

ABC GIT Y GITHUB





¿Que es git?

- ✓ Software libre de control de versiones.
- ✓ Finalidad: facilitar el desarrollo y mantenimiento de proyectos personales y/o grupales.



archivo1



archivo1_v1



archivo1_v2



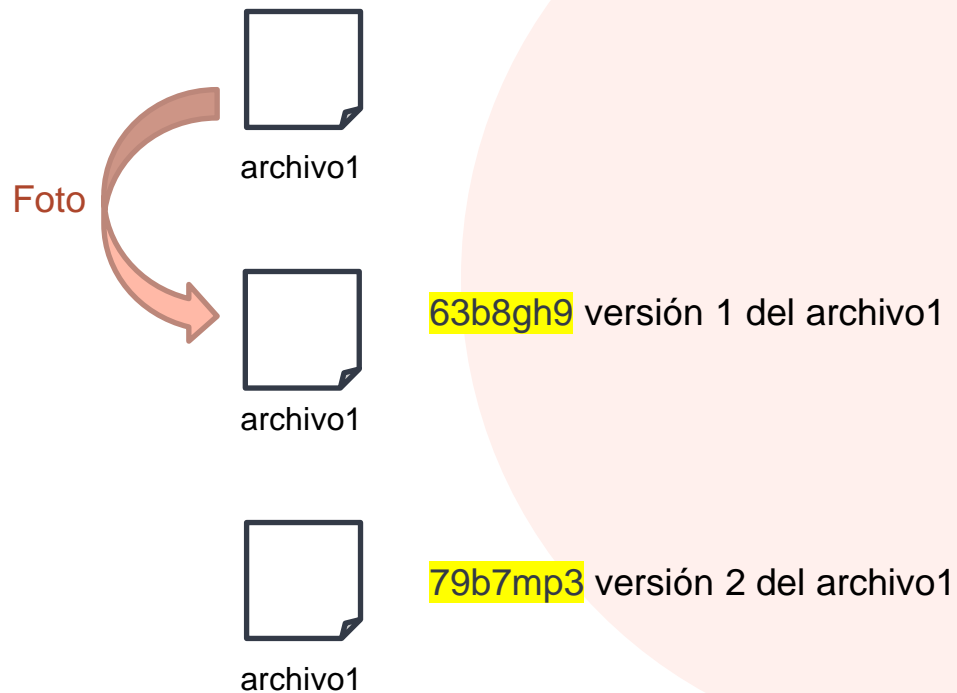
archivo1_final



MANEJO DE ARCHIVOS SIN GIT



MANEJO DE ARCHIVOS CON GIT





Porque usar git?

- ✓ Porque podrias retroceder o visualizar el cambio que te interesa.
- ✓ Porque me ayuda a poder trabajar con muchos archivos sin estrezarme en perder algun cambio.





¿Que es GitHub?

- ✓ Plataforma de desarrollo colaborativo para alojar proyectos utilizando el sistema de control de versiones de git.
- ✓ Se almacena por defecto en forma publica.
- ✓ Plataforma más importantes de colaboracion para proyectos open source
- ✓ Red social para proyectos





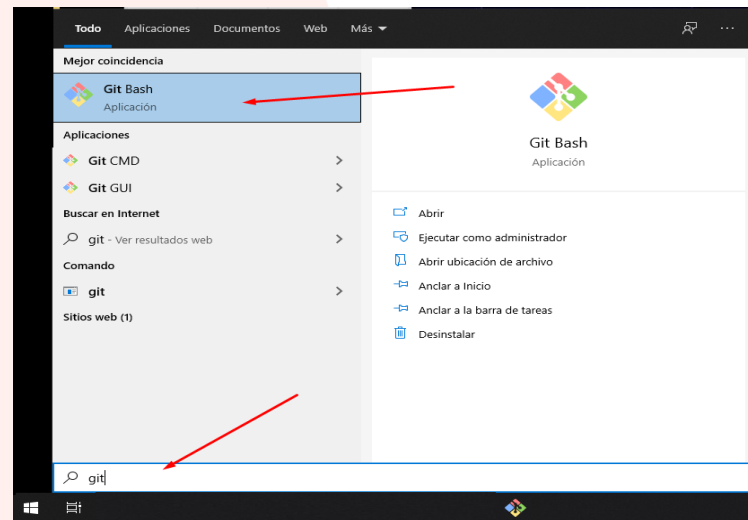
INSTALAR GIT EN NUESTRO EQUIPO

- ✓ Link para la descarga: <https://git-scm.com/downloads>
- ✓ Clic al instalador y hacer next a todos los pasos.

```
MINGW64:/c/Users/Skull

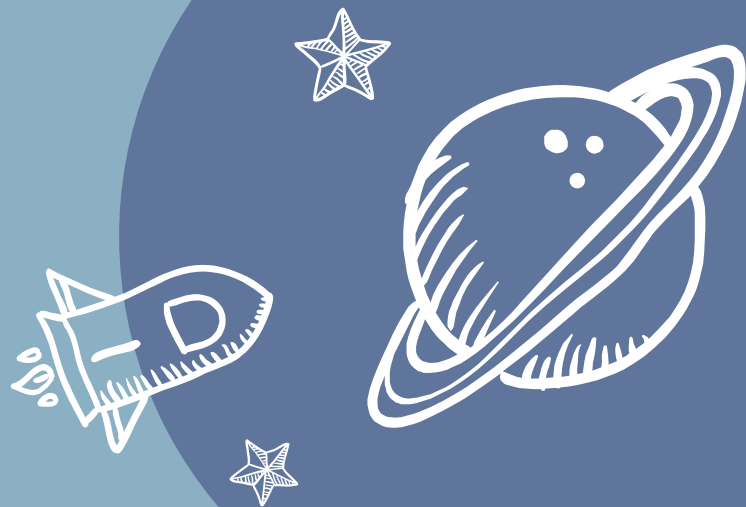
Skull@DESKTOP-01DM1KK MINGW64 ~
$ git --version
git version 2.29.2.windows.2

Skull@DESKTOP-01DM1KK MINGW64 ~
$ .....
```



Configurando git

Aprenderemos a pushear :v





- ✓ **git config --global user.name "MayuCH"**
Nombre con el que se va a guardar los cambios y versiones que se realicen.
- ✓ **git config --global user.email mayu@gmail.com**
Aquí se debe poner el correo que usamos en github.
- ✓ **git config -l**
Listar las configuraciones de git.

Configuraciones





✓ **git init**

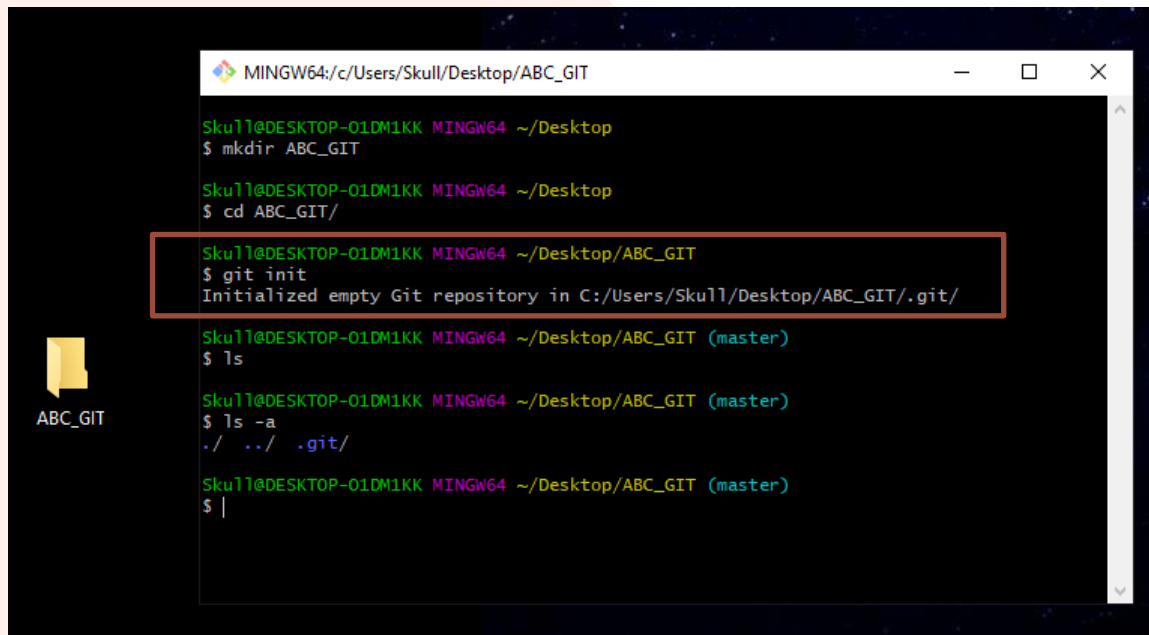
Inicializar el control de versiones.

✓ **ls**

Listar los archivos.

✓ **ls -a**

Listamos los archivos + archivos ocultos.



```
MINGW64:/c:/Users/Skull/Desktop/ABC_GIT
Sku11@DESKTOP-01DM1KK MINGW64 ~/Desktop
$ mkdir ABC_GIT
Sku11@DESKTOP-01DM1KK MINGW64 ~/Desktop
$ cd ABC_GIT/
Sku11@DESKTOP-01DM1KK MINGW64 ~/Desktop/ABC_GIT
$ git init
Initialized empty Git repository in C:/Users/Skull/Desktop/ABC_GIT/.git/
Sku11@DESKTOP-01DM1KK MINGW64 ~/Desktop/ABC_GIT (master)
$ ls
Sku11@DESKTOP-01DM1KK MINGW64 ~/Desktop/ABC_GIT (master)
$ ls -a
./  ../  .git/
Sku11@DESKTOP-01DM1KK MINGW64 ~/Desktop/ABC_GIT (master)
$ |
```

Inicializando el repositorio local



- ✓ **git status**

Nos indica que ha pasado con nuestros archivos.

- ✓ **git add index.html**

Nos ayuda a hacerle seguimiento a nuestro archivo.

- ✓ **git add .**

Nos ayuda a hacerle seguimiento a todos los archivos.

```
Skull@DESKTOP-01DM1KK MINGW64 ~/Desktop/ABC_GIT (master)
$ ls
index.html

Skull@DESKTOP-01DM1KK MINGW64 ~/Desktop/ABC_GIT (master)
$ git status
On branch master

No commits yet

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

nothing added to commit but untracked files present (use "git add" to track)

Skull@DESKTOP-01DM1KK MINGW64 ~/Desktop/ABC_GIT (master)
$ git add index.html

Skull@DESKTOP-01DM1KK MINGW64 ~/Desktop/ABC_GIT (master)
$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   index.html
```

Seguimiento del repositorio local



- ✓ **git commit -m "Mi 1er commit"**

Nos ayuda a escribir un mensaje para ese cambio.

- ✓ **git log**

Nos Podemos ver todos los commit realizados.

```
Sku11@DESKTOP-01DM1KK MINGW64 ~/Desktop/ABC_GIT (master)
$ git commit -m "Mi 1er commit"
[master (root-commit) 324e7bf] Mi 1er commit
1 file changed, 13 insertions(+)
create mode 100644 index.html
```

```
Sku11@DESKTOP-01DM1KK MINGW64 ~/Desktop/ABC_GIT (master)
$ git log
commit 324e7bfd3621a579b542356aee19d21d6e11d9bf (HEAD -> master)
Author: MayumyCH <heidich8@gmail.com>
Date: Tue Mar 30 01:00:46 2021 -0500

    Mi 1er commit
```

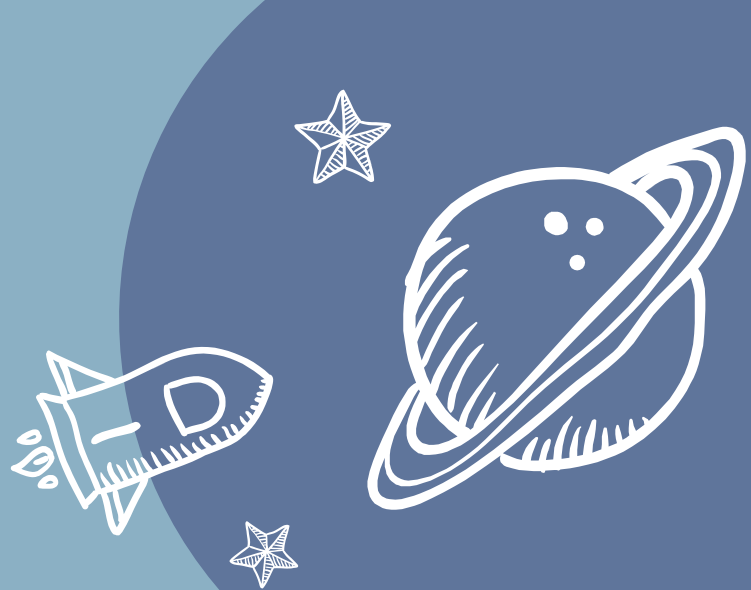
Estructura de un commit

Código Hash / Autor / Fecha / Mensaje

Guardar cambios al repositorio local

Creando un repositorio Github

Allá vamos GitHub :3 !!






Create a new repository

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

Repository template
Start your repository with a template repository's contents.

No template

Owner * **Repository name ***

 MayumyCH /

Great repository names are: [ABC_GIT_GITHUB is available.](#) inspiration? How about [special-octo-engine](#)?

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.

☐ **Add a README file**
This is where you can write a long description for your project. [Learn more.](#)

☐ **Add .gitignore**
Choose which files not to track from a list of templates. [Learn more.](#)

☐ **Choose a license**
A license tells others what they can and can't do with your code. [Learn more.](#)

Create repository





Quick setup — if you've done this kind of thing before



Set up in Desktop

or

HTTPS

SSH

https://github.com/MayumyCH/ABC_GIT_GITHUB.git

Get started by [creating a new file](#) or [uploading an existing file](#). We recommend every repository include a [README](#), [LICENSE](#), and [.gitignore](#).

...or create a new repository on the command line

```
echo "# ABC_GIT_GITHUB" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/MayumyCH/ABC_GIT_GITHUB.git
git push -u origin main
```

...or push an existing repository from the command line

```
git remote add origin https://github.com/MayumyCH/ABC_GIT_GITHUB.git
git branch -M main
git push -u origin main
```



MINGW64/c/Users/Skull/Desktop/ABC_GIT

```
Skull@DESKTOP-01DM1KK MINGW64 ~/Desktop/ABC_GIT (master)
$ git remote add origin https://github.com/MayumyCH/ABC_GIT_GITHUB.git
```

```
Skull@DESKTOP-01DM1KK MINGW64 ~/Desktop/ABC_GIT (master)
$ git branch -M main
```

```
Skull@DESKTOP-01DM1KK MINGW64 ~/Desktop/ABC_GIT (main)
$ git push -u origin main
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Delta compression using up to 16 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 432 bytes | 432.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/MayumyCH/ABC_GIT_GITHUB.git
 * [new branch]      main -> main
Branch 'main' set up to track remote branch 'main' from 'origin'.
```

```
Skull@DESKTOP-01DM1KK MINGW64 ~/Desktop/ABC_GIT (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean

Skull@DESKTOP-01DM1KK MINGW64 ~/Desktop/ABC_GIT (main)
```

Quick setup — if you've done this kind of thing before

Set up in Desktop or **HTTPS** **SSH** https://github.com/MayumyCH/ABC_GIT_GITHUB.git

Get started by [creating a new file](#) or [uploading an existing file](#). We recommend every repository include

...or create a new repository on the command line

```
echo "# ABC_GIT_GITHUB" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/MayumyCH/ABC_GIT_GITHUB.git
git push -u origin main
```

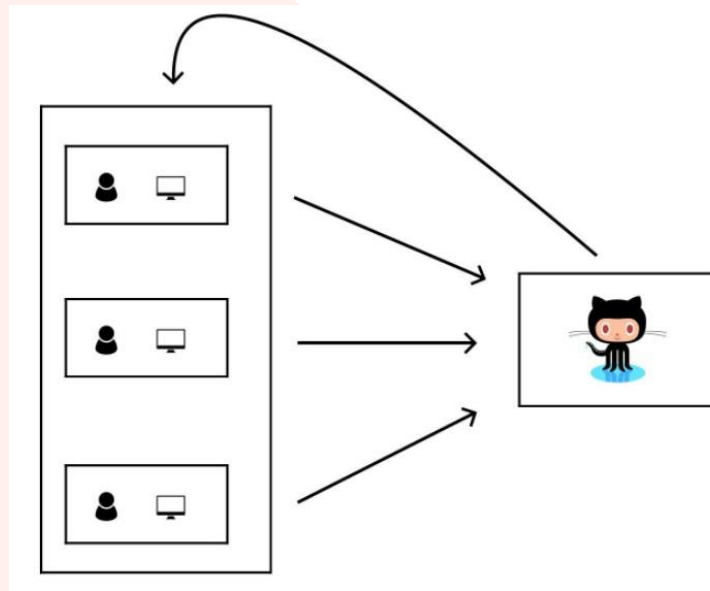
...or push an existing repository from the command line

```
git remote add origin https://github.com/MayumyCH/ABC_GIT_GITHUB.git
git branch -M main
git push -u origin main
```

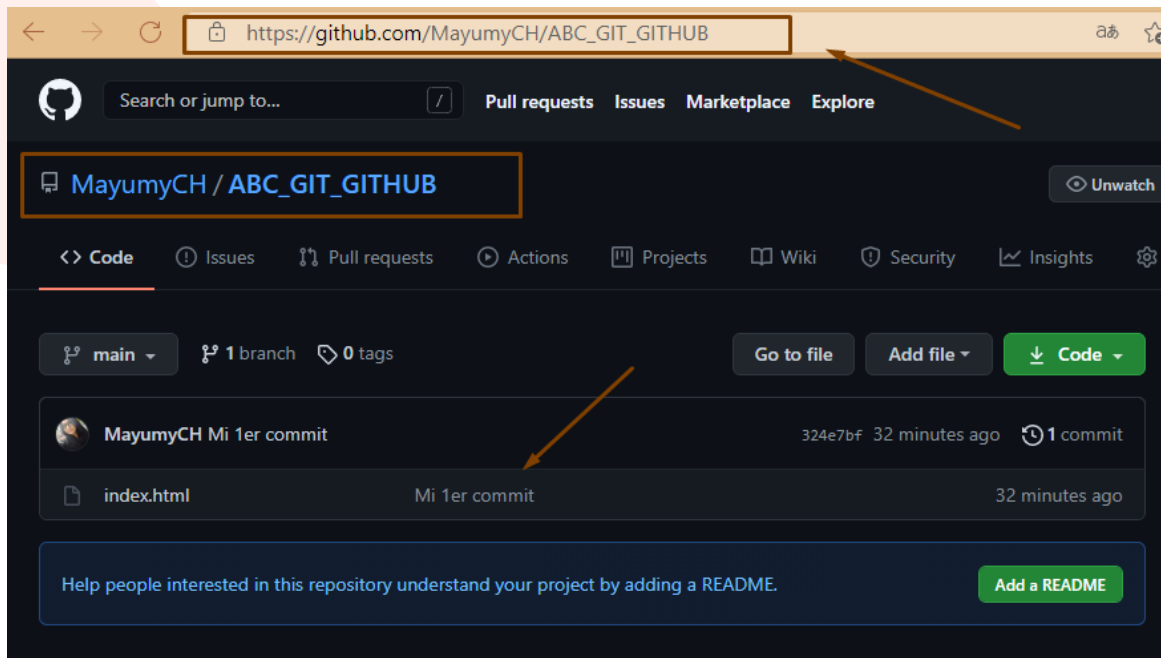
Conectando repositorio local con repositorio remoto



- ✓ **git remote add origin url_github**
Crea una nueva conexión a un repositorio remoto
- ✓ **git Branch -M main**
Cambiando el nombre de la rama actual a main.
- ✓ **git push -u origin main**
Enviar todo el contenido del repositorio local al repositorio remoto.
- ✓ **git pull origin main**
Descarga todos los cambios del repositorio remoto al repositorio local.



Explicación Repo local <-> Repo remoto



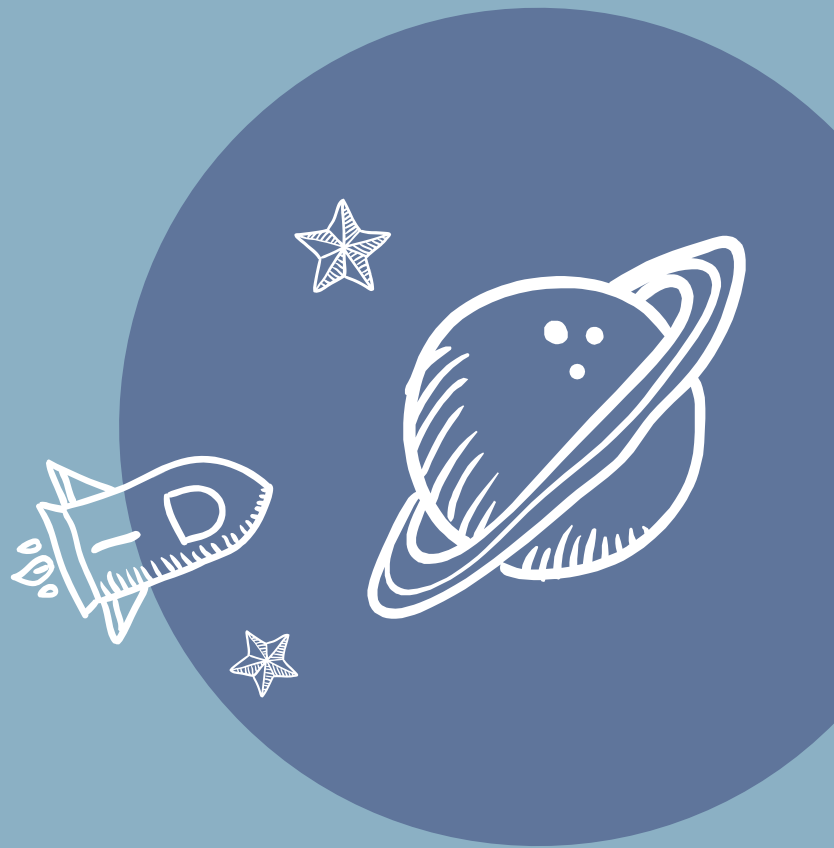
Mi primer repositorio





Bajar cambios de un repositorio

Allá vamos GitHub de nuevo :3 !!



Una vez ubicado en el repositorio de interes.
Nos vamos a la opcion :



Se tiene 3 opciones:

✓ **Clone**

Bajar el repositorio tal cual esta en el github con todos los registros de commits realizados. Nos vamos al terminal y ejecutamos lo siguiente:

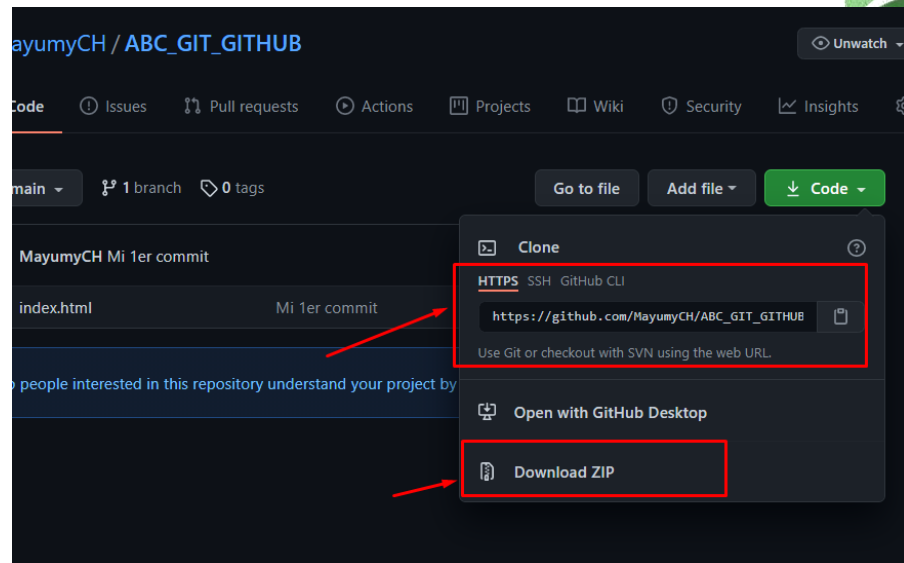
```
git clone url_repositorio
```

✓ **Github Desktop**

Interface que conecta el git con el github, evitamos escribir comandos en la terminal.

✓ **Download ZIP**

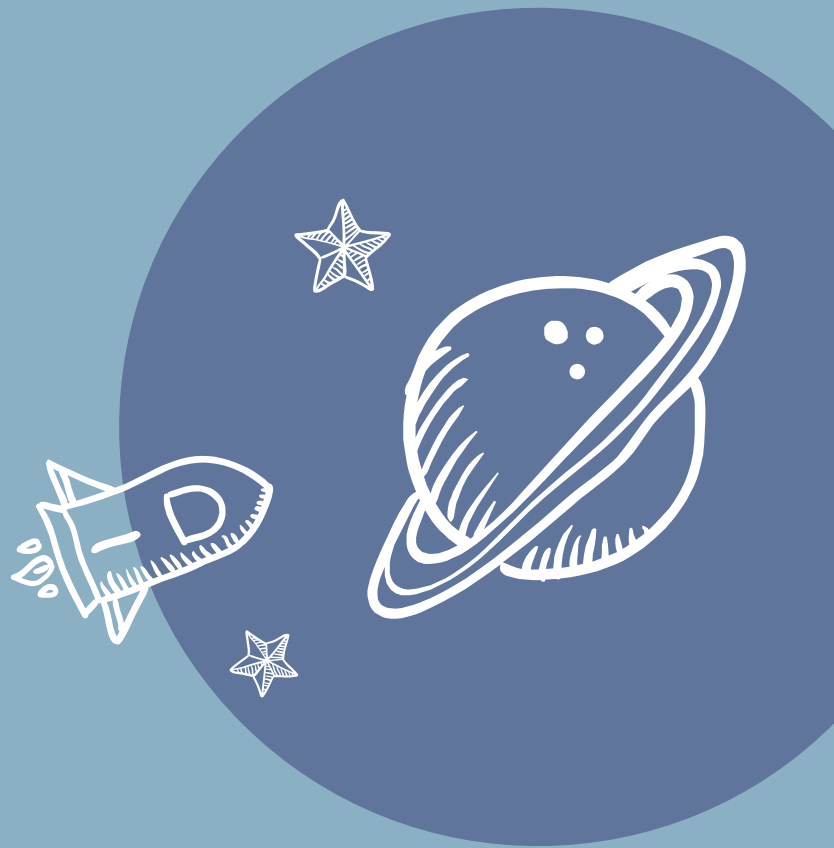
Descargar los documentos en la ultima version; esta no contendra ningun historial de los cambios realizados.



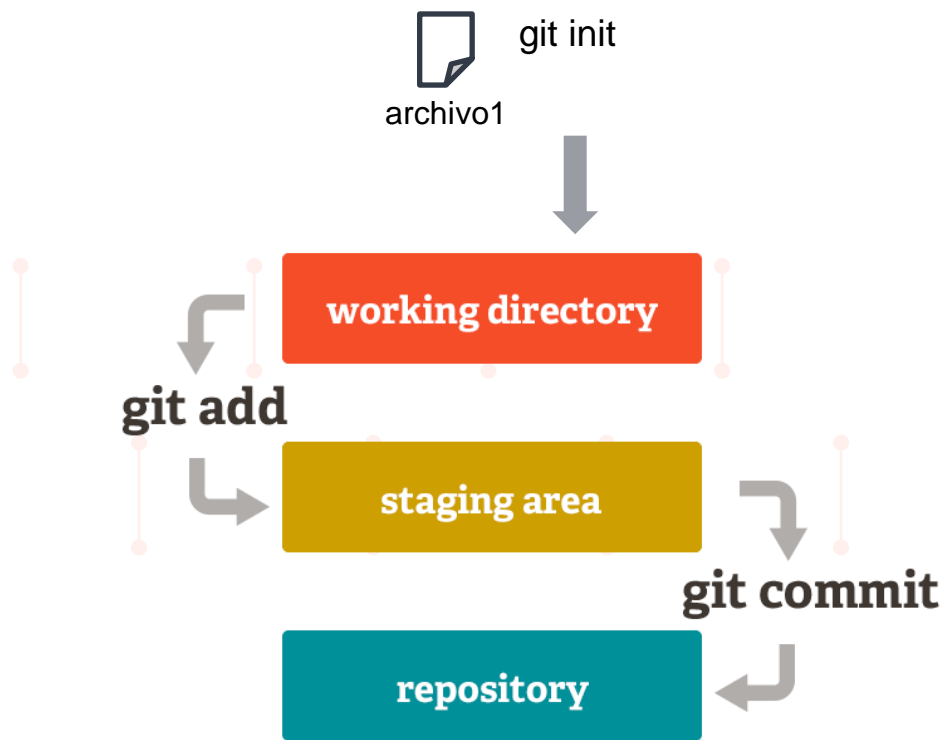


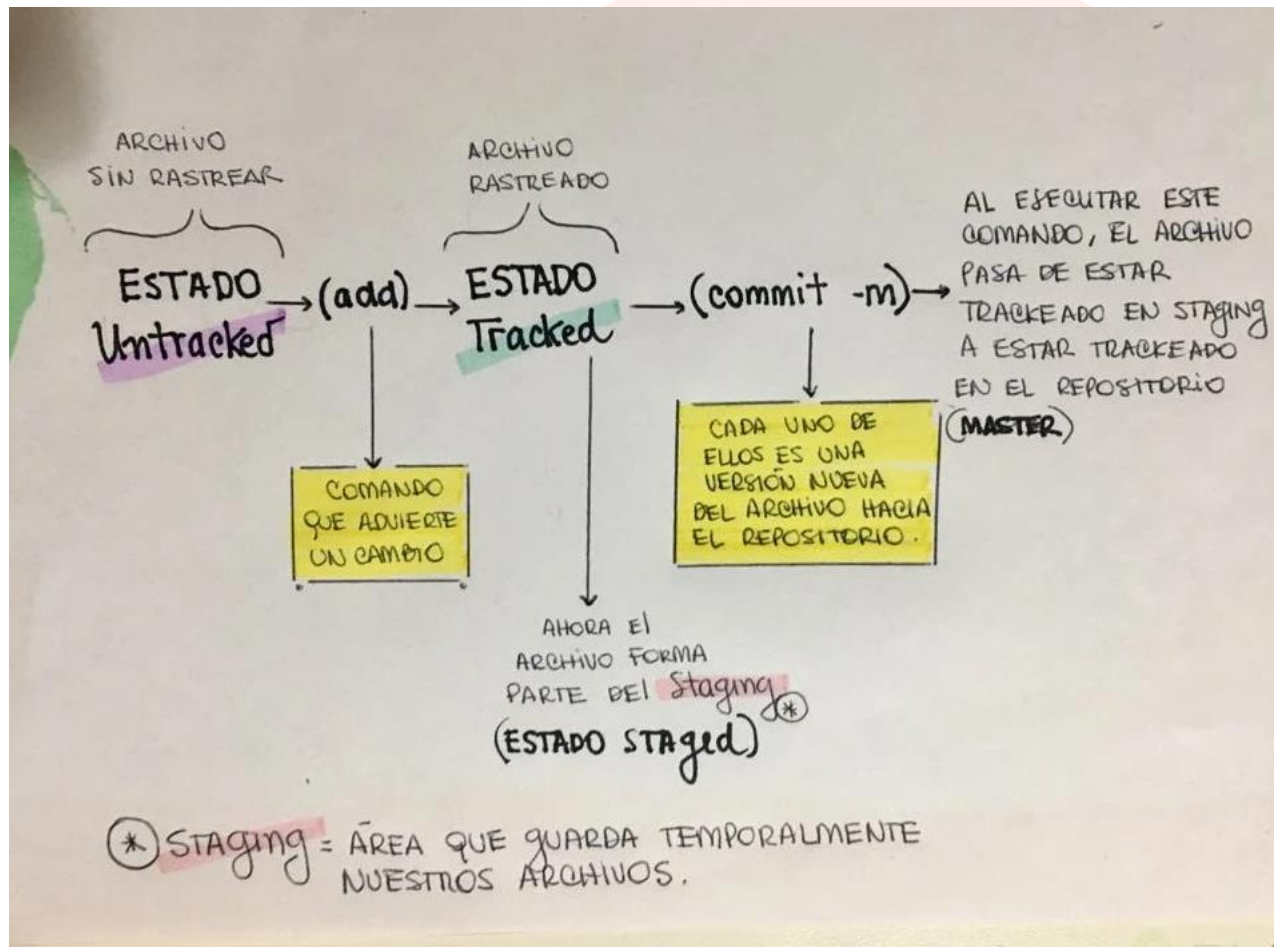
Explorando conceptos !

Allá vamos!!



Ciclo básico de trabajo con Git





Estado del archivo

Resumen !



Subir un proyecto a un Repositorio

```
git init
git add .
git commit -m "mensaje"
git remote add origin link_del_nuevo_repositorio
git Branch -M main
git push -u origin main
```

Bajar un Repositorio

```
git clone url_repositorio
git pull origin main
```

Subir cambios a un Repositorio

```
git add .
git commit -m "mensaje"
git pull origin main
git push origin main
```

