**Atividade Prévia – Trabalho Prático**

|  |  |
| --- | --- |
| **Disciplina** | **AIM – Análise de Imagens** |
| **Aluno** |  |

**Objetivos**

Após a realização da Atividade Prévia do Trabalho Prático, você terá:

* Configurado seu ambiente local para utilização do Jupyter Notebook.
* Executado testes inicias com análise de imagens.

**Enunciado**

O Anaconda é uma distribuição gratuita e de código aberto das linguagens de programação Python e R para computação científica, que visa simplificar o gerenciamento e a implantação de pacotes. A distribuição inclui pacotes de ciência de dados adequados para Windows, Linux e macOS.

Vamos configurar nosso ambiente local seguindo as instruções do site oficial: <https://docs.conda.io/projects/conda/en/latest/user-guide/install/index.html>

**Atividades**

1. Seguir o manual de instalação escolhendo o sistema operacional utilizado pelo aluno.
2. Criar ambiente com Python 3 utilizando o comando abaixo:
   1. conda create -n py36 python=3.6 anaconda
3. Ativar ambiente com Python 3 utilizando o comando abaixo:
   1. conda activate py36
4. Instalar Bibliotecas:
   1. pip install opencv-python
5. Abrir Jupyter Notebook utilizando o comando abaixo:
   1. jupyter notebook
6. Teste os seguintes códigos abaixo no seu notebook:

Nota: Teste diferentes combinações quando possível, lembre-se de usar imagens que você tem localmente.

Celula 1 *# Importação das bibliotecas*   
**import** **cv2** **import** **matplotlib.pyplot** **as** **plt**   
image = cv2.imread('lenna.jpeg')   
plt.imshow(cv2.cvtColor(image, cv2.COLOR\_BGR2RGB))   
print(image.shape)

Celula 2

blue = image.copy()

blue[:,:,1] = 0

blue[:,:,2] = 0

green = image.copy()

green[:,:,0] = 0

green[:,:,2] = 0

red = image.copy()

red[:,:,0] = 0

red[:,:,1] = 0

Celula 3

plt.imshow(cv2.cvtColor(blue, cv2.COLOR\_BGR2RGB))

Celula 4

plt.imshow(cv2.cvtColor(green, cv2.COLOR\_BGR2RGB))

Celula 5

plt.imshow(cv2.cvtColor(red, cv2.COLOR\_BGR2RGB))

Celula 6

img = cv2.imread('p.png')

img = cv2.cvtColor(img, cv2.COLOR\_BGR2GRAY)

img = (255-img)

scale\_percent = 5 # percent of original size

width = int(img.shape[1] \* scale\_percent / 100)

height = int(img.shape[0] \* scale\_percent / 100)

dim = (width, height)

# resize image

resized = cv2.resize(img, dim, interpolation = cv2.INTER\_AREA)

plt.imshow(resized, cmap='gray')

print(resized.shape)

Celula 7

print(resized)

print(resized.shape)