Comandos Git

Ingeniería de sistemas y computación

Integrantes

Manuel Eduardo Alarcon Aza

Profesor

William Javier Matallana Porras

Universidad de Cundinamarca – UDEC

Chía

2025

**Tabla de contenido**

[**Introducción** 3](#_Toc190953070)

[**Objetivo** 3](#_Toc190953071)

[**Configuración nombre de usuario y correo electrónico** 3](#_Toc190953072)

[**Creación repositorio** 4](#_Toc190953073)

[**Git Init** 4](#_Toc190953074)

[**Git Clone** 5](#_Toc190953075)

[**Git Status** 5](#_Toc190953076)

[**Git add** 5](#_Toc190953077)

[**Git Commit** 6](#_Toc190953078)

[**Git Push** 7](#_Toc190953079)

[**Git Switch** 7](#_Toc190953080)

[**Git Branch** 8](#_Toc190953081)

[**Git Pull** 9](#_Toc190953082)

[**Como deshacer o reversar un git commit** 9](#_Toc190953083)

[**Conclusión** 9](#_Toc190953084)

[**Referencias** 9](#_Toc190953085)

## **Introducción**

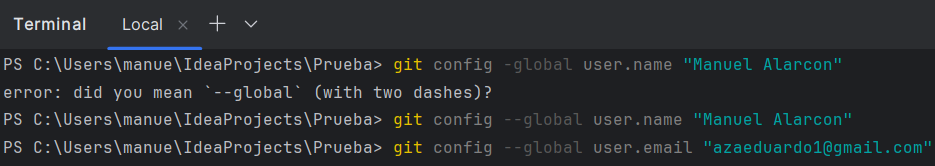
Este manual pretende que identifiquemos y visualizaremos algunos de los comandos más utilizados de Git, facilitando como podemos llegar a utilizarlos en nuestros entornos de trabajo y como estos aportan a la hora de trabajar con repositorios locales y remotos.

## **Objetivo**

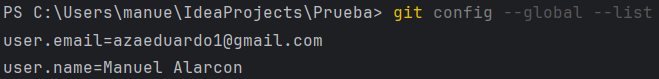
El objetivo de este manual es identificar los principales comandos de git, cómo se utilizan y cómo podemos llegar a aplicarlos en un entorno de trabajo cotidiano.

## **Configuración nombre de usuario y correo electrónico**

Para configurar el nombre de usuario debemos escribir en el terminal el comando git config - -global user.name “Nombre” y para el correo escribimos el comando git config - - global user.email “correoejemplo@gmail.com”

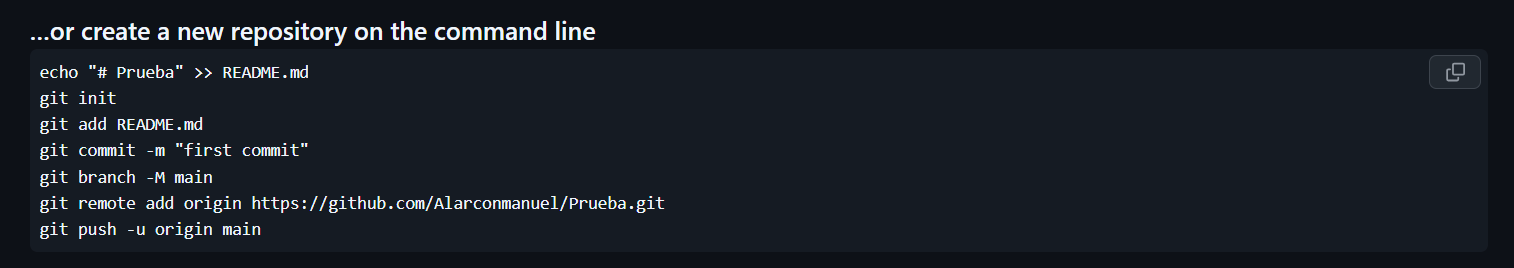


Y comprobamos que haya quedado registrado escribiendo en el terminal git config --global --list y nos deberá aparecer el nombre y correo que registramos.



## **Creación repositorio**

Para la creación de un repositorio luego de estar creado el proyecto en el local nos dirigimos a Github y creamos un nuevo repositorio dejándolo público y posteriormente copiamos esta información en el terminal





Hecho esta ya quedarían enlazados nuestro repositorio local y remoto en GitHub.

## **Git Init**

La función de este comando es el crear un nuevo repositorio de Git, inicializar un nuevo repositorio vacío o para poner un proyecto bajo un control de revisiones



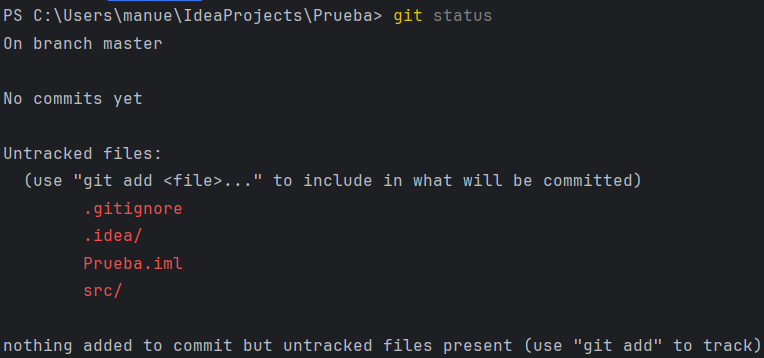
## **Git Clone**

Este comando se utiliza para crear una copia o clonar un repositorio remoto y para utilizarlo creamos una carpeta damos clic derecho seleccionamos Open git bash here y escribimos git clone y por último pegamos la url del repositorio.



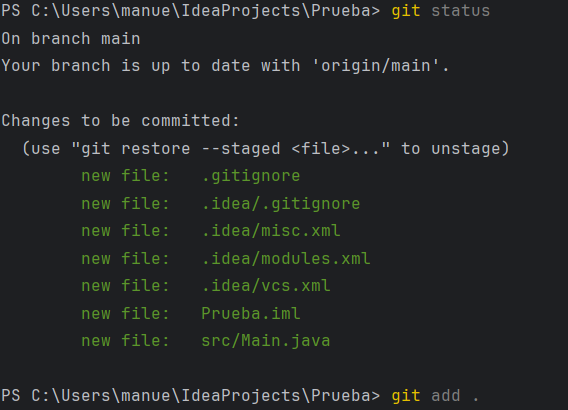
## **Git Status**

Este comando sirve para ver el estado de los archivos, esto nos ayuda a saber que archivos tenemos, si hemos modificado alguno y si están siendo rastreados o no, en el terminal se escribe git status y nos da esta información.



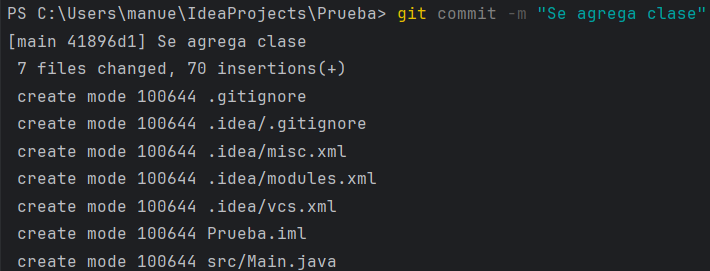
## **Git add**

Este comando sirve para agregar todo lo que llevemos trabajado a el área de ensayo además es importante ya que si no lo utilizamos no podremos ejecutar git commit posteriormente.



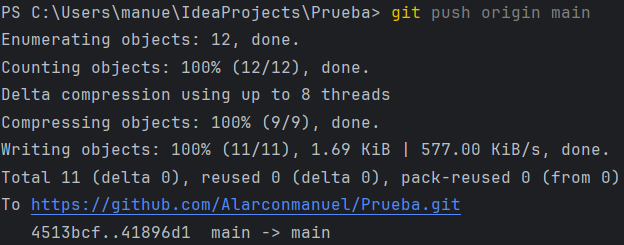
## **Git Commit**

Este comando nos permite guardar todos los cambios hechos en un repositorio local, en el terminal se escribe git commit -m “mensaje”, el mensaje que va entre comillas debe ser claro y especifico.



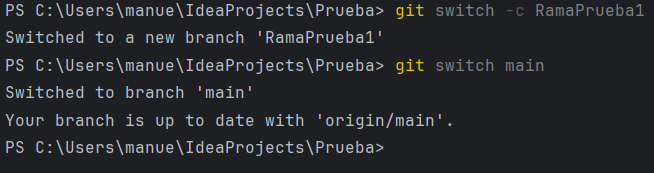
## **Git Push**

Este comando se utiliza para enviar los cambios realizados en una rama local de un repositorio a un repositorio remoto, se utiliza escribiendo en el terminal git push origin “nombre de la rama” en este caso main, y para verificarlo revisamos en Github que hayan sido cargados los cambios.



## **Git Switch**

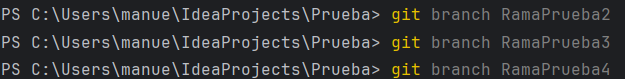
Este comando se utiliza para crear ramas e ir cambiando entre estas, para crear y pasarnos de una vez a la rama nueva utilizamos git switch -c “nombre de la nueva rama” y para solo cambiar de rama es git switch “nombre de la rama”.

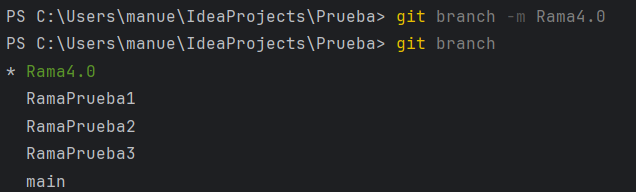
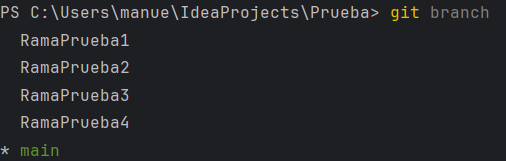
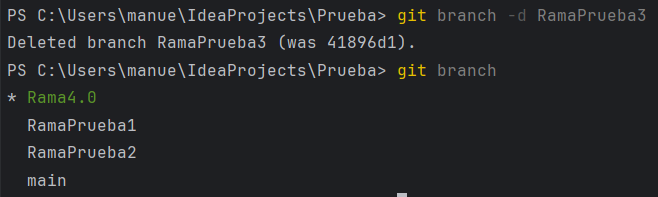


## **Git Branch**

Este comando nos permite crear, modificar, enumerar o eliminar ramas, los comandos para usarlos en el terminal son los siguientes:

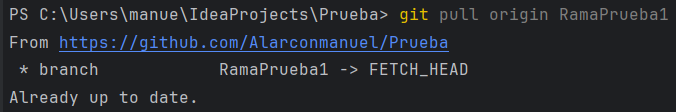
* + git Branch – visualizar las ramas existentes
  + git branch NuevaRama – crea una nueva rama
  + git branch -d <branch> - elimina el branch indicado, evitando la eliminación si están presentes commit no fusionados
  + git branch -D <branch> - elimina el branch indicado sin comprobar la presencia de commit no fusionados
  + git branch -m <branch> - cambia el nombre del branch actual
  + git branch -a - enumera branch remotos



## **Git Pull**

Nos sirve para descargar los cambios que se hallan realizado en el repositorio remoto y mantener el local actualizado



## **Como deshacer o reversar un git commit**

Para deshacer un git commit hay dos formas una si no se ha hecho git push y una cuando ya se haya hecho. Para cuando no se ha realizado git push podemos utilizar git reset, si queremos mantener los cambios escribimos en el terminal git reset --soft HEAD~1; HEAD~1 el programa lo interpreta como que se desea volver a la versión anterior y el parámetro soft lo que no elimina los cambios si no que los mantiene localmente y si no queremos mantener los cambios remplazando soft por hard, quedando git reset --hard HEAD~1.

Cuando ya hemos hecho git push debemos utilizar git revert 74a1092, esto crea un nuevo commit que deshace los cambios hechos por el anterior.

## **Conclusión**

El uso de comandos de git nos ayuda en el trabajo colaborativo de desarrollo de software, haciendo uso de los comandos principales y el trabajo con ramas además del manejo de repositorios locales y remotos.

## **Referencias**

* [Configuración de un repositorio de Git](https://www.atlassian.com/es/git/tutorials/setting-up-a-repository)
* [Guardar cambios en Git](https://www.atlassian.com/es/git/tutorials/saving-changes)
* [Comando Git Branch](https://aulab.es/articulos-guias-avanzadas/93/el-comando-git-branch-en-git)
* [Deshacer cambios en Git](https://www.atlassian.com/es/git/tutorials/undoing-changes)
* [Cómo deshacer el último commit con Git](https://midu.dev/como-deshacer-el-ultimo-commit-git/)