

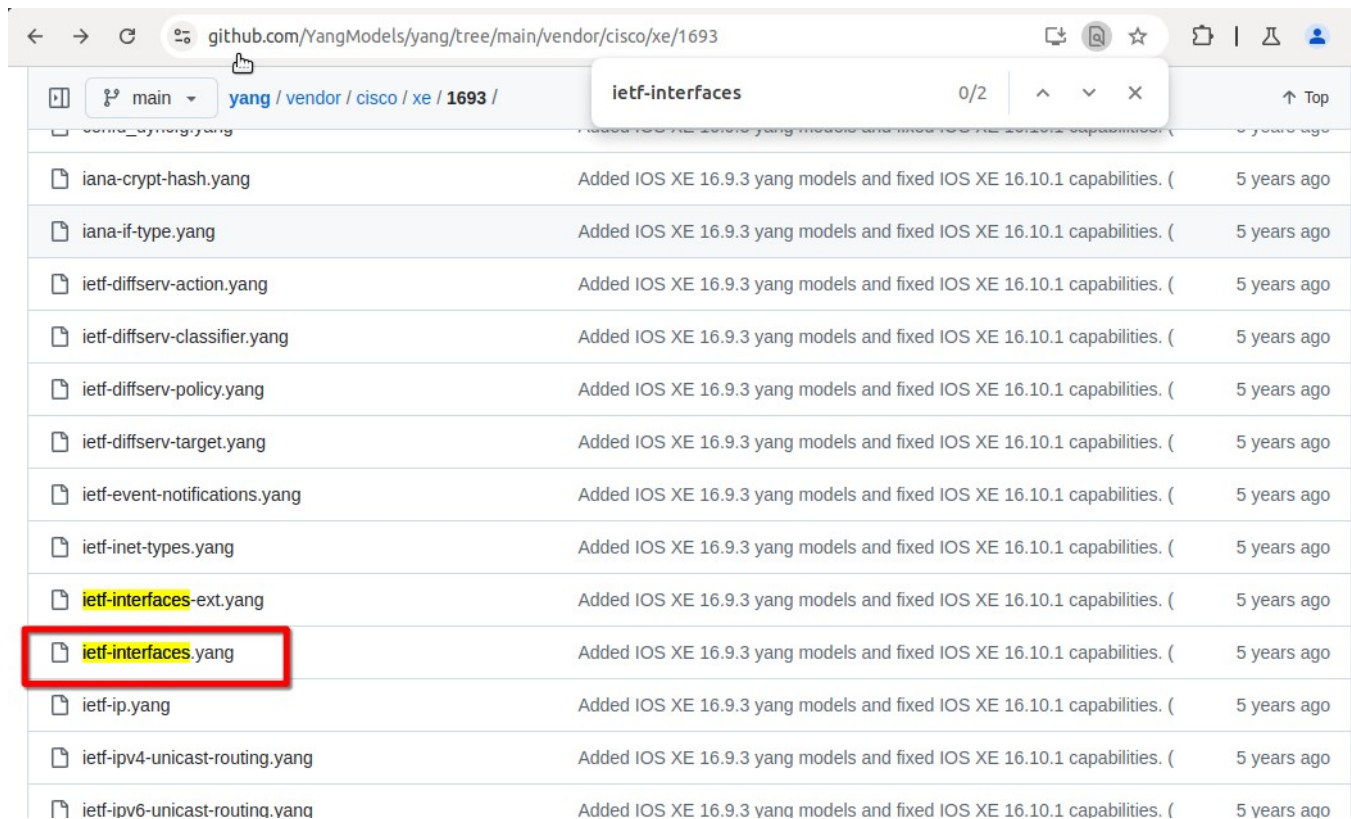
Laboratorio 10: Exploramos YANG Models.

Sergio Sebastian Pezo Jimenez - 20224087G

Inicializamos la DEVASC VM.

Exploramos un modelo YANG en GitHub.

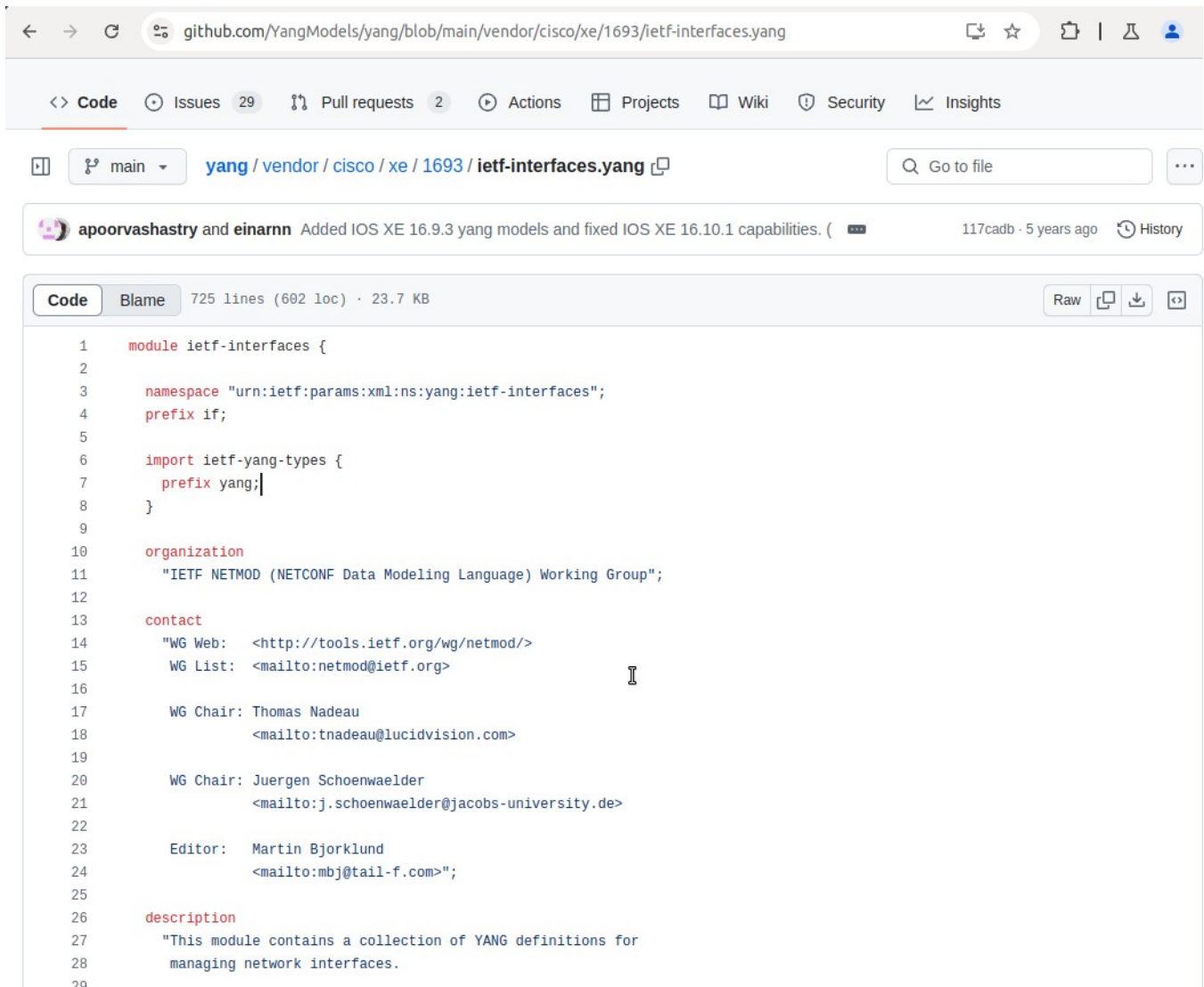
Navegamos en el repositorio <https://github.com/YangModels/yang>, hasta encontrar nuestro archivo `.yang` de interés



The screenshot shows a web browser displaying the GitHub repository `github.com/YangModels/yang/tree/main/vendor/cisco/xe/1693`. The breadcrumb navigation shows the path `yang / vendor / cisco / xe / 1693 /`. A search bar at the top right contains the text `ietf-interfaces` and shows `0/2` results. Below the search bar, a list of files is displayed. The file `ietf-interfaces.yang` is highlighted with a red rectangular box. The list of files includes:

File Name	Description	Time
<code>iana-crypt-hash.yang</code>	Added IOS XE 16.9.3 yang models and fixed IOS XE 16.10.1 capabilities. (5 years ago
<code>iana-if-type.yang</code>	Added IOS XE 16.9.3 yang models and fixed IOS XE 16.10.1 capabilities. (5 years ago
<code>ietf-diffserv-action.yang</code>	Added IOS XE 16.9.3 yang models and fixed IOS XE 16.10.1 capabilities. (5 years ago
<code>ietf-diffserv-classifier.yang</code>	Added IOS XE 16.9.3 yang models and fixed IOS XE 16.10.1 capabilities. (5 years ago
<code>ietf-diffserv-policy.yang</code>	Added IOS XE 16.9.3 yang models and fixed IOS XE 16.10.1 capabilities. (5 years ago
<code>ietf-diffserv-target.yang</code>	Added IOS XE 16.9.3 yang models and fixed IOS XE 16.10.1 capabilities. (5 years ago
<code>ietf-event-notifications.yang</code>	Added IOS XE 16.9.3 yang models and fixed IOS XE 16.10.1 capabilities. (5 years ago
<code>ietf-inet-types.yang</code>	Added IOS XE 16.9.3 yang models and fixed IOS XE 16.10.1 capabilities. (5 years ago
<code>ietf-interfaces-ext.yang</code>	Added IOS XE 16.9.3 yang models and fixed IOS XE 16.10.1 capabilities. (5 years ago
<code>ietf-interfaces.yang</code>	Added IOS XE 16.9.3 yang models and fixed IOS XE 16.10.1 capabilities. (5 years ago
<code>ietf-ip.yang</code>	Added IOS XE 16.9.3 yang models and fixed IOS XE 16.10.1 capabilities. (5 years ago
<code>ietf-ipv4-unicast-routing.yang</code>	Added IOS XE 16.9.3 yang models and fixed IOS XE 16.10.1 capabilities. (5 years ago
<code>ietf-ipv6-unicast-routing.yang</code>	Added IOS XE 16.9.3 yang models and fixed IOS XE 16.10.1 capabilities. (5 years ago

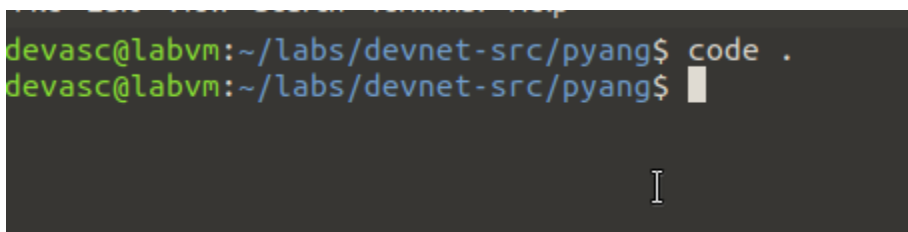
Archivo que define un modelo de datos para la gestión de interfaces de red, basado en la especificación del IETF.



The screenshot shows a GitHub repository page for the file `ietf-interfaces.yang` located at `github.com/YangModels/yang/blob/main/vendor/cisco/xe/1693/ietf-interfaces.yang`. The repository is named `yang` and the file is part of the `vendor/cisco/xe/1693` directory. The file is 725 lines long (602 loc) and 23.7 KB in size. The code is displayed in a light blue theme with line numbers on the left. The code defines a YANG module for managing network interfaces, including namespace, prefix, import, organization, contact, and description sections.

```
1 module ietf-interfaces {
2
3     namespace "urn:ietf:params:xml:ns:yang:ietf-interfaces";
4     prefix if;
5
6     import ietf-yang-types {
7         prefix yang;
8     }
9
10    organization
11        "IETF NETMOD (NETCONF Data Modeling Language) Working Group";
12
13    contact
14        "WG Web: <http://tools.ietf.org/wg/netmod/>
15         WG List: <mailto:netmod@ietf.org>
16
17         WG Chair: Thomas Nadeau
18                 <mailto:tnadeau@lucidvision.com>
19
20         WG Chair: Juergen Schoenwaelder
21                 <mailto:j.schoenwaelder@jacobs-university.de>
22
23         Editor: Martin Bjorklund
24               <mailto:mbj@tail-f.com>";
25
26    description
27        "This module contains a collection of YANG definitions for
28         managing network interfaces.
29
```

Copiamos el modelo a una carpeta `pyang` en nuestra máquina virtual, así abrirlo en VS Code.

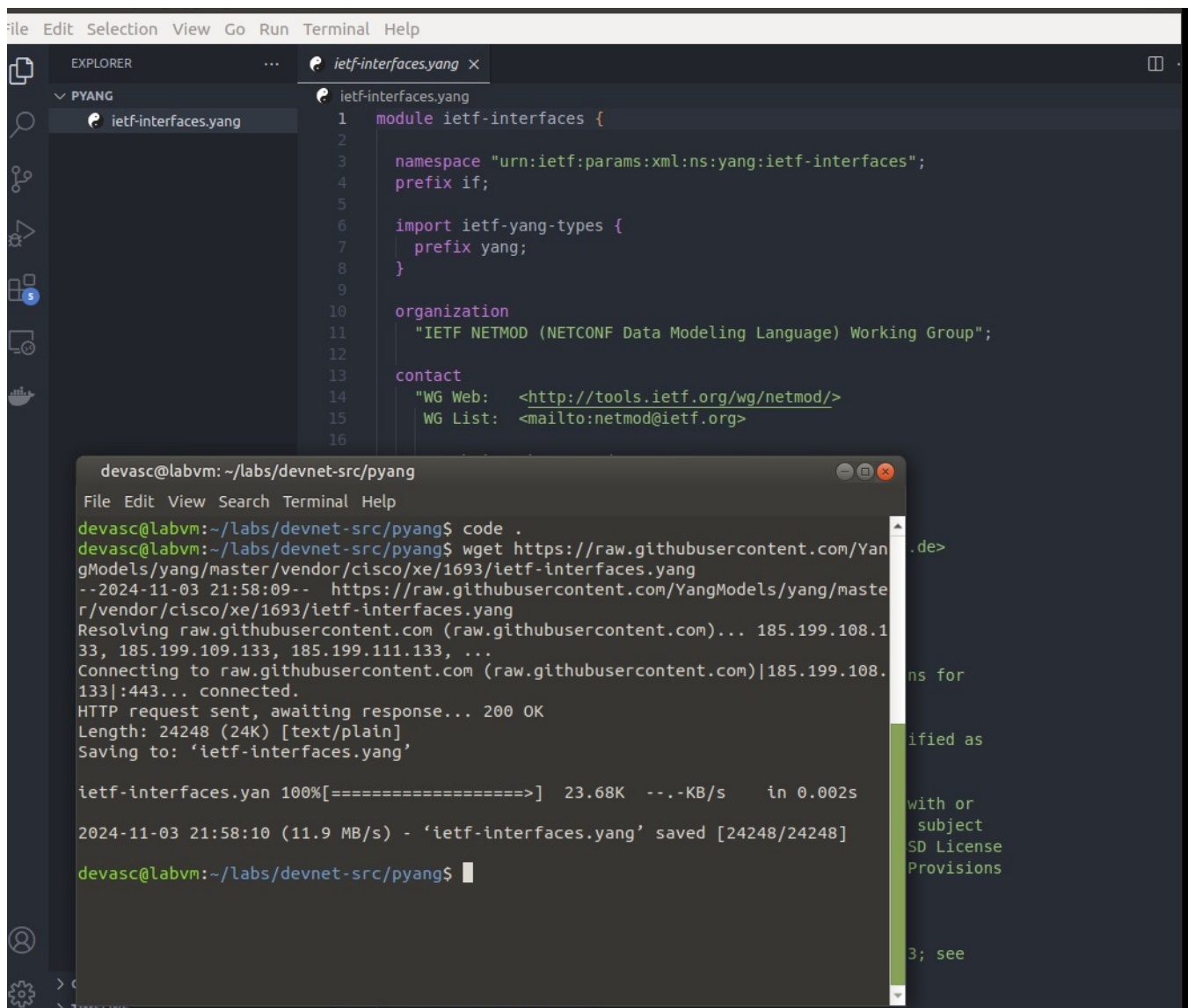


The screenshot shows a terminal window with the following commands and output:

```
devasc@labvm:~/labs/devnet-src/pyang$ code .
devasc@labvm:~/labs/devnet-src/pyang$
```

Y copiamos el archivo `.yang` en nuestra carpeta con `wget` en la url con nuestro archivo de interés

`https://raw.githubusercontent.com/YangModels/yang/master/vendor/cisco/xe/1693/ietf-interfaces.yang`, además descargamos una extensión para visualizar la sintaxis.



Exploramos un modelo YANG usando pyyang.

Una vez tenemos nuestro archivo yang en nuestro dispositivo, Trabajemos con tal usando `pyang`.

Así que primero verificamos si la tenemos ya instalada.

```
devasc@labvm:~/labs/devnet-src/pyang$ pyang -v
pyang 2.2.1
```

Ahora exploraremos las opciones para la transformación del modelo YANG.

```
devasc@labvm:~/labs/devnet-src/pyang$ pyang -h | more
Usage: pyang [options] [<filename>...]

Validates the YANG module in <filename> (or stdin), and all its dependencies.

Options:
-h, --help                Show this help message and exit
-v, --version             Show version number and exit
-V, --verbose
-e, --list-errors         Print a listing of all error and warning codes and
                           exit.
--print-error-code        On errors, print the error code instead of the error
                           message.
--msg-template=MSG_TEMPLATE
                           Template used to display error messages. This is a
                           python new-style format string used to format the
                           message information with keys file, line, code, type
                           and msg. Example: --msg-template='{file} || {line} ||
                           {code} || {type} || {level} || {msg}'
-W WARNING                If WARNING is 'error', treat all warnings as errors
```

Así que transformemos nuestro modelo en un formato de árbol así encontrar y leer mejor y más rápido este formato de archivos.

```
devasc@labvm:~/labs/devnet-src/pyang$ pyang -f tree ietf-interfaces.yang
ietf-interfaces.yang:6: error: module "ietf-yang-types" not found in search path
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
```

Conclusiones

1. Se aprendió a navegar y seleccionar modelos YANG en GitHub, lo que nos permite identificar y utilizar modelos de datos relevantes para la gestión de redes.
2. Se utilizaron herramientas como `pyang` para transformar y visualizar modelos YANG, facilitando su comprensión y aplicación en entornos de red.