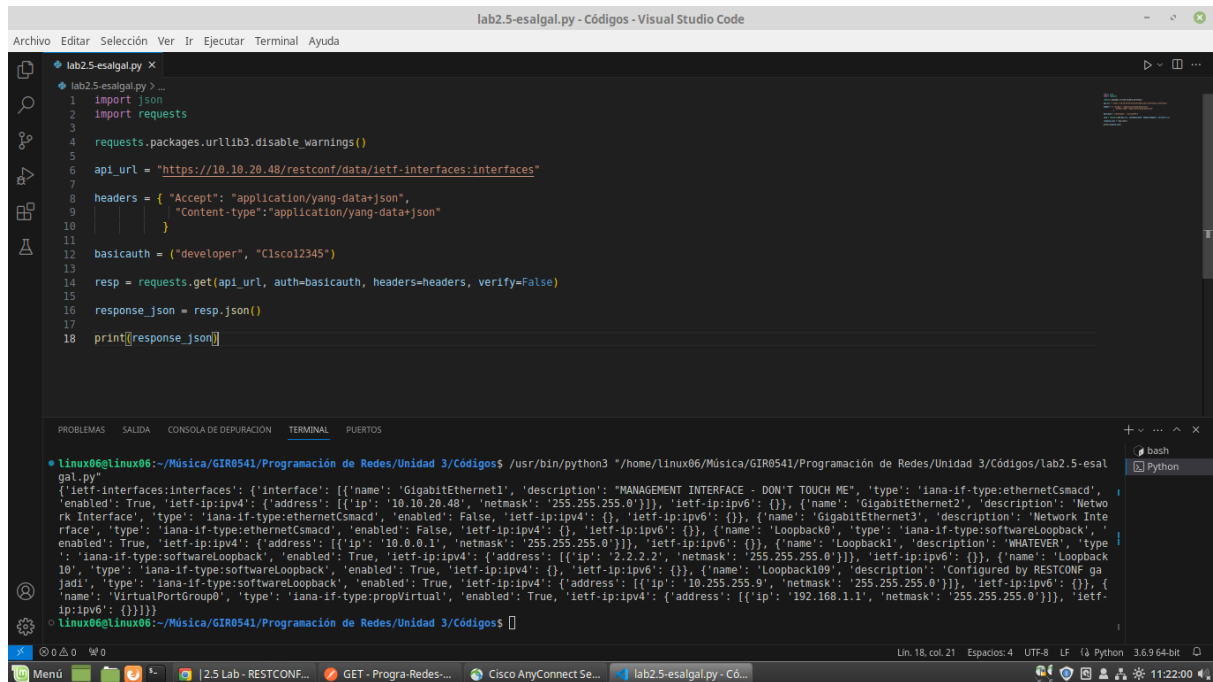


Nombre: Eder Gael Saldaña Galván
Fecha: 27/Noviembre/2023

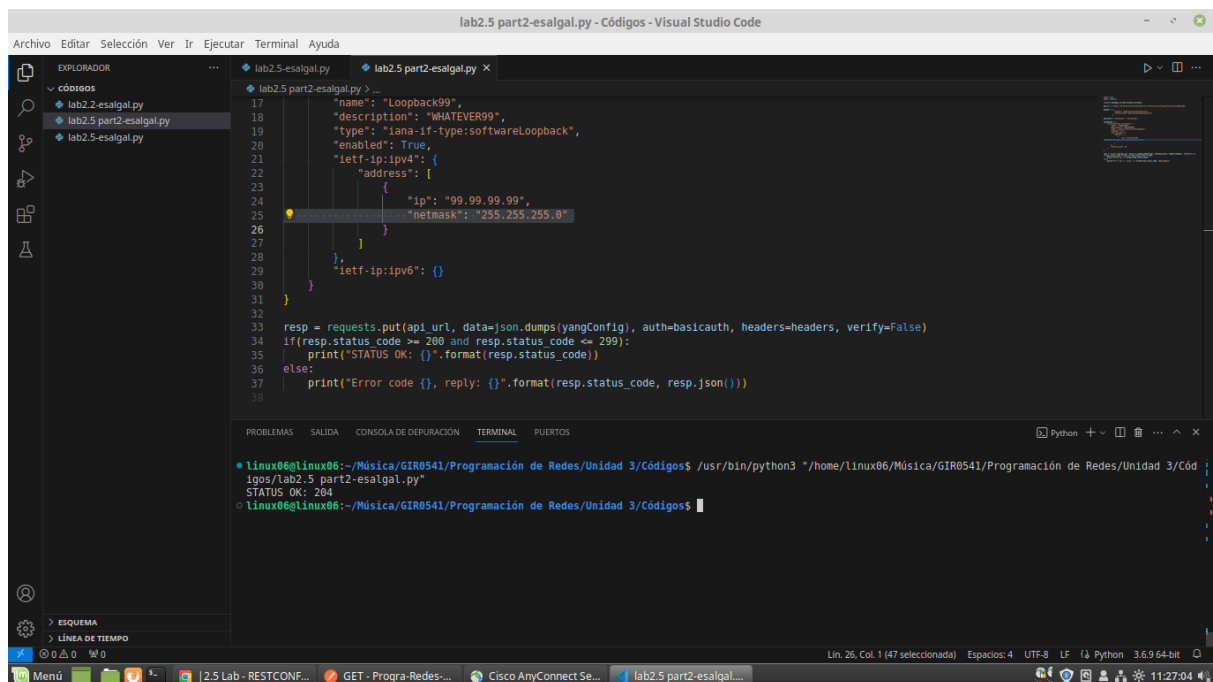
Grupo: GIR0541
Laboratorio: 2.5 Parte 1 y 2



The screenshot shows the Visual Studio Code editor with the file `lab2.5-esalgal.py` open. The code is a Python script that uses the `requests` library to send a GET request to a RESTCONF endpoint. The script defines the API URL, headers, and basic authentication credentials. It then sends the request and prints the response JSON.

```
1 import json
2 import requests
3
4 requests.packages.urllib3.disable_warnings()
5
6 api_url = "https://10.10.20.48/restconf/data/ietf-interfaces:interfaces"
7
8 headers = { "Accept": "application/yang-data+json",
9             "Content-type": "application/yang-data+json"
10            }
11
12 basicauth = ("developer", "Cisc0l2345")
13
14 resp = requests.get(api_url, auth=basicauth, headers=headers, verify=False)
15
16 response_json = resp.json()
17
18 print(response_json)
```

The terminal output shows the execution of the script, displaying the JSON response from the RESTCONF endpoint. The response includes details about the network interfaces, such as `GigabitEthernet1`, `GigabitEthernet2`, `GigabitEthernet3`, `Loopback0`, `Loopback1`, `Loopback10`, `Loopback109`, `SoftwareLoopback`, and `VirtualPortGroup0`.



The screenshot shows the Visual Studio Code editor with the file `lab2.5 part2-esalgal.py` open. The code is a Python script that uses the `requests` library to send a PUT request to a RESTCONF endpoint. The script defines the API URL, headers, and basic authentication credentials. It then sends the request and prints the response status code and JSON.

```
17
18 "description": "Loopback99",
19 "type": "iana-if-type:softwareLoopback",
20 "enabled": True,
21 "ietf-ip:ipv4": {
22     "address": [
23         {
24             "ip": "99.99.99.99",
25             "netmask": "255.255.255.0"
26         }
27     ],
28     "ietf-ip:ipv6": {}
29 }
30
31 resp = requests.put(api_url, data=json.dumps(yangConfig), auth=basicauth, headers=headers, verify=False)
32
33 if (resp.status_code >= 200 and resp.status_code <= 299):
34     print("STATUS OK: {}".format(resp.status_code))
35 else:
36     print("Error code {}, reply: {}".format(resp.status_code, resp.json()))
37
38
```

The terminal output shows the execution of the script, displaying the status code and the JSON response from the RESTCONF endpoint. The status code is 204, indicating a successful PUT request.