

Instalar el módulo pyang de Python y verificar si se ha instalado

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

C:\Users\info>pip install --no-binary pyang pyang
DEPRECATION: --no-binary currently disables reading from the cache of locally built wheels. In the future --no-binary will not influence the wheel cache. pip 23.1 will enforce this behaviour change. A possible replacement is to use the --no-cache-dir option. You can use the flag --use-feature=no-binary-enable-wheel-cache to test the upcoming behaviour. Discussion can be found at https://github.com/pypa/pip/issues/11453
Requirement already satisfied: pyang in c:\users\info\appdata\local\programs\python\python310\lib\site-packages (2.5.3)
Requirement already satisfied: lxml in c:\users\info\appdata\local\programs\python\python310\lib\site-packages (from pyang) (4.9.1)

C:\Users\info>pyang -v
pyang 2.5.3

C:\Users\info>
```

Explore los modelos YANG en el repositorio GitHub de YangModels/yang.

The image shows a GitHub web interface for the repository 'YangModels/yang'. The breadcrumb navigation path is 'yang / vendor / cisco / xe / 1693 / ietf-interfaces-ext.yang', with the entire path highlighted by a red circle. Below this, a commit by 'apoorvashty' is shown with the message 'Added IOS XE 16.9.3 yang models and fixed IOS XE 16.10.1 capabilities.' The code editor displays the start of a YANG model file, with line numbers 1 through 17 visible. The code defines a module 'ietf-interfaces-ext' with a namespace, prefixes, and imports.

Utilice la herramienta de línea de comandos pyang para transformar los modelos YANG

```
C:\WINDOWS\system32\cmd.exe
C:\Users\info\Documents\practica>pyang -f tree ietf-interfaces.yang.txt
module: ietf-interfaces
+--rw interfaces
|   +--rw interface* [name]
|   |   +--rw name                string
|   |   +--rw description?        string
|   |   +--rw type                identityref
|   |   +--rw enabled?            boolean
|   |   +--rw link-up-down-trap-enable? enumeration {if-mib}?
+--ro interfaces-state
|   +--ro interface* [name]
|   |   +--ro name                string
|   |   +--ro type                identityref
|   |   +--ro admin-status        enumeration {if-mib}?
|   |   +--ro oper-status         enumeration
|   |   +--ro last-change?        yang:date-and-time
|   |   +--ro if-index            int32 {if-mib}?
|   |   +--ro phys-address?       yang:phys-address
|   |   +--ro higher-layer-if*    interface-state-ref
|   |   +--ro lower-layer-if*    interface-state-ref
|   |   +--ro speed?              yang:gauge64
|   +--ro statistics
|   |   +--ro discontinuity-time  yang:date-and-time
|   |   +--ro in-octets?          yang:counter64
|   |   +--ro in-unicast-pkts?    yang:counter64
|   |   +--ro in-broadcast-pkts?  yang:counter64
|   |   +--ro in-multicast-pkts?  yang:counter64
|   |   +--ro in-discards?        yang:counter32
|   |   +--ro in-errors?          yang:counter32
|   |   +--ro in-unknown-protos?  yang:counter32
|   |   +--ro out-octets?         yang:counter64
|   |   +--ro out-unicast-pkts?   yang:counter64
|   |   +--ro out-broadcast-pkts? yang:counter64
|   |   +--ro out-multicast-pkts? yang:counter64
|   |   +--ro out-discards?       yang:counter32
|   |   +--ro out-errors?         yang:counter32
C:\Users\info\Documents\practica>
```

Conclusiones: los modelos yang son un lenguaje que sirve para crear solicitudes de configuracion dispositivos remotos, esta escrito en un sistema que pueda ser interpretado por los humanos. Con Python se utiliza para crear un scrip para enviar configuraciones a dispositivos remotos, como se hara en los siguientes laboratorios.