

5:

Figura 13: Mensaje de conflicto al hacer **merge** entre ramas.

```
isaac@campos-vm:~/repositorio_personal/LaboratorioGit/gatosGit$ git status
On branch main
Your branch is ahead of 'origin/main' by 1 commit.
  (use "git push" to publish your local commits)

nothing to commit, working tree clean
isaac@campos-vm:~/repositorio_personal/LaboratorioGit/gatosGit$ git merge feature
Auto-merging gatosGit/gatos.py
CONFLICT (content): Merge conflict in gatosGit/gatos.py
Automatic merge failed; fix conflicts and then commit the result.
isaac@campos-vm:~/repositorio_personal/LaboratorioGit/gatosGit$
```

7:

Figura 18: Inclusión de los cambios creados al repertorio remoto.

```
isaac@campos-vm:~/repositorio_personal/LaboratorioGit/gatosGit$ git push origin main:main
Enumerating objects: 15, done.
Counting objects: 100% (15/15), done.
Delta compression using up to 4 threads
Compressing objects: 100% (9/9), done.
Writing objects: 100% (9/9), 1.04 KiB | 1.04 MiB/s, done.
Total 9 (delta 2), reused 0 (delta 0)
remote: Resolving deltas: 100% (2/2), completed with 1 local object.
To github.com:Damiandcm/LaboratorioGit.git
   1ae709f..4735fe8  main -> main
isaac@campos-vm:~/repositorio_personal/LaboratorioGit/gatosGit$
```

Figura 19: Pull request del branch **main**.

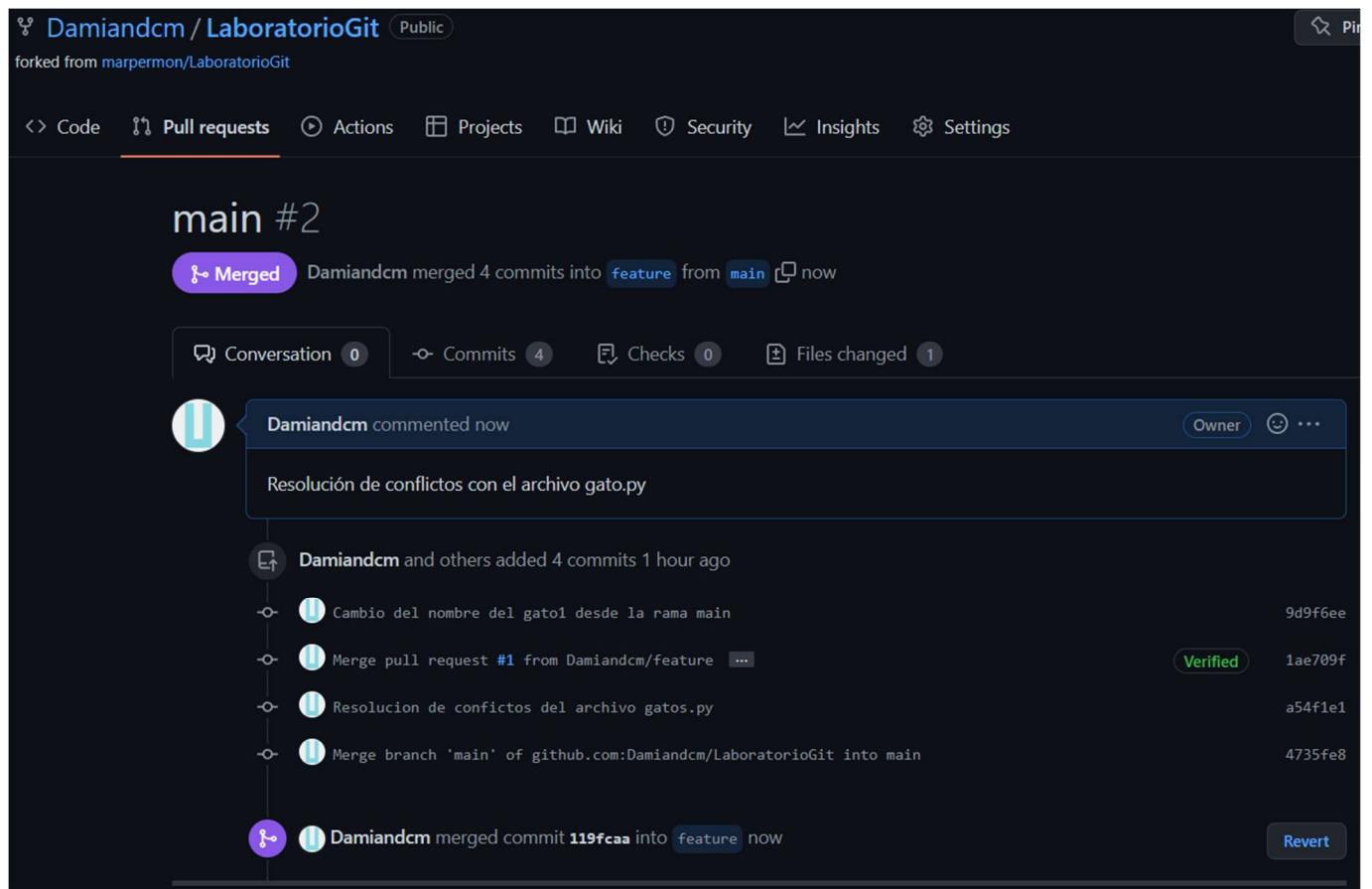


Figura 20: Ejecución del comando `git log --oneline --graph`.

```
isaac@campos-vm:~/repositorio_personal/LaboratorioGit/gatosGit$ git log --oneline --graph
* 4735fe8 (HEAD -> main, origin/main, origin/HEAD) Merge branch 'main' of github.com:Damiancdm/LaboratorioGit into main
|
| * 1ae709f Merge pull request #1 from Damiancdm/feature
|
| * a54f1e1 Resolucion de confictos del archivo gatos.py
|
| |
| | |
| | * ea98f8c (origin/feature, feature) Agregar gato3
| | * af2fe38 Cambio al nombre del gato1
| |
| |
| * / 9d9f6ee Cambio del nombre del gato1 desde la rama main
|
|
| * 1640128 included gato2 and list printing
| * c2b877a corrected mistakes
| * d02a6d2 included gato1
| * 7ab0adc included functions file
| * a13fd08 first commit
isaac@campos-vm:~/repositorio_personal/LaboratorioGit/gatosGit$
```

Figura 21: Uso del comando `git log --graph`.

```
--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 22: Uso del comando `git log --oneline`.

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

Figura 23: Propósito de `oneline`.

```
• oneline

    <hash> <title line>

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

9:

Figura 24: archivo `gatos.py` resultante.

```
isaac@campos-vm:~/repositorio_personal/LaboratorioGit/gatosGit$ cat gatos.py
import funcionesTarea

gato1 = funcionesTarea.gato()
gato1.nombre = "Mishi-fu"
gato1.color = "negro"
gato1.edad = 4
gato1.aniadirlista()

gato2 = funcionesTarea.gato()
gato2.nombre = "Pelusa"
gato2.color = "cafe"
gato2.edad = 3
gato2.aniadirlista()

gato3 = funcionesTarea.gato()
gato3.nombre = "Simon"
gato3.color = "Amarillo"
gato3.edad = 2
gato3.aniadirlista()

funcionesTarea.printlista()
```

*Figura Ejecución del archivo **gatos.py**.*

```
isaac@campos-vm:~/Desktop$ ./gatos.py
Mishi-fu, 4 annos, color: negro
Pelusa, 3 annos, color: cafe
Simon, 2 annos, color: Amarillo
isaac@campos-vm:~/Desktop$
```