

Área académica: tecnologías de la información y comunicación

Programa educativo: Tecnologías de la información área infraestructura de redes digitales

Docente: Gabriel Barrón Rodríguez

Alumno:

Bueno González Felipe Neri Francisco

Numero de control: 1222100410

Actividad 1



a. **¿Qué es la programabilidad basada en modelos? ¿Por qué se desarrolló?** es un enfoque de diseño que permite a los usuarios crear y modificar fácilmente sistemas y dispositivos de red. Se desarrolló para mejorar la eficiencia y la interoperabilidad de los sistemas de red, facilitando la implementación de nuevas tecnologías y soluciones.

b. **¿Qué propósito proporcionan los nuevos protocolos y estándares de YANG, NETCONF y RESTCONF?** una forma estandarizada de interactuar con dispositivos de red, permitiendo una mayor automatización y simplificación de las operaciones de configuración y administración de redes.

c. **¿SNMP va a desaparecer?** no desaparecerá, pero su función se ha ampliado y se ha integrado con otros sistemas de gestión de redes, como YANG, NETCONF y RESTCONF.

d. **¿Qué tiene que ver YANG con esto?** es facilitar la automatización y la gestión de dispositivos de red, permitiendo a los usuarios interactuar de manera eficiente y efectiva con ellos

e. **Diferencias entre el lenguaje YANG, los modelos de datos YANG y los datos YANG.**

- Lenguaje YANG: Es un lenguaje utilizado para describir modelos de datos de dispositivos de red.
- Modelos de datos YANG: Son estructuras de datos que definen los atributos y relaciones entre ellos en un dispositivo de red.
- Datos YANG: Son instancias específicas de los modelos de datos YANG que representan la configuración y estado de un dispositivo de red.

f. **¿Qué es un modelo de datos?** es una representación formal de la estructura y relaciones de los datos en un sistema; pero para YANG un modelo de datos define cómo se organizan y relacionan los atributos de un dispositivo de red

g. **¿Qué es OpenConfig?** es un proyecto de la IETF que define un lenguaje de configuración abierta para dispositivos de red. Su objetivo es permitir una mayor interoperabilidad y automatización en la configuración y administración de dispositivos de red, reduciendo la dependencia de protocolos específicos de dispositivo.

Evidenciar configuración de YANG (no se te olvide evidencia con perfil de alumnado).

The screenshot shows a web browser window with the following elements:

- Browser Tabs:** DevNet Lab 13 Instruments, Unidad Temática 11 Online, Exploring YANG Models with..., WhatsApp.
- Address Bar:** developer.cisco.com/learning/tracks/EN-Networking-v2/intro-device-level/interfaces/intro-yang/exploring-yang-models-with-pyng/
- Page Header:** DevNet | Learning Lab Center
- Left Sidebar:** A list of navigation links including 'Introduction to Model Driven Programmability (ex. NETCONF/YANG)', 'Module Overview', 'What and Why of Model Driven Programmability', 'Introducing YANG Data Modeling for the Network', 'Introduction', 'Get Up With YANG Code Examples', 'What is YANG?', 'Working with YANG Data Models', 'Exploring YANG Models with pyng', 'Exploring Network Device Data in YANG', 'Summary', 'Exploring YANG Models with NETCONF', and 'Exploring YANG Models with RESTCONF'.
- Main Content Area:**
 - Section 7: The `pyng` module can generate representations of a YANG model. Run the following command to generate a clear-text tree view of the model.
 - Code Snippet:

```
cd ~/src/dev-ops-code/intro-yang/models  
pyng -f tree self-interfaces.yang
```
 - Text: Expand to see the expected response.
 - Section 8: Some things to notice about the output.
 - Bulleted List:
 - The module `self-interfaces` provides two "containers":
 - `interfaces`
 - `interfaces-state`
 - Within each "container", there is a "list" called "interface".
 - You can identify a single instance of an interface by a unique "key" of [name]
 - Every "list" attribute (for example, name, description, type) has the following details:
 - Either read-write (`rw`) or read-only (`ro`).
 - Some are optional (?).
 - Explicitly defined data types.
 - Section 9: Included OpenConfig YANG model for "interfaces" in the `models` directory. Look at it and compare what you see to the IETF version.
 - Code Snippet:

```
cd ~/src/dev-ops-code/intro-yang/models  
pyng -f tree openconfig-self-interfaces.yang
```
- Right Sidebar:** A user profile for 'neri bueno' with a 'Dashboard' link and a 'Logout' button.
- Terminal:** A terminal window showing the output of the `pyng` command, displaying a tree structure of YANG model data.

Evidenciar Exploración con Modelos YANG

The screenshot shows a web browser window with the URL <https://developer.cisco.com/learning/tracks/EN-Networking-v0/intro-device-level/interfaces/intro-yang/exploring-yang-models-with-pyang/>. The page is part of the Cisco DevNet Learning Labs Center. The main content area is titled 'Introducing YANG' and contains the following text:

7. The `pyang` module can generate representations of a YANG model. Run the following command to generate a clear-text tree view of the model:

```
cd ~/src/dev-data-code/intro-mdp/yang/models
pyang -t tree ietf-interfaces.yang
```

Expand to see the expected response

8. Some things to notice about the output:

- The module `ietf-interfaces` provides two "containers":
 - `interfaces`
 - `interfaces-state`
- Within each "container", there is a "list" called "interface".
- You can identify a single instance of an interface by a unique "key" of [name].
- Every "leaf" attribute (for example, name, description, type) has the following details:
 - Either read-write (`rw`) or read-only (`ro`).
 - Some are optional (?).
 - Explicitly defined data types.

9. Included OpenConfig YANG model for "interfaces" in the `models\` directory. Look at it and compare what you see to the IETF version.

```
cd ~/src/dev-data-code/intro-mdp/yang/models
pyang -t tree openconfig-interfaces.yang
```

The terminal window on the right shows the output of the `pyang` command, displaying a tree view of the YANG model structure.

Evidenciar Exploración Modelos de Datos YANG con NETCONF

The screenshot shows a web browser window with the URL <https://developer.cisco.com/learning/tracks/EN-Networking-v0/intro-device-level/interfaces/intro-netconf/walking-through-automating-your-network-with-netconf/>. The page is part of the Cisco DevNet Learning Labs Center. The main content area is titled 'Exploring YANG Models with NETCONF' and contains the following text:

After you make the network configuration changes, save them to the startup configuration. NETCONF supports the ability for vendors to create native RPC operations for activities that are specific to their platform. IOS XE devices support a `save-config` operation as part of one of the native data models. Let's see how it works.

- Open the file `save_config.py` in a text editor. Let's walk through the new parts to observe.
- To craft a custom RPC, you are using another object from the `ncclient` package. Here you import `ncx`.
- Rather than creating a NETCONF `session` or `connection`, this time you are explicitly calling the RPC from a model.
- As you are sending a custom RPC, use the `dispatch` method to send the custom operation.

5. Close the script and run it from the terminal. Look at the body of the reply returned. You should see a clear indication of success.

```
cd ~/src/dev-data-code/intro-netconf
python save_config.py
```

The terminal window on the right shows the output of the `python save_config.py` command, displaying the result of the `save-config` operation.

Evidenciar Exploración Modelos de Datos YANG con RESTCONF

The screenshot shows a web browser window with the URL <https://developer.cisco.com/learning/tracks/EN-Networking-v0/intro-device-level/interfaces/intro-restconf/step-1-breaking-down-restconf-communications/>. The page is part of the Cisco DevNet Learning Labs Center. The main content area is titled 'Exploring YANG Models with RESTCONF' and contains the following text:

So, using model details, you can construct the URLs for RESTCONF requests.

List of Interfaces

`https://<ADDRESS>/restconf/data/ietf-interfaces:interfaces`

► Example curl command

Note: Curl is a command-line tool for transferring data specified with URL syntax.

A Specific Interface

`http://<ADDRESS>/restconf/data/ietf-interfaces:interfaces/interface=INTERFACE_NAME`

► Example curl command

Just the name and description fields

`http://<ADDRESS>/restconf/data/ietf-interfaces:interfaces/interface=INTERFACE_NAME?fields=name&fields=description`

► Example curl command

The terminal window on the right shows the output of the `curl` command, displaying the result of the RESTCONF request.