

▼ ESTI019 - 3QS2021 - CSM - Minami

Lab3 - Codificação de Imagem por DCT e Animação

Objetivos:

1. Produzir um vídeo de animação para o Grupo com Blender
2. Efetuar conversões entre espaços de cores
3. Comparar arquivos comprimidos JPEG
4. Efetuar compressão de imagem com DCT

```
import librosa
import librosa.display
import IPython.display
from google.colab import drive
ir, sr = librosa.load('/content/drive/My Drive/Colab Notebooks/Lab_3/lab_3.m4a')
IPython.display.Audio(ir, rate = sr)

/usr/local/lib/python3.7/dist-packages/librosa/core/audio.py:165: UserWarning
  warnings.warn("PySoundFile failed. Trying audioread instead.")

0:01 / 2:00
```

1. *Produzir um vídeo de animação para o Grupo com Blender*

Vejam o vídeo de como efetuar uma animação básica usando o Blender (instalação recomendada 2.8):

<https://youtu.be/xz4t1j2-1gs>

Link do vídeo produzido em blender pelo grupo:

<https://drive.google.com/file/d/126HGRnNSOaVVy1ovgkY-Btc6-cdpkiYF/view>

▼ 2. *Efetuar conversões entre espaços de cores*

```
import numpy as np
```

```

import cv2 as cv
import matplotlib.pyplot as plt
from google.colab import drive
from pathlib import Path

from google.colab import drive
drive.mount('/content/drive')

!ls -l '/content/drive/My Drive/Colab Notebooks/Lab_3'    # verifica se montou o dri
#bgr1 = cv.imread('/content/drive/My Drive/Colab Notebooks/Lab_3/messi5.jpg') # lei
bgr1 = cv.imread('/content/drive/My Drive/Colab Notebooks/Lab_3/messi5.jpg')
print(type(bgr1))
altura, largura, camadas = bgr1.shape
print("Resolução: ", largura, " x ", altura, "PIXELS. ", camadas, " camadas.")

Drive already mounted at /content/drive; to attempt to forcibly remount, call
total 12411
-rw----- 1 root root 786486 Dec 17 06:24 'Cópia de lena.bmp'
-rw----- 1 root root 72937 Dec 17 06:24 'Cópia de messi5.jpg'
-rw----- 1 root root 33591 Dec 17 20:34 grupo05.jpg
-rw----- 1 root root 74562 Dec 17 20:34 grupo25.jpg
-rw----- 1 root root 1425085 Dec 17 13:38 grupo.png
-rw----- 1 root root 6822 Dec 17 20:34 jefferson05.jpg
-rw----- 1 root root 17912 Dec 17 20:34 jefferson25.jpg
-rw----- 1 root root 374872 Dec 17 05:14 jefferson.png
-rw----- 1 root root 7798791 Dec 17 20:34 Lab3_Cod_Imagen_por_DCT_e_Animacã
-rw----- 1 root root 7055 Dec 17 20:34 lena05.jpg
-rw----- 1 root root 15815 Dec 17 20:34 lena25.jpg
-rw----- 1 root root 786486 Oct 1 13:28 lena.bmp
-rw----- 1 root root 13636 Dec 17 20:34 lucas05.jpg
-rw----- 1 root root 29011 Dec 17 20:34 lucas25.jpg
-rw----- 1 root root 410282 Dec 17 05:15 lucas.png
-rw----- 1 root root 72937 Oct 4 2019 messi5.jpg
-rw----- 1 root root 6862 Dec 17 20:34 muriel05.jpg
-rw----- 1 root root 12296 Dec 17 20:34 muriel25.jpg
-rw----- 1 root root 330293 Dec 17 05:14 muriel.png
-rw----- 1 root root 7652 Dec 17 20:34 victor05.jpg
-rw----- 1 root root 15610 Dec 17 20:34 victor25.jpg
-rw----- 1 root root 403638 Dec 17 05:14 victor.png
<class 'numpy.ndarray'>
Resolução: 548 x 342 PIXELS. 3 camadas.

```

Separa os canais e re-arranja para formar imagem RGB

```

b1, g1, r1 = cv.split(bgr1)
rgb2 = cv.merge([r1,g1,b1])
# Q1 - O que foi feito aqui?

```

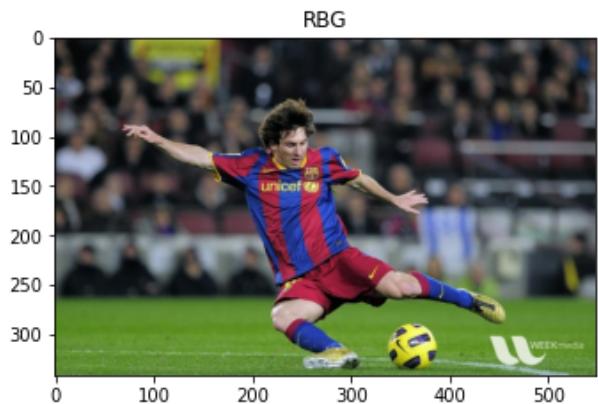
Imprime cores trocadas (BGR) e reais (RGB)

```

plt.figure(figsize=[12, 5])
plt.subplot(121); plt.imshow(bgr1); plt.title('BGR')
plt.subplot(122); plt.imshow(rgb2); plt.title('RBG')

```

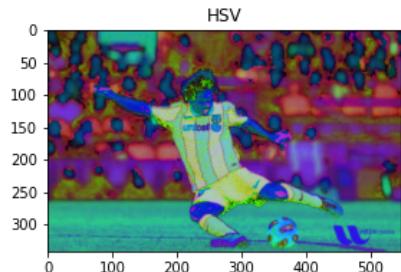
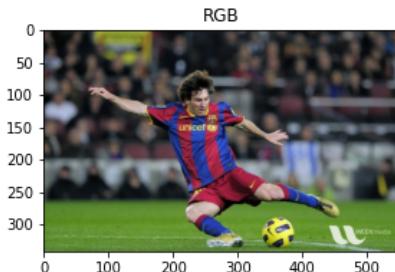
```
Text(0.5, 1.0, 'RGB')
```



Converte para os formatos YCrCb e HSV

```
ycrcb = cv.cvtColor(bgr1, cv.COLOR_BGR2YCrCb)
hsv = cv.cvtColor(bgr1, cv.COLOR_BGR2HSV)
plt.figure(figsize=[15,6])
plt.subplot(131); plt.imshow(rgb2); plt.title('RGB')
plt.subplot(132); plt.imshow(ycrcb); plt.title('YCrCb')
plt.subplot(133); plt.imshow(hsv); plt.title('HSV')
```

```
Text(0.5, 1.0, 'HSV')
```



Separação das Camadas RGB individualmente

```
imageR = rgb2.copy()
imageR[:, :, 1:3] = 0
imageG = rgb2.copy()
imageG[:, :, 0] = 0; imageG[:, :, 2] = 0
imageB = rgb2.copy()
imageB[:, :, 0:2] = 0
# Q2 - O que foi feito aqui?
```

```
plt.figure(figsize=[15,6])
plt.subplot(131); plt.imshow(imageR); plt.title('RGB_Camada R')
plt.subplot(132); plt.imshow(imageG); plt.title('RGB_Camada G')
plt.subplot(133); plt.imshow(imageB); plt.title('RGB_Camada B')
```

```
Text(0.5, 1.0, 'RGB_Camada_B')
```



Separação dos Canais YCbCr

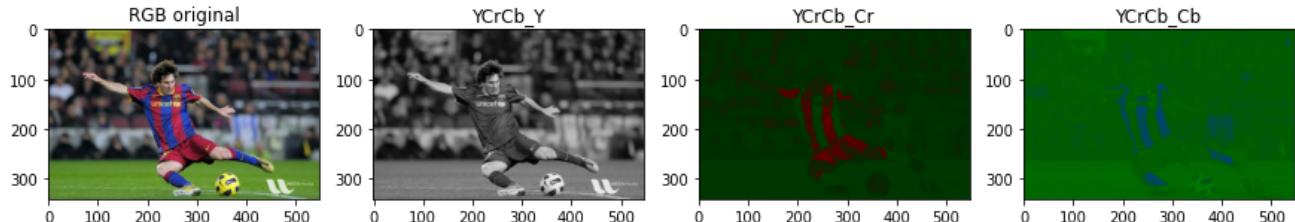
```
y1, cr1, cb1 = cv.split(ycrcb)
imageCR = ycrcb.copy()
imageCR[:, :, 0] = 0
imageCR[:, :, 2] = 0
Cr = cv.cvtColor(imageCR, cv.COLOR_YCrCb2RGB)

imageCB = ycrcb.copy()
imageCB[:, :, 0] = 0
imageCB[:, :, 1] = 0
Cb = cv.cvtColor(imageCB, cv.COLOR_YCrCb2RGB)

plt.figure(figsize=[15, 5])
plt.subplot(141); plt.imshow(rgb2); plt.title('RGB original')
plt.subplot(142); plt.imshow(y1, cmap='gray'); plt.title('YCrCb_Y')
plt.subplot(143); plt.imshow(Cr); plt.title('YCrCb_Cr')
plt.subplot(144); plt.imshow(Cb); plt.title('YCrCb_Cb')
```



```
Text(0.5, 1.0, 'YCrCb_Cb')
```



▼ Com as Imagens do Grupo:

1. Faça o mesmo com uma imagem de cada integrante do grupo e
2. Com a foto montagem de todos os do grupo, lembrando das roupas com cores diferentes, preferencialmente (R, G e B).

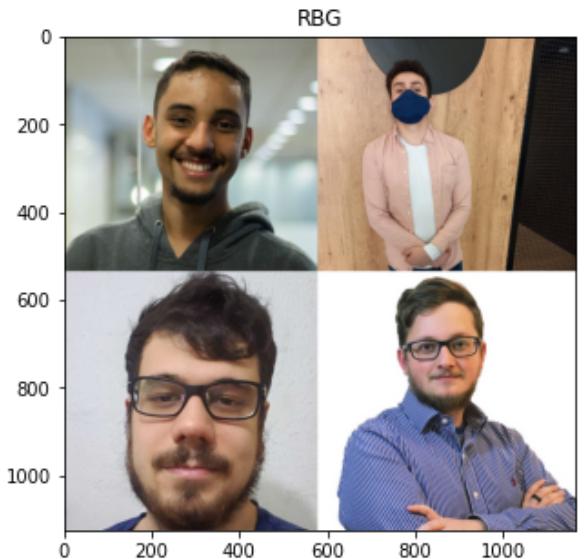
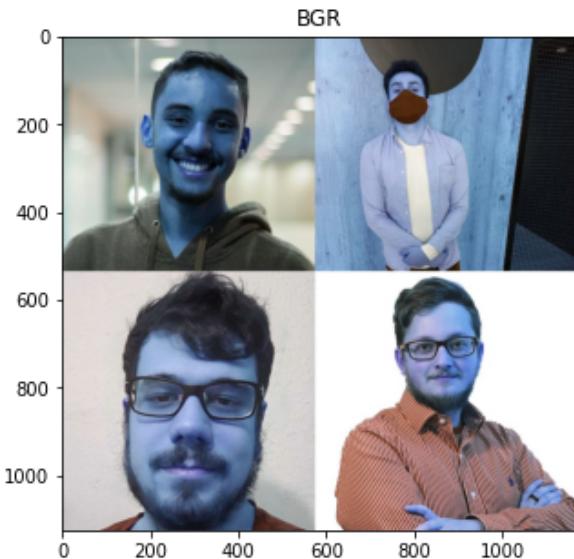
```
bgr1 = cv.imread('/content/drive/My Drive/Colab Notebooks/Lab_3/grupo.png')
altura, largura, camadas = bgr1.shape
```

```
print("Resolução: ", largura, " x ", altura, "PIXELS. ", camadas, " camadas.")  
Resolução: 1166 x 1125 PIXELS. 3 camadas.
```

```
b1, g1, r1 = cv.split(bgr1)  
rgb2 = cv.merge([r1,g1,b1])  
# Q1 - O que foi feito aqui?
```

```
plt.figure(figsize=[12, 5])  
plt.subplot(121); plt.imshow(bgr1); plt.title('BGR')  
plt.subplot(122); plt.imshow(rgb2); plt.title('RGB')
```

```
Text(0.5, 1.0, 'RGB')
```



```
ycrcb = cv.cvtColor(bgr1, cv.COLOR_BGR2YCrCb)  
hsv = cv.cvtColor(bgr1, cv.COLOR_BGR2HSV)  
plt.figure(figsize=[15,6])  
plt.subplot(131); plt.imshow(rgb2); plt.title('RGB')  
plt.subplot(132); plt.imshow(ycrcb); plt.title('YCrCb')  
plt.subplot(133); plt.imshow(hsv); plt.title('HSV')
```

```

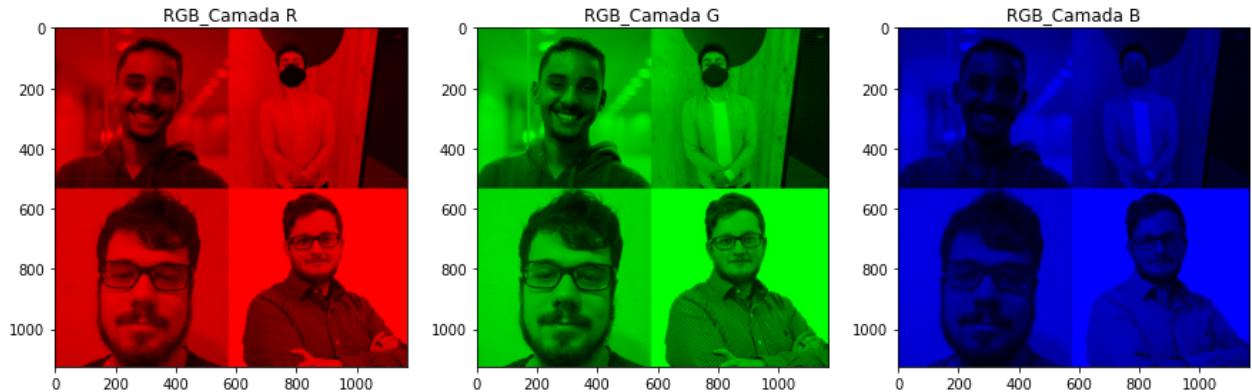
Text(0.5, 1.0, 'HSV')

imageR = rgb2.copy()
imageR[:, :, 1:3] = 0
imageG = rgb2.copy()
imageG[:, :, 0] = 0; imageG[:, :, 2] = 0
imageB = rgb2.copy()
imageB[:, :, 0:2] = 0
# Q2 - O que foi feito aqui?

plt.figure(figsize=[15,6])
plt.subplot(131); plt.imshow(imageR); plt.title('RGB_Camada R')
plt.subplot(132); plt.imshow(imageG); plt.title('RGB_Camada G')
plt.subplot(133); plt.imshow(imageB); plt.title('RGB_Camada B')

```

Text(0.5, 1.0, 'RGB_Camada B')



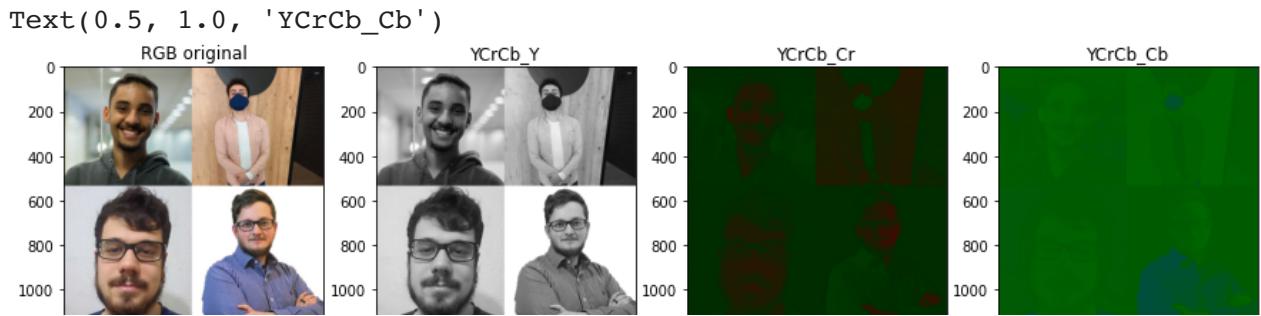
```

y1, cr1, cb1 = cv.split(ycrcb)
imageCR = ycrcb.copy()
imageCR[:, :, 0] = 0
imageCR[:, :, 2] = 0
Cr = cv.cvtColor(imageCR, cv.COLOR_YCrCb2RGB)

imageCB = ycrcb.copy()
imageCB[:, :, 0] = 0
imageCB[:, :, 1] = 0
Cb = cv.cvtColor(imageCB, cv.COLOR_YCrCb2RGB)

plt.figure(figsize=[15, 5])
plt.subplot(141); plt.imshow(rgb2); plt.title('RGB original')
plt.subplot(142); plt.imshow(y1, cmap='gray'); plt.title('YCrCb_Y')
plt.subplot(143); plt.imshow(Cr); plt.title('YCrCb_Cr')
plt.subplot(144); plt.imshow(Cb); plt.title('YCrCb_Cb')

```



▼ Victor

Clique duas vezes (ou pressione "Enter") para editar

```

Text(0.5, 1.0, 'YCrCb_Cb')
RGB original          YCrCb_Y           YCrCb_Cr           YCrCb_Cb
0   0   0   0
200 200 200 200
400 400 400 400
600 600 600 600
800 800 800 800
1000 1000 1000 1000

bgr1 = cv.imread('/content/drive/My Drive/Colab Notebooks/Lab_3/victor.png')
altura, largura, camadas = bgr1.shape
print("Resolução: ", largura, " x ", altura, "PIXELS. ", camadas, " camadas.")
b1, g1, r1 = cv.split(bgr1)
rgb2 = cv.merge([r1,g1,b1])
# Q1 - O que foi feito aqui?
plt.figure(figsize=[12, 5])
plt.subplot(121); plt.imshow(bgr1); plt.title('BGR')
plt.subplot(122); plt.imshow(rgb2); plt.title('RGB')
ycrcb = cv.cvtColor(bgr1, cv.COLOR_BGR2YCrCb)
hsv = cv.cvtColor(bgr1, cv.COLOR_BGR2HSV)
plt.figure(figsize=[15,6])
plt.subplot(131); plt.imshow(rgb2); plt.title('RGB')
plt.subplot(132); plt.imshow(ycrcb); plt.title('YCrCb')
plt.subplot(133); plt.imshow(hsv); plt.title('HSV')
imageR = rgb2.copy()
imageR[:, :, 1:3] = 0
imageG = rgb2.copy()
imageG[:, :, 0] = 0; imageG[:, :, 2] = 0
imageB = rgb2.copy()
imageB[:, :, 0:2] = 0
# Q2 - O que foi feito aqui?

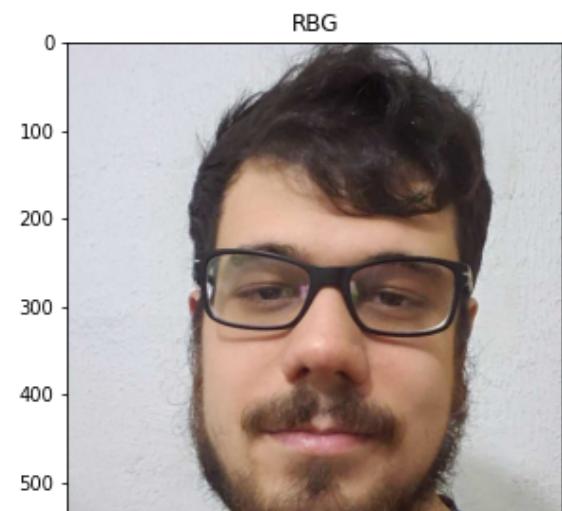
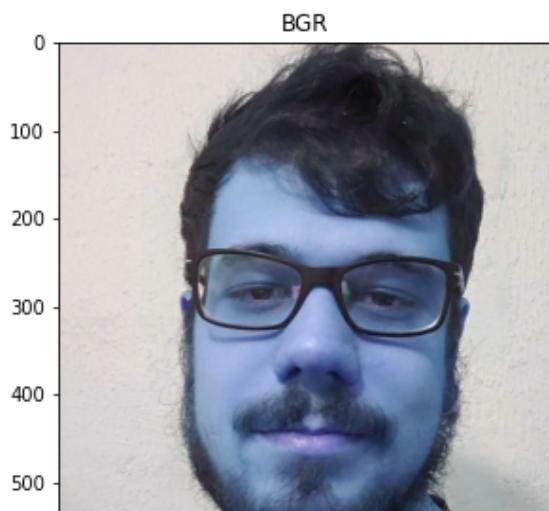
plt.figure(figsize=[15,6])
plt.subplot(131); plt.imshow(imageR); plt.title('RGB_Camada R')
plt.subplot(132); plt.imshow(imageG); plt.title('RGB_Camada G')
plt.subplot(133); plt.imshow(imageB); plt.title('RGB_Camada B')
y1, cr1, cb1 = cv.split(ycrcb)
imageCR = ycrcb.copy()
imageCR[:, :, 0] = 0
imageCR[:, :, 2] = 0
Cr = cv.cvtColor(imageCR, cv.COLOR_YCrCb2RGB)

imageCB = ycrcb.copy()
imageCB[:, :, 0] = 0
imageCB[:, :, 1] = 0
Cb = cv.cvtColor(imageCB, cv.COLOR_YCrCb2RGB)

```

```
plt.figure(figsize=[15, 5])
plt.subplot(141); plt.imshow(rgb2); plt.title('RGB original')
plt.subplot(142); plt.imshow(y1, cmap='gray'); plt.title('YCrCb_Y')
plt.subplot(143); plt.imshow(Cr); plt.title('YCrCb_Cr')
plt.subplot(144); plt.imshow(Cb); plt.title('YCrCb_Cb')
```

Resolução: 564 x 585 PIXELS. 3 camadas.
Text(0.5, 1.0, 'YCrCb_Cb')



▼ Muriel



```
bgr1 = cv.imread('/content/drive/My Drive/Colab Notebooks/Lab_3/muriel.png')
altura, largura, camadas = bgr1.shape
print("Resolução: ", largura, " x ", altura, "PIXELS. ", camadas, " camadas.")
b1, g1, r1 = cv.split(bgr1)
rgb2 = cv.merge([r1,g1,b1])
# Q1 - O que foi feito aqui?
plt.figure(figsize=[12, 5])
plt.subplot(121); plt.imshow(bgr1); plt.title('BGR')
plt.subplot(122); plt.imshow(rgb2); plt.title('RGB')
ycrcb = cv.cvtColor(bgr1, cv.COLOR_BGR2YCrCb)
hsv = cv.cvtColor(bgr1, cv.COLOR_BGR2HSV)
plt.figure(figsize=[15,6])
plt.subplot(131); plt.imshow(rgb2); plt.title('RGB')
plt.subplot(132); plt.imshow(ycrcb); plt.title('YCrCb')
plt.subplot(133); plt.imshow(hsv); plt.title('HSV')
imageR = rgb2.copy()
imageR[:, :, 1:3] = 0
imageG = rgb2.copy()
imageG[:, :, 0] = 0; imageG[:, :, 2] = 0
imageB = rgb2.copy()
imageB[:, :, 0:2] = 0
# Q2 - O que foi feito aqui?

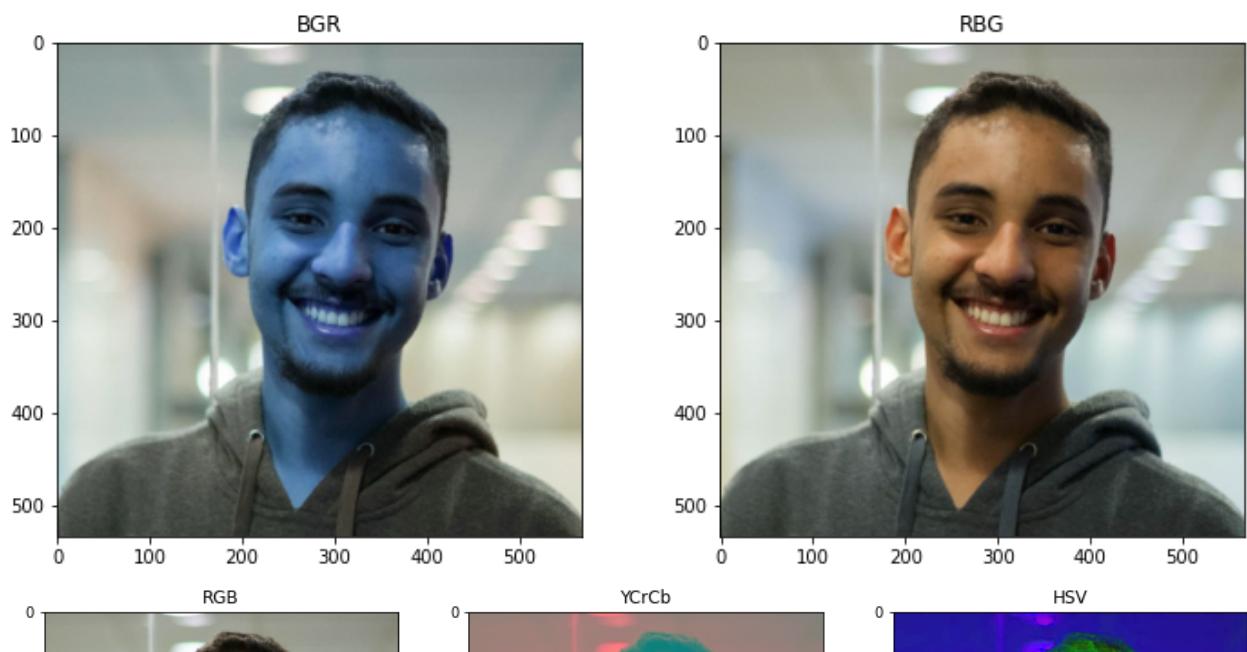
plt.figure(figsize=[15,6])
plt.subplot(131); plt.imshow(imageR); plt.title('RGB_Camada R')
plt.subplot(132); plt.imshow(imageG); plt.title('RGB_Camada G')
plt.subplot(133); plt.imshow(imageB); plt.title('RGB_Camada B')
y1, cr1, cb1 = cv.split(ycrcb)
imageCR = ycrcb.copy()
imageCR[:, :, 0] = 0
imageCR[:, :, 2] = 0
```

```
Cr = cv.cvtColor(imageCR, cv.COLOR_YCrCb2RGB)

imageCB = ycrcb.copy()
imageCB[:, :, 0] = 0
imageCB[:, :, 1] = 0
Cb = cv.cvtColor(imageCB, cv.COLOR_YCrCb2RGB)

plt.figure(figsize=[15, 5])
plt.subplot(141); plt.imshow(rgb2); plt.title('RGB original')
plt.subplot(142); plt.imshow(y1, cmap='gray'); plt.title('YCrCb_Y')
plt.subplot(143); plt.imshow(Cr); plt.title('YCrCb_Cr')
plt.subplot(144); plt.imshow(Cb); plt.title('YCrCb_Cb')
```

Resolução: 567 x 534 PIXELS. 3 camadas.
Text(0.5, 1.0, 'YCrCb_Cb')

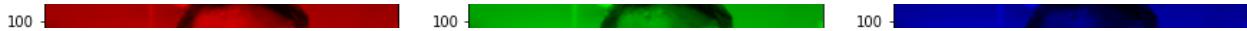


▶ Lucas

[] ↳ 3 células ocultas



▼ Jefferson



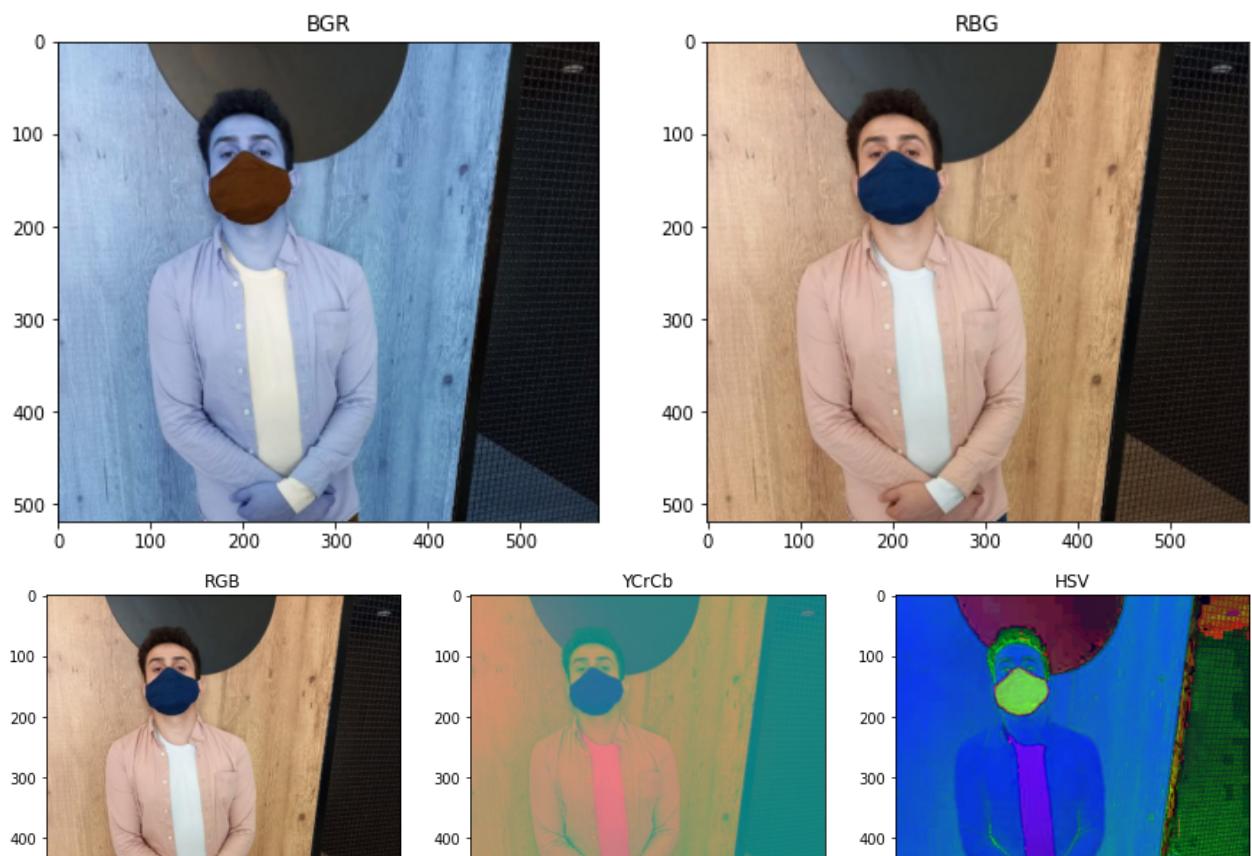
```
bgr1 = cv.imread('/content/drive/My Drive/Colab Notebooks/Lab_3/jefferson.png')
altura, largura, camadas = bgr1.shape
print("Resolução: ", largura, " x ", altura, "PIXELS. ", camadas, " camadas.")
b1, g1, r1 = cv.split(bgr1)
rgb2 = cv.merge([r1,g1,b1])
# Q1 - O que foi feito aqui?
plt.figure(figsize=[12, 5])
plt.subplot(121); plt.imshow(bgr1); plt.title('BGR')
plt.subplot(122); plt.imshow(rgb2); plt.title('RBG')
ycrcb = cv.cvtColor(bgr1, cv.COLOR_BGR2YCrCb)
hsv = cv.cvtColor(bgr1, cv.COLOR_BGR2HSV)
plt.figure(figsize=[15,6])
plt.subplot(131); plt.imshow(rgb2); plt.title('RGB')
plt.subplot(132); plt.imshow(ycrcb); plt.title('YCrCb')
plt.subplot(133); plt.imshow(hsv); plt.title('HSV')
imageR = rgb2.copy()
imageR[:, :, 1:3] = 0
imageG = rgb2.copy()
imageG[:, :, 0] = 0; imageG[:, :, 2] = 0
imageB = rgb2.copy()
imageB[:, :, 0:2] = 0
# Q2 - O que foi feito aqui?
```

```
plt.figure(figsize=[15,6])
plt.subplot(131); plt.imshow(imageR); plt.title('RGB_Camada R')
plt.subplot(132); plt.imshow(imageG); plt.title('RGB_Camada G')
plt.subplot(133); plt.imshow(imageB); plt.title('RGB_Camada B')
y1, cr1, cb1 = cv.split(ycrcb)
imageCR = ycrcb.copy()
imageCR[:, :, 0] = 0
imageCR[:, :, 2] = 0
Cr = cv.cvtColor(imageCR, cv.COLOR_YCrCb2RGB)

imageCB = ycrcb.copy()
imageCB[:, :, 0] = 0
imageCB[:, :, 1] = 0
Cb = cv.cvtColor(imageCB, cv.COLOR_YCrCb2RGB)

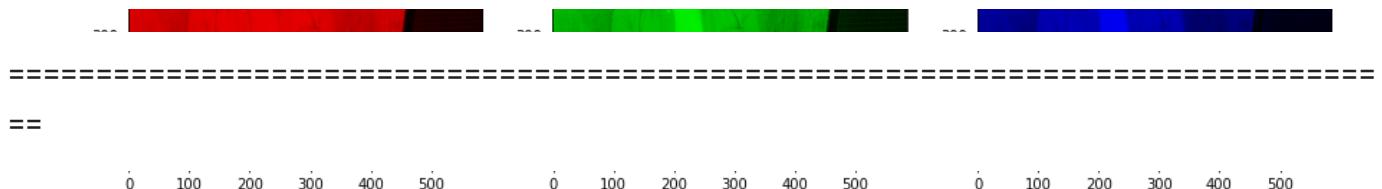
plt.figure(figsize=[15, 5])
plt.subplot(141); plt.imshow(rgb2); plt.title('RGB original')
plt.subplot(142); plt.imshow(y1, cmap='gray'); plt.title('YCrCb_Y')
plt.subplot(143); plt.imshow(Cr); plt.title('YCrCb_Cr')
plt.subplot(144); plt.imshow(Cb); plt.title('YCrCb_Cb')
```

Resolução: 585 x 519 PIXELS. 3 camadas.
Text(0.5, 1.0, 'YCrCb_Cb')



▼ 3. Comparar arquivos comprimidos JPEG

COMPRESSÃO DE IMAGENS COM PERDAS



- O formato JPEG permite compressão da imagem ao salvá-la num arquivo com o comando `imwrite()`.
- A compressão afeta a qualidade da imagem, sendo controlada pelo parâmetro `IMWRITE_JPEG_QUALITY` entre 0-100, sendo que quanto maior, melhor a qualidade. O default é 95.

```
bgr = cv.imread('/content/drive/My Drive/Colab Notebooks/Lab_3/lena.bmp') # format

print(type(bgr))
print(np.shape(bgr))
# salva com menor qualidade, fatores 25 e 5
cv.imwrite('/content/drive/My Drive/Colab Notebooks/Lab_3/lena25.jpg', bgr, [cv.IMW
cv.imwrite('/content/drive/My Drive/Colab Notebooks/Lab_3/lena05.jpg', bgr, [cv.IMW
```

```

# leitura para visualização e conversão para acertar a cor
rgb = cv.cvtColor(bgr, cv.COLOR_BGR2RGB)
bgr25 = cv.imread('/content/drive/My Drive/Colab Notebooks/Lab_3/lena25.jpg'); rgb2
bgr05 = cv.imread('/content/drive/My Drive/Colab Notebooks/Lab_3/lena05.jpg'); rgb0

plt.figure(figsize=[15,6])
plt.subplot(131); plt.imshow(rgb); plt.title('RGB Original')
plt.subplot(132); plt.imshow(rgb25); plt.title('JPEG fator 25')
plt.subplot(133); plt.imshow(rgb05); plt.title('JPEG fator 05')

<class 'numpy.ndarray'>
(512, 512, 3)
Text(0.5, 1.0, 'JPEG fator 05')

```

▼ COM AS FOTOS DO GRUPO

1. Repita o procedimento para cada uma das fotos dos integrantes do grupo e para a foto-montagem do grupo todo
2. Leia o tamanho dos arquivos (em bytes) e faça uma tabela comparando os tamanhos originais e os comprimidos e calcule a porcentagem de compressão de cada arquivo destes tamanhos na tabela construída

```

bgr = cv.imread('/content/drive/My Drive/Colab Notebooks/Lab_3/grupo.png') # forma

# salva com menor qualidade, fatores 25 e 5
cv.imwrite('/content/drive/My Drive/Colab Notebooks/Lab_3/grupo25.jpg', bgr, [cv.IM
cv.imwrite('/content/drive/My Drive/Colab Notebooks/Lab_3/grupo05.jpg', bgr, [cv.IM

# leitura para visualização e conversão para acertar a cor
rgb = cv.cvtColor(bgr, cv.COLOR_BGR2RGB)
bgr25 = cv.imread('/content/drive/My Drive/Colab Notebooks/Lab_3/grupo25.jpg'); rgb
bgr05 = cv.imread('/content/drive/My Drive/Colab Notebooks/Lab_3/grupo05.jpg'); rgb

plt.figure(figsize=[15,6])

```

```

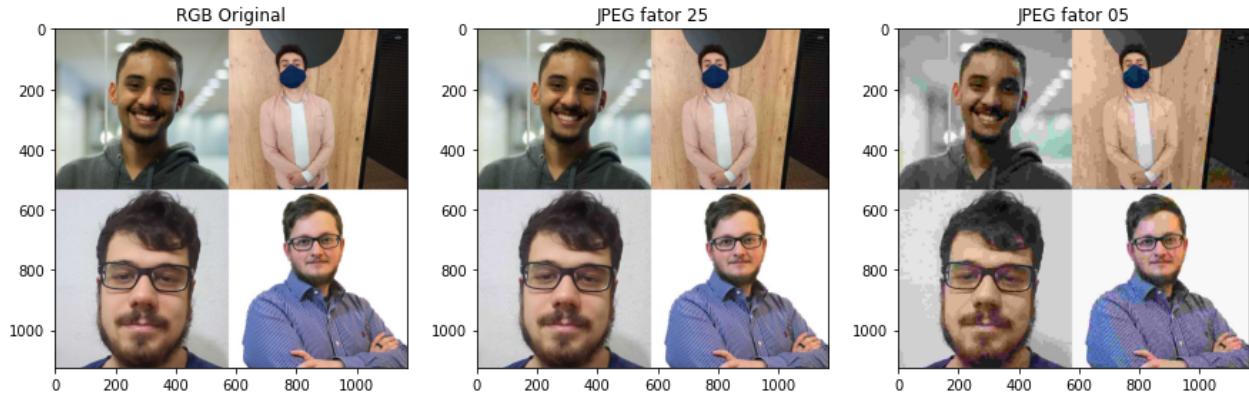
plt.subplot(131); plt.imshow(rgb); plt.title('RGB Original')
plt.subplot(132); plt.imshow(rgb25); plt.title('JPEG fator 25')
plt.subplot(133); plt.imshow(rgb05); plt.title('JPEG fator 05')

#identificando o fator de compressao
size_grupo = [Path('/content/drive/My Drive/Colab Notebooks/Lab_3/grupo.png').stat(),
              Path('/content/drive/My Drive/Colab Notebooks/Lab_3/grupo25.jpg').st]

s1 = str(size_grupo[1]) + ' ({0:.2f}%)'.format(size_grupo[1] / size_grupo[0] * 100)
s2 = str(size_grupo[2]) + ' ({0:.2f}%)'.format(size_grupo[2] / size_grupo[0] * 100)
print('Original',size_grupo[0])
print('Compressao de 25', s1)
print('Compressao de 05', s2)

```

Original 1425085
 Compressao de 25 74562 (5.23%)
 Compressao de 05 33591 (2.36%)



▼ Victor

```

bgr = cv.imread('/content/drive/My Drive/Colab Notebooks/Lab_3/victor.png') # forma

# salva com menor qualidade, fatores 25 e 5
cv.imwrite('/content/drive/My Drive/Colab Notebooks/Lab_3/victor25.jpg', bgr, [cv.I
cv.imwrite('/content/drive/My Drive/Colab Notebooks/Lab_3/victor05.jpg', bgr, [cv.I

# leitura para visualização e conversão para acertar a cor
rgb = cv.cvtColor(bgr, cv.COLOR_BGR2RGB)
bgr25 = cv.imread('/content/drive/My Drive/Colab Notebooks/Lab_3/victor25.jpg'); rg
bgr05 = cv.imread('/content/drive/My Drive/Colab Notebooks/Lab_3/victor05.jpg'); rg

plt.figure(figsize=[15,6])

```

```

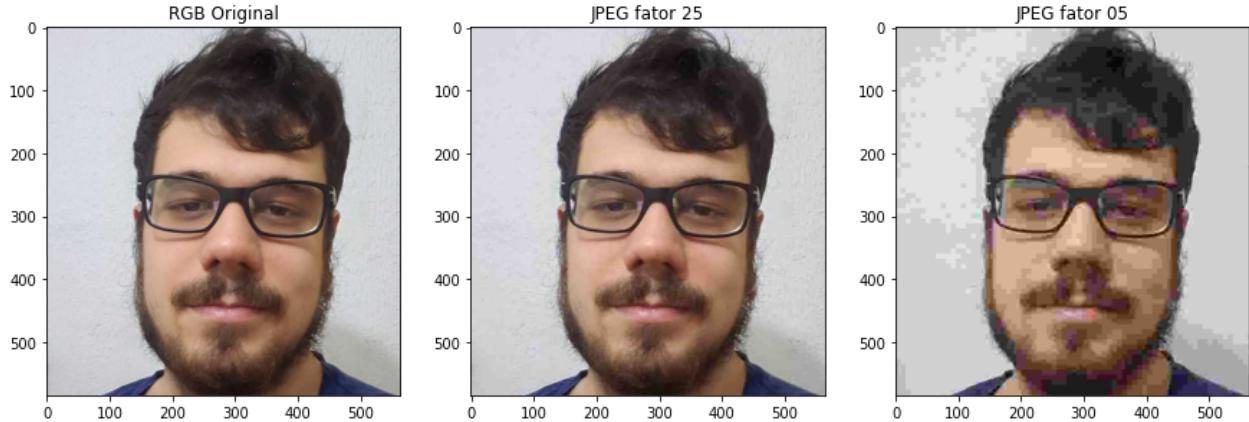
plt.subplot(131); plt.imshow(rgb); plt.title('RGB Original')
plt.subplot(132); plt.imshow(rgb25); plt.title('JPEG fator 25')
plt.subplot(133); plt.imshow(rgb05); plt.title('JPEG fator 05')

#identificando o fator de compressao
size_victor = [Path('/content/drive/My Drive/Colab Notebooks/Lab_3/victor.png').stat().st_size, Path('/content/drive/My Drive/Colab Notebooks/Lab_3/victor25.jpg').stat().st_size]
size_victor[1] / size_victor[0] * 100
size_victor[2] / size_victor[0] * 100

s1 = str(size_victor[1]) + ' ({0:.2f}%)'.format(size_victor[1] / size_victor[0] * 100)
s2 = str(size_victor[2]) + ' ({0:.2f}%)'.format(size_victor[2] / size_victor[0] * 100)
print('Original', size_victor[0])
print('Compressao de 25', s1)
print('Compressao de 05', s2)

```

Original 403638
 Compressao de 25 15610 (3.87%)
 Compressao de 05 7652 (1.90%)



▼ Muriel

```

bgr = cv.imread('/content/drive/My Drive/Colab Notebooks/Lab_3/muriel.png') # forma original

# salva com menor qualidade, fatores 25 e 5
cv.imwrite('/content/drive/My Drive/Colab Notebooks/Lab_3/muriel25.jpg', bgr, [cv.IWRITE_JPEG_QUALITY, 25])
cv.imwrite('/content/drive/My Drive/Colab Notebooks/Lab_3/muriel05.jpg', bgr, [cv.IWRITE_JPEG_QUALITY, 5])

# leitura para visualização e conversão para acertar a cor
rgb = cv.cvtColor(bgr, cv.COLOR_BGR2RGB)
bgr25 = cv.imread('/content/drive/My Drive/Colab Notebooks/Lab_3/muriel25.jpg'); rg

```

```

bgr05 = cv.imread('/content/drive/My Drive/Colab Notebooks/Lab_3/muriel05.jpg'); rg

plt.figure(figsize=[15,6])
plt.subplot(131); plt.imshow(rgb); plt.title('RGB Original')
plt.subplot(132); plt.imshow(rgb25); plt.title('JPEG fator 25')
plt.subplot(133); plt.imshow(rgb05); plt.title('JPEG fator 05')

#identificando o fator de compressao
size_muriel = [Path('/content/drive/My Drive/Colab Notebooks/Lab_3/muriel.png').sta
                Path('/content/drive/My Drive/Colab Notebooks/Lab_3/muriel25.jpg').s

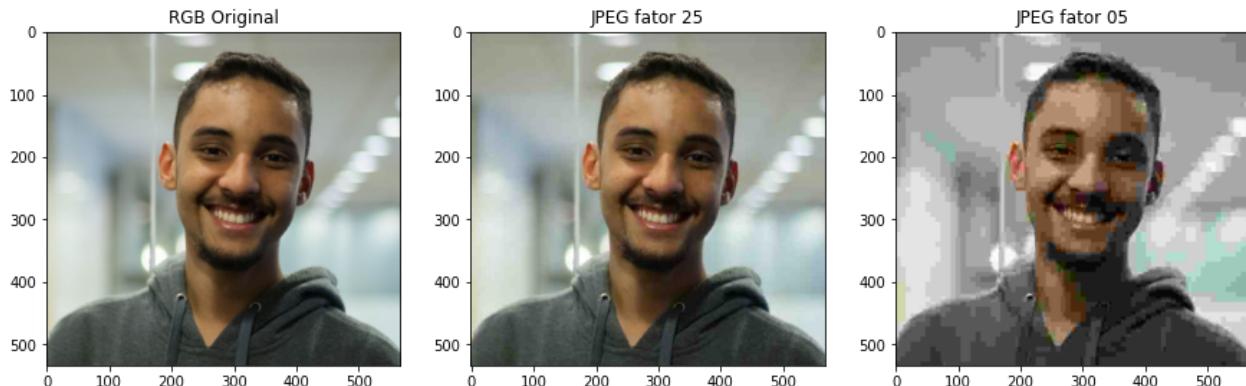
```

```

s1 = str(size_muriel[1]) + ' ({0:.2f}%)'.format(size_muriel[1] / size_muriel[0] * 1
s2 = str(size_muriel[2]) + ' ({0:.2f}%)'.format(size_muriel[2] / size_muriel[0] * 1
print('Original',size_muriel[0])
print('Compressao de 25', s1)
print('Compressao de 05', s2)

```

Original 330293
 Compressao de 25 12296 (3.72%)
 Compressao de 05 6862 (2.08%)



▼ Lucas

```

bgr = cv.imread('/content/drive/My Drive/Colab Notebooks/Lab_3/lucas.png') # forma

# salva com menor qualidade, fatores 25 e 5
cv.imwrite('/content/drive/My Drive/Colab Notebooks/Lab_3/lucas25.jpg', bgr, [cv.IM
cv.imwrite('/content/drive/My Drive/Colab Notebooks/Lab_3/lucas05.jpg', bgr, [cv.IM

# leitura para visualização e conversão para acertar a cor

```

```

rgb = cv.cvtColor(bgr, cv.COLOR_BGR2RGB)
bgr25 = cv.imread('/content/drive/My Drive/Colab Notebooks/Lab_3/lucas25.jpg'); rgb
bgr05 = cv.imread('/content/drive/My Drive/Colab Notebooks/Lab_3/lucas05.jpg'); rgb

plt.figure(figsize=[15,6])
plt.subplot(131); plt.imshow(rgb); plt.title('RGB Original')
plt.subplot(132); plt.imshow(rgb25); plt.title('JPEG fator 25')
plt.subplot(133); plt.imshow(rgb05); plt.title('JPEG fator 05')

#identificando o fator de compressao
size_lucas = [Path('/content/drive/My Drive/Colab Notebooks/Lab_3/lucas.png').stat(),
              Path('/content/drive/My Drive/Colab Notebooks/Lab_3/lucas25.jpg').st

```

```

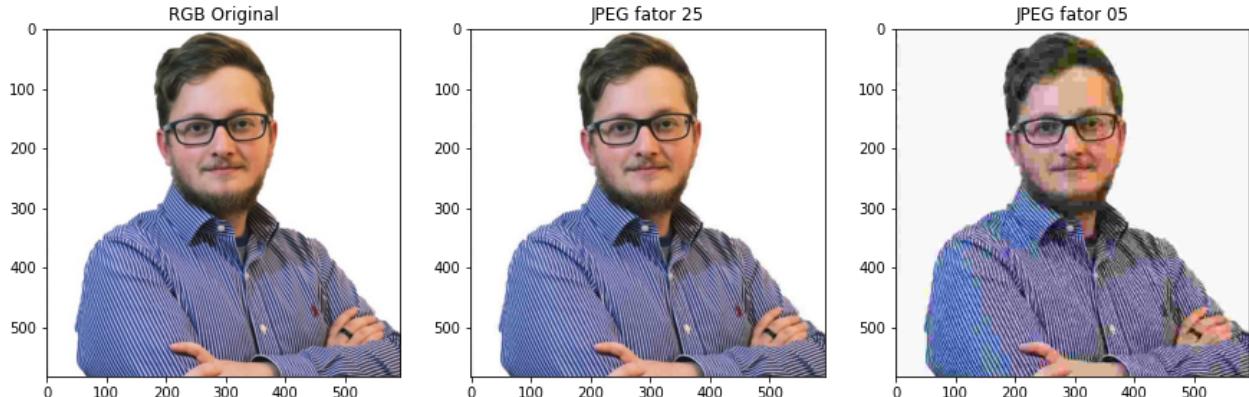
s1 = str(size_lucas[1]) + ' ({0:.2f}%)'.format(size_lucas[1] / size_lucas[0] * 100)
s2 = str(size_lucas[2]) + ' ({0:.2f}%)'.format(size_lucas[2] / size_lucas[0] * 100)
print('Original',size_lucas[0])
print('Compressao de 25', s1)
print('Compressao de 05', s2)

```

Original 410282

Compressao de 25 29011 (7.07%)

Compressao de 05 13636 (3.32%)



▼ Jefferson

```

bgr = cv.imread('/content/drive/My Drive/Colab Notebooks/Lab_3/jefferson.png') # f
# salva com menor qualidade, fatores 25 e 5
cv.imwrite('/content/drive/My Drive/Colab Notebooks/Lab_3/jefferson25.jpg', bgr, [c

```

```

cv.imwrite('/content/drive/My Drive/Colab Notebooks/Lab_3/jefferson05.jpg', bgr, [c

# leitura para visualização e conversão para acertar a cor
rgb = cv.cvtColor(bgr, cv.COLOR_BGR2RGB)
bgr25 = cv.imread('/content/drive/My Drive/Colab Notebooks/Lab_3/jefferson25.jpg');
bgr05 = cv.imread('/content/drive/My Drive/Colab Notebooks/Lab_3/jefferson05.jpg');

plt.figure(figsize=[15,6])
plt.subplot(131); plt.imshow(rgb); plt.title('RGB Original')
plt.subplot(132); plt.imshow(rgb25); plt.title('JPEG fator 25')
plt.subplot(133); plt.imshow(rgb05); plt.title('JPEG fator 05')

#identificando o fator de compressao
size_jefferson = [Path('/content/drive/My Drive/Colab Notebooks/Lab_3/jefferson.png',
Path('/content/drive/My Drive/Colab Notebooks/Lab_3/jefferson25.jpg'

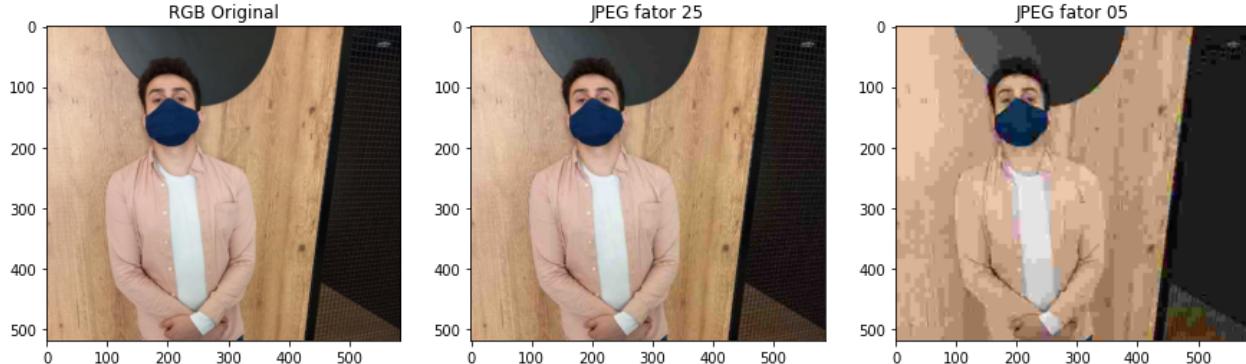
s1 = str(size_jefferson[1]) + ' ({0:.2f}%)'.format(size_jefferson[1] / size_jefferson[0])
s2 = str(size_jefferson[2]) + ' ({0:.2f}%)'.format(size_jefferson[2] / size_jefferson[0])
print('Original',size_jefferson[0])
print('Compressao de 25', s1)
print('Compressao de 05', s2)

```

Original 374872

Compressao de 25 17912 (4.78%)

Compressao de 05 6822 (1.82%)



Resumo das tabelas de compressao

Para o grupo

Original: 1425085

Compressao de 25: 74562 (5.23%)

Compressao de 05: 33591 (2.36%)

Para o victor

Original: 403638

Compressao de 25: 15610 (3.87%)

Compressao de 05: 7652 (1.90%)

Para o muriel

Original: 330293

Compressao de 25: 12296 (3.72%)

Compressao de 05: 6862 (2.08%)

Para o lucas

Original: 410282

Compressao de 25: 29011 (7.07%)

Compressao de 05: 13636 (3.32%)

Para o jefferson

Original: 374872

Compressao de 25: 17912 (4.78%)

Compressao de 05: 6822 (1.82%)

▼ TRANSFORMADA DISCRETA COSSENO

▼ 4. Efetuar compressão de imagem com DCT

Nesta parte calcule a DCT em bloco de 8x8 da imagem, referente à bola

```



```

```

(1)
Matriz 8x8: componente Y original
[[156 168 139 155 154 160 162 155]
 [156 166 139 155 154 160 159 154]
 [156 165 140 154 152 157 156 152]
 [159 167 143 156 152 158 155 155]
 [161 167 144 157 154 160 157 158]
 [160 164 141 154 152 160 156 157]
 [154 154 153 156 153 154 156 154]
 [156 148 154 159 156 150 154 154]]

(2)
Imagen Y 8x8 (formato ponto flutuante)
[[0.61 0.66 0.55 0.61 0.6 0.63 0.64 0.61]
 [0.61 0.65 0.55 0.61 0.6 0.63 0.62 0.6 ]
 [0.61 0.65 0.55 0.6 0.6 0.62 0.61 0.6 ]
 [0.62 0.65 0.56 0.61 0.6 0.62 0.61 0.61]
 [0.63 0.65 0.56 0.62 0.6 0.63 0.62 0.62]
 [0.63 0.64 0.55 0.6 0.6 0.63 0.61 0.62]
 [0.6 0.6 0.6 0.61 0.6 0.6 0.61 0.6 ]
 [0.61 0.58 0.6 0.62 0.61 0.59 0.6 0.6 ]]

(3)
DCT de Y (ponto flutuante)
[[ 4.87 0. 0.05 0.07 0. -0.03 -0.08 -0.09]
 [ 0.01 -0. 0.02 0.04 -0.02 -0.03 -0.05 -0.04]
 [-0.01 -0.02 -0.02 -0.02 0. 0.02 0.01 0.03]]

```

```

[ 0.02 -0.01  0.02  0.02 -0.01 -0.01 -0.01 -0.02]
[ 0.01  0.   -0.   -0.01  0.01  0.   0.   0.   ]
[-0.01  0.   -0.   -0.01 -0.01  0.   0.   0.   ]
[-0.01  0.   0.01  0.01  0.   -0.   -0.01 -0.01]
[ 0.   -0.   -0.01 -0.01 -0.   0.   0.   0.01]]
(4)
DCT de Y (formato inteiro)
[[1242    0   13   17    0   -6  -21  -22]
 [ 2   -1   5    9   -6  -7  -12  -10]
 [-3   -4   -6   -4    1   5    1    7]
 [ 5   -2   5    4   -2  -3  -2   -4]
 [ 3    1   0   -1    2    0    0    0]
 [-2    0   0   -2   -2    0    0    1]
 [-1    0   2    2    1    0   -1   -1]
 [ 0    0   -1   -1    0    1    0    1]]
```

ZERANDO manualmente da diagonal da DCT as componentes AC

```

dct8x8fz = dct8x8f.copy()
dct8x8fz[0,7] = 0
dct8x8fz[1,6:8] = 0
dct8x8fz[2,5:8] = 0
dct8x8fz[3,4:8] = 0
dct8x8fz[4,3:8] = 0
dct8x8fz[5,2:8] = 0
dct8x8fz[6,1:8] = 0
dct8x8fz[7,0:8] = 0
print( np.around(dct8x8fz, decimals = 2))
```

```

[[ 4.87  0.    0.05  0.07  0.   -0.03 -0.08  0.   ]
 [ 0.01 -0.   0.02  0.04 -0.02 -0.03  0.   0.   ]
 [-0.01 -0.02 -0.02 -0.02  0.   0.   0.   0.   ]
 [ 0.02 -0.01  0.02  0.02  0.   0.   0.   0.   ]
 [ 0.01  0.   -0.   0.   0.   0.   0.   0.   ]
 [-0.01  0.   0.   0.   0.   0.   0.   0.   ]
 [-0.01  0.   0.   0.   0.   0.   0.   0.   ]
 [ 0.   0.   0.   0.   0.   0.   0.   0.   ]]
```

Bloco Original e Reconstruído com Zeros das componentes AC da diagonal para baixo zerados

```

bloco8x8recz = cv.idct(dct8x8fz)

plt.subplot(121); plt.imshow(bloco8x8,'gray'); plt.title('Bloco Original')
plt.subplot(122); plt.imshow(bloco8x8recz,'gray'); plt.title('Bloco Reconstruído co
```



▼ Escolha outro bloco de 8x8 da imagem e:

1. refaça este procedimento zerando mais DUAS DIAGONAIS ACIMA DA PRINCIPAL além destas
2. Compare e comente as imagens do bloco original e reconstruída

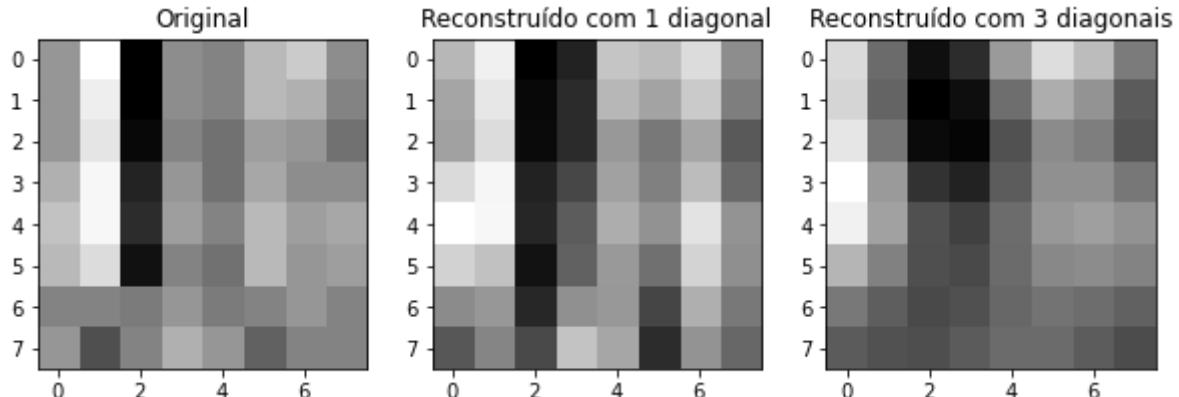
```
dct8x8fz2 = dct8x8f.copy()
dct8x8fz2[0,5:8] = 0
dct8x8fz2[1,4:8] = 0
dct8x8fz2[2,3:8] = 0
dct8x8fz2[3,2:8] = 0
dct8x8fz2[4,1:8] = 0
dct8x8fz2[5,0:8] = 0
dct8x8fz2[6,0:8] = 0
dct8x8fz2[7,0:8] = 0
print(np.around(dct8x8fz2, decimals = 2))

bloco8x8recz2 = cv.idct(dct8x8fz2)

plt.figure(figsize=[10,6])
plt.subplot(131); plt.imshow(bloco8x8,'gray'); plt.title('Original')
plt.subplot(132); plt.imshow(bloco8x8recz,'gray'); plt.title('Reconstruído com 1 diagonal')
plt.subplot(133); plt.imshow(bloco8x8recz2,'gray'); plt.title('Reconstruído com 3 diagonais')
```

```
[[ 4.87  0.     0.05  0.07  0.     0.     0.     0.     ],
 [ 0.01 -0.     0.02  0.04  0.     0.     0.     0.     ],
 [-0.01 -0.02 -0.02  0.     0.     0.     0.     0.     ],
 [ 0.02 -0.01  0.     0.     0.     0.     0.     0.     ],
 [ 0.01  0.     0.     0.     0.     0.     0.     0.     ],
 [ 0.     0.     0.     0.     0.     0.     0.     0.     ],
 [ 0.     0.     0.     0.     0.     0.     0.     0.     ],
 [ 0.     0.     0.     0.     0.     0.     0.     0.     ]]
```

Text(0.5, 1.0, 'Reconstruído com 3 diagonais')



É possível observar, como resultado do processo de reconstrução, a concentração de energia em regiões específicas, destacados em preto pela figura, os pontos são significativamente dispersos nas regiões mais brancas, ressaltando as diagonais nulas. Observa-se que a eliminação de componentes provoca perda de maior frequência na imagem, caracterizando um processo de compressão.

Clique duas vezes (ou pressione "Enter") para editar

-X-X-X-