## Declaração de Vetores

Para ler quatro notas e calcular e exibir a média artimética dessas notas, o mais comum é declarar quatro variáveis para as notas, ler cada uma das notas e somente depois de terminar a entrada de dados, calcular e exibir a média ao mesmo tempo, sem a necessidade da criação de uma variável para isso, conforme demonstrado na Figura 1.

Figura 1: Ler 4 notas, calcular e exibir a média aritmética

```
programa
{
    funcao inicio()
    {
        real nota1, nota2, nota3, nota4

        escreva("Digite a 1ª nota: ")
        leia(nota1)

        escreva("Digite a 2ª nota: ")
        leia(nota2)

        escreva("Digite a 3ª nota: ")
        leia(nota3)

        escreva("Digite a 4ª nota: ")
        leia(nota4)

        escreva("Média: ",(nota1+nota2+nota3+nota4)/4)
    }
}
```

Fonte: Elaborado pela autora

Apesar de ser possível obter o mesmo resultado declarando somente uma variável para a leitura das quatro notas e outra para a média. Para este programa, após cada entrada de notas, seu valor é somado na variável media, pois seu valor é perdido quando se utiliza a mesma variável para receber dados novamente (Figura 2). Então, realiza-se a soma na variável media para no final apenas efetuar a divisão por quatro para efetuar o cálculo e exibir o resultado da média aritmética de quatro notas.

Figura 2: Ler 4 notas, calcular e exibir a média aritmética (simplificado)

```
programa
{
    funcao inicio()
    {
        real nota, media=0.0

        escreva("Digite a 1ª nota: ")
        leia(nota)
        media += nota

        escreva("Digite a 2ª nota: ")
        leia(nota)
        media += nota

        escreva("Digite a 3ª nota: ")
        leia(nota)
        media += nota

        escreva("Digite a 4ª nota: ")
        leia(nota)
        media += nota

        escreva("Média: ",media/4)
    }
}
```

Fonte: Elaborado pela autora

Nesta situação em que há apenas quatro notas, tanto o primeiro programa quanto o segundo não ficam com o código tão longo. Mas se for necessário efetuar a média aritmética com mais notas, como por exemplo cem notas, esta solução não é prática, pois o programa final não ficará complexo, porém muito longo devido a duplicidade necessária para a entrada das notas.

Para trabalhar com um conjunto de dados, há um tipo de variável denominada de vetor ou *array* unidimensional por ter somente uma dimensão, isto é, pode ter sua representação em uma linha ou em uma coluna. É uma variável que armazena diversos valores do mesmo tipo.

No Portugol Studio, os vetores também são declarados, iniciando pelo seu tipo, seguido do seu nome e em seguida a sua dimensão:

**tipo_de_dado** nome_variável_vetor[dimensão]

O vetor também pode ser inicializado, onde cada valor é separado por vírgula:

**tipo_de_dado** nome_variável_vetor[dimesão] = {valores_iniciais}

## VOCÊ SABIA?

A dimensão em um vetor pode ser opcional se em sua declaração já tiver valores atribuídos. Neste caso a sua dimensão será de acordo com a quantidade de valores separados por vírgula:

**tipo_de_dado** nome_variável_vetor[] = {valores_iniciais}

Para acessar os valores de um vetor é necessário informar a posição desejada, que em programação é chamada de índice.

No Portugol Studio, como na grande maioria das linguagens de programação, o índice é um número inteiro, o primeiro elemento do vetor está no índice 0 (zero) e o último está no índice dimensão-1.

A Figura 3 apresenta exemplos de declarações de vetores, bem como atribuição de valores por meio dos índices.

Figura 3: Declaração de vetores no Portugol Studio

```
programa
{
    funcao inicio()
    {
        real nota[4] // Declaração de vetor de 4 posições

        // Declaração de vetores com valores inicias
        inteiro mega[6] = {12,28,33,45,51,55}
        logico resposta[] = {falso,verdadeiro,falso}

        // Atribuindo valor para a primeira posição do vetor
        nota[0]=7.5
    }
}
```

Fonte: Elaborado pela autora

## Vetores

O vetor, também chamado de *array* unidimensional é uma estrutura de dados que armazena valores de um mesmo tipo, como um conjunto de variáveis de mesmo tipo que possui um único nome (Figura 4).

Figura 4: Representação de Vetor em linha

| índice: | 0 | 1 | 2 | | | | n-1 |
|---|---|---|---|---|---|---|---|
| conteúdo: | **Ana** | **Lia** | **Jô** | | | | **Bia** |

Fonte: Elaborado pela autora

A atribuição de valores no vetor é feita diretamente em cada posição do vetor através do índice.

Para exemplificar, a Figura 5 demostra o uso de vetor para armazenar as quatro notas ao invés de quatro variáveis. Em seguida efetua o cálculo da média aritmética e apresenta o resultado.

Figura 5: Ler 4 notas, calcular e exibir a média aritmética (com vetor)

```
programa
{
    funcao inicio()
    {
        real nota[4]

        escreva("Digite a 1ª nota: ")
        leia(nota[0])

        escreva("Digite a 2ª nota: ")
        leia(nota[1])

        escreva("Digite a 3ª nota: ")
        leia(nota[2])

        escreva("Digite a 4ª nota: ")
        leia(nota[3])

        escreva("Média: ",(nota[0]+nota[1]+nota[2]+nota[3])/4)
    }
}
```

Fonte: Elaborado pela autora

O programa da Figura 5, apesar de utilizar vetor ainda não é o ideal

para o uso de muitas notas. Para melhorar, como vários trechos são iguais, a solução é utilizar estrutura de repetição.

A Figura 6 demonstra o código de um programa para ler quatro notas e calcular e exbir a média aritmética dessas notas, de forma que já está preparado para efetuar o cálculo de quantidades maiores de notas. Para isso basta alterar somente o valor da constante dimesao.

Figura 6: Ler 4 notas, calcular e exibir a média aritmética (com vetor e estrutura de repetição)

```
programa
{
    funcao inicio()
    {
        const inteiro DIMENSAO = 4
        real nota[DIMENSAO], media=0.0

        para (inteiro i=0; i<DIMENSAO; i++)
        {
            escreva("Digite a "+(i+1)+"ª nota: ")
            leia(nota[i])
            media += nota[i]
        }

        escreva("Média: ",media/DIMENSAO)
    }
}
```

Fonte: Elaborado pela autora

No Portugol Studio, há um exemplo de uso de vetor em Bibliotecas → Util → Tamanho do Vetor (Figura 7).

Nesta situação, o vetor é declarado com valores, o que não o obriga a informação da sua dimensão. Mas para se utilizar a estrutura de repetição para passar por todos os elementos do vetor através do índice, é necessário informar no comando "**para**" a condição de parada do laço que é o tamanho do vetor menos um. Então, utiliza-se a função "**numero_elementos**" encontrada na biblioteca "**Util**", que tem a finalidade de informar o número de elementos que o vetor possui, ou seja, a sua dimensão:

**numero_elementos (**nome do vetor**)**

Figura 7: Exemplo vetor.por do Portugol Studio

```
programa
{
    inclua biblioteca Util --> u

    funcao inicio()
    {
        // Cria um vetor com 5 elementos
        inteiro vet[] = { 4, 2, 9, 3, 8 }

        // Experimente substituir o vetor acima por este e veja que
        // o programa consegue percorrer normalmente o novo vetor

        // inteiro vet[] = { 4, 2, 9, 3, 8, 6, 5, 6, 2, 3, 9, 1 }

        inteiro elementos = u.numero_elementos(vet)

        escreva("O vetor possui ", elementos, " elementos:\n\n")

        // Utilizamos o valor obtido anteriormente para percorrer
        // o vetor e exibir seus valores
        para (inteiro elemento = 0; elemento < elementos; elemento++)
        {
            se (elemento == 0)
            {
                escreva("{ ")
            }

            escreva(vet[elemento])

            se (elemento < elementos - 1)
            {
                escreva(", ")
            }

            se (elemento == elementos - 1)
            {
                escreva(" }")
            }
        }

        escreva("\n")
    }
}
```

Fonte: portugol-studio.jar, 2022

## Declaração de Matrizes

Em uma situação em que o cálculo da média aritmética deve ser separada por disciplina pode-se criar um vetor para cada uma das disciplinas (Figura 8). E para cada um dos vetores a programação se repete

para efetuar o cálculo, apresentando o resultado separado por disciplina.

Figura 8: Exemplo de declaração de vários vetores

```
programa
{
    funcao inicio()
    {
        real nota1[4], nota2[4], nota3[4]
        real media1, media2, media3

        /*
         * Cada vetor representa uma disciplina.
         * Ler as notas para cada um dos vetores.
         * Calcular a média aritmética para cada vetor
         * Exibir cada média aritmética.
         *
         */
    }
}
```

Fonte: Elaborado pela autora

Porém, para esta situação, o ideal é a declaração de matriz que é uma vetor com mais de uma dimensão (linha e coluna). Também chamado de *array* multidimensional pois é um vetor de vetores.

A matriz é um tipo de variável que armazena diversos valores do mesmo tipo em linhas e colunas. É declarada inciando pelo seu tipo, seguida do seu nome e as dimensões de linhas e colunas em sequencia:

**tipo_de_dado** nome_variável_matriz[dimensão_linha] [dimensão_coluna]

A matriz, como ocorre com os vetores, também pode ser inicializada, colocando cada linha da matriz entre chaves, separadas por vírgula:

**tipo_de_dado** nome_variável_matriz[dimensão_linha] [dimensão_coluna] = {{valores_inicias},{valores_inicias},...}

## VOCÊ SABIA?

A dimensão em uma matriz pode ser opcional se em sua declaração já tiver valores atribuídos. Neste caso a sua dimensão será de acordo com a quantidade de valores em linhas e colunas separados por vírgula:

**tipo_de_dado** nome_variável_vetor[][] = {{valores_inicias},{valores_inicias},...}

Para acessar os valores de uma matriz é necessário informar a posição desejada do índice em cada uma das duas dimensções (linha e coluna).

Lembrando que no Portugol Studio, o índice inicia-se na posição 0 (zero) e a última posição é dimensão -1 tanto para as linhas quanto para as colunas.

A Figura 9 demonstra exemplos de declarações de matrizes e a atribuição de valores por meio dos índices.

Figura 9: Declaração de matrizes no Portugol Studio

```
programa
{
    funcao inicio()
    {
        real nota[4][3] // Declaração de matriz com 4 linhas e 3 colunas

        // Declaração de matrizes com valores inicias
        inteiro apto[2][4] = {{11,12,13,14},{21,22,23,24}}
        caracter resposta[][] = {{'A','C'},{'B','D'},{'A','E'}}

        // Atribuindo valor para a primeira posição da matriz
        nota[0][0]=8.5
    }
}
```

Fonte: Elaborado pela autora

## Matrizes

A matriz, também chamado de array multidimensional é uma estrutura de dados que armazena valores de um mesmo tipo, como um conjunto de variáveis de mesmo tipo que possui um único nome. A Figura 10 representa uma matriz de duas dimensões (linha e coluna).

Figura 10: Representação de Matriz

| | Coluna 0 | Coluna 1 | Coluna 2 | | Coluna n-1 |
|---|---|---|---|---|---|
| Linha 0 | **2.0** | **5.5** | **7.2** | | **9.1** |
| Linha 1 | **4.5** | **9.0** | **6.6** | | **1.2** |
| Linha 2 | **4.7** | **9.3** | **5.5** | | **4.2** |
| | | | | | |
| Linha n-1 | **9.3** | **8.2** | **3.3** | | **2.4** |

Fonte: Elaborado pela autora

A atribuição de valores na matriz é feita diretamente em cada posição da matriz através de índices que representam a linha e coluna.

Para exemplificar, a Figura 11 demostra o uso de matriz para armazenar quatro notas de duas disciplinas. Em seguida efetua o cálculo da média aritmética de cada uma das disciplinas e apresenta o resultado. Para esta situação, cada linha é uma disciplina e as colunas contém as notas de cada uma delas. Portanto, a matriz é 2x4, representando as duas disciplinas (linhas) e as quatro notas (colunas). Não há uma ordem correta para representar as dimensões de linhas e colunas, pois isso depende da organização do conteúdo na matriz de acordo com seu propósito. Então, se inverter as notas em linhas e disciplinas em colunas, também estará correto se ficar claro para o usuário do programa como os dados serão organizados para utilizá-los conforme os objetivos apresentados.

Figura 11: Ler 4 notas de 2 disciplinas, calcular e exibir a média aritmética

```
programa
{
    funcao inicio()
    {
        real nota[2][4]

        escreva("Digite a 1ª nota da 1ª disicplina: ")
        leia(nota[0][0])
        escreva("Digite a 2ª nota da 1ª disicplina: ")
        leia(nota[0][1])
        escreva("Digite a 3ª nota da 1ª disicplina: ")
        leia(nota[0][2])
        escreva("Digite a 4ª nota da 1ª disicplina: ")
        leia(nota[0][3])
        escreva("\n")

        escreva("Digite a 1ª nota da 2ª disicplina: ")
        leia(nota[1][0])
        escreva("Digite a 2ª nota da 2ª disicplina: ")
        leia(nota[1][1])
        escreva("Digite a 3ª nota da 2ª disicplina: ")
        leia(nota[1][2])
        escreva("Digite a 4ª nota da 2ª disicplina: ")
        leia(nota[1][3])
        escreva("\n")

        escreva("Média da 1ª disicplina: ",(nota[0][0]+nota[0][1]+nota[0][2]+nota[0][3])/4)
        escreva("\n")
        escreva("Média da 2ª disicplina: ",(nota[1][0]+nota[1][1]+nota[1][2]+nota[1][3])/4)
    }
}
```

Fonte: Elaborado pela autora

O programa da Figura 11, apesar de utilizar matriz ainda não é o ideal para o uso de muitas disciplinas e muitas notas. Para melhorar, como vários trechos são iguais, a solução é utilizar estrutura de repetição.

A Figura 12 demonstra o código de um programa para ler quatro notas de duas disciplinas, calcular e exbir a média aritmética dessas notas por disciplinas, de forma que já está preparado para efetuar o cálculo de quantidades maiores de notas e disciplinas. Para isso basta alterar somente o valor da constante dimesao de linha e coluna.

Para armazenar a média de cada uma das disciplinas o vetor é o tipo de variável adequado quando se trata de apenas uma dimensão. Mas como a definição da dimensão está em uma constante é necessário iniciar com uma estrutura de repetição para inicializar com zero todas as posições do vetor, pois o mesmo é somado com seu próprio valor.

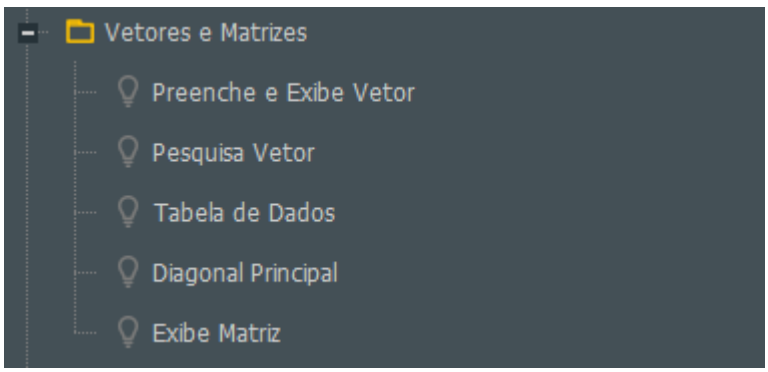
Figura 12: Ler 4 notas de 2 disciplinas, calcular e exibir a média aritmética (com estrutura de repetição)

```
programa
{
    funcao inicio()
    {
        const inteiro DIMENSAO_LINHA = 2
        const inteiro DIMENSAO_COLUNA = 4
        real nota[DIMENSAO_LINHA][DIMENSAO_COLUNA]
        real media[DIMENSAO_LINHA]

        para (inteiro i=0; i<DIMENSAO_LINHA; i++)
        {
            media[i] = 0.0
        }

        para (inteiro linha=0; linha<DIMENSAO_LINHA; linha++)
        {
            para (inteiro coluna=0; coluna<DIMENSAO_COLUNA; coluna++)
            {
                escreva("Digite a "+(coluna+1)+"ª nota da "+(linha+1)+"ª disciplina: ")
                leia(nota[linha][coluna])
                media[linha] += nota[linha][coluna]
            }
            escreva("\n")
        }

        para (inteiro i=0; i<DIMENSAO_LINHA; i++)
        {
            escreva("Média da "+(i+1)+"ª disciplina: ",media[i]/DIMENSAO_COLUNA,"\n")
        }
    }
}
```

Fonte: Elaborado pela autora

No Portugol Studio, há exemplos de programas com vetores e matrizes encontrados em Exemplos -> Vetores e Matrizes, conforme apresentado na Figura 13.

Figura 13: Algoritmos com vetores e matrizes do Portugol Studio




Fonte: portugol-studio.jar, 2022

# VOCÊ SABIA?

No Portugol Studio há uma função da biblioteca **Util** denominada **sorteia** que sorteia um número entre os valores informados, incluindo os valores inicial e final:

**sorteia(**valor inicial, valor final**)**

Esta função é muito útil para preencher vetores e/ou matrizes com números aleatórios.

# PARA PRATICAR

1. Criar um vetor de 10 elementos com números inteiros aleatórios de 1 até 100. Ordenar o vetor em ordem crescente e exibir seus valores separados por tabulação.

2. Criar um vetor denominado mega de 6 elementos com números inteiros aleatórios de 1 a 60. Ler 6 números e armazená-los em um vetor. Apresentar quais são os números iguais entre os dois vetores.

3. Ler o número de moradores de cada um dos apartamentos de um edifício de 10 andares com 4 apartamentos por andar. Apresentar a quantidade de moradores e a média aritmética.

4. Criar uma matriz 4x2 com números inteiros aleatórios de 1 a 100. Apresentar a posição da matriz em que se encontram o menor e o maior número armazenado, bem como seus respectivos valores.

5. Criar duas matrizes 3x3 com números inteiros aleatórios de 1 a 10. Calcular e apresentar a soma, a subtração e a multiplicação das matrizes.

# REFERÊNCIAS


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 abr. 2022.

ÁVILA, Walter Marlon Mamedes. **Algoritmo: Estrutura de vetores e matrizes.** Fábrica de Software, 2013. Disponível em: <http://fabrica.ms.senac.br/2013/06/algoritmo-estrutura-de-vetores-e-matrizes/>. Acesso em 01 abr. 2022.

BRANDÃO, Leônidas de Oliveira. **Introdução às matrizes**. Disponível em: <https://www.ime.usp.br/~leo/mac2166/2017-1/line_introducao_matrizes.html> Acesso em 05 abr. 2022.

FEOFILOFF, Paulo. **Vetores**. Disponível em: <https://www.ime.usp.br/~pf/algoritmos/aulas/array.html> Acesso em 01 abr. 2022.

GASPAR, Wagner. **VETOR – estrutura de dados homogênea (array unidimensional)**. Wagner Gastar, 2021. Disponível em: <https://wagnergaspar.com/vetor-estrutura-de-dados-homogenea-array-unidimensional/>. Acesso em 01 abr. 2022.