










 ManuelGTM / Sistema-Ventas



 Code  Issues  Pull requests  Actions  Projects  Security  Insights





☆ 0 stars  0 forks  1 watching  Activity

 Public repository

 main ▾



 Branches  Tags

 ManuelGTM ... 11 hours ago 

[View code](#)

 README.md

Sistema-Ventas

Manuel Gabriel Torres Malena 20-1346
Anthony C. Mena 14-0470

Pasos para usar git y github

Comandos para configurar git y github por el metodo SSH

1. Creacion de una cuenta en github

[Github](#)

2. Descargar git bash

[GitBash](#)

Comandos basicos de git para activar github y git en tu computador

OJO: Todos estos comandos se deben ejecutar por git bash

1. Informacion personal

- El nombre de usuario y el email van entre comillas dobles

```
git config --global user.name "Tu nombre de usuario"
git config --global user.email "Email usado en la cuenta de github"
```

2. Cambiar el nombre de la rama master a main

```
git config --global init.defaultBranch main
```

3. Activar los colores de la terminal

```
git config --global color.ui auto
```

4. Configurar el comportamiento de la rama principal

```
git config --global pull.rebase false
```

5. Confirmar si todo esta funcionando correctamente

- Se debe correr uno a la vez y este debe devolver tu nombre de usuario y el email de github.

```
git config --get user.name
```

-- enter y luego este

```
git config --get user.email
```

6. Creacion de la llave SSH

- Copiar el comando pero reemplazar el email de ejemplo con tu email

```
ssh-keygen -t ed25519 -C Tuemail@email.com <- reemplazalo con tu email de github
```

- te dira que lo guardes en un lugar, pero solo dale a enter hasta que te deje salir

7. Copiar tu llave ssh

```
cat ~/.ssh/id_ed25519.pub
```

- Te debera salir una llave criptografica con tu email al final como esta:

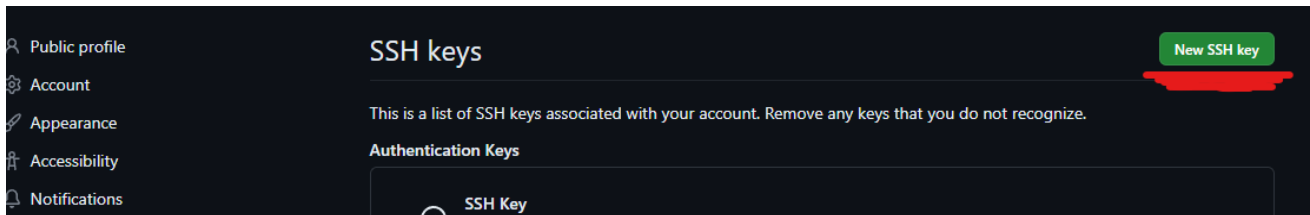
```
ssh-ed25519 AAAAC3NzaC1SRSTRSTRSTRTiQBc4P21jkXqasrtasrtdars3417yy1349877132M  
tuemail@email.com
```

- Copia esa llave con en email tambien

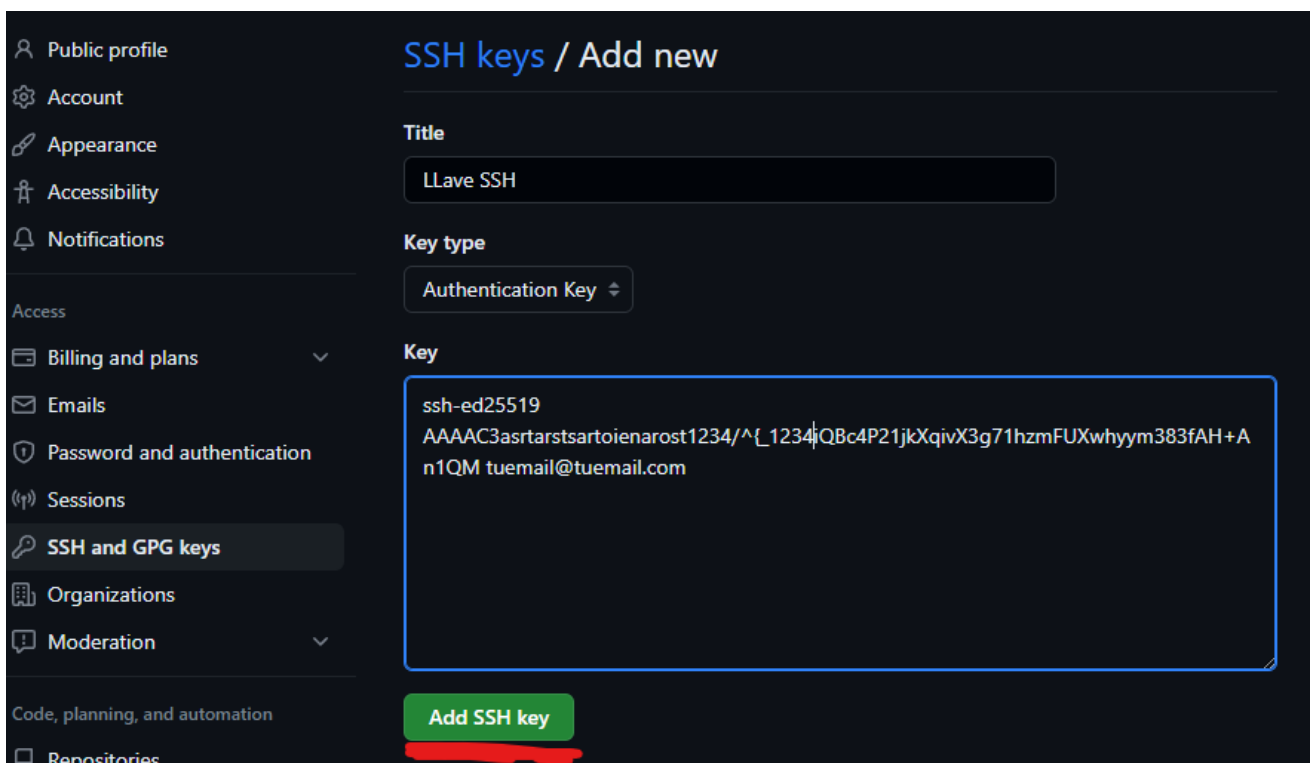
8. Ir a Github y guardar la llave

- [GitHub](#)

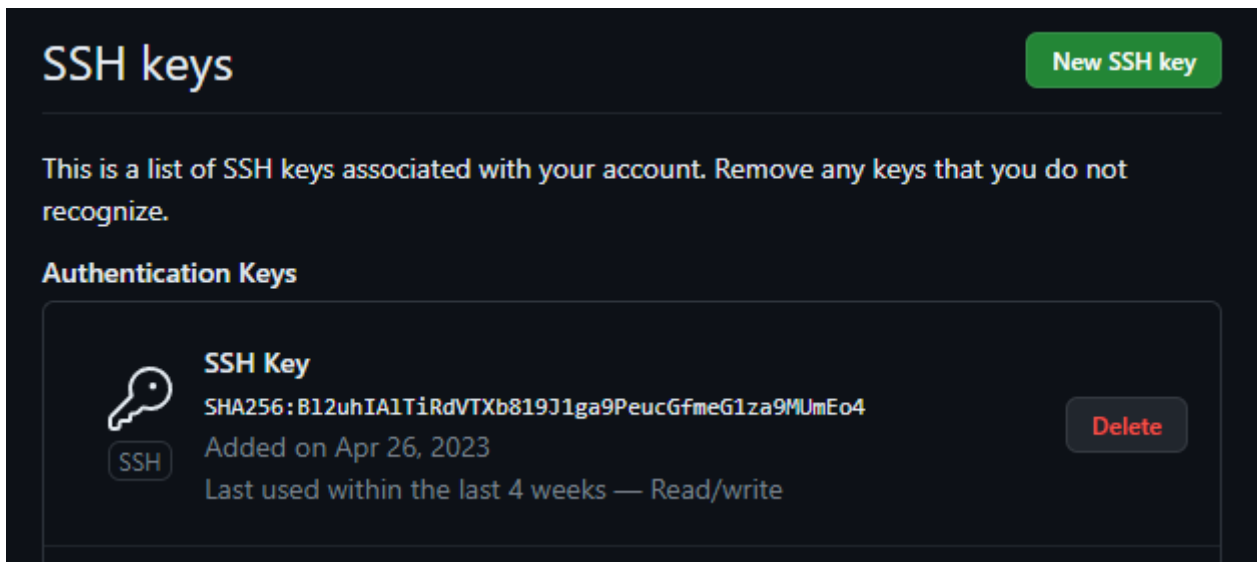
Pulsas el boton de agregar nueva llave



Pon un nombre a tu llave, pega la llave criptografica y la guardas



Tu llave ha sido guardada con exito



9. Prueba de la llave SSH

```
ssh -T git@github.com
```

Te debera salir:

The authenticity of host 'github.com (IP ADDRESS)' can't be established. ED25519 key fingerprint is SHA256:+DiY3wvvV6TuJJhbpZisF/zLDA0zPMSvHdkr4UvCOqU. Are you sure you want to continue connecting (yes/no)? <-- Le escribes yes

Y te saldra esto:

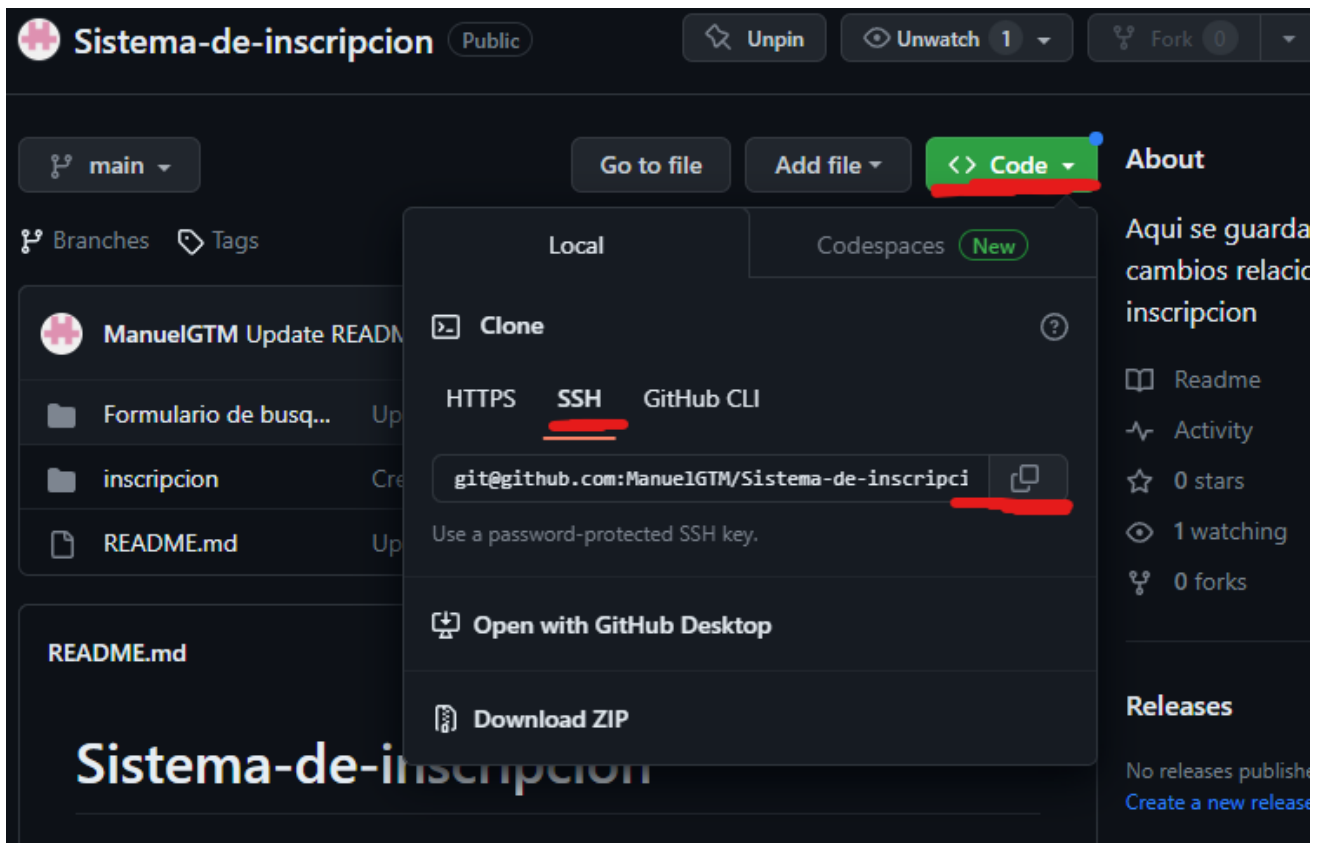
Hi (Tu nombre de usuario) You've successfully authenticated, but GitHub does not provide shell access.

Si esto salio, significa que has conectado con exito tu computador con github por el metodo SSH

Pasos para clonar un repositorio y realizar cambios

1. Copiar el enlace SSH del repositorio

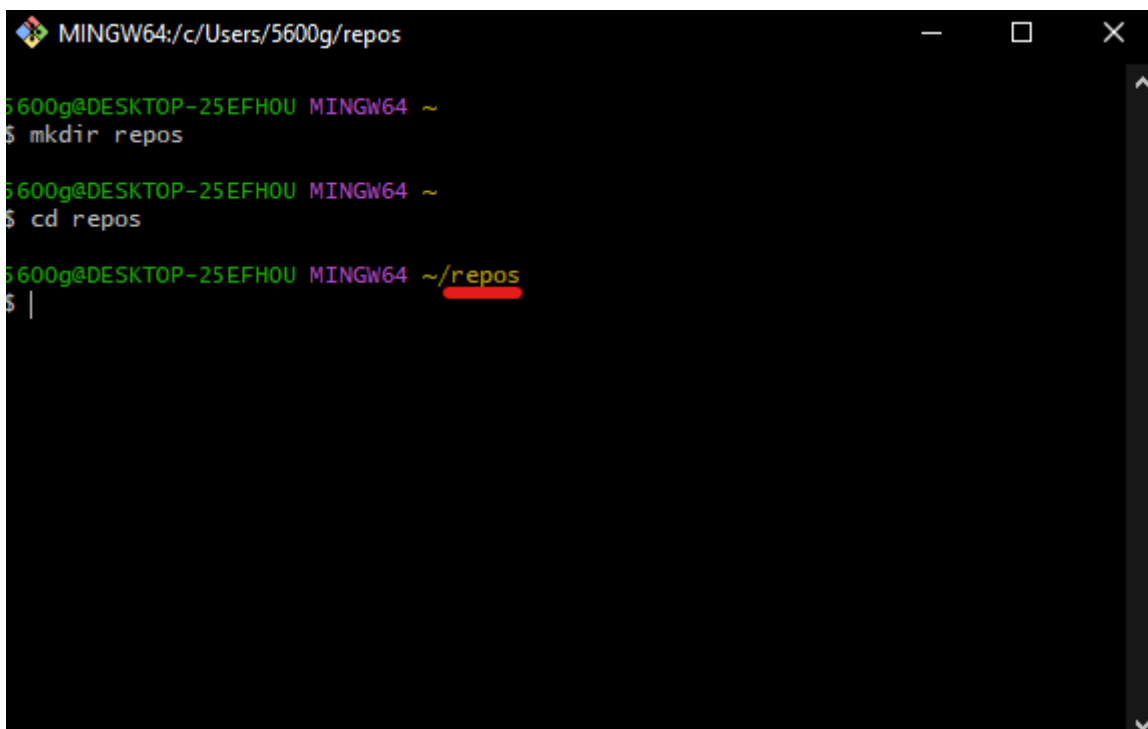
- Ir al boton que es de color verde que dice code
- Ojo: Dentro hay varias opciones pero elije la opcion SSH y copia el enlace



2. Crear una carpeta destinada a guardar el repositorio

Dentro de git bash ejecuta estos comandos

```
cd ~                <-- ir a la direccion principal
mkdir repos        <-- creacion de una carpeta --> mkdir ( make directory ) = Crear
carpeta
cd repos/          <-- entrar a la carpeta creada --> cd ( change directory ) =
cambiar de carpeta
```



- Si te das cuenta debe aparecer el nombre de tu carpeta en el lado derecho, esto significa que estas dentro de la carpeta deseada

3. Clonar el repositorio en la carpeta

Ejecuta este comando

```
git clone ( Aqui va el enlace que copiaste )
```

```
5600g@DESKTOP-25EFH0U MINGW64 ~/repos
$ git clone git@github.com:ManuelGTM/Sistema-de-inscripcion.git
Cloning into 'Sistema-de-inscripcion'...
remote: Enumerating objects: 232, done.
remote: Counting objects: 100% (232/232), done.
remote: Compressing objects: 100% (122/122), done.
remote: Total 232 (delta 99), reused 198 (delta 76), pack-reused 0
Receiving objects: 100% (232/232), 2.58 MiB | 746.00 KiB/s, done.
Resolving deltas: 100% (99/99), done.

5600g@DESKTOP-25EFH0U MINGW64 ~/repos
$ |
```

ls <--- Comando para ver los archivos dentro de la carpeta

- Te saldra algo como esto y de esta manera ya tienes el repositorio clonado dentro de tu computadora y listo para ser modificado

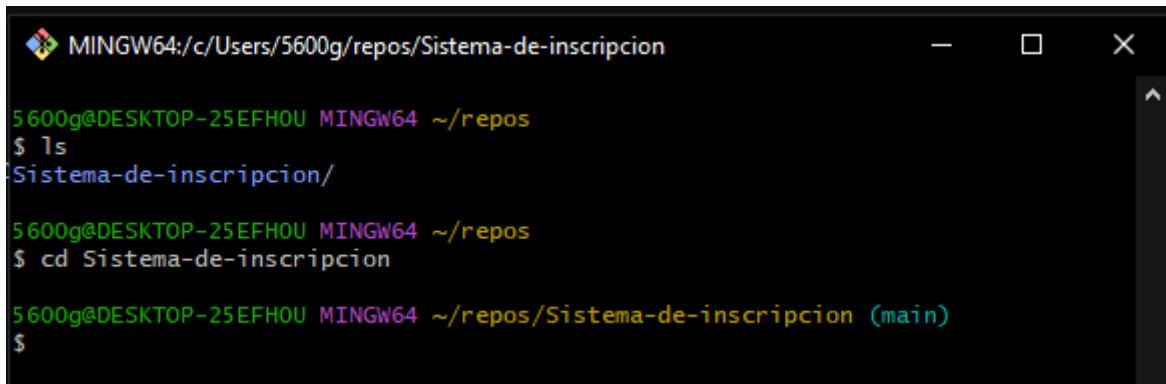
```
5600g@DESKTOP-25EFH0U MINGW64 ~/repos
$ ls
Sistema-de-inscripcion/

5600g@DESKTOP-25EFH0U MINGW64 ~/repos
$
```

Hacer modificaciones y subirlas al repositorio

Ojo: Debes entrar a la carpeta del repositorio con cd

```
cd ( nombre de la carpeta )
```

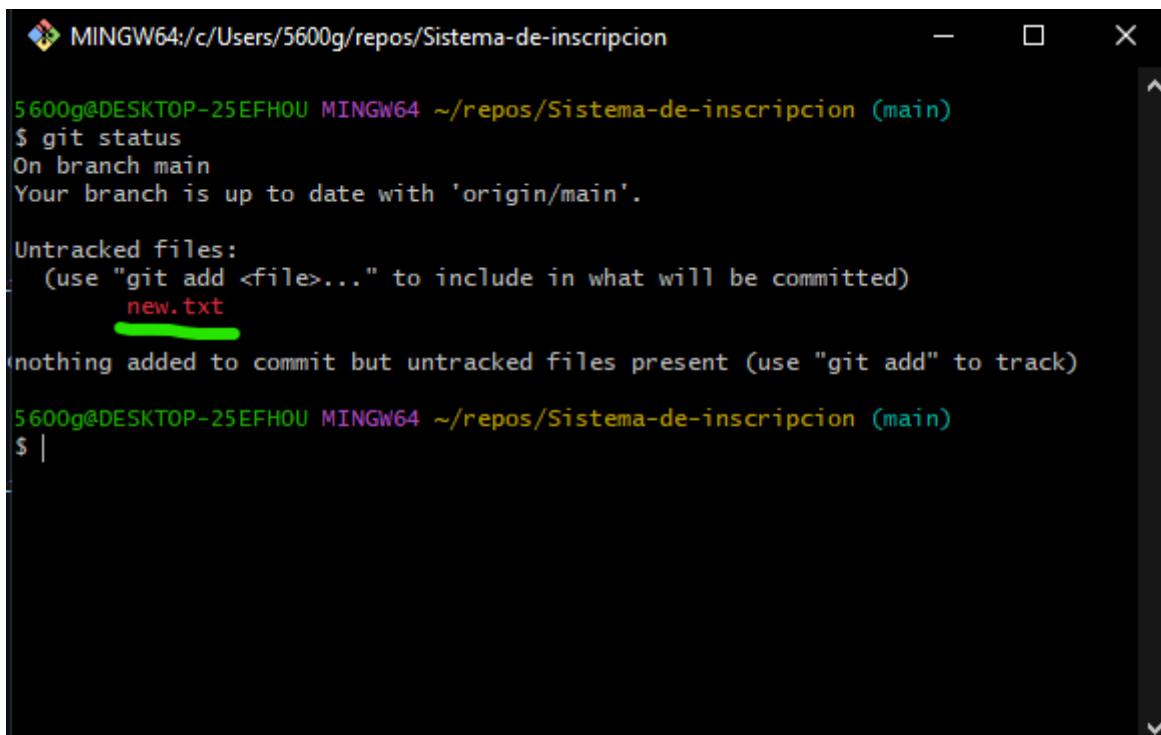


```
MINGW64:/c/Users/5600g/repos/Sistema-de-inscripcion
5600g@DESKTOP-25EFH0U MINGW64 ~/repos
$ ls
Sistema-de-inscripcion/
5600g@DESKTOP-25EFH0U MINGW64 ~/repos
$ cd Sistema-de-inscripcion
5600g@DESKTOP-25EFH0U MINGW64 ~/repos/Sistema-de-inscripcion (main)
$
```

1. Ejecutar este comando cada vez que hagas un cambio en los archivos

```
git status
```

- Este comando se usa para ver el estado de los archivos



```
MINGW64:/c/Users/5600g/repos/Sistema-de-inscripcion
5600g@DESKTOP-25EFH0U MINGW64 ~/repos/Sistema-de-inscripcion (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

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

nothing added to commit but untracked files present (use "git add" to track)
5600g@DESKTOP-25EFH0U MINGW64 ~/repos/Sistema-de-inscripcion (main)
$ |
```

- Como puedes ver el archivo esta de color rojo lo cual significa que ha sido modificado o creado pero no esta listo para ser enviado

2. Agregar este archivo al area de envio

```
git add .
```

- Este comando se usa para agregar los archivos al area de envio

Luego ejecutamos de nuevo `git status` para verificar el estado de archivo

```
5600g@DESKTOP-25EFH0U MINGW64 ~/repos/Sistema-de-inscripcion (main)
$ git add .

5600g@DESKTOP-25EFH0U MINGW64 ~/repos/Sistema-de-inscripcion (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   new.txt

5600g@DESKTOP-25EFH0U MINGW64 ~/repos/Sistema-de-inscripcion (main)
$ |
```

- Como puedes ver ahora nuestro archivo esta de color verde lo cual significa que esta listo para ser enviado

3. Guardar nuestro archivo en nuestro repositorio local

```
git commit -m "Un mensaje explicando lo que agregaste"
```

```
MINGW64:/c:/Users/5600g/repos/Sistema-de-inscripcion
5600g@DESKTOP-25EFH0U MINGW64 ~/repos/Sistema-de-inscripcion (main)
$ git commit -m "Agregando una archivo nuevo"
[main 7ad9f6a] Agregando una archivo nuevo
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 new.txt

5600g@DESKTOP-25EFH0U MINGW64 ~/repos/Sistema-de-inscripcion (main)
$ |
```

Luego ejecutamos de nuevo `git status` para verificar el estado de archivo

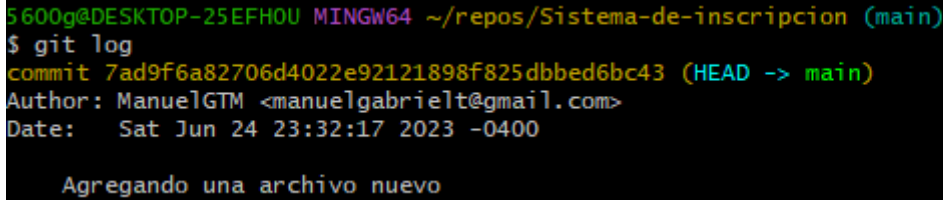
```
5600g@DESKTOP-25EFH0U MINGW64 ~/repos/Sistema-de-inscripcion (main)
$ git status
On branch main
Your branch is ahead of 'origin/main' by 1 commit.
  (use "git push" to publish your local commits)

nothing to commit, working tree clean
```

Esto significa que el archivo ha sido guardado en el repositorio local y ya no hay mas nada que agregar

4. Consultamos el historial de commits

```
git log
```



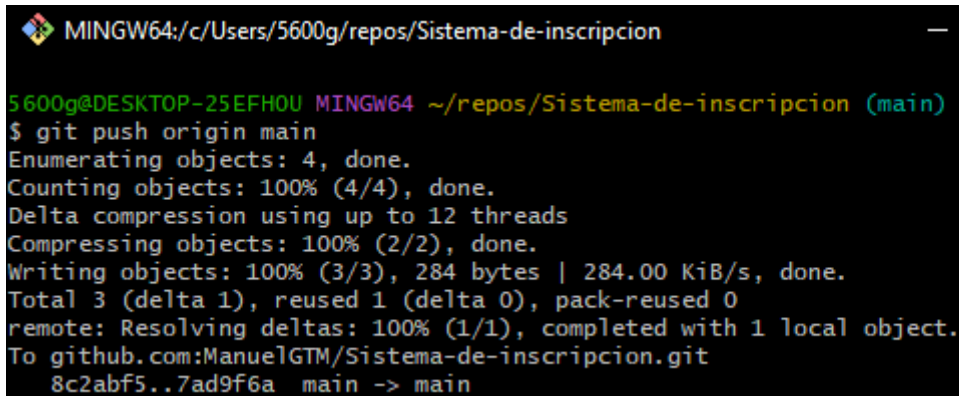
```
5600g@DESKTOP-25EFH0U MINGW64 ~/repos/Sistema-de-inscripcion (main)
$ git log
commit 7ad9f6a82706d4022e92121898f825dbbed6bc43 (HEAD -> main)
Author: ManuelGTM <manuelgabriel@gmail.com>
Date: Sat Jun 24 23:32:17 2023 -0400

    Agregando una archivo nuevo
```

- Al ejecutar este comando tendremos un reporte de todos los commits que hayamos realizado
- Tambien nos muestra quien fue el autor de este y cuando lo subio
- Para salir de este modo se presiona la letra **q**

5. Enviar nuestro cambios al repositorio

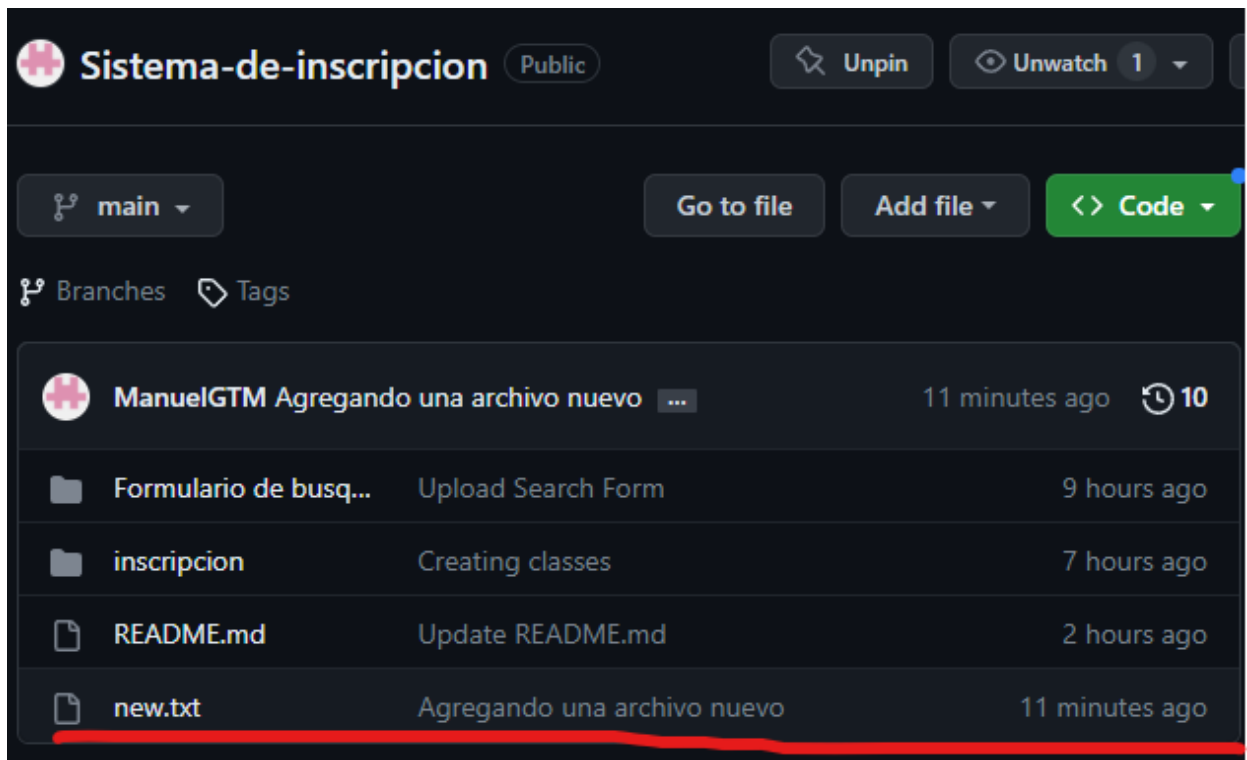
```
git push origin main
```



```
MINGW64:/c/Users/5600g/repos/Sistema-de-inscripcion

5600g@DESKTOP-25EFH0U MINGW64 ~/repos/Sistema-de-inscripcion (main)
$ git push origin main
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 12 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 284 bytes | 284.00 KiB/s, done.
Total 3 (delta 1), reused 1 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To github.com:ManuelGTM/Sistema-de-inscripcion.git
   8c2abf5..7ad9f6a  main -> main
```

- Con este comando enviamos todo los cambios a nuestro repositorio
- Y para ver nuestro archivo vamos al repositorio en github y confirmamos



Y aqui esta nuestro archivo en el repositorio guardado y seguro

6. Obtener los archivos modificados por un comapanero

```
git pull origin main
```

Con esto obtenemos los archivos que han sido cambiados por otro contribuidor y obtenemos sus cambios.

Releases

No releases published

Packages

No packages published

Languages

● C# 100.0%