Instalando a biblioteca *Ultralytics*

```
!pip install ultralytics
```

Importando YOLO da Ultralytics

```
from ultralytics import YOLO

model = YOLO('yolo11n.pt')
```

Importando as bibliotecas de renderização de imagens

```
import matplotlib.pyplot as plt
import cv2
```

Criando o diretório de imagens e salvando o caminho da imagem

```
!mkdir -p /content/imagens

image_path = "/content/imagens/img1.png"
```

Teste de sanidade para conferência do caminho salvo

```
image = cv2.imread(image_path)
image_rgb = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
plt.imshow(image_rgb)
plt.axis('off')
plt.show()
```

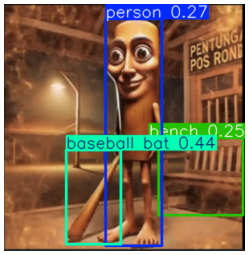
Resultado da aplicação da inferência do modelo na imagem existente

```
results = model(image_path)
```

```
image 1/1 /content/imagens/img1.png: 640x640 1 person, 1 bench, 1 baseball bat, 8.7ms
Speed: 10.7ms preprocess, 8.7ms inference, 346.4ms postprocess per image at shape (1, 3, 640, 640)
```

Mostra o que o modelo reconheceu e a porcentagem que ele acha que acertou

```
for result in results:
  boxes = result.boxes
  masks = result.masks
  keypoints = result.keypoints
  probs = result.probs
  obb = result.obb
  result.show()
```



Importando um novo modelo

```
model2 = YOLO('yolo11l.pt')
```

```
Downloading https://github.com/ultralytics/assets/releases/download/v8.3.0/yolo11l.pt to 'yolo11l.pt'...
100%|██████████| 49.0M/49.0M [00:00<00:00, 306MB/s]
```

Nova imagem

```
image2_path = "/content/imagens/img2.jpeg"
```

Novo teste de sanidade

```
image2 = cv2.imread(image2_path)
image2_rgb = cv2.cvtColor(image2, cv2.COLOR_BGR2RGB)
plt.imshow(image2_rgb)
plt.axis('off')
plt.show()
```

Nova inferência para a segunda imagem

```
results2 = model2(image2_path)
```

```
image 1/1 /content/imagens/img2.jpeg: 480x640 4 persons, 1 baseball bat, 1 baseball glove, 38.1ms
Speed: 2.7ms preprocess, 38.1ms inference, 1.7ms postprocess per image at shape (1, 3, 480, 640)
```

```
for result in results2:
  boxes = result.boxes
  masks = result.masks
  keypoints = result.keypoints
  probs = result.probs
  obb = result.obb
  result.show()
```