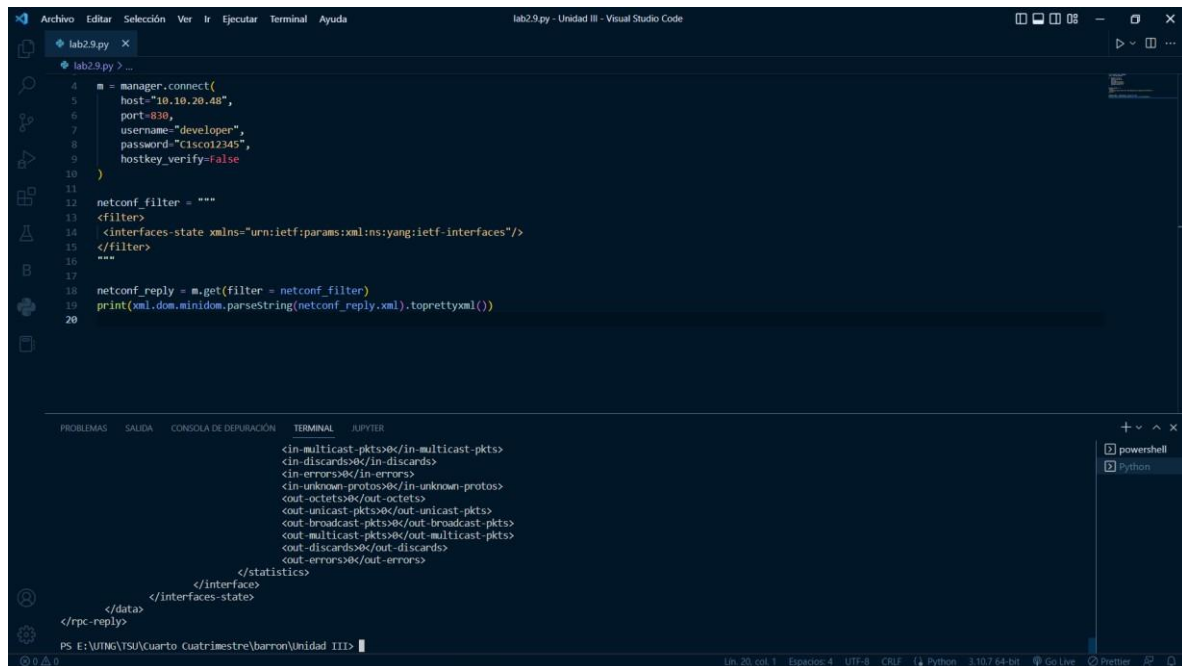


Utilice ncclient para recuperar la configuración en ejecución del dispositivo.

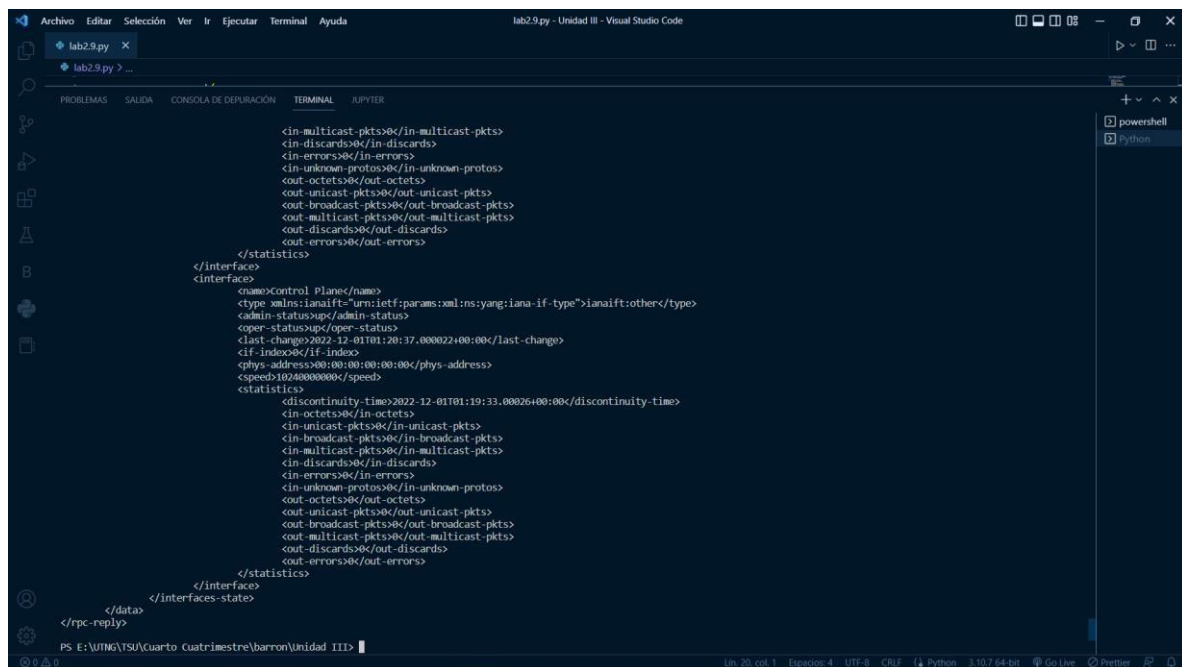


```
lab2.9.py
4 m = manager.connect(
5     host="10.10.20.48",
6     port=830,
7     username="developer",
8     password="Cisco12345",
9     hostkey_verify=False
10 )
11
12 netconf_filter = """
13 <filter>
14   <interfaces-state xmlns="urn:ietf:params:xml:ns:yang:ietf-interfaces"/>
15 </filter>
16 """
17
18 netconf_reply = m.get(filter = netconf_filter)
19 print(xml.dom.minidom.parsestring(netconf_reply.xml).toprettyxml())
20
```

```

<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>
PS E:\ITING\TSU\Cuarto Cuatrimestre\barron\Unidad III>
```

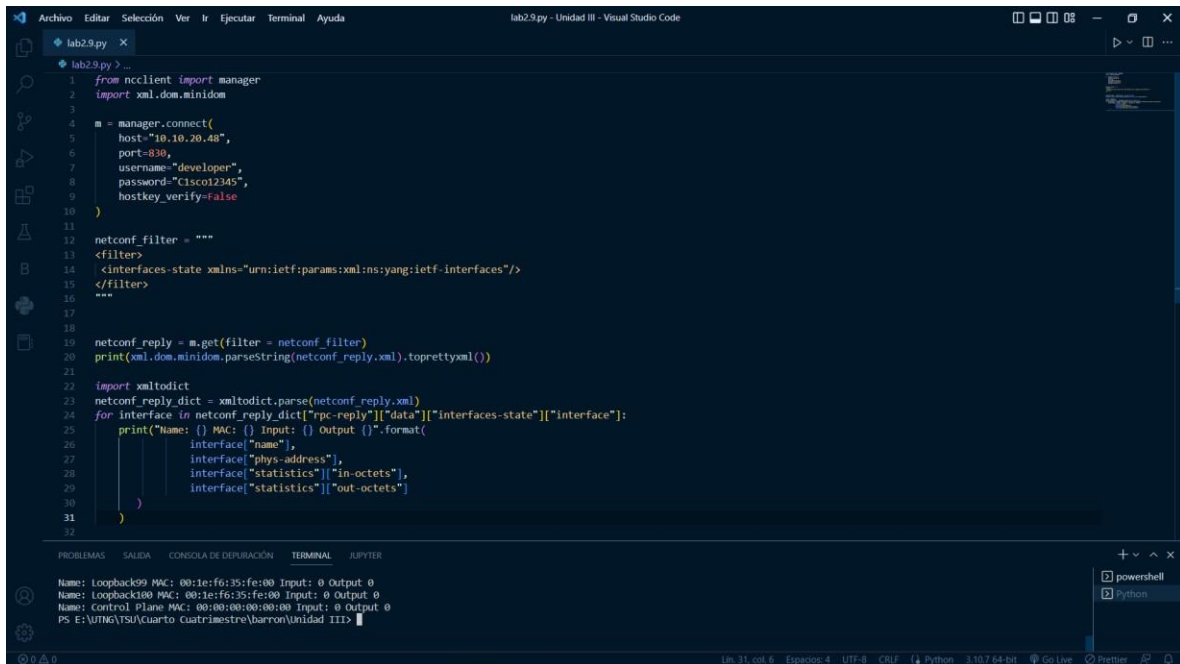
Salida:



```
lab2.9.py
4 m = manager.connect(
5     host="10.10.20.48",
6     port=830,
7     username="developer",
8     password="Cisco12345",
9     hostkey_verify=False
10 )
11
12 netconf_filter = """
13 <filter>
14   <interfaces-state xmlns="urn:ietf:params:xml:ns:yang:ietf-interfaces"/>
15 </filter>
16 """
17
18 netconf_reply = m.get(filter = netconf_filter)
19 print(xml.dom.minidom.parsestring(netconf_reply.xml).toprettyxml())
20
```

```

<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>Control Plane</name>
  <type xmlns:ianaif="urn:ietf:params:xml:ns:yang:iana-if-type">ianaif:other</type>
  <admin-status><admin-status>
    <oper-status><oper-status>
      <last-change>2022-12-01T01:20:17.000022+00:00</last-change>
      <if-index></if-index>
      <phys-address><phys-address>00:00:00:00:00:00</phys-address>
      <speed>1000000000</speed>
      <statistics>
        <discontinuity-time>2022-12-01T01:19:33.000026+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>
  </interfaces-state>
</data>
</rpc-reply>
PS E:\ITING\TSU\Cuarto Cuatrimestre\barron\Unidad III>
```

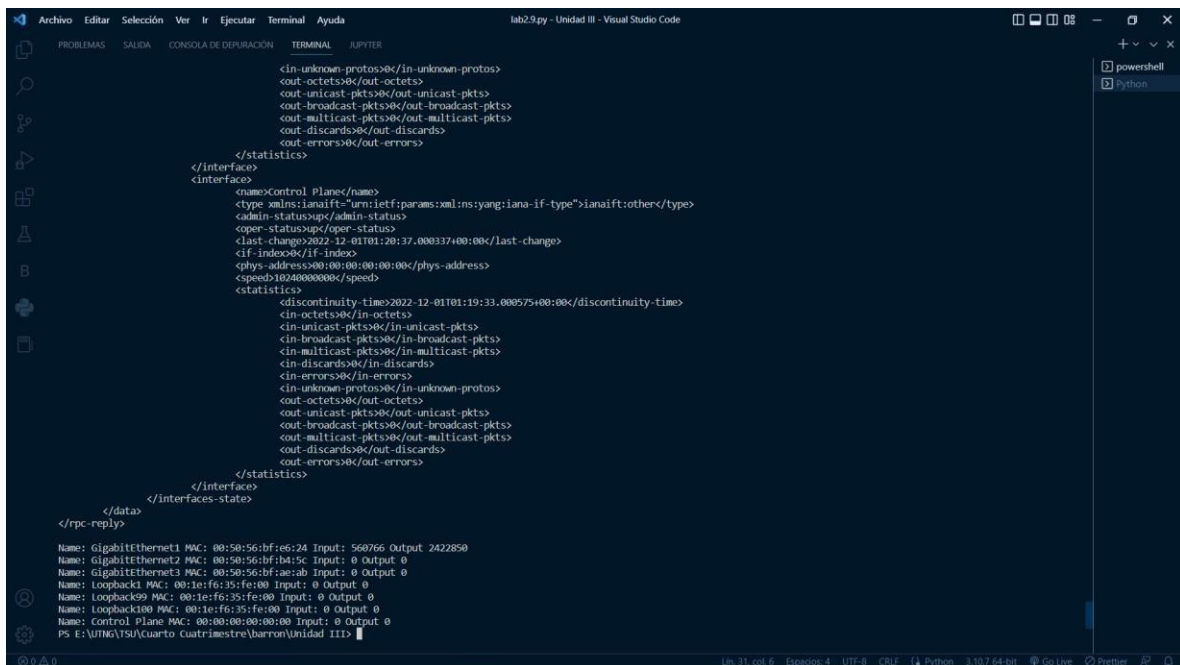


```
1 from ncclient import manager
2 import xml.dom.minidom
3
4 m = manager.connect(
5     host="10.10.20.48",
6     port=830,
7     username="developer",
8     password="Cisco12345",
9     hostkey_verify=False
10 )
11
12 netconf_filter = """
13 <filter>
14   <interfaces-state xmlns="urn:ietf:params:xml:ns:yang:ietf-interfaces"/>
15 </filter>
16 """
17
18 netconf_reply = m.get(filter=netconf_filter)
19 print(xml.dom.minidom.parsestring(netconf_reply.xml).toprettyxml())
20
21 import xmltodict
22 netconf_reply_dict = xmltodict.parse(netconf_reply.xml)
23 for interface in netconf_reply_dict["rpc-reply"]["data"]["interfaces-state"]["interface"]:
24     print("Name: {} MAC: {} Input: {} Output {}".format(
25         interface["name"],
26         interface["phys-address"],
27         interface["statistics"]["in-octets"],
28         interface["statistics"]["out-octets"]
29     ))
30
31
32
```

PROBLEMAS SALIDA CONSOLA DE DEPURACIÓN TERMINAL JUPYTER

Name: Loopback99 MAC: 00:1e:f6:35:fe:00 Input: 0 Output 0
Name: Loopback100 MAC: 00:1e:f6:35:fe:00 Input: 0 Output 0
Name: Control Plane MAC: 00:00:00:00:00:00 Input: 0 Output 0
PS E:\UTMG\TSU\Cuarto Cuatrimestre\harron\Unidad III>

Salida:



```
<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>Control Plane</name>
  <type xmlns:ianaif="urn:ietf:params:xml:ns:yang:iana-if-type">ianaif:others</type>
  <admin-status>up</admin-status>
  <oper-status>up</oper-status>
  <last-change>2022-12-01T01:20:37.000337+00:00</last-change>
  <if-index></if-index>
  <phys-address>00:00:00:00:00:00</phys-address>
  <speed>1024000000</speed>
  <statistics>
    <discontinuity-time>2022-12-01T01:19:33.000575+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>
</interfaces-state>
</data>
</rpc-reply>

Name: GigabitEthernet1 MAC: 00:50:56:bfb4:5c Input: 560766 Output 2422850
Name: GigabitEthernet2 MAC: 00:50:56:bfb4:5c Input: 0 Output 0
Name: GigabitEthernet3 MAC: 00:50:56:bfb4:5c Input: 0 Output 0
Name: Loopback1 MAC: 00:1e:f6:35:fe:00 Input: 0 Output 0
Name: Loopback99 MAC: 00:1e:f6:35:fe:00 Input: 0 Output 0
Name: Loopback100 MAC: 00:1e:f6:35:fe:00 Input: 0 Output 0
Name: Control Plane MAC: 00:00:00:00:00:00 Input: 0 Output 0
PS E:\UTMG\TSU\Cuarto Cuatrimestre\harron\Unidad III>
```

Conclusión: En este laboratorio igualmente se utilizó ncclient para conectarse y revisar la configuración que se estaba ejecutando en el router, mostrando una respuesta en XML ya formateada para su fácil lectura del humano, enviando comandos para revisar la configuración actual. Utilizando xmltodict que nos permite tratar archivos XML de una manera muy fácil, ya que convierte los archivos XML en una estructura de datos de tipo diccionario.

De igual manera, se vienen utilizando protocolos ya definidos en los laboratorios pasados, ahorita lo nuevo fue que se utilizó xmldict para formatear la respuesta en modo de diccionarios y así como ya lo mencione sea más legible para los humanos y se pueda interpretar mejor la respuesta que nos dio.