

Capítulo 9 - Vetores e Matrizes (Arrays)

Neste capítulo vamos falar sobre Vetores e Matrizes. Você vai aprender para que serve, como usar e, claro, fazer exercícios para fixar o aprendizado. Ao final deste capítulo você estará craque nesta estrutura de dados tão usada na programação.

O que são Vetores e Matrizes

Vetores e **Matrizes** são estruturas de dados bastante simples que podem nos ajudar muito quando temos um grande número de variáveis do mesmo tipo em um algoritmo.

Imagine o seguinte problema: Você precisa criar um algoritmo que lê o nome e as 4 notas de 50 alunos, calcular a média de cada aluno e informar quais foram aprovados e quais foram reprovados. Conseguiu imaginar quantas variáveis você vai precisar pra fazer este algoritmo?

Muitas né?

Vamos fazer uma continha rápida aqui: são 50 variáveis para armazenar os nomes dos alunos, 200 variáveis para armazenar as 4 notas de cada aluno (4 * 50) e por fim, 50 variáveis para armazenar as médias de cada aluno.

São 300 variáveis no total, sem contar a quantidade de linhas de código que você vai precisar para ler todos os dados, calcular as médias de cada aluno e apresentar todos os resultados.

Mas eu tenho uma boa notícia pra você! Nós não precisamos criar 300 variáveis! Podemos utilizar **Vetores** e **Matrizes** (também conhecidos como **ARRAYs**)!

Tá bom ... Mas o que são esses tais vetores e matrizes?

Vetor (**array** uni-dimensional) é uma variável que armazena várias variáveis do mesmo tipo. No problema apresentado anteriormente, nós podemos utilizar um vetor de 50 posições para armazenar os nomes dos 50 alunos.

Matriz (**array** multi-dimensional) é um **vetor** de **vetores**. No nosso problema, imagine uma matriz para armazenar as 4 notas de cada um dos 50 alunos. Ou seja, um vetor de 50 posições, e em cada posição do vetor, há outro vetor com 4 posições. Isso é uma matriz!

Cada item do vetor (ou matriz) é acessado por um número chamado de **índice**. Ou **index** em inglês.

Uma bela forma de pensar software é pensar graficamente... Então vamos imaginar no nosso exemplo dos nomes, notas e médias dos 50 alunos como seriam os vetores e matrizes graficamente para facilitar o entendimento do conceito.

**Dicas de Programação**

Vetor de nomes dos alunos

1	2	3	...	49	50
João	Pedro	Carlos	...	José	Maria

Matriz das notas dos alunos

	1	2	3	4
1	9,5	10	8	7,5
2	10	9	9	5,5
3	9	8,5	9,5	7
...
49	7	10	10	9
50	7	8,5	5,5	4

Podemos ver na imagem acima que cada posição do vetor é identificado por um número (chamado de **índice**), no caso da matriz são dois números (um na vertical e um na horizontal).

Claro que também pode existir matrizes com mais de duas dimensões, mas não precisa se prender a estes detalhes agora. ;)

No Visualg os vetores são declarados da seguinte maneira:

< nome da variável> vetor [1..<tamanho>] de <tipo de dados>

Exemplo:

nomesDosAlunos vetor [1..50] de caractere

E as matrizes assim:

< nome da variável> vetor [1..<tamanho 1>,1..<tamanho 2>] de <tipo de dados>

Exemplo:

notas vetor [1..50,1..4] de real

Pronto, agora você já sabe o que são **arrays**. Então vamos ver como implementá-los em um algoritmo.

Vetores e Matrizes na prática!

Nada como praticar para fixar um aprendizado, concorda?

Continuando com o nosso exemplo, vamos implementar um algoritmo para o cálculo das médias.

Neste algoritmo vamos usar algumas estruturas básicas já apresentadas nas lições anteriores, tais como a estrutura de repetição PARA (capítulo anterior) e a estrutura de decisão SE-ENTÃO-SENÃO (capítulo #5).

OBS: Neste exemplo vamos reduzir o número de alunos de 50 para 5, para facilitar a visualização do resultado.

Preste muita atenção no modo como é criado o Vetor e a Matriz e também a forma como cada posição é acessada, utilizando os contadores.

```
algoritmo "MediaDe5Alunos"

var

    nomes: vetor [1..5] de caractere
    notas: vetor [1..5,1..4] de real
    medias: vetor [1..5] de real
    contadorLoop1, contadorLoop2: inteiro

inicio

    //Leitura dos nomes e as notas de cada aluno
    PARA contadorLoop1 DE 1 ATE 5 FACA
        ESCREVA("Digite o nome do aluno(a) número ", contadorLoop1, " de 5: ")
        LEIA(nomes[contadorLoop1])
        PARA contadorLoop2 DE 1 ATE 4 FACA
            ESCREVA("Digite a nota ", contadorLoop2, " do aluno(a) ",
nomes[contadorLoop1], ": ")
            LEIA(notas[contadorLoop1, contadorLoop2])
        FIMPARA
    //CÁLCULO DAS MÉDIAS
    medias[contadorLoop1] := (notas[contadorLoop1, 1] + notas[contadorLoop1,
2] + notas[contadorLoop1, 3] + notas[contadorLoop1, 4]) / 4
    FIMPARA
```

```
//APRESENTAÇÃO DOS RESULTADOS
PARA contadorLoop1 DE 1 ATE 5 FAÇA
    SE medias[contadorLoop1] >= 6 ENTAO
        ESCREVAL("O aluno(a) ", nomes[contadorLoop1], " foi aprovado com as
notas  (", notas[contadorLoop1, 1], ", ", notas[contadorLoop1, 2], ", ",
notas[contadorLoop1, 3], ", ", notas[contadorLoop1, 4], ") e média: ",
medias[contadorLoop1])
    SENAo
        ESCREVAL("O aluno(a) ", nomes[contadorLoop1], " foi reprovado com as
notas  (", notas[contadorLoop1, 1], ", ", notas[contadorLoop1, 2], ", ",
notas[contadorLoop1, 3], ", ", notas[contadorLoop1, 4], ") e média: ",
medias[contadorLoop1])
    FIMSE
FIMPARA

finalgoritmo
```

Repare que os **arrays** (vetores ou matrizes) aliados à estrutura de repetição PARA é um ótimo recurso para algoritmos que precisam de muitas variáveis do mesmo tipo.

Um resultado do algoritmo acima pode ser observado a seguir:

```
Digite o nome do aluno(a) número 1 de 5: Gustavo
Digite a nota 1 do aluno(a) Gustavo: 9
Digite a nota 2 do aluno(a) Gustavo: 10
Digite a nota 3 do aluno(a) Gustavo: 9,5
Digite a nota 4 do aluno(a) Gustavo: 8
Digite o nome do aluno(a) número 2 de 5: João
Digite a nota 1 do aluno(a) João: 5
Digite a nota 2 do aluno(a) João: 6
Digite a nota 3 do aluno(a) João: 4,5
Digite a nota 4 do aluno(a) João: 7
Digite o nome do aluno(a) número 3 de 5: Pedro
Digite a nota 1 do aluno(a) Pedro: 7
Digite a nota 2 do aluno(a) Pedro: 8,5
Digite a nota 3 do aluno(a) Pedro: 6
```

```
Digite a nota 4 do aluno(a) Pedro: 7
Digite o nome do aluno(a) número 4 de 5: Luciana
Digite a nota 1 do aluno(a) Luciana: 10
Digite a nota 2 do aluno(a) Luciana: 7
Digite a nota 3 do aluno(a) Luciana: 7,5
Digite a nota 4 do aluno(a) Luciana: 8
Digite o nome do aluno(a) número 5 de 5: Augusto
Digite a nota 1 do aluno(a) Augusto: 5
Digite a nota 2 do aluno(a) Augusto: 5,5
Digite a nota 3 do aluno(a) Augusto: 7,5
Digite a nota 4 do aluno(a) Augusto: 6
O aluno(a) Gustavo foi aprovado com as notas ( 9, 10, 9.5, 8) e média: 9.125
O aluno(a) João foi reprovado com as notas ( 5, 6, 4.5, 7) e média: 5.625
O aluno(a) Pedro foi aprovado com as notas ( 7, 8.5, 6, 7) e média: 7.125
O aluno(a) Luciana foi aprovado com as notas ( 10, 7, 7.5, 8) e média: 8.125
O aluno(a) Augusto foi aprovado com as notas ( 5, 5.5, 7.5, 6) e média: 6
*** Fim da execução.
*** Feche esta janela para retornar ao Visualg.
```

Para você que é um iniciante em programação, este algoritmo pode parecer um pouco complexo, mas se prestar atenção, perceberá que os vetores e matrizes podem ser utilizados em muitos problemas. Por exemplo, armazenar os nomes dos funcionários de uma empresa.

Uma coisa importante a se observar é que os arrays são de tamanho fixo, ou seja, eles nascem e morrem com o mesmo tamanho. Se você precisar acrescentar um novo valor em um array e ele já estiver cheio, você deverá criar um novo array maior e realocar os valores do array antigo.

Mas pode ficar tranquilo que existem outras estruturas de dados que crescem dinamicamente, mas isso é assunto para um capítulo futuro ...

Conclusão

Como você pode perceber neste capítulo, Vetores e Matrizes são, na verdade, a mesma coisa: **ARRAY**

A diferença é que o vetor é um array de apenas 1 dimensão e a matriz é um array de 2 (ou mais) dimensões.

Os arrays também são conhecidos por **variáveis indexadas**.

Array é uma das estruturas de dados mais simples que existe e uma das mais utilizadas também. Acho que todas as linguagens de programação têm **arrays**, pelo menos ainda não conheço uma linguagem que não tenha.

É importante falar que estruturas de dados é o assunto seguinte a se aprender, depois de aprender lógica de programação. Existem muitos tipos de estruturas de dados, o array é o mais simples. Conhecer bem as estruturas de dados é o que vai determinar a sua facilidade em aprender qualquer linguagem de programação.

Os índices dos arrays podem mudar dependendo da linguagem, algumas linguagens começam os índices do array com 1 e outras com 0, essa é uma diferença muito comum que encontramos entre linguagens. No caso das linguagens que começam os arrays com o índice 0, o último elemento do array recebe o índice ($\text{<tamanho do array>} - 1$).

Te desafio a criar um jogo da velha!

Nossa vida é cheia de desafios e eles são muito importantes para evoluirmos e ultrapassar os nossos limites.

Pensando na sua evolução, eu tenho um desafio para você resolver! Neste desafio você poderá utilizar tudo que aprendeu até agora. Inclusive a matriz que você aprendeu neste capítulo!

Quero ver se você aprendeu mesmo!

O desafio é o seguinte:

Você deverá construir um jogo da velha.

Simples assim.

Não precisa ser um jogo muito elaborado. Vamos ver alguns requisitos.

1. As jogadas do jogo da velha deverão ser armazenadas numa matriz (3x3) de caractere, chamada "tabuleiro", cada posição desta matriz armazenará um dos valores: " ", "_", "X" ou "O", onde " " e "_" são posições *vazias* e "X" e "O" são *jogadas*. Abaixo uma representação gráfica desta matriz.

	1	2	3
1	___	___	___
2	___	___	___
3			

2. A cada jogada o programa deverá mostrar na tela a situação atual do "tabuleiro". Por exemplo:

	1	2	3
1	___	___	___
2	___	_X_	___
3	O		O

3. Terão dois jogadores no jogo. O programa deve solicitar o nome dos dois jogadores antes de começar o jogo.

4. A cada jogada o programa deverá perguntar separadamente as posições horizontal e vertical da jogada, nesta ordem.
5. Quando um jogador vencer o programa deve apresentar imediatamente o vencedor e a situação do "tabuleiro".

Este exercício não é simples, mas com um pouquinho de esforço e persistência tenho certeza que você consegue fazer esse jogo.