

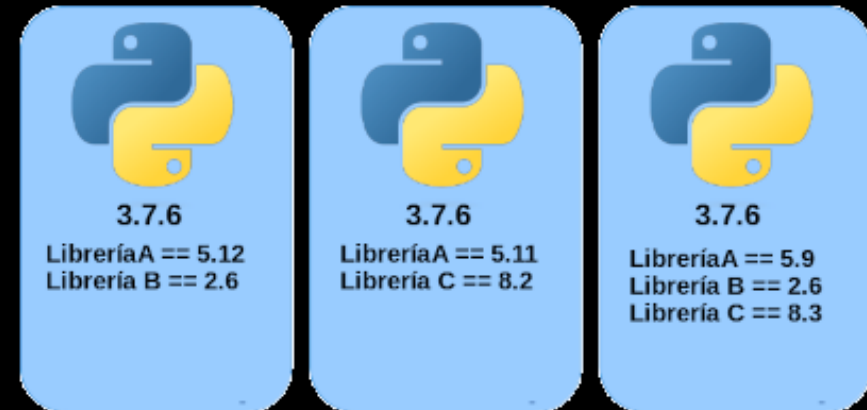
Introducción al Desarrollo de Aplicaciones en Android

- Ing. Skrauba Axel

Entornos Virtuales

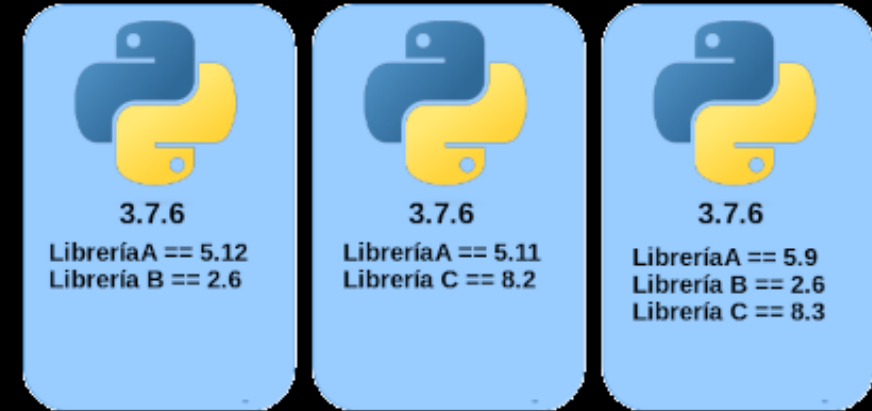
Un **entorno virtual de Python** es un ambiente creado con el objetivo de aislar recursos como librerías y entorno de ejecución, del sistema principal o de otros entornos virtuales.

Esto hace posible tener instaladas múltiples versiones de una misma librería sin crear ningún tipo de conflicto.



Entornos Virtuales

Un entorno virtual de Python es "un árbol de directorios autónomo que contiene una instalación de Python para una **versión particular de Python**, además de una serie de **paquetes** adicionales"



Diferentes aplicaciones pueden usar diferentes entornos virtuales. La aplicación A puede tener su propio entorno virtual con la versión 1.0 instalado mientras que la aplicación B tiene otro entorno virtual con la versión 2.0. Si la aplicación B requiere que una biblioteca se actualice a la versión 3.0, esto no afecta el entorno de la aplicación A.

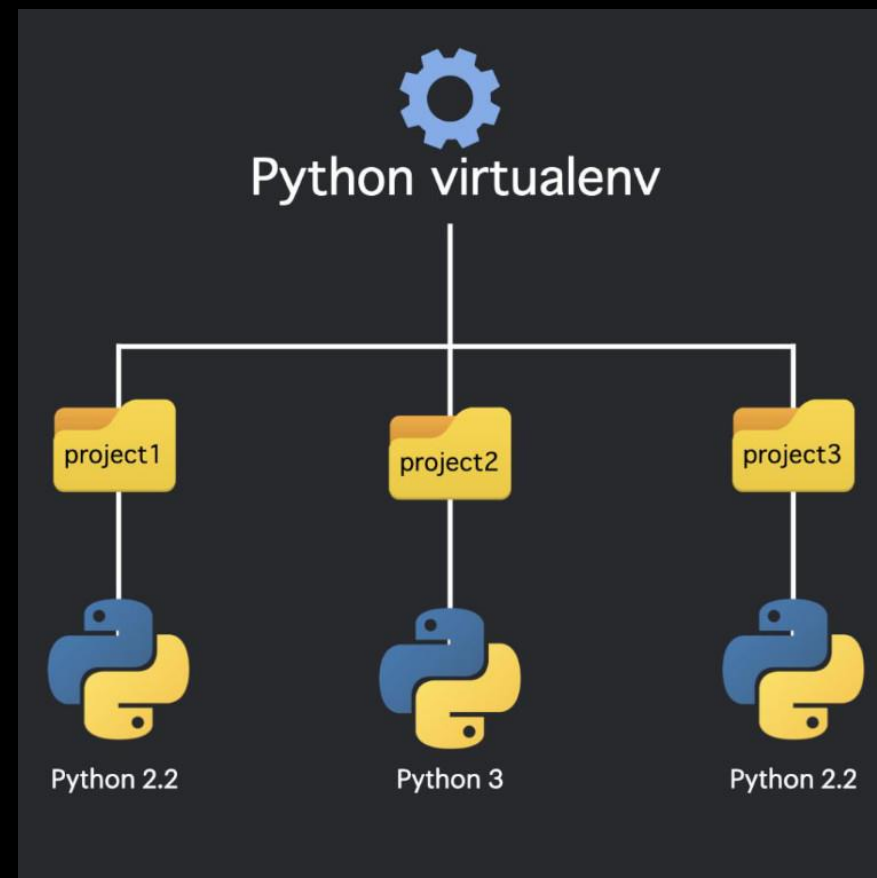
Virtualenv

¿Qué es *virtualenv*?

Es una herramienta para crear entornos Python aislados para proyectos diferentes. Permite tener dependencias y versiones de paquetes específicos para cada proyecto.

<https://pypi.org/project/virtualenv/>

<https://virtualenv.pypa.io/en/latest/installation.html>

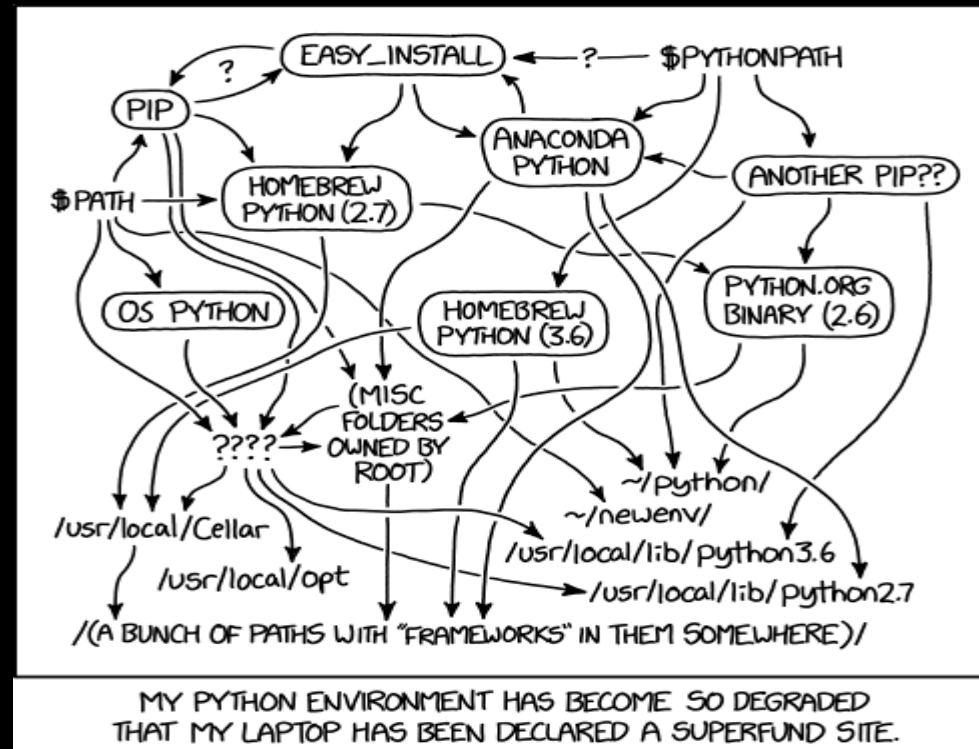


Virtualenv

Alternativas

- venv
- pyenv
- conda *environments*
- Poetry
- Autoenv
- Pew
- ...

Cada uno tiene sus pros



Un entorno virtual de Python es "un árbol de directorios autónomo que contiene una instalación de Python para una versión particular de Python, además de una serie de paquetes adicionales"

Virtualenv

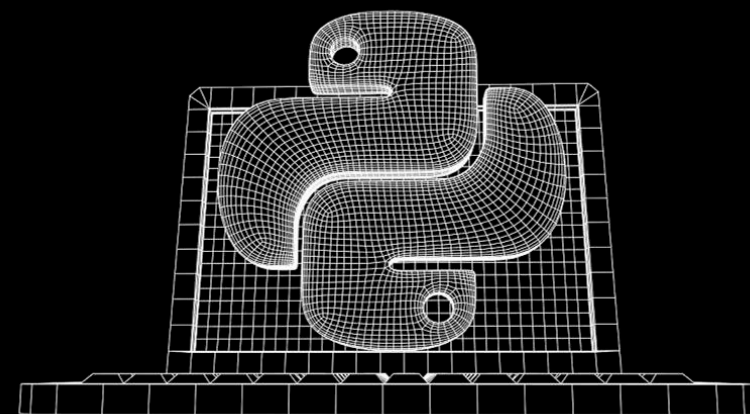
Configuración

Ubicación del entorno virtual

- Por defecto en el directorio donde se crea
- Opcional especificar ruta absoluta o relativa

Copiar el entorno

- Copiar toda la carpeta a otra ubicación
- Ejecutar el *activate* desde nueva ruta

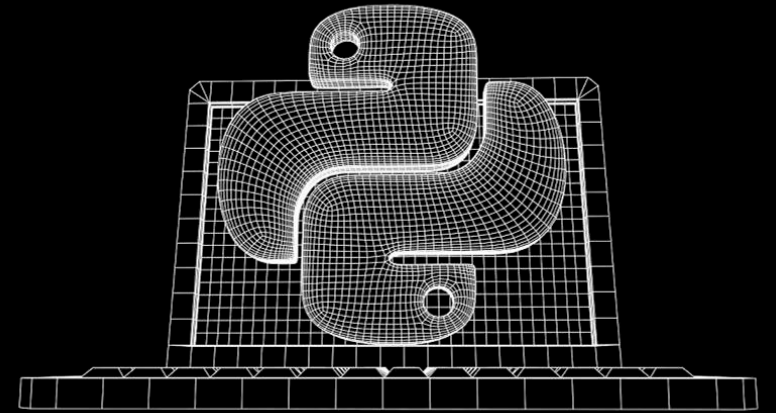


Virtualenv

Instalación

En una consola del S.O. (única vez):

```
$ pip install virtualenv
```



<https://virtualenv.pypa.io/en/latest/installation.html>

Virtualenv: Uso

En el directorio del nuevo proyecto

Creación:

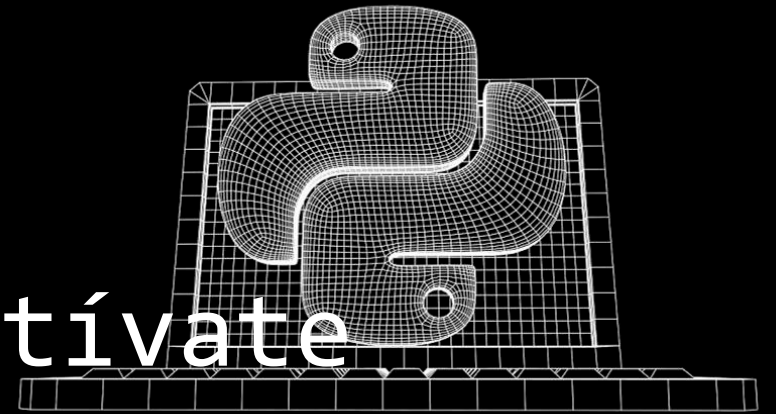
```
$ virtualenv nombre_env
```

Activar (Linux/macOS):

```
$ source nombre_env/bin/activate
```

Activar (Windows):

```
$ nombre_env\Scripts\activate
```

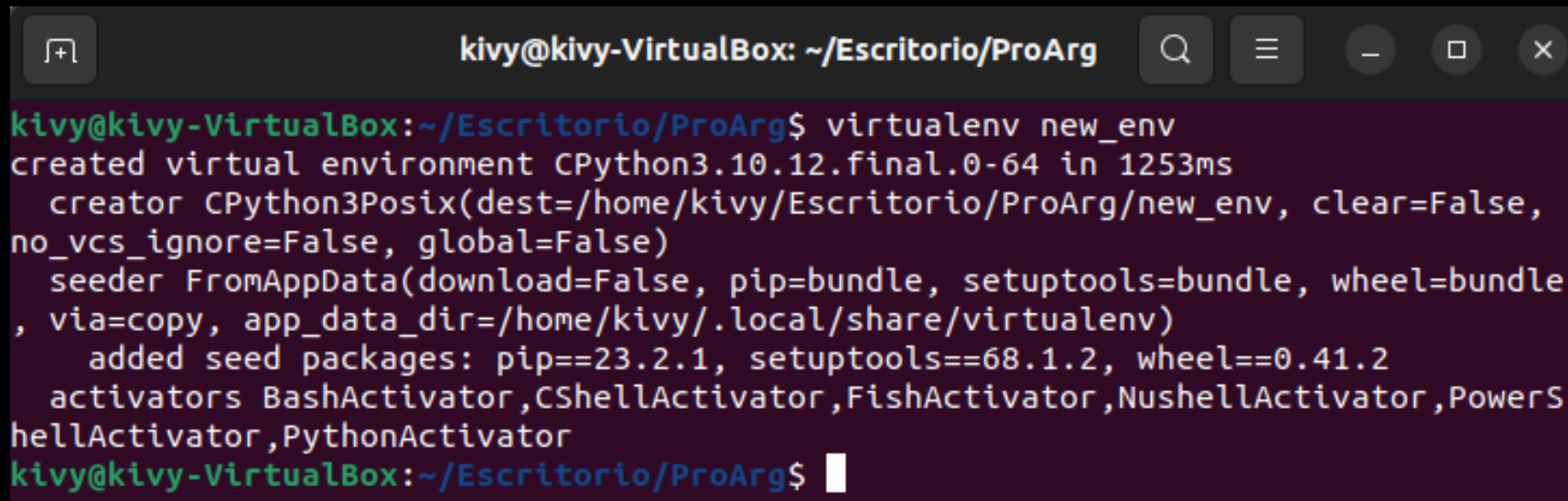


Virtualenv: Uso

En el directorio del nuevo proyecto

Creación:

\$ virtualenv nombre_env

A terminal window titled 'kivy@kivy-VirtualBox: ~/Escritorio/ProArg' with standard window controls. The terminal shows the execution of 'virtualenv new_env', which creates a virtual environment. The output includes details about the Python version (CPython3.10.12.final.0-64), the creator (CPython3Posix), the seeder (FromAppData), and the activators (BashActivator, CShellActivator, FishActivator, NushellActivator, PowershellActivator, PythonActivator).

```
kivy@kivy-VirtualBox:~/Escritorio/ProArg$ virtualenv new_env
created virtual environment CPython3.10.12.final.0-64 in 1253ms
  creator CPython3Posix(dest=/home/kivy/Escritorio/ProArg/new_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/kivy/.local/share/virtualenv)
    added seed packages: pip==23.2.1, setuptools==68.1.2, wheel==0.41.2
  activators BashActivator,CShellActivator,FishActivator,NushellActivator,PowerS
hellActivator,PythonActivator
kivy@kivy-VirtualBox:~/Escritorio/ProArg$
```

Virtualenv: Uso

En el directorio del nuevo proyecto

Activar (Linux/macOS):

\$ source nombre_env/bin/actíivate

Activado



```
kivy@kivy-VirtualBox: ~/Escritorio/ProArg
kivy@kivy-VirtualBox:~/Escritorio/ProArg$ virtualenv new_env
created virtual environment CPython3.10.12.final.0-64 in 1253ms
  creator CPython3Posix(dest=/home/kivy/Escritorio/ProArg/new_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/kivy/.local/share/virtualenv)
    added seed packages: pip==23.2.1, setuptools==68.1.2, wheel==0.41.2
  activators BashActivator,CShellActivator,FishActivator,NushellActivator,PowerS
hellActivator,PythonActivator
kivy@kivy-VirtualBox:~/Escritorio/ProArg$ source new_env/bin/activate
(new_env) kivy@kivy-VirtualBox:~/Escritorio/ProArg$
```

Comparación

Global S.O.

Entorno Virtual

```
kivy@kivy-VirtualBox: ~/Escritorio/ProArg
kivy@kivy-VirtualBox:~/Escritorio/ProArg$ pip list
Package          Version
-----
apturl            0.5.2
blinker           1.4
Brlapi            0.8.3
buildozer         1.5.0
certifi           2020.6.20
chardet           4.0.0
click             8.0.3
colorama          0.4.4
command-not-found 0.3
cryptography      3.4.8
cupshelpers       1.0
Cython            0.29.33
dbus-python       1.2.18
defer             1.0.6
distlib           0.3.7
distro            1.7.0
distro-info       1.1build1
filelock          3.12.2
httplib2          0.20.2
idna              3.3
importlib-metadata 4.6.4
jeepney           0.7.1
keyring           23.5.0
language-selector 0.1
launchpadlib      1.10.16
lazr.restfulclient 0.14.4
lazr.uri          1.0.6
louis             3.20.0
macaroonbakery    1.3.1
meson             0.61.2
more-itertools    8.10.0
```

```
kivy@kivy-VirtualBox: ~/Escritorio/ProArg
kivy@kivy-VirtualBox:~/Escritorio/ProArg$ virtualenv new_env
created virtual environment CPython3.10.12.final.0-64 in 1253ms
  creator CPython3Posix(dest=/home/kivy/Escritorio/ProArg/new_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/kivy/.local/share/virtualenv)
  added seed packages: pip==23.2.1, setuptools==68.1.2, wheel==0.41.2
  activators BashActivator,CShellActivator,FishActivator,NushellActivator,PowerS
hellActivator,PythonActivator
kivy@kivy-VirtualBox:~/Escritorio/ProArg$ source new_env/bin/activate
(new_env) kivy@kivy-VirtualBox:~/Escritorio/ProArg$ pip list
Package          Version
-----
pip              23.2.1
setuptools       68.1.2
wheel            0.41.2
(new_env) kivy@kivy-VirtualBox:~/Escritorio/ProArg$
```

Virtualenv: Uso

Instalar paquetes dentro del entorno:

`(new_env) $ pip install paquete_name`

```
(new_env) kivy@kivy-VirtualBox:~/Escritorio/ProArg$ pip install "kivy[base]"
Collecting kivy[base]
  Obtaining dependency information for kivy[base] from https://files.pythonhosted.org/packages/b0/4d/9c65432d3fdea1edc81f51c01db6f804a6dac9d5f9fa3ff1f0a5c18e127f/Kivy-2.2.1-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata
  Using cached Kivy-2.2.1-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata (13 kB)
Collecting Kivy-Garden>=0.1.4 (from kivy[base])
  Using cached Kivy_Garden-0.1.5-py3-none-any.whl (4.6 kB)
Collecting docutils (from kivy[base])
  Obtaining dependency information for docutils from https://files.pythonhosted.org/packages/26/87/f238c0670b94533ac0353a4e2a1a771a0cc73277b88bff23d3ae35a256c1/docutils-0.20.1-py3-none-any.whl.metadata
  Using cached docutils-0.20.1-py3-none-any.whl.metadata (2.8 kB)
Collecting pygments (from kivy[base])
  Obtaining dependency information for pygments from https://files.pythonhosted.org/packages/43/88/29adf0b44ba6ac85045e63734ae0997d3c58d8b1a91c914d240828d0d73d/Pygments-2.16.1-py3-none-any.whl.metadata
  Using cached Pygments-2.16.1-py3-none-any.whl.metadata (2.5 kB)
```

Virtualenv: Uso

Instalar paquetes dentro del entorno:

(new_env) \$ pip install paquete_name

Paquetes y
dependencias
instalados



```
(new_env) kivy@kivy-VirtualBox:~/Escritorio/ProArg$ pip list
Package            Version
-----
certifi             2023.7.22
charset-normalizer  3.3.0
docutils            0.20.1
idna                3.4
Kivy                2.2.1
Kivy-Garden         0.1.5
Pillow              10.0.1
pip                 23.2.1
Pygments            2.16.1
requests            2.31.0
setuptools          68.1.2
urllib3             2.0.6
wheel               0.41.2
(new_env) kivy@kivy-VirtualBox:~/Escritorio/ProArg$
```

Virtualenv: Uso

Desactivar entorno:

`(new_env) $ deactivate`

Global S.O. →

```
(new_env) kivy@kivy-VirtualBox:~/Escritorio/ProArg$ pip list
Package            Version
-----
certifi             2023.7.22
charset-normalizer  3.3.0
docutils            0.20.1
idna                3.4
Kivy                2.2.1
Kivy-Garden         0.1.5
Pillow              10.0.1
pip                 23.2.1
Pygments            2.16.1
requests            2.31.0
setuptools          68.1.2
urllib3             2.0.6
wheel               0.41.2
(new_env) kivy@kivy-VirtualBox:~/Escritorio/ProArg$ deactivate
kivy@kivy-VirtualBox:~/Escritorio/ProArg$
```

Dependencias (instalar)

```
(new_env) $ pip install buildozer
```

```
(new_env) $ pip install Cython==0.29.33
```

```
(new_env) $ pip install kivymd
```

```
(new_env) $ pip install plyer
```

<https://github.com/kivy/plyer/tree/master>

Extra: plyer

Plyer es una biblioteca de Python que proporciona una interfaz unificada y sencilla para acceder a características específicas del sistema operativo en aplicaciones de Python, incluidas las aplicaciones desarrolladas con Kivy en dispositivos Android.

Abstrae las diferencias en la implementación entre sistemas operativos y dispositivos, lo que permite a los desarrolladores acceder a las mismas funcionalidades en plataformas diferentes utilizando la misma API de Python. Esto facilita el desarrollo multiplataforma y simplifica el código.

Extra: plyer

Acceso a Características del Dispositivo:

Plyer proporciona acceso a una variedad de características del dispositivo, como la cámara, la ubicación, los sensores, la batería, el sistema de archivos, las notificaciones, la vibración y más.

Permite acceder a las funcionalidades del dispositivo mediante una API de Python, lo que les evita tener que aprender lenguajes de programación nativos de cada plataforma.

Extra: plyer

Solicitud de Permisos:

Cuando una aplicación intenta acceder a una característica protegida por permisos, el sistema operativo Android requerirá que la aplicación solicite permisos al usuario. Esto se hace generalmente a través de un cuadro de diálogo de solicitud de permisos.

Concesión de Permisos:

El usuario tiene la capacidad de otorgar o denegar permisos a la aplicación. La decisión del usuario es fundamental, y una aplicación debe estar preparada para funcionar de manera adecuada y segura tanto si se le otorgan como si se le deniegan permisos.

Extra: Pipenv

- **Pip** y **virtualenv** se usan en conjunto para mantener las dependencias de un entorno virtual, pero *pip* puede llegar a producir entornos diferentes, incluso con un mismo archivo *requirements.txt*, esto es algo que queremos evitar.
- El creador de **pipenv** diseñó su herramienta intentando resolver esa problemática.

Lectura:

https://dev.to/prox_sea/el-administrador-de-entornos-virtuales-que-no-conoces-pipenv-175a

Preguntas...

- Dudas
- Sugerencias
- Cuestiones
- Vamos al código...