

Diferencia entre python y anaconda:

La gestión de la descarga de los paquetes alojados en PyPI su instalación y control de dependencias

asociadas se puede realizar a través de dos herramientas principales:

****pip** (<https://pypi.org/project/pip/>), el gestor de paquetes estándar de Python.

****conda** (<https://docs.conda.io/en/latest/>) para la gestión de paquetes y dependencias.

Aunque algunas de las funcionalidades de estas dos herramientas se superponen, tienen diseños y propósitos distintos. Mientras pip es la herramienta recomendada para instalar paquetes desde PyPI, conda es un gestor multiplataforma que instala y gestiona paquetes desde el repositorio de Anaconda, y Anaconda Cloud (<https://anaconda.cloud/>). Mientras que con pip el software a instalar se compila a partir del código fuente de la librería elegida (lo que puede requerir la instalación previa de compiladores compatibles), los paquetes que maneja conda son directamente binarios (pre-compilados), y durante la instalación no ejecuta procesos de compilación. Otra diferencia es que los paquetes conda no se limitan a software Python, sino que también puede gestionar librerías de C o C++, paquetes R u otros softwares. Otra diferencia clave entre pip y conda es que esta última tiene la capacidad de crear entornos aislados en los que alojar diferentes versiones de Python y/o paquetes. Esto resulta particularmente útil si trabajamos con diferentes herramientas y versiones de las mismas, ya que se pueden dar conflictos entre las dependencias que impidan su instalación en un entorno único.