

Manipulación de Pixeles

Importar Librerías

```
In [ ]: 1 from matplotlib import pyplot as plt
        2 import argparse
        3 import cv2
```

Función para mostrar imágenes en Jupyter Notebooks y Google Colab

```
In [ ]: 1 def mostrar_imagen(titulo,imagen):
        2     imagen = cv2.cvtColor(imagen,cv2.COLOR_BGR2RGB)
        3     plt.imshow(imagen)
        4     plt.title(titulo)
        5     plt.grid(False)
        6     plt.show
```

Obtención y modificación de Pixeles

```
In [ ]: 1 args = {
        2     "imagen": "xxxxx.png"
        3 }
```

```
In [ ]: 1 # cargar la imagen, graba sus dimensiones (ancho y alto),
        2 # y luego muestra la imagen original
        3 imagen = cv2.imread(args["imagen"])
        4 (h, w) = imagen.shape[:2]
        5 mostrar_imagen("Original", imagen)
```

```
In [ ]: 1 # Las imagenes son arreglos Numpy -- con el origen (0,0) ubicado en
2 # la esquina superior izquierda de la imagen
3
4 (b, g, r) = imagen[0, 0]
5 print("Pixel en (0, 0) - Rojo: {}, Verde: {}, Azúl: {}".format(r, g, b))
6
7 # accede al pixel ubicado en x=50, y=20
8 (b, g, r) = imagen[20, 50]
9 print("Pixel en (50, 20) - Rojo: {}, Verde: {}, Azúl: {}".format(r, g, b))
10
11 # actualizar el pixel en (50, 20) and establecerlo a rojo
12 imagen[20, 50] = (0, 0, 255)
13 (b, g, r) = imagen[20, 50]
14 print("Pixel en (50, 20) - Rojo: {}, Verde: {}, Azúl: {}".format(r, g, b))
```

```
In [ ]: 1 # calcula el centro de la imagen, simplemente es el ancho y alto
2 # dividido por dos
3 (cX, cY) = (w // 2, h // 2)
4
5 # ya que estamos usando arreglos NumPy, podemos aplicar la selección
6 # de una parte del arreglo y grabar las regiones de interés de la
7 # imagen, como ejemplo grabamos la esquina superior izquierda de la
8 # imagen
9 t1 = imagen[0:cY, 0:cX]
10 mostrar_imagen("Esquina superior izquierda", t1)
```

```
In [ ]: 1 # de manera similar podemos hacer lo mismo con las otras esquinas
2 tr = image[0:cY, cX:w]
3 br = image[cY:h, cX:w]
4 bl = image[cY:h, 0:cX]
5 mostrar_imagen("Esquina superior derecha", tr)
6 mostrar_imagen("Esquina inferior derecha", br)
7 mostrar_imagen("Esquina inferior izquierda", bl)
```

```
In [ ]: 1 # hacer la esquina superior izquierda de color verde
        2 image[0:cY, 0:cX] = (0, 255, 0)
        3
        4 # Mostrar la imagen actualizada
        5 mostrar_imagen("Updated", image)
```