

Configurar un entorno virtual de Python

Asegúrese de que la versión del sistema Ubuntu sea lo suficientemente reciente

```
lsb_release -a
```

Impresión (o arriba)

Distributor ID: Ubuntu

Description: Ubuntu 24.04.3 LTS

Release: 24.04

Codename: noble

```
python3 -V
```

Impresión

Python 3.12.3

Crear una base para todos nuestros entornos virtuales

```
mkdir ${HOME}/.venv  
# Entonces podemos crear todos nuestros entornos virtuales en ${HOME}/.venv/
```

Crear nuestro entorno virtual

```
# Aquí nombramos el entorno virtual devpi  
cd ${HOME}/.venv/  
python3 -m venv devpi
```

O utilizando venvwrapper

```
mkvenv devpi
```

Una vez creado nuestro entorno, podemos instalar los elementos necesarios para empezar.

Actívalo

```
source ${HOME}/.venv/devpi/bin/activate
```

O utilizando venvwrapper

```
venv devpi
```

Instalar lo básico

```
# Instalar
python -m pip install -U pip
pip install wheel
pip install setuptools
pip install twine
pip install rst2pdf
pip install flit

# desactivar
deactivate
```

Guardar requisitos previos

Una vez completado el entorno virtual, conviene guardar el contexto. De esta forma, podrá restaurarlo rápidamente desde cero si es necesario.

Mientras está activado

```
pip freeze >requirements.txt

# Es buena idea mantener dichos archivos bajo control de versiones.
```

Recuperar

Mientras está activado

```
pip install -r requirements.txt
```

Activación automática

En muchos casos, queremos activar un proyecto al iniciar sesión. Podemos hacerlo automáticamente añadiendo algo como esto a .bashrc o .profile.

```
#
# Añadido para activación automática del entorno virtual
#
if [ -f "${HOME}/.venv/devpi/bin/activate" ] ; then
    source "${HOME}/.venv/devpi/bin/activate
fi
```

O utilizando venvwrapper

```
#
# Añadido para activación automática del entorno virtual
#
VENV_HOME=${HOME}/.venv
venv_wrapper=$(which venvwrapper.sh)
if [[ -n $venv_wrapper ]]; then
    source $venv_wrapper
    venv devpi
fi
```