



Processamento da Informação

Matrizes – Parte 1

Prof. Jesús P. Mena-Chalco
CMCC/UFABC

Q2/2018

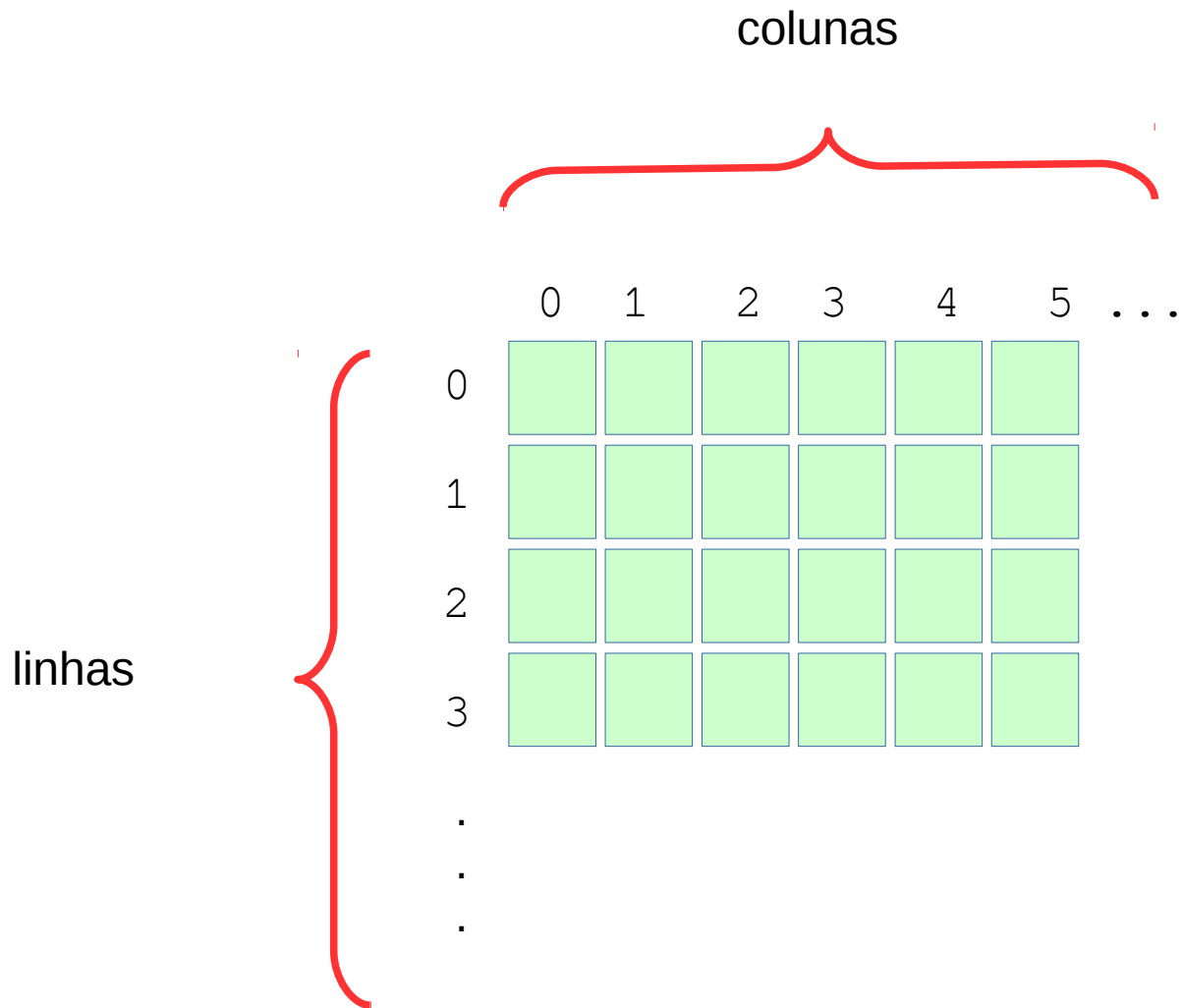
Matriz bidimensional em Java

Declaração de uma variável que representa uma **matriz bidimensional de inteiros** (10 linhas e 20 colunas)

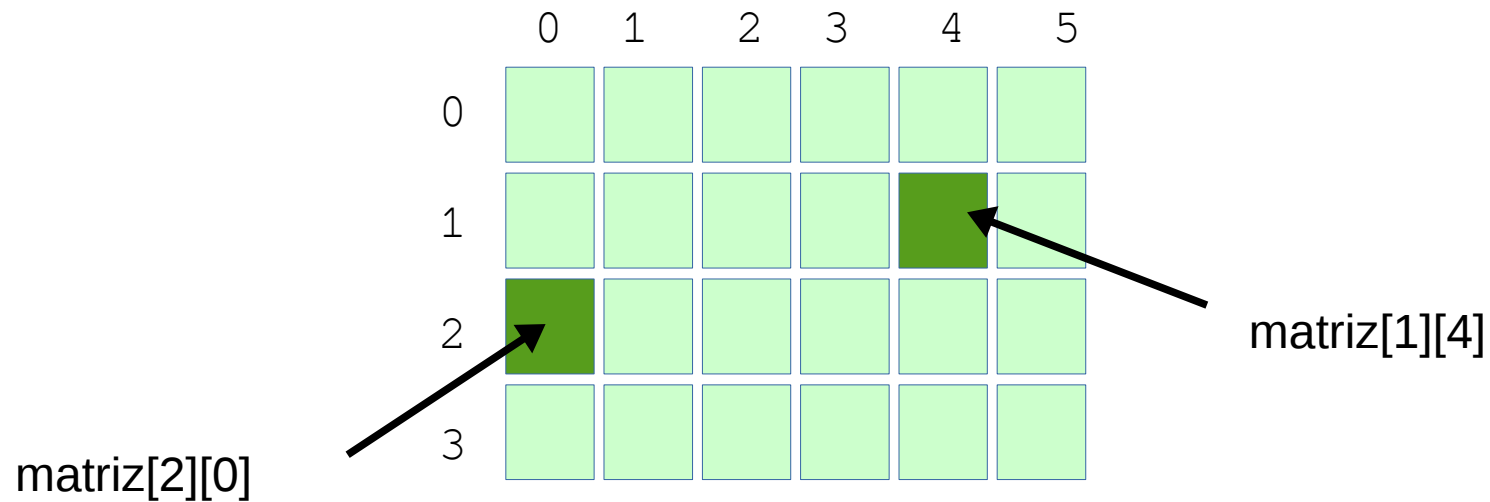
```
int matriz[][];  
matriz = new int[10][20];
```

```
int matriz[][] = new int[10][20];
```

Matriz bidimensional em Java

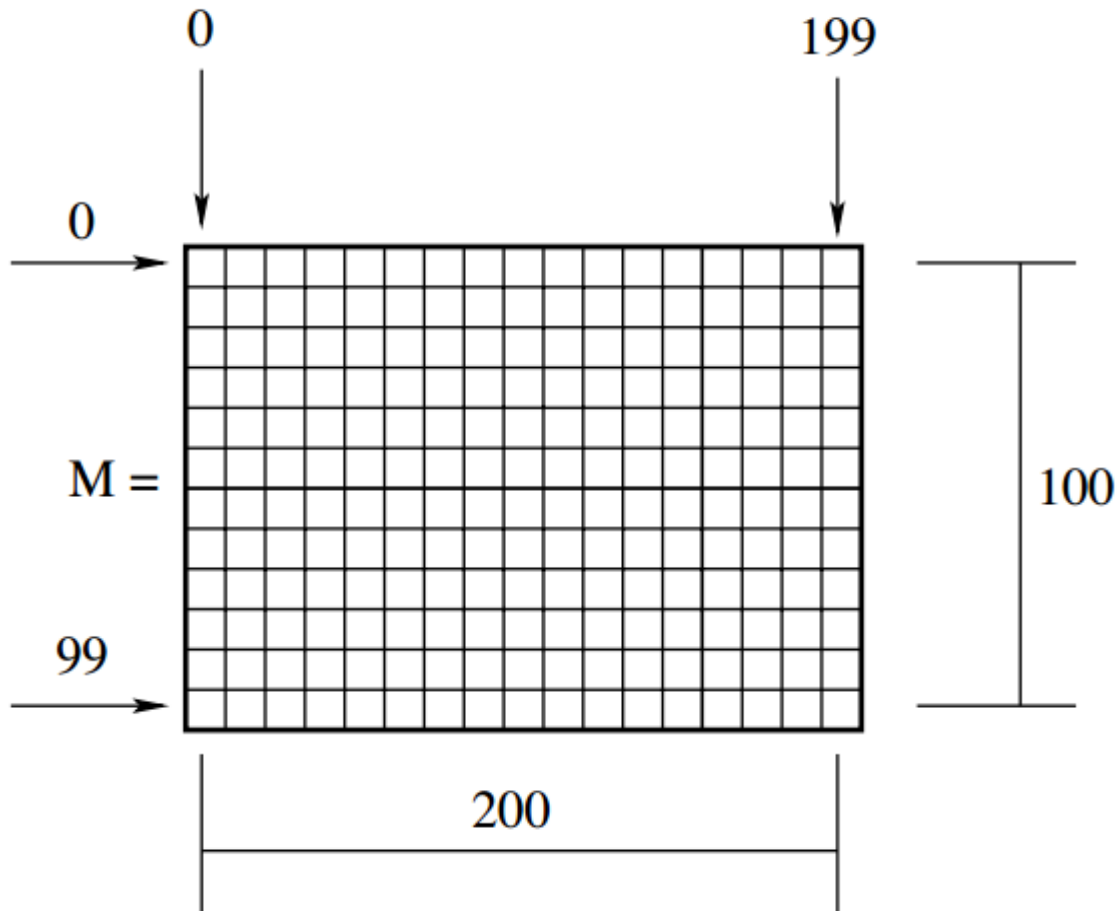


Matriz bidimensional em Java



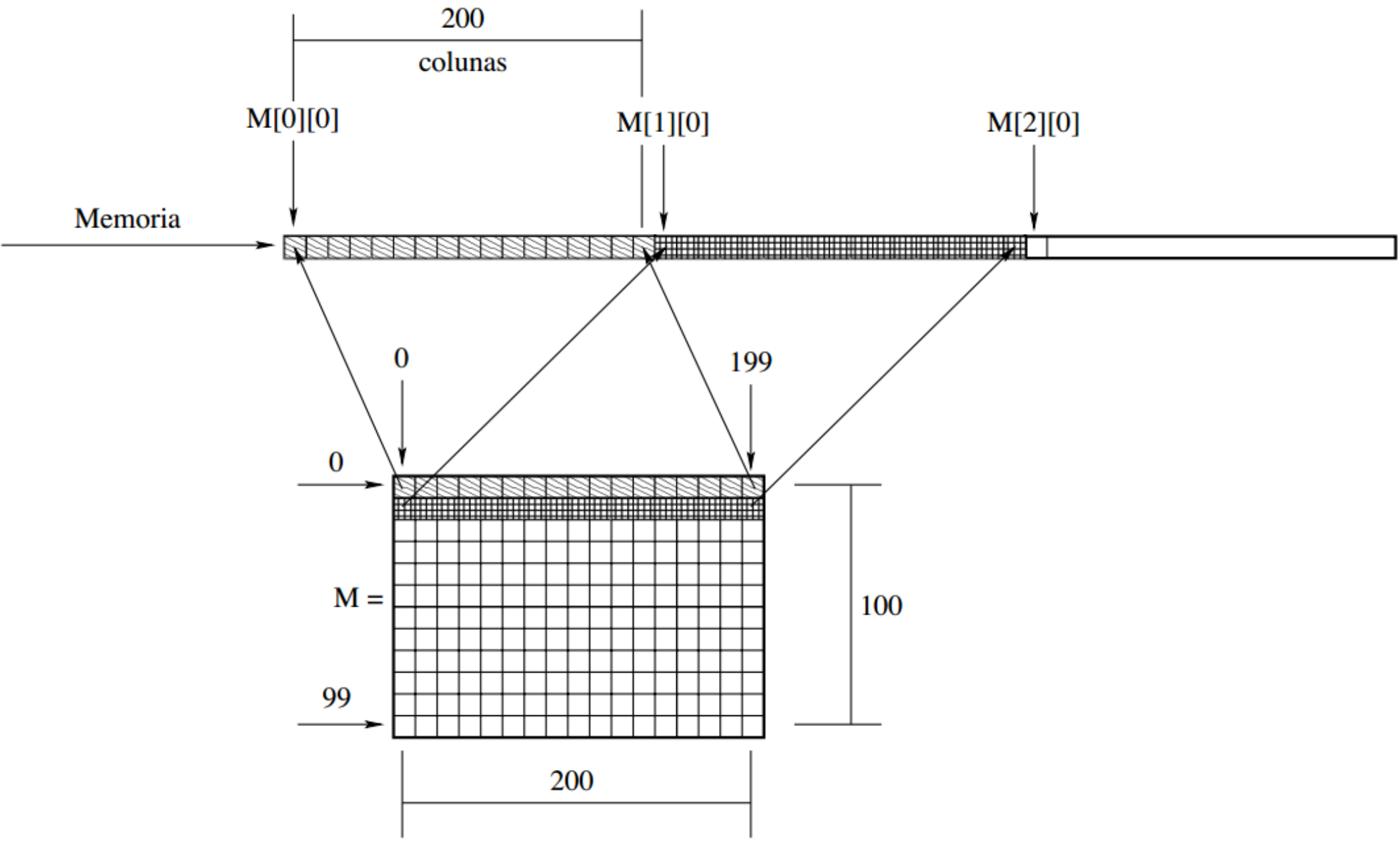
```
int M[][] = new int[100][200];
```

Declara uma matriz M
de 100 linhas
com 200 colunas
(20mil inteiros)

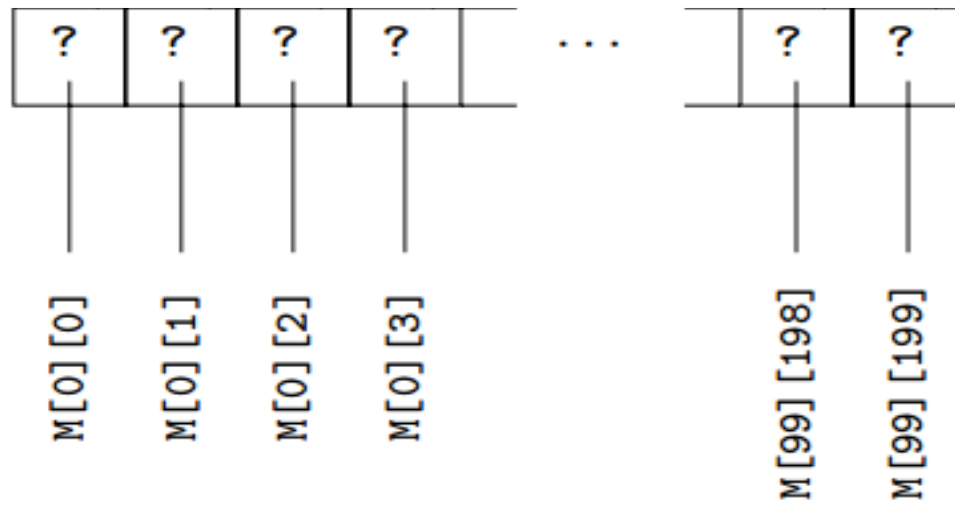


A memória do computador é linear!

Estrutura da matriz na memória do computador



Disposição dos 20mil elementos da matriz M na memória

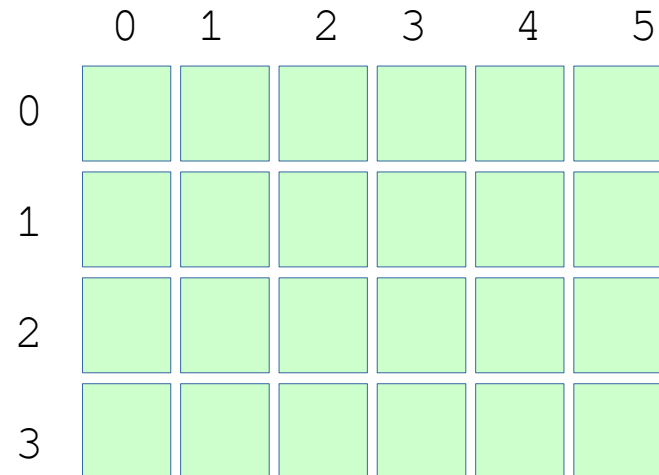


Erros comuns

- Considerar índices negativos
- Considerar índices que ultrapassam as dimensões da matriz

```
int M[] [] = new int [4] [6]
```

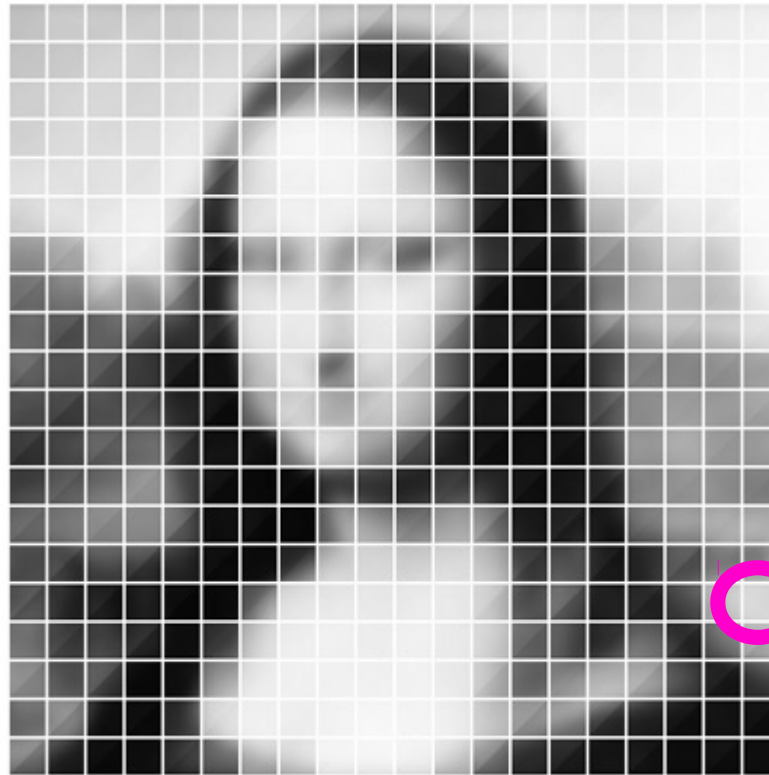
matriz[-1][4]



matriz[4][6]

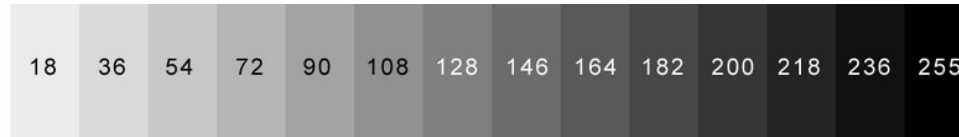


Matriz bidimensional (imagem em níveis de cinza)

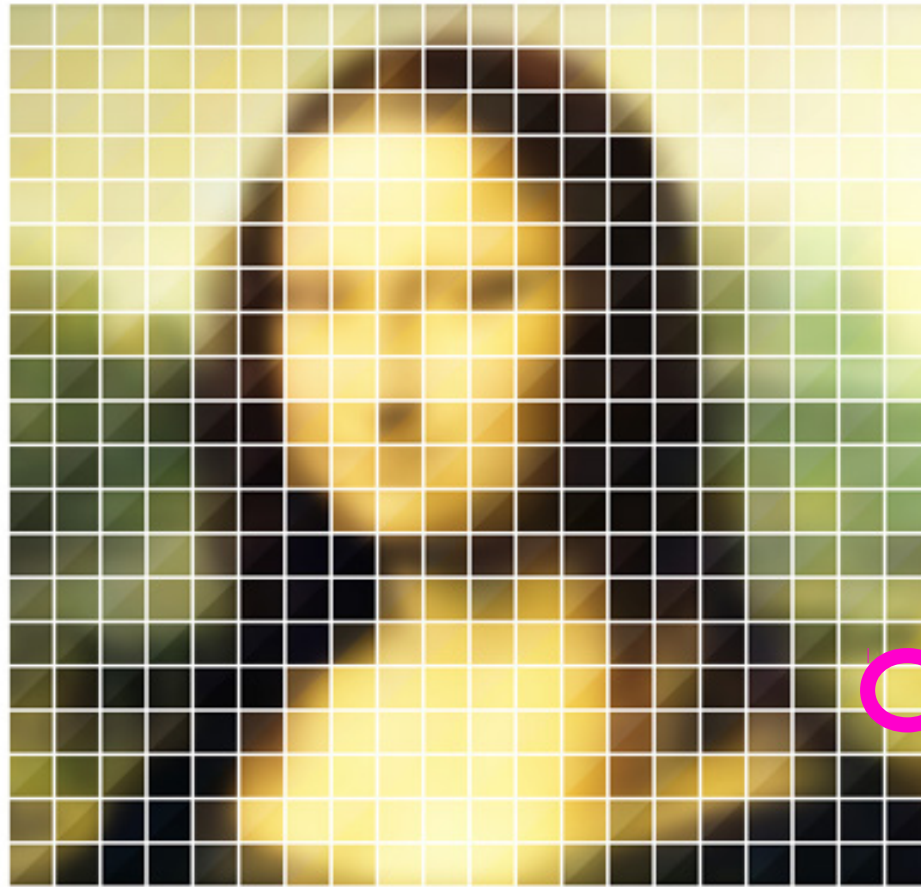


Nível=18

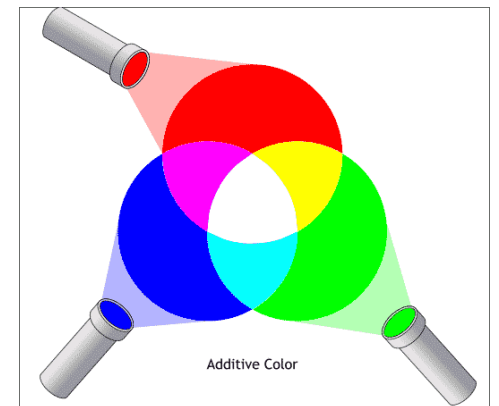
0 18 36 54 72 90 108 128 146 164 182 200 218 236 255



Matriz tridimensional (imagem em RGB)

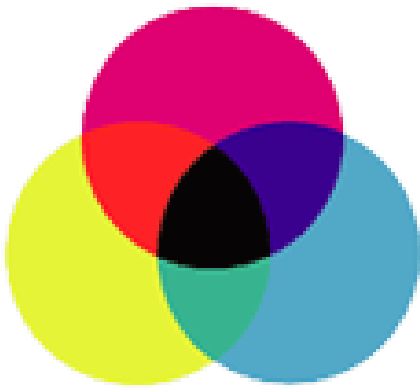


→ {Red, Green, Blue}



Sistema de cor?

Quando falamos
de cor, falamos
de luz



SISTEMA SUBTRATIVO

Sistema CMYK
(impressoras)

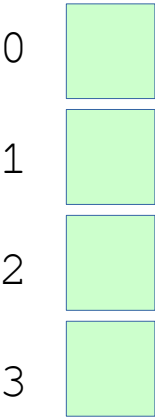


SISTEMA ADITIVO

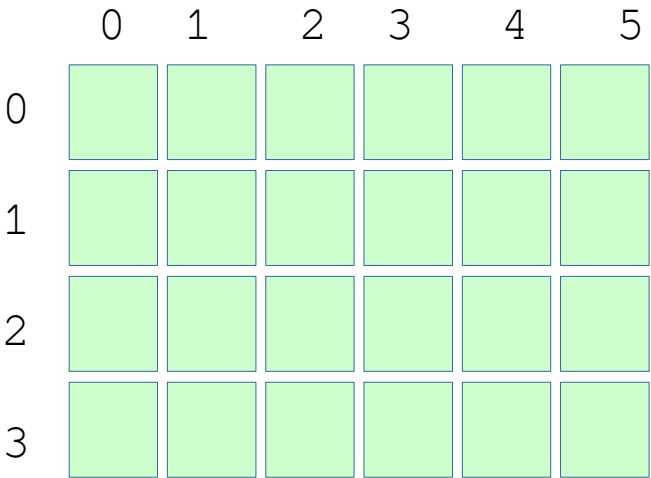
Sistema RGB
(projektor)

Matrizes

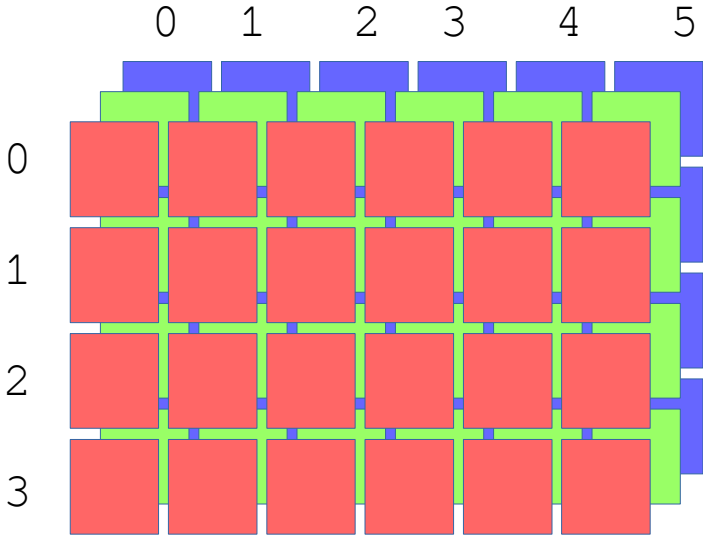
**Matriz unidimensional
(vetor/Array)**



**Matriz bidimensional
(2D)**

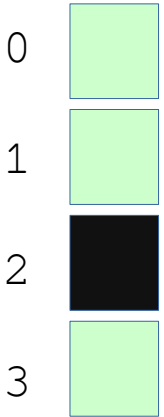


**Matriz tridimensional
(3D)**



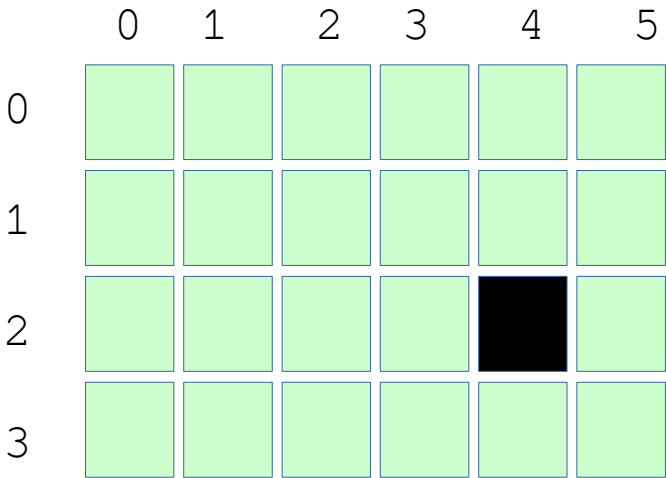
Matrizes

Matriz unidimensional (vetor/Array)



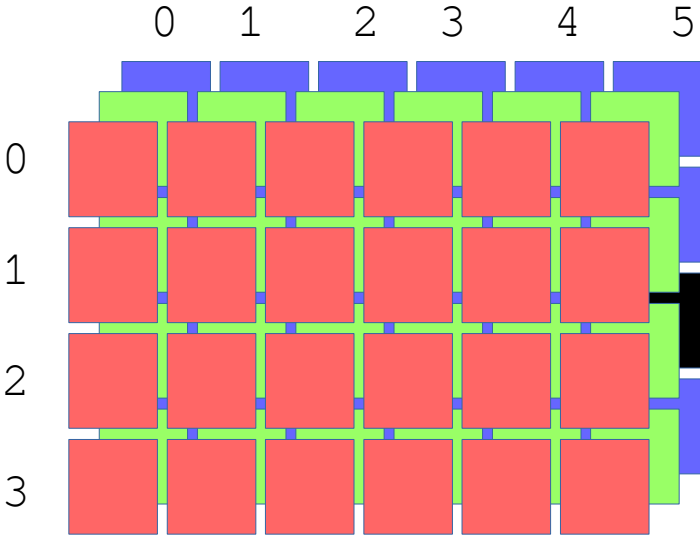
`M[2]`

Matriz bidimensional (2D)



`M[2][4]`

Matriz tridimensional (3D)



`M[2][5][2]`

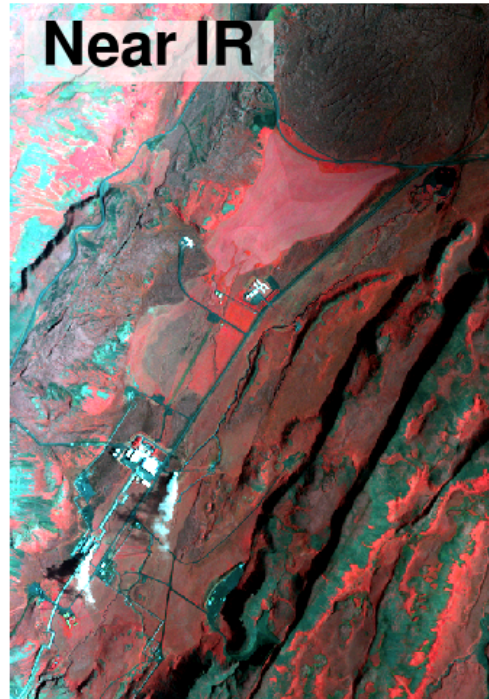
**Multi-Spectral Imaging
Documentation**



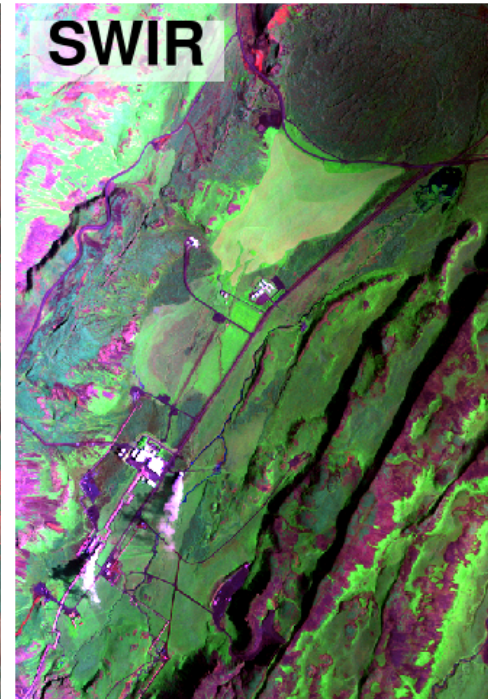
www.antoninocosentino.it



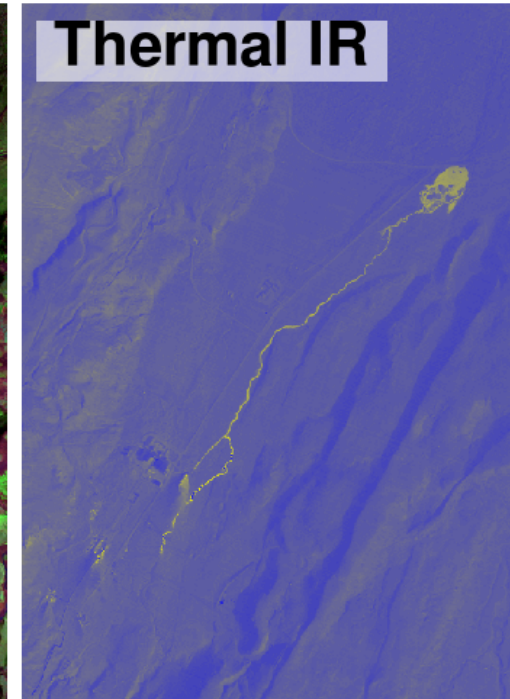
Visible



Near IR



SWIR



Thermal IR

Processamento de imagens multiespectrais (imagens do mesmo objeto, mas capturadas com diferentes comprimentos de onda **eletromagnéticas**)

Os aparelhos consideram diferentes comprimentos de onda para cada camada.

<http://all-geo.org/volcan01010/2013/01/processing-arsf-remote-sensing-data-with-open-source-gis-tools/>

Exemplo 01: Definição

```
int M[][] = new int[5][10];  
  
System.out.println(M.length);  
System.out.println(M[0].length);  
System.out.println(M[1].length);  
System.out.println(M[2].length);  
System.out.println(M[3].length);  
System.out.println(M[4].length);
```

```
5  
10  
10  
10  
10  
10
```


Exemplo 02: Listando os elementos

```
int i, j;
int M[][] = new int[5][10];

for (i=0; i<M.length; i=i+1) {
    for (j=0; j<M[0].length; j=j+1) {
        System.out.print(M[i][j]+" ");
    }
}
```

00

Exemplo 02: Listando os elementos

```
int i, j;  
int M[][] = new int[5][10];  
  
for (i=0; i<M.length; i=i+1) {  
    for (j=0; j<M[0].length; j=j+1) {  
        System.out.print(M[i][j]+" ");  
    }  
    System.out.print("\n");  
}
```

```
0000000000  
0000000000  
0000000000  
0000000000  
0000000000
```

Exemplo 03: Atribuindo valores

```
int i, j;
int M[][] = new int[5][10];

M[3][3] = 33;
M[4][7] = 47;
M[4][9] = 49;

for (i=0; i<M.length; i=i+1) {
    for (j=0; j<M[0].length-1; j=j+1) {
        System.out.print(M[i][j]+" ");
    }
    System.out.print(M[i][j)+"\n");
}
```

	0	1	2	3	4	5	6	7	8	9
0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0
3	0	0	0	33	0	0	0	0	0	0
4	0	0	0	0	0	0	0	47	0	49

Exemplo 04: Matriz identidade de ordem **n**

Vamos desenvolver um método que permita criar uma matriz identidade de **ordem n**.

Uma matriz identidade é uma matriz quadrada, onde todos os elementos da diagonal principal é 1, os demais elementos são 0s.

Assinatura:

```
static int[] [] matrizIdentidade( int n )
```

$$I_1 = [1], I_2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, I_3 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \dots, I_n = \begin{bmatrix} 1 & 0 & 0 & \dots & 0 \\ 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 1 \end{bmatrix}$$

Exemplo 04: Matriz identidade de ordem **n**

```
static int[][] matrizIdentidade( int n ) {  
    int i, j;  
    int M[][] = new int[n][n];
```

Exemplo 04: Matriz identidade de ordem **n**

```
static int[][] matrizIdentidade( int n ) {
    int i, j;
    int M[][] = new int[n][n];

    for (i=0; i<n; i=i+1) {
        for (j=0; j<n; j=j+1) {
            if (i==j) {
                M[i][j] = 1;
            }
            else {
                M[i][j] = 0;
            }
        }
    }

    return M;
}
```

Exemplo 04: Matriz identidade de ordem **n**

```
public static void main(String[] args)
{
    int i, j;
    int [][]M = matrizIdentidade(4);

    for (i=0; i<M.length; i=i+1) {
        for (j=0; j<M[0].length-1; j=j+1) {
            System.out.print(M[i][j]+" ");
        }
        System.out.print(M[i][j)+"\n");
    }
}
```

```
1 0 0 0
0 1 0 0
0 0 1 0
0 0 0 1
```

Exemplo 05: Matriz diagonal

Vamos desenvolver um método que permita **verificar** se uma matriz é diagonal.

Uma matriz diagonal é uma matriz quadrada onde os elementos que não pertencem à diagonal principal **são obrigatoriamente** iguais a zero.

Assinatura:

```
static boolean ehDiagonal( int M[] [] )
```

$$\begin{pmatrix} 2 & 0 & 0 & 0 & 0 \\ 0 & -3 & 0 & 0 & 0 \\ 0 & 0 & 7 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 11 \end{pmatrix}$$

Pode ser também `[][]M`

Exemplo 05: Matriz diagonal

```
static boolean ehDiagonal( int [][]M ) {
    int i, j;
    int n = M.length;
    int m = M[0].length;

    if (n!=m) {
        return false;
    }
    else {
        .
        .
        .
    }
}
```

Exemplo 05: Matriz diagonal

```
static boolean ehDiagonal( int [][]M ) {
    int i, j;
    int n = M.length;
    int m = M[0].length;

    if (n!=m) {
        return false;
    }
    else {
        for (i=0; i<n; i=i+1) {
            for (j=0; j<n; j=j+1) {
                if (i!=j && M[i][j]!=0) {
                    return false;
                }
            }
        }
        return true;
    }
}
```

Exemplo 06: Matriz transposta

Vamos desenvolver um método que permita calcular a matriz diagonal de uma matriz dada como entrada.

Assinatura:

```
static int [] [] transposta ( int [] [] M )
```

$$\begin{bmatrix} 6 & 4 & 24 \\ 1 & -9 & 8 \end{bmatrix}^T = \begin{bmatrix} 6 & 1 \\ 4 & -9 \\ 24 & 8 \end{bmatrix}$$

Exemplo 06: Matriz transposta

```
static int[][] transposta( int [][]M ) {
    int i, j;
    int n = M.length;
    int m = M[0].length;
    int [][]R = new int[m][n];

    for (i=0; i<n; i=i+1) {
        for (j=0; j<m; j=j+1) {
            R[j][i] = M[i][j];
        }
    }

    return R;
}
```