

## RESOLVIENDO CONFLICTOS EN GITHUB

1. HACEMOS GIT PUSH Y ENCONTRAMOS QUE EL REPOSITORIO REMOTO TIENE CAMBIOS QUE NO TIENE EL REPOSITORIO LOCAL PERO NO HAY CONFLICTO DE RAMAS

Siempre hacer `git pull origin <nombre-rama-remota>` antes de empezar a trabajar para actualizar el repo local con el remoto.

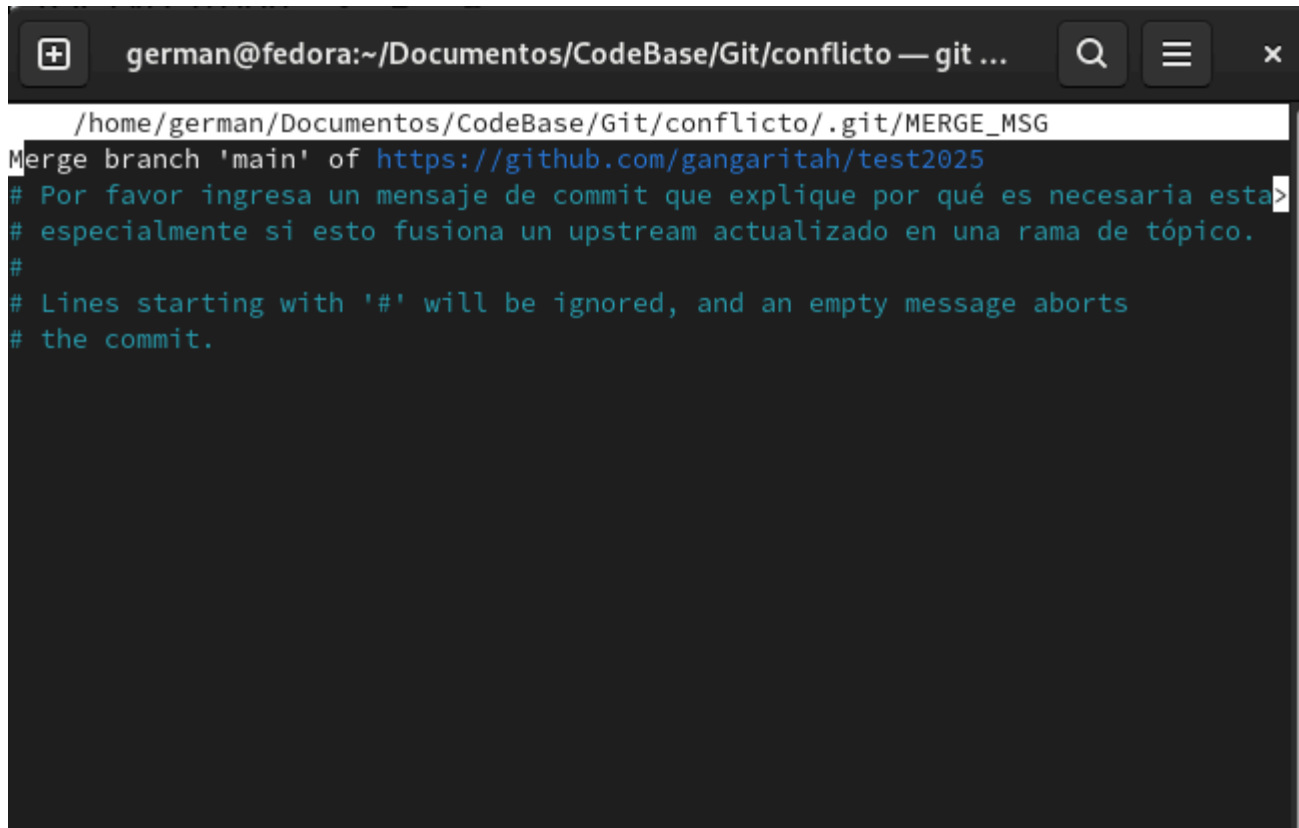
CASO: Si hacemos por ejemplo `git pull origin main` y nuestro repo remoto tiene cambios que no tenemos en el repo local, ya sea porque nosotros hayamos hecho esos cambios o uno de nuestros compañeros si la rama es compartida, obtendremos un mensaje como este

```
To https://github.com/gangaritah/test2025
! [rejected]        main -> main (fetch first)
error: falló el empuje de algunas referencias a 'https://github.com/gangaritah/test2025'
hint: Updates were rejected because the remote contains work that you do not
hint: have locally. This is usually caused by another repository pushing to
hint: the same ref. If you want to integrate the remote changes, use
hint: 'git pull' before pushing again.
hint: See the 'Note about fast-forwards' in 'git push --help' for details.
```

en este caso debemos hacer lo siguiente:

hacemos `git pull origin main` para actualizar nuestro repo local. Debemos tener activado **`git config --global pull.rebase false`** para que cuando se haga `git pull origin main` el código remoto se fusione(merge) con el local.

En este caso se abre el editor para confirmar la fusión del código de nuestro repositorio remoto con el local



```
german@fedora:~/Documentos/CodeBase/Git/conflicto — git ...  
/home/german/Documentos/CodeBase/Git/conflicto/.git/MERGE_MSG  
Merge branch 'main' of https://github.com/gangaritah/test2025  
# Por favor ingresa un mensaje de commit que explique por qué es necesaria esta  
# especialmente si esto fusiona un upstream actualizado en una rama de tópico.  
#  
# Lines starting with '#' will be ignored, and an empty message aborts  
# the commit.
```

Guardamos y veremos un mensaje que nos indica que la fusión ha sido exitosa

```
Desde https://github.com/gangaritah/test2025  
* branch      main      -> FETCH_HEAD  
Merge made by the 'ort' strategy.  
 README.md | 1 +  
 1 file changed, 1 insertion(+)
```

Ahora nuestro código local está emparejado con el remoto, es hora de volver a intentar hacer push.

Hacemos git push origin main y veremos un mensaje como este

```
Enumerando objetos: 9, listo.  
Contando objetos: 100% (8/8), listo.  
Compresión delta usando hasta 8 hilos  
Comprimiendo objetos: 100% (5/5), listo.  
Escribiendo objetos: 100% (5/5), 596 bytes | 596.00 KiB/s, listo.  
Total 5 (delta 2), reused 0 (delta 0), pack-reused 0 (from 0)  
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.  
To https://github.com/gangaritah/test2025  
 837f906..09c2fe1  main -> main
```

esto nos indica que nuestros cambios locales ahora se han fusionado en el repositorio remoto

## 2. HACEMOS GIT PUSH Y ENCONTRAMOS QUE EL REPOSITORIO REMOTO TIENE CAMBIOS QUE NO TIENE EL REPOSITORIO LOCAL Y HAY CONFLICTO DE RAMAS

Esto sucede cuando queremos enviar cambios al repo remoto y una o varias líneas de nuestro código local también se encuentran modificadas por cambios en el repo remoto.

Al hacer git push origin main por ejemplo, obtenemos el mensaje que nos indica que hay cambios divergentes, esto es, hay cambios en el repo remoto que no están en local

```
german@fedora:~/Documentos/CodeBase/Git/conflicto$ git push origin main
To https://github.com/gangaritah/test2025
 ! [rejected]        main -> main (fetch first)
error: falló el empuje de algunas referencias a 'https://github.com/gangaritah/test2025'
hint: Updates were rejected because the remote contains work that you do not
hint: have locally. This is usually caused by another repository pushing to
hint: the same ref. If you want to integrate the remote changes, use
hint: 'git pull' before pushing again.
hint: See the 'Note about fast-forwards' in 'git push --help' for details.
```

Aplicamos la misma estrategia, hacemos *git pull origin main* para actualizar nuestro repo local y nos damos cuenta que hay un conflicto de ramas, esto lo vemos en la consola

```
german@fedora:~/Documentos/CodeBase/Git/conflicto$ git pull origin main
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 3 (delta 2), reused 0 (delta 0), pack-reused 0 (from 0)
Desempaquetando objetos: 100% (3/3), 928 bytes | 928.00 KiB/s, listo.
Desde https://github.com/gangaritah/test2025
 * branch                main          -> FETCH_HEAD
    09c2fe1..2b4e80d      main          -> origin/main
Auto-fusionando index.js
CONFLICTO (contenido): Conflicto de fusión en index.js
Fusión automática falló; arregle los conflictos y luego realice un commit con el resultado.
```

y si vamos al editor de código, por ejemplo, veremos un conflicto en él

```
index.js 9+,! X JS app.js
JS index.js > ...
1  function name(params) {
2      console.log("HOLA EAM");
3  }
4
5  function nomina(params) {
6      <<<<<<< HEAD (Current Change)
7      console.log("ppppppppp");
8      =====
9      console.log("Nomina ok ok");
10     >>>>>>> 2b4e80deca2ff3c7213aacb0e84dbe1c75a37740 (Incoming Change)
11 }
12
```

Resolvemos el conflicto y guardamos

Hacemos un commit de resolución de conflicto

```
german@fedora:~/Documentos/CodeBase/Git/conflicto$ git commit -m "Commit de reso
lucion"
[main 01e5e0b] Commit de resolucion
```

volvemos a hacer git push origin main

```
german@fedora:~/Documentos/CodeBase/Git/conflicto$ git push origin main
Enumerando objetos: 10, listo.
Contando objetos: 100% (10/10), listo.
Compresión delta usando hasta 8 hilos
Comprimiendo objetos: 100% (6/6), listo.
Escribiendo objetos: 100% (6/6), 610 bytes | 305.00 KiB/s, listo.
Total 6 (delta 3), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (3/3), completed with 1 local object.
To https://github.com/gangaritah/test2025
  2b4e80d..01e5e0b  main -> main
```

vemos que ahora nuestro código local se ha fusionado exitosamente con el código del repo remoto

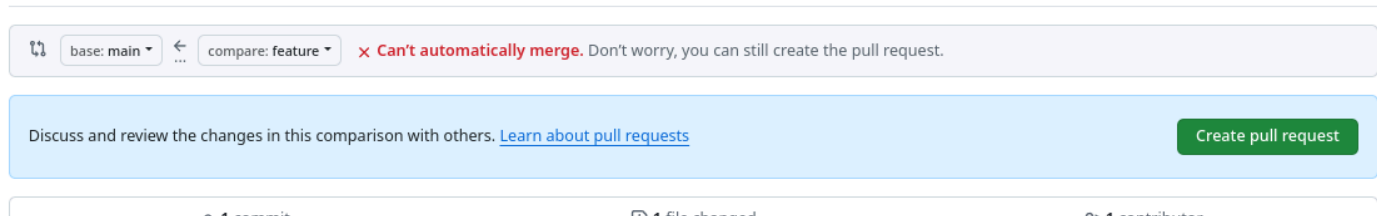
### 3. HACEMOS PULL REQUESTS Y SE GENERA UN CONFLICTO DE RAMAS

Cuando trabajamos en equipo y hacemos PR (pull request), se puede generar un conflicto de ramas, esto ocurre cuando alguien más ha modificado las mismas líneas de código en la rama de destino.

Por ejemplo, si hacemos pull request para fusionar las ramas, en este caso, la fusión sería de la rama base(main) con la rama secundaria (feature)

#### Comparing changes

Choose two branches to see what's changed or to start a new pull request. If you need to, you can also [compare across forks](#) or [learn more about diff comparisons](#).



Obtenemos una advertencia de conflictos al fusionar, en este caso no debemos hacer click en Create pull request, sino que hacemos git pull origin main estando en la rama feature(según el ejemplo) para ver los conflictos

```
* [new branch] feature -> feature
german@fedora:~/Documentos/CodeBase/Git/conflicto$ git pull origin main
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 3 (delta 2), reused 0 (delta 0), pack-reused 0 (from 0)
Desempaquetando objetos: 100% (3/3), 928 bytes | 464.00 KiB/s, listo.
Desde https://github.com/gangaritah/test2025
* branch          main          -> FETCH_HEAD
   01e5e0b..e0435a3 main          -> origin/main
Auto-fusionando index.js
CONFLICTO (contenido): Conflicto de fusión en index.js
Fusión automática falló; arregle los conflictos y luego realice un commit con el resultado.
```

Vemos que en la consola se marca el conflicto

Si vemos el editor tendremos las marcas que nos indican un conflicto de ramas

```
}

Accept Current Change | Accept Incoming Change
<<<<<< HEAD (Current Change)
console.log(-----)
=====
console.log(-----)
>>>>>> e0435a3fd2af3a53d
```

Resolvemos el conflicto y guardamos  
Hacemos un commit de resolución de conflicto

```
german@fedora:~/Documentos/CodeBase/Git/conflicto$ git add .
german@fedora:~/Documentos/CodeBase/Git/conflicto$ git commit -m "Commit de resolucion"
[feature 08e3a7f] Commit de resolucion
german@fedora:~/Documentos/CodeBase/Git/conflicto$
```


hacemos *git push origin feature* para enviar los cambios a la rama secundaria


```
german@fedora:~/Documentos/CodeBase/Git/conflicto$ git push origin feature
Enumerando objetos: 7, listo.
Contando objetos: 100% (7/7), listo.
Compresión delta usando hasta 8 hilos
Comprimiendo objetos: 100% (3/3), listo.
Escribiendo objetos: 100% (3/3), 316 bytes | 316.00 KiB/s, listo.
Total 3 (delta 2), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To https://github.com/gangaritah/test2025
  30504dc..08e3a7f  feature -> feature
```

Ahora ya podemos hacer nuevamente el pull request

### Comparing changes

Choose two branches to see what's changed or to start a new pull request. If you need to, you can also [compare across forks](#) or [learn more about diff comparisons](#).

 base: main

 compare: feature

✓ Able to merge. These branches can be automatically merged.

Discuss and review the changes in this comparison with others. [Learn about pull requests](#)

Create pull request

Vemos que ya no hay conflictos y damos click a Create pull request. Ahora esperamos que nuestro equipo revise y apruebe la fusión de nuestro código.