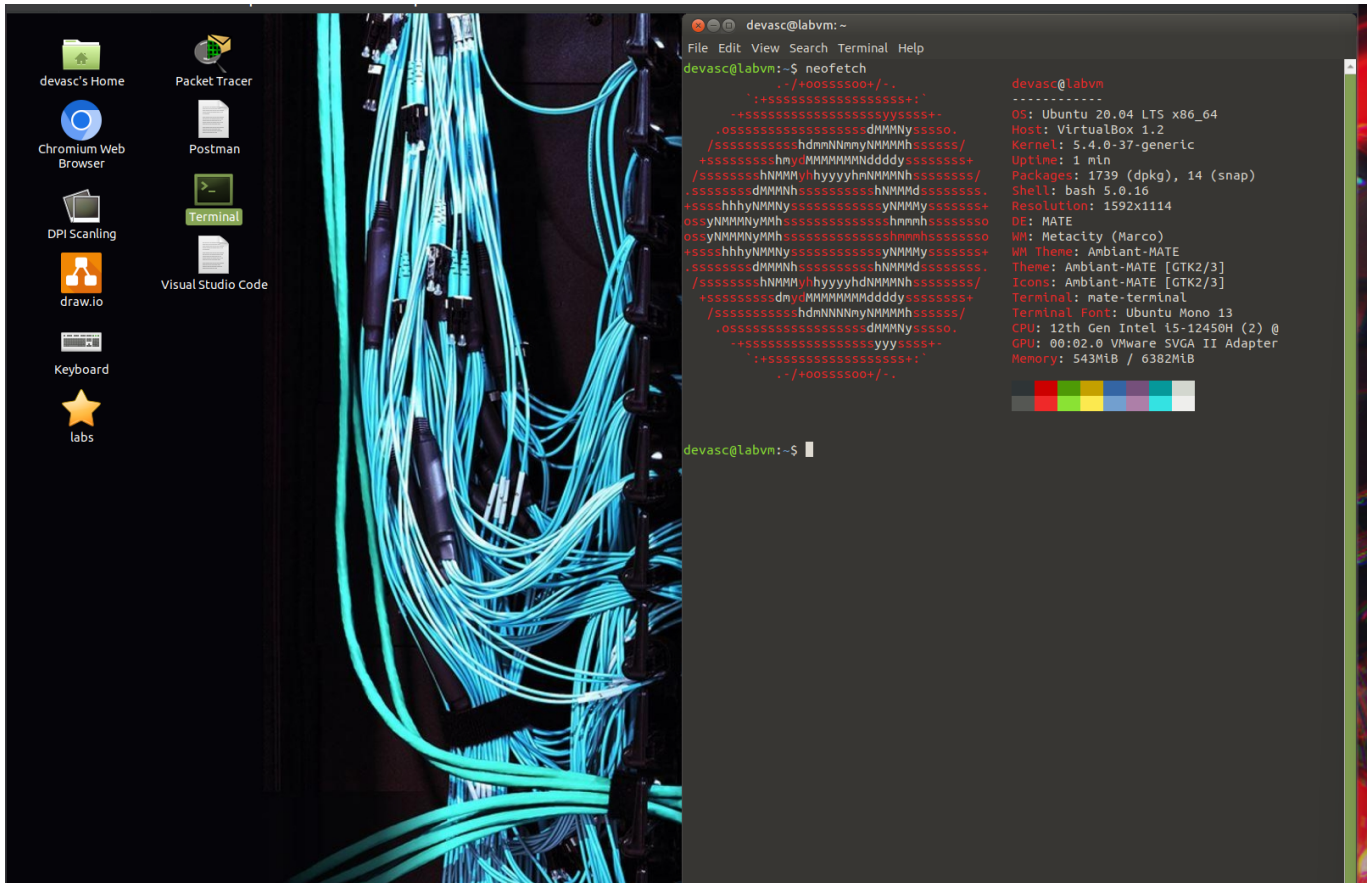


# Práctica Calificada N° 2

## Administración de Redes (CC 312)

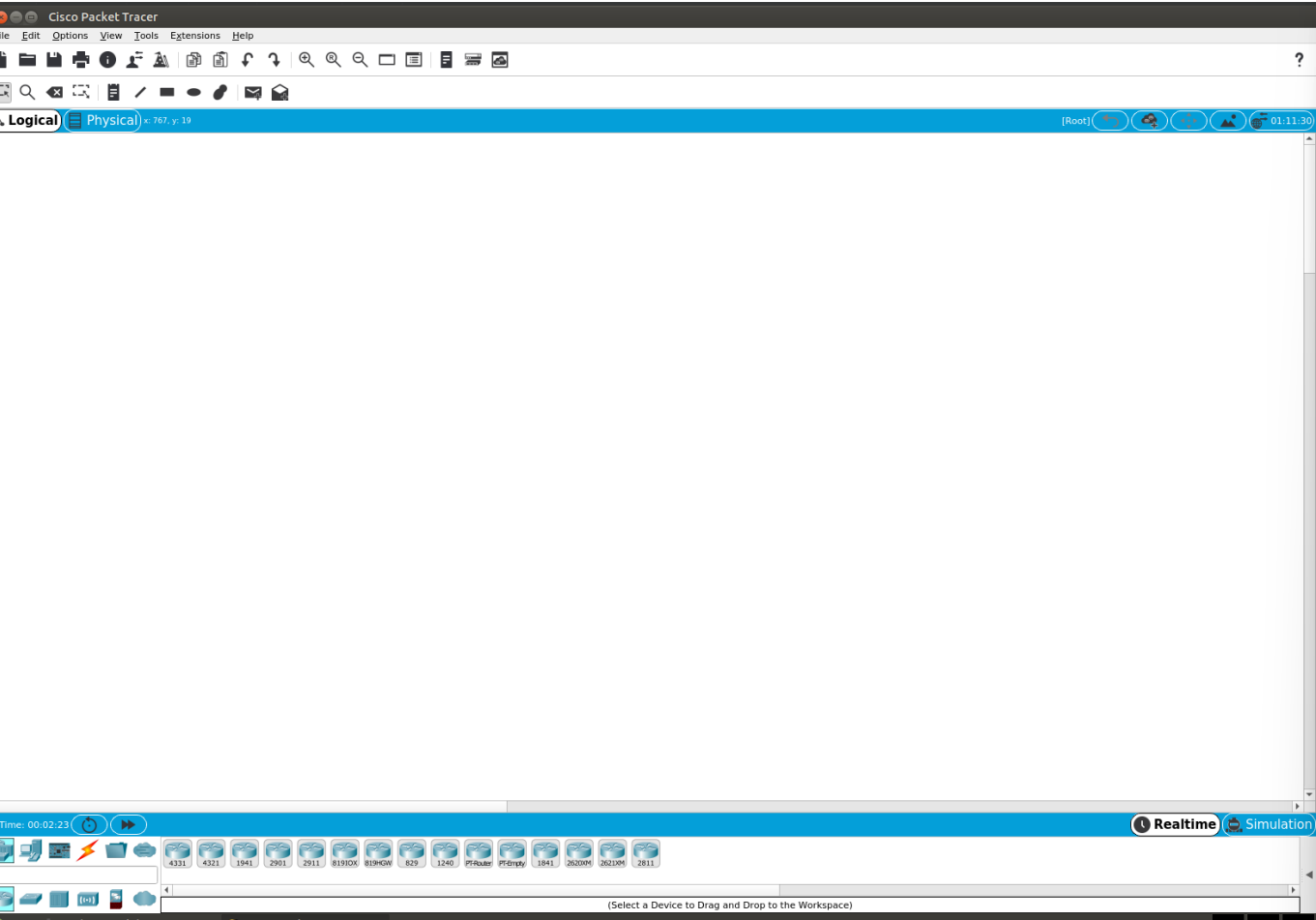
### Parte 1: Ejecutar la máquina virtual (Virtual Machine) de DEVASC.



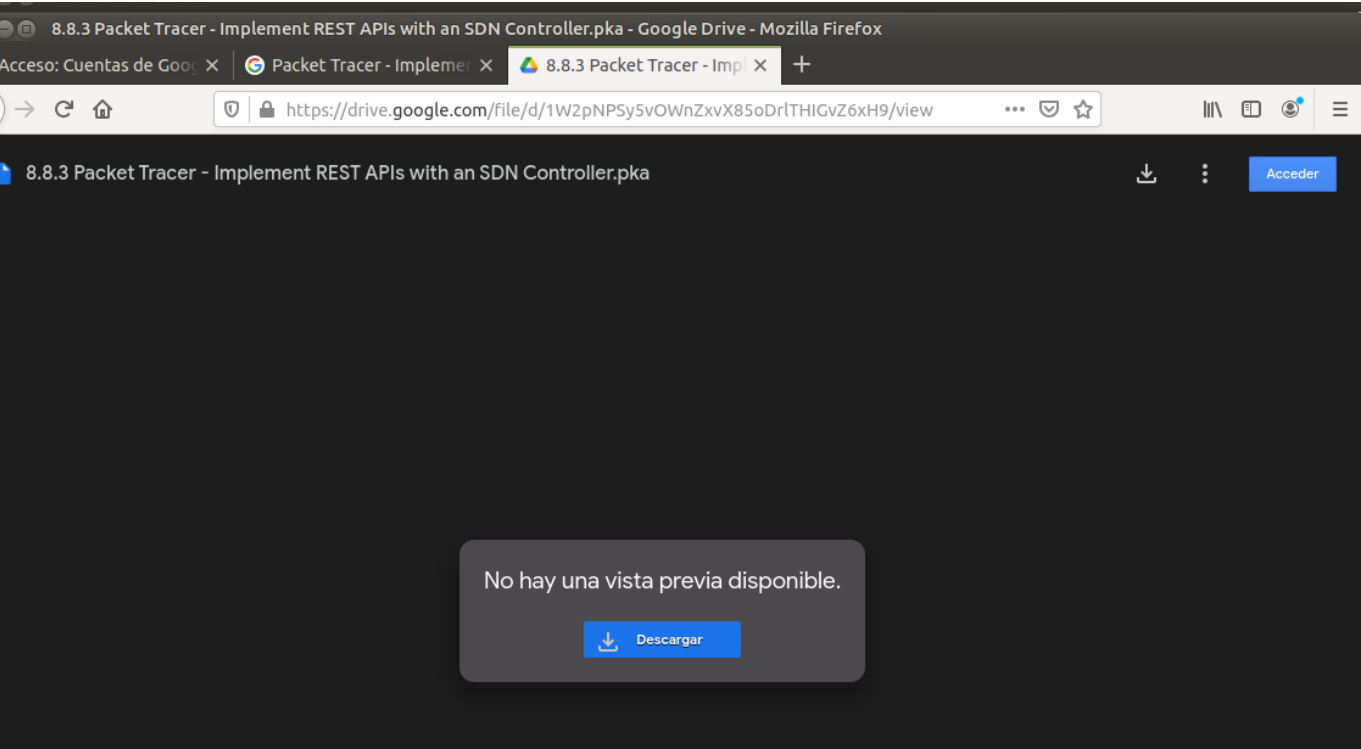
### Parte 2: Verificar la conectividad externa con el Packet Tracer

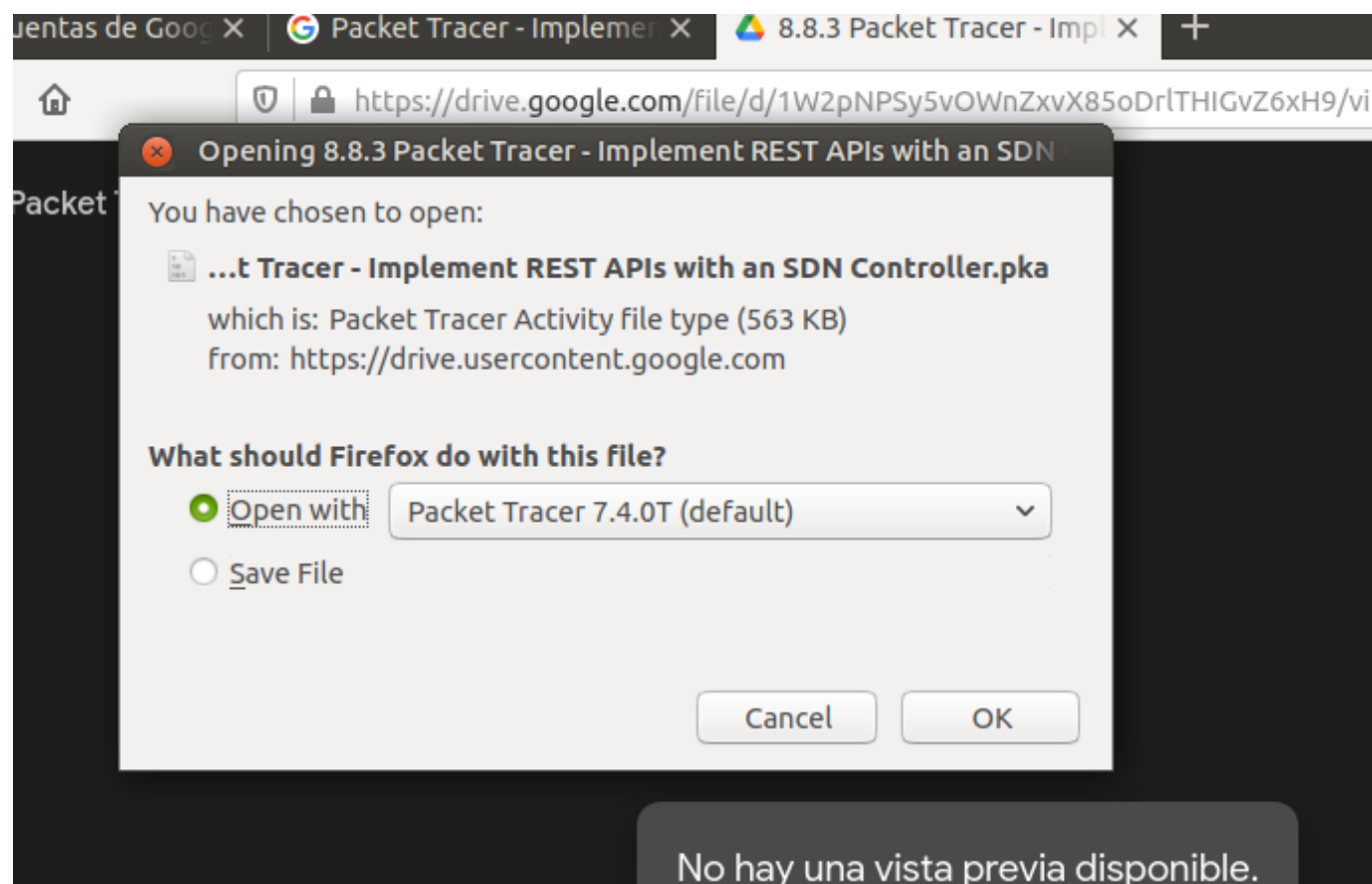
#### Paso 1: Abriendo Packet Tracer

Primero inicié sesión en Packet Tracer

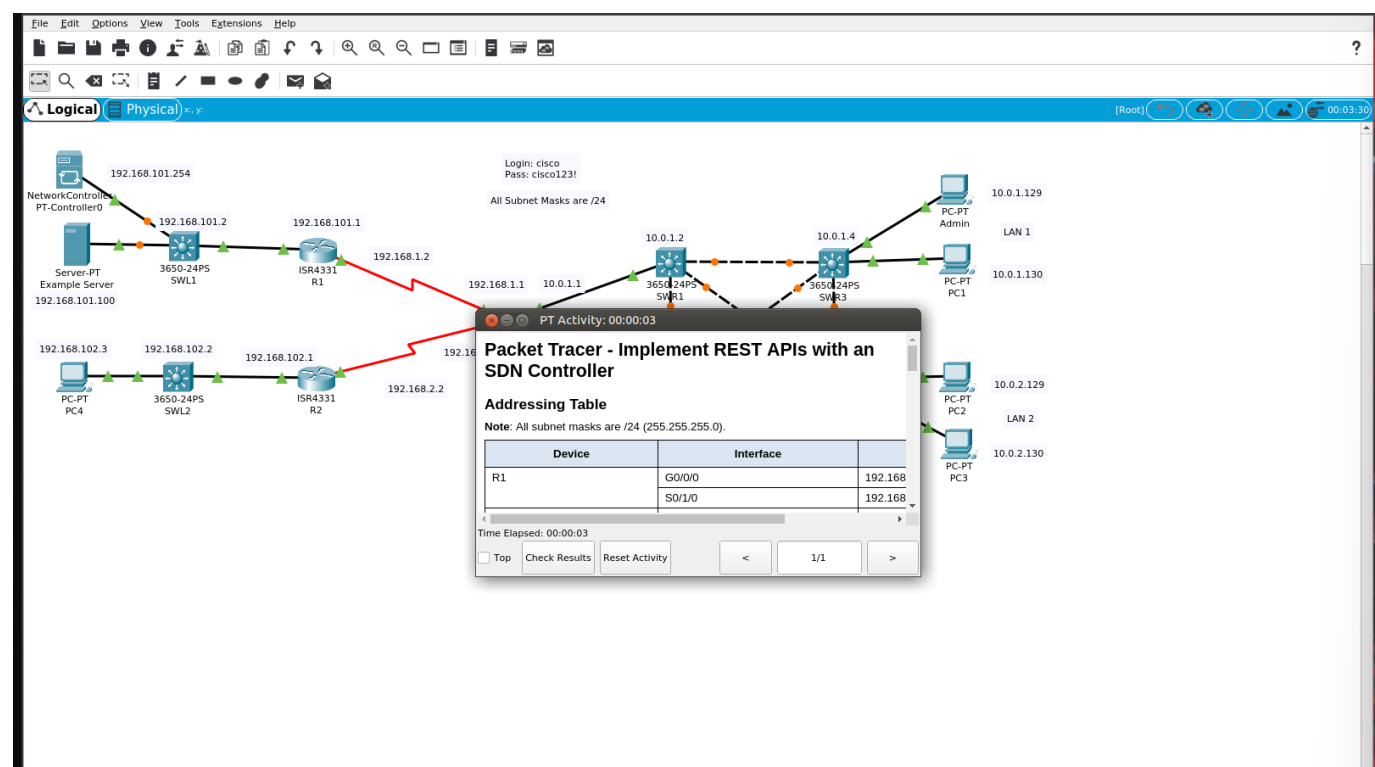


Luego descargué el archivo Packet Tracer - Implementar API REST con un SDN Controller.pka



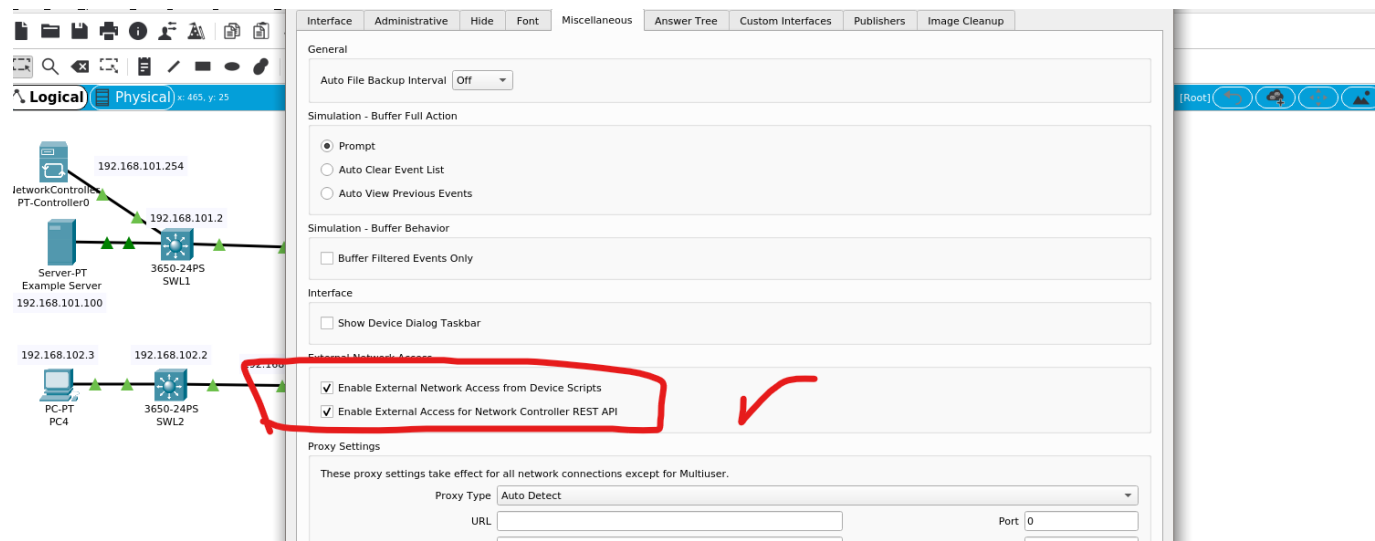


Entrando...

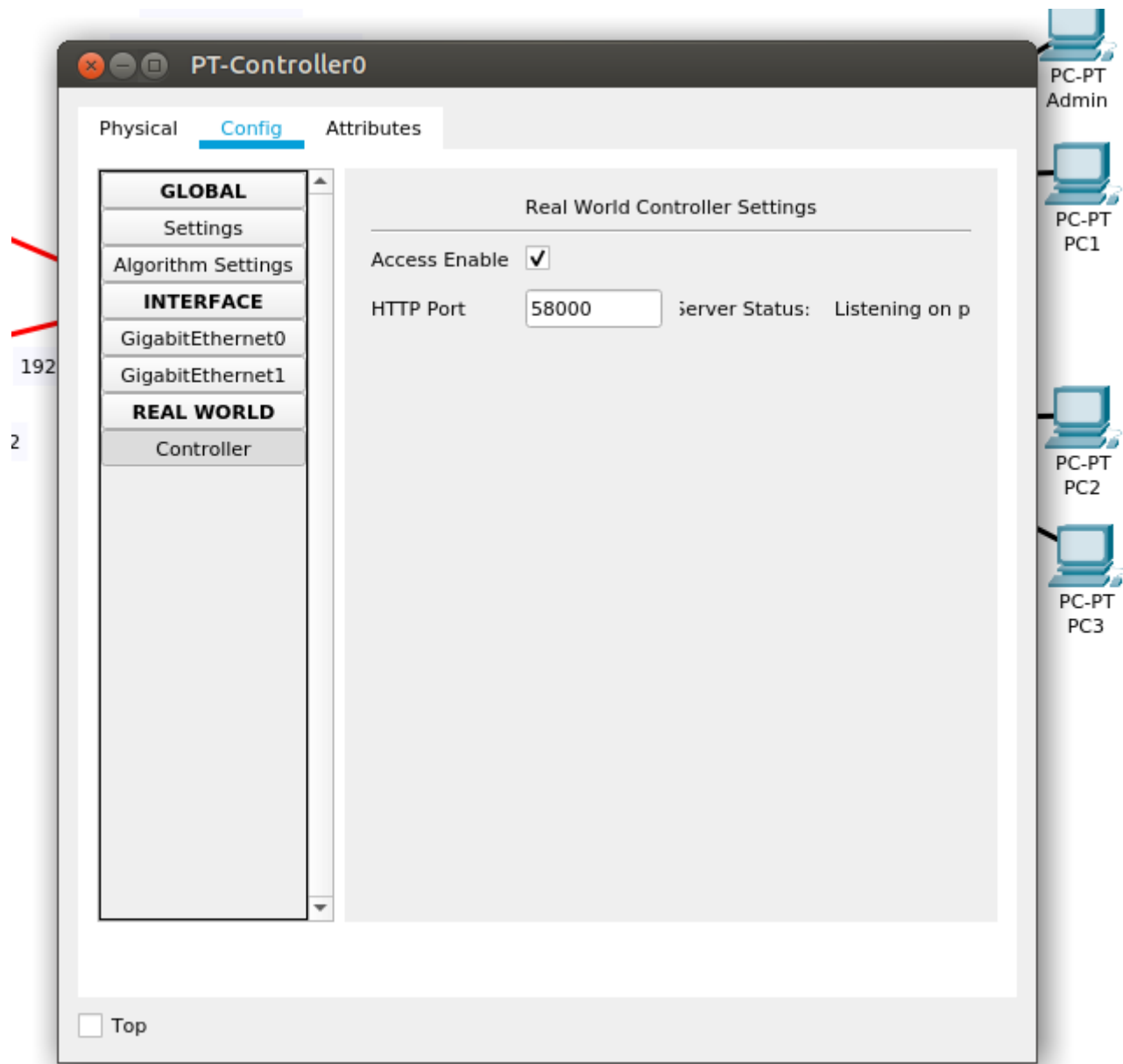


Paso 2: Verifique la configuración de Packet Tracer para el acceso externo.

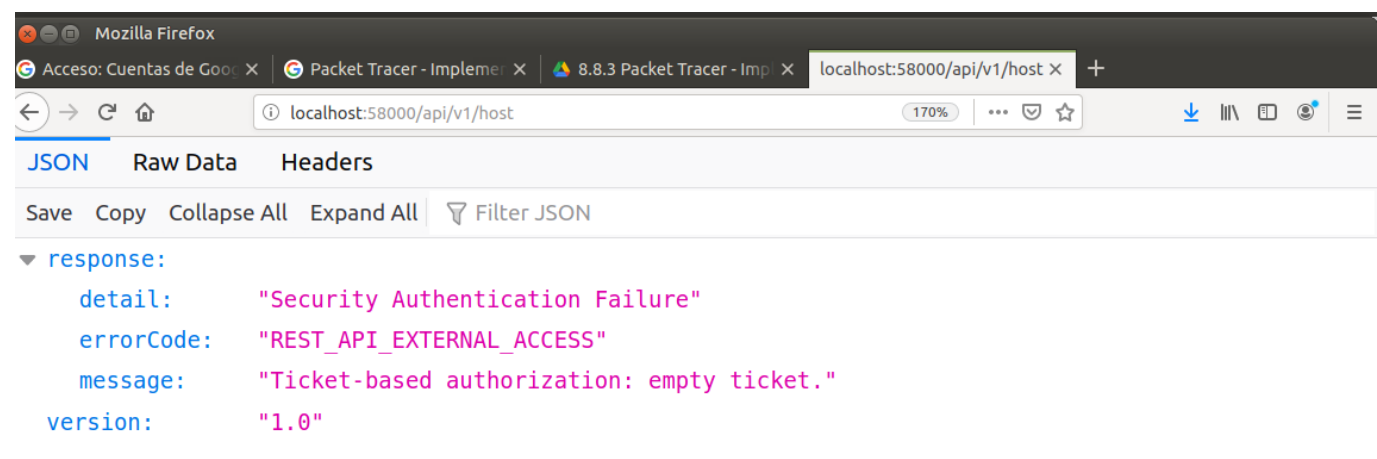
Viendo que esté habilitado el acceso externo con Rest API



Chequeando acceso para el controlador

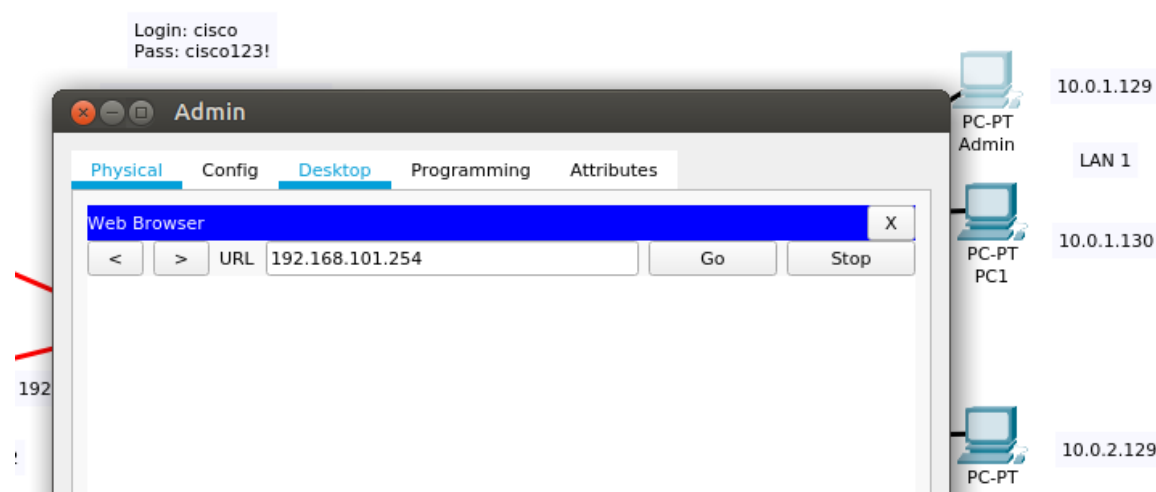


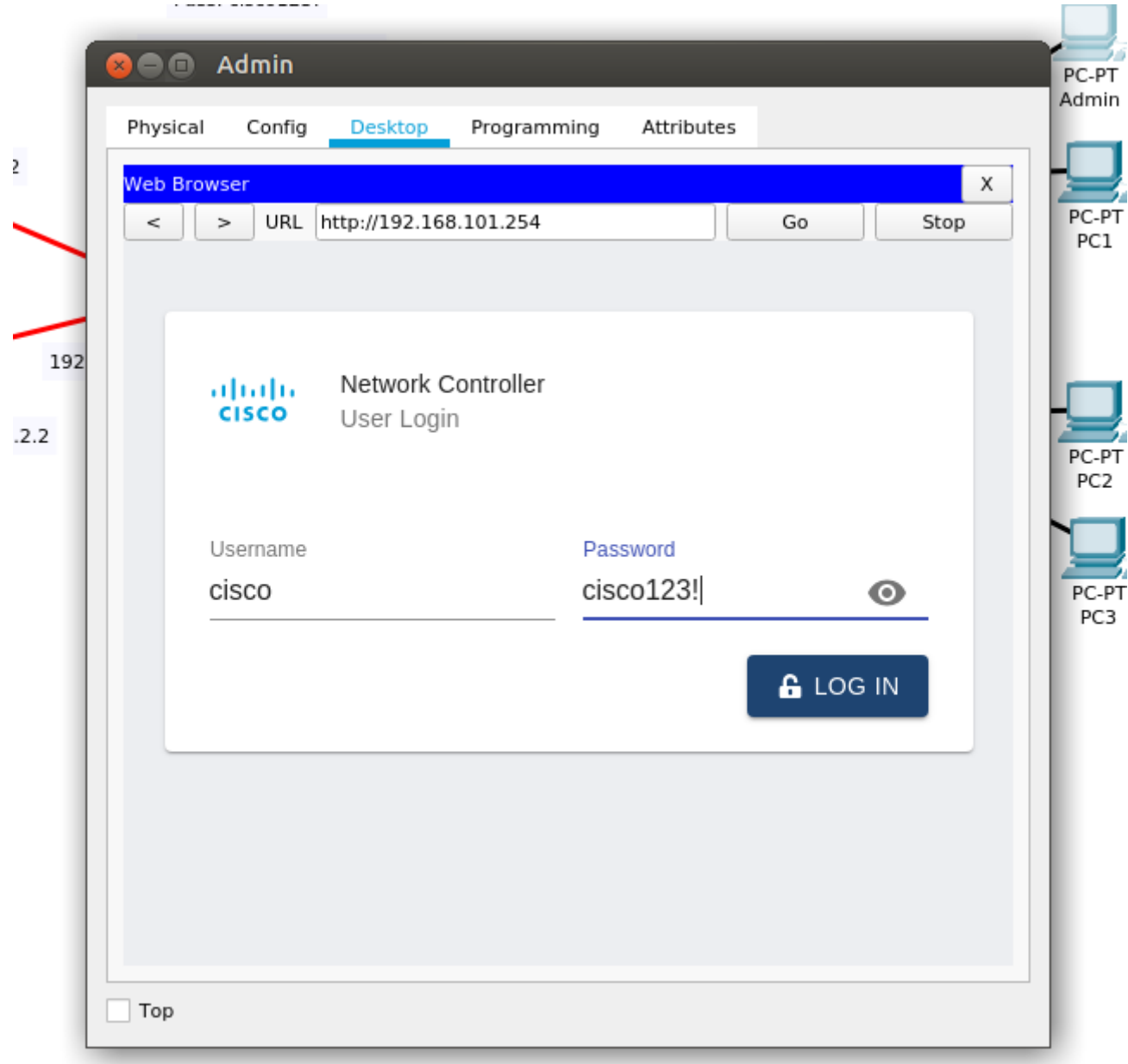
Paso 3: Compruebe que puede acceder a Packet Tracer desde otro programa en la VM DEVASC.

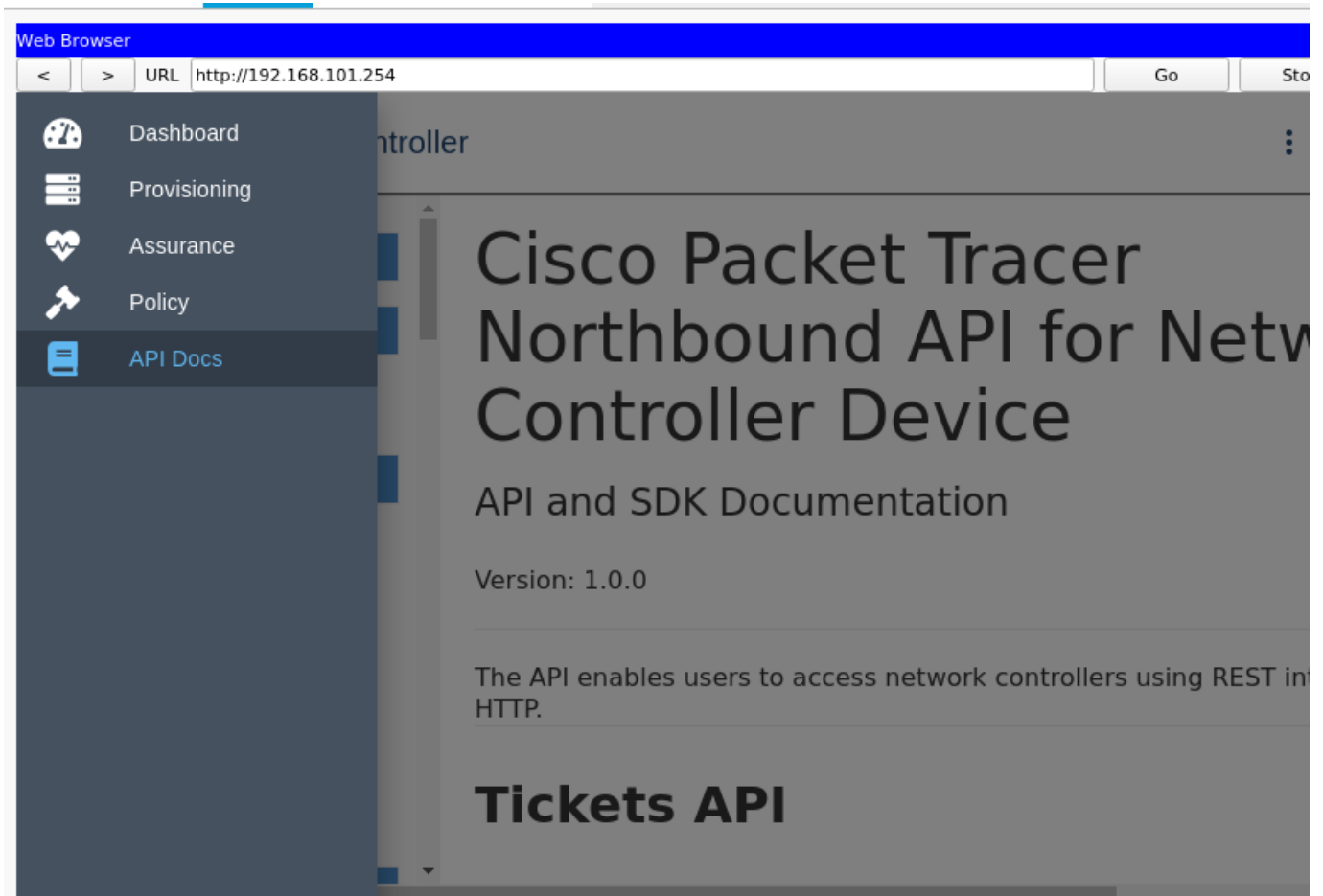


### Parte 3: Solicitar un token de autenticación con Postman

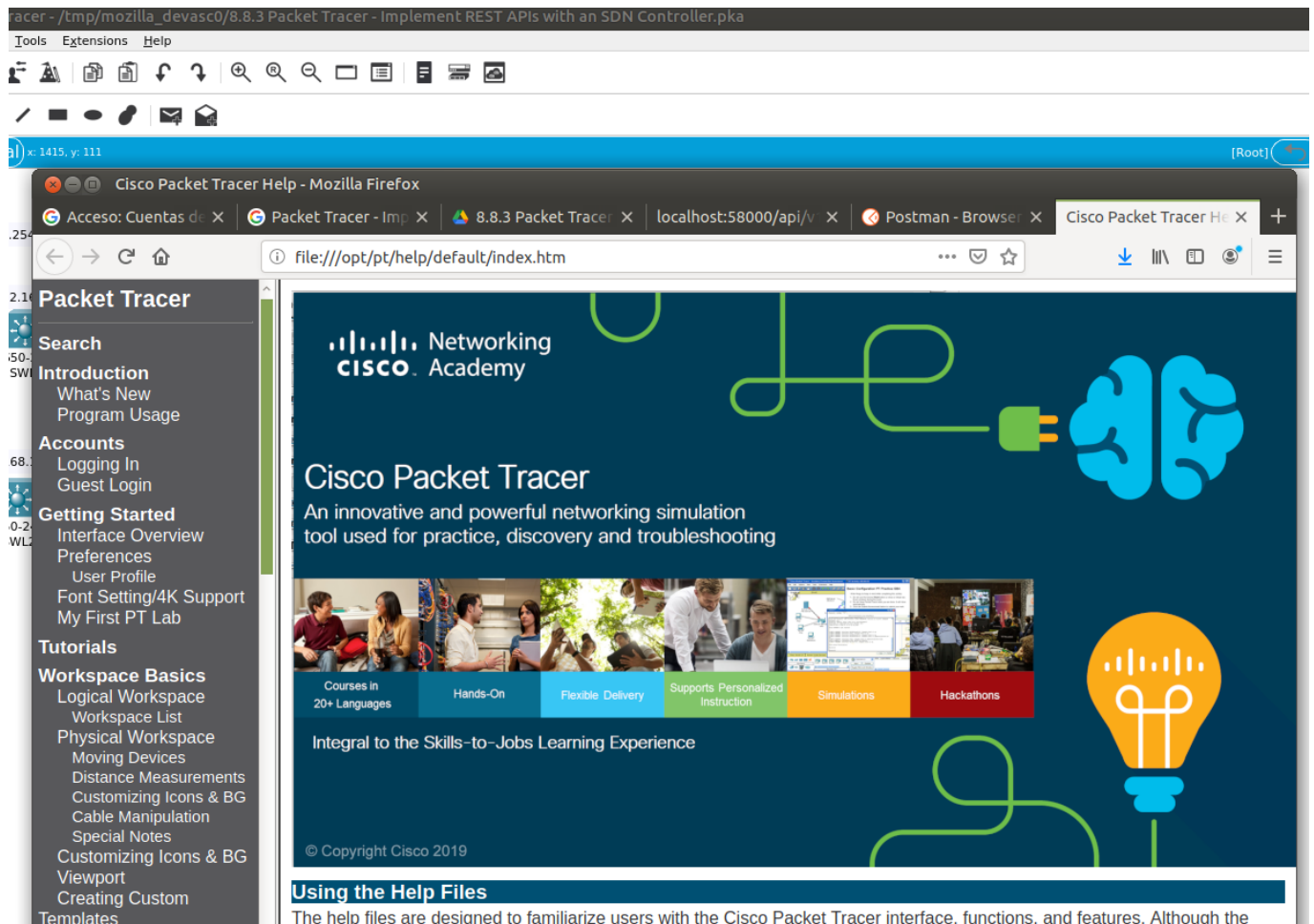
Entrando a la documentación de la API REST para el controlador



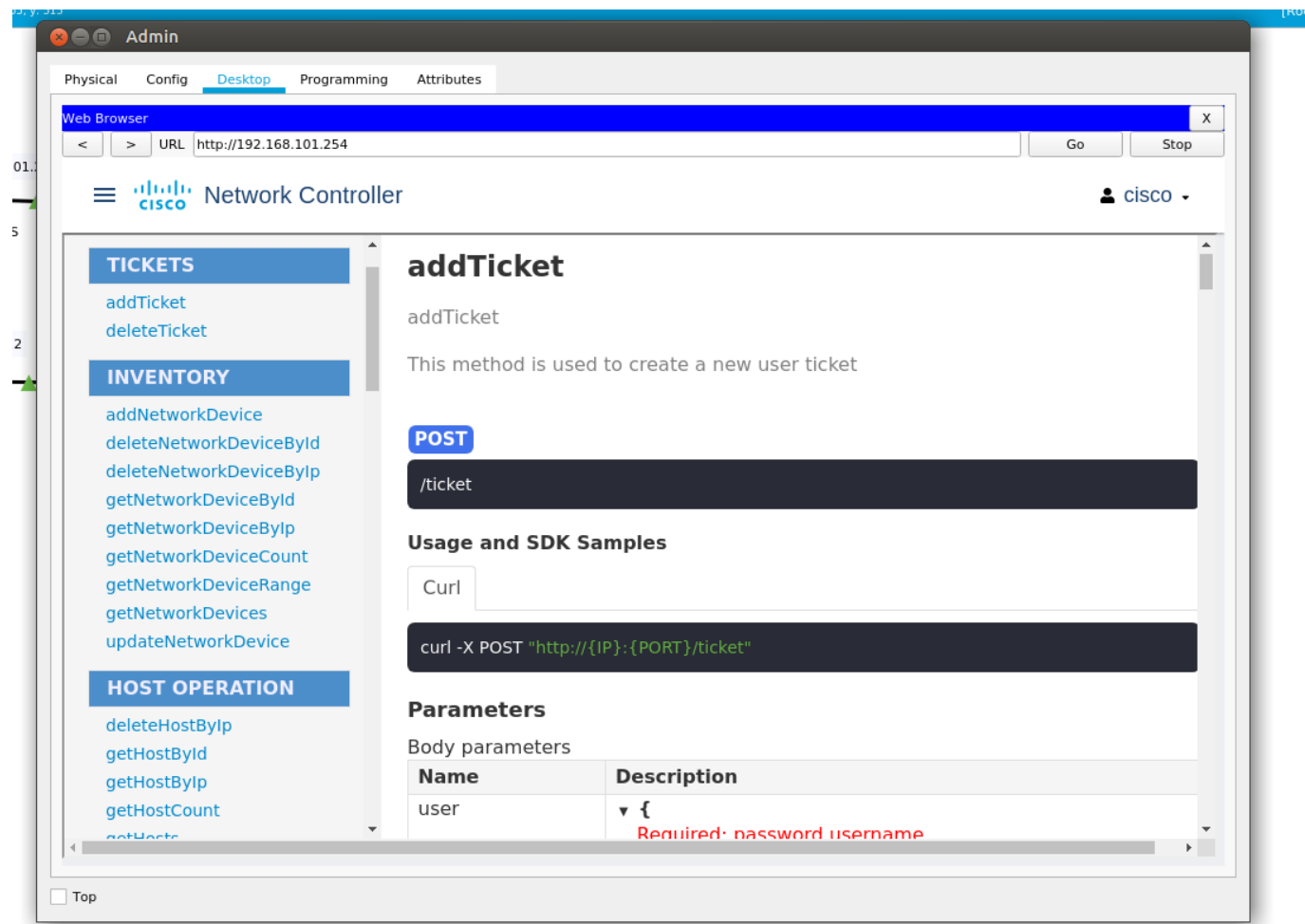




También desde Help -> Context



Adding a new request



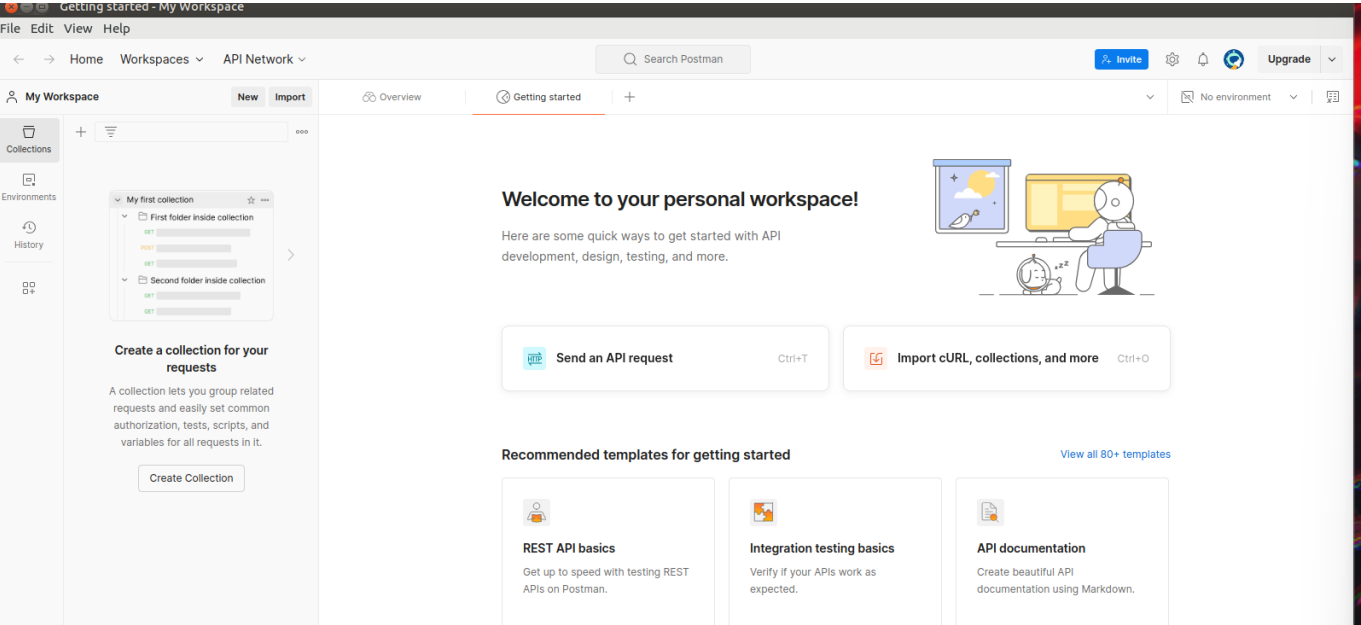
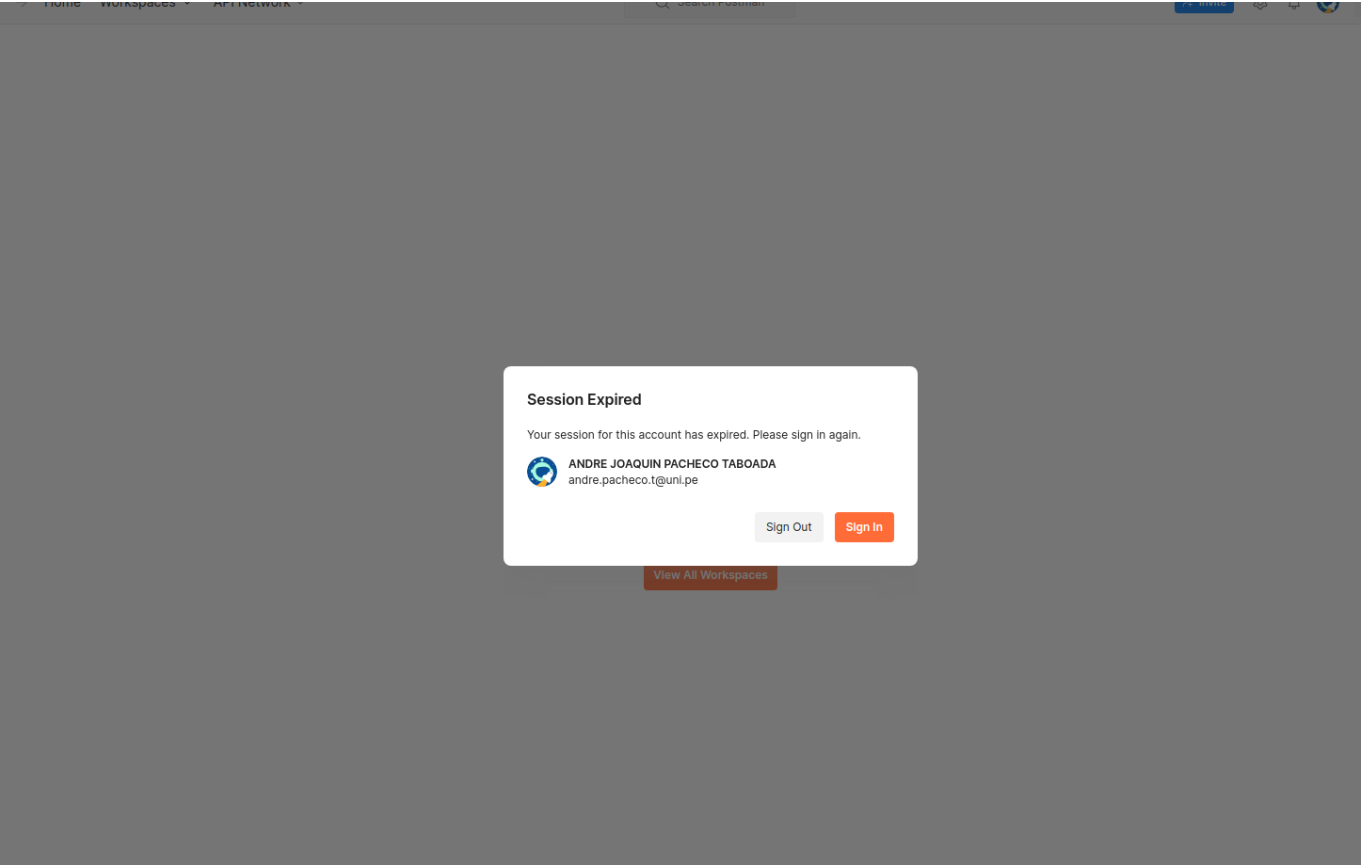
Usaremos el método POST

```
curl -X POST "http://{IP}:{PORT}/ticket"
```

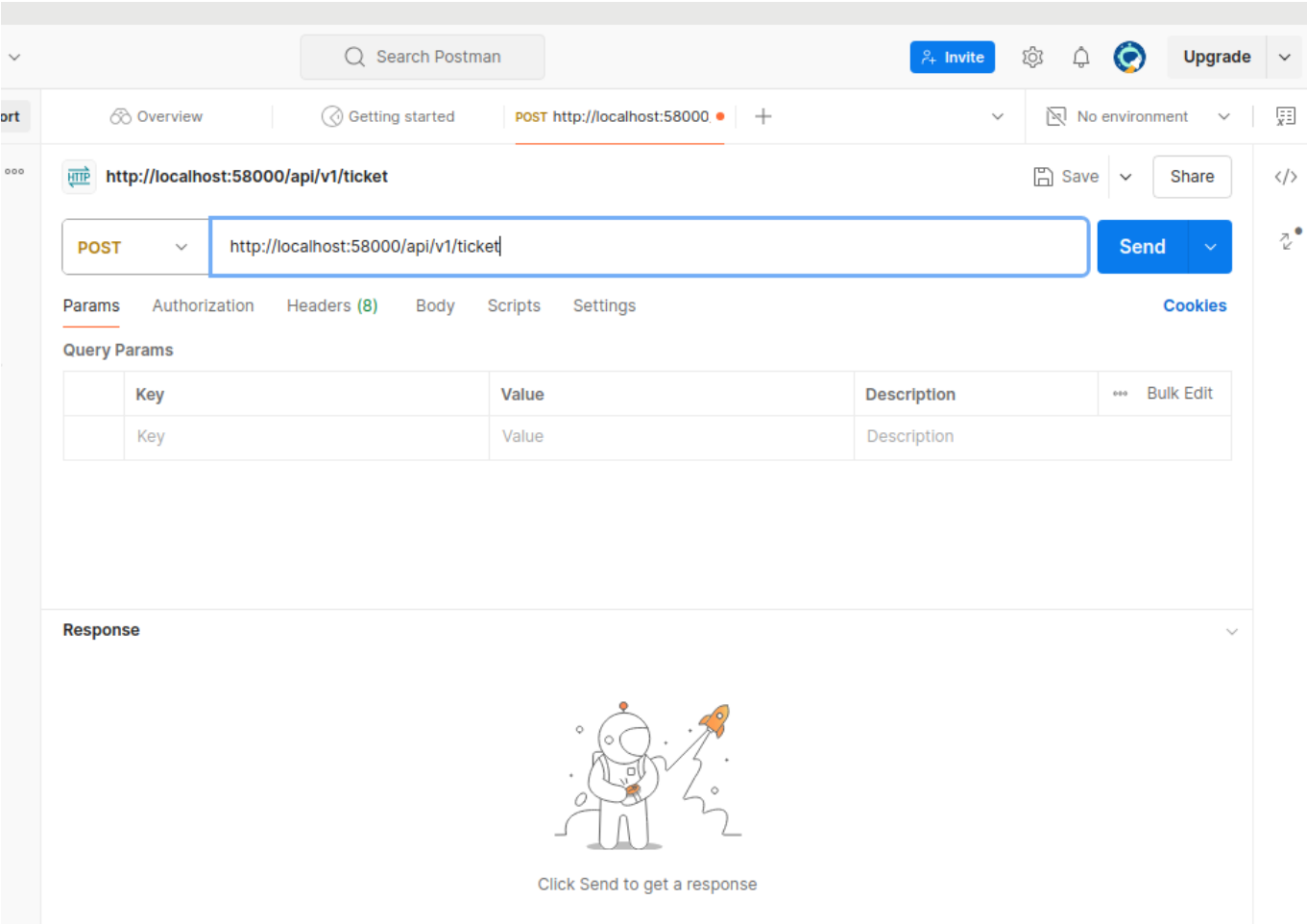
Entrando a Postman



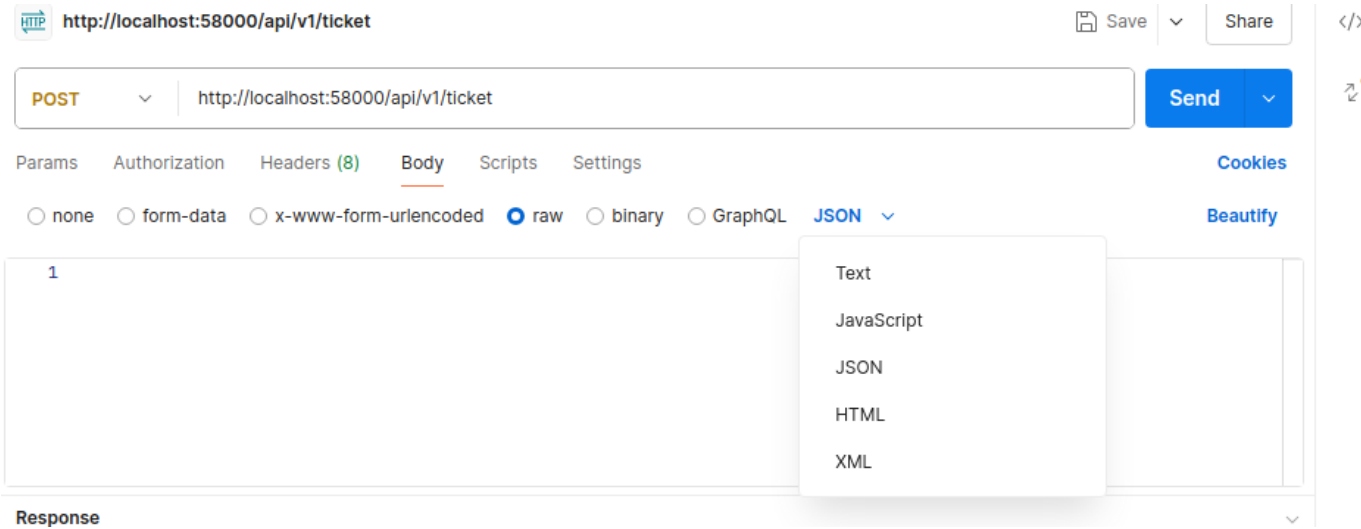
Iniciando sesión en Postman



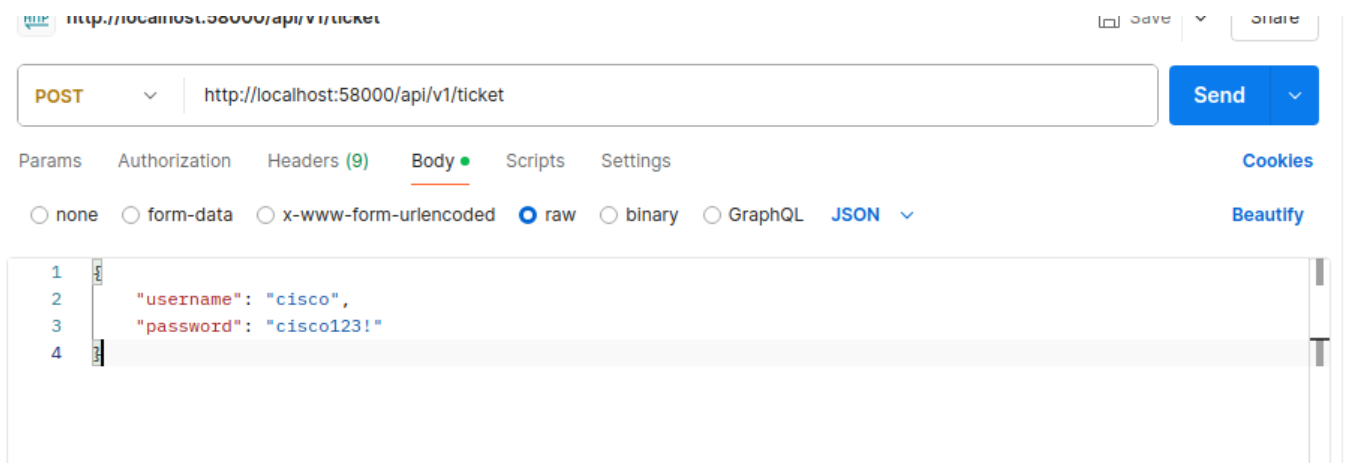
Seteando el request



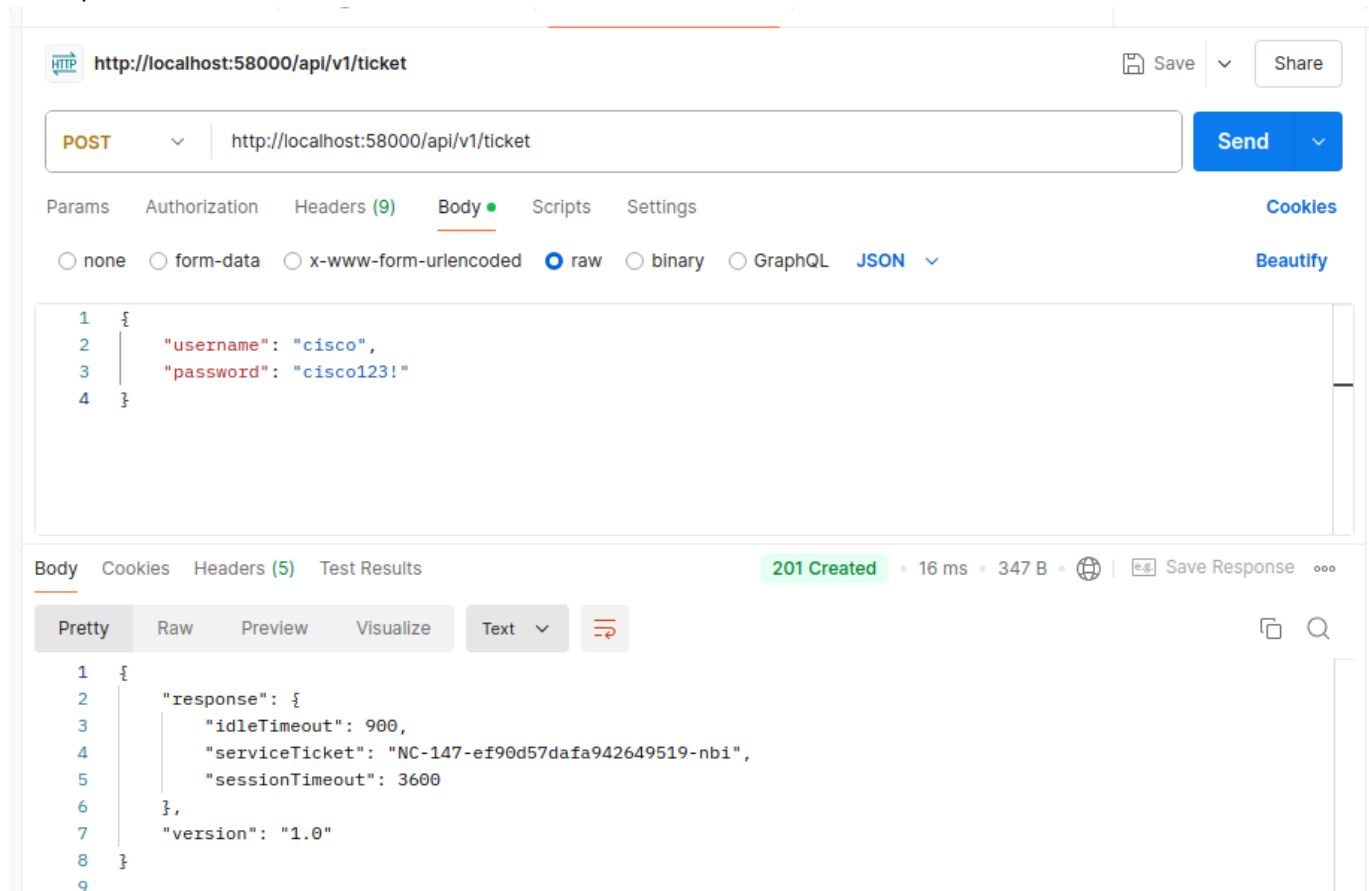
Pasando el body a raw -> JSON



Agregando el payload



La respuesta:

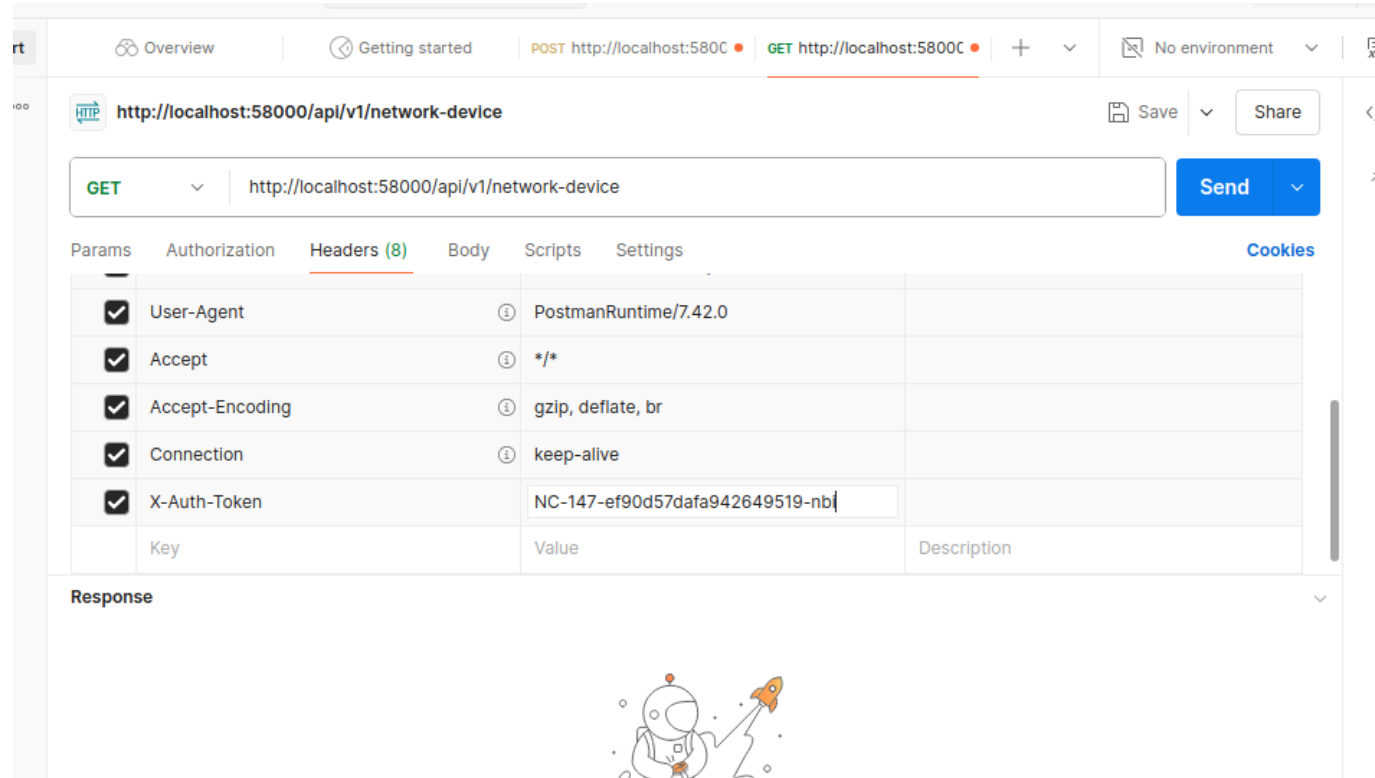


El ticket que me dió:

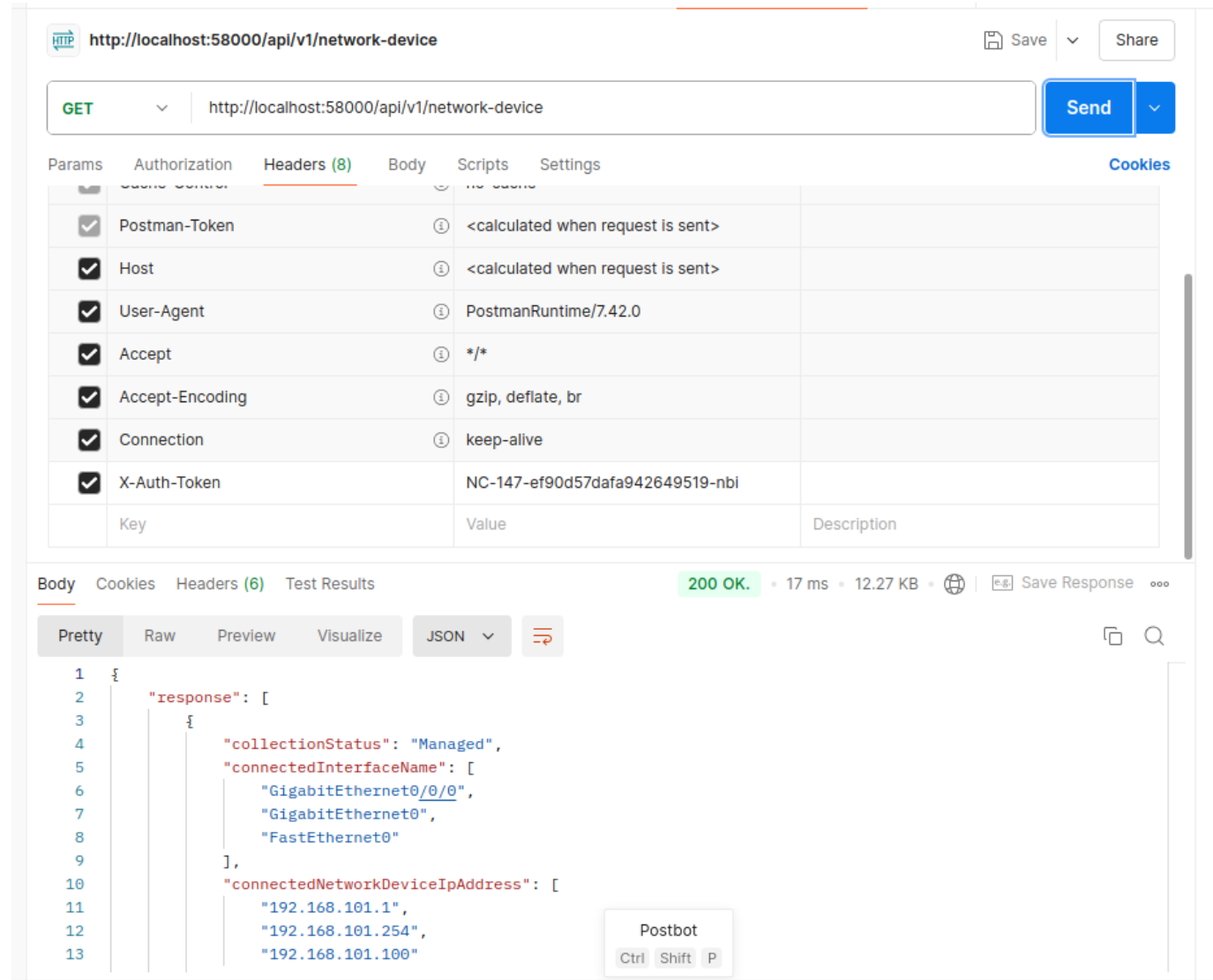
```
NC-147-ef90d57dafa942649519-nbi
```

## Parte 4: Enviar solicitudes REST con Postman

Creando nueva solicitud en Postman apuntando a `http://localhost:58000/api/v1/network-device` y agregando el ticket en el header como `X-Auth-Token` con el valor del ticket obtenido anteriormente.



Después de enviar la solicitud, se mostró el siguiente resultado:



Observo que se obtuvo el listado de dispositivos de red y toda la topología de la red, lo cual incluye cómo se conectan los dispositivos entre sí. Noté que se corresponden con los dispositivos que se ven en el Packet Tracer.

```
{
  "response": [
    {
      "collectionStatus": "Managed",
      "connectedInterfaceName": [
        "GigabitEthernet0/0/0",
        "GigabitEthernet0",
        "FastEthernet0"
      ],
      "connectedNetworkDeviceIpAddress": [
        "192.168.101.1",
        "192.168.101.254",
        "192.168.101.100"
      ],
      "connectedNetworkDeviceName": [
        "R1",
        "NetworkController",
        "Example Server"
      ],
      "errorDescription": "",
      "globalCredentialId": "53046ecc-88c3-49f6-9626-ca8ab9db6725",
      "hostname": "SWL1",
      "id": "CAT1010BT47-uuid",
      "interfaceCount": "29",
      "inventoryStatusDetail": "Managed",
      "lastUpdateTime": "11",
      "lastUpdated": "2020-06-11 18:19:42",
      "macAddress": "000C.CF42.2B11",
      "managementIpAddress": "192.168.101.2",
      "platformId": "3650",
      "productId": "3650-24PS",
      "reachabilityFailureReason": "",
      "reachabilityStatus": "Reachable",
      "serialNumber": "CAT1010BT47-",
      "softwareVersion": "16.3.2",
      "type": "MultiLayerSwitch",
      "upTime": "19 minutes, 2 seconds"
    },
    {
      "collectionStatus": "Managed",
      "connectedInterfaceName": [
        "GigabitEthernet1/0/1",
        "Serial0/1/0"
      ],
      "connectedNetworkDeviceIpAddress": [
        "192.168.101.2",
        "192.168.1.1"
      ],
    },
  ],
}
```

```
    "connectedNetworkDeviceName": [
      "SWL1",
      "R3"
    ],
    "errorDescription": "",
    "globalCredentialId": "53046ecc-88c3-49f6-9626-ca8ab9db6725",
    "hostname": "R1",
    "id": "FD01302XY2X-uuid",
    "interfaceCount": "6",
    "inventoryStatusDetail": "Managed",
    "ipAddresses": [
      "192.168.101.1",
      "192.168.1.2"
    ],
    "lastUpdateTime": "11",
    "lastUpdated": "2020-06-11 18:19:42",
    "macAddress": "00D0.5852.527D",
    "managementIpAddress": "192.168.1.2",
    "platformId": "ISR4300",
    "productId": "ISR4331",
    "reachabilityFailureReason": "",
    "reachabilityStatus": "Reachable",
    "serialNumber": "FD01302XY2X-",
    "softwareVersion": "15.4",
    "type": "Router",
    "upTime": "19 minutes, 2 seconds"
  },
  {
    "collectionStatus": "Managed",
    "connectedInterfaceName": [
      "GigabitEthernet1/0/1",
      "GigabitEthernet1/0/1",
      "Serial0/1/0",
      "Serial0/1/1"
    ],
    "connectedNetworkDeviceIpAddress": [
      "10.0.1.2",
      "",
      "192.168.1.2",
      "192.168.2.2"
    ],
    "connectedNetworkDeviceName": [
      "SWR1",
      "SWR2",
      "R1",
      "R2"
    ],
    "errorDescription": "",
    "globalCredentialId": "53046ecc-88c3-49f6-9626-ca8ab9db6725",
    "hostname": "R3",
    "id": "FD013026087-uuid",
    "interfaceCount": "6",
    "inventoryStatusDetail": "Managed",
    "ipAddresses": [
```

```
        "10.0.1.1",
        "10.0.2.1",
        "192.168.1.1",
        "192.168.2.1"
    ],
    "lastUpdateTime": "11",
    "lastUpdated": "2020-06-11 18:19:42",
    "macAddress": "00E0.B039.A39C",
    "managementIpAddress": "192.168.2.1",
    "platformId": "ISR4300",
    "productId": "ISR4331",
    "reachabilityFailureReason": "",
    "reachabilityStatus": "Reachable",
    "serialNumber": "FD013026087-",
    "softwareVersion": "15.4",
    "type": "Router",
    "upTime": "19 minutes, 2 seconds"
},
{
    "collectionStatus": "Managed",
    "connectedInterfaceName": [
        "GigabitEthernet0/0/0",
        "GigabitEthernet1/0/2",
        "GigabitEthernet1/0/3",
        "GigabitEthernet1/0/5"
    ],
    "connectedNetworkDeviceIpAddress": [
        "10.0.1.1",
        "10.0.1.4",
        "10.0.1.5",
        "10.0.1.3"
    ],
    "connectedNetworkDeviceName": [
        "R3",
        "SWR3",
        "SWR4",
        "SWR2"
    ],
    "errorDescription": "",
    "globalCredentialId": "53046ecc-88c3-49f6-9626-ca8ab9db6725",
    "hostname": "SWR1",
    "id": "CAT101021Z6-uuid",
    "interfaceCount": "29",
    "inventoryStatusDetail": "Managed",
    "lastUpdateTime": "11",
    "lastUpdated": "2020-06-11 18:19:42",
    "macAddress": "00E0.F915.E250",
    "managementIpAddress": "10.0.1.2",
    "platformId": "3650",
    "productId": "3650-24PS",
    "reachabilityFailureReason": "",
    "reachabilityStatus": "Reachable",
    "serialNumber": "CAT101021Z6-",
    "softwareVersion": "16.3.2",
```

```
"type": "MultiLayerSwitch",
"upTime": "19 minutes, 2 seconds"
},
{
  "collectionStatus": "Managed",
  "connectedInterfaceName": [
    "GigabitEthernet0/0/1",
    "GigabitEthernet1/0/2",
    "GigabitEthernet1/0/4",
    "GigabitEthernet1/0/5"
  ],
  "connectedNetworkDeviceIpAddress": [
    "10.0.2.1",
    "10.0.1.5",
    "10.0.1.4",
    "10.0.1.2"
  ],
  "connectedNetworkDeviceName": [
    "R3",
    "SWR4",
    "SWR3",
    "SWR1"
  ],
  "errorDescription": "",
  "globalCredentialId": "53046ecc-88c3-49f6-9626-ca8ab9db6725",
  "hostname": "SWR2",
  "id": "CAT1010JJ1H-uuid",
  "interfaceCount": "29",
  "inventoryStatusDetail": "Managed",
  "lastUpdateTime": "11",
  "lastUpdated": "2020-06-11 18:19:42",
  "macAddress": "00E0.B060.5317",
  "managementIpAddress": "10.0.1.3",
  "platformId": "3650",
  "productId": "3650-24PS",
  "reachabilityFailureReason": "",
  "reachabilityStatus": "Reachable",
  "serialNumber": "CAT1010JJ1H-",
  "softwareVersion": "16.3.2",
  "type": "MultiLayerSwitch",
  "upTime": "19 minutes, 2 seconds"
},
{
  "collectionStatus": "Managed",
  "connectedInterfaceName": [
    "GigabitEthernet1/0/1",
    "Serial0/1/1"
  ],
  "connectedNetworkDeviceIpAddress": [
    "192.168.102.2",
    "192.168.2.1"
  ],
  "connectedNetworkDeviceName": [
    "SWL2",
```



```

        "R3"
    ],
    "errorDescription": "",
    "globalCredentialId": "53046ecc-88c3-49f6-9626-ca8ab9db6725",
    "hostname": "R2",
    "id": "FD013022UJ0-uuid",
    "interfaceCount": "6",
    "inventoryStatusDetail": "Managed",
    "ipAddresses": [
        "192.168.102.1",
        "192.168.2.2"
    ],
    "lastUpdateTime": "11",
    "lastUpdated": "2020-06-11 18:19:42",
    "macAddress": "0060.4797.3DA5",
    "managementIpAddress": "192.168.2.2",
    "platformId": "ISR4300",
    "productId": "ISR4331",
    "reachabilityFailureReason": "",
    "reachabilityStatus": "Reachable",
    "serialNumber": "FD013022UJ0-",
    "softwareVersion": "15.4",
    "type": "Router",
    "upTime": "19 minutes, 2 seconds"
},
{
    "collectionStatus": "Managed",
    "connectedInterfaceName": [
        "GigabitEthernet0/0/0",
        "FastEthernet0"
    ],
    "connectedNetworkDeviceIpAddress": [
        "192.168.102.1",
        "192.168.102.3"
    ],
    "connectedNetworkDeviceName": [
        "R2",
        "PC4"
    ],
    "errorDescription": "",
    "globalCredentialId": "53046ecc-88c3-49f6-9626-ca8ab9db6725",
    "hostname": "SWL2",
    "id": "CAT101059L6-uuid",
    "interfaceCount": "29",
    "inventoryStatusDetail": "Managed",
    "lastUpdateTime": "11",
    "lastUpdated": "2020-06-11 18:19:42",
    "macAddress": "0090.2155.BB91",
    "managementIpAddress": "192.168.102.2",
    "platformId": "3650",
    "productId": "3650-24PS",
    "reachabilityFailureReason": "",
    "reachabilityStatus": "Reachable",
    "serialNumber": "CAT101059L6-",

```

```
    "softwareVersion": "16.3.2",
    "type": "MultiLayerSwitch",
    "upTime": "19 minutes, 2 seconds"
  },
  {
    "collectionStatus": "Managed",
    "connectedInterfaceName": [
      "GigabitEthernet1/0/2",
      "GigabitEthernet1/0/3",
      "GigabitEthernet1/0/5",
      "FastEthernet0",
      "FastEthernet0"
    ],
    "connectedNetworkDeviceIpAddress": [
      "10.0.1.3",
      "10.0.1.2",
      "10.0.1.4",
      "10.0.2.129",
      "10.0.2.130"
    ],
    "connectedNetworkDeviceName": [
      "SWR2",
      "SWR1",
      "SWR3",
      "PC2",
      "PC3"
    ],
    "errorDescription": "",
    "globalCredentialId": "53046ecc-88c3-49f6-9626-ca8ab9db6725",
    "hostname": "SWR4",
    "id": "CAT1010K0UR-uuid",
    "interfaceCount": "29",
    "inventoryStatusDetail": "Managed",
    "lastUpdateTime": "11",
    "lastUpdated": "2020-06-11 18:19:42",
    "macAddress": "0060.5C0D.E4AE",
    "managementIpAddress": "10.0.1.5",
    "platformId": "3650",
    "productId": "3650-24PS",
    "reachabilityFailureReason": "",
    "reachabilityStatus": "Reachable",
    "serialNumber": "CAT1010K0UR-",
    "softwareVersion": "16.3.2",
    "type": "MultiLayerSwitch",
    "upTime": "19 minutes, 2 seconds"
  },
  {
    "collectionStatus": "Managed",
    "connectedInterfaceName": [
      "GigabitEthernet1/0/2",
      "GigabitEthernet1/0/4",
      "GigabitEthernet1/0/5",
      "FastEthernet0",
      "FastEthernet0"
```

```
    ],
    "connectedNetworkDeviceIpAddress": [
      "10.0.1.2",
      "10.0.1.3",
      "10.0.1.5",
      "10.0.1.130",
      "10.0.1.129"
    ],
    "connectedNetworkDeviceName": [
      "SWR1",
      "SWR2",
      "SWR4",
      "Admin",
      "PC1"
    ],
    "errorDescription": "",
    "globalCredentialId": "53046ecc-88c3-49f6-9626-ca8ab9db6725",
    "hostname": "SWR3",
    "id": "CAT1010J4FO-uuid",
    "interfaceCount": "29",
    "inventoryStatusDetail": "Managed",
    "lastUpdateTime": "11",
    "lastUpdated": "2020-06-11 18:19:42",
    "macAddress": "0050.0F7C.0C09",
    "managementIpAddress": "10.0.1.4",
    "platformId": "3650",
    "productId": "3650-24PS",
    "reachabilityFailureReason": "",
    "reachabilityStatus": "Reachable",
    "serialNumber": "CAT1010J4FO-",
    "softwareVersion": "16.3.2",
    "type": "MultiLayerSwitch",
    "upTime": "19 minutes, 2 seconds"
  }
],
"version": "1.0"
}
```

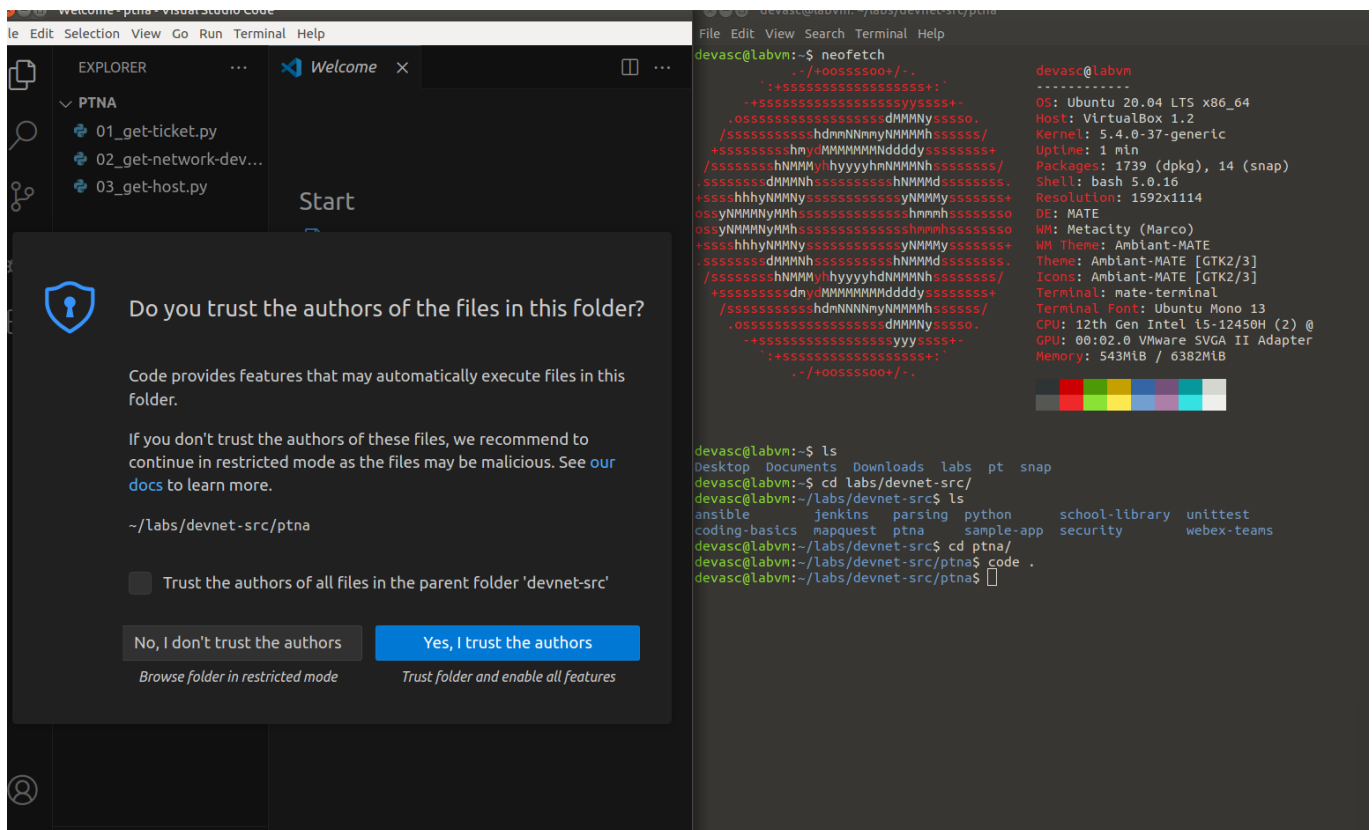
Noto los siguientes dispositivos:

- "hostname": "R1"
- "hostname": "R2"
- "hostname": "R3"
- "hostname": "SWR1"
- "hostname": "SWR2"
- "hostname": "SWR3"
- "hostname": "SWR4"

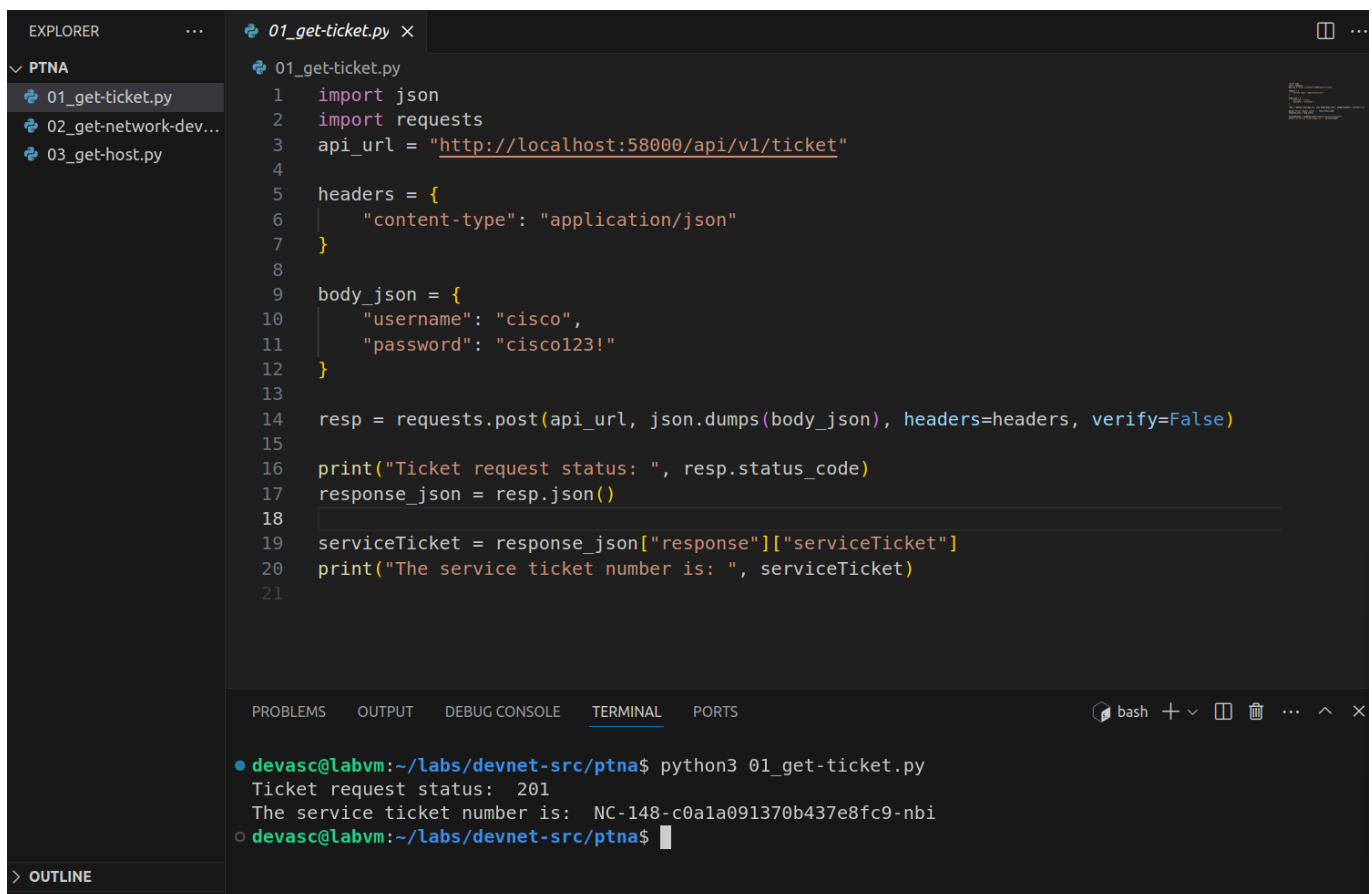
Con ello cerramos el Postman.

## Parte 5: Enviar solicitudes REST con código VS

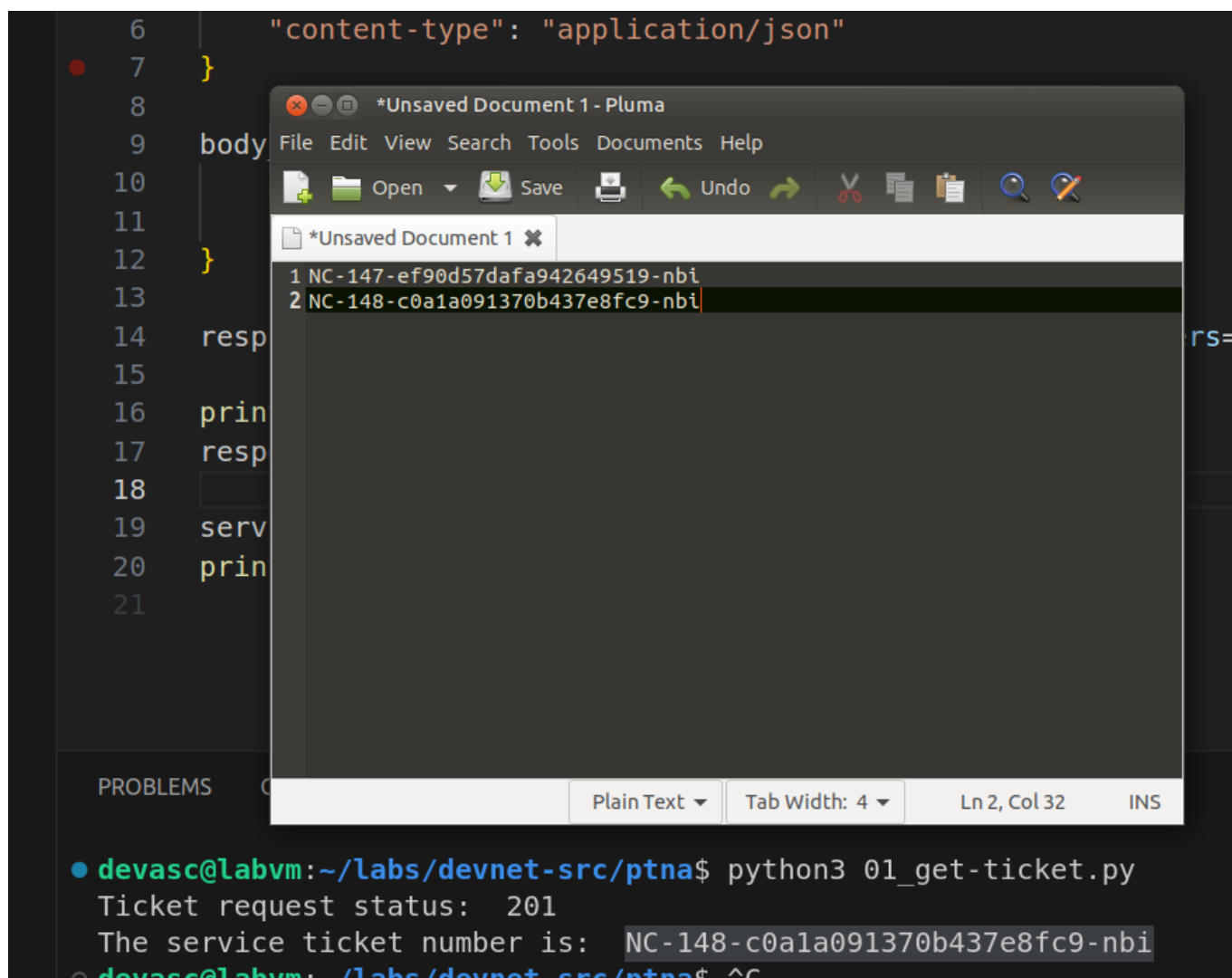
Entrando a la ruta devnet-src/ptna y entrando a VSC



Ejecutando el primer script .py



Observación: El ticket que se obtiene es DIFERENTE al que se obtuvo en Postman.



The screenshot shows a code editor with a Python script on the left and a terminal window at the bottom. A modal window titled '\*Unsaved Document 1 - Pluma' is open in the center, displaying a list of ticket numbers. The script on the left is partially visible, showing a JSON object with a 'content-type' field and a 'body' field. The terminal window shows the output of a Python script, indicating a successful ticket request with status 201 and a service ticket number.

```
6     "content-type": "application/json"
7 }
8
9 body
10
11
12 }
13
14 resp
15
16 prin
17 resp
18
19 serv
20 prin
21
```

\*Unsaved Document 1 - Pluma

File Edit View Search Tools Documents Help

Open Save Undo

\*Unsaved Document 1

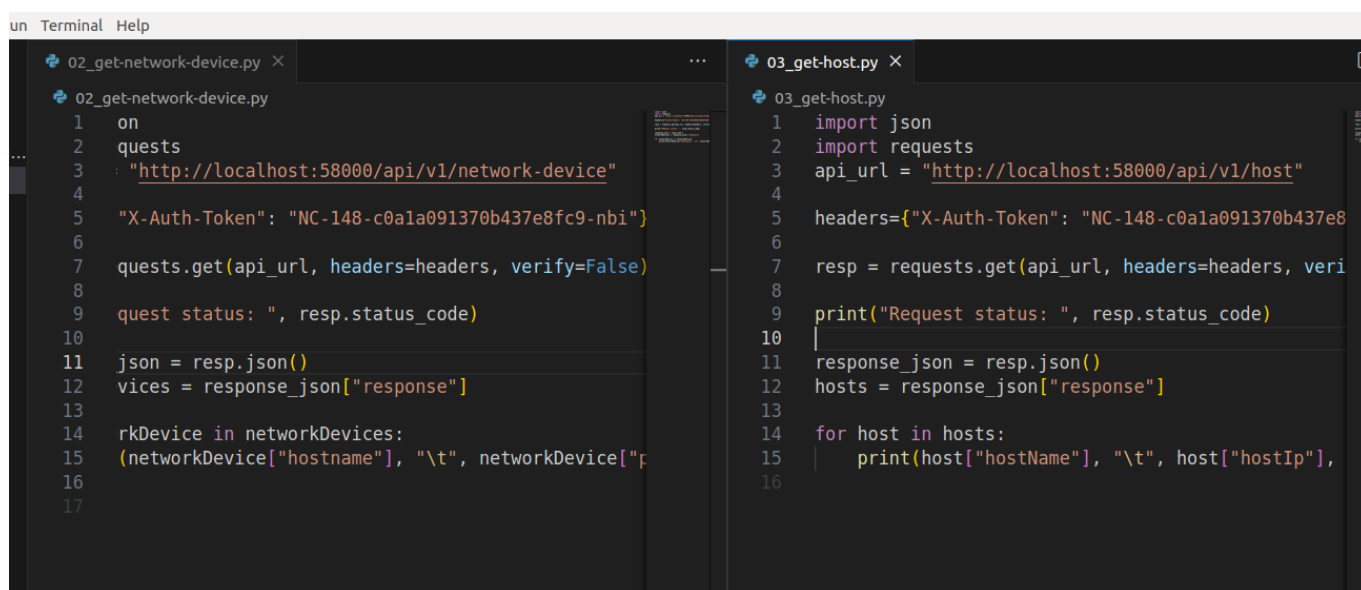
```
1 NC-147-ef90d57dafa942649519-nbi
2 NC-148-c0a1a091370b437e8fc9-nbi
```

PROBLEMS

Plain Text Tab Width: 4 Ln 2, Col 32 INS

```
● devasc@labvm:~/labs/devnet-src/ptna$ python3 01_get-ticket.py
Ticket request status: 201
The service ticket number is: NC-148-c0a1a091370b437e8fc9-nbi
devasc@labvm:~/labs/devnet-src/ptna$ ^C
```

Reemplazando el ticket en los otros 2 scripts .py y ejecutándolos:



The screenshot shows a code editor with two Python scripts side-by-side. The left script, '02\_get-network-device.py', is a REST client that sends a GET request to a network device API. The right script, '03\_get-host.py', is a REST client that sends a GET request to a host API. Both scripts use the 'requests' library and include the same 'X-Auth-Token' header.

```
02_get-network-device.py
1 on
2 quests
3 : "http://localhost:58000/api/v1/network-device"
4
5 "X-Auth-Token": "NC-148-c0a1a091370b437e8fc9-nbi"}
6
7 quests.get(api_url, headers=headers, verify=False)
8
9 quest status: ", resp.status_code)
10
11 json = resp.json()
12 vices = response_json["response"]
13
14 rkDevice in networkDevices:
15 (networkDevice["hostname"], "\t", networkDevice["p
16
17
```

```
03_get-host.py
1 import json
2 import requests
3 api_url = "http://localhost:58000/api/v1/host"
4
5 headers={"X-Auth-Token": "NC-148-c0a1a091370b437e8
6
7 resp = requests.get(api_url, headers=headers, veri
8
9 print("Request status: ", resp.status_code)
10
11 response_json = resp.json()
12 hosts = response_json["response"]
13
14 for host in hosts:
15     print(host["hostName"], "\t", host["hostIp"],
16
```

Ejecutando el segundo script .py

```
Problems (Ctrl+Shift+M)

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

● devasc@labvm:~/labs/devnet-src/ptna$ python3 01_get-ticket.py
Ticket request status: 201
The service ticket number is: NC-148-c0a1a091370b437e8fc9-nbi
○ devasc@labvm:~/labs/devnet-src/ptna$ ^C
● devasc@labvm:~/labs/devnet-src/ptna$ python3 02_get-network-device.py
Request status: 200
SWL1      3650      192.168.101.2
R1        ISR4300      192.168.1.2
R3        ISR4300      192.168.2.1
SWR1      3650      10.0.1.2
SWR2      3650      10.0.1.3
R2        ISR4300      192.168.2.2
SWL2      3650      192.168.102.2
SWR4      3650      10.0.1.5
SWR3      3650      10.0.1.4
○ devasc@labvm:~/labs/devnet-src/ptna$
```

Me otorgó la topología de la red, la cual es idéntica a la que se vió en Packet Tracer; esto incluyó la id de la plataforma y la ip de cada dispositivo.

```
print(networkDevice["hostname"], "\t", networkDevice["platformId"], "\t",
networkDevice["managementIpAddress"])
```

```
Problems (Ctrl+Shift+M)

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

● devasc@labvm:~/labs/devnet-src/ptna$ python3 01_get-ticket.py
Ticket request status: 201
The service ticket number is: NC-148-c0a1a091370b437e8fc9-nbi
○ devasc@labvm:~/labs/devnet-src/ptna$ ^C
● devasc@labvm:~/labs/devnet-src/ptna$ python3 02_get-network-device.py
Request status: 200
SWL1      3650      192.168.101.2
R1        ISR4300      192.168.1.2
R3        ISR4300      192.168.2.1
SWR1      3650      10.0.1.2
SWR2      3650      10.0.1.3
R2        ISR4300      192.168.2.2
SWL2      3650      192.168.102.2
SWR4      3650      10.0.1.5
SWR3      3650      10.0.1.4
○ devasc@labvm:~/labs/devnet-src/ptna$
```

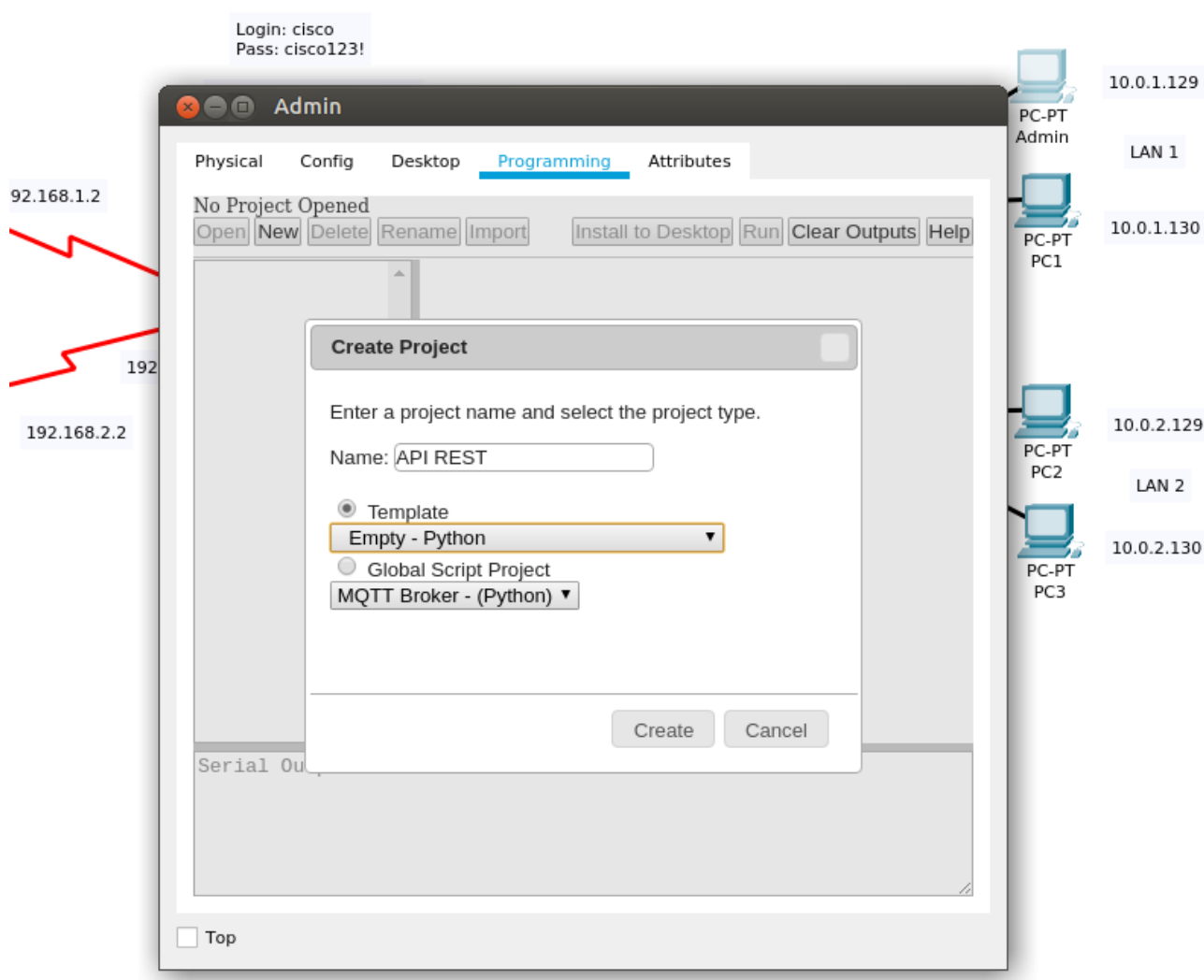
Usando el tercer script pude obtener la información de los hosts, incluyendo su nombre, dirección IP, dirección MAC y la interfaz a la que están conectados.

```
print(host["hostName"], "\t", host["hostIp"], "\t", host["hostMac"], "\t",
      host["connectedInterfaceName"])
```

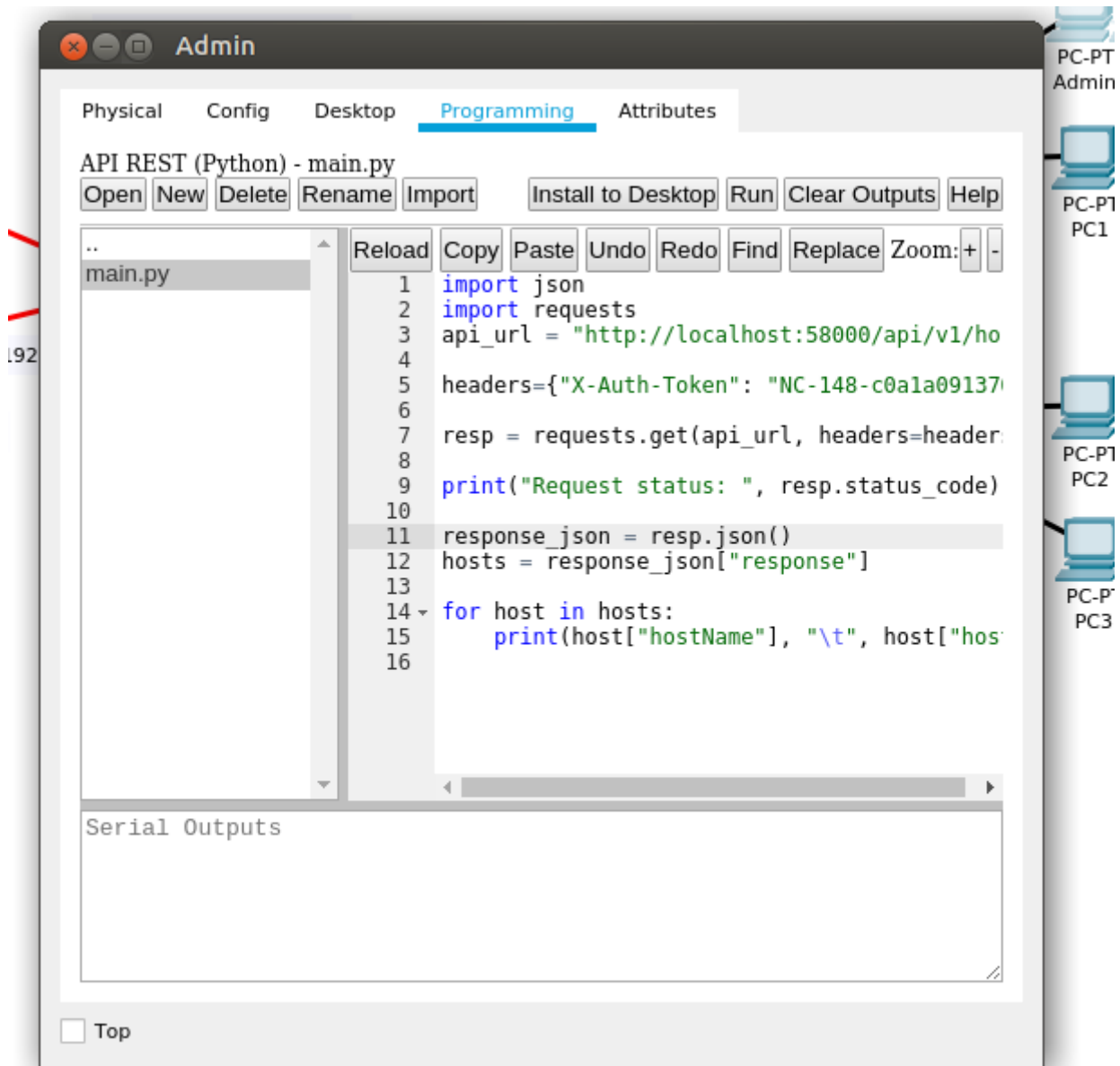
```
devasc@labvm: ~/labs/devnet-src/ptna$ python3 03_get-host.py
Request status: 200
PC4      192.168.102.3    00E0.F96C.155B    GigabitEthernet1/0/24
PC3      10.0.2.130    0004.9A42.C245    GigabitEthernet1/0/24
PC1      10.0.1.129    00E0.A330.3359    GigabitEthernet1/0/22
PC2      10.0.2.129    0060.47C1.A4DB    GigabitEthernet1/0/23
Admin    10.0.1.130    0050.0FCE.B095    GigabitEthernet1/0/21
Example Server  192.168.101.100    000A.413D.D793    GigabitEthernet1/0/3
```

## Parte 6: Enviar solicitudes REST dentro del Packet Tracer

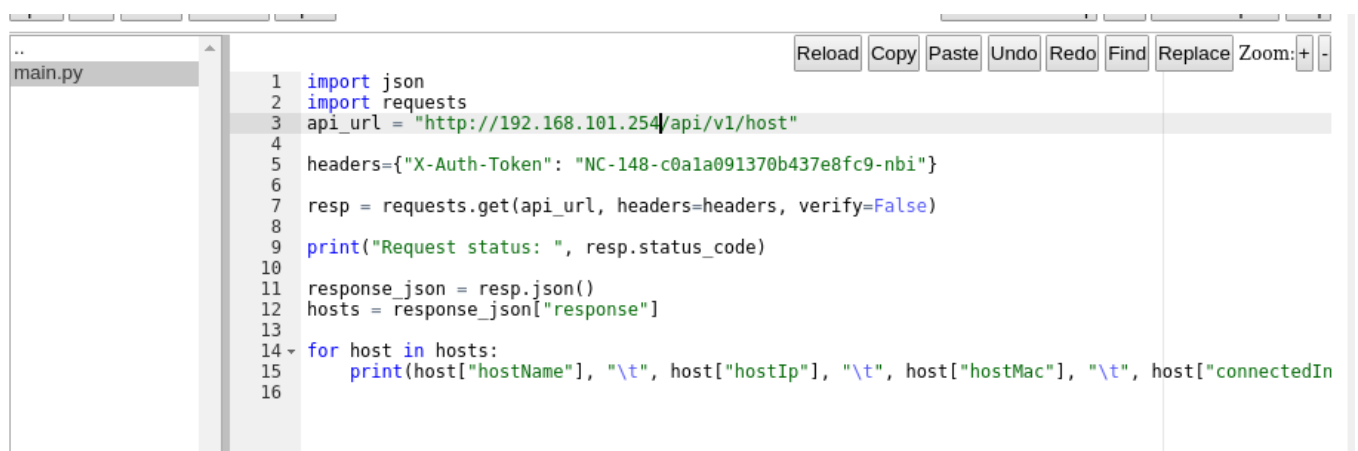
Entro a la pc Admin y creo un nuevo proyecto de programación con python llamado API REST.



Agregando el código del script 3 que se encuentra en devnet-src/ptna en main.py

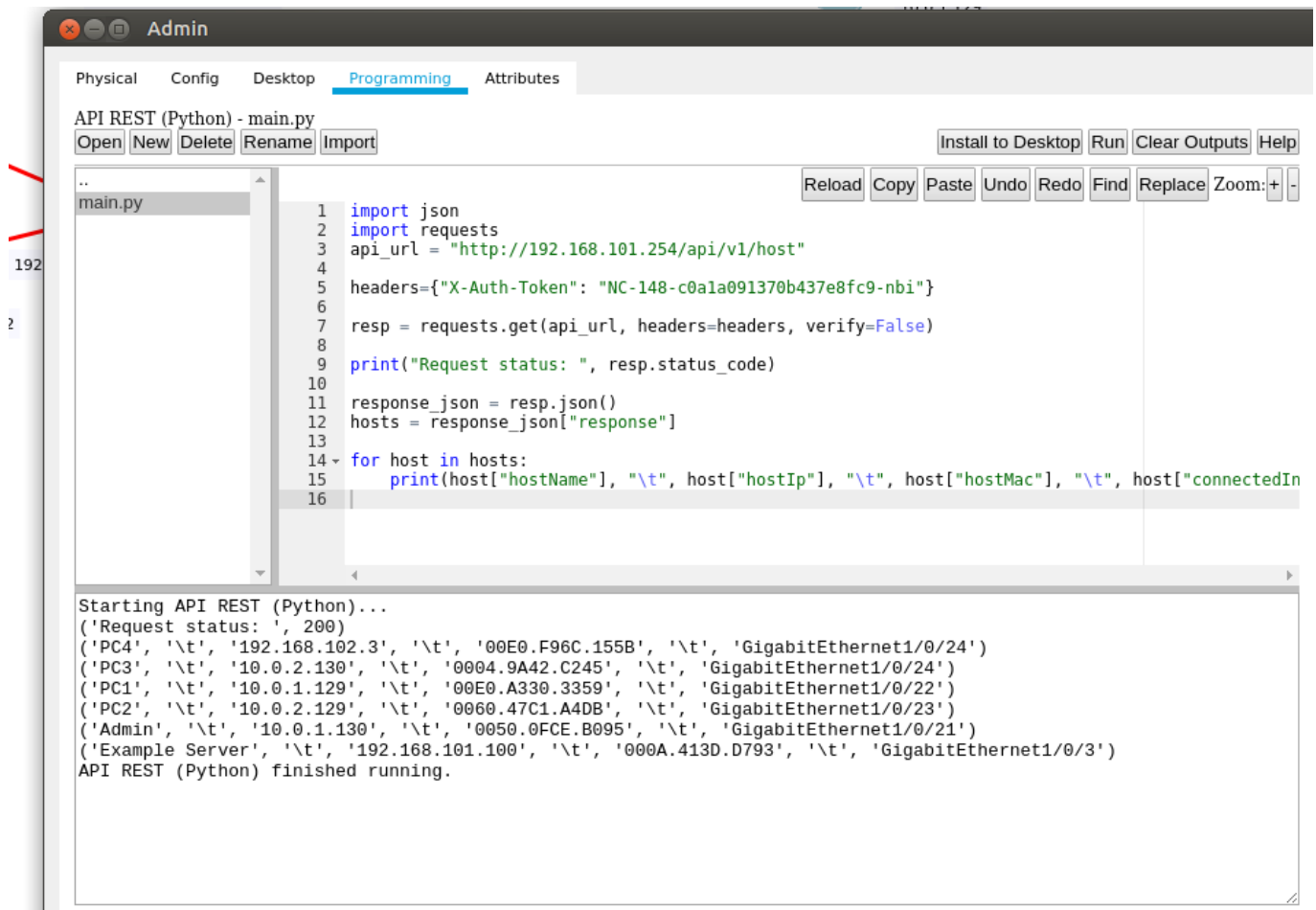


Cambiando el localhost por la ip del controlador 192.168.101.254



Al terminar de correr:





The screenshot shows the Admin IDE interface with the 'Programming' tab selected. The editor displays a Python script named 'main.py' with the following code:

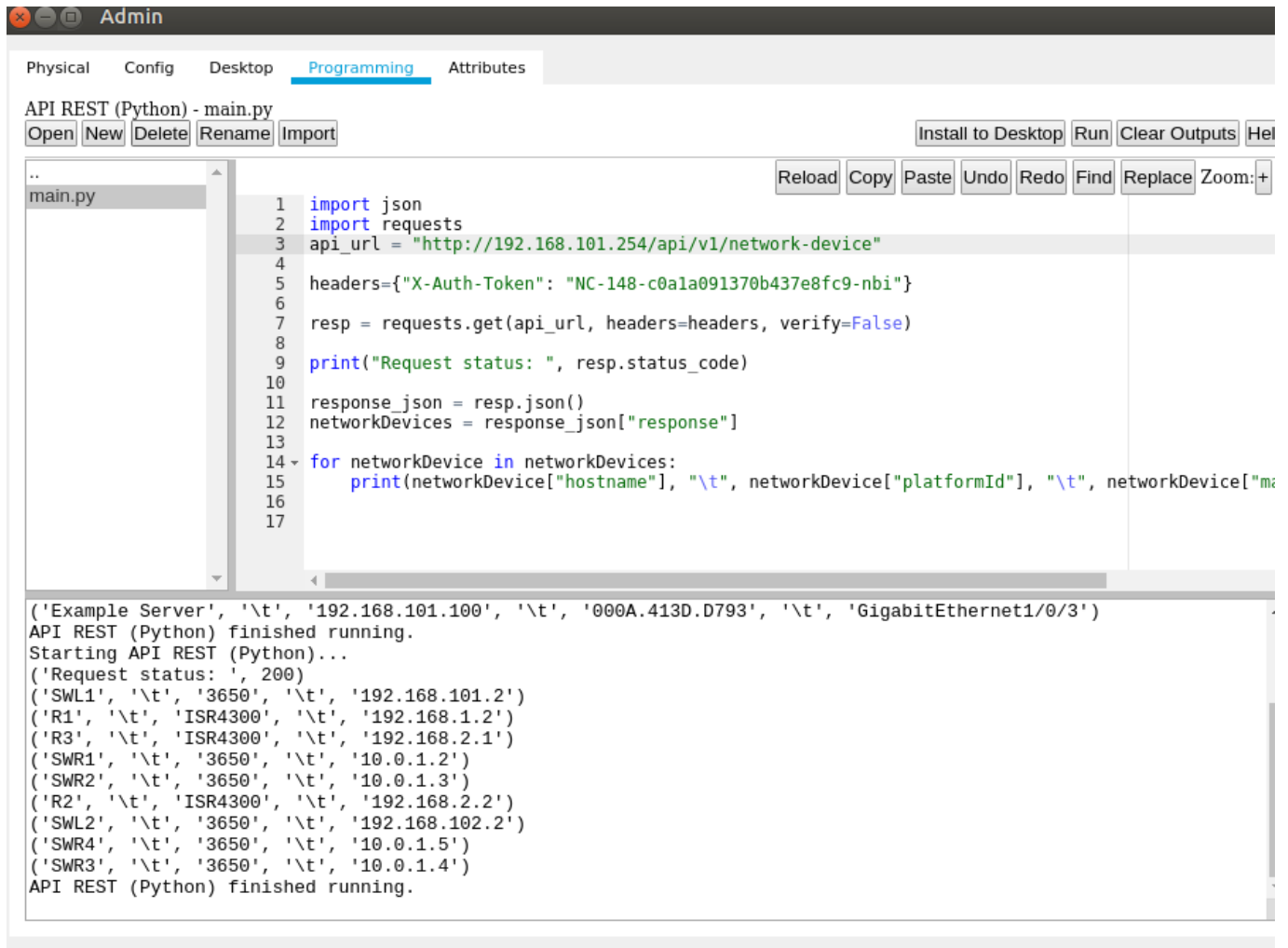
```
1 import json
2 import requests
3 api_url = "http://192.168.101.254/api/v1/host"
4
5 headers={"X-Auth-Token": "NC-148-c0a1a091370b437e8fc9-nbi"}
6
7 resp = requests.get(api_url, headers=headers, verify=False)
8
9 print("Request status: ", resp.status_code)
10
11 response_json = resp.json()
12 hosts = response_json["response"]
13
14 for host in hosts:
15     print(host["hostName"], "\t", host["hostIp"], "\t", host["hostMac"], "\t", host["connectedIn"])
16
```

The output console shows the following text:

```
Starting API REST (Python)...
('Request status: ', 200)
('PC4', '\t', '192.168.102.3', '\t', '00E0.F96C.155B', '\t', 'GigabitEthernet1/0/24')
('PC3', '\t', '10.0.2.130', '\t', '0004.9A42.C245', '\t', 'GigabitEthernet1/0/24')
('PC1', '\t', '10.0.1.129', '\t', '00E0.A330.3359', '\t', 'GigabitEthernet1/0/22')
('PC2', '\t', '10.0.2.129', '\t', '0060.47C1.A4DB', '\t', 'GigabitEthernet1/0/23')
('Admin', '\t', '10.0.1.130', '\t', '0050.0FCE.B095', '\t', 'GigabitEthernet1/0/21')
('Example Server', '\t', '192.168.101.100', '\t', '000A.413D.D793', '\t', 'GigabitEthernet1/0/3')
API REST (Python) finished running.
```

Ahora hacemos lo mismo para el script 2, reemplazando el localhost por la ip del controlador

192.168.101.254



The screenshot shows the Admin interface with the 'Programming' tab selected. The script 'main.py' is open, and its output is displayed in the console.

```
API REST (Python) - main.py
Open New Delete Rename Import Install to Desktop Run Clear Outputs Hel

1 import json
2 import requests
3 api_url = "http://192.168.101.254/api/v1/network-device"
4
5 headers={"X-Auth-Token": "NC-148-c0a1a091370b437e8fc9-nbi"}
6
7 resp = requests.get(api_url, headers=headers, verify=False)
8
9 print("Request status: ", resp.status_code)
10
11 response_json = resp.json()
12 networkDevices = response_json["response"]
13
14 for networkDevice in networkDevices:
15     print(networkDevice["hostname"], "\t", networkDevice["platformId"], "\t", networkDevice["mac"])
16
17
```

Output:

```
('Example Server', '\t', '192.168.101.100', '\t', '000A.413D.D793', '\t', 'GigabitEthernet1/0/3')
API REST (Python) finished running.
Starting API REST (Python)...
('Request status: ', 200)
('SWL1', '\t', '3650', '\t', '192.168.101.2')
('R1', '\t', 'ISR4300', '\t', '192.168.1.2')
('R3', '\t', 'ISR4300', '\t', '192.168.2.1')
('SWR1', '\t', '3650', '\t', '10.0.1.2')
('SWR2', '\t', '3650', '\t', '10.0.1.3')
('R2', '\t', 'ISR4300', '\t', '192.168.2.2')
('SWL2', '\t', '3650', '\t', '192.168.102.2')
('SWR4', '\t', '3650', '\t', '10.0.1.5')
('SWR3', '\t', '3650', '\t', '10.0.1.4')
API REST (Python) finished running.
```

Con ello concluimos la práctica!