# **Entornos virtuales para Python**







## ¿Por qué usar entornos virtuales?

- → Son <u>cajas</u> aisladas para los proyectos.
- → Cada entorno puede tener sus propias dependencias (librerías).
- → Los entornos no interfieren entre sí ni con la instalación global de Python.
- → Mantiene el sistema limpio.





#### venv

Viene incluido con Python 3 por lo que no es necesario instalarlo.

Crea un entorno:

python3 -m venv nombre\_del\_entorno

Activa el entorno en Linux/macOS:

source nombre\_del\_entorno/bin/activate





#### Activa el entorno en Windows:

nombre\_del\_entorno\Scripts\activate

### Instala paquetes:

pip install nombre\_del\_paquete

#### Desactiva el entorno:

nombre\_del\_entorno\Scripts\activate





## conda/miniconda

Descarga: <a href="https://www.anaconda.com/">https://www.anaconda.com/</a>

Miniconda es una versión más ligera de Anaconda que solo incluye el comando <u>conda</u> y sus dependencias.







#### Crea un entorno:

conda create -- name nombre\_del\_entorno

#### Activa el entorno en Linux/macOS:

conda activate nombre\_del\_entorno

#### Activa el entorno en Windows:

activate nombre\_del\_entorno





### Instala paquetes:

conda install nombre\_del\_paquete

Desactiva el entorno:

conda deactivate

Lista los entornos virtuales:

conda env list



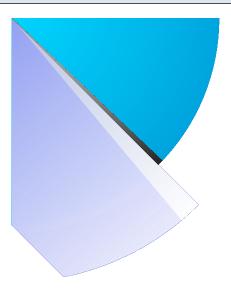


## poetry

Crea automáticamente un entorno virtual para cada proyecto.

Instalación (hay que instalar primero pipx):

pipx install poetry







## Crea un proyecto y lo activa:

poetry new nombre\_del\_proyecto

Esto crea un nuevo directorio con la estructura básica de un proyecto de Python y un archivo pyproject.toml donde se gestionan las dependencias.

Activa un proyecto que no está activado:

poetry shell





#### Desactiva el entorno:

exit

### Instala paquetes:

poetry add nombre\_del\_paquete

#### Lista los entornos virtuales:

poetry env list



