

MANUAL DE GIT

Guillermo Gallego Desdentado
2º DAW

Índice

1	Instalación de GIT, configuración y uso de GitHub.....	3
1.1	Inicializar el repositorio local.....	3
1.2	Crear enlace entre los repositorios local y remoto.....	3
1.3	Comprobar repositorios remotos.....	3
1.4	Crear rama “master”, primer commit y push.....	4
1.5	Crear una rama distinta a “master”	5
2	Añadir un proyecto.....	6
3	Subiendo este manual al repositorio remoto.....	7
4	Clonar repositorio de un compañero (Rafa).....	8
5	Modificar proyecto de un compañero (Rafa).....	8
6	GitKraken.....	10
6.1	Instalación, importar proyecto.....	10
6.2	Cambio de rama.....	10
6.3	Git add, git commit, git push – Actualizando el repositorio.....	11
7	Eliminar repositorios.....	12
8	Comandos de GIT.....	13

1 Instalación de GIT, configuración y uso de GitHub

→ Instalar GIT (Ubuntu 18.04) → **sudo apt-get install git**

→ Crear cuenta en GitHub

→ Crear un nuevo repositorio → New repository

* Añadir mínimo un nombre, opcional, descripción, licencia, etc

* Create repository

→ Establecer el repositorio local, dentro de la carpeta que creamos, con el comando → **git init**

→ Establecer el vínculo entre el repositorio remoto y el repositorio local con **git remote add + un alias + la url del repositorio remoto**.

1.1 Inicializar el repositorio local

Para inicializar el repositorio local, lo haremos con **git init**, que creará un nuevo subdirectorio llamado **.git** que contiene todos los archivos necesarios del repositorio.

```
willy@willy-laptop:~/Documentos/DAW2/Despliegue de Aplicaciones Web/Trabajos/Desafío  
1/repo_desafio1$ git init
```

1.2 Crear enlace entre los repositorios local y remoto

Con **git remote add rutarepositoriolocal rutarepositorioremoto**, creamos el enlace entre el repositorio local y el repositorio remoto.

```
willy@willy-laptop:~/Documentos/DAW2/Despliegue de Aplicaciones Web/Trabajos/Desafío  
1/repo_desafio1$ git remote add desafio1 https://github.com/willytal/desafio1.git
```

1.3 Comprobar repositorios remotos

Con **git remote -v**, vemos la información de todos los remotos que tenemos dados de alta para el repositorio local.

```
willy@willy-laptop:~/Documentos/DAW2/Despliegue de Aplicaciones Web/Trabajos/Desafío  
1/repo_desafio1$ git remote -v  
desafio1      https://github.com/willytal/desafio1.git (fetch)  
desafio1      https://github.com/willytal/desafio1.git (push)
```

1.4 Crear rama “master”, primer commit y push

- Configuramos la rama master con **git push --set-upstream rutarepositoriolocal master** (pedirá el usuario y contraseña de GitHub). Con esto hacemos un enlace entre la rama local y la rama remota.
- Después, añadimos un **archivo inicial**, un primer **commit**, seguido de un **push**.

```
willy@willy-laptop:~/Documentos/DAW2/Despliegue de Aplicaciones Web/Trabajos/Desafío
1/repo_desafio1$ git push --set-upstream desafio1 master
Username for 'https://github.com': willytal
Password for 'https://willytal@github.com':
Contando objetos: 3, listo.
Escribiendo objetos: 100% (3/3), 209 bytes | 209.00 KiB/s, listo.
Total 3 (delta 0), reused 0 (delta 0)
remote:
remote: Create a pull request for 'master' on GitHub by visiting:
remote:   https://github.com/willytal/desafio1/pull/new/master
remote:
To https://github.com/willytal/desafio1.git
 * [new branch]   master -> master
Rama 'master' configurada para hacer seguimiento a la rama remota 'master' de 'desafio1'.
```

- Crear un archivo inicial con, por ejemplo: **touch inicial**
- Añadimos el fichero **inicial** a la zona de preparación con: **git add inicial**
- Hacemos un primer commit con **git commit -m “mensaje”**:

```
willy@willy-laptop:~/Documentos/DAW2/Despliegue de Aplicaciones Web/Trabajos/Desafío
1/repo_desafio1$ git commit -m "inicial commit"
[master (commit-raíz) 49a2e92] inicial commit
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 inicial
```

- Realizamos un **git push** para enviar los cambios y refrescamos el navegador para comprobar que se ha subido el archivo inicial.

The screenshot displays the GitHub interface for a repository named 'desafio1' owned by 'willytal'. At the top, there are buttons for 'Watch', 'Star', and 'Fork'. Below this is a navigation bar with tabs for 'Code', 'Issues', 'Pull requests', 'Projects', 'Wiki', 'Insights', and 'Settings'. The main content area shows repository statistics: '1 commit', '1 branch', '0 releases', and '1 contributor'. There are buttons for 'New pull request', 'Create new file', 'Upload files', 'Find file', and 'Clone or download'. The commit history section shows a single commit titled 'initial commit' by 'willytal' 12 minutes ago. At the bottom, a light blue banner prompts the user to 'Help people interested in this repository understand your project by adding a README.' with an 'Add a README' button.

1.5 Crear una rama distinta a “master”

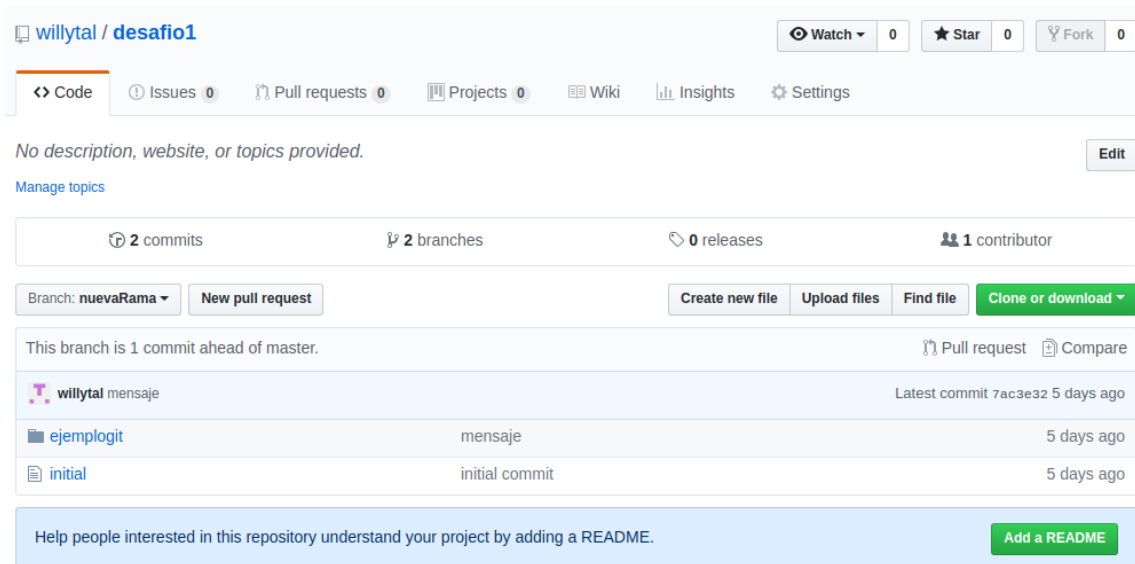
- Con comando **git checkout -b nuevaRama**, crea y hace checkout a la rama “nuevaRama”.
- Con **git push --set-upstream rutarepositoriolocal nuevaRama**, se asigna la rama local “nuevaRama” a la rama remota. Pedirá usuario y contraseña de GitHub.

```
willy@willy-laptop:~/Documentos/DAW2/Despliegue de Aplicaciones Web/Trabajos/Desafío
1/repo_desafio1$ git push --set-upstream desafio1 nuevaRama
Username for 'https://github.com': willytal
Password for 'https://willytal@github.com':
Contando objetos: 8, listo.
Delta compression using up to 4 threads.
Comprimiendo objetos: 100% (7/7), listo.
Escribiendo objetos: 100% (8/8), 1.01 KiB | 1.01 MiB/s, listo.
Total 8 (delta 0), reused 0 (delta 0)
To https://github.com/willytal/desafio1.git
 49a2e92..7ac3e32 nuevaRama -> nuevaRama
Rama 'nuevaRama' configurada para hacer seguimiento a la rama remota 'nuevaRama' de 'desafio1'.
```

2 Añadir un proyecto

Con el comando **git add rutadelproyectorlocal**, añadimos a la zona de preparación el proyecto elegido. (Se pueden añadir archivos individuales con **git add nombreachivo**, o enviar todos los archivos con **git add .**)

willy@willy-laptop:~/Documentos/DAW2/Despliegue de Aplicaciones Web/Trabajos/Desfío 1/repo_desafio1\$ **git add ejempligit**



- Hacemos **commit** para guardar los cambios en el repositorio:

```
willy@willy-laptop:~/Documentos/DAW2/Despliegue de Aplicaciones Web/Trabajos/Desfío 1/repo_desafio1$ git commit -m "mensaje"
[nuevaRama 7ac3e32] mensaje
3 files changed, 30 insertions(+)
create mode 100644 ejempligit/nbproject/project.properties
create mode 100644 ejempligit/nbproject/project.xml
create mode 100644 ejempligit/public_html/index.html
```

- Por último, enviamos los cambios con el comando **git push**.

```
willy@willy-laptop:~/Documentos/DAW2/Despliegue de Aplicaciones Web/Trabajos/Desfío 1/repo_desafio1$ git push
Username for 'https://github.com': willytal
Password for 'https://willytal@github.com':
Contando objetos: 3, listo.
Delta compression using up to 4 threads.
Comprimiendo objetos: 100% (3/3), listo.
Escribiendo objetos: 100% (3/3), 303.39 KiB | 17.85 MiB/s, listo.
Total 3 (delta 0), reused 0 (delta 0)
To https://github.com/willytal/desafio1.git
3a628d8..fbc5ae5 nuevaRama -> nuevaRama
```

3 Subiendo este manual al repositorio remoto

Con `git add nombrefichero` se agrega el manual al index.

```
willy@willy-laptop:~/Documentos/DAW2/Despliegue de Aplicaciones Web/Trabajos/Desfio 1/repo_desafio1$ git add Manual\ GIT.odt
```





Con `git commit -m "mensaje"` se guardan los cambios en el repositorio.

```
willy@willy-laptop:~/Documentos/DAW2/Despliegue de Aplicaciones Web/Trabajos/Desfio 1/repo_desafio1$ git commit -m "20180929-Manual GIT agregado"
En la rama nuevaRama
Tu rama está actualizada con 'desafio1/nuevaRama'.
```

Con `git push` se envían los cambios realizados.

```
willy@willy-laptop:~/Documentos/DAW2/Despliegue de Aplicaciones Web/Trabajos/Desfio 1/repo_desafio1$ git push
Username for 'https://github.com': willytal
Password for 'https://willytal@github.com':
Contando objetos: 3, listo.
Delta compression using up to 4 threads.
Comprimiendo objetos: 100% (3/3), listo.
Escribiendo objetos: 100% (3/3), 130.55 KiB | 26.11 MiB/s, listo.
Total 3 (delta 0), reused 0 (delta 0)
To https://github.com/willytal/desafio1.git
7ac3e32..3a628d8 nuevaRama -> nuevaRama
```

This branch is 2 commits ahead of master. [Pull request](#) [Compare](#)

 willytal	20180929-Manual GIT agregado	Latest commit 3a628d8 25 minutes ago
 ejemplogit	mensaje	5 days ago
 Manual GIT.odt	20180929-Manual GIT agregado	25 minutes ago
 initial	initial commit	5 days ago

Help people interested in this repository understand your project by adding a README. [Add a README](#)

4 Clonar repositorio de un compañero (Rafa)

En la carpeta dónde queremos clonar el repositorio remoto, ejecutamos `git clone rutarepositorioremoto`

```
willy@willy-laptop:~/Documentos/DAW2/Despliegue de Aplicaciones Web/Trabajos/
Repositorio Rafa$ git clone https://github.com/RafaelUrena/DespliegueAppWeb
Clonando en 'DespliegueAppWeb'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Desempaquetando objetos: 100% (3/3), listo.
```

Con `ls -l` podemos ver el repositorio remoto clonado en nuestro equipo, juntos a los permisos, usuarios y hora de creación..

```
willy@willy-laptop:~/Documentos/DAW2/Despliegue de Aplicaciones Web/Trabajos/
Repositorio Rafa$ ls -l
total 4
drwxr-xr-x 3 willy willy 4096 sep 29 17:15 DespliegueAppWeb
```

5 Modificar proyecto de un compañero (Rafa)

Para poder modificar el proyecto de otro usuario, primero nos tienen que añadir en **Settings/Collaborators**. Como el señor Rafael Ángel tiene los proyectos en una rama diferente de “master”, cambiamos de rama con `git checkout nombrerama`.

```
willy@willy-laptop:~/Documentos/DAW2/Despliegue de Aplicaciones Web/Trabajos/Rep
ositorioRafa/DespliegueAppWeb$ sudo git checkout nuevaRama
Rama 'nuevaRama' configurada para hacer seguimiento a la rama remota 'nuevaRama'
de 'origin'.
Cambiado a nueva rama 'nuevaRama'
```

Después importamos el proyecto con un editor de código, en este caso se utilizará **Atom** para modificar el proyecto.

```
<!DOCTYPE html>
<html lang="en" dir="ltr">
  <head>
    <meta charset="utf-8">
    <title>Rafael es un gamberro. Me deletea los trabajos</title>
  </head>
  <body>

    <header>
      <h1> HELLO RAFARLD! </h1>
    </header>

    <nav>
    </nav>

    <section>
    </section>

    <aside>
    </aside>

    <footer>
    </footer>

  </body>
</html>
```


Se comprueba con **git status**, el archivo modificado, en este caso el index.html

```
ositorioRafa/DespliegueAppWeb$ git status
En la rama nuevaRama
Tu rama está actualizada con 'origin/nuevaRama'.

Cambios no rastreados para el commit:
  (usa "git add <archivo>..." para actualizar lo que será confirmado)
  (usa "git checkout -- <archivo>..." para descartar los cambios en el directori
o de trabajo)

      modificado:      Proyecto/index.html
```

Con **git add nombrearchivo**, enviamos al area de preparación, los archivos con cambios.

```
willy@willy-laptop:~/Documentos/DAW2/Despliegue de Aplicaciones Web/Trabajos/Rep
ositorioRafa/DespliegueAppWeb$ git add Proyecto/index.html
```

Después se ejecuta un **git commit -m "mensaje"** para guardar los cambios en el repositorio.

```
willy@willy-laptop:~/Documentos/DAW2/Despliegue de Aplicaciones Web/Trabajos/Rep
ositorioRafa/DespliegueAppWeb$ git commit -m "index modificado, Rafael Ángel"
[nuevaRama 659d574] index modificado, Rafael Ángel
1 file changed, 3 insertions(+), 2 deletions(-)
```

Por último, con **git push**, enviamos los cambios al repositorio remoto.

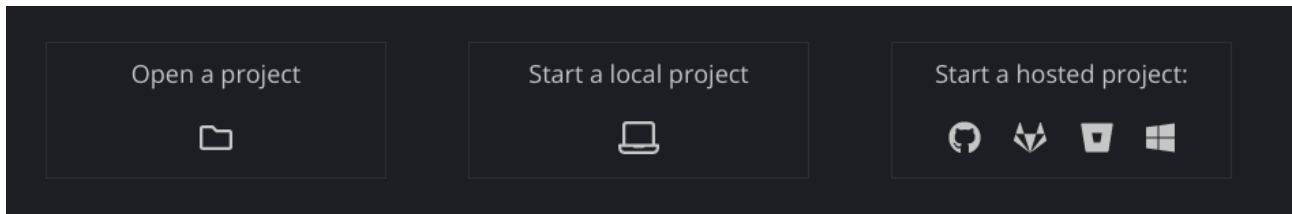
```
willy@willy-laptop:~/Documentos/DAW2/Despliegue de Aplicaciones Web/Trabajos/Rep
ositorioRafa/DespliegueAppWeb$ git push
Username for 'https://github.com': willytal
Password for 'https://willytal@github.com':
Contando objetos: 4, listo.
Delta compression using up to 4 threads.
Comprimiendo objetos: 100% (3/3), listo.
Escribiendo objetos: 100% (4/4), 437 bytes | 437.00 KiB/s, listo.
Total 4 (delta 2), reused 0 (delta 0)
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To https://github.com/RafaelUrena/DespliegueAppWeb.git
f221096..659d574 nuevaRama -> nuevaRama
```

6 GitKraken

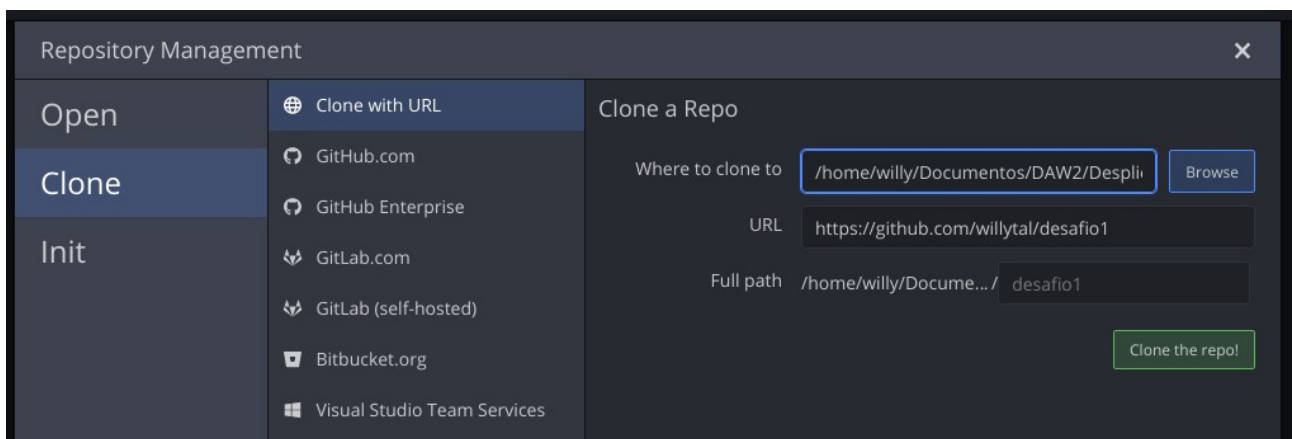
6.1 Instalación, importar proyecto.

Si no tenemos instalado GitKraken, basta con descargar el paquete deb de su página web (Ubuntu), <https://www.gitkraken.com/>, e iniciar sesión en GitHub una vez esté instalado.

En la pantalla principal de GitKraken, pulsamos sobre “Open a project”



Después, pinchamos sobre “Clone”, para clonar el repositorio remoto. (También podemos abrir un repositorio que ya tengamos creado o iniciar uno nuevo).



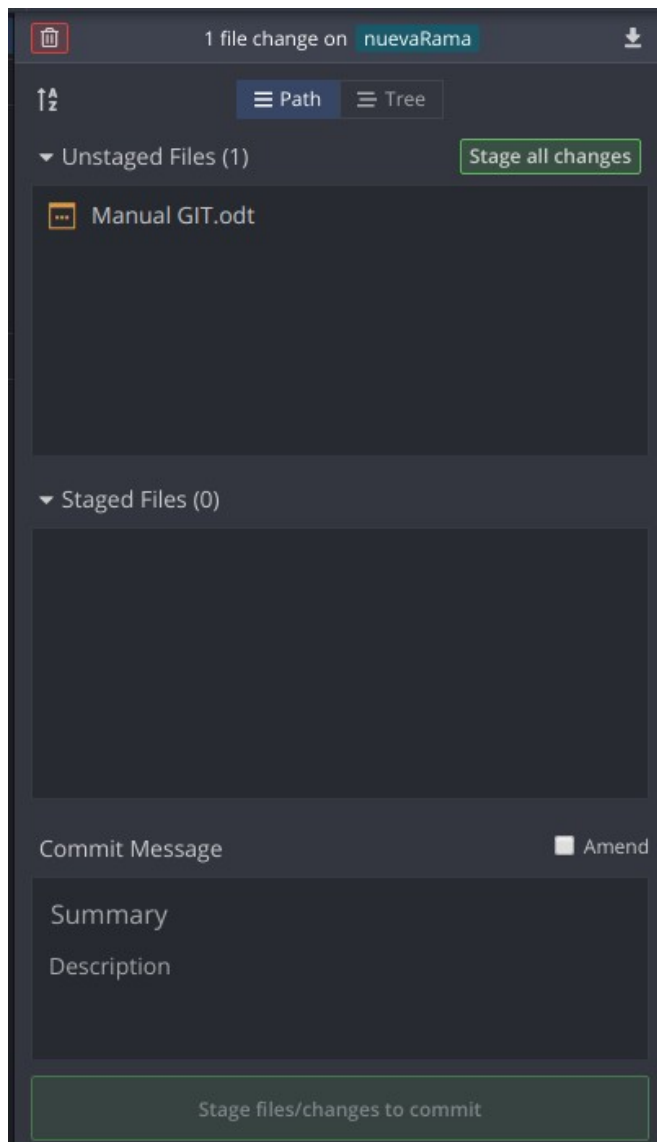
Especificamos la ruta dónde lo queremos clonar en nuestro equipo y la URL del repositorio remoto a clonar. Después hacemos click sobre “Clone the repo!”.

6.2 Cambio de rama

Una vez abierto, para cambiar a la rama “nuevaRama” (rama dónde tengo el proyecto), hacemos **doble click sobre “nuevaRama”**, y nos saldrá un mensaje que nos informará que se ha hecho un **checkout** a la rama seleccionada.



6.3 Git add, git commit, git push – Actualizando el repositorio



Cuando realizamos un cambio en un fichero, aparecerá reflejado en la parte derecha de la pantalla.

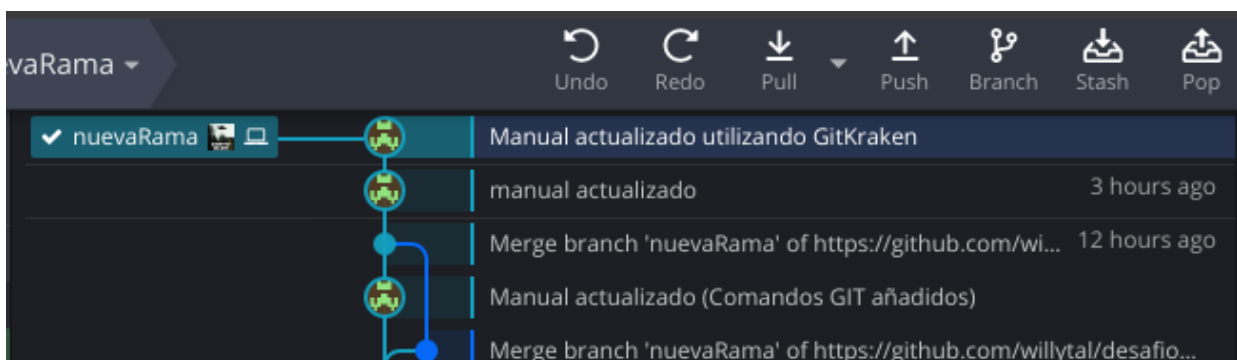
Pulsando el botón “**Stage all changes**” o arrastrando los ficheros que queremos actualizar en el repositorio, los enviamos a **Staged Files**. (Area de preparación - comando *git add* en Git).

Pulsando sobre “**Unstage all changes**” podemos sacar los archivos del area de preparación.

Por último escribimos la información que queremos reflejar en el **commit**, y pulsamos sobre “**Stage files/changes to commit**”.



Por último, pulsamos sobre el botón “**Push**”, para enviar los cambios al repositorio remoto.



7 Eliminar repositorios

Para eliminar repositorios locales, basta con borrar la carpeta dónde se encuentre el repositorio.

Si lo que queremos es eliminar el repositorio remoto, podemos hacerlo desde las opciones (“**Settings**”) en la página de GitHub, en el apartado de “**Danger Zone**”, “**Delete this repository**”.

Danger Zone

Make this repository private Please upgrade your plan to make this repository private.	
Transfer ownership Transfer this repository to another user or to an organization where you have the ability to create repositories.	Transfer
Archive this repository Mark this repository as archived and read-only.	Archive this repository
Delete this repository Once you delete a repository, there is no going back. Please be certain.	Delete this repository

8 Comandos de GIT

- **git config** → Configuración específica
- **git init** → Crear nuevo repositorio
- **git add** → Agrega archivos al index
- **git clone** → Revisar repositorios, si está en un servidor remoto → git clone usuario@93.178.160.57:/path/to/repository
- **git commit** → Guarda los cambios en el repositorio
- **git status** → Lista de los archivos
- **git push** → Envía los cambios que se han hecho
- **git checkout** → Crea ramas o cambia entre ellas
- **git remote** → Conecta a un repositorio remoto
- **git branch** → Lista, crea o borra ramas
- **git pull** → Fusiona todos los cambios que se han hecho en el repositorio local
- **git merge** → Fusiona una rama con otra
- **git diff** → Lista conflictos
- **git tag** → Marca commits específicos
- **git log** → Lista de commits en una rama junto a sus detalles
- **git reset** → Resetea el index y el directorio al último estado comprometido
- **git rm** → Borra archivos del index y del directorio en que se trabajan
- **git stash** → Salva cambios que no están por ser comprometidos inmediatamente
- **git show** → Información sobre cualquier objeto
- **git fetch** → Buscar todos los objetos de un repositorio

Otros comandos:

- **git ls-tree, git cat-file, git grep, gitk, git instaweb, git gc, git archive, git grune, git fsck, git rebase.**