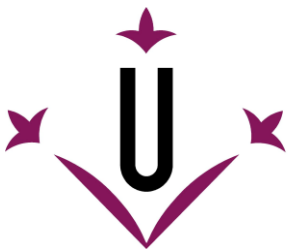


Recetas Git

Grado en Ingeniería Informática

Roberto García (roberto.garcia@udl.cat)
<http://griho.udl.cat/about/members/garcia-roberto.html>



Universitat de Lleida
Escola Politècnica Superior



Copiarse el repositorio

- `git clone git@github.com:user/repo.git`
 - Crearlo en GitHub y clonarlo facilita cosas como tener un `.gitignore` para Python, pero fundamentalmente enlazar repositorio local y remoto
 - Si no, sería necesario hacer:
`git remote add origin git@github.com:user/repo.git`

Introducir cambios

- Cambiar cosas:
 - Editar ficheros existentes, añadir, borrar,...
- Añadir nuevos ficheros a Git:
 - `git add <file|folder>`
 - Borrar de Git (pero no el fichero):
 - `git rm --cached <file>`
 - `git rm -r --cached <folder>`
- Para ver que falta por añadir:
 - `git status`

Guardar cambios (local)

- `git commit -m "descripción del cambio"`
- Ver últimos cambios:
 - `git log`
- Atajo
 - Para ficheros ya añadidos a Git, si los cambiamos:
 - `git commint -a -m "..."`, equivalente a:
 - `git add <file|dir>`
 - `git commit -m "..."`

Actualizar git local (con remoto)

- git fetch origin
- git merge origin/master
- Atajo:
 - git pull origin master

Compartir cambios (remoto)

- git push origin master
 - Siempre después de haber comprobado si hay cambios remotos con un git pull

Si nos equivocamos...

- Corregir el último commit (antes de hacer push) con un nuevo commit (y opcionalmente mensaje):
 - `git commit --amend`

Git avanzado

- Añadir nueva funcionalidad a “repo” de manera ordenada:
- Descargarlo:
 - `clone git@github.com:user/repo.git`
- Añadir rama nueva funcionalidad:
 - `git checkout -b my_new_feature`
- Editar código y guardar cambios:
 - `git commit ...`
- Mantenerse actualizado con lo último en master:
 - `git rebase master`
- Seguir trabajando y guardando cambios, y manteniéndose al día
 - `git commit ...`
 - `git rebase master`
- Con esto se completa la nueva funcionalidad
 - `git checkout master`
 - `git merge my_new_feature`