

1.1__Entornos Virtuales

June 13, 2025

Creado por:

Isabel Maniega

```
[1]: from IPython.display import Image
```

1 1.1 Entornos Virtuales

El objetivo de crear un entorno virtual es poder trabajar dentro del mismo sistema operativo en diferentes proyectos, tanto con distintas versiones de Python, así como con diferentes versiones de paquetes, así no interferir y crear conflictos entre proyectos.

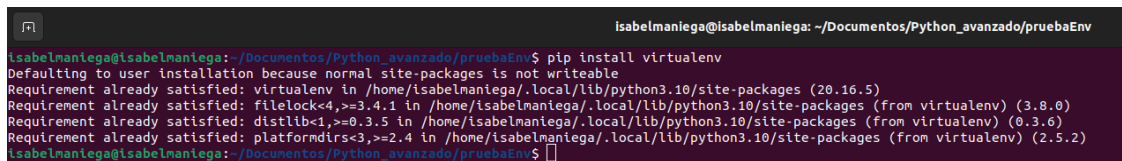
Para ello crearemos la carpeta del proyecto donde queremos trabajar **PruebaEnv**, abriremos el terminal en esa ubicación y realizaremos la instalación del paquete de entornos virtuales. Existen paquetes estándar de Python como es **venv** (<https://docs.python.org/es/3/library/venv.html>) o paquetes como **virtualenv** (<https://virtualenv.pypa.io/en/latest/>), es el caso que nos ocupa, usaremos virtualenv.

Primero realizamos la instalación del paquete con:

```
pip install virtualenv
```

```
[2]: Image(filename='./images/venv_1.png')
```

[2]:



A terminal window screenshot showing the command 'pip install virtualenv' being executed. The output indicates that virtualenv is already installed in the current environment, with details about the version (20.16.5) and the Python version (3.8.0). The terminal prompt is 'isabelmaniega@isabelmaniega: ~/Documentos/Python_avanzado/pruebaEnv\$'.

Una vez instalado crearemos el entorno virtual en la carpeta del proyecto:

```
virtualenv <nombre entorno>
```

Si queremos usar una versión de Python concreta en el entorno pondremos el comando `python=version python a usar`:

```
virtualenv --python=3.8 <nombre entorno>
```

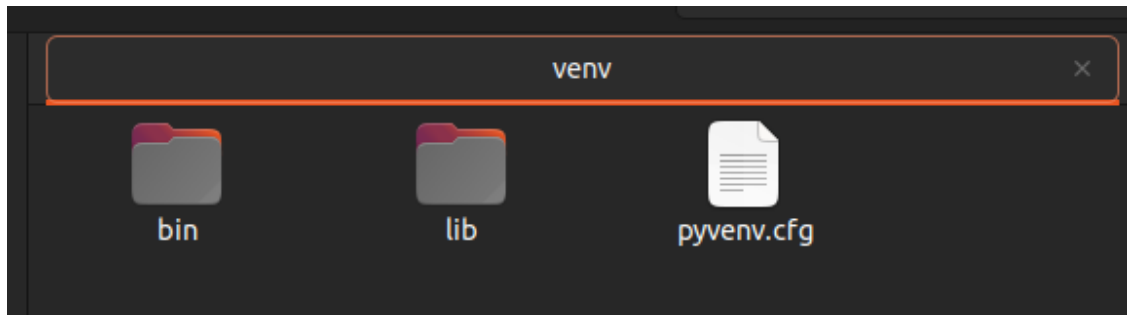
```
[3]: Image(filename='./images/venv_2.png')
```

[3]:

```
isabelmaniega@isabelmaniega:~/Documentos/Python_avanzado/pruebaEnv$ virtualenv env
created virtual environment CPython3.10.12.final.0-64 in 123ms
creator CPython3Posix(dest=/home/isabelmaniega/Documentos/Python_avanzado/pruebaEnv/env, clear=False, no_vcs_ignore=False, global=False)
seeder FromAppData(download=False, pip=bundle, setuptools=bundle, wheel=bundle, via=copy, app_data_dir=/home/isabelmaniega/.local/share/virtualenv)
added seed packages: pip==24.3.1, setuptools==75.5.0, wheel==0.45.1
activators BashActivator,CShellActivator,FishActivator,NushellActivator,PowerShellActivator,PythonActivator
isabelmaniega@isabelmaniega:~/Documentos/Python_avanzado/pruebaEnv$ virtualenv --python=3.8 venv
created virtual environment CPython3.8.20.final.0-64 in 100ms
creator CPython3Posix(dest=/home/isabelmaniega/Documentos/Python_avanzado/pruebaEnv/venv, clear=False, no_vcs_ignore=False, global=False)
seeder FromAppData(download=False, pip=bundle, setuptools=bundle, wheel=bundle, via=copy, app_data_dir=/home/isabelmaniega/.local/share/virtualenv)
added seed packages: pip==24.3.1, setuptools==75.5.0, wheel==0.45.1
activators BashActivator,CShellActivator,FishActivator,NushellActivator,PowerShellActivator,PythonActivator
isabelmaniega@isabelmaniega:~/Documentos/Python_avanzado/pruebaEnv$
```

[4]: Image(filename='./images/venv_3.png')

[4]:



Se crean varias carpetas entre ellas se encuentran:

- bin (Linux, Mac) o Scripts (Windows): donde se encuentran los comando de ejecución del entorno como, activar, desactivar, python, etc.
- lib: donde se encuentran los paquetes disponibles de Python para esa versión. En esta parte nos crea una pequeña copia de Python con los paquetes necesarios para que funcione de manera correcta. En esta carpeta, se realizarán las descargas de los paquetes necesarios para nuestro proyecto.

Para poder usar el entorno debemos de activarlo mediante el siguiente comando:

- linux o Mac: `source <nombre entorno>/bin/activate`
- Windows: `.\<nombre entorno>\Scripts\activate`

[5]: Image(filename='./images/venv_4.png')

[5]:

```
isabelmaniega@isabelmaniega:~/Documentos/Python_avanzado/pruebaEnv$ source venv/bin/activate
(venv) isabelmaniega@isabelmaniega:~/Documentos/Python_avanzado/pruebaEnv$
```

Se observa que estamos dentro del entorno porque nos pone entre paréntesis el nombre del entorno.

Una vez activo podemos realizar las instalaciones necesarias u observar que paquetes tenemos instalados con el comando de pip, como:

- `pip list` nos devuelve los paquetes instalados en el entorno.
- `pip install <paquete>` para instalarlo.

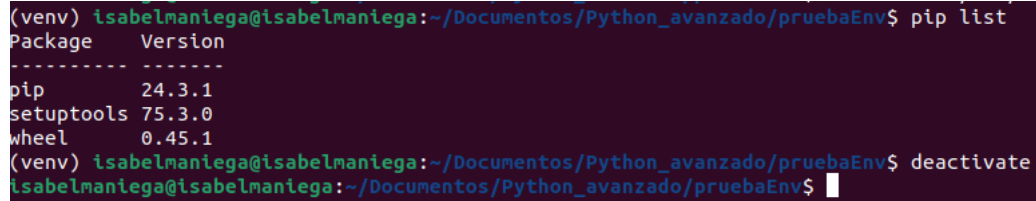
- `pip uninstall <paquete>` para desinstalarlo
- `pip freeze > requirements.txt` para extraer los paquetes con sus debidas veriones de paquetes del proyecto.
- `pip install -r requirements.txt` para instalar los requerimientos del proyecto, etc...

Cuando queramos dejar de usar el entorno solo debemos de poner:

`deactivate`

[6]: `Image(filename='./images/venv_5.png')`

[6]:



```
(venv) isabelmaniega@isabelmaniega:~/Documentos/Python_avanzado/pruebaEnv$ pip list
Package      Version
-----
pip          24.3.1
setuptools   75.3.0
wheel        0.45.1
(venv) isabelmaniega@isabelmaniega:~/Documentos/Python_avanzado/pruebaEnv$ deactivate
isabelmaniega@isabelmaniega:~/Documentos/Python_avanzado/pruebaEnv$
```

Creado por:

Isabel Maniega