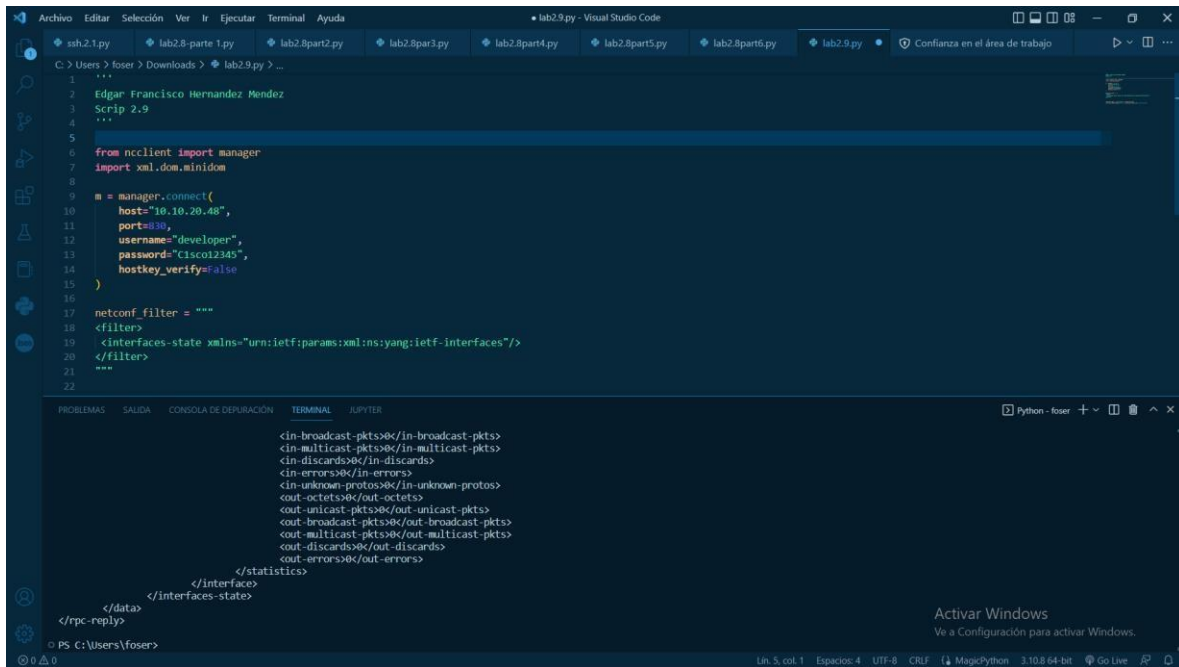


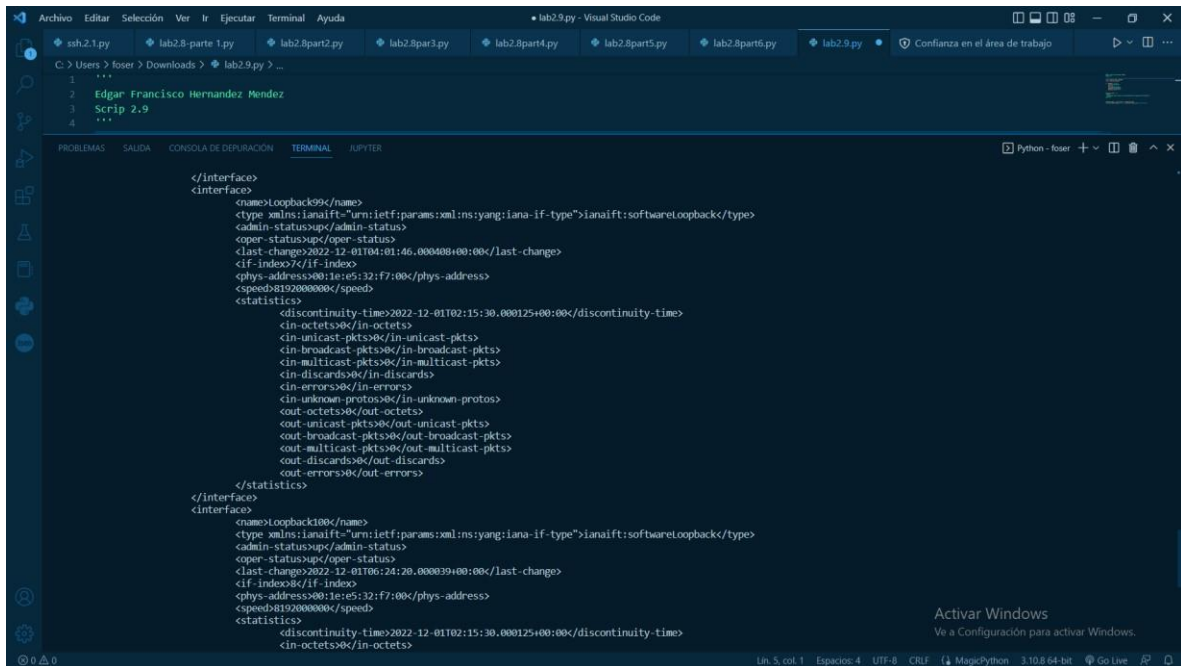
## Ncclient para recuperar la configuración en ejecución



```
1  ...
2  Edgar Francisco Hernandez Mendez
3  Scrip 2.9
4  ...
5
6  from ncclient import manager
7  import xml.dom.minidom
8
9  m = manager.connect(
10     host="10.10.20.48",
11     port=830,
12     username="developer",
13     password="cisco12345",
14     hostkey_verify=False
15 )
16
17 netconf_filter = """
18 <filter>
19   <interfaces-state xmlns="urn:ietf:params:xml:ns:yang:ietf-interfaces"/>
20 </filter>
21 """
22
```

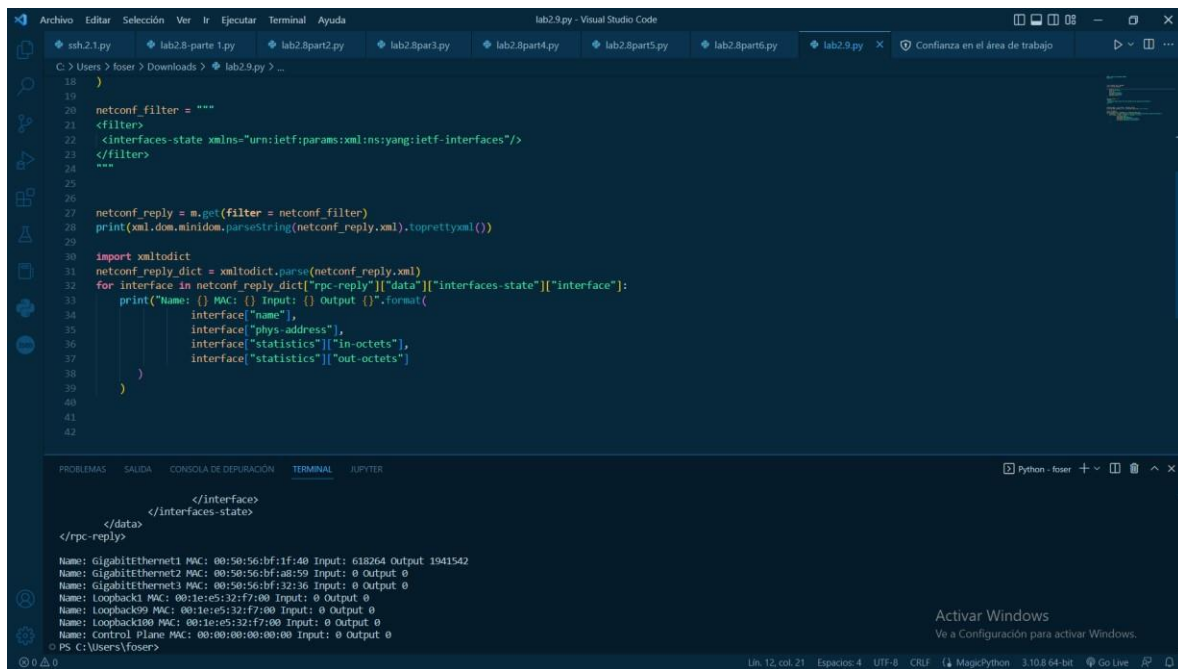
```
<in-broadcast-pkts></in-broadcast-pkts>
<in-multicast-pkts></in-multicast-pkts>
<in-discards></in-discards>
<in-errors></in-errors>
<in-unknown-protos></in-unknown-protos>
<out-octets></out-octets>
<out-unicast-pkts></out-unicast-pkts>
<out-broadcast-pkts></out-broadcast-pkts>
<out-multicast-pkts></out-multicast-pkts>
<out-discards></out-discards>
<out-errors></out-errors>
</statistics>
</interface>
</interfaces-state>
</data>
</rpc-reply>
```

## Salida:



```
</interface>
<interface>
  <name>looback99</name>
  <type xmlns:ianaif="urn:ietf:params:xml:ns:yang:iana-if-type">ianaif:software-loopback</type>
  <admin-status>up</admin-status>
  <oper-status>up</oper-status>
  <last-change>2022-12-01T04:01:46.000408+00:00</last-change>
  <if-index>7</if-index>
  <phys-address>00:1e:e5:32:f7:00</phys-address>
  <speed>102000000</speed>
  <statistics>
    <discontinuity-time>2022-12-01T02:15:30.000125+00:00</discontinuity-time>
    <in-octets></in-octets>
    <in-unicast-pkts></in-unicast-pkts>
    <in-broadcast-pkts></in-broadcast-pkts>
    <in-multicast-pkts></in-multicast-pkts>
    <in-discards></in-discards>
    <in-errors></in-errors>
    <in-unknown-protos></in-unknown-protos>
    <out-octets></out-octets>
    <out-unicast-pkts></out-unicast-pkts>
    <out-broadcast-pkts></out-broadcast-pkts>
    <out-multicast-pkts></out-multicast-pkts>
    <out-discards></out-discards>
    <out-errors></out-errors>
  </statistics>
</interface>
<interface>
  <name>looback100</name>
  <type xmlns:ianaif="urn:ietf:params:xml:ns:yang:iana-if-type">ianaif:software-loopback</type>
  <admin-status>up</admin-status>
  <oper-status>up</oper-status>
  <last-change>2022-12-01T06:24:20.000039+00:00</last-change>
  <if-index>8</if-index>
  <phys-address>00:1e:e5:32:f7:00</phys-address>
  <speed>102000000</speed>
  <statistics>
    <discontinuity-time>2022-12-01T02:15:30.000125+00:00</discontinuity-time>
    <in-octets></in-octets>
```

## Parte 2



The screenshot shows a Visual Studio Code window with a Python file named `lab2.9.py`. The script defines an XML filter for Netconf, retrieves data from a Netconf server, and prints the result in a structured format. The terminal output shows the XML response from the server, which includes details for several interfaces: GigabitEthernet1, GigabitEthernet2, GigabitEthernet3, Loopback1, Loopback99, and Loopback100, along with their MAC addresses, input/output rates, and control plane information.

```
18 )
19
20 netconf_filter = """
21 <filter>
22 <interfaces-state xmlns="urn:ietf:params:xml:ns:yang:ietf-interfaces"/>
23 </filter>
24 """
25
26 netconf_reply = m.get(filter = netconf_filter)
27 print(xml.dom.minidom.parseString(netconf_reply.xml).toprettyxml())
28
29 import xmltodict
30 netconf_reply_dict = xmltodict.parse(netconf_reply.xml)
31 for interface in netconf_reply_dict["rpc-reply"]["data"]["interfaces-state"]["interface"]:
32     print("Name: {} MAC: {} Input: {} Output: {}".format(
33         interface["name"],
34         interface["phys-address"],
35         interface["statistics"]["in-octets"],
36         interface["statistics"]["out-octets"]
37     ))
38
39
40
41
42
```

```
</interface>
</interfaces-state>
</data>
</rpc-reply>
Name: GigabitEthernet1 MAC: 00:50:56:b1:f1:40 Input: 618264 Output 1941542
Name: GigabitEthernet2 MAC: 00:50:56:b1:a8:50 Input: 0 Output 0
Name: GigabitEthernet3 MAC: 00:50:56:b1:32:36 Input: 0 Output 0
Name: Loopback1 MAC: 00:1e:e5:32:f7:00 Input: 0 Output 0
Name: Loopback99 MAC: 00:1e:e5:32:f7:00 Input: 0 Output 0
Name: Loopback100 MAC: 00:1e:e5:32:f7:00 Input: 0 Output 0
Name: Control Plane MAC: 00:00:00:00:00:00 Input: 0 Output 0
PS C:\Users\foser>
```

## Conclusión

Si la aplicación cliente solicita la salida en formato de etiqueta XML, el servidor NETCONF encierra su respuesta en el elemento de etiqueta de respuesta específico que corresponde al elemento de etiqueta de solicitud, que luego se encierra en un elemento de etiqueta.

Es bueno trabajar con este tipo de laboratorios, investigar de diferentes fuentes de información.