



git init: Para iniciar el sistema de git y configurar parámetros necesarios. **git config**

--list: Veo las configuraciones de mi git

git config --global user.name "nombre"

git config --global user.email "correo"

git config --global core.editor nano

git status: Me muestra si los archivos están en Staging o todavía faltan subirlos.. **git add:** Agregar los archivos a Staging y quedan listos para el Commit.

git add . :Agregar todos los archivos.

git commit: Subo los archivos al repositorio de git y creo una versión nueva.

- **git commit -m "mensaje"**

git log: muestra la lista de los commits.

- **git log --stat:** muestra los cambios específicos

- **git log -p:** muestra las diferencias introducidas en cada confirmación.

git show: muestra los últimos cambios sobre un archivo.

git diff: compara lo que hay en tu directorio de trabajo con lo que hay en tu área de preparación. El resultado te indica los cambios que has hecho y que todavía no has preparado. Puedo ver los cambios entre un commit y otro de un archivo. Muestra los cambios que todavía no están preparados.

git rm --cached: Elimina los archivos del área de Staging y del próximo commit, pero los mantiene en el disco duro.

git rm --force: Elimina los archivos de git y del disco duro.

git reset: Me permite volver en el tiempo a otra versión.

- **git reset n° de commit --hard:** cambia todo y vuelve a la versión del commit indicado.

- **git reset n° de commit --soft:** Vuelve al commit indicado pero mantiene lo que hay en staging. No toca el archivo de índice o el árbol de trabajo en absoluto (pero reinicia la cabeza <commit>, al igual que todos los modos). Esto deja todos los archivos modificados como "Cambios para confirmar", como lo git status diría.

Ramas

git branch nombre de la rama: Crea una rama nueva distinta a Master con el último commit de master.

git checkout nombre de la rama: Para moverse a una rama en particular.

git merge: une ramas. Debo estar en master y desde allí hacer merge trayendo la rama que quiero fusionar.

Puede haber conflictos cuando se hace el merge si se modificaron las líneas iguales en ambas ramas.

Para ver el historial de todas las ramas:

`git show-branch --all`



Readme: Un archivo que explica que es el repositorio.

Agregar el origen del repositorio de github

git remote add origin ruta de github donde subiré mi repositorio local.

Clonar un proyecto de Github para usar localmente

Entramos al repositorio de Github que queremos clonar y luego hacemos clic en el botón verde <> Code y copiamos la dirección de la solapa HTTP. En la terminal escribimos **git clone** y pegamos la dirección del repo que copiamos.

git remote -v: muestra el origen para enviar y recibir el repositorio

git push origin master: envió la rama master al repositorio de github

git pull origin master: Traigo la actualización, si la hubiera, del repositorio de Github. Es una buena práctica.

git pull origin master --allow-unrelated-histories: Para fusionar las historias de los commit en caso de que haya algo anterior en el repo remoto y surja un error al traer esa data.

Trabajo colaborativo

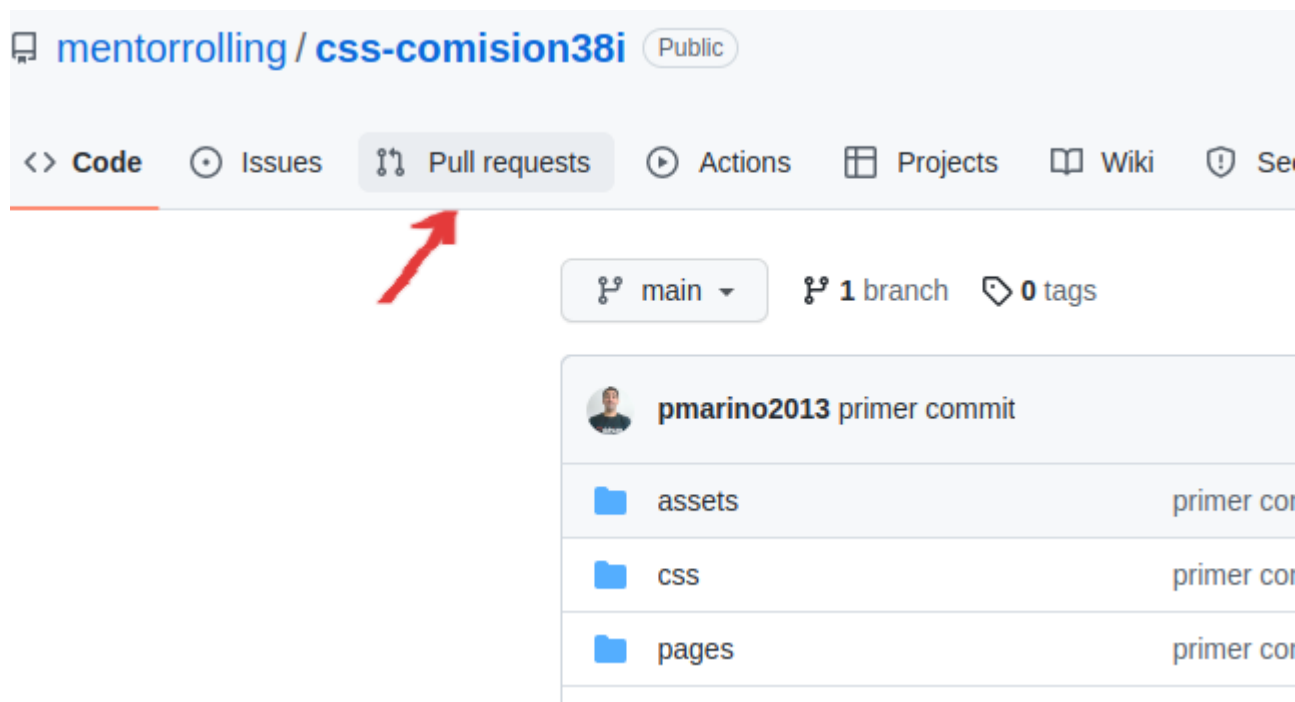
Si el repositorio es público cualquiera puede clonarlo, pero para subir los cambios con push se necesita agregar al usuario al repositorio.

Vamos a los *settings del repositorio*->*Manage access y agrego un colaborador*. La persona debe tener una cuenta en Github.

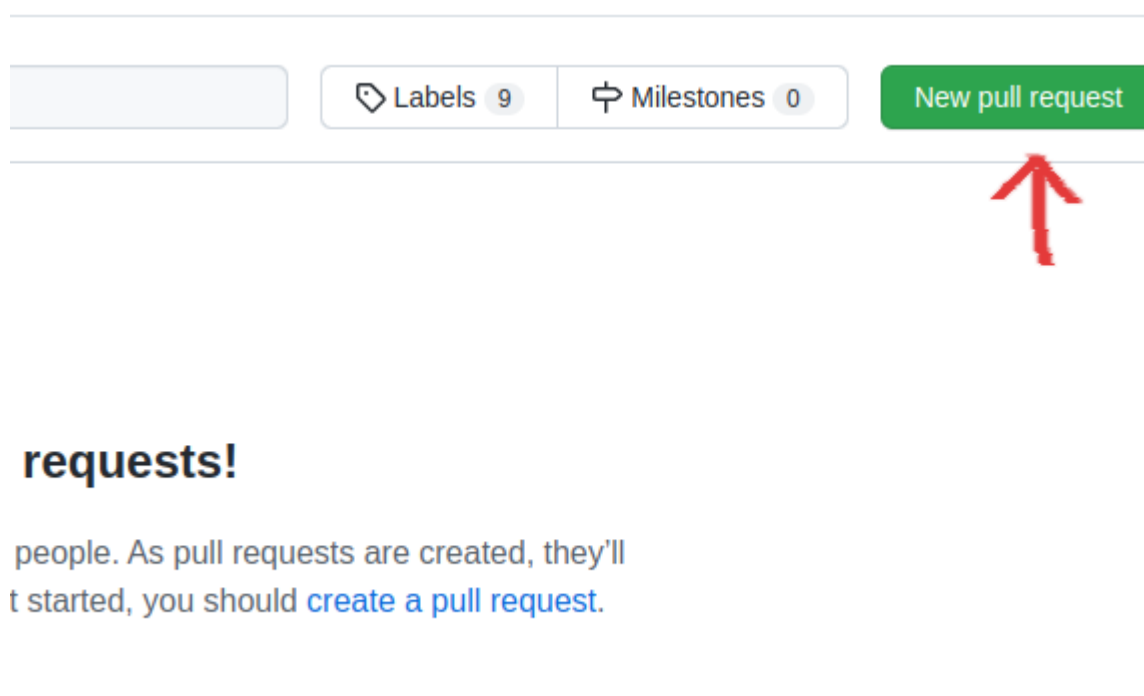
Le llegará un correo para que acepte la invitación.

Cómo hacer un Pull requests (PR)

Paso 1



Paso 2



Paso 3


En el margen superior elegir la rama de origen y la rama con la cuál queremos mergear o fusionar.

Si no hay ningún conflicto aparecerá un mensaje de color verde diciendo “*Able to merge*” lo que indica que se puede hacer la fusión de las ramas sin problema.




Una vez hecho eso tocamos en el botón *Create 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](#).

 base: main ← compare: Alejandro_Juan ✓ **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

 10 commits  15 files changed  1 contributor


Paso 4


A continuación escribimos un título al pull request y dejamos una descripción de lo realizado en la rama que vamos a unir a la rama principal para que quede constancia del porqué de la solicitud.

Presionamos el botón verde una vez finalizado.

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](#).

 base: main ← compare: Alejandro_Juan ✓ **Able to merge.** These branches can be automatically merged.




Tarea de actualización terminada

Write Preview

H B I  < >     @  

El diseño de los botones quedó como se había solicitado.

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


Create pull request


Paso 5

Una vez creado el pull request queda a la espera de que el encargado/a de verificar y aceptar las solicitudes le dé el OK para concluir el proceso y fusionar las ramas en cuestión.

Se pueden agregar mensajes de ser necesarios.


Add more commits by pushing to the **Alejandro_Juan** branch on **mentorrolling/javascript-22i**.










**This branch has no conflicts with the base branch**
Merging can be performed automatically.

Merge pull request

 or view [command line instructions](#).




WritePreview

H B I      @  

Leave a comment

Completa este campo

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

 Close pull request

Comment