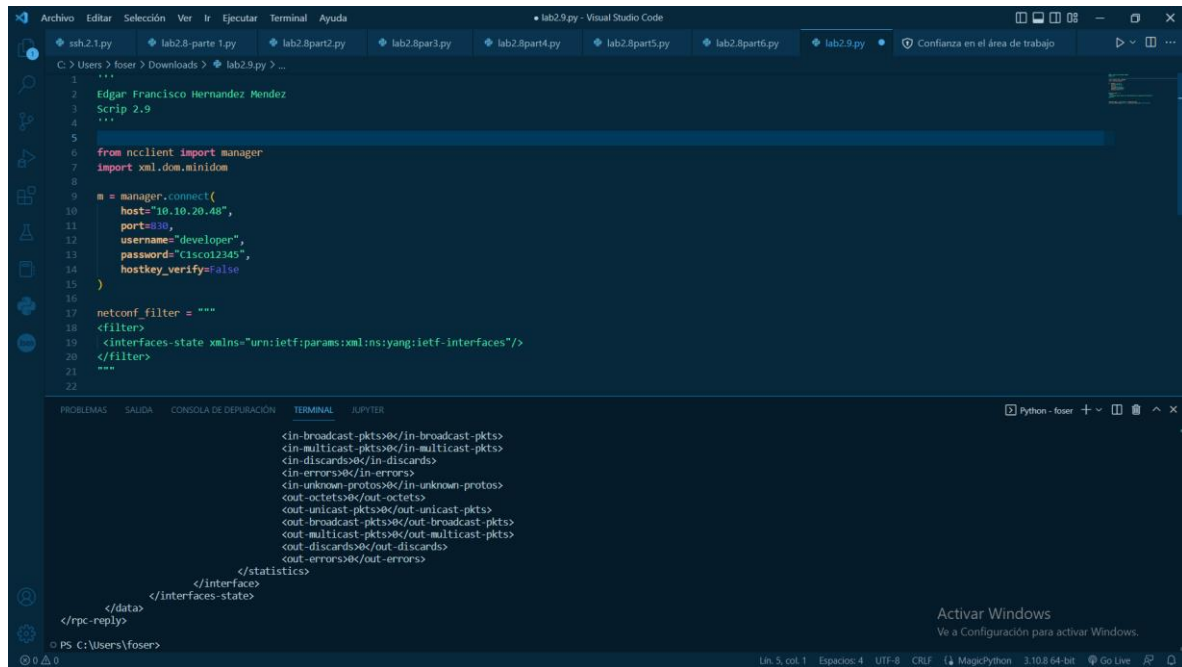


## Ncclient para recuperar la configuracion enn ejecucion



The screenshot shows a Visual Studio Code editor with a Python script named `lab2.9.py` open. The script uses the `ncclient` library to connect to a network device and retrieve the configuration. The terminal output shows the XML configuration for the interfaces.

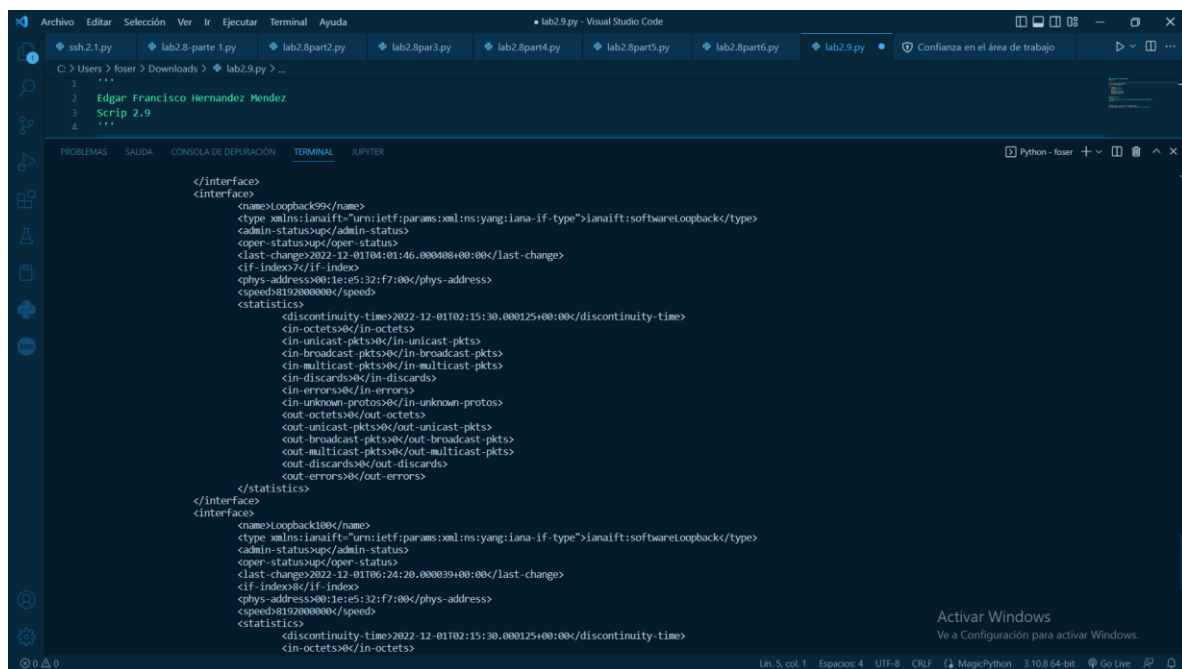
```
1  """
2  Edgar Francisco Hernandez Mendez
3  Scrip 2.9
4  """
5
6  from ncclient import manager
7  import xml.dom.minidom
8
9  m = manager.connect(
10     host="10.10.20.48",
11     port=830,
12     username="developer",
13     password="cisco12345",
14     hostkey_verify=False
15 )
16
17 netconf_filter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

```

<in-broadcast-pkts>0</in-broadcast-pkts>
<in-multicast-pkts>0</in-multicast-pkts>
<in-discards>0</in-discards>
<in-errors>0</in-errors>
<in-unknown-protos>0</in-unknown-protos>
<out-octets>0</out-octets>
<out-unicast-pkts>0</out-unicast-pkts>
<out-broadcast-pkts>0</out-broadcast-pkts>
<out-multicast-pkts>0</out-multicast-pkts>
<out-discards>0</out-discards>
<out-errors>0</out-errors>
</statistics>
</interface>
</interfaces-state>
</data>
</rpc-reply>

```

Salida:



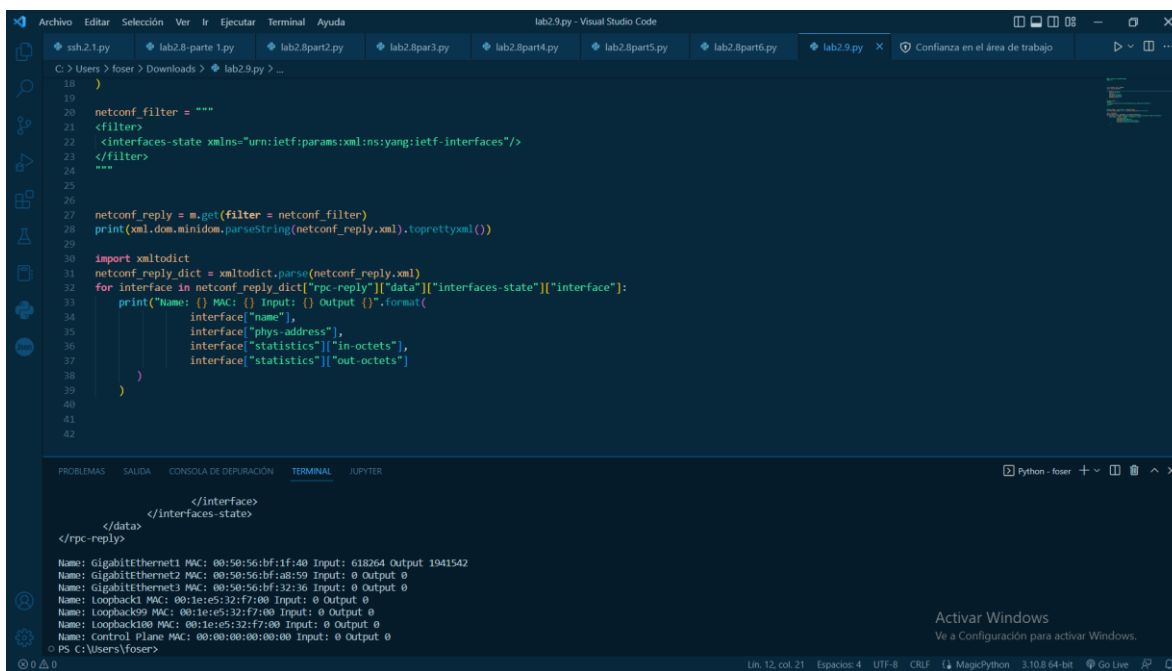
The screenshot shows the same Visual Studio Code editor with the `lab2.9.py` script. The terminal output now shows the XML configuration for the interfaces, including details like name, type, status, and statistics.

```

<interface>
<interface>
  <name>Loopback99</name>
  <type xmlns:ianaif="urn:ietf:params:xml:ns:yang:iana-if-type">ianaif:softwareloopback</type>
  <admin-status>up</admin-status>
  <oper-status>up</oper-status>
  <last-change>2022-12-01T04:01:46.000000+00:00</last-change>
  <if-index>7</if-index>
  <phys-address>00:1e:5:32:f7:00</phys-address>
  <speed>1000000000</speed>
  <statistics>
    <discontinuity-time>2022-12-01T02:15:30.000125+00:00</discontinuity-time>
    <in-octets>0</in-octets>
    <in-unicast-pkts>0</in-unicast-pkts>
    <in-broadcast-pkts>0</in-broadcast-pkts>
    <in-multicast-pkts>0</in-multicast-pkts>
    <in-discards>0</in-discards>
    <in-errors>0</in-errors>
    <in-unknown-protos>0</in-unknown-protos>
    <out-octets>0</out-octets>
    <out-unicast-pkts>0</out-unicast-pkts>
    <out-broadcast-pkts>0</out-broadcast-pkts>
    <out-multicast-pkts>0</out-multicast-pkts>
    <out-discards>0</out-discards>
    <out-errors>0</out-errors>
  </statistics>
</interface>
<interface>
  <name>Loopback100</name>
  <type xmlns:ianaif="urn:ietf:params:xml:ns:yang:iana-if-type">ianaif:softwareloopback</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:5:32:f7:00</phys-address>
  <speed>1000000000</speed>
  <statistics>
    <discontinuity-time>2022-12-01T02:15:30.000125+00:00</discontinuity-time>
    <in-octets>0</in-octets>

```

## Parte 2



The screenshot shows a Visual Studio Code editor window with a Python script in the main editor and its output in the terminal. The script is located at `C:\Users\foser\Downloads\lab2.9.py` and contains the following code:

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

The terminal output shows the XML response from the netconf client and a list of network interfaces with their MAC addresses and statistics:

```
</interface>
</interfaces-state>
</data>
</rpc-reply>
Name: GigabitEthernet1 MAC: 00:50:56:b3:1f:40 Input: 618264 Output 1941542
Name: GigabitEthernet2 MAC: 00:50:56:b3:a8:59 Input: 0 Output 0
Name: GigabitEthernet3 MAC: 00:50:56:b3:32:36 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
```

The terminal window title is "Python - foser" and the status bar at the bottom indicates "Lin. 12, col. 21", "Espacios: 4", "UTF-8", "CRLF", "MagicPython", "3.10.8 64-bit", and "Go Live".