Nombre Completo: Martin Giovaruzzo.

https://github.com/erkcgn/taller_Git

Taller de Git – GitHub

- ¿Cómo se crea un repositorio en Git? git init
- ¿Cómo se clona un repositorio en Git? git clone URL o Path
- ¿Cómo se crea una rama en Git? git branch nombreRama

```
MINGW64:/c/Users/Usuario/OneDrive/Escritorio/G&L/BECA - ACELERACION/tallerGit

Usuario@ERKCGN MINGW64 ~/OneDrive/Escritorio/G&L/BECA - ACELERACION/tallerGit (main)
$ git add .

Usuario@ERKCGN MINGW64 ~/OneDrive/Escritorio/G&L/BECA - ACELERACION/tallerGit (main)
$ git commit -m "inicio repo tallerGit"
[main (root-commit) d3ef5f7] inicio repo tallerGit
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 Taller Git.docx

Usuario@ERKCGN MINGW64 ~/OneDrive/Escritorio/G&L/BECA - ACELERACION/tallerGit (main)
$ git branch
* main

Usuario@ERKCGN MINGW64 ~/OneDrive/Escritorio/G&L/BECA - ACELERACION/tallerGit (main)
$ git branch feature-prueba

Usuario@ERKCGN MINGW64 ~/OneDrive/Escritorio/G&L/BECA - ACELERACION/tallerGit (main)
$ git branch
  feature-prueba
* main
```

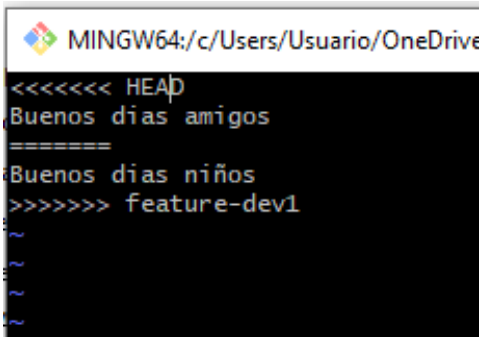
- ¿Cómo se fusionan dos ramas en Git? git merge nombreRama

```
Usuario@ERKCGN MINGW64 ~/OneDrive/Escritorio/G&L/BECA - ACELERACION/tallerGit
(main)
$ git merge feature-prueba2
Updating b6535eb..9a56b7d
Fast-forward
 .megaEjemplo.txt.swp   | Bin 0 -> 12288 bytes
 .superEjemplo.txt.swp  | Bin 0 -> 12288 bytes
 megaEjemplo.txt        |   1 +
 superEjemplo.txt       |   1 +
 4 files changed, 2 insertions(+)
 create mode 100644 .megaEjemplo.txt.swp
 create mode 100644 .superEjemplo.txt.swp
 create mode 100644 megaEjemplo.txt
```

```
Usuario@ERKCGN MINGW64 ~/OneDrive/Escritorio/G&L/BECA - ACELERACION/tallerGit (develop)
$ git merge feature-dev1
Auto-merging ramaDevelop.txt
CONFLICT (content): Merge conflict in ramaDevelop.txt
Automatic merge failed; fix conflicts and then commit the result.
```

- ¿Cómo se resuelve un conflicto de fusión en Git?

```
Usuario@ERKCGN MINGW64 ~/OneDrive/Escritorio/G&L/BECA - ACELERACION/tallerGit (develop|ME
GING)
$ git status
On branch develop
You have unmerged paths.
  (fix conflicts and run "git commit")
  (use "git merge --abort" to abort the merge)

Unmerged paths:
  (use "git add <file>..." to mark resolution)
        both modified:   ramaDevelop.txt

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

Usuario@ERKCGN MINGW64 ~/OneDrive/Escritorio/G&L/BECA - ACELERACION/tallerGit
(develop|MERGING)
$ vi ramaDevelop.txt
```



```
Usuario@ERKCGN MINGW64 ~/OneDrive/Escritorio/G&L/BECA - ACELERACION/tallerGit
(develop|MERGING)
$ git add .

Usuario@ERKCGN MINGW64 ~/OneDrive/Escritorio/G&L/BECA - ACELERACION/tallerGit
(develop|MERGING)
$ git commit -m "solucione conflicto"
[develop 12aedfb] solucione conflicto

Usuario@ERKCGN MINGW64 ~/OneDrive/Escritorio/G&L/BECA - ACELERACION/tallerGit
(develop)
$ git push origin develop
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 307 bytes | 307.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/erkcgn/taller_Git.git
   c29f24f..12aedfb  develop -> develop
```

Tener la ultima version, git fetch seguido git merge o git rebase. Econtrar el conflicto.. archivos como CONFLICT, resolver..., git add 'nombreArchivoConflicto' y luego git commit para completar la fusion.

- ¿Cómo se hace un push en Git? git push origin nombreRama

```
Usuario@ERKCGN MINGW64 ~/OneDrive/Escritorio/G&L/BECA - ACELERACION/tallerGit
(main)
$ git push origin main
Everything up-to-date
```

```
Usuario@ERKCGN MINGW64 ~/OneDrive/Escritorio/G&L/BECA - ACELERACION/tallerGit (develop)
$ git push origin develop
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (2/2), 243 bytes | 243.00 KiB/s, done.
Total 2 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
remote:
remote: Create a pull request for 'develop' on GitHub by visiting:
remote:      https://github.com/erkcgn/taller_Git/pull/new/develop
remote:
To https://github.com/erkcgn/taller_Git.git
 * [new branch]      develop -> develop
```

- ¿Cómo se hace un pull en Git? git pull origin NOMBRE_DE_LA_RAMA

```
Usuario@ERKCGN MINGW64 ~/OneDrive/Escritorio/G&L/BECA - ACELERACION/tallerGit
(feature-prueba)
$ git pull origin feature-prueba
From https://github.com/erkcgn/taller_Git
 * branch            feature-prueba -> FETCH_HEAD
Merge made by the 'ort' strategy.
 megaEjemplo.txt  | 1 +
 superEjemplo.txt | 1 +
 2 files changed, 2 insertions(+)
 create mode 100644 megaEjemplo.txt
```

```
Usuario@ERKCGN MINGW64 ~/OneDrive/Escritorio/G&L/BECA - ACELERACION/tallerGit
(main)
$ git pull origin main
remote: Enumerating objects: 1, done.
remote: Counting objects: 100% (1/1), done.
remote: Total 1 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (1/1), 619 bytes | 154.00 KiB/s, done.
From https://github.com/erkcgn/taller_Git
 * branch            main       -> FETCH_HEAD
   75f8942..07d8e95  main       -> origin/main
Updating 75f8942..07d8e95
Fast-forward
 .megaEjemplo.txt.swo  | Bin 0 -> 12288 bytes
 .megaEjemplo.txt.swp  | Bin 12288 -> 12288 bytes
 .nuevoArchivo.txt.swp | Bin 0 -> 12288 bytes
 ejemplo.txt           |   0
 lalala.txt            |   0
 nuevoArchivo.txt      |   0
 probandoStash.txt     |   0
 7 files changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 .megaEjemplo.txt.swo
 create mode 100644 .nuevoArchivo.txt.swp
 create mode 100644 ejemplo.txt
 create mode 100644 lalala.txt
 create mode 100644 nuevoArchivo.txt
 create mode 100644 probandoStash.txt
```

- ¿Cómo se crea una etiqueta en Git? git tag vX.0.0

```
Usuario@ERKCGN MINGW64 ~/OneDrive/Escritorio/G&L/BECA - ACELERACION/tallerGit
(main)
$ git tag v2.0.0

Usuario@ERKCGN MINGW64 ~/OneDrive/Escritorio/G&L/BECA - ACELERACION/tallerGit
(main)
$ git log --oneline
75f8942 (HEAD -> main, tag: v2.0.0, origin/main) otro nuevoArchivo
```

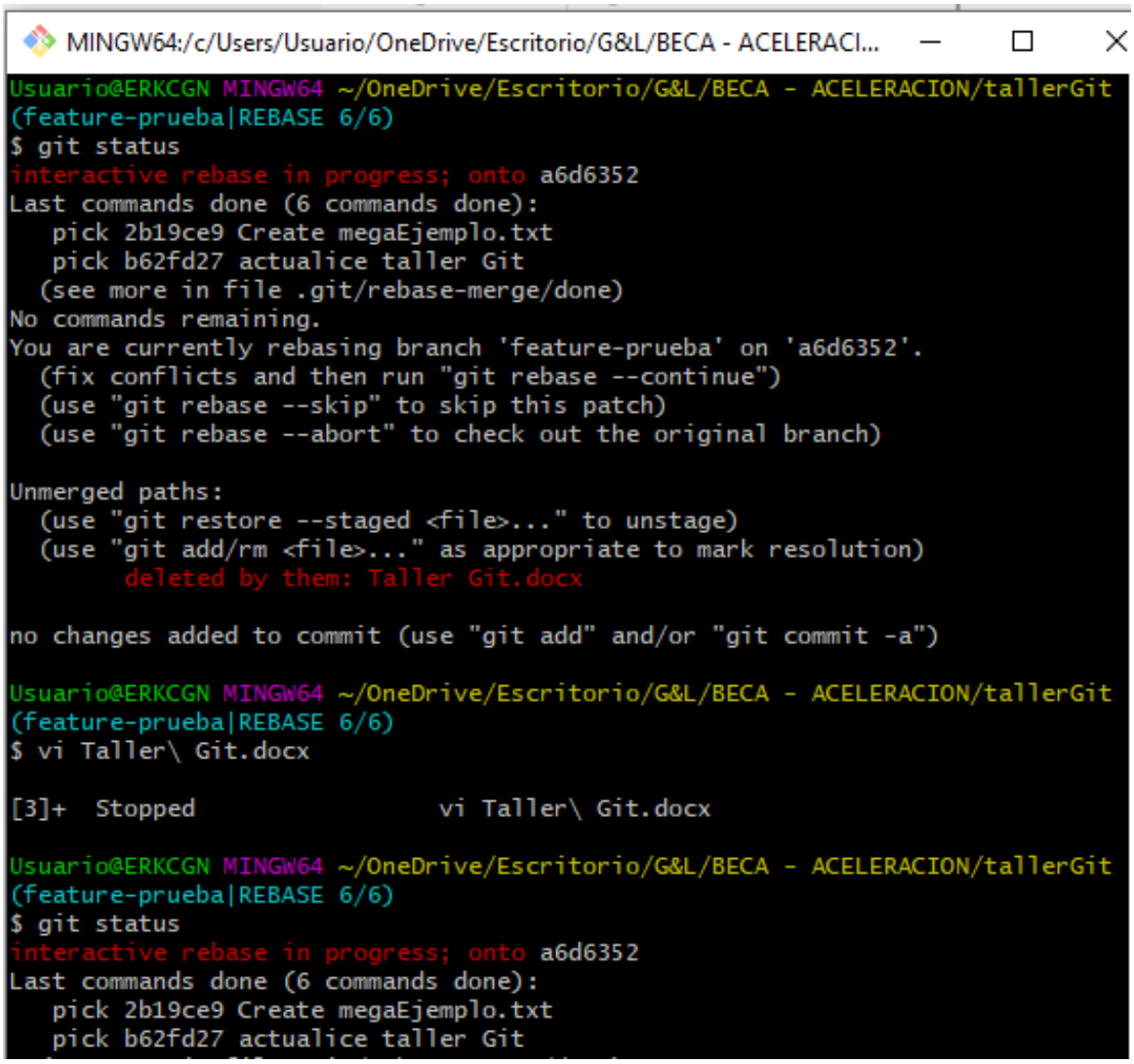
```
Usuario@ERKCGN MINGW64 ~/OneDrive/Escritorio/G&L/BECA - ACELERACION/tallerGit
(main)
$ git push --tags
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/erkcgn/taller_Git.git
 * [new tag]         v1.0.0 -> v1.0.0
 * [new tag]         v2.0.0 -> v2.0.0
```

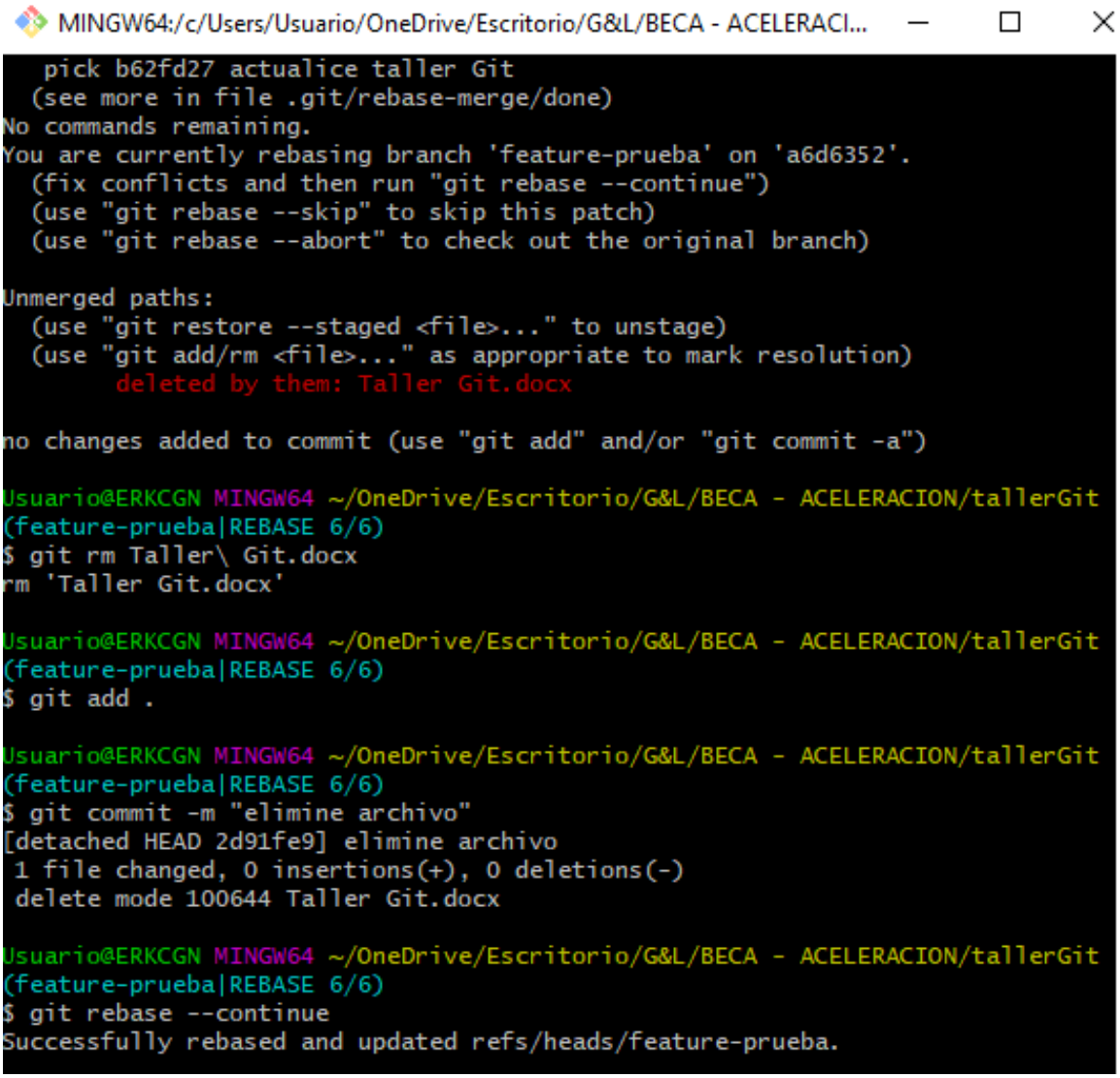
- ¿Cómo se revierte un commit en Git? git revert "hash" luego git push

```
Usuario@ERKCGN MINGW64 ~/OneDrive/Escritorio/G&L/BECA - ACELERACION/tallerGit
(feature-prueba)
$ git revert 009bfc7
[feature-prueba 0be37a5] Revert "cree archivo cielo"
 1 file changed, 0 insertions(+), 0 deletions(-)
 delete mode 100644 cielo.txt

Usuario@ERKCGN MINGW64 ~/OneDrive/Escritorio/G&L/BECA - ACELERACION/tallerGit
(feature-prueba)
$ git push origin feature-prueba
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 8 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 415 bytes | 415.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/erkcgn/taller_Git.git
   e3a3270..0be37a5  feature-prueba -> feature-prueba
```

- ¿Cómo se hace un rebase en Git? git rebase feature-xxxx
- ¿Cómo se resuelve un conflicto de rebase en Git?

```
Usuario@ERKCGN MINGW64 ~/OneDrive/Escritorio/G&L/BECA - ACELERACION/tallerGit
(feature-prueba)
$ git rebase feature-prueba3
CONFLICT (modify/delete): Taller Git.docx deleted in b62fd27 (actualice taller
 Git) and modified in HEAD.  Version HEAD of Taller Git.docx left in tree.
error: could not apply b62fd27... actualice taller Git
hint: Resolve all conflicts manually, mark them as resolved with
hint: "git add/rm <conflicted_files>", then run "git rebase --continue".
hint: You can instead skip this commit: run "git rebase --skip".
hint: To abort and get back to the state before "git rebase", run "git rebase
--abort".
Could not apply b62fd27... actualice taller Git
```


MINGW64:/c/Users/Usuario/OneDrive/Escritorio/G&L/BECA - ACELERACI...
Usuario@ERKCGN MINGW64 ~/OneDrive/Escritorio/G&L/BECA - ACELERACION/tallerGit
(feature-prueba|REBASE 6/6)
$ git status
interactive rebase in progress; onto a6d6352
Last commands done (6 commands done):
pick 2b19ce9 Create megaEjemplo.txt
pick b62fd27 actualice taller Git
(see more in file .git/rebase-merge/done)
No commands remaining.
You are currently rebasing branch 'feature-prueba' on 'a6d6352'.
(fix conflicts and then run "git rebase --continue")
(use "git rebase --skip" to skip this patch)
(use "git rebase --abort" to check out the original branch)
Unmerged paths:
(use "git restore --staged <file>..." to unstage)
(use "git add/rm <file>..." as appropriate to mark resolution)
deleted by them: Taller Git.docx
no changes added to commit (use "git add" and/or "git commit -a")
Usuario@ERKCGN MINGW64 ~/OneDrive/Escritorio/G&L/BECA - ACELERACION/tallerGit
(feature-prueba|REBASE 6/6)
$ vi Taller\ Git.docx
[3]+ Stopped vi Taller\ Git.docx
Usuario@ERKCGN MINGW64 ~/OneDrive/Escritorio/G&L/BECA - ACELERACION/tallerGit
(feature-prueba|REBASE 6/6)
$ git status
interactive rebase in progress; onto a6d6352
Last commands done (6 commands done):
pick 2b19ce9 Create megaEjemplo.txt
pick b62fd27 actualice taller Git



1- git status(ver archivos con conflicto)

2 - buscar marcadores <<<<<<<, ======= y >>>>>>>

3- editar el codigo en conflicto y eliminar los marcadores.

4- git add .

5- git rebase --continue

- ¿Cómo se usa el comando cherry-pick en Git? git cherry-pick "asdf123" hash del commit

```
Usuario@ERKCGN MINGW64 ~/OneDrive/Escritorio/G&L/BECA - ACELERACION/tallerGit
(main)
$ git cherry-pick a6d6352
[main 75f8942] otro nuevoArchivo
 Date: Tue May 9 09:19:03 2023 -0300
 2 files changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 nuevoArchivo2.txt
```

- ¿Cómo se usa el comando stash en Git?

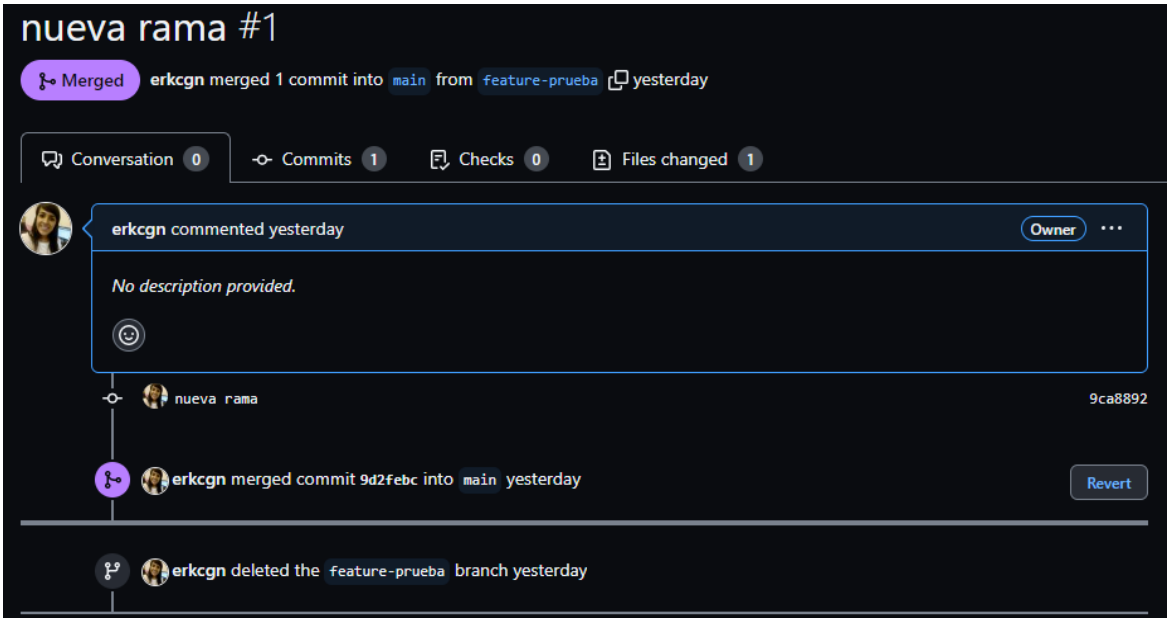
```
Usuario@ERKCGN MINGW64 ~/OneDrive/Escritorio/G&L/BECA - ACELERACION/tallerGit
(feature-prueba)
$ touch probandoStash.txt

Usuario@ERKCGN MINGW64 ~/OneDrive/Escritorio/G&L/BECA - ACELERACION/tallerGit
(feature-prueba)
$ git add .

Usuario@ERKCGN MINGW64 ~/OneDrive/Escritorio/G&L/BECA - ACELERACION/tallerGit
(feature-prueba)
$ git stash
Saved working directory and index state WIP on feature-prueba: 0be37a5 Revert
"cree archivo cielo"

Usuario@ERKCGN MINGW64 ~/OneDrive/Escritorio/G&L/BECA - ACELERACION/tallerGit
(feature-prueba)
$ git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.

Usuario@ERKCGN MINGW64 ~/OneDrive/Escritorio/G&L/BECA - ACELERACION/tallerGit
(main)
$ git checkout feature-prueba
Switched to branch 'feature-prueba'

Usuario@ERKCGN MINGW64 ~/OneDrive/Escritorio/G&L/BECA - ACELERACION/tallerGit
(feature-prueba)
$ git stash apply
On branch feature-prueba
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   probandoStash.txt
```

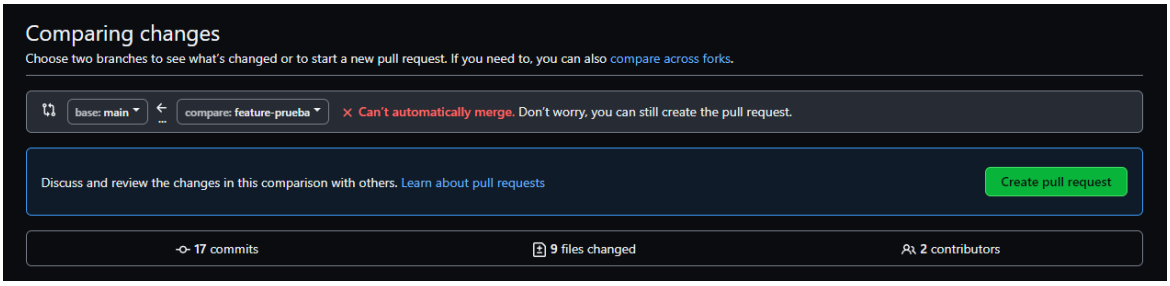
1- git stash (no quiero commitear ni perder el trabajo)

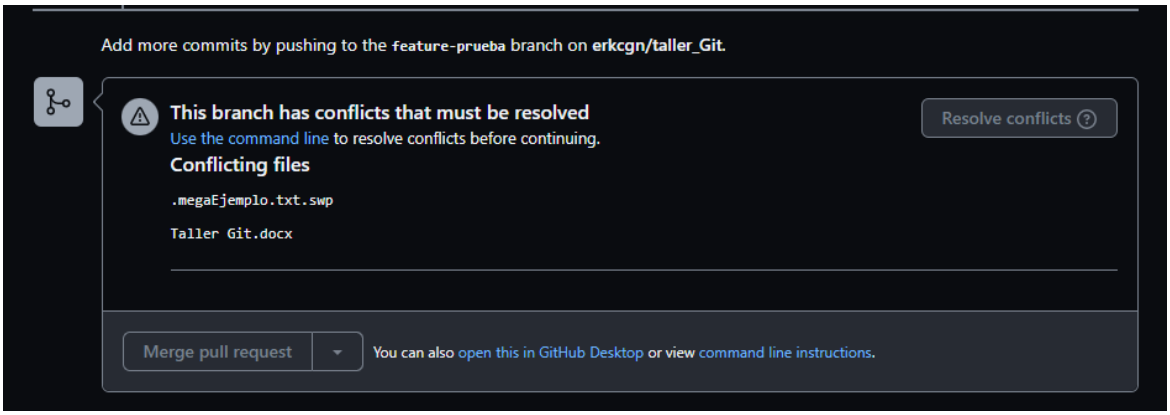
2 -retomo lo pendiente con git stash pop (index) o git stash apply (luego git stash clear).

- ¿Cómo se crea un pull request en GitHub?

pedido de validacion para unificar una rama a la otra, resolucion de conflictos y merge.

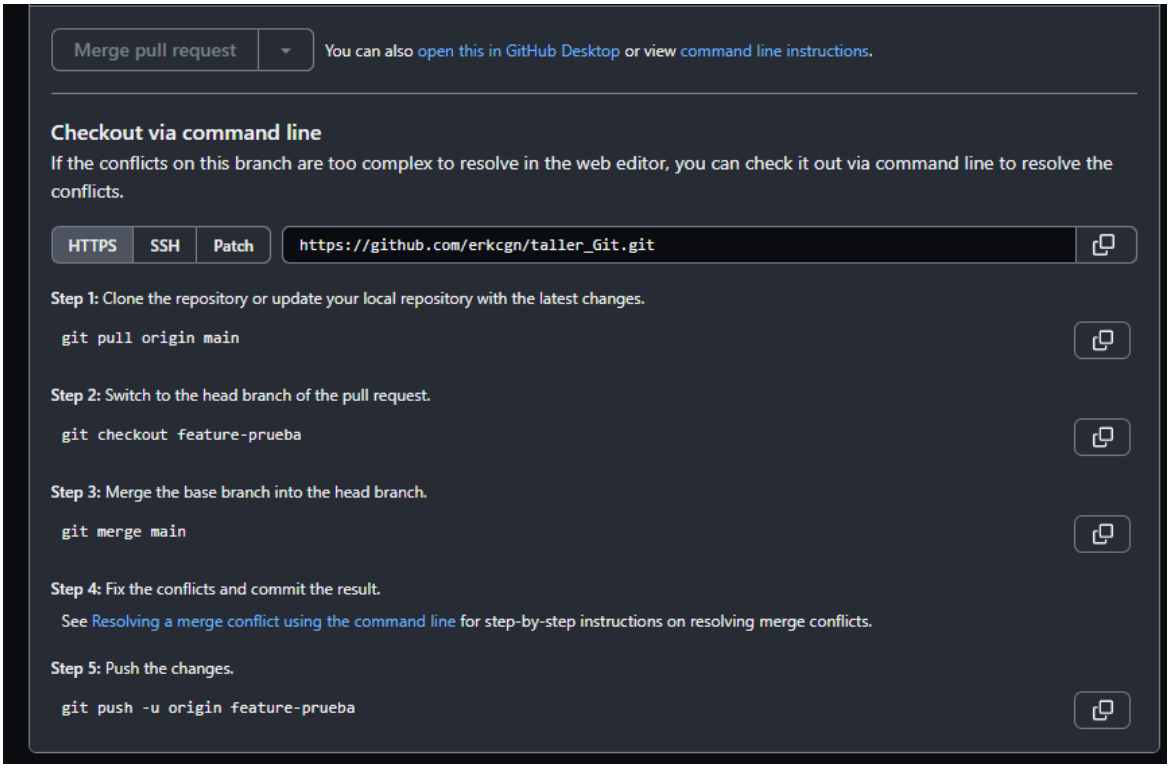


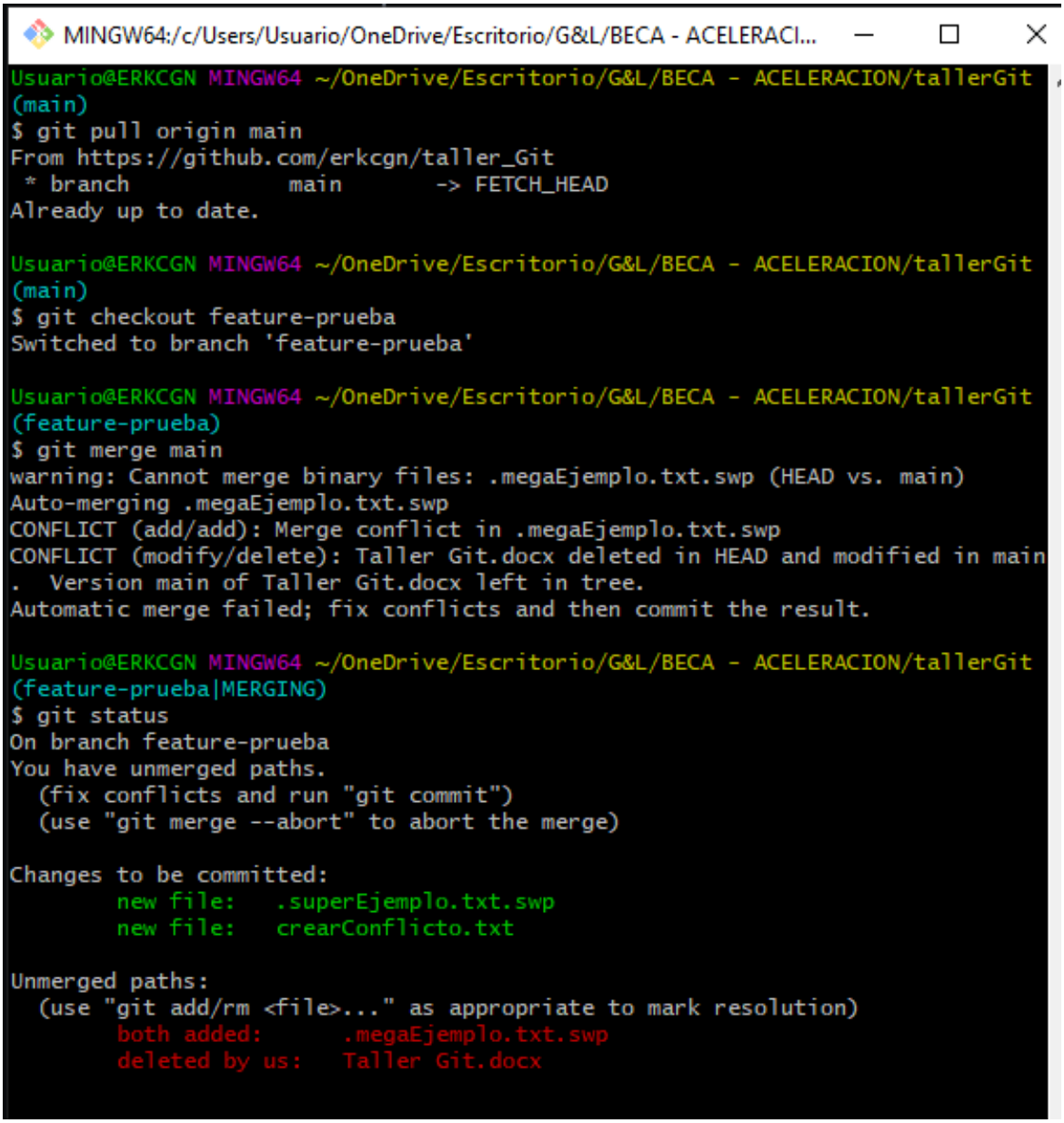


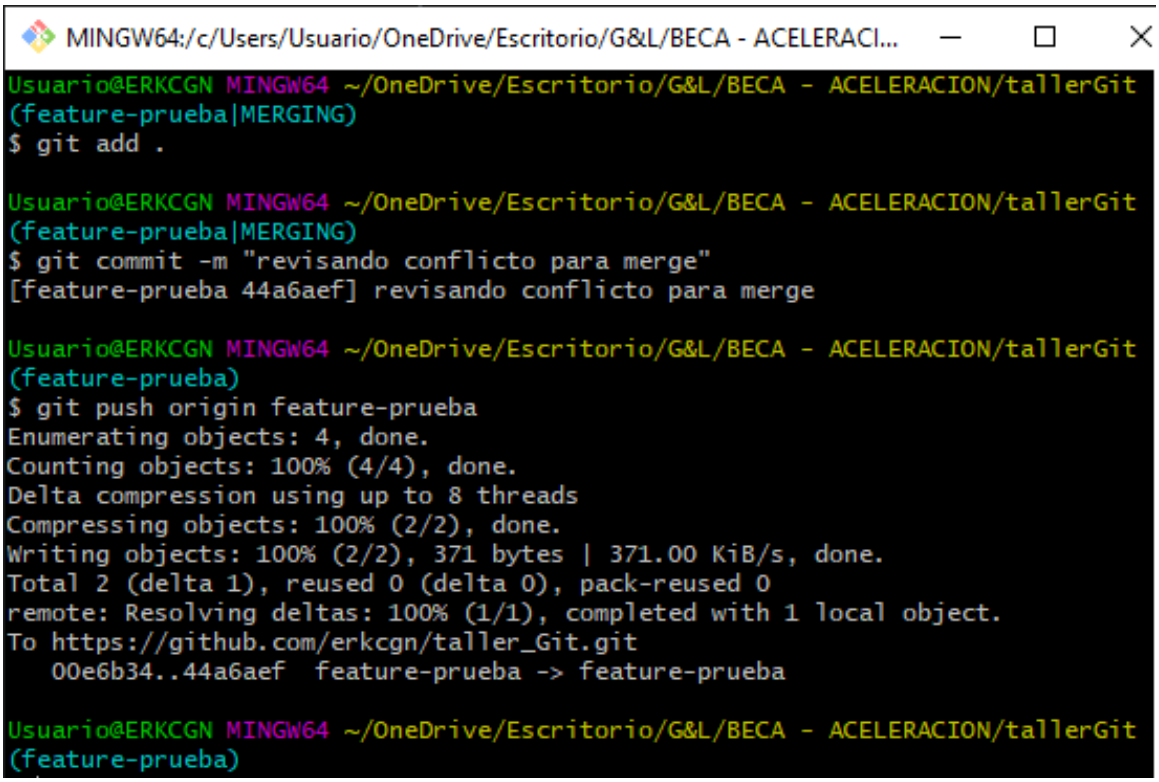


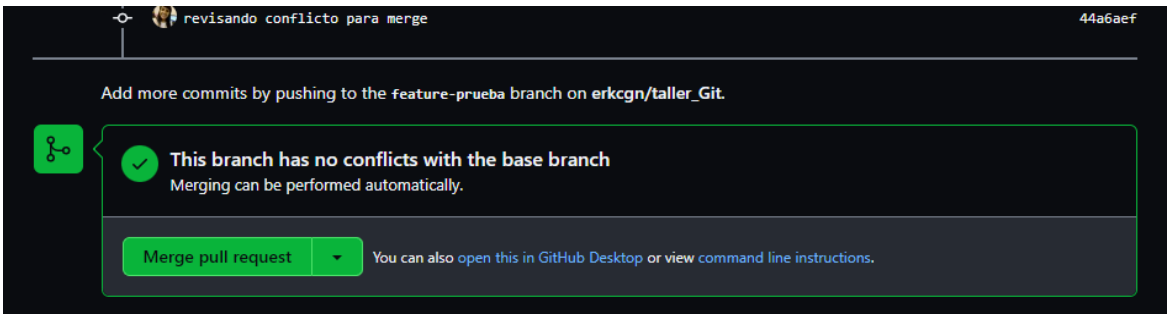
por ejemplo si creo una nueva rama github me propone realizar un compare & pull requests, acepto y al abrirse puedo escribir un comentario y luego crearlo (Create pull request). Luego me da la opcion de Merge pull request (en caso de que no haya conflictos).

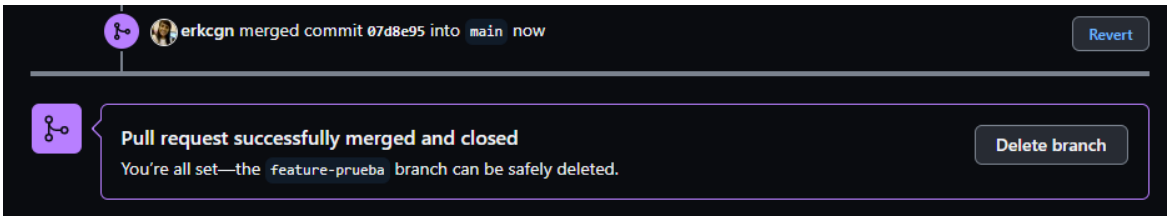
- ¿Cómo se revisan los cambios en un pull request en GitHub?


Merge pull request
You can also open this in GitHub Desktop or view command line instructions.
Checkout via command line
If the conflicts on this branch are too complex to resolve in the web editor, you can check it out via command line to resolve the conflicts.
HTTPS
SSH
Patch
https://github.com/erkcgn/taller_Git.git
Step 1: Clone the repository or update your local repository with the latest changes.
git pull origin main
Step 2: Switch to the head branch of the pull request.
git checkout feature-prueba
Step 3: Merge the base branch into the head branch.
git merge main
Step 4: Fix the conflicts and commit the result.
See Resolving a merge conflict using the command line for step-by-step instructions on resolving merge conflicts.
Step 5: Push the changes.
git push -u origin feature-prueba


MINGW64:/c/Users/Usuario/OneDrive/Escritorio/G&L/BECA - ACELERACI...
Usuario@ERKCGN MINGW64 ~/OneDrive/Escritorio/G&L/BECA - ACELERACION/tallerGit (main)
$ git pull origin main
From https://github.com/erkcgn/taller_Git
 * branch   main   -> FETCH_HEAD
Already up to date.
Usuario@ERKCGN MINGW64 ~/OneDrive/Escritorio/G&L/BECA - ACELERACION/tallerGit (main)
$ git checkout feature-prueba
Switched to branch 'feature-prueba'
Usuario@ERKCGN MINGW64 ~/OneDrive/Escritorio/G&L/BECA - ACELERACION/tallerGit (feature-prueba)
$ git merge main
warning: Cannot merge binary files: .megaEjemplo.txt.swp (HEAD vs. main)
Auto-merging .megaEjemplo.txt.swp
CONFLICT (add/add): Merge conflict in .megaEjemplo.txt.swp
CONFLICT (modify/delete): Taller Git.docx deleted in HEAD and modified in main. Version main of Taller Git.docx left in tree.
Automatic merge failed; fix conflicts and then commit the result.
Usuario@ERKCGN MINGW64 ~/OneDrive/Escritorio/G&L/BECA - ACELERACION/tallerGit (feature-prueba|MERGING)
$ git status
On branch feature-prueba
You have unmerged paths.
  (fix conflicts and run "git commit")
  (use "git merge --abort" to abort the merge)
Changes to be committed:
        new file: .superEjemplo.txt.swp
        new file: crearConflicto.txt
Unmerged paths:
  (use "git add/rm <file>..." as appropriate to mark resolution)
        both added: .megaEjemplo.txt.swp
        deleted by us: Taller Git.docx







En otros casos tambien desde Resolve conflicts en el pull request que los presente, me lleva a la interface de usuario de github donde se visualiza el conflicto:

<<<<< ramaX blabla

==========

asdfasd >>>>> main

aqui borramos lo que no nos interese y nos habilita el boton Mark as resolved al clickear nos muestra un tilde verde y un boton Committing merge! y ahi si nos permite el Merge pull request.

- ¿Cómo se acepta un pull request en GitHub?

  explique arriba

- ¿Cómo se trabaja con submódulos en Git?

  git submodule add con la URL del proyecto que desea empezar a rastrear

  https://git-scm.com/book/es/v2/Herramientas-de-Git-Submódulos

- ¿Cómo se usa Git bisect para encontrar un commit problemático?

```
Usuario@ERKCGN MINGW64 ~/OneDrive/Escritorio/G&L/BECA - ACELERACION/tallerGit
(feature-prueba)
$ git bisect start

Usuario@ERKCGN MINGW64 ~/OneDrive/Escritorio/G&L/BECA - ACELERACION/tallerGit
(feature-prueba|BISECTING)
$ git bisect good e0019af

Usuario@ERKCGN MINGW64 ~/OneDrive/Escritorio/G&L/BECA - ACELERACION/tallerGit
(feature-prueba|BISECTING)
$ git bisect bad 2b19ce9
Some good revs are not ancestors of the bad rev.
git bisect cannot work properly in this case.
Maybe you mistook good and bad revs?
```

git bisect start

git bisect good <commit> donde <commit> es el identificador del commit que funciona correctamente

git bisect bad <commit> donde <commit> es el identificador del commit donde se presenta el problema.

Git seleccionará un commit en el medio del rango y te pedirá que pruebes si el problema aún está presente. Si el problema aún existe, ejecuta el comando git bisect bad, si el problema no está presente, ejecuta el comando git bisect good.

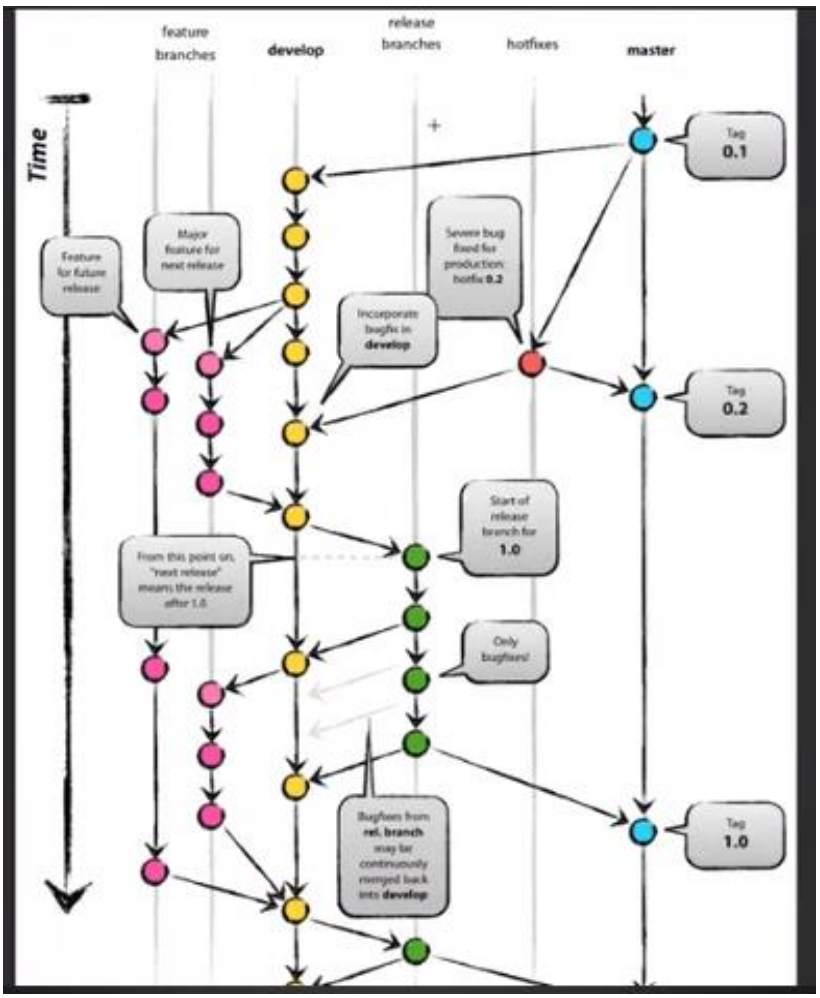
Cuando Git encuentre el commit problemático, te indicará el identificador de ese commit. Puedes ver los detalles del commit con el comando git log <commit>.

Cuando hayas terminado de usar Git bisect, ejecuta el comando git bisect reset para volver al estado anterior a la búsqueda de bisect.

- ¿Cómo se configura un flujo de trabajo de Git flow?

cual va a ser el flujo de trabajo! marco de trabajo!

```
Usuario@ERKCGN MINGW64 ~ (main)
$ git flow
usage: git flow <subcommand>

Available subcommands are:
   init      Initialize a new git repo with support for the branching model.
   feature   Manage your feature branches.
   bugfix    Manage your bugfix branches.
   release   Manage your release branches.
   hotfix    Manage your hotfix branches.
   support   Manage your support branches.
   version   Shows version information.
   config    Manage your git-flow configuration.
   log       Show log deviating from base branch.

Try 'git flow <subcommand> help' for details.
```



crear rama feature: git flow feature start "nombreRama"

git flow feature finish "nombreRama"

git flow release start 0.1 -- se fusiona en main

git flow release finish 0.1

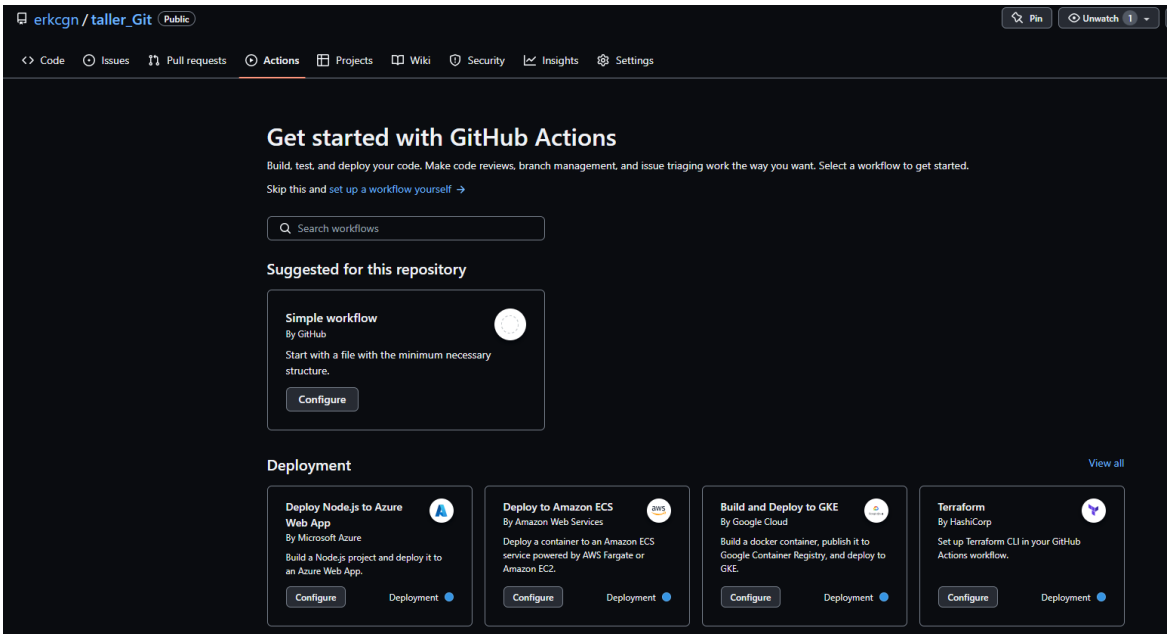
se fusiona con la rama main, esto tiene un commit y se relaciona a una etiqueta. y se fusiona con develop, si hay un problema se crea un hotfix.. git flow hotfix start hotfix_1.

- ¿Cómo se configura un flujo de trabajo de GitHub flow?

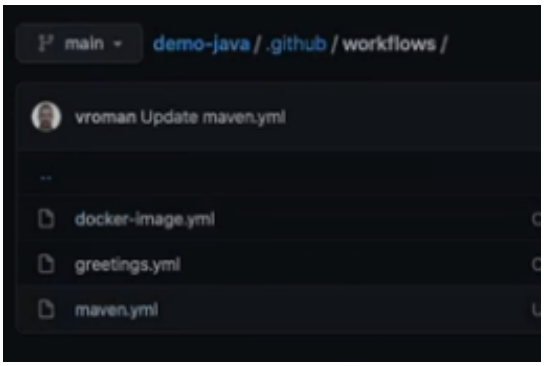
Dentro de "Actions":

GitHub Actions es una plataforma de integración y despliegue continuos (IC/DC) que te permite automatizar tu mapa de compilación, pruebas y despliegue. Puedes crear flujos de trabajo y crear y probar cada solicitud de cambios en tu repositorio o desplegar solicitudes de cambios fusionadas a producción

workflow --> eventos --> jobs --> steps --> actions --> runners



o crearlo a mano:



se configura, nombre, eventos, trabajo -->

19 lines (17 sloc) | 444 Bytes

```yaml
name: Greetings

on:
  pull_request:
    branches:
      - main

jobs:
  greeting:
    runs-on: ubuntu-latest
    permissions:
      issues: write
      pull-requests: write
    steps:
    - uses: actions/first-interaction@v1
      with:
        repo-token: ${{ secrets.GITHUB_TOKEN }}
        issue-message: 'Message that will be displayed on users first issue'
        pr-message: 'Message that will be displayed on users first pull request'
```