

LINGUAGEM DE PROGRAMAÇÃO I

AULA 02
LOOPS E ARRAYs

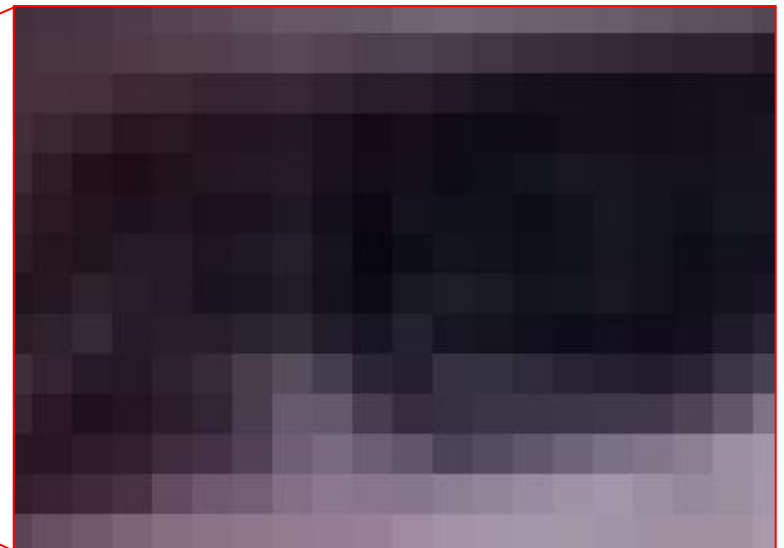
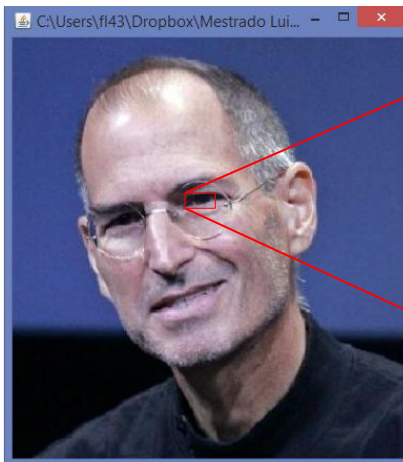
LUIS GUSTAVO ARAUJO

01

REVISÃO DA
AULA ANTERIOR

Sabemos também que o computador é muito bom com números. **Assim foi preciso criar uma forma de representar imagens digitais através de números.**

Imagens digitais são arquivos compostos por uma sequência de pontos (pixels) na vertical e horizontal.

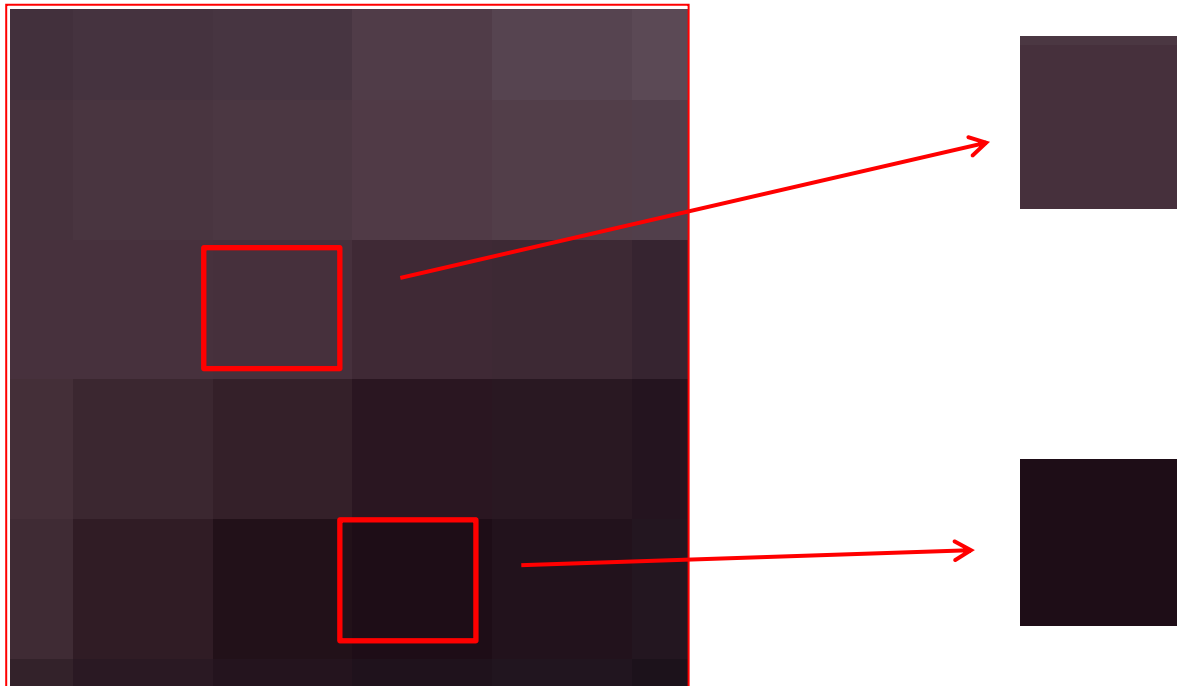


PIXEL:

Pixels são elementos da imagens

Cada pixel sabe sua cor

E também sabe sua posição



MATRIZ DE PIXEL:

Uma imagem é uma matriz de pixels

| | 0 | 1 | 2 | 3 |
|---|----|----|----|----|
| 0 | 15 | 12 | 13 | 10 |
| 1 | 9 | 7 | 2 | 1 |
| 2 | 6 | 3 | 9 | 10 |

REFERENCIANDO A MATRIZ:

As posições na matriz podem ser referenciadas com **x** e **y** ou **horizontal** e **vertical**

O elemento **(1,0)** na matriz **é 12**





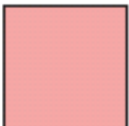
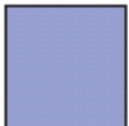


O elemento **(0,2)** na matriz **é 6**

| | 0 | 1 | 2 | 3 |
|---|----|----|----|----|
| 0 | 15 | 12 | 13 | 10 |
| 1 | 9 | 7 | 2 | 1 |
| 2 | 6 | 3 | 9 | 10 |

CODIFICAÇÃO DA COR:

Cada componente da cor (vermelho, azul, verde) é codificado com um byte.

As cores variam de **(0,0,0) preto** até **(255,255,255) branco**. E mudam conforme os valores (a mistura das cores)

| | 0 | 1 | 2 | 3 |
|---|------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------|
| 0 |  255, 30, 30 |  30, 30, 255 |  30, 255, 30 |  0, 0, 0 |
| 1 |  255, 150, 150 |  150, 150, 255 |  150, 255, 150 |  200, 200, 200 |

CODIFICAÇÃO DA COR:

Existem diversas formas de codificação da cor:

CMYK (Ciano, Magenta, Amarelo, Preto)

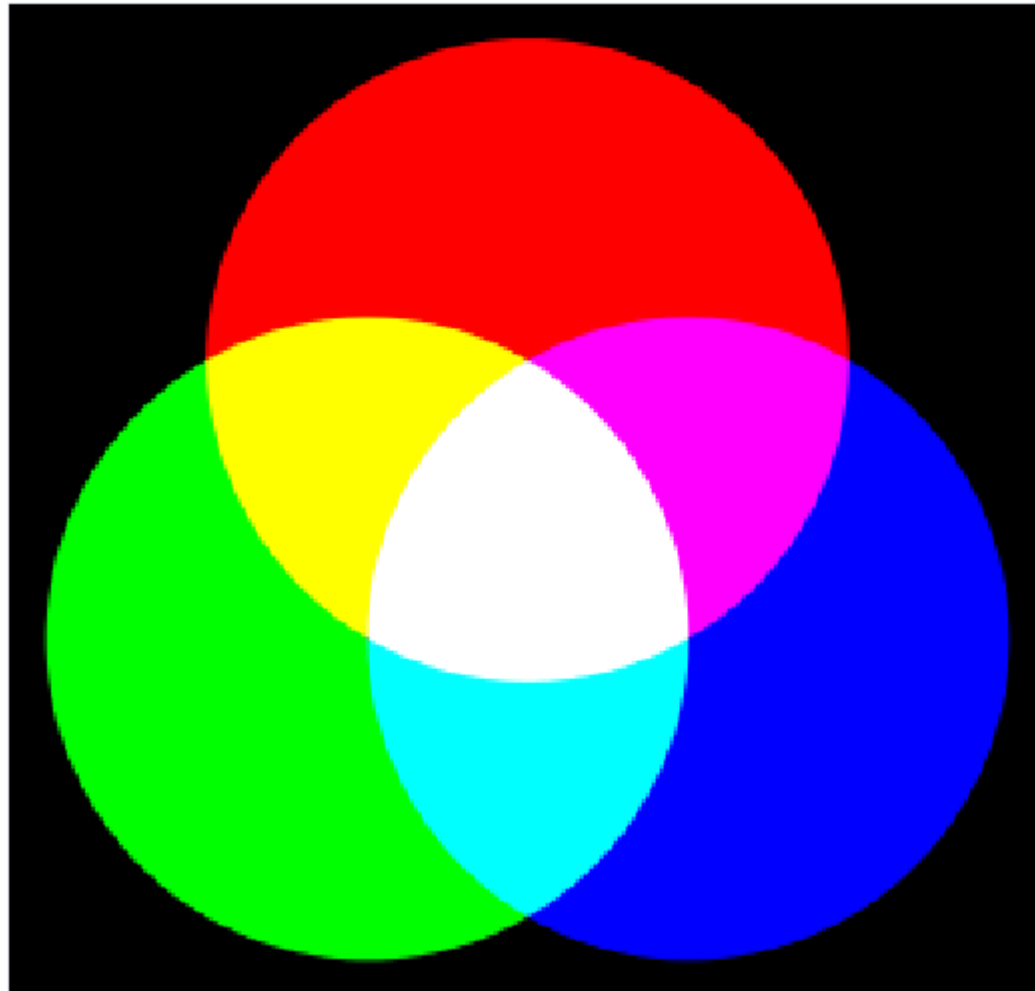
HSB (Matiz, Saturação, Brilho)

HSV (Matiz, Saturação, Valor)

Nós iremos usar o **RBG** (**Red/Vermelho**, **Blue/Azul** e **Green/Vermelho**)

Esse padrão é o usado comumente pelos computadores.

MISTURANDO CORES:



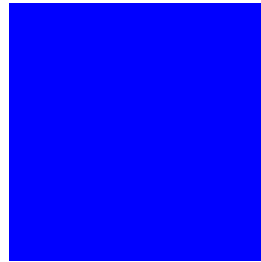
MISTURANDO CORES:

Vermelho

 $(255, 0, 0)$

+

Azul

 $(0, 0, 255)$

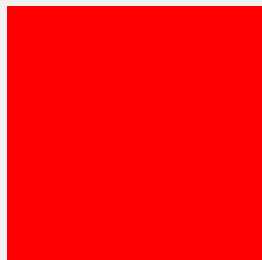
=

Magenta

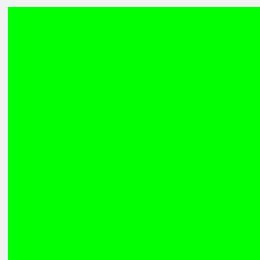
 $(255, 0, 255)$ $(255, 0, 0)$ $(255, 0, 0)$

 $(255, 0, 255)$

Vermelho

 $(255, 0, 0)$

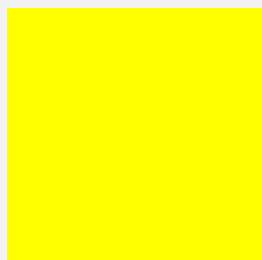
Verde

 $(0, 255, 0)$

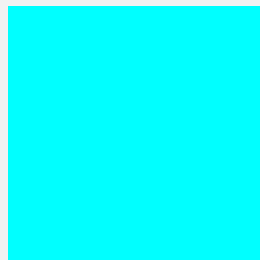
Azul

 $(0, 0, 255)$

Amarelo

 $(255, 255, 0)$

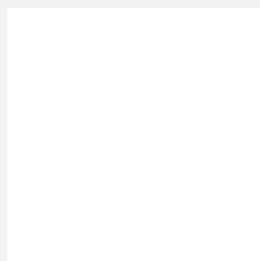
Ciano

 $(0, 255, 255)$

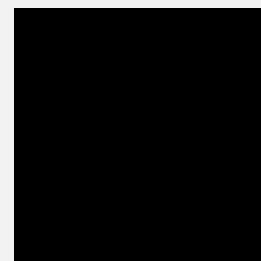
Magenta

 $(255, 0, 255)$

Branco

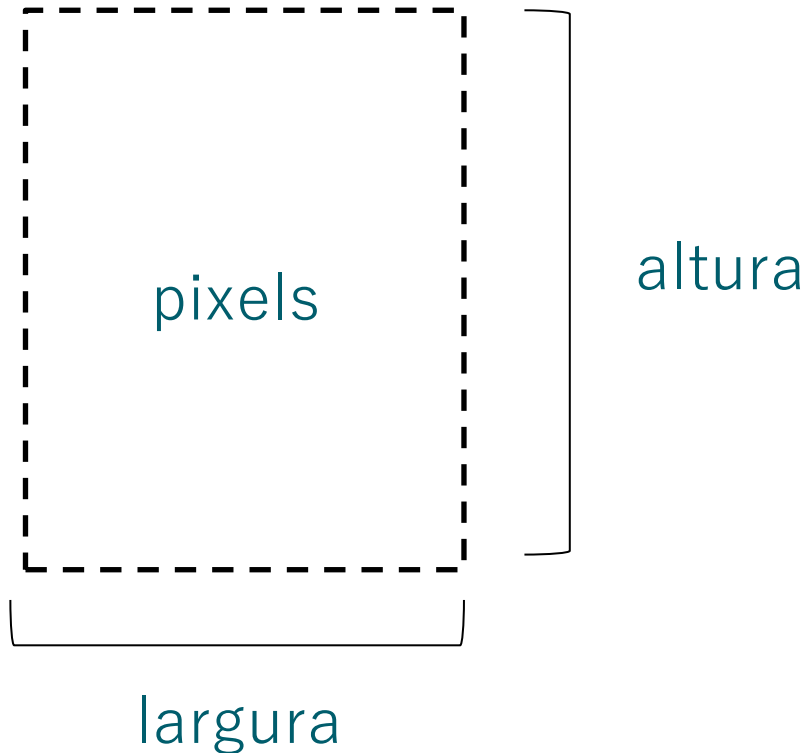
 $(255, 255, 255)$

Preto

 $(0, 0, 0)$

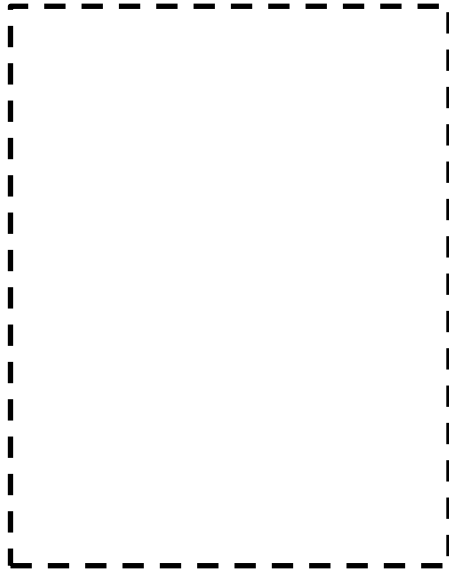
CLASSES (ATRIBUTOS)

Picture não é real, apenas um modelo.



CLASSES (MÉTODOS)

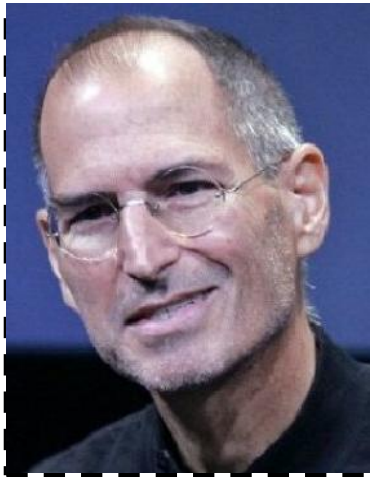
Picture não é real, apenas um modelo.



Eu posso me exibir, posso de
passar os meus pixels, posso
te dizer qual a minha largura
e altura...

OBJETOS

Objetos, ao contrário das Classes, são reais. São implementações das Classes. Todos os **métodos** e **atributos** definidos no modelo, ou classe, estarão no **objeto**.



300 px

100 px

Minha altura é 300 pixel e largura é 100 pixels. O meu pixel na posição 0,0 é azul!

EXIBINDO UMA IMAGEM

Através do objeto do tipo Picture podemos exibir a imagem

```
caminho = FileChooser.pickAFile( )
```

```
imagem = Picture (caminho)
```

```
imagem.show( )
```

