

# Etapa 6 — Vetores e Matrizes (Arrays and Matrices)

Trilha: Lógica de Programação

# O que são Vetores e Matrizes?

- **Vetores (Arrays) e Matrizes (Matrices) são estruturas de dados.**
- Permitem **armazenar vários valores** em uma mesma variável.
- Facilitam o **acesso e a manipulação de conjuntos de dados.**
- **Exemplo:**

Armazenar 10 notas de alunos em um vetor, em vez de 10 variáveis diferentes.

# O que é um Vetor (Array)?

- Um **vetor** é uma **lista linear de elementos** do mesmo tipo.
- Cada elemento é identificado por um **índice (index)**.
- O **primeiro índice normalmente é 0 ou 1**, dependendo da linguagem.
- **Exemplo visual:**

Índice	1	2	3	4	5
Valor	10	8	9	7	6

# Declaração de Vetores

- **Sintaxe genérica:**

declare notas[5] como real

- **Atribuição e acesso:**

notas[1]  $\leftarrow$  8.5

escreva notas[1]

- **Saída:**

8.5

- **Dica:**

Os colchetes [] indicam a posição do elemento dentro do vetor.

# Preenchendo um Vetor com Loop

- **Exemplo:**

para i de 1 ate 5 faca

    escreva "Digite a nota ", i, ":"

    leia notas[i]

fimpara

- **Para exibir todas as notas:**

para i de 1 ate 5 faca

    escreva "Nota ", i, " = ", notas[i]

fimpara

- **Uso comum:**

Percorrer vetores inteiros com loops (laços).

# Operações Comuns em Vetores

- Somar todos os elementos
- Calcular média
- Encontrar o maior ou menor valor
- **Exemplo:**

soma  $\leftarrow$  0

para i de 1 ate 5 faça

    soma  $\leftarrow$  soma + notas[i]

fimpara

media  $\leftarrow$  soma / 5

escreva "Média = ", media

# O que é uma Matriz (Matrix)?

- Uma **matriz** é uma **tabela bidimensional (duas dimensões)**.
- Possui **linhas (rows)** e **colunas (columns)**.
- Cada elemento é identificado por **dois índices**.
- **Exemplo visual:**

	Coluna 1	Coluna 2	Coluna 3
Linha 1	5	8	9
Linha 2	2	4	6

# Declaração de Matrizes

- **Sintaxe genérica:**

declare matriz[2][3] como inteiro

- **Atribuição e acesso:**

matriz[1][2]  $\leftarrow$  8

escreva matriz[1][2]

- **Saída:**

8

- **Dica:**

O primeiro índice é a **linha**, o segundo é a **coluna**.



# Preenchendo Matrizes

- **Exemplo:**

```
para i de 1 ate 2 faca
    para j de 1 ate 3 faca
        escreva "Digite o valor [", i, ",", j, "]:"
        leia matriz[i][j]
    fimpara
fimpara
```

- **Exibir matriz:**

```
para i de 1 ate 2 faca
    para j de 1 ate 3 faca
        escreva matriz[i][j], " "
    fimpara
    escreva ""
fimpara
```

# Aplicações de Matrizes

- Tabelas e planilhas.
- Representação de notas por aluno e disciplina.
- Mapas e grades em jogos.
- Processamento de imagens e cálculos matemáticos.
- Dica:

Matrizes são **a base** para estruturas mais complexas como **matrizes tridimensionais e vetores de vetores**.

# Exemplo Prático: Soma de Matrizes

- **Algoritmo:**

declare  $A[2][2]$ ,  $B[2][2]$ ,  $C[2][2]$  como inteiro

para  $i$  de 1 ate 2 faca

    para  $j$  de 1 ate 2 faca

$C[i][j] \leftarrow A[i][j] + B[i][j]$

    fimpara

fimpara

- **Resultado:**

$C$  contém a soma elemento a elemento das matrizes  $A$  e  $B$ .

# Diferença Entre Vetor e Matriz

Estrutura	Dimensões	Exemplo de acesso
Vetor	1D	notas[3]
Matriz	2D	matriz[2][3]

- **Dica:**

Pense no vetor como uma **linha** e na matriz como uma **tabela**.

# Erros Comuns

- Usar índices fora do limite (ex: acessar notas[6] em vetor[5]).
- Confundir linha e coluna.
- Não inicializar os elementos antes de usar.
- Sempre **defina o tamanho corretamente e teste os índices!**

# Exemplo Completo

algoritmo "notas\_alunos"

inicio

declare notas[5] como real

para i de 1 ate 5 faca

escreva "Digite a nota ", i, ":"

leia notas[i]

fimpara

soma  $\leftarrow$  0

para i de 1 ate 5 faca

soma  $\leftarrow$  soma + notas[i]

fimpara

media  $\leftarrow$  soma / 5

escreva "Média = ", media

fim

- **Saída:**

Média = 8.2 (exemplo)

# Dica de Lógica

- Use vetores para listas simples.
- Use matrizes para tabelas.
- Use loops aninhados para percorrer estruturas bidimensionais.

“Dominar vetores e matrizes é o primeiro passo para entender estruturas de dados mais avançadas.”

# Conclusão

- **Vetores (Arrays) e Matrizes (Matrices)** permitem **armazenar e manipular múltiplos valores** de forma organizada.
- O uso dessas estruturas reduz **o número de variáveis** e **simplifica o código**.
- São a **base para estruturas mais complexas**, como listas, filas e árvores.
- Entender como percorrer vetores e matrizes com **loops (for / while)** é essencial para resolver problemas mais avançados.
- Resumo essencial:  
“Vetores e matrizes são a ponte entre a lógica básica e a programação estruturada.”