

12. Entornos Virtuales y Paquetes

Las aplicaciones en Python usualmente hacen uso de paquetes y módulos que no forman parte de la librería estándar. Las aplicaciones a veces necesitan una versión específica de una librería, debido a que dicha aplicación requiere que un bug particular haya sido solucionado o bien la aplicación ha sido escrita usando una versión obsoleta de la interfaz de la librería.

Esto significa que tal vez no sea posible para una instalación de Python cumplir los requerimientos de todas las aplicaciones. Si la aplicación A necesita la versión 1.0 de un módulo particular y la aplicación B necesita la versión 2.0, entonces los requerimientos entran en conflicto e instalar la versión 1.0 o 2.0 dejará una de las aplicaciones sin funcionar.

La solución a este problema es crear un entorno virtual, un directorio que contiene una instalación de Python de una versión en particular, además de unos cuantos paquetes adicionales.

Diferentes aplicaciones pueden entonces usar entornos virtuales diferentes. Para resolver el ejemplo de requerimientos en conflicto citado anteriormente, la aplicación A puede tener su propio entorno virtual con la versión 1.0 instalada mientras que la aplicación B tiene otro entorno virtual con la versión 2.0.

Si la aplicación B requiere que actualizara la librería a la versión 3.0, esto no afectará el entorno virtual de la aplicación A.

12.2. Creando Entornos Virtuales



El script usado para crear y manejar entornos virtuales es pyvenv. pyvenv normalmente instalará la versión mas reciente de Python que tengas disponible; el script también es instalado con un número de versión, con lo que si tienes múltiples versiones de Python en tu sistema puedes seleccionar una versión de Python específica ejecutando python3 o la versión que desees.

Para crear un entorno virtual, decide en que carpeta quieres crearlo y ejecuta el módulo venv como script con la ruta a la carpeta:

```
python3 -m venv tutorial-env
```

Esto creará el directorio tutorial-env si no existe, y también creará directorios dentro de él que contienen una copia del intérprete de Python y varios archivos de soporte.

Una ruta común para el directorio de un entorno virtual es .venv. Ese nombre mantiene el directorio típicamente escondido en la consola y fuera de vista mientras le da un nombre que explica cuál es el motivo de su existencia. También permite que no colisione con los ficheros de definición de variables de entorno .env que algunas herramientas soportan.

Una vez creado el entorno virtual, podrás activarlo.

En Windows, ejecuta:

```
tutorial-env\Scripts\activate.bat
```

