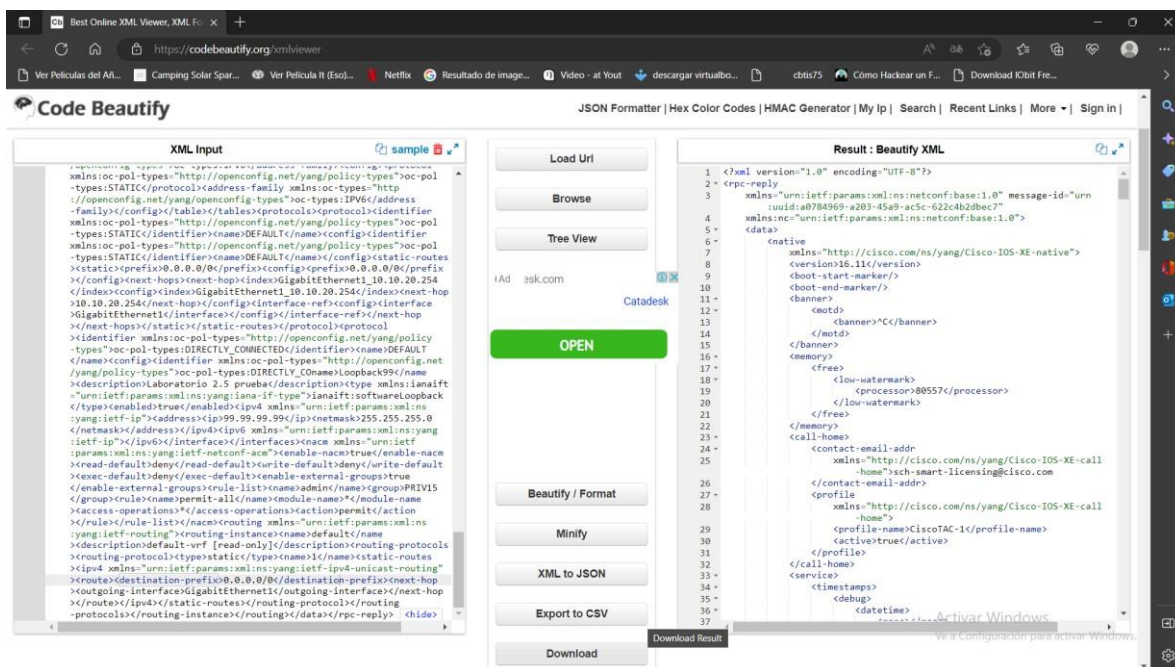


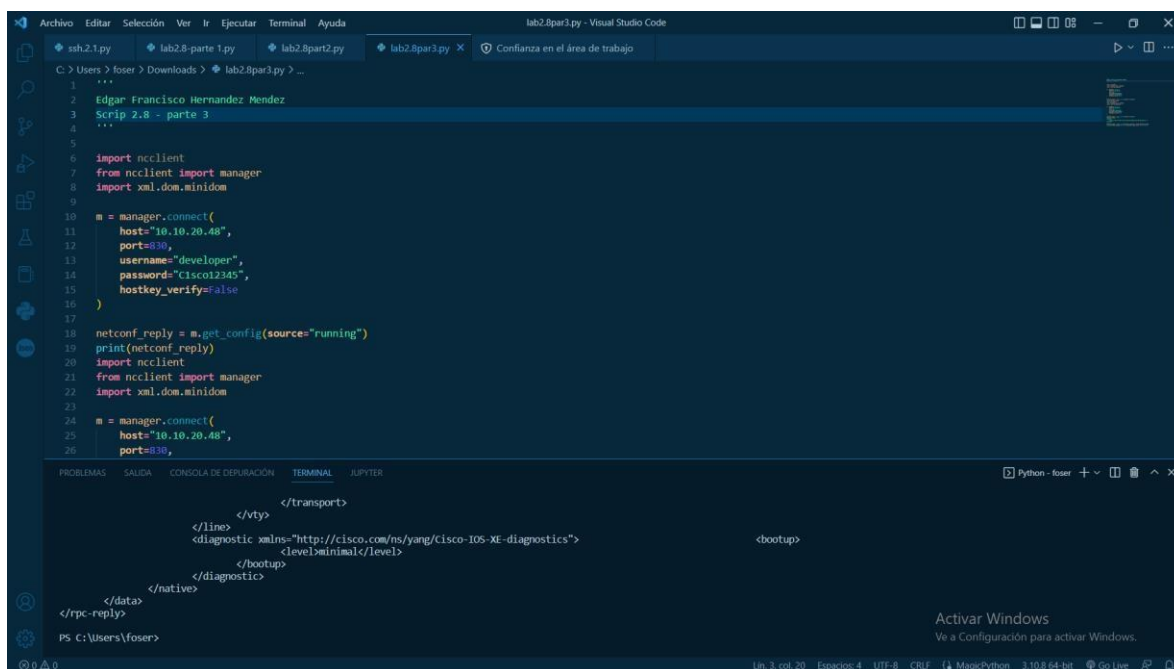
Se usa ncclient para recuperar la configuracion en ejecucion del dispositivo

[illegible]

Se utiliza CodeBeautify.com para evaluar



Usar la función `toprettyxml()` para embeceller

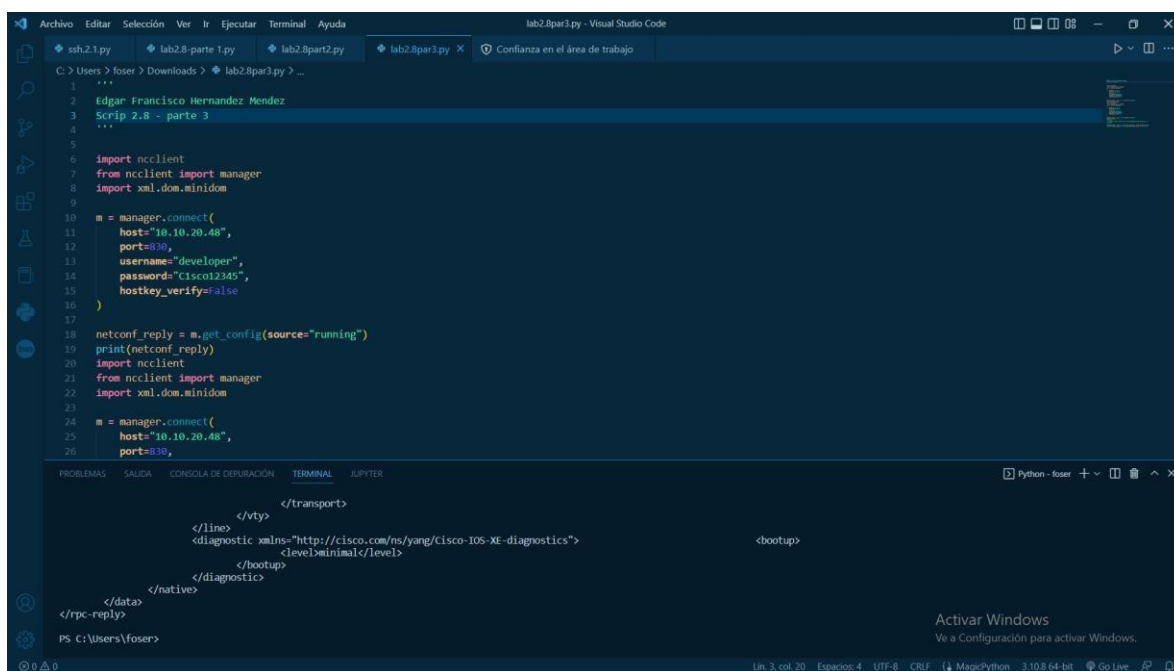


The screenshot shows a Visual Studio Code window with a Python script in the editor and its output in the terminal. The script is a Python program that connects to a Cisco device via Netmiko and retrieves the configuration. The terminal output shows the XML representation of the configuration, which is not yet pretty-printed.

```
1 ***
2 Edgar Francisco Hernandez Mendez
3 Scrip 2.8 - parte 3
4 ***
5
6 import ncclient
7 from ncclient import manager
8 import xml.dom.minidom
9
10 m = manager.connect(
11     host="10.10.20.48",
12     port=830,
13     username="developer",
14     password="Cisco12345",
15     hostkey_verify=False
16 )
17
18 netconf_reply = m.get_config(source="running")
19 print(netconf_reply)
20 import ncclient
21 from ncclient import manager
22 import xml.dom.minidom
23
24 m = manager.connect(
25     host="10.10.20.48",
26     port=830,
```

```
</transport>
</vty>
</line>
<diagnostic xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-diagnostics"
<level>minimal</level>
</bootup>
</diagnostic>
</native>
</data>
</rpc-reply>
PS C:\Users\foser>
```

recuperar una configuración definida por un modelo YANG específico

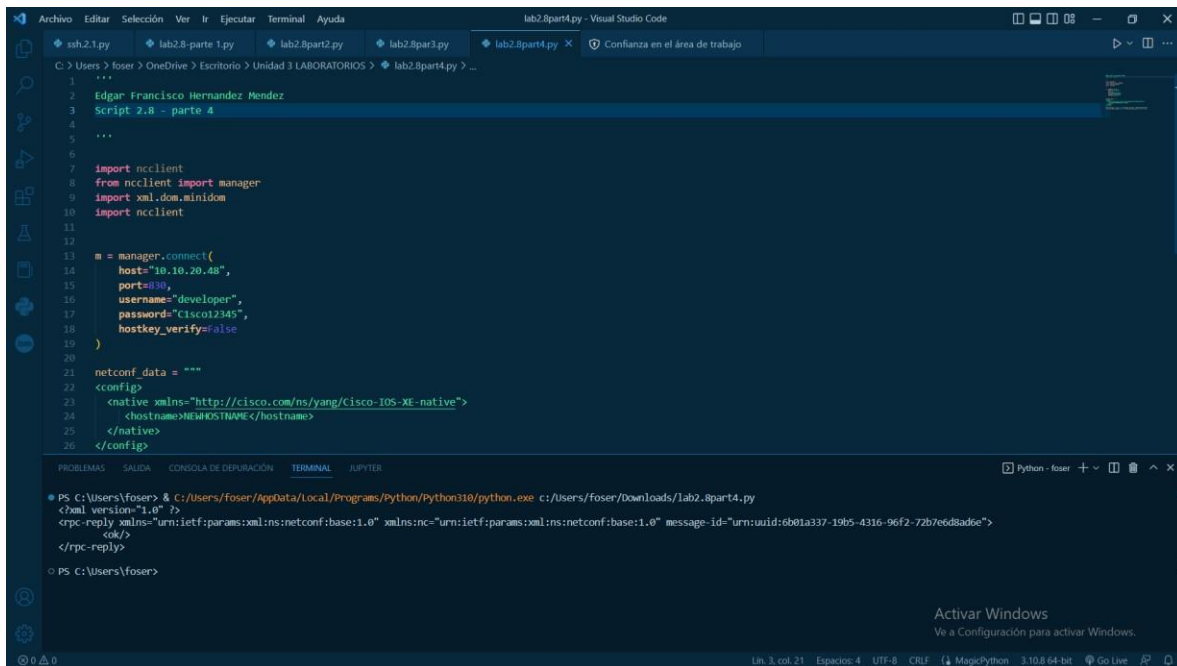


The screenshot shows a Visual Studio Code window with a Python script in the editor and its output in the terminal. The script is a Python program that connects to a Cisco device via Netmiko and retrieves the configuration. The terminal output shows the XML representation of the configuration, which is not yet pretty-printed.

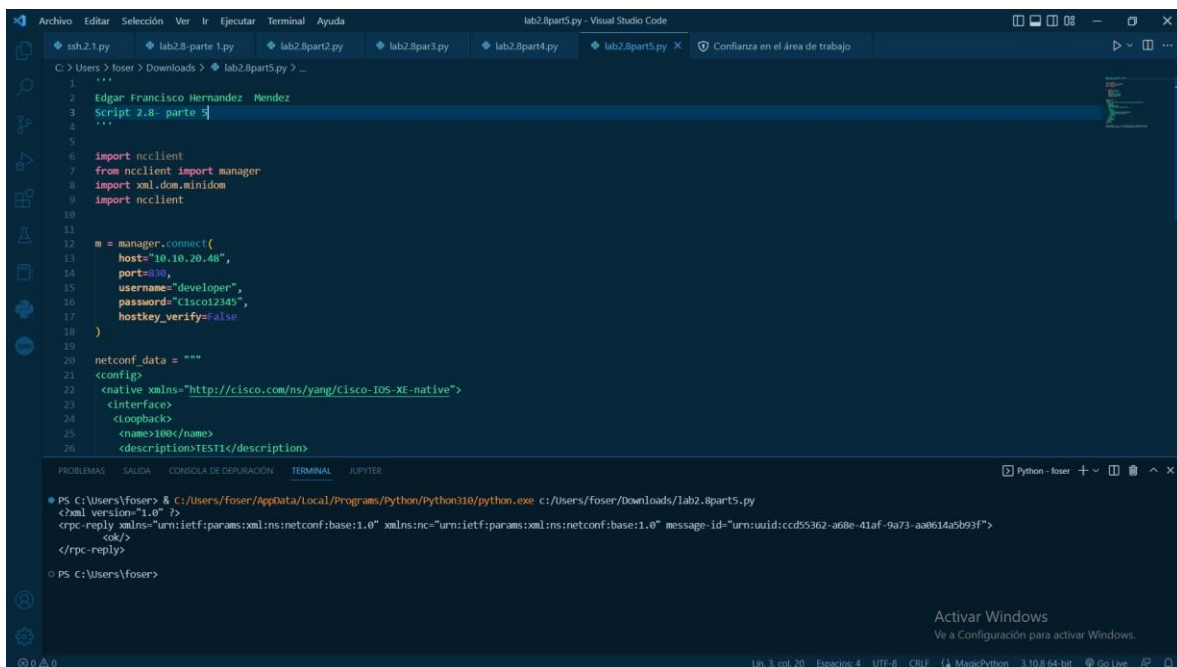
```
1 ***
2 Edgar Francisco Hernandez Mendez
3 Scrip 2.8 - parte 3
4 ***
5
6 import ncclient
7 from ncclient import manager
8 import xml.dom.minidom
9
10 m = manager.connect(
11     host="10.10.20.48",
12     port=830,
13     username="developer",
14     password="Cisco12345",
15     hostkey_verify=False
16 )
17
18 netconf_reply = m.get_config(source="running")
19 print(netconf_reply)
20 import ncclient
21 from ncclient import manager
22 import xml.dom.minidom
23
24 m = manager.connect(
25     host="10.10.20.48",
26     port=830,
```

```
</transport>
</vty>
</line>
<diagnostic xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-diagnostics"
<level>minimal</level>
</bootup>
</diagnostic>
</native>
</data>
</rpc-reply>
PS C:\Users\foser>
```

Crema una interfaz loopback y la salida da ( OK)

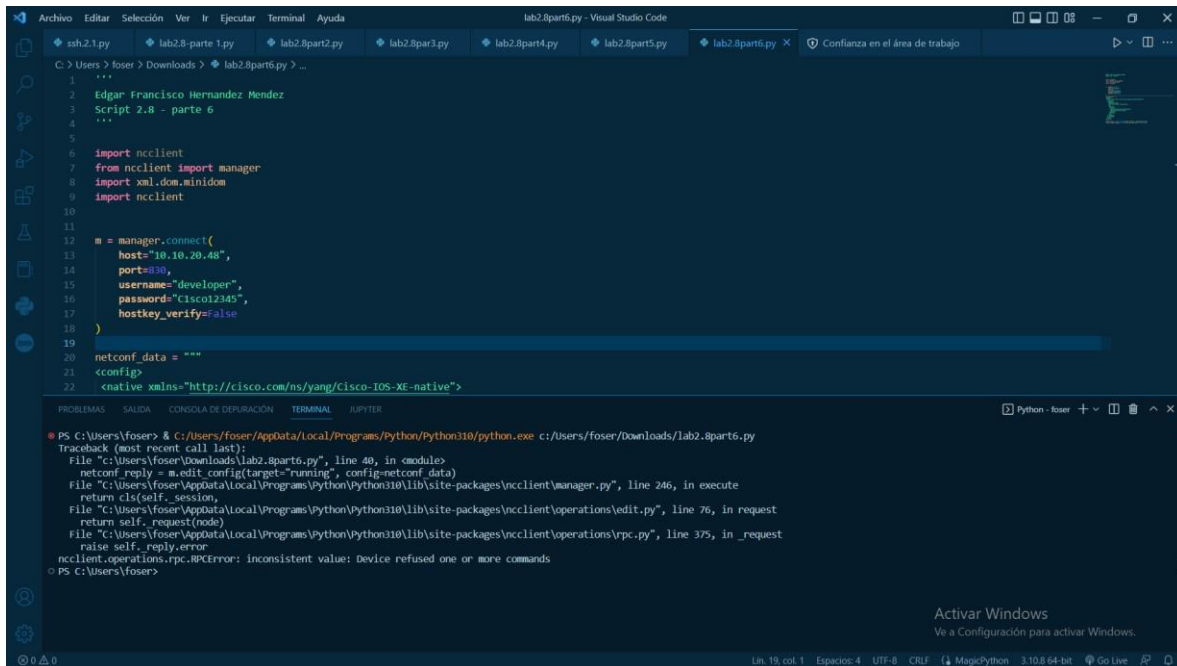


```
lab2.8part4.py - Visual Studio Code
C:\Users\foser> OneDrive > Escritorio > Unidad 3 LABORATORIOS > lab2.8part4.py > ...
1 ***
2 Edgar Francisco Hernandez Mendez
3 Script 2.8 - parte 4
4 ***
5
6
7 import ncclient
8 from ncclient import manager
9 import xml.dom.minidom
10 import ncclient
11
12
13 m = manager.connect(
14     host="10.10.20.48",
15     port=830,
16     username="developer",
17     password="Cisc012345",
18     hostkey_verify=False
19 )
20
21 netconf_data = """
22 <config>
23   <native xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-native">
24     <hostname>HUBIKOSTNAME</hostname>
25   </native>
26 </config>
27 """
28
29 PS C:\Users\foser> & C:\Users\foser\AppData\Local\Programs\Python\Python310\python.exe c:\Users\foser\Downloads\lab2.8part4.py
<?xml version="1.0" ?>
<rpc-reply xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" xmlns:nc="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="urn:uuid:6b01a337-19b5-4316-96f2-72b7e6d8ad6e">
  <ok/>
</rpc-reply>
PS C:\Users\foser>
```



```
lab2.8part5.py - Visual Studio Code
C:\Users\foser> Downloads > lab2.8part5.py > ...
1 ***
2 Edgar Francisco Hernandez Mendez
3 Script 2.8- parte 5
4 ***
5
6
7 import ncclient
8 from ncclient import manager
9 import xml.dom.minidom
10 import ncclient
11
12
13 m = manager.connect(
14     host="10.10.20.48",
15     port=830,
16     username="developer",
17     password="Cisc012345",
18     hostkey_verify=False
19 )
20
21 netconf_data = """
22 <config>
23   <native xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-native">
24     <interface>
25       <loopback>
26         <name>100</name>
27         <description>TEST1</description>
28       </loopback>
29     </interface>
30   </native>
31 </config>
32 """
33
34 PS C:\Users\foser> & C:\Users\foser\AppData\Local\Programs\Python\Python310\python.exe c:\Users\foser\Downloads\lab2.8part5.py
<?xml version="1.0" ?>
<rpc-reply xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" xmlns:nc="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="urn:uuid:ccd55362-a68e-41af-9a73-aa0614a5b93f">
  <ok/>
</rpc-reply>
PS C:\Users\foser>
```

Intente crear una nueva interfaz de loopback con una dirección IP en conflicto.



The screenshot shows a Visual Studio Code window with a file explorer on the left displaying a directory of files named lab2.8part1.py through lab2.8part6.py. The main editor area shows the content of lab2.8part6.py, which is a Python script for connecting to a Cisco device via ncclient and retrieving configuration data. The script includes imports for ncclient, xml, and minidom, and defines a netconf\_data variable with a specific YANG model URI. The bottom panel shows the terminal output, which includes a traceback indicating an RPCError: inconsistent value: Device refused one or more commands.

```
1 '''
2 Edgar Francisco Hernandez Mendez
3 Script 2.8 - parte 6
4 '''
5
6 import ncclient
7 from ncclient import manager
8 import xml.dom.minidom
9 import ncclient
10
11
12 m = manager.connect(
13     host="10.10.20.48",
14     port=830,
15     username="developer",
16     password="cisco12345",
17     hostkey_verify=False
18 )
19
20 netconf_data = """
21 <config>
22 <native xmlns="http://cisco.com/ns/yang/cisco-IOS-XE-native">
```

PS C:\Users\foser> & C:\Users\foser\AppData\Local\Programs\Python\Python310\python.exe c:\Users\foser\Downloads\lab2.8part6.py  
Traceback (most recent call last):  
File "C:\Users\foser\Downloads\lab2.8part6.py", line 40, in <module>  
netconf\_reply = m.edit\_config(target='running', config=netconf\_data)  
File "C:\Users\foser\AppData\Local\Programs\Python\Python310\lib\site-packages\ncclient\manager.py", line 246, in execute  
return cls(self, session, request, response)  
File "C:\Users\foser\AppData\Local\Programs\Python\Python310\lib\site-packages\ncclient\operations\edit.py", line 76, in request  
return self.\_request(node)  
File "C:\Users\foser\AppData\Local\Programs\Python\Python310\lib\site-packages\ncclient\operations\rpc.py", line 375, in \_request  
raise self.\_reply.error  
ncclient.operations.rpc.RPCError: inconsistent value: Device refused one or more commands  
PS C:\Users\foser>

## Conclusión

¿Qué comandos utilizo frecuentemente?

Ncclient,xml y los importamos

¿Qué tipo de información obtengo?

Se obtiene mediante CodeBeautify.com en entorno gráfico y después también podemos ver. La función toprettyxml() para embeceller y recuperar una configuración definida por un modelo YANG específico

¿Cuál es el sentido de evaluar la respuesta?

Una interfaz loopback y la salida da ( OK)

¿Qué tipo de configuración se requiere?

Un Netconf (netconf\_data)

¿Qué es una interfaz loopback?

La interfaz de **Loopback** es la interfaz virtual más utilizada y se usa en casi todos los **Router**

Interfaces IP sin numerar. Una dirección sin numerar puede tomar prestada una dirección de interfaz Loopback para guardar las direcciones IP de la red.