

Programación 2

Universidad de Cundinamarca

Fabián Guillermo Meléndez Gaitán

Comandos Git

Sede Chía

2025

Guía comandos GIT

1. git config --global user.name

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

Prueba la nueva tecnología PowerShell multiplataforma https://aka.ms/pscore6

PS C:\Users\User\IdeaProjects\Actividad> git config --global user.name "Fabian Melendez"
PS C:\Users\User\IdeaProjects\Actividad>
```

Este comando se utiliza para configurar el nombre de usuario que se asociará con los commits en Git.

2. git config --global user.email

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

Prueba la nueva tecnología PowerShell multiplataforma https://aka.ms/pscore6

PS C:\Users\User\IdeaProjects\Actividad> git config --global user.name "Fabian Melendez"
PS C:\Users\User\IdeaProjects\Actividad> git config --global user.email "fabianmelendez321@gmail.com"
PS C:\Users\User\IdeaProjects\Actividad>
```

Se utiliza para configurar la dirección de correo electrónico con el que se asociará los commits en Git.

3. git init

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

Prueba la nueva tecnología PowerShell multiplataforma https://aka.ms/pscore6

PS C:\Users\User\IdeaProjects\Actividad> git init
```

Se utiliza para crear un nuevo repositorio de Git, generando una carpeta oculta llamada ".git" donde se almacena toda la información del repositorio.

4. `git commit -m "nombre del proyecto"`

```
PS C:\Users\User\IdeaProjects\Actividad> git commit -m "nombre del proyecto"
On branch main
Your branch is up to date with 'origin/main'.

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        .gitignore
        .idea/
        Actividad.iml
        src/

nothing added to commit but untracked files present (use "git add" to track)
PS C:\Users\User\IdeaProjects\Actividad>
```

Este comando tiene como función guardar los cambios en el historial del repositorio, donde se deja un mensaje, generalmente describiendo los cambios realizados.

5. `git push origin main`

```
Your branch is up to date with 'origin/main'.

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        .gitignore
        .idea/
        Actividad.iml
        src/

nothing added to commit but untracked files present (use "git add" to track)
PS C:\Users\User\IdeaProjects\Actividad> git push origin main
Everything up-to-date
PS C:\Users\User\IdeaProjects\Actividad>
```

Se utiliza para enviar los commits locales al repositorio remoto en la rama que se escriba, en este caso es la rama "main".

6. git status

```
PS C:\Users\User\IdeaProjects\Actividad> git status
On branch main
Your branch is up to date with 'origin/main'.

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        .gitignore
        .idea/
        Actividad.iml
        src/

nothing added to commit but untracked files present (use "git add" to track)
PS C:\Users\User\IdeaProjects\Actividad> █
```

Se utiliza para ver el estado del repositorio y los archivos, muestra los cambios modificados e informa si hay cambios preparados para commit.

7. git add .

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

Prueba la nueva tecnología PowerShell multiplataforma https://aka.ms/powershell

PS C:\Users\User\IdeaProjects\Actividad> git add .
PS C:\Users\User\IdeaProjects\Actividad> █
```

Se utiliza para agregar todos los archivos modificados al +área de preparación, estando ya listos para un commit.

8. git branch

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

Prueba la nueva tecnología PowerShell multiplataforma https://aka.ms/pscore6

PS C:\Users\User\IdeaProjects\Actividad> git branch
* main
PS C:\Users\User\IdeaProjects\Actividad>
```

Este comando se usa para ver y administrar las ramas del repositorio Git, resaltando en verde la rama en la que estamos trabajando en el momento.

9. git switch -c "nombre"

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

Prueba la nueva tecnología PowerShell multiplataforma https://aka.ms/pscore6

PS C:\Users\User\IdeaProjects\Actividad> git switch -c "desarrollo"
Switched to a new branch 'desarrollo'
PS C:\Users\User\IdeaProjects\Actividad>
```

Este comando se utiliza para agregar una nueva rama al repositorio, en la cual se podrá trabajar, se le puede asignar el nombre que se desee.

10. git switch "rama"

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

Prueba la nueva tecnología PowerShell multiplataforma https://aka.ms/pscore6

PS C:\Users\User\IdeaProjects\Actividad> git switch desarrollo
Already on 'desarrollo'
PS C:\Users\User\IdeaProjects\Actividad>
```

Se utiliza para cambiar a la rama del repositorio que escribamos, podemos comprobarlo usando el comando "git branch".

11. git branch -D "rama"

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

Prueba la nueva tecnología PowerShell multiplataforma https://aka.ms/pscore6

PS C:\Users\User\IdeaProjects\Actividad> git branch -D prueba2
Deleted branch prueba2 (was fa6c732).
PS C:\Users\User\IdeaProjects\Actividad>
```

Este comando se utiliza para borrar una rama creada del repositorio de manera forzada, aun si tiene cambios no cargados o fucionados.

12. git pull origin "rama"

```
PS C:\Users\User\IdeaProjects\Actividad> git pull origin main
From https://github.com/Fabian-Melendez/Actividad
 * branch                main                -> FETCH_HEAD
Already up to date.
PS C:\Users\User\IdeaProjects\Actividad>
```

Se utiliza para descargar el código que se encuentre en la rama de la que deseemos, en este caso utilizamos la rama "main".

13. git clone "URL"

```
MINGW64:/c/Users/User/Desktop/carpeta de prueba

User@DESKTOP-D4K75HQ MINGW64 ~/Desktop/carpeta de prueba
$ git clone "URL"
```

Este comando se utiliza para clonar un repositorio o código teniendo la URL del mismo, se hace abriendo o creando una carpeta deseada, y seleccionando la opción "Open Git Bash here".

14. git config --list

```
PS C:\Users\ESTUDIANTE\IdeaProjects\practica> 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
credential.https://dev.azure.com.usehttppath=true
```

Este comando nos muestra todas las configuraciones de git que están activas en el sistema, provienen de tres niveles, “—system”, “—global” y “—local”.

15. git config --global --unset user.name

```
Prueba la nueva tecnología PowerShell multiplataforma https://aka.ms/pscore6

PS C:\Users\ESTUDIANTE\IdeaProjects\practica> git config --global --unset user.name
PS C:\Users\ESTUDIANTE\IdeaProjects\practica>
```

Este comando se utiliza para desasociar el nombre de usuario propuesto antes con el comando git config -- global user.name

16. git config --global --unset user.email

```
Other
  --[no-]default <value>
                        with --get, use default value when missing entry
  --[no-]comment <value>
                        human-readable comment string (# will be prepended as needed)
  --[no-]fixed-value    use string equality when comparing values to value pattern
  --[no-]includes        respect include directives on lookup

PS C:\Users\ESTUDIANTE\IdeaProjects\practica> git config --global --unset user.email
PS C:\Users\ESTUDIANTE\IdeaProjects\practica>
```

Este comando se usa para desasociar el email de usuario registrado anteriormente con el comando “git config --global user.email”, podemos confirmar estos cambios con el comando “git config --list”

17. git branch -r

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

Prueba la nueva tecnología PowerShell multiplataforma https://aka.ms/pscore6

PS C:\Users\ESTUDIANTE\IdeaProjects\practica> git branch -r
origin/main
PS C:\Users\ESTUDIANTE\IdeaProjects\practica>
```

Se utiliza para mostrar una lista de las ramas remotas que hay en un repositorio, sirve para corroborar cambios.

18. git fetch --all

```
PS C:\Users\ESTUDIANTE\IdeaProjects\practica> git fetch --all
PS C:\Users\ESTUDIANTE\IdeaProjects\practica>
```

Se utiliza para actualizar la información de todas las ramas remotas, sin modificar el código pero no fusiona los cambios automáticamente en la rama actual.

19. git log

```
PS C:\Users\ESTUDIANTE\IdeaProjects\practica> git log
commit 2a5f9fc565686c7e84ae474c914e2e98f6e56e4b (HEAD -> main, origin/main, origin/HEAD)
Author: Fabian Melendez <fabianmelendez321@gmail.com>
Date: Thu Feb 20 16:50:49 2025 -0500

    first commit
PS C:\Users\ESTUDIANTE\IdeaProjects\practica>
```

La función de este comando es mostrarnos el historial de commits en un repositorio, en casi de necesitar modificar algo.

20. git reflog

```
PS C:\Users\ESTUDIANTE\IdeaProjects\practica> git reflog
6edcf6a (HEAD -> main, origin/main, origin/HEAD) HEAD@{0}: commit: primeros cambios
2a5f9fc HEAD@{1}: Branch: renamed refs/heads/main to refs/heads/main
2a5f9fc HEAD@{3}: commit (initial): first commit
PS C:\Users\ESTUDIANTE\IdeaProjects\practica>
```

Este comando muestra el historial de referencias de git, o sea los cambios en las ramas y los movimientos en commits, merges o resets.

21. git log --oneline

```
PS C:\Users\ESTUDIANTE\IdeaProjects\practica> git log --oneline
6edcf6a (HEAD -> main, origin/main, origin/HEAD) primeros cambios
2a5f9fc first commit
PS C:\Users\ESTUDIANTE\IdeaProjects\practica>
```

Nos muestra el historial de commits en formato compacto, donde cada commit representa una línea, también se usa para ver los ID de los commits en caso de necesitar hacer un revert

22. git revert

```
revert primeros cambios  
  
This reverts commit 6edcf6ac048c4becf5a3c6618dd0c2a9709f7c09.  
  
Please enter the commit message for your changes. Lines starting  
with '#' will be ignored, and an empty message aborts the commit  
.  
.  
On branch main  
Your branch is up to date with 'origin/main'.  
.  
Changes to be committed:  
  deleted:   .gitignore  
  deleted:   .idea/.gitignore  
  deleted:   .idea/misc.xml  
git/COMMIT_EDITMSG [unix] (17:44 20/02/2025)
```

Este comando se usa para deshacer un commit, en caso de haber enviado algún archivo de manera errónea, creando un nuevo commit especificado, no borra el historial.

23. git merge

```
PS C:\Users\User\IdeaProjects\Actividad> git merge main  
Auto-merging src/Main.java  
CONFLICT (content): Merge conflict in src/Main.java  
Automatic merge failed; fix conflicts and then commit the result.  
PS C:\Users\User\IdeaProjects\Actividad> 
```

Este comando se usa para combinar dos ramas en git, su propósito es integrar los cambios de una rama en otra, primero activamos la rama en la que queremos implementar los cambios y luego después de “merge” integramos los cambios de la rama que deseamos.

24. git rebase

```
PS C:\Users\User\IdeaProjects\Actividad> git switch main  
Switched to branch 'main'  
Your branch is up to date with 'origin/main'.  
PS C:\Users\User\IdeaProjects\Actividad> git branch  
  desarrollo  
* main  
PS C:\Users\User\IdeaProjects\Actividad> git rebase desarrollo  
Successfully rebased and updated refs/heads/main.  
PS C:\Users\User\IdeaProjects\Actividad> 
```

Se usa para mover una rama a otro punto del historial del proyecto, reescribiendo los commits para mantener el historial más limpio, y no utilizar merges innecesariamente.