

Implementar ramas, hacer cambios y fusión de estas.

En git tenemos la posibilidad de crear entornos o 'caminos' diferentes en cada repositorio, a estos distintos entornos se les llama rama. Cada rama nace con el contenido en base de donde se cree, es decir, si tenemos la rama 'main' y creamos una rama 'develop' esta nueva rama se creará con el contenido que, hasta el momento, tiene 'main'. Después de crear la rama, esta será una ramificación de la rama de donde se creó, pero en cierto modo será independiente de esta, ya que tendrá sus propios commits, versiones o cambios, cambios que podrán ser integrados a otra rama fusionándose.

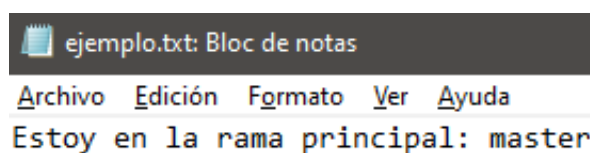
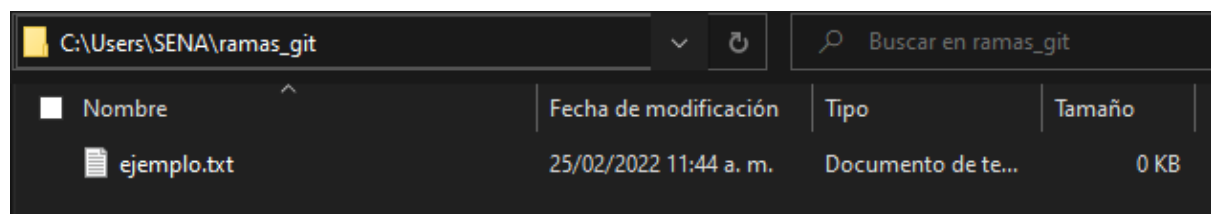
Se puede trabajar con ramas desde un repositorio local o remoto.

Ramas en repositorio local:

Iniciemos un repositorio, yo lo haré en una carpeta con nombre 'ramas_git'.

```
PS C:\Users\SENA\ramas_git> git init
Initialized empty Git repository in C:/Users/SENA/ramas_git/.git/
PS C:\Users\SENA\ramas_git> _
```

Cuando creamos el repositorio y hacemos nuestro primer commit, git automáticamente crea una rama principal llamada master. Comprobémoslo creando un archivo y guardándolo con git, yo lo haré con un archivo de texto llamado ejemplo.txt.



Creamos el archivo y si revisamos el estado de git, veremos que no está en seguimiento. Lo añadiremos para que git lleve un seguimiento de sus versiones y crearemos un commit o guardado de versión.

```

PS C:\Users\SENA\ramas_git> git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        ejemplo.txt

nothing added to commit but untracked files present (use "git add" to track)
PS C:\Users\SENA\ramas_git> git add .
PS C:\Users\SENA\ramas_git> git commit -m "Iniciación del repositorio con el archivo ejemplo.txt"
[master (root-commit) 96f2f02] Inicialización del repositorio con el archivo ejemplo.txt
 1 file changed, 1 insertion(+)
 create mode 100644 ejemplo.txt
PS C:\Users\SENA\ramas_git> git status
On branch master
nothing to commit, working tree clean
PS C:\Users\SENA\ramas_git> _

```

Ya con este primer commit, git creará la rama git por defecto, podemos verla con **git branch**

```

PS C:\Users\SENA\ramas_git> git branch
* master
PS C:\Users\SENA\ramas_git> _

```

Las ramas podemos renombrarlas, resetearlas, crearlas o eliminarlas.

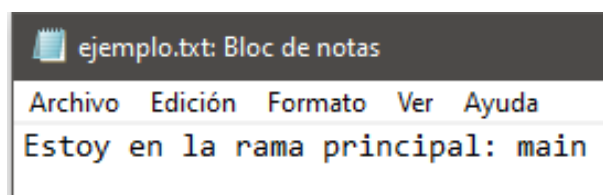
Primero vamos a modificar el nombre de 'master' a 'main' con el comando **git branch -M main**

```

PS C:\Users\SENA\ramas_git> git branch -M main
PS C:\Users\SENA\ramas_git> git branch
* main
PS C:\Users\SENA\ramas_git> _

```

Cambiaré el contenido de nuestro archivo ejemplo.txt para que el nombre de las ramas concuerden.



```

Windows PowerShell
PS C:\Users\SENA\ramas_git> git add .
PS C:\Users\SENA\ramas_git> git commit -m "Cambio de nombre de rama en ejemplo.txt"
[main 529bd4c] Cambio de nombre de rama en ejemplo.txt
1 file changed, 1 insertion(+), 1 deletion(-)
PS C:\Users\SENA\ramas_git>

```

Para crear una nueva rama y poder modificar este archivo sin que afecte a la rama main, utilizamos el comando **git checkout -B <nombre_rama>**:

```

Windows PowerShell
PS C:\Users\SENA\ramas_git> git branch
* main
PS C:\Users\SENA\ramas_git> git checkout -B develop
Switched to a new branch 'develop'
PS C:\Users\SENA\ramas_git> git branch
* develop
  main
PS C:\Users\SENA\ramas_git> _

```

Ahora tenemos dos ramas, main y develop. Por ahora, develop es un clon de main, es decir, tiene exactamente los mismos cambios y commits:

```

Windows PowerShell
PS C:\Users\SENA\ramas_git> git log
commit 529bd4c589abe155266d9ac20f1aabbf77f3332e1 (HEAD -> develop, main)
Author:
Date: Tue Mar 1 06:29:49 2022 -0500

    Cambio de nombre de rama en ejemplo.txt

commit 96f2f02348d2ec2311ae4ff242cf596d8daaf11f
Author:
Date: Fri Feb 25 11:53:33 2022 -0500

    Inicialización del repositorio con el archivo ejemplo.txt
PS C:\Users\SENA\ramas_git> _

```

Como vemos, los commits son los mismos. Vamos a hacer un cambio. Primero que todo asegurarnos que estamos en la rama 'develop' con el comando **git branch**:

```

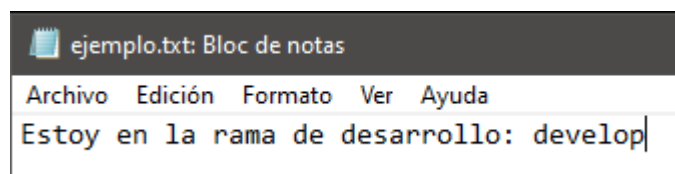
Windows PowerShell
PS C:\Users\SENA\ramas_git> git branch
* develop
  main
PS C:\Users\SENA\ramas_git>

```

Si llegamos a estar en la rama 'main' podemos desplazarnos entre ramas con el comando **git checkout <nombre_rama>**:

```
Windows PowerShell
PS C:\Users\SENA\ramas_git> git checkout main
Switched to branch 'main'
PS C:\Users\SENA\ramas_git> git branch
  develop
* main
PS C:\Users\SENA\ramas_git> git checkout develop
Switched to branch 'develop'
PS C:\Users\SENA\ramas_git> git branch
* develop
  main
PS C:\Users\SENA\ramas_git>
```

Estando en la rama develop, vamos a hacer un cambio en el archivo ejemplo.txt. Lo agregamos al staging area y le haremos un commit.



```
PS C:\Users\SENA\ramas_git> git status
On branch develop
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   ejemplo.txt

no changes added to commit (use "git add" and/or "git commit -a")
PS C:\Users\SENA\ramas_git> git add .\ejemplo.txt
PS C:\Users\SENA\ramas_git> git commit -m "Modificando nombre de la rama."
[develop 3e63559] Modificando nombre de la rama.
 1 file changed, 1 insertion(+), 1 deletion(-)
PS C:\Users\SENA\ramas_git> git status
On branch develop
nothing to commit, working tree clean
PS C:\Users\SENA\ramas_git> _
```

Si vemos el contenido del archivo ejemplo.txt en la rama develop, este será diferente al que hay en la main. Comprobémoslo:

```

PS C:\Users\SENA\ramas_git> git branch
* develop
  main
PS C:\Users\SENA\ramas_git> Get-Content .\ejemplo.txt
Estoy en la rama de desarrollo: develop
PS C:\Users\SENA\ramas_git> git checkout main
Switched to branch 'main'
PS C:\Users\SENA\ramas_git> git branch
  develop
* main
PS C:\Users\SENA\ramas_git> Get-Content .\ejemplo.txt
Estoy en la rama principal: main
PS C:\Users\SENA\ramas_git> _

```

Vemos que el contenido es diferente, ya que develop tiene sus propios commits. Los commits del repositorio los podemos revisar con el comando **git log**:

```

PS C:\Users\SENA\ramas_git> git branch
  develop
* main
PS C:\Users\SENA\ramas_git> git checkout develop
Switched to branch 'develop'
PS C:\Users\SENA\ramas_git> git branch
* develop
  main
PS C:\Users\SENA\ramas_git> git log
commit 3e6355933169e9ef9f10531207dfdbb0d28034ec (HEAD -> develop)
Author:
Date: Tue Mar 1 07:14:55 2022 -0500

    Modificando nombre de la rama.

commit 529bd4c589abe155266d9ac20f1aabbf77f3332e1 (main)
Author:
Date: Tue Mar 1 06:29:49 2022 -0500

    Cambio de nombre de rama en ejemplo.txt

commit 96f2f02348d2ec2311ae4ff242cf596d8daaf11f
Author:
Date: Fri Feb 25 11:53:33 2022 -0500

    Inicialización del repositorio con el archivo ejemplo.txt
PS C:\Users\SENA\ramas_git> _

```

Hasta el momento has aprendido a:

- Crear ramas en git.
- Cambiar de nombre a una rama.
- Cambiar entre ramas.
- Ver los commits de las ramas.

Ahora supongamos que en 'develop' tenemos cambios importantes que queremos pasar a la rama 'main' ¿Cómo hacemos esto?, para esto utilizamos la fusión de ramas, la cual consiste en juntar cambios de una rama con otra y, en caso de que haya conflictos, decidir qué cambios aceptar en la mezcla.

Para hacer una fusión o **merge**, nos ubicamos en la rama que queremos que se fusione con otra, en este caso, 'main'.

```
PS C:\Users\SENA\ramas_git> git checkout main
Switched to branch 'main'
PS C:\Users\SENA\ramas_git>
```

Ya en la rama, utilizaremos el comando **git merge <nombre_rama>** para fusionar las ramas:

```
PS C:\Users\SENA\ramas_git> git merge develop
Updating 529bd4c..3e63559
Fast-forward
 ejemplo.txt | 2 +-
 1 file changed, 1 insertion(+), 1 deletion(-)
PS C:\Users\SENA\ramas_git>
```

Ahora la rama 'main' tiene los cambios de la rama 'develop', veamos el estado de la rama:

```
PS C:\Users\SENA\ramas_git> git status
On branch main
nothing to commit, working tree clean
PS C:\Users\SENA\ramas_git> _
```

Vemos que todo está limpio, no hay cambios sin guardar, pero si miramos el contenido de ejemplo.txt...

```
PS C:\Users\SENA\ramas_git> git branch
  develop
* main
PS C:\Users\SENA\ramas_git> Get-Content .\ejemplo.txt
Estoy en la rama de desarrollo: develop
PS C:\Users\SENA\ramas_git>
```

Así es, los cambios que llevábamos en 'develop' ahora están también en 'main' y en los logs las dos ramas aparecen en el mismo nivel de commit, en el último.

```

PS C:\Users\SENA\ramas_git> git log
commit 3e6355933169e9ef9f10531207dfdbb0d28034ec (HEAD -> _main, develop)
Author:
Date: Tue Mar 1 07:14:55 2022 -0500

    Modificando nombre de la rama.

commit 529bd4c589abe155266d9ac20f1aabbf77f3332e1
Author:
Date: Tue Mar 1 06:29:49 2022 -0500

    Cambio de nombre de rama en ejemplo.txt

commit 96f2f02348d2ec2311ae4ff242cf596d8daaf11f
Author:
Date: Fri Feb 25 11:53:33 2022 -0500

    Inicialización del repositorio con el archivo ejemplo.txt
PS C:\Users\SENA\ramas_git> _

```

Las dos ramas se encuentran en HEAD, es decir, en el último commit.

Ya si no necesitamos más la rama de 'develop', podemos eliminarla con el comando **git branch -D <nombre_rama>**:

```

PS C:\Users\SENA\ramas_git> git branch -D develop
Deleted branch develop (was 3e63559).
PS C:\Users\SENA\ramas_git> git branch
* _main
PS C:\Users\SENA\ramas_git> _

```

Tenemos que tener en cuenta que:

- Las ramas no se pueden eliminar si hay cambios en el staging area o si hay cambios no añadidos o no removidos.
- La eliminación de una rama **NO SOLICITA CONFIRMACIÓN** por lo que hay que tener cuidado.

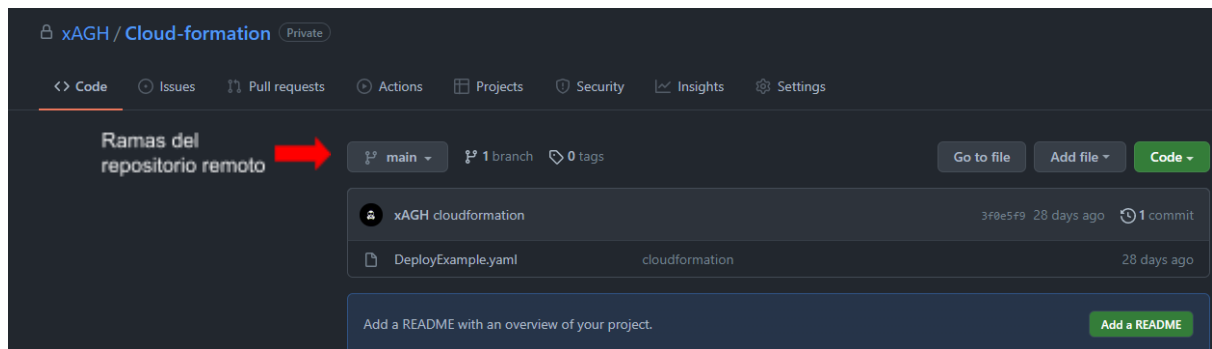
Las ramas se pueden subir a un repositorio remoto, en caso de que exista con el comando **git push <url o alias del repositorio remoto> <rama>**:

Ramas en repositorio remoto:

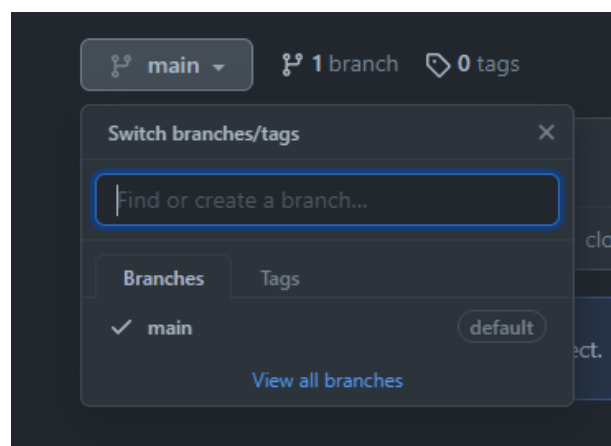
Ya vimos cómo manejar ramas en un repositorio local, ahora miraremos cómo trabajar con ramas en un repositorio remoto en Github.

En este caso, tomaré un repositorio remoto ya creado y con un archivo de configuración ya que hasta este punto debes de saber cómo crear tu cuenta en GitHub, como crear un repositorio remoto y cómo iniciarlo.

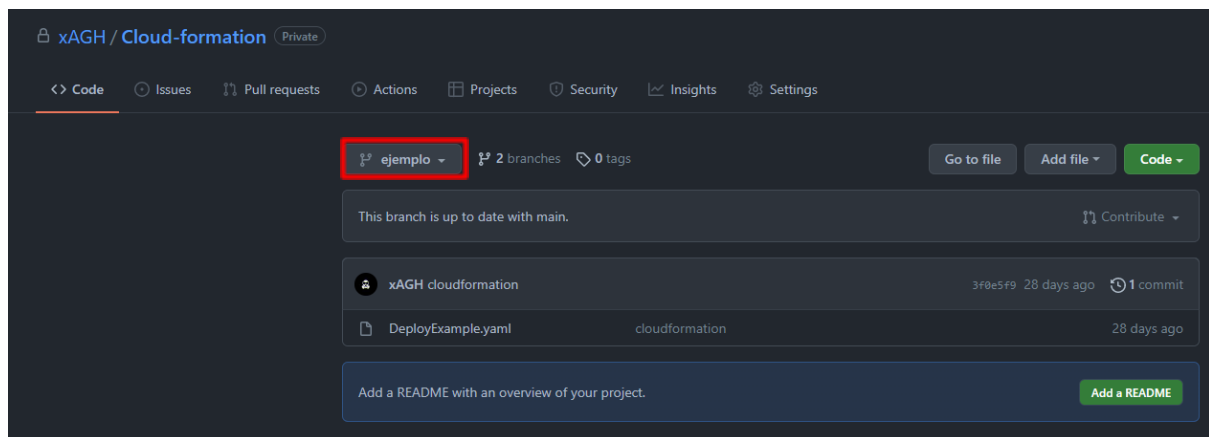
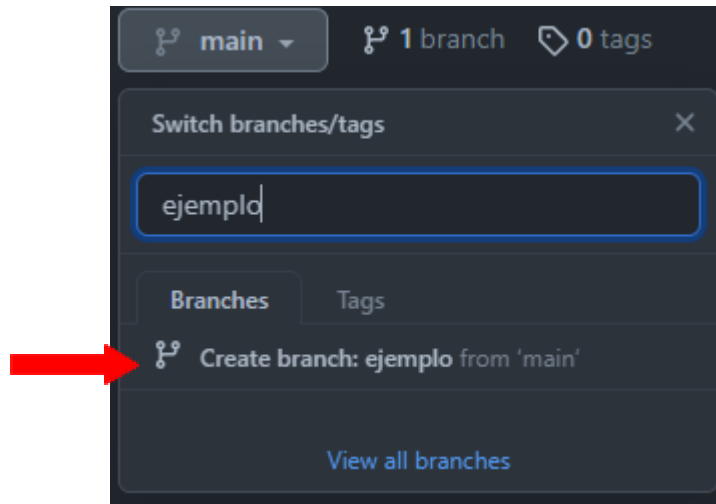
Este es nuestro repositorio:



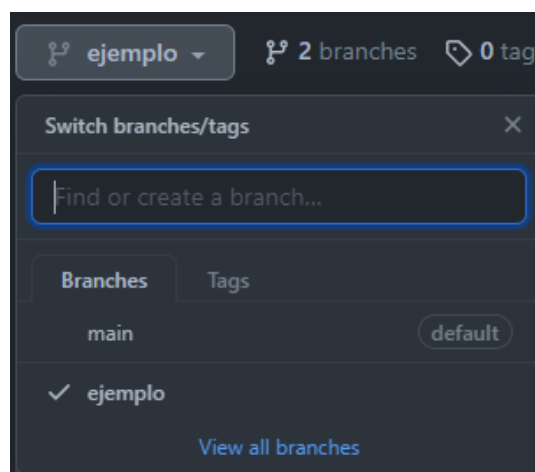
Si desplegamos las ramas, veremos sólo una.



Para crear una rama desde el repositorio remoto, buscamos en la barra de búsqueda el nombre que queramos poner a nuestra rama, en mi caso 'ejemplo'. Si esta rama no existe en el repositorio remoto, git la creará, si existe, simplemente la busca.



Ya la rama está creada. Si desplegamos las ramas, veremos las dos.



Ahora podemos hacer cambios en los archivos y estos quedarán guardados exclusivamente en su rama hasta que se fusionen.

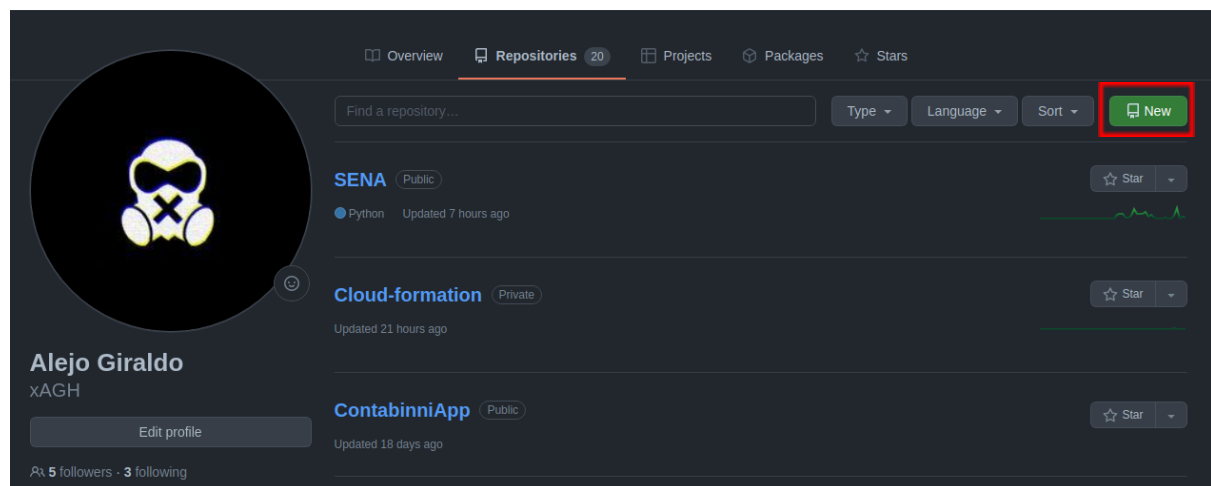
Solicitar revisiones y autorizaciones:

En grupos de trabajo grandes, es necesario que los commits subidos al repositorio remoto sean revisados y aprobados para evitar problemas en el despliegue o introducir bugs en el código, aquí vamos a simular una de estas situaciones, vamos a preparar nuestro entorno:

Repositorio local:

```
~/ramas_git > git init
Inicializado repositorio Git vacío en /home/alejo/ramas_git/.git/
~/ramas_git master > echo "Desde la rama main" > ejemplo.txt
~/ramas_git master ?1 > git add .
~/ramas_git master +1 > git commit -m "Inicio repo con ejemplo.txt"
[master (commit-raíz) 5c6ee54] Inicio repo con ejemplo.txt
1 file changed, 1 insertion(+)
create mode 100644 ejemplo.txt
~/ramas_git master > git branch -M main
~/ramas_git main > git checkout -B develop
Cambiado a nueva rama 'develop'
~/ramas_git develop > echo "Modificado desde develop" > ejemplo.txt
~/ramas_git develop !1 > git add .
~/ramas_git develop +1 > git commit -m "Modificacion en develop de ejemplo.txt"
[develop 605fd5b] Modificacion en develop de ejemplo.txt
1 file changed, 1 insertion(+), 1 deletion(-)
~/ramas_git develop > |
```


Ahora creamos el repositorio remoto para enlazarlo con el local:



Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Owner *

 xAGH ▾

Repository name *

/ ✓

Great repository name. ramas_git is available. [Porable](#). Need inspiration? How about [verbose-octo-winner?](#)

Description (optional)



Public

Anyone on the internet can see this repository. You choose who can commit.



Private

You choose who can see and commit to this repository.

Initialize this repository with:

Skip this step if you're importing an existing repository.



Add a README file

This is where you can write a long description for your project. [Learn more.](#)



Add .gitignore

Choose which files not to track from a list of templates. [Learn more.](#)



Choose a license

A license tells others what they can and can't do with your code. [Learn more.](#)

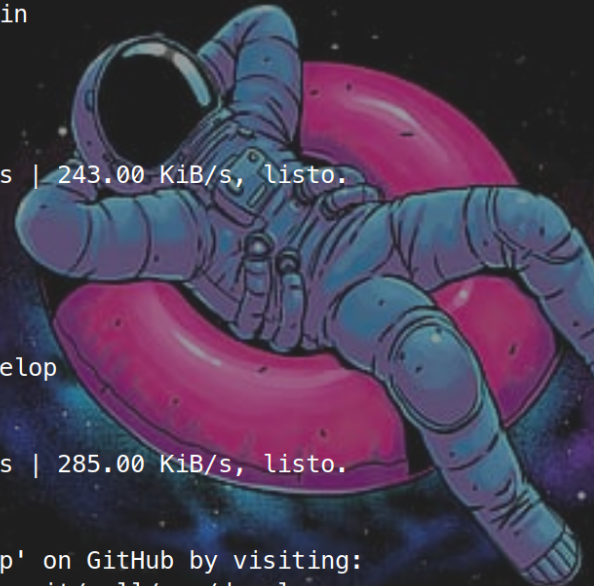
Create repository

Conexión a repositorio remoto:

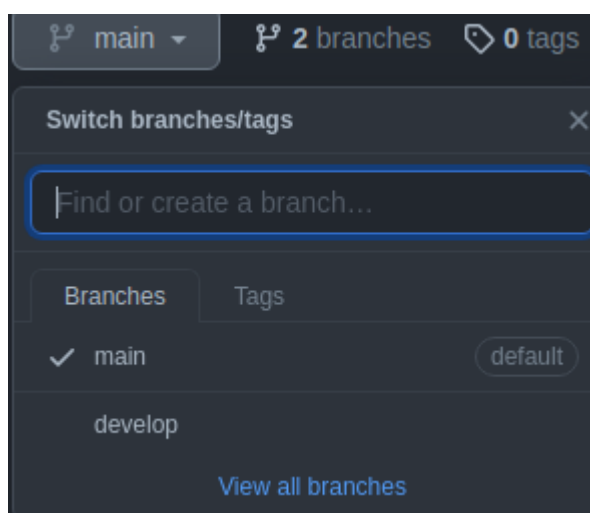
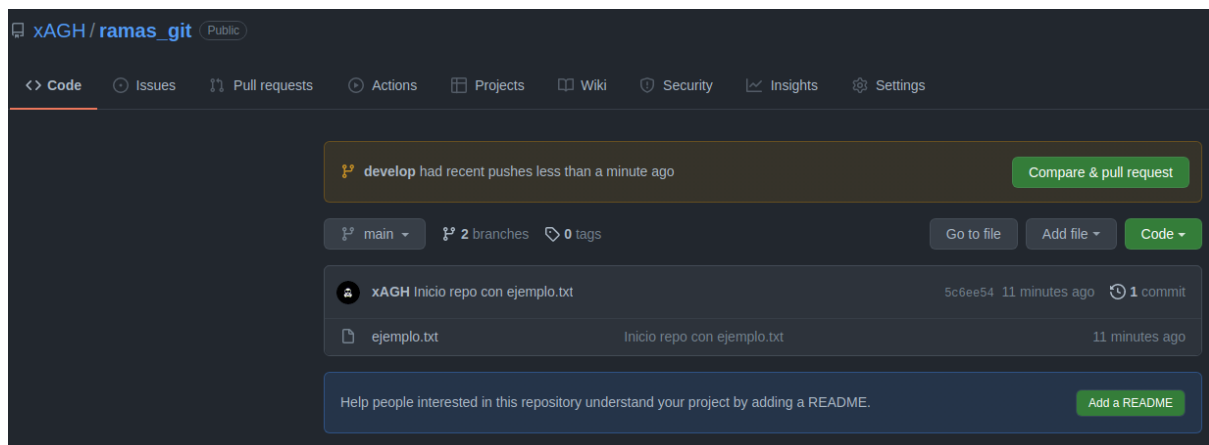
```
~/ramas_git develop > git checkout main
Cambiado a rama 'main'
~/ramas_git main > git remote add origin https://github.com/xAGH/ramas_git.git
~/ramas_git main > git push
fatal: La rama actual main no tiene una rama upstream.
Para realizar un push de la rama actual y configurar el remoto como upstream, use

    git push --set-upstream origin main

~/ramas_git main > git push origin main
Enumerando objetos: 3, listo.
Contando objetos: 100% (3/3), listo.
Escribiendo objetos: 100% (3/3), 243 bytes | 243.00 KiB/s, listo.
Total 3 (delta 0), reusado 0 (delta 0)
To https://github.com/xAGH/ramas_git.git
 * [new branch]      main -> main
~/ramas_git main > git checkout develop
Cambiado a rama 'develop'
~/ramas_git develop > git push origin develop
Enumerando objetos: 5, listo.
Contando objetos: 100% (5/5), listo.
Escribiendo objetos: 100% (3/3), 285 bytes | 285.00 KiB/s, listo.
Total 3 (delta 0), reusado 0 (delta 0)
remote:
remote: Create a pull request for 'develop' on GitHub by visiting:
remote:      https://github.com/xAGH/ramas_git/pull/new/develop
remote:
To https://github.com/xAGH/ramas_git.git
 * [new branch]      develop -> develop
~/ramas_git develop > |
```

An illustration of an astronaut in a blue spacesuit floating in space next to a large, pink, inflatable ring. The astronaut is positioned as if they are about to enter or have just exited the ring. The background is a dark, starry space.

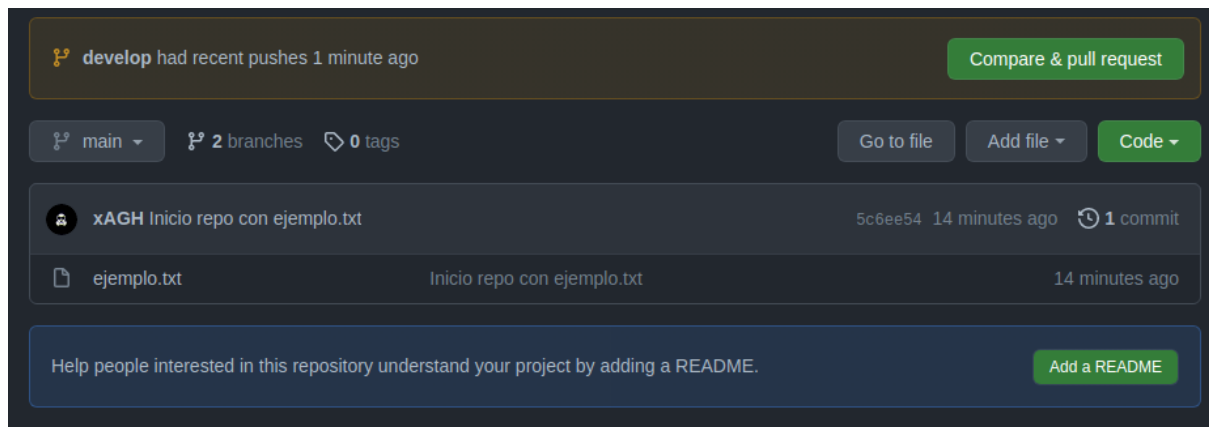
Ahora en nuestro repo remoto tenemos:



Como vemos en nuestra rama main nos aparece una alerta que nos dice que en 'develop' hay cambios que no están en la main, creamos otro archivo en 'develop' para visualizar mejor los cambios:

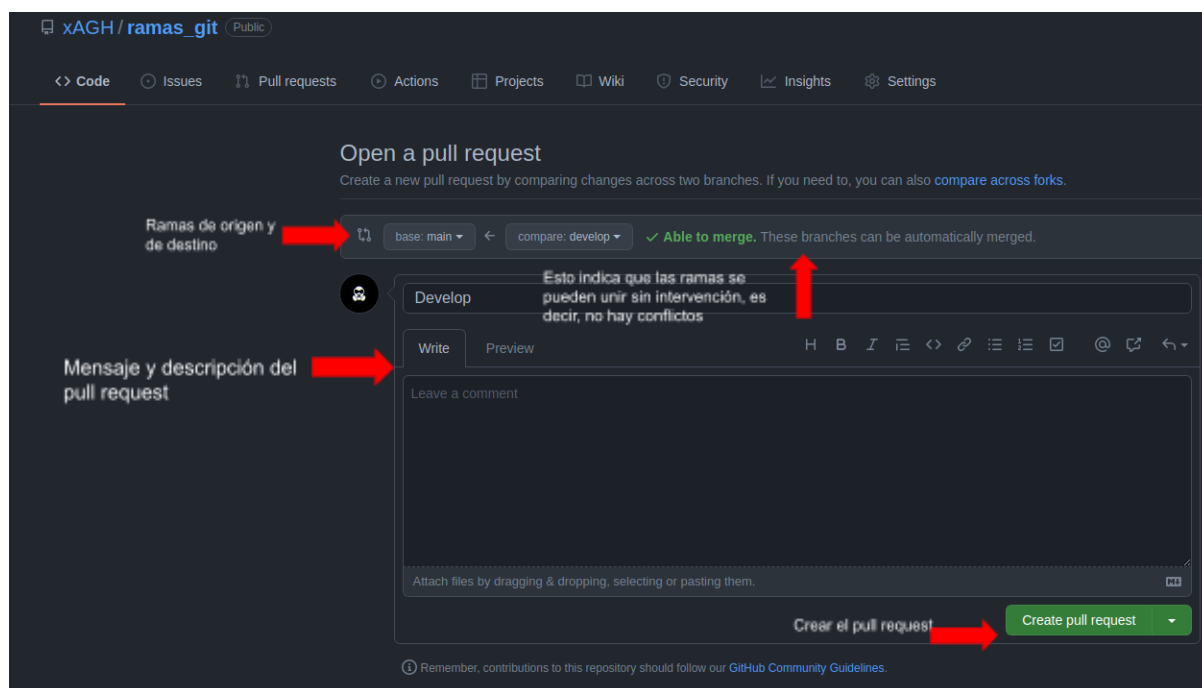
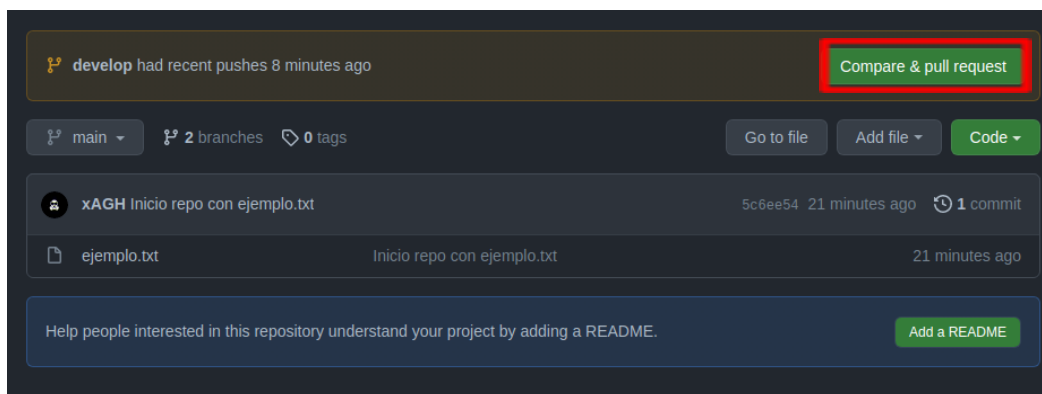
```
~/ramas_git develop > echo "Otro archivo mas" > Otro.txt
~/ramas_git develop ?1 > git add .
~/ramas_git develop +1 > git commit -m "Se agrega otro archivo mas"
[develop 4181678] Se agrega otro archivo mas
1 file changed, 1 insertion(+)
create mode 100644 Otro.txt
~/ramas_git develop > git push origin develop
Enumerando objetos: 4, listo.
Contando objetos: 100% (4/4), listo.
Compresión delta usando hasta 8 hilos
Comprimiendo objetos: 100% (2/2), listo.
Escribiendo objetos: 100% (3/3), 301 bytes | 301.00 KiB/s, listo.
Total 3 (delta 0), reusado 0 (delta 0)
To https://github.com/xAGH/ramas_git.git
605fd5b..4181678 develop -> develop
~/ramas_git develop > |
```

Esto nos aparece en el repositorio remoto:

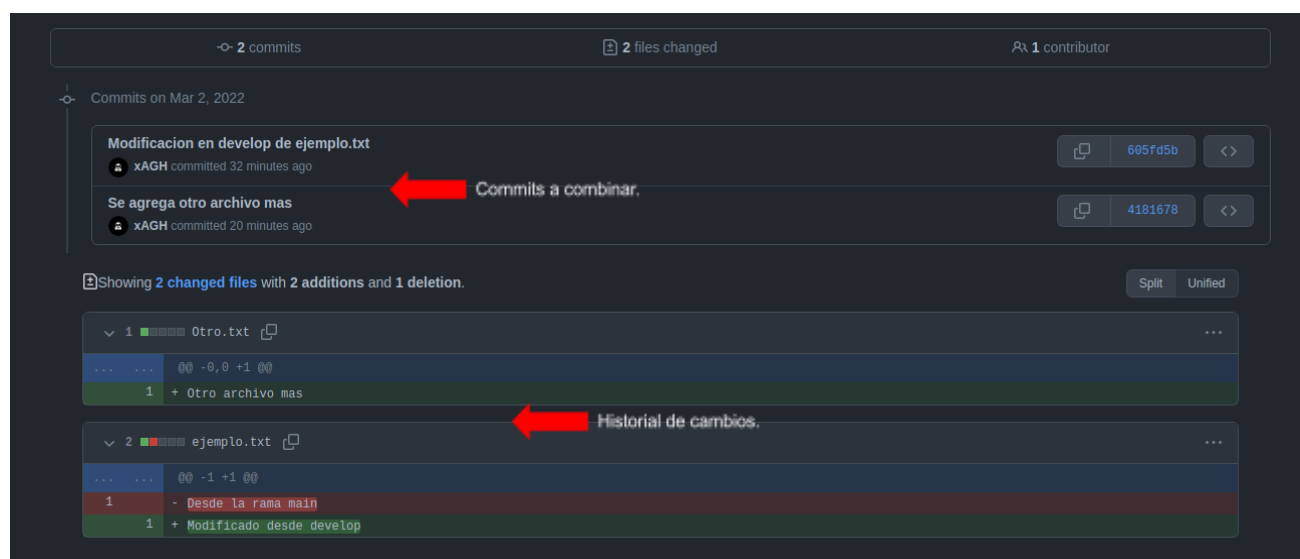


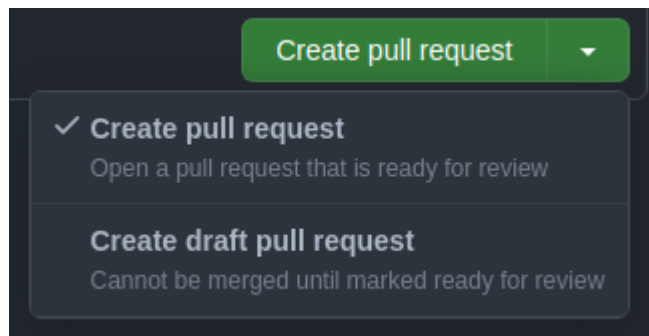
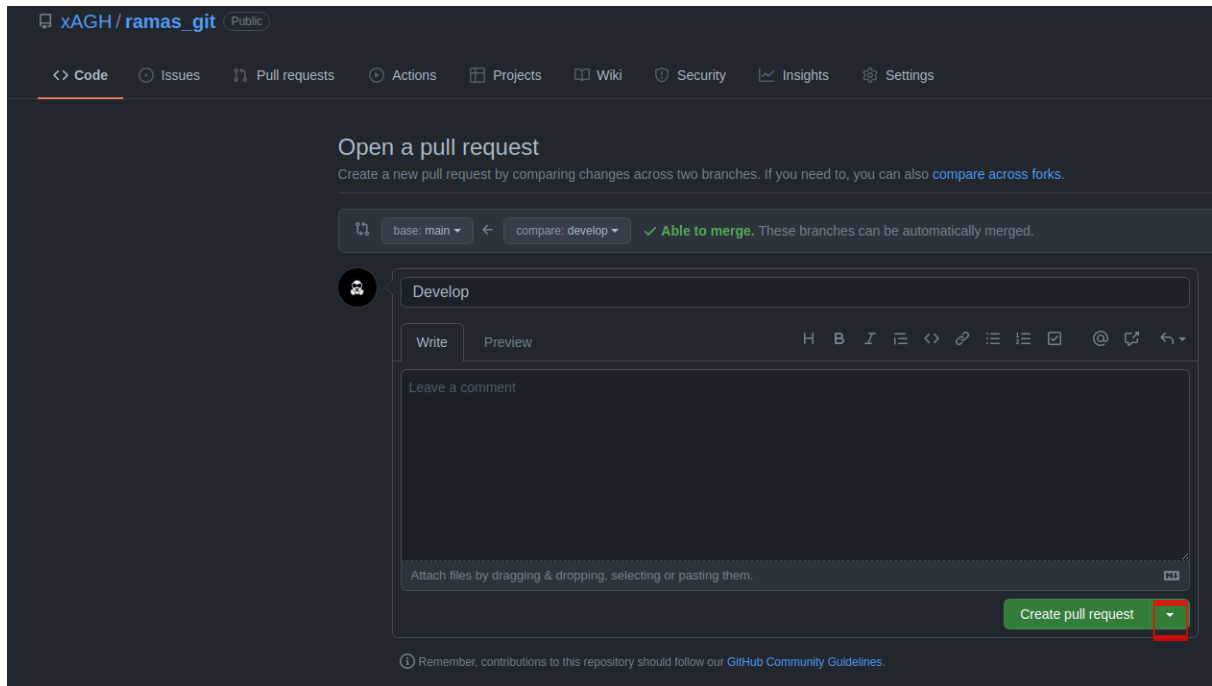
Un pull request es una solicitud que hace develop hacia main para integrar los cambios, como yo soy el dueño y único colaborador del repositorio podré hacer y aceptar el pull request por mi cuenta.

Vamos a ver como funciona un pull request:

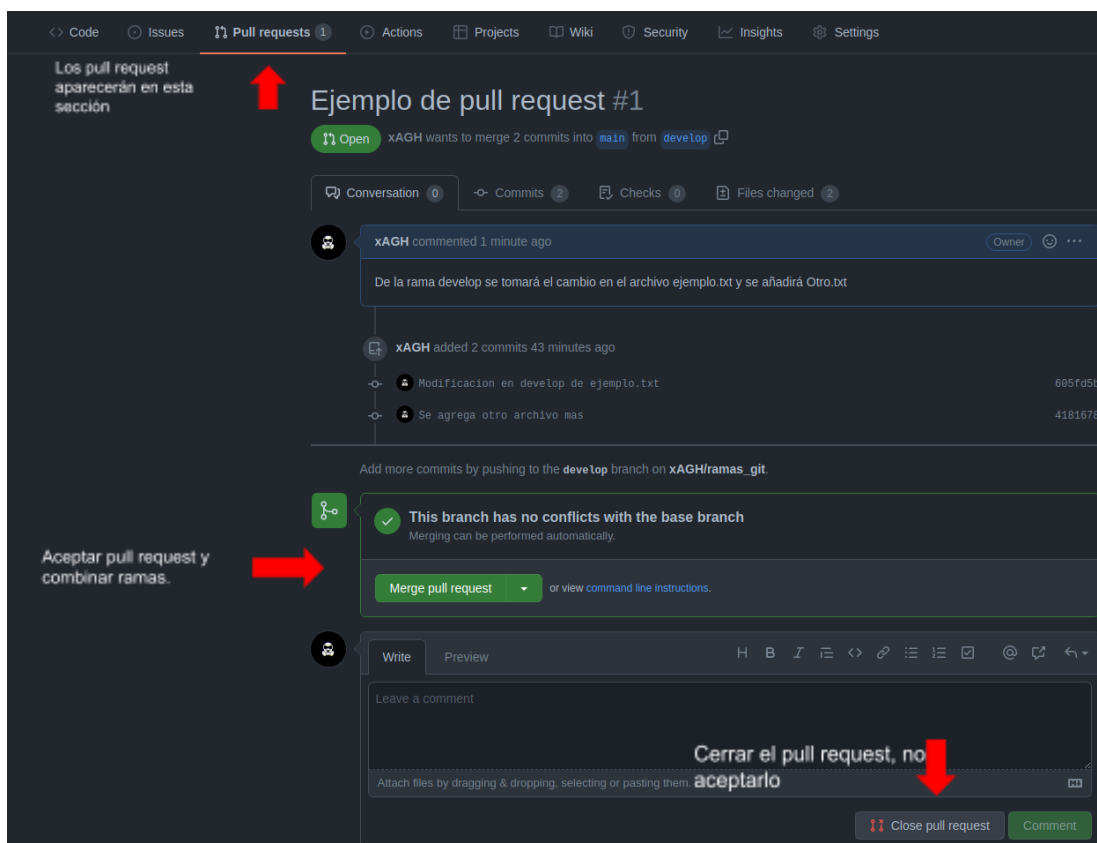
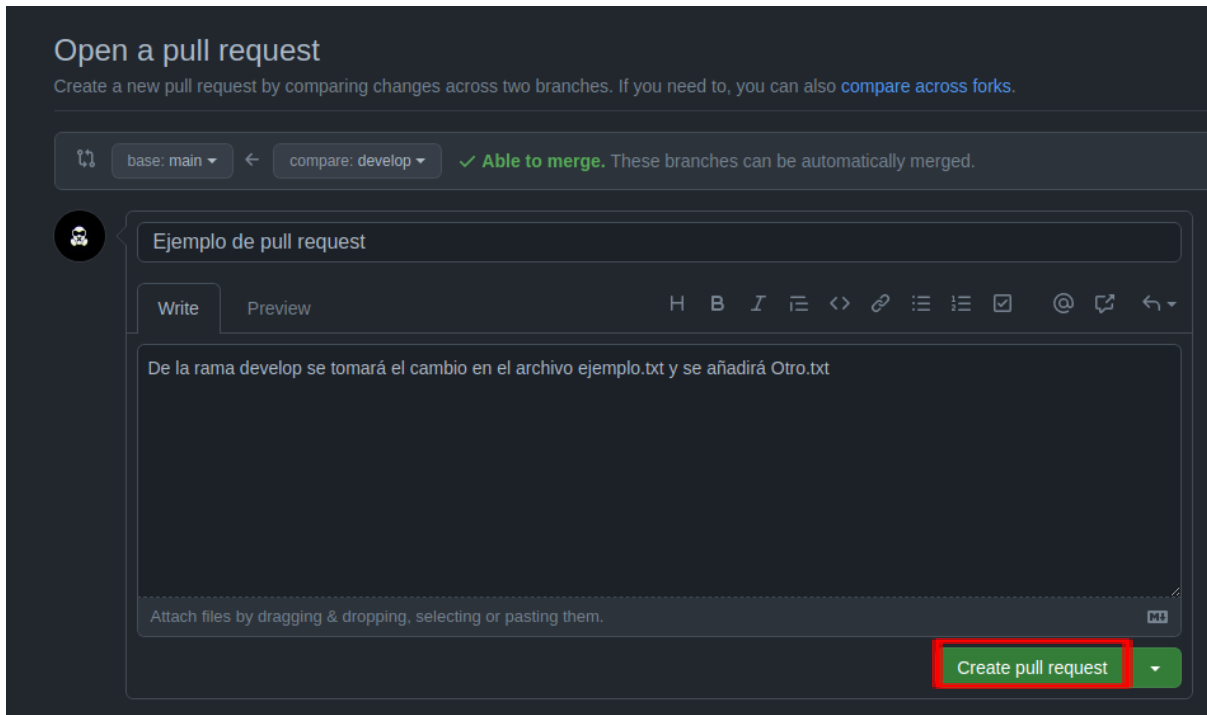


Más abajo tenemos:





Create pull request: Crea un pull request listo para ser revisado y aprobado.
Create draft pull request: Crea un pull request que debe ser marcado como apropiado para ser revisado y aprobado.




Vamos a aceptarlo:

[Code](#) [Issues](#) [Pull requests](#) [Actions](#) [Projects](#) [Wiki](#) [Security](#) [Insights](#) [Settings](#)


Ejemplo de pull request #1

Open xAGH wants to merge 2 commits into `main` from `develop`

Conversation 0 Commits 2 Checks 0 Files changed 2

 xAGH commented 1 minute ago Owner


De la rama develop se tomará el cambio en el archivo ejemplo.txt y se añadirá Otro.txt

 xAGH added 2 commits 43 minutes ago

Modificacion en develop de ejemplo.txt 605fd5b


Se agrega otro archivo mas 4181678

Add more commits by pushing to the `develop` branch on `xAGH/ramas_git`.

 This branch has no conflicts with the base branch

Merging can be performed automatically.

Merge pull request or view command line instructions.


 Write Preview

Leave a comment

Attach files by dragging & dropping, selecting or pasting them.

Close pull request Comment

Add more commits by pushing to the `develop` branch on `xAGH/ramas_git`.

 Aceptando el pull request

Me parecen correcto los cambios


Confirm merge Cancel

[Code](#) [Issues](#) [Pull requests](#) [Actions](#) [Projects](#) [Wiki](#) [Security](#) [Insights](#) [Settings](#)


Ejemplo de pull request #1

Merged xAGH merged 2 commits into `main` from `develop` now

Conversation 0 Commits 2 Checks 0 Files changed 2


 xAGH commented 9 minutes ago Owner


De la rama develop se tomará el cambio en el archivo ejemplo.txt y se añadirá Otro.txt

 xAGH added 2 commits 1 hour ago

Modificacion en develop de ejemplo.txt 605fd5b


Se agrega otro archivo mas 4181678

 xAGH merged commit e758b61 into `main` now Revert

 Pull request successfully merged and closed

You're all set—the `develop` branch can be safely deleted.

Delete branch

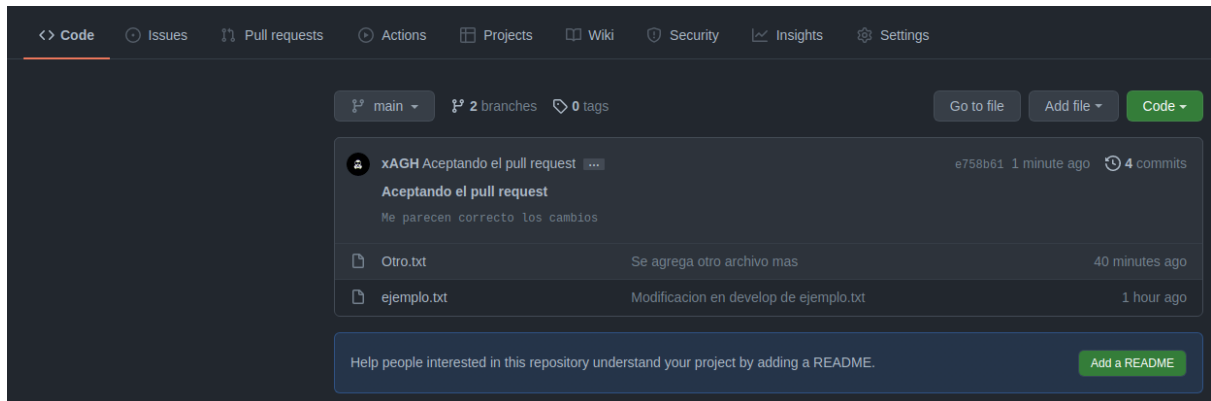
 Write Preview

Leave a comment

Attach files by dragging & dropping, selecting or pasting them.

Comment

El pull request ha sido aceptado y las ramas combinado:



Ya con esto hemos aprendido a hacer pull request, revisarlas y aceptarlas o rechazarlas.