

# Guía Instalación ns-3 versión 3.33

## 2021-1

Bienvenidas y bienvenidos a nuestra guía de instalación de *ns-3* junto con la librería *OpenAI Gym*, aquí encontrarás una orientación paso a paso de los procedimientos que puedes realizar para llevar a cabo una instalación exitosa. Esta guía es escrita para el sistema operativo Ubuntu 20.04.2.0 de escritorio, si deseas consultar detalles para otros sistemas operativos te recomendamos revisar este recurso:

<https://www.nsnam.org/wiki/Installation>

### Antes de comenzar

Recomendamos fuertemente que inicien actividades de preparación del entorno de desarrollo en fechas tempranas a la entrega, no es algo que se pueda tomar a la ligera, toma su tiempo la instalación de *ns-3*.

La instalación es posible hacerla para Windows 10 usando WSL (Windows Subsystem for Linux), sin embargo, no se garantiza que las interfaces graficas se desplieguen ya que depende de librerías que deben ser instaladas de manera adicional, nuestra recomendación es usar un software de virtualización (VirtualBox, VMware, entre otros) e instalar Ubuntu en él o usar propiamente Ubuntu como Sistema Operativo de su dispositivo.

***Python 2 ya no es compatible con Python Software Foundation:*** A partir del 1 de enero de 2020, la rama 2.x del lenguaje de programación Python ya no es compatible con sus creadores Python Software Foundation, por lo tanto, esta guía está centrada para instalar paquetes para Python 3.x.



## Paso 1: Instala paquetes para ns-3.33

Desde la terminal de Ubuntu (puedes abrirla con la combinación de teclas *Control + Alt + T*) ejecuta los siguientes comandos para instalar los paquetes requeridos:

Requisitos mínimos para los usuarios de C++:

```
sudo apt-get install g++ python3
```

Requisitos mínimos para los usuarios de la API de Python 3:

```
sudo apt-get install g++ python3 python3-dev pkg-config sqlite3
```

Requisitos mínimos para Python 3:

```
sudo apt-get install python3-setuptools git
```

Requisitos para Netanim animator:

```
sudo apt-get install qt5-default mercurial
```

Requisitos ns-3-pyviz visualizer:

```
sudo apt-get install gir1.2-gooocanvas-2.0 python3-gi python3-gi-cairo python3-pygraphviz gir1.2-gtk-3.0 ipython3
```

Soporte para emulación distribuida basada en MPI:

```
sudo apt-get install openmpi-bin openmpi-common openmpi-doc libopenmpi-dev
```

Soporte para bake build tool:

```
sudo apt-get install autoconf cvs bzip2 unrar
```

Debugging:

```
sudo apt-get install gdb valgrind
```



Soporte para el programa de verificación de estilo de código `utils/check-style.py`

```
sudo apt-get install uncrustify
```

Doxygen y documentación en línea relacionada:

```
sudo apt-get install doxygen graphviz imagemagick
```

```
sudo apt-get install texlive texlive-extra-utils texlive-latex-extra texlive-font-utils dvipng latexmk
```

Manual y tutorial de ns-3 escritos en reStructuredText para Sphinx:

```
sudo apt-get install python3-sphinx dia
```

Soporte de la biblioteca científica GNU (GSL) para modelos de error WiFi 802.11b más precisos (no es necesario para OFDM):

```
sudo apt-get install gsl-bin libgsl-dev libgsl23 libgslcblas0
```

Para leer trazas de paquetes pcap:

```
sudo apt-get install tcpdump
```

Soporte de base de datos para statistics framework:

```
sudo apt-get install sqlite sqlite3 libsqlite3-dev
```

Versión basada en XML de la tienda de configuración (requiere libxml2 >= versión 2.7):

```
sudo apt-get install libxml2 libxml2-dev
```

Soporte para generar enlaces de Python modificados:

```
sudo apt-get install cmake libc6-dev libc6-dev-i386 libclang-6.0-dev llvm-6.0-dev automake python3-pip
```

```
sudo python3 -m pip install --user cxxfilt
```

Un sistema de configuración basado en GTK:

```
sudo apt-get install libgtk-3-dev
```

Para experimentar con máquinas virtuales y ns-3:

```
sudo apt-get install vtun lxc uml-utilities
```

Soporte para el módulo openflow (requiere libxml2-dev si no está instalado arriba):

```
sudo apt-get install libxml2 libxml2-dev
```

## Paso 2: Descarga ns-3

Descargaremos una copia de ns-3 usando git, escoge la carpeta de su preferencia donde quieres clonar el repositorio de ns-3, usa la siguiente instrucción:

```
git clone https://gitlab.com/nsnam/ns-3-allinone.git
```

Luego usa:

```
cd ns-3-allinone
ls
```

Y encontraras los siguientes archivos:

```
build.py* constants.py dist.py* download.py* README util.py
```

Desde este directorio podemos descargar varias versiones de ns-3 (por ejemplo, la versión 3.29), para nuestro caso descargaremos la versión 3.33 usado la siguiente instrucción:

```
./download.py -n ns-3.33
```

Después de descargarlo, tendremos nuevos directorios (hemos marcado con azul el directorio que posee la versión 3.33 de ns-3):

```
Bake          constants.py  download.py   ns-3.33      __pycache__
util.py        build.py      dist.py       netanim      pybindgen
README
```



### Paso 3: Instala paquetes para OpenAI Gym

Ejecuta las siguientes instrucciones para instalar los paquetes necesarios:

```
sudo apt-get install libzmq5 libzmq5-dev
```

```
sudo apt-get install libprotobuf-dev
```

```
sudo apt-get install protobuf-compiler
```

### Paso 4: Descarga OpenAI Gym

En un directorio diferente al de ns-3 descarga la librería OpenAI Gym desde el siguiente enlace:

<https://codeload.github.com/tkn-tub/ns3-gym/tar.gz/ns3-gym-1.0.0>

**Nota:** Puedes descargarlo desde un navegador web o usando el comando:

```
curl -O https://codeload.github.com/tkn-tub/ns3-gym/tar.gz/ns3-gym-1.0.0
```

(que debes instalar «curl» primero con: `sudo apt install curl`)

En nuestro caso hemos descargado el archivo en el mismo directorio raíz donde se encuentra la carpeta ns-3-allione:

```
ns-3-allione      ns3-gym-ns3-gym-1.0.0
```

Descomprimos el archivo con la instrucción:

```
tar -xzf ns3-gym-1.0.0
```

Renombramos la carpeta ns3-gym-ns3-gym-1.0.0 como opengym con la instrucción:

```
mv ns3-gym-ns3-gym-1.0.0/ opengym
```

Luego movemos la carpeta opengym al directorio ns-3-allione/ns-3.33/contrib/ con la instrucción:

```
mv opengym/ ns-3-allione/ns-3.33/contrib/
```



Verificamos que se encuentre la carpeta `opengym` en el directorio establecido:

```
cd ns-3-allione/ns-3.33/contrib/
```

Debemos encontrar algo así:

```
opengym      wscript
```

## Paso 5: Preparación compilación OpenAI Gym

Para ahorrar tiempo, primero descargamos ns-3.33 y OpenAI Gym, de esta forma sólo realizamos una sola compilación para ambos, sin embargo, debemos ajustar primero algunas cosas para la librería.

Ingresamos a la carpeta `opengym`:

```
cd opengym/
```

Compilamos los mensajes de Protobuf manualmente:

```
protoc -I=./model/ --python_out=./model/ns3gym/ns3gym  
./model/messages.proto
```

Instalamos el módulo para python 3 ns3gym:

```
pip3 install ./model/ns3gym
```

## Paso 6: Compilación

Debemos ubicarnos en dentro de la carpeta `ns-3-allinone` y allí ejecutamos la siguiente instrucción (*este proceso toma su tiempo*):

```
sudo ./build.py --enable-examples --enable-tests
```

**Nota:** Aparecerán algunas advertencias en la compilación sugiriendo cambios en el código fuente de ns-3, para nuestro propósito podemos ignorarlas. En ocasiones no hay suficiente espacio de memoria principal (memoria de tipo RAM) para ejecutar toda la compilación, en este caso volvemos a ejecutar el comando y continuara la compilación en el punto donde se detuvo, en caso contrario, por favor cierre todas las aplicaciones y vuelva a intentarlo sin abrir otro programa durante el proceso de compilación.

Si la compilación fue un éxito debemos ver en la consola el siguiente mensaje:

```
'Build' finished successfully
```



Tras terminar la compilación anterior, nos ubicamos dentro de la carpeta `ns-3.33`:

```
cd ns-3.33/
```

Luego ejecutamos la siguiente instrucción:

```
sudo ./waf configure -d debug --enable-examples --enable-tests
```

Nos mostrara una lista de elementos habilitados y deshabilitados, si todo es correcto nos mostrara el siguiente mensaje en la terminal:

```
'configure' finished successfully
```

Ahora de nuevo ejecutamos el comando `waf`:

```
sudo ./waf
```

**Opcional:** Puede verificar la *instalación* ejecutando la siguiente instrucción en la misma carpeta de `ns-3.33` (esta instrucción toma su tiempo en completarse):

```
sudo ./test.py
```

## Paso 7: Prueba OpenAI Gym

Esta prueba es para verificar la librería OpenAI Gym, al mismo tiempo se verifica el funcionamiento de `ns-3`. Debemos abrir dos terminales en el sistema operativo, uno para iniciar la simulación y el otro para iniciar el agente:

**Terminal 1:** Dentro de la carpeta `ns-3.33` ejecutamos la siguiente instrucción para iniciar el código de la simulación:

```
sudo ./waf --run "contrib/opengym/examples/opengym"
```

**Terminal 2:** Desde la carpeta de `ns-3.33` ingresamos a la carpeta de `opengym`:

```
cd ./contrib/opengym/examples/opengym/
```

Luego ejecutamos la siguiente instrucción para iniciar el agente que se conectara a la simulación:

```
python3 ./test.py --start=0
```

Por último, te felicitamos por llegar hasta este punto de la *guía de instalación de ns-3.33 con OpenAI Gym*, buena suerte en tu proyecto.

