TIC

Módulo 1

LECCIÓN 3

Instalaciones de módulos y paquetes

TIC

Instalador de paquetes de Python

Python se ha convertido en líder en investigación sobre inteligencia artificial. La minería de datos, una de las disciplinas científicas modernas más prometedoras, también utiliza Python. Matemáticos, psicólogos, genetistas, meteorólogos, lingüistas: todas estas personas ya usan Python, o si aún no lo hacen, estamos seguros de que lo harán muy pronto. No hay forma de escapar de esta tendencia.

Por supuesto, no tiene ningún sentido hacer que todos los usuarios de Python escriban su código desde cero, manteniéndolos perfectamente aislados del mundo exterior y de los logros de otros programadores. Esto sería antinatural y contraproducente.

Lo más preferible y eficiente es permitir que todos los miembros de la comunidad de Python intercambien libremente sus códigos y experiencias. En este modelo, nadie está obligado a empezar a trabajar desde cero, ya que existe una alta probabilidad de que alguien más haya estado trabajando en el mismo problema (o uno muy similar).

Para hacer girar este mundo, se deben establecer y mantener en movimiento dos entidades básicas: un repositorio centralizado de todos los paquetes de software disponibles; y una herramienta que permite a los usuarios acceder al repositorio. Ambas entidades ya existen y se pueden utilizar en cualquier momento.

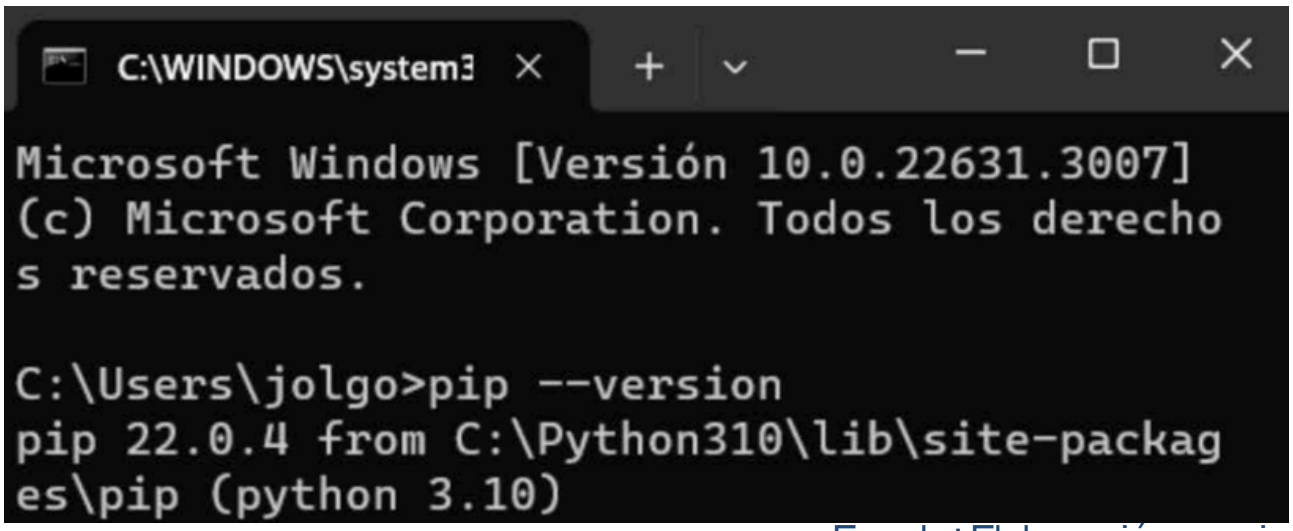
Fuente: Qué es PIP, consultado en enero de 2024, Using Python's pip to Manage Your Projects' Dependencies

El repositorio (o repo para abreviar) que mencionamos se llama PyPI (es la abreviatura de Python Package Index) y lo mantiene un grupo de trabajo llamado Packaging Working Group, una parte de la Python Software Foundation, cuya tarea principal es apoyar a los desarrolladores de Python en la diseminación de código eficiente.

PyPI es completamente gratuito, puedes tomar un código y usarlo; no encontrarás cajero ni guardia de seguridad. Por supuesto, esto no te exime de ser cortés y honesto. Debes obedecer todos los términos de la licencia, así que no olvides leerlos.

Para trabajar con PYPI, es necesario contar con el manejador de paquetes PIP, para verificar que ya cuente con una versión de PIPP instalado solo necesita ejecutar en consola.

```
Unset
pip3 --version
```



Fuente: Elaboración propia

La ausencia de este mensaje puede significar que la variable PATH apunta incorrectamente a la ubicación de los binarios de Python o no apunta a ellos en absoluto; o que no se ha instalado Python aún.

La forma más fácil de reconfigurar la variable PATH es reinstalar Python, indicando al instalador que lo configure en el proceso.

Ahora que ya conocen que se pueden instalar paquetes, es bueno hablar de las dependencias.

Dependencias

Cuando se busca crear un aplicativo que utilice módulo o paquetes de terceros, es posible que ese paquete a su vez utilice otro paquete de otro tercero, y ese otro paquete de otro tercero puede que también utilice un nuevo paquete de un tercero diferente y así sucesivamente. En teoría eso quiere decir que se necesita instalar cada uno de esos paquetes para poder continuar con la funcionalidad buscada pero, en la práctica no es así.

PIP puede descubrir, identificar y resolver todas las dependencias. Además, puede hacerlo de la manera más inteligente, evitando descargas y reinstalaciones innecesarias.

¿Cómo usar PIP?

El comando de ayuda puede indicar qué opciones se tienen con PIP.

```
Unset
//Ayuda general
> pip help

//Ayuda de un comando particular
> pip help install

//Listado de paquetes instalados
> pip list
```

```
C:\Users\jolgo>pip list
Package           Version
----------------- -----------
numpy             1.26.3
pandas            2.1.4
pip               22.0.4
python-dateutil 2.8.2
pytz              2023.3.post1
setuptools        58.1.0
six               1.16.0
tzdata            2023.4
```

Fuente: Elaboración propia

¿Qué tal probar ahora con el comando show, qué resultado se obtuvo?