



hazme una guía sobre como se crea un entorno virtual de python, como se instalan los módulos, como se instala un archivo de requirements 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. ^[1] ^[2]

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: ^[3] ^[1]

- En Linux/macOS:
 - `cd ruta/al/proyecto`
 - `python3 -m venv venv` (o `python -m venv venv` según tu instalación) ^[1] ^[3]
 - Activar: `source venv/bin/activate` (verás el prefijo `(venv)` en la terminal). ^[3]
- En Windows:
 - `cd ruta\al\proyecto`
 - `py -m venv venv` o `python -m venv venv` ^[4] ^[3]
 - Activar: `venv\Scripts\activate` (aparecerá `(venv)` en el prompt). ^[4] ^[3]

Para desactivar el entorno basta ejecutar `deactivate` en la terminal. ^[3]

2. Instalar módulos con pip

Con el entorno virtual activo, cualquier instalación con `pip` queda aislada del sistema. ^[1] ^[3]

- Instalar un paquete concreto:
 - `pip install requests` (ejemplo). ^[1]
- Ver qué está instalado en tu entorno:
 - `pip list` o `pip freeze`. ^[5] ^[6]
- Desinstalar un paquete:

- `pip uninstall requests.`^[1]

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.^{[3] [1]}

3. requirements.txt: crear e instalar

El archivo `requirements.txt` sirve para definir dependencias del proyecto y poder replicarlas en otro entorno.^{[7] [6]}

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

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.^{[5] [7]}

Si alguna dependencia falla, se puede editar el archivo para quitar o cambiar esa línea y volver a ejecutar la instalación.^{[6] [8]}

4. Iniciar Git en el proyecto

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

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

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.^{[10] [2]}

5. Conectar con GitHub

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

- 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.^[2]
- `git add .`
- `git commit -m "Descripción del cambio"`
- `git push` para enviar a GitHub.^{[9] [2]}

Si quieres, 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://rukbottoiland.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
11. <https://es.stackoverflow.com/questions/148159/git-cómo-conectar-un-repositorio-local-nuevo-con-un-repositorio-en-github-exis>
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://tincode.es/blog/generar-instalar-requirements-txt-python-django>
20. <https://www.youtube.com/watch?v=APpm0o5YoCo>