

Ejercicios: Clase 2

1. Módulos y `pip`

- a. Escriba un módulo que a través del paquete `sys` de la biblioteca estándar de Python, obtenga en una variable la versión de Python que se está utilizando y la muestre como output en pantalla.
- b. En el mismo, pruebe importar y utilizar distintos paquetes de la biblioteca estándar de Python. Luego, con `sys.modules` investigue y muestre los módulos importados (evitando módulos privados y submódulos). Muestre estos en pantalla y opcionalmente guárdelos en un archivo.
- c. Utilice `pip` para generar un archivo de requerimientos en formato `.txt`.
- d. Investigue los comandos de `pip` y obtenga la lista de las bibliotecas desactualizadas. Opcionalmente, escriba una función de Bash (o de Python) que actualice estas bibliotecas.

2. `venv` y `pyenv`

- a. Cree un entorno virtual usando `venv`, instale al menos dos bibliotecas en el mismo, y configure Visual Studio Code (o su IDE preferida) de manera tal de poder ejecutar el script del punto 1.a utilizando el ejecutable de Python del entorno virtual.
- b. Instale una versión anterior de Python utilizando `pyenv`, cree un entorno virtual con la misma y repita los pasos del ejercicio 2.a.

3. `setuptools`

- a. Genere dos módulos de Python, uno de ellos con definición de variables y el otro con funciones que utilicen nuevas bibliotecas, o las del ejercicio 2. En otro módulo, escriba código a partir de las funciones y variables definidas. Ejecute este último con el entorno virtual generado.
Para este ejercicio se recomienda, obtener datos de algún módulo de Python o fuente externa (a partir de un archivo o una API), hacer transformación de los mismos y luego guardar su resultado en un archivo.

- b. Investigue el concepto de Linters para Python, instale `pylint` o algún otro paquete de capacidades similares y utilícelo sobre los módulos previamente generados.
- c. Cree requerimientos productivos y de desarrollo para su proyecto. La biblioteca utilizada como Linter deberá pertenecer a los requerimientos de desarrollo.
- d. Genere un paquete a partir de un archivo `setup.py` con lo desarrollado. Modifique los `import`s de sus módulos para reflejar la nueva estructura.
- e. Repita lo realizado utilizando `poetry`.
- f. Opcionalmente, buildee y publique su paquete a `PyPI` utilizando los comandos de `poetry`.