

CAPTURAS DEL CODIGO DEL SERVIDOR PYTHON.

- Importamos los paquetes que utilizaremos
- Tambien utilizamos las variables de entorno

```
Servidor-Python > server.py > ...
  Click here to ask Blackbox to help you code faster
1  from flask import Flask, request, jsonify
2  from dotenv import load_dotenv
3  import boto3
4  import base64
5  import os
6
7  app = Flask(__name__)
8
9  # Cargar variables de entorno desde el archivo .env
10 load_dotenv()
11
12 # Obtener las variables de entorno
13 AWS_ACCESS_KEY_ID = os.getenv('AWS_ACCESS_KEY_ID')
14 AWS_SECRET_ACCESS_KEY = os.getenv('AWS_SECRET_ACCESS_KEY')
15 AWS_REGION = os.getenv('AWS_REGION')
16
17 # Configurar el cliente de Rekognition
18 rekognition_client = boto3.client('rekognition',
19                                  aws_access_key_id=AWS_ACCESS_KEY_ID,
20                                  aws_secret_access_key=AWS_SECRET_ACCESS_KEY,
21                                  region_name=AWS_REGION)
22
23 @app.route('/tarea3-<carne>', methods=['POST'])
24 def procesar_imagen(carne):
25     if 'file' not in request.files:
26         return jsonify({'error': 'No se proporcionó ningún archivo'}), 400
27
28     file = request.files['file']
29     if file.filename == '':
```

- Cargamos la imagen al servidor y despues sacamos su base 64
- utilizamos boto3 para el cliente de aws

```

28     file = request.files['file']
29     if file.filename == '':
30         return jsonify({'error': 'No se seleccionó ningún archivo'}), 400
31
32     # Guardar la imagen en el servidor
33     filename = file.filename
34     file_path = os.path.join('./uploads', filename)
35     file.save(file_path)
36
37     # Leer la imagen en formato base64
38     with open(file_path, "rb") as image_file:
39         encoded_string = base64.b64encode(image_file.read()).decode()
40
41     # Eliminar la imagen del servidor
42     os.remove(file_path)
43
44     # Enviar la imagen a Rekognition
45     response = rekognition_client.detect_labels(
46         Image={'Bytes': base64.b64decode(encoded_string)})
47
48
49     # Obtener las etiquetas
50     relevant_data = []
51     for label in response['Labels']:
52         label_data = {
53             'Nombre': label['Name'],
54             'Confianza': label['Confidence'],
55             'Detalles': label['Instances'],
56             'Textos': label.get('TextInstances', [])
57         }
58         relevant_data.append(label_data)
59
60     return jsonify({'imagen': filename, 'datos_relevantes': relevant_data}), 200
61
62 if __name__ == '__main__':
63     app.run(debug=True)
64
```

- Recibimos la respuesta y parsemos los campos que queremos
- Devolvemos los campos en forma de json

```

# Obtener las etiquetas
relevant_data = []
for label in response['Labels']:
    label_data = {
        'Nombre': label['Name'],
        'Confianza': label['Confidence'],
        'Detalles': label['Instances'],
        'Textos': label.get('TextInstances', [])
    }
    relevant_data.append(label_data)

return jsonify({'imagen': filename, 'datos_relevantes': relevant_data}), 200

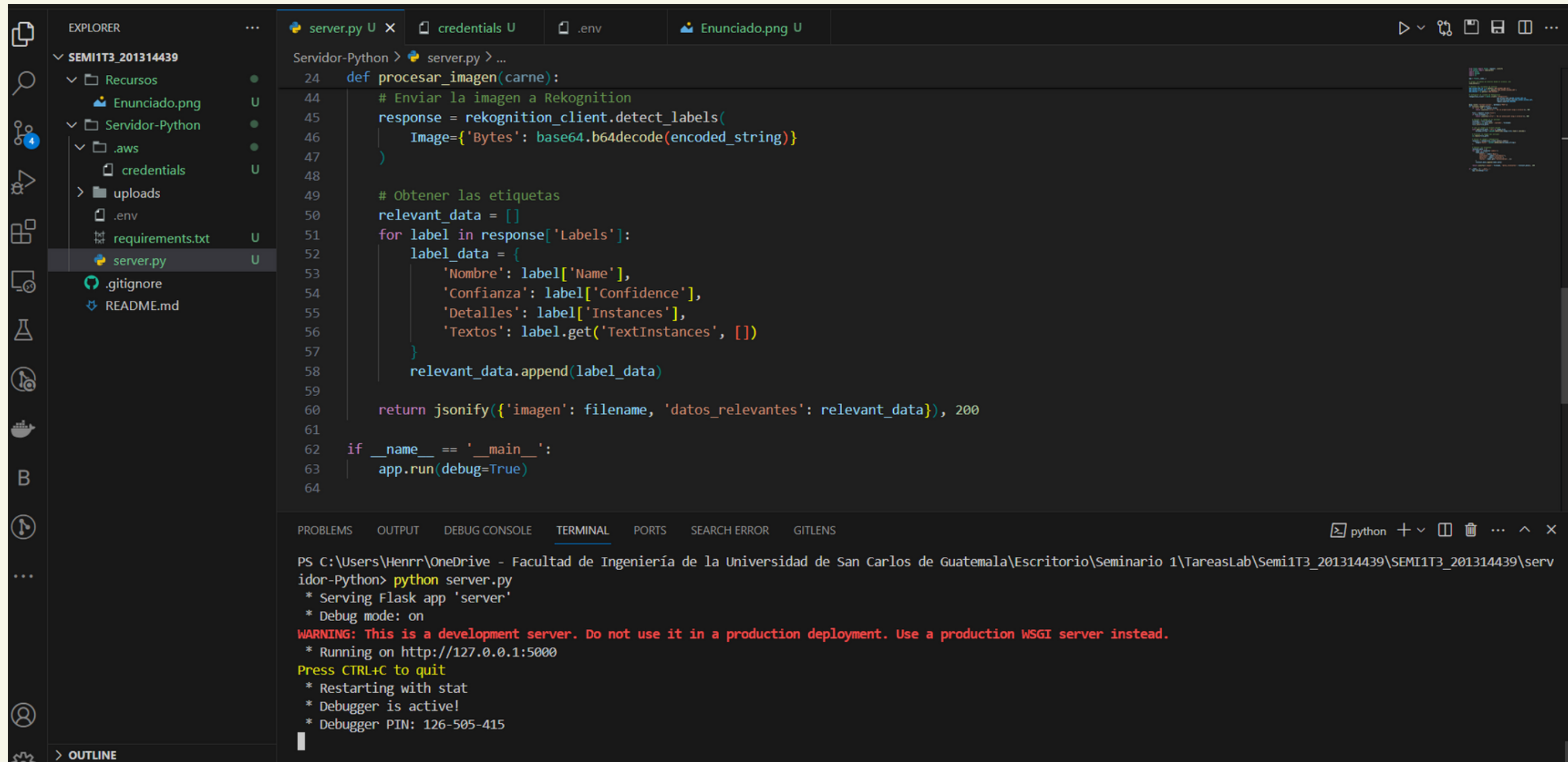
if __name__ == '__main__':
    app.run(debug=True)

```

CAPTURA DEL SERVIDOR CORRIENDO.

- vemos que el puerto esta en el la siguiente ruta

http://localhost:5000



The screenshot shows a Visual Studio Code editor with a Python project. The Explorer sidebar on the left shows the file structure: SEMI1T3_201314439 > Recursos > Enunciado.png, SEMI1T3_201314439 > Servidor-Python > .aws > credentials, SEMI1T3_201314439 > Servidor-Python > uploads > .env, SEMI1T3_201314439 > Servidor-Python > requirements.txt, SEMI1T3_201314439 > Servidor-Python > server.py, SEMI1T3_201314439 > .gitignore, SEMI1T3_201314439 > README.md. The main editor shows the server.py file with the following code:

```
24 def procesar_imagen(carne):
44     # Enviar la imagen a Rekognition
45     response = rekognition_client.detect_labels(
46         Image={'Bytes': base64.b64decode(encoded_string)}
47     )
48
49     # Obtener las etiquetas
50     relevant_data = []
51     for label in response['Labels']:
52         label_data = {
53             'Nombre': label['Name'],
54             'Confianza': label['Confidence'],
55             'Detalles': label['Instances'],
56             'Textos': label.get('TextInstances', [])
57         }
58         relevant_data.append(label_data)
59
60     return jsonify({'imagen': filename, 'datos_relevantes': relevant_data}), 200
61
62 if __name__ == '__main__':
63     app.run(debug=True)
64
```

The bottom panel shows the TERMINAL output:

```
PS C:\Users\Henrr\OneDrive - Facultad de Ingeniería de la Universidad de San Carlos de Guatemala\Escritorio\Seminario 1\TareasLab\Semi1T3_201314439\SEMI1T3_201314439\serv
idor-Python> python server.py
* Serving Flask app 'server'
* Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on http://127.0.0.1:5000
Press CTRL+C to quit
* Restarting with stat
* Debugger is active!
* Debugger PIN: 126-505-415
```


CAPTURA DEL JSON INGRESADO Y RUTA CREADA (POSTMAN).

server.py U × credentials U .env Enunciado.png U

Servidor-Python > server.py > ...

```
24 def procesar_imagen(carne):
44     # Enviar la imagen a Rekognition
45     response = rekognition_client.detect_labels(
46         Image={'Bytes': base64.b64decode(encoded_string)}
47     )
48
49     # Obtener las etiquetas
50     relevant_data = []
51     for label in response['Labels']:
52         label_data = {
53             'Nombre': label['Name'],
54             'Confianza': label['Confidence'],
55             'Detalles': label['Instances'],
56             'Textos': label.get('TextInstances', [])
57         }
58         relevant_data.append(label_data)
59
60     return jsonify({'imagen': filename, 'datos_relevantes': relevant_data})
61
62 if __name__ == '__main__':
63     app.run(debug=True)
64
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS SEARCH ERROR GITLENS

PS C:\Users\Henrr\OneDrive - Facultad de Ingeniería de la Universidad de San Carlos -> python server.py
* Serving Flask app 'server'
* Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment
* Running on http://127.0.0.1:5000
Press CTRL+C to quit
* Restarting with stat
* Debugger is active!
* Debugger PIN: 126-505-415
127.0.0.1 - - [04/Mar/2024 21:57:28] "POST /tarea3-201314439 HTTP/1.1" 200 -

GET http://localhost:5000/tarea3-201314439

POST http://localhost:5000/tarea3-201314439

Params Auth Headers (8) Body Pre-req. Tests Settings

form-data

Key	Value	Description
<input checked="" type="checkbox"/> file	File	Will-Smith-...
Key	Text	Value

Body

200 OK 1311 ms 4.02 KB Save as example

Pretty Raw Preview Visualize JSON

```
1 {
2   "datos_relevantes": [
3     {
4       "Confianza": 98.0057601928711,
5       "Detalles": [
6         {
7           "BoundingBox": {
8             "Height": 0.45350298285484314,
9             "Left": 0.20361244678497314,
10            "Top": 0.38804176449775696,
11            "Width": 0.5543646216392517
12          },
13          "Confidence": 98.0057601928711
14        }
15      ],
16      "Nombre": "Adult",
17      "Textos": []
18    }
19  ]
20 }
```

Se envia la imagen como tal no como base 64 si no se hace la conversion dentro del servidor como podemos ver en esta area

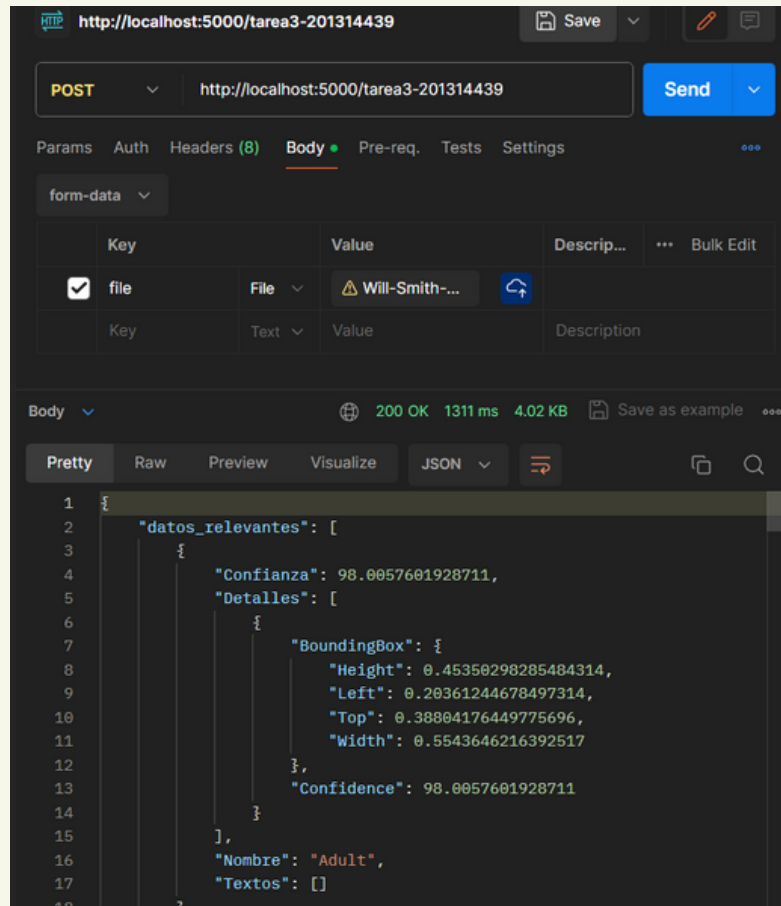
hacemos uso del endpoint:

<http://localhost:5000/tarea3-201314439>

Imagen que se cargo



CAPTURA DE LAS ETIQUETAS QUE DEVUELVA REKOGNITION PARA LA IMAGEN ENVIADA.



18	}	35	}	53	"Detalles": [70	"Nombre": "Standing",	87	"Detalles": [],
19	{	36		54		71	"Textos": []	88	"Nombre": "Street",
20		37	"Confianza": 98.0057601928711,	55	"Detalles": [72		89	"Textos": []
21	"Confianza": 98.0057601928711,	38		56	{	73		90	
22	"Detalles": [39	{	57	"BoundingBox": {	74	"Confianza": 79.71790313720703,	91	
23	{	40	"BoundingBox": {	58	"Height": 0.45350298285484314,	75	"Detalles": [],	92	"Confianza": 78.4168472290039,
24	"BoundingBox": {	41	"Height": 0.45350298285484314,	59	"Left": 0.20361244678497314,	76	"Nombre": "City",	93	"Detalles": [],
25	"Height": 0.45350298285484314,	42	"Left": 0.20361244678497314,	60	"Top": 0.38804176449775696,	77	"Textos": []	94	"Nombre": "Urban",
26	"Left": 0.20361244678497314,	43	"Top": 0.38804176449775696,	61	"Width": 0.5543646216392517	78		95	"Textos": []
27	"Top": 0.38804176449775696,	44		62		79		96	
28	"Width": 0.5543646216392517	45	},	63	"Confidence": 98.0057601928711	80		97	
29	},	46	"Confidence": 98.0057601928711	64],	81	"Confianza": 78.4168472290039,	98	"Confianza": 75.09879302978516,
30	"Confidence": 98.0057601928711	47	},	65	"Nombre": "Person",	82	"Detalles": [],	99	"Detalles": [],
31	},	48	"Nombre": "Man",	66	"Textos": []	83	"Nombre": "Road",	100	"Nombre": "Arch",
32	"Nombre": "Male",	49	"Textos": []	67		84	"Textos": []	101	"Textos": []
33	"Textos": []	50	},	68		85		102	
34	}	51	{	69	"Confianza": 86.2803955070125,	86		103	
35		52	"Confianza": 98.0057601928711,	70	"Detalles": [],	87	"Confianza": 78.4168472290039,	104	"Confianza": 75.09879302978516,
		53	"Detalles": [71	"Nombre": "Standing",		"Detalles": [],		

105	"Detalles": [],	123	"Detalles": [138	"Confianza": 63.23008728027344,	156	"Confianza": 57.6481819152832,	163	"Detalles": [],
106	"Nombre": "Architecture",	124	{	139	"Detalles": [],	157	"Detalles": [],	164	"Nombre": "Housing",
107	"Textos": []	125	"BoundingBox": {	140	"Nombre": "Building",	158	"Nombre": "Mosque Of Cordoba",	165	"Textos": []
108		126	"Height": 0.09246286749839783,	141	"Textos": []	159		166	
109		127	"Left": 0.8736579418182373,	142		160		167	
110	"Confianza": 75.09879302978516,	128	"Top": 0.45089128613471985,	143		161		168	"Confianza": 55.08332824707031,
111	"Detalles": [],	129	"Width": 0.09440509229898453	144	"Confianza": 63.23008728027344,	162	"Confianza": 55.303565979003906,	169	"Detalles": [],
112	"Nombre": "Gothic Arch",	130	},	145	"Detalles": [],	163	"Detalles": [],	170	"Nombre": "Performer",
113	"Textos": []	131	"Confidence": 65.78656005859375	146	"Nombre": "Clock Tower",	164	"Nombre": "Housing",	171	"Textos": []
114		132	},	147	"Textos": []	165		172	
115		133	},	148		166		173	
116	"Confianza": 67.42974853515625,	134	"Nombre": "Plant",	149		167		174	"Confianza": 55.08332824707031,
117	"Detalles": [],	135	"Textos": []	150	"Confianza": 63.23008728027344,	168	"Confianza": 55.08332824707031,	175	"Detalles": [],
118	"Nombre": "Landmark",	136		151	"Detalles": [],	169	"Detalles": [],	176	"Nombre": "Solo Performance",
119	"Textos": []	137		152	"Nombre": "Tower",	170	"Nombre": "Performer",	177	"Textos": []
120		138	"Confianza": 63.23008728027344,	153	"Textos": []	171	"Textos": []	178	
121		139	"Detalles": [],	154		172		179	
122	"Confianza": 65.78656005859375,	140	"Nombre": "Building",	155		173		180	"imagen": "Will-Smith-en-Angtigua-Guatemala.jpg"

LINK GITHUB DE CODIGO EJECUTADO:

[HTTPS://GITHUB.COM/HENRRYBRAN-](https://github.com/HENRRYBRAN-)

HUB/SEMI1T3_201314439/TREE/MAIN