Code showing cutom model used for prediction using our own endpoint URL and probability of object detection for 5 images

Import libraries

In [68]:
```
from msrest.authentication import ApiKeyCredentials
from azure.cognitiveservices.vision.customvision.prediction import CustomVisionP
from azure.storage.blob import (
    BlobServiceClient,
    generate_blob_sas,
    BlobSasPermissions,
)
from azure.storage.blob import generate_blob_sas, BlobSasPermissions
from datetime import datetime, timedelta
from azure.cognitiveservices.vision.customvision.prediction.models import ImageU
```

Define credentials

In [ ]:
```
ENDPOINT      = 'https://pythonejercicio-prediction.cognitiveservices.azure.com/
PREDICTION_KEY = 'prediction_key_here'  '
PROJECT_ID     = "9b1fa97e-0db3-46ee-93b4-9eead57c48d2"
PUBLISHED_NAME = "Iteration2"
```

In [ ]:
```
ACCOUNT_URL     = "https://001finalproject.blob.core.windows.net/"
ACCOUNT_KEY     = "account_key_here"
CONTAINER_NAME  = "step4"
BLOB_PREFIX     = ""
```

In [72]:
```
blob_service = BlobServiceClient(account_url=ACCOUNT_URL, credential=ACCOUNT_KEY
container    = blob_service.get_container_client(CONTAINER_NAME)

def sas_url(blob_name: str) -> str:
    sas = generate_blob_sas(
        account_name    = blob_service.account_name,
        container_name = CONTAINER_NAME,
        blob_name      = blob_name,
        account_key    = ACCOUNT_KEY,
        permission     = BlobSasPermissions(read=True),
        expiry         = datetime.utcnow() + timedelta(hours=1),
        version        = "2023-11-03"
    )
    return f"{ACCOUNT_URL}{CONTAINER_NAME}/{blob_name}?{sas}"

blob_urls = [
    sas_url(b.name)
    for b in container.list_blobs()
    if b.name.lower().endswith((".jpg", ".jpeg", ".png"))
]

print(f"▶️ {len(blob_urls)} imágenes encontradas.")
```

```
▶️ 5 imágenes encontradas.
```

In [73]:
```
credentials = ApiKeyCredentials(in_headers={"Prediction-key": PREDICTION_KEY})
predictor   = CustomVisionPredictionClient(ENDPOINT, credentials)

# --- Recorrer blobs e inferir por contenido (NO URL) ---
for blob in container.list_blobs():
    if not blob.name.lower().endswith((".jpg", ".jpeg", ".png")):
        continue

    blob_client = blob_service.get_blob_client(container=CONTAINER_NAME, blob=bl
    img_bytes   = blob_client.download_blob().readall()

    res = predictor.detect_image(PROJECT_ID, PUBLISHED_NAME, img_bytes)

    print(f"\n📷 {blob.name}")
    hits = [p for p in res.predictions if p.probability > THRESHOLD]

    if not hits:
        print(f"   ❌ Sin objetos > {THRESHOLD:.0%}")
    else:
        for p in hits:
            print(f"   ✅ {p.tag_name:7s} → {p.probability:.2%}")
```

```
📷 lighter_test_set_1of5.jpg
   ✅ Lighter → 54.83%
   ✅ Lighter → 44.31%
   ✅ Lighter → 41.76%
   ✅ Lighter → 27.41%

📷 lighter_test_set_2of5.jpg
   ✅ Lighter → 49.67%
   ✅ Lighter → 45.28%
   ✅ Lighter → 31.26%
   ✅ Lighter → 30.93%

📷 lighter_test_set_3of5.jpg
   ✅ Lighter → 63.36%
   ✅ Lighter → 30.63%
   ✅ Lighter → 29.97%

📷 lighter_test_set_4of5.jpg
   ✅ Lighter → 70.95%
   ✅ Lighter → 59.93%

📷 lighter_test_set_5of5.jpg
   ✅ Lighter → 26.78%
```