Definitions

Made by: Filip, Ignacio, Diego, Alex.

Git checkout, this command switches you to a different branch within your local repository. It doesn't directly access remote repositories.

Git merge is often used in conjunction with git checkout to select the current branch and then merge them together.

Git push allows you to push your commits from your local branch of your local git repository to the remote repository.

Git pull is better than fetch and merge, its more used the git pull, because git pull it’s a mix of the other two.

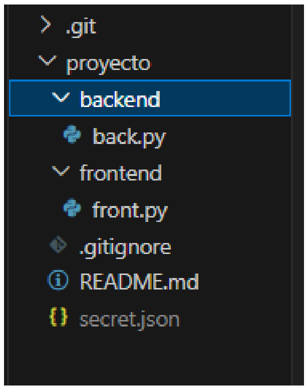
Git stash takes uncommitted changes, both those that are ready and those that are not, saves them aside for later use and then undoes them in the code you are working on. You can use stash pop and stash apply.

Git rebase The main reason for using it is to maintain a linear Project history, that is, to see how the main branch has progressed since it was started.

Git fetch allows you to update your local repository with the latest changes from a remote repository, but without merging those changes into your local working branches.

Steps

- We have a main master branch, with the files, folders, etc...

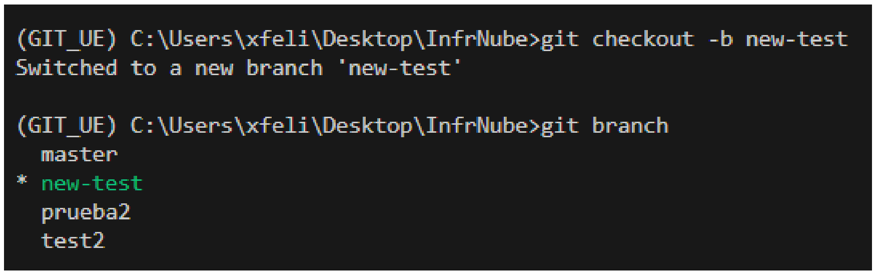


```
> .git
v proyecto
  v backend
      back.py
  v frontend
      front.py
    .gitignore
    README.md
    secret.json
```

- We push to send all the files to the main branch from the repository.

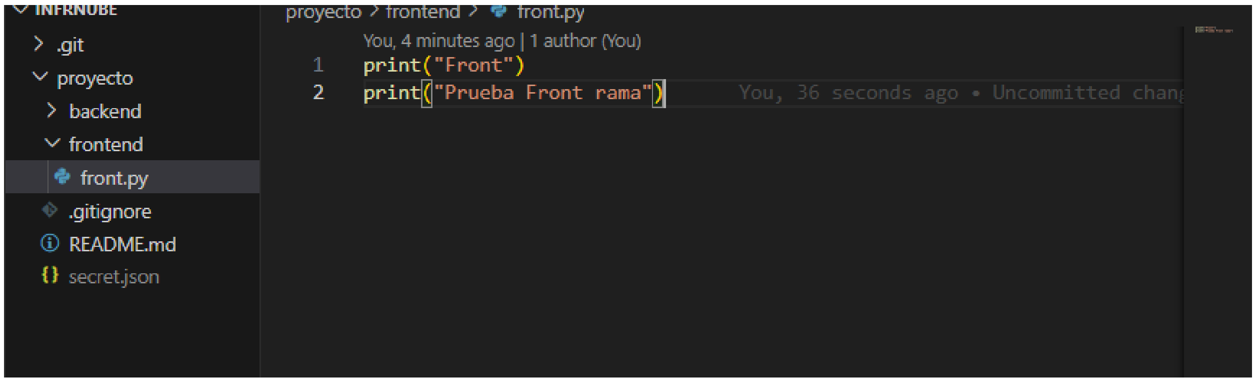
>**git push** test3 (Repository) master (Main branch)

- We create a new branch, we modify files or create new ones.



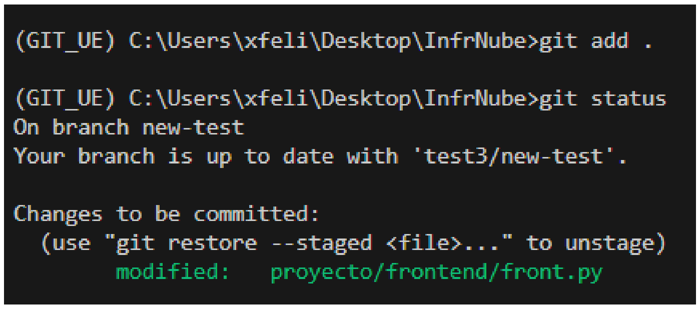
```
(GIT_UE) C:\Users\xfeli\Desktop\InfrNube>git checkout -b new-test
Switched to a new branch 'new-test'

(GIT_UE) C:\Users\xfeli\Desktop\InfrNube>git branch
  master
* new-test
  prueba2
  test2
```



```
print("Front")
print("Prueba Front rama")        You, 36 seconds ago • Uncommitted chang
```

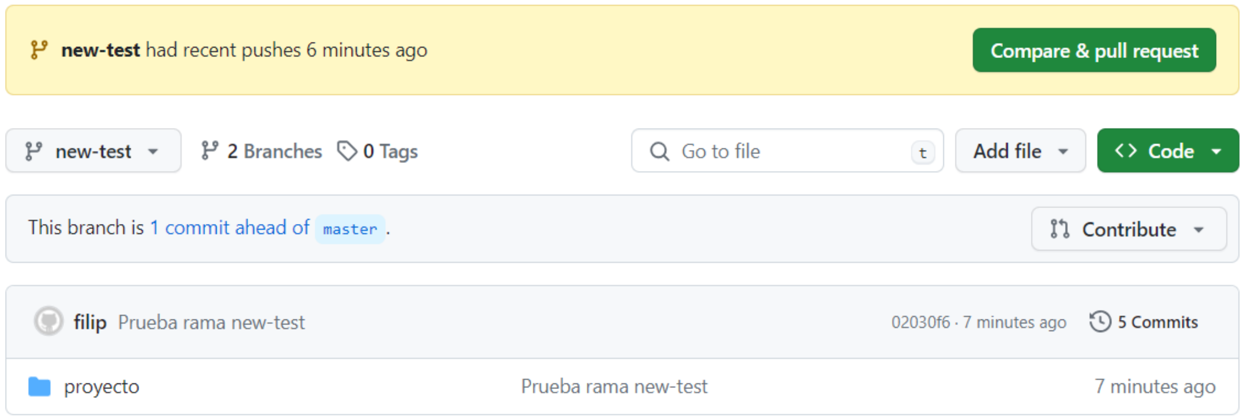
- First, we add the changes and then push to upload them in the new branch created from the repository.



```
(GIT_UE) C:\Users\xfeli\Desktop\InfrNube>git add .

(GIT_UE) C:\Users\xfeli\Desktop\InfrNube>git status
On branch new-test
Your branch is up to date with 'test3/new-test'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   proyecto/frontend/front.py
```



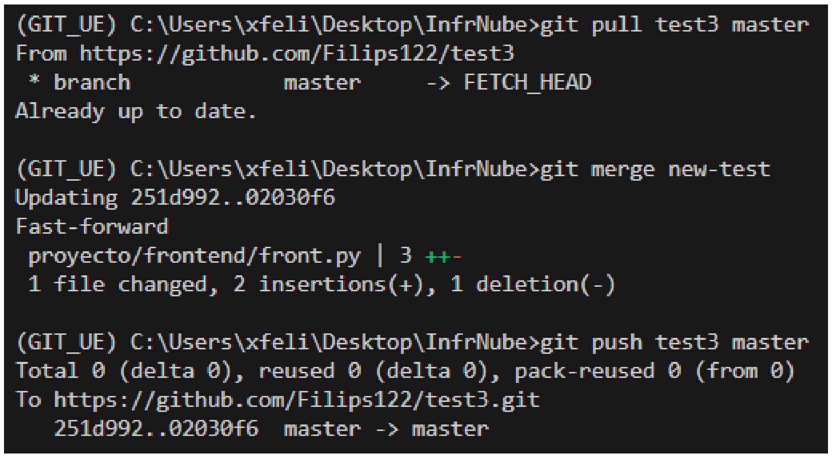
This branch is 1 commit ahead of master.

filip Prueba rama new-test	02030f6 · 7 minutes ago	5 Commits
proyecto	Prueba rama new-test	7 minutes ago

- We switch to the main branch and pull to have the original files and avoid conflicts.

>**git checkout** master

- While we are in the main branch, we merge the second branch to main one.



```
(GIT_UE) C:\Users\xfeli\Desktop\InfrNube>git pull test3 master
From https://github.com/Filips122/test3
 * branch            master     -> FETCH_HEAD
Already up to date.

(GIT_UE) C:\Users\xfeli\Desktop\InfrNube>git merge new-test
Updating 251d992..02030f6
Fast-forward
 proyecto/frontend/front.py | 3 ++-
 1 file changed, 2 insertions(+), 1 deletion(-)

(GIT_UE) C:\Users\xfeli\Desktop\InfrNube>git push test3 master
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/Filips122/test3.git
   251d992..02030f6  master -> master
```

- After we merge both branches we make a push to send it to repository.

- If we use fetch it would be like merge but instead of branches, between repositories and it is a good practice if you don't want to "mess it up" in the main repository.

>**git fetch** test3

- This is to see the difference between actual branch and the repository you want to merge at.

>**git diff** master (Actual branch) test3/master (Repository that you want to merge)

- We do rebase if we want to have a cleaner history and not with all the changes as in merge.

>**git checkout** new-test

>**git rebase** main

>**git rebase** --abort