

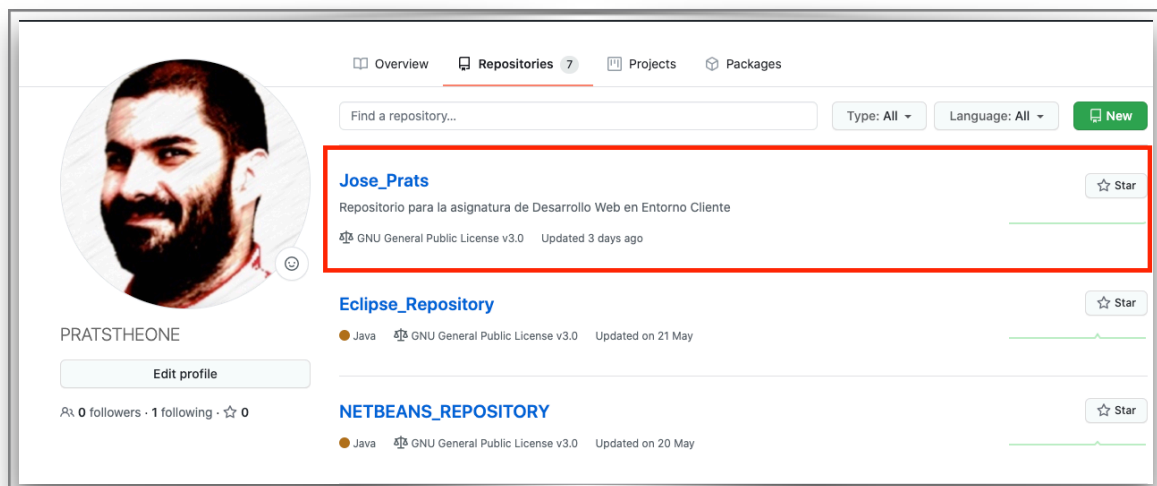
## UD1

# Tutorial de Git - Ejercicios

## Ejercicios

Antes de Empezar:

1 - Crea un repositorio en algún servidor de git (lo más práctico usar un proyecto nuevo en GitHub). Utiliza como nombre del repositorio tu ***primerNombre\_primerApellido***



[https://github.com/PRATSTHEONE/Jose\\_Prats.git](https://github.com/PRATSTHEONE/Jose_Prats.git)

2 - Toma capturas de los comandos introducidos y del repositorio en GitHub que vayan reflejando los cambios y la resolución de los ejercicios.

## Ejercicio 0

Busca en internet información sobre los siguientes comandos:

**git log** y **git status**

Diferencias entre ellos, cómo aplicarlos para todas las ramas, etc.

!!! Te recomiendo que los vayas usando, te darán información valiosa !!!

---

Git log sirve para ver el histórico de commits, estando situados en la carpeta de nuestro proyecto. El listado de commits suele aparecer en orden invertido, queriendo decir que los últimos realizados aparecen como primeros. Ejemplo:

```
commit b2c07b2f6bf6910c3a05aff3c3c2dc8efb096aa5
Author: Miguel Angel Alvarez <malvarez@desarrolloweb.com>
Date:   Fri Feb 24 19:53:11 2017 -0300

    Cambios última clase

commit fa42f3ba4dc9b0112dd1e302f4a33b73be64e539
Author: Miguel Angel Alvarez <malvarez@desarrolloweb.com>
Date:   Fri Feb 17 18:11:19 2017 -0200

    clase 2 de node

commit 35446abb25913d0bb383e9759a405de88fc46409
Author: Miguel Angel Alvarez <malvarez@desarrolloweb.com>
Date:   Fri Feb 17 16:00:58 2017 -0200

    componentes antes de clase 2 nodejs

commit bda125319b4df54bc9fbf6596c83bcd5d0fe9a03
Author: Miguel Angel Alvarez <malvarez@desarrolloweb.com>
Date:   Mon Feb 13 18:23:03 2017 -0200

    Clase 2 de polymerfire
```

Si el repositorio tiene ramas (branch) y queremos que el log muestre más información de las ramas existentes, sus uniones (merges) y cosas así, podemos hacer uso de la opción `--graph`.

`git log --graph --oneline`

```
* e181272 Esto es un merge con mensaje
|\
| * 071d7ee commit en experimental para test
| * bc3b0c2 Yeah! git rules!
| * 27d0926 Merge branch 'experimental'
| \
|  \
|   |
|   | * a7c67c8 commit experimental
|   | * cb8877e Segundo cambio experimental
|   | * e2a99f8 cambio experimental
|   | * 03bbd72 cambio importante en master
|   |
|   |
|   | * 51aae71 primer commit
```

Cómo se puede observar en la captura anterior tras introducir el comando descrito, se muestran los commit en una línea y las ramas en las que estábamos con sus diferentes operaciones.

Por otro lado, git status muestra la lista de los archivos que se han cambiado junto con los archivos que están por ser añadidos o comprometidos. En otras palabras: muestra el estado del repositorio. Ejemplo:

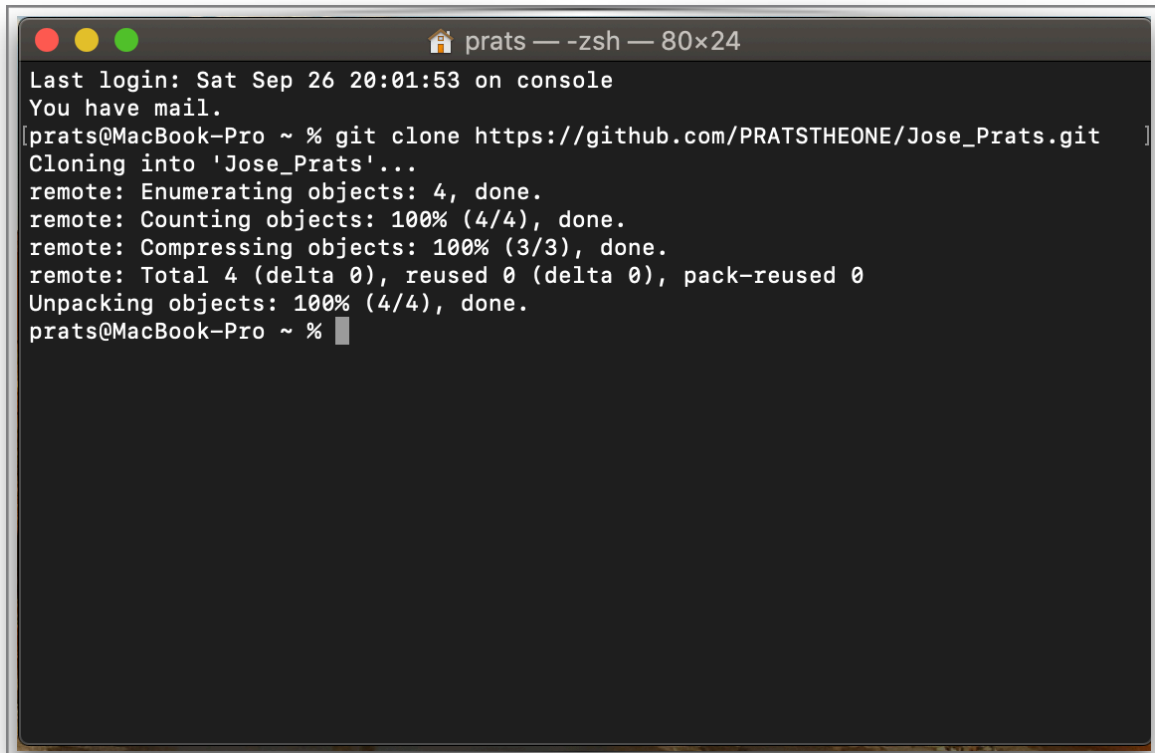
```
git-test[master]$ git status
On branch master
nothing to commit, working tree clean
git-test[master]$ git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

        modified:   test.txt

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

## Ejercicio 1

- Haz un clonado en local de tu proyecto/repositorio.

A terminal window titled 'prats — -zsh — 80x24' with standard macOS window controls (red, yellow, green buttons). The terminal output shows the execution of a git clone command. It starts with 'Last login: Sat Sep 26 20:01:53 on console' and 'You have mail.' followed by the command '[prats@MacBook-Pro ~ % git clone https://github.com/PRATSTHEONE/Jose\_Prats.git ]'. The output of the command is: 'Cloning into 'Jose\_Prats'...', 'remote: Enumerating objects: 4, done.', 'remote: Counting objects: 100% (4/4), done.', 'remote: Compressing objects: 100% (3/3), done.', 'remote: Total 4 (delta 0), reused 0 (delta 0), pack-reused 0', 'Unpacking objects: 100% (4/4), done.', and finally 'prats@MacBook-Pro ~ %' with a cursor. The window has a dark background and light-colored text.

```
prats — -zsh — 80x24
Last login: Sat Sep 26 20:01:53 on console
You have mail.
[prats@MacBook-Pro ~ % git clone https://github.com/PRATSTHEONE/Jose_Prats.git ]
Cloning into 'Jose_Prats'...
remote: Enumerating objects: 4, done.
remote: Counting objects: 100% (4/4), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 4 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (4/4), done.
prats@MacBook-Pro ~ %
```

- Mostrar el historial de cambios del repositorio. Se supone que debemos estar en la rama "master".

Creo un fichero llamado primer fichero.txt

The screenshot shows a GitHub repository interface. At the top, it indicates the current branch is 'master', there is 1 branch, and 0 tags. Below this, a commit history table is displayed:

File	Commit Message	Time
LICENSE	Initial commit	3 days ago
README.md	Initial commit	3 days ago
primer fichero.txt	Create primer fichero.txt	now

Below the table, the content of the README.md file is shown, which reads: **Jose\_Prats**.

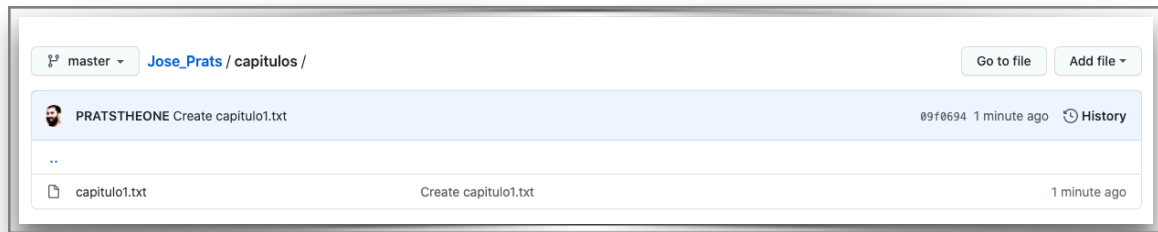
Luego, usé git log y...

The screenshot shows a terminal window titled 'Jose\_Prats — -zsh — 80x24'. The user has executed the command 'git log', and the output is as follows:

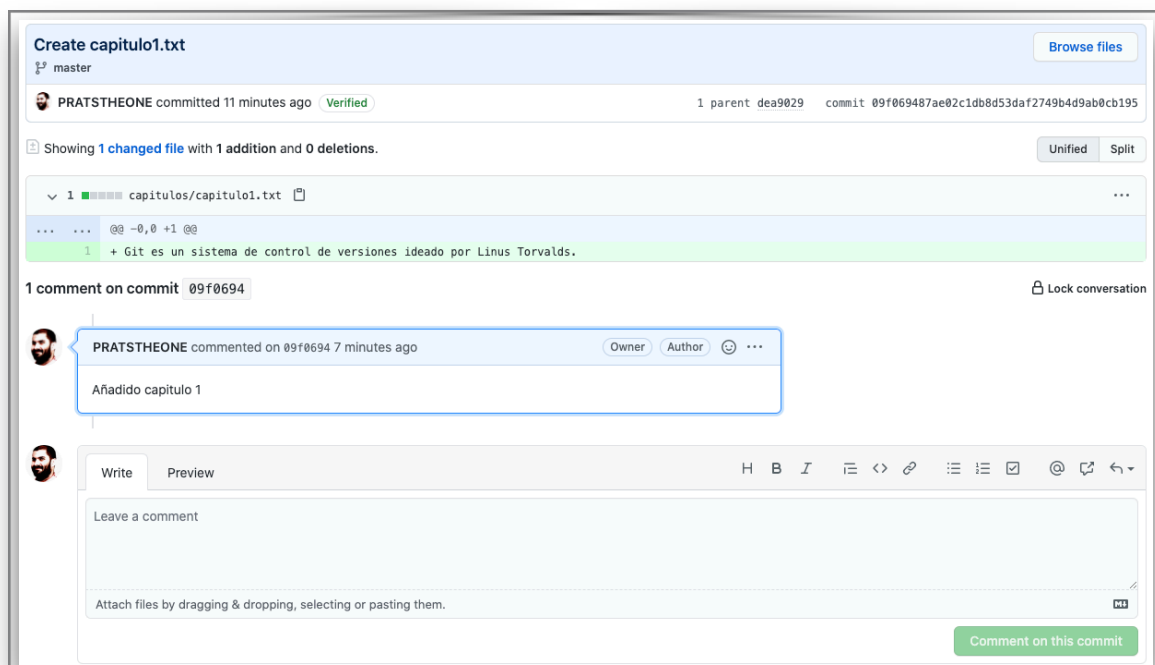
```
[prats@MacBook-Pro jose_prats % git log
commit 37fe62ff9448e265e17c6953e28abcb115740108 (HEAD -> master, origin/master,
origin/HEAD)
Author: PRATSTHEONE <55916482+PRATSTHEONE@users.noreply.github.com>
Date:   Wed Sep 23 13:25:15 2020 +0200

    Initial commit
prats@MacBook-Pro jose_prats %
```

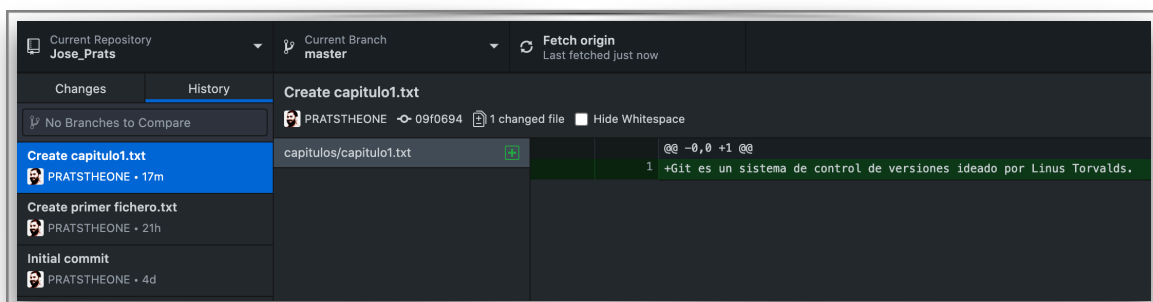
- Crear una carpeta "capítulos" y crear dentro de ella el fichero capitulo1.txt con el siguiente texto: "Git es un sistema de control de versiones ideado por Linus Torvalds."



- Añade el nuevo archivo a git, lo que se suele definir como [añadir el archivo a la zona “temporal”](#).
- Hacer un commit de los cambios con el mensaje “Añadido capítulo 1.”



- Volver a mostrar el historial de cambios del repositorio.



## Ejercicio 2

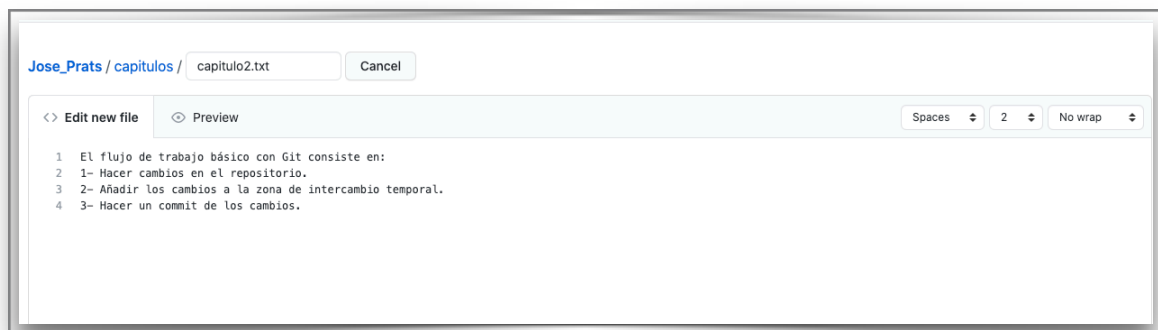
- Crear el fichero capítulo 2.txt en la carpeta capítulos con el siguiente texto:

*El flujo de trabajo básico con Git consiste en:*

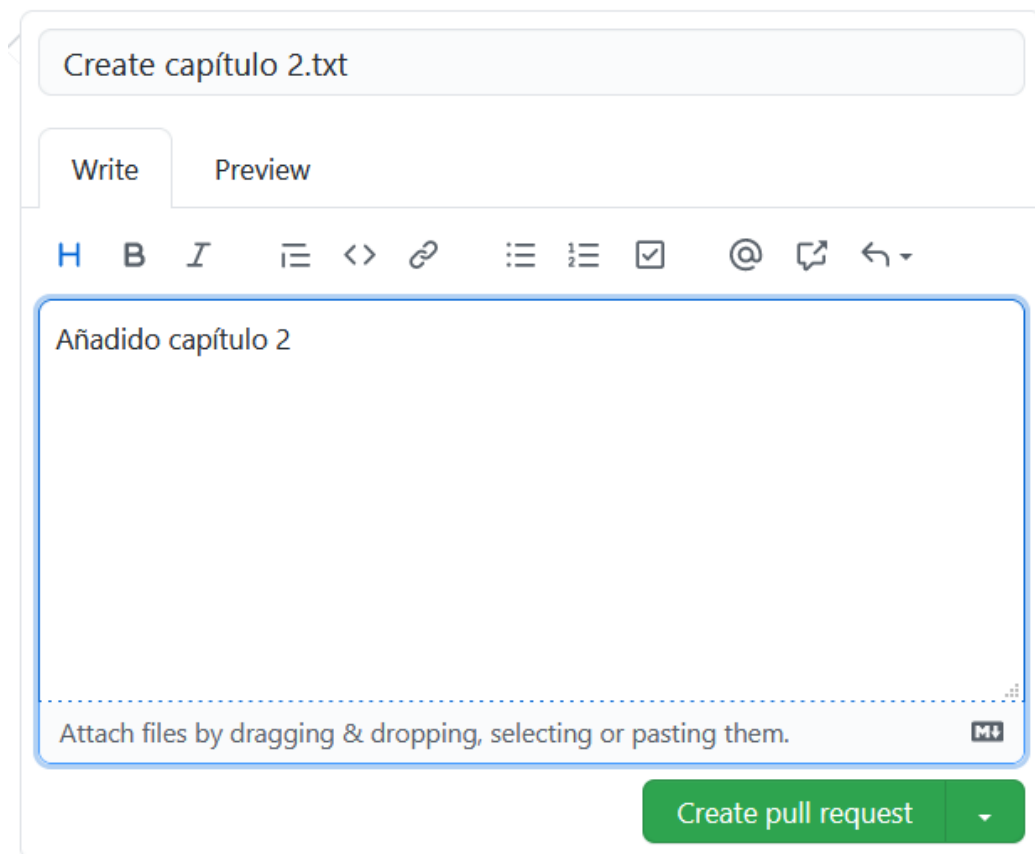
*1- Hacer cambios en el repositorio.*

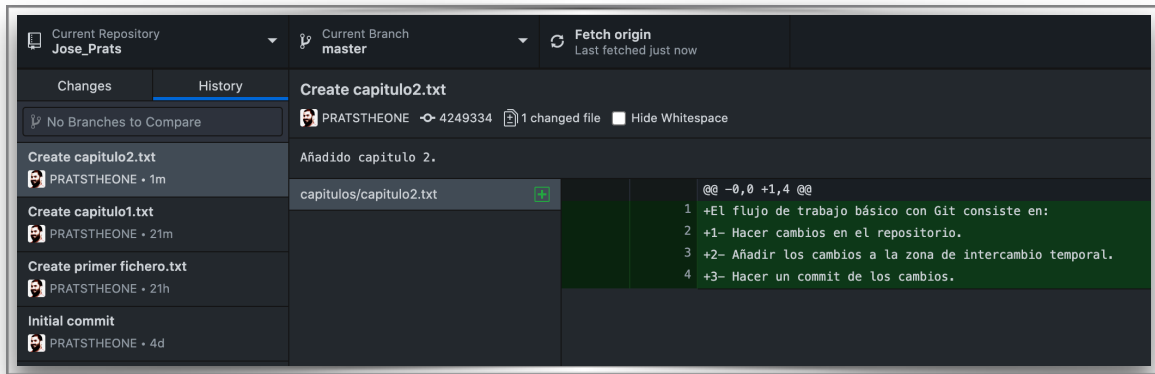
*2- Añadir los cambios a la zona de intercambio temporal.*

*3- Hacer un commit de los cambios.*



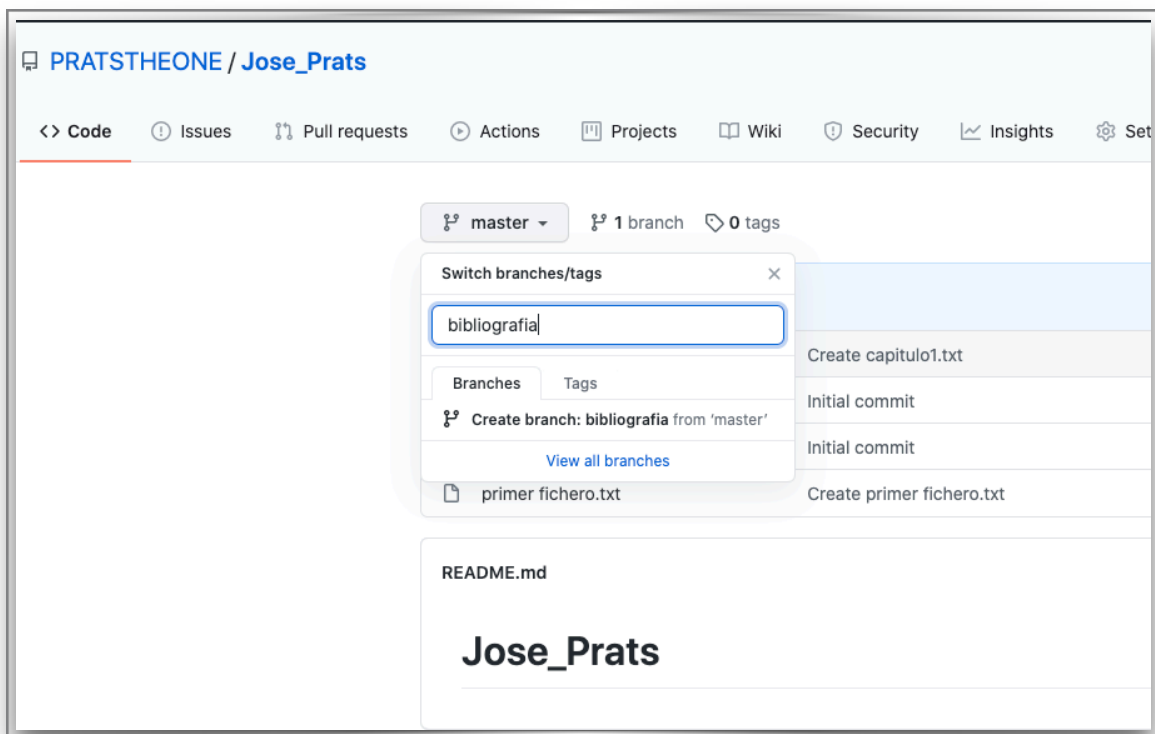
- Añadir los cambios a la zona de intercambio temporal.
- Hacer un commit de los cambios con el mensaje “Añadido capítulo 2.”

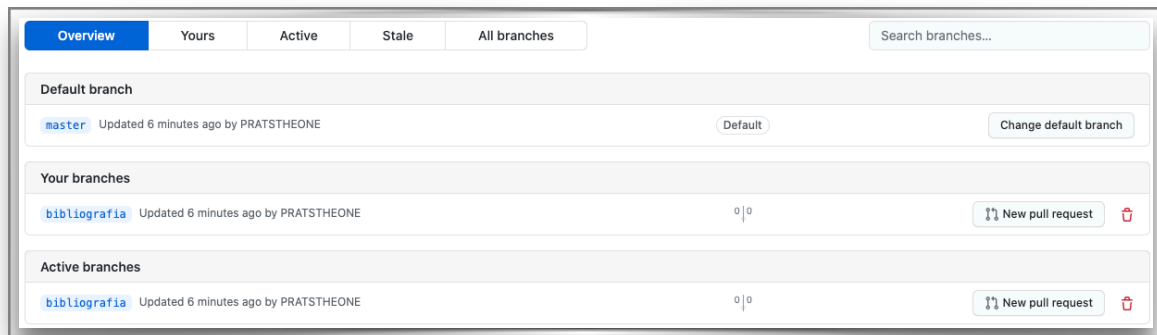




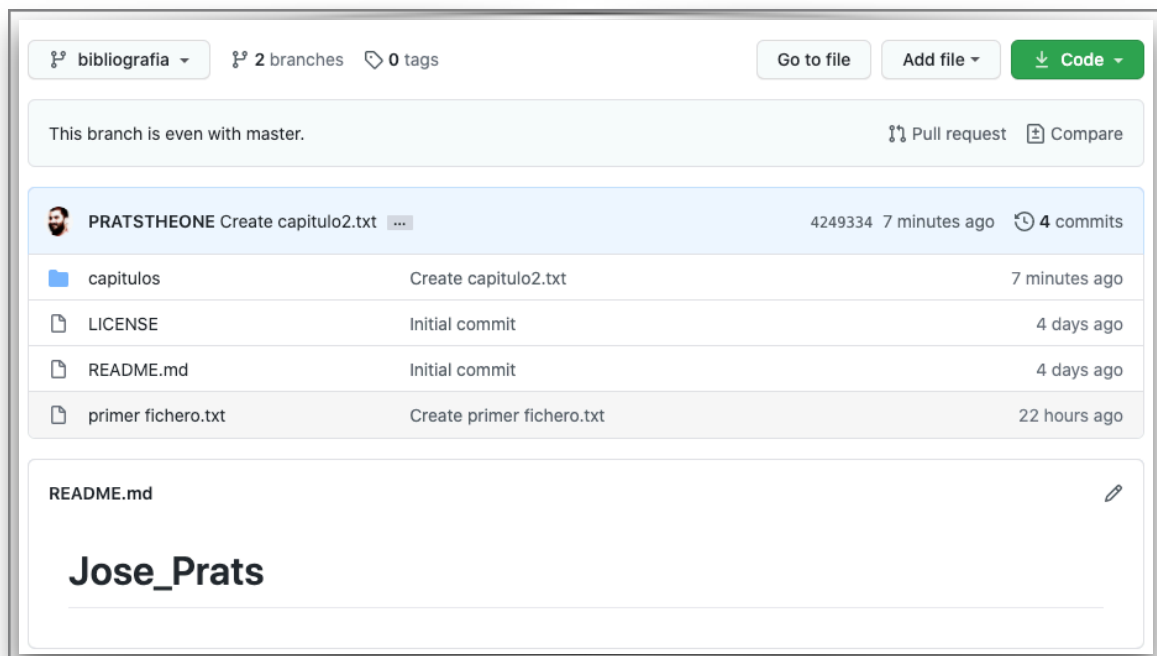
### Ejercicio 3

- Crear una nueva rama bibliografía y mostrar las ramas del repositorio.

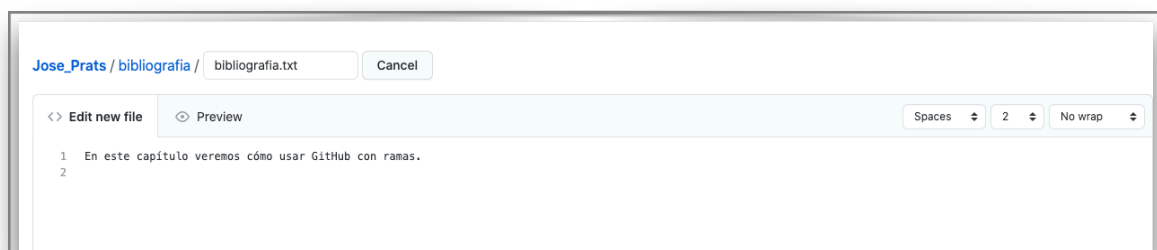




- Posicionate en la nueva rama creada.
- Muestra que ahora si estás en la nueva rama creada.

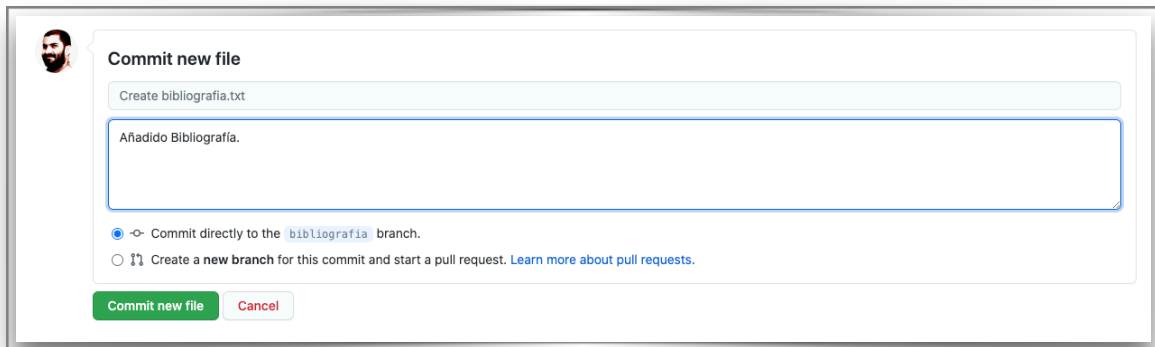


- Crear el fichero bibliografia/bibliografia.txt y añadir el texto siguiente: *En este capítulo veremos cómo usar GitHub con ramas.*





- Añadir los cambios a la zona de intercambio temporal.
- Hacer un commit con el mensaje “Añadido Bibliografía.”



**Commit new file**

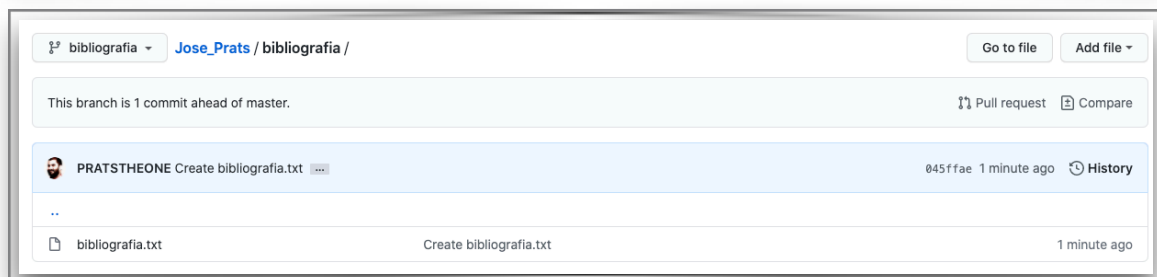
Create bibliografia.txt

Añadido Bibliografía.

☒ Commit directly to the `bibliografia` branch.

☐ Create a new branch for this commit and start a pull request. [Learn more about pull requests.](#)

**Commit new file** **Cancel**



`bibliografia` **Jose\_Prats** / bibliografia /

Go to file Add file

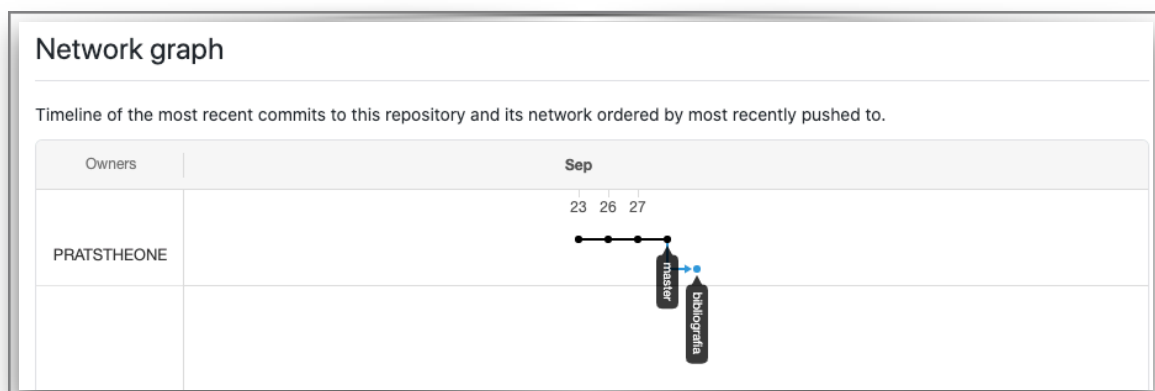
This branch is 1 commit ahead of master. Pull request Compare

**PRATSTHEONE** Create bibliografia.txt 045ffae 1 minute ago History

..

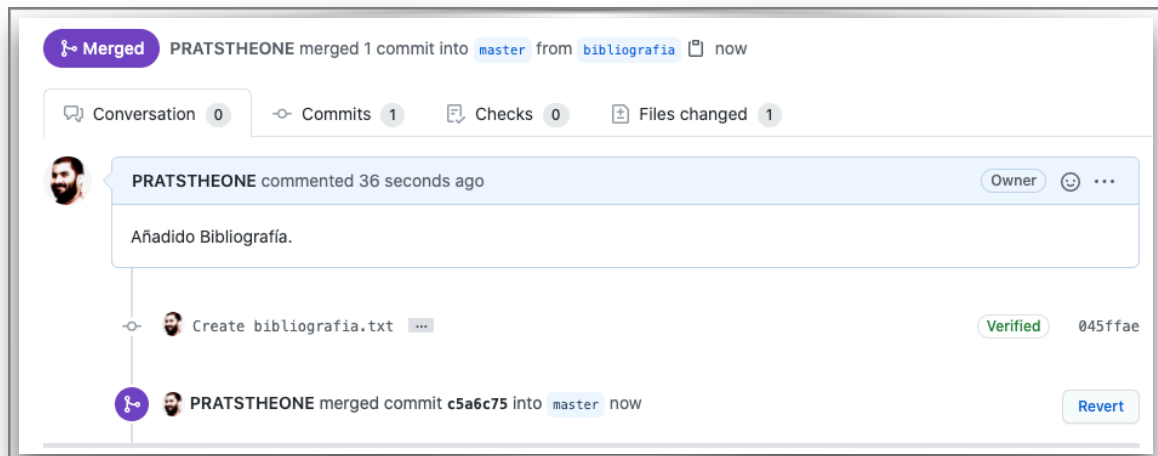
`bibliografia.txt` Create bibliografia.txt 1 minute ago

- Mostrar la historia del repositorio incluyendo todas las ramas

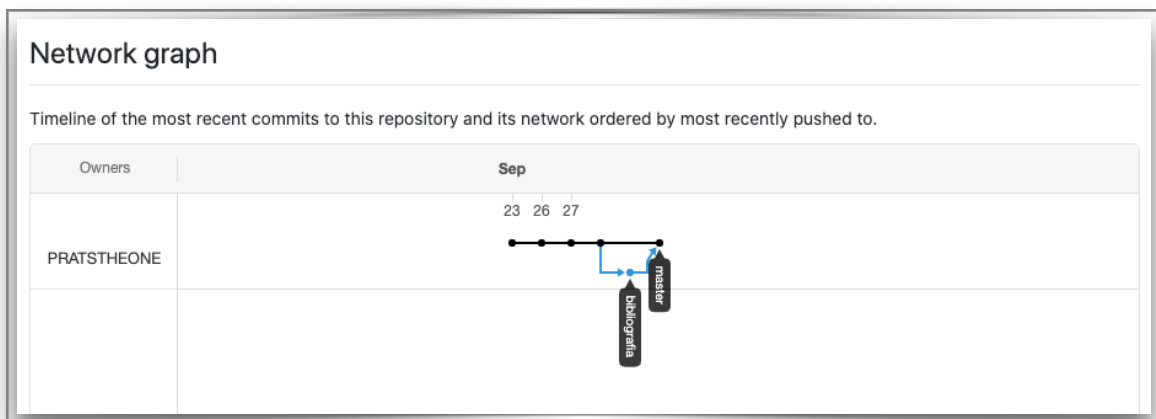


## Ejercicio 4

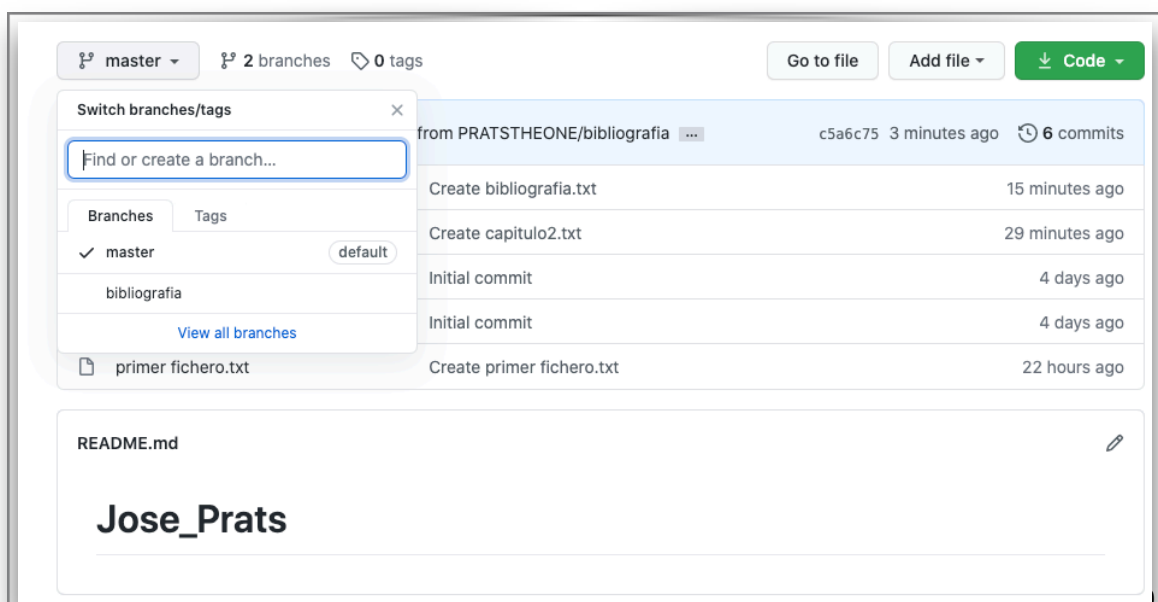
- Fusionar la rama bibliografía con la rama master.



- Mostrar la historia del repositorio incluyendo todas las ramas.

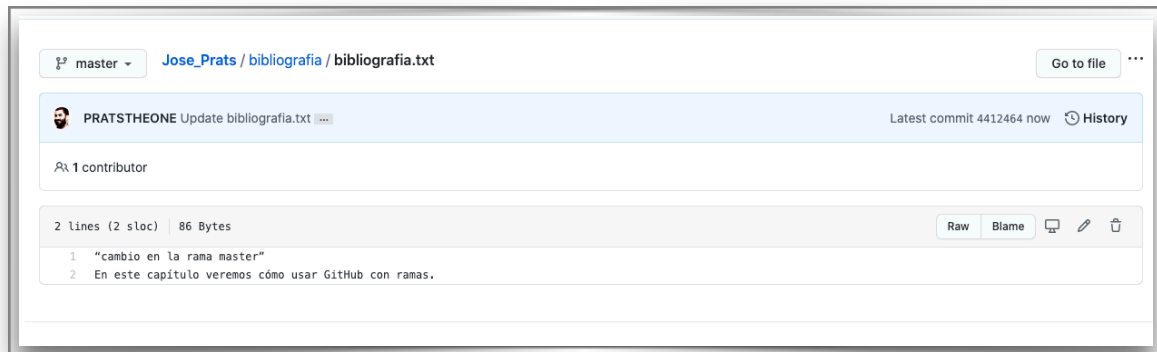


- Mostrar de nuevo la historia del repositorio incluyendo todas las ramas.
- Sube todos los cambios realizados en este ejercicio a tu repositorio.

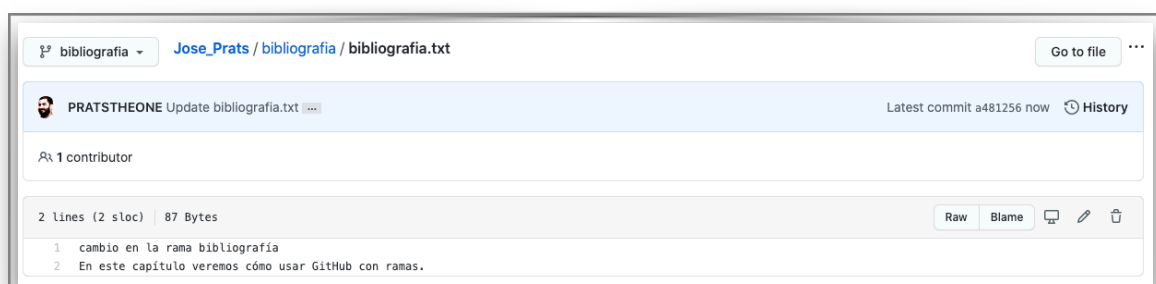


## Ejercicio 5

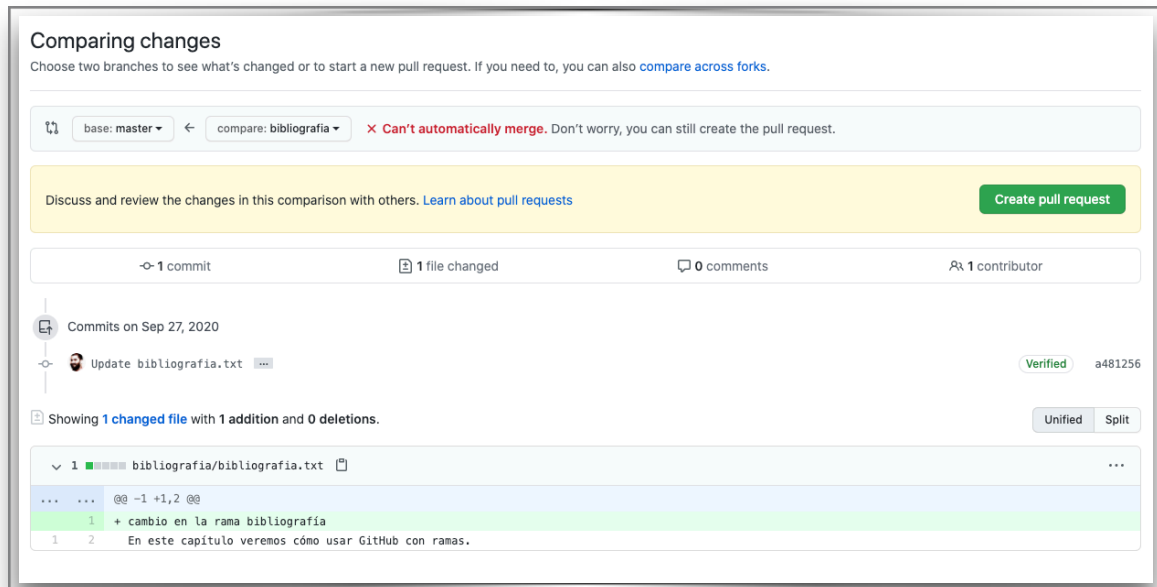
- Posicionate en la rama master, y en el archivo “bibliografía.txt” haz un cambio en la primera línea del fichero (Por ejemplo, “cambio en la rama master”) y guardalo. Comitea los cambios.



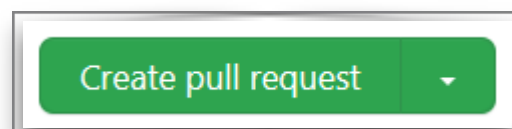
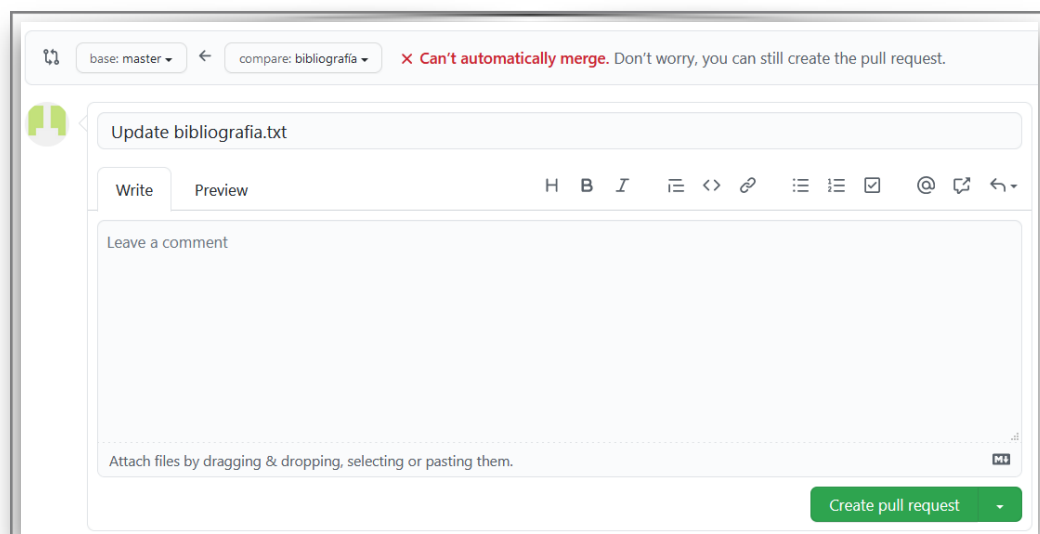
- Posicionate en la rama bibliografía, y en el archivo “bibliografía.txt” haz un cambio en la primera línea del fichero (Por ejemplo, “cambio en la rama bibliografía”)y guardalo. Comitea los cambios.

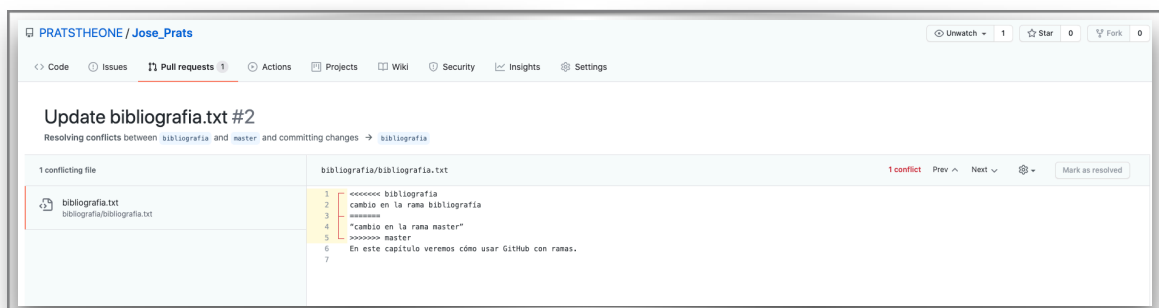
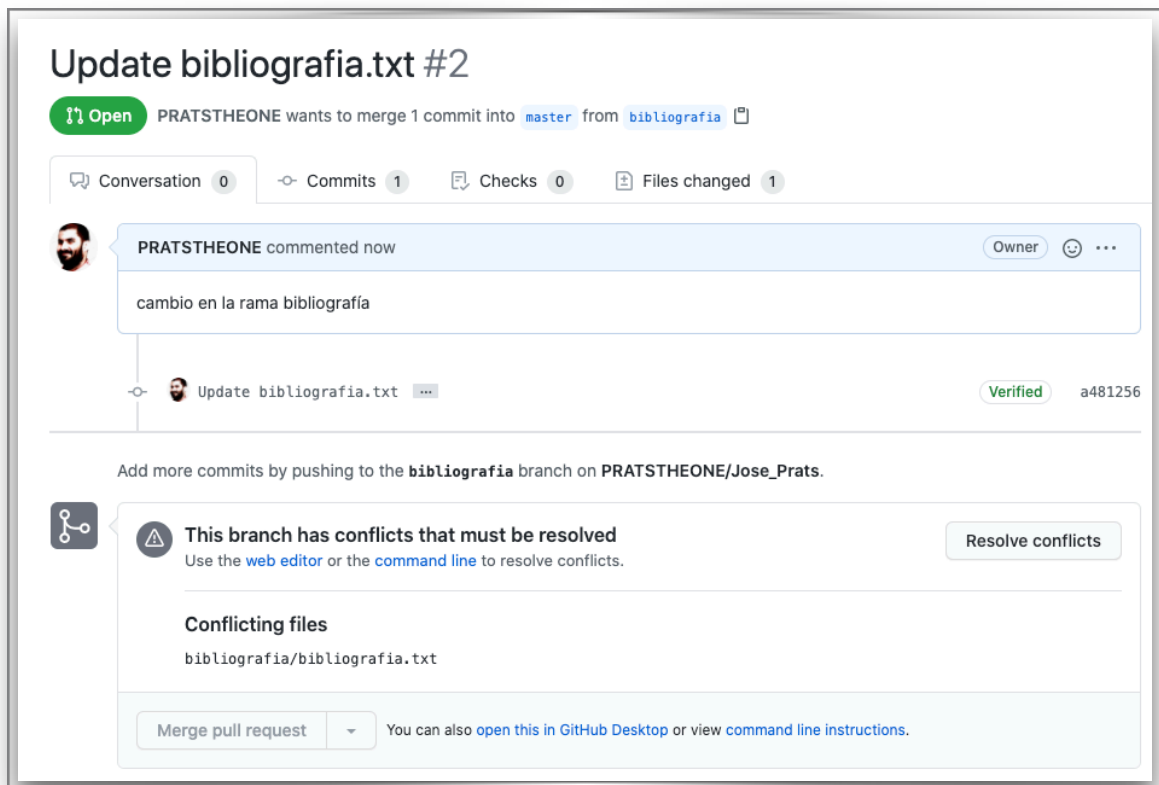


- Ahora vamos a intentar fusionar la rama “bibliografía” dentro de “máster”. Como verás, nos va a dar un conflicto.



Trata de buscar información por internet de como resolver un conflicto en la fusión de ramas, resuélvelo, explica cómo lo has hecho, fusiona la rama y súbelo a tu repositorio Git. ¿Has visto como Git ha transformado el archivo bibliografía.txt cuando has intentado hacer la fusión y ha habido un conflicto?





Mark as resolved

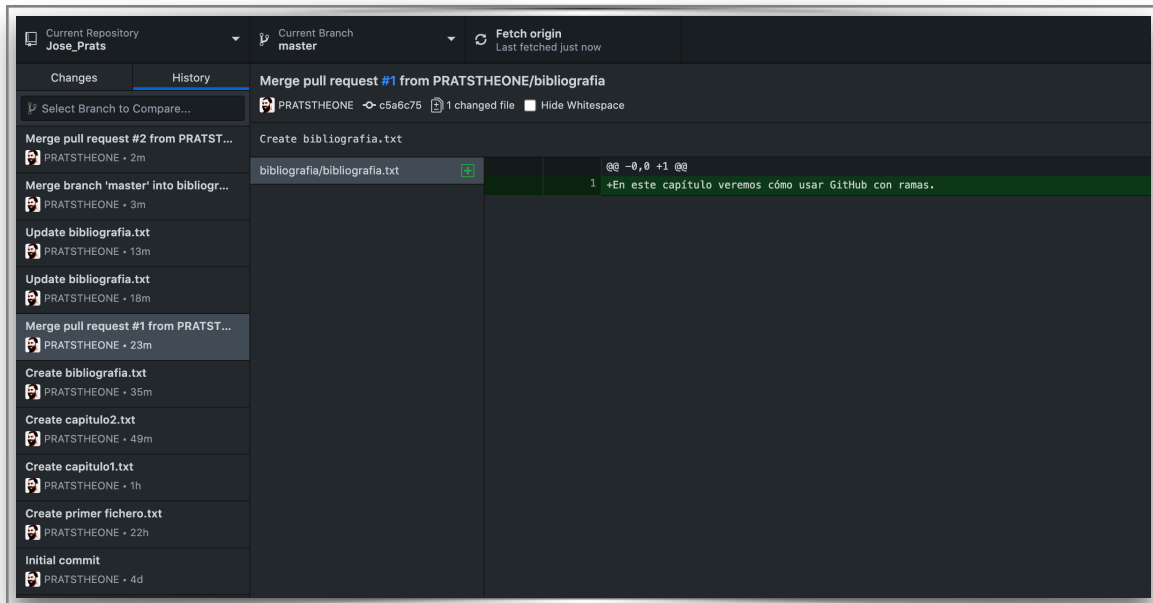
Commit merge

✓ Resolved

- Una vez solucionado el conflicto, sube los cambios a GitHub.

The screenshot displays a GitHub pull request interface. At the top, a purple banner indicates the pull request has been merged. Below this, a summary shows that PRATSTHEONE merged 2 commits into the master branch from the bibliografia branch 34 seconds ago. The interface includes tabs for Conversation (1), Commits (1), Checks (0), and Files changed (1). The commit history shows a comment from PRATSTHEONE 7 minutes ago about a change in the bibliografia branch, followed by an update to bibliografia.txt (Verified, a481256). Another comment from PRATSTHEONE 3 minutes ago mentions resolving a version problem in the bibliografia files when merging branches. This is followed by a merge commit (Verified, c4bd3f6) and a final comment from PRATSTHEONE stating the resolution of the version problem. A system message confirms the pull request was successfully merged and closed, advising that the bibliografia branch can be safely deleted. At the bottom, there is a comment input area with a 'Comment' button.

Confirm merge



Enlace Git: [https://github.com/PRATSTHEONE/Jose\\_Prats.git](https://github.com/PRATSTHEONE/Jose_Prats.git)