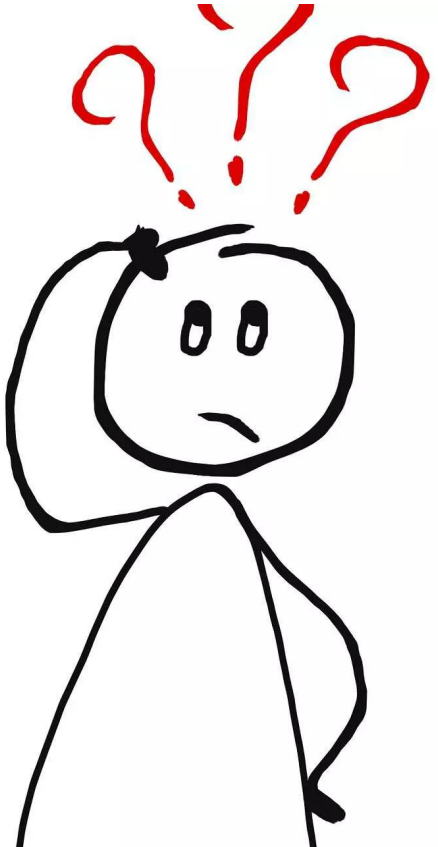
ALGORITMOS E ESTRUTURA DE DADOS

Curso de Engenharia de Software
Lucas Sampaio Leite

O que são vetores e matrizes?

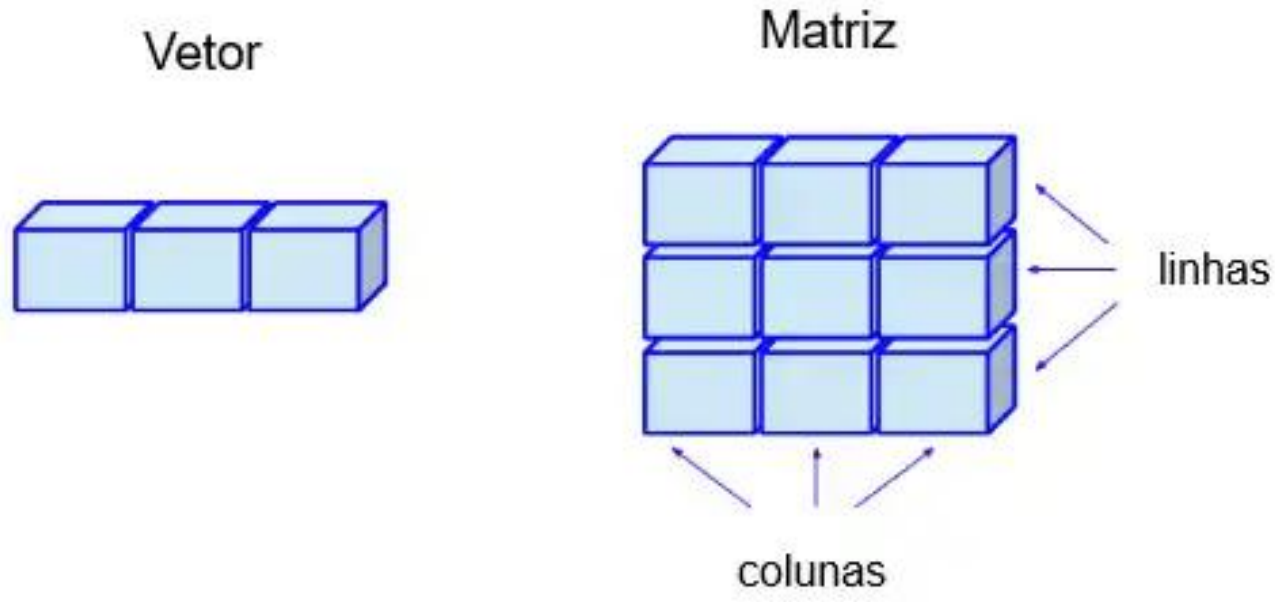
INSTITUTO FEDERAL
Pernambuco



Vetores e Matrizes

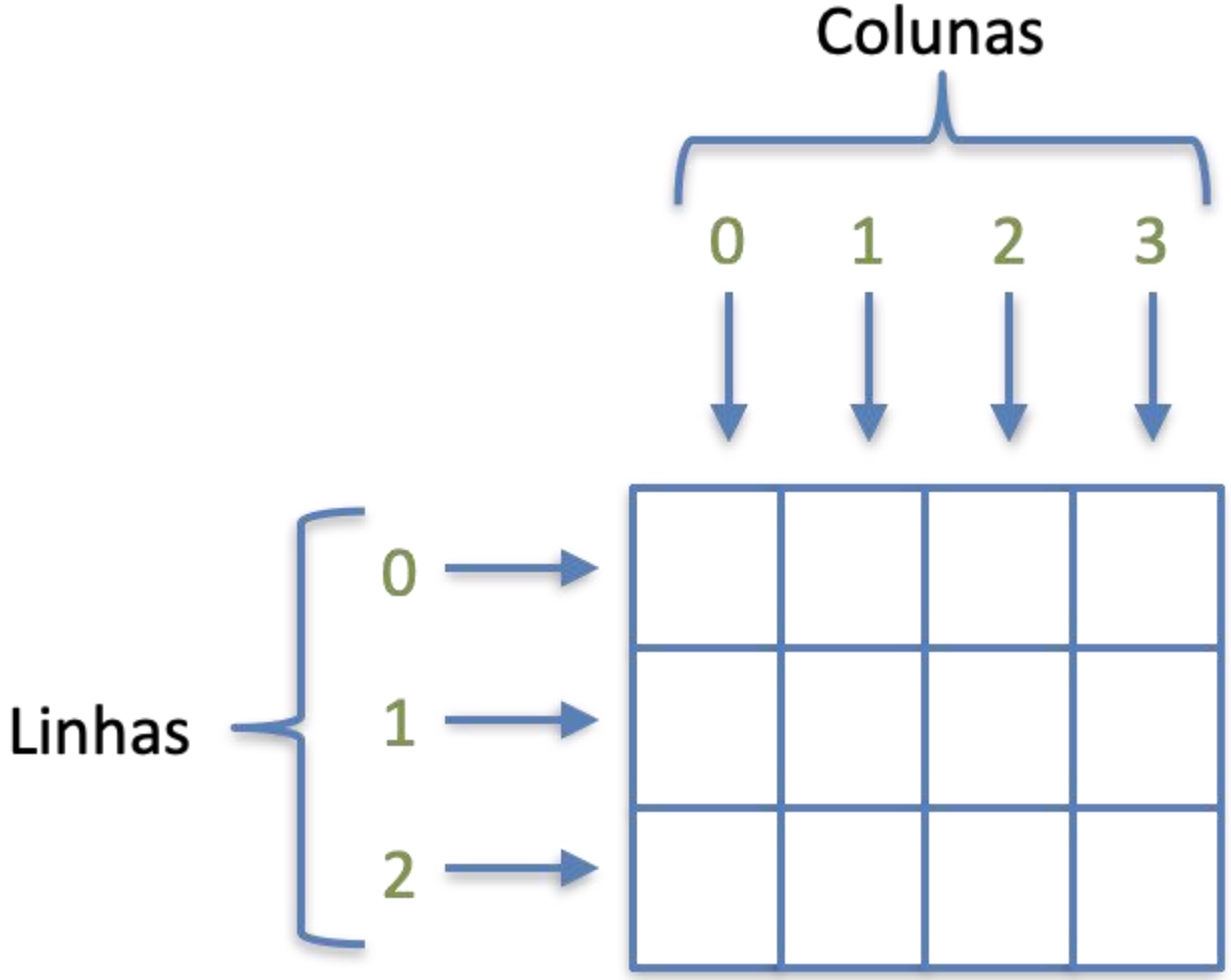
INSTITUTO FEDERAL
Pernambuco

- Um **vetor** (ou array unidimensional) é uma estrutura de dados que armazena uma **sequência de elementos do mesmo tipo**, organizados em uma **única dimensão** e acessados por índices.
- Uma **matriz** é uma extensão dos vetores: é um **array multidimensional**, podendo ter duas dimensões (linhas e colunas) ou mais, como matrizes tridimensionais e n-dimensionais.



Vetores e Matrizes

INSTITUTO FEDERAL
Pernambuco



Vetores

INSTITUTO FEDERAL
Pernambuco

- Um vetor é uma estrutura de dados linear que armazena uma coleção ordenada de elementos do mesmo tipo.
- Cada elemento é identificado por um índice, permitindo acesso direto e eficiente.
- O tamanho do vetor é fixo, sendo definido no momento de sua declaração.
- Para acessar um elemento, utiliza-se seu índice correspondente, que geralmente começa em 0 e vai até tamanho − 1.

Vetores

INSTITUTO FEDERAL
Pernambuco

- Declaração e inicialização de um vetor de inteiros com 5 elementos

```
vetor: array[0..4] of integer
```

- Inicialização dos elementos do vetor:

```
vetor[0] := 10
vetor[1] := 20
vetor[2] := 30
vetor[3] := 40
vetor[4] := 50
```

Vetores

INSTITUTO FEDERAL
Pernambuco

- Declaração e inicialização de um vetor de inteiros com 5 elementos

vetor: array[0..4] of integer

- Inicialização dos elementos do vetor:

vetor[0] := 10

vetor[1] := 20

vetor[2] := 30

vetor[3] := 40

vetor[4] := 50

```c
#include <stdio.h>

int main() {
    int vetor[5];

    vetor[0] = 10;
    vetor[1] = 20;
    vetor[2] = 30;
    vetor[3] = 40;
    vetor[4] = 50;

    return 0;
}
```

Exercício rápido

- Sendo a, b e o vetor x iguais a:

a = 2 **b =** 4 **x =**

0	1	2	3	4	5	6	7	8	9
2	6	8	3	10	9	1	21	33	14

- Escreva o valor correspondente de:

a) x[a+1] =
b) x[a+2] =
c) x[a+3] =
d) x[a*4] =
e) x[a*1] =
f) x[a*2] =
g) x[a*3] =
h) x[x[a+b]] =
i) x[a+b] =
j) x[8-x[2]] =
k) x[x[4]] =
l) x[x[x[7]]] =
m) x[x[1]*x[4]] =
n) x[x[a+4]] =

Exercício rápido

INSTITUTO FEDERAL
Pernambuco

- Sendo a, b e o vetor x iguais a:

a = 2 **b =** 4 **x =**

0	1	2	3	4	5	6	7	8	9
2	6	8	3	10	9	1	21	33	14

- Escreva o valor correspondente de:

a) x[a+1] = x[2+1] = x[3] = 3
b) x[a+2] = x[2+2] = x[4] = 10
c) x[a+3] = x[2+3] = x[5] = 9
d) x[a*4] = x[2*4] = x[8] = 33
e) x[a*1] = x[2*1] = x[2] = 8
f) x[a*2] = x[2*2] = x[4] = 10
g) x[a*3] = x[2*3] = x[6] = 1
h) x[x[a+b]] = x[x[2+4]] = x[x[6]] = x[1] = 6
i) x[a+b] = x[2+4] = x[6] = 1
j) x[8-x[2]] = x[8-8] = x[0] = 2
k) x[x[4]] = x[10] = Inválido
l) x[x[x[7]]] = x[x[21]] = Inválido
m) x[x[1]*x[4]] = x[6*10] = x[60] = Inválido
n) x[x[a+4]] = x[x[2+4]] = x[x[6]] = x[1] = 6

Vetores

INSTITUTO FEDERAL
Pernambuco

- Declaração de um vetor em C:
  - <tipo> identificador [tamanho];

```
int w[4];
```

  - <tipo> identificador [] = { valor1, valor2, ... };

```
int v[] = {1, 2, 3, 4};
```

Vetores

INSTITUTO FEDERAL
Pernambuco

- Declaração de um vetor em C:
  - <tipo> identificador [tamanho];

```
int w[4];
```

  - <tipo> identificador [] = { valor1, valor2, ... };

```
int v[] = {1, 2, 3, 4};
```

Quando o tamanho não é especificado, o compilador o determina automaticamente com base na quantidade de elementos fornecidos no inicializador.

Vetores

INSTITUTO FEDERAL
Pernambuco

- vetor: array[0..4] of integer = [10, 20, 30, 40, 50]

    Para cada elemento em vetor:

        Faça algo com o elemento

- Percorrendo o vetor e imprimindo cada elemento:

    Para cada elemento em vetor:

        Imprimir elemento

Vetores

```c
#include <stdio.h>

int main() {

    int vetor[5] = {10, 20, 30, 40, 50};

    printf("Percorrendo o vetor e dobrando cada elemento:\n");
    for (int i = 0; i < 5; i++) {
        int resultado = vetor[i] * 2;
        printf("Elemento %d dobrado = %d\n", vetor[i], resultado);
    }

    printf("\nPercorrendo o vetor e imprimindo cada elemento:\n");
    for (int i = 0; i < 5; i++) {
        printf("vetor[%d] = %d\n", i, vetor[i]);
    }

    return 0;
}
```

INSTITUTO FEDERAL
Pernambuco

Vetores

INSTITUTO FEDERAL
Pernambuco

```
Percorrendo o vetor e dobrando cada elemento:
Elemento 10 dobrado = 20
Elemento 20 dobrado = 40
Elemento 30 dobrado = 60
Elemento 40 dobrado = 80
Elemento 50 dobrado = 100

Percorrendo o vetor e imprimindo cada elemento:
vetor[0] = 10
vetor[1] = 20
vetor[2] = 30
vetor[3] = 40
vetor[4] = 50
```

Exercícios rápidos

INSTITUTO FEDERAL
Pernambuco

1. Leia 5 números inteiros e armazene em um vetor. Mostre a soma de todos os elementos.

2. Leia 10 números inteiros para um vetor. Mostre o maior valor armazenado.

3. Leia 7 números inteiros e depois um número X. Verifique se X está no vetor e informe a posição (índice). Caso não exista, mostre “não encontrado”.

Matrizes

INSTITUTO FEDERAL
Pernambuco

- Enquanto um vetor possui apenas uma dimensão, uma matriz pode armazenar informações em múltiplas dimensões, como as matrizes bidimensionais, tridimensionais e até estruturas com mais dimensões.
- Em uma matriz bidimensional, cada elemento é acessado por dois índices, que representam sua linha e coluna.
- O tamanho da matriz é definido no momento de sua declaração, indicando o número de linhas e colunas (ou dimensões adicionais, se houver).
- De maneira conceitual, uma matriz pode ser vista como um vetor de vetores, em que cada vetor interno corresponde a uma linha da matriz.

Matrizes

INSTITUTO FEDERAL
Pernambuco

- Declaração e inicialização de uma matriz 2x3 de inteiros:

```
matriz: array[0..1, 0..2] of integer
```

- Inicialização dos elementos da matriz:

```
matriz[0][0] := 1
matriz[0][1] := 2
matriz[0][2] := 3
matriz[1][0] := 4
matriz[1][1] := 5
matriz[1][2] := 6
```

Matrizes

INSTITUTO FEDERAL
Pernambuco

- Declaração e inicialização de uma matriz 2x3 de inteiros:

matriz: array[0..1, 0..2] of integer

- Inicialização dos elementos da matriz:

matriz[0][0] := 1
matriz[0][1] := 2
matriz[0][2] := 3
matriz[1][0] := 4
matriz[1][1] := 5
matriz[1][2] := 6

```c
#include <stdio.h>

int main() {
    int matriz[2][3];

    matriz[0][0] = 1;
    matriz[0][1] = 2;
    matriz[0][2] = 3;

    matriz[1][0] = 4;
    matriz[1][1] = 5;
    matriz[1][2] = 6;

    return 0;
}
```

Vetores

INSTITUTO FEDERAL
Pernambuco

- Declaração de uma matriz em C:
  - <tipo> identificador[linhas][colunas];

```
int m[3][4];
```

  - <tipo> identificador[linhas][colunas] = { valores, valores, ... };

```
int matriz[2][3] = {1, 2, 3, 4, 5, 6};
```

  - <tipo> identificador[][colunas] = { valores, valores, ... };;

```
int matriz[][3] = {1, 2, 3, 4, 5, 6};
```

Vetores

INSTITUTO FEDERAL
Pernambuco

- Declaração de uma matriz em C:
  - <tipo> identificador[linhas][colunas] = { { valores }, { valores }, ... };

```
int matriz[2][3] = {{1, 2, 3}, {4, 5, 6}};
```

  - <tipo> identificador[][colunas] = { { valores }, { valores }, ... };

```
int matriz[][3] = {{1, 2, 3}, {4, 5, 6}};
```

Matrizes

- matriz: array[0..2][0..2] of integer

```
Para cada linha em matriz:
    Para cada coluna em linha:
        Faça algo com o elemento
```

Matrizes

```c
int main() {
    int matriz[2][3];

    matriz[0][0] = 1;
    matriz[0][1] = 2;
    matriz[0][2] = 3;

    matriz[1][0] = 4;
    matriz[1][1] = 5;
    matriz[1][2] = 6;

    printf("Matriz 2x3:\n");
    for (int i = 0; i < 2; i++) {
        for (int j = 0; j < 3; j++) {
            printf("%d ", matriz[i][j]);
        }
        printf("\n");
    }
    return 0;
}
```

Matrizes

```
Matriz 2x3:
1 2 3
4 5 6
```

```c
int main() {
    int matriz[2][3];

    matriz[0][0] = 1;
    matriz[0][1] = 2;
    matriz[0][2] = 3;

    matriz[1][0] = 4;
    matriz[1][1] = 5;
    matriz[1][2] = 6;

    printf("Matriz 2x3:\n");
    for (int i = 0; i < 2; i++) {
        for (int j = 0; j < 3; j++) {
            printf("%d ", matriz[i][j]);
        }
        printf("\n");
    }
    return 0;
}
```

INSTITUTO FEDERAL
Pernambuco

Exercícios

INSTITUTO FEDERAL
Pernambuco

1. Leia uma matriz 3x3 de inteiros. Mostre a soma de todos os elementos.
2. Leia uma matriz 4x4. Mostre o maior valor presente na matriz.
3. Crie um algoritmo que receba uma matriz[5][5] e verifique se esta matriz é identidade.
4. Crie um algoritmo que receba uma matriz[5][5] e verifique se esta matriz é triangular superior.

Dúvidas

INSTITUTO FEDERAL
Pernambuco

ALGORITMOS E ESTRUTURA DE DADOS

Curso de Engenharia de Software
Lucas Sampaio Leite