

DESARROLLO DE SOFTWARE

ACTIVIDAD 4

Diego Manuel Delgado Velarde



Facultad de ciencias – Universidad Nacional de Ingeniería

EJERCICIO 1



OBJETIVO

Practicar la creacion, fusion y eliminacion de ramas, así como la resolucion de conflictos que puedan surgir durante la fusion.

Crear una nueva rama para una característica:

- Crea una nueva rama llamada feature/advanced-feature desde la rama main:

```
$ git branch feature/advanced-feature
```

```
$ git checkout feature/advanced-feature
```

Modificar archivos en la nueva rama:

- Edita el archivo main.py para incluir una función adicional:

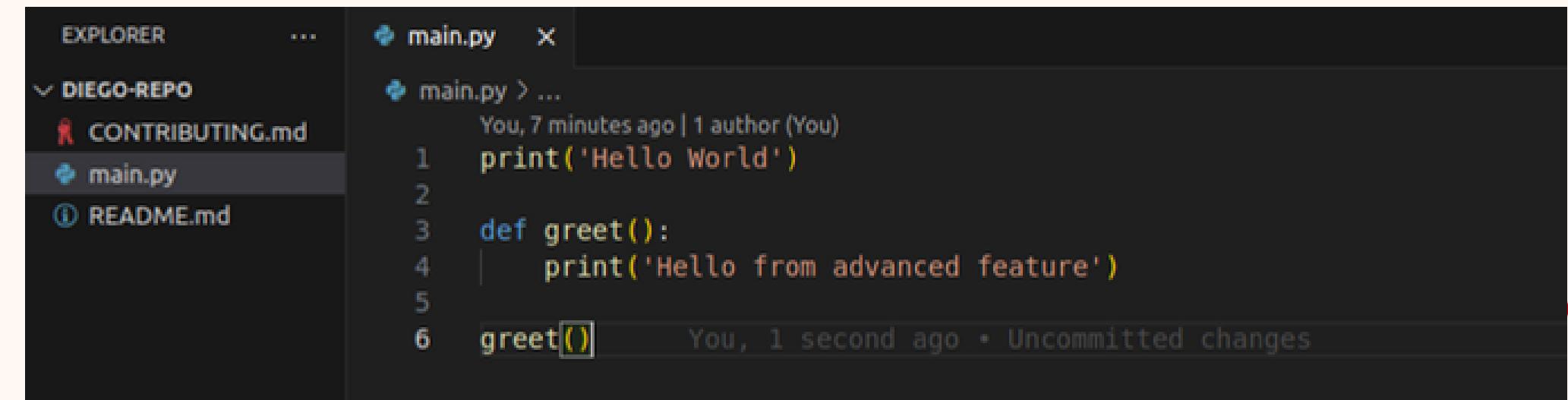
```
def greet():
    print('Hello from advanced feature')
```

```
greet()
```

- Añade y confirma estos cambios en la rama feature/advanced-feature:

```
$ git add main.py
```

```
$ git commit -m "Add greet function in advanced feature"
```



```
diegodev@HPavilion:~/Desktop/dev-practice/diego-repo$ git checkout -b feature/advanced-feature
Switched to a new branch 'feature/advanced-feature'
diegodev@HPavilion:~/Desktop/dev-practice/diego-repo$ code .
diegodev@HPavilion:~/Desktop/dev-practice/diego-repo$ git add main.py
diegodev@HPavilion:~/Desktop/dev-practice/diego-repo$ git commit -m "Add greet function in advanced feature"
[feature/advanced-feature debc732] Add greet function in advanced feature
 1 file changed, 5 insertions(+)
diegodev@HPavilion:~/Desktop/dev-practice/diego-repo$
```

Simular un desarrollo paralelo en la rama master:

- Cambia de nuevo a la rama main:

```
$ git checkout main
```

- Añade y confirma estos cambios en la rama master:

```
$ git add main.py
```

```
$ git commit -m "Update main.py message in main branch"
```

- Edita el archivo main.py de forma diferente (por ejemplo, cambia el mensaje del print original):

```
print('Hello World - updated in main')
```

```
EXPLORER ...  
DIEGO-REPO  
CONTRIBUTING.md  
main.py M  
README.md  
You, 41 seconds ago | 1 author (You)  
1 | print('Hello World - updated in main')
```

Intentar fusionar la rama feature/advanced-feature en main:

- Fusiona la rama feature/advanced-feature en master:

```
$ git merge feature/advanced-feature
```

- Después de resolver el conflicto, añade el archivo resuelto y completa la fusión:

```
$ git add main.py
```

```
$ git commit -m "Resolve merge conflict between main and feature/advanced-feature"
```

Eliminar la rama fusionada:

Resolver el conflicto de fusión:

- Git generará un conflicto en main.py. Abre el archivo y resuelve el conflicto manualmente, eligiendo cómo combinar las dos versiones.

```
main.py > ...  
You, 13 seconds ago | 1 author (You)  
1 print('Hello World - updated in main and merge with feature')  
2  
3 def greet():  
4 | print('Hello from advanced feature')  
5
```

```
diegodev@HPavilion:~/Desktop/dev-practice/diego-repo$ git checkout main  
Switched to branch 'main'  
diegodev@HPavilion:~/Desktop/dev-practice/diego-repo$ code .  
diegodev@HPavilion:~/Desktop/dev-practice/diego-repo$ git add main.py  
diegodev@HPavilion:~/Desktop/dev-practice/diego-repo$ git commit -m "Update main.py message in main branch"  
[main de1c18c] Update main.py message in main branch  
1 file changed, 1 insertion(+), 1 deletion(-)  
diegodev@HPavilion:~/Desktop/dev-practice/diego-repo$ git merge feature/advanced-feature  
Auto-merging main.py  
CONFLICT (content): Merge conflict in main.py  
Automatic merge failed; fix conflicts and then commit the result.  
diegodev@HPavilion:~/Desktop/dev-practice/diego-repo$ code .  
diegodev@HPavilion:~/Desktop/dev-practice/diego-repo$ git commit -m "Resolve merge conflict between main and feature/advanced-feature"  
[main 76d4bcf] Resolve merge conflict between main and feature/advanced-feature  
diegodev@HPavilion:~/Desktop/dev-practice/diego-repo$ git branch -d feature/advanced-feature  
Deleted branch feature/advanced-feature (was debc732).  
diegodev@HPavilion:~/Desktop/dev-practice/diego-repo$
```



EJERCICIO 2



OBJETIVO

Aprender a navegar y manipular el historial de commits usando comandos avanzados de Git

Ejecutando el siguiente comando, veremos en muchísimo más detalle los commits realizados en el repositorio, mostrándose los cambios realizados:

```
$ git log -p
```

```
diegodev@HPavilion: ~/Desktop/dev-practice/diego-repo

commit 76d4bcfbc73f0c910fb4a8a4d9b63bfd526e5f52 (HEAD -> main)
Merge: de1c18c debc732
Author: Diego <manuelchema999@gmail.com>
Date:   Sun Apr 13 12:47:29 2025 -0500

    Resolve merge conflict between main and feature/advanced-feature

commit de1c18c3c777c9c95d462a331a00007ca32b37b8
Author: Diego <manuelchema999@gmail.com>
Date:   Sun Apr 13 12:44:52 2025 -0500

    Update main.py message in main branch

diff --git a/main.py b/main.py
index df1dc68..78f55b5 100644
--- a/main.py
+++ b/main.py
@@ -1 +1 @@
-print('Hello World')
+print('Hello World - updated in main')
\ No newline at end of file

commit debc7324a3c2698f93e2252f5e51aecd32aed3db
Author: Diego <manuelchema999@gmail.com>
Date:   Sun Apr 13 12:35:00 2025 -0500

    Add greet function in advanced feature

diff --git a/main.py b/main.py
index df1dc68..7c50a20 100644
--- a/main.py
+++ b/main.py
@@ -1 +1,6 @@
 print('Hello World')
+
+def greet():
+    print('Hello from advanced feature')
+
+greet()
\ No newline at end of file

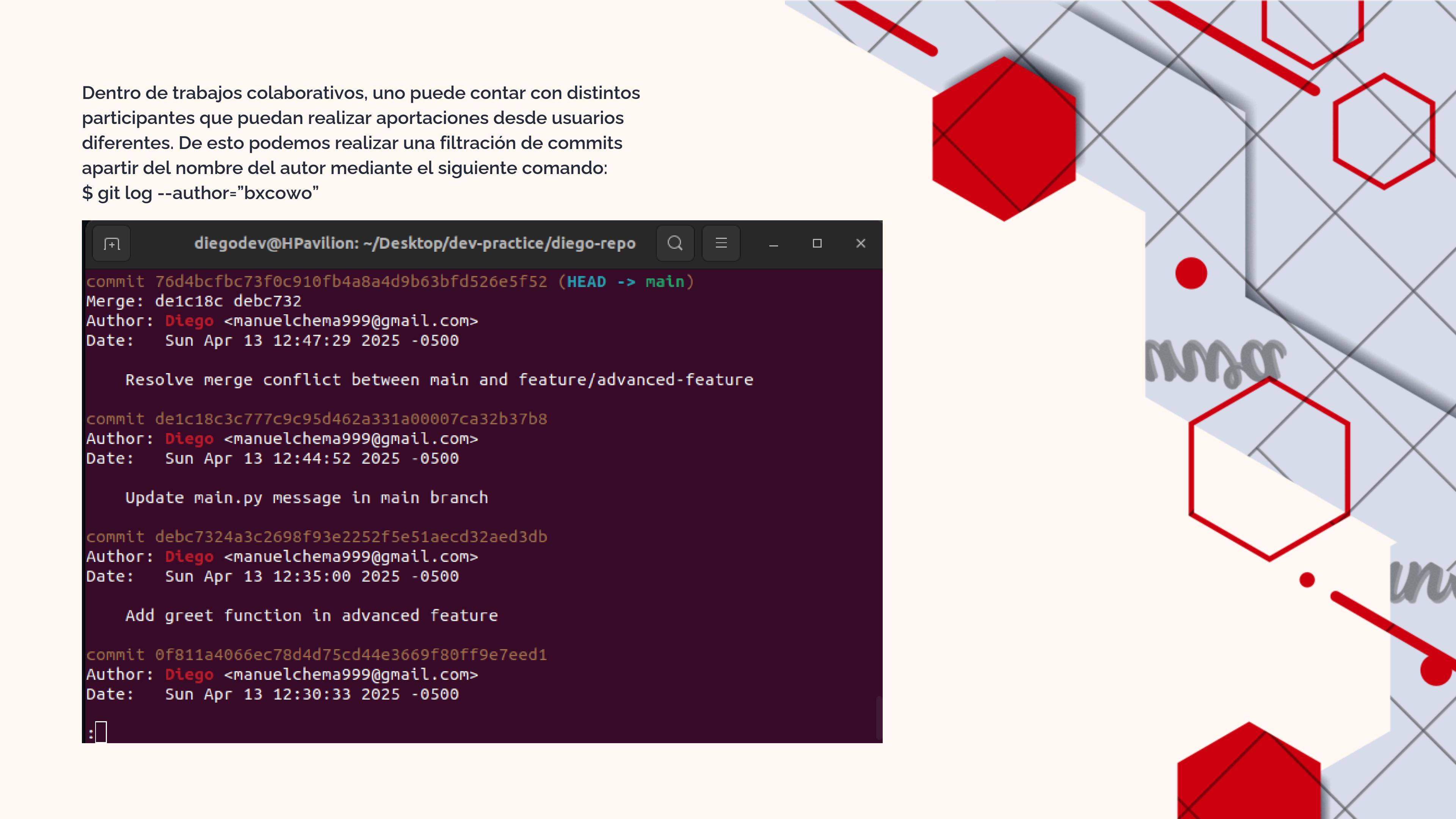
commit 0f811a4066ec78d4d75cd44e3669f80ff9e7eed1
Author: Diego <manuelchema999@gmail.com>
Date:   Sun Apr 13 12:30:33 2025 -0500

    Add main.py

diff --git a/main.py b/main.py
new file mode 100644
index 0000000..df1dc68
--- /dev/null
+++ b/main.py
@@ -0,0 +1 @@
+print('Hello World')
```

Dentro de trabajos colaborativos, uno puede contar con distintos participantes que puedan realizar aportaciones desde usuarios diferentes. De esto podemos realizar una filtración de commits apartir del nombre del autor mediante el siguiente comando:

```
$ git log --author="bxcowo"
```



```
diegodev@HPavilion: ~/Desktop/dev-practice/diego-repo
```

```
commit 76d4bcfbc73f0c910fb4a8a4d9b63bfd526e5f52 (HEAD -> main)
Merge: de1c18c debc732
Author: Diego <manuelchema999@gmail.com>
Date:   Sun Apr 13 12:47:29 2025 -0500

    Resolve merge conflict between main and feature/advanced-feature

commit de1c18c3c777c9c95d462a331a00007ca32b37b8
Author: Diego <manuelchema999@gmail.com>
Date:   Sun Apr 13 12:44:52 2025 -0500

    Update main.py message in main branch

commit debc7324a3c2698f93e2252f5e51aec32aed3db
Author: Diego <manuelchema999@gmail.com>
Date:   Sun Apr 13 12:35:00 2025 -0500

    Add greet function in advanced feature

commit 0f811a4066ec78d4d75cd44e3669f80ff9e7eed1
Author: Diego <manuelchema999@gmail.com>
Date:   Sun Apr 13 12:30:33 2025 -0500
```

```
:
```

Sin embargo, si sucediese algún error durante el desarrollo y se necesitase regresar a una versión anterior se pueden usar el siguiente comando que permita dicha reversión:

```
$ git revert HEAD
```

Se nos inicializará en la carpeta .git el archivo COMMIT_EDITMSG donde podremos un mensaje para el commit regresado.

Ahora si verificasemos nuestro historial de commits usando \$ git log, notaremos los cambios hechos.

```
diegodev@HPavilion:~/Desktop/dev-practice/diego-repo$ git revert HEAD  
[main 75fe508] Revert "add nuevo print"  
 1 file changed, 2 deletions(-)  
diegodev@HPavilion:~/Desktop/dev-practice/diego-repo$ git log --oneline  
75fe508 (HEAD -> main) Revert "add nuevo print"  
19874bc add nuevo print  
76d4bcf Resolve merge conflict between main and feature/advanced-feature  
de1c18c Update main.py message in main branch  
debc732 Add greet function in advanced feature  
0f811a4 Add main.py  
4496723 Set up the repository base documentation  
diegodev@HPavilion:~/Desktop/dev-practice/diego-repo$ code .  
diegodev@HPavilion:~/Desktop/dev-practice/diego-repo$
```

Así mismo, podemos realizar un rebase interactivo que permita gestionar varios commits, para el siguiente ejemplo combinaremos 3 commits en uno solo, esto simplifica de gran forma los logs generados para una mejor gestión. Usaremos el siguiente comando:

```
$ git rebase -i HEAD~3
```

Es apartir de este comando que estamos considerando los tres últimos commits desde HEAD para alguno de los commandos presentes en los comentarios, usaremos squash para combinarlos todos en uno solo.



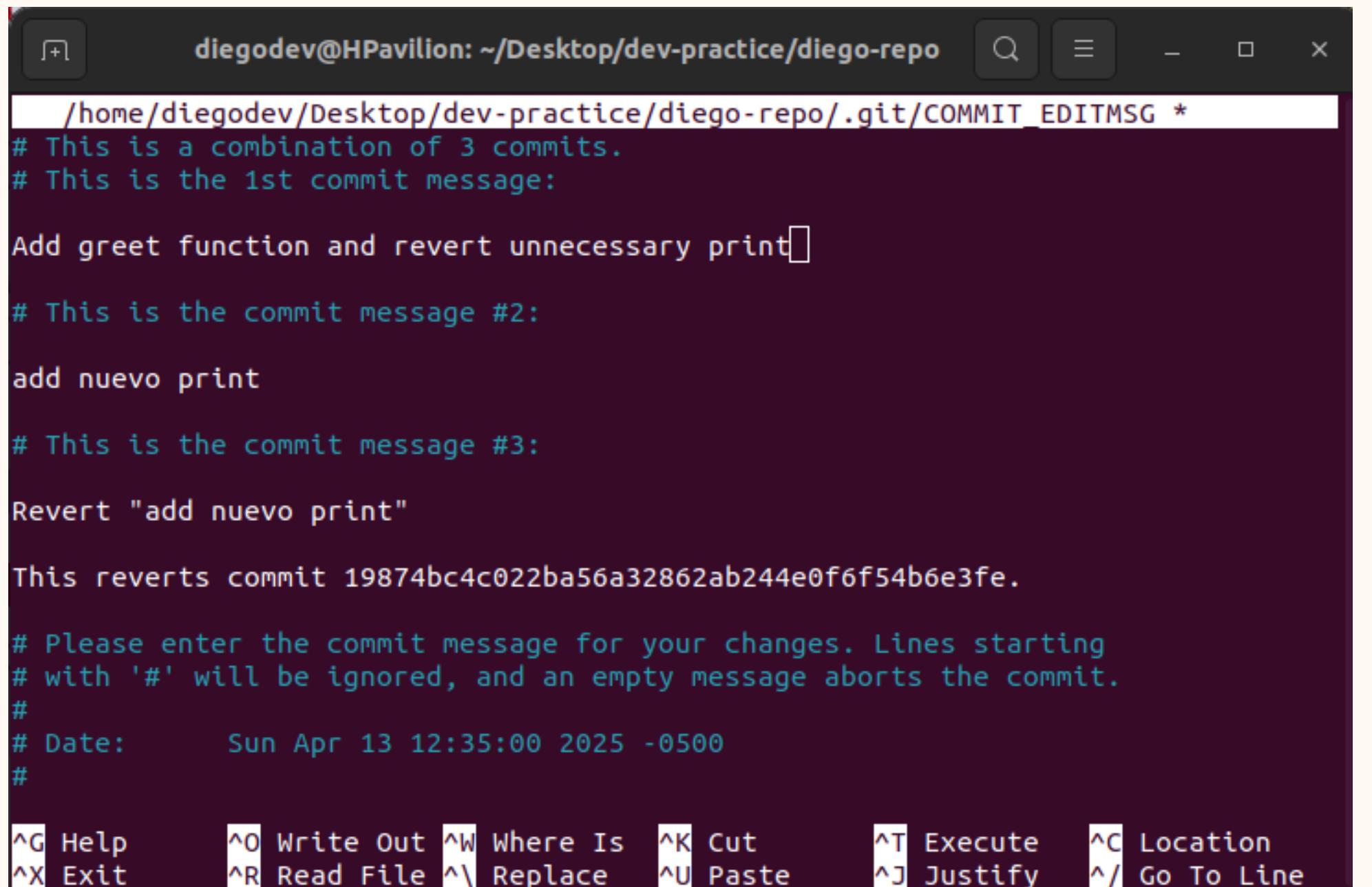
```
diegodev@HPavilion: ~/Desktop/dev-practice/diego-repo
```

```
...dev/Desktop/dev-practice/diego-repo/.git/rebase-merge/git-rebase-todo *
pick debc732 Add greet function in advanced feature
squash 19874bc add nuevo print
squash 75fe508 Revert "add nuevo print"

# Rebase de1c18c..75fe508 onto de1c18c (3 commands)
#
# Commands:
# p, pick <commit> = use commit
# r, reword <commit> = use commit, but edit the commit message
# e, edit <commit> = use commit, but stop for amending
# s, squash <commit> = use commit, but meld into previous commit
# f, fixup [-C | -c] <commit> = like "squash" but keep only the previous
#                               commit's log message, unless -C is used, in which case
#                               keep only this commit's message; -c is same as -C but
#                               opens the editor
# x, exec <command> = run command (the rest of the line) using shell
# b, break = stop here (continue rebase later with 'git rebase --continue')
# d, drop <commit> = remove commit
# l, label <label> = label current HEAD with a name
# t, reset <label> = reset HEAD to a label

^G Help      ^O Write Out  ^W Where Is   ^K Cut        ^T Execute    ^C Location
^X Exit      ^R Read File  ^\ Replace    ^U Paste      ^J Justify    ^/ Go To Line
```

Al haber realizado correctamente la ejecución del comando, en nuestro caso tuvimos conflicto y se solucionaron en el editor, ahora se nos abrirá nuevamente el editor para la implementación de un mensaje asociado a ellos.



```
diegodev@HPavilion: ~/Desktop/dev-practice/diego-repo
```

```
/home/diegodev/Desktop/dev-practice/diego-repo/.git/COMMIT_EDITMSG *
```

```
# This is a combination of 3 commits.
```

```
# This is the 1st commit message:
```

```
Add greet function and revert unnecessary print
```

```
# This is the commit message #2:
```

```
add nuevo print
```

```
# This is the commit message #3:
```

```
Revert "add nuevo print"
```

```
This reverts commit 19874bc4c022ba56a32862ab244e0f6f54b6e3fe.
```

```
# Please enter the commit message for your changes. Lines starting
# with '#' will be ignored, and an empty message aborts the commit.
#
# Date:      Sun Apr 13 12:35:00 2025 -0500
#
^G Help      ^O Write Out  ^W Where Is  ^K Cut        ^T Execute  ^C Location
^X Exit      ^R Read File  ^\ Replace   ^U Paste     ^J Justify  ^/ Go To Line
```

Finalmente, dentro de nuestros logs podremos observar los últimos cambios realizados como una combinación de los 3 últimos commits realizados.

```
diegodev@HPavilion: ~/Desktop/dev-practice/diego-repo
```

```
diegodev@HPavilion:~/Desktop/dev-practice/diego-repo$ git rebase -i HEAD~3
fatal: It seems that there is already a rebase-merge directory, and
I wonder if you are in the middle of another rebase. If that is the
case, please try
    git rebase (--continue | --abort | --skip)
If that is not the case, please
    rm -fr ".git/rebase-merge"
and run me again. I am stopping in case you still have something
valuable there.

diegodev@HPavilion:~/Desktop/dev-practice/diego-repo$ git add main.py
diegodev@HPavilion:~/Desktop/dev-practice/diego-repo$ git rebase --continue
[detached HEAD a2d6b42] Add greet function and print
 1 file changed, 4 insertions(+), 1 deletion(-)
[detached HEAD eac7c95] Add greet function and revert unnecessary print
  Date: Sun Apr 13 12:35:00 2025 -0500
  1 file changed, 4 insertions(+), 1 deletion(-)
Successfully rebased and updated refs/heads/main.
diegodev@HPavilion:~/Desktop/dev-practice/diego-repo$ git log --oneline
eac7c95 (HEAD -> main) Add greet function and revert unnecessary print
de1c18c Update main.py message in main branch
0f811a4 Add main.py
4496723 Set up the repository base documentation
diegodev@HPavilion:~/Desktop/dev-practice/diego-repo$
```

Otra forma alternativa de visualización de commits será a través de una representación gráfica que permita un entendimiento mucho más adecuado del log asociado al repositorio. Para realizarlo ejecutaremos el siguiente comando:

```
$ git log --graph --oneline --all
```

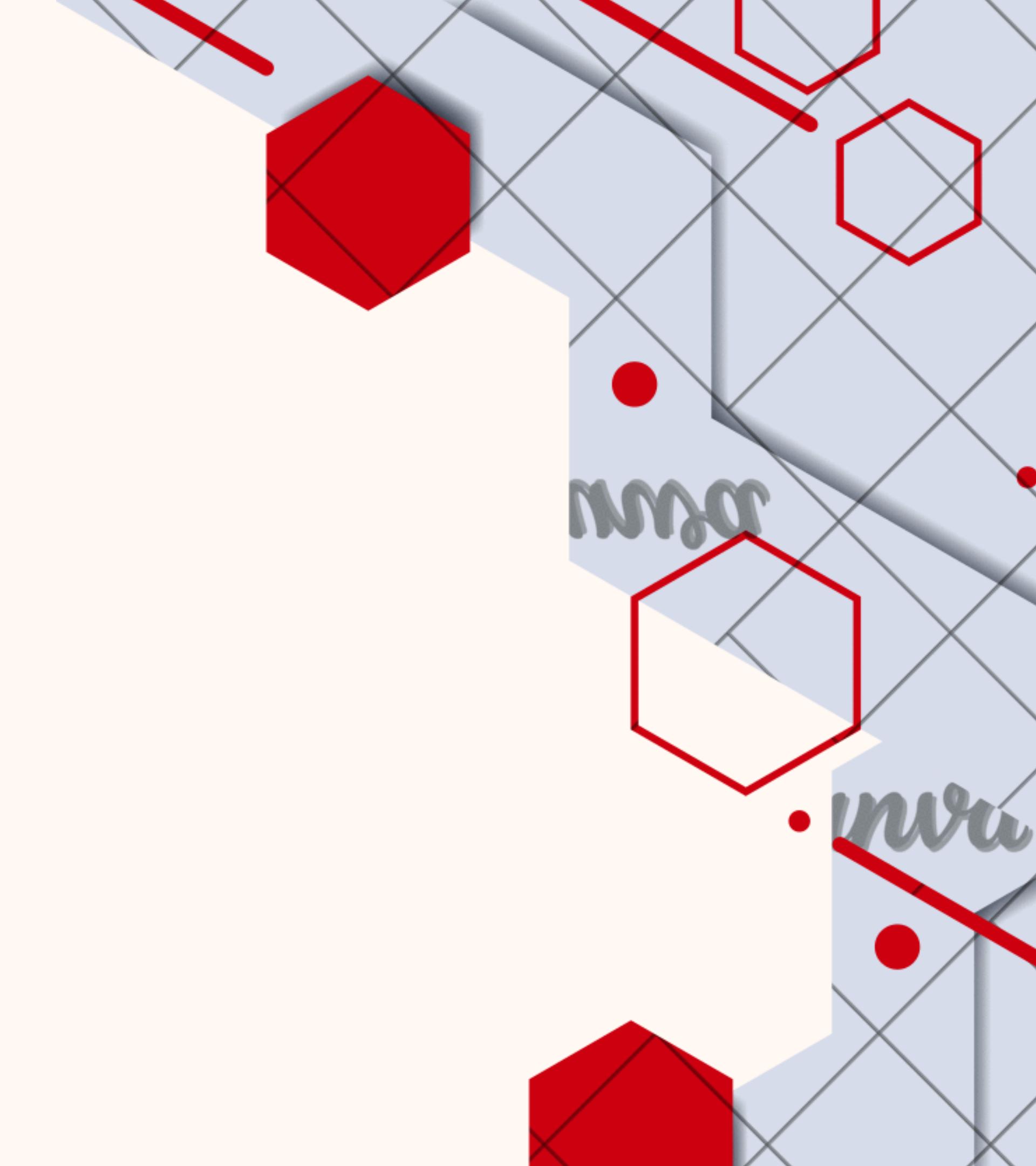
```
diegodev@HPavilion:~/Desktop/dev-practice/diego-repo$ git log --graph --oneline  
--all  
* eac7c95 (HEAD -> main) Add greet function and revert unnecessary print  
* de1c18c Update main.py message in main branch  
* 0f811a4 Add main.py  
* 4496723 Set up the repository base documentation  
diegodev@HPavilion:~/Desktop/dev-practice/diego-repo$
```

EJERCICIO 3



OBJETIVO

Practicar la creación de ramas desde commits específicos y comprender cómo Git maneja las referencias históricas.



Crear una nueva rama desde un commit específico:

Modificar y confirmar cambios en la nueva rama:

Fusionar los cambios en la rama principal:

Explorar el historial después de la fusión:

Eliminar la rama bugfix/rollback-feature:

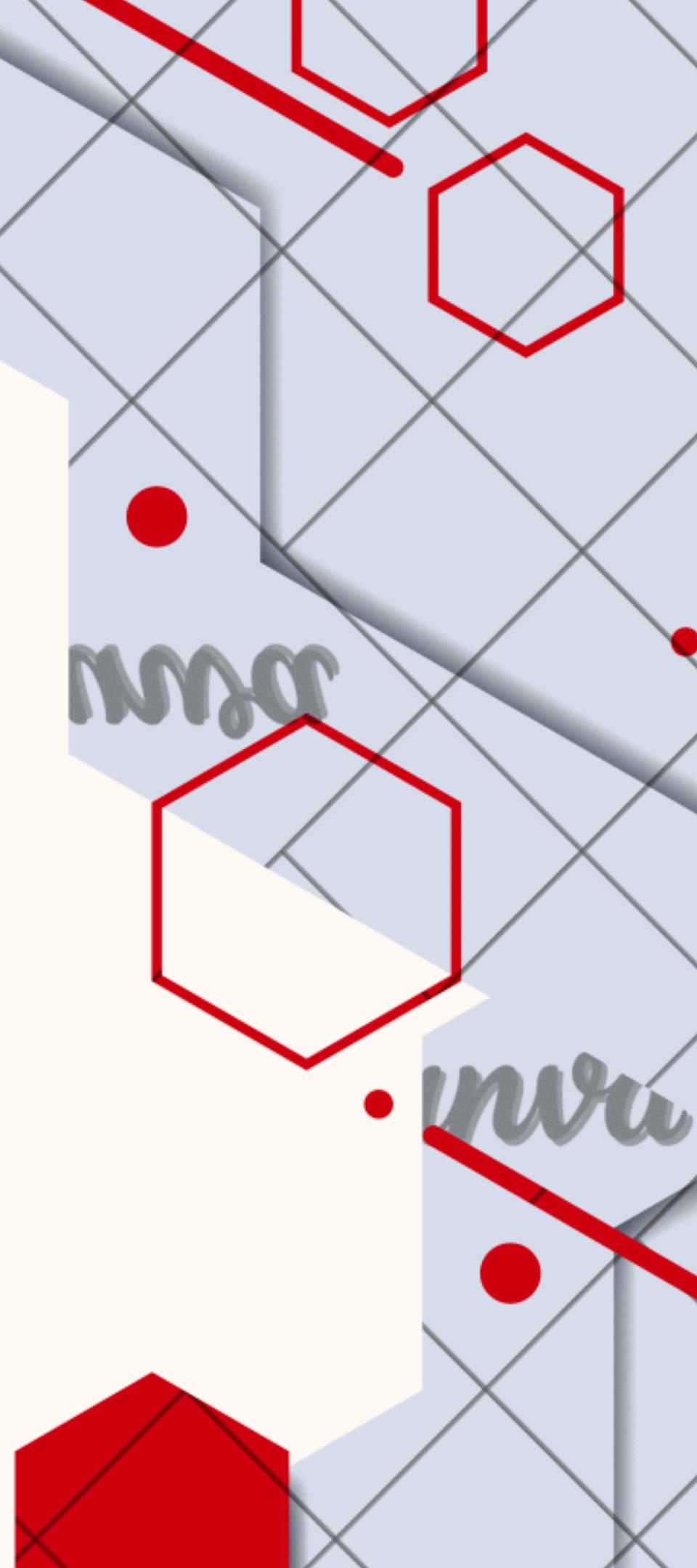
```
diegodev@HPavilion:~/Desktop/dev-practice/diego-repo$ git log --oneline
eac7c95 (HEAD -> main) Add greet function and revert unnecessary print
de1c18c Update main.py message in main branch
0f811a4 Add main.py
4496723 Set up the repository base documentation
diegodev@HPavilion:~/Desktop/dev-practice/diego-repo$ git branch bugfix/rollback-feature de1c18c
diegodev@HPavilion:~/Desktop/dev-practice/diego-repo$ git checkout bugfix/rollback-feature
Switched to branch 'bugfix/rollback-feature'
diegodev@HPavilion:~/Desktop/dev-practice/diego-repo$ code .
diegodev@HPavilion:~/Desktop/dev-practice/diego-repo$ git add main.py
diegodev@HPavilion:~/Desktop/dev-practice/diego-repo$ git commit -m "Fix bug in rollback feature"
[bugfix/rollback-feature 59a0216] Fix bug in rollback feature
 1 file changed, 4 insertions(+), 1 deletion(-)
diegodev@HPavilion:~/Desktop/dev-practice/diego-repo$ git checkout main
Switched to branch 'main'
diegodev@HPavilion:~/Desktop/dev-practice/diego-repo$ git merge bugfix/rollback-feature
Auto-merging main.py
CONFLICT (content): Merge conflict in main.py
Automatic merge failed; fix conflicts and then commit the result.
diegodev@HPavilion:~/Desktop/dev-practice/diego-repo$ code .
diegodev@HPavilion:~/Desktop/dev-practice/diego-repo$ git add main.py
diegodev@HPavilion:~/Desktop/dev-practice/diego-repo$ git commit -m "update main"
[main abc5d78] update main
diegodev@HPavilion:~/Desktop/dev-practice/diego-repo$ git branch -d bugfix/rollback-feature
Deleted branch bugfix/rollback-feature (was 59a0216).
diegodev@HPavilion:~/Desktop/dev-practice/diego-repo$
```

EJERCICIO 4



OBJETIVO

Comprender cómo usar git reset y git restore para deshacer cambios en el historial y en el área de trabajo



Hacer cambios en el archivo main.py:

Usar git reset para deshacer el commit:

```
diegodev@HPavilion:~/Desktop/dev-practice/diego-repo$ code .
diegodev@HPavilion:~/Desktop/dev-practice/diego-repo$ git add main.py
diegodev@HPavilion:~/Desktop/dev-practice/diego-repo$ git commit -m "Introduce a change to be reset"
[main 134f99e] Introduce a change to be reset
 1 file changed, 1 insertion(+), 1 deletion(-)
diegodev@HPavilion:~/Desktop/dev-practice/diego-repo$ git reset --hard HEAD~1
HEAD is now at abc5d78 update main
diegodev@HPavilion:~/Desktop/dev-practice/diego-repo$ git log --oneline
abc5d78 (HEAD -> main) update main
59a0216 Fix bug in rollback feature
eac7c95 Add greet function and revert unnecessary print
de1c18c Update main.py message in main branch
0f811a4 Add main.py
4496723 Set up the repository base documentation
diegodev@HPavilion:~/Desktop/dev-practice/diego-repo$
```

Usar git restore para deshacer cambios no confirmados:

```
diegodev@HPavilion:~/Desktop/dev-practice/diego-repo$ echo "Another line in README" >> README.md
diegodev@HPavilion:~/Desktop/dev-practice/diego-repo$ git status
On branch main
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")
diegodev@HPavilion:~/Desktop/dev-practice/diego-repo$ git restore README.md
diegodev@HPavilion:~/Desktop/dev-practice/diego-repo$ git status
On branch main
nothing to commit, working tree clean
diegodev@HPavilion:~/Desktop/dev-practice/diego-repo$
```

EJERCICIO 5



OBJETIVO

Simular un flujo de trabajo colaborativo utilizando ramas y pull requests.

Crear un nuevo repositorio remoto:

- Usa GitHub o GitLab para crear un nuevo repositorio remoto y clónalo localmente: \$ git clone <URL-del-repositorio>

```
diegodev@HPavilion:~/Desktop/dev-practice$ git clone git@github.com:Ox-Chema-x0/team-practice.git
Cloning into 'team-practice'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (3/3), done.
diegodev@HPavilion:~/Desktop/dev-practice$
```

Crear una nueva rama para desarrollo de una característica:

- En tu repositorio local, crea una nueva rama feature/team-feature:
\$ git branch feature/team-feature
\$ git checkout feature/team-feature

```
diegodev@HPavilion:~/Desktop/dev-practice$ git status
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean
diegodev@HPavilion:~/Desktop/dev-practice$ git branch feature/team-feature
diegodev@HPavilion:~/Desktop/dev-practice$ git checkout feature/team-feature
Switched to branch 'feature/team-feature'
```

Realizar cambios y enviar la rama al repositorio remoto:

- Realiza cambios en los archivos del proyecto y confírmalos:

```
$ echo "print('Collaboration is key!') > collaboration.py
```

```
$ git add .
```

```
$ git commit -m "Add collaboration script"
```

```
diegodev@HPavilion:~/Desktop/dev-practice/team-practice$ echo "print('Collaboration is key!') > collaboration.py"
diegodev@HPavilion:~/Desktop/dev-practice/team-practice$ git add .
diegodev@HPavilion:~/Desktop/dev-practice/team-practice$ git commit -m "Add collaboration script"
[feature/team-feature 4cbad76] Add collaboration script
 1 file changed, 1 insertion(+)
  create mode 100644 collaboration.py
diegodev@HPavilion:~/Desktop/dev-practice/team-practice$
```

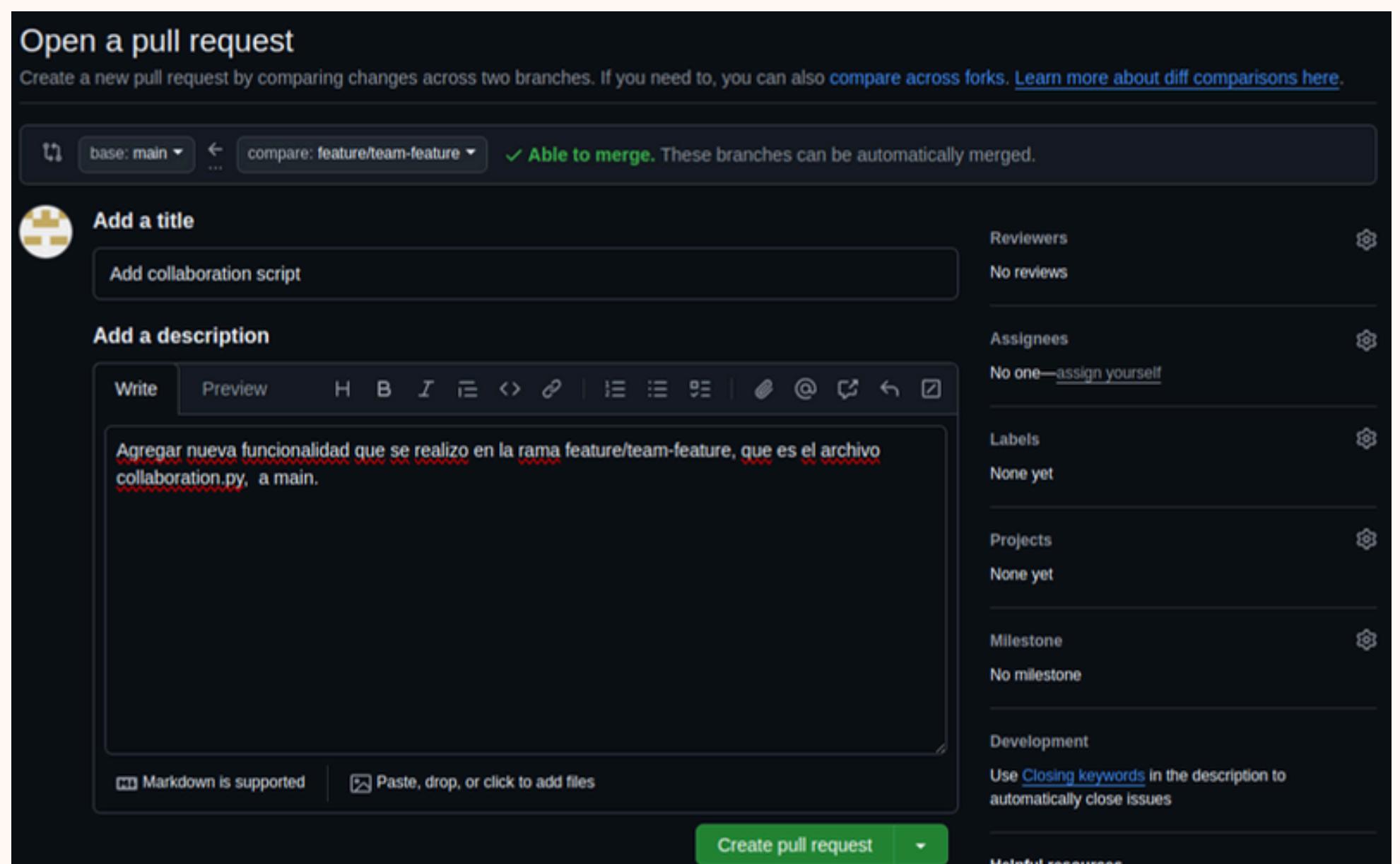
- Envía la rama al repositorio remoto:

```
$ git push origin feature/team-feature
```

```
diegodev@HPavilion:~/Desktop/dev-practice/team-practice$ git push origin feature/team-feature
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 12 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 325 bytes | 325.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
remote:
remote: Create a pull request for 'feature/team-feature' on GitHub by visiting:
remote:     https://github.com/Ox-Chema-x0/team-practice/pull/new/feature/team-feature
remote:
To github.com:Ox-Chema-x0/team-practice.git
 * [new branch]      feature/team-feature -> feature/team-feature
```

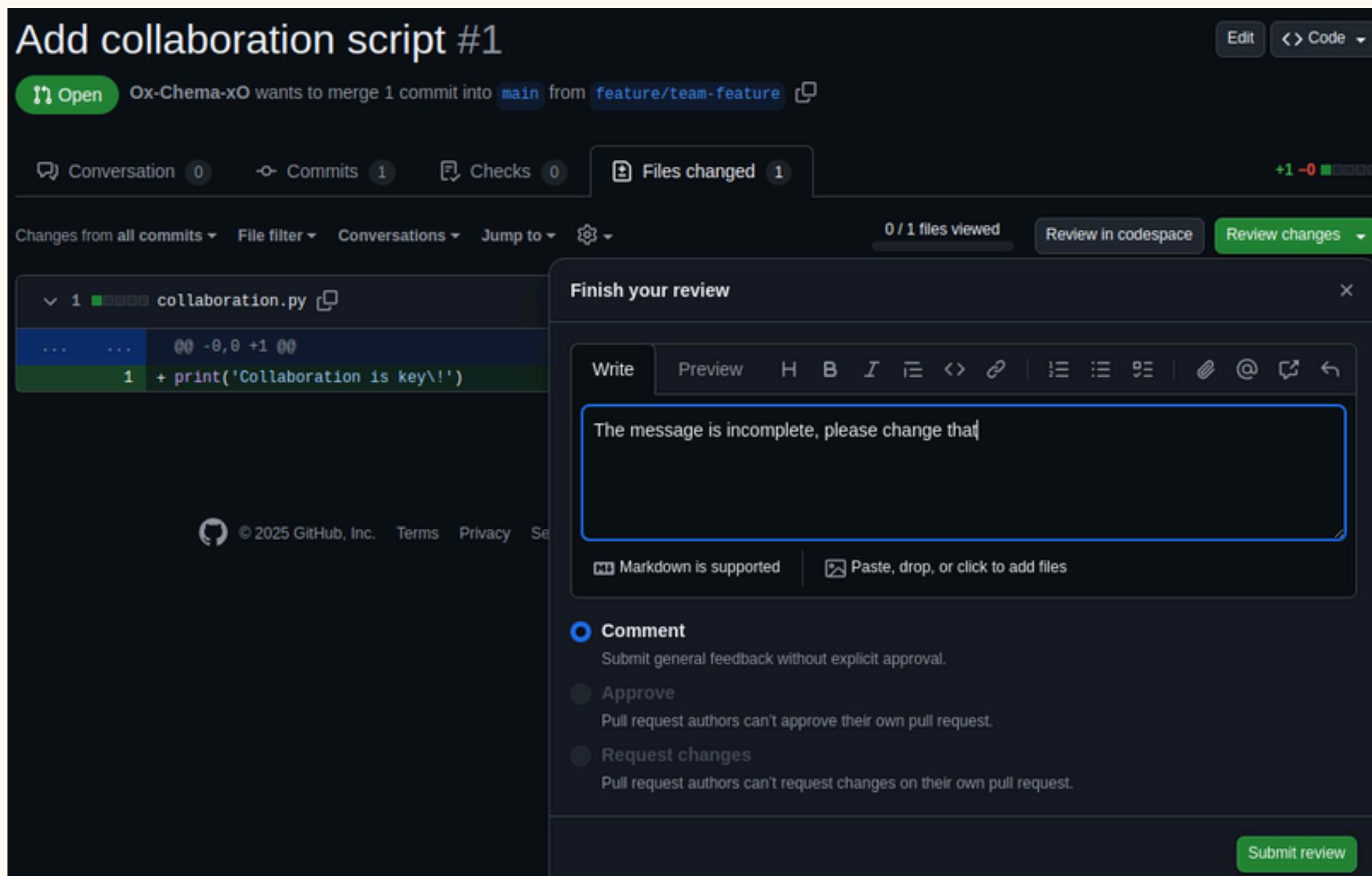
Abrir un Pull Request:

- Abre un Pull Request (PR) en la plataforma remota (GitHub/GitLab) para fusionar feature/team-feature con la rama main.
- Añade una descripción detallada del PR, explicando los cambios realizados y su propósito.



Revisar y fusionar el Pull Request:

- Simula la revisión de código, comenta en el PR y realiza cualquier cambio necesario basado en la retroalimentación.



Revisar y fusionar el Pull Request:

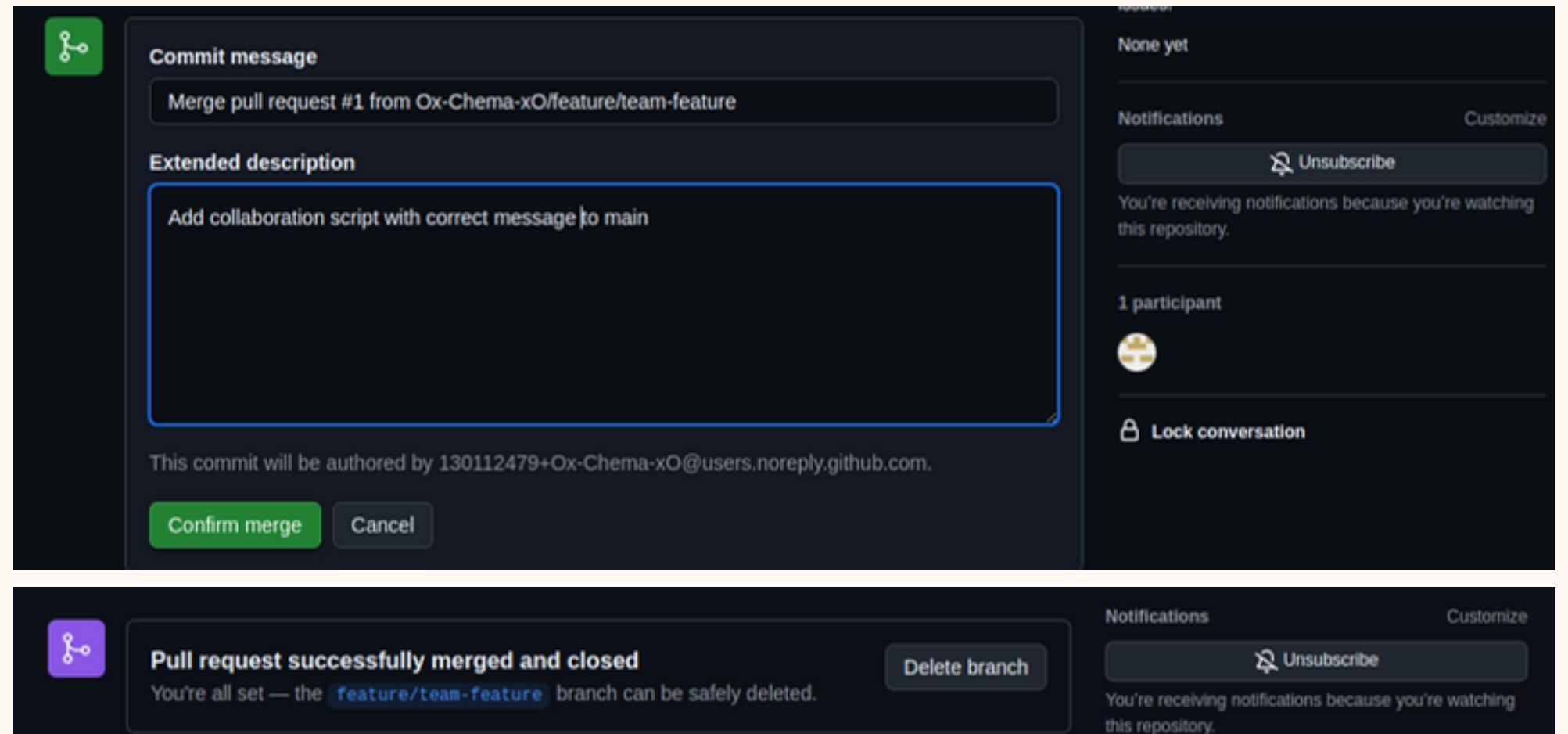
- Simula la revisión de código, comenta en el PR y realiza cualquier cambio necesario basado en la retroalimentación.

```
diegodev@HPavilion:~/Desktop/dev-practice/team-practice$ git branch
* feature/team-feature
  main
diegodev@HPavilion:~/Desktop/dev-practice/team-practice$ git status
On branch feature/team-feature
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:   collaboration.py

no changes added to commit (use "git add" and/or "git commit -a")
diegodev@HPavilion:~/Desktop/dev-practice/team-practice$ git add collaboration.py
diegodev@HPavilion:~/Desktop/dev-practice/team-practice$ git commit -m "Mensaje correcto para collaboration file"
[feature/team-feature fbeea8a] Mensaje correcto para collaboration file
 1 file changed, 1 insertion(+), 1 deletion(-)
diegodev@HPavilion:~/Desktop/dev-practice/team-practice$ git push origin feature/team-feature
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 12 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 372 bytes | 372.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To github.com:Ox-Chema-x0/team-practice.git
  4cbad76..fbeea8a  feature/team-feature -> feature/team-feature
diegodev@HPavilion:~/Desktop/dev-practice/team-practice$
```

Revisar y fusionar el Pull Request:

- Una vez aprobado, fusiona el PR en la rama main.



Eliminar la rama remota y local:

- Despues de la fusión, elimina la rama tanto local como remotamente:

```
$ git branch -d feature/team-feature
```

```
$ git push origin --delete feature/team-feature
```

```
diegodev@HPavilion:~/Desktop/dev-practice/team-practice$ git branch -d feature/team-feature
Deleted branch feature/team-feature (was fbeea8a).
diegodev@HPavilion:~/Desktop/dev-practice/team-practice$ git push origin --delete feature/team-feature
To github.com:Ox-Chema-xO/team-practice.git
 - [deleted]      feature/team-feature
diegodev@HPavilion:~/Desktop/dev-practice/team-practice$
```

EJERCICIO 6



OBJETIVO

Aprender a aplicar commits específicos a otra rama utilizando git cherry-pick y a guardar temporalmente cambios no confirmados utilizando git stash.

Hacer cambios en main.py y confirmarlos:

- Realiza y confirma varios cambios en main.py en la rama main:

```
$ echo "print('Cherry pick this!')" >> main.py  
$ git add main.py  
$ git commit -m "Add cherry-pick example"
```

```
diegodev@HPavilion:~/Desktop/dev-practice/activ4-git$ echo "print('Cherry pick this!')" >> main.py  
diegodev@HPavilion:~/Desktop/dev-practice/activ4-git$ git add main.py  
diegodev@HPavilion:~/Desktop/dev-practice/activ4-git$ git commit -m "Add cherry-pick example"  
[main ef2d04d] Add cherry-pick example  
 1 file changed, 1 insertion(+)
```

Crear una nueva rama y aplicar el commit específico:

- Crea una nueva rama feature/cherry-pick y aplícale el commit específico:

```
$ git branch feature/cherry-pick  
$ git checkout feature/cherry-pick
```

```
diegodev@HPavilion:~/Desktop/dev-practice/activ4-git$ git branch feature/cherry-pick  
diegodev@HPavilion:~/Desktop/dev-practice/activ4-git$ git checkout feature/cherry-pick  
Switched to branch 'feature/cherry-pick'
```

Crear una nueva rama y aplicar el commit específico:

- Crea una nueva rama feature/cherry-pick y aplícale el commit específico:

```
$ git log  
$ git cherry-pick <commit-hash>
```

```
diegodev@HPavilion:~/Desktop/dev-practice/activ4-git$ git log  
commit ef2d04d85243b4a1b695351f3a7142aa06277a49 (HEAD -> main)  
Author: Diego <manuelchema999@gmail.com>  
Date:   Fri Apr 4 09:32:40 2025 -0500  
  
    Add cherry-pick example  
  
commit 49b487def6559570da8d2354fd46d9c461bdeeca (feature/cherry-pick)  
Author: Diego <manuelchema999@gmail.com>  
Date:   Thu Apr 3 08:35:44 2025 -0500  
  
    add main.py  
  
commit 444795fad309ff90d18b5783a2e84dc0baf75910  
Author: Diego <manuelchema999@gmail.com>  
Date:   Thu Apr 3 08:32:43 2025 -0500  
  
    Initial commit with README.md
```

```
diegodev@HPavilion:~/Desktop/dev-practice/activ4-git$ git switch feature/cherry-pick  
Switched to branch 'feature/cherry-pick'  
diegodev@HPavilion:~/Desktop/dev-practice/activ4-git$ git cherry-pick ef2d04d85243b4a1b695351f3a71  
42aa06277a49  
[feature/cherry-pick b14aae1] Add cherry-pick example  
Date: Fri Apr 4 09:32:40 2025 -0500  
1 file changed, 1 insertion(+)
```

Guardar temporalmente cambios no confirmados:

- Realiza algunos cambios en main.py pero no los confirme:

```
$ echo "This change is stashed" >> main.py
```

```
$ git status
```

```
diegodev@HPavilion:~/Desktop/dev-practice/activ4-git$ echo "This change is stashed" >> main.py
diegodev@HPavilion:~/Desktop/dev-practice/activ4-git$ git status
On branch main
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:   main.py

no changes added to commit (use "git add" and/or "git commit -a")
```

- Guarda temporalmente estos cambios utilizando git stash:

```
$ git stash
```

```
diegodev@HPavilion:~/Desktop/dev-practice/activ4-git$ git stash
Saved working directory and index state WIP on main: ef2d04d Add cherry-pick example
diegodev@HPavilion:~/Desktop/dev-practice/activ4-git$ git status
On branch main
nothing to commit, working tree clean
```

Aplicar los cambios guardados:

- Realiza otros cambios y confírmalos si es necesario.
- Luego, recupera los cambios guardados anteriormente:

```
$ git stash pop
```

Aplicar los cambios guardados:

- Realiza otros cambios y confírmalos si es necesario.
- Luego, recupera los cambios guardados anteriormente:

```
$ git stash pop
```

```
diegodev@HPavilion:~/Desktop/dev-practice/activ4-git$ echo "new change for practice that" >> main.py
```

```
diegodev@HPavilion:~/Desktop/dev-practice/activ4-git$ git add main.py  
diegodev@HPavilion:~/Desktop/dev-practice/activ4-git$ git commit -m "Add new change into main.py"  
[main 8078ca4] Add new change into main.py  
 1 file changed, 1 insertion(+)
```

- Luego, recupera los cambios guardados anteriormente:

```
$ git stash pop
```

```
diegodev@HPavilion:~/Desktop/dev-practice/activ4-git$ git stash pop  
Auto-merging main.py  
CONFLICT (content): Merge conflict in main.py  
On branch main  
Unmerged paths:  
  (use "git restore --staged <file>..." to unstage)  
  (use "git add <file>..." to mark resolution)  
    both modified:  main.py  
  
no changes added to commit (use "git add" and/or "git commit -a")  
The stash entry is kept in case you need it again.
```

Revisar el historial y confirmar la correcta aplicación de los cambios:

- Usa git log para revisar el historial de commits y verificar que todos los cambios se han aplicado correctamente.

```
commit ba38e4d26d489c22a6db3a2acdb0f38b04393e17 (HEAD -> main)
Author: Diego <manuelchema999@gmail.com>
Date:   Fri Apr 4 10:35:48 2025 -0500

    Resolve conflicts after git stash pop

commit 8078ca4002e8afdf56d5f82148b34fae89c6979ff
Author: Diego <manuelchema999@gmail.com>
Date:   Fri Apr 4 10:26:44 2025 -0500

    Add new change into main.py

commit ef2d04d85243b4a1b695351f3a7142aa06277a49
Author: Diego <manuelchema999@gmail.com>
Date:   Fri Apr 4 09:32:40 2025 -0500

    Add cherry-pick example

commit 49b487def6559570da8d2354fd46d9c461bdeeca
Author: Diego <manuelchema999@gmail.com>
Date:   Thu Apr 3 08:35:44 2025 -0500

    add main.py
```



**MUCHAS
GRACIAS**