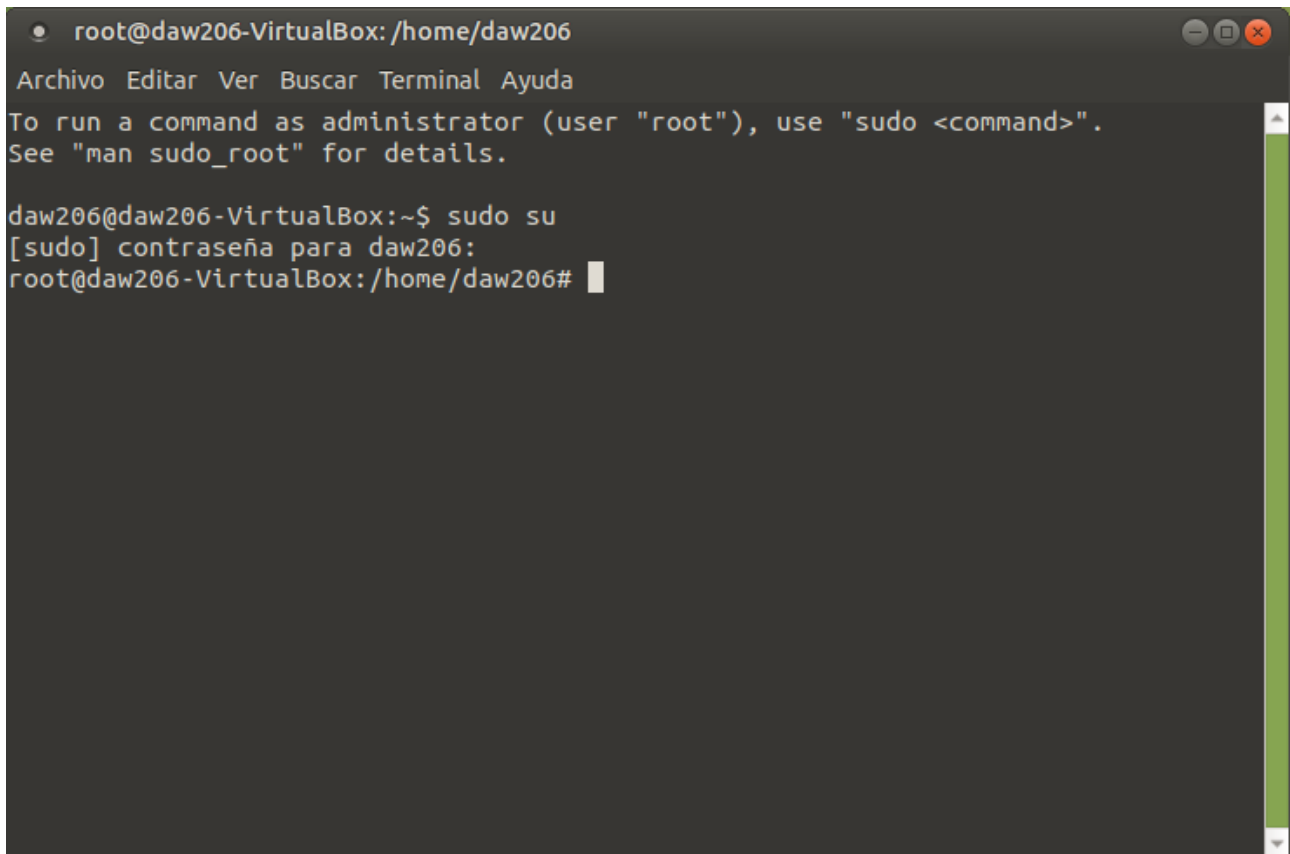


Instalacion de Git y su uso

Para realizar la instalación de git en una maquina de distribucion linux debian los pasos a realizar son los siguientes :

Accedemos como super usuario para tener los permisos de instalacion :

A screenshot of a terminal window titled 'root@daw206-VirtualBox: /home/daw206'. The window has a menu bar with 'Archivo', 'Editar', 'Ver', 'Buscar', 'Terminal', and 'Ayuda'. Below the menu bar, there is a message: 'To run a command as administrator (user "root"), use "sudo <command>". See "man sudo_root" for details.' The terminal shows the command 'daw206@daw206-VirtualBox:~\$ sudo su' being entered, followed by the prompt '[sudo] contraseña para daw206:'. The user has entered a password, and the prompt has changed to 'root@daw206-VirtualBox: /home/daw206#'.

```
root@daw206-VirtualBox: /home/daw206
Archivo Editar Ver Buscar Terminal Ayuda
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

daw206@daw206-VirtualBox:~$ sudo su
[sudo] contraseña para daw206:
root@daw206-VirtualBox: /home/daw206#
```

Lo siguientes es introducir la linea de comando que hace la referencia a la instalacion de git :

```
root@daw206-VirtualBox: /home/daw206# apt-get install git
```

aceptamos los terminos y comenzara con la instalacion

```

Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes adicionales:
  git-man liberror-perl
Paquetes sugeridos:
  git-daemon-run | git-daemon-sysvinit git-doc git-el git-email git-gui gitk
  gitweb git-cvs git-mediawiki git-svn
Se instalarán los siguientes paquetes NUEVOS:
  git git-man liberror-perl
0 actualizados, 3 nuevos se instalarán, 0 para eliminar y 169 no actualizados.
Se necesita descargar 4.731 kB de archivos.
Se utilizarán 33,9 MB de espacio de disco adicional después de esta operación.
¿Desea continuar? [S/n] █

```

Una vez realizado ya todo esto tendremos nuestro git instalado y podremos comenzar a usarlo

Comenzando con git

Para empezar con git debemos de crear un repositorio el cual contendrá unos archivos los cuales utilizara git y donde guardaremos nuestros archivos para compartir, para esto lo que haremos es posicionarnos en la carpeta en la que queramos crear el repositorio y utilizar el comando git init.

```

root@daw206-VirtualBox:/home/daw206# git init
Iniciado repositorio Git vacío en /home/daw206/.git/

```

Una vez tenemos nuestro repositorio creado lo siguiente, como es obvio, comenzar a introducir documentos/ trabajos que queramos controlar las versiones, en mi caso será un archivo prueba.txt, para llevar a cabo esto utilizaremos los siguientes comandos de la imagen:

```

root@daw206-VirtualBox:/home/daw206# git add prueba.txt
root@daw206-VirtualBox:/home/daw206# git commit -m 'version inicial'

```

Esto es que nos añadirá al repositorio el documento que vayamos a seguir su control y con el commit lo que nos hará es confirmar los posibles cambios que hemos realizado, decir que antes de hacer eso debemos crear una cuenta en github para que no nos dé ningún fallo, si todo ha salido bien nos aparecerá algo como esto

```

root@daw206-VirtualBox:/home/daw206# git commit -m 'version inicial'
[master (commit-raíz) 7f8e97d] version inicial
1 file changed, 1 insertion(+)
create mode 100644 prueba.txt
root@daw206-VirtualBox:/home/daw206# █

```

Pero como es normal no podemos tener todo en la misma raiz, ya que no tendremos ningun tipo de oranizacion, para esto se utilizan las ramas o subdirectorios que se crean de la siguiente forma:

```
root@daw206-VirtualBox:/home/daw206# git checkout -b pruebas
Cambiado a nueva rama 'pruebas'
```

Este comando lo que nos permitira es crear la rama que desamos y a su vez movernos para directamente comenzar a trabajar en ella si no, de otra forma seria utilizando las lineas

\$ git branch prueba

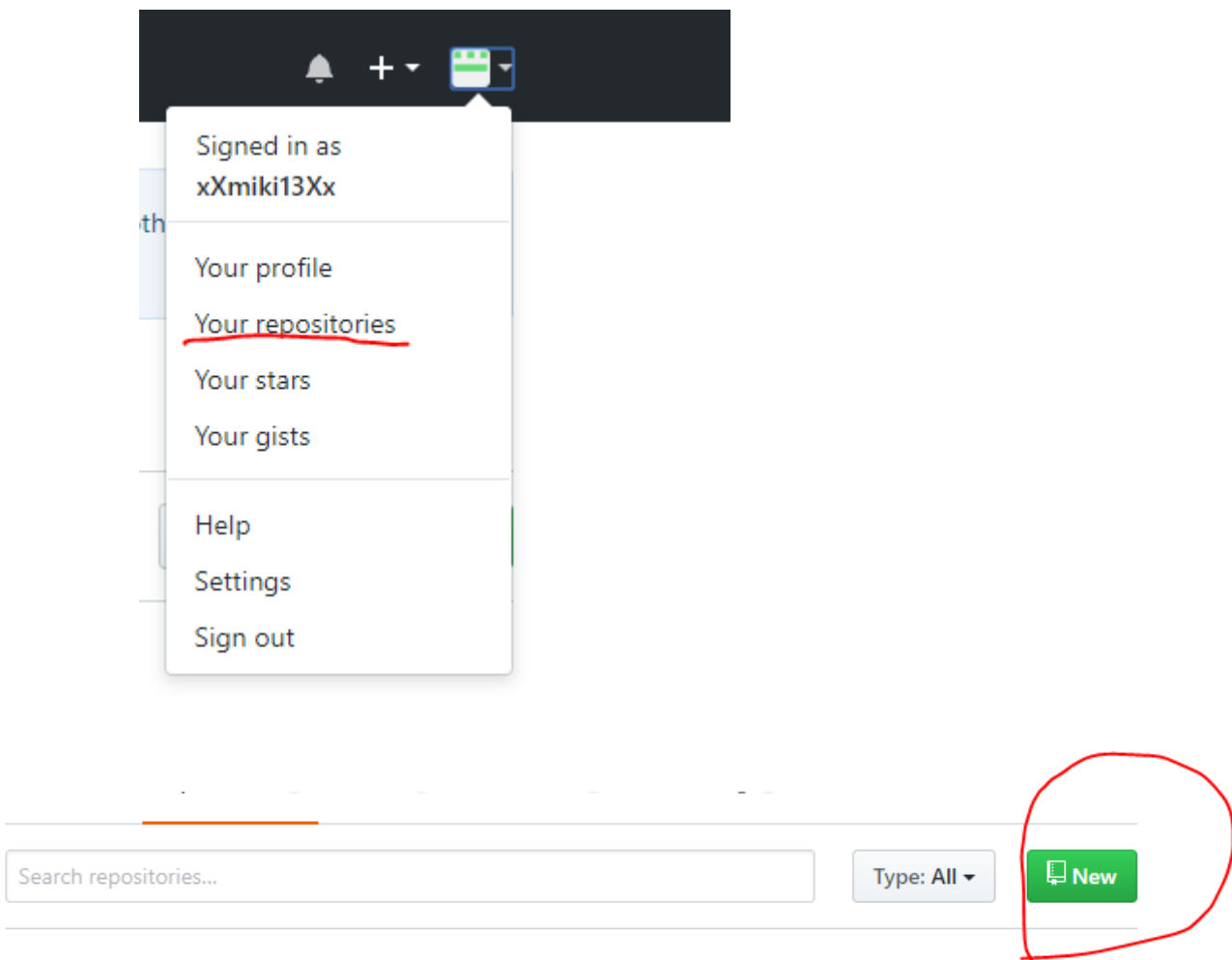
\$ git checkout prueba

que simplemente seria la version larga del checkout -b.


Comenzando GitHub

Una vez que ya hemos realizado todo de forma de comando lo utilizaremos de nuevo pero esta vez sera mediante GitHub que seria como un entorno grafico de lo que hemos visto hasta ahora pero en el cual aun necesitamos unos comandos, lo primero que debemos hacer es entrar en github.com y crear nuestra cuenta gratuita.

Despues de realizar la creacion de cuenta y la validacion crearemos nuestro primer repositorio para estos iremos a nuestro perfil > repositorios > nuevo repositorios




Owner


 xXmiki13Xx ▾

Repository name

Great repository names are short and memorable. Need inspiration? How about [fuzzy-octo-succotash](#).


Description (optional)

☒  **Public**
Anyone can see this repository. You choose who can commit.

☐  **Private**
You choose who can see and commit to this repository.

☐ **Initialize this repository with a README**
This will let you immediately clone the repository to your computer. Skip this step if you're importing an existing repository.

Add .gitignore: **None** ▾

Add a license: **None** ▾ 

Create repository

al poner el nombre y dando a crear repositorio ya nos mandara al repositorio para trabajar directamente, al estar vacio nos mostrara formas de agregar un trabajo a nuestro repositorio, desde linea de terminal seria de la siguiente forma añadiendo el archivo prueba.txt

```
root@daw206-VirtualBox:/home/daw206# git add prueba.txt
root@daw206-VirtualBox:/home/daw206# git remote add origin https://github.com/xXmiki13Xx/pruebas.git
root@daw206-VirtualBox:/home/daw206# git push -u origin master
```

despues de esto nos pedira un usuario y contraseña que corresponderan a nuestra cuenta de github

Pero para añadir/clonar el repositorio de otra persona ya sea porque vamos a trabajar con ella tendremos que acceder al proyecto que queremos/quieran compartir y seleccionar clone or download, nos dara una direccion la cual tendremos que dar si queremos que copien nuestro repositorio o recibir en caso de ser nosotros quien queramos clonar y una vez realizado tendremos que ejecutar en un terminal el siguiente comandos

```
git clone https://usuario@bitbucket.org/usuario/repositorio.git
```

y asi se nos añadiria a nuestro repositorio.

```
daw206@daw206:~/Despliegue$ git init
Inicializado repositorio Git vacío en /home/daw206/Despliegue/.git/
daw206@daw206:~/Despliegue$ git clone https://github.com/HannaDAW2/Proyecto.git
Clonando en 'Proyecto'...
remote: Enumerating objects: 29, done.
remote: Counting objects: 100% (29/29), done.
remote: Compressing objects: 100% (20/20), done.
remote: Total 29 (delta 5), reused 29 (delta 5), pack-reused 0
Desempaquetando objetos: 100% (29/29), listo.
daw206@daw206:~/Despliegue$
```

Después de realizar esto solo tendremos que entrar con github y tendremos nuestro repositorio clonado y ya podremos editarlo, la edición desde el terminal sería añadir el repositorio/ o archivo para el seguimiento de git con un 'git add archivo/repositorio' y después de ya una vez realizado cualquier cambio solo será necesario realizar un commit para aplicar los cambios y para que la otra persona simplemente realice un pull y obtenga el documento con todos los cambios que se hayan realizado


```
daw206@daw206:~/Despliegue/Proyecto/Desafio/public_html$ git add imagen.html
daw206@daw206:~/Despliegue/Proyecto/Desafio/public_html$ git commit -m 'cambio en el proyecto'
[master d1de2db] cambio en el proyecto
1 file changed, 2 insertions(+), 1 deletion(-)
```

Ya realizado esto solo nos quedará subir los cambios con un git push y la rama donde se llevarán los cambios, es decir en mi caso en el repositorio clonado sería:

```
git push origin master/Proyecto/public_html/
```

Y ya persona solo necesitará realizar un pull desde local o cuando entre a su cuenta en github.com le aparecerá en el repositorio del proyecto que se han realizado modificaciones y cuando

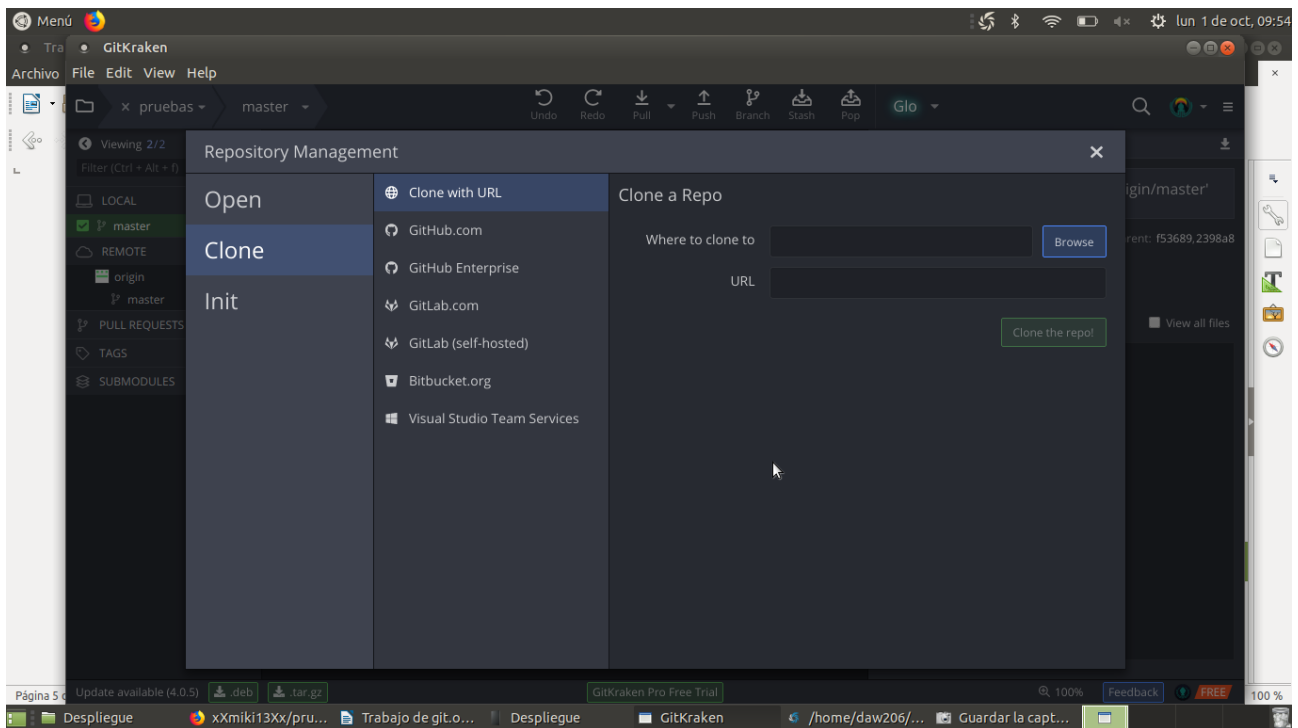
Ej:

 prueba.txt	Commit Hanna	7 minutes ago
--	--------------	---------------

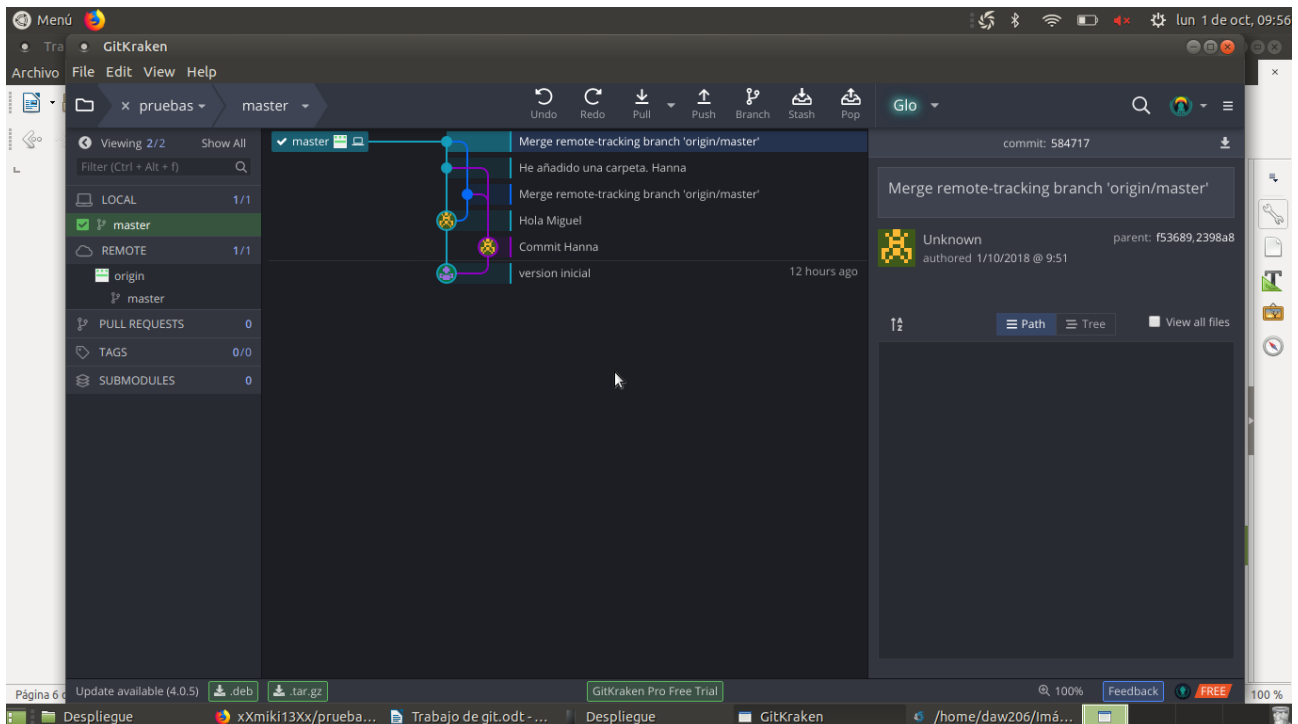
En caso que queramos eliminar un repositorio lo único que tenemos que hacer es lanzar el comando `rm -rf .git`

Utilizando gitKraken

Para utilizar gitkraken lo primero que debemos de hacer es clonar el repositorio

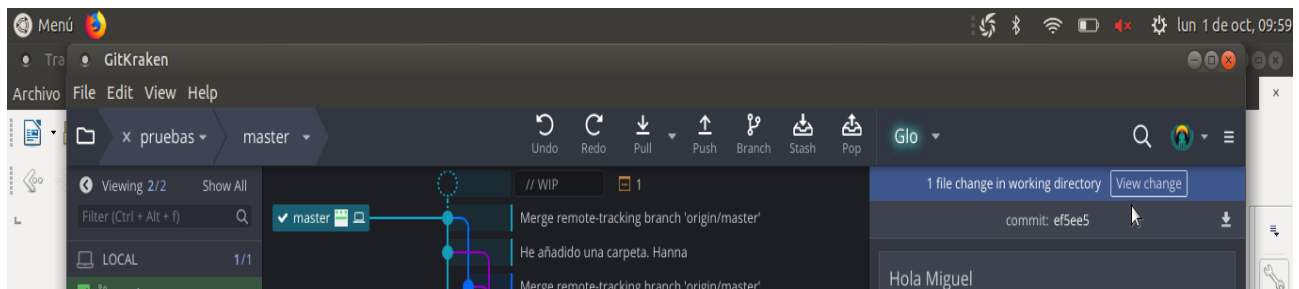


una vez que hemos realizado la clonacion despues de poner donde queremos clonarlo y desde que url, nos dara la opcion de abrir el directorio para empezar a trabajar, despues de esto nos saldra la estructura del repositorio

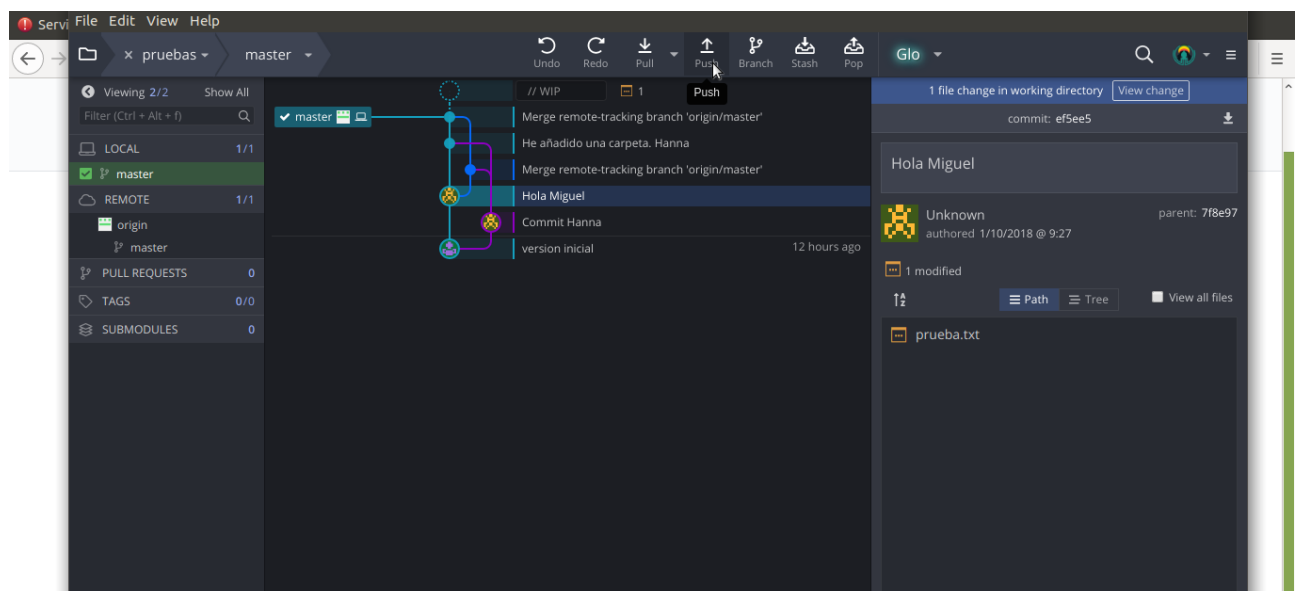


desde aquí ya podremos realizar cualquier cambio, solo tenemos que tener en cuenta de cuando realicemos los cambios mandar un push, recordar que se necesita tener privilegios de colaborador en el gitKraken.

Una vez se hayan realizado cambios se nos avisará de estos



y ya lo único que nos falta por hacer es realizar un push para enviar todos los cambios que hemos realizado

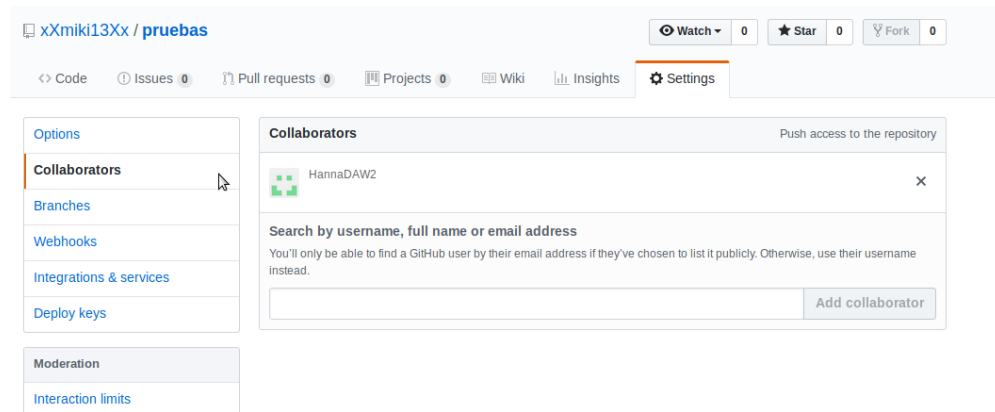
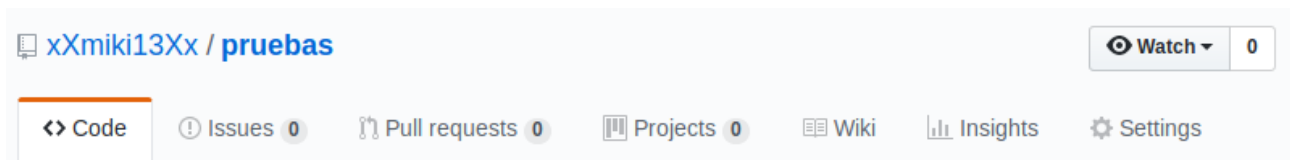


Para realizar un merge/ fusión en gitkraken es tan sencillo como seleccionar y arrastrar.

Detalles a tener en cuenta

Si vamos a compartir o vamos a trabajar en algún proyecto tenemos que tener presente que necesitaremos recibir o otorgar los privilegios de colaborador si no, no nos permitirá realizar un

push o nos saltara algun tipo de error o no se realizara, para esto debemos dirigirnos al proyecto, setting y en collaborators, buscamos al usuario al cual queremos otorgarle los permisos.



Y con esto se arreglaria el problema al mandar un push con los cambios realizados.

Recordar siempre antes de realizar el push, de realizar un commit para confirmar que los cambios que se han realizado se guarden en el documento