

# Lógica de Programação

## Aula 13 - Estruturas de dados homogêneas – Matrizes



## Apresentação da Aula

---

Você lembra que as estruturas de dados homogêneas são classificadas em dois tipos? Em **vetores** (também conhecidos como *arrays*) e em **matrizes**. O primeiro tipo é também conhecido como **estruturas unidimensional**, e o segundo tipo, as **matrizes**, é, no mínimo, **bidimensional**. Além disso, você viu que os vetores são variáveis que utilizam índices para permitir o acesso aos seus espaços de armazenamento e aprendeu como declará-los e utilizá-los.

Nesta aula, você continuará a conhecer as **estruturas de dados homogêneas**, mas, agora, trabalhará especificamente com as **matrizes**, que, como já dito, são estruturas com capacidade **bidimensional** ou **n-dimensional**.



### Objetivos

- Conhecer o conceito de estruturas de dados homogêneas **bidimensionais**.
- Explorar o conceito e a sintaxe das **matrizes**.
- Construir algoritmos por meio das **matrizes**.

# Classificação das Estruturas de Dados Homogêneas

---

Em aulas anteriores, você viu o que são as estruturas de dados homogêneas e teve a oportunidade de conhecer as estruturas homogêneas **unidimensionais**. Além disso, observou que essas estruturas são variáveis com diversos espaços de armazenamento e que, para acessar cada um desses espaços, é necessário utilizar um índice.

Os vetores (*arrays*) são estruturas **unidimensionais**, pois possuem apenas uma dimensão. Em outras palavras, para acessar os espaços de armazenamento dessa estrutura, é necessário informar somente um índice.

Já as **matrizes** são estruturas também **homogêneas**, porém, elas possuem duas dimensões ou mais. Agora, você irá conhecer e trabalhar especificamente com as **matrizes bidimensionais**, isto é, com duas dimensões.

As **matrizes bidimensionais** podem ser representadas por meio de tabelas. Você recordar-se das tabelas estudadas na disciplina de Introdução às Tecnologias da Informação? Em uma tabela, para cada espaço destinado ao armazenamento de informação (células), são necessárias duas informações para determinar o seu endereço. No caso das planilhas, uma das informações diz respeito à definição da coluna, geralmente representada por letras. As linhas, por sua vez, são definidas com base em números. Observe a Figura 01, na qual é apresentada uma tabela com 10 linhas e 10 colunas.

### Figura 01 - Exemplo de Tabela

[illegible]

A Figura 02 apresenta a mesma tabela da Figura 01, mas com alguns dos seus espaços ocupados. Por exemplo, na primeira linha, temos os espaços das colunas **A**, **B** e **C**, que estão ocupados. Já na sexta linha, temos o espaço da coluna **E** e, por último, na linha 10, nós temos o espaço da coluna **H** ocupado. Para que seja possível determinar qual desses espaços se desejar acessar, é necessário sempre informar qual é a coluna e qual é a linha pretendida.

**Figura 02 - Exemplo de Tabela com conteúdos**

[illegible]

Nas **matrizes**, como estrutura de dados homogênea **bidimensional**, das linguagens de programação, geralmente, as informações que definem o endereço de cada espaço da tabela são índices numéricos, apesar que, há linguagens que permitem a definição de índices a partir de outros tipos de dados. No Potigol, ambos os índices serão sempre valores inteiros.

Como explicado, uma matriz **bidimensional** pode ser representada por meio de uma tabela. A Figura 03 apresenta uma matriz, cujos índices são números inteiros que estão informados acima e à esquerda dessa tabela.

Na **matriz**, observe que, internamente, em cada um dos espaços (quadrados), há dois números: o primeiro deles representa o número da linha, e o segundo representa o número da coluna. Você lembra que, na aula sobre vetores, foi dito que os índices de um vetor, na linguagem **Potigol**, são iniciados a partir do valor 0 (zero)? O mesmo vale para as **matrizes**, cujo primeiro índice (tanto para a linha como para as colunas) será sempre zero. Neste exemplo, nós temos uma matriz de cinco colunas e três linhas.

**Figura 03** - Exemplo de Matriz Bidimensional 3x5

	0	1	2	3	4
0	[0,0]	[0,1]	[0,2]	[0,3]	[0,4]
1	[1,0]	[1,1]	[1,2]	[1,3]	[1,4]
2	[2,0]	[2,1]	[2,2]	[2,3]	[2,4]

## Sintaxe de Matrizes

---

Lembra-se da sintaxe para se declarar uma variável homogênea **unidimensional** do tipo texto (vetor)? Caso você não esteja se recordando, o trecho de código abaixo reapresenta como realizar esse procedimento. No exemplo, define-se uma variável homogênea denominada **alunos**, cujo tipo é texto e com 10 espaços de armazenamento (índices de 0 até 9).

```
1 var alunos := vetor[10]
```

Agora, observe como realizar a declaração de uma variável **bidimensional**, também do tipo **texto**, porém, com 5 linhas e 10 colunas. Observe que o comando que define uma matriz é semelhante ao de declaração de um vetor. A diferença refere-se ao par de números separados por vírgula, pois, para a **matriz**, são necessários dois valores inteiros a fim de indicar a quantidade de linhas e colunas.

```
1 var matriz_alunos := matriz[5,10]
```

Lembre-se de que o índice utilizado entre os colchetes de um vetor indica a posição que se deseja acessar na variável. O mesmo vale para as **matrizes bidimensionais**, com a diferença de que, no caso delas, será necessário informar dois índices: um para a linha e outro para a coluna. Veja como ficaria a sintaxe para acessar a terceira linha e a segunda coluna do vetor **matriz\_aluno**, armazenar o nome de um aluno e, em seguida, escrever na tela o nome armazenado.

```
1 var matriz_alunos := matriz[5,10]
2 # Armazenando um nome de aluno.
3 matriz_alunos[2][1] := "Uriel Lira Lopes"
4 escreva "O nome do aluno é: {matriz_alunos[2][1]}."
5
```

Você recorda que, nos vetores, é possível substituir o índice numérico por uma variável primitiva do tipo **inteiro**? Assim, é possível utilizar as estruturas de repetição para simplificar a manipulação dos dados de um vetor. A mesma solução vale para as **matrizes**, no entanto, para acessar todas as posições (espaços da matriz), será necessário utilizar duas estruturas de repetição, pois é imprescindível uma variável de controle para indicar o índice das linhas e outra variável para indicar o índice das colunas.

No exemplo a seguir, considere que cada linha da matriz representa um aluno e que cada coluna representa a nota de uma disciplina. Desse modo, cada linha possui 3 colunas que representarão as notas de 3 disciplinas.

```

1 var notas_de_3disciplinas := matriz[5,3]
2
3 # ler notas de 3 disciplinas para 5 alunos
4 para linha de 0 até 4 faça
5   para coluna de 0 até 2 faça
6     escreva "Para o aluno {linha+1} informe a nota da Disciplina {coluna+1}:"
7     notas_de_3disciplinas[linha][coluna] := leia_inteiro
8   fim
9 fim
10
11 # imprimir as notas de 3 disciplinas de 5 alunos
12 para linha de 0 até 4 faça
13   para coluna de 0 até 2 faça
14     escreva "Aluno {linha+1} - Disciplina {coluna+1} - Nota {notas_de_3disciplinas[linha][coluna]}"
15   fim
16 fim
17

```

A Figura 04 representa graficamente a matriz **notas\_de\_3disciplinas[][]**. Observe que, na lateral esquerda, temos os valores de L1, L2, L3, L4 e L5, que estão associados às linhas de 1 até 5, e, imediatamente ao lado, estão os respectivos índices, que vão de 0 (zero) até 4 e são utilizados para acessar uma dessas linhas. Por exemplo, se você quiser acessar a última linha (L5), será necessário utilizar o índice 4 dentro do primeiro colchete da variável **notas\_de\_3disciplinas[4][ ]**.

Já na parte superior, há os identificadores C1, C2 e C3, que representam as colunas 1, 2 e 3. Logo abaixo desses identificadores, estão os índices de cada coluna, relativos aos valores de 0 (zero) até 2. Para indicar qual das colunas se deseja acessar, é necessário inserir o respectivo índice no segundo colchete da variável **notas\_de\_3disciplinas[ ][ ]**. É importante destacar que, ao acessar qualquer posição em uma matriz, você terá de inserir sempre ambos os índices (o de linha e o de coluna) para poder indicar corretamente qual posição se deseja acessar nela.

**Figura 04** - Matriz e índices

		C1	C2	C3
		0	1	2
L1	0			
L2	1			
L3	2			
L4	3			
L5	4			

## Uso de Matrizes

---

As **matrizes** são utilizadas, geralmente, para o armazenamento de dados que necessitam ser endereçados por mais de um índice. Um bom exemplo em que há o uso de **matrizes** são as imagens digitais que você vê no computador, no *tablet* ou no celular. Uma imagem é formada por centenas de pontos de informação, um ao lado do outro, e cada um desses pontos de informação possui um par de coordenadas (que determina as linhas e colunas) na **matriz**. A informação armazenada em cada um desses pontos, no caso das imagens, é a cor que o ponto deve possuir.

Muitos outros problemas computacionais utilizam as **matrizes** para representar uma estrutura multidimensional de dados. A exemplo de tais problemas, temos jogos (um tabuleiro de xadrez), gráficos, sistemas de equações (construção de



**matrizes**, álgebra linear), além de diversas outras situações.



## Atividade 01

### Preenchendo e imprimindo o conteúdo de uma matriz

Construa um algoritmo que receberá os elementos para preencher uma **matriz** de 5 linhas e 10 colunas. O preenchimento da **matriz** ocorrerá da esquerda para a direita, de cima para baixo.

Após preencher toda a **matriz**, o seu algoritmo deverá imprimi-la no formato de 5 linhas e 10 colunas. Os números de uma mesma linha devem ser separados por um único espaço em branco entre eles.



## Atividade 02

### Preenchendo e imprimindo o conteúdo de uma matriz

2

Construa um algoritmo que receberá os elementos para preencher uma **matriz** de 5 linhas e 10 colunas. O preenchimento da **matriz** ocorrerá da esquerda para a direita, de cima para baixo.

Após preencher toda a **matriz**, o seu algoritmo deverá imprimi-la no formato de 5 linhas e 10 colunas. Os números de uma mesma linha devem ser separados por um único espaço em branco entre eles.

Em seguida, você deverá imprimir novamente a mesma **matriz**, no entanto, deverá invertê-la e realizar a impressão no formato de 10 linhas e 5 colunas. Para isso, você precisará considerar que cada linha será uma coluna e que cada coluna será uma linha. Lembre-se de que os números de uma mesma coluna (que agora serão impressos como linhas) devem ser separados por um único espaço em branco entre eles.



## Atividade 03

### Somando em uma matriz

Construa um algoritmo que receberá um número inteiro entre 1 e 10 e que, em seguida, receberá os elementos para preencher uma **matriz** de 5 linhas e 10 colunas. O preenchimento da **matriz** ocorrerá da esquerda para a direita, de cima para baixo.

Após preencher toda a **matriz**, o seu algoritmo realizará a soma de todos os elementos presentes na coluna informada (o valor, entre 1 e 10, informado no início do programa).



## Resumo

---

Nesta aula, você exercitou o uso das estruturas de dados **homogêneas** e trabalhou com as estruturas **bidimensionais**, construindo algoritmos e utilizando essas estruturas associadas ao uso das **estruturas de repetição**. Caso você tenha encontrado alguma dificuldade, converse com seu mediador para poder sanar o quanto antes as suas dúvidas. O completo entendimento das estruturas homogêneas **unidimensionais** e **bidimensionais** é importante para a construção dos seus próximos conhecimentos sobre a construção de algoritmos.



## Referências

---

**Linguagem Potigol: Programação para todos.** Disponível em: <<http://potigol.github.io/>>. Acesso em: 07 jul. 2018.