

- Git clone: sirve para copiar un repositorio en tu máquina local. He usado este comando tras realizar el fork del repositorio de la práctica 1. El fork permite copiar un repositorio pero está en GitHub, por eso para tenerlo en el ordenador realizas un git clone.

```
● @AmaliaVaro →/workspaces/p1 (main) $ cd /workspaces
● @AmaliaVaro →/workspaces $ git clone https://github.com/AmaliaVaro/p1-fork
  Cloning into 'p1-fork'...
remote: Enumerating objects: 6, done.
remote: Counting objects: 100% (1/1), done.
remote: Total 6 (delta 0), reused 0 (delta 0), pack-reused 5 (from 1)
Receiving objects: 100% (6/6), done.
○ @AmaliaVaro →/workspaces $
```

- Git status: muestra el estado actual del repositorio, indica en qué rama estás, los archivos que has modificado y que archivos están listos para hacer commit. En esta práctica lo he utilizado para comprobar que no tenía que hacer commit.

```
● @AmaliaVaro →/workspaces/p1 (main) $ git status
  On branch main
    Your branch is up to date with 'origin/main'.

    nothing to commit, working tree clean
```

- Git add: te permite seleccionar los cambios, añadiéndolos al staging área, preparando así el próximo commit. En esta práctica he utilizado este comando para indicar que quería todos los cambios para el commit.

```
● @AmaliaVaro →/workspaces/p1 (main) $ mkdir src
○ @AmaliaVaro →/workspaces/p1 (main) $ cd /src
  bash: cd: /src: No such file or directory
● @AmaliaVaro →/workspaces/p1 (main) $ cd /workspaces/p1/src
● @AmaliaVaro →/workspaces/p1/src (main) $ touch index.html
@AmaliaVaro →/workspaces/p1/src (main) $ cat > index.html
<!DOCTYPE html>
<html>
<head><title></title></head>
<body></body>
</html>
@AmaliaVaro →/workspaces/p1/src (main) $ cat index.html
<!DOCTYPE html>
<html>
<head><title></title></head>
<body></body>
</html>
@AmaliaVaro →/workspaces/p1/src (main) $ cd /workspaces/p1
● @AmaliaVaro →/workspaces/p1 (main) $ git add .
```

- Git commit: añade los cambios al repositorio local con un mensaje corto y descriptivo de los cambios. Deben realizarse de manera periódica y no hacer commits que tengan muchos cambios. En la práctica he realizado el commit tras haber seleccionado con add todos los cambios, para tenerlo

en mi repositorio local.

```
● @AmaliaVaro →/workspaces/p1 (main) $ git commit -m "edicion de index.html"
[main 9763a67] edicion de index.html
 1 file changed, 5 insertions(+)
  create mode 100644 src/index.html
```

- Git push: sirve para subir los cambios que tengo en el repositorio local al repositorio remoto, en este caso GitHub. Lo he utilizado para subir el commit que he realizado a GitHub de tal forma que todo el mundo pueda verlo.

```
● @AmaliaVaro →/workspaces/p1 (main) $ git push origin main
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 2 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (4/4), 395 bytes | 395.00 KiB/s, done.
Total 4 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/AmaliaVaro/p1
  4302958..9763a67  main -> main
```

- Git checkout: sirve para cambiar de rama. Lo hemos usado para crear una nueva rama y escribir en el fichero. Tras subir los cambios al repositorio remoto hemos vuelto a cambiar de rama, en este caso a la rama main.

```
● @AmaliaVaro →/workspaces/p1 (main) $ git checkout -b feat/add-body
Switched to a new branch 'feat/add-body'
@AmaliaVaro →/workspaces/p1 (feat/add-body) $ git add .
@AmaliaVaro →/workspaces/p1 (feat/add-body) $ git commit -m "feat: add body"
[feat/add-body 9642cd3] feat: add body
 1 file changed, 1 insertion(+), 1 deletion(-)
@AmaliaVaro →/workspaces/p1 (feat/add-body) $ git push origin feat/add-body
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 2 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (4/4), 410 bytes | 410.00 KiB/s, done.
Total 4 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
remote:
remote: Create a pull request for 'feat/add-body' on GitHub by visiting:
remote:     https://github.com/AmaliaVaro/p1/pull/new/feat/add-body
remote:
To https://github.com/AmaliaVaro/p1
 * [new branch]      feat/add-body -> feat/add-body
@AmaliaVaro →/workspaces/p1 (feat/add-body) $ git cheout main
git: 'cheout' is not a git command. See 'git --help'.

The most similar command is
  checkout
● @AmaliaVaro →/workspaces/p1 (feat/add-body) $ git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.
```