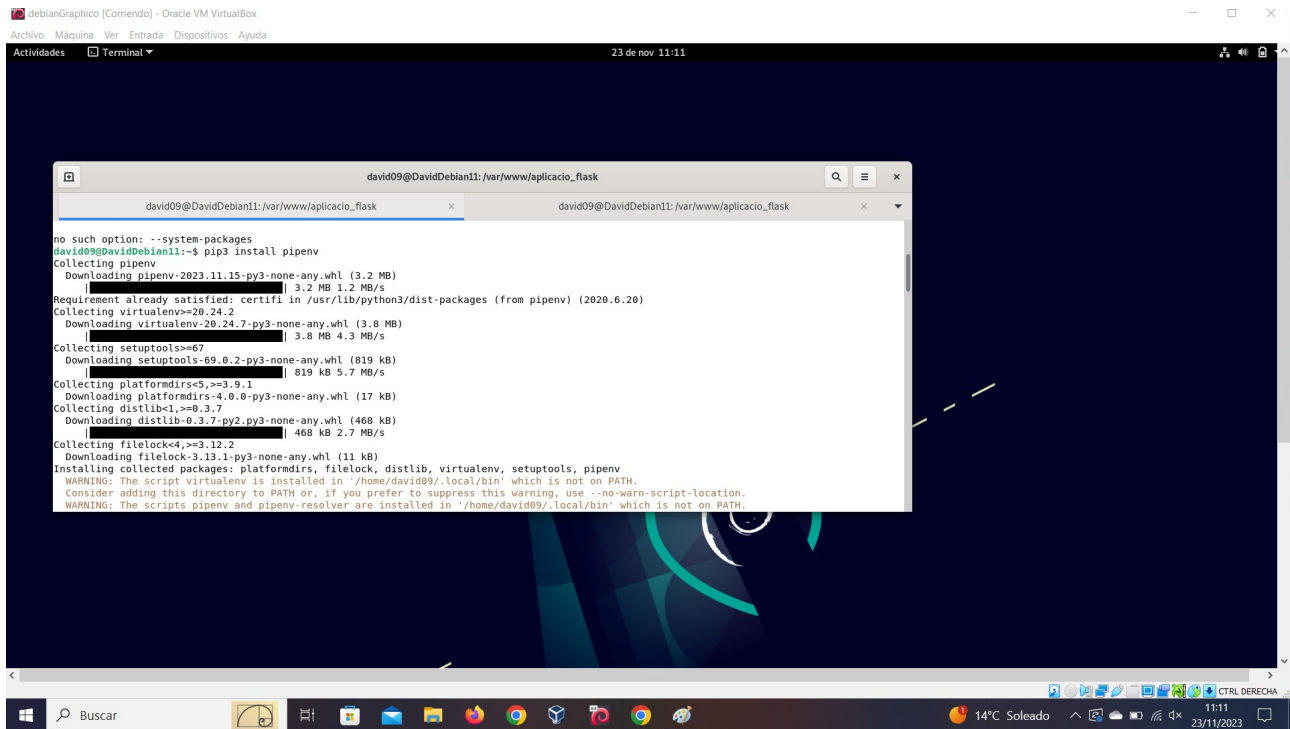


FLASK

Primero instalamos el gestor de paquetes de python:

`sudo apt-get update` (antes de hacer cualquier instalacion)

`sudo apt-get install python3-pip`

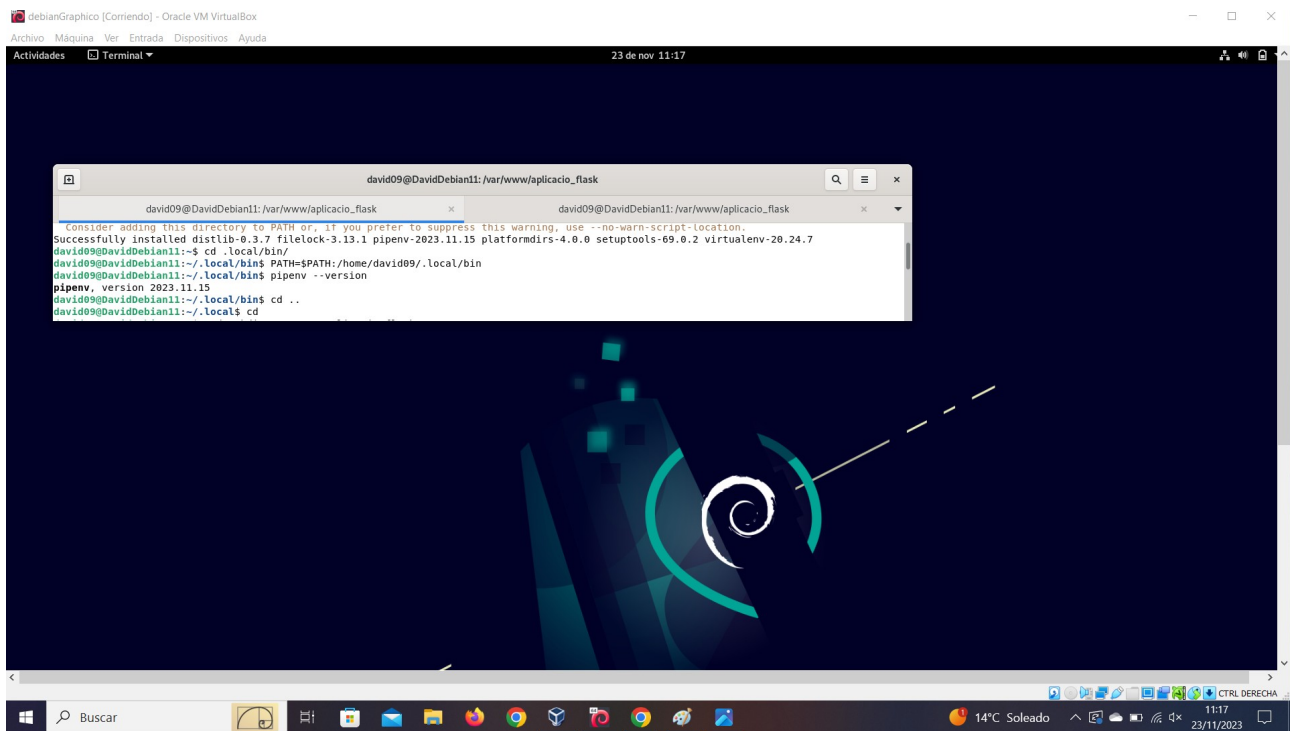


```
debianGraphico [Comando] - Oracle VM VirtualBox
Archivo Máquina Ver Entrada Dispositivos Ayuda
23 de nov 11:11

david09@DavidDebian11: /var/www/aplicacio_flask
no such option: --system-packages
david09@DavidDebian11:~$ pip3 install pipenv
Collecting pipenv
  Downloading pipenv-2023.11.15-py3-none-any.whl (3.2 MB)
    | 3.2 MB 1.2 MB/s
Requirement already satisfied: certifi in /usr/lib/python3/dist-packages (from pipenv) (2020.6.20)
Collecting virtualenv<20.24.2
  Downloading virtualenv-20.24.7-py3-none-any.whl (3.8 MB)
    | 3.8 MB 4.3 MB/s
Collecting setuptools>=67
  Downloading setuptools-69.0.2-py3-none-any.whl (819 kB)
    | 819 kB 5.7 MB/s
Collecting platformdirs<5,>=3.0.1
  Downloading platformdirs-4.0.0-py3-none-any.whl (17 kB)
Collecting distlib<1,>=0.3.7
  Downloading distlib-0.3.7-py2.py3-none-any.whl (468 kB)
    | 468 kB 2.7 MB/s
Collecting filelock<4,>=3.12.2
  Downloading filelock-3.13.1-py3-none-any.whl (11 kB)
Installing collected packages: platformdirs, filelock, distlib, virtualenv, setuptools, pipenv
WARNING: The script virtualenv is installed in '/home/david09/.local/bin' which is not on PATH.
Consider adding this directory to PATH or, if you prefer to suppress this warning, use --no-warn-script-location.
WARNING: The scripts pipenv and pipenv-resolver are installed in '/home/david09/.local/bin' which is not on PATH.
```

Usamos el comando `pip3 install pipenv` que es la manera para gestionar los entornos virtuales.

Después comprobamos que la versión instalada para ello usamos `pipenv --version`



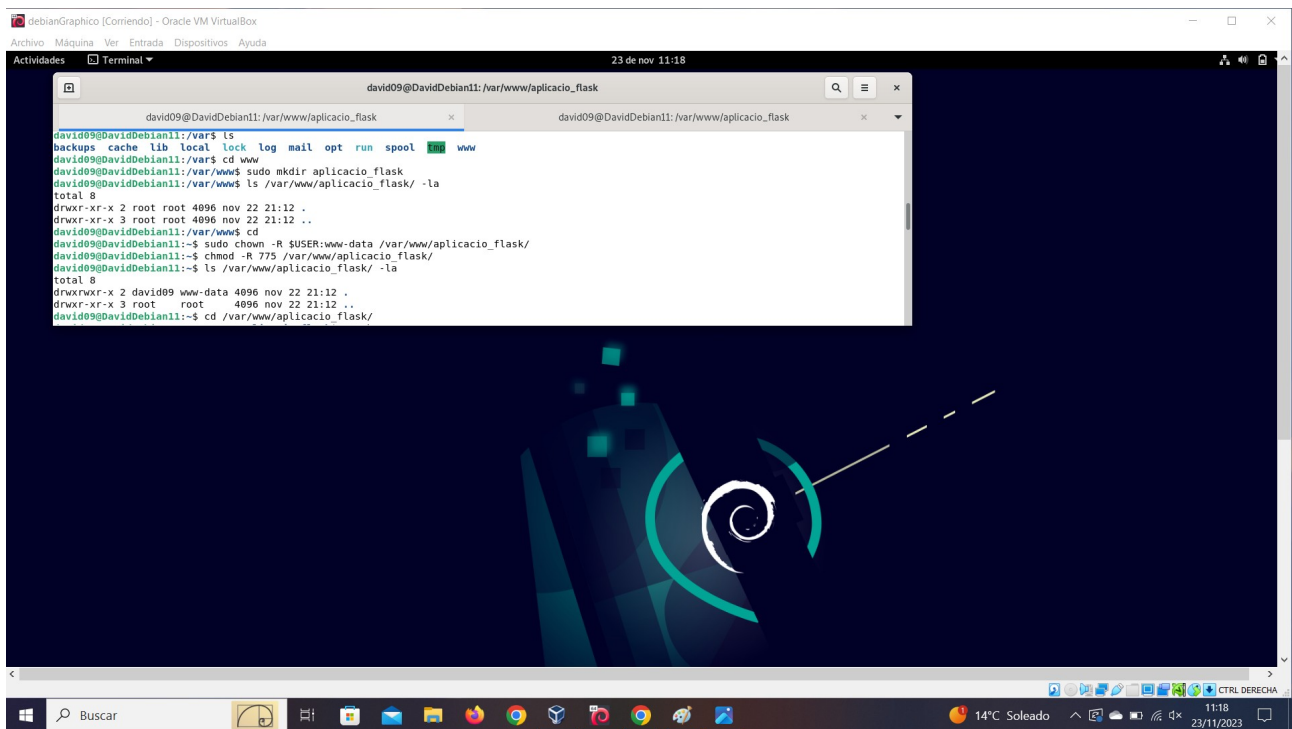
Vamos a crear un directorio en el cual vamos a guardar nuestro proyecto utilizando el siguiente comando: `sudo mkdir /var/www/el nombre de la aplicación` (www no esta creado deberia de crearlo en caso de fallo crea primero www y luego dentro de www el nombre de la aplicación) en mi caso le llamare `aplicacio_flask`.

Al crearlo vemos que los permisos pertenecen a root, para cambiarlo debemos ponerle los permisos al usuario.

`sudo chown -R $USER:www-data /var/www/nombre de la aplicacion`(en mi caso `aplicacio_flask`).

Y le ponemos los permisos para que pueda ser leida por todo el que quiera utilizando el siguiente comando.

`chmod -R 775 /var/www/nombre de la aplicacio`(en mi caso `aplicacio_flask`).

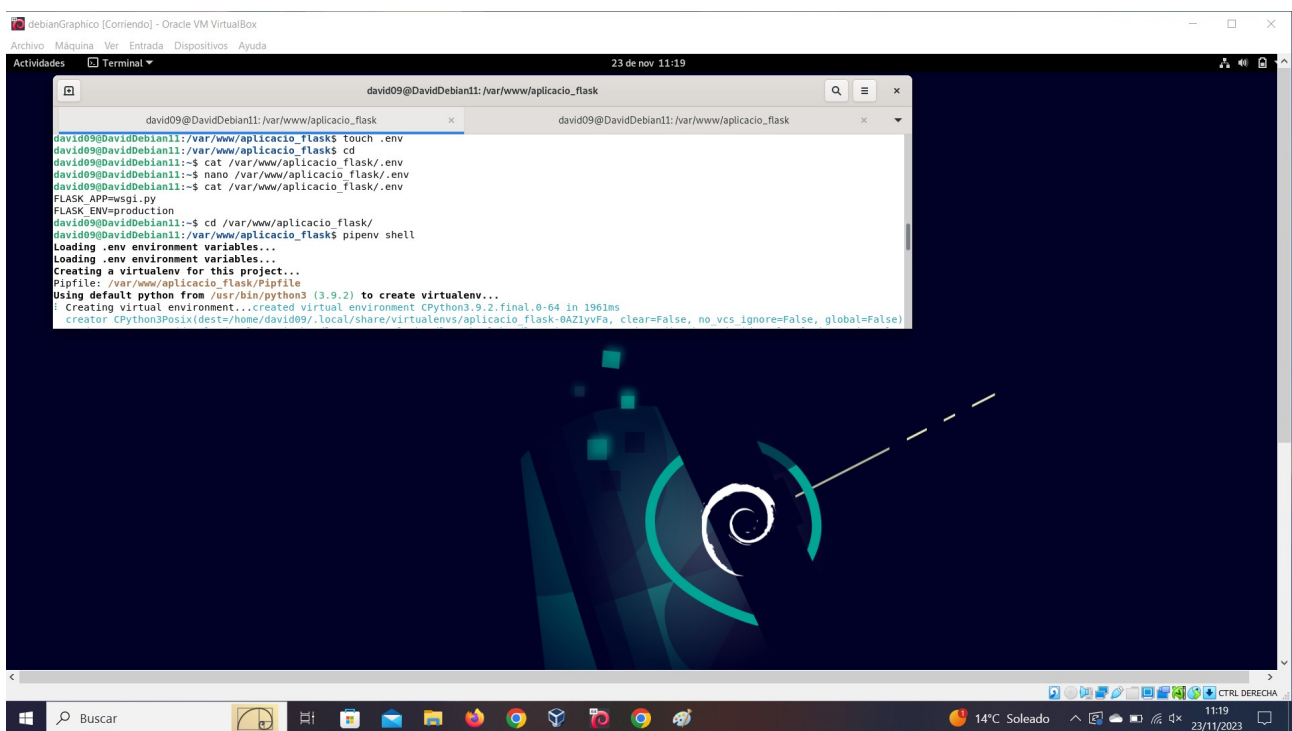


dentro del directorio de nuestra aplicación creamos un archivo oculto llamado .env con las variables de entorno necesario:

utilizando el siguiente comando: touch .env

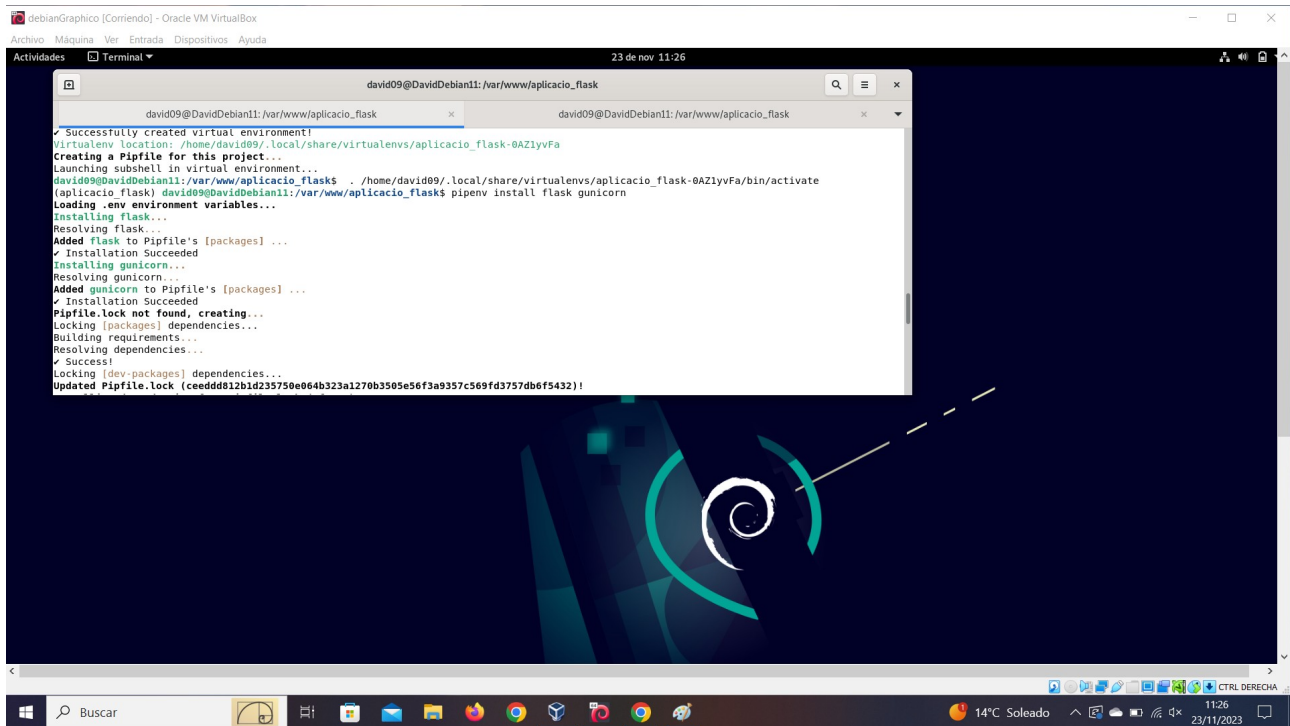
Iniciaremos en nuestro entorno virtual pipenv es el encargado de que desde el fichero env se cargaran todas las variables de entorno de manera automatica.

Pipenv shell



Luego nos metemos dentro del entorno virtual mediante `./home/nombre_usuario/.local/share/virtualenvs/aplicacio_flask(numero y letras aleatorios)/bin/activate` y dentro de entorno queremos que pipen instale todas las dependencias necesarias para nuestro proyecto para ello utilizaremos el siguiente comando:

`pipenv install flask gunicorn.`

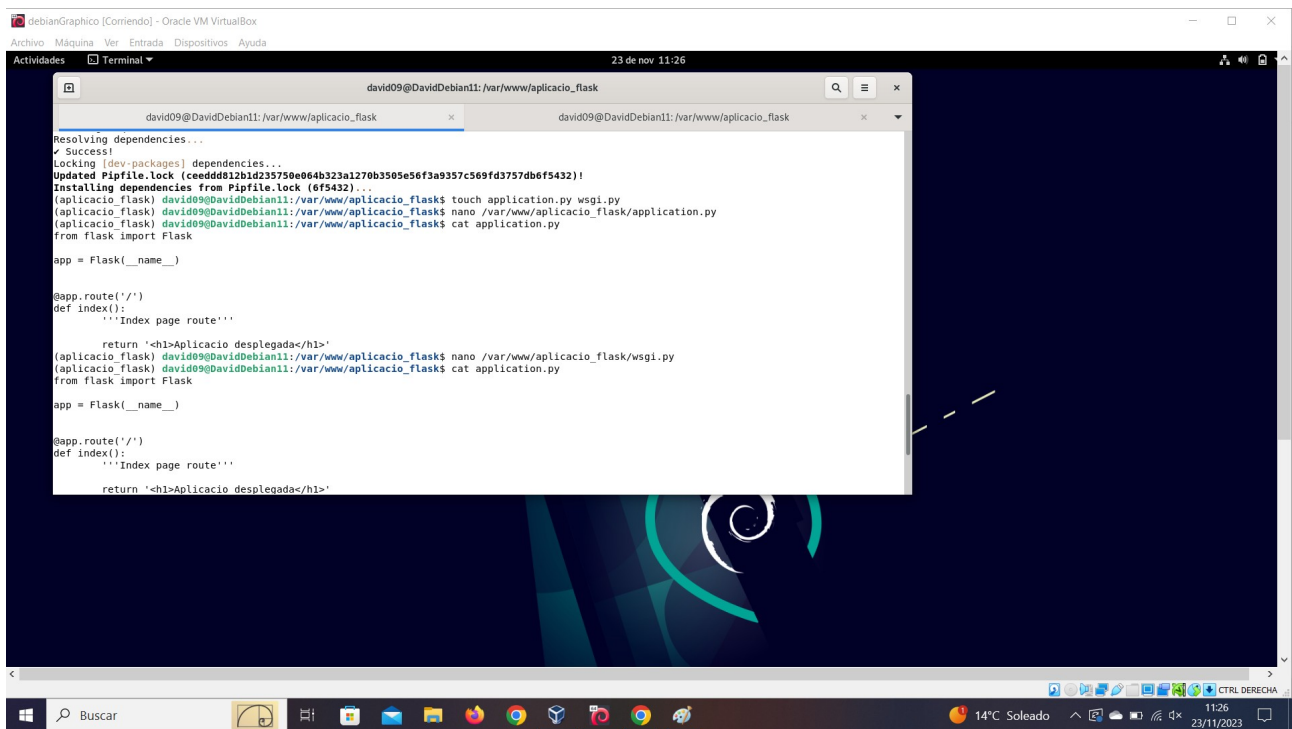


```
debianGraphico [Corriendo] - Oracle VM VirtualBox
Archivo  Máquina  Ver  Entrada  Dispositivos  Ayuda
Actividades  Terminal
23 de nov 11:26

david09@DavidDebian11: /var/www/aplicacio_flask
david09@DavidDebian11: /var/www/aplicacio_flask
david09@DavidDebian11: /var/www/aplicacio_flask
✓ Successfully created virtual environment!
Virtualenv location: /home/david09/.local/share/virtualenvs/aplicacio_flask-0AZIyvFa
Creating a Pipfile for this project...
Launching subshell in virtual environment...
david09@DavidDebian11: /var/www/aplicacio_flask$ ./home/david09/.local/share/virtualenvs/aplicacio_flask-0AZIyvFa/bin/activate
(aplicacio_flask) david09@DavidDebian11: /var/www/aplicacio_flask$ pipenv install flask gunicorn
Loading .env environment variables...
Installing flask...
Resolving flask...
Added flask to Pipfile's [packages] ...
✓ Installation Succeeded
Installing gunicorn...
Resolving gunicorn...
Added gunicorn to Pipfile's [packages] ...
✓ Installation Succeeded
Pipfile.lock not found, creating...
Locking [packages] dependencies...
Building requirements...
Resolving dependencies...
✓ Success!
Locking [dev-packages] dependencies...
Updated Pipfile.lock (ceedd812b1d235750e064b323a1270b350e56f3a9357c569fd3757db6f5432)!
```

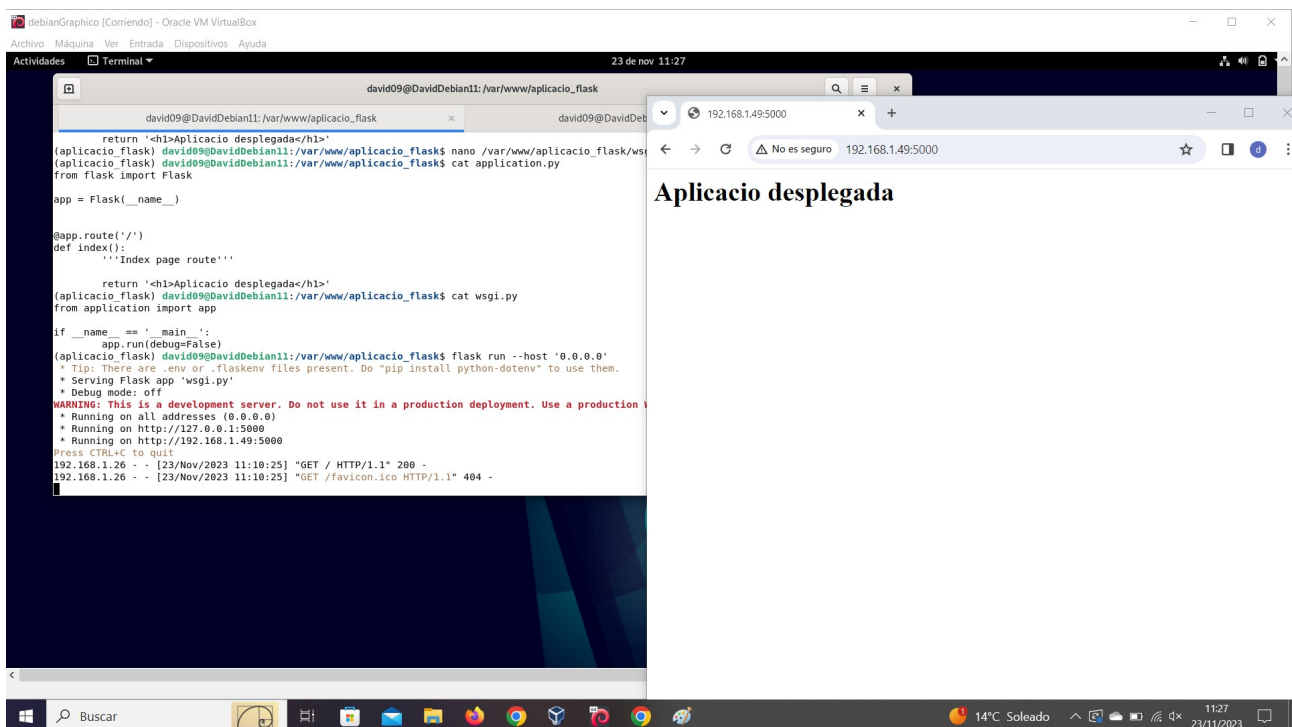
Vamos hacer una aplicación flask , modo PoC (proof of concept), los archivos que tendrán la aplicación serán 2 archivos que estarán en oculto.

Utilizaremos el siguiente comando. Touch nombredeaplicacion.py nombre2deaplicacion.py
y los editaremos.



Vamos a lanzar la aplicación ahora para ver si funciona utilizamos el siguiente comando: `Flask run --host '0.0.0.0'` (le indicamos dirección 0.0.0.0 es porque queremos que esuche todas las interfaces que tenga.)

ahora si ponemos la maquina virtual(yo lo he lanzado desde mi portatil personal poniendo la ip de mi debian) debería salirnos lo siguiente.



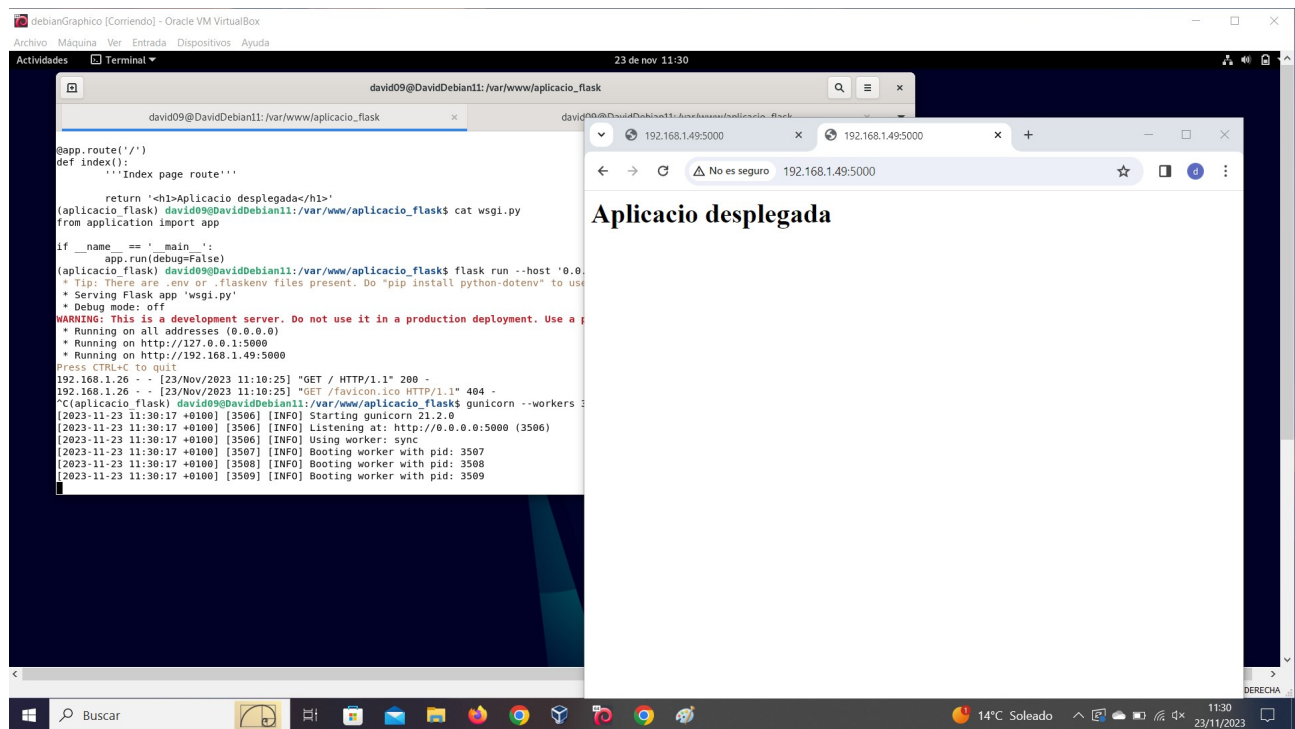
Después de ver que la aplicación funciona vamos a realizar lo mismo en Gunicorn, lo logico que si funciona en uno funcione en otro

usando el comando `unicorn -workers num -bind 0.0.0.0:5000 wsgi:app`

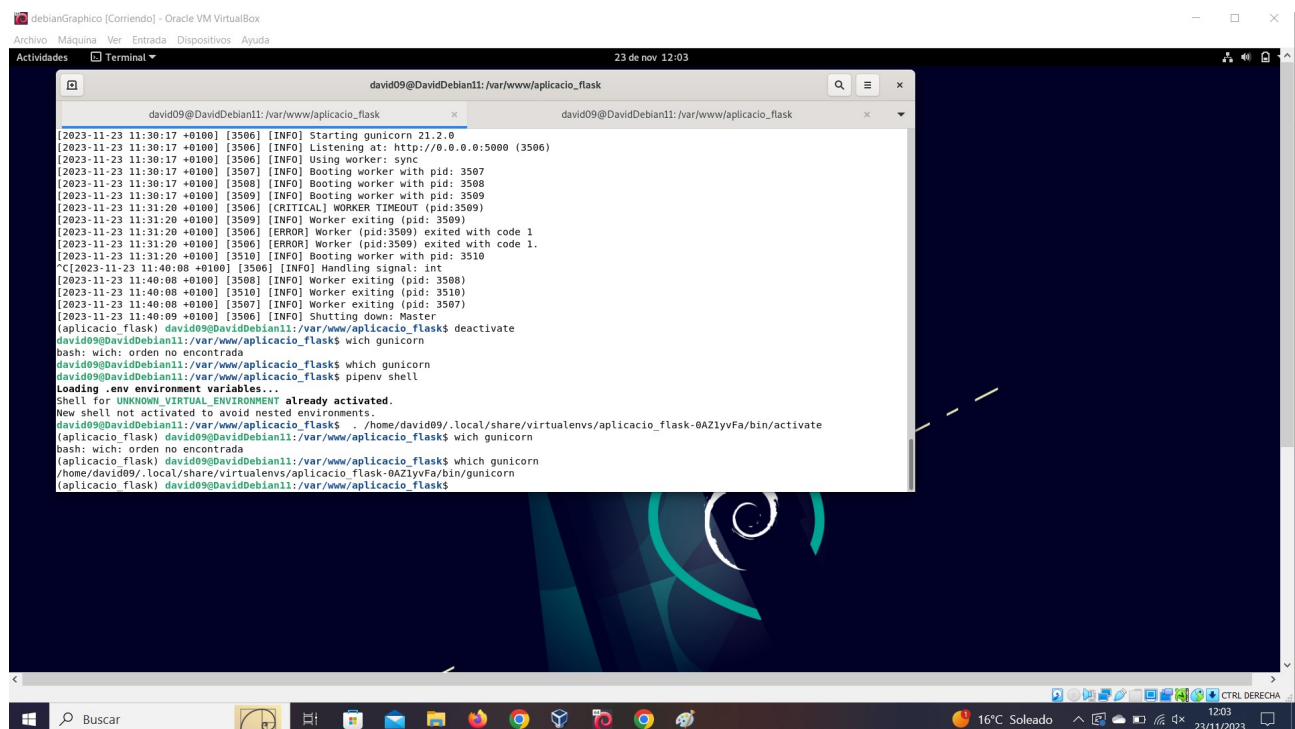
workers: numero de hijos que queremos que coja.

Bind: escucha las peticiones de sus interfaces de la red y en el puerto 5000

wsgi:app nombre de archivo de la extension .py(que hemos hecho en oculto antes).



Dentro del entorno virtual necesitamos guardarnos cual es la ruta desde cual unicorn se esta ejecutando en mi caso al preguntarlo utilizando `which` me da la siguiente ruta.

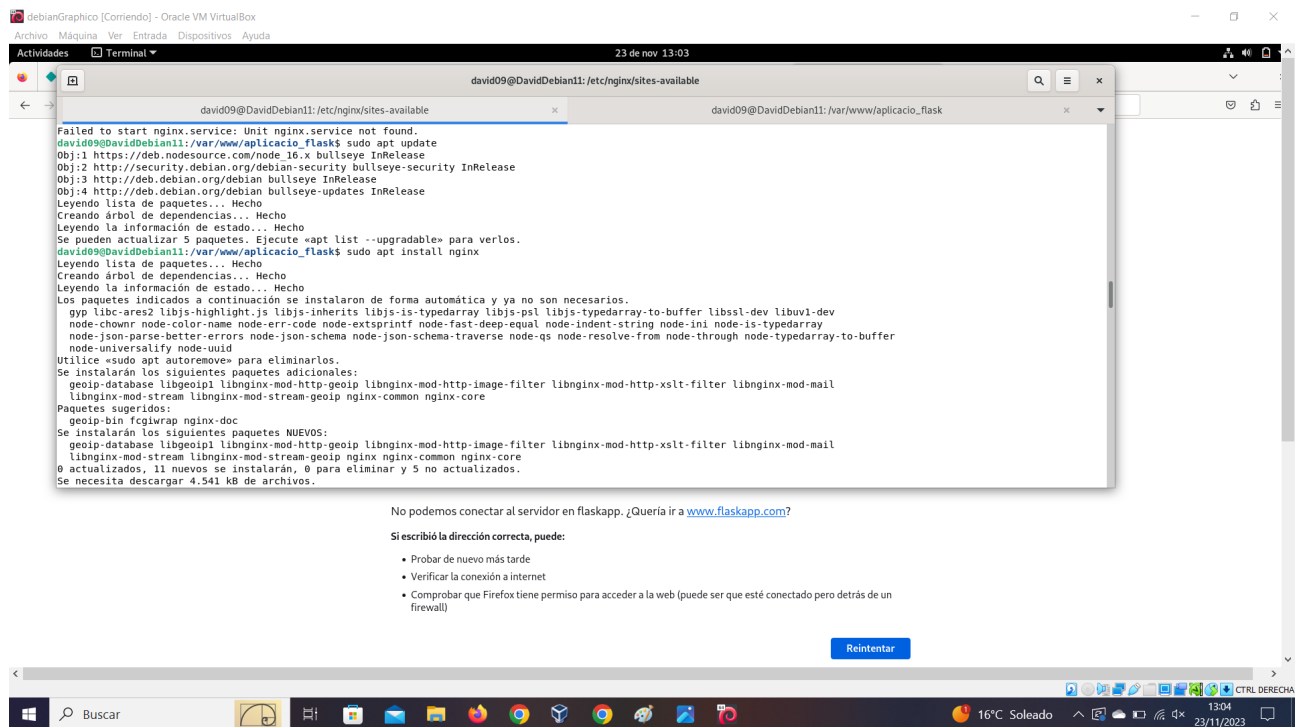


Para salirnos del entorno virtual utilizamos deactivate y veras que nos saldrá del entorno virtual.

Vamos ahora a realizar lo mismo pero usando nginx.

Primero comprobamos si lo tenemos instalado en el caso de no ser así lo instalamos con los 2 siguientes comandos:

Hacemos sudo apt update y despues sudo apt install nginx.



```
david09@DavidDebian11: /etc/nginx/sites-available
Failed to start nginx.service: Unit nginx.service not found.
david09@DavidDebian11: /var/www/aplicacio_flask$ sudo apt update
Obj:1 https://deb.nodesource.com/node_16.x bullseye InRelease
Obj:2 https://security.debian.org/debian-security bullseye-security InRelease
Obj:3 http://deb.debian.org/debian bullseye InRelease
Obj:4 http://deb.debian.org/debian bullseye-updates InRelease
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
Se pueden actualizar 5 paquetes. Ejecute «apt list --upgradable» para verlos.
david09@DavidDebian11: /var/www/aplicacio_flask$ sudo apt install nginx
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
Los paquetes indicados a continuación se instalaron de forma automática y ya no son necesarios.
gyp lib-ares2 libjs-highlight.js libjs-inherits libjs-is-typedarray libjs-psl libjs-typedarray-to-buffer libssl-dev libuv1-dev
node-chokidar node-color-name node-code node-extsprintf node-fast-deep-equal node-indent-string node-ini node-is-typedarray
node-json-parse-better-errors node-json-schema node-json-schema-traverse node-qs node-resolve-from node-through node-typedarray-to-buffer
node-universalify node-uuid
Utilice «sudo apt autoremove» para eliminarlos.
Se instalarán los siguientes paquetes adicionales:
geoip-database libgeoip1 libnginx-mod-http-geoip libnginx-mod-http-image-filter libnginx-mod-http-xslt-filter libnginx-mod-mail
libnginx-mod-stream libnginx-mod-stream-geoip nginx-common nginx-core
Paquetes sugeridos:
geoip-bin fcgiwrap nginx-doc
Se instalarán los siguientes paquetes NUEVOS:
geoip-database libgeoip1 libnginx-mod-http-geoip libnginx-mod-http-image-filter libnginx-mod-http-xslt-filter libnginx-mod-mail
libnginx-mod-stream libnginx-mod-stream-geoip nginx nginx-common nginx-core
0 actualizados, 11 nuevos se instalarán, 0 para eliminar y 5 no actualizados.
Se necesita descargar 4.541 kB de archivos.
```

No podemos conectar al servidor en flaskapp. ¿Quería ir a www.flaskapp.com?

Si escribió la dirección correcta, puede:

- Probar de nuevo más tarde
- Verificar la conexión a internet
- Comprobar que Firefox tiene permiso para acceder a la web (puede ser que esté conectado pero detrás de un firewall)

Reintentar

Después vamos a iniciar nginx y lo hacemos utilizando sudo systemctl start nginx.

Para comprobar que esta funcionando sudo systemctl status nginx.

Ahora vamos a habilitarlo y a lanzarlo con los siguiente comando(el archivo previo que hemos creado dentro de systemd)

```
systemctl enable nombre_del_servicio_creado
```

```
systemctl start nombre_del_servicio_creado
```

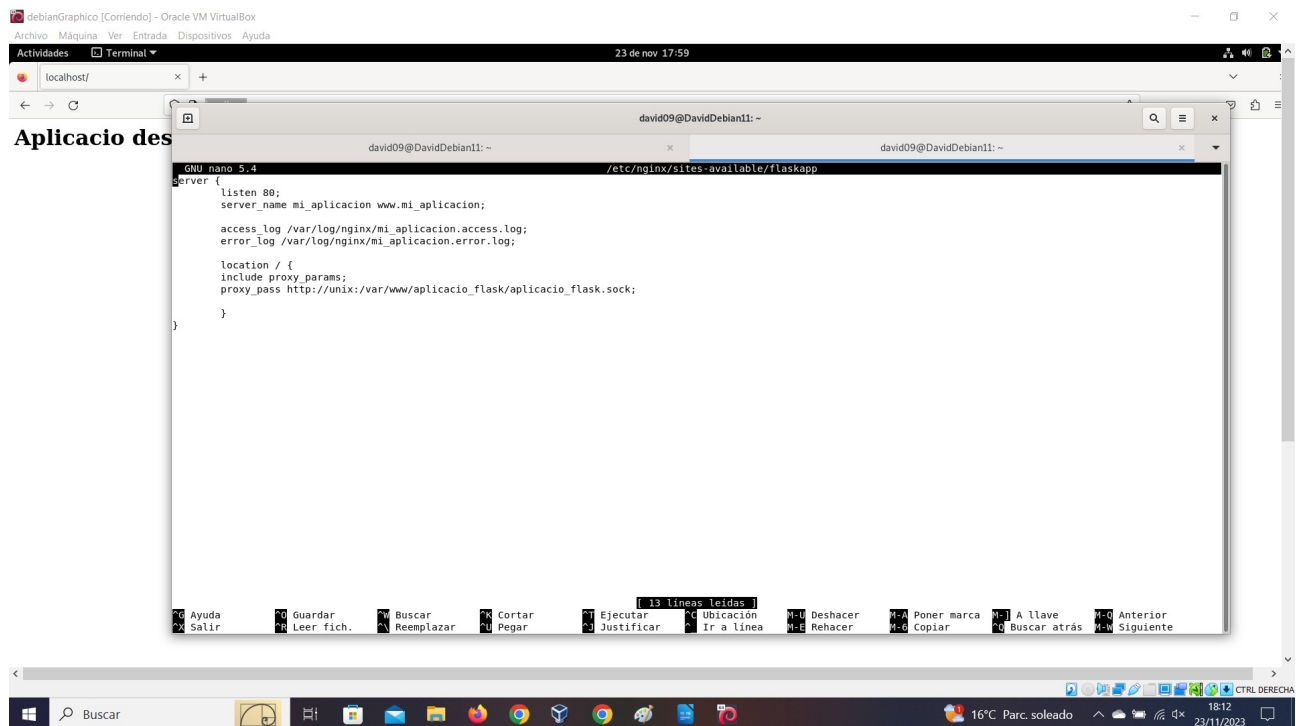
Vamos a crearnos un archivo con el nombre de aplicación y dentro de el vamos a configurar para la web que querramos. Para ello debe estar en /etc/nginx/sites-available/nombreDeTuAplicacion.

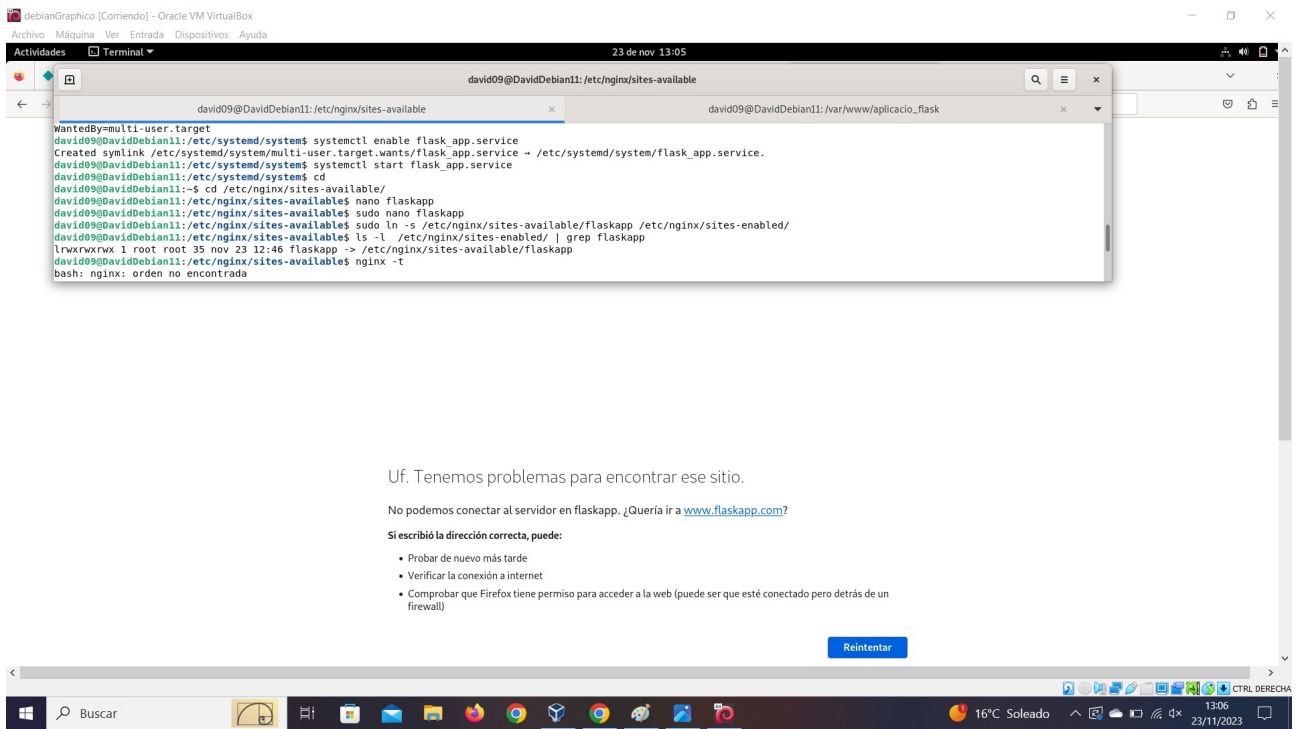
Al tener el archivo creado tenemos que crear un link simbolico para que sepa donde tiene que llamar.

Utilizando el comando `sudo ln -s /etc/nginx/sites-available/nombreDeTuAplicacion /etc/nginx/sites-enabled/`

Y si queremos asegurarnos que el enlace esta bien hecho podemos utilizar el siguiente comando.

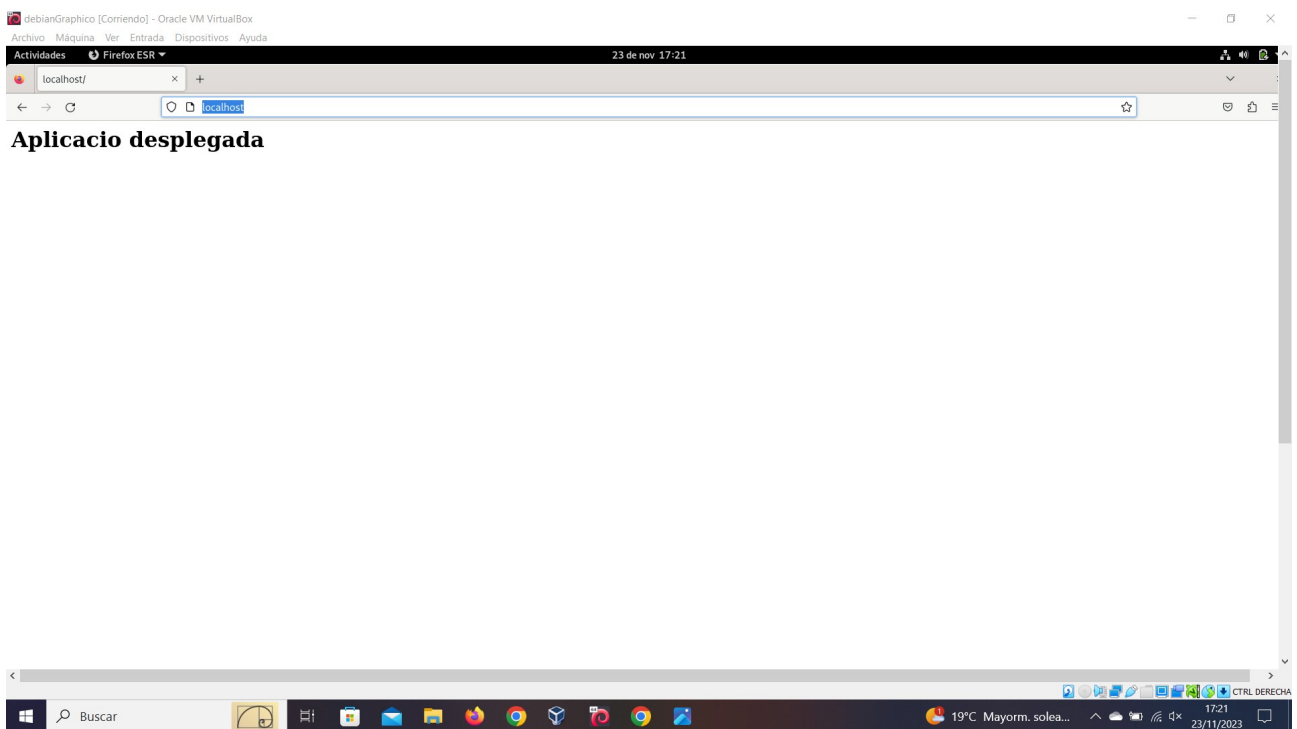
```
ls -l /etc/nginx/sites-enabled/ | grep flaskapp
```



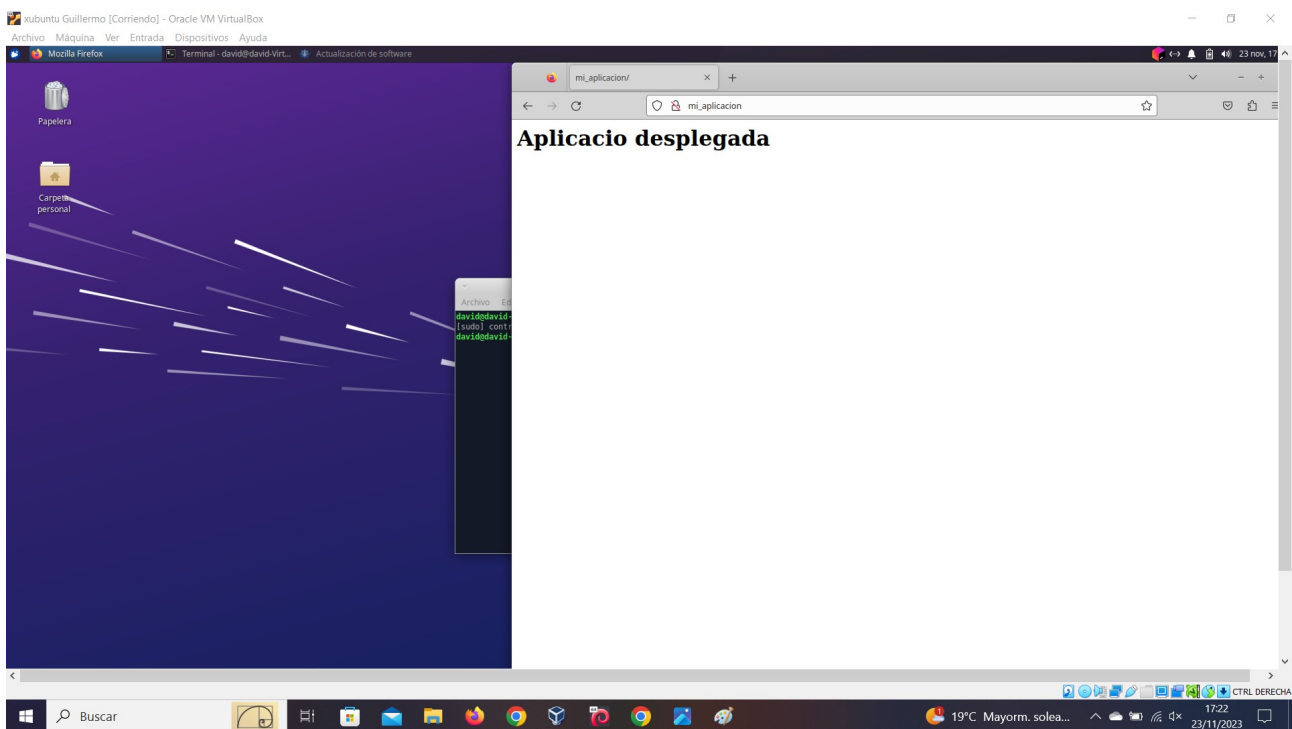


despues de configurarlo tenemos que reiniciar nginx para que coja los cambios que hemos realizado para ello hacemos un `systemctl restart nginx` y comprobamos que todo esta bien y no hay ningun error `systemctl status nginx`.

Si todo esta correcto al poner `http://mi_aplicacion`(en mi caso) o http://www.mi_aplicacion debería de funcionar aunque puedes usar tambien localhost dentro de tu debian o configuramos en otra maquina virtual ubuntu (vamos a `/etc/hosts` ponemos la ip y el nombre) y deberia de dejarnos entrar



(desde la debian)



(configurando hosts con la ip de la debian y el nombre desde una maquina virtual ubuntu)