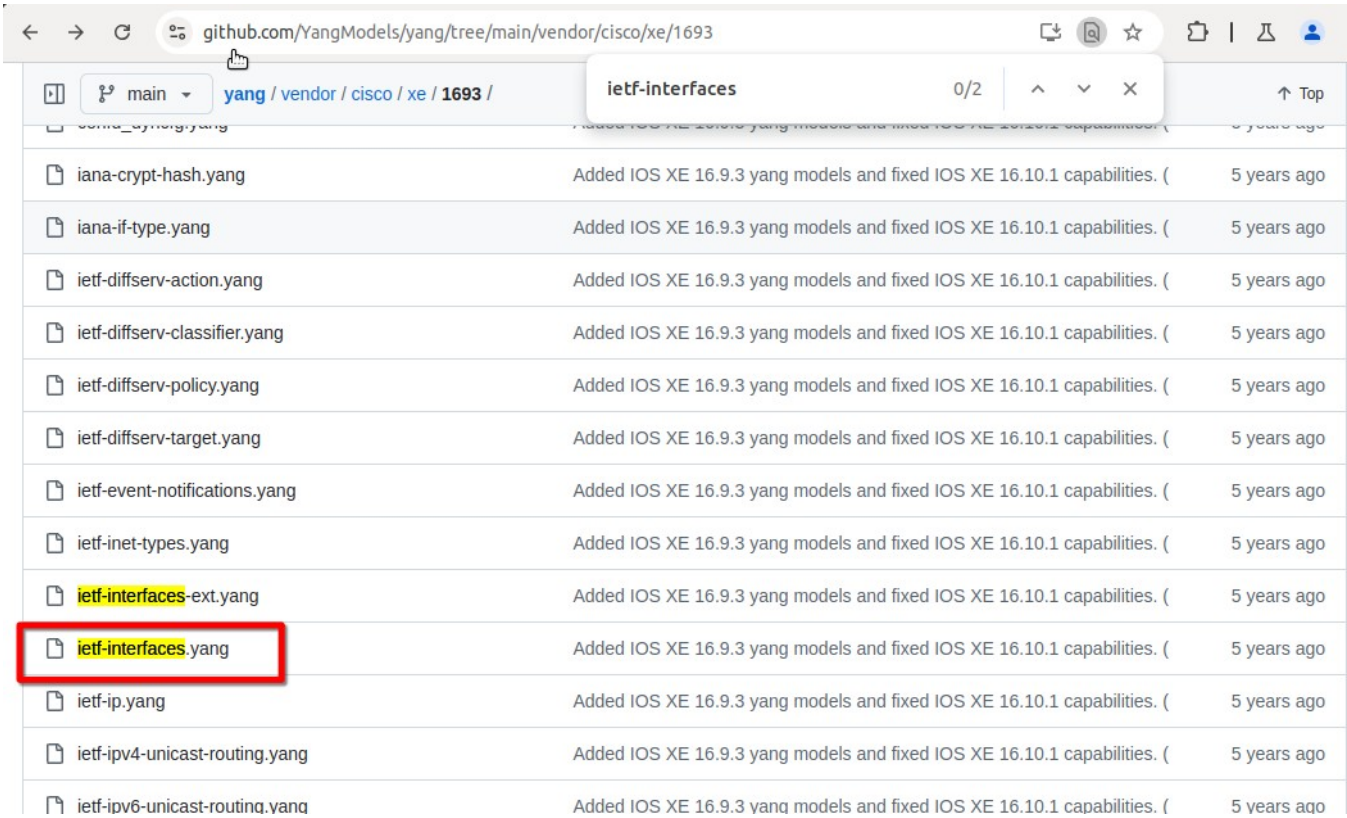
Laboratorio 10: Exploramos YANG Models.

Sergio Sebastian Pezo Jimenez - 20224087G

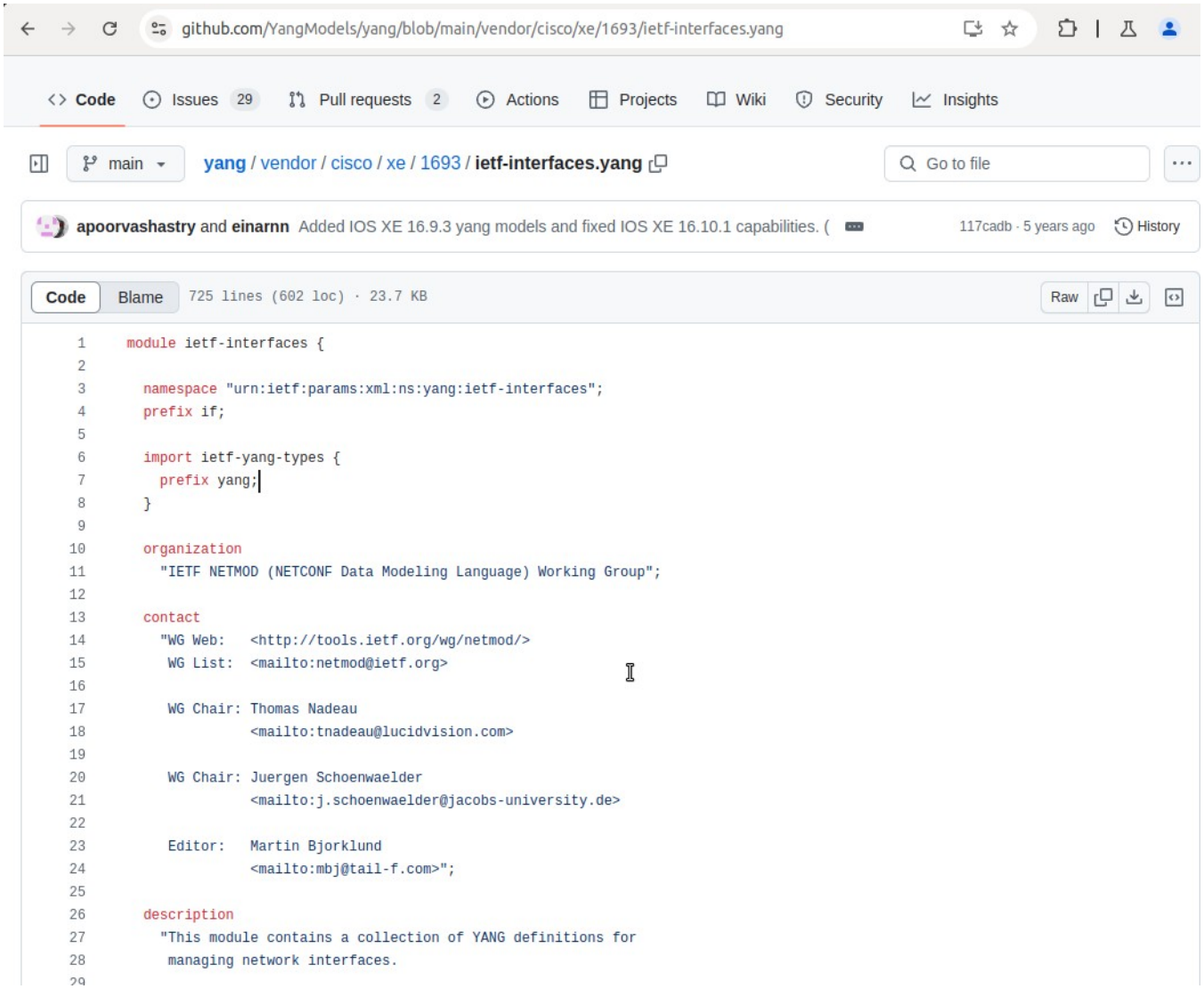
Inicializamos la DEVASC VM.

Exploramos un modelo YANG en GitHub.

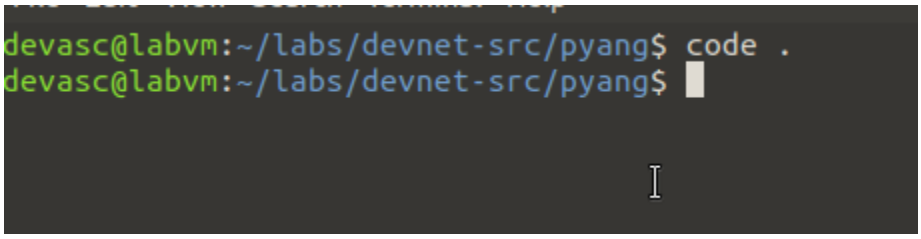
Navegamos en el repositorio `https://github.com/YangModels/yang`, hasta encontrar nuestro archivo `.yang` de interés



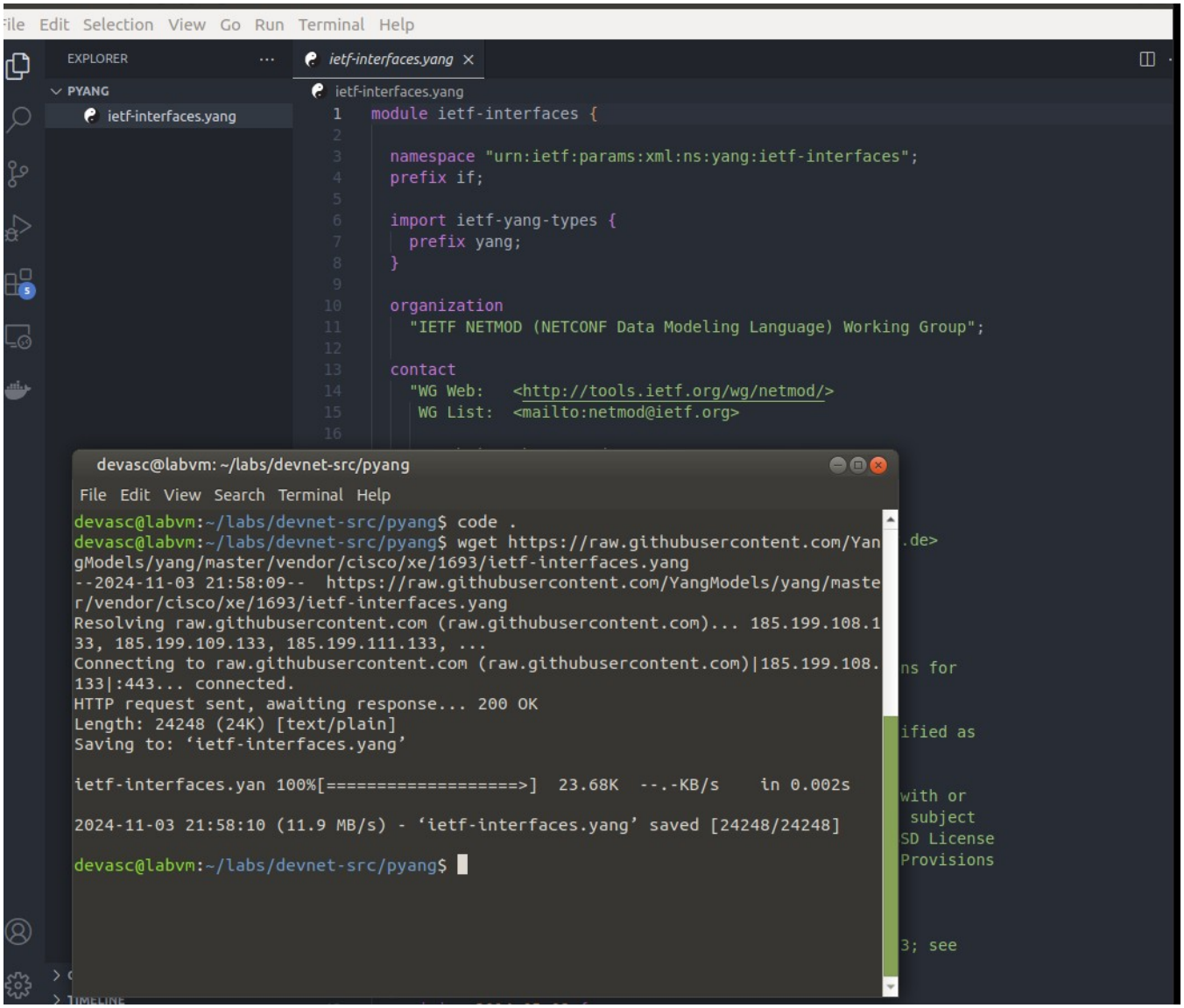
Arhcivo que define un modelo de datos para la gestión de interfaces de red, basado en la especificación del IETF.



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



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-interfa`, 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 tranformación 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 tranformemos 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
devasc@labvm:~/labs/devnet-src/pyang$
```

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.