gitCreate a new branch

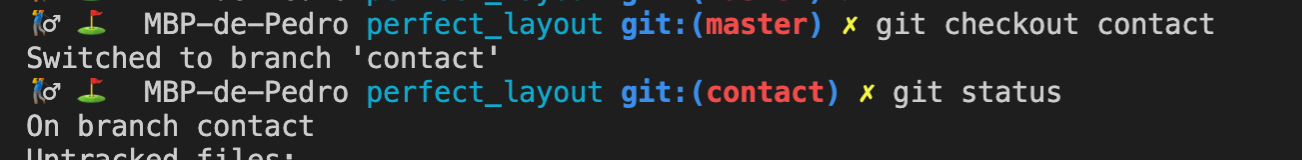
git branch <branch name>

See the branchs

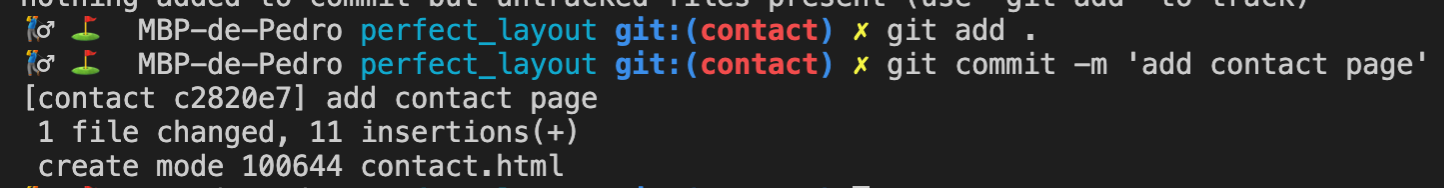
git branch



Move a another branch

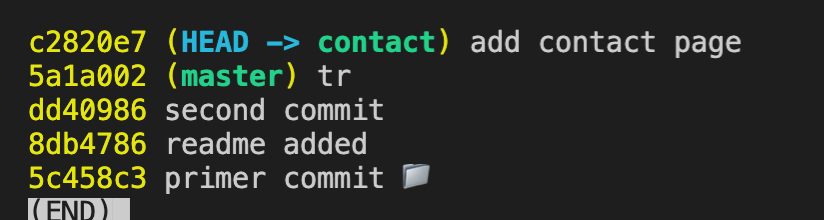


The new commits will be on ‘contact’ branch.



And now, HEAD (last commit in the active branch) is on ‘contact’ branch.

git log --oneline

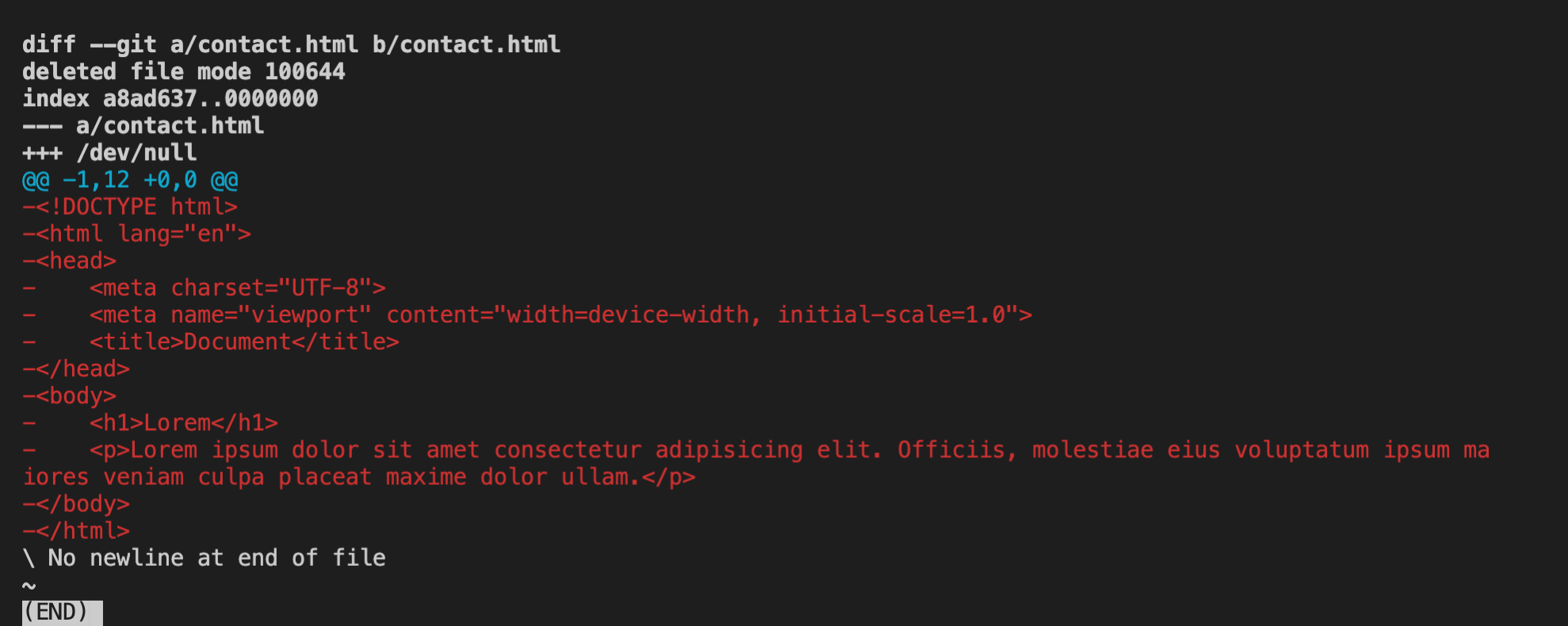


Differences between branches

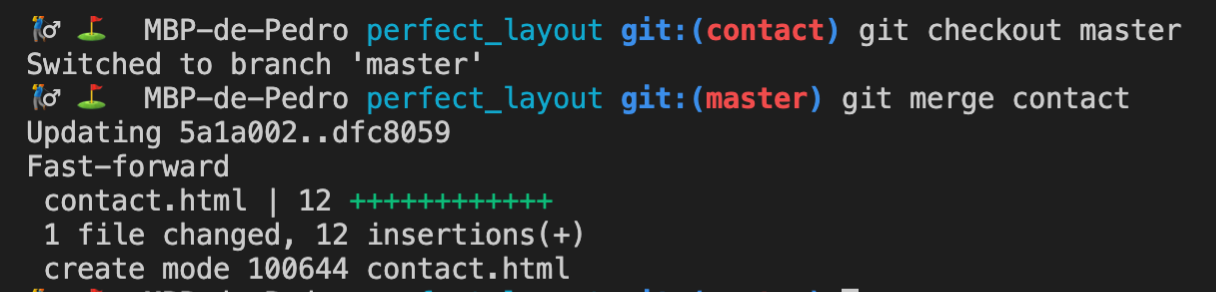
git diff <branch name> <branch name>

e.g.

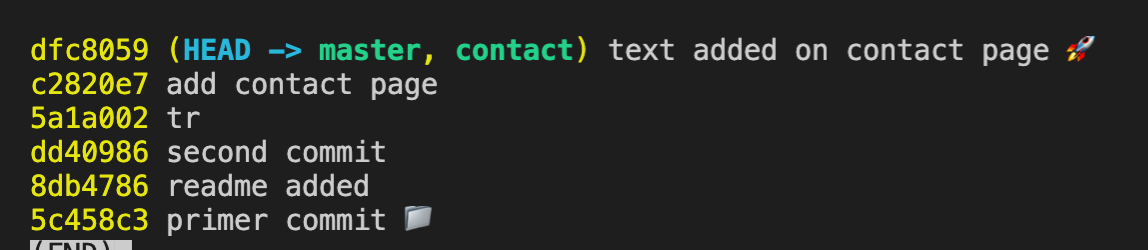
git diff contact master



Fast forward merge



Now HEAD is on both, contact & master.

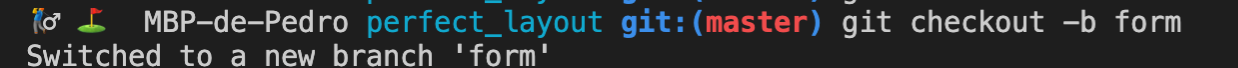


Delete branch

git branch -d <branch name>

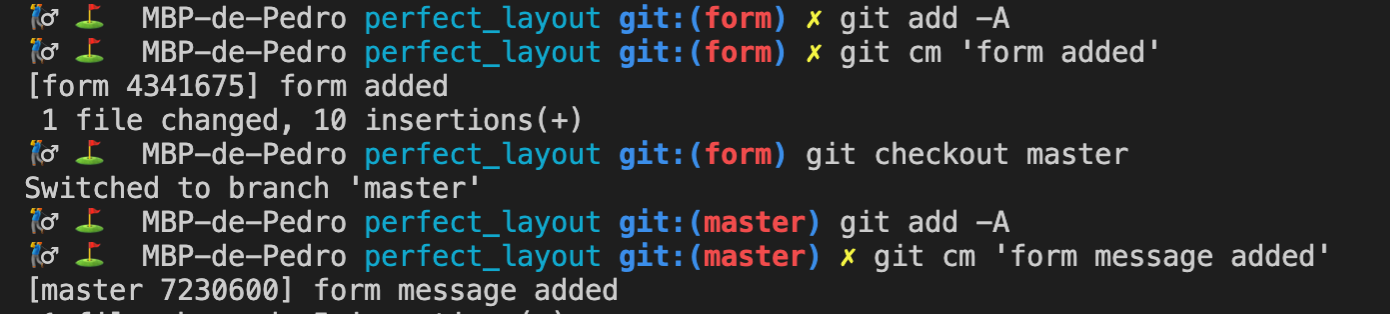
Create and move to a new branch in one step

git checkout -b <branch name>

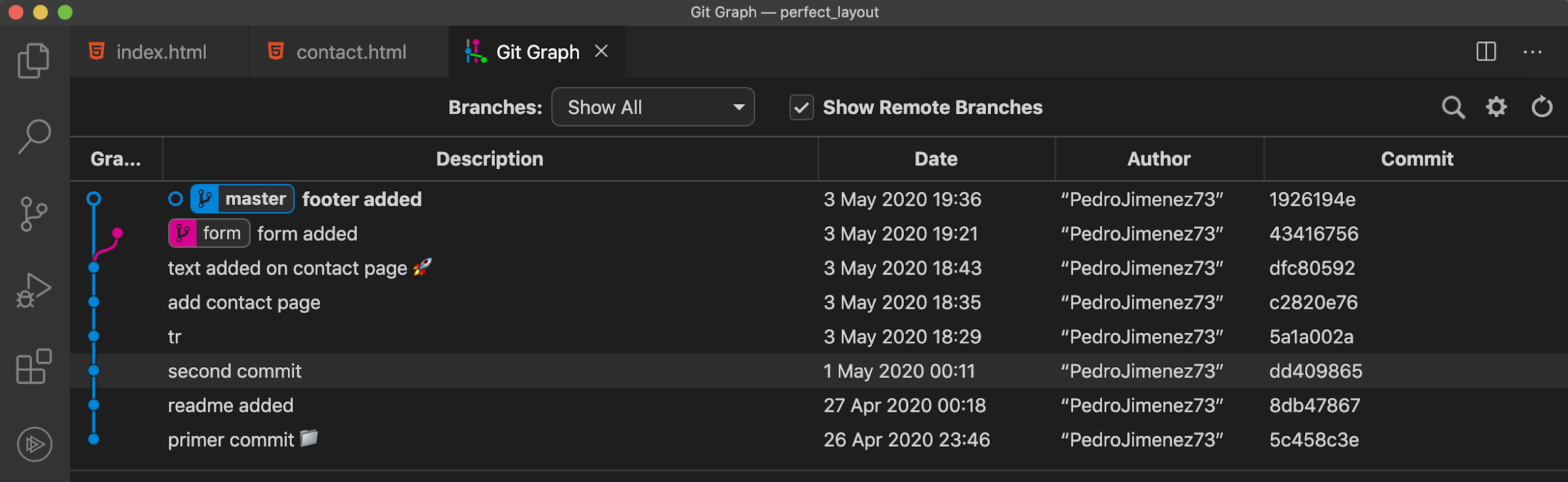


Auto merge

Set two commits on both branches

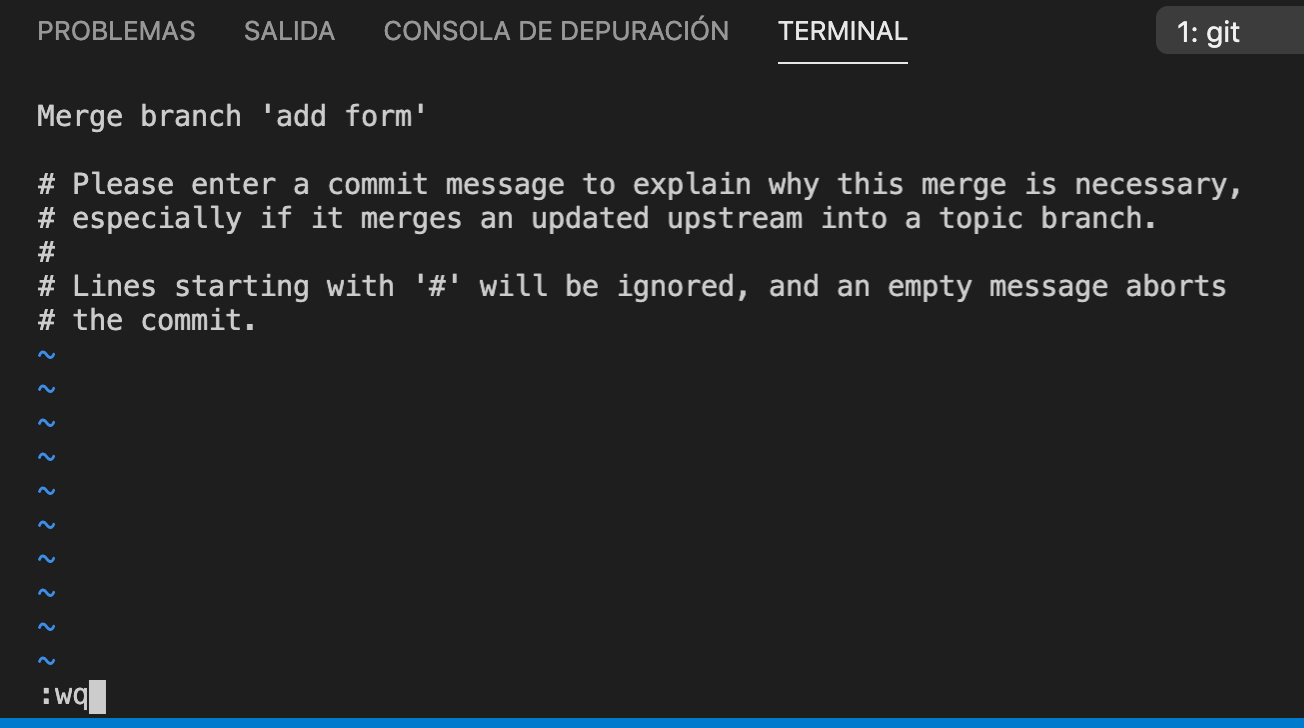


see how branches diverging

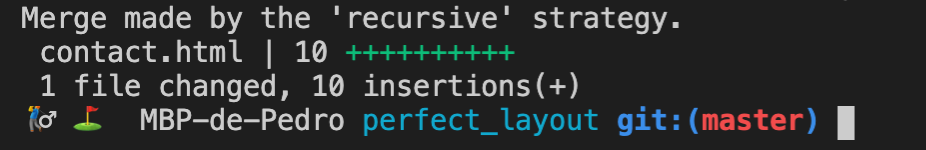


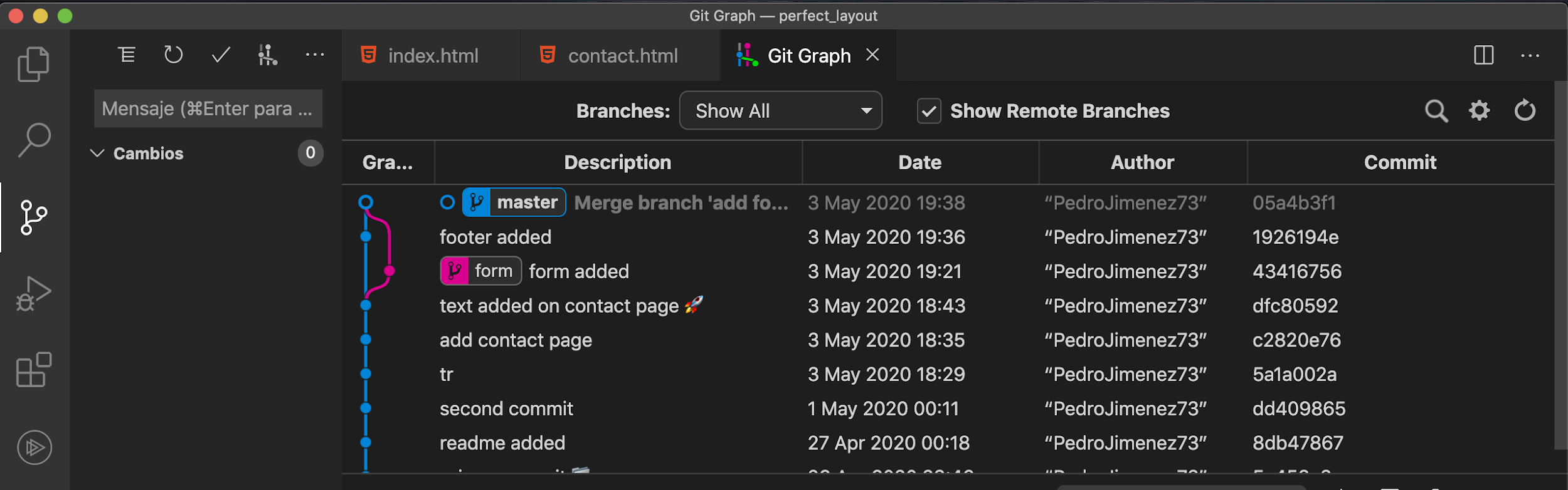
Now…

git merge form



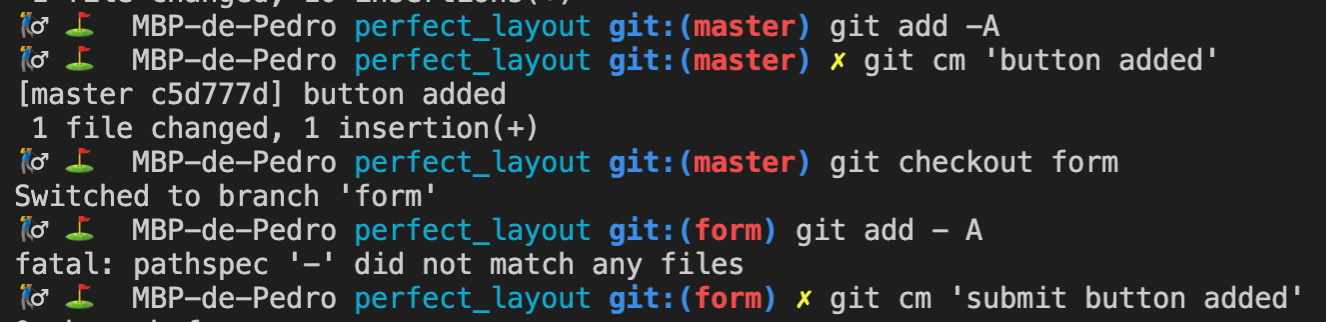
save with esc + :wq





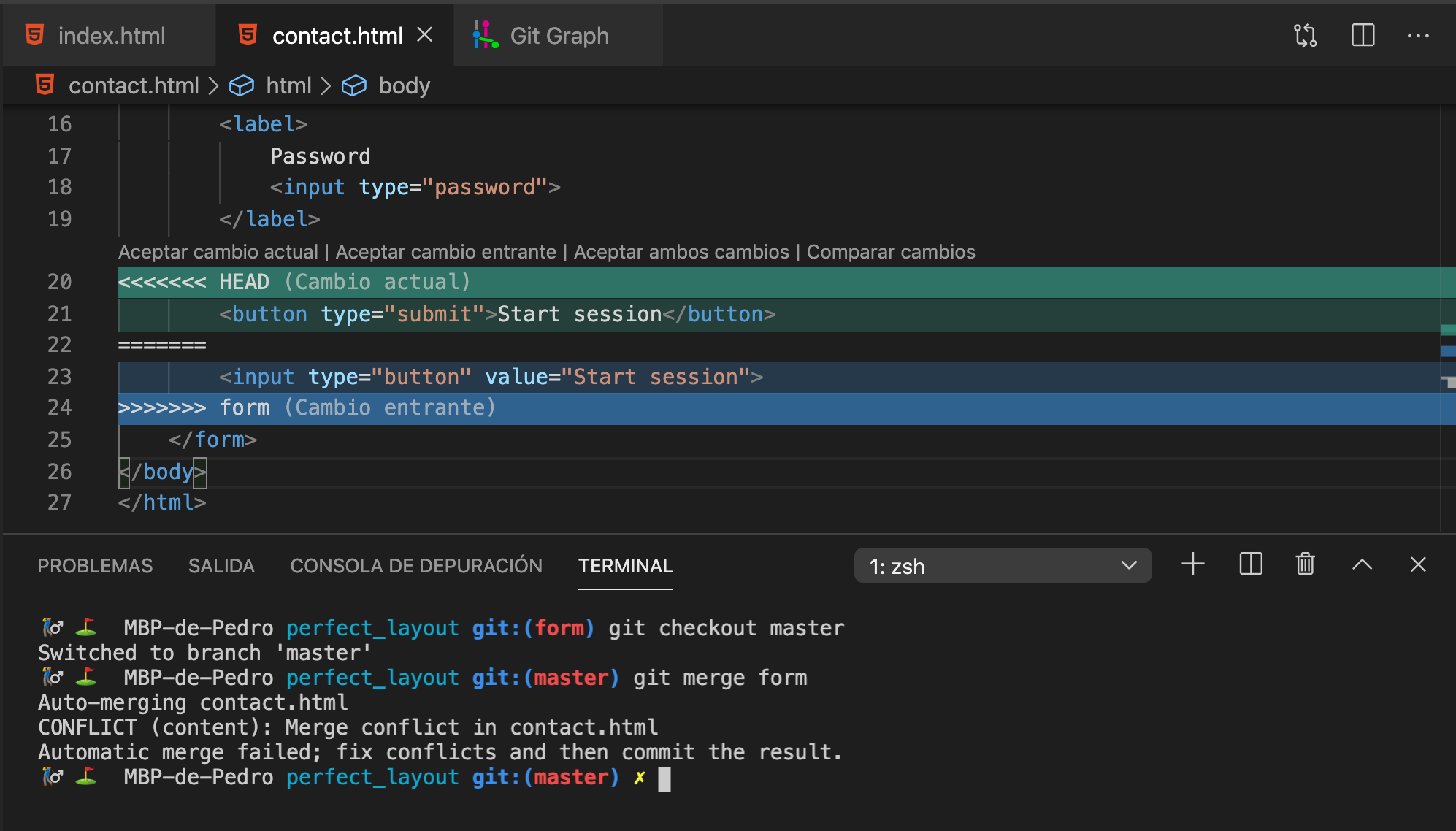
Manual merge (to resolve conflicts)

Make two changes on same file in differents branches.

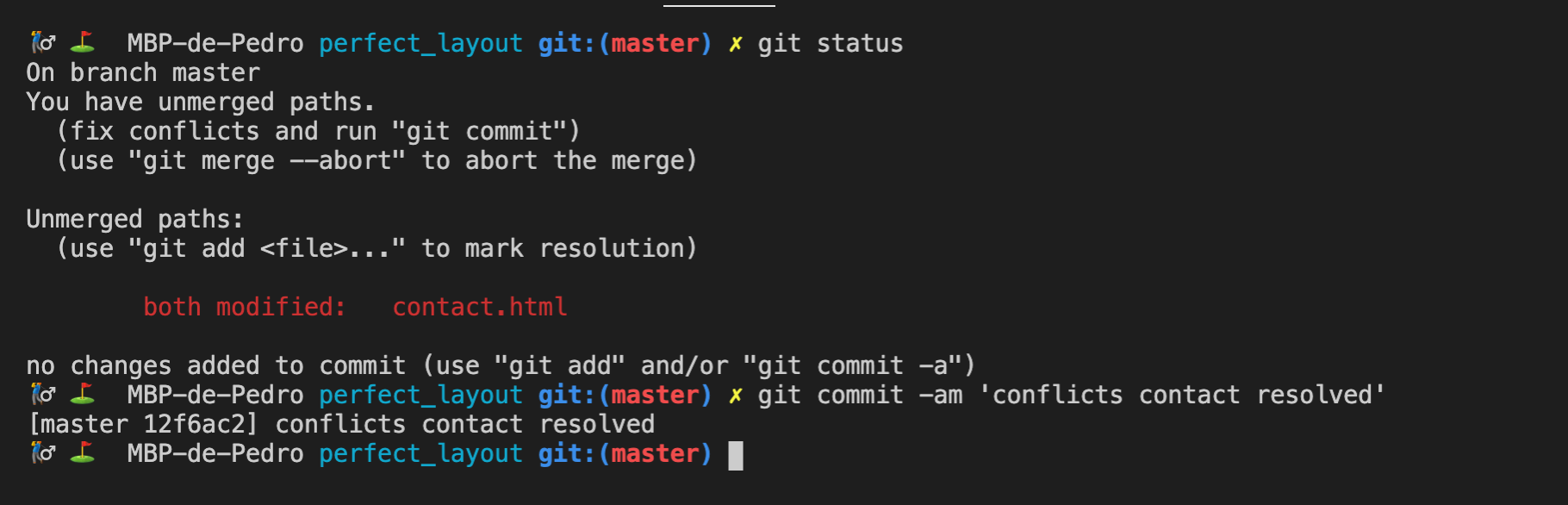




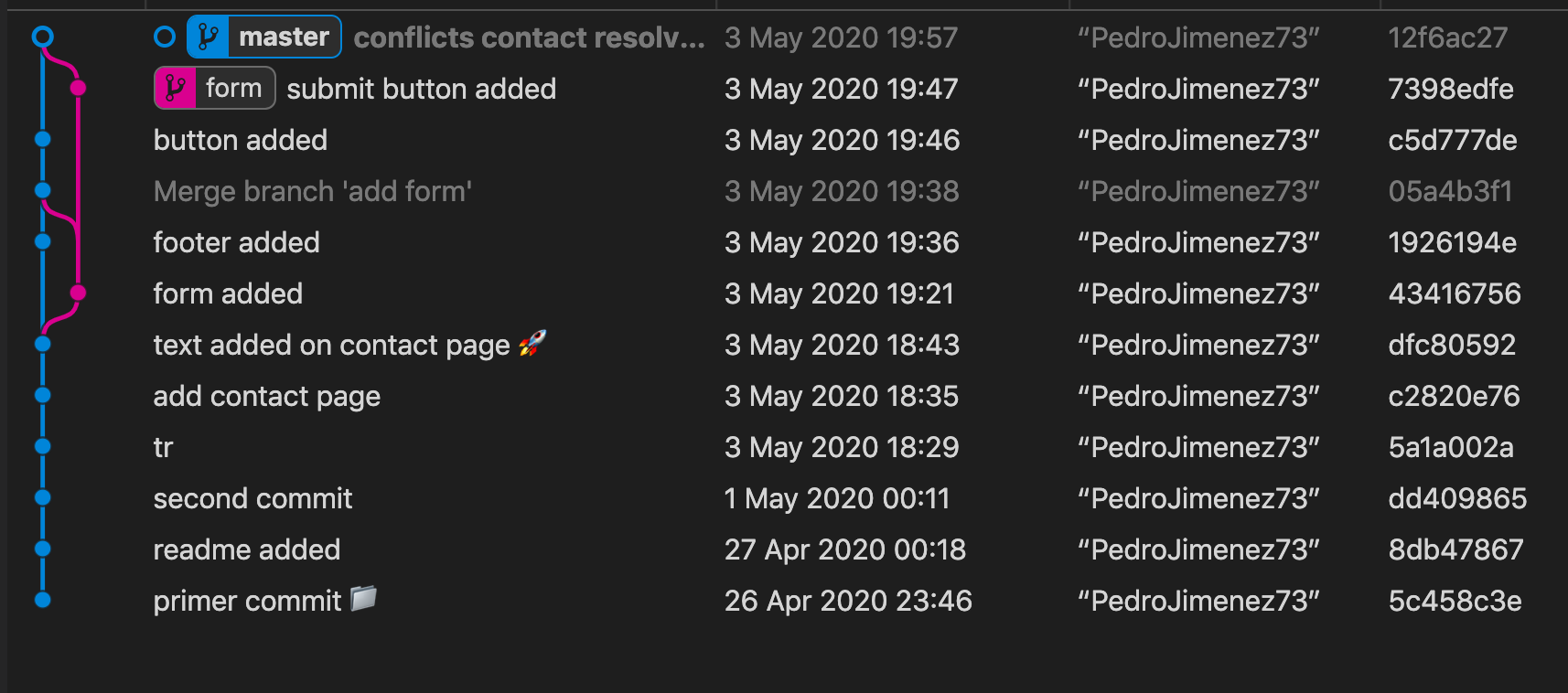
Try to merge…



To resolve conflicts, just delete the git anotations on file and save, and add file to staged and set a new commit.



And…



Finally, delete the branch:

git branch -d form

Recover branch deleted

git checkout -b form 45b2c6f9e790e7aca20feb2ce0055114594248ab

Create tag

git tag -a v1.0.0 -m 'First release'

See tags

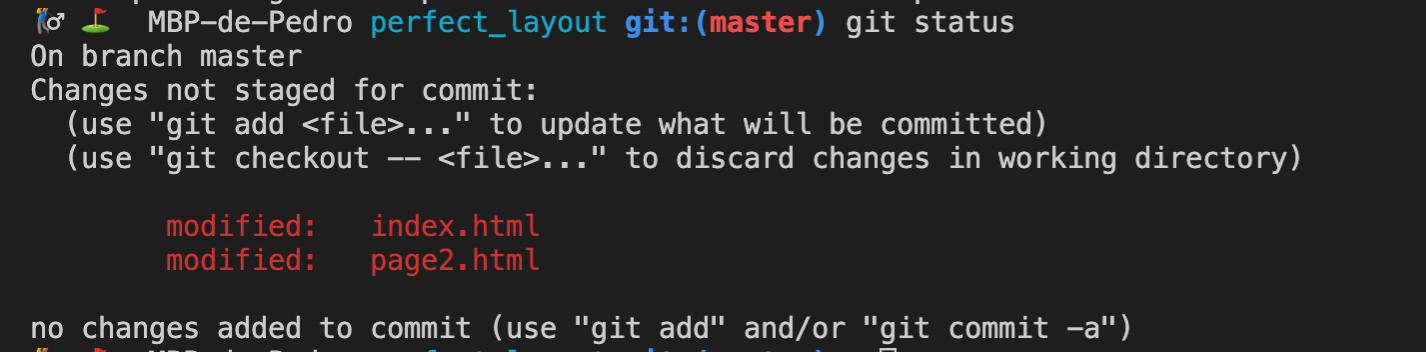
git tag

Delete tag

git tag -d v1.0.0

Create stash (hide work in progress files)

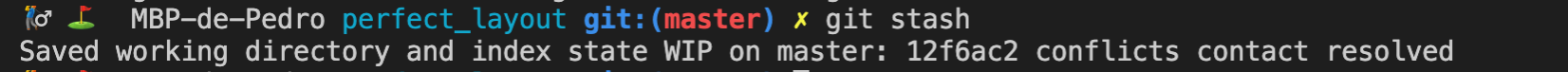
Make any changes on several files and git status

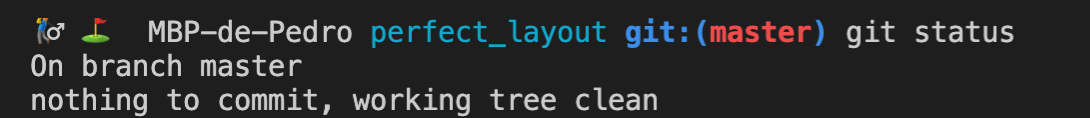


At this time if you need fix a bug in others files and don’t want commit this changes, you can set aside this work in progress to a stash and later get them back.

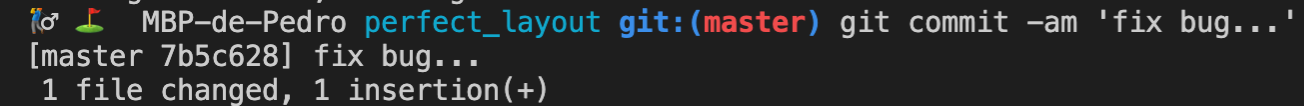
So, before commit other things, you can:

git stash





Now you can fix the bugs and commit:





And recover the work in progress stash:

