

UV: An extremely fast Python package and project manager, written in Rust.

👤 Integrantes:

- 👤 Fabián Pérez
- 👤 Redactor: Juan Calderón

📄 Material de apoyo:

- 👤 **Diapositivas:** [Ver presentacion](#)
-

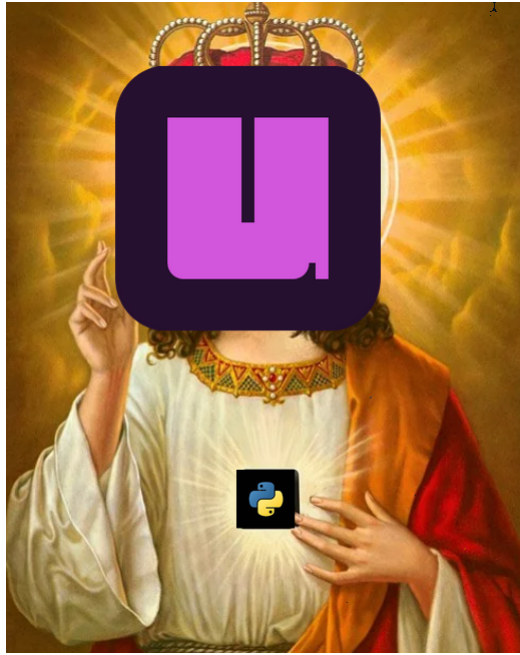
Introducción: ¿Por qué deberías cambiar tu flujo de trabajo de Python?

Si trabajas con **Python**, sabes que los **entornos virtuales** son la columna vertebral de cualquier proyecto serio. Nos permiten aislar las dependencias y evitar el temido "infierno de dependencias" que surge al trabajar en múltiples proyectos con diferentes requisitos.

Tradicionalmente, hemos dependido de `venv` para crear el entorno y de `pip` para instalar los paquetes. Pero, ¿y si te dijera que existe una herramienta **hasta 10 veces más rápida** que puede manejar ambos procesos de forma integrada y moderna?

Conoce a `uv`, la nueva herramienta de gestión de paquetes y entornos que está revolucionando la comunidad Python. En este tutorial, aprenderás paso a paso a instalar `uv`, crear entornos virtuales

y gestionar dependencias de forma relámpago.



⚙️ I. Preparación: Instalando uv

Antes de empezar a volar con la gestión de entornos, necesitamos instalar la herramienta uv.

Requisitos Previos

Asegúrate de tener una versión reciente de **Python** instalada en tu sistema.

1. Opción Recomendada: Instalación del Binario (Ultrarrápida)

Esta es la forma más rápida y estable de instalar el binario de uv directamente en tu sistema:

Para Linux y macOS (usando curl):

```
$ curl -Lsf https://astral.sh/uv/install.sh | sh
```

Para Windows (usando PowerShell):

```
$ powershell -ExecutionPolicy Bypass -c "irm https://astral.sh/uv/install.ps1 | iex"
```

2. Opción Alternativa: Instalación como Paquete Python

Si prefieres instalarlo a través de Python (por ejemplo, dentro de otro entorno virtual o usando `pipx`), también es posible:

```
$ pip install uv
```

Para verificar que la instalación fue exitosa, ejecuta:

```
$ uv --version
```

II. Crear el Entorno Virtual: `uv`

`venv` vs. `uv init`

`uv` ofrece dos comandos potentes para empezar tu proyecto. Ambos son mucho más rápidos que el método tradicional.

Opción A: Creación Pura del Entorno con

`uv venv` (Reemplazo directo de `python -m venv`)

Este comando es ideal si solo quieres la carpeta `.venv/` en un directorio existente:

```
$ mkdir mi-proyecto-rapido
$ cd mi-proyecto-rapido
$ uv venv
```

Opción B: Inicialización Completa con `uv init` (Recomendado para Nuevos

Proyectos)

Este comando es más amplio y genera archivos clave para la configuración moderna de Python:

```
$ mkdir mi-proyecto-nuevo
$ cd mi-proyecto-nuevo
$ uv init
# Esto crea: .venv/, pyproject.toml, .gitignore, y más.
```

Activación del Entorno

Una vez creado (con cualquiera de los comandos anteriores), debes **activar** el entorno virtual:

Sistema Operativo	Comando de Activación
Linux/macOS	<code>\$ source .venv/bin/activate</code>
Windows (CMD)	<code>\$.venv\Scripts\activate</code>
Windows (PowerShell)	<code>\$.venv\Scripts\Activate.ps1</code>

Verás el nombre del entorno (`.venv`) aparecer al inicio de tu línea de comandos, indicando que está activo.

III. Instalación de Herramientas de Deep Learning con `uv add`

Aquí es donde `uv` brilla, permitiendo instalar librerías complejas como **PyTorch** de manera ultrarrápida.

1. Instalación de NumPy y PyTorch

Para ejecutar un proyecto de *Deep Learning*, instalaremos la base (**NumPy**) y el framework (**PyTorch**) junto con librerías de soporte (**torchvision** y **torchaudio**).

a. Instalación de NumPy:

```
$ uv add numpy
```

b. Instalación de PyTorch (con soporte CUDA/GPU para rendimiento):

```
# Reemplaza 'cu121' con la versión de CUDA instalada en tu sistema (ej. cu118, cu121, etc.)  
$ uv add torch torchvision torchaudio --index-url https://download.pytorch.org/whl/cu121
```

“ **Nota:** La URL del índice es crucial para obtener la versión correcta y optimizada para la GPU.

2. Eliminación de Paquetes

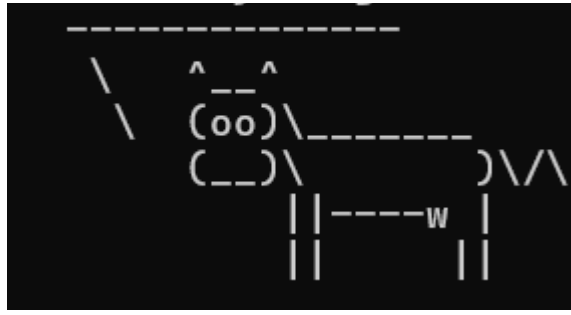
Para desinstalar un paquete (ej. si decides cambiar de PyTorch a TensorFlow):

```
$ uv remove torch
```

IV. Ejecutando Herramientas Comunes con `uv`

`uv` no solo instala paquetes; también puede ejecutar herramientas directamente desde el entorno virtual sin instalarlas globalmente, usando `uv run` o `uvx`.

Herramienta	Propósito	Comando <code>uv</code>
Jupyter Notebook	Entorno interactivo para desarrollo de ciencia de datos.	<code>\$ uv run jupyter notebook</code>
Ruff	Linter y formateador de código Python de alto rendimiento.	<code>\$ uv run ruff check .</code>
pycowsay	Una librería de prueba divertida para verificar la ejecución.	<code>\$ uv run pycowsay "hola soy Jorge"</code>



V. Más comandos de `uv`

Para una gestión completa del entorno virtual, aquí tienes comandos adicionales útiles:

Comando	Propósito
<code>\$ uv tree</code>	Muestra el árbol de dependencias del proyecto de forma jerárquica.
<code>\$ uv python list</code>	Lista todas las instalaciones de Python disponibles en tu sistema.
<code>\$ uv add --dev ruff</code>	Agrega un paquete (ej. Ruff) específicamente como dependencia de desarrollo (no requerida en producción).
<code>\$ uv run python</code>	Inicia el intérprete REPL (shell) de Python dentro del entorno activo.
<code>\$ uv sync</code>	Sincroniza el entorno virtual con el archivo de bloqueo (lock file) del proyecto, asegurando que las versiones sean exactas.
<code>\$ uv self update</code>	Actualiza el ejecutable de <code>uv</code> a su última versión.
<code>\$ uv run --env-file .env app.py</code>	Ejecuta un script de Python (<code>app.py</code>) cargando variables de entorno desde un archivo <code>.env</code> .

VI. Desactivación y Conclusión

Desactivación del Entorno

Cuando termines de trabajar, desactiva el entorno escribiendo `deactivate`:

Conclusión

uv no es solo una herramienta más rápida; es un paso hacia la modernización de todo el ecosistema de dependencias de Python. Al integrar la creación del entorno (**venv**) con la gestión de paquetes (**pip**), ofrece un flujo de trabajo más limpio, consistente y eficiente.

Si valoras tu tiempo y buscas optimizar tus procesos de desarrollo en Python, **uv es la herramienta que necesitas implementar hoy mismo.**

Referencias

- **Instalación de **uv** e instrucciones de uso**
 - Comunidad de **discord** de **uv**
 - Repositorio de **GitHub** de **uv**
-

Revision #5

Created 29 August 2025 23:08:16 by Jorge Garcia

Updated 4 December 2025 03:45:29 by Juan Calderón