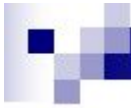




# Técnicas de Programação e Algoritmos - TPA

– Vetor



## Introdução

Vetores e Matrizes em geral são caracterizadas por se tratarem de uma única variável de um determinado tamanho que armazena várias informações do mesmo tipo.

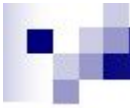
Essas informações são gravadas na memória sequencialmente e são referenciadas através de índices.

Em vez de declarar 100 variáveis para armazenar por exemplo o nome de 100 alunos, utilizando vetor, basta declarar apenas uma única variável de tamanho 100. Ou seja, em cada posição do vetor será armazenado o nome de cada aluno.

Vetores – Unidimensionais

Matrizes – Multidimensionais

São muito utilizados por exemplo, em desenvolvimento de jogos, análise de conjunto de dados, desenhos, cálculos matemáticos entre outros.



## Vetor

É uma variável que possui apenas uma única dimensão.

A declaração de vetores em java deve obedecer a seguinte sintaxe:

```
Tipo[ ] nome_variável = new Tipo[tamanho];
```

O Tipo deve ser especificado de acordo com o tipo de informação que será armazenada no vetor (ex. int float, char, String...).

E o tamanho representa a quantidade máxima de elementos que podem ser armazenados neste vetor.

É importante dizer que os vetores começam pelo índice 0 (zero) que guarda o primeiro elemento do vetor. Posição 1 equivale ao índice 0.

## Vetor – Exemplo de Declaração

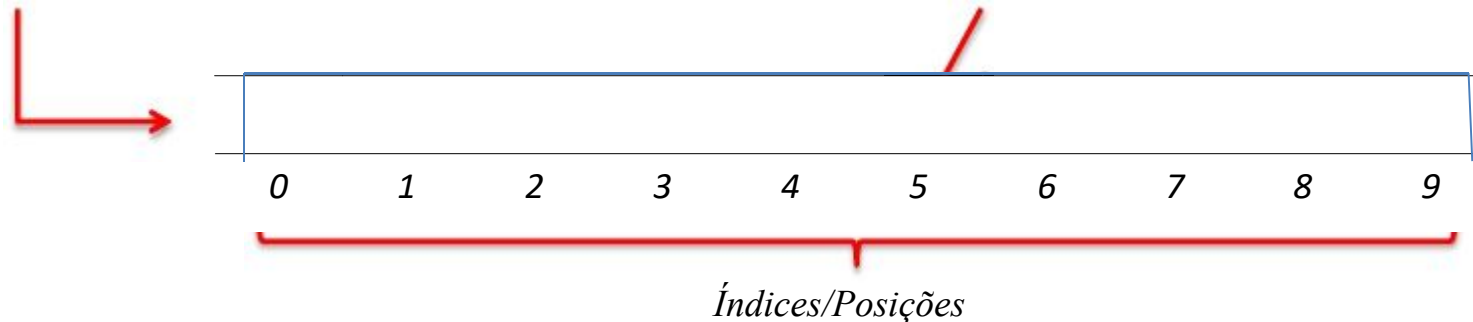
Para entender melhor, considere que seja necessário declarar um vetor do tipo inteiro que contenha 10 elementos. Isto é feito da seguinte forma:

```
int[ ] vetor_exemplo = new int[10];
```

A variável **vetor\_exemplo** possui 10 posições, sendo que seus índices vão de 0 a 9.

*Nome da variável*

*Conteúdo da posição (na caso está vazia)*



## Exemplo 1 – Sem uso de Vetor

Neste exemplo, o programa irá armazenar o nome de 10 pessoas e depois exibi-los na tela.

Observem que foram necessárias a declaração de 10 variáveis, onde cada uma armazenará o nome de uma pessoa.

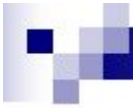
```
Exemplo_1A.java
1 import javax.swing.JOptionPane;
2
3 public class Exemplo_1A {
4
5     public static void main(String[] args) {
6         // criando variáveis
7         String n1,n2,n3,n4,n5,n6,n7,n8,n9,n10;
8         // recebendo valores para cada variável criada
9         n1=JOptionPane.showInputDialog("Digite o 1º nome:: ");
10        n2=JOptionPane.showInputDialog("Digite o 2º nome:: ");
11        n3=JOptionPane.showInputDialog("Digite o 3º nome:: ");
12        n4=JOptionPane.showInputDialog("Digite o 4º nome:: ");
13        n5=JOptionPane.showInputDialog("Digite o 5º nome:: ");
14        n6=JOptionPane.showInputDialog("Digite o 6º nome:: ");
15        n7=JOptionPane.showInputDialog("Digite o 7º nome:: ");
16        n8=JOptionPane.showInputDialog("Digite o 8º nome:: ");
17        n9=JOptionPane.showInputDialog("Digite o 9º nome:: ");
18        n10=JOptionPane.showInputDialog("Digite o 10º nome:: ");
19        // apresentando os valores digitados em cada variável
20        System.out.println("1º nome:: "+ n1);
21        System.out.println("2º nome:: "+ n2);
22        System.out.println("3º nome:: "+ n3);
23        System.out.println("4º nome:: "+ n4);
24        System.out.println("5º nome:: "+ n5);
25        System.out.println("6º nome:: "+ n6);
26        System.out.println("7º nome:: "+ n7);
27        System.out.println("8º nome:: "+ n8);
28        System.out.println("9º nome:: "+ n9);
29        System.out.println("10º nome:: "+ n10);
30    }
31
32 }
33
```

## Exemplo 1 – Sem uso de Vetor - Resultado

Foram digitados os 10 nomes e na sequência foram exibidos na tela, todos os nomes na ordem em que foram digitados.



```
Saída - JavaApplication2 (run) ✖  
run:  
1º nome:: Maria  
2º nome:: Pedro  
3º nome:: Antonio  
4º nome:: Cássio  
5º nome:: Joana  
6º nome:: Luciana  
7º nome:: Sérgio  
8º nome:: Paulo  
9º nome:: Edson  
10º nome:: Welligton  
CONSTRUÍDO COM SUCESSO
```



## Exemplo 2 – Com uso de Vetor

Com base no exemplo anterior, o propósito é o mesmo, o programa irá armazenar o nome de 10 pessoas e depois exibi-los na tela.

Observem que foi necessária a declaração de apenas 1 variável para armazenar os 10 nomes e uma outra do tipo *int* para referenciar o índice no vetor.

- No próximo slide a apresentação do código fonte.

```
1  import javax.swing.JOptionPane;
2
3  public class Exemplo_1B {
4
5      public static void main(String[] args) {
6          // criação da variável nomes com 10 posições ( 0 a 9)
7          String[] nomes = new String[10];
8          // criação da variável de controle do for
9          int i;
10
11         for(i=0;i<10;i++)
12         {
13             // recebendo valores para cada variável criada
14             nomes[i]=JOptionPane.showInputDialog("Digite o "+(i+1)+"º nome:: ");
15         }
16
17         for(i=0;i<10;i++)
18         {
19             // apresentando os valores digitados em cada variável
20             System.out.println((i+1)+"º nome:: "+ nomes[i]);
21         }
22     }
23
24
25
26 }
27
```



## Exemplo 2 – Com uso de Vetor - Resultado

Foram digitados os 10 nomes e na sequência foram exibidos na tela, todos os nomes na ordem em que foram digitados.

Se compararmos com o resultado do exemplo 1, veremos que o formato de saída são idênticos.



```
run:
1º nome:: Maria
2º nome:: Pedro
3º nome:: Antonio
4º nome:: Cássio
5º nome:: Joana
6º nome:: Luciana
7º nome:: Sérgio
8º nome:: Paulo
9º nome:: Edson
10º nome:: Welligton
CONSTRUÍDO COM SUCESSO
```

## Exemplo 2 – Com uso de Vetor – Analisando o 1º Bloco

Imediatamente após a declaração das variáveis, este bloco permite que sejam armazenados os 10 nomes de pessoas.

A estrutura de repetição **for** é responsável por repetir as instruções por 10 vezes, onde *i* é a variável de controle deste bloco e será incrementada até chegar em 10, ou seja, até que a condição seja falsa.

Na saída irá aparecer o resultado da soma de *i*+1, ou seja, a primeira vez *i* vale **0**, então somando 0 e 1 aparecerá **1** na tela. “*Digite o 1º nome:* “. O usuário digitará o primeiro nome que será armazenado na posição **0** do vetor.

No segundo momento, após *i* ser incrementada, a mesma passa a valer **1**, aparecendo **2** na tela “*Digite o 2º nome:* “. O usuário digitará o segundo nome que será armazenado na posição **1** do vetor. E assim sucessivamente:

```
11  
12  
13  
14  
15  
16  
for(i=0;i<10;i++)  
{  
    // recebendo valores para cada variável criada  
    nomes[i]=JOptionPane.showInputDialog("Digite o "+(i+1)+"º nome:: ");  
}
```

## Exemplo 2 – Com uso de Vetor – Analisando o 2º Bloco

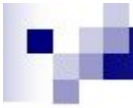
Depois da execução do 1º Bloco, que foi responsável por armazenar os nomes no vetor, este 2º Bloco é responsável por exibir os dados que estão armazenados no vetor.

Na saída (i+1), segue o mesmo adotado no 1º Bloco, apenas para identificar ao usuário uma sequencia.

Na saída [i], será exibido primeiramente o nome que está armazenado no vetor na posição 0, pois i inicia com valor 0. Em uma segunda iteração, será exibido o nome que está na posição 1 do vetor e assim sucessivamente.

```
17  
18  
19  
20  
21
```

```
for(i=0;i<10;i++)  
{  
    // apresentando os valores digitados em cada variável  
    System.out.println((i+1)+"º nome:: "+ nomes[i]);  
}
```



## Com ou Sem uso de Vetor? - Comparativos

Analizando as modificações necessárias para tal adaptação, o uso de vetor se faz necessário para uma melhor codificação, onde no desenvolvimento não precisou realizar grandes modificações. Sem o uso de vetor seriam necessárias por volta 270 linhas de código a mais para adaptar ao novo cenário, o que não é viável.

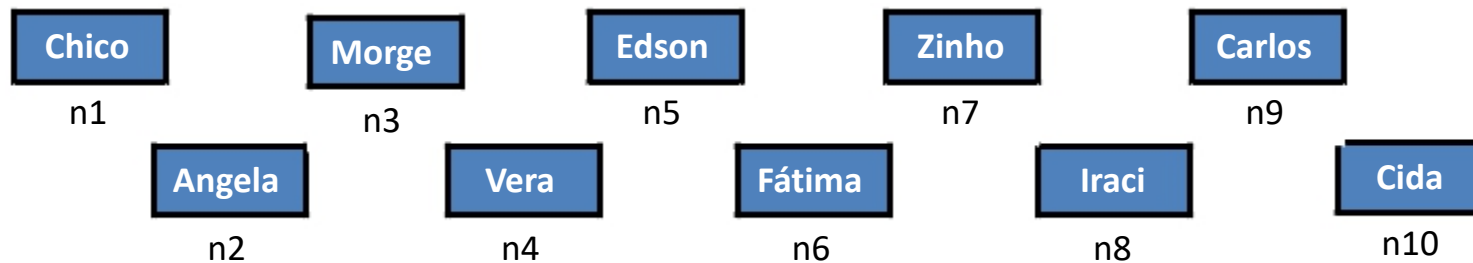
Além do mais, se o cliente ainda quiser solicitar mais adaptações, como por exemplo, saber qual é o nome que possui mais caracteres, o nome que possui menos caracteres, quantidade de nomes que iniciam com vogal ou mesmo para fazer uma simples pesquisa para saber se existe um nome cadastrado.

Então, se não usarmos vetores, as implementações sugeridas pelo cliente com certeza não atenderão as expectativas.

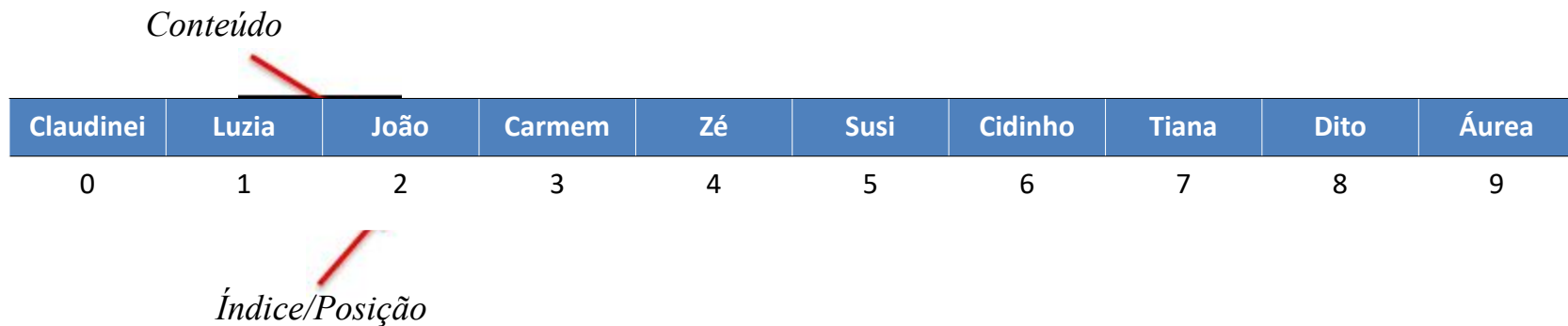
Neste caso e outros que veremos, é necessário o uso de vetores.

## Diferença no Armazenamento

**Sem uso de Vetor:** os valores são armazenados em variáveis independentes.



**Com uso de Vetor:** os valores são armazenados em uma única variável.



## Exemplo 3 – Transformando string em vetor de char

Neste exemplo será apresentado um método que transforma/converte uma string em um vetor de caracteres. O usuário digitará uma palavra e serão exibidas apenas as vogais desta palavra.

```
3  import javax.swing.JOptionPane;
4
5  public class Testes {
6
7      public static void main(String[] args)
8      {
9          String palavra;
10         int i;
11         // recebendo valor para variável criada
12         palavra=JOptionPane.showInputDialog("Digite uma palavra:: ");
13         System.out.println("\n vogais da palavra: \t"+ palavra+"\n");
14         // a função toCharArray() separa cada caracter da palavra
15         char[] vetor=palavra.toCharArray();
16         int tamanho = vetor.length;
17
18         for(i=0;i<tamanho;i++)
19         {
20             // apresentando as vogais da palavra digitada, verifica se a letra do momento é uma vogal
21             if(vetor[i]=='a' || vetor[i]=='e' || vetor[i]=='i' || vetor[i]=='o' || vetor[i]=='u')
22             {
23                 System.out.println( vetor[i]+" ");
24             }
25         }
26     }
27 }
```

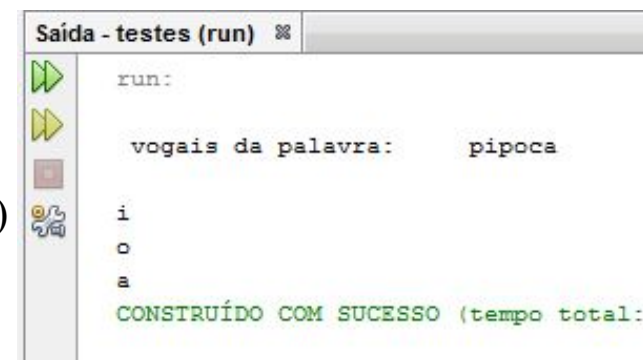
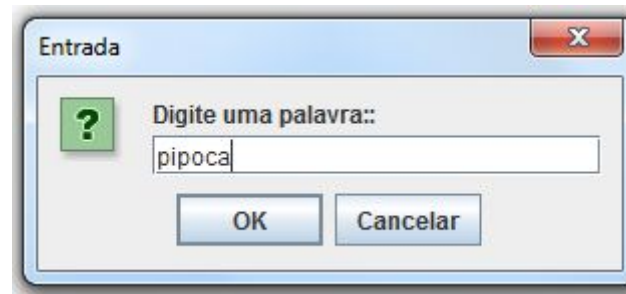
## Exemplo 3 – Transformando string em vetor de char - Resultado

O resultado apresentado exibiu apenas as vogais da palavra digitada pelo usuário.

Neste exemplo somente foram tratadas as vogais minúsculas, mas poderiam ser realizados outros testes como comparar maiúsculas e também vogais com acentuação.

A instrução: **char[] vetor = palavra.ToCharArray();** (linha 15) realiza a transformação de uma string em um vetor de caracteres, pois para realizar o teste para saber se cada elemento do vetor é uma vogal, deve ser comparado caractere por caractere.

A instrução: **int tamanho = vetor.length;** (linha 16) realiza a verificação de quantos caracteres existem no vetor e armazena esta quantidade na variável **tamanho**, para que possa ser usada como limite para o número de iterações na estrutura de repetição **for**.



## Exemplo 4 – De 1 a 10 Salários Mínimos

Neste exemplo serão apresentados os valores correspondentes a quantidade de 1 a 10 salários mínimos.

```
2
3 import javax.swing.JOptionPane;
4
5 public class Testes {
6
7     public static void main(String[] args)
8     {
9         int i;
10        double sal_minimo=1000.00;
11        double[] salarios = new double[10];
12
13        //atribuição do cálculo no vetor salarios
14        for(i=0;i<salarios.length;i++)
15        {
16            salarios[i] = (i+1)*sal_minimo;
17        }
18
19        //Exibição dos dados calculados no vetor salarios
20        for(i=0;i<salarios.length;i++)
21        {
22            System.out.println((i+1)+" Salário(s) Mínimo(s) = "+salarios[i)+"\n");
23        }
24    }
25 }
```



## Exemplo 4 – De 1 a 10 Salários Mínimos – Resultado

Saída::



```
run:
1 Salário(s) Mínimo(s) = 1000.0
2 Salário(s) Mínimo(s) = 2000.0
3 Salário(s) Mínimo(s) = 3000.0
4 Salário(s) Mínimo(s) = 4000.0
5 Salário(s) Mínimo(s) = 5000.0
6 Salário(s) Mínimo(s) = 6000.0
7 Salário(s) Mínimo(s) = 7000.0
8 Salário(s) Mínimo(s) = 8000.0
9 Salário(s) Mínimo(s) = 9000.0
10 Salário(s) Mínimo(s) = 10000.0

CONSTRUÍDO COM SUCESSO (tempo total:
```