



**hazme una guía sobre como se crea un entorno virtual de python, como se instalan los módulos, como se instala un archivo de requierements o como se crea este archivo. Y también como se pone en marcha el control de versiones con git y como se conecta a Github**

Para un flujo típico de proyecto Python moderno: crea un entorno virtual, instala dependencias (manual o con requirements.txt) y luego versiona el código con Git conectado a GitHub.<sup>[1]</sup> <sup>[2]</sup>

## 1. Crear y usar entorno virtual

Entra en la carpeta de tu proyecto y crea el entorno virtual con el módulo estándar venv. Cambia el comando según sistema:<sup>[3]</sup> <sup>[1]</sup>

- En Linux/macOS:

- cd ruta\al\proyecto
- python3 -m venv venv (o python -m venv venv según tu instalación)<sup>[1]</sup> <sup>[3]</sup>
- Activar: source venv/bin/activate (verás el prefijo (venv) en la terminal).<sup>[3]</sup>

- En Windows:

- cd ruta\al\proyecto
- py -m venv venv o python -m venv venv<sup>[4]</sup> <sup>[3]</sup>
- Activar: venv\Scripts\activate (aparecerá (venv) en el prompt).<sup>[4]</sup> <sup>[3]</sup>

Para desactivar el entorno basta ejecutar deactivate en la terminal.<sup>[3]</sup>

## 2. Instalar módulos con pip

Con el entorno virtual activo, cualquier instalación con pip queda aislada del sistema.<sup>[1]</sup> <sup>[3]</sup>

- Instalar un paquete concreto:

- pip install requests (ejemplo).<sup>[1]</sup>

- Ver qué está instalado en tu entorno:

- pip list o pip freeze.<sup>[5]</sup> <sup>[6]</sup>

- Desinstalar un paquete:

- `pip uninstall requests.`<sup>[1]</sup>

Es importante que el prompt tenga (`venv`) o el nombre de tu entorno, para asegurarte de que los paquetes se instalan ahí y no globalmente.<sup>[3]</sup> <sup>[1]</sup>

### 3. requirements.txt: crear e instalar

El archivo `requirements.txt` sirve para definir dependencias del proyecto y poder replicarlas en otro entorno.<sup>[7]</sup> <sup>[6]</sup>

- Crear automáticamente a partir de lo instalado en tu entorno virtual:
  - `pip freeze > requirements.txt` (desde la raíz del proyecto).<sup>[8]</sup> <sup>[5]</sup>
- Crear/editar manualmente:
  - Crea `requirements.txt` y escribe una dependencia por línea, por ejemplo `requests==2.32.0.`<sup>[7]</sup>

Para instalar todo lo definido en `requirements.txt` dentro de un entorno (normalmente recién creado y activado):

- `pip install -r requirements.txt` desde la carpeta donde está el archivo.<sup>[5]</sup> <sup>[7]</sup>

Si alguna dependencia falla, se puede editar el archivo para quitar o cambiar esa línea y volver a ejecutar la instalación.<sup>[6]</sup> <sup>[8]</sup>

### 4. Iniciar Git en el proyecto

En la carpeta del proyecto, el flujo básico con Git es:

- Inicializar el repo local:
  - `git init.`<sup>[9]</sup> <sup>[2]</sup>
- Ver el estado:
  - `git status.`<sup>[2]</sup>
- Añadir archivos al índice:
  - `git add .` (todo) o `git add archivo.py.`<sup>[9]</sup> <sup>[2]</sup>
- Crear el primer commit:
  - `git commit -m "Primer commit".`<sup>[2]</sup> <sup>[9]</sup>

Antes de hacer commits, crea un `.gitignore` para evitar subir cosas como `venv/`, ficheros temporales, etc. Un contenido típico mínimo es incluir `venv/` y archivos de caché de Python.<sup>[10]</sup> <sup>[2]</sup>

## 5. Conectar con GitHub

Primero crea un repositorio vacío en GitHub (sin subir nada todavía) usando la interfaz web.  
Luego, en tu repo local:<sup>[2]</sup>

- Configura tu identidad (una vez por máquina):
  - `git config --global user.name "Tu Nombre"`
  - `git config --global user.email "tu_email@example.com".[2]`
- Añade el remoto (HTTPS de ejemplo):
  - `git remote add origin https://github.com/tu_usuario/tu_repo.git.[9] [2]`
- Verifica:
  - `git remote -v.[11] [2]`
- Sube la rama principal (main o master, según lo que uses):
  - `git branch -M main (si quieres llamarla main).[2]`
  - `git push -u origin main.[9] [2]`

A partir de ahí, el ciclo típico es:

- Hacer cambios en el código.
- `git status` para ver qué cambió.<sup>[2]</sup>
- `git add .`
- `git commit -m "Descripción del cambio"`
- `git push` para enviar a GitHub.<sup>[9] [2]</sup>

Si quieras, en un siguiente mensaje se puede montar una “chuleta” conjunta (script de comandos típicos) para un proyecto nuevo: crear venv, generar requirements.txt y dejar todo subido a GitHub de una vez.

\*\*

1. <https://docs.python.org/es/3.13/tutorial/venv.html>
2. <https://docs.github.com/es/repositories/creating-and-managing-repositories/quickstart-for-repositories>
3. <https://pythonscouts.com/entorno-virtual-python/>
4. <https://platzi.com/tutoriales/1927-intro-selenium/12448-crea-entornos-virtuales-en-windows-en-3-pasos/>
5. <https://gist.github.com/kamikaze-lab/7d5987ff86223e1bf686>
6. <https://learn.microsoft.com/es-es/visualstudio/python/managing-required-packages-with-requirements-txt?view=visualstudio>
7. <https://rukbotoland.com/blog/como-instalar-paquetes-python-con-requirementstxt/>
8. <https://learn.microsoft.com/es-es/visualstudio/python/managing-required-packages-with-requirements-txt?view=vs-2022>
9. <https://www.steamacademy.com.co/post/conectar-una-carpeta-con-un-repositorio-en-github-usando-git>

10. [https://www.youtube.com/watch?v=PXADC\\_SaiM4](https://www.youtube.com/watch?v=PXADC_SaiM4)
11. <https://es.stackoverflow.com/questions/148159/git-c%C3%B3mo-conectar-un-repositorio-local-nuevo-con-un-repositorio-en-github-existe>
12. <https://docs.python.org/es/3.8/library/venv.html>
13. <https://micro.recursospython.com/recursos/como-crear-un-entorno-virtual-venv.html>
14. <https://www.youtube.com/watch?v=39ij9YX4JfE>
15. <https://platzi.com/tutoriales/1378-python-practico/3077-crear-ambiente-virtual-con-virtualenv/>
16. <https://www.youtube.com/watch?v=xB2V5J3G42U>
17. <https://www.youtube.com/watch?v=N9PdRkXOQ4w>
18. <https://help.dreamhost.com/hc/es/articles/115000695551-Instalar-y-usar-virtualenv-con-Python-3>
19. <https://tancode.es/blog/generar-instalar-requirements-txt-python-django>
20. <https://www.youtube.com/watch?v=APpm0o5YoCo>