Proyecto de Software

Tema

- Entornos Virtuales.

Entornos Virtuales

- Multiples versiones del lenguajes (Python 2.X, Python 3.X).

- Multiples versiones de librerías.

- Requerimientos mínimos.

Entornos Virtuales

- Herramienta que nos permite tener un ambiente para un proyecto con sus dependencias y librerías.

- Nos permite trabajar en un mismo equipo con diferentes versiones de lenguaje y paquetes.

- Nos ayuda a replicar el ambiente de desarrollo

- Virtualenv, venv, Conda, poetry, ¿otros?

Entornos Virtuales - Pip

- Pip es un sistema de gestión de paquetes.

- Primero, comprobamos si tenemos correctamente instalado pip.

```
sudo apt-get install python3-pip python3-dev build-essential
```

Entornos Virtuales - Pip

- Nos facilita conocer cuáles son los paquetes y sus versiones para poder distribuirlo.

```
pip freeze > requirements.txt
```

- Podemos ver cuáles son los paquetes mirando requirements.txt

```
Click==7.0
Flask==1.1.1
itsdangerous==1.1.0
Jinja2==2.10.1
MarkupSafe==1.1.1
pkg-resources==0.0.0
Werkzeug==0.16.0
```

Entornos Virtuales - Pip

- Algunos comandos útiles

```
pip list
pip install <nombre_del_paquete>==<version>
pip show <nombre_del_paquete>
```

Entornos Virtuales - Pip

- Para desplegar el proyecto en otra plataforma ejecutamos el siguiente comando:

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

Entornos Virtuales

- Instalamos virtualenv

```
pip install virtualenv
```

Entornos Virtuales

- Nos posicionamos en la carpeta del proyecto o, en su defecto, la creamos:

```
mkdir miProyecto
cd miProyecto
```

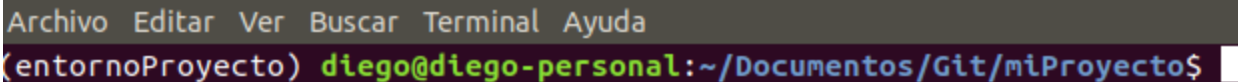# Entornos Virtuales

- Inicializamos el entorno virtual:

```
virtualenv entornoProyecto
```

Entornos Virtuales

- Activamos el entorno virtual:

```
source entornoProyecto/bin/activate
```


- Para desactivar el entorno virtual:

```
deactivate
```

Entornos Virtuales

- Desde la versión de Python 3.3 contamos con venv

```
python3 -m venv /path/to/new/virtual/environment
```

- Por lo general:

```
python3 -m venv venv
```

Entornos Virtuales

- Una herramienta útil para gestionar entornos virtuales es **virtualenvwrapper**:

```
pip install virtualenvwrapper
```

```
diego@diego-personal:~/Documentos/Git/miProyecto$ pip install virtualenvwrapper
Collecting virtualenvwrapper
  Downloading https://files.pythonhosted.org/packages/c1/6b/2f05d73b2d2f2410b48b90d3783a0034c26afa534a4a95ad5f1178d61191/virtualenvwrapper-4.8.4.tar.gz (334kB)
    100% |                                | 337kB 1.1MB/s
Collecting virtualenv (from virtualenvwrapper)
  Downloading https://files.pythonhosted.org/packages/8b/12/8d4f45b8962b03ac9efefe5ed5053f6b29334d83e438b4fe379d21c0cb8e/virtualenv-16.7.5-py2.py3-none-any.whl (3.3MB)
    100% |                                | 3.3MB 204kB/s
Collecting virtualenv-clone (from virtualenvwrapper)
  Downloading https://files.pythonhosted.org/packages/ba/f8/50c2b7dbc99e05fce5e5b9d9a31f37c988c99acd4e8dedd720b7b8d4011d/virtualenv_clone-0.5.3-py2.py3-none-any.whl
Collecting stevedore (from virtualenvwrapper)
  Downloading https://files.pythonhosted.org/packages/b1/e1/f5ddbd83f60b03f522f173c03e406c1bff8343f0232a292ac96aa633b47a/stevedore-1.31.0-py2.py3-none-any.whl (43kB)
    100% |                                | 51kB 1.6MB/s
Collecting pbr!=2.1.0,>=2.0.0 (from stevedore->virtualenvwrapper)
  Downloading https://files.pythonhosted.org/packages/46/a4/d5c83831a3452713e4b4f126149bc4fbda170f7cb16a86a00ce57ce0e9ad/pbr-5.4.3-py2.py3-none-any.whl (110kB)
    100% |                                | 112kB 1.3MB/s
Collecting six>=1.10.0 (from stevedore->virtualenvwrapper)
  Cache entry deserialization failed, entry ignored
  Using cached https://files.pythonhosted.org/packages/73/fb/00a976f728d0d1fecfe898238ce23f502a721c0ac0ecfedb80e0d88c64e9/six-1.12.0-py2.py3-none-any.whl
Building wheels for collected packages: virtualenvwrapper
  Running setup.py bdist_wheel for virtualenvwrapper ... done
  Stored in directory: /home/diego/.cache/pip/wheels/70/d7/39/a522e494b0e145a1bec42f45a6e542f097c20d0be3ec26866e
Successfully built virtualenvwrapper
```

Poetry

- Gestor de paquetes
- Gestor de dependencias
- Entornos virtuales

Poetry

```
curl -sSL https://install.python-poetry.org | python3 -
```

```
Retrieving Poetry metadata

# Welcome to Poetry!

This will download and install the latest version of Poetry,
a dependency and package manager for Python.

It will add the `poetry` command to Poetry's bin directory, located at:

/home/diego/.local/bin

You can uninstall at any time by executing this script with the --uninstall option,
and these changes will be reverted.

Installing Poetry (1.1.15): Done

Poetry (1.1.15) is installed now. Great!

You can test that everything is set up by executing:

`poetry --version`
```

Poetry

```
poetry init
```

```
Compatible Python versions [^3.8]:  3.10.4

Would you like to define your main dependencies interactively? (yes/no) [yes]
You can specify a package in the following forms:
  - A single name (requests)
  - A name and a constraint (requests@^2.23.0)
  - A git url (git+https://github.com/python-poetry/poetry.git)
  - A git url with a revision (git+https://github.com/python-poetry/poetry.git#develop)
  - A file path (../my-package/my-package.whl)
  - A directory (../my-package/)
  - A url (https://example.com/packages/my-package-0.1.0.tar.gz)

Search for package to add (or leave blank to continue): flask
Found 20 packages matching flask

Enter package # to add, or the complete package name if it is not listed:
 [0] Flask
 [1] Flask-Zipper
 [2] Flask-DB
 [3] flask-beans
 [4] Flask-Sessions
 [5] Flask-SimpleACL
```

Poetry

```
Add a package:

Would you like to define your development dependencies interactively? (yes/no) [yes]
Search for package to add (or leave blank to continue): pytest
Found 20 packages matching pytest

Enter package # to add, or the complete package name if it is not listed:
 [0] pytest
 [1] pytest123
 [2] 131228_pytest_1
 [3] pytest-symbols
 [4] pytest-circleci
 [5] pytest-parallel
 [6] sphinx_pytest
 [7] pytest-vcrpandas
 [8] pytest-briefcase
 [9] pytest-black
 > 0
Enter the version constraint to require (or leave blank to use the latest version):
Using version ^7.1.2 for pytest

Add a package:
```

Poetry

```
Generated file

[tool.poetry]
name = "ejemplo-proyecto-2022"
version = "0.1.0"
description = "Ejemplo Proyecto de Software"
authors = ["Diego Vilches <dvilches@linti.unlp.edu.ar>"]

[tool.poetry.dependencies]
python = "3.10.4"
Flask = "^2.2.2"

[tool.poetry.dev-dependencies]
pytest = "^7.1.2"

[build-system]
requires = ["poetry-core>=1.0.0"]
build-backend = "poetry.core.masonry.api"


Do you confirm generation? (yes/no) [yes]
```

Poetry

Algunos comandos útiles

```
poetry self update
poetry env info
poetry new admin
poetry add flask=2.2.2
poetry add pytest --group dev
poetry run flask --version
poetry build
```

Enlaces útiles

```
https://pypi.org/project/virtualenv/
https://virtualenv.pypa.io/en/stable/
https://github.com/pyenv/pyenv#simple-python-version-management-pyenv
https://docs.conda.io/projects/conda/en/latest/commands.html#conda-vs-pip-vs-virtualenv-commands
https://python-poetry.org
https://www.paradigmadigital.com/dev/poetry-gestor-paquetes-python/
https://www.jetbrains.com/help/pycharm/poetry.html
```