



si monta la “chuleta” conjunta

Aquí va una chuleta compacta para arrancar un proyecto Python desde cero con venv, requirements, Git y GitHub.[\[1\]](#) [\[2\]](#)

1. Crear proyecto y entorno virtual

```
# Crear carpeta y entrar
mkdir mi_proyecto
cd mi_proyecto

# Crear entorno virtual (Linux/macOS)
python3 -m venv venv

# Crear entorno virtual (Windows)
# py -m venv venv (o python -m venv venv)

# Activar entorno (Linux/macOS)
source venv/bin/activate

# Activar entorno (Windows)
# venv\Scripts\activate
```

2. Instalar paquetes y generar requirements.txt

```
# Con el entorno ACTIVO
pip install requests flask # ejemplo

# Ver lo instalado
pip list

# Generar requirements.txt
pip freeze > requirements.txt
```

Si alguien clona el repo o montas un entorno nuevo:

```
# Activar el entorno virtual del proyecto
# (source venv/bin/activate o venv\Scripts\activate)

# Instalar dependencias del proyecto
pip install -r requirements.txt
```

3. Crear .gitignore básico para Python

Crea un archivo `.gitignore` en la raíz del proyecto con algo así (adaptable):

```
venv/  
__pycache__/  
*.py[cod]  
.log  
.env
```

Esto evita subir el entorno virtual, cachés y archivos temporales al repositorio.[\[3\]](#) [\[4\]](#)

4. Inicializar Git y primer commit

```
# Inicializar repo local  
git init  
  
# Ver estado  
git status  
  
# Añadir archivos (incluye requirements.txt y .gitignore)  
git add .  
  
# Crear primer commit  
git commit -m "Inicializa proyecto Python con venv y requirements"
```

Si no lo configuraste antes, define tu usuario global (una sola vez por máquina):

```
git config --global user.name "Tu Nombre"  
git config --global user.email "tu_email@example.com"
```

5. Crear repo en GitHub y conectar

1. En GitHub, crea un repositorio vacío `mi_proyecto` (sin README, sin `.gitignore`, sin licencia).[\[5\]](#) [\[6\]](#)
2. Copia la URL HTTPS, por ejemplo: `https://github.com/tu_usuario/mi_proyecto.git`.[\[7\]](#)

En tu terminal, dentro de la carpeta del proyecto:

```
# Añadir remoto  
git remote add origin https://github.com/tu_usuario/mi_proyecto.git  
  
# Asegurar que la rama principal se llama main (opcional, pero estándar)  
git branch -M main
```

```
# Subir al remoto  
git push -u origin main
```

A partir de ahí, el ciclo diario es:

```
# Haces cambios en el código...  
  
git status  
git add .  
git commit -m "Describe el cambio"  
git push
```

Si quieras, se puede hacer otra chuleta solo con comandos habituales de Git (ramas, stash, rebase, etc.) adaptada a tu flujo de trabajo.

**

1. <https://www.hostinger.com/es/tutoriales/comandos-de-git>
2. <https://github.com/samuel-emrys-blogs/python-dev-env>
3. <https://www.datacamp.com/es/tutorial/gitignore>
4. <https://j2logo.com/gitignore-para-que-sirve/>
5. <https://welearndata.com/git/primeros-pasos-con-git-github/>
6. <https://docs.github.com/es/repositories/creating-and-managing-repositories/quickstart-for-repositories>
7. https://entrenamiento-frameworks-web-python.readthedocs.io/es/latest/leccion1/entorno_desarrollo.html
8. https://es.linkedin.com/posts/midudev_he-creado-una-chuleta-de-git-para-programadores-activity-7365365468666245122-BIQb
9. <https://docs.python.org/es/3.8/library/venv.html>
10. <https://docs.github.com/es/actions/how-tos/deploy/deploy-to-third-party-platforms/python-to-azure-app-service>
11. <https://stackoverflow.com/questions/64747598/gitignore-for-a-python-venv-based-in-the-root-of-a-git-project>
12. <https://x.com/midudev/status/1919753781165588769>
13. <https://docs.github.com/es/actions/reference/workflows-and-actions/workflow-syntax>
14. <https://stackoverflow.com/questions/42306410/gitignore-syntax-how-to-exclude-virtualenv-sub-directories>
15. <https://www.facebook.com/midudev.frontend/posts/chuleta-de-git-para-programadores/1218489043618659/>
16. https://www.reddit.com/r/learnpython/comments/16koffh/python_dependency_management_on_github/
17. <https://www.youtube.com/watch?v=tIFF78yIIHQ>
18. <https://api.rpubs.com/justorfc/Guiacodespacescopicot>
19. https://www.reddit.com/r/Python/comments/1o2viq3/uv_cheatsheet_with_most_commonuseful_commands/
20. <https://j2logo.com/comandos-git-principales/>

21. <https://github.com/fgrweb/chuleta-git>