

**CENTRO DE ENSEÑANZA TECNICA INDUSTRIAL**



## **Practica 004**

### **Dibujar sobre la imagen y ROI**

**Gonzalez Yañez Leonardo Isaac**

**28/04/2025**

# Objetivo de la práctica

- Dibujar elementos (líneas, rectángulos, círculos y texto) sobre una imagen cargada.
- Definir y extraer una **Región de Interés (ROI)** de la imagen.
- Mostrar la imagen completa con dibujos y la región de interés recortada en ventanas separadas.

## Desarrollo

### Carga de la imagen

Se cargó una imagen a color utilizando OpenCV:

```
python
CopiarEditar
img = cv2.imread('tu_imagen.jpg')
```

### Dibujo sobre la imagen

Se realizaron los siguientes dibujos:

- **Línea verde:** Desde el punto (20, 20) hasta (280, 20), con grosor de 2 píxeles.
- **Rectángulo azul:** De las coordenadas (50, 50) a (250, 120), contorno de 2 píxeles.
- **Círculo rojo:** En el centro de la imagen (150, 84), de radio 30 píxeles y relleno sólido.
- **Texto blanco:** La palabra "ROI practica" colocada en la coordenada (50, 160), con tamaño de letra 0.6 y grosor 2.

### Definición de ROI

Se definió una región rectangular dentro de la imagen:

```
python
CopiarEditar
roi = img[50:120, 50:250]
```

Esta región contiene parte del rectángulo azul dibujado.

### Visualización

Se mostraron dos ventanas:

- Una ventana con la imagen completa y los dibujos.
- Una segunda ventana solo con la Región de Interés (ROI).

---

## Resultados

- Se pudo observar claramente los dibujos superpuestos a la imagen.
- El texto y las formas fueron visibles correctamente dentro de los límites de la imagen (300x168 píxeles).
- La Región de Interés fue recortada exitosamente y mostrada en una ventana aparte.

---

## Conclusiones

- Aprendí a **dibujar formas y textos** sobre imágenes usando funciones de OpenCV como `cv2.line`, `cv2.rectangle`, `cv2.circle` y `cv2.putText`.
- Comprendí la importancia de **respetar las dimensiones de la imagen** al definir coordenadas para los dibujos.
- Pude **segmentar una parte específica de la imagen (ROI)** para futuros trabajos de procesamiento de imágenes, como enfoque en detalles importantes o reducción de área de análisis.

## Código completo

```
import cv2

import numpy as np

# Cargar la imagen

img = cv2.imread('tu_imagen.jpg')

# Tamaño de la imagen

print("Tamaño de la imagen:", img.shape) # Debería imprimir (168, 300, 3)

# Dibujar una línea (en la parte superior)

cv2.line(img, (20, 20), (280, 20), (0, 255, 0), 2)

# Dibujar un rectángulo (en el centro)
```

```
cv2.rectangle(img, (50, 50), (250, 120), (255, 0, 0), 2)

# Dibujar un círculo (en el centro de la imagen)
cv2.circle(img, (150, 84), 30, (0, 0, 255), -1)

# Escribir texto (abajo de la imagen)
font = cv2.FONT_HERSHEY_SIMPLEX
cv2.putText(img, 'ROI practica', (50, 160), font, 0.6, (255, 255, 255), 2,
cv2.LINE_AA)

# Definir una región de interés (ROI) dentro de la imagen
roi = img[50:120, 50:250]

# Mostrar la imagen con los dibujos
cv2.imshow('Imagen con Dibujos', img)

# Mostrar solo la ROI recortada
cv2.imshow('ROI – Región de Interés', roi)

cv2.waitKey(0)
cv2.destroyAllWindows()
```