



## Práctica de laboratorio 10: Explorar los modelos YANG

### Versión en inglés:

<https://itexamanswers.net/8-3-5-lab-explore-yang-models-answers.html>

## Objetivos

**Parte 1: Iniciar la máquina virtual de DEVASC.**

**Parte 2: Explorar un modelo de YANG en GitHub.**

**Parte 3: Explorar un modelo YANG usando pyang.**

## Aspectos básicos/Situación

Los modelos YANG definen la estructura exacta, los tipos de datos, la sintaxis y las reglas de validación para el contenido de los mensajes intercambiados entre un dispositivo gestionado y otro sistema que se está comunicando con el dispositivo. Trabajar con archivos usando el lenguaje YANG puede ser un poco abrumador para el nivel de detalles de estos archivos.

En este laboratorio, aprenderá cómo utilizar la herramienta **pyang** de código abierto, para transformar los modelos de datos de YANG a partir de archivos, utilizando el lenguaje YANG en un formato mucho más fácil de leer. Usando la transformación de vista “árbol”, identificará cuáles son los elementos clave del modelo YANG de interfaces ietf.

## Recursos necesarios

- Una computadora con el sistema operativo de su elección.
- VirtualBox o VMware.
- Máquina virtual DEVASC.

## Instrucciones

### Parte 1: Iniciar la máquina virtual de DEVASC.

Si aún no ha completado el **laboratorio - Instalar el DEVASC-LAB**, hágalo ahora. Si se ha completado ya, inicie la máquina virtual DEVASC.

### Parte 2: Explorar un modelo de YANG en GitHub.

En esta parte, instalará el módulo pyang en su máquina virtual DEVASC y explorará como transforma los archivos YANG. Pyang simplifica el trabajo con archivos YANG. El módulo viene con una línea ejecutable de comandos pyang que transforma los archivos YANG en un formato más legible para el ser humano.

### Paso 1: Explorar los modelos de Cisco IOS XE YANG en el repositorio de GitHub.

- a. Abra Chromium y busque <https://github.com/YangModels/yang>.

- b. En la rama **maestra**, vaya a los modelos de YANG para Cisco IOS XE (versión 16.9.3), haciendo clic en los siguientes directorios: **proveedor > cisco > xe > 1693**.
- c. Desplácese por debajo de todos los modelos Cisco YANG y encuentre dónde comienzan los modelos IETF. Busque **ietf-interfaces.yang**.
- d. Haga clic en **ietf-interfaces.yang** y desplácese por todos los nodos de contenedores, nodos de hoja y nodos de lista. Si está familiarizado con la salida del comando Mostrar interfaces IOS (show interfaces IOS), entonces debe reconocer algunos o todos los nodos. Por ejemplo, alrededor de la línea 221 verá la hoja habilitada.

```
hoja habilitada {
  Tipo boolean;
  por defecto «true»;
  descripción
    «Esta hoja contiene el estado configurado y deseado del
    E&M.
    Los sistemas que implementan el IF-MIB utilizan el valor de este
    en el almacén de datos 'en ejecución' para establecer
    if-mib.ifAdminStatus a 'arriba' o 'abajo' después de una ifEntry
    se ha iniciado, como se describe en RFC 2863.
    Los cambios en esta hoja en el almacén de datos 'en ejecución' son
    reflejado en ifAdminStatus, pero si ifAdminStatus es
    cambiado sobre SNMP, esta hoja no se va a ver afectada.»;
  referencia
    «RFC 2863: MIB del Grupo de Interfaces - IFAdminStatus»;
}
```

### Paso 2: Copiar el modelo ietf-interfaces.yang en una carpeta de su Máquina Virtual.

- a. Abra VS Code.
  - b. Haga clic en **Archivo > Abrir carpeta...** y navegue al directorio **devnet-src**.
  - c. Haga clic en **Aceptar**.
  - d. Abra una ventana de terminal en VS Código: **Terminal > Nueva Terminal**.
  - e. Cree un subdirectorio llamado **pyang** en el directorio **/devnet-src**.
- ```
devasc @labvm: ~/labs/devnet-src$ mkdir pyang
devasc@labvm:~/labs/devnet-src$
```
- f. Vuelva a la pestaña de Chromium donde el modelo **ietf-interfaces.yang** sigue abierto. Desplácese hacia atrás, a la parte superior, si es necesario y haga clic en Raw para mostrar solo los datos del modelo YANG.
  - g. Seleccione y copie la dirección URL.
  - h. En la terminal, vaya a la carpeta pyang.
  - i. Utilice **wget** para guardar el archivo ietf-interfaces.yang sin procesar.

```
devasc @labvm: ~/labs/devnet-src/pyang$ wget
https://raw.githubusercontent.com/YangModels/yang/master/vendor/cisco/xe/1693
/ietf-interfaces.yang
--2020-06-22 20:42:20--
https://raw.githubusercontent.com/YangModels/yang/master/vendor/cisco/xe/1693/ietf-
interfaces.yang
```

```
Resolviendo raw.githubusercontent.com (raw.githubusercontent.com)... 151.101.0.133,
151.101.192.133, 151.101.128.133, ...
Conectando a raw.githubusercontent.com (raw.githubusercontent.com)
|151.101.0.133|:443... conectado.
HTTP request sent, awaiting response... 200 OK
Longitud: 24248 (24K) [texto/liso]
Guardar en: 'ietf-interfaces.yang'

ietf-interface 100% 23.68K --.-Kb/s en 0.05s

2020-06-22 20:42:21 (439 kB/s) - 'ietf-interfaces.yang' guardado [24248/24248]

devasc @labvm: ~/labs/devnet-src/pyang$
Ahora tiene una versión local del modelo ietf-interfaces.yang que puede manipular con pyang.
```

### Parte 3: Explorar un modelo de YANG usando pyang.

En esta parte, usted instalará el módulo **pyang** en su máquina virtual DEVASC y explorará cómo transforma el modelo YANG que copió de GitHub. Pyang simplifica el trabajo con archivos YANG. El módulo viene con un ejecutable de línea de comandos **pyang** que transforma los archivos YANG en un formato más legible por humanos.

#### Paso 1: Verificar que pyang esté instalado y actualizado.

- En un código VS, abra la ventana terminal
- Verifique que pyang ya esté instalado con el comando **pyang -v**. Su número de versión puede ser diferente al que se muestra aquí. También se puede

```
devasc @labvm: ~/labs/devnet-src$ pyang -v
pyang 2.2.1
devasc@labvm:~/labs/devnet-src$
```
- (Opcional) Usted puede verificar que tiene las últimas actualizaciones de pyang usando el siguiente comando **pip3**. Cualquier actualización después de escribir este laboratorio se descargará y se instalará.

```
devasc @labvm: ~/labs/devnet-src$ pip3 install pyang --upgrade
Requisito ya actualizado: pyang en ./local/lib/python3.8/site-packages (2.2.1)
Requisito ya satisfecho, omitiendo la actualización: lxml en
./local/lib/python3.8/site-packages (de pyang) (4.5.0)
devasc@labvm:~/labs/devnet-src$
```

#### Paso 2: Transformar el modelo ietf-interfaces.yang.

- Vaya al directorio **pyang**.

```
devasc@labvm:~/labs/devnet-src$ cd pyang
devasc@labvm:~/labs/devnet-src/pyang$
```
- Ingrese **pyang -h** | para explorar más las opciones para transformar el modelo YANG. Busque la opción **-f** como se muestra a continuación. Usará la opción de formato de **árbol**.

```
devasc@labvm:~/labs/devnet-src/pyang$ pyang -h | más
Uso: pyang [opciones] [<filename>...]
```

Valida el módulo YANG en <filename> (o stdin), y todas sus dependencias.

Options

-h, -help Mostrar este mensaje de ayuda y salir  
-v, -version Mostrar el número de versión y salir

<output omitted>

**-f FORMAT**, -format=Formato

Convertir a FORMAT. Los formatos admitidos son: yang, yin, dsdl, jstree, jsonxsl, capacidad, identificadores, jtox, UML, nombre, Omni, **árbol**, depende, sample-xml-esqueleto

<output omitted>

devasc@labvm:~/labs/devnet-src/pyang\$

- c. Transforme el modelo **ietf-interfaces.yang** en un modelo de formato de árbol con el siguiente comando. Observe que la **hoja habilitada** es mucho más fácil de encontrar y leer en este formato.

devasc@labvm:~/labs/devnet-src/pyang\$ **pyang -f árbol ietf-interfaces.yang**

ietf-interfaces.yang:6: error: módulo «ietf-yang-types» no fue encontrado en la ruta de búsqueda

módulo: ietf-interfaces

```
interfaces +-rw
| +--rw interface* [Nombre]
| +-cadena de nombre rw
| +--rw descripción? cadena
| +--rw tipo identifiref
| +--rw habilitado? booleano
| +-rw link-up-down-trap-enable? enumeración {if-mib}?
+-ro interfaces-estado
  +--ro interface* [Nombre]
    +-ro cadena de nombre
    +--ro tipo identifiref
    +-ro admin-estado enumeración {if-mib}?
    +-ro enumeración de oper-estado
    +-ro último cambio? yang: fecha y hora
    +-ro if-index int32 {if-mib}?
    +-ro phys-dirección? yang:phys-dirección
    +-ro capa superior-si* interface-state ref
    +-ro capa inferior-si* interface-state ref
    ¿Velocidad de +ro? yang:gauge64
    +--ro estadísticas
      +-ro tiempo de discontinuidad yang: fecha y hora
      +-ro en octetos? yang:counter64
      +-ro in-unicast-pkts? yang:counter64
      +-ro in-broadcast pkts? yang:counter64
      +-ro en multidifusión pkts? yang:counter64
      +-ro en descartes? yang:counter32
      +-ro en errores? yang:counter32
      +-ro in-unknown-protos? yang:counter32
      +-ro octetos de salida? yang:counter64
      +-ro fuera-unicast-pkts? yang:counter64
      +-ro fuera-broadcast-pkts? yang:counter64
      +-ro fuera-multicast-pkts? yang:counter64
      +-ro fuera-descartes? yang:counter32
```

```
+-ro fuera-errores? yang:counter32
devasc@labvm:~/labs/devnet-src/pyang$
```