Universidad de Costa Rica

Facultad de Ingeniería
Escuela de Ingeniería Eléctrica
IE-0117 Programación Bajo Plataformas Abiertas
III-ciclo 2022

Laboratorio Paso 10

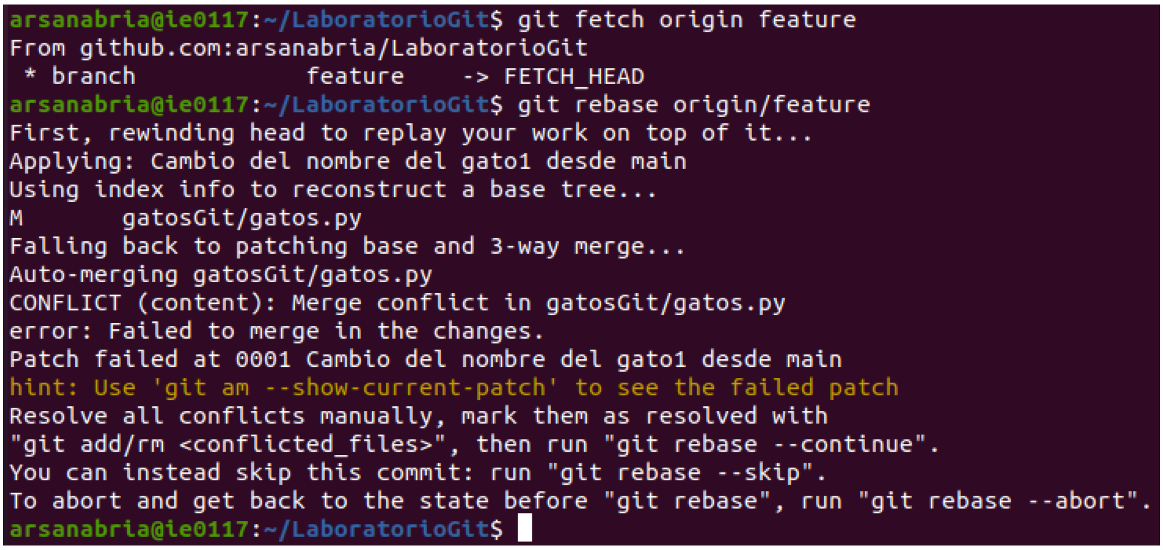
GIT

Aroon Sanabria Torres B97205
Grupo 901

Profesor: Julián Gairaud

10 de febrero de 2023

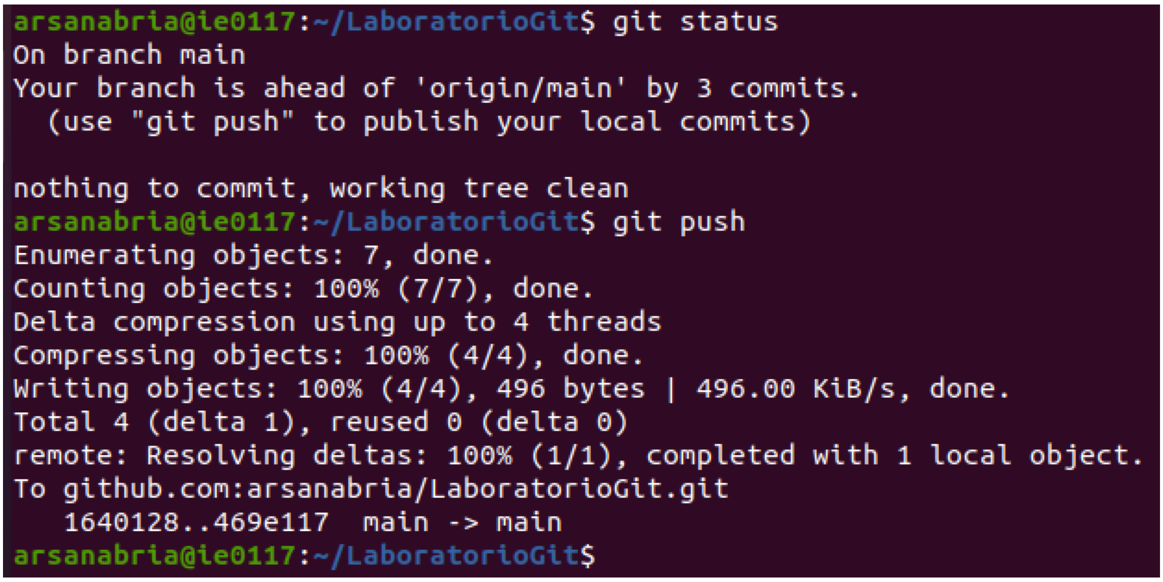
1. Captura de los conflictos entre ambas ramas.



```
arsanabria@ie0117:~/LaboratorioGit$ git fetch origin feature
From github.com:arsanabria/LaboratorioGit
 * branch            feature    -> FETCH_HEAD
arsanabria@ie0117:~/LaboratorioGit$ git rebase origin/feature
First, rewinding head to replay your work on top of it...
Applying: Cambio del nombre del gato1 desde main
Using index info to reconstruct a base tree...
M       gatosGit/gatos.py
Falling back to patching base and 3-way merge...
Auto-merging gatosGit/gatos.py
CONFLICT (content): Merge conflict in gatosGit/gatos.py
error: Failed to merge in the changes.
Patch failed at 0001 Cambio del nombre del gato1 desde main
hint: Use 'git am --show-current-patch' to see the failed patch
Resolve all conflicts manually, mark them as resolved with
"git add/rm <conflicted_files>", then run "git rebase --continue".
You can instead skip this commit: run "git rebase --skip".
To abort and get back to the state before "git rebase", run "git rebase --abort".
arsanabria@ie0117:~/LaboratorioGit$ 
```

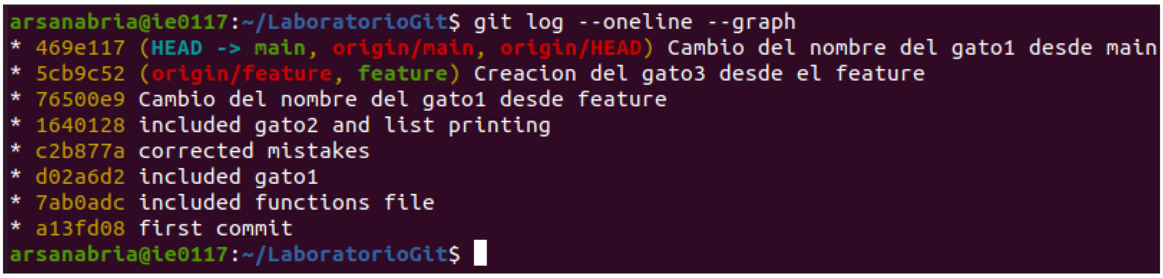
Figura 1: Conflicto de contenido entre ambas ramas.

2. Inclusión de los cambios al repositorio.



```
arsanabria@ie0117:~/LaboratorioGit$ git status
On branch main
Your branch is ahead of 'origin/main' by 3 commits.
  (use "git push" to publish your local commits)

nothing to commit, working tree clean
arsanabria@ie0117:~/LaboratorioGit$ git push
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 4 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (4/4), 496 bytes | 496.00 KiB/s, done.
Total 4 (delta 1), reused 0 (delta 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To github.com:arsanabria/LaboratorioGit.git
   1640128..469e117  main -> main
arsanabria@ie0117:~/LaboratorioGit$
```

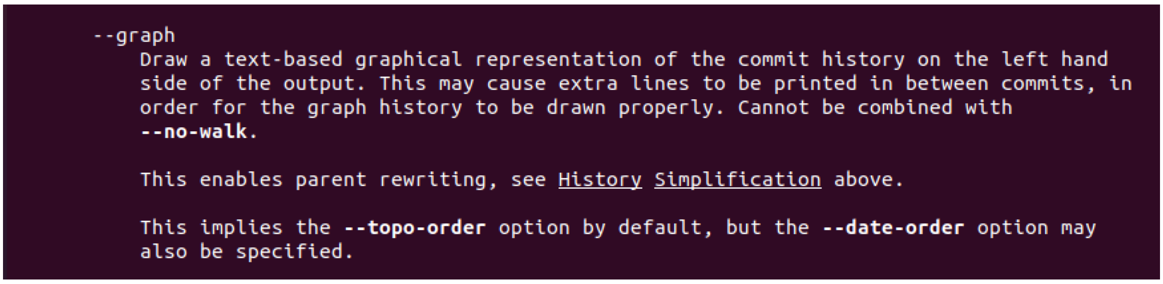
Figura 2: Inclusión de los cambios al repositorio con el comando *git push*.



```
arsanabria@ie0117:~/LaboratorioGit$ git log --oneline --graph
* 469e117 (HEAD -> main, origin/main, origin/HEAD) Cambio del nombre del gato1 desde main
* 5cb9c52 (origin/feature, feature) Creacion del gato3 desde el feature
* 76500e9 Cambio del nombre del gato1 desde feature
* 1640128 included gato2 and list printing
* c2b877a corrected mistakes
* d02a6d2 included gato1
* 7ab0adc included functions file
* a13fd08 first commit
arsanabria@ie0117:~/LaboratorioGit$ 
```

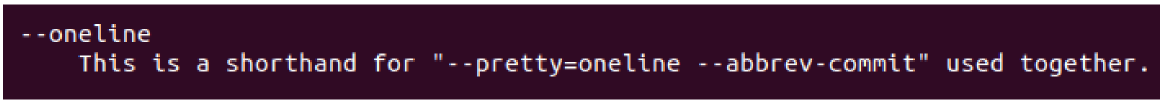
Figura 3: Uso del comando *git log oneline graph*.

3. Manual de git log.



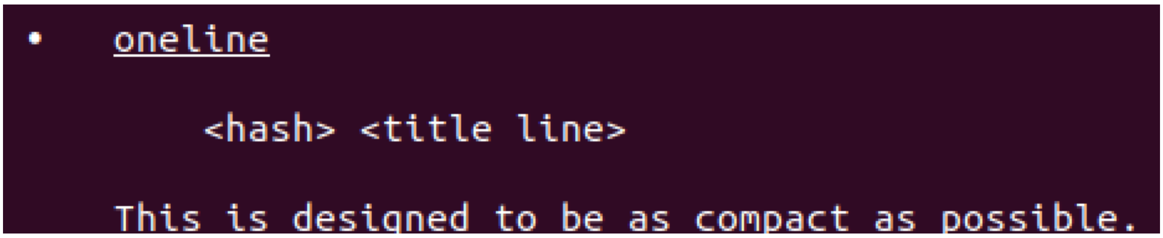
```
    --graph
        Draw a text-based graphical representation of the commit history on the left hand
        side of the output. This may cause extra lines to be printed in between commits, in
        order for the graph history to be drawn properly. Cannot be combined with
        --no-walk.

        This enables parent rewriting, see History Simplification above.

        This implies the --topo-order option by default, but the --date-order option may
        also be specified.
```

Figura 4: Manual de graph.



```
--oneline
    This is a shorthand for "--pretty=oneline --abbrev-commit" used together.
```

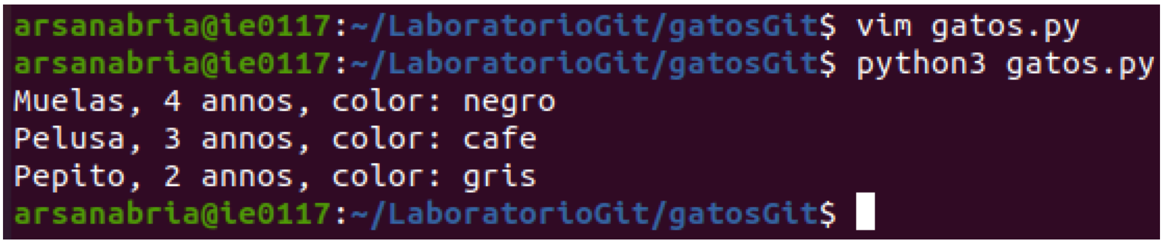
Figura 5: Significado de oneline.



```
•   oneline

        <hash> <title line>

    This is designed to be as compact as possible.
```

Figura 6: Función de oneline

4. Archivo gatos.py corriendo en la computadora.



```
arsanabria@ie0117:~/LaboratorioGit/gatosGit$ vim gatos.py
arsanabria@ie0117:~/LaboratorioGit/gatosGit$ python3 gatos.py
Muelas, 4 annos, color: negro
Pelusa, 3 annos, color: cafe
Pepito, 2 annos, color: gris
arsanabria@ie0117:~/LaboratorioGit/gatosGit$ 
```

Figura 7: Ejecución del archivo gatos.py.