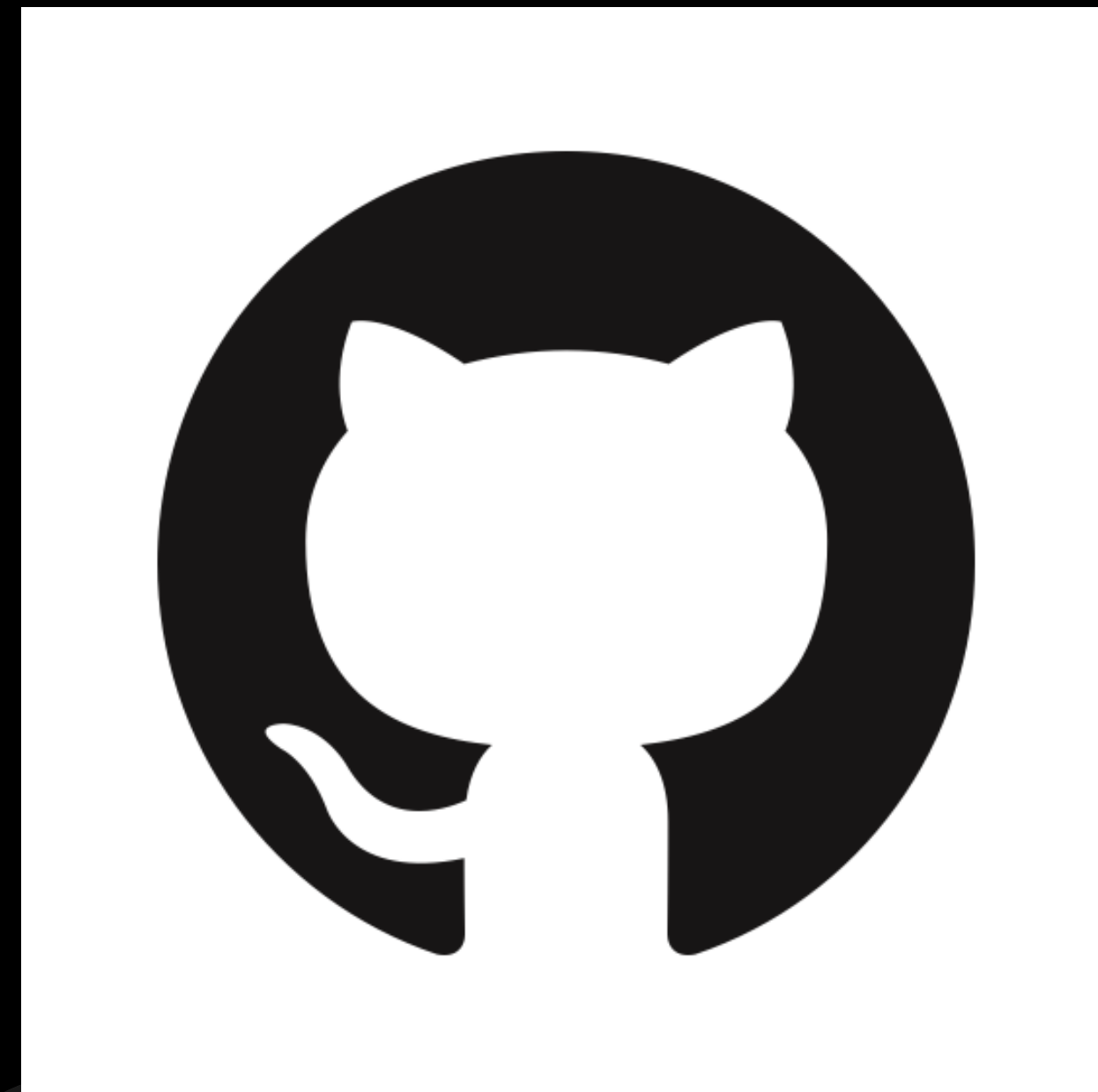




IAW

# GIT Y GITHUB

les las Fuentezuelas



2023 - HÉCTOR ONIEVA GUTIÉRREZ



# Índice

01 ¿QUÉ ES GIT?

02 PRERREQUISITOS

03 COMANDOS DE GIT

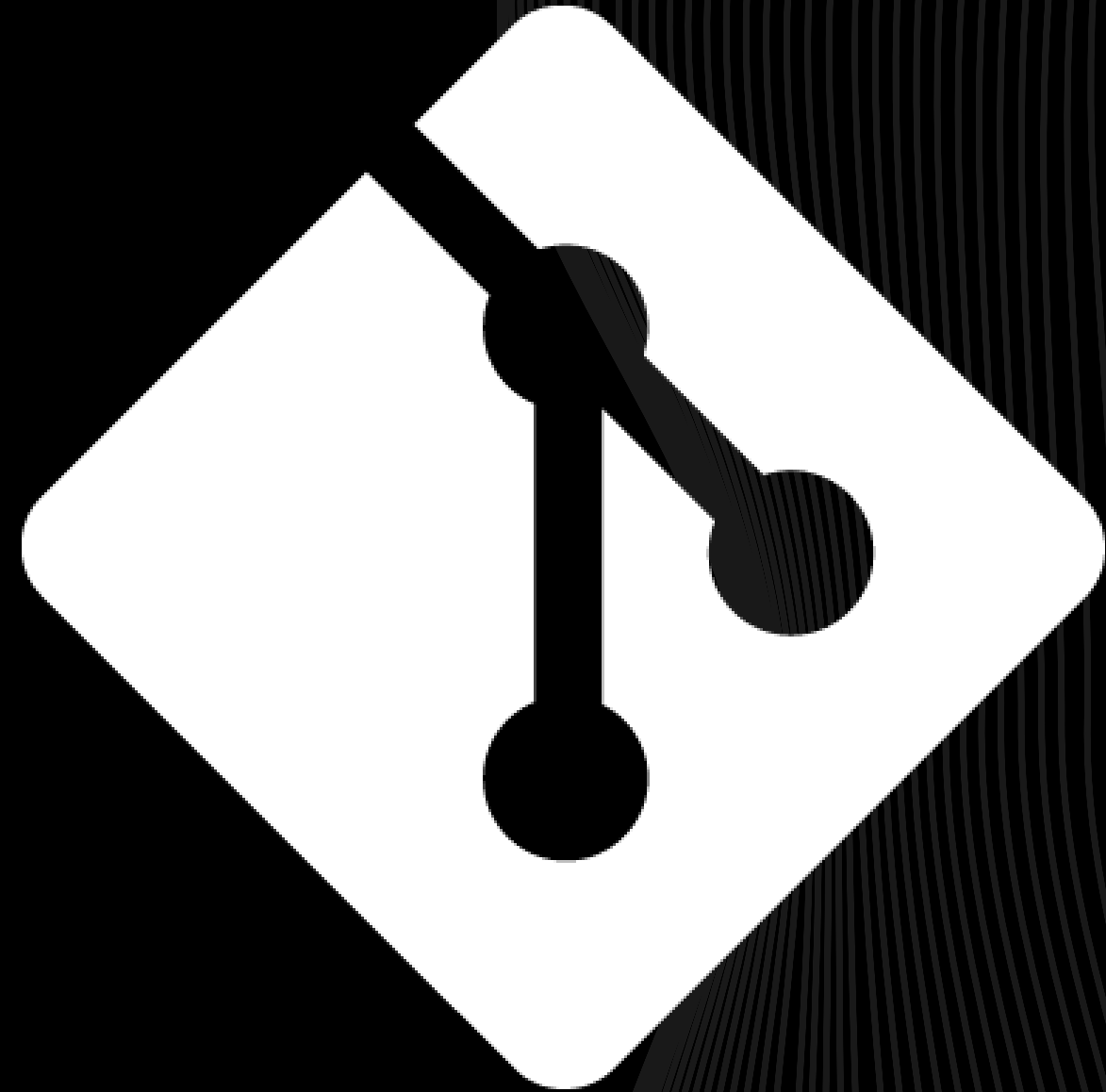
04 INSTALACIÓN DE GIT

05 CONFIGURACIÓN DE GIT

06 CREACIÓN DE REPOSITORIOS  
LOCALES

# 1.¿QUÉ ES GIT?

-Git es un sistema de control de versiones (VCS, por sus siglas en inglés) de código abierto ampliamente utilizado en el desarrollo de software. Fue creado por Linus Torvalds en 2005 y se ha convertido en una herramienta esencial para rastrear y gestionar cambios en el código fuente de proyectos de programación. A continuación, se explican los conceptos clave de Git y su importancia



1

1.1

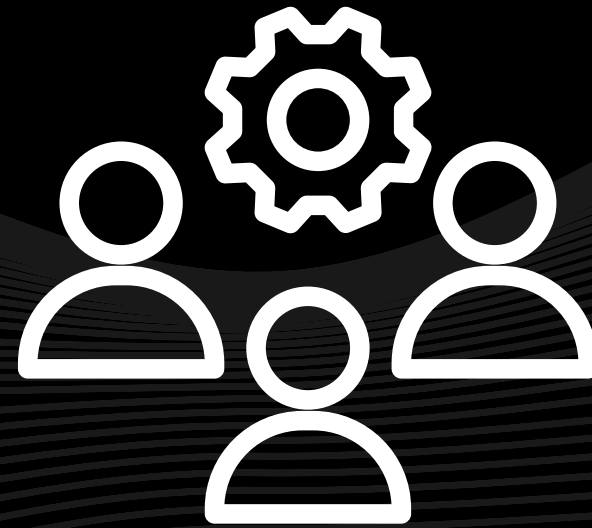
1.2

1.1

# 1.1 ¿Para que se usa Git/Github?



**Manejo de  
repositorios y ramas**



**Poder trabajar en  
equipo**



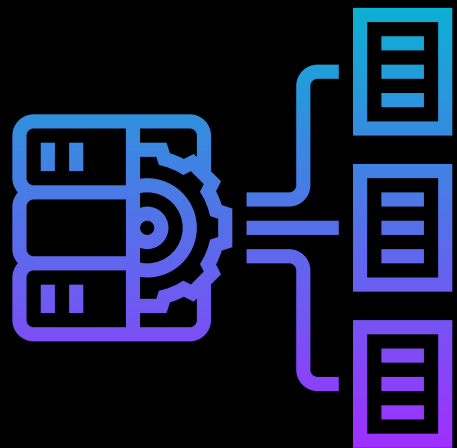
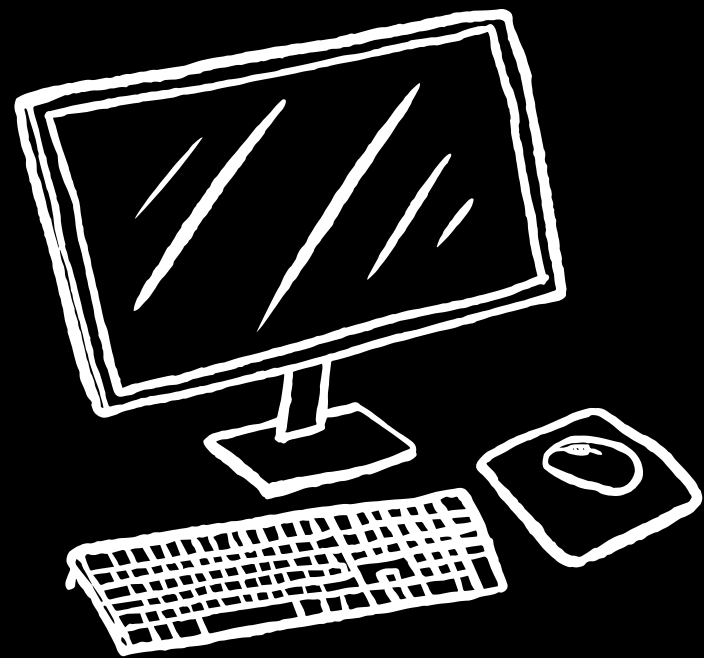
**Volver a un archivo  
en el momento que  
estaba correcto**

**1**

**1.2**

1

## 1.2 Terminología básica



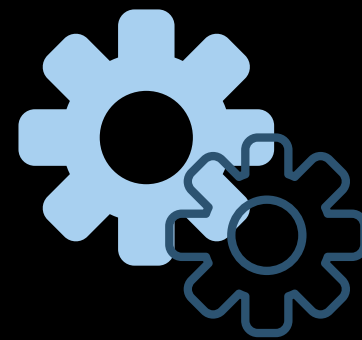
### Clon

Es una copia exacta del repositorio.



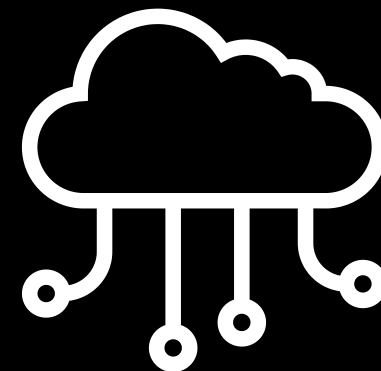
### GIT

Es un sistema de control de versiones: un software que permite registrar el historial de cambios de un proyecto



### Repositorio

Es todo proyecto que está siendo seguido por GIT, ya tiene un historial de GIT en el que se están registrando sus cambios



### Commit

Es cada uno de los cambios registrados en el historial de GIT. Cada uno de los desarrolladores manda los commits de los cambios que ha hecho. No es automático, cada desarrollador tiene que decir qué hizo y por qué.



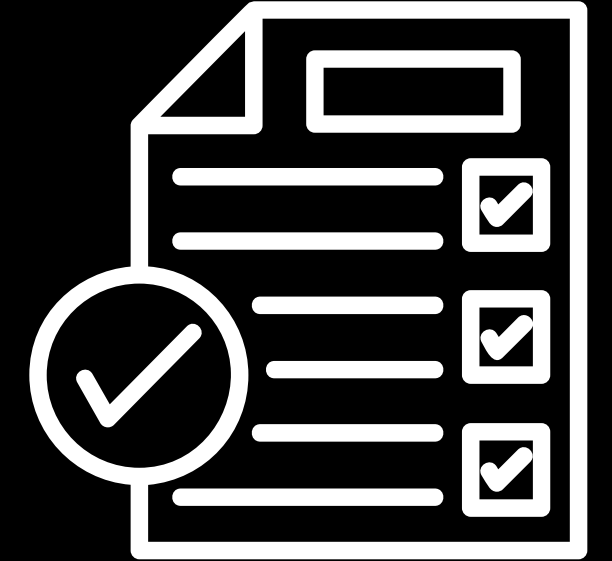
### Ramas

Son nuevos caminos que toma el proyecto. La rama principal se llama máster y es donde está el proyecto que sale a producción.

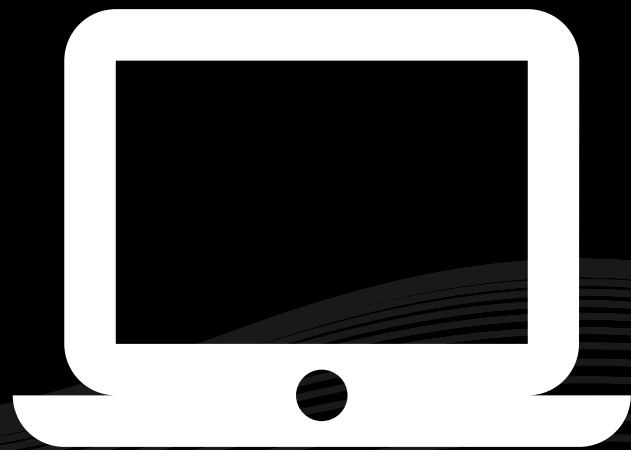
1.1

## 2. Prerrequisitos

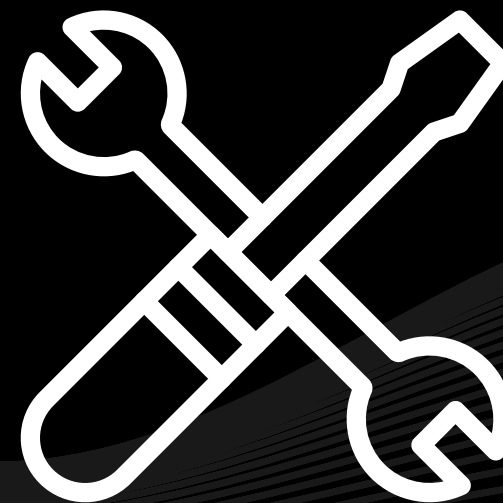
Para poder tener Git o Github en nuestro equipo debemos cumplir los siguientes requisitos



**Un ordenador con  
Git instalado**



**Una herramienta  
para usar Git**



**Una cuenta de  
Github**





# 3. Comandos de GIT

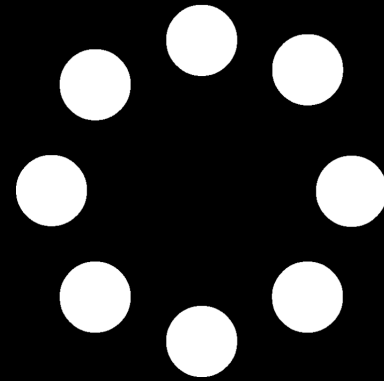
**git init**



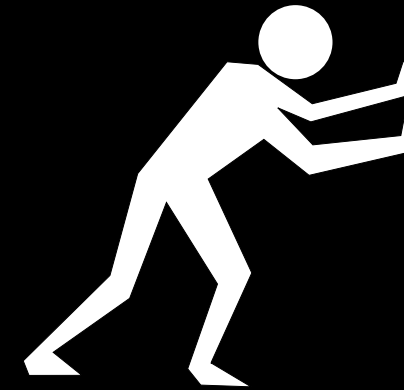
**git commit -m**



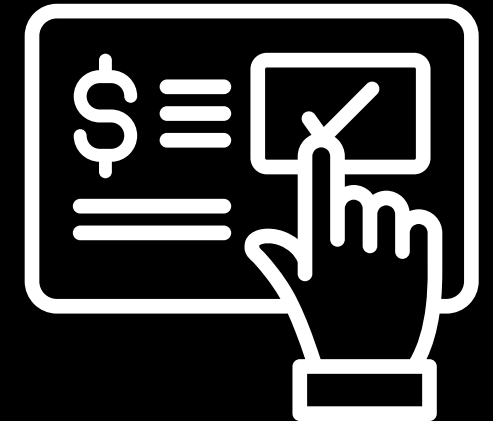
**git status**



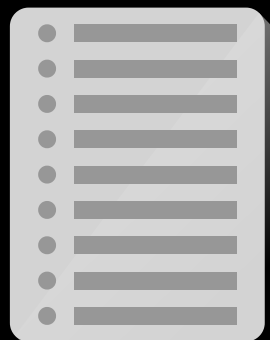
**git push**



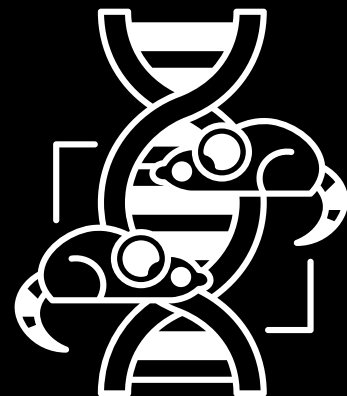
**git checkout**



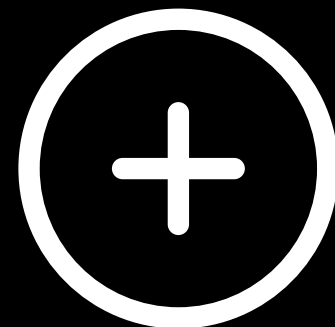
**git log**



**git clone**



**git add**



**git diff**



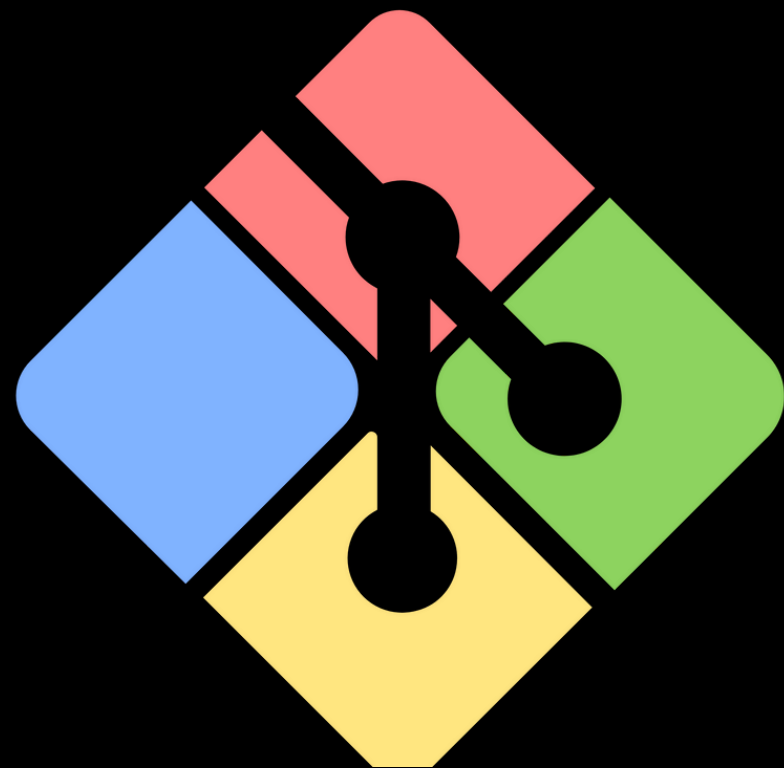
**git add <file>**



# 4.Instalación de GIT

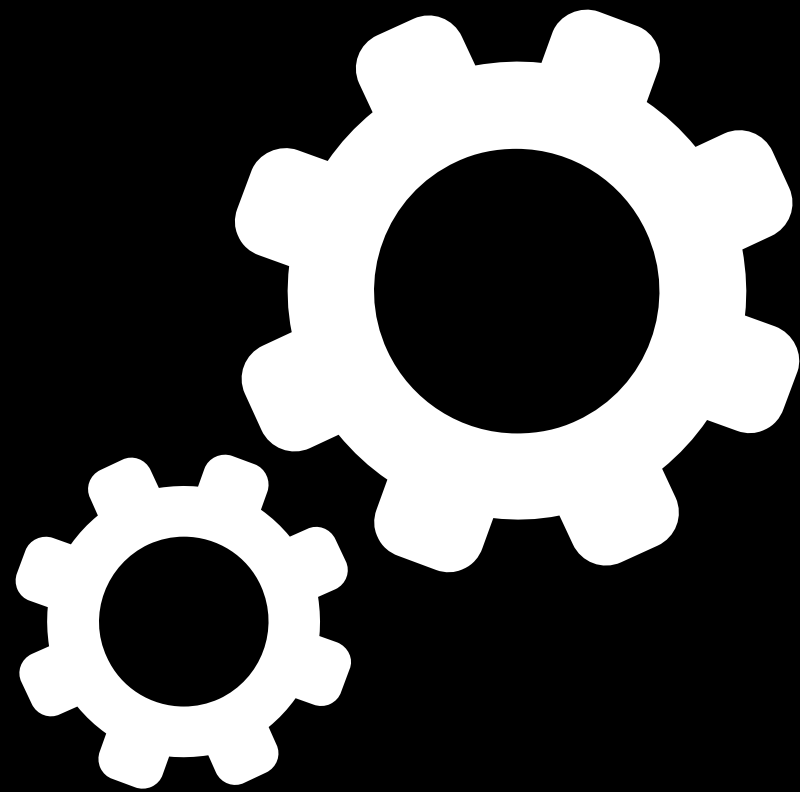
Para instalar GIT, tendremos que descargar la versión de Windows, esta se encuentra en el siguiente enlace.

<https://git-scm.com/downloads>





# 5. Configuración de Git



**GIT VERSION**  
Nos muestra la versión

```
MINGW64:/d/Git-test  
regina@DESKTOP-3FF7K39 MINGW64 /d/Git-test  
$ git version  
git version 2.37.2.windows.2
```

**GIT CONFIG**  
Nos permite modificar las  
distintas configuraciones

```
regina@DESKTOP-3FF7K39 MINGW64 /d/Git-test  
$ git config --global user.name "regina"
```

**ls**  
Ver directorios

```
regina@DESKTOP-3FF7K39 MINGW64 /d/Git-test  
$ ls  
'folder 1'/'
```

**ll**  
Ver con más detalle la  
información del contenido de  
la carpeta

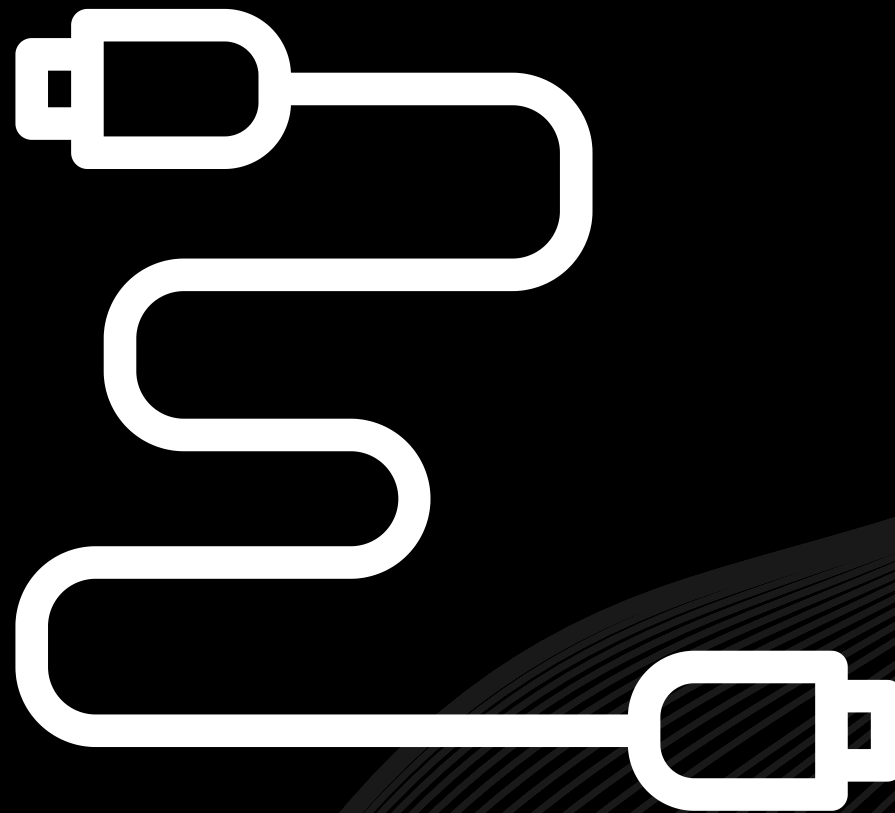
```
regina@DESKTOP-3FF7K39 MINGW64 /d/Git-test  
$ ll  
total 0  
drwxr-xr-x 1 regina 197121 0 Aug 19 14:04 'folder 1'/'
```

# 6. Creación de repositorios locales

6

Una vez configurado Github, abarcaremos una cantidad de comandos para la creación de repositorios locales

6.1



6.2

6.1



## 6.1 GIT INIT



Para iniciar nuestro GIT local

```
regina@DESKTOP-3FF7K39 MINGW64 /d/Git-test  
$ git init  
Initialized empty Git repository in D:/Git-test/.git/  
regina@DESKTOP-3FF7K39 MINGW64 /d/Git-test master)  
$
```



## 6.2 RAMAS DE GIT

### GIT BRANCH NUEVO-FEATURE

Situarse en nueva  
rama

```
regina@DESKTOP-3FF7K39 MINGW64 /d/git
$ git checkout nuevo-feature
Switched to branch 'nuevo-feature'
```

```
regina@DESKTOP-3FF7K39 MINGW64 /d/git
$ git checkout nuevo-fea
Switched to branch 'nuev
```

### GIT BRANCH

Ver Ramas locales y  
remotes

### GIT CHECKOUT NUEVO-FEATURE

Ramas creadas  
localmente

```
regina@DESKTOP-3FF7K39 MINGW64 /d/git
$ git branch
main
* nuevo-feature
```

```
regina@DESKTOP-3FF7K39 MINGW64 /d/git
$ git branch -a
main
* nuevo-feature
remotes/origin/HEAD
remotes/origin/main
```

### GIT BRANCH A

Listar todas las ramas

6

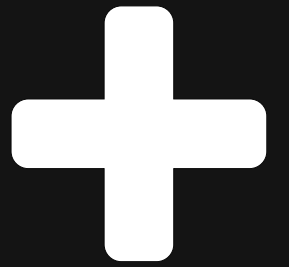
## 6.3 Más comandos para creación de repositorios locales



6.2

### GIT ADD

Con este comando añadiremos todo el staging area



### GIT RM

Por si queremos eliminar un archivo o directorio

### GIT COMMIT

Para realizar la transición del staggin area al repositorio local



### GIT REMOTE

Permite conectar el repositorio local con un remoto



# FIN

2023-Héctor Onieva Gutiérrez  
honigut0312@gmail.coms