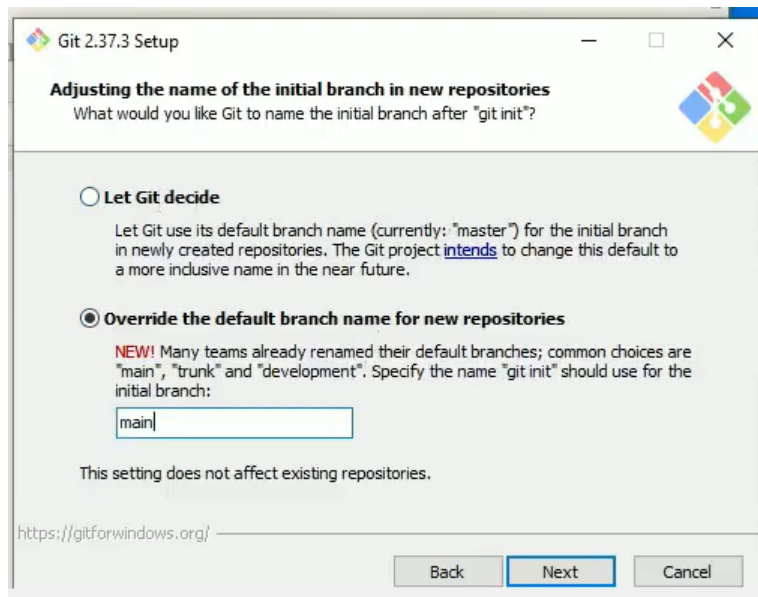


GIT

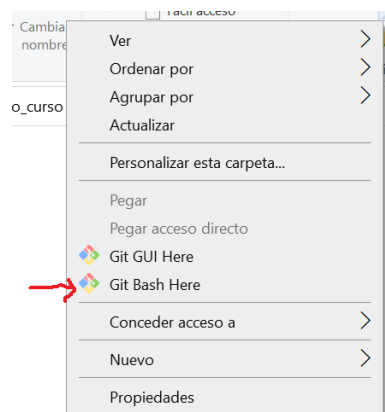
<https://www.youtube.com/watch?v=3iKiHUUsDgY>

Instalación

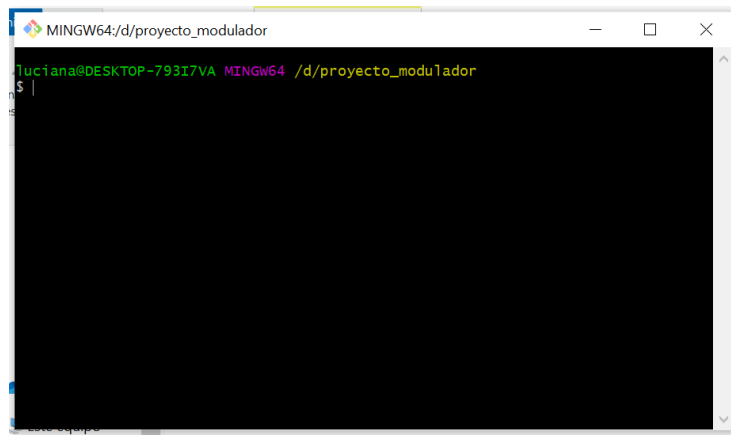
Descargar desde: <https://git-scm.com/downloads>. Instalar.



Para usarlo, una vez instalado, pararse en la carpeta donde esta el proyecto con el que quiero trabajar. Boton derecho del maus abrir Git Bash



Abre el Git Bash:



Configuracion inicial (se hace solo una vez)

Configuro el usuario:

git config --global user.name "luciana"

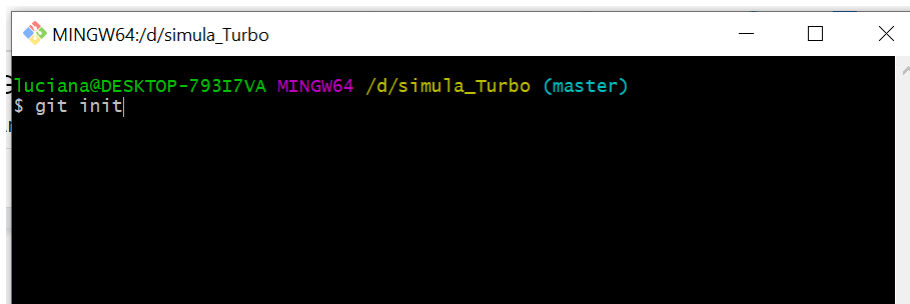
git config --global user.email "lucianadm55@gmail.com"

AYUDA: para pegar en el GitBash son las teclas SHIFT INSERT (en vez de ctrl v)

Crear un proyecto nuevo:

Abro el Git Bash en la carpeta del proyecto nuevo y pongo:

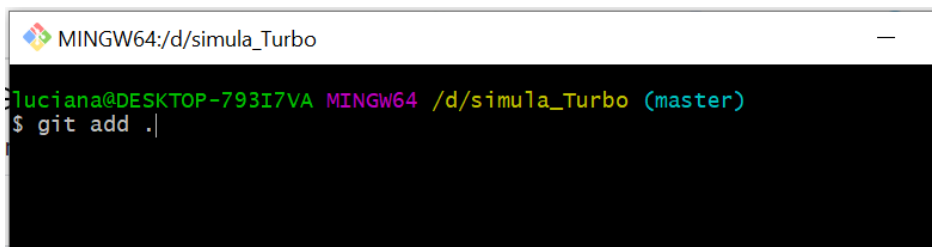
git init ⇒ Inicio el proyecto, esto se hace solo una vez para inicializar el proyecto



Hago las modificaciones en la carpeta, creo archivos, modifico archivos, etc

Despues hago los siguientes comandos:

git add . =>subir muchos archivos



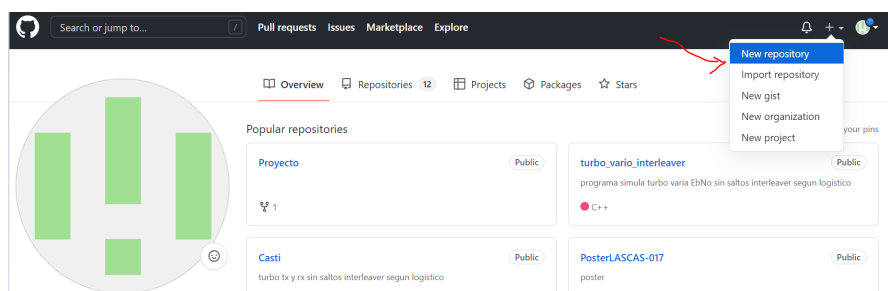
git commit -m "aca el mensj"=>para mandar al respoistorio del stagin

```
MINGW64:/d/simula_Turbo
luciana@DESKTOP-793I7VA MINGW64 /d/simula_Turbo (master)
$ git commit -m"cambie algo del programa"
```

Hasta aca fue todo en mi compu, ahora quiero mandarlo a la web al Github.

Asocio el proyecto con un repositorio de Github: (lo subo a la nube y despues siempre que modifiko algo subo la actualizacion)

Entro a Github (<https://github.com/login>) me logueo, saco usuario si no tengo, y voy a la cruz arriba a la derecha:




Me abre:

Create a new repository

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

Owner * Repository name *

 lucianadm /

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

Description (optional)

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

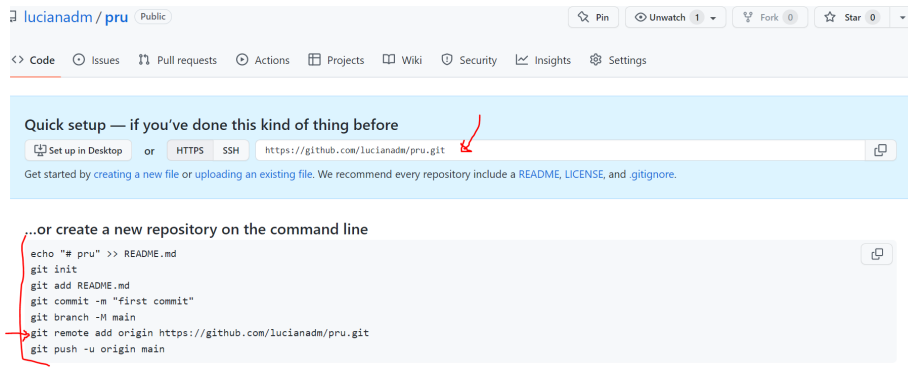
☐ **Private**
You choose who can see and commit to this repository.

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

New repository ⇒ Creo un nuevo repositorio.

Elijo el nombre, el resto se puede dejar como esta.

Abre esta ventana, copiar la direccion (flecha roja arriba) que vamos a usar en el GitBash para asociarlo con el proyecto que tenemos en la compu



Ahora de nuevo en el Gitbash:

git remote add origin <https://github.com/lucianadm/pru.git> ⇒ esto se hace una unica vez y ya queda asociado el github con el proyecto.(origin es el repositorio remoto)
git push origin main mando el proyecto. (mando main a origin)

=====

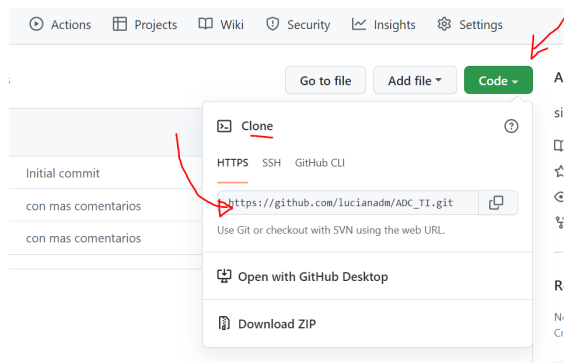
Cada vez que modifico algun archivo o agrego algo (para que se actualice en el Github):

git add .
git commit -m "aca el mensj"
git push origin main

=====

CLONAR UN REPOSITORIO:

Si quiero bajarme un proyecto de otro y hacer modificaciones y subirselas,
tengo que clonar el proyecto (bajarlo a mi pc y que quede asociado)
Busco la direccion en Github, en code copio la direccion:



Abro el GitBash en una carpeta donde quiero que se me copie la carpeta del proyecto y pongo:

git clone dir_q_copio_del_github
(para pegar en el gitbash SHIFT INSERT)

CUIDADO!!!! con el gitbash tengo que estar parado en la carpeta que bajé!

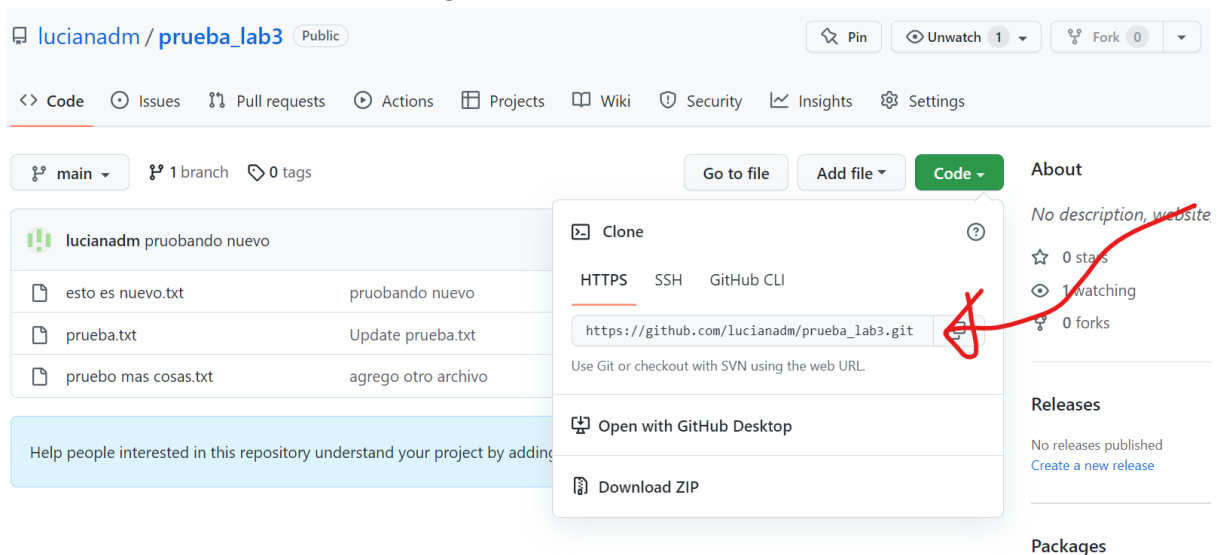
Ejemplo:

Estoy von el gitbash en el disco D:

```
luciana@DESKTOP-793I7VA MINGW64 /D
$
```

Clono desde github el repositorio Prueba_Lab3

Para ello copio la direccion del github de ese repositorio:



y desde el gitbah:

```
MINGW64:/D
luciana@DESKTOP-793I7VA MINGW64 ~ (master)
$ cd /D/

luciana@DESKTOP-793I7VA MINGW64 /D
$ git clone https://github.com/lucianadm/prueba_lab3.git
```

Me crea en el disco D una carpeta que se llama prueba_lab3 donde va a estar el repositorio.

Modifico lo que quiera de esa carpeta.

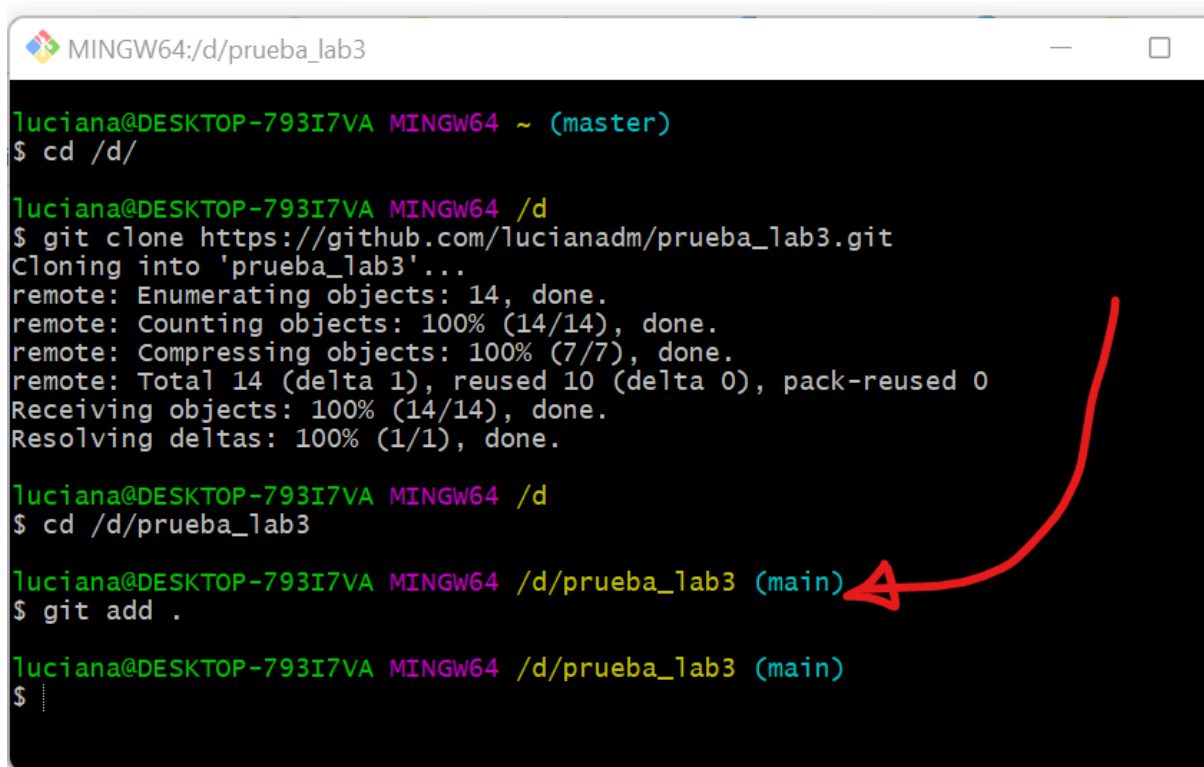
Ahora me tengo que parar con el gitbash en la carpeta prueba_lab3!!!

y recién ahí hago:

git add .

git commit -m"comento los cambios"

git push origin main



```
MINGW64:/d/prueba_lab3

luciana@DESKTOP-793I7VA MINGW64 ~ (master)
$ cd /d/

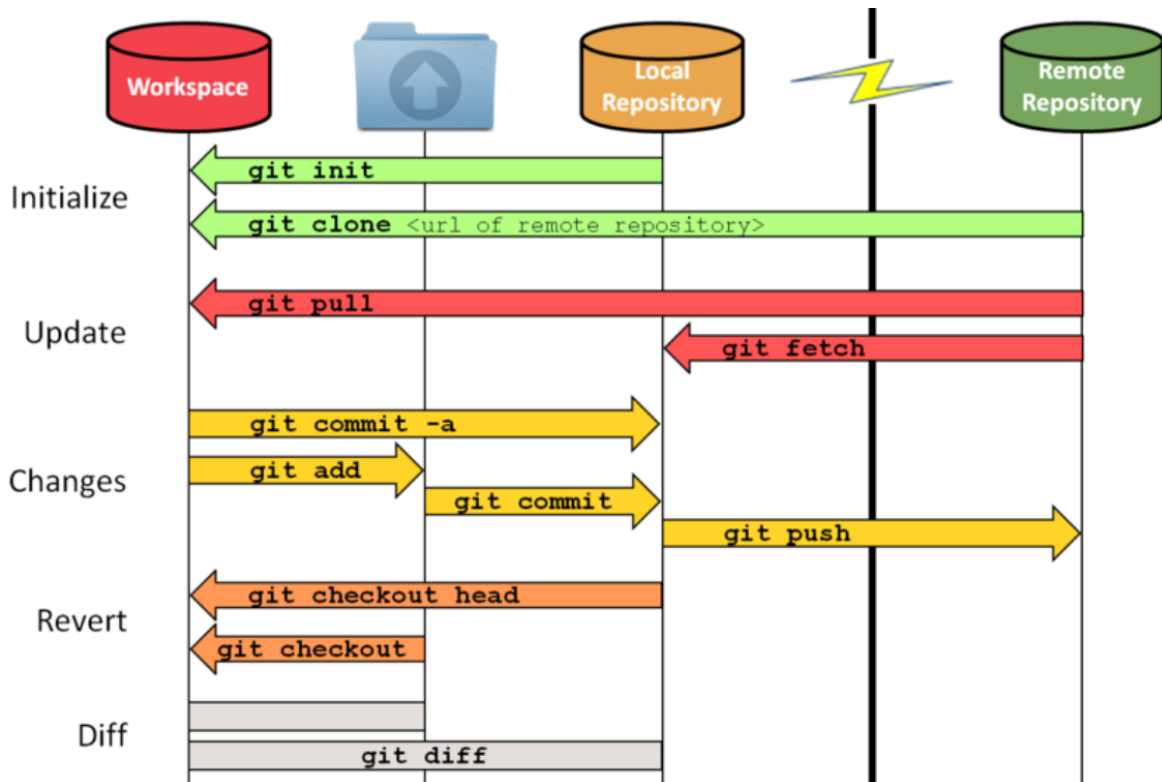
luciana@DESKTOP-793I7VA MINGW64 /d
$ git clone https://github.com/lucianadm/prueba_lab3.git
Cloning into 'prueba_lab3'...
remote: Enumerating objects: 14, done.
remote: Counting objects: 100% (14/14), done.
remote: Compressing objects: 100% (7/7), done.
remote: Total 14 (delta 1), reused 10 (delta 0), pack-reused 0
Receiving objects: 100% (14/14), done.
Resolving deltas: 100% (1/1), done.

luciana@DESKTOP-793I7VA MINGW64 /d
$ cd /d/prueba_lab3

luciana@DESKTOP-793I7VA MINGW64 /d/prueba_lab3 (main)
$ git add .

luciana@DESKTOP-793I7VA MINGW64 /d/prueba_lab3 (main)
$
```

=====



PARA QUE ALGUNOS ARCHIVOS O CARPETAS NO ME LAS SUBA AL REPO:

creo un archivo `.gitignore`

y ahí adentro es una lista directamente con el nombre del archivo o la carpeta, por ej:

`apunte.pdf`

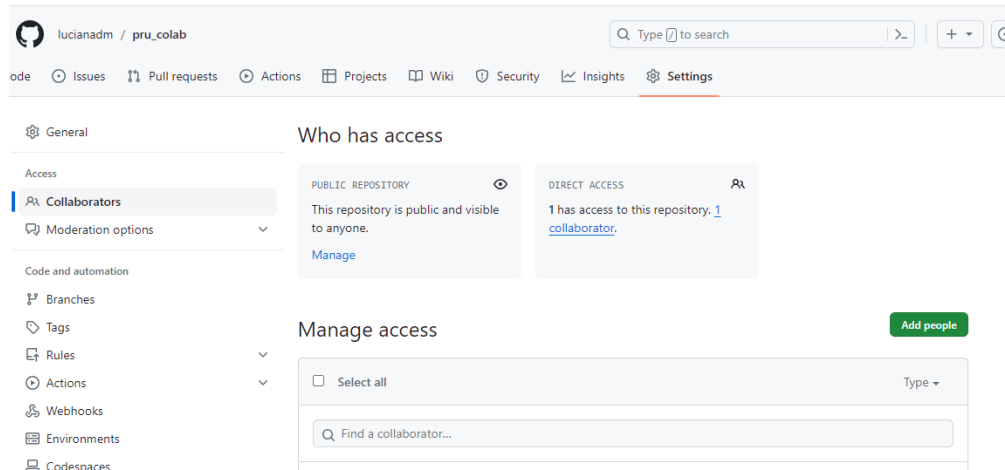
`/Carpeta`

No me sube ni el archivo `apunte.pdf` ni la carpeta `Carpeta`, ambos están en la carpeta del repositorio local y no se suben al github.

TRABAJO EN GRUPO:

Uno es el coordinador, crea el proyecto (Main) en su compu y lo asocia con un repositorio del Github, y agrega a los colaboradores:

En el proyecto en Github va a Settings (arriba a la derecha), Collaborators (izquierda arriba)



Se agregan los colaboradores en Add people con el usuario de Github.

Cada colaborador:

Clona en su PC el proyecto

Genera una rama nueva a partir del proyecto, para no modificar el proyecto principal y desp se une, de la siguiente forma:

Ejecuta el GitBash en una carpeta vacia:

Primero clono el proyecto que creo el coordinador.

git clone nombre_del_proyecto

me paro en la carpeta del proyecto con el Gitbash (cerrarlo y ejecutarlo nuevamente sobre la del repo):

git branch mi_rama⇒ creo la rama

git checkout mi_rama => me paso a la rama nombre_de_la_rama para q lo q haga se cargue ahi

Trabajo en mi compu en el proyecto (que ahora es una rama)

Cuando termino subo al Github los cambios (a mi rama)

git add .

git commit -m "comentario"

git push origin mi_rama

```
luciana@DESKTOP-793I7VA MINGW64 /d/pru_branch/TyDD (rama_lu)
$ git add .
luciana@DESKTOP-793I7VA MINGW64 /d/pru_branch/TyDD (rama_lu)
$ git commit -m "agrego a mi rama?"
[rama_lu 078622e] agrego a mi rama?
1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 AGREGO_ARCH_LUCIANA.txt
luciana@DESKTOP-793I7VA MINGW64 /d/pru_branch/TyDD (rama_lu)
$ git push origin rama_lu
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 286 bytes | 286.00 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.
```


Cada vez que voy a trabajar:

Me bajo a mi rama la ultima version del proyecto:

me paro con Gitbash en mi repositorio local (que se llama igual q el main pero es mi rama).

hago: **git checkout mi_rama**

git pull origin main ==>descargo en mi rama lo q esta en origin (a mi pc)

Despues lo actualizo en github con:

git add .

git commit -m"comentario"

git push origin mi_rama

=====

CUANDO EL COLABORADOR QUIERE PASAR A LA RAMA PPAL (MAIN) LOS CAMBIOS QUE HIZO EN SU RAMA:

Pasos para fusionar la rama con la MAIN

(ref: https://www.w3docs.com/snippets/git/best-and-safe-way-to-merge-a-git-branch-into-master.html#google_vignette)

Si sos un colaborador del proyecto (que es el MAIN). Suponiendo que haya creado una nueva rama "rama_lu" desde main.

Trabajas en esta rama, pero también quieres que se actualice con confirmaciones de git en main que otros desarrolladores han hecho de vez en cuando.

PASOS:

git fetch ==>The initial command to run is git fetch for getting the latest updates of your repository.

git rebase origin/main ==>The git rebase command will bring the latest commits of main to your branch.

git checkout main ==>Switch to the main branch

git pull origin main ==> Get the latest changes from main

git merge mi_rama ==> Then merge the changes

git push origin main ==> The final step is pushing local changes to the remote repository

MAS INFO:

Videos del Profesor Alejandro Zapata:

<https://www.youtube.com/watch?v=ptXiQwE535s&list=PLoCpUTIZIYORkDzYwdunkVf-KlqGjyoot>

- GIT y GitHub (tutorial en español). Inicio Rápido para Principiantes:

https://www.youtube.com/watch?v=hWglK8nWh60&list=PLPI81lqbj-4l8i-x2b5_MG58tZfgKmJls

- Fundamentos de GIT: <https://bluuweb.github.io/tutorial-github/guia/fundamentos.html>

- GitHub: <https://bluuweb.github.io/tutorial-github/guia/github.html>

Si en una compu alguien ya usa github y tiene asociada su cuenta , hay que eliminarla y asociarla a la nuestra antes de poder trabajar.

En Windows entrar en:

Panel de control\Cuentas de usuario\Administrador de credenciales

=> credenciales de Windows

Ahi elimino la cuenta de github que haya y tengo que desde el gitbash asociar mi usuario y email.