

Guía Profesional v4.0 – Flujo Completo GitHub + VS Code

Esta guía documenta el flujo completo de trabajo profesional con GitHub y Visual Studio Code para proyectos Python: desde el inicio de una tarea, commits, sincronización, integración, taggeo de versiones estables y creación de Releases en GitHub. Aplicable a proyectos Signal Mapper, Canalizaciones, control_documental_mvp y SSAA.

1. Flujo General

- Trabajar siempre desde develop.
- Crear ramas por feature, bugfix o refactor.
- Nunca trabajar directo en main.
- main siempre representa una versión estable.

2. Inicio de una Tarea

- `git checkout develop`
- `git pull --rebase origin develop`
- `git checkout -b feature/nombre-claro`

3. Commit

- `git status`
- `git add .`
- `git commit -m "feat(modulo): descripcion clara"`

4. Sincronización

- `git pull --rebase origin develop`
- Resolver conflictos si aparecen
- `git rebase --continue`

5. Push

- `git push -u origin feature/nombre-claro`
- Luego: `git push`

6. Integración a develop

- Usar Pull Request en GitHub (recomendado)
- O merge local controlado
- develop queda actualizado

7. Paso a main

- `git checkout main`
- `git pull --rebase origin main`
- `git merge --no-ff develop`
- `git push origin main`

8. Tag de Versión Estable

- `git tag -a vX.Y.Z -m "Release vX.Y.Z"`
- `git push origin vX.Y.Z`

9. Release en GitHub

- Ir a GitHub → Releases → Draft new release
- Seleccionar el tag creado
- Agregar notas de la versión
- Publicar el release

10. Buenas Prácticas

- Commits pequeños y claros
- Tags solo en main
- Historial limpio y trazable