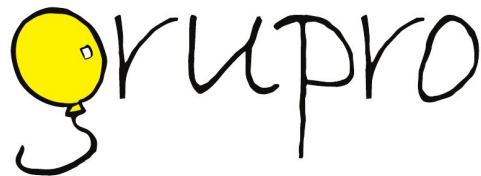




Curso de Programação Nível Básico



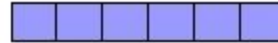
Universidade Federal da Bahia
Instituto de Computação
Departamento de Ciência da Computação

MATRIZES

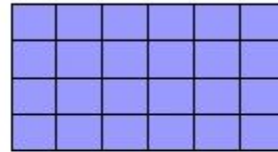
Matrizes

- Matrizes nada mais são do que vetores com múltiplas dimensões.

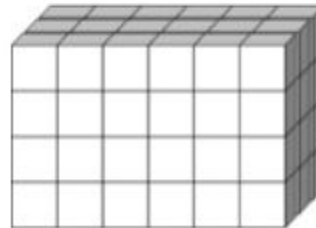
■ Vetor



■ Matriz 2D



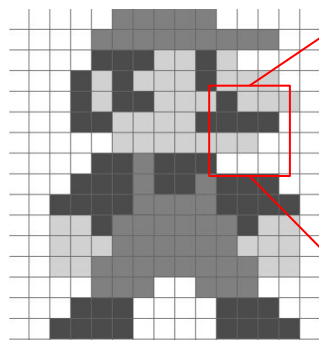
■ Matriz 3D



Aplicações com vetores de diferentes dimensões

Uma imagem em escala de cinza é uma matriz (linhas X colunas) de valores, com cada campo (pixel) contendo um valor de intensidade.

- Edições, filtros de efeitos e reconhecimento de padrões em imagens costumam analisar/alterar o valor de um pixel em relação aos seus vizinhos em uma matriz



L
I
N
H
A
S

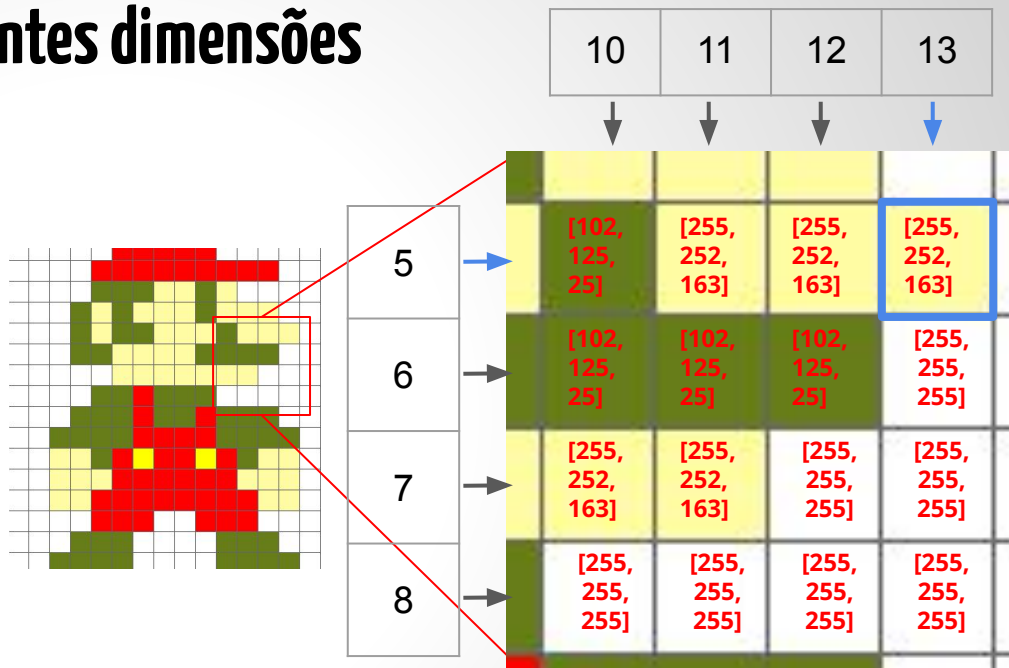
COLUMNAS

209	75	209	209	
75	75	75	75	
209	209	209	255	
75	255	255	255	

Aplicações com vetores de diferentes dimensões

Uma imagem colorida é uma matriz com 3 dimensões, pois cada campo possui valores de RGB (vermelho, verde e azul).

- Para mudar a cor do pixel na ponta do nariz, teríamos que acessar a matriz na linha 5, coluna 13 e alterar os 3 valores dessa posição.

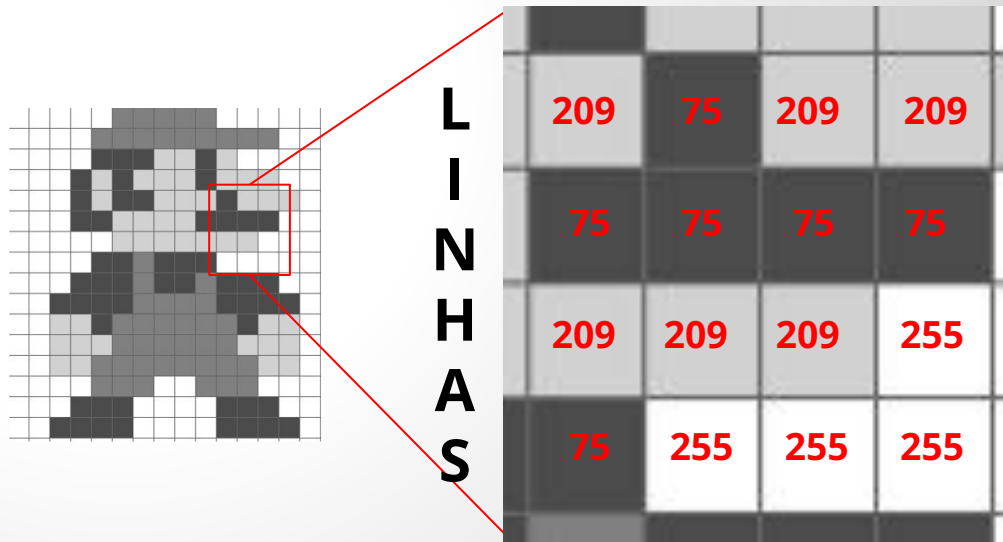


Matrizes

- A declaração de uma matriz é semelhante a de um vetor, porém deve-se adicionar as novas dimensões:

```
int jogo[4];
```

```
int jogo[4][4];
```



Matrizes

- No caso de uma matriz 2D, o primeiro tamanho corresponde à quantidade de linhas, e o segundo à quantidade de colunas.

```
int jogo[4][4];
```

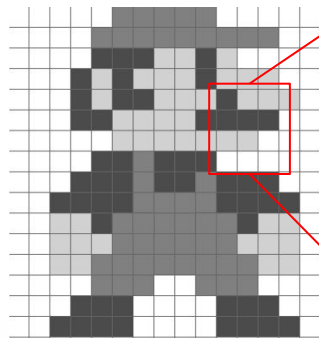
```
jogo[0][0] = 209;
```

```
jogo[1][2] = 75;
```

```
jogo[2][3] = 255;
```

```
cin >> mat[1][4];
```

```
//CUIDADO, ÍNDICE FORA DA MATRIZ
```



L
I
N
H
A
S

COLUMNAS

209	75	209	209
75	75	75	75
209	209	209	255
75	255	255	255

Matrizes

- Como declarar as matrizes multidimensionais:
 - tipo nome[tamanho1][tamanho2][tamanho3]....[tamanhoN];

//Cria matriz com zeros

```
int i, j, altura=10, largura=20;
```

```
int mat2D[altura][largura];
```

```
for(i=0; i < altura; i++)
```

```
    for(j=0; j < largura; j++)
```

```
        mat2D[ i ][ j ] = 0;
```

OU

//Lê matriz da entrada padrão

```
int i,j, altura, largura;
```

```
cin >> altura >> largura;
```

```
int mat2D[altura][largura];
```

```
for(i=0; i < altura; i++)
```

```
    for(j=0; j < largura; j++)
```

```
        cin >> mat2D[ i ][ j ];
```


Acessando uma linha

```
int i,j, altura, largura;  
cin >> altura >> largura;  
int mat2D[altura][largura];  
for(i=0; i < altura; i++)  
    for(j=0; j < largura; j++)  
        cin >> mat2D[ i ][ j ];
```

```
int linha = 1;  
for (j=0; j<largura; j++)  
    cout << mat2D[linha][j] << endl;
```

		COLUNAS			
		0	1	2	3
L I N H A S	0	1	2	3	4
	1	5	6	7	8
	2	9	10	11	12

Saída

5
6
7
8

Acessando uma coluna

```
int i,j, altura, largura;  
cin >> altura >> largura;  
int mat2D[altura][largura];  
for(i=0; i < altura; i++)  
    for(j=0; j < largura; j++)  
        cin >> mat2D[ i ][ j ];
```

```
int coluna = 0;  
for (i=0; i<altura; i++)  
    cout << mat2D[i][coluna] << endl;
```

		COLUNAS			
		0	1	2	3
LINHAS	0	1	2	3	4
	1	5	6	7	8
	2	9	10	11	12

Saída:

```
1  
5  
9
```

Imprimindo matriz - elemento a elemento, linha por linha

```
int i,j, altura, largura;  
cin >> altura >> largura;  
int mat2D[altura][largura];  
for(i=0; i < altura; i++)  
    for(j=0; j < largura; j++)  
        cin >> mat2D[ i ][ j ];  
for (i=0; i<altura; i++){  
    for (j=0; j<largura; j++)  
        cout << mat2D[i][j] << " ";  
    cout << endl;  
}
```

		COLUNAS			
		0	1	2	3
L I N H A S	0	1	2	3	4
	1	5	6	7	8
	2	9	10	11	12

Saída:

```
1 2 3 4  
5 6 7 8  
9 10 11 12
```

Imprimindo matriz - elemento a elemento, coluna por coluna

```
int i,j, altura, largura;  
cin >> altura >> largura;  
int mat2D[altura][largura];  
for(i=0; i < altura; i++)  
    for(j=0; j < largura; j++)  
        cin >> mat2D[ i ][ j ];  
for (j=0; j<largura; j++){  
    for (i=0; i<altura; i++)  
        cout << mat2D[i][j] << " ";  
    cout << endl;  
}
```

Saída:

```
1 5 9  
2 6 10  
3 7 11  
4 8 12
```



Me Add Aí

- Descrição
 - Calcule a soma de duas matrizes.
- Entrada
 - Na primeira linha o número de linhas e o de colunas das matrizes. Nas N próximas linhas os elementos da primeira matriz. Após essas N linhas, temos outras N linhas com os elementos da segunda matriz.
- Saída
 - N linhas contendo os elementos da matriz que resulta da soma das duas matrizes lidas.

Me Add Aí

```
#include <iostream>
using namespace std;
int main() {
    int N, M, i, j;
    cin >> N >> M;
    int A[N][M], B[N][M], C[N][M];
    // Lê a matriz A
    for (i=0; i < N; i++)
        for (j=0; j < M; j++)
            cin >> A[i][j];
    // Lê a matriz B
    for (i=0; i < N; i++)
        for (j=0; j < M; j++)
            cin >> B[i][j];
```

```
// Calcula a matriz C
for (i=0; i < N; i++)
    for (j=0; j < M; j++)
        C[i][j] = A[i][j] + B[i][j];
// Imprime a matriz C
for (i=0; i < N; i++) {
    for (j=0; j < M-1; j++)
        cout << C[i][j] << " ";
    cout << C[i][j] << endl;
}
}
```