

MANUAL GIT

JEISON FABIAN MALAGON BENAVIDES

ING. WILLIAM ALEXANDER MATALLANA

UNIVERSIDAD DE CUNDINAMARCA EXTENSIÓN CHIA

PROGRAMA DE INGENIERÍA DE SISTEMAS

FACULTAD DE INGENIERÍA

2025



Tabla de contenido

Introducción:	3
Objetivos:.....	3
Comandos:	3
1. Git config --list:	3
2. Git config --global user.email:	4
3. Git config --global user.name:	4
4. git config --global --unset user.email:	4
5. git config --global --unset user.name:	4
6. Git init.....	4
7. Git clone.....	5
8. Git status	5
9. Git add	5
10. Git commit -m ""	6
11. Git push origin rama.....	6
12. Git pull origin rama	6
13. Git branch -D rama	7
14. Git swtich -c rama.....	7
15. Git switch rama	7
16. Git fetch --all	7
17. Git Branch -r	8
18. Git remote -v	8
19. Git log	8
20. Git reflog	8
21. Git log --oneline	9
22. git merge rama.....	9
23. Git revert	9
24. Git push origin - - delete rama	10
"Como asociar un archivo a un repositorio existente"	10
Conclusiones:	11
Referencias:.....	12

Introducción:

Git es un sistema de control de versiones que permite llevar un historial de los cambios realizados en los archivos de un proyecto, facilitando el trabajo colaborativo y la organización del código.

A través de sus comandos podemos configurar, gestionar y sincronizar nuestros proyectos entre el repositorio local y el repositorio remoto.

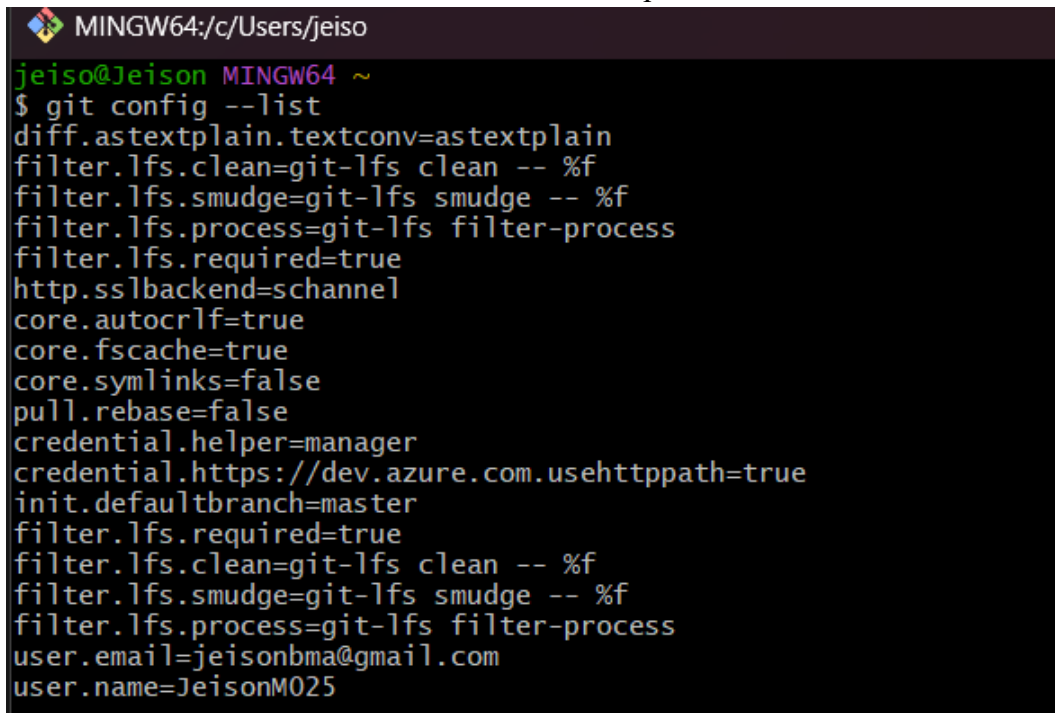
Git nos facilitan el trabajo en equipo, dándonos opciones de trabajar en ramas individuales sin llegar a modificar el código o proyecto en producción.

Objetivos:

- Comprender el uso de la herramienta GIT
- Aprender comandos básicos, configuración de entorno, clonación y subir cambios a repositorio de GITHUB
- Identificar histórico de cambios de un proyecto clonado.

Comandos:

1. `Git config --list`: muestra la configuración actual de GIT en el entorno, trae información como nombre, email, editor de texto predeterminado



```
MINGW64:/c/Users/jeiso
jeiso@Jeison MINGW64 ~
$ git config --list
diff.astextplain.textconv=astextplain
filter.lfs.clean=git-lfs clean -- %f
filter.lfs.smudge=git-lfs smudge -- %f
filter.lfs.process=git-lfs filter-process
filter.lfs.required=true
http.sslbackend=openssl
core.autocrlf=true
core.fscache=true
core.symlinks=false
pull.rebase=false
credential.helper=manager
credential.https://dev.azure.com.usehttppath=true
init.defaultbranch=master
filter.lfs.required=true
filter.lfs.clean=git-lfs clean -- %f
filter.lfs.smudge=git-lfs smudge -- %f
filter.lfs.process=git-lfs filter-process
user.email=jeisonbma@gmail.com
user.name=JeisonM025
```

2. `Git config --global user.email`: Se usa para configurar el correo asociado a los commits en git a nivel global.

```
jeiso@Jeison MINGW64 ~  
$ Git config --global user.email "jeisonbma@gmail.com"  
  
jeiso@Jeison MINGW64 ~  
$ |
```

3. `Git config --global user.name`: Se usa para configura el usuario asociado a los commits en git a nivel global.

```
jeiso@Jeison MINGW64 ~  
$ Git config --global user.name "JeisonM025"  
  
jeiso@Jeison MINGW64 ~  
$ |
```

4. `git config --global --unset user.email`: borra el correo configurado anteriormente de manera global.

```
jeiso@Jeison MINGW64 ~  
$ git config --global --unset user.email  
  
jeiso@Jeison MINGW64 ~  
$ |
```

5. `git config --global --unset user.name`: borra el usuario configurado anteriormente de manera global

```
jeiso@Jeison MINGW64 ~  
$ git config --global --unset user.name  
  
jeiso@Jeison MINGW64 ~  
$
```

6. `Git init`: este comando inicializa un nuevo proyecto GIT, crea un subdirectorio llamado `.git/` ese directorio contiene toda la información para que git empiece a rastrear todos los cambios.

```
MINGW64:/c/Users/jeiso/OneDrive/Documentos/Linea profundización 3/Ejemplo
jeiso@Jeison MINGW64 ~/OneDrive/Documentos/Linea profundización 3
$ cd Ejemplo

jeiso@Jeison MINGW64 ~/OneDrive/Documentos/Linea profundización 3/Ejemplo
$ git init
Initialized empty Git repository in C:/Users/jeiso/OneDrive/Documentos/Linea pro
fundización 3/Ejemplo/.git/

jeiso@Jeison MINGW64 ~/OneDrive/Documentos/Linea profundización 3/Ejemplo (maste
r)
$ |
```

7. **Git clone:** clona repositorio en la ruta donde le indiquemos, solo se necesita el link del repositorio.

```
jeiso@Jeison MINGW64 ~/OneDrive/Documentos/Linea profundización 3/Ejemplo (maste
r)
$ git clone https://github.com/JeisonM025/Ejercicio_1sep.git
Cloning into 'Ejercicio_1sep'...
remote: Enumerating objects: 16, done.
remote: Counting objects: 100% (16/16), done.
remote: Compressing objects: 100% (12/12), done.
remote: Total 16 (delta 2), reused 16 (delta 2), pack-reused 0 (from 0)
Receiving objects: 100% (16/16), done.
Resolving deltas: 100% (2/2), done.

jeiso@Jeison MINGW64 ~/OneDrive/Documentos/Linea profundización 3/Ejemplo (master)
$
```

8. **Git status:** este comando sirve para ver el estado actual del repositorio, como la rama, si hay archivos modificados que no se han agregado, si hay cambios listos para el commit.

```
jeiso@Jeison MINGW64 ~/OneDrive/Documentos/Linea profundización 3/Ejemplo (master)
$ cd Ejercicio_1sep/

jeiso@Jeison MINGW64 ~/OneDrive/Documentos/Linea profundización 3/Ejemplo/Ejercicio
_1sep (master)
$ git status
On branch master
Your branch is up to date with 'origin/master'.

nothing to commit, working tree clean
```

9. **Git add . :** este comandando sirve para agregar todo los cambios nuevos al repositorio, aun no se actualiza el repositorio, queda en zona de preparación,

pendiente por commit y push.

```
jeiso@Jeison MINGW64 ~/OneDrive/Documentos/Linea profundización 3/Ejemplo/Ejercicio_1sep (master)
$ git add .

jeiso@Jeison MINGW64 ~/OneDrive/Documentos/Linea profundización 3/Ejemplo/Ejercicio_1sep (master)
$ |
```

10. **Git commit -m ""**: este comando se usa para guardar en el histórico de git los cambios que están en zona de preparación, es decir los cambios añadidos con el git add .

```
jeiso@Jeison MINGW64 ~/OneDrive/Documentos/Linea profundización 3/Ejemplo/Ejercicio_1sep (master)
$ git commit -m "Cambios 1"
[master 0e08d80] Cambios 1
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 git.txt

jeiso@Jeison MINGW64 ~/OneDrive/Documentos/Linea profundización 3/Ejemplo/Ejercicio_1sep (master)
$ |
```

11. **Git push origin rama**: este comando se usa para enviar los commits de tu rama local al repositorio remoto

```
jeiso@Jeison MINGW64 ~/OneDrive/Documentos/Linea profundización 3/Ejemplo/Ejercicio_1sep (master)
$ git push origin master
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), 374 bytes | 374.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/JeisonM025/Ejercicio_1sep.git
220e7b6..0e08d80 master -> master

jeiso@Jeison MINGW64 ~/OneDrive/Documentos/Linea profundización 3/Ejemplo/Ejercicio_1sep (master)
$
```

12. **Git pull origin rama**: este comando se usa para traer los cambios desde una rama del repositorio remoto y fusionarlos con tu rama local actual.

```
jeiso@Jeison MINGW64 ~/OneDrive/Documentos/Linea profundización 3/Ejemplo/Ejercicio_1sep (master)
$ git pull origin master
From https://github.com/JeisonM025/Ejercicio_1sep
* branch          master      -> FETCH_HEAD
Already up to date.

jeiso@Jeison MINGW64 ~/OneDrive/Documentos/Linea profundización 3/Ejemplo/Ejercicio_1sep (master)
$
```

13. `Git branch -D rama`: este comando sirve para eliminar una rama local en Git de manera forzada.

```
jeiso@Jeison MINGW64 ~/OneDrive/Documentos/Linea profundizaci
master)
$ git branch -D prueba
Deleted branch prueba (was 0e08d80).

jeiso@Jeison MINGW64 ~/OneDrive/Documentos/Linea profundizaci
master)
$ |
```

14. `Git switch -c rama`: este comando sirve para crear una nueva rama y cambiarte a ella inmediatamente.

```
jeiso@Jeison MINGW64 ~/OneDrive/Documentos/Linea profundizaci
master)
$ git switch -c rama1
Switched to a new branch 'rama1'

jeiso@Jeison MINGW64 ~/OneDrive/Documentos/Linea profundizaci
rama1)
$ |
```

15. `Git switch rama`: este comando se usa para cambiarte a una rama que ya existe en tu repositorio local.

```
jeiso@Jeison MINGW64 ~/OneDrive/Documentos/Linea profundizaci
rama1)
$ git switch master
Switched to branch 'master'
Your branch is up to date with 'origin/master'.

jeiso@Jeison MINGW64 ~/OneDrive/Documentos/Linea profundizaci
master)
$ |
```

16. `Git fetch --all`: Este comando se usa para descargar todas las actualizaciones desde todos los remotos configurados, pero sin fusionarlas en tus ramas locales todavía.

```
jeiso@Jeison MINGW64 ~/OneDrive/Documentos/Linea profundizaci
master)
$ git fetch --all

jeiso@Jeison MINGW64 ~/OneDrive/Documentos/Linea profundizaci
master)
$ |
```

17. **Git Branch -r**: Este comando sirve para listar todas las ramas remotas que Git conoce en tu repositorio.

```
jeiso@Jeison MINGW64 ~/OneDrive/Documentos/Linea profun
master)
$ git branch -r
origin/HEAD -> origin/master
origin/master
```

18. **Git remote -v**: Este comando sirve para mostrar las URL de los repositorios remotos que tu repo local tiene configurados.

```
jeiso@Jeison MINGW64 ~/OneDrive/Documentos/Linea profundización 3/Ejemplo
master)
$ git remote -v
origin https://github.com/JeisonM025/Ejercicio_1sep.git (fetch)
origin https://github.com/JeisonM025/Ejercicio_1sep.git (push)
```

19. **Git log**: Este comando sirve para ver el historial de commits en Git.

```
MINGW64/c/Users/jeiso/OneDrive/Documentos/Linea profundización 3/Ejemplo/Ejercicio_...
jeiso@Jeison MINGW64 ~/OneDrive/Documentos/Linea profundización 3/Ejemplo/Ejercicio
master)
$ git log
commit 0e08d80ee8b42704a070ff7205194446435f925f (HEAD -> master, origin/master, or
D, ramal)
Author: JeisonM025 <jeisonbma@gmail.com>
Date: Tue Sep 2 18:47:06 2025 -0500

Cambios 1

commit 220e7b62714c563eac3dafa41ab3dad94c021a3b
Author: JeisonM025 <jeisonbma@gmail.com>
Date: Mon Sep 1 19:33:15 2025 -0500

Ejercicio final

commit 959c6452a40c7b1dd1fda8466106f69e070759d1
Author: JeisonM025 <jeisonbma@gmail.com>
Date: Mon Sep 1 18:25:11 2025 -0500

Ejercicio

jeiso@Jeison MINGW64 ~/OneDrive/Documentos/Linea profundización 3/Ejemplo/Ejercicio
master)
```

20. **Git reflog**: Sirve para mostrar el historial de movimientos del puntero HEAD, es decir, todas las acciones que has hecho en tu repositorio local.


```
jeiso@Jeison MINGW64 ~/OneDrive/Documentos/Linea profundización 3/Ejemplo/Ejercicio_1sep
master)
$ git reflog
0e08d80 (HEAD -> master, origin/master, origin/HEAD, rama1) HEAD@{0}: checkout: moving t
m rama1 to master
0e08d80 (HEAD -> master, origin/master, origin/HEAD, rama1) HEAD@{1}: checkout: moving t
m master to rama1
0e08d80 (HEAD -> master, origin/master, origin/HEAD, rama1) HEAD@{2}: checkout: moving t
m prueba to master
0e08d80 (HEAD -> master, origin/master, origin/HEAD, rama1) HEAD@{3}: checkout: moving t
m master to prueba
0e08d80 (HEAD -> master, origin/master, origin/HEAD, rama1) HEAD@{4}: commit: Cambios 1
220e7b6 HEAD@{5}: clone: from https://github.com/JeisonM025/Ejercicio_1sep.git

jeiso@Jeison MINGW64 ~/OneDrive/Documentos/Linea profundización 3/Ejemplo/Ejercicio_1sep
master)
$
```

21. `Git log --oneline`: Este comando sirve para ver el historial de commits de forma resumida: cada commit aparece en una sola línea.

```
master)
$ git log --oneline
0e08d80 (HEAD -> master, origin/master, origin/HEAD, rama1) Cambios 1
220e7b6 Ejercicio final
959c645 Ejercicio

jeiso@Jeison MINGW64 ~/OneDrive/Documentos/Linea profundización 3/Ejemplo/Ej

```

22. `git merge rama`: Sirve para unir los cambios de otra rama a la rama en la que estás actualmente.

```
jeiso@Jeison MINGW64 ~/OneDrive/Documentos/Linea profundización 3/Ejemp
master)
$ git merge prueba
merge: prueba - not something we can merge

jeiso@Jeison MINGW64 ~/OneDrive/Documentos/Linea profundización 3/Ejemp
master)

```

23. `Git revert`: Sirve para deshacer un commit, pero creando un nuevo commit inverso.



```
MINGW64:/c/Users/jeiso/OneDrive/Documentos/Linea profundización 3/Ejemplo/Ejercicio_...
revert "Cambios 1"

This reverts commit 0e08d80ee8b42704a070ff7205194446435f925f.

Please enter the commit message for your changes. Lines starting
with '#' will be ignored, and an empty message aborts the commit.

On branch master
Your branch is up to date with 'origin/master'.

Changes to be committed:
  deleted:    git.txt

git/COMMIT_EDITMSG [unix] (19:11 02/09/2025)
```

24. Git push origin --delete ramo: Sirve para eliminar una rama en el repositorio remoto

```
jeiso@Jeison MINGW64 ~/OneDrive/Documentos/Linea profundización 3/Ejercicio_1sep (master)
$ git push origin --delete ramo1
To https://github.com/JeisonM025/Ejercicio_1sep.git
- [deleted]          ramo1

jeiso@Jeison MINGW64 ~/OneDrive/Documentos/Linea profundización 3/Ejercicio_1sep (master)
```

"Como asociar un archivo a un repositorio existente"

- Se inicia con git init
- Conectamos con el repositorio: git remote add origin

```
MINGW64:/c/Users/jeiso/OneDrive/Documentos/Linea profundización 3/Pract...
jeiso@Jeison MINGW64 ~/OneDrive/Documentos/Linea profundización 3/Practica
$ git init
Initialized empty Git repository in C:/Users/jeiso/OneDrive/Documentos/Linea pro
fundización 3/Practica/.git/

jeiso@Jeison MINGW64 ~/OneDrive/Documentos/Linea profundización 3/Practica (mast
er)
$ git remote add origin https://github.com/JeisonM025/Ejercicio_1sep.git

jeiso@Jeison MINGW64 ~/OneDrive/Documentos/Linea profundización 3/Practica (mast
er)
$
```

- Agregar nuevo archivo git add

```
jeiso@Jeison MINGW64 ~/OneDrive/Documentos/Linea profundización 3/
er)
$ git add nuevo.txt

jeiso@Jeison MINGW64 ~/OneDrive/Documentos/Linea profundización 3/
er)
```

- Git commit y git push

```
jeiso@Jeison MINGW64 ~/OneDrive/Documentos/Linea profundización 3/
er)
$ git commit -m "Nuevo archivo"
[master (root-commit) 6d4df5f] Nuevo archivo
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 nuevo.txt

jeiso@Jeison MINGW64 ~/OneDrive/Documentos/Linea profundización 3/
er)
jeiso@Jeison MINGW64 ~/OneDrive/Documentos/Linea profundización 3/
Ejercicio_1sep (main)
$ git push origin main
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
remote:
remote: Create a pull request for 'main' on GitHub by visiting:
remote:   https://github.com/JeisonM025/Ejercicio_1sep/pull/new/main
remote:
To https://github.com/JeisonM025/Ejercicio_1sep.git
 * [new branch]      main -> main

jeiso@Jeison MINGW64 ~/OneDrive/Documentos/Linea profundización 3/
Ejercicio_1sep (main)
```

Conclusiones:

- El manual permitió comprender la utilidad de Git como herramienta de control de versiones y trabajo colaborativo

- Se logró aprender y aplicar los comandos básicos de Git, así como la configuración del entorno de trabajo, la clonación de repositorios y el procedimiento para subir cambios a GitHub
- Se identificó la importancia de consultar el historial de cambios de un proyecto clonado con `git log`.

el uso de Git y GitHub nos ayuda un flujo de trabajo ordenado y eficiente, constituyéndose en una competencia fundamental para el desarrollo de proyectos

Referencias:

- Git, manual de uso git, <https://git-scm.com/book/es/v2/Inicio---Sobre-el-Control-de-Versiones-Fundamentos-de-Git>