

Instale el módulo de Python ncclient

```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows [Versión 10.0.19044.2251]
(c) Microsoft Corporation. Todos los derechos reservados.

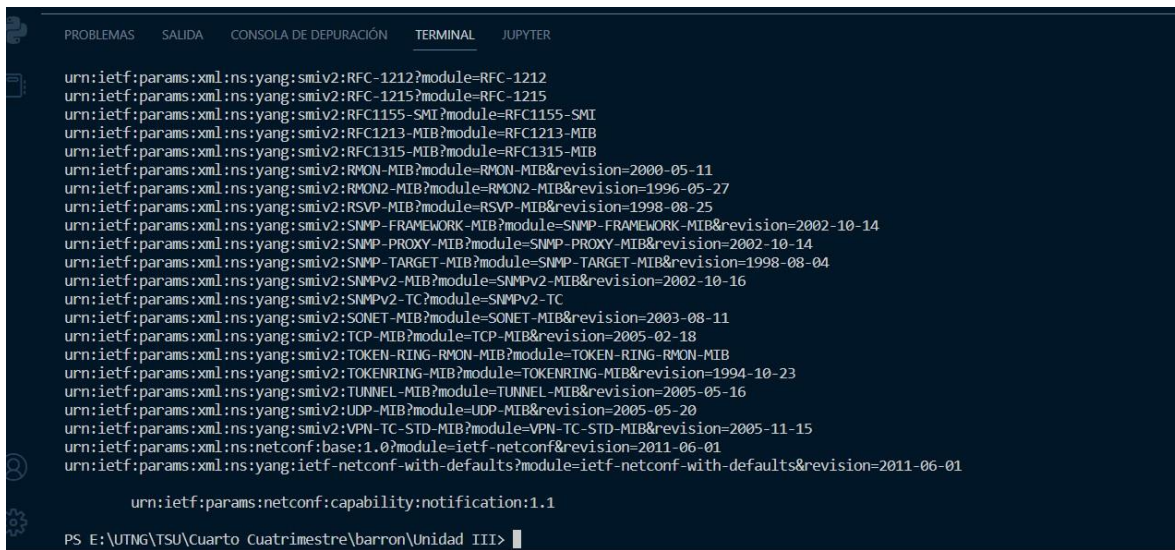
C:\Users\info>pip install ncclient
Requirement already satisfied: ncclient in c:\users\info\appdata\local\programs\python\python310\lib\site-packages (0.6.13)
Requirement already satisfied: setuptools>0.6 in c:\users\info\appdata\local\programs\python\python310\lib\site-packages (from ncclient) (63.2.0)
Requirement already satisfied: paramiko>=1.15.0 in c:\users\info\appdata\local\programs\python\python310\lib\site-packages (from ncclient) (2.11.0)
Requirement already satisfied: lxml>=3.3.0 in c:\users\info\appdata\local\programs\python\python310\lib\site-packages (from ncclient) (4.9.1)
Requirement already satisfied: six in c:\users\info\appdata\local\programs\python\python310\lib\site-packages (from ncclient) (1.16.0)
Requirement already satisfied: cryptography>=2.5 in c:\users\info\appdata\local\programs\python\python310\lib\site-packages (from paramiko>=1.15.0->ncclient) (38.0.1)
Requirement already satisfied: bcrypt>=3.1.3 in c:\users\info\appdata\local\programs\python\python310\lib\site-packages (from paramiko>=1.15.0->ncclient) (4.0.1)
Requirement already satisfied: pynacl>=1.0.1 in c:\users\info\appdata\local\programs\python\python310\lib\site-packages (from paramiko>=1.15.0->ncclient) (1.5.0)
Requirement already satisfied: cffi>=1.12 in c:\users\info\appdata\local\programs\python\python310\lib\site-packages (from cryptography>=2.5->paramiko>=1.15.0->ncclient) (1.15.1)
Requirement already satisfied: pycparser in c:\users\info\appdata\local\programs\python\python310\lib\site-packages (from cffi>=1.12->cryptography>=2.5->paramiko>=1.15.0->ncclient) (2.21)

C:\Users\info>
```

Enumere las capacidades del IOS XE: modelos YANG admitidos

```
lab27.py
1 from ncclient import manager
2
3 m = manager.connect(
4     host="10.10.20.48",
5     port=830,
6     username="developer",
7     password="Cisco12345",
8     hostkey_verify=False
9 )
10
11 print("#Supported Capabilities (YANG models):")
12 for capability in m.server_capabilities:
13     print(capability)
14
15
PROBLEMAS SALIDA CONSOLA DE DEPURACIÓN TERMINAL JUPYTER
urn:ietf:params:xml:ns:yang:smiv2:RFC-1212?module=RFC-1212
urn:ietf:params:xml:ns:yang:smiv2:RFC-1215?module=RFC-1215
urn:ietf:params:xml:ns:yang:smiv2:RFC1155-SMI?module=RFC1155-SMI
urn:ietf:params:xml:ns:yang:smiv2:RFC1213-MIB?module=RFC1213-MIB
urn:ietf:params:xml:ns:yang:smiv2:RFC1315-MIB?module=RFC1315-MIB
urn:ietf:params:xml:ns:yang:smiv2:RMON-MIB?module=RMON-MIB&revision=2000-05-11
urn:ietf:params:xml:ns:yang:smiv2:RMON2-MIB?module=RMON2-MIB&revision=1996-05-27
urn:ietf:params:xml:ns:yang:smiv2:RSVP-MIB?module=RSVP-MIB&revision=1998-08-25
urn:ietf:params:xml:ns:yang:smiv2:SNMP-FRAMEWORK-MIB?module=SNMP-FRAMEWORK-MIB&revision=2002-10-14
urn:ietf:params:xml:ns:yang:smiv2:SNMP-PROXY-MIB?module=SNMP-PROXY-MIB&revision=2002-10-14
urn:ietf:params:xml:ns:yang:smiv2:SNMP-TARGET-MIB?module=SNMP-TARGET-MIB&revision=1998-08-04
urn:ietf:params:xml:ns:yang:smiv2:SNMPv2-MIB?module=SNMPv2-MIB&revision=2002-10-16
urn:ietf:params:xml:ns:yang:smiv2:SNMPv2-TC?module=SNMPv2-TC
urn:ietf:params:xml:ns:yang:smiv2:SOCKET-MIB?module=SOCKET-MIB&revision=2003-08-11
urn:ietf:params:xml:ns:yang:smiv2:TCP-MIB?module=TCP-MIB&revision=2005-02-18
urn:ietf:params:xml:ns:yang:smiv2:TOKEN-RING-RMON-MIB?module=TOKEN-RING-RMON-MIB
urn:ietf:params:xml:ns:yang:smiv2:TOKENRING-MIB?module=TOKENRING-MIB&revision=1994-10-23
urn:ietf:params:xml:ns:yang:smiv2:TUNNEL-MIB?module=TUNNEL-MIB&revision=2005-05-16
urn:ietf:params:xml:ns:yang:smiv2:UDP-MIB?module=UDP-MIB&revision=2005-05-20
urn:ietf:params:xml:ns:yang:smiv2:VPN-TC-STD-MIB?module=VPN-TC-STD-MIB&revision=2005-11-15
urn:ietf:params:xml:ns:yang:ietf-netconf:base:1.0?module=ietf-netconf&revision=2011-06-01
urn:ietf:params:xml:ns:yang:ietf-netconf-with-defaults?module=ietf-netconf-with-defaults&revision=2011-06-01
urn:ietf:params:netconf:capability:notification:1.1
PS E:\UTING\YSU\Cuarto Cuatrimestre\harron\Unidad III>
```

Salida:



```
urn:ietf:params:xml:ns:yang:smiv2:RFC-1212?module=RFC-1212
urn:ietf:params:xml:ns:yang:smiv2:RFC-1215?module=RFC-1215
urn:ietf:params:xml:ns:yang:smiv2:RFC1155-SMI?module=RFC1155-SMI
urn:ietf:params:xml:ns:yang:smiv2:RFC1213-MIB?module=RFC1213-MIB
urn:ietf:params:xml:ns:yang:smiv2:RFC1315-MIB?module=RFC1315-MIB
urn:ietf:params:xml:ns:yang:smiv2:RMON-MIB?module=RMON-MIB&revision=2000-05-11
urn:ietf:params:xml:ns:yang:smiv2:RMON2-MIB?module=RMON2-MIB&revision=1996-05-27
urn:ietf:params:xml:ns:yang:smiv2:RSVP-MIB?module=RSVP-MIB&revision=1998-08-25
urn:ietf:params:xml:ns:yang:smiv2:SNMP-FRAMEWORK-MIB?module=SNMP-FRAMEWORK-MIB&revision=2002-10-14
urn:ietf:params:xml:ns:yang:smiv2:SNMP-PROXY-MIB?module=SNMP-PROXY-MIB&revision=2002-10-14
urn:ietf:params:xml:ns:yang:smiv2:SNMP-TARGET-MIB?module=SNMP-TARGET-MIB&revision=1998-08-04
urn:ietf:params:xml:ns:yang:smiv2:SNMPv2-MIB?module=SNMPv2-MIB&revision=2002-10-16
urn:ietf:params:xml:ns:yang:smiv2:SNMPv2-TC?module=SNMPv2-TC
urn:ietf:params:xml:ns:yang:smiv2:SONET-MIB?module=SONET-MIB&revision=2003-08-11
urn:ietf:params:xml:ns:yang:smiv2:TCP-MIB?module=TCP-MIB&revision=2005-02-18
urn:ietf:params:xml:ns:yang:smiv2:TOKEN-RING-RMON-MIB?module=TOKEN-RING-RMON-MIB
urn:ietf:params:xml:ns:yang:smiv2:TOKENRING-MIB?module=TOKENRING-MIB&revision=1994-10-23
urn:ietf:params:xml:ns:yang:smiv2:TUNNEL-MIB?module=TUNNEL-MIB&revision=2005-05-16
urn:ietf:params:xml:ns:yang:smiv2:UDP-MIB?module=UDP-MIB&revision=2005-05-20
urn:ietf:params:xml:ns:yang:smiv2:VPN-TC-STD-MIB?module=VPN-TC-STD-MIB&revision=2005-11-15
urn:ietf:params:xml:ns:yang:ietf-netconf:base:1.0?module=ietf-netconf&revision=2011-06-01
urn:ietf:params:xml:ns:yang:ietf-netconf-with-defaults?module=ietf-netconf-with-defaults&revision=2011-06-01

urn:ietf:params:netconf:capability:notification:1.1

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

Conclusión: nccliente es una biblioteca de Python para clientes NETCONF. Su objetivo es ofrecer una API intuitiva que mapee sensiblemente la naturaleza codificada por XML de NETCONF a las construcciones y expresiones idiomáticas de Python, y facilitar la escritura de scripts de administración de red.

Otras características clave son:

- Admite todas las operaciones y capacidades definidas en RFC 6241.

- Solicitar canalización.

- Solicitudes de RPC asíncronas.

- Mantener XML fuera del camino a menos que sea realmente necesario.

- Extensible. Se pueden agregar fácilmente nuevas asignaciones de transporte y capacidades / operaciones.

El Protocolo de configuración de red (NETCONF) es un protocolo de gestión de red con base en XML que da un procedimiento programable para configurar y registrar dispositivos de red. NETCONF ha sido determinado en RFC 4741 por el Conjunto laboral de ingeniería de Internet (IETF) y inspeccionado en RFC 6241.

NETCONF da estándares por medio de los cuales los administradores de red y los desarrolladores de aplicaciones tienen la posibilidad de registrar configuraciones de dispositivos de red y obtener el estado del dispositivo de red inmediatamente.

Los paquetes NETCONF permanecen en formato XML y el protocolo NETCONF tiene una poderosa capacidad de filtrado. Cada campo de datos tiene un nombre de factor fijo y una postura. Por consiguiente, los dispositivos del mismo abastecedor tienen la posibilidad de usar el mismo modo de ingreso y modo de visualización de resultados. Los dispositivos de diferentes proveedores tienen la posibilidad de conseguir el mismo impacto por medio del mapeo XML. Esta funcionalidad permite el desarrollo de programa de terceros y la personalización del programa NMS en el ámbito de diversas proveedores y dispositivos. Gracias a dicho programa NMS, NETCONF simplifica la configuración del dispositivo y optimización la eficiencia de la configuración del dispositivo.