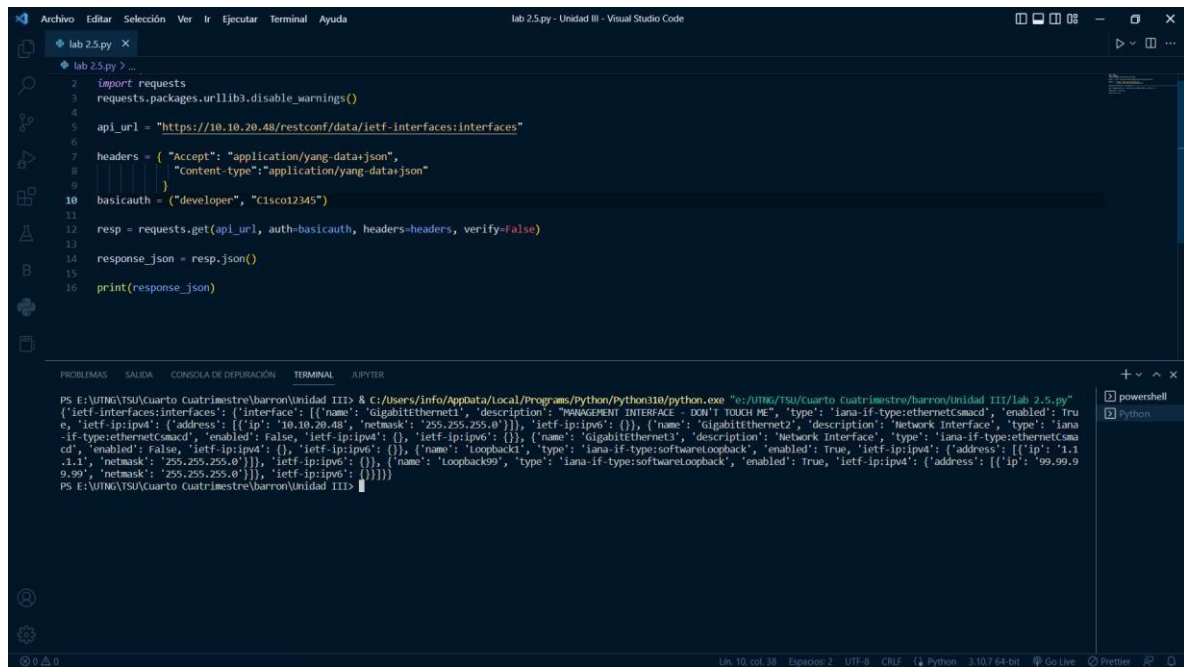


RESTCONF in Python



The screenshot shows a Visual Studio Code window with a Python file named 'lab 2.5.py'. The script uses the 'requests' library to perform a GET request to a RESTCONF endpoint. The terminal output displays the JSON response from the server, which contains network interface information.

```
1 import requests
2 requests.packages.urllib3.disable_warnings()
3
4 api_url = "https://10.10.20.48/restconf/data/ietf-interfaces:interfaces"
5
6
7 headers = { "Accept": "application/yang-data+json",
8             "Content-type": "application/yang-data+json"
9           }
10 basicauth = ("developer", "Cisco12345")
11
12 resp = requests.get(api_url, auth=basicauth, headers=headers, verify=False)
13
14 response_json = resp.json()
15
16 print(response_json)
```

```
PS E:\UTING\TSU\Cuarto Cuatrimestre\barron\Unidad III> & C:\Users\info\AppData\Local\Programs\Python\Python310\python.exe "e:\UTING\TSU\Cuarto Cuatrimestre\barron\Unidad III\lab 2.5.py"
{'ietf-interfaces:interfaces': {'interface': [{'name': 'GigabitEthernet1', 'description': 'MANAGEMENT INTERFACE - DON'T TOUCH ME', 'type': 'iana-if-type:ethernetCsmacd', 'enabled': True, 'ietf-ip:ipv4': {'address': [{'ip': '10.10.20.48', 'netmask': '255.255.255.0'}]}, 'ietf-ip:ipv6': {}}, {'name': 'GigabitEthernet2', 'description': 'Network Interface', 'type': 'iana-if-type:ethernetCsmacd', 'enabled': False, 'ietf-ip:ipv4': {}, 'ietf-ip:ipv6': {}}, {'name': 'GigabitEthernet3', 'description': 'Network Interface', 'type': 'iana-if-type:ethernetCsmacd', 'enabled': False, 'ietf-ip:ipv4': {}, 'ietf-ip:ipv6': {}}, {'name': 'Loopback1', 'type': 'iana-if-type:softwareLoopback', 'enabled': True, 'ietf-ip:ipv4': {'address': [{'ip': '1.1.1.1', 'netmask': '255.255.255.0'}]}, 'ietf-ip:ipv6': {}}, {'name': 'Loopback99', 'type': 'iana-if-type:softwareLoopback', 'enabled': True, 'ietf-ip:ipv4': {'address': [{'ip': '99.99.99.99', 'netmask': '255.255.255.0'}]}, 'ietf-ip:ipv6': {}}]}}
```

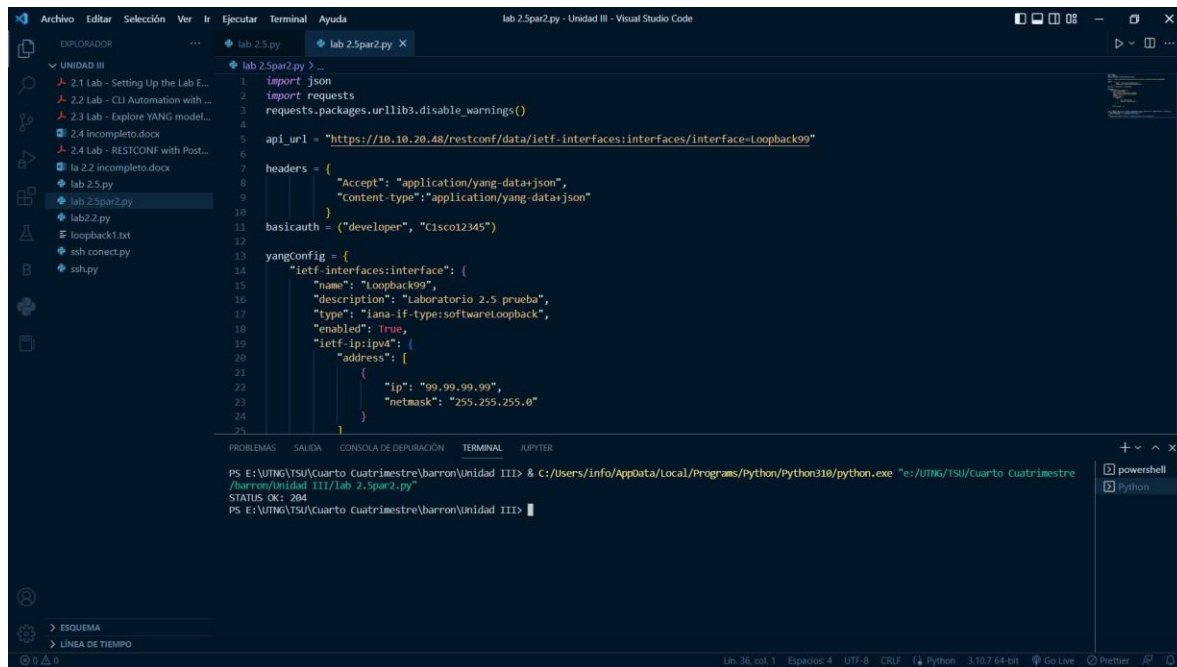
Salida:



This block shows the terminal output of the Python script, displaying the JSON response from the RESTCONF endpoint. The output is a nested dictionary representing the network interface configuration.

```
PS E:\UTING\TSU\Cuarto Cuatrimestre\barron\Unidad III> & C:\Users\info\AppData\Local\Programs\Python\Python310\python.exe "e:\UTING\TSU\Cuarto Cuatrimestre\barron\Unidad III\lab 2.5.py"
{'ietf-interfaces:interfaces': {'interface': [{'name': 'GigabitEthernet1', 'description': 'MANAGEMENT INTERFACE - DON'T TOUCH ME', 'type': 'iana-if-type:ethernetCsmacd', 'enabled': True, 'ietf-ip:ipv4': {'address': [{'ip': '10.10.20.48', 'netmask': '255.255.255.0'}]}, 'ietf-ip:ipv6': {}}, {'name': 'GigabitEthernet2', 'description': 'Network Interface', 'type': 'iana-if-type:ethernetCsmacd', 'enabled': False, 'ietf-ip:ipv4': {}, 'ietf-ip:ipv6': {}}, {'name': 'GigabitEthernet3', 'description': 'Network Interface', 'type': 'iana-if-type:ethernetCsmacd', 'enabled': False, 'ietf-ip:ipv4': {}, 'ietf-ip:ipv6': {}}, {'name': 'Loopback1', 'type': 'iana-if-type:softwareLoopback', 'enabled': True, 'ietf-ip:ipv4': {'address': [{'ip': '1.1.1.1', 'netmask': '255.255.255.0'}]}, 'ietf-ip:ipv6': {}}, {'name': 'Loopback99', 'type': 'iana-if-type:softwareLoopback', 'enabled': True, 'ietf-ip:ipv4': {'address': [{'ip': '99.99.99.99', 'netmask': '255.255.255.0'}]}, 'ietf-ip:ipv6': {}}]}}
```

Modificar la configuración de la interfaz con RESTCONF en Python



```
1 import json
2 import requests
3 requests.packages.urllib3.disable_warnings()
4
5 api_url = "https://10.10.20.48/restconf/data/ietf-interfaces:interfaces/interface-Loopback99"
6
7 headers = {
8     "Accept": "application/yang-data+json",
9     "Content-type": "application/yang-data+json"
10 }
11
12 basicauth = ("developer", "Cisco12345")
13
14 yangConfig = {
15     "ietf-interfaces:interface": {
16         "name": "Loopback99",
17         "description": "Laboratorio 2.5 prueba",
18         "type": "iana-if-type:software-loopback",
19         "enabled": True,
20         "ietf-ip:ipv4": {
21             "address": {
22                 "ip": "99.99.99.99",
23                 "netmask": "255.255.255.0"
24             }
25         }
26     }
27 }
```

```
PS E:\UTNG\TSU\Cuarto Cuatrimestre\barron\Unidad III> & C:\Users\info\AppData\Local\Programs\Python\Python310\python.exe "e:/UTNG/TSU/Cuarto Cuatrimestre/barron/Unidad III/lab 2.5par2.py"
STATUS OK: 204
PS E:\UTNG\TSU\Cuarto Cuatrimestre\barron\Unidad III>
```

Salida:

```
PS E:\UTNG\TSU\Cuarto Cuatrimestre\barron\Unidad III> & C:\Users\info\AppData\Local\Programs\Python\Python310\python.exe "e:/UTNG/TSU/Cuarto Cuatrimestre/barron/Unidad III/lab 2.5par2.py"
STATUS OK: 204
PS E:\UTNG\TSU\Cuarto Cuatrimestre\barron\Unidad III>
```

Conclusión: en este ejercicio se realizó la misma consulta que en laboratorio anterior con postman pero ahora con una interfaz de solo código de Python utilizando el protocolo REQUEST y los modelos Yang para la salida de la ejecución.

¿Qué es RESTCONF?

El propósito primordial de RESTCONF es permitir que las aplicaciones web (que usan el protocolo HTTP) accedan a un dispositivo de red para la compra de datos de configuración y estado.

RESTCONF se divide primordialmente en 2 capas: capa de contenido y capa de protocolo.

Capa de contenido: define una recolección de objetos para ser operados, definidos en lenguaje YANG, y distingue entre datos de administración y de configuración.

Capa de protocolo: Usa de manera directa el protocolo HTTP (+TLS) para conceder funcionalidades diversas del contenido.

Interacción entre NETCONF y RESTCONF

NETCONF es un protocolo para mandar configuraciones a los dispositivos de red. Posibilita a los usuarios centrarse en los datos de configuración en vez de en los comandos de configuración.

No obstante, diferentes dispositivos necesitan diferentes construcciones de configuración.

Esta diferencia se resuelve por medio de la definición del modelo yang, que es un lenguaje de modelado de datos. Los usuarios solamente deben mejorar la configuración solicitada en el modelo yang para configurar los accesorios subyacente.

El papel de RESTCONF está en la capa de servicio web. Conforme con la URL, rellena la información del modelo yang para crear la composición de configuración del dispositivo que corresponde. Después, según esto, se puede crear el XML para mandar la solicitud al dispositivo que corresponde para su configuración.