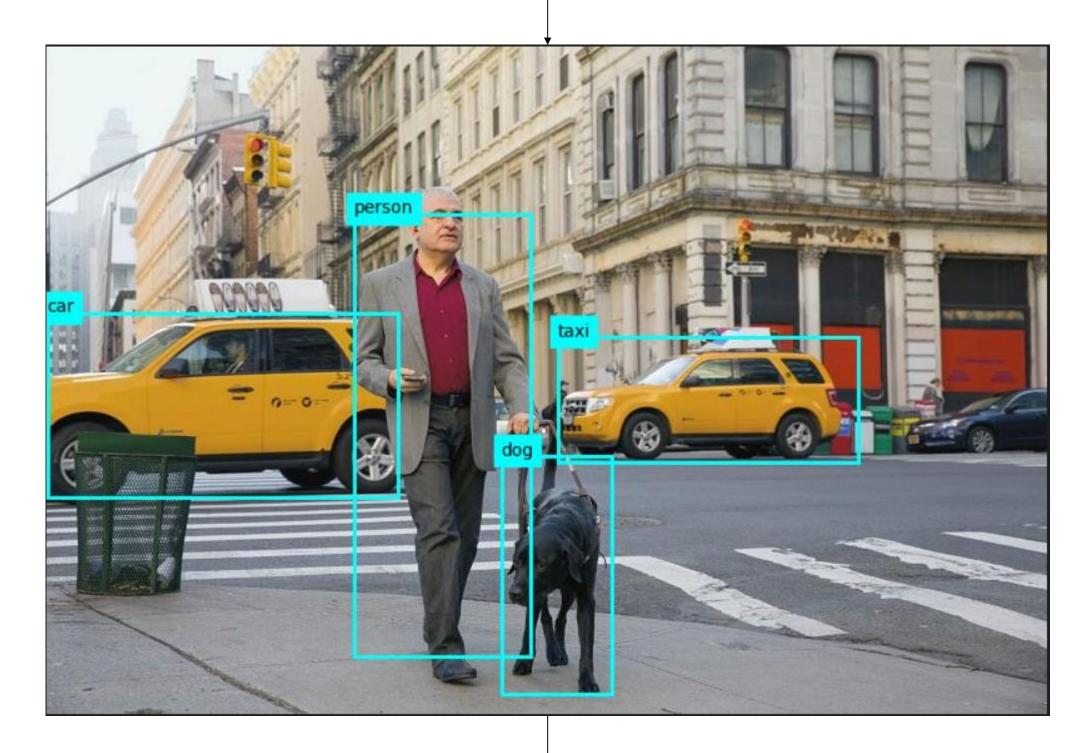
Se pueden obtener etiquetas relevantes

Get image tags
if result.tags is not None:
 print("\nTags:")
 for tag in result.tags.list:
 print(" Tag: '{}' (confidence: {:.2f}%)".format(tag.name, tag.confidence * 100))

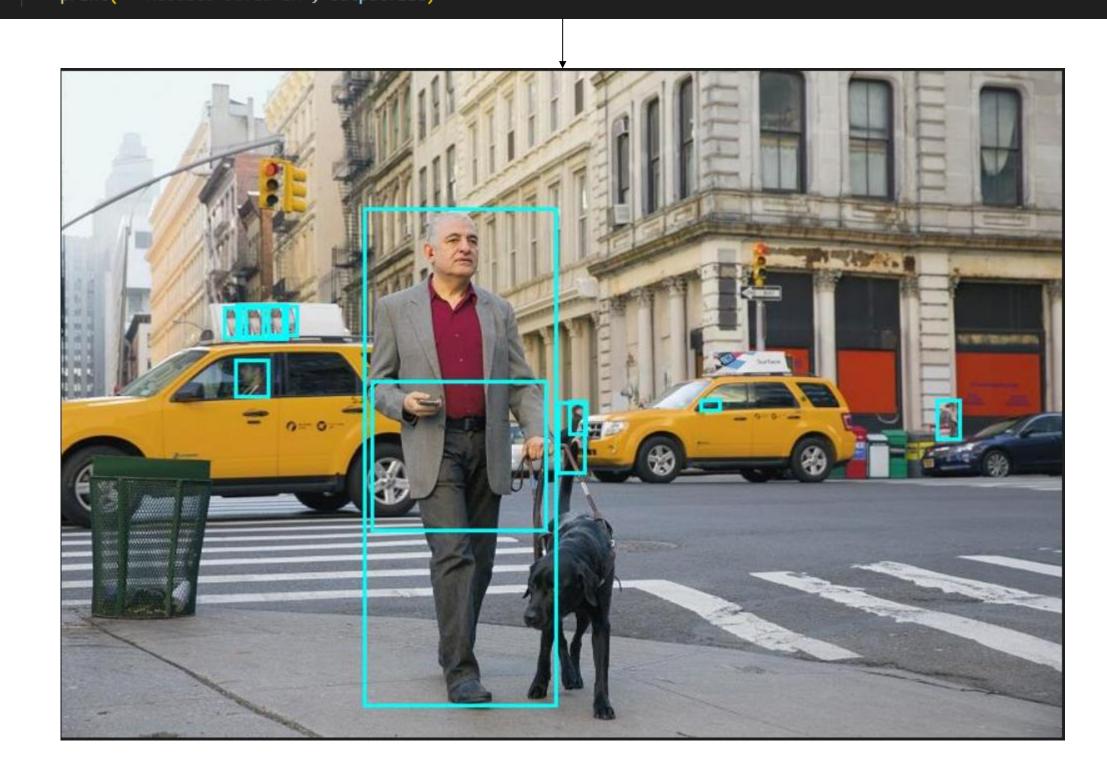
Se pueden obtener objetos en la imágen

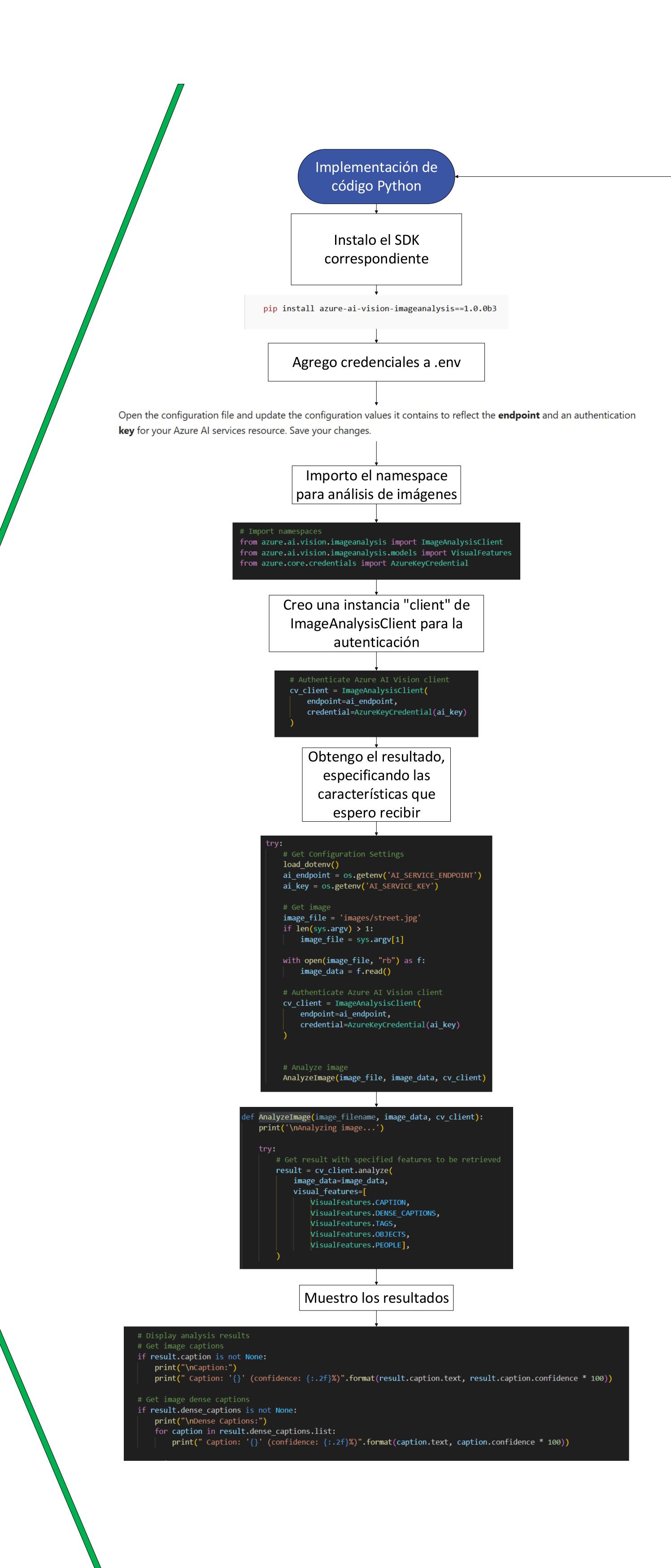
```
t objects in the image
if result.objects is not None:
print("\nObjects in image:")
# Prepare image for drawing
image = Image.open(image_filename)
fig = plt.figure(figsize=(image.width/100, image.height/100))
plt.axis('off')
draw = ImageDraw.Draw(image)
color = 'cyan'
 for detected_object in result.objects.list:
   # Print object name
   print(" {} (confidence: {:.2f}%)".format(detected_object.tags[0].name, detected_object.tags[0].confidence * 100))
    # Draw object bounding box
    r = detected_object.bounding_box
   bounding_box = ((r.x, r.y), (r.x + r.width, r.y + r.height))
draw.rectangle(bounding_box, outline=color, width=3)
    plt.annotate(detected_object.tags[0].name,(r.x, r.y), backgroundcolor=color)
  # Save annotated image
  plt.imshow(image)
 plt.tight_layout(pad=0)
outputfile = 'objects.jpg'
 fig.savefig(outputfile)
print(' Results saved in', outputfile)
```



Se pueden detectar y ubicar personas en la imágen

```
Get people in the image
 result.people is not None:
print("\nPeople in image:")
 # Prepare image for drawing
 image = Image.open(image_filename)
fig = plt.figure(figsize=(image.width/100, image.height/100))
  plt.axis('off')
 draw = ImageDraw.Draw(image)
 color = 'cyan'
  for detected_people in result.people.list:
    # Draw object bounding box
    r = detected_people.bounding_box
    bounding_box = ((r.x, r.y), (r.x + r.width, r.y + r.height))
    draw.rectangle(bounding_box, outline=color, width=3)
    # Return the confidence of the person detected
    print(" {} (confidence: {:.2f}%)".format(detected_people.bounding_box, detected_people.confidence * 100))
  # Save annotated image
  plt.imshow(image)
  plt.tight_layout(pad=0)
 outputfile = 'people.jpg'
 fig.savefig(outputfile)
print(' Results saved in', outputfile)
```





Analyze Images with Azure Al Vision

Provision an Azure Al Services resource

If you don't already have one in your subscription, you'll need to provision an Azure Al Services resource.

- 1. Open the Azure portal at https://portal.azure.com, and sign in using the Microsoft account associated with your Azure subscription.
- Select Create a resource.
 In the search bar, search for Azure Al services, select Azure Al Services, and create an Azure Al services multi-service account resource with the following settings:
- Subscription: Your Azure subscription
- Resource group: Choose or create a resource group (if you are using a restricted subscription, you may not have permission to create a new resource group - use the one provided)
- **Region**: Choose from East US, West US, France Central, Korea Central, North Europe, Southeast Asia, West Europe, or East
- Name: Enter a unique name
- Pricing tier: Standard S0
- *Azure Al Vision 4.0 full feature sets are currently only available in these regions.
- 4. Select the required checkboxes and create the resource.5. Wait for deployment to complete, and then view the deployment details.
- 6. When the resource has been deployed, go to it and view its **Keys and Endpoint** page. You will need the endpoint and one of the keys from this page in the next procedure.

Adiciono el paquete nuguet con el admin de paquetes o por terminal

dotnet add package Azure.AI.Vision.ImageAnalysis -v 1.0.0-beta.3

Agrego credenciales a appsettingst.json

Open the configuration file and update the configuration values it contains to reflect the **endpoint** and an authentication **key** for your Azure AI services resource. Save your changes.

Importo el namespace para análisis de imágenes

// Import namespaces
using Azure.AI.Vision.ImageAnalysis

Creo una instancia "client" de ImageAnalysisClient para la autenticación