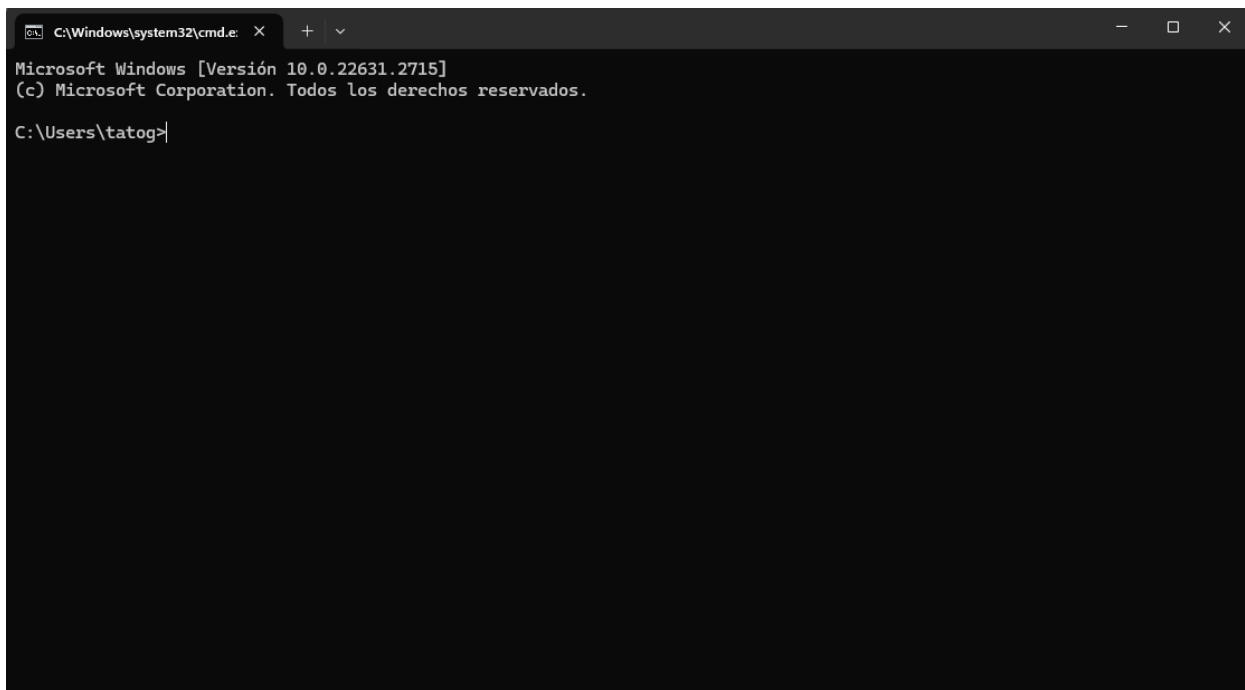


# Git

Es importante descargar Git, al descargarlo se obtienen los comandos necesarios para utilizarlo en la consola. Estos comandos son esenciales para realizar tareas como la clonación de repositorios, la creación de ramas, la realización de confirmaciones (commits) y la sincronización con repositorios remotos.

## Consola

Lo principal es abrir una consola, claro está que va a cambiar para cuando lo hagamos en Linux pero los comandos vienen siendo los mismos que en Windows. Esto es más que nada para que tengan una base para empezar en GitHub.



ahora es necesario entrar a la carpeta donde está ubicado el archivo que se desea subir. para esto usaremos el siguiente comando

- cd ruta del archivo
- cd Desktop

```
C:\Windows\system32\cmd.e: X + v
Microsoft Windows [Versión 10.0.22631.2715]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\tatog>cd Desktop
C:\Users\tatog\Desktop>|
```

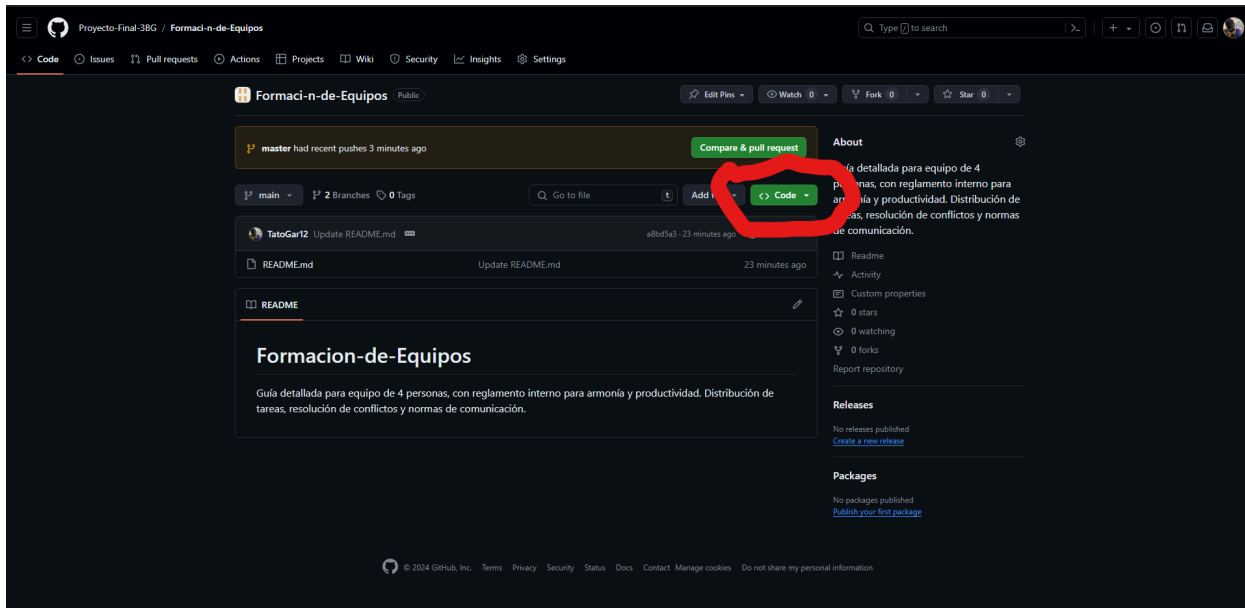
una vez que estemos en el mismo lugar de alojamiento de nuestro archivo, esto lo podemos confirmar con un tree (windows) o ls (linux) debemos iniciar git, eso se hace con el siguiente comando

- git init

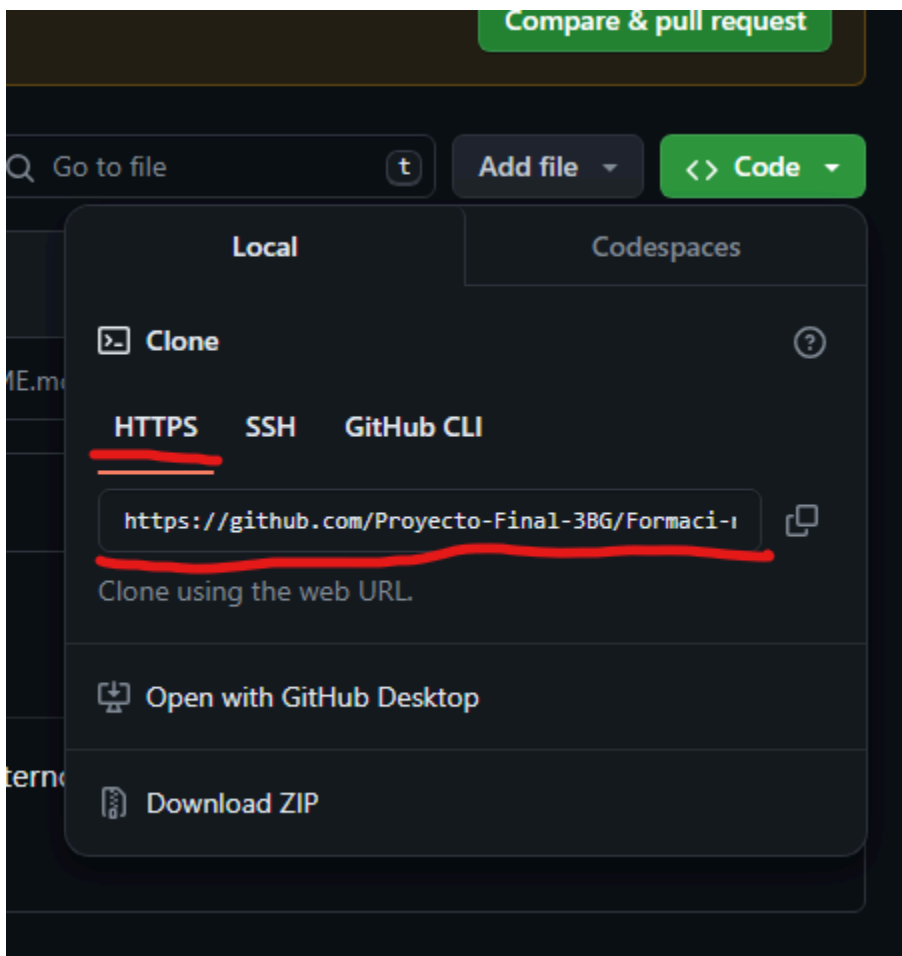
```
C:\Users\tatog>cd Desktop
C:\Users\tatog\Desktop>git init
Reinitialized existing Git repository in C:/Users/tatog/Desktop/.git/
C:\Users\tatog\Desktop>
```

si tenemos git instalado y colocamos el comando bien, nos saltaría la siguiente respuesta. una vez realizado esto, necesitamos conectarnos al directorio que esta en github. en windows se hace con su link y en linux con una clave ssh.

para obtener estas, vamos a nuestro repositorio en la nuba y le damos al siguiente boton.



este nos abra un pequeño panel con varias opciones, como lo estamos haciendo en windows tomaremos el link.



con esto en mano, volvemos a la consola y nos conectamos de manera remota al directorio con el siguiente comando:

- `git remote add origin <link copiado previamente>`

```
C:\Users\tatog\Desktop\Git>git remote add origin https://github.com/Proyecto-Final-3BG/GitGub.git
C:\Users\tatog\Desktop\Git>
```

Nuestro amigo git, es un hombre de pocas palabras, pero no te preocupes, si no te sale nada es porque lo hiciste bien.

ahora procederemos a añadir nuestro archivo o carpeta que deseamos subir, aca hay varias formas

podemos ingresar el nombre del archivo o ubicar un punto para marcar todos los archivos

- `git add <nombrearchivo>` este solo suma el archivo que hayas ingresado
- `git add .` este suma todos los archivos que hayan en la carpeta git

por lo momento solo usaremos el primer ejemplo

```
C:\Users\tatog\Desktop\Git>git add GitHub-Tutorial
C:\Users\tatog\Desktop\Git>git status
```

como git no nos dice nada, podemos usar el comando `git status` para saber si la carpeta o archivo se sumo a lista.

- `git status`

```
C:\Users\tatog\Desktop\Git>git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   GitHub-Tutorial/Prueba.txt
```

el echo que este en verde nos indica que lo hicimos de manera correcta, en caso de estar en rojo significa que no esta en la cola para subirse.

ahora, necesitamos ubicar los commits, esto es algo sencillo, aunque necesitamos saber que son, vienen siendo como las aclaraciones que van a ir con el archivo a la hora de subirlo, por ejemplo, si cambiaste el nombre del archivo, el commit seria algo como “se cambio el nombre del archivo”, es un registro basico por así decirlo, todos podrán verlo.

entonces, usaremos el siguiente comando:

- `git commit -m “Hola mundo”`

```
C:\Users\tatog\Desktop\Git>git commit -m "hola mundo"
[master (root-commit) 1aa5089] hola mundo
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 GitHub-Tutorial/Prueba.txt
```

en caso de hacerlo bien, nos marcara a que archivo se le dio dicho comentario así como el comentario que se dio.

ahora hay que especificar el gmail y el usuario que tenemos para subirlo a github

- git config --global user.email "tu\_email@gmail.com"
- git config --global user.name "Tu Nombre"

usando estos dos comandos, en uno pondremos el correo de git, y en el otro el usuario de git.

```
C:\Users\tatog\Desktop\Git>git config --global user.email "facunddogarcia@gmail.com"
C:\Users\tatog\Desktop\Git>git config --global user.name "TatoGar12"
```

no nos dira mucho, y eso es una buena señal

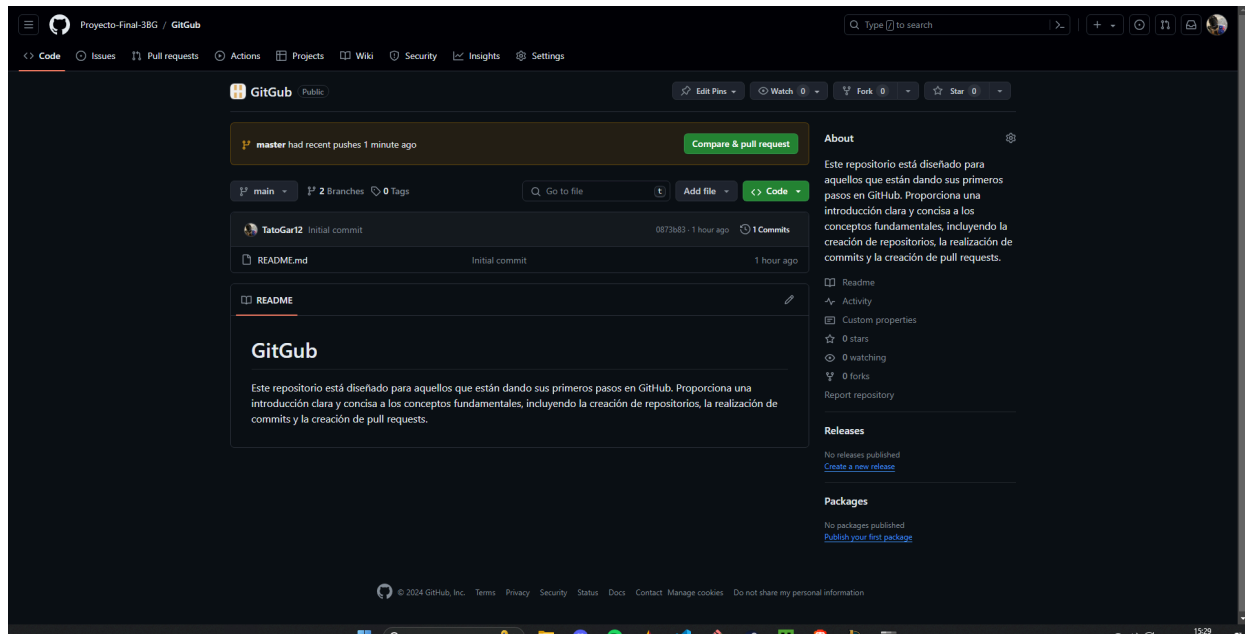
ahora necesitamos hacer el push, el primer push se realiza de la siguiente forma:

- git push --set-upstream origin master

donde dice origin master viene siendo la rama a donde ira, por el momento vamos a dejarla en master aunque tambien esta main.

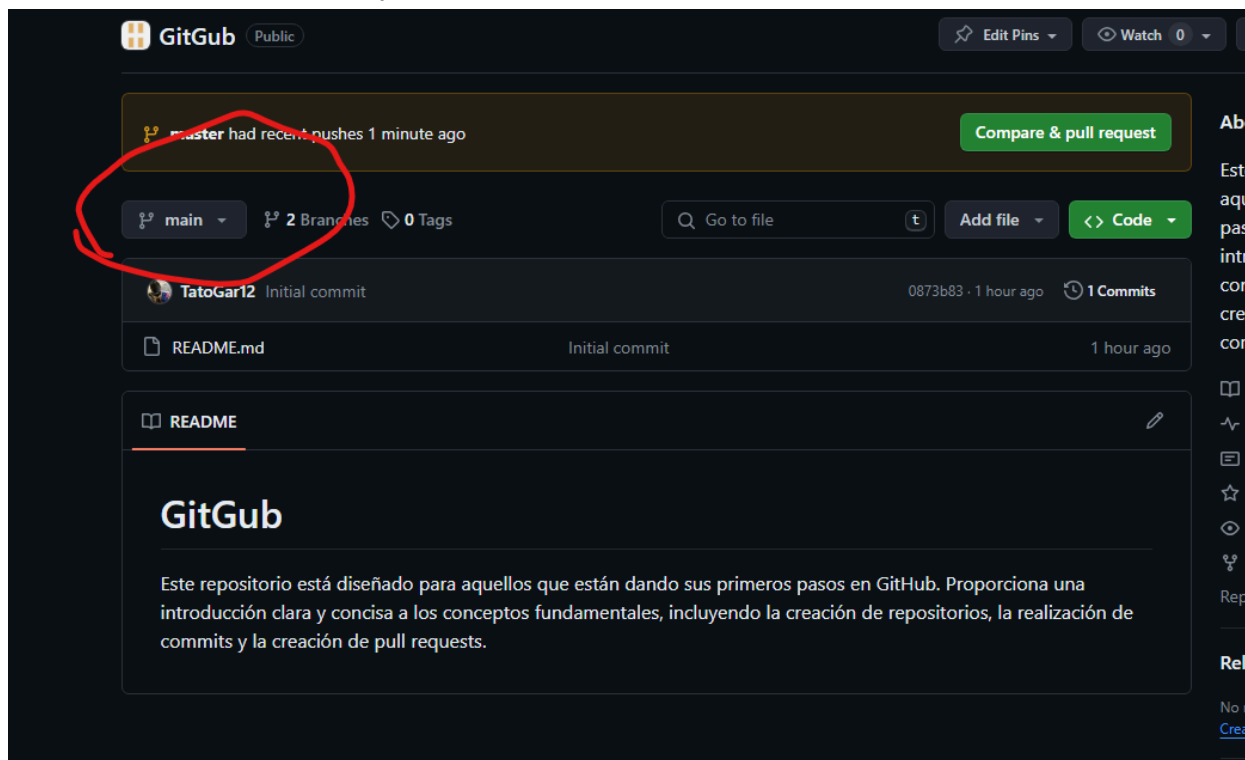
```
C:\Users\tatog\Desktop\Git>git push --set-upstream origin master
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Writing objects: 100% (4/4), 266 bytes | 266.00 KiB/s, done.
Total 4 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
remote:
remote: Create a pull request for 'master' on GitHub by visiting:
remote:   https://github.com/Proyecto-Final-3BG/GitGub/pull/new/master
remote:
To https://github.com/Proyecto-Final-3BG/GitGub.git
 * [new branch]      master -> master
branch 'master' set up to track 'origin/master'.
```

en caso de hacerlo bien, nos tira esto de manera directa, significa que nuestro archivo ya se subio al directorio en la rama master, aunque tambien puede que te tire unas lineas adicionales que te pedirá tu correo de git así como la contraseña para poder ingresar a github.

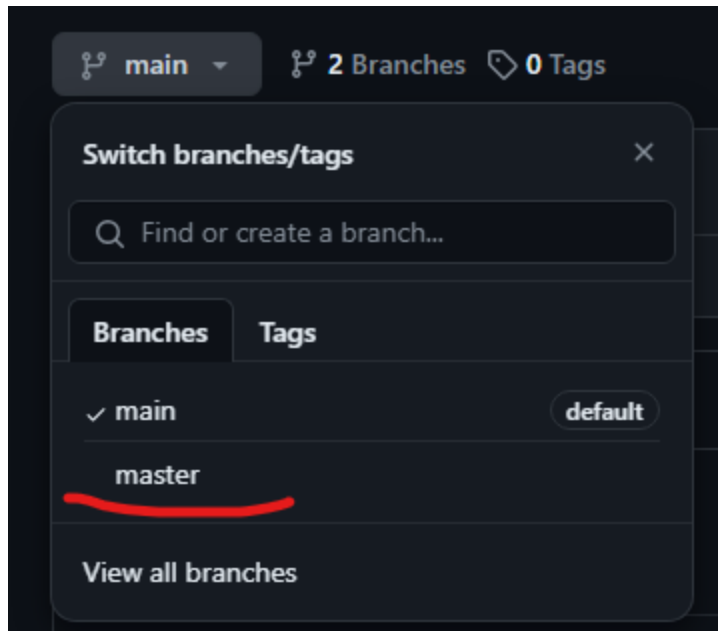


como ven, una ves aca no nos muestra nuestro archivo, eso es porque lo hicimos en la rama “master” y estamos en la main

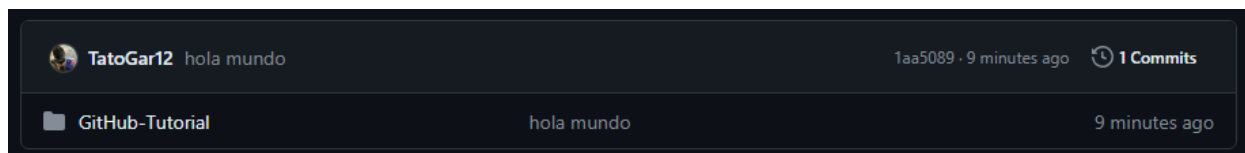
para cambiar de rama es muy simple.



damos click, y nos despliega un pequeño manel, donde tocaremos la rama “master”



al marcar la rama master nos tendría que aparecer el archivo que subimos en dicha rama.



como ven aca tenemos nuestro archivo ya subido.