

FUNDAMENTOS EM LÓGICA DE PROGRAMAÇÃO

Atividade: Começando a pensar logicamente e a desenvolver softwares utilizando o Portugol Studio

Tema: Coleção de dados - MATRIZ

INDICADORES ASSOCIADOS

2 - Analisa e avalia o funcionamento de computadores e periféricos em ambientes computacionais.

3 - Codifica programas computacionais utilizando lógica de programação e respeitando boas práticas de programação.

5 - Desenvolver capacidades linguísticas de modo a saber usar adequadamente a linguagem oral e escrita em diferentes situações e contextos.

8 - Utilizar estruturas de dados definindo-as e aplicando-as adequadamente nos programas.

COLEÇÃO DE DADOS - MATRIZ

- Você se lembra desse trecho, dito no material anterior?
 - Basicamente, o *VETOR* é uma coleção de dados;
 - A *MATRIZ*, por sua vez, é uma coleção de vetores.
- Aprendemos juntos o que é um vetor, agora vamos entender o que define uma matriz;
- Vamos usar uma analogia utilizada anteriormente:
- Lembra-se do exemplo do congestionamento de trânsito, na trilha de *VETORES*?
- Vamos continuar no exemplo do trânsito, mas agora migraremos para um pedágio;
- De Curitiba para Joinville, você encontrará no caminho dois pedágios;
- Quando você vai passar ele, tem várias pistas, correto?
- Cada uma das pistas do pedágio costuma ter vários veículos enfileirados, para passar ao outro lado;
- Numa analogia:

- A pista em que estamos parados no pedágio é o vetor;
 - Os veículos são os elementos que compõe esse vetor;
- Certo, mas e o pedágio, o que é?
- O pedágio é a matriz;
 - Você se lembra que “A *MATRIZ*, por sua vez, é uma coleção de vetores”?
- Portanto, nessa analogia:
 - O pedágio é formado por trilhas que possuem veículos;
 - Portanto, uma matriz é formada por vários vetores que possui várias informações.
- Lembre-se sempre de que:
 - O vetor é composto por vários dados;
 - A matriz é composta por vários vetores.
- “Matriz (array multi-dimensional) é um vetor de vetores” (ALVES);

Figura 1- Pedágio



Fonte: Autores, 2023.

- É muito próximo do conceito de linhas e colunas;
- Na prática, talvez, isso fique mais claro para você;
- Então, vamos para ela!
- Como funciona tudo isso na prática?
- Vamos descobrir em seguida, mas antes:

- Certifique-se que você conseguiu compreender o uso do *PARA FAÇA* no material de *VETORES*;
- Se você entendeu como funciona essa união de funções, pode prosseguir;
- Caso não tenha ficado claro para você:

PARE TUDO O QUE ESTÁ FAZENDO E NOS CHAME
É DE SUMA IMPORTÂNCIA QUE VOCÊ ENTENDA ISSO PARA QUE PROSSIGA

- Nossa atividade se dará da seguinte forma:
 - Faremos a criação de uma matriz que possua 2 linhas e 3 colunas:
 - Em outras palavras, a matriz é composta por 2 vetores com 3 elementos dentro de cada um.

Figura 2 - Desenvolvimento do código-fonte de matrizes

```
programa
{
    funcao inicio()
    {
        inteiro matriz[2][3]
```

Fonte: Autores, 2023.

- Declarada a nossa matriz, precisamos informar dados, certo?
- Nesse exemplo, faremos isso de maneira individualizada:
 - Ou seja, posição por posição.

Figura 3 - Desenvolvimento do código-fonte de matrizes

```
// 1ª Linha
matriz[0][0] = 10
matriz[0][1] = 20
matriz[0][2] = 30

// 2ª Linha
matriz[1][0] = 40
matriz[1][1] = 50
matriz[1][2] = 60
```

Fonte: Autores, 2023.

- Lembra que a matriz foi declarada assim: [2][3]?
- Então, isso quer dizer que eu possuo duas linhas;
- Lembra também que em vetores a primeira posição é 0, não 1?

- Pois bem, matrizes não fogem a essa regra;
- A posição é sempre POSIÇÃO – 1:
 - Linha 1 = Posição 0
 - Linha 2 = Posição 1
 - Linha 3 = Posição 2
- E o mesmo se aplica as colunas:
 - Coluna 1 = Posição 0
 - Coluna 2 = Posição 1
 - Coluna 3 = Posição 2
- Por isso, no nosso exemplo temos:
 - Linha 1 x Coluna 1 = 10
 - Linha 1 x Coluna 2 = 20
 - Linha 1 x Coluna 3 = 30
 - Linha 2 x Coluna 1 = 40
 - Linha 2 x Coluna 2 = 50
 - Linha 2 x Coluna 3 = 60
- Por fim, precisamos imprimir nossa matriz em tela:
- Para isso, percorreremos cada linha e coluna para resgatar o valor que tem lá;
- Criaremos, então, o seguinte trecho:

Figura 4 - Desenvolvimento do código-fonte de matrizes

```
para(inteiro i=0; i<2; i++){
    para(inteiro j=0; j<3; j++){
        escreva("Posição[" + (i + 1) + "][ " + (j + 1) + "] = " + matriz[i][j] + "\n")
    }
}
```

Fonte: Autores, 2023.

- Agora sim!
- Nosso código-fonte está completo.

Figura 5 - Desenvolvimento do código-fonte de matrizes

```
programa
{
    funcao inicio()
    {
        inteiro matriz[2][3]

        // 1ª Linha
        matriz[0][0] = 10
        matriz[0][1] = 20
        matriz[0][2] = 30

        // 2ª Linha
        matriz[1][0] = 40
        matriz[1][1] = 50
        matriz[1][2] = 60

        escreva("Vamos imprimir a matriz\n")

        para(inteiro i=0; i<2; i++){
            para(inteiro j=0; j<3; j++){
                escreva("Posição[" + (i + 1) + "][", (j + 1) + "] = " + matriz[i][j] + "\n")
            }
        }
    }
}
```

Fonte: Autores, 2023.

- Só que temos um “problema”;
- Nossos dados estão corretos, então por qual motivo nossa saída de dados não segue o formato de uma matriz?
- Por quê está saindo tudo em separado, assim?

Figura 6 - Desenvolvimento do código-fonte de matrizes

```
Vamos imprimir a matriz
Posição[1][1] = 10
Posição[1][2] = 20
Posição[1][3] = 30
Posição[2][1] = 40
Posição[2][2] = 50
Posição[2][3] = 60

Programa finalizado. Tempo de execução: 188 milissegundos
```

Fonte: Autores, 2023.

- Simples: você não está imprimindo valores como uma matriz;
- Você ainda está imprimindo como se fossem várias linhas:
- Uma linha embaixo da outra.
- Então, como podemos ajustar isso?
- Fácil!
- Você já usou o “\n” em algum momento?

- Então, essa cadeia de caracteres é interpretada como um saldo de linha pelo computador.
- Logo, faremos um pequeno ajuste em nosso código-fonte:
- Você trocará a sua saída de dados atual por esse novo trecho;
- Repare que o trecho foi um pouquinho alterado, mas nada muito gritante:

Figura 7 - Desenvolvimento do código-fonte de matrizes

```
para(inteiro i=0; i<2; i++){
    para(inteiro j=0; j<3; j++){
        escreva("[", (i + 1), "]" + (j + 1) + "]= " + matriz[i][j] + " ")
    }
    escreva("\n")
}
```

Fonte: Autores, 2023.

- Agora sim!
- Nossa saída tá certinha né?
- Mas, o que mudou?
- Basicamente, tiramos o salto de linha (“\n”) de dentro da troca de colunas;
- Repare que, na versão anterior, dentro do “para(inteiro j=0; j<3; j++)” existia um “\n”:
 - Ou seja, nosso salto de linha estava atrelado aos valores do vetor;
- Toda a vez que eu acessava um valor eu pulo a linha;
- Como tenho 6 valores, salto 6 vezes de linha.
- Agora, ele está localizado dentro, bem no finalzinho do “para(inteiro i=0; i<2; i++)”:
- Ou seja, ligado aos vetores que compõem a matriz;
- Aqui, toda a vez que eu vou verificar um outro vetor, eu salto a linha.
- Como tenho 2 vetores, com 6 elementos, salto 2 vezes de linha na saída.
- Então nosso código-fonte está finalizado assim:

Figura 8 - Desenvolvimento do código-fonte de matrizes

```
programa
{
    funcao inicio()
    {
        inteiro matriz[2][3]

        // 1ª Linha
        matriz[0][0] = 10
        matriz[0][1] = 20
        matriz[0][2] = 30

        // 2ª Linha
        matriz[1][0] = 40
        matriz[1][1] = 50
        matriz[1][2] = 60

        escreva("Vamos imprimir a matriz\n")

        para(inteiro i=0; i<2; i++){
            para(inteiro j=0; j<3; j++){
                escreva("[", (i + 1), "][" + (j + 1) + "]= " + matriz[i][j] + " ")
            }
            escreva("\n")
        }
    }
}
```

Fonte: Autores, 2023.

- Conseguimos chegar ao fim dessa trilha de conhecimento!
- Bora para os exercícios.

REFERÊNCIAS BIBLIOGRÁFICAS

ALVES, Gustavo Furtado de Oliveira. **O que são Vetores e Matrizes (arrays)**. Disponível em: <https://dicasdeprogramacao.com.br/o-que-sao-vetores-e-matrizes-arrays/>. Acesso em: 06 fev. 2023.

ND MAIS (org.). **Aumento do pedágio na BR-101 em SC**. Disponível em: <https://ndmais.com.br/webstories/aumento-do-pedagio-na-br-101-em-sc/>. Acesso em: 06 fev. 2023.