



**Facultad de:** *Infraestructura de Redes Digitales.*

**Nombre del Alumno(a):**

*Jessica Quetzali Castillo Avilés*

**Matrícula:** *1221100373*

**Materia:**

*PROGRAMACIÓN DE REDES.*

**Nombre de la Actividad:**

*Laboratorio 2.5*

**Docente:**

*Gabriel Barrón Rodríguez*

*Dolores Hidalgo C.I.N.; a 07 de Diciembre 2022*



Universidad Tecnológica  
del Norte de Guanajuato  
Organismo Público Descentralizado del Gobierno del Estado

## Lab 2.5 – RESTCONF with Python.

The screenshot displays a Windows desktop environment during a lab session. In the background, a WordPad window titled "2.5 Lab - RESTCONF with Python" is open. In the foreground, an IDLE Shell 3.9.1 window shows the execution of a Python script. The script defines a RESTCONF client and attempts to connect to a Cisco IOS XE device. The output shows a successful connection and a list of enabled programmability features.

```
Python 3.9.1 (tags/v3.9.1:1e5d33e, Dec 7 2020, 17:08:21) [MSC v.1927 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:/Users/jecql/Documents/2.5.py =====
Traceback (most recent call last):
  File "C:/Users/jecql/Documents/2.5.py", line 2, in <module>
    import requests
ModuleNotFoundError: No module named 'requests'
>>>
===== RESTART: C:/Users/jecql/Documents/2.5.py =====
{'ietf-interfaces:interfaces': {'interface': [{'name': 'GigabitEthernet1', 'description': 'MANAGEMENT INTERFACE - DON'T TOUCH ME', 'type': 'iana-if-type:ethernetCsmacd', 'enabled': True, 'ietf-ip:ipv4': {'address': [{'ip': '10.10.20.48', 'netmask': '255.255.255.0'}]}, 'ietf-ip:ipv6': {}}, {'name': 'GigabitEthernet2', 'description': 'Network Interface', 'type': 'iana-if-type:ethernetCsmacd', 'enabled': False, 'ietf-ip:ipv4': {}, 'ietf-ip:ipv6': {}}, {'name': 'GigabitEthernet3', 'description': 'Network Interface', 'type': 'iana-if-type:ethernetCsmacd', 'enabled': False, 'ietf-ip:ipv4': {}, 'ietf-ip:ipv6': {}}, {'name': 'Loopback1', 'description': 'LABORATORIO 2.2', 'type': 'iana-if-type:softwareLoopback', 'enabled': True, 'ietf-ip:ipv4': {'address': [{'ip': '1.1.1.1', 'netmask': '255.255.255.0'}]}, 'ietf-ip:ipv6': {}}, {'name': 'Loopback99', 'description': 'LABORATORIO 2.2 99', 'type': 'iana-if-type:softwareLoopback', 'enabled': True, 'ietf-ip:ipv4': {'address': [{'ip': '99.99.99.99', 'netmask': '255.255.255.0'}]}, 'ietf-ip:ipv6': {}}]}}
>>>
```

A PuTTY terminal window is also open, showing the login process for a Cisco CSR1000v. The terminal output includes the following table:

Interface	IP-Address	OK?	Method	Status	Protocol
GigabitEthernet1	10.10.20.48	YES	NVRAM	up	up
GigabitEthernet2	unassigned	YES	NVRAM	administratively down	down
GigabitEthernet3	unassigned	YES	NVRAM	administratively down	down
Loopback1	1.1.1.1	YES	manual	up	up
Loopback98	98.98.98.98	YES	other	up	up
Loopback99	99.99.99.99	YES	other	up	up

The terminal also shows the command `ietf-ip:ipv4` and its output, which is a JSON object representing the IPv4 configuration for the interface.



## CONCLUSIONES:

Se utilizaron las variables creadas como parámetros para el método `request.get()`. Este método envía una solicitud HTTP GET a la API RESTCONF. Además se le asignó el resultado de la solicitud a un nombre de variable `resp`. Esa variable contendrá la respuesta JSON de la API. Si la solicitud tiene éxito, el JSON contendrá el modelo de datos YANG devuelto.

### Código Descripción Humana

201	Creado OK, respuesta correcta frente a un POST.
204	Modificado OK, respuesta correcta frente a un PUT
400	Error en la petición, suele ocurrir frente a URIs mal armadas o con campos incorrectos en el cuerpo del mensaje.
401	Error en las credenciales.
404	No se encuentra el recurso. Posible problema en la URI
409	Conflicto en la creación. En un POST, significa que ya existe el recurso (se debería modificar con PUT).

Estos son algunos de los códigos respondidos en una request, puede darte cualquiera de estas, dependiendo a cada situación, si hay un error de escritura, si no se encuentra, si se encontró exitosamente.

`encabezados` = un parámetro que se asigna a la variable de encabezados

`verificar = falso`= deshabilita la verificación del certificado SSL cuando se realiza la solicitud

`resp.status_code`= El código de estado HTTP en la respuesta de solicitud de API

Además de poder crear exitosamente con postman, también se puede en una terminal de python, además podemos definir un ciclo que nos diga si la rest fue respondida con éxito o por el contrario no.