



TSU en
Desarrollo de Software
Multiplataforma



FACULTAD DE: *TSU DESARROLLO DE SOFTWARE
MULTIPLATAFORMA*

ALUMNO(A): NESTOR EMMANUEL BRIONES RAMÍREZ

MATRÍCULA: *1220100321*

MATERIA: APlicaciones Web

NOMBRE DE LA ACTIVIDAD: frameworks para software
empresarial.

DOCENTE: GABRIEL BARRÓN RODRIGUEZ

1. instalación de *VirtualBox*

VirtualBox es un software de virtualización desarrollado por Oracle que podemos encontrar instalado en los equipos de los laboratorios del Departamento de Informática y Sistemas.

Este software permite que podamos crear y ejecutar máquinas virtuales (incluso corriendo sistemas operativos) dentro de nuestro propio sistema, el cual, actuará como anfitrión. Para instalar el VirtualBox en nuestro propio equipo, podemos acudir al siguiente enlace: <https://www.virtualbox.org/>



Ilustración 1. Vista de la página de inicio del sitio web de *VirtualBox*

Dentro de la opción *Downloads* encontramos diversas opciones de descarga. Dependiendo del sistema operativo de nuestro sistema anfitrión elegiremos uno u otro sistema operativo. Los disponibles son:

- Microsoft Windows

- Sistemas operativos OS X
- Distribuciones Linux
- Sistemas operativos Solaris

A continuación se muestran los pasos para la instalación en el sistema operativo Windows. Haciendo doble clic sobre el ícono del fichero descargado desde virtualbox.org se iniciará el asistente de instalación:



Ilustración 2. Vista del wizard de instalación de VirtualBox

Al avanzar en el asistente se nos presentará la posibilidad de seleccionar qué opciones del VirtualBox queremos instalar. Seleccionaremos VirtualBox Aplicación, lo cual incluye la aplicación completa, los módulos de soporte de USB, los módulos de red y el soporte para Python.

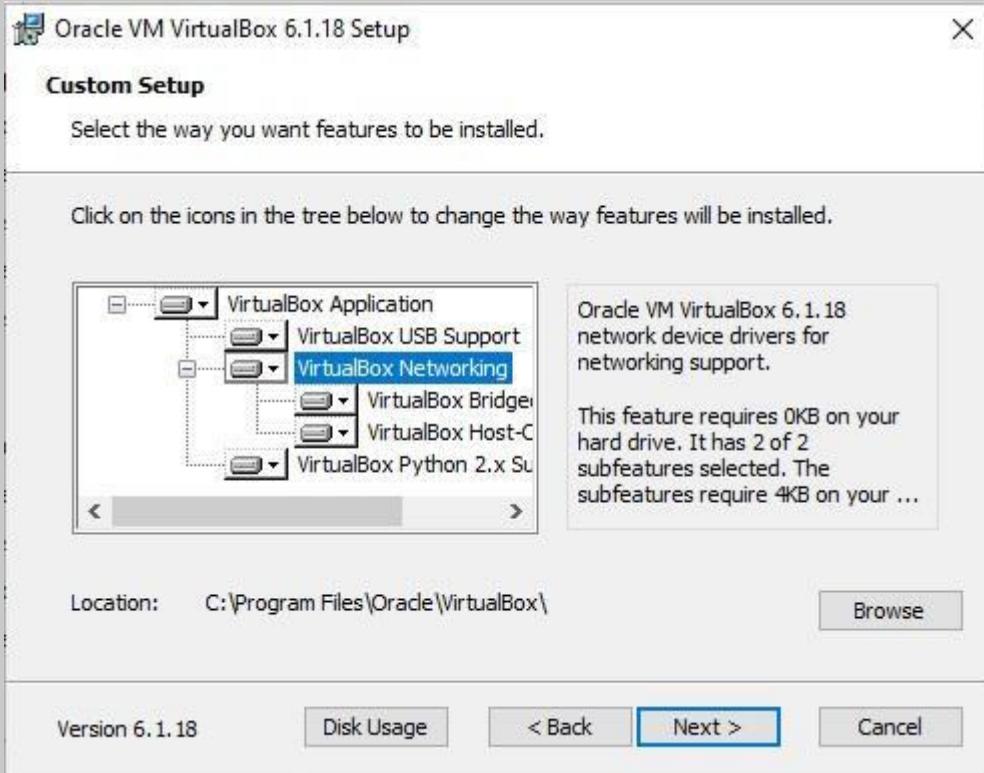


Ilustración 3. Opciones de instalación

Al seleccionar siguiente se nos pide si deseamos crear un acceso directo en el escritorio hacia la aplicación VirtualBox, un acceso dentro de menú de aplicaciones, un acceso en la barra de inicio rápido y por último asociar las extensiones de ficheros de VirtualBox a la propia aplicación (Ilustración 4).

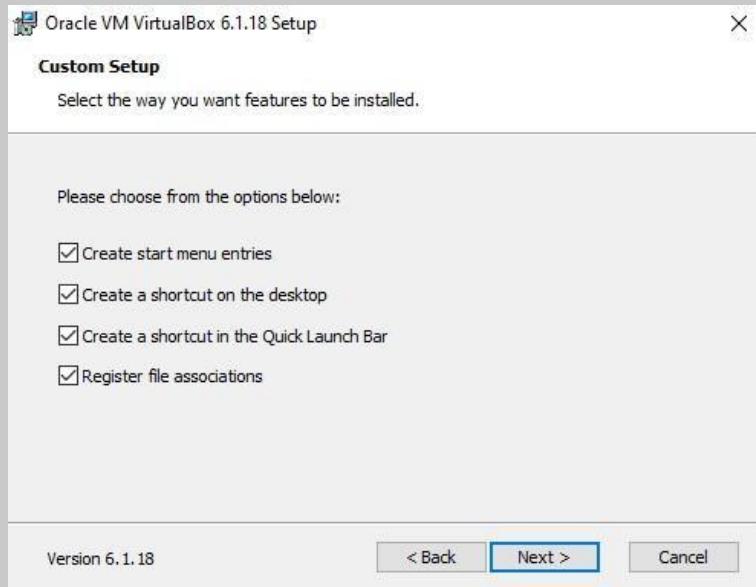


Ilustración 4. Opciones para la creación de accesos directos

En la Ilustración 5 se puede observar que, durante el proceso de instalación, habrá un reseteo temporal de la conexión de red en el equipo anfitrión.



Ilustración 5. Aviso de desconexión temporal de la red durante la instalación

El proceso continúa indicando que la instalación comenzará (Ilustración 6) y una vez realiza, se alerta de la posibilidad de ejecutar VirtualBox nada más finalizar la instalación (Ilustración 7).

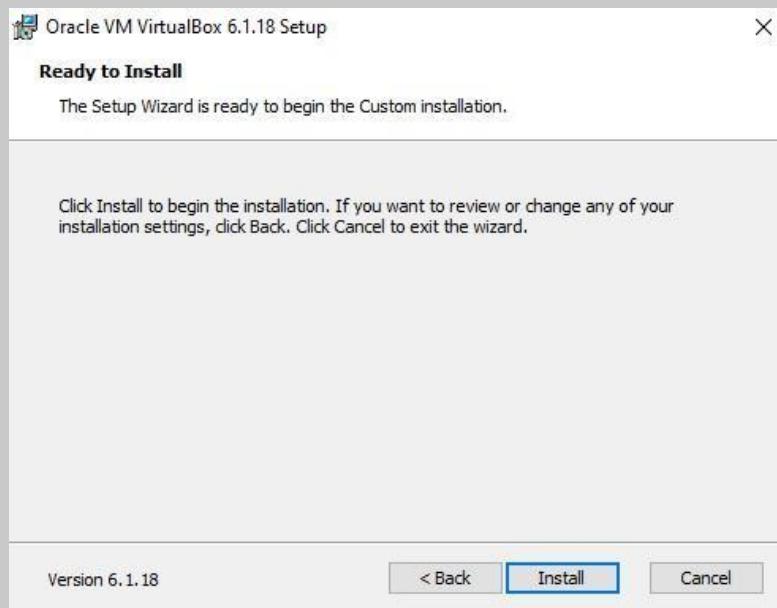


Ilustración 6. Confirmación final de instalación



Ilustración 7. Confirmación de instalación de VirtualBox

En la página de descarga de VirtualBox también se encuentra el paquete de extensiones opcional, el cual es recomendable instalar de igual forma (**VirtualBox Extensión Pack**).

2. Creación de máquinas virtuales

También tenemos la posibilidad de crear una máquina virtual nueva (no importada o creada previamente). Para poder realizar el proceso necesitamos un fichero imagen (ISO) que puede ser descargado desde las webs del sistema operativo que vayamos a instalar.

Vamos a realizar el ejemplo de creación de una máquina virtual corriendo la distribución Ubuntu. El primer paso sería acudir a la web de Ubuntu y descargar el fichero ISO.

En la web ubuntu.com podemos encontrar una opción de Download, a partir de la cual nos descarga la ISO correspondiente (ilustración 9)

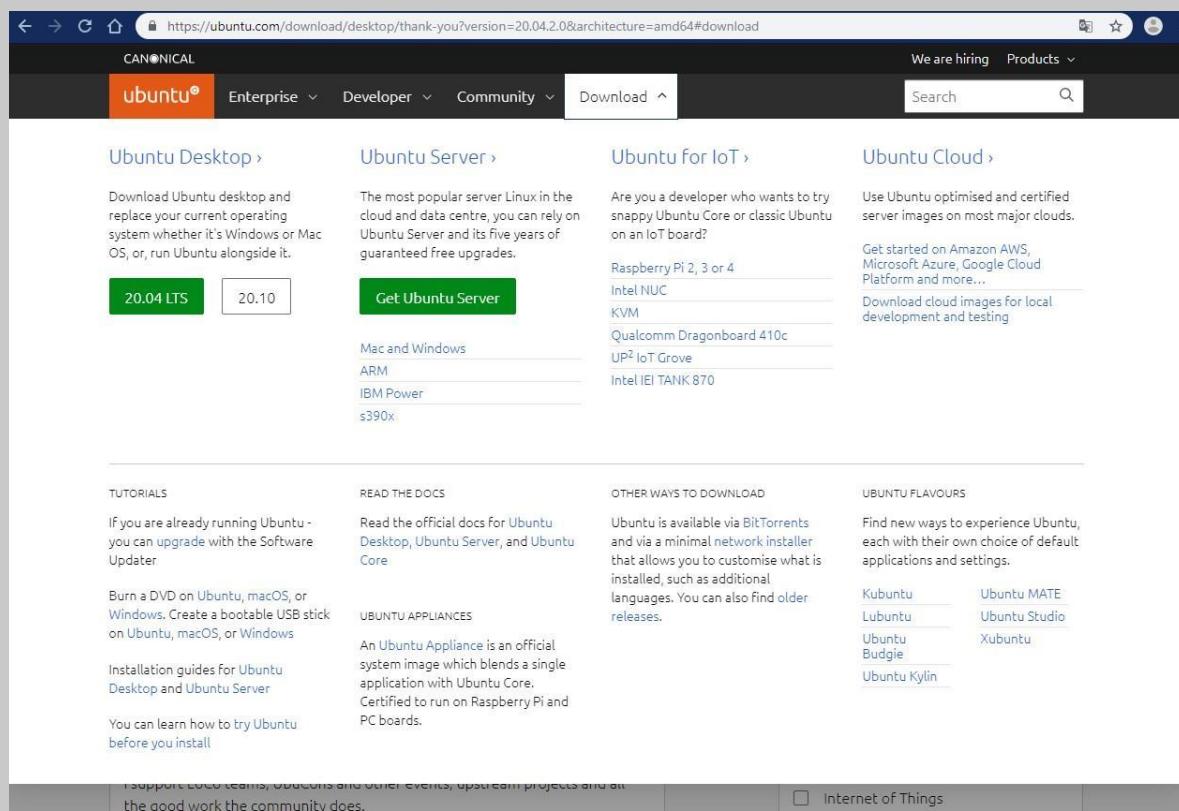


Ilustración 9. Descarga de la ISO de Ubuntu

Una vez descargado podemos arrancar VirtualBox para iniciar el proceso de creación de la máquina. Seleccionamos la opción NUEVA y nos aparecerá la imagen de la ilustración 10:

Indicaremos un nombre para la máquina, y el sistema operativo que deseamos instalar. En nuestro caso LINUX y UBUNTU 64bits, respectivamente.

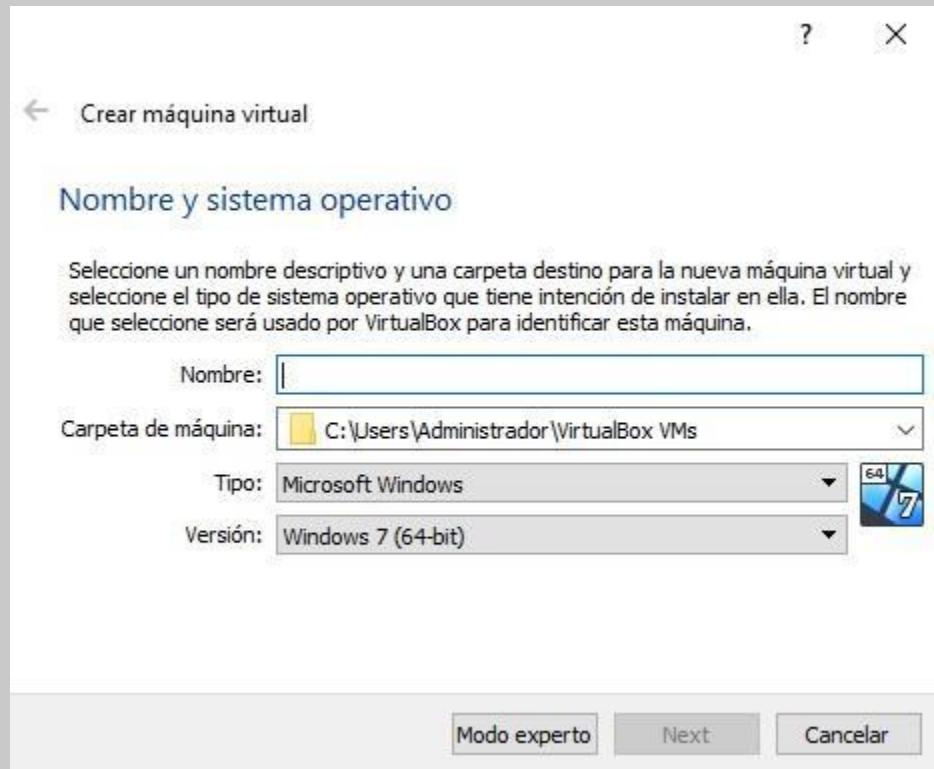


Ilustración 10. Ricard de creación de máquinas virtuales

En la siguiente pantalla aparece la posibilidad de asignar la memoria a nuestra máquina virtual.

En nuestro caso asignaremos 2Gb de RAM (Ilustración 11).

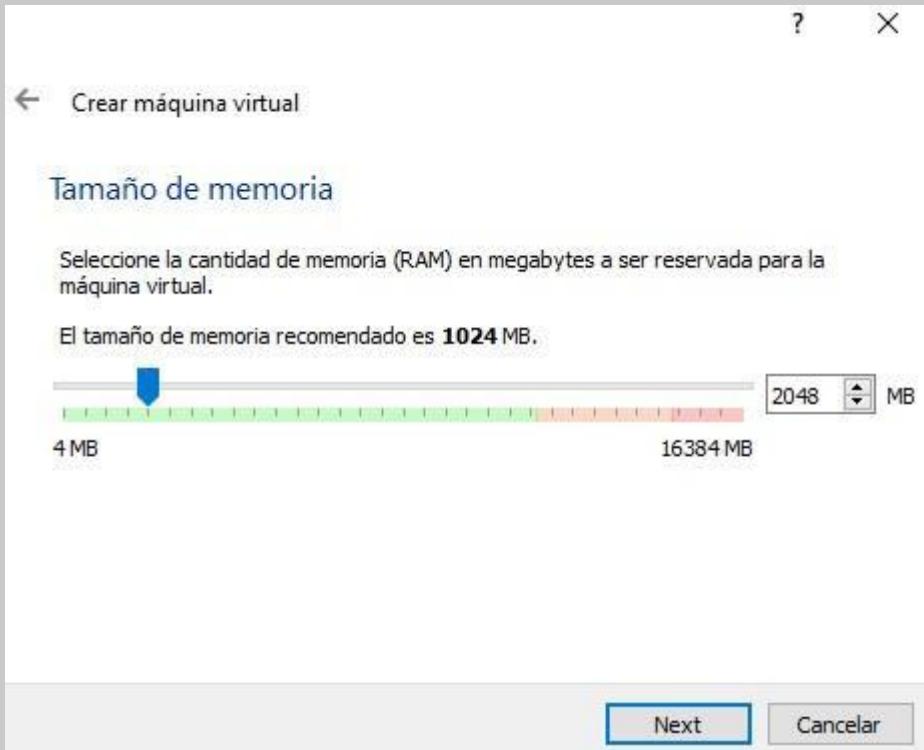


Ilustración 10. Cantidad de memoria RAM a asignar a la máquina virtual

A continuación, se nos solicita que indiquemos si deseamos crear un nuevo disco duro para esta máquina. Indicamos que sí, seleccionamos formato VDI, tamaño fijo y un espacio, por ejemplo, de 10Gb.

Una vez creada la máquina virtual la tendremos disponible dentro del VirtualBox. Podremos crear tantas máquinas como sea necesario.

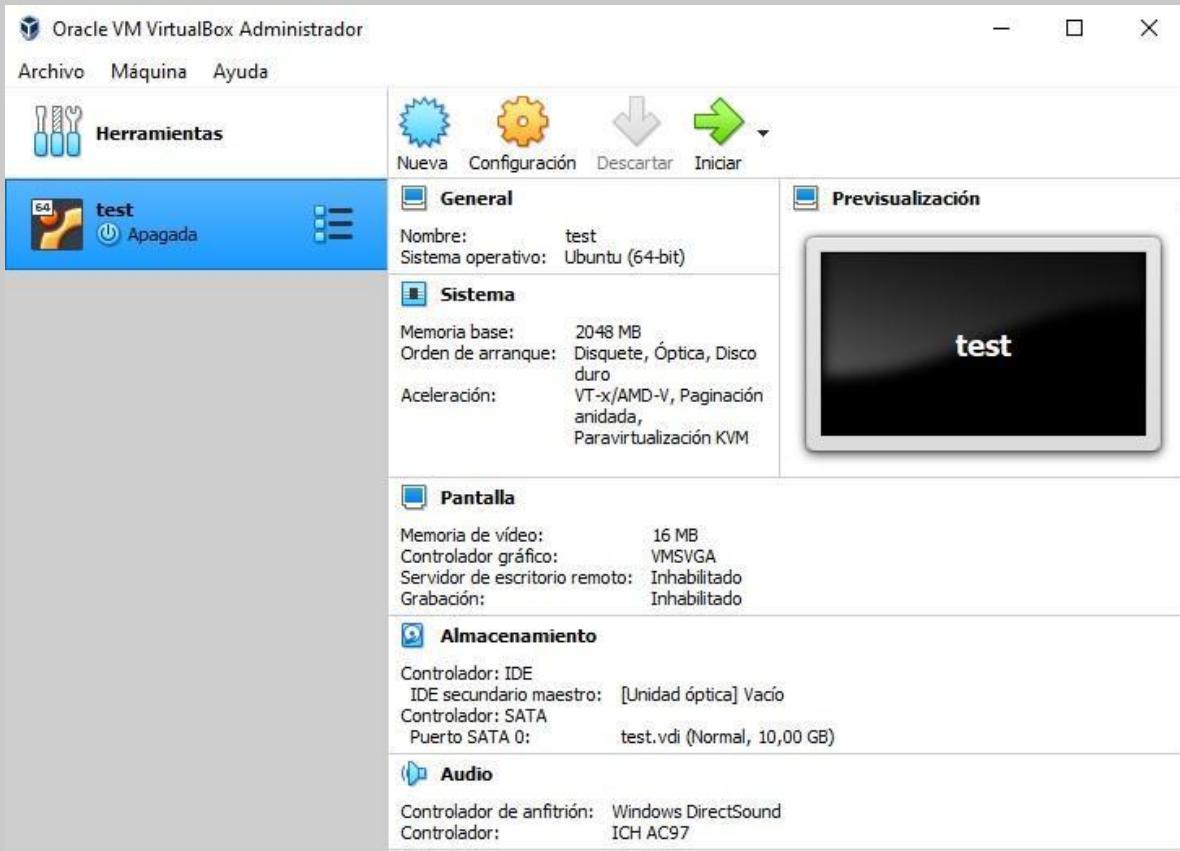


Ilustración 11. Características la máquina virtual

El siguiente paso será conectarle la imagen ISO que nos descargamos del Ubuntu como si fuese una unidad de CDROM, de forma que la máquina virtual arranque desde dicha ISO y se inicie el proceso de instalación.

Para conseguirlo iremos a CONFIGURACION -> ALMACENAMIENTO

Y en la unidad óptica le incorporaremos el fichero ISO tal y como se ve en la ilustración 13.

Arrancaremos la máquina virtual y comprobaremos que se inicia la ISO (ilustración 14).

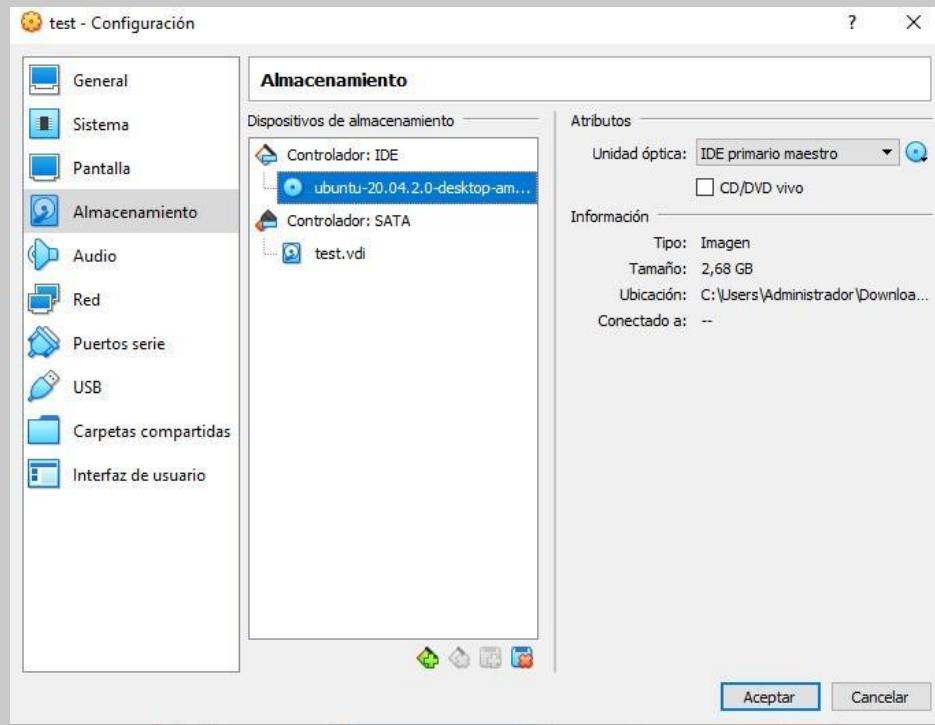


Ilustración 12. Imagen ISO de *Ubuntu Server* bajo la controladora *IDE*

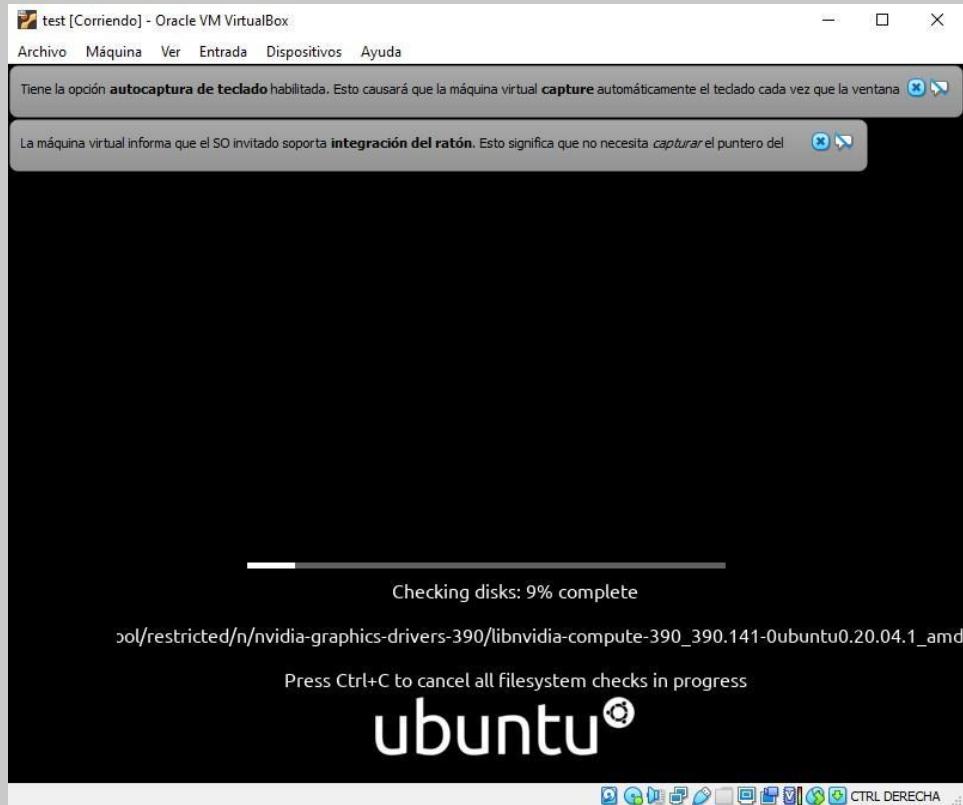


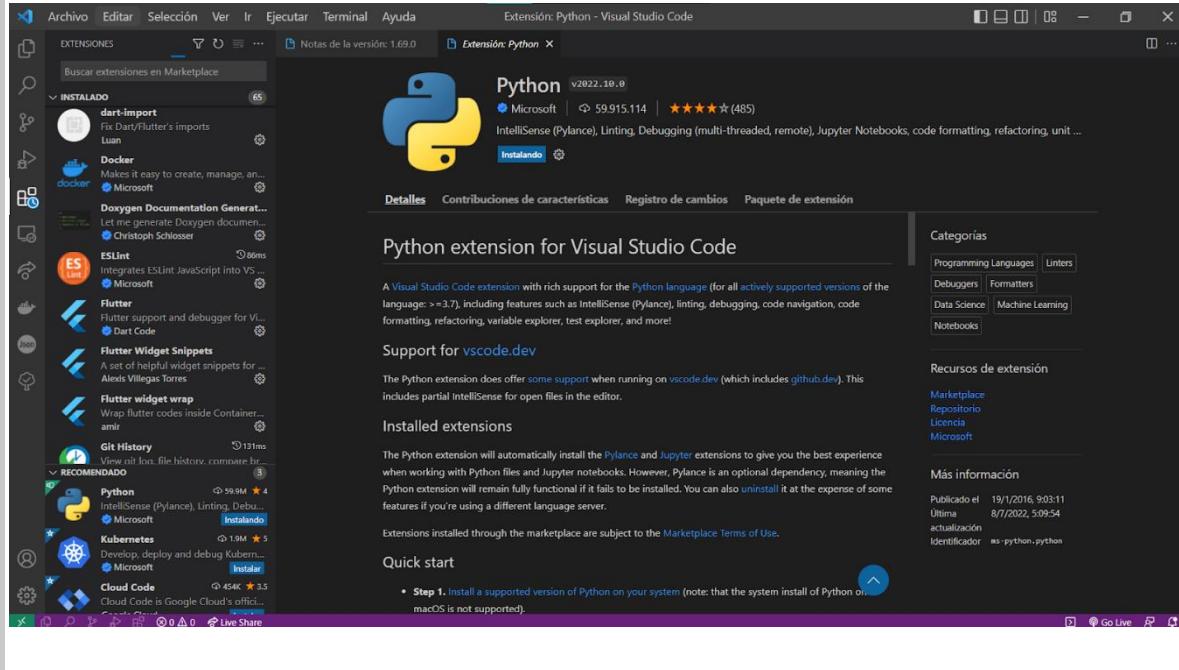
Ilustración 13. Vista de la máquina virtual creada antes de comenzar el proceso de instalación del SO

3. Instalación de visual estudio y sus extensiones

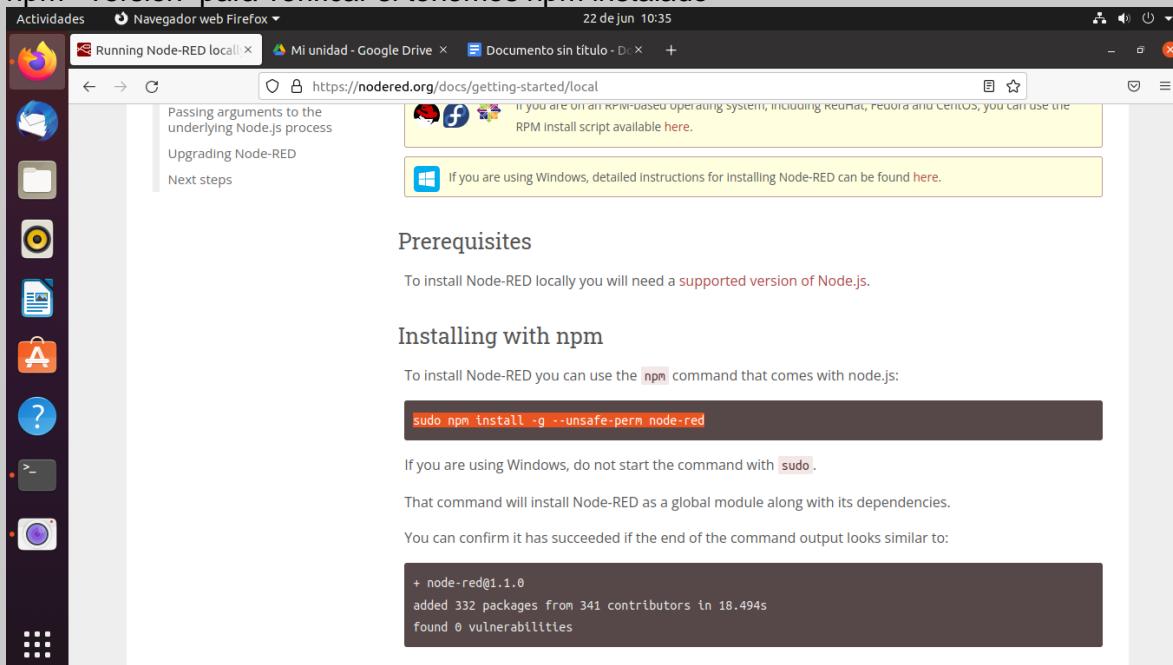
Para instalar VS Conde usando el comando de terminal, siga estos pasos:

1. Abierto **terminal**.
2. Ingrese el siguiente comando y presione **Iniciar sesión**:
sudo apt-get update
3. Ingrese el siguiente comando y presione **Iniciar sesión**:
sudo apt install software-properties-common apt-transport-https wget
4. Ingrese el siguiente comando para importar la clave Microsoft GPC y presione **Iniciar sesión**:
wget -q https://packages.microsoft.com/keys/microsoft.asc -O- | sudo apt-key add -
5. Ingrese el siguiente comando para habilitar el repositorio de VS Code y presione **Iniciar sesión**:
sudo add-apt-repository "deb https://packages.microsoft.com/repos/vscode stable main"
6. Ingrese el siguiente comando para instalar Visual Studio Code y presione **Iniciar sesión**:
sudo apt-get install code

Extensiones de la visual



primero utilizamos el comando
node –versión y nos dará el comando para instalar node
sudo apt install node
después pondremos
npm –version para verificar si tenemos npm instalado



esta página es para iniciar el node js
usaremos los comandos que nos marca
<https://nodered.org/docs/getting-started/local>

1- pondremos el primer comando para iniciar node js
sudo npm install -g --unsafe-perm node-red

Actividades Terminal 22 de jun 10:29

```
nestor@NestorB:~$ sudo apt install npm
No se ha encontrado la orden «npm», pero se puede instalar con:
sudo apt install npm
nestor@NestorB:~$ sudo apt install npm
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Los paquetes indicados a continuación se instalaron de forma automática y ya no
son necesarios.
  docker-ce-rootless-extras docker-scan-plugin slirp4netns
Utilice «sudo apt autoremove» para eliminarlos.
Se instalarán los siguientes paquetes adicionales:
  gyp javascript-common libbjs-inherits libbjs-ts-typedarray libjs-psl
  libbjs-typedarray-to-buffer libnode-dev libpython2-stdlib
  libpython2.7-minimal libpython2.7-stdlib libssl1.1 libuv1-dev
  node-abbrev node-ajv node-ansi node-ansi-align node-ansi-regex
  node-ansi-styles node-ansi-types node-aproba node-archy
  node-are-we-there-yet node-asap node-asn1 node-assert-plus node-asynckit
  node-aws-sign2 node-aws4 node-balanced-match node-bcrypt-pbkdf node-bl
  node-bluebird node-boxen node-brace-expansion node-builtins-modules
  node-builtins node-cache node-call-limit node-camelcase node-caseless
That command will install Node-RED as a global module along with its dependencies.
```

You can confirm it has succeeded if the end of the command output looks similar to:

```
+ node-red@1.1.0
added 332 packages from 341 contributors in 18.494s
found 0 vulnerabilities
```

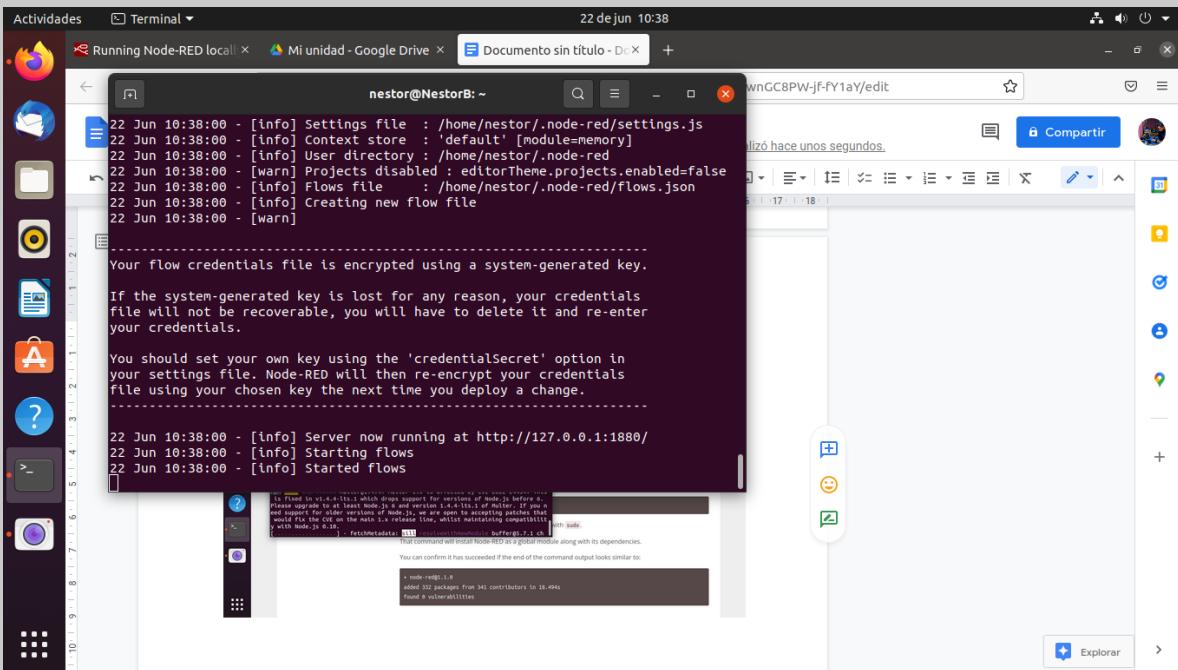
Actividades Terminal 22 de jun 10:36

```
nestor@NestorB:~$ sudo npm install -g --unsafe-perm node-red
Configurando node-gauge (2.7.4-1) ...
Configurando node-normalize-package-data (2.5.0-1) ...
Configurando node-configstore (5.0.1-1) ...
Configurando node-boxen (4.2.0-2) ...
Configurando node-npmlog (4.1.2-2) ...
Configurando node-yargs (15.3.0-1) ...
Configurando node-cacheable (11.3.3-2) ...
Configurando node-read-package-json (2.1.1-1) ...
Configurando node-gyp (6.1.0-3) ...
Configurando node-libpax (10.2.1-2) ...
Configurando npm (6.14.4+ds-ubuntu2) ...
Procesando disparadores para man-db (2.9.1-1) ...
Procesando disparadores para desktop-file-utils (0.24-1ubuntu3) ...
Procesando disparadores para mime-support (3.64ubuntu1) ...
Procesando disparadores para gnome-menus (3.36.0-ubuntu1) ...
Procesando disparadores para libc-bin (2.31-0ubuntu9.9) ...
nestor@NestorB:~$ sudo npm install -g --unsafe-perm node-red
npm WARN deprecated multer@1.4.4: Multer 1.x is affected by CVE-2022-24434. This
is fixed in v1.4.4-lts.1 which drops support for versions of Node.js before 6.
Please upgrade to at least Node.js 6 and version 1.4.4-lts.1 of Multer. If you n
eed support for older versions of Node.js, we are open to accepting patches that
would fix the CVE on the main 1.x release line, whilst maintaining compatibility
with Node.js 0.10.
[.....] - fetchMetadata: sill resolveWithNewModule buffer@5.7.1 ch
That command will install Node-RED as a global module along with its dependencies.
```

You can confirm it has succeeded if the end of the command output looks similar to:

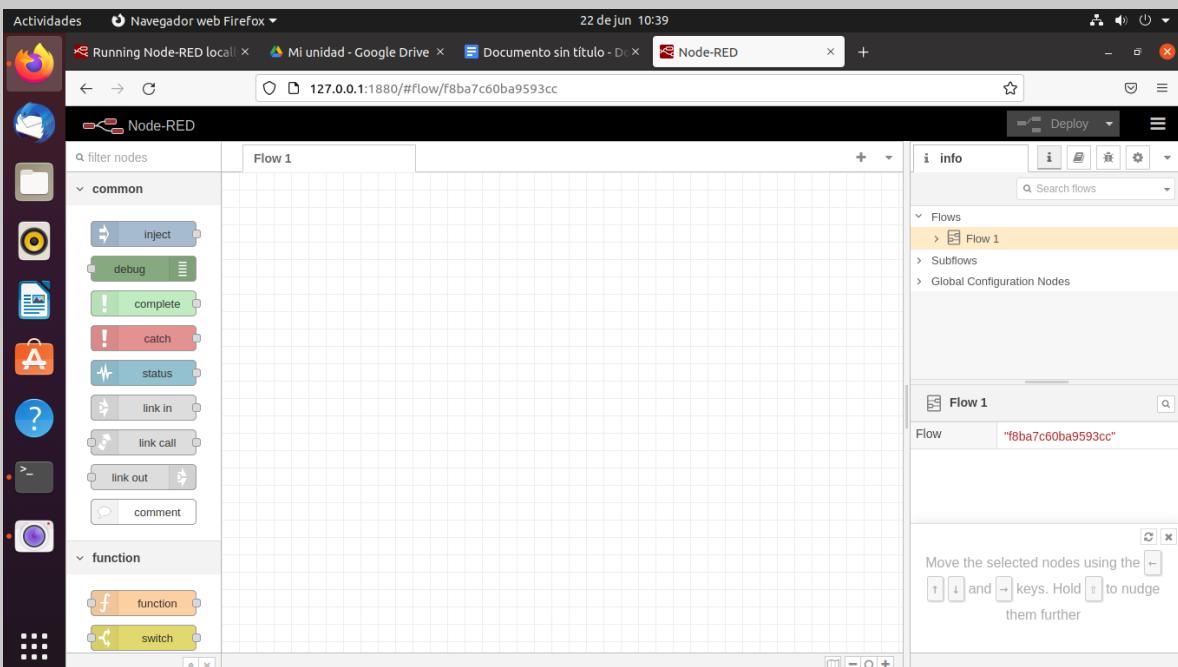
```
+ node-red@1.1.0
added 332 packages from 341 contributors in 18.494s
found 0 vulnerabilities
```

damos clic con control a la dirección que nos muestra



nos abrirá la interface de node

ACTUALIZAMOS NUESTRO EQUIPO Y EACCEDEMOS A SUPER USUARIO



esta página nos muestra cómo como instalar mosquito
<https://mosquitto.org/download/>

The screenshot shows a Firefox browser window with the URL <https://mosquitto.org/download/>. The page content includes sections for Mac, Linux distributions with snap support (containing a bullet point for `sudo snap install mosquitto`), Debian (with two bullet points about Debian proper and repositories), Raspberry Pi (mentioning availability through the main repository and a link to the Debian repository), and Ubuntu (with a note about Eclipse Foundation cookie policy). At the bottom, there's a dark bar with 'Decline' and 'Allow cookies' buttons.

el siguiente comando fue donde se instala mosquito con el snap

The screenshot shows a terminal window titled 'nestor@NestorB: ~'. It displays the following log output:

```
22 Jun 10:38:00 - [info] Flows file      : /home/nestor/.node-red/flows.json
22 Jun 10:38:00 - [info] Creating new flow file
22 Jun 10:38:00 - [warn]

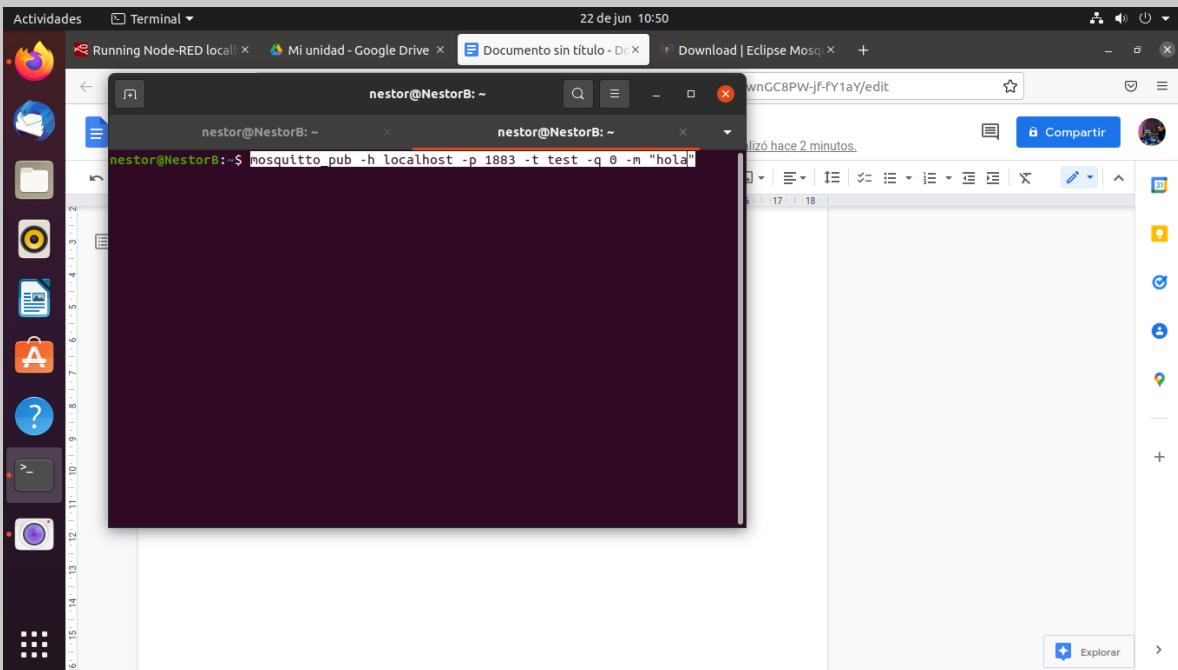
-----
Your flow credentials file is encrypted using a system-generated key. Mosquitto
If the system-generated key is lost for any reason, your credentials
file will not be recoverable, you will have to delete it and re-enter
your credentials.

You should set your own key using the 'credentialSecret' option in
your settings file. Node-RED will then re-encrypt your credentials
file using your chosen key the next time you deploy a change.
-----

22 Jun 10:38:00 - [info] Server now running at http://127.0.0.1:1880/ appearing in Debian as part of the normal Debian procedures.
22 Jun 10:38:00 - [info] Starting flows
22 Jun 10:38:00 - [info] Started flows
^C22 Jun 10:43:05 - [info] Stopping flows
22 Jun 10:43:05 - [info] Stopped flows
nestor@NestorB: $ snap install mosquitto
Se ha instalado mosquitto 2.0.14 por Mosquitto Team (mosquitto)
nestor@NestorB: $ mosquitto_sub -h localhost -p 1883 -t test[
```

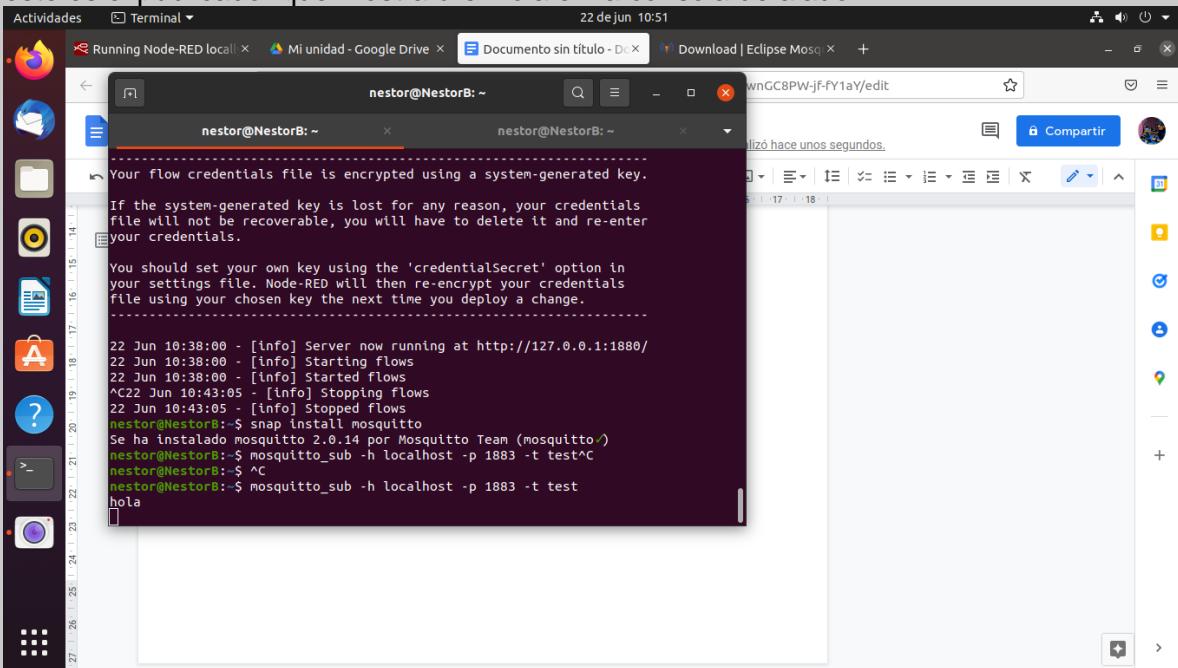
Below the terminal, there's a note about Debian repositories and a link to the Debian repository, followed by the 'Ubuntu' section of the cookie policy bar.

mosquitto_sub -h localhost -p 1883 -t test
mosquitto suscriptor es la parte o programa que se usa en el puerto 1883



```
nestor@NestorB:~$ mosquitto_pub -h localhost -p 1883 -t test -q 0 -m "hola"
```

mosquitto_pub -h localhost -p 1883 -t test -q 0 -m "hola"
este es el publicador que mostrará el hola en la consola de alado



```
Your flow credentials file is encrypted using a system-generated key.  
If the system-generated key is lost for any reason, your credentials  
file will not be recoverable, you will have to delete it and re-enter  
your credentials.  
  
You should set your own key using the 'credentialSecret' option in  
your settings file. Node-RED will then re-encrypt your credentials  
file using your chosen key the next time you deploy a change.  
  
22 Jun 10:38:00 - [info] Server now running at http://127.0.0.1:1880/  
22 Jun 10:38:00 - [info] Starting flows  
22 Jun 10:38:00 - [info] Started Flows  
^C22 Jun 10:43:05 - [info] Stopping flows  
22 Jun 10:43:05 - [info] Stopped flows  
nestor@nestorB:~$ snap install mosquitto  
Se ha instalado mosquitto 2.0.14 por Mosquitto Team (mosquitto)  
nestor@nestorB:~$ mosquitto_sub -h localhost -p 1883 -t test^C  
nestor@nestorB:~$ ^C  
nestor@nestorB:~$ mosquitto_sub -h localhost -p 1883 -t test  
hola
```

aquí nos dio la después en la otra terminal

2- vamos a checar el ip, estos son los comandos por usar
sudo apt install net-tools
ifconfig

3- Descargaremos Arduino para Linux
usamos la página: <https://www.arduino.cc/en/software>

The screenshot shows a Firefox browser window with the URL <https://www.arduino.cc/en/software>. The page header includes the Arduino logo and navigation links for HARDWARE, SOFTWARE, CLOUD, DOCUMENTATION, COMMUNITY, BLOG, and ABOUT. The main content is titled "Future Version of the Arduino IDE" and features a section for "Arduino IDE 2.0 RC (2.0.0-rc7)". It highlights improvements like a faster editor, modern interface, autocompletion, code navigation, and a live debugger. A note mentions moving to stable status with Release Candidate builds and encourages community feedback via a survey form. Another section for "Nightly builds" is present. On the right, there's a "DOWNLOAD OPTIONS" sidebar with links for Windows (Win 10 and newer, 64 bits), Windows MSI Installer, Windows ZIP file, Linux (AppImage 64 bits (X86-64)), Linux ZIP file 64 bits (X86-64), and macOS (10.14: "Mojave" or newer, 64 bits). A "Help" button is at the bottom right.

The screenshot shows a terminal window on a Linux system with the command prompt "nestor@NestorB: ~". The user has run the command "mosquitto_pub -h localhost -p 1883 -t test -q 0 -m "holo"" and then tried to change directories to "Download" and "Downloads" but received errors indicating they don't exist. The user then navigated to the "Descargas" directory and attempted to extract a file named "arduino-1.8.19-linux64.tar.xz". A file manager window titled "xz" is open, showing the file "arduino-1.8.19-linux64.tar.xz" selected. The status bar at the bottom of the terminal window indicates "«arduino-1.8.19-linux64.tar.xz» seleccionado (128.6 MB)".

4- descomprimimos el zip con el siguiente comando
tar -xvf arduino-1.8.19-linux64.tar.xz

5- nos adentramos a la carpeta que se descomprimio
y pondremos el siguiente comando
dando permisos con
sudo chmod +x ./install.sh
instalamos con: sudo ./install.sh

```
Actividades Terminal 22 de jun 11:18
Running Node-RED local x Mi unidad - Google Drive x INSTALA MOSQUITO Y N x Download | Eclipse Mosc x Support the Arduino IDE x +
nestor@NestorB: ~/Descargas/arduino-1.8.19 ~ nestor@NestorB: ~/Descargas/arduin... ~
arduino-1.8.19/tools/WiFi101/tool/firmwares/NINA/1.2.1/
arduino-1.8.19/tools/WiFi101/tool/firmwares/NINA/1.2.1-NINA_W102.bin
arduino-1.8.19/tools/WiFi101/tool/firmwares/NINA/1.2.1/NINA_W102-Uno_WiFi_Rev2.b
in
arduino-1.8.19/tools/WiFi101/tool/firmwares/NINA/1.4.5/
arduino-1.8.19/tools/WiFi101/tool/firmwares/NINA/1.4.5/NINA_W102-Nano_RP2040_Con
nect.bin
arduino-1.8.19/tools/WiFi101/tool/firmwares/NINA/1.4.5/NINA_W102.bin
arduino-1.8.19/tools/WiFi101/tool/firmwares/NINA/1.4.5/NINA_W102-Uno_WiFi_Rev2.b
in
arduino-1.8.19/tools/WiFi101/tool/firmwares/NINA/1.4.4/
arduino-1.8.19/tools/WiFi101/tool/firmwares/NINA/1.4.4/NINA_W102.bin
arduino-1.8.19/tools/WiFi101/tool/firmwares/NINA/1.4.4/NINA_W102-Uno_WiFi_Rev2.b
in
arduino-1.8.19/tools/howto.txt
nestor@NestorB:~/Descargas$ cd arduino-1.8.19
nestor@NestorB:~/Descargas/arduino-1.8.19$ sudo chmod +x ./install.sh
[sudo] contraseña para nestor:
nestor@NestorB:~/Descargas/arduino-1.8.19$ sudo ./install.sh
Adding desktop shortcut, menu item and file associations for Arduino IDE...
done!
nestor@NestorB:~/Descargas/arduino-1.8.19$
```

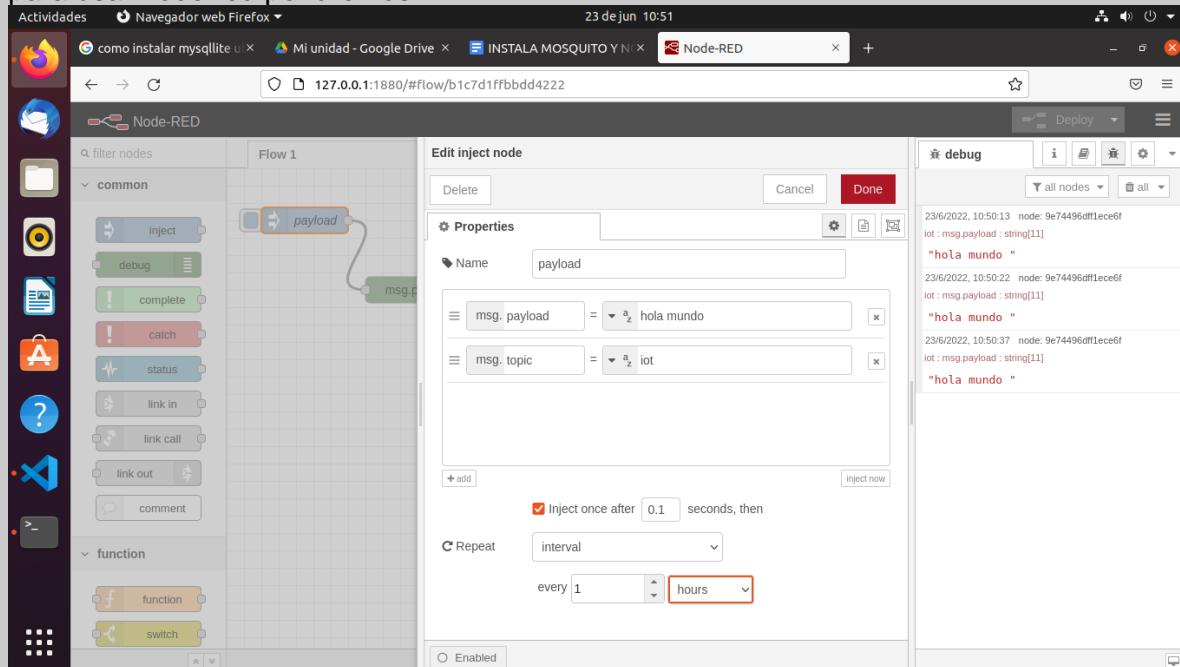
ya está Arduino en nuestro escritorio

instalamos Django como se muestra

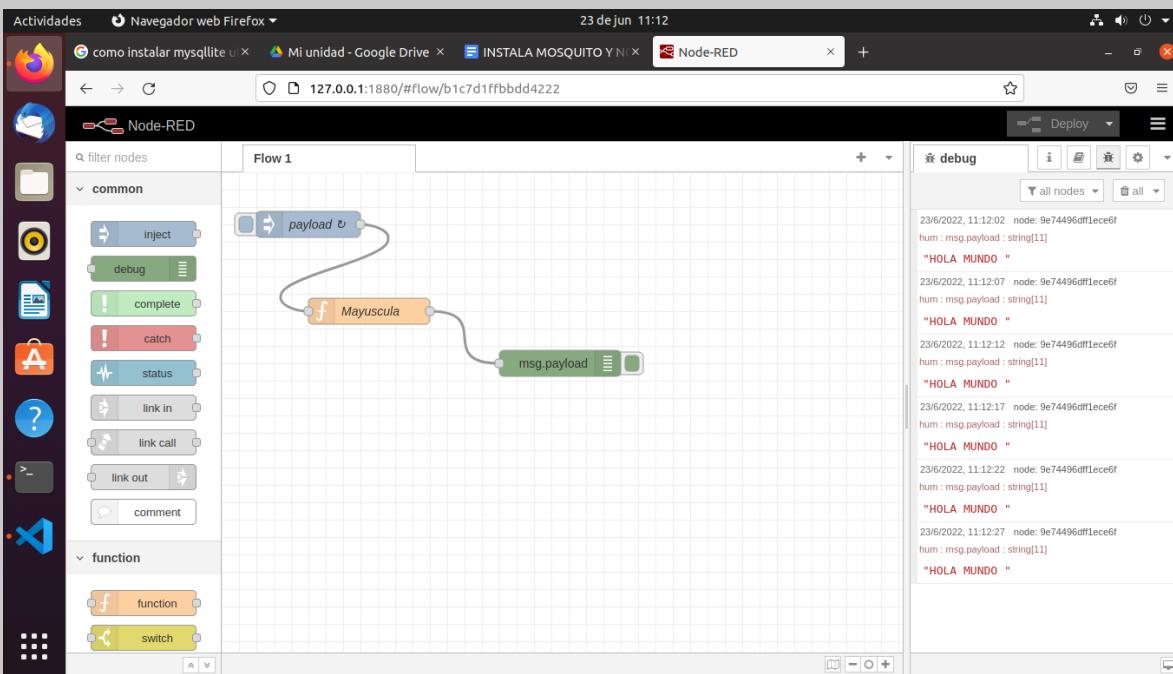
```
nestor@NestorB:~/django/hello-django1$ source .venv/bin/activate
(.venv) nestor@NestorB:~/django/hello-django1$ sudo pip install django
Collecting django
  Downloading Django-4.0.5-py3-none-any.whl (8.0 MB)
    8.0 MB 1.6 MB/s
Collecting backports.zoneinfo; python_version < "3.9"
  Downloading backports.zoneinfo-0.2.1-cp38-cp38-manylinux1_x86_64.whl (74 kB)
Collecting sqlparse>=0.2.2
  Downloading sqlparse-0.4.2-py3-none-any.whl (42 kB)
Collecting asgiref<4,>=3.4.1
  Downloading asgiref-3.5.2-py3-none-any.whl (22 kB)
Installing collected packages: backports.zoneinfo, sqlparse, asgiref, django
Successfully installed asgiref-3.5.2 backports.zoneinfo-0.2.1 django-4.0.5 sqlpa
rse-0.4.2
(.venv) nestor@NestorB:~/django/hello-django1$
```

instalamos django como se muestra

para usar node red pondremos



tenemos todas las configuraciones de todos los sensores o partes del programa que queramos usar



pay load guarda el mensaje que queramos poner
la función nos dice que hacer o en este caso cambiar de minúsculas a mayúsculas
y el último es las veces que se repetirá el mensaje.

NECESITAREMOS SQL Y SQLITE APSRA ESTO UTILIZAREMOS LOS SIGUIENTES COMANDOS

sudo apt update

sudo apt upgrade

```
sudo apt install mysql-server
```

abrimos el programa con _mysql

The screenshot shows a Linux desktop environment with a terminal window open. The terminal window title is "root@NestorB: /home/nestor". The terminal content includes:

```
root@NestorB: /home/nestor
root@NestorB: ~
Sé encontraron errores al procesar:
  docker-ce
E: Sub-process /usr/bin/dpkg returned an error code (1)
root@NestorB:/home/nestor# mysql
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 8
Server version: 8.0.29-0ubuntu0.20.04.3 (Ubuntu)

Copyright (c) 2000, 2022, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> quit
Bye
root@NestorB:/home/nestor# sudo apt update
Err:1 http://mx.archive.ubuntu.com/ubuntu focal InRelease
  Fallo temporal al resolver «mx.archive.ubuntu.com»
  Obj:1:2 http://security.ubuntu.com/ubuntu focal-security InRelease
  Obj:3 https://download.docker.com/linux/ubuntu focal InRelease
  Obj:4 http://mx.archive.ubuntu.com/ubuntu focal-updates InRelease
```

TUTORIAL DJANGO

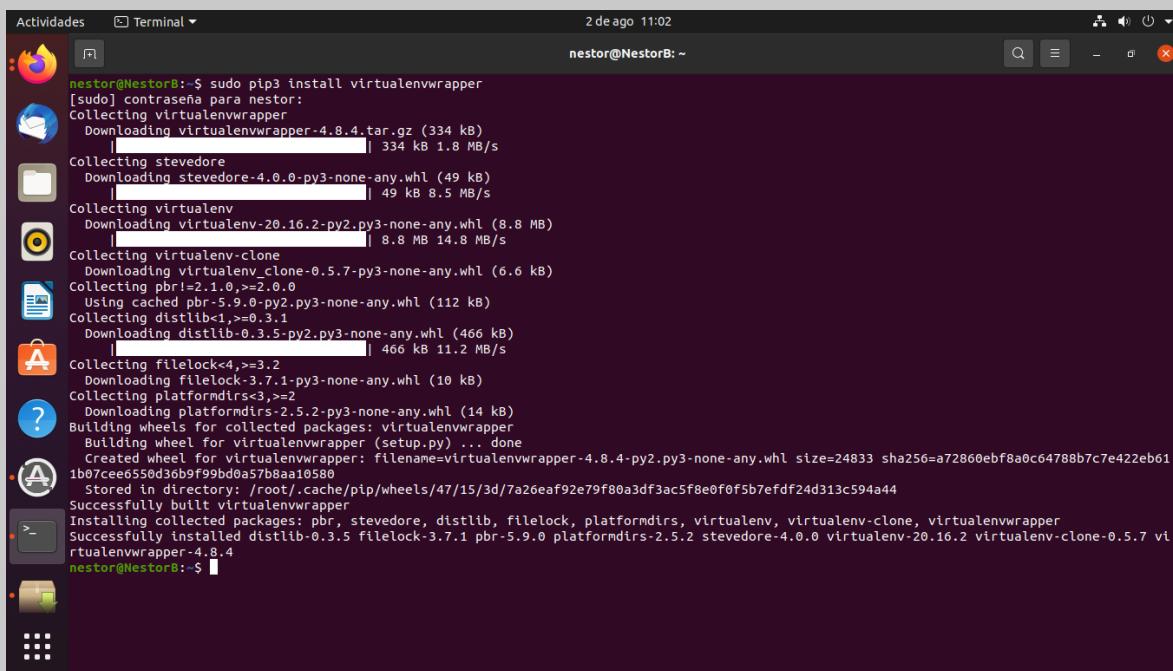
Creación de entorno virtual

- Conocer la versión de Python.

```
python3 -v
```

- Abrir una terminal

```
sudo pip3 install virtualenvwrapper
```



```
nestor@NestorB:~$ sudo pip3 install virtualenvwrapper
[sudo] contraseña para nestor:
Collecting virtualenvwrapper
  Downloading virtualenvwrapper-4.8.4.tar.gz (334 kB)
    ━━━━━━━━━━━━━━━━ 334 kB 1.8 MB/s
Collecting stevedore
  Downloading stevedore-4.0.0-py3-none-any.whl (49 kB)
    ━━━━━━━━━━━━━━ 49 kB 8.5 MB/s
Collecting virtualenv
  Downloading virtualenv-20.16.2-py2.py3-none-any.whl (8.8 MB)
    ━━━━━━━━━━━━━━ 8.8 MB 14.8 MB/s
Collecting virtualenv-clone
  Downloading virtualenv_clone-0.5.7-py3-none-any.whl (6.6 kB)
Collecting pbr!=2.1.0,>=2.0.0
  Using cached pbr-5.9.0-py2.py3-none-any.whl (112 kB)
Collecting distlib<1,>=0.3.1
  Downloading distlib-0.3.5-py2.py3-none-any.whl (466 kB)
    ━━━━━━━━━━━━━━ 466 kB 11.2 MB/s
Collecting filelock<4,>=3.2
  Downloading filelock-3.7.1-py3-none-any.whl (10 kB)
Collecting platformdirs<3,>=2
  Downloading platformdirs-2.5.2-py3-none-any.whl (14 kB)
Building wheels for collected packages: virtualenvwrapper
  Building wheel for virtualenvwrapper (setup.py) ... done
    Created wheel for virtualenvwrapper: filename=virtualenvwrapper-4.8.4-py2.py3-none-any.whl size=24833 sha256=a72860ebf8a0c64788b7c7e422eb61
1b07cee6550d33b09f99bd0a57b8aa1058a
  Stored in directory: /root/.cache/pip/wheels/47/15/3d/7a26eaf92e79f80a3df3ac5f8e0f0f5b7efdf24d313c594a44
Successfully built virtualenvwrapper
Installing collected packages: pbr, stevedore, distlib, filelock, platformdirs, virtualenv, virtualenv-clone, virtualenvwrapper
Successfully installed distlib-0.3.5 filelock-3.7.1 pbr-5.9.0 platformdirs-2.5.2 stevedore-4.0.0 virtualenv-20.16.2 virtualenv-clone-0.5.7 virtualenvwrapper-4.8.4
nestor@NestorB:~$
```

- Modificar el archivo ~/.bashrc

```
sudo nano ~/.bashrc
```

```
Actividades Terminal 2 de ago 11:03
nestor@NestorB: ~
GNU nano 4.8 /home/nestor/.bashrc
#!/bin/bash
# This file is executed on login by the non-interactive shell.
# See /usr/share/doc/bash/examples/startup-files (in the package bash-doc)
# for examples

# If not running interactively, don't do anything
case $- in
    *t*);;
    *) return;;
esac

# don't put duplicate lines or lines starting with space in the history.
# See bash(1) for more options
HISTCONTROL=ignoreboth

# append to the history file, don't overwrite it
shopt -s histappend

# for setting history length see HISTSIZE and HISTFILESIZE in bash(1)
HISTSIZE=1000
HISTFILESIZE=2000

# check the window size after each command and, if necessary,
# update the values of LINES and COLUMNS.
shopt -s checkwinsize

# If set, the pattern "##" used in a pathname expansion context will
# match all files and zero or more directories and subdirectories.
#shopt -s globstar

# make less more friendly for non-text input files, see lesspipe(1)
[ -x /usr/bin/lesspipe ] && eval "$(SHELL=/bin/sh lesspipe)"

# set variable identifying the chroot you work in (used in the prompt below)
if [ -z "${debian_chroot:-}" ] && [ -r /etc/debian_chroot ]; then
    debian_chroot=$(cat /etc/debian_chroot)
fi

^G Ver ayuda ^O Guardar ^W Buscar ^K Cortar Texto ^J Justificar ^C Posición ^M-D Deshacer ^M-A Marcar texto ^M-] A llave
^X Salir ^R Leer fich. ^A Reemplazar ^U Pegar ^T Ortografía ^I Ir a línea ^M-B Rehacer ^M-G Copiar ^Q Buscar atrás
```

Agregar las siguientes líneas

```
export WORKON_HOME=$HOME/.virtualenvs
export VIRTUALENVWRAPPER_PYTHON=/usr/bin/python3
export PROJECT_HOME=$HOME/Devel
source /usr/local/bin/virtualenvwrapper.sh
```

```
Actividades Terminal 2 de ago 11:06
nestor@NestorB: ~
GNU nano 4.8 /home/nestor/.bashrc
#!/bin/bash
# This file is executed on login by the non-interactive shell.
# See /usr/share/doc/bash/examples/startup-files (in the package bash-doc)
# for examples
export WORKON_HOME=$HOME/.virtualenvs
export VIRTUALENVWRAPPER_PYTHON=/usr/bin/python3
export PROJECT_HOME=$HOME/Devel
source /usr/local/bin/virtualenvwrapper.sh

# If not running interactively, don't do anything
case $- in
    *t*);;
    *) return;;
esac

# don't put duplicate lines or lines starting with space in the history.
# See bash(1) for more options
HISTCONTROL=ignoreboth

# append to the history file, don't overwrite it
shopt -s histappend

# for setting history length see HISTSIZE and HISTFILESIZE in bash(1)
HISTSIZE=1000
HISTFILESIZE=2000

# check the window size after each command and, if necessary,
# update the values of LINES and COLUMNS.
shopt -s checkwinsize

# If set, the pattern "##" used in a pathname expansion context will
# match all files and zero or more directories and subdirectories.
#shopt -s globstar

# make less more friendly for non-text input files, see lesspipe(1)
[ 121 líneas leidas ]
^G Ver ayuda ^O Guardar ^W Buscar ^K Cortar Texto ^J Justificar ^C Posición ^M-D Deshacer ^M-A Marcar texto ^M-] A llave
^X Salir ^R Leer fich. ^A Reemplazar ^U Pegar ^T Ortografía ^I Ir a línea ^M-B Rehacer ^M-G Copiar ^Q Buscar atrás
```

- volver a cargar

```

source ~/.bashrc
Actividades Terminal 2 de ago 11:07
nestor@NestorB: ~
Collecting virtualenv-clone
  Downloading virtualenv_clone-0.5.7-py3-none-any.whl (6.6 kB)
Collecting pbr!=2.1.0,>=2.0.0
  Using cached pbr-5.9.0-py2.py3-none-any.whl (112 kB)
Collecting distlib<1,>=0.3.1
  Downloading distlib-0.3.5-py2.py3-none-any.whl (466 kB)
    |
    ||██████████| 466 kB 11.2 MB/s
Collecting filelock<4,>=3.2
  Downloading filelock-3.7.1-py3-none-any.whl (10 kB)
Collecting platformdirs<3,>=2
  Downloading platformdirs-2.5.2-py3-none-any.whl (14 kB)
Building wheels for collected packages: virtualenvwrapper
  Building wheel for virtualenvwrapper (setup.py) ... done
    Created wheel for virtualenvwrapper: filename=virtualenvwrapper-4.8.4-py2.py3-none-any.whl size=24833 sha256=a72860ebf8a0c64788b7c7e422eb61
1b07cee6550d36b9f99bd0a57b8aa10580
  Stored in directory: /root/.cache/pip/wheels/47/15/3d/7a26eaf92e79f80a3df3ac5f8e0f5b7efdf24d313c594a44
Successfully built virtualenvwrapper
Installing collected packages: pbr, stevedore, distlib, filelock, platformdirs, virtualenv, virtualenv-clone, virtualenvwrapper
Successfully installed distlib-0.3.5 filelock-3.7.1 pbr-5.9.0 platformdirs-2.5.2 stevedore-4.0.0 virtualenv-20.16.2 virtualenv-clone-0.5.7 virtualenvwrapper-4.8.4
nestor@NestorB: $ sudo nano ~/.bashrc
nestor@NestorB: $ sudo nano ~/.bashrc
nestor@NestorB: $ volver a cargar
volver: no se encontró la orden
nestor@NestorB: $ source ~/.bashrc
virtualenvwrapper.user_scripts creating /home/nestor/.virtualenvs/premkproject
virtualenvwrapper.user_scripts creating /home/nestor/.virtualenvs/postmkproject
virtualenvwrapper.user_scripts creating /home/nestor/.virtualenvs/initialize
virtualenvwrapper.user_scripts creating /home/nestor/.virtualenvs/premkvirtualenv
virtualenvwrapper.user_scripts creating /home/nestor/.virtualenvs/postmkvirtualenv
virtualenvwrapper.user_scripts creating /home/nestor/.virtualenvs/prermvirtualenv
virtualenvwrapper.user_scripts creating /home/nestor/.virtualenvs/postrmvirtualenv
virtualenvwrapper.user_scripts creating /home/nestor/.virtualenvs/predeactivate
virtualenvwrapper.user_scripts creating /home/nestor/.virtualenvs/postdeactivate
virtualenvwrapper.user_scripts creating /home/nestor/.virtualenvs/preactivate
virtualenvwrapper.user_scripts creating /home/nestor/.virtualenvs/postactivate
virtualenvwrapper.user_scripts creating /home/nestor/.virtualenvs/get_env_details
nestor@NestorB: $ 
```

- Crear un directorio de trabajo
- En la terminal ejecutar el siguiente comando:

`mkvirtualenv apliweb_virtualenv`

```

Actividades Terminal 2 de ago 11:13
nestor@NestorB: ~/ProyectoParedVerde
ib07cee6550d36b9f99bd0a57b8aa10580
  Stored in directory: /root/.cache/pip/wheels/47/15/3d/7a26eaf92e79f80a3df3ac5f8e0f5b7efdf24d313c594a44
Successfully built virtualenvwrapper
Successfully installed distlib-0.3.5 filelock-3.7.1 pbr-5.9.0 platformdirs-2.5.2 stevedore-4.0.0 virtualenv-20.16.2 virtualenv-clone-0.5.7 virtualenvwrapper-4.8.4
nestor@NestorB: $ sudo nano ~/.bashrc
nestor@NestorB: $ sudo nano ~/.bashrc
nestor@NestorB: $ volver a cargar
volver: no se encontró la orden
nestor@NestorB: $ source ~/.bashrc
virtualenvwrapper.user_scripts creating /home/nestor/.virtualenvs/premkproject
virtualenvwrapper.user_scripts creating /home/nestor/.virtualenvs/postmkproject
virtualenvwrapper.user_scripts creating /home/nestor/.virtualenvs/initialize
virtualenvwrapper.user_scripts creating /home/nestor/.virtualenvs/premkvirtualenv
virtualenvwrapper.user_scripts creating /home/nestor/.virtualenvs/postmkvirtualenv
virtualenvwrapper.user_scripts creating /home/nestor/.virtualenvs/prermvirtualenv
virtualenvwrapper.user_scripts creating /home/nestor/.virtualenvs/postrmvirtualenv
virtualenvwrapper.user_scripts creating /home/nestor/.virtualenvs/predeactivate
virtualenvwrapper.user_scripts creating /home/nestor/.virtualenvs/postdeactivate
virtualenvwrapper.user_scripts creating /home/nestor/.virtualenvs/preactivate
virtualenvwrapper.user_scripts creating /home/nestor/.virtualenvs/postactivate
virtualenvwrapper.user_scripts creating /home/nestor/.virtualenvs/get_env_details
nestor@NestorB: $ cd ProyectoParedVerde
nestor@NestorB:~/ProyectoParedVerde$ mkvirtualenv apliweb_virtualenv
created virtual environment CPython3.8.10.final.0-64 in 1487ms
  creator CPython3Posix(dest=/home/nestor/.virtualenvs/apliweb_virtualenv, clear=False, no_vcs_ignore=False, global=False)
  seeder FromAppdata(download=False, pip=bundle, setuptools=bundle, wheel=bundle, via_copy, app_data_dir=/home/nestor/.local/share/virtualenv )
    added seed packages: pip==22.2.1, setuptools==63.2.0, wheel==0.37.1
  activators BashActivator,CShellActivator,FishActivator,NushellActivator,PowerShellActivator,PythonActivator
virtualenvwrapper.user_scripts creating /home/nestor/.virtualenvs/apliweb_virtualenv/bin/predeactivate
virtualenvwrapper.user_scripts creating /home/nestor/.virtualenvs/apliweb_virtualenv/bin/postdeactivate
virtualenvwrapper.user_scripts creating /home/nestor/.virtualenvs/apliweb_virtualenv/bin/preactivate
virtualenvwrapper.user_scripts creating /home/nestor/.virtualenvs/apliweb_virtualenv/bin/postactivate
virtualenvwrapper.user_scripts creating /home/nestor/.virtualenvs/apliweb_virtualenv/bin/get_env_details
(apliweb_virtualenv) nestor@NestorB:~/ProyectoParedVerde$ ^C
(apliweb_virtualenv) nestor@NestorB:~/ProyectoParedVerde$ 
```

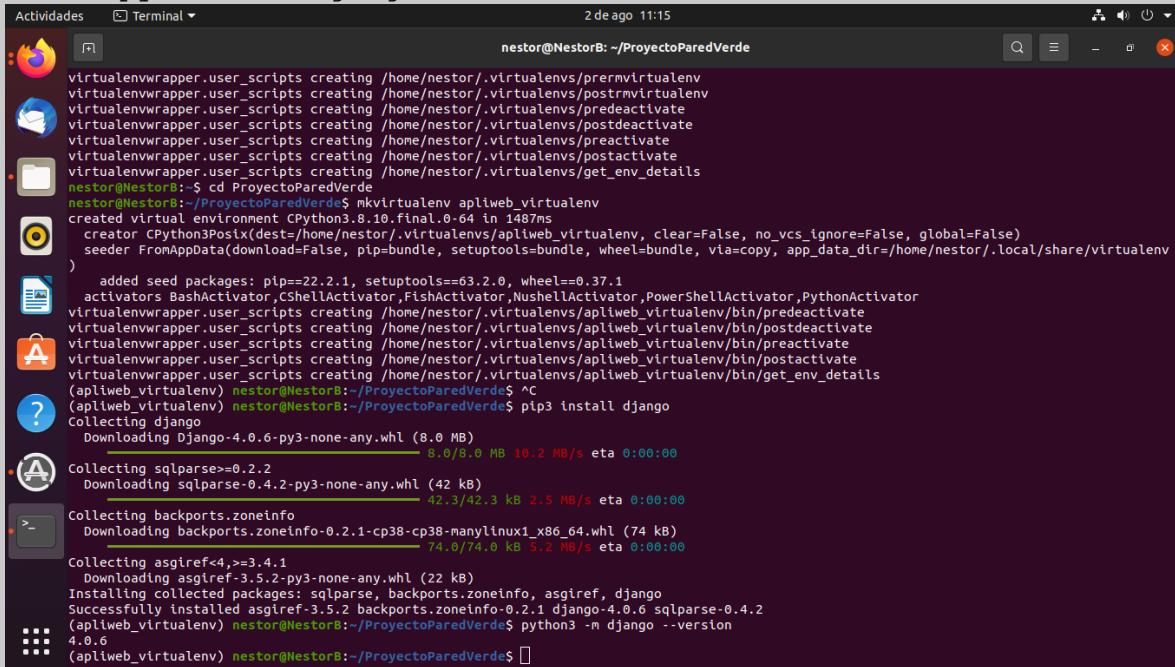
Creando Proyecto Django

- Instalar módulo Django

```
pip3 install Django
```

- verificar de instalación de módulo Django

```
python3 -m Django --version
```



```
nestor@NestorB: ~/ProyectoParedVerde
nestor@NestorB:~/ProyectoParedVerde$ python3 -m Django --version
Django 4.0.6
nestor@NestorB:~/ProyectoParedVerde$
```

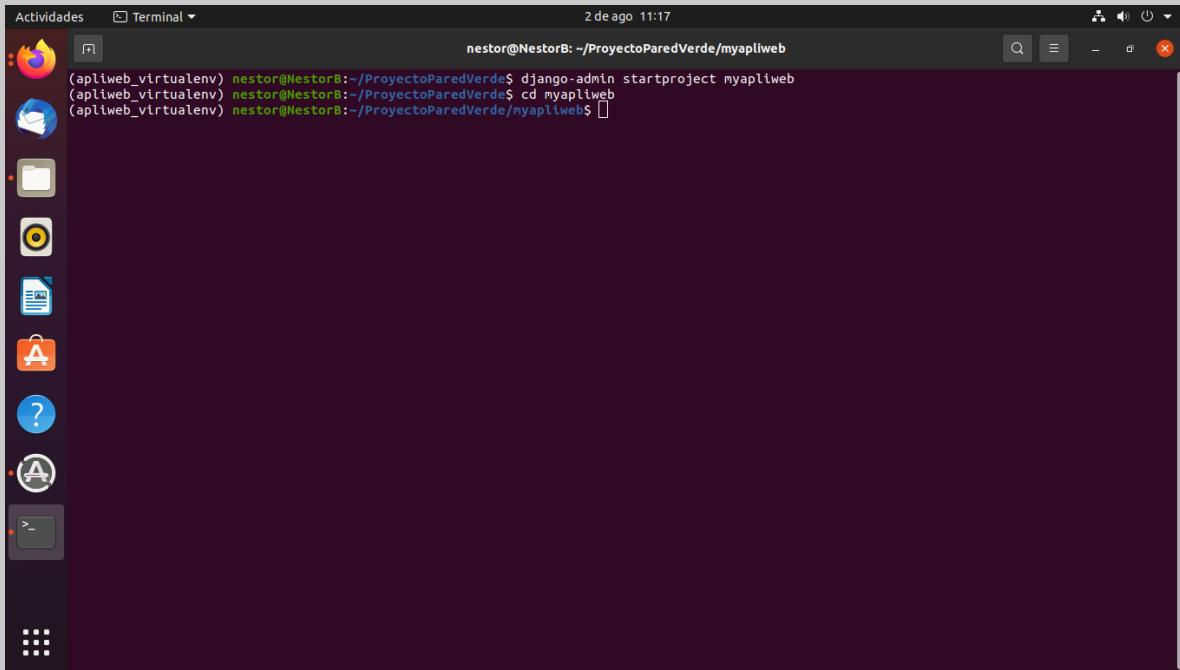
El proceso es sencillo:

- Usar la herramienta **django-admin** para crear la carpeta del proyecto, los ficheros de plantillas básicos y el script de gestión del proyecto (**manage.py**).
- Usar **manage.py** para crear una o más *aplicaciones*.
- Registrar las nuevas aplicaciones para incluirlas en el proyecto.
- Conectar el mapeado url de cada aplicación.

Crear el esqueleto del proyecto

Abrir una terminal y ejecutar los siguientes comandos

```
django-admin startproject myapiweb
cd myapiweb
```



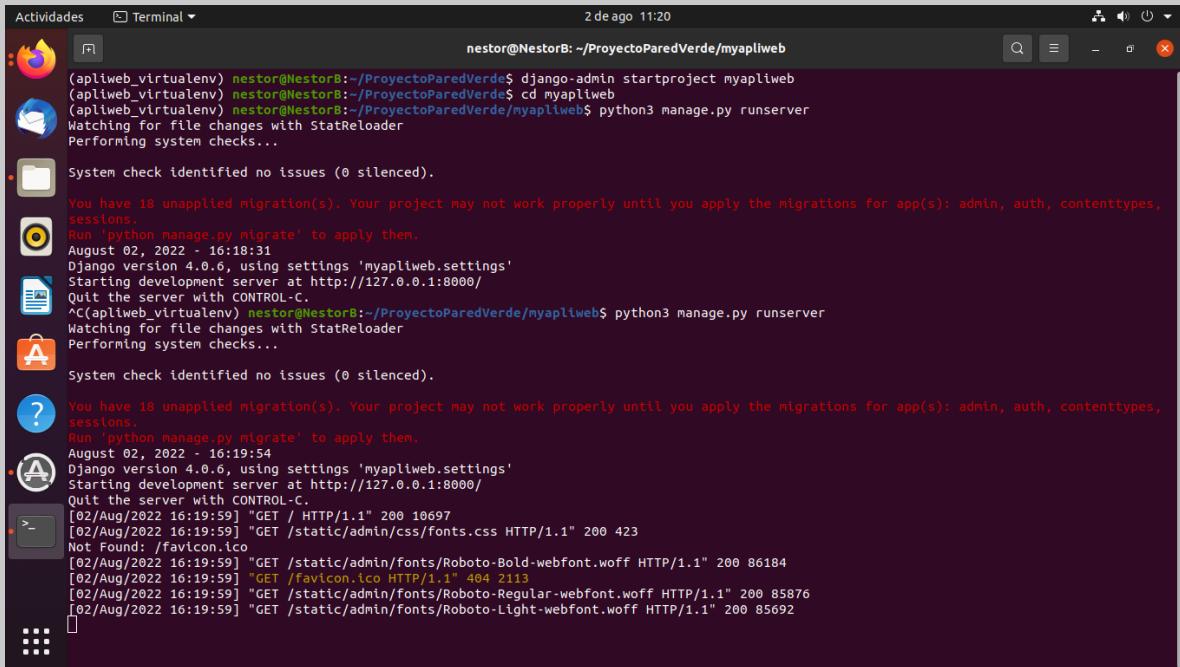
Actividades Terminal 2 de ago 11:17

```
(apliweb_virtualenv) nestor@NestorB:~/ProyectoParedVerde/myapiweb$ django-admin startproject myapiweb
(apliweb_virtualenv) nestor@NestorB:~/ProyectoParedVerde$ cd myapiweb
(apliweb_virtualenv) nestor@NestorB:~/ProyectoParedVerde/myapiweb$ 
```

Ejecutar orden para levantar el servidor

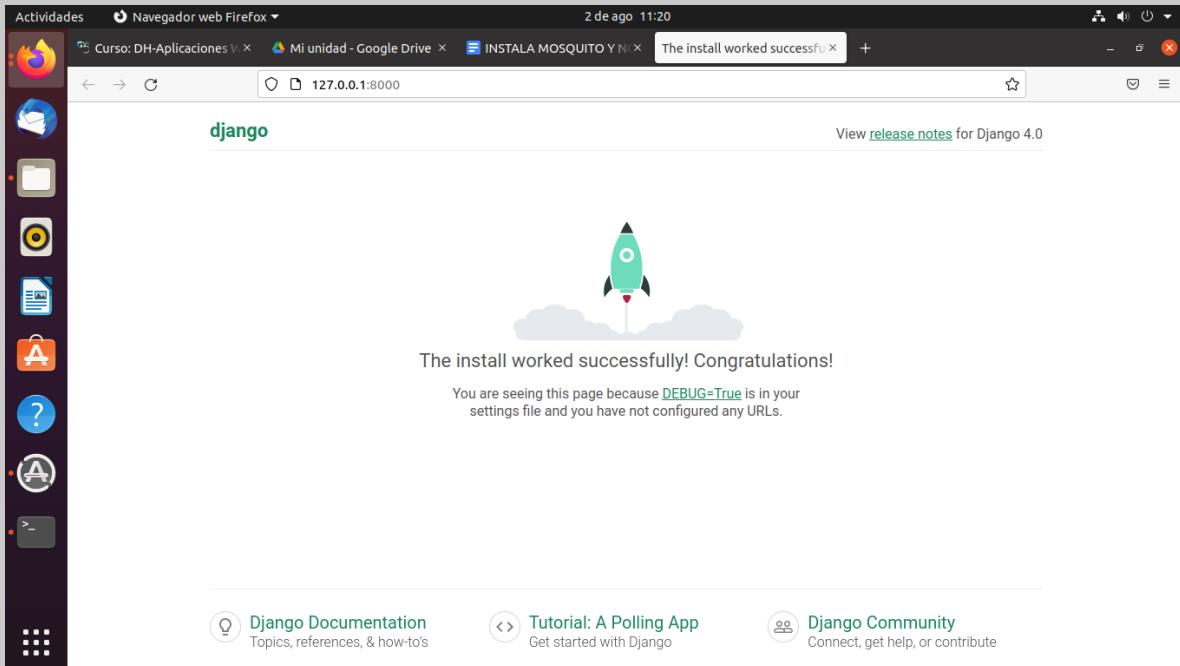
```
python3 manage.py runserver
```

Ingresar a la URL <http://127.0.0.1/8000>



Actividades Terminal 2 de ago 11:20

```
(apliweb_virtualenv) nestor@NestorB:~/ProyectoParedVerde/myapiweb$ django-admin startproject myapiweb
(apliweb_virtualenv) nestor@NestorB:~/ProyectoParedVerde$ cd myapiweb
(apliweb_virtualenv) nestor@NestorB:~/ProyectoParedVerde/myapiweb$ python3 manage.py runserver
Watching for file changes with StatReloader
Performing system checks...
System check identified no issues (0 silenced).
You have 18 unapplied migration(s). Your project may not work properly until you apply the migrations for app(s): admin, auth, contenttypes, sessions.
Run 'python manage.py migrate' to apply them.
August 02, 2022 - 16:18:31
Django version 4.0.6, using settings 'myapiweb.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CONTROL-C.
^C(apliweb_virtualenv) nestor@NestorB:~/ProyectoParedVerde/myapiweb$ python3 manage.py runserver
Watching for file changes with StatReloader
Performing system checks...
System check identified no issues (0 silenced).
You have 18 unapplied migration(s). Your project may not work properly until you apply the migrations for app(s): admin, auth, contenttypes, sessions.
Run 'python manage.py migrate' to apply them.
August 02, 2022 - 16:19:54
Django version 4.0.6, using settings 'myapiweb.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CONTROL-C.
[02/Aug/2022 16:19:59] "GET / HTTP/1.1" 200 10697
[02/Aug/2022 16:19:59] "GET /static/admin/css/fonts.css HTTP/1.1" 200 423
Not Found: /favicon.ico
[02/Aug/2022 16:19:59] "GET /static/admin/fonts/Roboto-Bold-webfont.woff HTTP/1.1" 200 86184
[02/Aug/2022 16:19:59] "GET /favicon.ico HTTP/1.1" 404 2113
[02/Aug/2022 16:19:59] "GET /static/admin/fonts/Roboto-Regular-webfont.woff HTTP/1.1" 200 85876
[02/Aug/2022 16:19:59] "GET /static/admin/fonts/Roboto-Light-webfont.woff HTTP/1.1" 200 85692
```



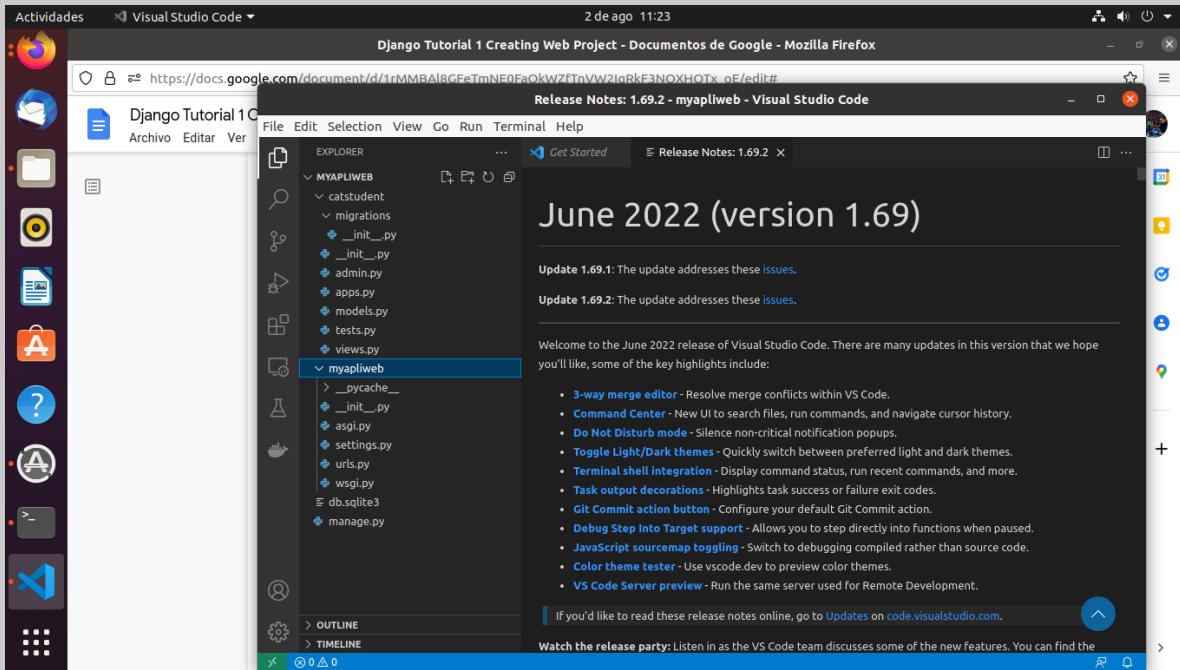
Creación de la aplicación student

Ejecutar la instrucción en la terminal

```
python3 manage.py startapp catstudent
```

A screenshot of a terminal window titled 'Terminal'. The command 'python3 manage.py startapp catstudent' is run, and the terminal output shows the creation of the 'catstudent' application within the project directory. It includes messages about migrations, system checks, and the starting of the development server at 'http://127.0.0.1:8000'.

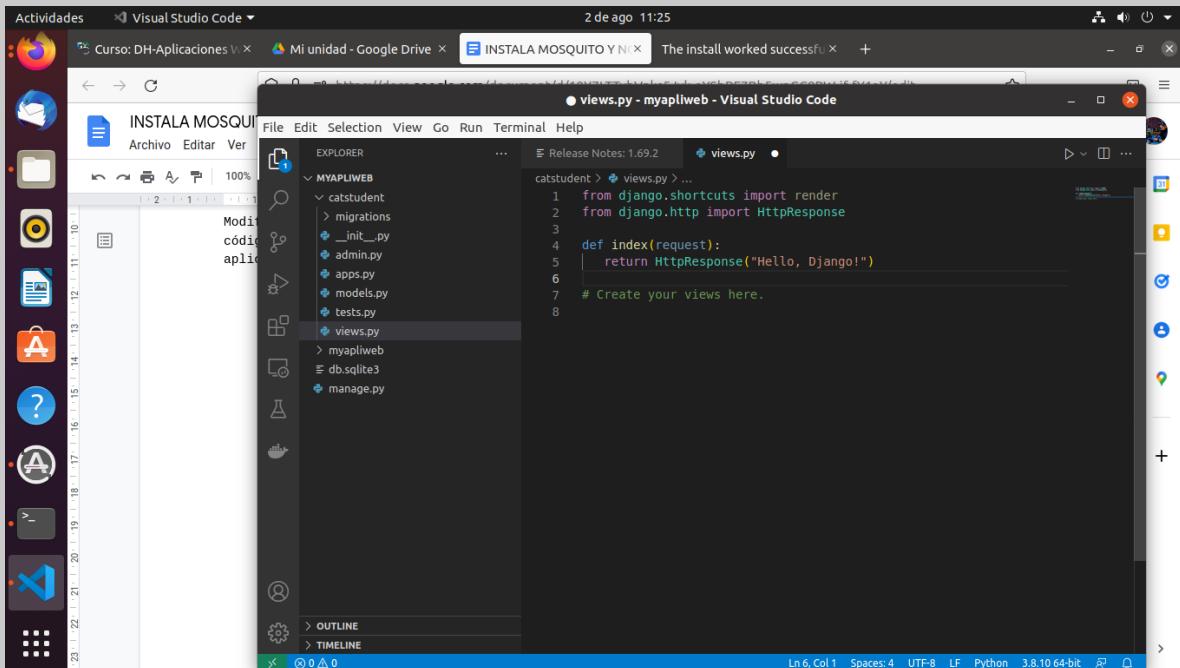
ponemos después el comando code . y abrirá visual studio code



Modifique **catstudent/views.py** para que coincida con el siguiente código, crea una vista única para la página de inicio de la aplicación:

```
from django.http import HttpResponseRedirect

def index(request):
    return HttpResponseRedirect("Hello, Django!")
```

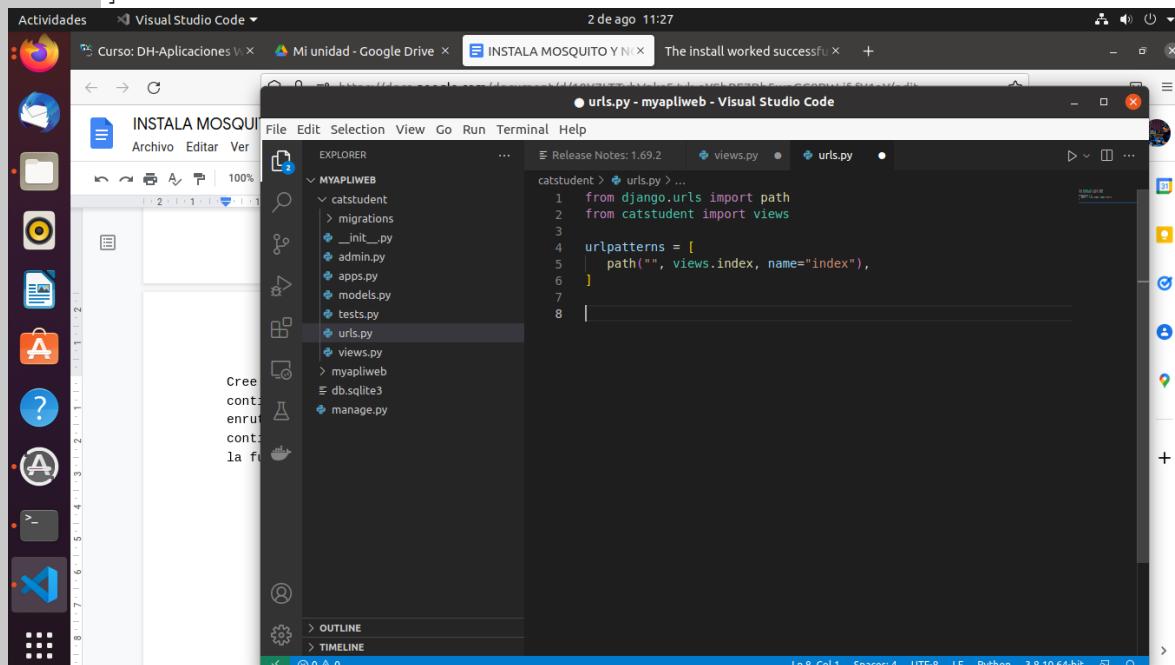


Cree un archivo, **catstudent/urls.py**, con el contenido a continuación. El archivo **urls.py** es donde especifica patrones para

enrutar diferentes URL a sus vistas apropiadas. El siguiente código contiene una ruta para asignar la URL raíz de la aplicación ("") a la función `views.index` que acaba de agregar a `catstudent/views.py`:

```
from django.urls import path
from catstudent import views

urlpatterns = [
    path("", views.index, name="index"),
]
```

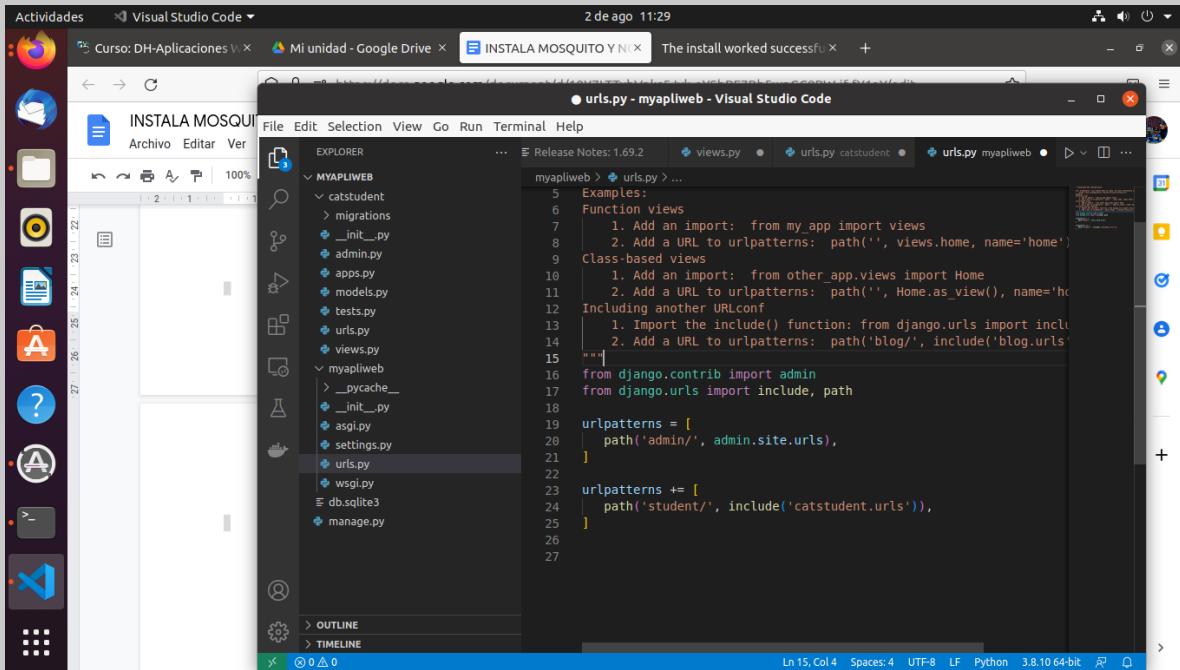


La carpeta `myapliweb` también contiene un archivo `urls.py`, que es donde realmente se maneja el enrutamiento de URL. Abra `myapliweb/urls.py` y modificalo para que coincida con el siguiente código. Este código extrae `catstudent/urls.py` de la aplicación usando `django.urls.include`, que mantiene las rutas de la aplicación contenidas dentro de la aplicación. Esta separación es útil cuando un proyecto contiene varias aplicaciones.

```
from django.contrib import admin
from django.urls import include, path

urlpatterns = [
    path('admin/', admin.site.urls),
]

urlpatterns += [
    path('student/', include('catstudent.urls')),
]
```



Registrar la aplicación

Abre el fichero de ajustes del proyecto **myapliweb/myapliweb/settings.py** y encuentra la definición de la lista **INSTALLED_APPS**. Añade a continuación una nueva línea al final de la lista, como se muestra en negrilla abajo.

```
INSTALLED_APPS = [  
    'django.contrib.admin',  
    'django.contrib.auth',  
    'django.contrib.contenttypes',  
    'django.contrib.sessions',  
    'django.contrib.messages',  
    'django.contrib.staticfiles',  
    ''catstudent.apps.CatstudentConfig',  
]
```

```

INSTALA MOSQUITO Y NIX
2 de ago 11:31
File Edit Selection View Go Run Terminal Help
EXPLORER
  myapiweb > settings.py ...
  19 # QUICK-START DEVELOPMENT SETTINGS - unsuitable for production
  20 # See https://docs.djangoproject.com/en/4.0/howto/deployment/checklist/
  21
  22 # SECURITY WARNING: keep the secret key used in production secret!
  23 SECRET_KEY = 'django-insecure-d$)^8l6=g-m1+p_ci$hve(iewe5v_qj1'
  24
  25 # SECURITY WARNING: don't run with debug turned on in production!
  26 DEBUG = True
  27
  28 ALLOWED_HOSTS = []
  29
  30 # Application definition
  31 INSTALLED_APPS = [
  32     'django.contrib.admin',
  33     'django.contrib.auth',
  34     'django.contrib.contenttypes',
  35     'django.contrib.sessions',
  36     'django.contrib.messages',
  37     'django.contrib.staticfiles',
  38     'catstudent.apps.CatstudentConfig',
  39 ]
  40
  41
  42 MIDDLEWARE = [
  43     'django.middleware.security.SecurityMiddleware',
  44     'django.contrib.sessions.middleware.SessionMiddleware',
  45     'django.middleware.common.CommonMiddleware',
  46     'django.middleware.csrf.CsrfViewMiddleware',
  47 ]

```

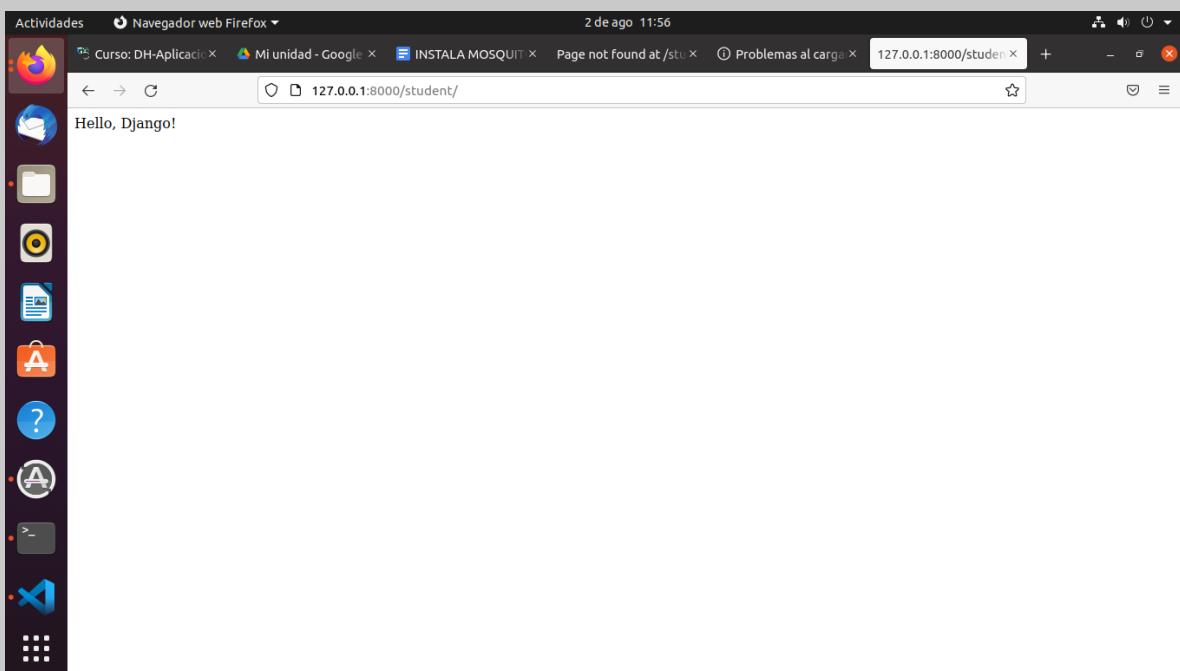
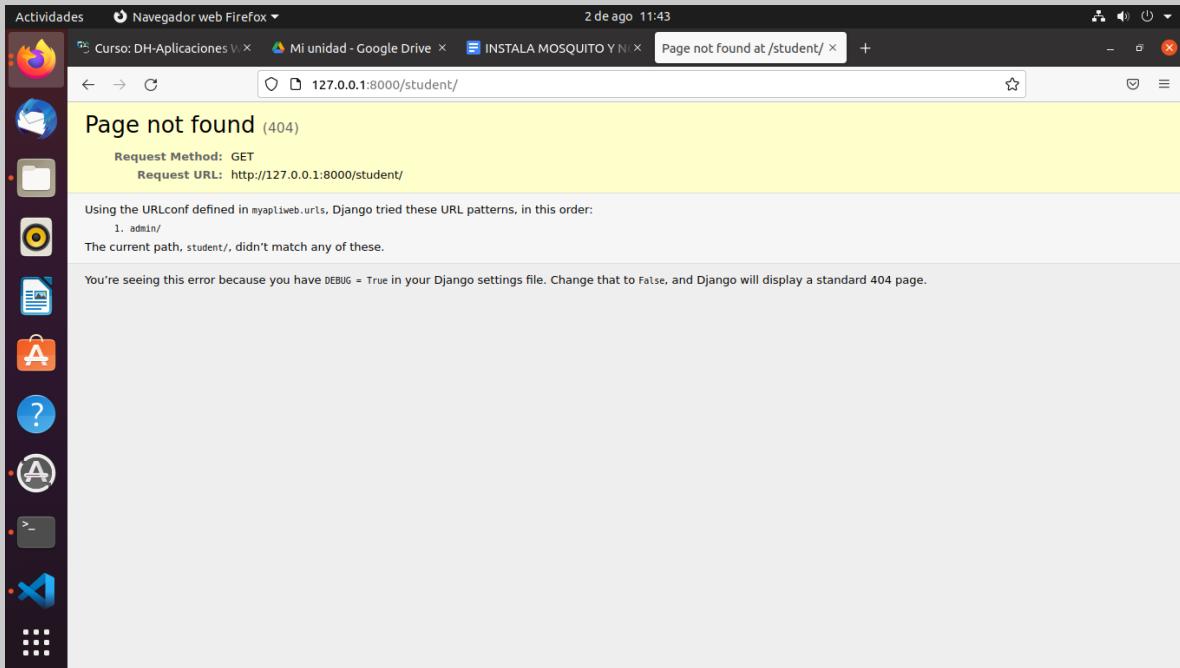
guardamos todo nuestro proyecto con control+s
levantamos el servidor ### python3 manage.py runserver ###

```

nestor@NestorB: ~/ProyectoParedVerde/myapiweb
System check identified no issues (0 silenced).
You have 18 unapplied migration(s). Your project may not work properly until you apply the migrations for app(s): admin, auth, contenttypes, sessions.
Run 'python manage.py migrate' to apply them.
August 02, 2022 - 16:19:54
Django version 4.0.6, using settings 'myapiweb.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CONTROL-C.
[02/Aug/2022 16:19:59] "GET / HTTP/1.1" 200 10697
[02/Aug/2022 16:19:59] "GET /static/admin/css/fonts.css HTTP/1.1" 200 423
Not Found: /favicon.ico
[02/Aug/2022 16:19:59] "GET /static/admin/fonts/Roboto-Bold-webfont.woff HTTP/1.1" 200 86184
[02/Aug/2022 16:19:59] "GET /favicon.ico HTTP/1.1" 404 2113
[02/Aug/2022 16:19:59] "GET /static/admin/fonts/Roboto-Regular-webfont.woff HTTP/1.1" 200 85876
[02/Aug/2022 16:19:59] "GET /static/admin/fonts/Roboto-Light-webfont.woff HTTP/1.1" 200 85692
^C(applweb_virtulenv) nestor@NestorB:~/ProyectoParedVerde/myapiweb$ python3 manage.py startapp catstudent
(applweb_virtulenv) nestor@NestorB:~/ProyectoParedVerde/myapiweb$ code .
(applweb_virtulenv) nestor@NestorB:~/ProyectoParedVerde/myapiweb$ python3 manage.py runserver
Watching for file changes with StatReloader
Performing system checks...
System check identified no issues (0 silenced).
You have 18 unapplied migration(s). Your project may not work properly until you apply the migrations for app(s): admin, auth, contenttypes, sessions.
Run 'python manage.py migrate' to apply them.
August 02, 2022 - 16:37:15
Django version 4.0.6, using settings 'myapiweb.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CONTROL-C.
[02/Aug/2022 16:37:29] "GET / HTTP/1.1" 200 10697
[02/Aug/2022 16:37:30] "GET /static/admin/css/fonts.css HTTP/1.1" 200 423
[02/Aug/2022 16:37:30] "GET /static/admin/fonts/Roboto-Bold-webfont.woff HTTP/1.1" 304 0
Not Found: /favicon.ico
[02/Aug/2022 16:37:30] "GET /static/admin/fonts/Roboto-Regular-webfont.woff HTTP/1.1" 304 0
[02/Aug/2022 16:37:30] "GET /static/admin/fonts/Roboto-Light-webfont.woff HTTP/1.1" 304 0

```

Ingresá la URL <http://127.0.0.1:8000/> y después agregamos a la url <http://127.0.0.1:8000/student/>



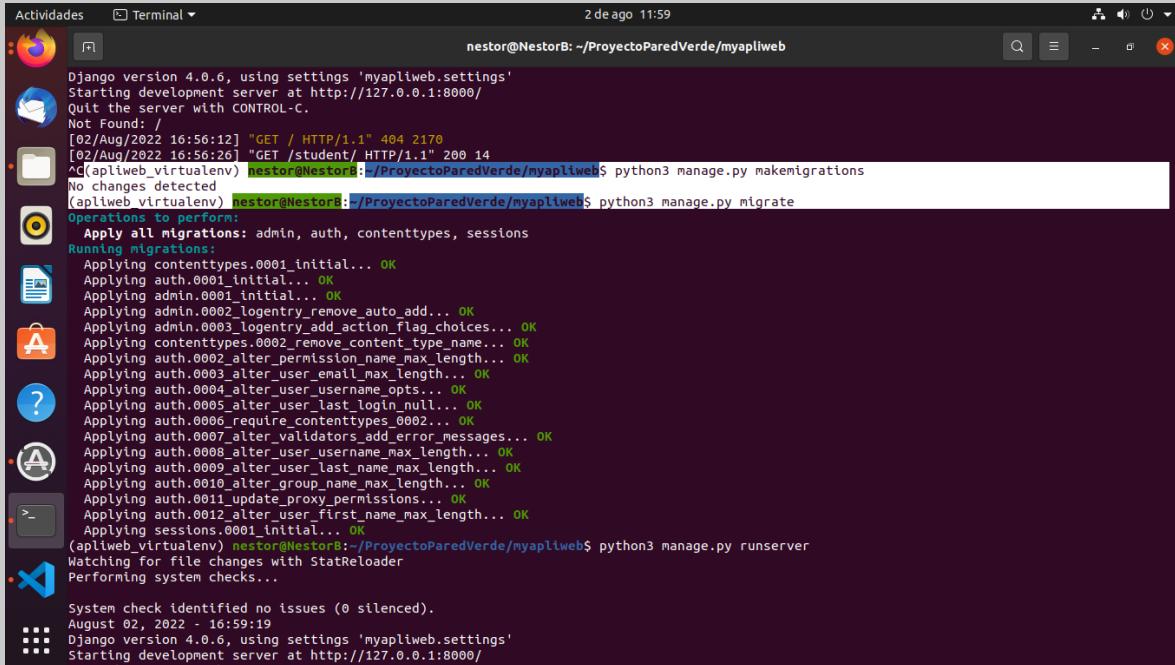
Base de Datos

Django usa un Object-Relational-Mapper (ORM) para mapear las definiciones de Modelos en el código Django con la estructura de datos utilizada por la base de datos subyacente. A medida que cambiamos nuestras definiciones de modelos, Django sigue la pista a los cambios y puede crear scripts de migración de la base de datos (en [/myapliweb/catstudent/migrations/](#)) para migrar

automáticamente la estructura de datos subyacente en la base de datos para igualarse al modelo.

Ejecuta los siguientes comandos

```
python3 manage.py makemigrations  
python3 manage.py migrate
```

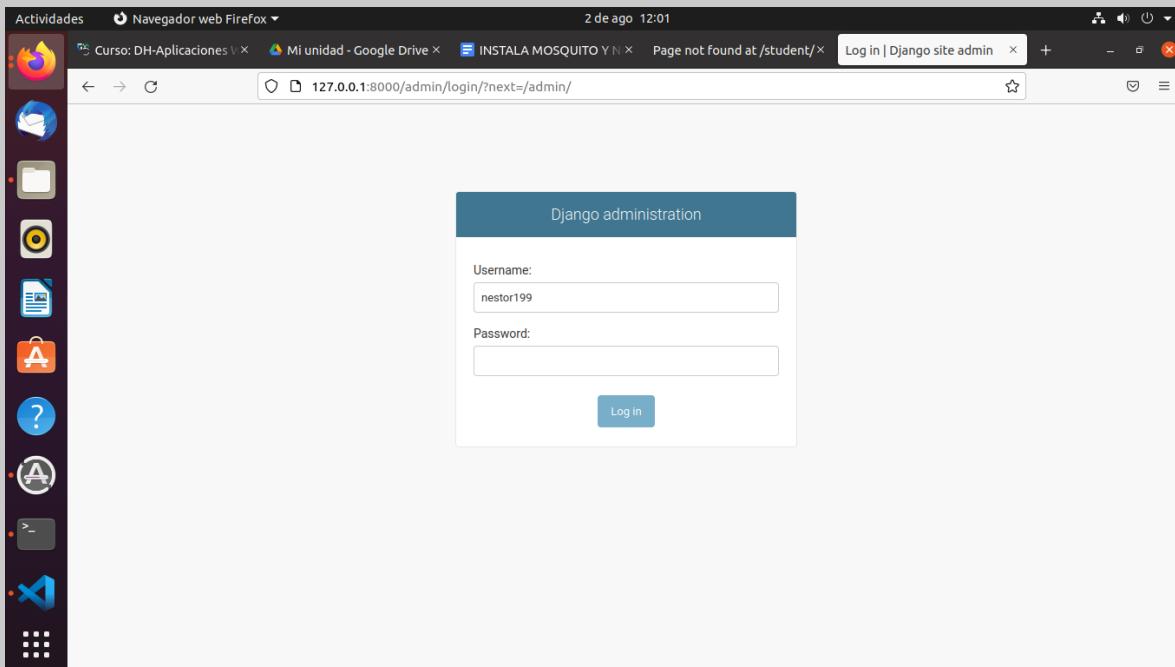


```
Django version 4.0.6, using settings 'myapiweb.settings'  
Starting development server at http://127.0.0.1:8000/  
Quit the server with CONTROL-C.  
Not Found: /  
[02/Aug/2022 16:56:12] "GET / HTTP/1.1" 404 2170  
[02/Aug/2022 16:56:26] "GET /student/ HTTP/1.1" 200 14  
^C[aplweb_virtenv] nestor@NestorB:~/ProyectoParedVerde/myapiweb$ python3 manage.py makemigrations  
No changes detected  
(aplweb_virtenv) nestor@NestorB:~/ProyectoParedVerde/myapiweb$ python3 manage.py migrate  
Operations to perform:  
  Apply all migrations: admin, auth, contenttypes, sessions  
Running migrations:  
  Applying contenttypes.0001_initial... OK  
  Applying auth.0001_initial... OK  
  Applying admin.0001_initial... OK  
  Applying admin.0002_logentry_remove_auto_add... OK  
  Applying auth.0003_logentry_add_action_flag_choices... OK  
  Applying contenttypes.0002_remove_content_type_name... OK  
  Applying auth.0002_alter_permission_name_max_length... OK  
  Applying auth.0003_alter_user_email_max_length... OK  
  Applying auth.0004_alter_user_username_opts... OK  
  Applying auth.0005_alter_user_last_login_null... OK  
  Applying auth.0006_require_contenttypes_0002... OK  
  Applying auth.0007_alter_validators_add_error_messages... OK  
  Applying auth.0008_alter_user_username_max_length... OK  
  Applying auth.0009_alter_user_last_name_max_length... OK  
  Applying auth.0010_alter_group_name_max_length... OK  
  Applying auth.0011_update_proxy_permissions... OK  
  Applying auth.0012_alter_user_first_name_max_length... OK  
  Applying sessions.0001_initial... OK  
(aplweb_virtenv) nestor@NestorB:~/ProyectoParedVerde/myapiweb$ python3 manage.py runserver  
Watching for file changes with StatReloader  
Performing system checks...  
  
System check identified no issues (0 silenced).  
August 02, 2022 - 16:59:19  
Django version 4.0.6, using settings 'myapiweb.settings'  
Starting development server at http://127.0.0.1:8000/  
Quit the server with CONTROL-C.
```

Guardar los archivos y ejecutar la aplicación

```
python3 manage.py runserver
```

Ingresar la url <http://127.0.0.1:8000/admin/>



Crear un usuario

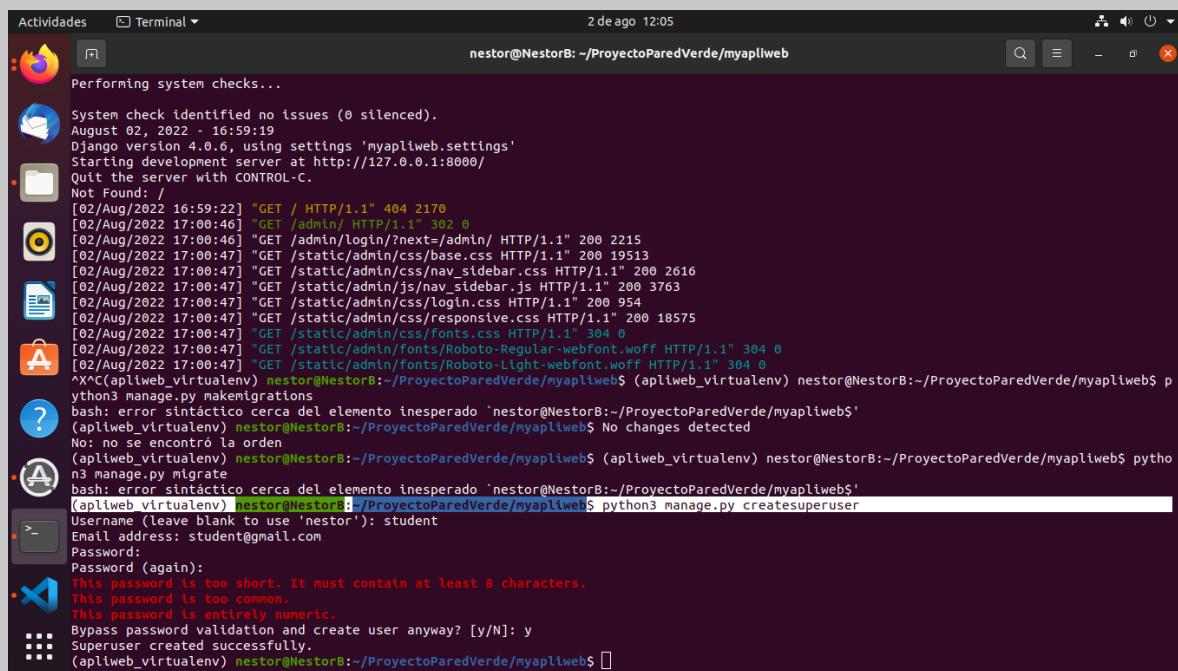
Una vez hecha la migración ejecutar el siguiente comando para crear un nuevo usuario

```
python3 manage.py createsuperuser
```

Ingresá la siguiente información:

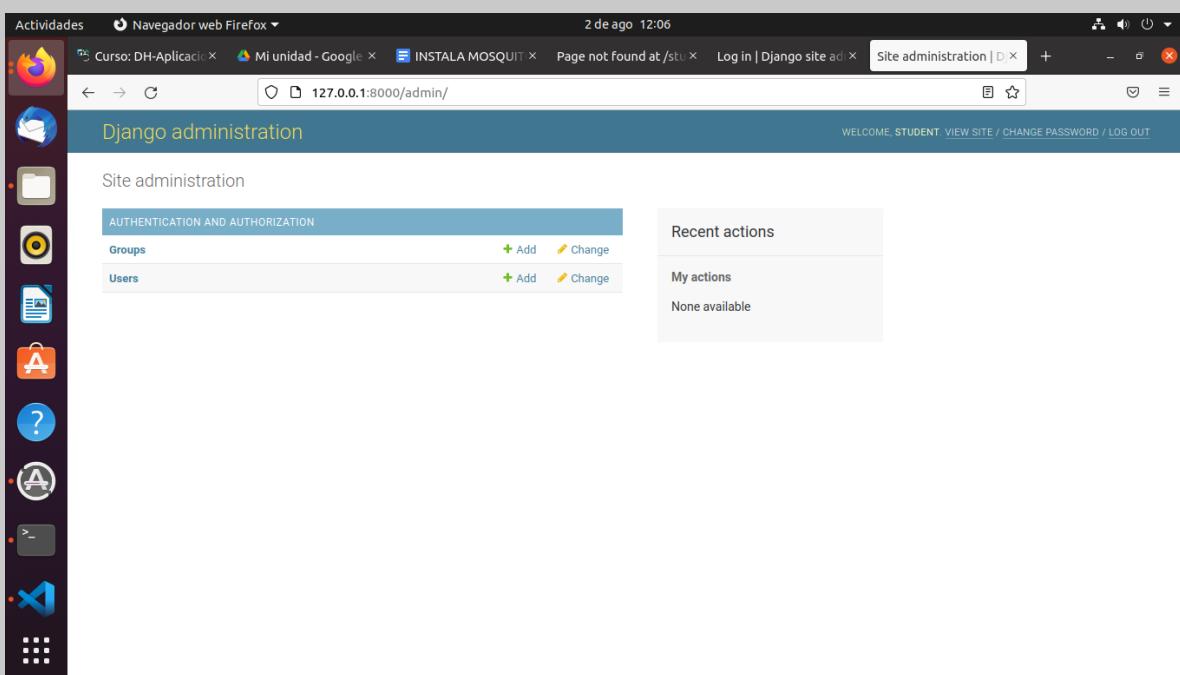
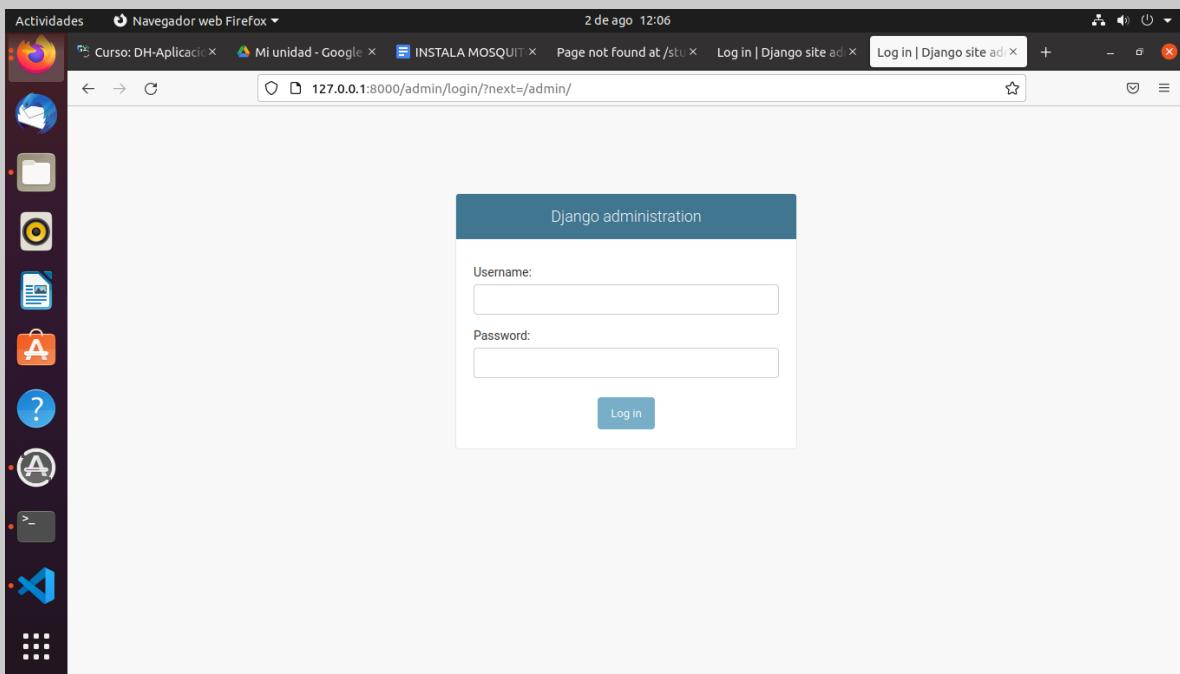
```
username: student
email: student@gmail.com
password: student@01
Aceptas los cambios
Bypass password validation and create user anyway?
[y/N]: y
Superuser created successfully.
```

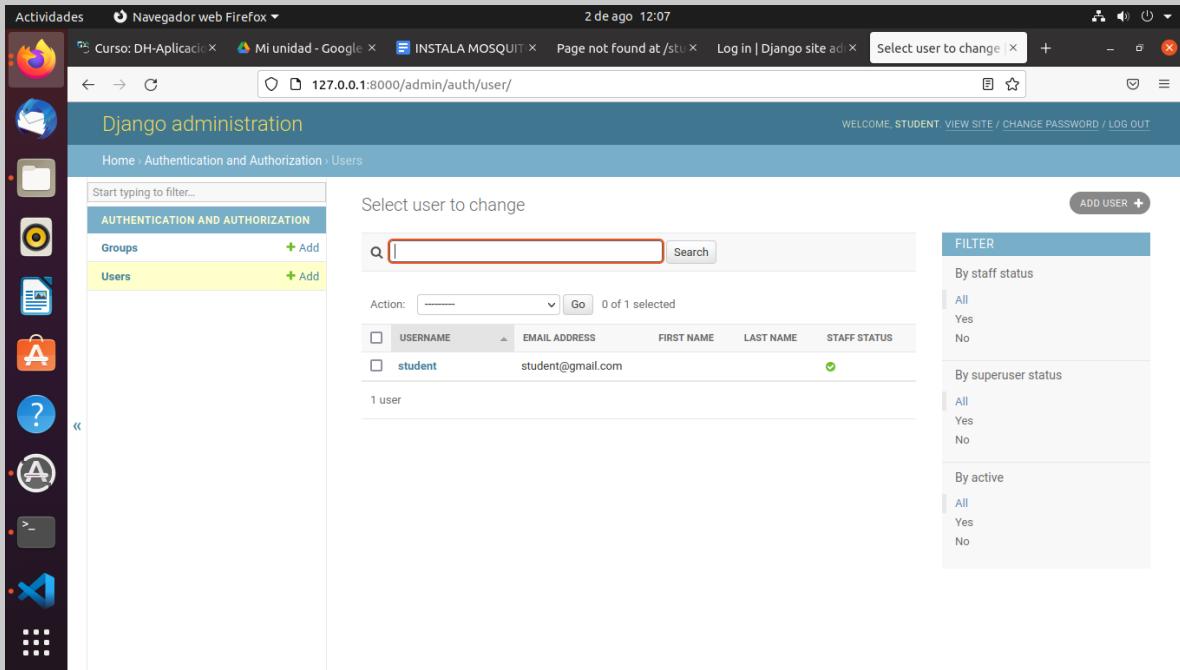
Ingresás a la URL <http://127.0.0.1:8000/admin/login/?next=/admin/>



The screenshot shows a terminal window titled "nestor@NestorB: ~/ProyectoParedVerde/myapiweb". The terminal output is as follows:

```
nestor@NestorB: ~/ProyectoParedVerde/myapiweb$ python3 manage.py createsuperuser
Username (leave blank to use 'nestor'): student
Email address: student@gmail.com
Password:
Password (again):
This password is too short. It must contain at least 8 characters.
This password is too common.
This password is entirely numeric.
Bypass password validation and create user anyway? [y/N]: y
Superuser created successfully.
(nestor@NestorB: ~/ProyectoParedVerde/myapiweb$ )
```





CONCLUSIONES

Este trabajo presenta la experiencia académica con el uso de Python y Django para el desarrollo rápido de prototipos de sistemas de información para estudiantes con una leve formación en el área de tecnología para las ciencias de la información. Según esto, se presentan las siguientes conclusiones: Python es un lenguaje de programación adecuado para un fácil y rápido dominio; Python es un lenguaje de programación actual con marcos de trabajo adecuados para el desarrollo rápido de soluciones informáticas; Django es una herramienta que permite la abstracción para trabajar con sistemas de base de datos (por ejemplo, no se requiere conocimiento directo de SQL); y gracias al gran desarrollo de Python como lenguaje de programación para el análisis de datos, los marcos de trabajo como Django son perfectamente integrables para el desarrollo de soluciones web para el análisis de datos.

Bibliografías

<https://docs.oracle.com/cd/E19957-01/821-1692/installmac/index.html>

<https://www.robertoserrano.pro/blog/installacion-de-visual-studio-code-en-ubuntu-20-04/#:~:text=Si%20prefieres%20usar%20la%20aplicaci%C3%B3n,una%20actualizaci%C3%B3n%20se%20actualizara%20automaticamente.>

<https://mosquitto.org/download/>

<https://openwebinars.net/blog/como-instalar-git-en-ubuntu/>

<https://ubunlog.com/python-3-9-como-instalar-en-ubuntu-20-04/#:~:text=1.2.2%20Compilaci%C3%B3n~Instalaci%C3%B3n%20de%20Python%203.9,la%20pagina%20web%20de%20Python.>

<https://www.linuxcapable.com/es/how-to-install-sqlite-3-on-ubuntu-20-04-lts/>