Pràctica GIT

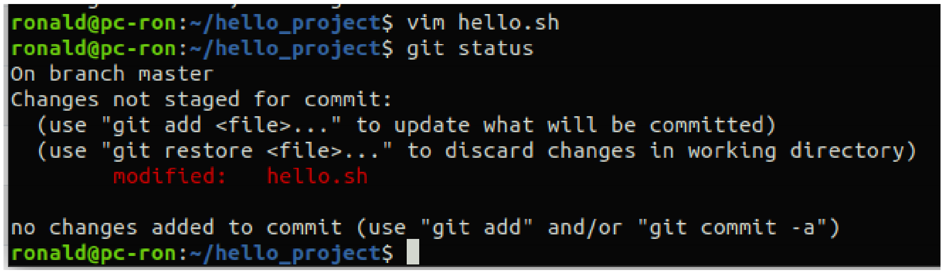
Haureu d'entregar un document amb el nom: **M05-UF2-PT-GIT-Cognoms-Nom.ext**

Per cada **exercici** heu de fer una alguna captura o copiar el codi que ho documenti. Procureu fer `git status` o `git log` descriptius

DESFER CANVIS EN EL WORKSPACE (ABANS DE FER STAGE)

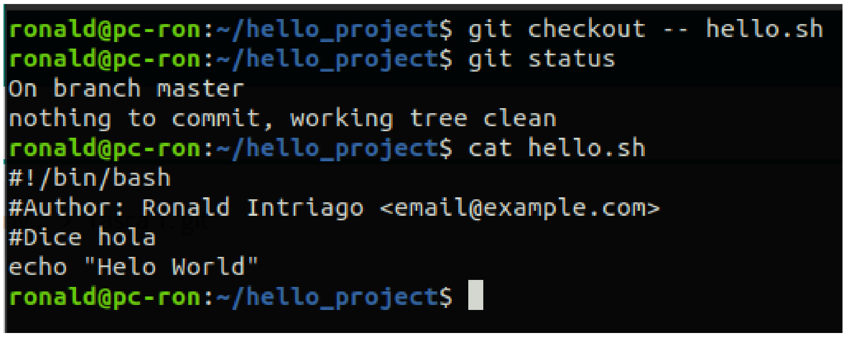
- Editem un fitxer

  Se ha cambioado **Helo** por **Hello**
- `git status`



```
ronald@pc-ron:~/hello_project$ vim hello.sh
ronald@pc-ron:~/hello_project$ git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   hello.sh

no changes added to commit (use "git add" and/or "git commit -a")
ronald@pc-ron:~/hello_project$ 
```

- Volem eliminar els canvis d'aquest fitxer del workspace (que encara no hem fet stage):
- `git checkout -- nomFitxer`
- Si mirem el fitxer, ja no tenim els canvis (i són irrecuperables)

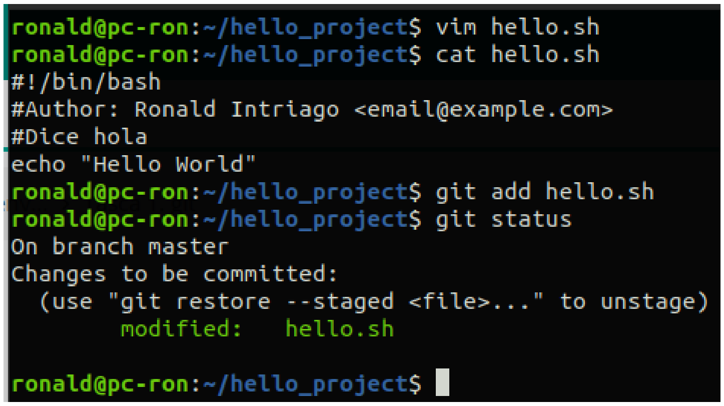


```
ronald@pc-ron:~/hello_project$ git checkout -- hello.sh
ronald@pc-ron:~/hello_project$ git status
On branch master
nothing to commit, working tree clean
ronald@pc-ron:~/hello_project$ cat hello.sh
#!/bin/bash
#Author: Ronald Intriago <email@example.com>
#Dice hola
echo "Helo World"
ronald@pc-ron:~/hello_project$ 
```

DESFER CANVIS EN EL STAGE (ABANS DE FER COMMIT)

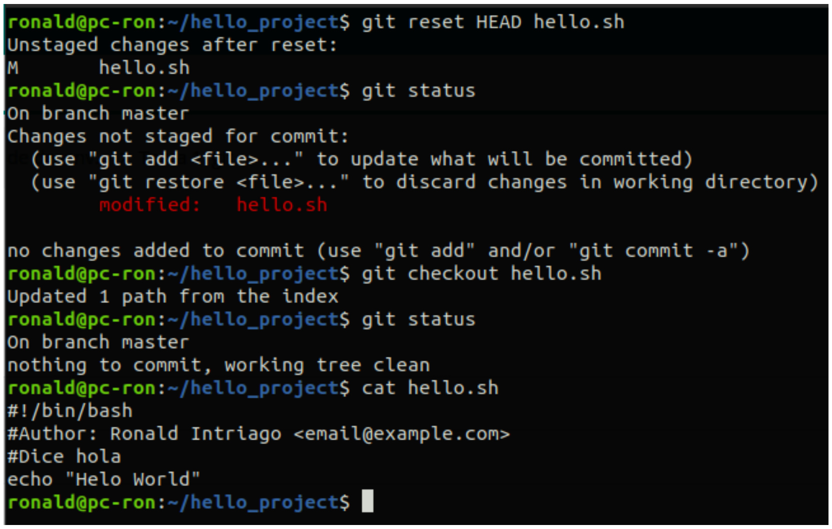
- Editem un fitxer

  Se ha cambioado **Helo** por **Hello**
- `git add nomFitxer`
- `git status`



```
ronald@pc-ron:~/hello_project$ vim hello.sh
ronald@pc-ron:~/hello_project$ cat hello.sh
#!/bin/bash
#Author: Ronald Intriago <email@example.com>
#Dice hola
echo "Hello World"
ronald@pc-ron:~/hello_project$ git add hello.sh
ronald@pc-ron:~/hello_project$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   hello.sh

ronald@pc-ron:~/hello_project$
```

- Traiem els canvis de stage:
- `git reset HEAD nomFitxer`
- `git status`
- Ara ja podem suprimir els canvis, com en l'apartat anterior:
- `git checkout nomFitxer`

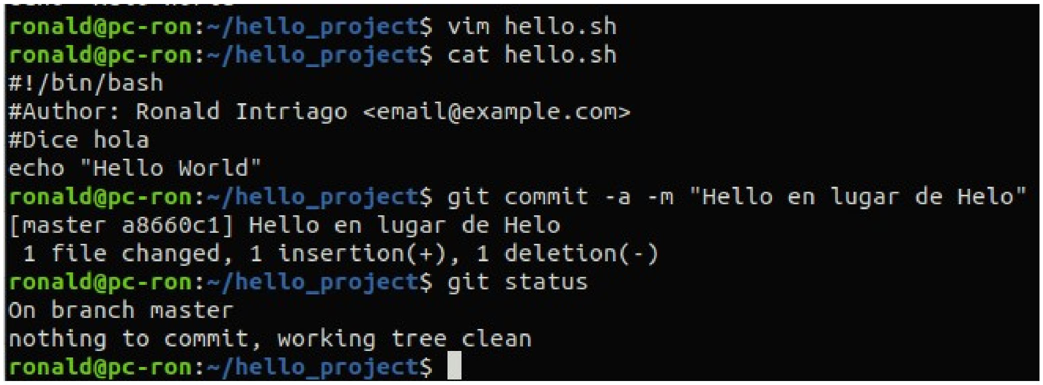


```
ronald@pc-ron:~/hello_project$ git reset HEAD hello.sh
Unstaged changes after reset:
M       hello.sh
ronald@pc-ron:~/hello_project$ git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   hello.sh

no changes added to commit (use "git add" and/or "git commit -a")
ronald@pc-ron:~/hello_project$ git checkout hello.sh
Updated 1 path from the index
ronald@pc-ron:~/hello_project$ git status
On branch master
nothing to commit, working tree clean
ronald@pc-ron:~/hello_project$ cat hello.sh
#!/bin/bash
#Author: Ronald Intriago <email@example.com>
#Dice hola
echo "Helo World"
ronald@pc-ron:~/hello_project$
```

DESFER UN COMMIT

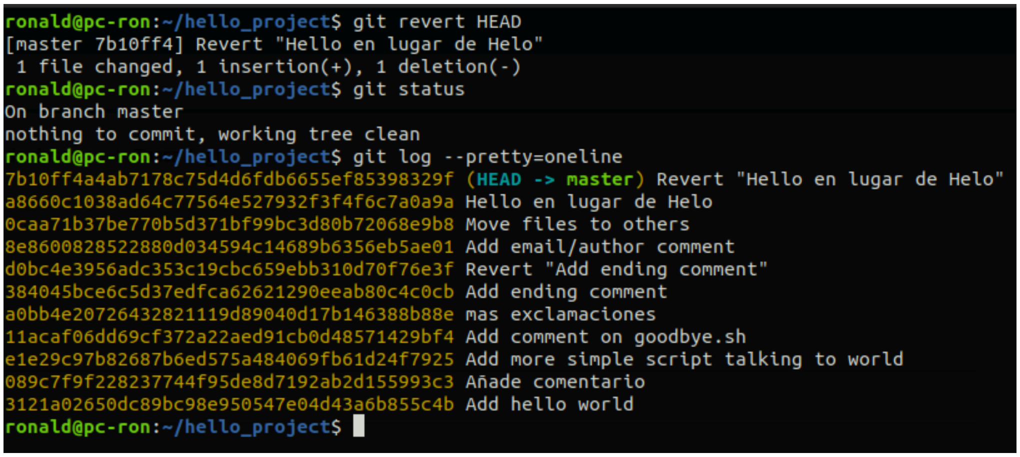
- Editem un fitxer

  Se ha cambioado **Helo** por **Hello**
- `git commit -a -m "Missatge del commit"`



```
ronald@pc-ron:~/hello_project$ vim hello.sh
ronald@pc-ron:~/hello_project$ cat hello.sh
#!/bin/bash
#Author: Ronald Intriago <email@example.com>
#Dice hola
echo "Hello World"
ronald@pc-ron:~/hello_project$ git commit -a -m "Hello en lugar de Helo"
[master a8660c1] Hello en lugar de Helo
 1 file changed, 1 insertion(+), 1 deletion(-)
ronald@pc-ron:~/hello_project$ git status
On branch master
nothing to commit, working tree clean
ronald@pc-ron:~/hello_project$ 
```

- Desfem el canvi
- `git revert HEAD` (se'ns obre l'editor i canviem, o no, el comentari)
- **El canvi, però queda a l'històric (és més segur)**



```
ronald@pc-ron:~/hello_project$ git revert HEAD
[master 7b10ff4] Revert "Hello en lugar de Helo"
 1 file changed, 1 insertion(+), 1 deletion(-)
ronald@pc-ron:~/hello_project$ git status
On branch master
nothing to commit, working tree clean
ronald@pc-ron:~/hello_project$ git log --pretty=oneline
7b10ff4a4ab7178c75d4d6fdb6655ef85398329f (HEAD -> master) Revert "Hello en lugar de Helo"
a8660c1038ad64c77564e527932f3f4f6c7a0a9a Hello en lugar de Helo
0caa71b37be770b5d371bf99bc3d80b72068e9b8 Move files to others
8e8600828522880d034594c14689b6356eb5ae01 Add email/author comment
d0bc4e3956adc353c19cbc659ebb310d70f76e3f Revert "Add ending comment"
384045bce6c5d37edfca62621290eeab80c4c0cb Add ending comment
a0bb4e20726432821119d89040d17b146388b88e mas exclamaciones
11acaf06dd69cf372a22aed91cb0d48571429bf4 Add comment on goodbye.sh
e1e29c97b82687b6ed575a484069fb61d24f7925 Add more simple script talking to world
089c7f9f228237744f95de8d7192ab2d155993c3 Añade comentario
3121a02650dc89bc98e950547e04d43a6b855c4b Add hello world
ronald@pc-ron:~/hello_project$ 
```

FER DIFF

i comentar què es compara en cada cas

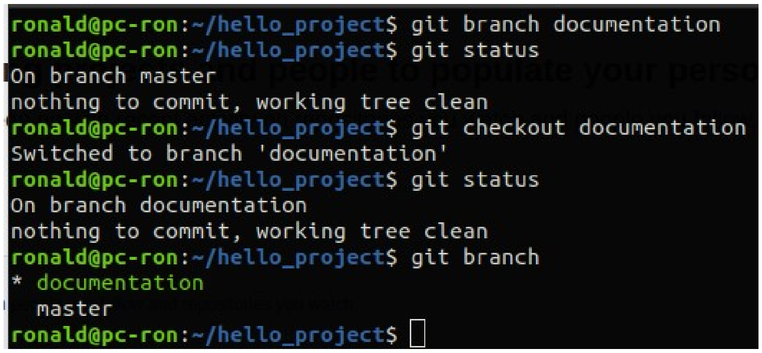
- git diff

  Muestra los cambios de los archivos que no se han hecho stag aún
- git diff HEAD

  Mustra todos los cambios de los archivos.

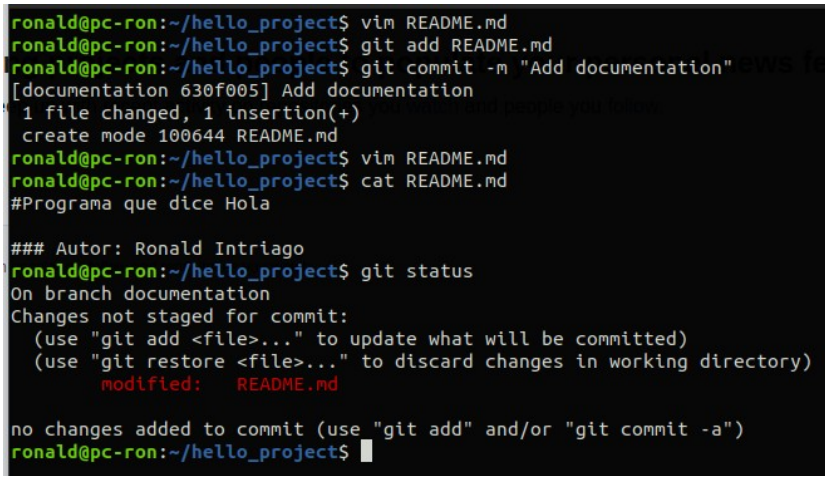
CREACIÓ D'UNA BRANCA

- `git branch documentation`
- `git checkout documentation`
- o
- `git checkout -b documentation`
- Per veure on estem
- `git status`
- o
- `git branch`



```
ronald@pc-ron:~/hello_project$ git branch documentation
ronald@pc-ron:~/hello_project$ git status
On branch master
nothing to commit, working tree clean
ronald@pc-ron:~/hello_project$ git checkout documentation
Switched to branch 'documentation'
ronald@pc-ron:~/hello_project$ git status
On branch documentation
nothing to commit, working tree clean
ronald@pc-ron:~/hello_project$ git branch
* documentation
  master
ronald@pc-ron:~/hello_project$ 
```

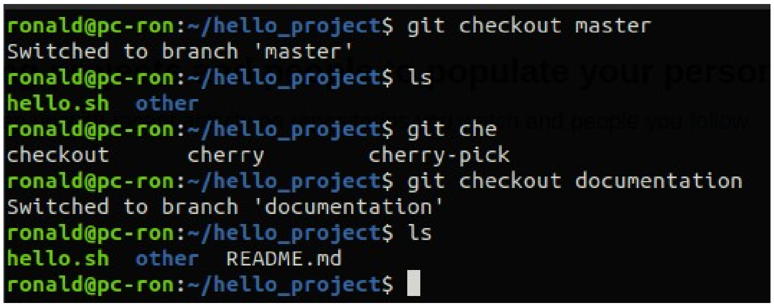
- Afegim coses a la branca. Posem un README:
  - A program that says hello.
- `git add README`
- `git commit -m "Add documentation"`
- Fem algun canvi més (afegir autor, llicència...)



```
ronald@pc-ron:~/hello_project$ vim README.md
ronald@pc-ron:~/hello_project$ git add README.md
ronald@pc-ron:~/hello_project$ git commit -m "Add documentation"
[documentation 630f005] Add documentation
 1 file changed, 1 insertion(+)
 create mode 100644 README.md
ronald@pc-ron:~/hello_project$ vim README.md
ronald@pc-ron:~/hello_project$ cat README.md
#Programa que dice Hola

### Autor: Ronald Intriago
ronald@pc-ron:~/hello_project$ git status
On branch documentation
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   README.md

no changes added to commit (use "git add" and/or "git commit -a")
ronald@pc-ron:~/hello_project$ 
```

MOURE'S PER LES BRANQUES

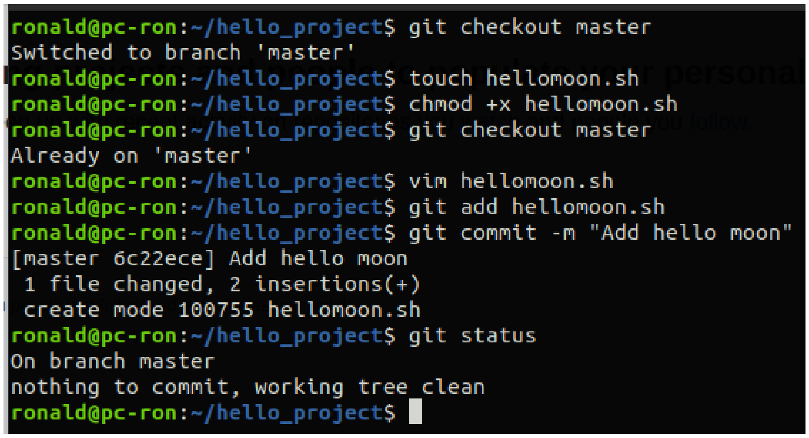
- Anem a la branca master
- `git checkout master`
- Mirem els fitxers que tenim amb ls.
- Tornem a documentation
- `git checkout documentation`



```
ronald@pc-ron:~/hello_project$ git checkout master
Switched to branch 'master'
ronald@pc-ron:~/hello_project$ ls
hello.sh  other
ronald@pc-ron:~/hello_project$ git che
checkout      cherry        cherry-pick
ronald@pc-ron:~/hello_project$ git checkout documentation
Switched to branch 'documentation'
ronald@pc-ron:~/hello_project$ ls
hello.sh  other  README.md
ronald@pc-ron:~/hello_project$ 
```

- CANVIS A master
- Anem a la branca master i afegim el fitxer hellomoon.sh
- `git checkout master`
- `#!/bin/sh`

- `echo "Hello moon!"`
- `git add hellomoon.sh`
- `git commit -m "Add hello moon"`

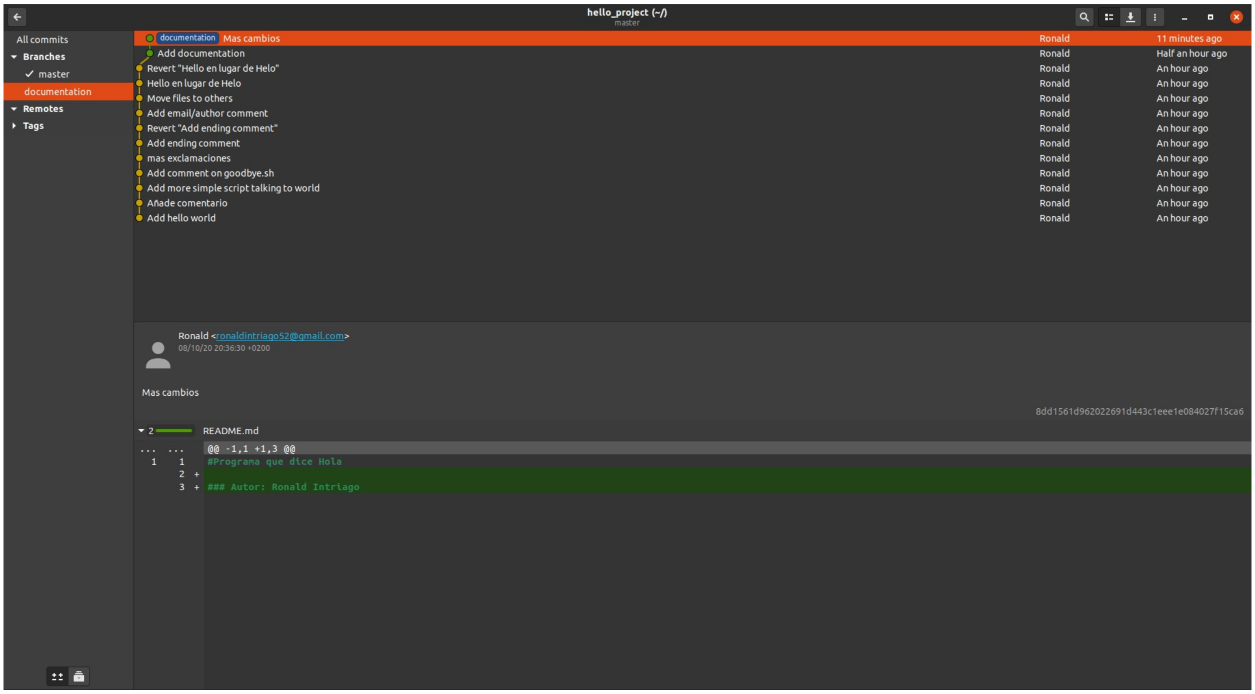


```
ronald@pc-ron:~/hello_project$ git checkout master
Switched to branch 'master'
ronald@pc-ron:~/hello_project$ touch hellomoon.sh
ronald@pc-ron:~/hello_project$ chmod +x hellomoon.sh
ronald@pc-ron:~/hello_project$ git checkout master
Already on 'master'
ronald@pc-ron:~/hello_project$ vim hellomoon.sh
ronald@pc-ron:~/hello_project$ git add hellomoon.sh
ronald@pc-ron:~/hello_project$ git commit -m "Add hello moon"
[master 6c22ece] Add hello moon
 1 file changed, 2 insertions(+)
 create mode 100755 hellomoon.sh
ronald@pc-ron:~/hello_project$ git status
On branch master
nothing to commit, working tree clean
ronald@pc-ron:~/hello_project$ 
```

- (Mireu el gitk o gitg!!!)
    - Gitg l’haureu d’instal.lar

      Instalado con el comando: **sudo apt-get install gitg**
    - O instal.leu alguna altra eina gràfica:
        - https://git-scm.com/book/en/v2/Appendix-A%3A-Git-in-Other-Environments-Graphical-Interfaces
        - O gitkraken
        - Penseu que després treballarem amb Eclipse git.

VEURE LES BRANQUES DES D’UNA EINA GRÀFICA



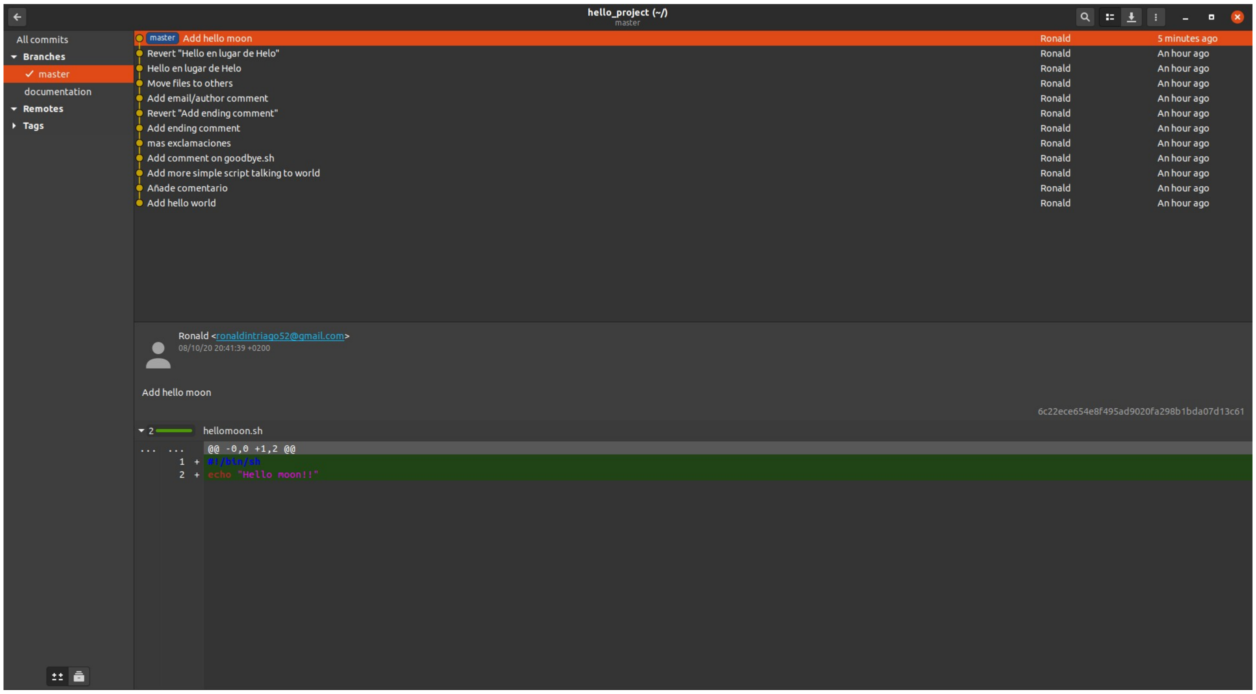
Mas cambios	Ronald	11 minutes ago
Add documentation	Ronald	Half an hour ago
Revert "Hello en lugar de Helo"	Ronald	An hour ago
Hello en lugar de Helo	Ronald	An hour ago
Move files to others	Ronald	An hour ago
Add email/author comment	Ronald	An hour ago
Revert "Add ending comment"	Ronald	An hour ago
Add ending comment	Ronald	An hour ago
mas exclamaciones	Ronald	An hour ago
Add comment on goodbye.sh	Ronald	An hour ago
Add more simple script talking to world	Ronald	An hour ago
Añade comentario	Ronald	An hour ago
Add hello world	Ronald	An hour ago

Ronald <ronaldintriago52@gmail.com>
08/10/20 20:36:30 +0200

Mas cambios

8dd1561d962022691d443c1eee1e084027f15ca6

```
2  README.md
...  ...  @@ -1,1 +1,3 @@
  1    1  #Programa que dice Hola
       2 +
       3 + ### Autor: Ronald Intriago
```



Add hello moon	Ronald	5 minutes ago
Revert "Hello en lugar de Helo"	Ronald	An hour ago
Hello en lugar de Helo	Ronald	An hour ago
Move files to others	Ronald	An hour ago
Add email/author comment	Ronald	An hour ago
Revert "Add ending comment"	Ronald	An hour ago
Add ending comment	Ronald	An hour ago
mas exclamaciones	Ronald	An hour ago
Add comment on goodbye.sh	Ronald	An hour ago
Add more simple script talking to world	Ronald	An hour ago
Añade comentario	Ronald	An hour ago
Add hello world	Ronald	An hour ago

Ronald <ronaldintriago52@gmail.com>
08/10/20 20:41:39 +0200

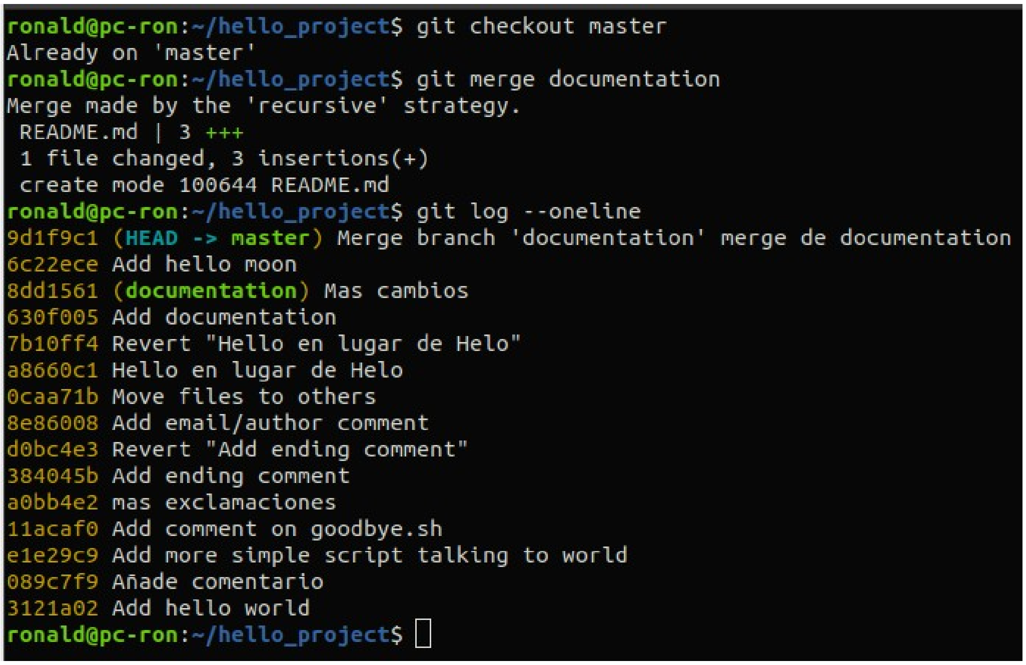
Add hello moon

6c22ece654e8f495ad9020fa298b1bda07d13c61

```
2  hellomoon.sh
...  ...  @@ -0,0 +1,2 @@
       1 + #!/bin/sh
       2 + echo "Hello moon!!"
```

FUISONAR BRANQUES (MERGE)

- Anirem a master i incorporarem els canvis de documentation
- `git checkout master`
- `git merge documentation`
- `git log --oneline`



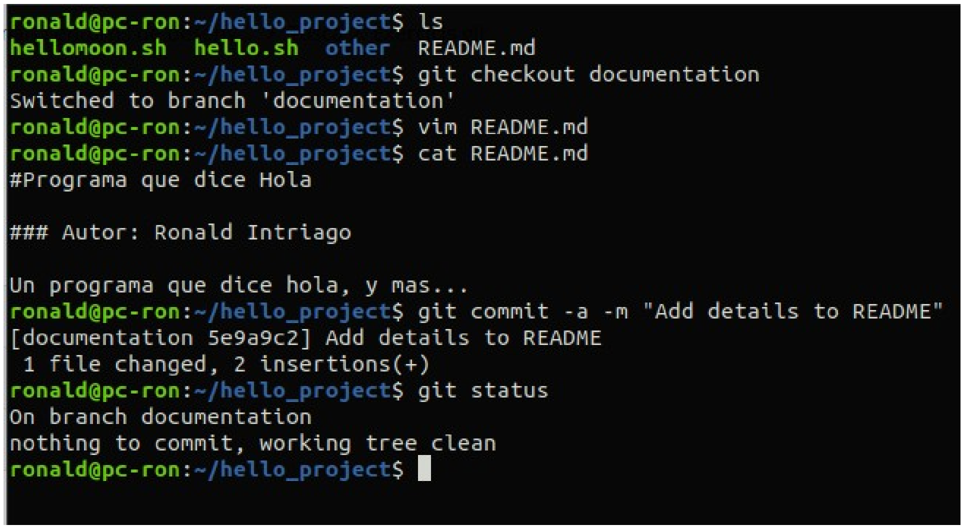
```
ronald@pc-ron:~/hello_project$ git checkout master
Already on 'master'
ronald@pc-ron:~/hello_project$ git merge documentation
Merge made by the 'recursive' strategy.
 README.md | 3 +++
 1 file changed, 3 insertions(+)
 create mode 100644 README.md
ronald@pc-ron:~/hello_project$ git log --oneline
9d1f9c1 (HEAD -> master) Merge branch 'documentation' merge de documentation
6c22ece Add hello moon
8dd1561 (documentation) Mas cambios
630f005 Add documentation
7b10ff4 Revert "Hello en lugar de Helo"
a8660c1 Hello en lugar de Helo
0caa71b Move files to others
8e86008 Add email/author comment
d0bc4e3 Revert "Add ending comment"
384045b Add ending comment
a0bb4e2 mas exclamaciones
11acaf0 Add comment on goodbye.sh
e1e29c9 Add more simple script talking to world
089c7f9 Añade comentario
3121a02 Add hello world
ronald@pc-ron:~/hello_project$ 
```

ESBORRAR BRANQUES

- `git branch -d nombranca`   (No cal fer-ho)

CREANT CONFLICTES

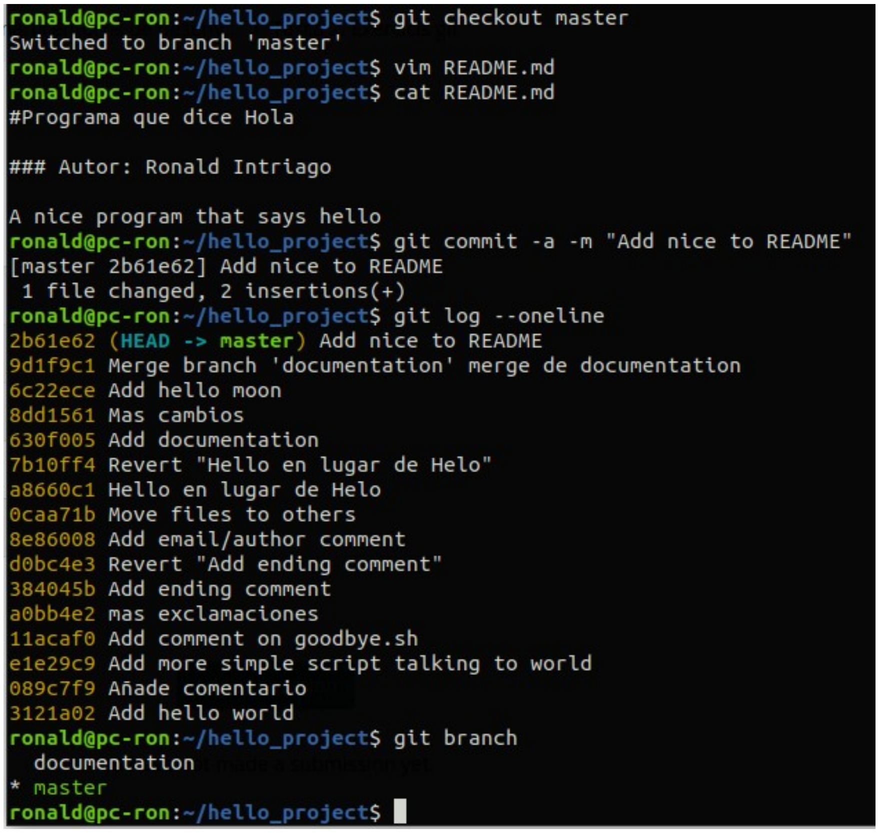
- Anem a documentation:
- `git checkout documentation`
- editem el README:
  - A program that says hello and more.
- `git commit -a -m "Add details to README"`



```
ronald@pc-ron:~/hello_project$ ls
hellomoon.sh  hello.sh  other  README.md
ronald@pc-ron:~/hello_project$ git checkout documentation
Switched to branch 'documentation'
ronald@pc-ron:~/hello_project$ vim README.md
ronald@pc-ron:~/hello_project$ cat README.md
#Programa que dice Hola

### Autor: Ronald Intriago

Un programa que dice hola, y mas...
ronald@pc-ron:~/hello_project$ git commit -a -m "Add details to README"
[documentation 5e9a9c2] Add details to README
 1 file changed, 2 insertions(+)
ronald@pc-ron:~/hello_project$ git status
On branch documentation
nothing to commit, working tree clean
ronald@pc-ron:~/hello_project$ 
```

- Anem a master:
- `git checkout master`
- editem el README:
  - A nice program that says hello.
- `git commit -a -m "Add nice to README"`

- `git log…`
- `git branch`

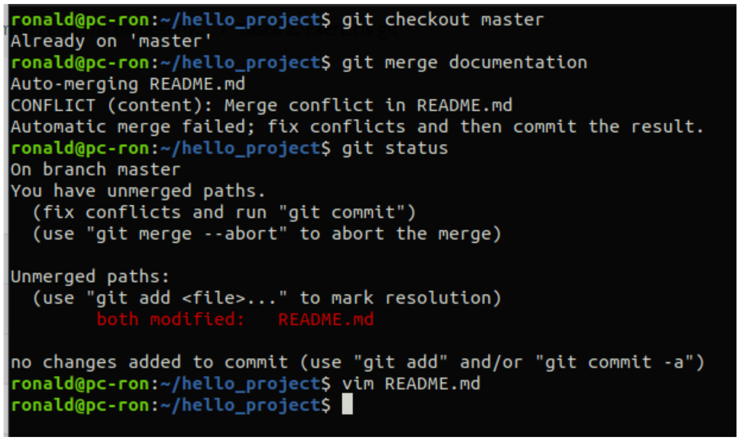


```
ronald@pc-ron:~/hello_project$ git checkout master
Switched to branch 'master'
ronald@pc-ron:~/hello_project$ vim README.md
ronald@pc-ron:~/hello_project$ cat README.md
#Programa que dice Hola

### Autor: Ronald Intriago

A nice program that says hello
ronald@pc-ron:~/hello_project$ git commit -a -m "Add nice to README"
[master 2b61e62] Add nice to README
 1 file changed, 2 insertions(+)
ronald@pc-ron:~/hello_project$ git log --oneline
2b61e62 (HEAD -> master) Add nice to README
9d1f9c1 Merge branch 'documentation' merge de documentation
6c22ece Add hello moon
8dd1561 Mas cambios
630f005 Add documentation
7b10ff4 Revert "Hello en lugar de Helo"
a8660c1 Hello en lugar de Helo
0caa71b Move files to others
8e86008 Add email/author comment
d0bc4e3 Revert "Add ending comment"
384045b Add ending comment
a0bb4e2 mas exclamaciones
11acaf0 Add comment on goodbye.sh
e1e29c9 Add more simple script talking to world
089c7f9 Añade comentario
3121a02 Add hello world
ronald@pc-ron:~/hello_project$ git branch
  documentation
* master
ronald@pc-ron:~/hello_project$ 
```

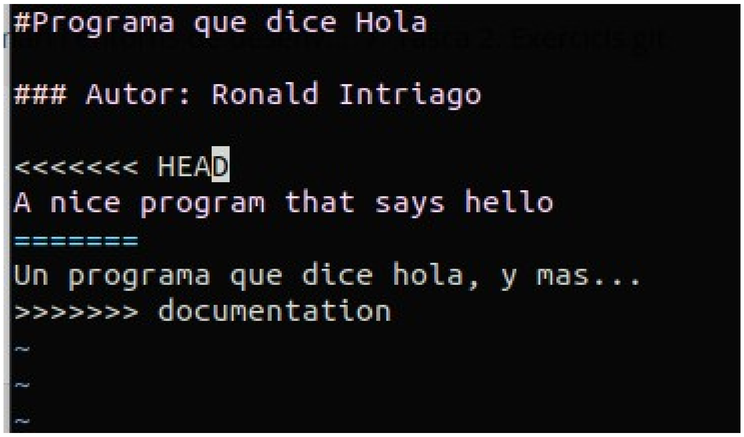
RESOLUCIÓ DE CONFLICTES

- Ara intentem fer merge: ens situem a master i fem merge
- `git checkout master`
- `git merge documentation`
- Ens dóna un missatge dient que no ha pogut resoldre el conflicte...
- `git status`
- Veiem quin és el fitxer en conflicte i l'editem. El fitxer estarà marcat
    - `<<<<<<< HEAD`
    - `què hi ha a HEAD (master)`
    - `=======`
    - `què hi ha a documentation`
    - `>>>>>>> documentation`
    - Ho arreglem posant la frase
    - `A nice program that says hello and more.`

  - i treient les línees <<< >>> i ===
- git commit -a



```
ronald@pc-ron:~/hello_project$ git checkout master
Already on 'master'
ronald@pc-ron:~/hello_project$ git merge documentation
Auto-merging README.md
CONFLICT (content): Merge conflict in README.md
Automatic merge failed; fix conflicts and then commit the result.
ronald@pc-ron:~/hello_project$ git status
On branch master
You have unmerged paths.
  (fix conflicts and run "git commit")
  (use "git merge --abort" to abort the merge)

Unmerged paths:
  (use "git add <file>..." to mark resolution)
        both modified:   README.md

no changes added to commit (use "git add" and/or "git commit -a")
ronald@pc-ron:~/hello_project$ vim README.md
ronald@pc-ron:~/hello_project$ 
```

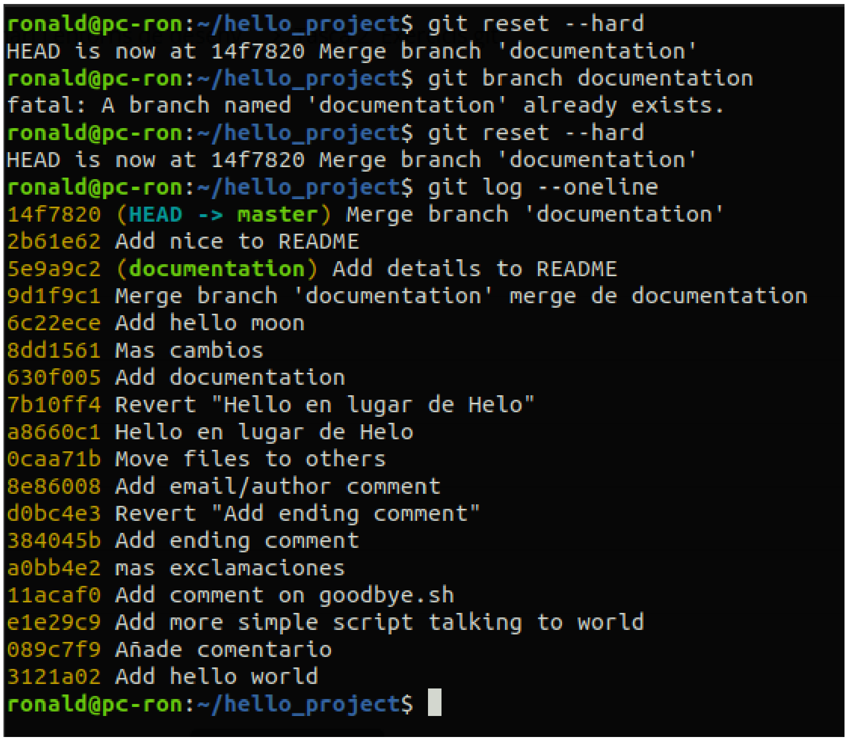


```
#Programa que dice Hola

### Autor: Ronald Intriago

<<<<<<< HEAD
A nice program that says hello
=======
Un programa que dice hola, y mas...
>>>>>>> documentation
~
~
~
```

https://www.atlassian.com/git/tutorials/using-branches/merge-conflicts

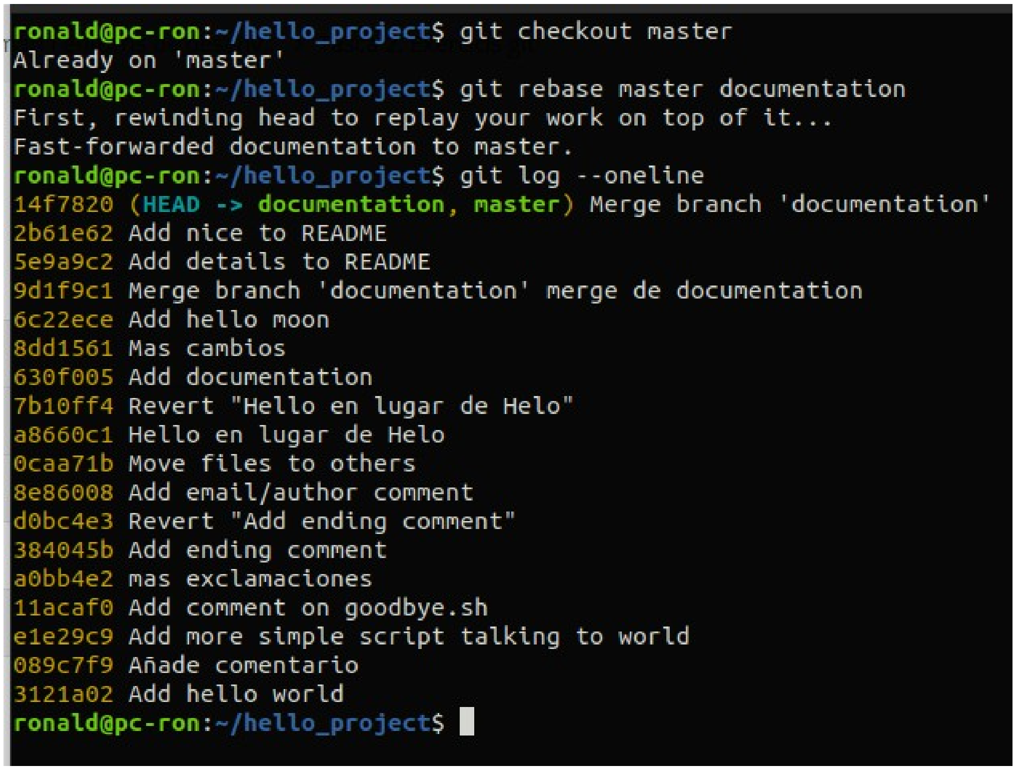
REBASE

- Farem un `reset --hard` fins al punt abans de fer el primer merge amb master. Hem de fer el `reset --hard` a les dues branques. Master ha d'estar en el canvi "Add hello moon" i documentation ha d'estar al commit "Add license"

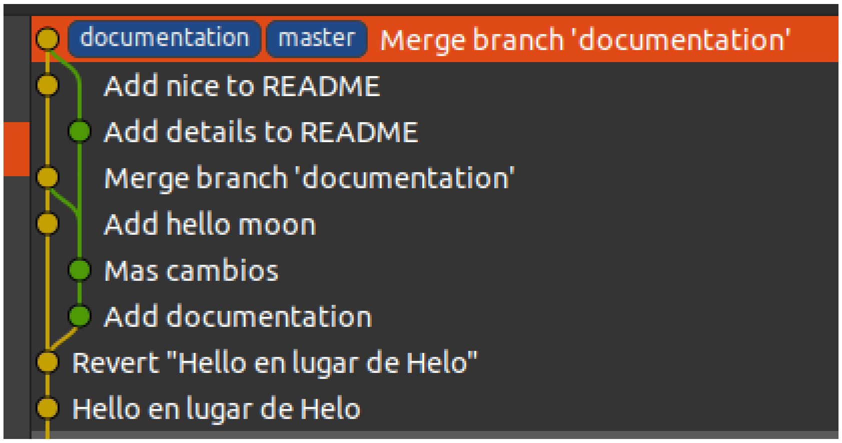


```
ronald@pc-ron:~/hello_project$ git reset --hard
HEAD is now at 14f7820 Merge branch 'documentation'
ronald@pc-ron:~/hello_project$ git branch documentation
fatal: A branch named 'documentation' already exists.
ronald@pc-ron:~/hello_project$ git reset --hard
HEAD is now at 14f7820 Merge branch 'documentation'
ronald@pc-ron:~/hello_project$ git log --oneline
14f7820 (HEAD -> master) Merge branch 'documentation'
2b61e62 Add nice to README
5e9a9c2 (documentation) Add details to README
9d1f9c1 Merge branch 'documentation' merge de documentation
6c22ece Add hello moon
8dd1561 Mas cambios
630f005 Add documentation
7b10ff4 Revert "Hello en lugar de Helo"
a8660c1 Hello en lugar de Helo
0caa71b Move files to others
8e86008 Add email/author comment
d0bc4e3 Revert "Add ending comment"
384045b Add ending comment
a0bb4e2 mas exclamaciones
11acaf0 Add comment on goodbye.sh
e1e29c9 Add more simple script talking to world
089c7f9 Añade comentario
3121a02 Add hello world
ronald@pc-ron:~/hello_project$ 
```

- Ara enlloc de fer merge farem rebase:
- `git checkout master`
- `git rebase master documentation`
- Veieu la diferència? Queda més net, però no s'ha de fer quan ja ho hem pujat a un altre repositori. **Només s'ha de fer per canvis locals encara no pujats**.



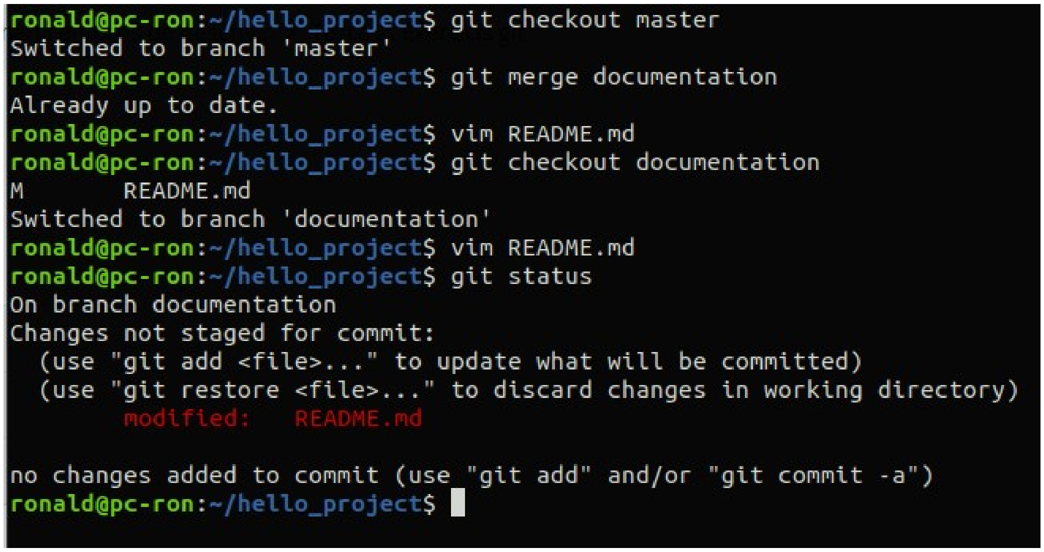
```
ronald@pc-ron:~/hello_project$ git checkout master
Already on 'master'
ronald@pc-ron:~/hello_project$ git rebase master documentation
First, rewinding head to replay your work on top of it...
Fast-forwarded documentation to master.
ronald@pc-ron:~/hello_project$ git log --oneline
14f7820 (HEAD -> documentation, master) Merge branch 'documentation'
2b61e62 Add nice to README
5e9a9c2 Add details to README
9d1f9c1 Merge branch 'documentation' merge de documentation
6c22ece Add hello moon
8dd1561 Mas cambios
630f005 Add documentation
7b10ff4 Revert "Hello en lugar de Helo"
a8660c1 Hello en lugar de Helo
0caa71b Move files to others
8e86008 Add email/author comment
d0bc4e3 Revert "Add ending comment"
384045b Add ending comment
a0bb4e2 mas exclamaciones
11acaf0 Add comment on goodbye.sh
e1e29c9 Add more simple script talking to world
089c7f9 Añade comentario
3121a02 Add hello world
ronald@pc-ron:~/hello_project$ 
```



```
Merge branch 'documentation'
Add nice to README
Add details to README
Merge branch 'documentation'
Add hello moon
Mas cambios
Add documentation
Revert "Hello en lugar de Helo"
Hello en lugar de Helo
```

o

- Ara podem fer un merge de documentation a master, veureu, que l'únic que fa és avançar master fins a documentation (merge fast forward)
- `git checkout master`
- `git merge documentation`
- Fer alguns canvis més a master i documentation per tornar a separar les branques.

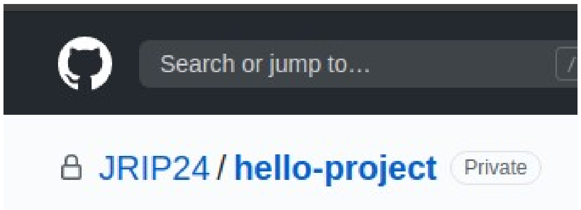


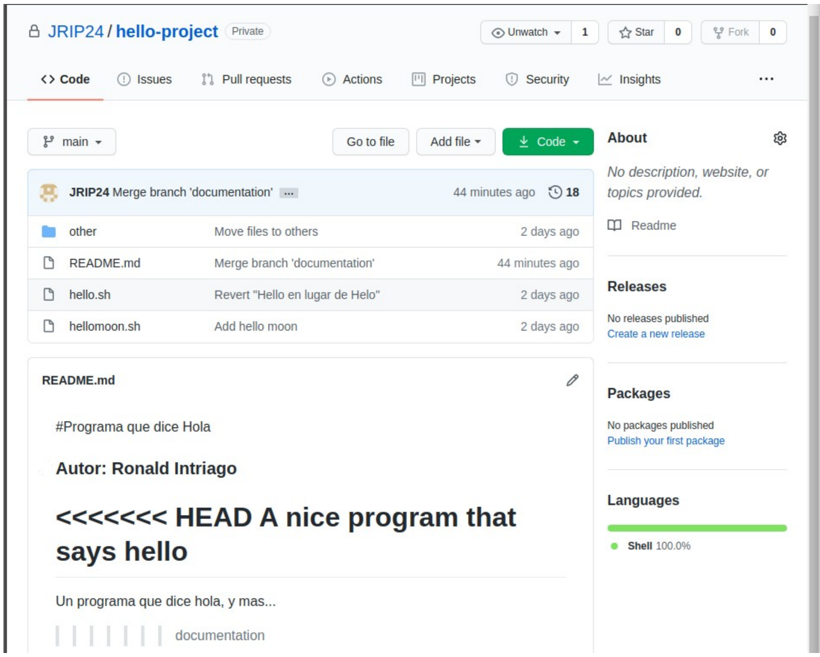
```
ronald@pc-ron:~/hello_project$ git checkout master
Switched to branch 'master'
ronald@pc-ron:~/hello_project$ git merge documentation
Already up to date.
ronald@pc-ron:~/hello_project$ vim README.md
ronald@pc-ron:~/hello_project$ git checkout documentation
M       README.md
Switched to branch 'documentation'
ronald@pc-ron:~/hello_project$ vim README.md
ronald@pc-ron:~/hello_project$ git status
On branch documentation
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   README.md

no changes added to commit (use "git add" and/or "git commit -a")
ronald@pc-ron:~/hello_project$
```

PUJAR UN REPOSITORI LOCAL A UN REPOSITORI REMOT

Creeu un projecte al repositori remot (GitLab) i pugeu el projecte

Assegureu-vos que heu pujat el vostre repositori local amb totes les branques usant 'get push'.





JRIP24 Merge branch 'documentation' ... 44 minutes ago 18

other	Move files to others	2 days ago
README.md	Merge branch 'documentation'	44 minutes ago
hello.sh	Revert "Hello en lugar de Helo"	2 days ago
hellomoon.sh	Add hello moon	2 days ago

README.md

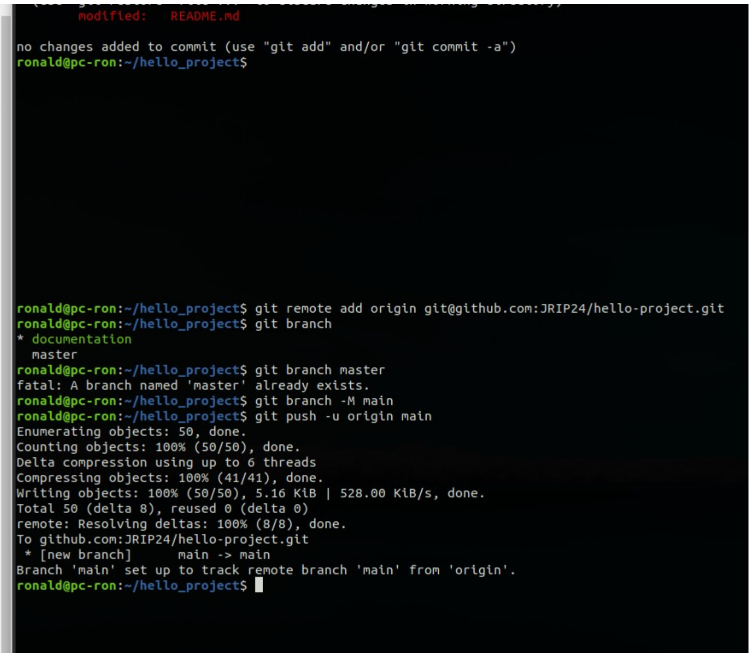
#Programa que dice Hola

Autor: Ronald Intriago

<<<<<<< HEAD A nice program that says hello

Un programa que dice hola, y mas...

| | | | | | | documentation

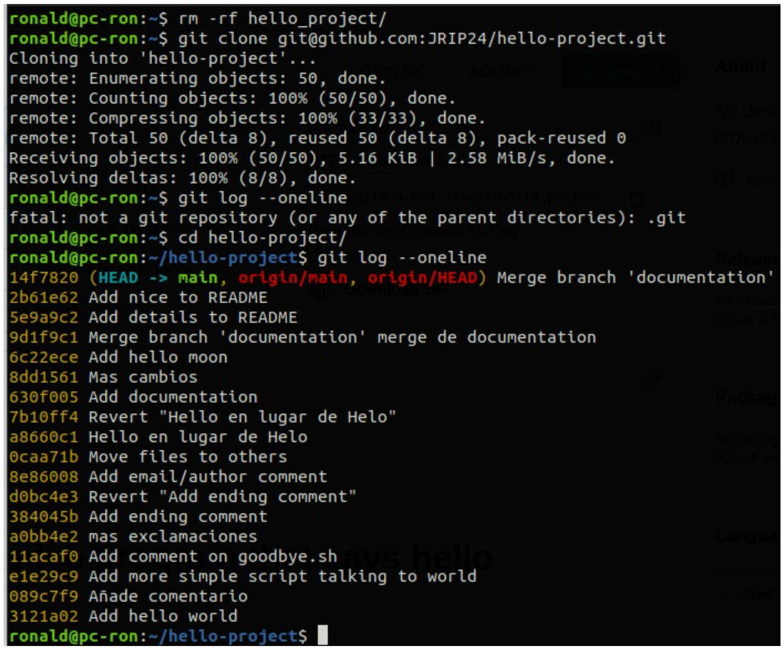


```
        modified:   README.md

no changes added to commit (use "git add" and/or "git commit -a")
ronald@pc-ron:~/hello_project$




ronald@pc-ron:~/hello_project$ git remote add origin git@github.com:JRIP24/hello-project.git
ronald@pc-ron:~/hello_project$ git branch
* documentation
  master
ronald@pc-ron:~/hello_project$ git branch master
fatal: A branch named 'master' already exists.
ronald@pc-ron:~/hello_project$ git branch -M main
ronald@pc-ron:~/hello_project$ git push -u origin main
Enumerating objects: 50, done.
Counting objects: 100% (50/50), done.
Delta compression using up to 6 threads
Compressing objects: 100% (41/41), done.
Writing objects: 100% (50/50), 5.16 KiB | 528.00 KiB/s, done.
Total 50 (delta 8), reused 0 (delta 0)
remote: Resolving deltas: 100% (8/8), done.
To github.com:JRIP24/hello-project.git
 * [new branch]      main -> main
Branch 'main' set up to track remote branch 'main' from 'origin'.
ronald@pc-ron:~/hello_project$ 
```

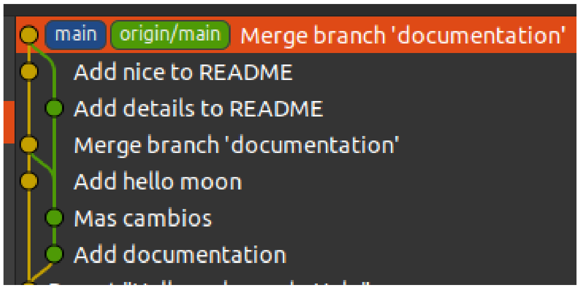
CLONAR UN REPOSITORI REMOT

Esborrarem el directori local i clonarem el directori remot. O el clonarem en un altre directori.

- Si mireu a git (o gitg, o gitk), veureu més etiquetes (origin/HEAD, origin/master i origin/documentation)



```
ronald@pc-ron:~$ rm -rf hello_project/
ronald@pc-ron:~$ git clone git@github.com:JRIP24/hello-project.git
Cloning into 'hello-project'...
remote: Enumerating objects: 50, done.
remote: Counting objects: 100% (50/50), done.
remote: Compressing objects: 100% (33/33), done.
remote: Total 50 (delta 8), reused 50 (delta 8), pack-reused 0
Receiving objects: 100% (50/50), 5.16 KiB | 2.58 MiB/s, done.
Resolving deltas: 100% (8/8), done.
ronald@pc-ron:~$ git log --oneline
fatal: not a git repository (or any of the parent directories): .git
ronald@pc-ron:~$ cd hello-project/
ronald@pc-ron:~/hello-project$ git log --oneline
14f7820 (HEAD -> main, origin/main, origin/HEAD) Merge branch 'documentation'
2b61e62 Add nice to README
5e9a9c2 Add details to README
9d1f9c1 Merge branch 'documentation' merge de documentation
6c22ece Add hello moon
8dd1561 Mas cambios
630f005 Add documentation
7b10ff4 Revert "Hello en lugar de Helo"
a8660c1 Hello en lugar de Helo
0caa71b Move files to others
8e86008 Add email/author comment
d0bc4e3 Revert "Add ending comment"
384045b Add ending comment
a0bb4e2 mas exclamaciones
11acaf0 Add comment on goodbye.sh
e1e29c9 Add more simple script talking to world
089c7f9 Añade comentario
3121a02 Add hello world
ronald@pc-ron:~/hello-project$ 
```



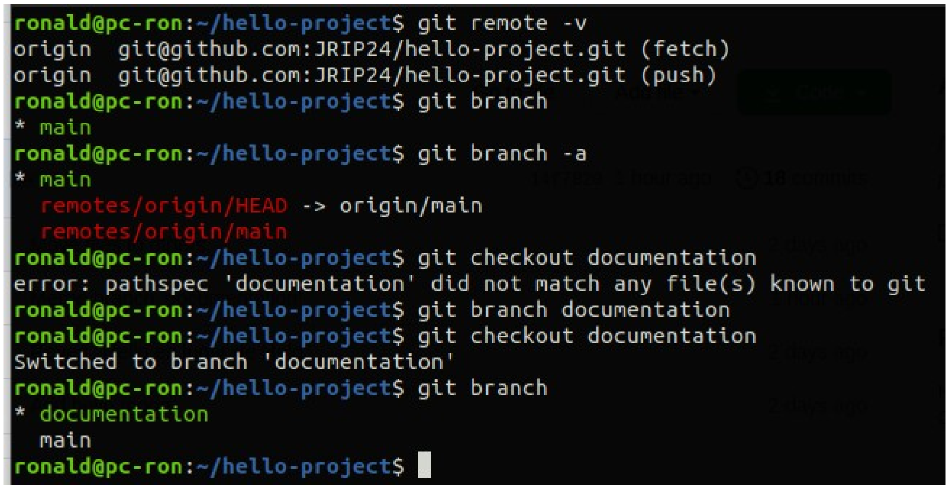
```
Merge branch 'documentation'
Add nice to README
Add details to README
Merge branch 'documentation'
Add hello moon
Mas cambios
Add documentation
```

- Això es deu a que tenim anotat el repositori remot del qual hem clonat. Podem veure-ho fent:
- `git remote -v`
- Si mirem les branques veiem que només tenim la master:
- `git branch`
- Llistem també les branques remotes:
- `git branch -a`

  Al subir los archivos cree una **branch** con nombre, **main**, por lo que no tengo **master**, sino **main**.

- Fixeu-vos que de fet només tenim en local la branca master (no la de documentation).
- Però quan fem `git checkout documentation`, ja la tindrem disponible.

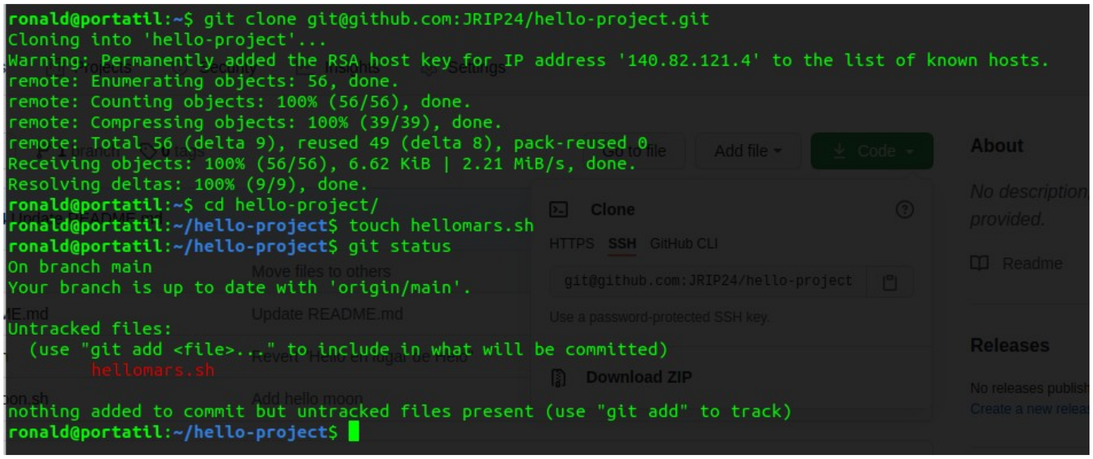
  Al subir el proyecto a GitHub se eliminaron las ramas. Por lo que la creé.



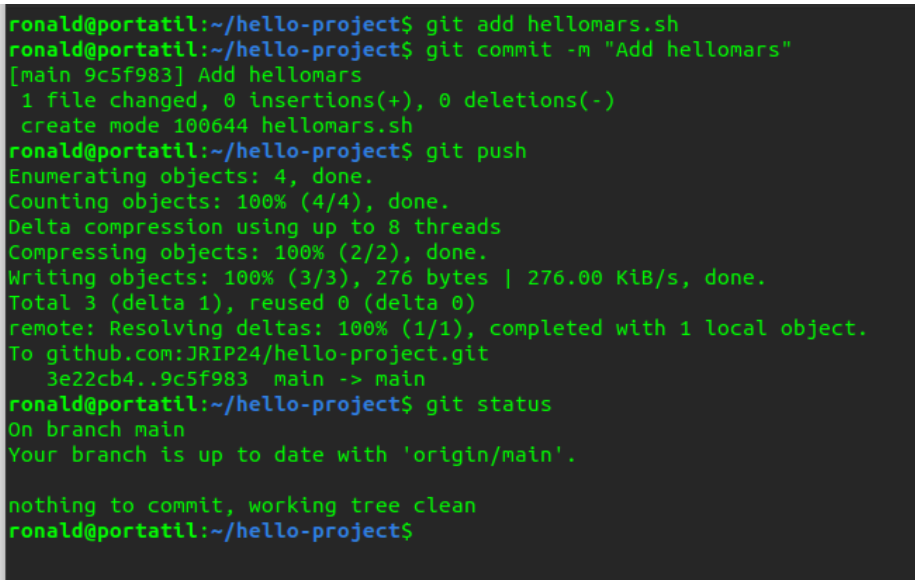
```
ronald@pc-ron:~/hello-project$ git remote -v
origin  git@github.com:JRIP24/hello-project.git (fetch)
origin  git@github.com:JRIP24/hello-project.git (push)
ronald@pc-ron:~/hello-project$ git branch
* main
ronald@pc-ron:~/hello-project$ git branch -a
* main
  remotes/origin/HEAD -> origin/main
  remotes/origin/main
ronald@pc-ron:~/hello-project$ git checkout documentation
error: pathspec 'documentation' did not match any file(s) known to git
ronald@pc-ron:~/hello-project$ git branch documentation
ronald@pc-ron:~/hello-project$ git checkout documentation
Switched to branch 'documentation'
ronald@pc-ron:~/hello-project$ git branch
* documentation
  main
ronald@pc-ron:~/hello-project$ 
```

RECUPERAR CANVIS FETS EN EL REPOSITORI REMOT

- Ara li donarem permisos a un company de classe per tal que pugui usar el repositori remot i li demanarem que faci el següent canvi en el repositori (git clone, fer el canvi, git commit, git push).
- El canvi serà el següent:
  - Afegim el fitxer hellomars.sh a la branca master

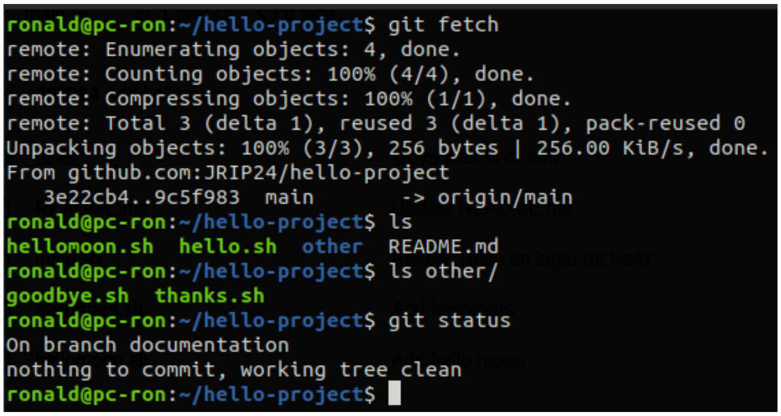


```
ronald@portatil:~$ git clone git@github.com:JRIP24/hello-project.git
Cloning into 'hello-project'...
Warning: Permanently added the RSA host key for IP address '140.82.121.4' to the list of known hosts.
remote: Enumerating objects: 56, done.
remote: Counting objects: 100% (56/56), done.
remote: Compressing objects: 100% (39/39), done.
remote: Total 56 (delta 9), reused 49 (delta 8), pack-reused 0
Receiving objects: 100% (56/56), 6.62 KiB | 2.21 MiB/s, done.
Resolving deltas: 100% (9/9), done.
ronald@portatil:~$ cd hello-project/
ronald@portatil:~/hello-project$ touch hellomars.sh
ronald@portatil:~/hello-project$ git status
On branch main
Your branch is up to date with 'origin/main'.

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        hellomars.sh

nothing added to commit but untracked files present (use "git add" to track)
ronald@portatil:~/hello-project$ 
```

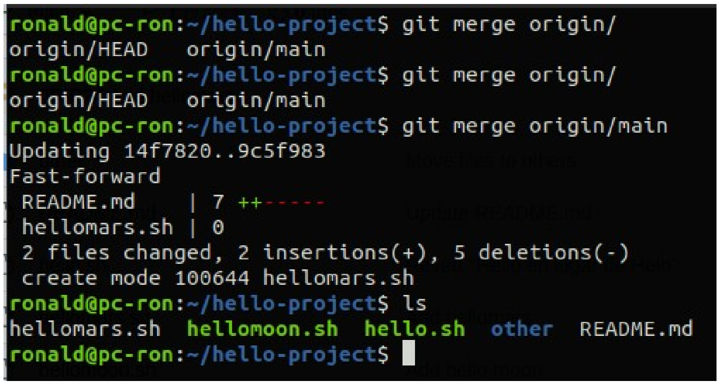


```
ronald@portatil:~/hello-project$ git add hellomars.sh
ronald@portatil:~/hello-project$ git commit -m "Add hellomars"
[main 9c5f983] Add hellomars
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 hellomars.sh
ronald@portatil:~/hello-project$ git push
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 276 bytes | 276.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To github.com:JRIP24/hello-project.git
   3e22cb4..9c5f983  main -> main
ronald@portatil:~/hello-project$ git status
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean
ronald@portatil:~/hello-project$
```

- Recuperem el canvi del repositori remot que ens ha pujat el company amb la següent comanda:
- `git fetch`
- Hem recuperat la informació de la branca remota, però no l'hem afegit a la nostra branca local.
- Si mireu el directori de treball no hi ha el fitxer others/hellomars.sh
- Si volem incorporar els canvis a la branca local hem de fer:
- `git merge origin/master`



```
ronald@pc-ron:~/hello-project$ git fetch
remote: Enumerating objects: 4, done.
remote: Counting objects: 100% (4/4), done.
remote: Compressing objects: 100% (1/1), done.
remote: Total 3 (delta 1), reused 3 (delta 1), pack-reused 0
Unpacking objects: 100% (3/3), 256 bytes | 256.00 KiB/s, done.
From github.com:JRIP24/hello-project
   3e22cb4..9c5f983  main       -> origin/main
ronald@pc-ron:~/hello-project$ ls
hellomoon.sh  hello.sh  other  README.md
ronald@pc-ron:~/hello-project$ ls other/
goodbye.sh  thanks.sh
ronald@pc-ron:~/hello-project$ git status
On branch documentation
nothing to commit, working tree clean
ronald@pc-ron:~/hello-project$ 
```



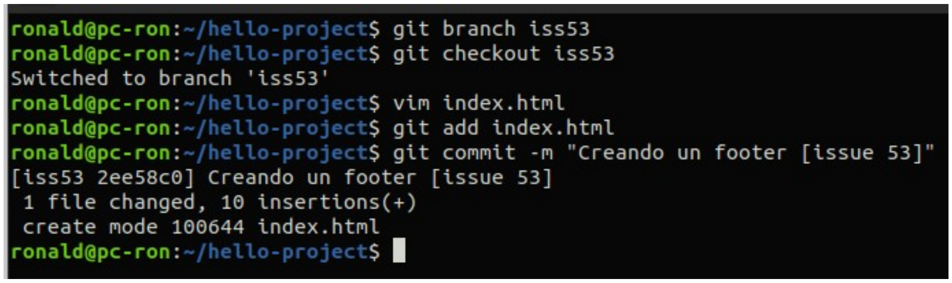
```
ronald@pc-ron:~/hello-project$ git merge origin/
origin/HEAD   origin/main
ronald@pc-ron:~/hello-project$ git merge origin/
origin/HEAD   origin/main
ronald@pc-ron:~/hello-project$ git merge origin/main
Updating 14f7820..9c5f983
Fast-forward
 README.md    | 7 ++-----
 hellomars.sh | 0
 2 files changed, 2 insertions(+), 5 deletions(-)
 create mode 100644 hellomars.sh
ronald@pc-ron:~/hello-project$ ls
hellomars.sh  hellomoon.sh  hello.sh  other  README.md
ronald@pc-ron:~/hello-project$ 
```

REPRODUÏU EL PROBLEMA PRESENTAT A CLASSE

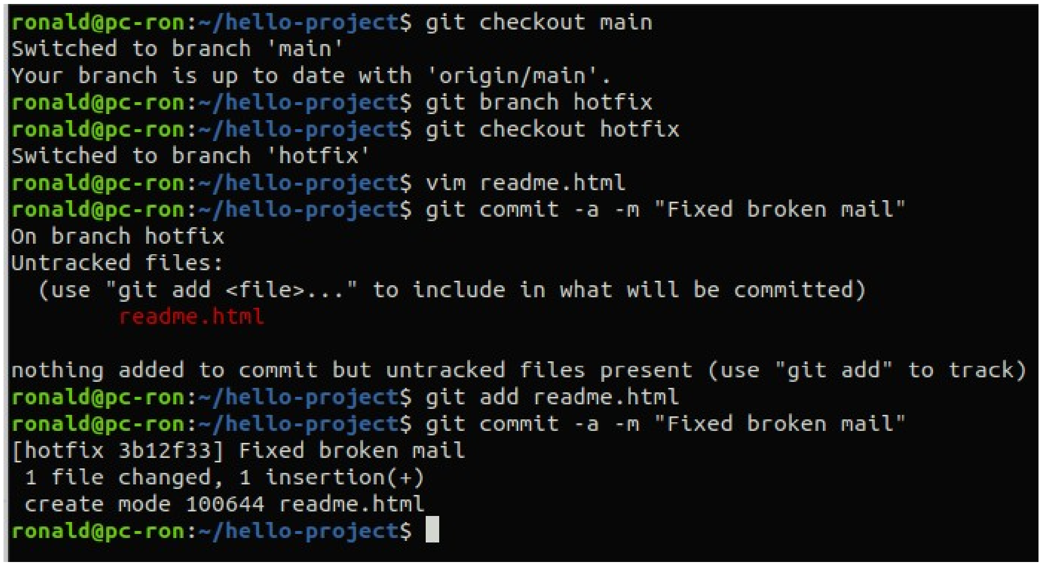
(GIT Avanzado.pdf. Problema branques master/isss53 hotfix)

Primer modificant fitxers diferents a les branques, de manera que no hi hagi conflicte en fer el merge, i després modificant el mateix fitxer en dues branques per generar un conflicte i solucionar-lo

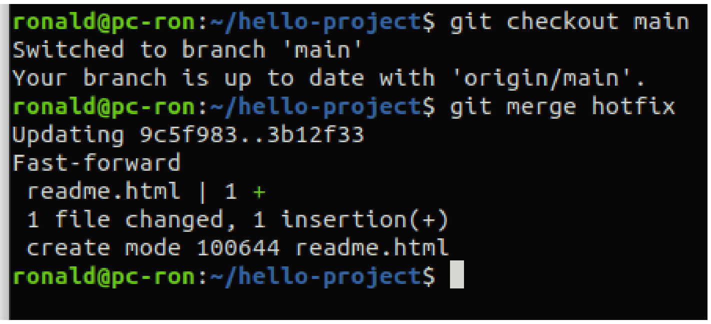
Merge sin conflicto



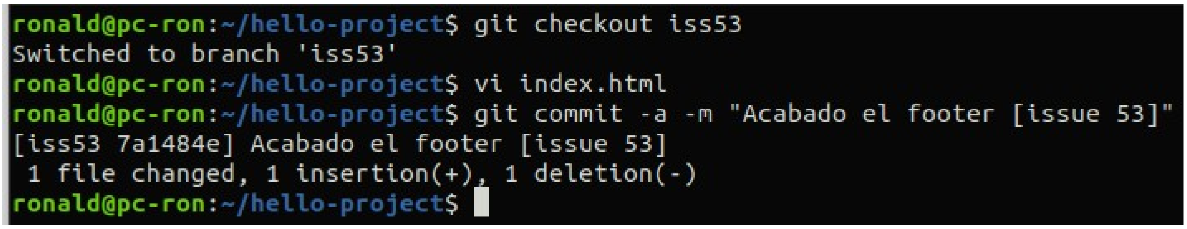
```
ronald@pc-ron:~/hello-project$ git branch iss53
ronald@pc-ron:~/hello-project$ git checkout iss53
Switched to branch 'iss53'
ronald@pc-ron:~/hello-project$ vim index.html
ronald@pc-ron:~/hello-project$ git add index.html
ronald@pc-ron:~/hello-project$ git commit -m "Creando un footer [issue 53]"
[iss53 2ee58c0] Creando un footer [issue 53]
 1 file changed, 10 insertions(+)
 create mode 100644 index.html
ronald@pc-ron:~/hello-project$ 
```



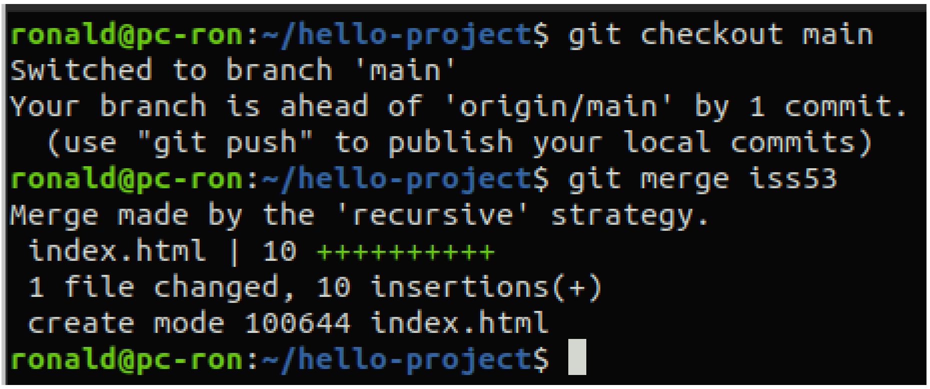
```
ronald@pc-ron:~/hello-project$ git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.
ronald@pc-ron:~/hello-project$ git branch hotfix
ronald@pc-ron:~/hello-project$ git checkout hotfix
Switched to branch 'hotfix'
ronald@pc-ron:~/hello-project$ vim readme.html
ronald@pc-ron:~/hello-project$ git commit -a -m "Fixed broken mail"
On branch hotfix
Untracked files:
  (use "git add <file>..." to include in what will be committed)
        readme.html

nothing added to commit but untracked files present (use "git add" to track)
ronald@pc-ron:~/hello-project$ git add readme.html
ronald@pc-ron:~/hello-project$ git commit -a -m "Fixed broken mail"
[hotfix 3b12f33] Fixed broken mail
 1 file changed, 1 insertion(+)
 create mode 100644 readme.html
ronald@pc-ron:~/hello-project$ 
```



```
ronald@pc-ron:~/hello-project$ git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.
ronald@pc-ron:~/hello-project$ git merge hotfix
Updating 9c5f983..3b12f33
Fast-forward
 readme.html | 1 +
 1 file changed, 1 insertion(+)
 create mode 100644 readme.html
ronald@pc-ron:~/hello-project$ 
```

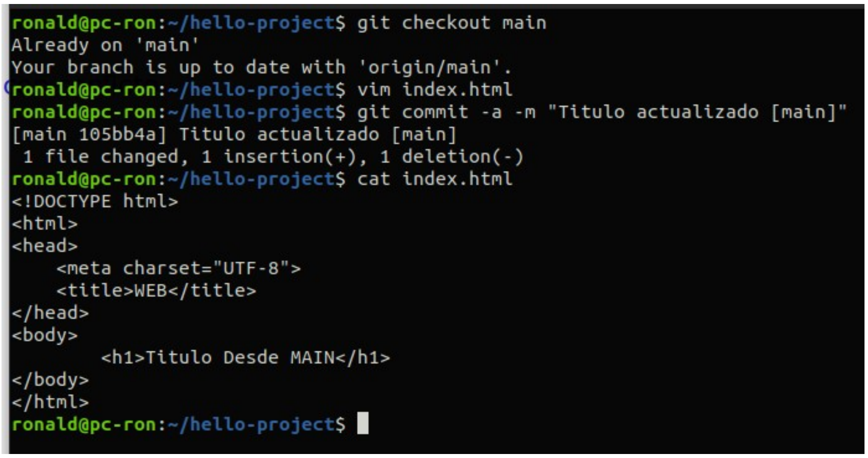


```
ronald@pc-ron:~/hello-project$ git checkout iss53
Switched to branch 'iss53'
ronald@pc-ron:~/hello-project$ vi index.html
ronald@pc-ron:~/hello-project$ git commit -a -m "Acabado el footer [issue 53]"
[iss53 7a1484e] Acabado el footer [issue 53]
 1 file changed, 1 insertion(+), 1 deletion(-)
ronald@pc-ron:~/hello-project$ 
```

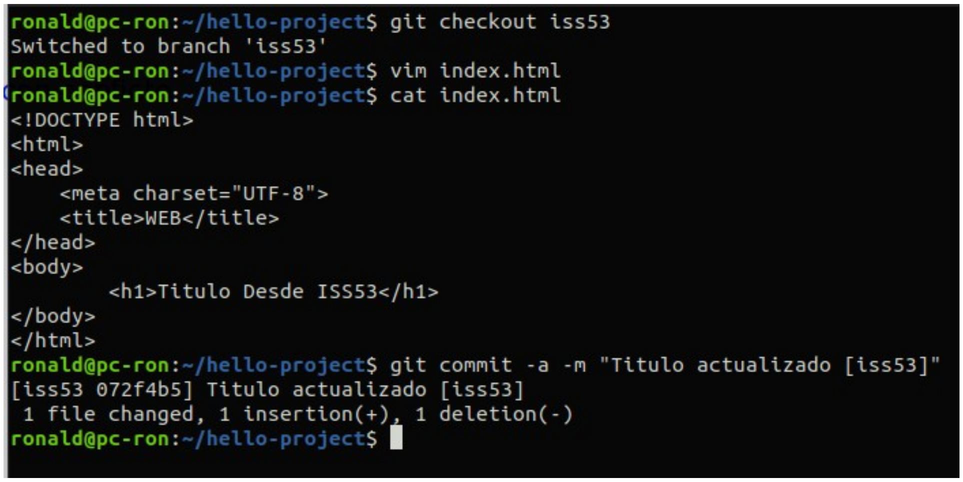


```
ronald@pc-ron:~/hello-project$ git checkout main
Switched to branch 'main'
Your branch is ahead of 'origin/main' by 1 commit.
  (use "git push" to publish your local commits)
ronald@pc-ron:~/hello-project$ git merge iss53
Merge made by the 'recursive' strategy.
 index.html | 10 ++++++++++
 1 file changed, 10 insertions(+)
 create mode 100644 index.html
ronald@pc-ron:~/hello-project$ 
```

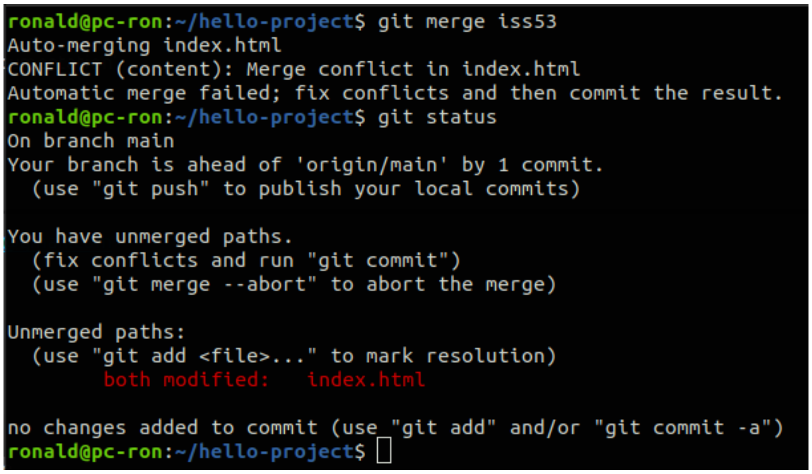
Merge con conficto



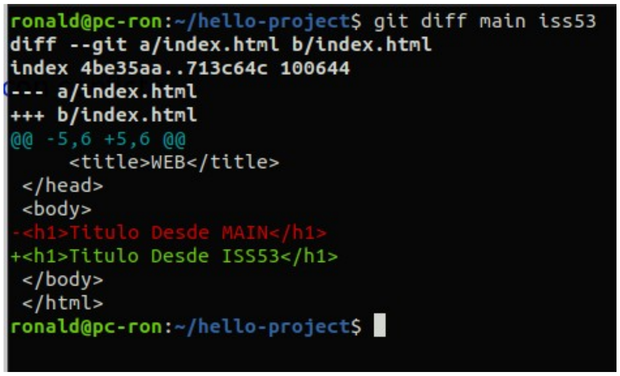
```
ronald@pc-ron:~/hello-project$ git checkout main
Already on 'main'
Your branch is up to date with 'origin/main'.
ronald@pc-ron:~/hello-project$ vim index.html
ronald@pc-ron:~/hello-project$ git commit -a -m "Titulo actualizado [main]"
[main 105bb4a] Titulo actualizado [main]
 1 file changed, 1 insertion(+), 1 deletion(-)
ronald@pc-ron:~/hello-project$ cat index.html
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>WEB</title>
</head>
<body>
        <h1>Titulo Desde MAIN</h1>
</body>
</html>
ronald@pc-ron:~/hello-project$
```



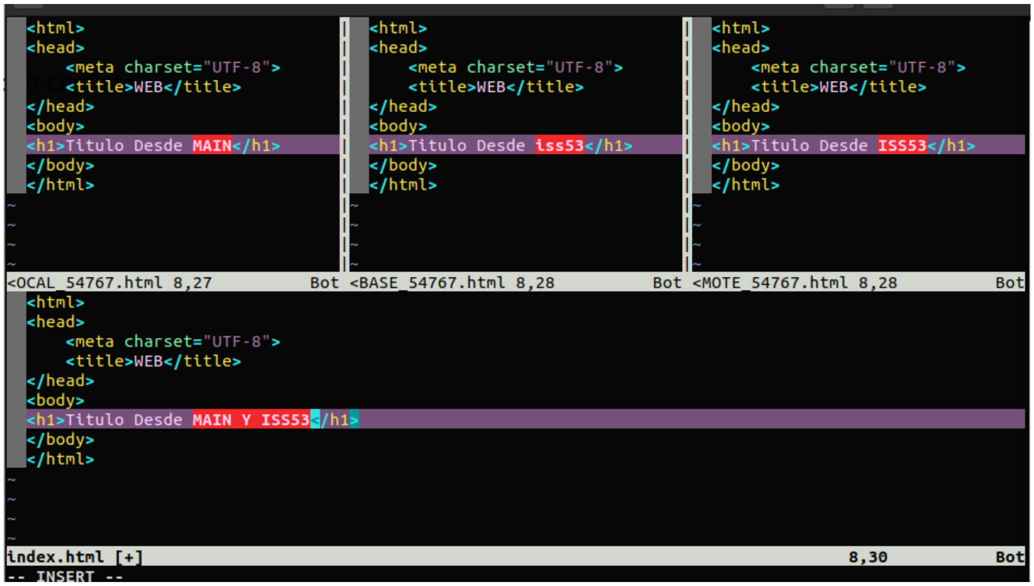
```
ronald@pc-ron:~/hello-project$ git checkout iss53
Switched to branch 'iss53'
ronald@pc-ron:~/hello-project$ vim index.html
ronald@pc-ron:~/hello-project$ cat index.html
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>WEB</title>
</head>
<body>
        <h1>Titulo Desde ISS53</h1>
</body>
</html>
ronald@pc-ron:~/hello-project$ git commit -a -m "Titulo actualizado [iss53]"
[iss53 072f4b5] Titulo actualizado [iss53]
 1 file changed, 1 insertion(+), 1 deletion(-)
ronald@pc-ron:~/hello-project$
```



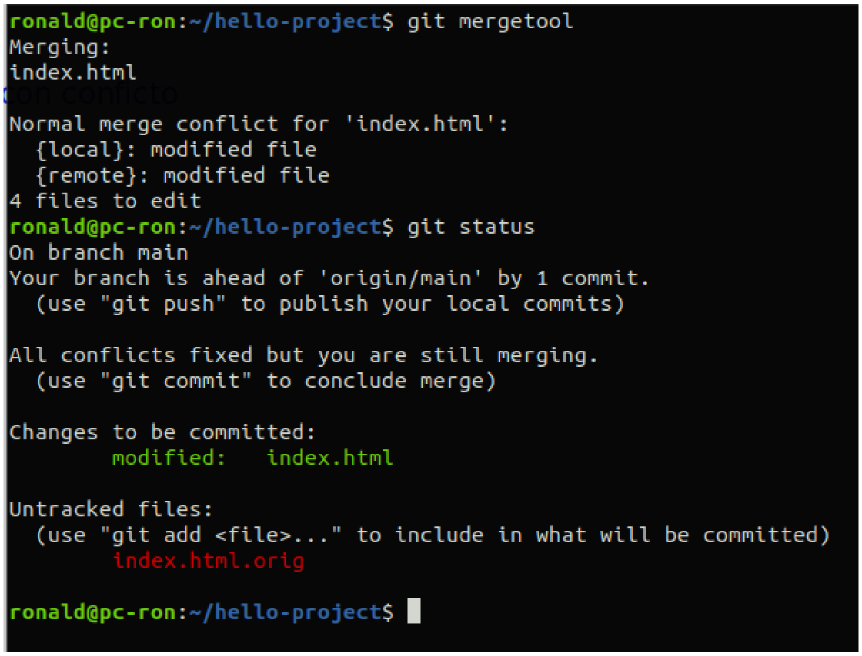
```
ronald@pc-ron:~/hello-project$ git merge iss53
Auto-merging index.html
CONFLICT (content): Merge conflict in index.html
Automatic merge failed; fix conflicts and then commit the result.
ronald@pc-ron:~/hello-project$ git status
On branch main
Your branch is ahead of 'origin/main' by 1 commit.
  (use "git push" to publish your local commits)

You have unmerged paths.
  (fix conflicts and run "git commit")
  (use "git merge --abort" to abort the merge)

Unmerged paths:
  (use "git add <file>..." to mark resolution)
        both modified:   index.html

no changes added to commit (use "git add" and/or "git commit -a")
ronald@pc-ron:~/hello-project$
```



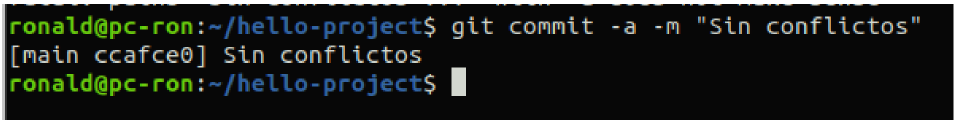
```
ronald@pc-ron:~/hello-project$ git diff main iss53
diff --git a/index.html b/index.html
index 4be35aa..713c64c 100644
--- a/index.html
+++ b/index.html
@@ -5,6 +5,6 @@
     <title>WEB</title>
 </head>
 <body>
-<h1>Titulo Desde MAIN</h1>
+<h1>Titulo Desde ISS53</h1>
 </body>
 </html>
ronald@pc-ron:~/hello-project$
```



```
<html>                            <html>                            <html>
<head>                            <head>                            <head>
    <meta charset="UTF-8">            <meta charset="UTF-8">            <meta charset="UTF-8">
    <title>WEB</title>                <title>WEB</title>                <title>WEB</title>
</head>                           </head>                           </head>
<body>                            <body>                            <body>
<h1>Titulo Desde MAIN</h1>        <h1>Titulo Desde iss53</h1>       <h1>Titulo Desde ISS53</h1>
</body>                           </body>                           </body>
</html>                           </html>                           </html>
~                                 ~                                 ~
~                                 ~                                 ~
~                                 ~                                 ~
~                                 ~                                 ~
<OCAL_54767.html 8,27        Bot <BASE_54767.html 8,28        Bot <MOTE_54767.html 8,28        Bot
<html>
<head>
    <meta charset="UTF-8">
    <title>WEB</title>
</head>
<body>
<h1>Titulo Desde MAIN Y ISS53</h1>
</body>
</html>
~
~
~
~
index.html [+]                                                              8,30           Bot
-- INSERT --
```



```
ronald@pc-ron:~/hello-project$ git mergetool
Merging:
index.html

Normal merge conflict for 'index.html':
  {local}: modified file
  {remote}: modified file
4 files to edit
ronald@pc-ron:~/hello-project$ git status
On branch main
Your branch is ahead of 'origin/main' by 1 commit.
  (use "git push" to publish your local commits)

All conflicts fixed but you are still merging.
  (use "git commit" to conclude merge)

Changes to be committed:
        modified:   index.html

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        index.html.orig

ronald@pc-ron:~/hello-project$
```



```
ronald@pc-ron:~/hello-project$ git commit -a -m "Sin conflictos"
[main ccafce0] Sin conflictos
ronald@pc-ron:~/hello-project$ 
```