

GUIA DE USUARIO PARA COMANDOS BASICOS GIT HUB

HELDA YESENIA VALBURNA CASTRO

UNIVERSIDAD DE CUNDINAMARCA

FACULTAD DE INGENIERIA – NOCHE

LINEA DE PROFUNDIZACION III

ALEXANDER MATLLANA

CHIA, CUNDINAMARCA

1-SEPTIEMBRE-2025

Tabla de contenido

1. Introducción
2. Objetivo
 - 2.1.1. 2.1Objetivo general
 - 2.1.2. 2.2Objetivo específico
3. Desarrollo del Taller
 - 3.1. Git config – -list
 - 3.2. Git config – -global user.email
 - 3.3. Git config – -global user.name
 - 3.4. Git config - - global - -unset user.email
 - 3.5. Git config - - global - -unset user.name
 - 3.6. Git init
 - 3.7. Git clone
 - 3.8. Git status
 - 3.9. Git add .
 - 3.10. Git commit -m “”
 - 3.11. Git push origin rama
 - 3.12. Git pull origin rama
 - 3.13. Git branch -D rama
 - 3.14. Git swtich -c rama
 - 3.15. Git switch rama
 - 3.16. Git fetch - -all
 - 3.17. Git Branch -r
 - 3.18. Git remote -v
 - 3.19. Git log
 - 3.20. Git reflog
 - 3.21. Git log --oneline
 - 3.22. Git merge
 - 3.23. Git revert
4. . Conclusiones
5. Referencias Bibliográfica

1.Introducción

En base a los conocimientos adquiridos durante las últimas semanas, se presenta el siguiente manual de usuario, con un propósito, guiar a cualquier internauta en la comprensión y aplicación de los comandos básicos de **GitHub**.

Este manual describe de manera clara el concepto de cada uno de los comandos, acompañado de su respectiva explicación sobre cómo se ejecutan dentro de un entorno de desarrollo en **Java**.

La finalidad es ofrecer una herramienta práctica y accesible que facilite a los usuarios la integración del trabajo colaborativo a través de GitHub.

2.Objetivo

2.1 Objetivo general

Compartir con los usuarios un manual claro del uso de los comandos básicos dando una descripción de cada uno y la forma en la que puede ser ejecutado en un entorno **JAVA**.

2.2Objetivo específico

2.2.1 Identifica los comandos básicos para **GIT HUB**

2.2.2 Descripción de cada uno de los comandos y como ejecutarlos.

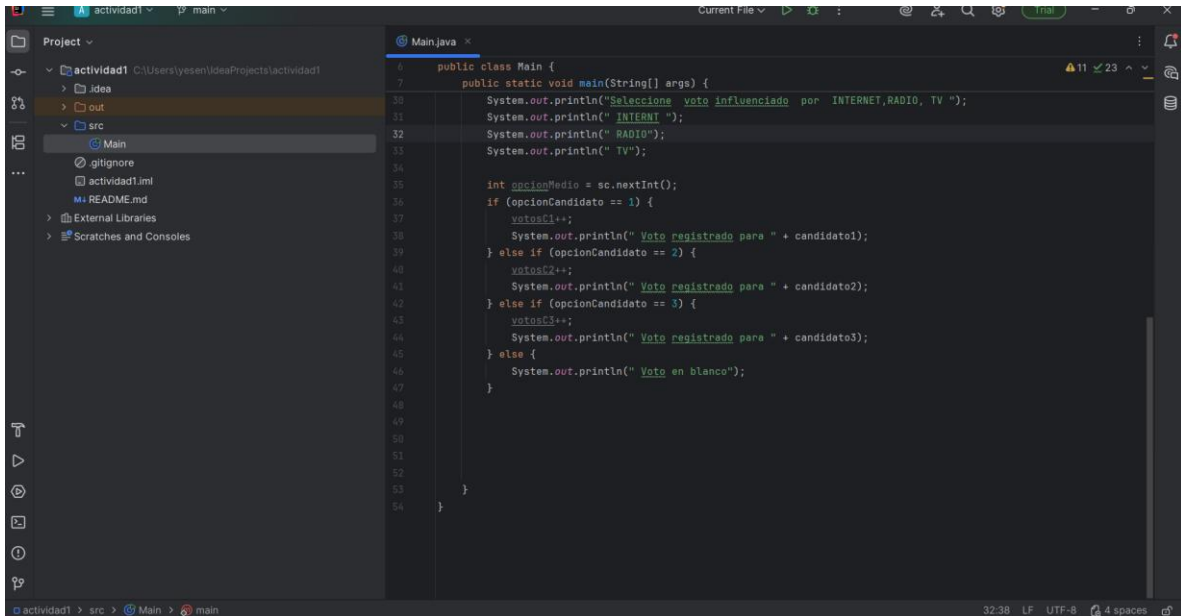
2.2.3 Aplicar los conocimientos adquiridos.

3.Desarrollo del Taller

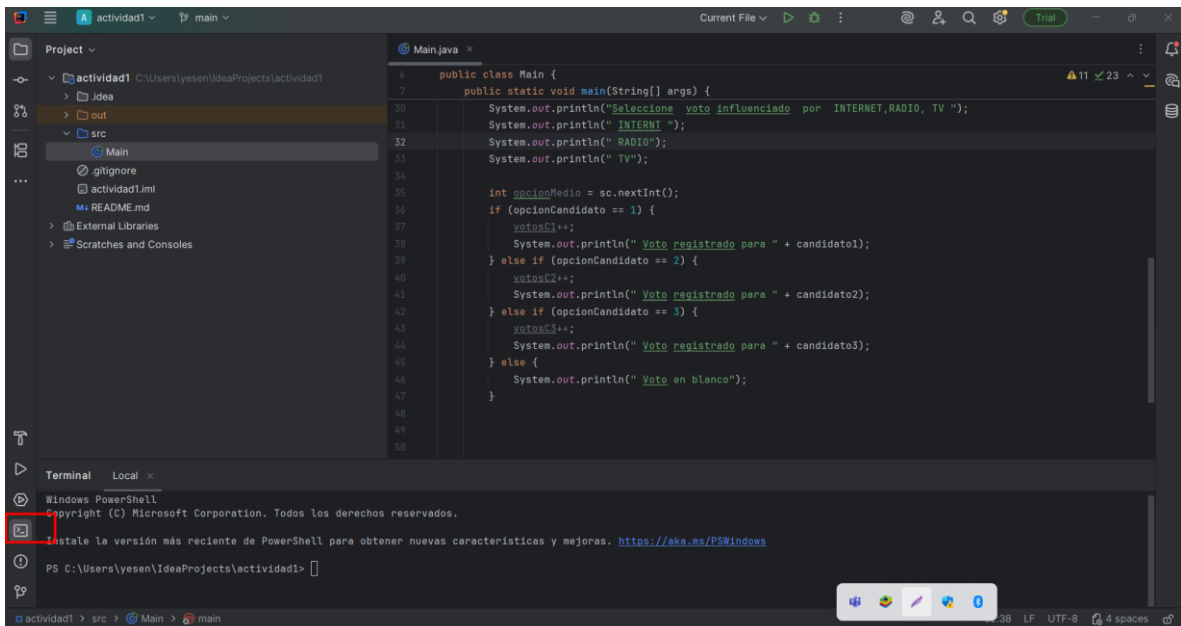
Descripción de comandos

Para ingresar los comandos debemos inicialmente abrir una terminal.

Nuestro panel principal



Aquí mostramos seleccionamos el icono de abrir terminal



Después ingresamos los comandos.

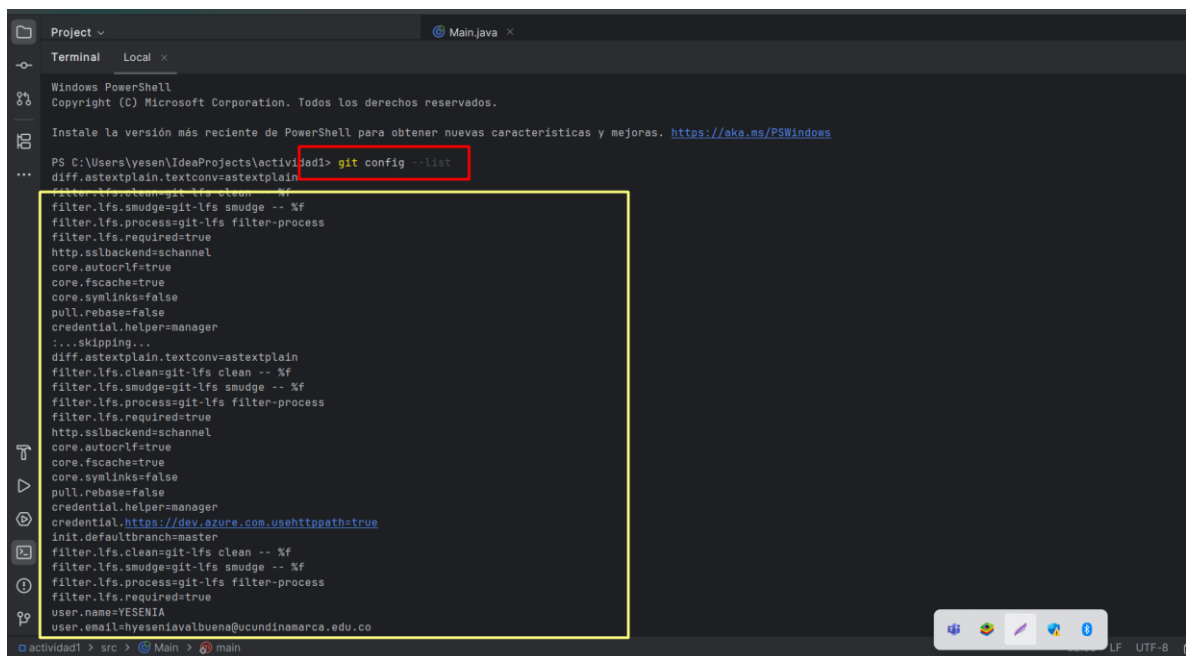
- **Git config --list**

Con este comando podemos identificar la configuración actual de nuestro GIT HUB.

Ingresamos nuestro comando en la terminal.

-EL comando es el recuadro subrayado en rojo

-En el recuadro amarillo podemos identificar la configuración de nuestro equipo, como el correo, el usuario entre otros.



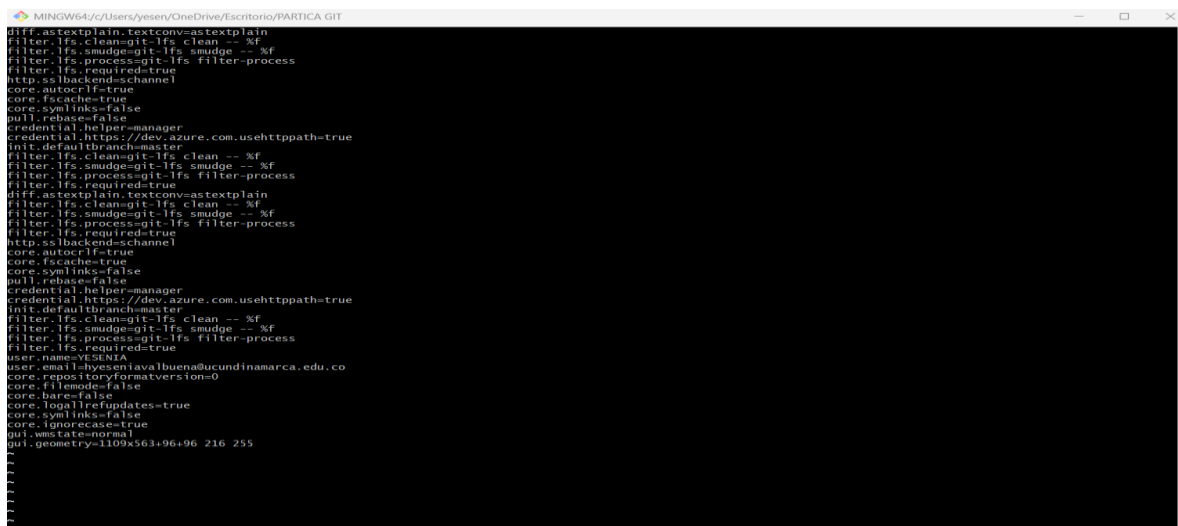
```
Windows PowerShell
Copyright (C) Microsoft Corporation. Todos los derechos reservados.

Instale la versión más reciente de PowerShell para obtener nuevas características y mejoras. https://aka.ms/PSWindows

PS C:\Users\yessen\IdeaProjects\actividad1> 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=schannel
core.autocrlf=true
core.fsckcache=true
core.symlinks=false
pull.rebase=false
credential.helper=manager
...skipping...
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=schannel
core.autocrlf=true
core.fsckcache=true
core.symlinks=false
pull.rebase=false
credential.helper=manager
credential.https://dev.azure.com.usehttppath=true
init.defaultbranch=master
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
user.name=VESENIA
user.email=hyesseniaibuenaaucundinamarca.edu.co
```

GIT BASH



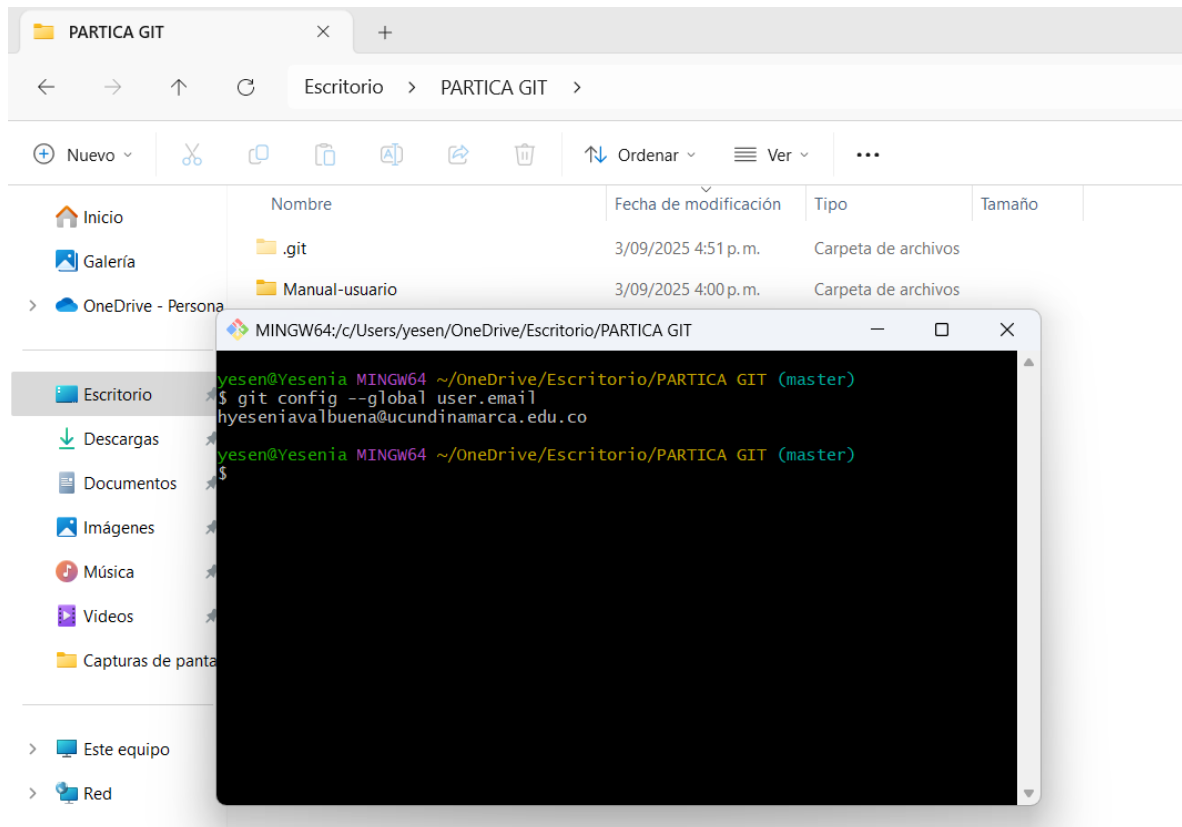
```
MINGW64/c:/Users/yessen/OneDrive/Escritorio/PARTICA GIT

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=schannel
core.autocrlf=true
core.fsckcache=true
core.symlinks=false
pull.rebase=false
credential.helper=manager
credential.https://dev.azure.com.usehttppath=true
init.defaultbranch=master
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
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=schannel
core.autocrlf=true
core.fsckcache=true
core.symlinks=false
pull.rebase=false
credential.helper=manager
credential.https://dev.azure.com.usehttppath=true
init.defaultbranch=master
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
user.name=VESENIA
user.email=hyesseniaibuenaaucundinamarca.edu.co
core.repositoryformatversion=0
core.filemode=false
core.bare=false
core.logallrefupdates=true
core.symlinks=false
core.ignorecase=true
gui.wmstate=normal
gui.geometry=1109x563+96+96 216 255
```

- **Git config --global user.email**

Muestra el correo global del usuario que se asociará a los *commits*.

- EL comando es el recuadro subrayado en rojo Git config --global user.email

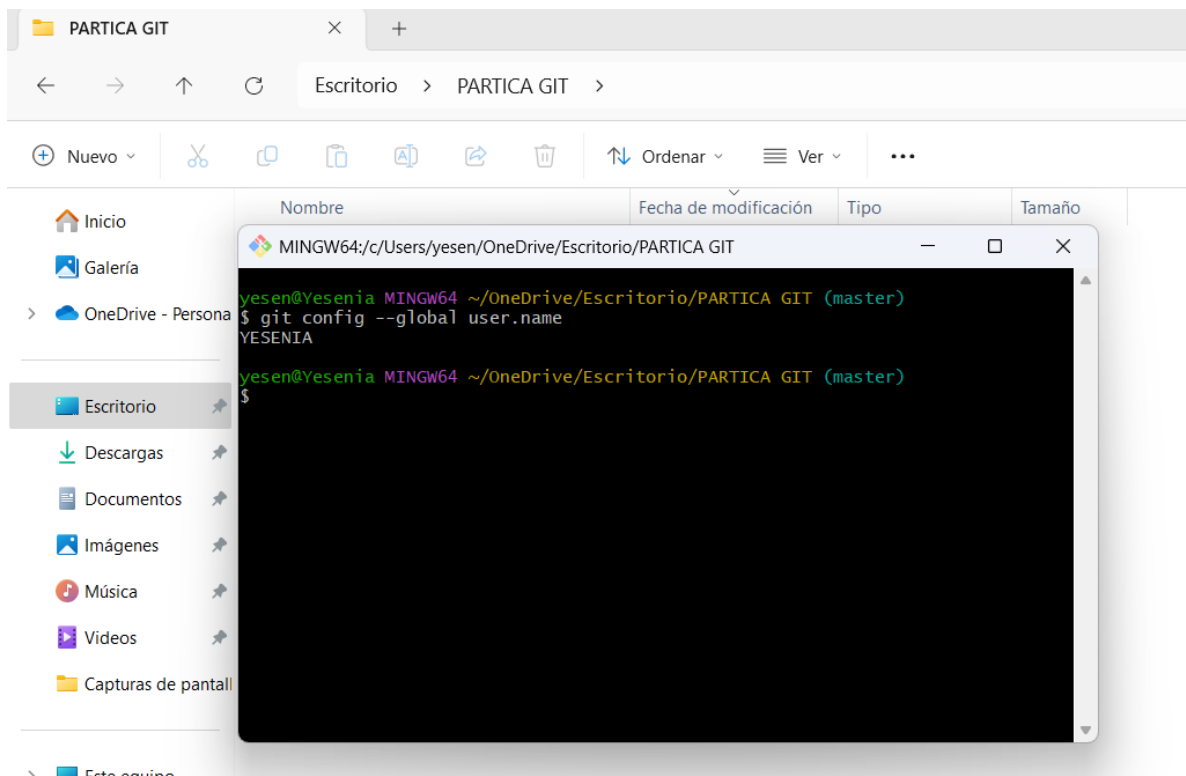


- **Git config --global user.name**

Muestra el usuario que se asociará a los *commits*.

Infresamos comando

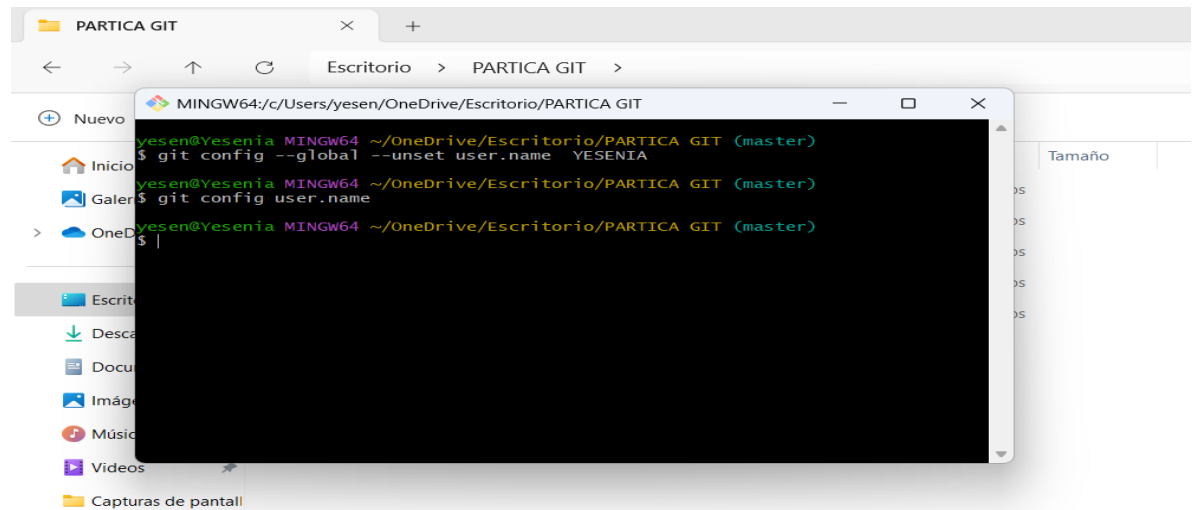
mue



Muestra el usuario asociado en este caso YESENIA

- **Git config - - global - -unset user.email**

Elimina la configuración global del correo electrónico.

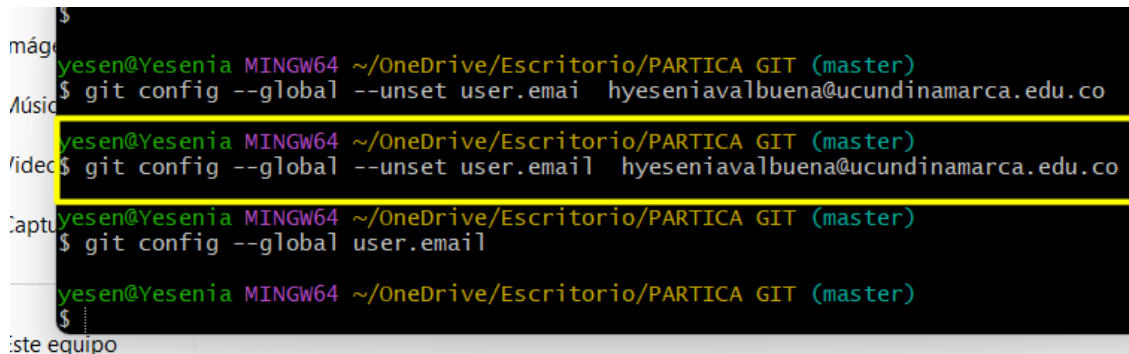


Agregamos comando de git **config user.name** para verificar que se halla eliminado el usuario.

- **Git config --global --unset user.name**

Elimina la configuración global del nombre de usuario

Ingresamos el comando

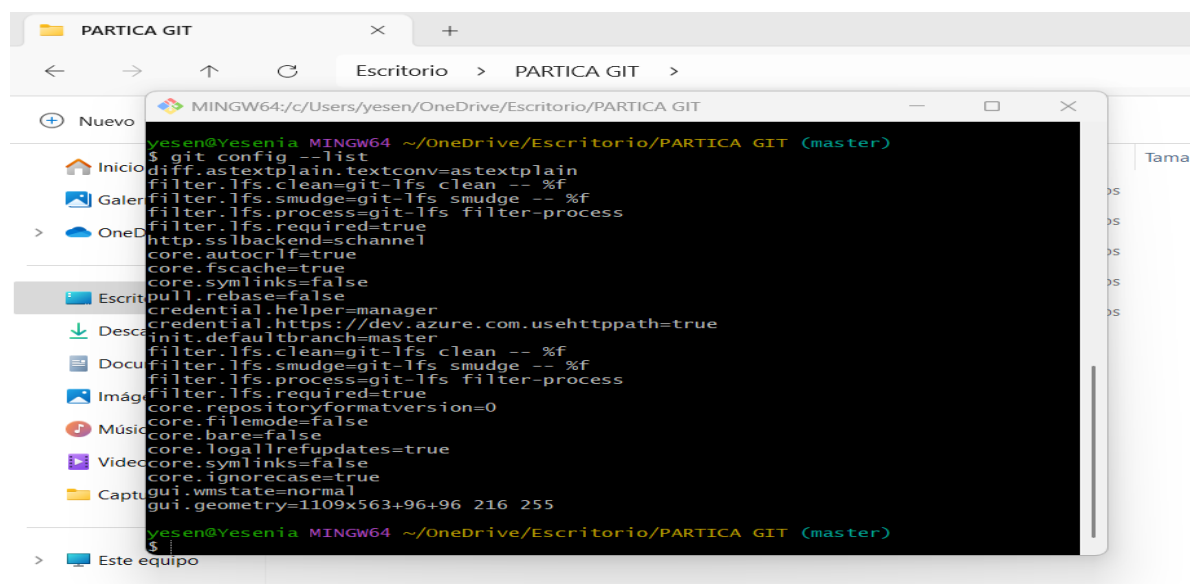


```
yesen@Yesenia MINGW64 ~/OneDrive/Escritorio/PARTICA GIT (master)
$ git config --global --unset user.email hyeseniaavalbuena@ucundinamarca.edu.co
yesen@Yesenia MINGW64 ~/OneDrive/Escritorio/PARTICA GIT (master)
$ git config --global --unset user.email hyeseniaavalbuena@ucundinamarca.edu.co
yesen@Yesenia MINGW64 ~/OneDrive/Escritorio/PARTICA GIT (master)
$ git config --global user.email
yesen@Yesenia MINGW64 ~/OneDrive/Escritorio/PARTICA GIT (master)
$
```

Agregamos comando de git **config user.email** para verificar que se halla eliminado el email.

Para confirmar que estos cambios se hallan realizado agregamos el comando

git config --list

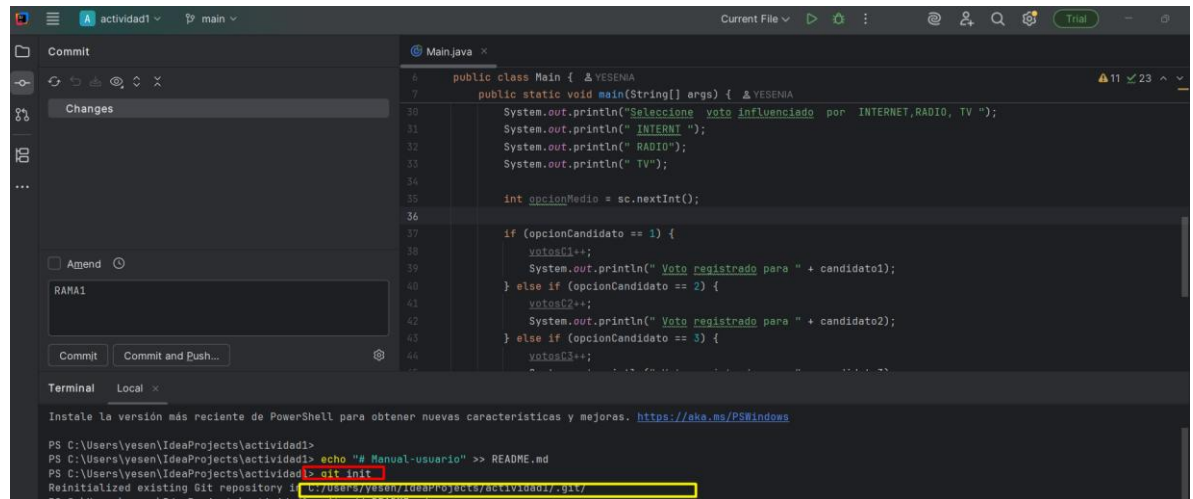


```
yesen@Yesenia MINGW64 ~/OneDrive/Escritorio/PARTICA GIT (master)
$ git config --list
diff.astextplain.textconv=astextplain
filter.lfs.clean=git-lfs clean -- %f
filter.lfs.smudge=git-lfs smudge -- %f
filter.lfs.process=git-lfs filter-process
filter.lfs.required=true
http.sslbackend=schannel
core.autocrlf=true
core.fscache=true
core.symlinks=false
pull.rebase=false
credential.helper=manager
init.defaultbranch=master
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
core.repositoryformatversion=0
core.filemode=false
core.bare=false
core.logallrefupdates=true
core.symlinks=false
core.ignorecase=true
gui.wmstate=normal
gui.geometry=1109x563+96+96 216 255
yesen@Yesenia MINGW64 ~/OneDrive/Escritorio/PARTICA GIT (master)
$
```

Que nos muestra que ya no se encuentra configurado.

- **Git init**

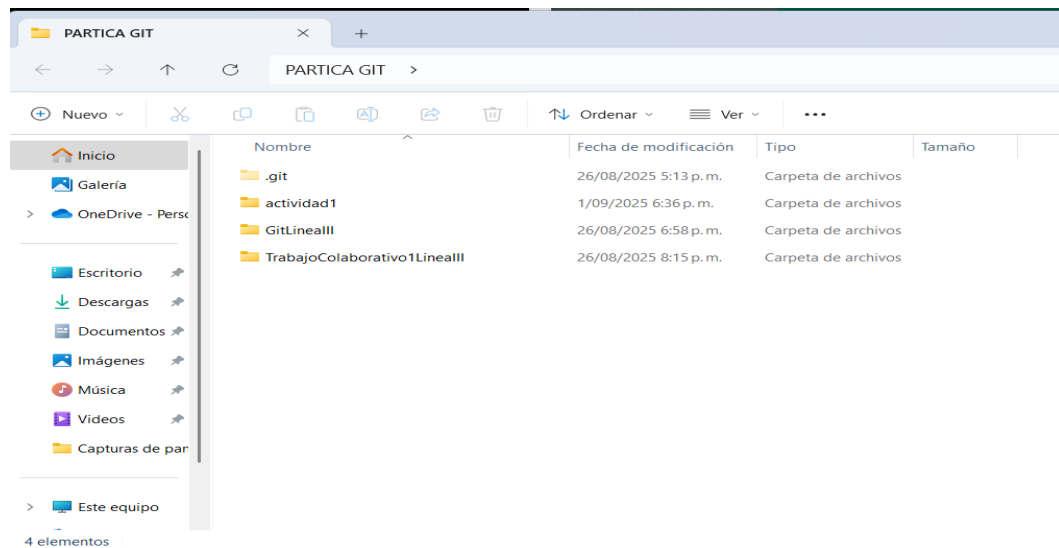
Crear un repositorio en el GIT que se encuentra asociado o carpeta actual.



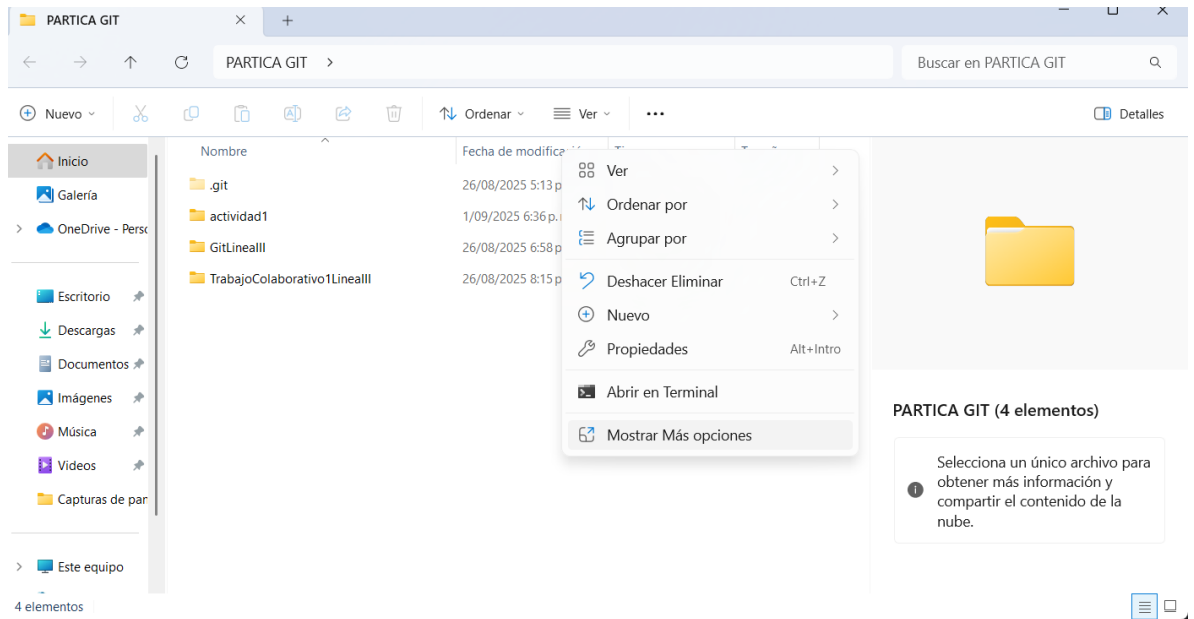
- **Git clone**

Descarga un repositorio remoto de GitHub hacia el equipo local.

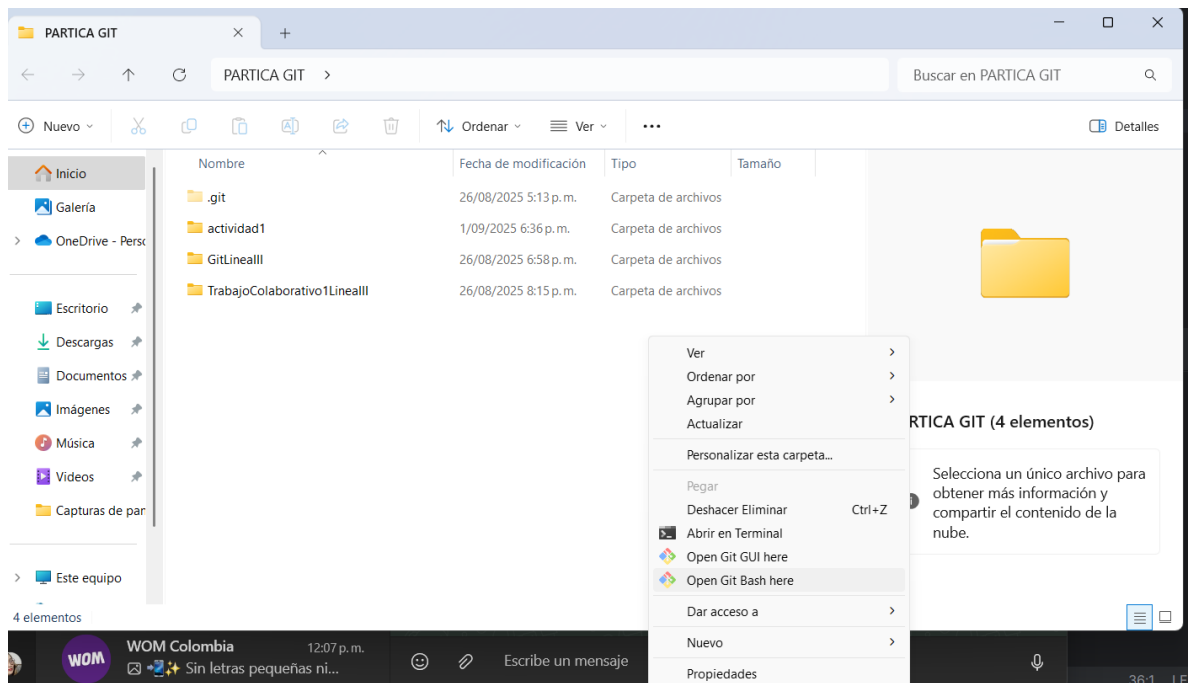
Para hacer la clonación debemos crear una carpeta



Clic derecho , mostrar mas opciones



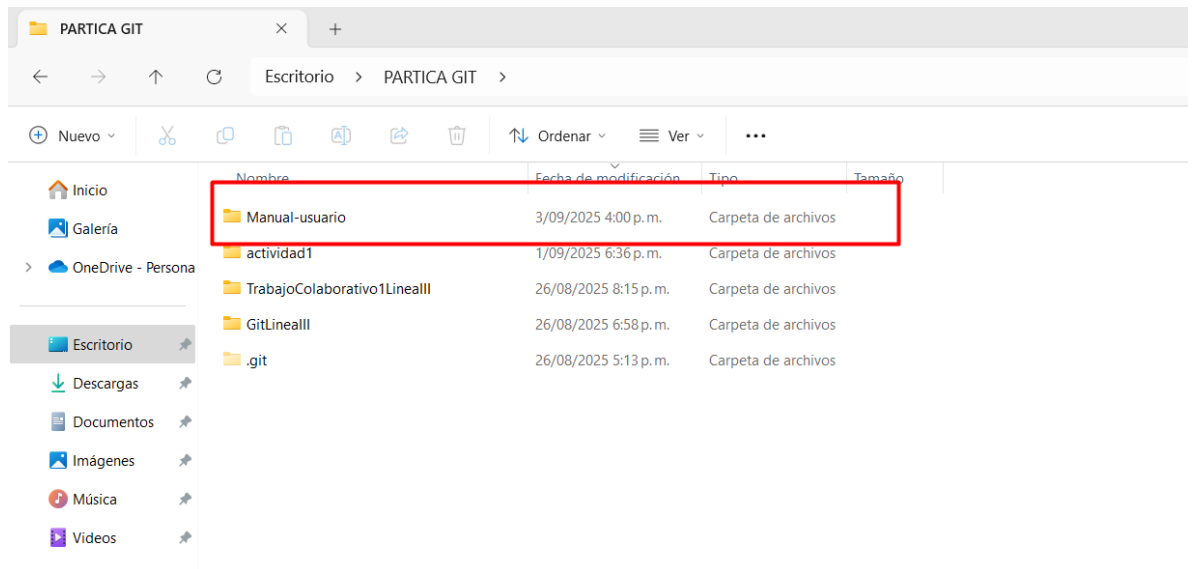
Open git bash here



```
MINGW64:/c/Users/yesen/OneDrive/Escritorio/PARTICA GIT
yesen@Yesenia MINGW64 ~/OneDrive/Escritorio/PARTICA GIT (master)
$ git clone ^[[200~https://github.com/HELDAYESE/Manual-usuario.git~
Cloning into 'Manual-usuario.git'...
fatal: protocol '?[200~https' is not supported

yesen@Yesenia MINGW64 ~/OneDrive/Escritorio/PARTICA GIT (master)
$
```

Este crea una carpeta con el nombre que le asignamos en el repositorio.

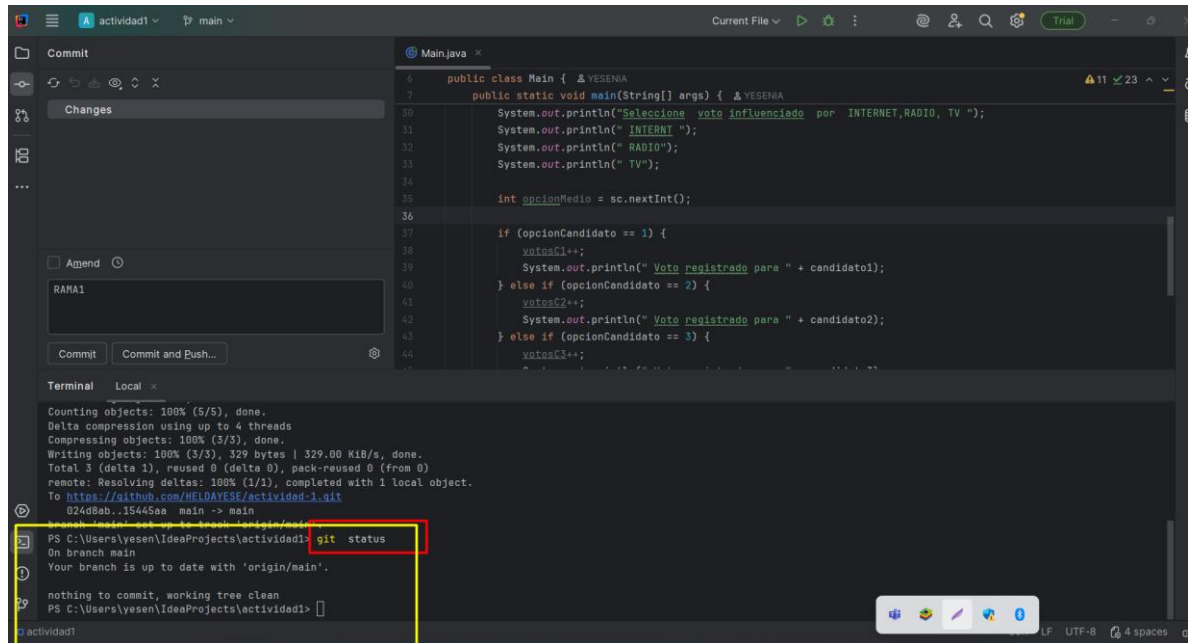


- **Git status**

Muestra el estado de los archivos.

Recuadro rojo muestra el comando

El amarillo la muestra el estado de los archivos que tenemos actualmente.

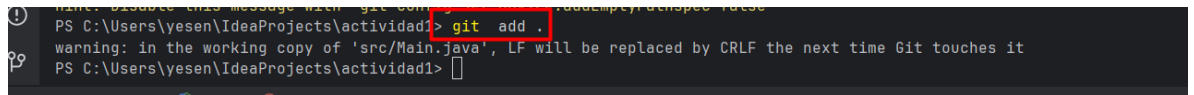


- **Git add .**

A Agregar cambios a staging

Escribimos en comando en la terminal

Nos muestra el cambio

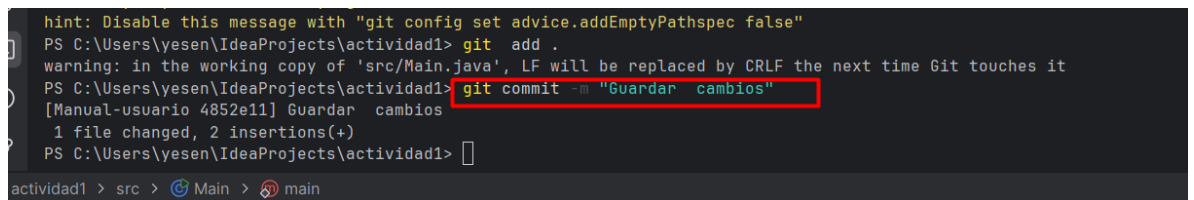


- **Git commit -m ""**

Guarda los cambios en el repositorio local con un mensaje descriptivo

Escribimos en comando en la terminal

Nos muestra el cambio



- **Git push origin rama**

Subir cambios al remoto a la rama en especifico

Nos muestra como aplica los cambios dentro de nuestro repositorio

```
[Manual-usuario 4852e11] Guardar cambios
1 file changed, 2 insertions(+)
PS C:\Users\yesen\IdeaProjects\actividad1> git push origin Manual-usuario
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 4 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (4/4), 378 bytes | 378.00 KiB/s, done.
Total 4 (delta 2), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To https://github.com/HELDAYESE/actividad-1.git
15445aa..4852e11 Manual-usuario -> Manual-usuario
PS C:\Users\yesen\IdeaProjects\actividad1>
```

- **Git pull origin rama**

Descarga y combina en local los cambios de la rama remota

Vemos el comando

Y from sobre el cual nos encontramos trabajando

```
PS C:\Users\yesen\IdeaProjects\actividad1> git pull origin Manual-usuario
From https://github.com/HELDAYESE/actividad-1
* branch          Manual-usuario -> FETCH_HEAD
Already up to date.
PS C:\Users\yesen\IdeaProjects\actividad1>
```

- **Git branch -D rama**

Elimina de forma forzada una rama local.

```
error: branch 'RAMA1' not found
PS C:\Users\yesen\IdeaProjects\actividad1> git branch -d Manual-usuario
error: the branch 'Manual-usuario' is not fully merged
hint: If you are sure you want to delete it, run 'git branch -D Manual-usuario'
hint: Disable this message with "git config set advice.forceDeleteBranch false"
PS C:\Users\yesen\IdeaProjects\actividad1>
```

- **Git switch -c rama**

Crea una nueva rama y cambia a ella

```
switch
PS C:\Users\yesen\IdeaProjects\actividad1> git switch -c Manual
Switched to a new branch 'Manual'
PS C:\Users\yesen\IdeaProjects\actividad1>
```

- **Git switch rama**

Cambia a una rama existente.

Agregamos el comando y nos fuimos a nuestra rama main

```
PS C:\Users\yesen\IdeaProjects\actividad1> git switch main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.
PS C:\Users\yesen\IdeaProjects\actividad1>
```

- **Git fetch - -all**

Descarga información de todas las ramas remotas sin fusionarlas

- **Git Branch -r**

lista todas las ramas remotas disponibles

Agregamos el comando

Nos muestra las ramas que tenemos actualmente

```
* branch          Manual-usuario -> FETCH_HEAD
Already up to date.
PS C:\Users\yesen\IdeaProjects\actividad1> git branch -r
  origin/Manual-usuario
  origin/RAMA1
  origin/main
PS C:\Users\yesen\IdeaProjects\actividad1>
```

- **Git remote -v**

Muestra las direcciones (URLs) de los repositorios remotos asociados al proyecto

```
PS C:\Users\yesen\IdeaProjects\actividad1> git remote -v
origin  https://github.com/HELDAYESE/actividad-1.git (fetch)
origin  https://github.com/HELDAYESE/actividad-1.git (push)
PS C:\Users\yesen\IdeaProjects\actividad1>
```

- **Git log**

Muestra el historial completo de commits, con información detallada (autor, fecha, hash, mensaje)

```
PS C:\Users\yesen\IdeaProjects\actividad1> git log
commit 15445aae96e4d96e5ed8e3dca7d313121a6277a4 (HEAD -> main, origin/main, origin/HEAD)
Author: YESENIA <hyesenialbuena@ucundinamarca.edu.co>
Date:   Wed Sep 3 15:42:26 2025 -0500

    first commit

commit 431b8309f571037362e00bf728f3e99957201bc8 (origin/RAMA1)
Author: YESENIA <hyesenialbuena@ucundinamarca.edu.co>
Date:   Tue Sep 2 20:33:33 2025 -0500

    RAMA1

commit 024d8ab73b0c52082001b684c17a223ba8ba383d
Author: YESENIA <hyesenialbuena@ucundinamarca.edu.co>
Date:   Mon Sep 1 20:29:00 2025 -0500

    actividad1

commit ae28502f33f965fb2ac5636a2d8e4de2fc078d0c
Author: YESENIA <hyesenialbuena@ucundinamarca.edu.co>
Date:   Mon Sep 1 20:27:59 2025 -0500

    ...skipping...
commit 15445aae96e4d96e5ed8e3dca7d313121a6277a4 (HEAD -> main, origin/main, origin/HEAD)
Author: YESENIA <hyesenialbuena@ucundinamarca.edu.co>
Date:   Wed Sep 3 15:42:26 2025 -0500

    first commit

commit 431b8309f571037362e00bf728f3e99957201bc8 (origin/RAMA1)
Author: YESENIA <hyesenialbuena@ucundinamarca.edu.co>
Date:   Tue Sep 2 20:33:33 2025 -0500

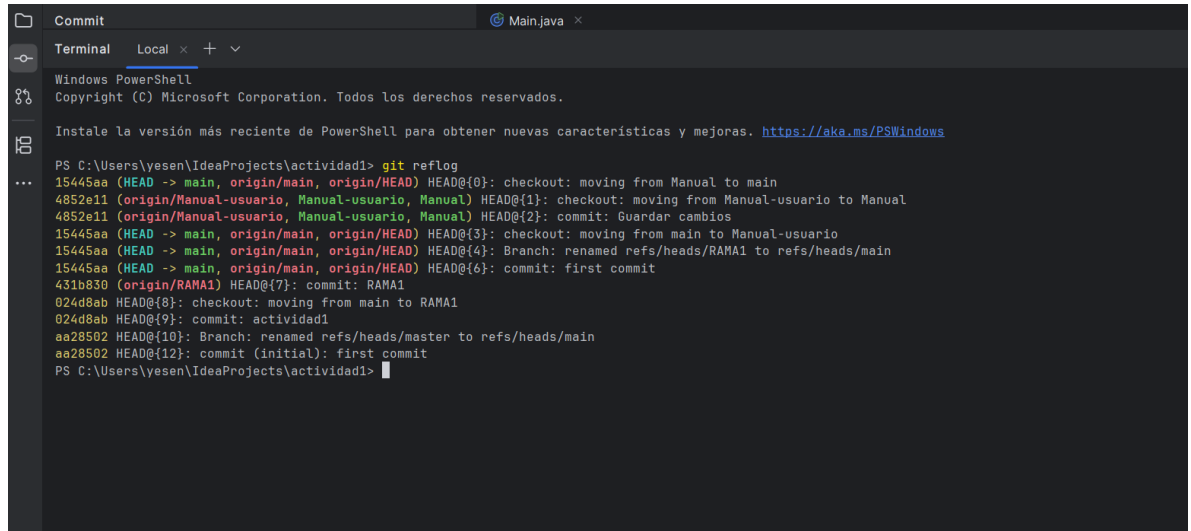
    ....
```

- **Git reflog**

Insertamos comando

Muestra el historial de movimientos de la referencia HEAD.

Es Útil para recuperar commits eliminados o volver a un estado anterior.

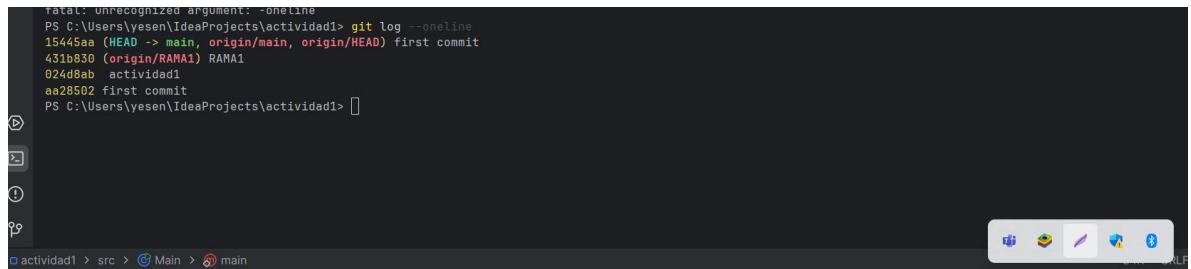


```
PS C:\Users\yesen\IdeaProjects\actividad1> git reflog
15445aa (HEAD -> main, origin/main, origin/HEAD) HEAD@{0}: checkout: moving from Manual to main
4852e11 (origin/Manual-usuario, Manual-usuario, Manual) HEAD@{1}: checkout: moving from Manual-usuario to Manual
4852e11 (origin/Manual-usuario, Manual-usuario, Manual) HEAD@{2}: commit: Guardar cambios
15445aa (HEAD -> main, origin/main, origin/HEAD) HEAD@{3}: checkout: moving from main to Manual-usuario
15445aa (HEAD -> main, origin/main, origin/HEAD) HEAD@{4}: Branch: renamed refs/heads/RAMA1 to refs/heads/main
15445aa (HEAD -> main, origin/main, origin/HEAD) HEAD@{5}: commit: first commit
431b830 (origin/RAMA1) HEAD@{6}: commit: RAMA1
024d8ab HEAD@{8}: checkout: moving from main to RAMA1
024d8ab HEAD@{9}: commit: actividad1
aa28502 HEAD@{10}: Branch: renamed refs/heads/master to refs/heads/main
aa28502 HEAD@{12}: commit (initial): first commit
PS C:\Users\yesen\IdeaProjects\actividad1>
```

- **Git log --oneline**

Agregamos el comando

Muestra el historial de commits en una sola línea por commit Es

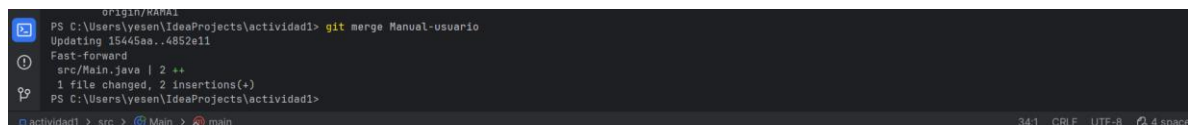


```
PS C:\Users\yesen\IdeaProjects\actividad1> git log --oneline
15445aa (HEAD -> main, origin/main, origin/HEAD) first commit
431b830 (origin/RAMA1) RAMA1
024d8ab actividad1
aa28502 first commit
PS C:\Users\yesen\IdeaProjects\actividad1>
```

- **Git merge**

- **Agregamos el comando**

Fusiona los cambios de la rama indicada en la rama actual.

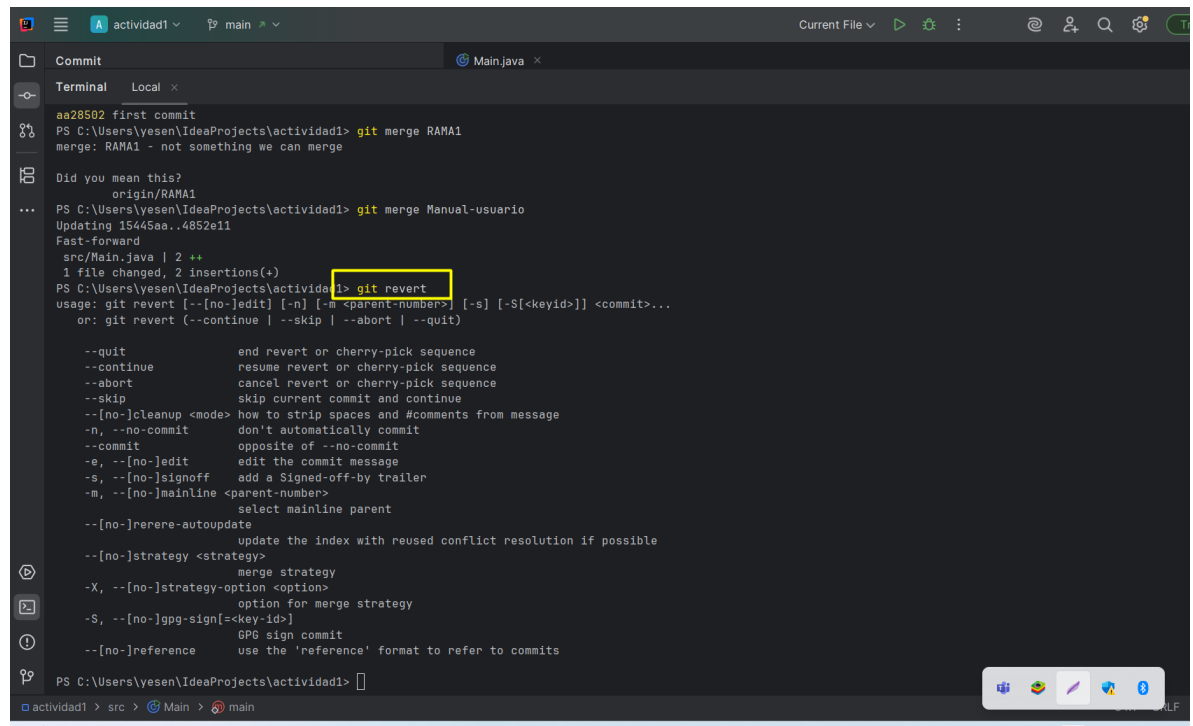


```
PS C:\Users\yesen\IdeaProjects\actividad1> git merge Manual-usuario
Updating 15445aa..4852e11
Fast-forward
 src/Main.java | 2 ++
 1 file changed, 2 insertions(+)
PS C:\Users\yesen\IdeaProjects\actividad1>
```

- **Git revert**

Agregamos el comando

Crea un nuevo commit que **revierte** los cambios de un commit anterior, sin borrar el historial.



```
aa28502 first commit
PS C:\Users\yesen\IdeaProjects\actividad1> git merge RAMA1
merge: RAMA1 - not something we can merge

Did you mean this?
  origin/RAMA1
...
PS C:\Users\yesen\IdeaProjects\actividad1> git merge Manual-usuario
Updating 15445aa..4852e11
Fast-forward
 src/Main.java | 2 ++
 1 file changed, 2 insertions(+)
PS C:\Users\yesen\IdeaProjects\actividad1> git revert
usage: git revert [--no-edit] [-n] [-m <parent-number>] [--] [-S<keyid>] <commit>...
or: git revert (--continue | --skip | --abort | --quit)

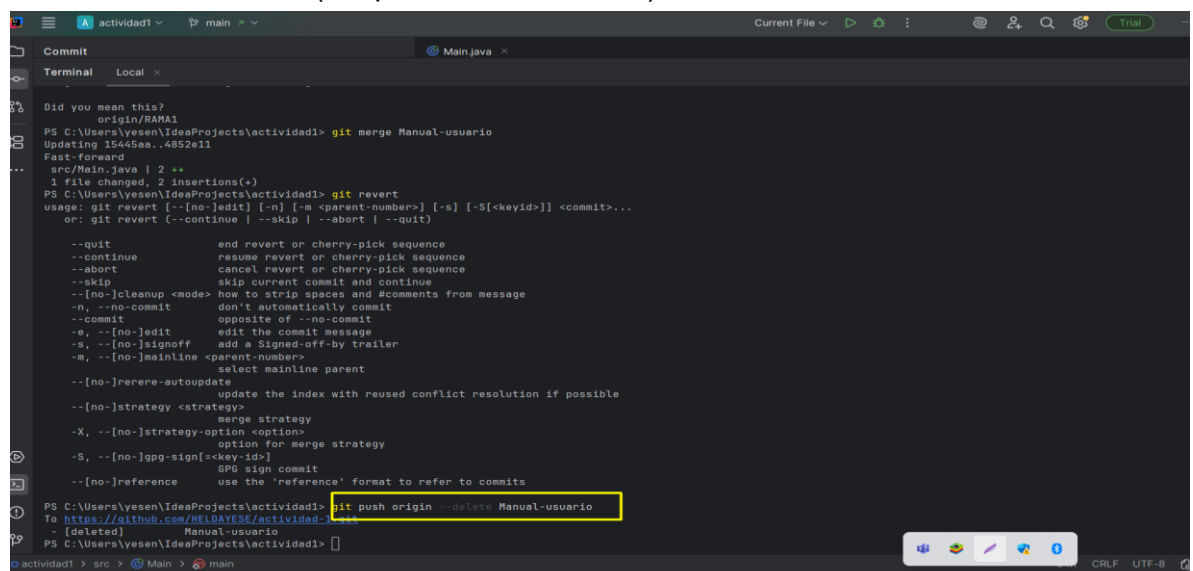
--quit          end revert or cherry-pick sequence
--continue      resume revert or cherry-pick sequence
--abort         cancel revert or cherry-pick sequence
--skip          skip current commit and continue
--[no-]cleanup <mode> how to strip spaces and #comments from message
-n, --no-commit don't automatically commit
--commit        opposite of --no-commit
-e, --[no-]edit  edit the commit message
-s, --[no-]signoff add a Signed-off-by trailer
-m, --[no-]mainline <parent-number>
                  select mainline parent
--[no-]rerere-autoupdate
                  update the index with reused conflict resolution if possible
--[no-]strategy <strategy>
                  merge strategy
-X, --[no-]strategy-option <option>
                  option for merge strategy
-S, --[no-]pgp-sign[=<key-id>]
                  GPG sign commit
--[no-]reference
                  use the 'reference' format to refer to commits

PS C:\Users\yesen\IdeaProjects\actividad1>
```

- Git push origin - - delete ramagit

Agregamos comando

Eliminar rama remota (después de fusionar PR)



```
Did you mean this?
  origin/RAMA1
...
PS C:\Users\yesen\IdeaProjects\actividad1> git merge Manual-usuario
Updating 15445aa..4852e11
Fast-forward
 src/Main.java | 2 ++
 1 file changed, 2 insertions(+)
PS C:\Users\yesen\IdeaProjects\actividad1> git revert
usage: git revert [--no-edit] [-n] [-m <parent-number>] [--] [-S<keyid>] <commit>...
or: git revert (--continue | --skip | --abort | --quit)

--quit          end revert or cherry-pick sequence
--continue      resume revert or cherry-pick sequence
--abort         cancel revert or cherry-pick sequence
--skip          skip current commit and continue
--[no-]cleanup <mode> how to strip spaces and #comments from message
-n, --no-commit don't automatically commit
--commit        opposite of --no-commit
-e, --[no-]edit  edit the commit message
-s, --[no-]signoff add a Signed-off-by trailer
-m, --[no-]mainline <parent-number>
                  select mainline parent
--[no-]rerere-autoupdate
                  update the index with reused conflict resolution if possible
--[no-]strategy <strategy>
                  merge strategy
-X, --[no-]strategy-option <option>
                  option for merge strategy
-S, --[no-]pgp-sign[=<key-id>]
                  GPG sign commit
--[no-]reference
                  use the 'reference' format to refer to commits

PS C:\Users\yesen\IdeaProjects\actividad1> git push origin --delete Manual-usuario
To https://github.com/HELDAVESE/actividad1:
- [deleted]          Manual-usuario
PS C:\Users\yesen\IdeaProjects\actividad1>
```

En la parte inferior nos muestra como la rama ha sido eliminada.

4. Conclusiones

Se aplicaron los conocimientos adquiridos dentro del aula de clase en base a los comandos básicos usados en GIT HUB.

Se identifico pequeños errores en los comandos que fueron corregidos para poder ser ejecutados de manera adecuada.

Se hizo reconocimiento de los comandos y la forma en la que se debe ser aplicada, como la eliminación y creación de email y user name dentro del repositorio.

Se aplicaron de manera adecuada los comandos requeridos por el docente.

Se tuvo en cuenta a detalle la forma en la que se debe usar cada uno de estos comando y trate de ser lo mas explicita posible con cada uno y la forma en la que se debe ingresar para poder ejecutar dentro y fuera de la terminal en este caso en el bash.

5. Referencias bibliográficas

<https://git-scm.com/book/es/v2>

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

<https://chatgpt.com>

<https://github.com/HELDAYESE/Manual-usuario>

<https://www.youtube.com/watch?v=bZWeidxYAw4>

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