

UNIVERSIDAD DE COSTA RICA

ESCUELA DE INGENIERÍA ELÉCTRICA

LABORATORIO GID

IE-0117 Programación Bajo Plataformas Abiertas

Git/GitHub

Profesor:

Julián Gairaud

Estudiante:

Carlos Andrés Cordero Retana B92317

III Semestre 2022

Índice

1. Indicaciones	1
1.1. “Fork” del repositorio	1
1.2. Creación de la rama: “feature”	1
1.2.1. “Commit” que guarda el cambio al nombre del objeto:gato1	2
1.2.2. “Commit” que guarda la creación del objeto:gato3	2
1.2.3. Envío de la rama “feature” al repositorio remoto	3
1.3. Cambio del gato1 desde la rama “main”	4
1.4. Inclusión de los “commits” de la rama “feature” en la rama “main”	4
1.5. Conflicto mostrado en el editor texto NANO	5
1.6. Solución del conflicto de forma manual	6
1.7. Inclusión de los cambios en el repertorio remoto	7
1.8. Empleo del comando: “git log –oneline –graph”	8
1.9. Empleo del comando –help para conocer la funcionalidad de los comandos: –graph y –oneline	8
1.10.Resultado de ejecutar el archivo:“gatos.py”	9
1.11.Creación de un “commit” al añadir un PDF con diferentes screenshots al repositorio local	9
1.12.“Push” de la rama “main” local a la rama “main” del repositorio remoto	9

1. Indicaciones

1.1. “Fork” del repositorio

En primer lugar, se hizo “fork” del repositorio público al repositorio personal, como se muestra en la figura 1. Luego se realiza una copia del mismo en la computadora, tal y como se muestra en la figura 2

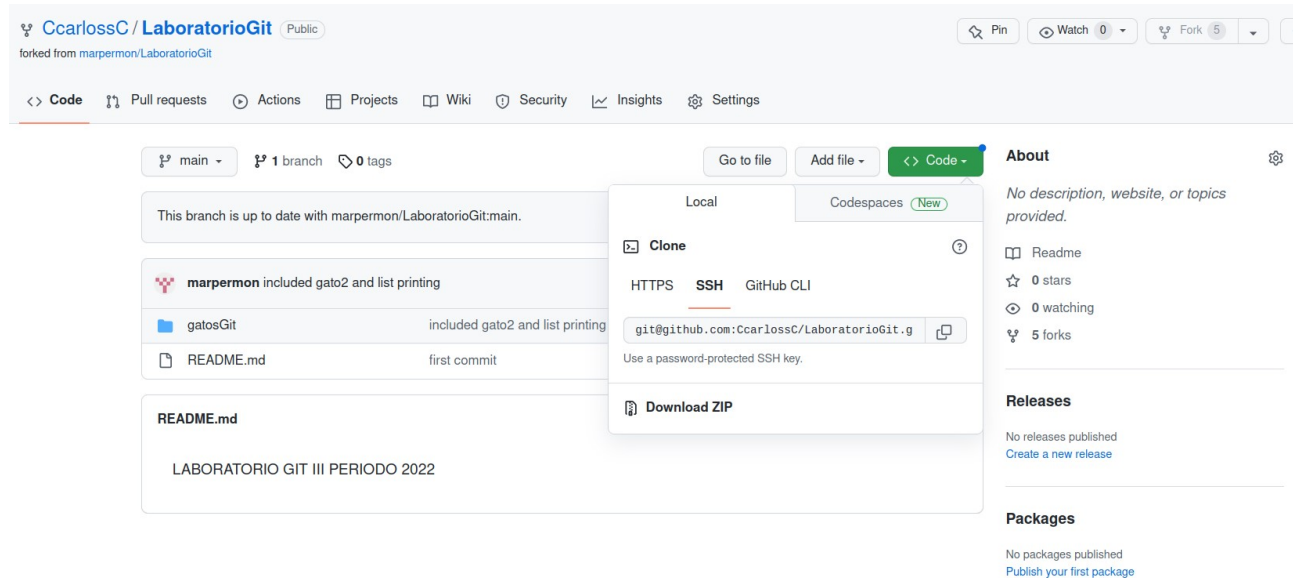


Figura 1: “Fork” del repositorio público

```
ccarlos@carlos-c:~/personal_workbench$ git clone git@github.com:CcarlossC/LaboratorioGit.git
Cloning into 'LaboratorioGit'...
remote: Enumerating objects: 19, done.
remote: Counting objects: 100% (19/19), done.
remote: Compressing objects: 100% (14/14), done.
Receiving objects: 100% (19/19), done.
Resolving deltas: 100% (2/2), done.
remote: Total 19 (delta 2), reused 19 (delta 2), pack-reused 0
ccarlos@carlos-c:~/personal_workbench$
```

Figura 2: Copia del repositorio en la computadora mediante el comando “git clone”

1.2. Creación de la rama: “feature”

```
ccarlos@carlos-c:~/personal_workbench/LaboratorioGit$ git checkout -b feature
Switched to a new branch 'feature'
ccarlos@carlos-c:~/personal_workbench/LaboratorioGit$ git status
On branch feature
nothing to commit, working tree clean
ccarlos@carlos-c:~/personal_workbench/LaboratorioGit$
```

Figura 3: Creación de la rama “feature”

1.2.1. “Commit” que guarda el cambio al nombre del objeto:gato1

```
ccarlos@carlos-c:~/personal_workbench/LaboratorioGit$ git checkout feature
Switched to branch 'feature'
ccarlos@carlos-c:~/personal_workbench/LaboratorioGit$ nano gatosGit/gatos.py
ccarlos@carlos-c:~/personal_workbench/LaboratorioGit$ git add gatosGit/gatos.py
ccarlos@carlos-c:~/personal_workbench/LaboratorioGit$ git commit -m "Cambio del nombre a gato1 desde feature"
[feature 4ca823e] Cambio del nombre a gato1 desde feature
1 file changed, 1 insertion(+), 1 deletion(-)
ccarlos@carlos-c:~/personal_workbench/LaboratorioGit$ git log
commit 4ca823e7e45618cf7883d39558141e253b4837cf (HEAD -> feature)
Author: CcarlossC <124691128+CcarlossC@users.noreply.github.com>
Date: Thu Feb 9 18:08:33 2023 -0600

    Cambio del nombre a gato1 desde feature

commit 1640128b5cd29d26384cbfa07ba2ceca7e336d21 (origin/main, origin/feature, origin/HEAD, main)
Author: marpermon <maria.perezmonge@ucr.ac.cr>
Date: Sat Jan 14 14:17:16 2023 -0600

    included gato2 and list printing
```

Figura 4: Comandos empleados para crear un primer “commit” que contiene el cambio del nombre del objeto: gato1

1.2.2. “Commit” que guarda la creación del objeto:gato3

```
GNU nano 4.8                                gatosGit/gatos.py
import funcionesTarea

gato1 = funcionesTarea.gato()
gato1.nombre = "Oreo"
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 = "Estrella"
gato3.color = "Blanco"
gato3.edad = 6
gato3.aniadirlista()

funcionesTarea.printlista()
```

Figura 5: Modificación del archivo para crear el objeto: gato3

```

ccarlos@carlos-c:~/personal_workbench/LaboratorioGit$ nano gatosGit/gatos.py
ccarlos@carlos-c:~/personal_workbench/LaboratorioGit$ git add gatosGit/gatos.py
ccarlos@carlos-c:~/personal_workbench/LaboratorioGit$ git commit -m "Se crea el objeto gato3"
[feature 6158073] Se crea el objeto gato3
1 file changed, 8 insertions(+)
ccarlos@carlos-c:~/personal_workbench/LaboratorioGit$ git log
commit 615807323a9817c87448bab5f7f35f411b2e7af5 (HEAD -> feature)
Author: CcarlossC <124691128+CcarlossC@users.noreply.github.com>
Date: Thu Feb 9 18:18:55 2023 -0600

    Se crea el objeto gato3

commit 4ca823e7e45618cf7883d39558141e253b4837cf
Author: CcarlossC <124691128+CcarlossC@users.noreply.github.com>
Date: Thu Feb 9 18:08:33 2023 -0600

    Cambio del nombre a gato1 desde feature

commit 1640128b5cd29d26384cbfa07ba2ceca7e336d21 (origin/main, origin/feature, origin/HEAD, main)
Author: marpermon <maria.perezmonge@ucr.ac.cr>
Date: Sat Jan 14 14:17:16 2023 -0600

    included gato2 and list printing

```

Figura 6: Creación del “commit” que contempla la creación del objeto:gato3

1.2.3. Envío de la rama “feature” al repositorio remoto

```

ccarlos@carlos-c:~/personal_workbench/LaboratorioGit$ #__Sincronizamos
ccarlos@carlos-c:~/personal_workbench/LaboratorioGit$ git fetch origin feature
From github.com:CcarlossC/LaboratorioGit
* branch          feature      -> FETCH_HEAD
ccarlos@carlos-c:~/personal_workbench/LaboratorioGit$ git rebase origin/feature
Current branch feature is up to date.
ccarlos@carlos-c:~/personal_workbench/LaboratorioGit$ # Subimos el branch actualizado con los 2 commits
ccarlos@carlos-c:~/personal_workbench/LaboratorioGit$ git push origin feature:feature
Enumerating objects: 11, done.
Counting objects: 100% (11/11), done.
Delta compression using up to 3 threads
Compressing objects: 100% (8/8), done.
Writing objects: 100% (8/8), 904 bytes | 452.00 KiB/s, done.
Total 8 (delta 2), reused 0 (delta 0)
remote: Resolving deltas: 100% (2/2), completed with 1 local object.
To github.com:CcarlossC/LaboratorioGit.git
    1640128..6158073  feature -> feature
ccarlos@carlos-c:~/personal_workbench/LaboratorioGit$

```

Figura 7: Sincronización entre las dos ramas observadas, así como el comando empleado para subir la rama: “feature” al repositorio remoto

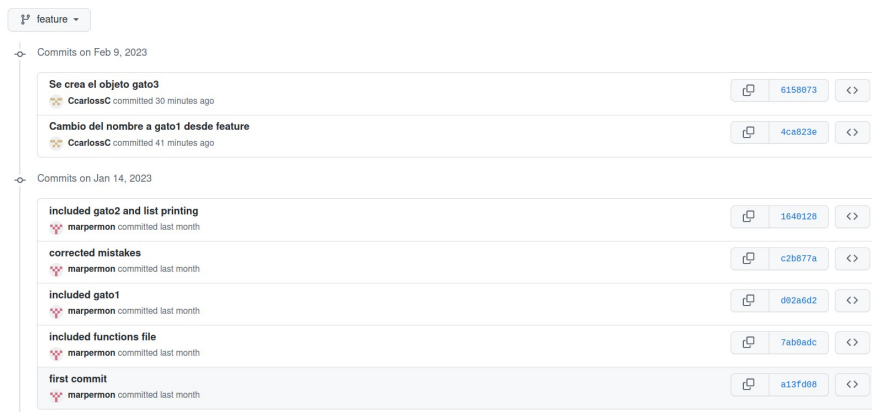


Figura 8: Verificación de que la rama creada se haya subido de forma exitosa al repositorio remoto

1.3. Cambio del gato1 desde la rama “main”

```
ccarlos@carlos-c:~/personal_workbench/LaboratorioGit$ git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.
ccarlos@carlos-c:~/personal_workbench/LaboratorioGit$ nano gatosGit/gatos.py
ccarlos@carlos-c:~/personal_workbench/LaboratorioGit$ git add gatosGit/gatos.py
ccarlos@carlos-c:~/personal_workbench/LaboratorioGit$ git commit -m "Cambio del nombre al gato1 desde main"
[main 45e4bd1] Cambio del nombre al gato1 desde main
1 file changed, 1 insertion(+), 1 deletion(-)
ccarlos@carlos-c:~/personal_workbench/LaboratorioGit$ git log
commit 45e4bd17b2bec44210d5b7033421744bcc0e42d2 (HEAD -> main)
Author: CcarlossC <124691128+CcarlossC@users.noreply.github.com>
Date: Thu Feb 9 18:56:59 2023 -0600

    Cambio del nombre al gato1 desde main
```

Figura 9: “Commit” creado para cambiar el nombre al gato1 desde la rama: “main”

1.4. Inclusión de los “commits” de la rama “feature” en la rama “main”

```
ccarlos@carlos-c:~/personal_workbench/LaboratorioGit$ # Hacemos merge de la feature al main
ccarlos@carlos-c:~/personal_workbench/LaboratorioGit$ 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
ccarlos@carlos-c:~/personal_workbench/LaboratorioGit$ git rebase feature
First, rewinding head to replay your work on top of it...
Applying: Cambio del nombre al 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 al 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".
ccarlos@carlos-c:~/personal_workbench/LaboratorioGit$
```

Figura 10: Resultado de hacer “merge” de la rama “feature” hacia la rama “main”

1.5. Conflicto mostrado en el editor texto NANO



```
GNU nano 4.8
import funcionesTarea

gato1 = funcionesTarea.gato()
<<<<<< HEAD
gato1.nombre = "Oreo"
=====
gato1.nombre = "Avena"
>>>>>> Cambio del nombre al gato1 desde main
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 = "Estrella"
gato3.color = "Blanco"
gato3.edad = 6
gato3.aniadirlista()

funcionesTarea.printlista()
```

Figura 11: Conflicto mostrado en el editor texto “NANO”

1.6. Solución del conflicto de forma manual

A screenshot of the GNU nano 4.8 text editor. The editor has a dark purple background with light green text. The code is written in Python and defines three cat objects, each with attributes for name, color, and age, and then adds them to a list. Finally, it prints the list.

```
GNU nano 4.8
import funcionesTarea

gato1 = funcionesTarea.gato()
gato1.nombre = "Avena"
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 = "Estrella"
gato3.color = "Blanco"
gato3.edad = 6
gato3.aniadirlista()

funcionesTarea.printlista()
```

Figura 12: Edición del archivo:gatos.py para solucionar el conflicto que se observa en la figura 11


```
ccarlos@carlos-c:~/personal_workbench/LaboratorioGit$ nano gatosGit/gatos.py
ccarlos@carlos-c:~/personal_workbench/LaboratorioGit$ nano gatosGit/gatos.py
ccarlos@carlos-c:~/personal_workbench/LaboratorioGit$ git add gatosGit/gatos.py
ccarlos@carlos-c:~/personal_workbench/LaboratorioGit$ git rebase --continue
Applying: Cambio del nombre al gato1 desde main
ccarlos@carlos-c:~/personal_workbench/LaboratorioGit$ git log
commit 1fa353e9195bf48a39a12a8c8fc56a7da31bb05e (HEAD -> main)
Author: CcarlossC <124691128+CcarlossC@users.noreply.github.com>
Date: Thu Feb 9 18:56:59 2023 -0600

    Cambio del nombre al gato1 desde main

commit 615807323a9817c87448bab5f7f35f411b2e7af5 (origin/feature, feature)
Author: CcarlossC <124691128+CcarlossC@users.noreply.github.com>
Date: Thu Feb 9 18:18:55 2023 -0600

    Se crea el objeto gato3

commit 4ca823e7e45618cf7883d39558141e253b4837cf
Author: CcarlossC <124691128+CcarlossC@users.noreply.github.com>
Date: Thu Feb 9 18:08:33 2023 -0600

    Cambio del nombre a gato1 desde feature

commit 1640128b5cd29d26384cbfa07ba2ceca7e336d21 (origin/main, origin/HEAD)
Author: marpermon <maria.perezmonge@ucr.ac.cr>
Date: Sat Jan 14 14:17:16 2023 -0600
```

Figura 13: Comandos empleados para terminar de resolver el conflicto de la figura 11, y verificación de que se resolvió correctamente.

1.7. Inclusión de los cambios en el repertorio remoto

```
ccarlos@carlos-c:~/personal_workbench/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
ccarlos@carlos-c:~/personal_workbench/LaboratorioGit$ git fetch origin main
From github.com:CcarlossC/LaboratorioGit
* branch          main          -> FETCH_HEAD
ccarlos@carlos-c:~/personal_workbench/LaboratorioGit$ git rebase origin/main
Current branch main is up to date.
ccarlos@carlos-c:~/personal_workbench/LaboratorioGit$ git push origin main:main
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 3 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (4/4), 432 bytes | 432.00 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:CcarlossC/LaboratorioGit.git
1640128..1fa353e  main -> main
ccarlos@carlos-c:~/personal_workbench/LaboratorioGit$
```

Figura 14: Sincronización entre las dos ramas observadas, así como el comando empleado para subir los cambios de la rama: “main” local a la rama “main” del repositorio remoto

1.8. Empleo del comando: “git log –oneline –graph”

```
ccarlos@carlos-c:~/personal_workbench/LaboratorioGit$ git log --oneline --graph
* 1fa353e (HEAD -> main, origin/main, origin/HEAD) Cambio del nombre al gato1 desde main
* 6158073 (origin/feature, feature) Se crea el objeto gato3
* 4ca823e Cambio del nombre a gato1 desde feature
* 1640128 included gato2 and list printing
* c2b877a corrected mistakes
* d02a6d2 included gato1
* 7ab0adc included functions file
* a13fd08 first commit
```

Figura 15: Resultado de emplear el comando git log –oneline –graph

1.9. Empleo del comando –help para conocer la funcionalidad de los comandos: –graph y –oneline

```
--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 16: Funcionalidad del comando “–graph”

```
--abbrev-commit
Instead of showing the full 40-byte hexadecimal commit object name, show only a partial
prefix. Non default number of digits can be specified with "--abbrev=<n>" (which also
modifies diff output, if it is displayed).

This should make "--pretty=oneline" a whole lot more readable for people using
80-column terminals.

--no-abbrev-commit
Show the full 40-byte hexadecimal commit object name. This negates --abbrev-commit and
those options which imply it such as "--oneline". It also overrides the
log.abbrevCommit variable.

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

Figura 17: Significado del comando “–oneline”

described below (see `git-config(1)`). Here are the details of the built-in formats:

- oneline

<hash> <title line>

This is designed to be as compact as possible.

Figura 18: Razón por la cual fue diseñado el comando “oneline”

1.10. Resultado de ejecutar el archivo: “gatos.py”

```
ccarlos@carlos-c:~/personal_workbench/LaboratorioGit$ python3 gatosGit/gatos.py
Avena, 4 annos, color: negro
Pelusa, 3 annos, color: cafe
Estrella, 6 annos, color: Blnaco
ccarlos@carlos-c:~/personal_workbench/LaboratorioGit$
```

Figura 19: Resultado de ejecutar el archivo: “gatos.py”

1.11. Creación de un “commit” al añadir un PDF con diferentes screenshots al repositorio local

```
ccarlos@carlos-c:~/personal_workbench/LaboratorioGit$ ls
gatosGit README.md screenshots.pdf
ccarlos@carlos-c:~/personal_workbench/LaboratorioGit$ git add screenshots.pdf
ccarlos@carlos-c:~/personal_workbench/LaboratorioGit$ git commit -m "Creacion del PDF con screenshots"
[main 6f90495] Creacion del PDF con screenshots
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 screenshots.pdf
ccarlos@carlos-c:~/personal_workbench/LaboratorioGit$ git log
commit 6f90495562c2d9382df3c3c397e641941f39e776 (HEAD -> main)
Author: CcarlossC <124691128+CcarlossC@users.noreply.github.com>
Date: Fri Feb 10 10:17:50 2023 -0600

    Creacion del PDF con screenshots

commit 1fa353e9195bf48a39a12a8c8fc56a7da31bb05e (origin/main, origin/HEAD)
Author: CcarlossC <124691128+CcarlossC@users.noreply.github.com>
Date: Thu Feb 9 18:56:59 2023 -0600

    Cambio del nombre al gato1 desde main
```

Figura 20: Creación de un “commit” al añadir un PDF con diferentes screenshots al repositorio local

1.12. “Push” de la rama “main” local a la rama “main” del repositorio remoto

```
ccarlos@carlos-c:~/personal_workbench/LaboratorioGit$ git push origin main:main
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 3 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 593.69 KiB | 6.18 MiB/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To github.com:CcarlossC/LaboratorioGit.git
    1fa353e..6f90495  main -> main
ccarlos@carlos-c:~/personal_workbench/LaboratorioGit$
```

Figura 21: “Push” de la rama “main” local a la rama “main” del repositorio remoto

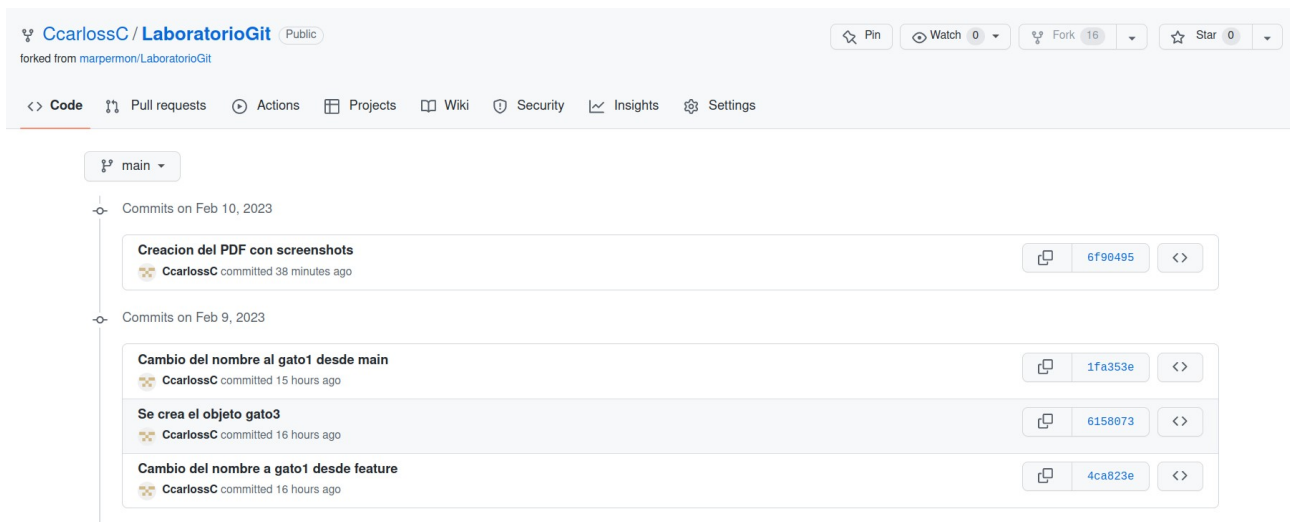


Figura 22: Verificación de que se haya actualizado la rama “main” de forma correcta.