

Desafio pixel syngenta digital

Nome: Luiz Claudio dos Santos Cazarin Junior

Desenvolvimento usando jupyter

Utilizando python e as biblioteca NumPy e matplotlib

```
1 import numpy as np
2 import matplotlib.pyplot as plt
```

Atribuimos a imagem a uma variável usando o método plt.imread()

Passando o arquivo da imagem como parâmetro

```
1 img = plt.imread('Syngenta.bmp')
```

Verificamos o tipo do arquivo

```
1 type(img)
```

Sabendo que é um objeto de array n-dimensional homogêneo

```
1 img.shape
```

Atribuimos o arquivo a uma variável

```
1 a = img[:, :, :]
```

img[linha, coluna, camada]

Usamos a função unique para pegar apenas os valores únicos

```
1 np.unique(a)
```

Descobrimos que existe 4 valores (4 camadas)

Sabemos que as cores são representadas em uma escala de 0 a 255

Sendo 0 = preto e 255 = branco em decimal

Ao analisarmos a imagem em um editor

Descobrimos que a cor do pixel verde contem os seguintes valores #60c000 que são representados em hexadecimal

Podemos representar em #RGB sendo 60 o valor do Red, c0 o valor do Green e 00 o valor do Blue

convertendo os valores retornados da ultima funcao [0, 96, 192, 255] em hexadecimal

Sabemos que os valores sao iguais a 96 = 60, e 192 = c0, ou seja, a combinacao dessas cores forma o pixel verde (#60c000)

Pegando a soma dos valores únicos [0, 96, 192, 255]

```
1 np.unique(a, return_counts=True)
2 print(np.unique(a, return_counts=True))
```

Ao somar os valores dos pixels, retornamos [376006, 298, 298, 127398]

sabendo que o segundo e o terceiro valor são responsáveis pela cor verde

*descobrimos que existem **298** pixels verdes*