

Agenda 07: Vetores

Conceitos trabalhados:

Sumário

- Momento de Reflexão
- Por que Aprender?
- Para Começar o Assunto
- Mergulhando no Tema
- Ampliando Horizontes
- Resumindo o Estudo

Momento de Reflexão

Olá, estudante :)

Boas-vindas à sétima agenda do Módulo 1! Após estudar as principais estruturas de todas as linguagens de programação, terá contato com um novo conceito: **vetores**, um tipo especial de variáveis.

Para começar, que tal conhecer uma nova analogia gastronômica? Você certamente já experimentou **chocolate** não é mesmo? Mais do que isso, se tiver o mesmo apreço pelo doce como a maioria das pessoas têm, conseguirá lembrar de marcas, formatos e sabores dessa guloseima.



Entre os formatos de chocolate, o mais conhecido é a barra formada por quadradinhos.



Então, quando pensamos em uma barra, imaginamos que ela é composta por pedaços menores.



Esses pedaços podem ter recheios, de um único sabor, na mesma barra.

Antes que você corra para o mercado mais próximo para satisfazer seu desejo pelo doce, vamos entender a relação da barra de chocolate com vetores?

Assim como uma única barra de chocolate armazena quantidades de recheio em unidades menores, **um único vetor é capaz de armazenar diversos valores em suas casas**. Compramos barras de chocolate em lugar de bombons individuais pela praticidade: adquirimos uma quantidade maior do produto levando um único item (uma barra). Além disso, a barra costuma ser mais econômica que o bombom individual. Na programação, utilizamos vetores em lugar de variáveis convencionais também pela praticidade: **isso facilita o acesso aos dados e economiza espaço na memória do computador**. Deste modo, o processo de leitura e gravação de um dado em uma variável é simplificado graças às estruturas de repetição.

Nesta agenda, você verá como essa simplificação é feita. Vamos lá?

Por que Aprender?

Até este momento, você trabalhou com variáveis simples, que só recebem um dado por vez. Para atender às exigências da programação, você aprenderá a lidar com uma técnica de programação que permite trabalhar com um **agrupamento de várias informações** dentro de uma mesma variável, tornando a programação muito **mais rápida e eficiente**: os vetores.

Para Começar o Assunto



“Era uma pessoa igual a cem mil outras pessoas. Mas, eu fiz dela um amigo, agora ela é única no mundo”

O pequeno príncipe

Jonas, um adepto da tecnologia e estudante de Lógica de Programação, recebeu uma importante tarefa de seus pais: criar um pequeno programa para homenagear os amigos de longa data de seu avô, Marco, na festa de 80 anos que aconteceria na semana seguinte.

Para isso, os pais de Jonas devem conseguir inserir os nomes dos cinco amigos a serem homenageados no programa, que deverá gerar uma pequena mensagem.

Mas, confeccionando o programa, Jonas percebeu que o que parecia simples se tornou muito extenso: como cada nome estava em uma variável, seria impossível utilizar uma estrutura de repetição para satisfazer o problema...

Você consegue imaginar como Jonas poderá desenvolver o programa para que ele fique pronto a tempo e de forma eficiente? Sim, vetores podem ajudar a resolver essa questão! Explore os materiais a seguir para entender melhor de que maneira.

Mergulhando no Tema

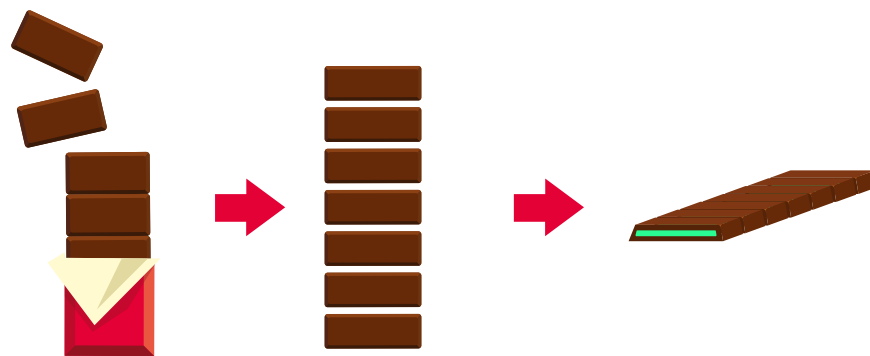
O que são vetores?

Durante seus estudos em Lógica de Programação, você já passou por alguma situação em que fosse necessária a declaração de muitas variáveis para o mesmo propósito? Alguma vez você já se perguntou se para calcular a média entre 10 notas de um aluno seria necessário declarar 10 variáveis?

Ao longo das agendas anteriores, você trabalhou apenas com variáveis simples, que só recebem um dado por vez. Mas há uma técnica de programação que permite o **agrupamento de várias informações** dentro de uma mesma variável, fazendo com que a programação seja mais rápida e eficiente: são os **vetores**.



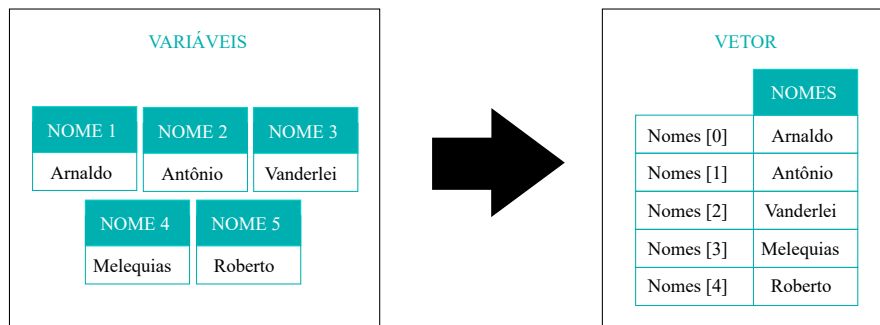
Para fazermos uma analogia gastronômica, vetores são como barras de chocolate: uma barra é composta por diversos quadradinhos, que podem armazenar recheios, desde que sejam todos do mesmo sabor.



Assim como uma única barra de chocolate armazena quantidades de recheio em unidades menores, **um único vetor é capaz de armazenar diversos valores em suas casas**. Compramos barras de chocolate em vez de bombons individuais pela praticidade: adquirimos uma quantidade maior do produto levando em um único item (uma barra). Além disso, a barra costuma ser mais econômica que o bombom individual. Na programação, utilizamos vetores em vez de variáveis convencionais também pela praticidade: **isso facilita o acesso aos dados e economiza espaço na memória do computador**.

Vetores são variáveis especiais que têm a capacidade de receber diversos valores, desde que todos sejam do mesmo tipo (int, double, String etc). São também conhecidos como **Matrizes Unidimensionais**.

Vetores foram desenvolvidos para que leituras e gravações de dados repetitivos sejam simplificados por meio das **estruturas de repetição**, que você conheceu nas agendas anteriores. Uma vez que valores individuais devem sempre ser lidos individualmente, não é possível ler N variáveis dentro de uma única estrutura de repetição, então utilizamos vetores.



Agora que você sabe que vetores são variáveis que podem receber mais de um valor, pense e responda: em quais casos já estudados podemos substituir variáveis repetidas por vetores?

Refleta sobre este novo conceito antes de prosseguir com a leitura.

Conhecendo a estrutura do vetor

Para entender melhor a importância desse tipo de variável, vamos tentar resolver a seguinte situação-problema:



Jennifer, uma professora do curso técnico de Eventos e formada em Programação de Computadores, decidiu criar um programa que confeccione automaticamente uma lista de convidados para o próximo evento da turma do 1º módulo do curso. Para testar se o programa funcionará corretamente, Jennifer deseja primeiramente cadastrar apenas 10 convidados e exibi-los na mesma ordem em que foram cadastrados.

Para isso, criou o seguinte programa em Java:

```
import javax.swing.JOptionPane;

public class Principal {

    public static void main(String[] args) {
        String convidado1 = "";
        String convidado2 = "";
        String convidado3 = "";
        String convidado4 = "";
        String convidado5 = "";
        String convidado6 = "";
        String convidado7 = "";
        String convidado8 = "";
        String convidado9 = "";
        String convidado10 = "";
        String listafinal = "";
        convidado1 = JOptionPane.showInputDialog("Digite o nome do 1º Convidado");
        convidado2 = JOptionPane.showInputDialog("Digite o nome do 2º Convidado");
        convidado3 = JOptionPane.showInputDialog("Digite o nome do 3º Convidado");
        convidado4 = JOptionPane.showInputDialog("Digite o nome do 4º Convidado");
        convidado5 = JOptionPane.showInputDialog("Digite o nome do 5º Convidado");
        convidado6 = JOptionPane.showInputDialog("Digite o nome do 6º Convidado");
        convidado7 = JOptionPane.showInputDialog("Digite o nome do 7º Convidado");
        convidado8 = JOptionPane.showInputDialog("Digite o nome do 8º Convidado");
        convidado9 = JOptionPane.showInputDialog("Digite o nome do 9º Convidado");
        convidado10 = JOptionPane.showInputDialog("Digite o nome do 10º Convidado");
        listafinal = "Os convidados da festa são: " +
        convidado1 + ", " + convidado2 + ", " +
        convidado3 + ", " + convidado4 +
        ", " + convidado5 + ", " + convidado6 +
        ", " + convidado7 + ", " +
        convidado8 + ", " + convidado9 + ", " + convidado10;
        JOptionPane.showMessageDialog(null, listafinal);
    }
}
```


Apesar do programa funcionar sem problemas, a codificação ficou muito repetitiva, não acha?

Isso ocorre por que o programa está fazendo apenas uma função básica, sem muitos incrementos. Com vetores, podemos reorganizar esse código para utilizar **menos da metade** dos comandos, e com uma vantagem: não é necessário acrescentar nenhuma linha de código extra, caso seja necessário cadastrar 10, 100 ou até 1000 convidados.

Como trabalhar com um Vetor?

Em primeiro lugar, para trabalhar com um vetor, devemos **declará-lo**, assim como fazemos com uma variável. A diferença é que nessa declaração devemos especificar a quantidade de “**casas**” que o vetor terá.

Para o computador, cada casa é um espaço no qual poderá ser escrito um valor. A primeira casa do vetor sempre será a **casa de número zero**, devido à numeração binária utilizada em qualquer meio computacional, que sempre tem início em zero.

Atenção!

Assim como uma variável, você **nunca deve declarar um vetor sem necessidade**. Um problema comum para iniciantes é a declaração de mais casas no vetor do que realmente necessita, ou o **estouro do vetor**, que é a tentativa alocar mais valores no vetor do que ele suporta.

Declarando um vetor

Para declarar um vetor é muito simples, basta seguir o seu padrão conforme a tabela abaixo:

PSEUDOCÓDIGO

nomevariavel **como conjunto**[inicio..fim] **de** tipo tipo[] nomevariavel = **new** tipo[10]

JAVA

Repare que tanto o pseudocódigo quanto o fluxograma utilizam a mesma forma para declarar um vetor. Isso acontece porque ambas as notações utilizam a mesma linguagem como base: o Português Estruturado.

As diferenças entre uma variável comum e um vetor são:

- O vetor sempre deve estar acompanhado da “casa” em que o dado deve ser lido/gravado;
- A primeira casa do vetor sempre será nomeada como casa número 0.

Na prática, se declaramos um vetor de 10 casas em Java, ele alocará na memória as casas numeradas de 0 a 9. Veja o exemplo abaixo, parte da solução do programa de Jennifer:

```
System.out.println("O convidado é " + convidados [4]);
```

↓

Convidados[10]	
convidados[0]	Zenaide
convidados[1]	Julio
convidados[2]	Luci
convidados[3]	Melvin
convidados[4]	Marta
convidados[5]	Blaino
convidados[6]	Hilda
convidados[7]	Kirla
convidados[8]	Lourdes
convidados[9]	Clovis

→ O convidado é Marta

Na videoaula a seguir, o professor Rogério Silva apresenta a estrutura de vetores aplicada em Java.



Conhecendo alguns exemplos

Voltando ao exemplo, que tal verificar como ficará o problema da Jennifer utilizando vetores?

Seguindo as particularidades do algoritmo e da Linguagem de Programação Java, veja abaixo a solução para o problema da professora Jennifer utilizando vetor:

PSEUDOCÓDIGO

```

Programa convidaEvento
  Declare
    convidados como conjunto[1..10] de caractere
    contador como inteiro
    listafinal como caractere
  Inicio
    listafinal &0;- "Os convidados são: "
    Para contador &0;- 1 até 10 faça
      Escreva("Digite o convidado número ", contador, ": ")
      Leia(convidados[contador])
    Fim-Para
    Para contador &0;- 1 até 10 faça
      listafinal &0;- listafinal + convidados[contador]+ " "
    Fim-Para
    Escreva(listafinal)
  Fim
  
```

Em Java:

```

import javax.swing.JOptionPane;

public class PrincipalVetor {
    public static void main(String[] args) {
        String[] convidados = new String[10];
        int contador = 0;
        //Definir a frase que antecede o nome dos
        convidados
        String listafinal = "Os convidados da festa
        são: ";
        //Laço de repetição para ler cada um dos
        convidados.
        for (contador =0;contador < 10;contador ++){
            convidados[contador]=JOptionPane.showInputDialog("Digite o " +
            (contador+1) + " convidado");
            //Como o vetor inicia-se com zero,
            utilizamos contador+1 para identificar ao usuário qual
            //é o número do convidado no qual ele está
            digitando corretamente.
        }
        //Laço de repetição para alocar em uma variável
        o nome de cada um dos candidatos
        for(contador = 0; contador < 10; contador ++){
            //Alocação dos dados na variável,
            adicionando novos valores a cada passagem do laço.
            listafinal = listafinal +
            convidados[contador] + " ";
        }
    }
}
  
```

```
        }  
        //exibição dos dados armazenados em listafinal.  
        JOptionPane.showMessageDialog(null,  
listafinal);  
    }  
}
```

Você percebeu o quanto o código fonte **diminuiu de tamanho** após a aplicação do vetor? O uso do vetor pode ser um pouquinho mais trabalhoso no início, mas ele consegue realizar o trabalho de muitas variáveis utilizando apenas uma! Esse é um ótimo aliado de todo bom programador!

Como você já sabe, os conceitos que estudou nas agendas anteriores são fundamentais para que você consiga desenvolver programas cada vez mais complexos. Procure sempre utilizá-los conjuntamente. Na videoaula a seguir, o professor Sandro Valérius apresenta um exemplo de programa com vetores que também utiliza o laço de repetição for.

DS - Lógica - Vetores



Ampliando Horizontes

Onde encontro mais informações sobre vetores?

Para aprofundar o tema desta agenda, você pode consultar os materiais indicados a seguir, confira!

https://eadtec.cps.sp.gov.br/midiateca/arquivos/7_apostila_de_algoritmo_estruturado_saletebufoni.pdf

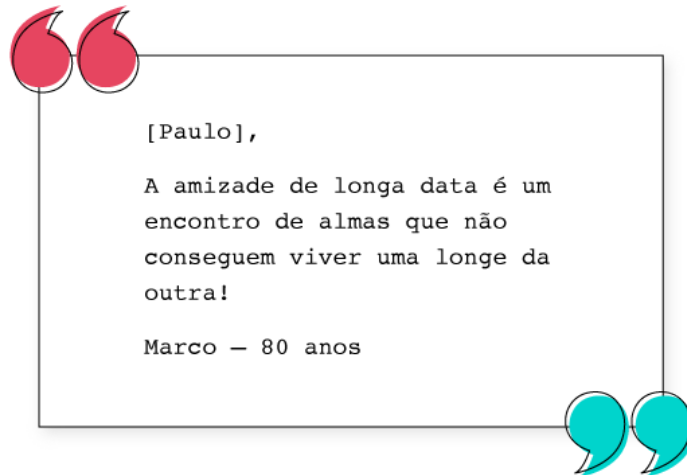
Essa apostila apresenta diversos conceitos que podem ser úteis durante seus estudos, mas para mais informações sobre vetores acesse o **Capítulo 5 – p. 51**.



Lógica de programação e estruturas de dados com aplicações em Java. PUGA, Sandra; RISSETTI, Gerson. Editora Pearson, 2009.

Resumindo o Estudo

Nesta agenda, você aprendeu que vetores foram desenvolvidos para que leituras e gravações de dados repetitivos sejam simplificados, facilitando a criação de programas como o desenvolvido por Jonas, para a festa de seu avô.



Como você já sabe, sempre **há mais de uma forma de se estruturar um programa ou código fonte**, de modo a obter o mesmo resultado ao final. Quando falamos de vetores, não é diferente, por isso nesta agenda você conheceu a forma básica dos vetores, abordando apenas uma dimensão. Mas, dependendo da circunstância, também é possível utilizar as **matrizes**, que são o tema da próxima agenda.

Até lá!

