

Comandos básicos de Git

Git init

Crea un repositorio

```
git init
```

Git Clone

Para descargarse todo el contenido del repositorio

```
git clone <https://link-con-nombre-del-repositorio>
```

Git status

Nos da toda la información sobre la rama que tenemos seleccionada actualmente. Si esta actualizada, si hay pendiente algún cambio para confirmar, enviar, o descargar, archivos en preparación (staged, con add hecho) o sin seguimiento (untracked), archivos creados, modificados o eliminados

```
git status
```

Git add

Cuando realizamos cambios en un archivo, solo ocurren en local y deben de marcarse para queda en estado de seguido para añadir en el próximo commit. Se pueden marcar uno, o todos

```
git add <archivo>
```

```
git add .
```

Git commit

Este comando implica establecer un punto donde los cambios realizados son versionados, y por tanto establecemos un punto de control al que podremos volver. Requiere de un mensaje asociado, que debe ser descriptivo para otros desarrolladores de los cambios que se están implementando.

```
git commit -m "mensaje de confirmación"
```

Git reset

Nos permite vaciar el seguimiento o un archivo determinado. Es el contrario a Git add

```
git reset
```

```
git reset <archivo>
```

Nos permite retirar un commit (solo localmente), conservando el código, o descartándolo

```
git reset --mixed
```

```
git reset --hard
```

Git diff

Muestra los cambios que tiene el archivo, respecto del último commit de trabajo:

```
git diff <archivo>
```

Git Branch

Para trabajar con ramas del repositorio

Obtener nombre de la rama donde estamos trabajando

```
git branch
```

Crear rama

```
git branch <nombre-de-la-rama>
```

Borrar rama

```
git branch -d <nombre-de-la-rama>
```

Git checkout

Es el comando que nos permite cambiar de una rama a otra, o de commit dentro de la misma rama

```
git checkout <nombre-de-la-rama o id-de-commit>
```

También permite restaurar los archivos a la versión del último commit:

```
git checkout <nombre archivo o "-- ." para todos>
```

Git merge

Sirve para traerse los cambios **desde** una rama **especificada** en el comando, **hasta la que está actualmente seleccionada**:

```
git merge <nombre-de-la-rama-desde-la-que-traer-los-cambios>
```

Git tag

Crea una etiqueta de versión en el commit actual de trabajo:

```
git tag <nombre-tag>
```

Git Pull

Sirve para traerse los últimos cambios que hay en el origen o repositorio remoto

```
git pull <nombre-tag>
```

Git push

Este comando hace que las versiones de cambios definidas con el commit, no solo se guarden de forma local, si no que queden publicadas en remoto

```
git push
```

```
git push <nombre-remoto> <nombre-de-tu-rama>
```

Si vas a publicar por primera vez la rama

```
git push --set-upstream <nombre-remoto> <nombre-de-tu-rama>
```

Git log

Permite ver el historial de commits

```
git log
```

```
git log --graph
```

Git stash

Permiten enviar cambios a una pila de cambios considerados “borradores”

Enviar cambios al stash

```
git stash
```

Traer último cambio del stash (último apilado)

```
git branch
```

Ver lista de cambios en la pila de borradores

```
git stash pop
```

Recuperar un cambio en concreto

```
git stash apply <id-borrador>
```

Borrar un borrador apilado

```
git stash drop <id>
```