UNDA | Python Academy

Clase 9 | Entornos virtuales

DATA & ANALYTICS ARGENTINA

UNIVERSITY OF DATA · UNDA · DATA & ANALYTICS

NTT DATA

¿PARA QUE NECESITAMOS ENTORNOS VIRTUALES?

✓ ¿Para que necesitamos entornos virtuales?

NTT DATA

Para la mantenibilidad de nuestros proyectos

Cuando desarrollamos nuestros proyectos utilizamos un conjunto de librerías con su correspondiente versionado. Esto se conoce como las **dependencias** de un proyecto.

Si utilizaremos el entorno general de Python y actualizamos las librerías corremos el **riesgo** de que los proyectos desarrollados anteriormente dejen de funcionar.

Los entornos virtuales nos permiten **independizar** las dependencias de un proyecto de las dependencias de otro.

¿QUE COMPONE UN ENTORNO VIRTUAL?

¿QUE COMPONE UN ENTORNO VIRTUAL?

NTT DATA

Versión de Python

Por ejemplo podemos trabajar en proyectos con una versión de Python 3.6 y otros con Python 3.11

Entorno Virtual

Librerias de terceros

Las librerías tienen sus propias versiones.

¿QUE HERRAMIENTAS SE USAN PARA CONSTRUIR ENTORNOS VIRTUALES?

DIFERENTES MANERAS DE CREARLOS

NTT DATA

VENV

- Nativo con Python > 3.3
- Fácil de utilizar
- No permite versionado con Python 2

VIRTUALENV

- Requiere instalación
- Fácil de utilizar
- Permite versionado de Python 2
- Tiene opciones para usuarios avanzados

PIPENV

- Similar a virtualenv
- Reemplaza a pip
- Usa pipfiles para tener un registro de todas las librerías del entorno
- Fue idealmente pensada para entornos de producción.

CONDA

- Combina entornos virtuales con manager de paquetes
- Es ideal para ciencia de datos
- Permite versionado de Python
- Puede no ser la mejor opción para etapas de deployment

NTT DATA

VENV

1. Crear el entorno virtual con *–m venv “nombre del entorno”*

2. Activar el entorno virtual:
   - *\nombre del entorno\Scripts\activate.bat* (En Windows)
   - *Source nombre del entorno/bin/actívate* (En Linux/Mac)

3. Instalar librerías con *pip install*. Opcionalmente podemos crear un requirements.txt y hacer un *pip install –r requirements.txt*

NTT DATA

VIRTUALENV

1. Instalar virtualenv con *pip install virtualenv*

2. Crear el entorno virtual con –m *virtualenv “nombre del entorno”.* Opcionalmente se puede agregar *--python version*

3. Activar el entorno virtual:
   - *\nombre del entorno\Scripts\actívate.bat* (En Windows)
   - *Source nombre del entorno/bin/actívate* (En Linux/Mac)

4. Instalar librerías con *pip install*. Opcionalmente podemos crear un requirements.txt y hacer un *pip install –r requirements.txt*

NTT DATA

PIPENV

1. Instalar pipenv con *pip install pipenv*

2. Crear el entorno virtual con *pipenv shell*
Los entornos tienen un nombre autoasignado y no crean un directorio adicional

3. Instalar librerías con *pipenv install*. Opcionalmente podemos crear un requirements.txt y hacer un *pip install –r requierements.txt*

4. Se puede trabajar con *requierements.txt* pero un pipfile es introducido como su reemplazo. Para saber más leer https://realpython.com/pipenv-guide/

✓ PIPENV

1. Separa en entornos de desarrollo (**dev-packages**) y producción (**packages**).
2. Incluye la version de Python
3. Con el commando *pipenv lock* generamos un archivo **pipfile.lock** que no debe ser editado manualmente y sirve para congelar las dependencias al momento de llevar a producción.

```
[[source]]
url = https://pypi.org/simple
verify_ssl = true
name = "pypi“

[packages]

[dev-packages]

[requires]
python_version = "3.11"
```

CONDA

1. Instalar manualmente anaconda o miniconda (https://www.anaconda.com/)

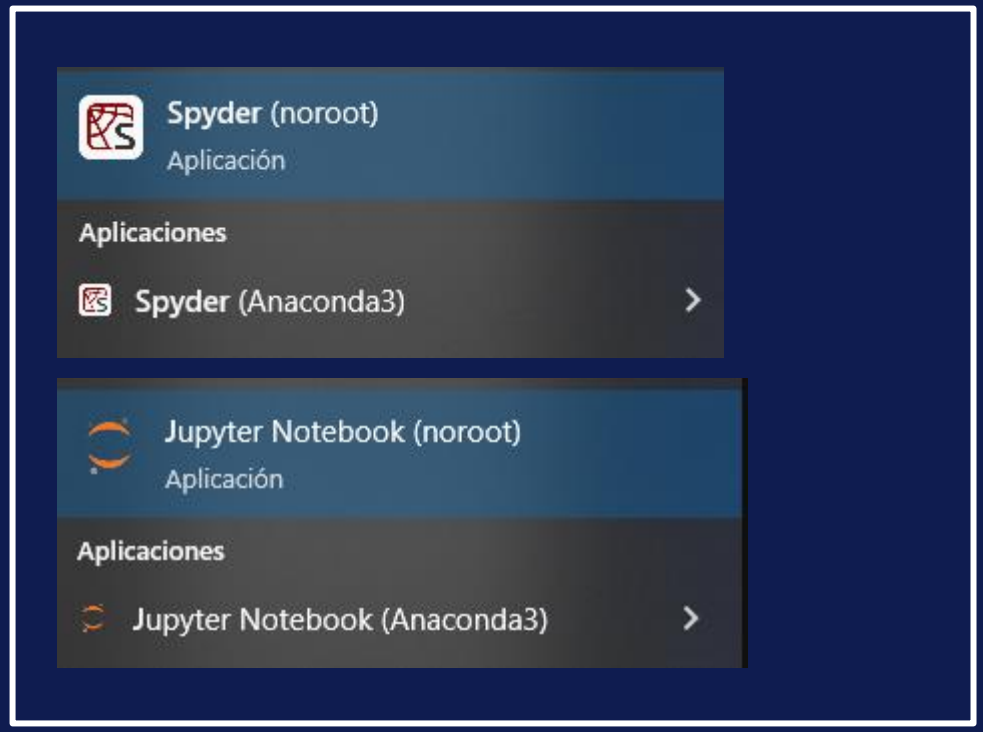
2. Con la **anaconda prompt** crear el entorno virtual con *conda create –name “nombre del environment”*

3. Instalar librerías con *conda install*. Si el paquete no está en los repositorios de anaconda podemos usar pip (Ocasionalmente esto causa algunos problemas)

4. Activar con *source activate “nombre del environment”*

✓ CONDA ENVIRONMENTS

1. Todo esto se puede hacer con la GUI de Anaconda Navigator
2. La creación de un nuevo entorno también “duplica” las aplicaciones instaladas con Anaconda como spyder o jupyter notebook
3. Podemos listar los entornos creados con *conda env list*



Anaconda | Using Pip in a Conda Environment

NTT DATA

Best Practices Checklist

Use pip only after conda

- install as many requirements as possible with conda, then use pip
- pip should be run with –upgrade-strategy only-if-needed (the default)
- Do not use pip with the –user argument, avoid all “users” installs

Use conda environments for isolation

- create a conda environment to isolate any changes pip makes
- environments take up little space thanks to hard links
- care should be taken to avoid running pip in the “root” environment

Recreate the environment if changes are needed

- once pip has been used conda will be unaware of the changes
- to install additional conda packages it is best to recreate the environment

Store conda and pip requirements in text files

- package requirements can be passed to conda via the –file argument
- pip accepts a list of Python packages with -r or –requirements
- conda env will export or create environments based on a file with conda and pip requirements

¡GRACIAS!

DATA & ANALYTICS ARGENTINA

UNIVERSITY OF DATA
UNDA
DATA & ANALYTICS

NTT DATA