Conceptos Básicos de Procesamiento de Imágenes

Estudiante: Karla Abreu | 2023-0081

In [1]:
```
#Importar Librerias
import cv2 as cv
import numpy as np
import matplotlib.pyplot as plt
```

In [2]:
```
#Crear una imagen en blanco
imagen = np.zeros((500, 500, 3), dtype=np.uint8)
```

Dibujar Figuras Geométricas

In [ ]:
```
#Dibujar una linea
cv.line(imagen, (0, 0), (499, 499), (255, 0, 0), thickness=2)
```

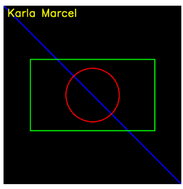
In [ ]:
```
#Dibujar rectángulo verde
#Centro de la imagen
center_x, center_y = 250, 250
#Ancho y alto del rectángulo
rect_width, rect_height =350, 200

top_left = (center_x - rect_width // 2, center_y - rect_height // 2)
bottom_right = (center_x + rect_width // 2, center_y + rect_height // 2)


cv.rectangle(imagen, top_left, bottom_right, (0, 255, 0), thickness=2)
```

In [ ]:
```
#Dibujar un circulo
cv.circle(imagen, (250, 250), 75, (0, 0, 255), thickness=2)
```

Agregar texto a la imagen

In [ ]:
```
#Escribir texto en la imagen en color amarillo
cv.putText(imagen, "Karla Marcel", (10, 30), cv.FONT_HERSHEY_SIMPLEX, 1, (0, 255, 255), thickness=2)
```

In [7]:
```
#Visualizar

#Convertir de BGR a RGB
img_rgb = cv.cvtColor(imagen, cv.COLOR_BGR2RGB)

plt.imshow(img_rgb)
plt.axis("off")
plt.show()
```




In [10]:
```
#Guardar Imagen
cv.imwrite("original.png", imagen)
```

Out[10]:
```
True
```

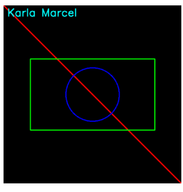
Manipulación de Imágenes

In [9]:
```
#Convertir la imagen a escala de grises
escala_grises = cv.cvtColor(imagen, cv.COLOR_BGR2GRAY)

#visualizar
plt.imshow(escala_grises)
plt.axis("off")
plt.show()
```


In [ ]:
```
#Guardar Imagen
cv.imwrite("grayscale.png", escala_grises)
```

Out[ ]:
```
True
```

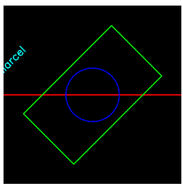
In [ ]:
```
#Redimensionar la imagen al doble de su tamaño original (1000x1000 pixeles)
redimensionar = cv.resize(imagen, (1000, 1000))

#visualizar
plt.imshow(redimensionar)
plt.axis("off")
plt.show()
```




In [ ]:
```
#Guardar Imagen
cv.imwrite("resized.png", redimensionar)
```

Out[ ]:
```
True
```

In [ ]:
```
#Rotar la imagen 45 grados alrededor de su centro
center = (250, 250)
rot_mat_45 = cv.getRotationMatrix2D(center, 45, 1.0)
rotated = cv.warpAffine(imagen, rot_mat_45, (500, 500))

#visualizar
plt.imshow(rotated)
plt.axis("off")
plt.show()
```

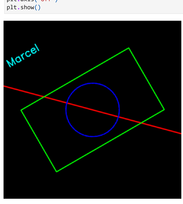



In [ ]:
```
#Guardar Imagen
cv.imwrite("rotated.png", rotated)
```

Out[ ]:
```
True
```

Manipulación Avanzada de Imágenes

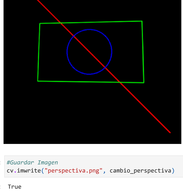
In [ ]:
```
#Girar la imagen 30 grados
rot_mat_30 = cv.getRotationMatrix2D(center, 30, 1.0)
rotated30 = cv.warpAffine(imagen, rot_mat_30, (500, 500))

#visualizar
plt.imshow(rotated30)
plt.axis("off")
plt.show()
```




In [ ]:
```
#Guardar Imagen
cv.imwrite("rotated30.png", rotated30)
```

Out[ ]:
```
True
```

In [ ]:
```
#Transformación de perspectiva:
src_points = np.float32([[0, 0], [500, 0], [500, 500], [0, 500]])
dst_points = np.float32([[50, 50], [450, 30], [470, 470], [30, 450]])
persp_mat = cv.getPerspectiveTransform(src_points, dst_points)
cambio_perspectiva = cv.warpPerspective(imagen, persp_mat, (500, 500))

plt.imshow(cambio_perspectiva)
plt.axis("off")
plt.show()
```




In [ ]:
```
#Guardar Imagen
cv.imwrite("perspectiva.png", cambio_perspectiva)
```

Out[ ]:
```
True
```

In [ ]:
```
#Máscara de transparencia
mascara_transparencia = np.zeros((500, 500), dtype=np.uint8)
cv.circle(mascara_transparencia, (250, 250), 100, 255, thickness=-1)

#Convertir la imagen original a BGRA para poder incluir un canal alfa y aplicar la máscara
img_bgra = cv.cvtColor(imagen, cv.COLOR_BGR2BGRA)
img_bgra[:, :, 3] = mascara_transparencia

plt.imshow(mascara_transparencia)
plt.axis("off")
plt.show()
```

In [ ]:
```
#Guardar Imagen
cv.imwrite("mascara_transparencia.png", mascara_transparencia)
```

Out[ ]:
```
True
```