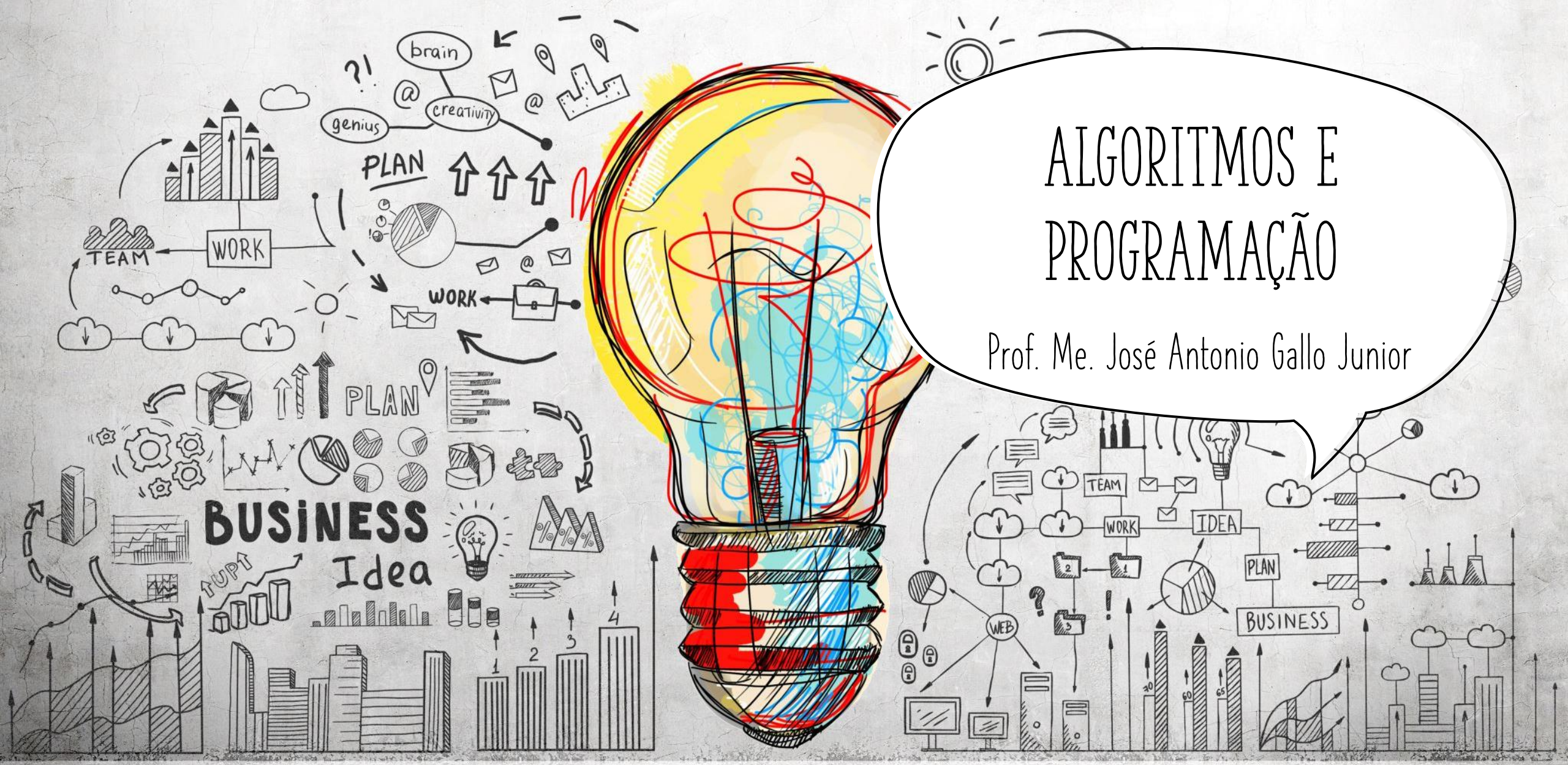


ALGORITMOS E PROGRAMAÇÃO

Prof. Me. José Antonio Gallo Junior

VETORES E MATRIZES

Quando precisamos guardar **vários valores do mesmo tipo**, como as notas de uma turma ou os preços de uma lista de produtos, não é prático criar uma variável diferente para cada dado.
Para isso usamos **Vetores** e **Matrizes**.

- **Vetores** permitem armazenar vários valores em uma única variável, organizados em posições numeradas.
- **Matrizes** são como tabelas, com linhas e colunas, permitindo guardar dados em duas dimensões.

Essas estruturas facilitam o trabalho com grandes quantidades de informações e deixam os algoritmos mais organizados.

VETORES

Guardar a **nota de um aluno** é simples: basta usar uma variável do tipo real.

Mas como armazenar as notas de **toda uma turma**? Teríamos que criar uma variável para cada aluno?

E se a quantidade de alunos mudasse de turma para **turma**?

E se além das notas quiséssemos também os **nomes**?

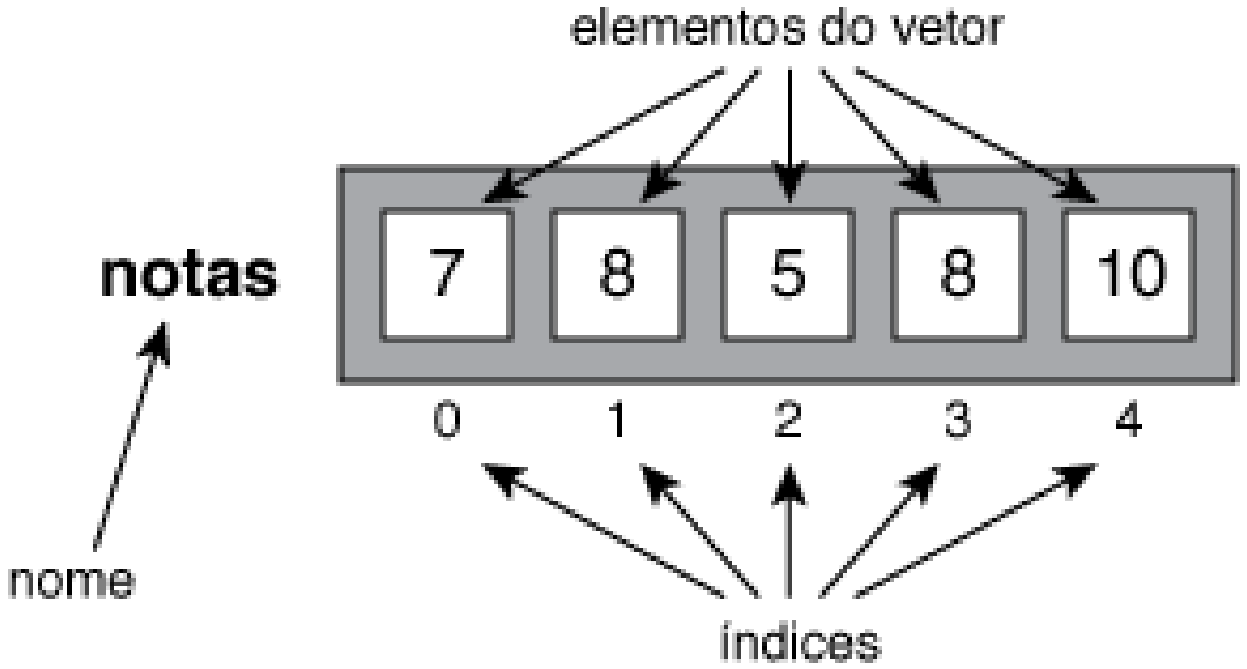
Criar muitas variáveis deixaria o programa confuso e difícil de controlar.

VETORES

Para resolver isso, usamos uma **estrutura de dados** que agrupa vários valores em um único nome: o **vetor**.

Um **vetor** pode ser entendido como uma **variável com várias posições**, onde cada posição guarda um valor do mesmo tipo.

VETORES



VETORES

Assim como as variáveis comuns, os vetores precisam ser **declarados**.

A declaração segue uma lógica parecida:

1. Primeiro indicamos o **tipo** dos valores que ele vai guardar.
2. Depois damos um **nome** para o vetor.
3. Por fim, definimos a **dimensão** (quantas posições ele terá) entre colchetes.

VETORES - SINTAXE

```
inteiro vetor[5]
caracter vetor2[200]

//vetores inicializados
real vetor3[2] = {1.4,2.5}
logico vetor4[4] = {verdadeiro, falso, verdadeiro, verdadeiro}
cadeia vetor5[] = {"Questão", "Fundamental"}

//Mudando o valor do vetor5 na posição 0 de "Questão" para "Pergunta"
vetor5[0] = "Pergunta"
```

VETORES

Cada valor dentro de um vetor é chamado de **elemento**, e para acessar um elemento precisamos indicar a **posição** dele no vetor.

Essa posição é identificada por um número chamado **índice**.

O **índice** é sempre um número inteiro colocado entre **colchetes [ ]**, logo após o nome do vetor.

Por padrão, a **primeira posição** de um vetor tem índice **0** (assim como a maioria das linguagens de programação).

VETORES

A última posição depende do tamanho do vetor.

- **Exemplo**: um vetor com **10 elementos** possui as posições **0 a 9**.
- Já um vetor com **4 elementos** possui as posições **0 a 3**.

Isso significa que o número total de elementos do vetor é sempre igual ao seu **tamanho**, mas o índice começa em **0** e vai até **tamanho - 1**.

VETORES - EXEMPLO

```
funcao inicio()
{
    //Declaração de um vetor de inteiros
    // de cinco posições já inicializado.
    inteiro vetor[5] = {15,22,8,10,11}

    //Imprime o valor 15 correspondente
    // ao primeiro elemento do vetor.
    escreva(vetor[0])
    escreva("\n")
```

VETORES - EXEMPLO

```
//Imprime o segundo elemento do vetor
escreva(vetor[1])
escreva("\n")

//Imprime o  valor 11 correspondente
// ao último elemento do vetor
escreva(vetor[4])
```

VETORES - EXEMPLO

```
    //Declaração de um vetor de reais de dez posições
    real outro_vetor[10]

    //Declaração de um vetor de caracteres onde o tamanho
    // é definido pela quantidade de elementos da inicialização
    caracter nome[] = {'P','o','r','t','u','g','o','l'}
}
```

MATRIZES

Para entender bem o que é uma **matriz**, é importante lembrar primeiro do conceito de **vetor**.

Um **vetor** já nos ajuda a guardar vários valores, mas ele é **linear** (uma única linha de dados).

Agora imagine que precisamos guardar as **três notas de quatro alunos diferentes**.

Se usássemos apenas vetores, isso ficaria confuso.

Para esse tipo de situação existe uma estrutura mais adequada: a **matriz**.

MATRIZES

Uma **matriz** pode ser vista como um **vetor com mais de uma dimensão** (normalmente duas)

.Enquanto o vetor organiza os dados em uma linha, a matriz organiza em forma de **tabela**, com **linhas** e **colunas**.

A tabela ao lado ilustra uma matriz que permite armazenar 3 notas de 4 alunos.

Posições	0	1	2
0	10	9	6.7
1	6	8	10
2	8	7	4.5
3	5.2	3.3	0.3

MATRIZES

Cada **linha** da matriz pode representar, por exemplo, um **aluno**, e cada **coluna** pode guardar uma de suas **notas**.

No caso de uma turma com 4 alunos e 3 notas cada, teremos uma matriz com 4 linhas e 3 colunas.

Assim como acontece com os vetores, **todos os elementos da matriz são do mesmo tipo** (inteiros, reais, etc.).

MATRIZES

Na **declaração** de uma matriz, devemos informar:

1. O **tipo de dado** que será armazenado.
2. O **nome da matriz**.
3. O número de **linhas**.
4. O número de **colunas**.

MATRIZES

Para acessar um elemento da **matriz** – seja para atribuir ou ler um valor – usamos **dois índices**:

- O **primeiro índice** indica a **linha**.
- O **segundo índice** indica a **coluna**.

Esses índices sempre aparecem entre **colchetes [ ]** logo após o nome da matriz e devem ser valores inteiros (constantes ou variáveis).

MATRIZES - SINTAXE

```
//Declaração de uma matriz de números reais com 5 linhas e 3 colunas
real nome_da_variavel[5][3]

//Gravar um valor na matriz na posição 0
//(1ª linha) e 1 (2ª coluna)
Nome_da_variavel[0][1] = 2.5
```

Posições	0	1	2
0		2.5	
1			
2			
3			
4			

MATRIZES - EXEMPLO

```
funcao inicio()
{
    //Declaração de uma matriz de inteiros
    // de duas linhas e duas colunas já inicializado.
    inteiro matriz[2][2] = {{15,22},{10,11}}

    //Atribui -1 na primeira linha e segunda
    //coluna da matriz
    matriz[0][1] = -1
```

MATRIZES - EXEMPLO

```
//Imprime o valor 15 correspondente
// a primeira linha e primeira coluna da matriz.
inteiro i = 0
escreva(matriz[i][0], "\n")

//Imprime o valor 11 correspondente
// a última linha e última coluna da matriz.
escreva(matriz[1][1])
```

MATRIZES - EXEMPLO

```
//Declaração de uma matriz de caracteres onde o tamanho
// de linhas e colunas são definidos pela inicialização
caracter jogo_velha[][] = {{'X', 'O', 'X'}
                           ,{'O', 'X', 'O'}
                           ,{' ', ' ', 'X'}}

//Declaração de uma matriz de reais de
// duas linhas e quatro colunas.
real outra_matriz[2][4]
```

VAMOS PRATICAR!?!