

# GitHub Comandos Avanzados

Bryan Ivan Montiel Ortega

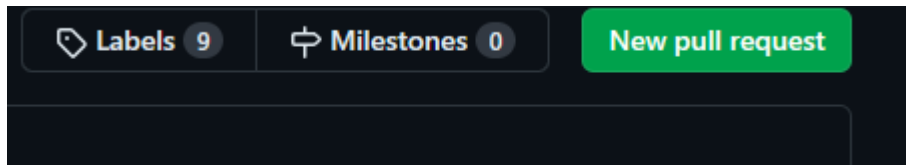
Octubre 2023

# Índice

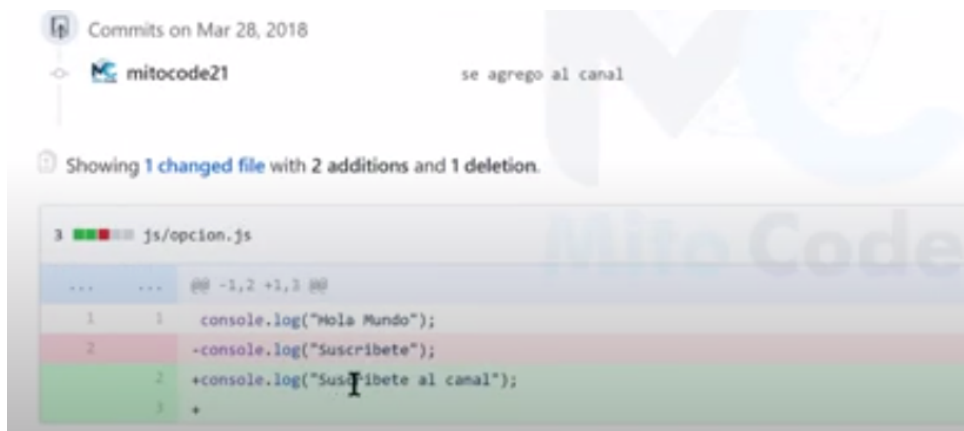
Índice.....	1
Pull Request.....	2
Fork.....	4
Rebase.....	6
Bibliografía.....	8

# Pull Request

Para hacer un 'Pull Request' seleccionas en la pestaña de 'pull request' en el repositorio copia en tu cuenta y posteriormente está la opción como se ve a continuación:



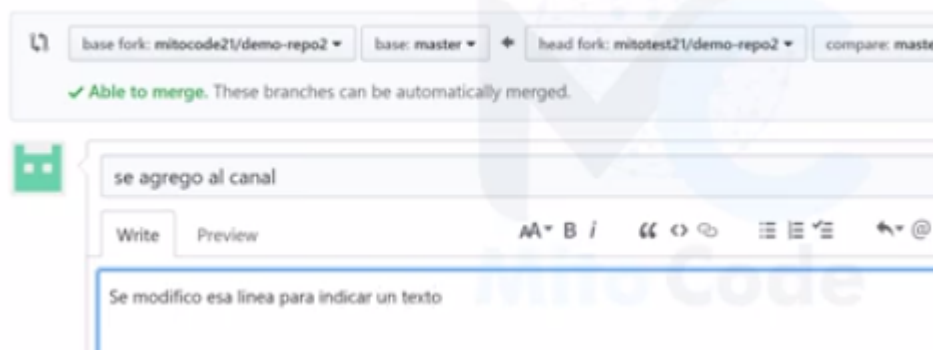
Posteriormente se puede ver una ventana resumen con los cambios hechos respecto a la versión original.



Después de ello se abriría otra ventana donde se pedirá añadir un asunto de la 'pull request'

## Open a pull request

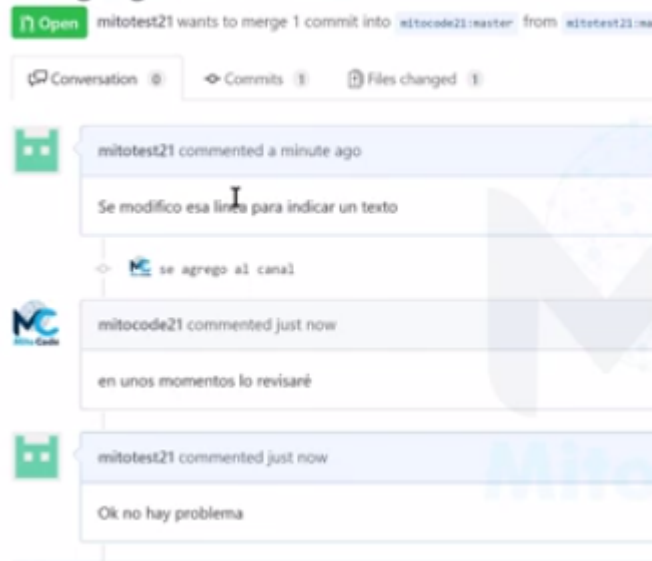
Create a new pull request by comparing changes across two branches. If you need to, you can also [compare across forks](#).



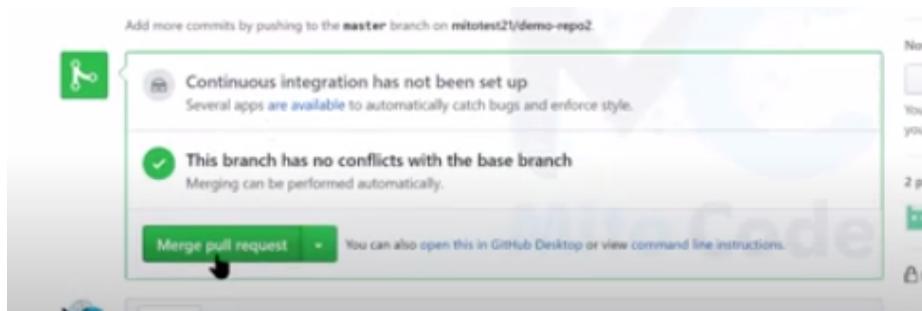
Manda un mensaje si no hay problemas con la rama principal.

Desde el otro lado o el repositorio original llegará una notificación con el 'pull request'. Y podría notificarse por mensaje o hacer alguna cola de mensajes para estar en contacto con el equipo de trabajo.

## se agrego al canal #2



Una vez que el usuario del repositorio original haga sus revisiones y decida juntar los cambios se puede hacer el merge como se ve en la imagen

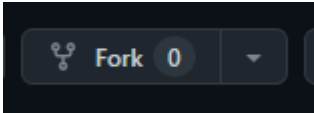


De esta manera es posible trabajar colaborativamente con un equipo de trabajo, hacer revisiones y comentarios mientras se trabaja en distintos aspectos de un proyecto.

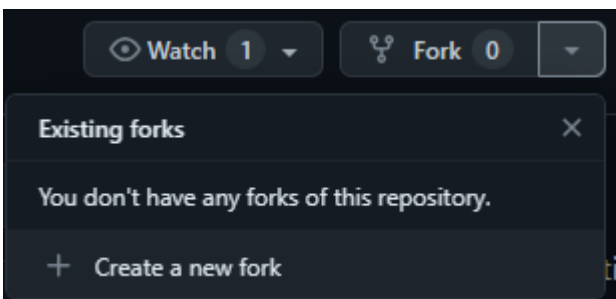
# Fork

Cuando no aparece interesante un repositorio y quisiéramos tener una copia para poder hacer nuestros cambios o una propia adaptación del repositorio.

Dentro de un repositorio en GitHub en el extremo superior derecho de un repositorio puedes ver la siguiente opción.

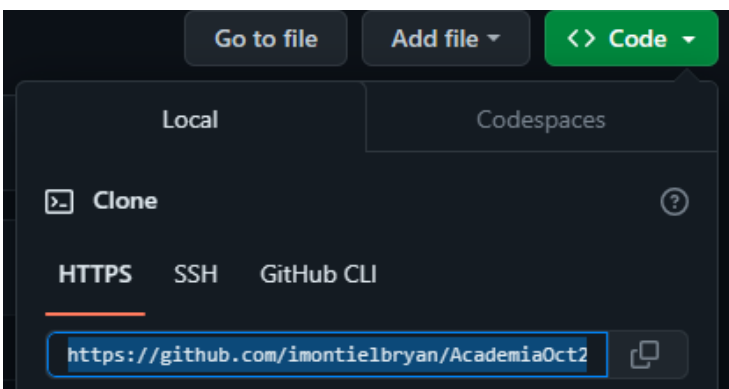


Y al abrir el menú desplegable puedes copiarlo en tu cuenta.



Si estás haciendo cambios estos no se verán en el repositorio original sino en tu copia en tu cuenta de GitHub.

Para poder comenzar a hacer cambios localmente puedes crear una carpeta donde comenzarás a trabajar con la copia local en tu equipo y hacer un 'clone' de la copia de tu cuenta. Primero debes obtener la dirección desde el repositorio



Cuando hayas ubicado en tu sistema la dirección de la carpeta donde quieres alojar tu versión local puedes teclear 'git clone' junto a la dirección copiada como se ve a continuación:

```
mitocode@mitocode: ~/git/folder4
mitocode@mitocode:~/git/folder4$ git clone https://github.com/mitocode21/angular-tour-of-heroes.git
Cloning into 'angular-tour-of-heroes'...
remote: Counting objects: 1579, done.
remote: Total 1579 (delta 0), reused 0 (delta 0), pack-reused 1579
Receiving objects: 100% (1579/1579), 268.83 KiB | 131.00 KiB/s, done.
Resolving deltas: 100% (972/972), done.
Checking connectivity... done.
```

Al hacer un cambio en una carpeta específica de nuestra versión local la podemos subir en nuestra copia del repositorio, recordando que después del 'commit' se hace por medio del 'push' origin y el nombre de la rama principal (en el ejemplo de la imagen es 'master' pero actualmente puede aparecer más común con 'main') y por último introducir la cuenta y clave de la cuenta.

```
mitocode@mitocode:~/git/folder4$ cd angular-tour-of-heroes/
mitocode@mitocode:~/git/folder4/angular-tour-of-heroes$ git add .
mitocode@mitocode:~/git/folder4/angular-tour-of-heroes$ git commit -m "se elimin
o algo en dashboard"
[master 36195b4] se elimino algo en dashboard
1 file changed, 1 insertion(+), 1 deletion(-)
mitocode@mitocode:~/git/folder4/angular-tour-of-heroes$ git push origin master
Username for 'https://github.com': mitocode21@gmail.com
Password for 'https://mitocode21@gmail.com@github.com':
Counting objects: 5, done.
Compressing objects: 100% (5/5), done.
Writing objects: 100% (5/5), 429 bytes | 0 bytes/s, done.
Total 5 (delta 4), reused 0 (delta 0)
remote: Resolving deltas: 100% (4/4), completed with 4 local objects.
```

Los cambios se verán en GitHub. Los cambios de tu copia de GitHub ya los puedes hacer desde tu equipo localmente.

# Rebase

Te trae los cambios de la rama principal hacia tu rama de desarrollo y lo reorganiza dentro de tu código.

Es una manera de juntar una rama hacia la rama principal. Aunque puede tomarse de precaución que el 'comando' puede reescribir la historia de los commits que tienes en el repositorio, por lo que debe revisarse dos veces si se está haciendo con cuidado.

Pensando que se tiene una rama principal 'master' y una rama 'desarrollo' se haría lo siguiente. Primero por medio del comando 'checkout' se traslada hacia la rama desarrollo y dentro se hacen dos commits.

```
lucho@DESKTOP-8LMTQUM MINGW64 ~/Mis documentos/Tutoriales/cursohtml (desarrollo)
$ touch archivo1.txt && git add archivo1.txt && git commit -m "archivo1.txt"
[desarrollo 990633c] archivo1.txt
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 archivo1.txt

lucho@DESKTOP-8LMTQUM MINGW64 ~/Mis documentos/Tutoriales/cursohtml (desarrollo)
$ touch archivo2.txt && git add archivo2.txt && git commit -m "archivo2.txt"
[desarrollo da7bfe9] archivo2.txt
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 archivo2.txt
```

Y después de esto, suponiendo un proyecto colaborativo, se hace un cambio en la rama principal o 'master', para hacerlo en tu cuenta primero puedes dirigirte hacia la rama 'master' y hacer los cambios.

```
lucho@DESKTOP-8LMTQUM MINGW64 ~/Mis documentos/Tutoriales/cursohtml (desarrollo)
$ git checkout master
Switched to branch 'master'
Your branch is up to date with 'origin/master'.

lucho@DESKTOP-8LMTQUM MINGW64 ~/Mis documentos/Tutoriales/cursohtml (master)
$ touch archivo3.txt && git add archivo3.txt && git commit -m "archivo3.txt"
```

Después, nuevamente suponiendo algo colaborativo, en la rama 'desarrollo' puede existir un cambio posterior al hecho en 'master'.

```
lucho@DESKTOP-8LMTQUM MINGW64 ~/Mis documentos/Tutoriales/cursohtml (desarrollo)
$ touch archivo4.txt && git add archivo4.txt && git commit -m "archivo4.txt"
```

En la siguiente imagen se hace la unión de la rama 'desarrollo' con la rama 'master'. Para ello se usa el comando de git

```
git rebase master
```

```
lucho@DESKTOP-8LMTQUM MINGW64 ~/Mis documentos/Tutoriales/cursohtml (desarrollo)
$ git rebase master
Successfully rebased and updated refs/heads/desarrollo.
```

Al agregarlo es posible volver a hacer un log de los cambios y el cambio en la rama 'master' se verá reflejado (en este caso se puede ver el cambio de 'archivo3.txt')

```
$ git log --oneline --graph -10
* 5a2257d (HEAD -> desarrollo) archivo4.txt
* 988bae3 archivo2.txt
* 24d0576 archivo1.txt
* 245ee57 (master) archivo3.txt
* 9d688c1 (origin/master, origin/HEAD) Agregado el a
```



# Bibliografía

1. DECODE\_Programacion. (2021, mayo 28). ⚠️ ¿CÓMO HACER un REBASE EN GIT? [2022] ⚡  
CURSO COMPLETO DE GIT y GITHUB. YouTube.  
[https://youtu.be/AfmszlBoOrQ?si=\\_NeqlvUnpMexfn-H](https://youtu.be/AfmszlBoOrQ?si=_NeqlvUnpMexfn-H)
2. MitoCode. (2018, marzo 24). Curso de Git y Github - 13 fork. YouTube.  
[https://youtu.be/9YUaf-uxuRM?si=5NHeNogEZow\\_Hdpe](https://youtu.be/9YUaf-uxuRM?si=5NHeNogEZow_Hdpe)
3. MitoCode. (2018, marzo 31). Curso de Git y Github - 14 pull request. YouTube.  
<https://youtu.be/8yXR1MJx9kM?si=nExUVXvBAdUB88Xe>