



**UDEC**  
UNIVERSIDAD DE  
CUNDINAMARCA

Universidad De Cundinamarca

Ingeniería De Sistemas

Programacion II

Diego Alejandro Garcia Olaya

Grupo

C-109

Comandos Git

## Tabla de contenido

Tabla de contenido .....	2
Comandos Git Introducción.....	4
Git.....	4
GitHub.....	4
Editor de código .....	5
Creación de un repositorio desde Git paso a paso .....	5
Git Status.....	5
Git Init.....	5
Git add .....	6
Git commit -m “” .....	6
Git push origib main .....	6
Clonación del proyecto.....	7
git clone url .....	7
Ramificaciones.....	8
Git switch -c rama .....	8
git switch rama .....	8
git push origin rama .....	9
git pull origen main .....	9
git branch -D rama.....	10

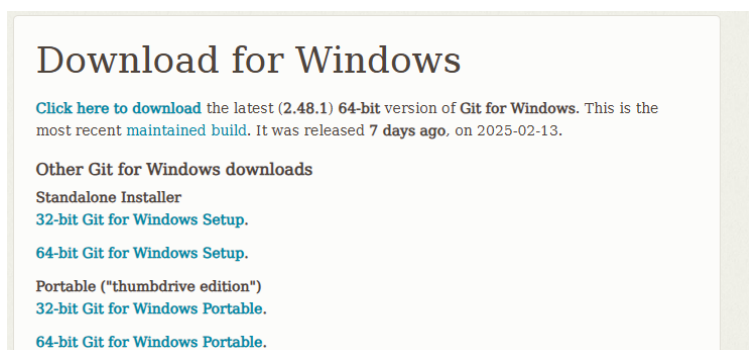
git fetch -all .....	10
git branch -r .....	10
git merge rama.....	11
git push origin --delete rama .....	11
Git config.....	12
git config --global user.email .....	12
git config --global user.name .....	12
git config --list .....	12
git config --global --unset user.name .....	13
git config --global --unset user.gmail .....	13
Historial .....	14
git log .....	14
git log --oneline.....	14
git reflog.....	14
Git reverse.....	15

## Comandos Git Introducción

Para poder subir un proyecto hay que tener en cuenta unas bases necesarias para poder hacer el proceso de manera adecuada, para poder alojar el código para que otros miembros de la comunidad puedan verlo e incluso colaborar para mejorarlo.

## Git

Podemos empezar con la instalación de Git, es una herramienta de código abierto, nos permite realizar un seguimiento al código fuente. Comenzamos con la instalación de este con la instalación pertinente para nuestro equipo, teniendo en cuenta nuestro sistema operativo y bits que dispone nuestro equipo.



<https://git-scm.com/downloads/win>

## GitHub

con una cuenta en esta plataforma nos ofrece un servicio basado en la nube que aloja repositorios de Git. Donde nos permite crear una cuenta y alojar nuestro código.

Utiliza el menú desplegable de la esquina superior derecha y selecciona Nuevo Repositorio.

Escribe un nombre para el repositorio. También puedes añadir una descripción muy precisa si así lo deseas

Elige el sea más conveniente: público, interno o privado. Haz clic en Crear repositorio.

## Editor de código

En nuestro caso estamos utilizando IntelliJ IDEA editor de código para el cual lo utilizamos para programar en la versión de Java JDK21. Abrimos el programa y vamos a crear una carpeta donde se va a alojar nuestro proyecto, abrimos la terminal y colocamos los siguientes comandos en el orden en el que aparecen a continuación.

## Creación de un repositorio desde Git paso a paso

### Git Status

te proporcionará información útil según tu situación actual

```
PS C:\ProgramJava\comados_GitHub> 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/
  comados_GitHub.iml
  src/
```

### Git Init

para iniciar git en dicha carpeta

```
PS C:\ProgramJava\comados_GitHub> git init
Reinitialized existing Git repository in C:/ProgramJava/comados_GitHub/.git/
```

## Git add .

para agregar y añadir todos los ficheros

```
PS C:\ProgramJava\comados_GitHub> git add .  
warning: in the working copy of '.gitignore', LF will be replaced by CRLF the next time Git touches it  
warning: in the working copy of 'src/Main.java', LF will be replaced by CRLF the next time Git touches it
```

## Git commit -m ""

“Creado el proyecto inicial” (prepara los ficheros que quieres subir con breve explicación acerca de los cambios)

```
PS C:\ProgramJava\comados_GitHub> git commit -m "Comandos"  
[main 4c1236b] Comandos  
7 files changed, 71 insertions(+)  
create mode 100644 .gitignore  
create mode 100644 .idea/.gitignore  
create mode 100644 .idea/misc.xml  
create mode 100644 .idea/modules.xml  
create mode 100644 .idea/vcs.xml  
create mode 100644 comados_GitHub.iml  
create mode 100644 src/Main.java
```

## Git push origin main

sube los cambios con el comando final

```
PS C:\ProgramJava\comados_GitHub> git push origin main
Enumerating objects: 12, done.
Counting objects: 100% (12/12), done.
Delta compression using up to 4 threads
Compressing objects: 100% (9/9), done.
Writing objects: 100% (11/11), 1.72 KiB | 1.72 MiB/s, done.
Total 11 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/Dagarcia0/Comandos.git
    ea1be94..4c1236b  main -> main
PS C:\ProgramJava\comados_GitHub>
PS C:\ProgramJava\comados_GitHub>
```

Si todos los pasos se han hecho de manera correcta como se mostró en las imágenes, al entrar al link podemos observar cómo se subió el repositorio de manera correcta si problema alguno.

## Clonación del proyecto

Copiamos la url del proyecto ya creado y ahora vamos a nuestra carpeta designada para archivar nuestros códigos, vamos a las opciones de ver mas y entramos a esa opción que se muestra a continuación



## git clone url

utilizamos ese comando para clonar

```

Lenovo@DESKTOP-3A43619 MINGW64 ~/OneDrive/Escritorio/JAVA (master)
$ git clone https://github.com/Dagarcia0/Comandos.git
Cloning into 'Comandos'...
remote: Enumerating objects: 14, done.
remote: Counting objects: 100% (14/14), done.
remote: Compressing objects: 100% (10/10), done.
remote: Total 14 (delta 0), reused 14 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (14/14), done.

```

Ya clonado el código, vamos al IntelliJ donde copiamos los códigos que nos da GitHub Y comenzamos con los comandos para crear una nueva rama para no trabajar sobre la rama MAIN con los comandos.

## Ramificaciones

### Git switch -c rama

creamos la rama “rama\_practica” y se cambia a esta nueva rama creada.

```

PS D:\Users\Lenovo\programacion II\clonados github\practica> git switch -c practica
Switched to a new branch 'practica'

```

### git Branch

para confirmar que si estamos trabajando sobre la rama que creamos

```

PS D:\Users\Lenovo\programacion II\clonados github\practica> git branch
main
* practica

```

### git switch rama

cambia a una rama existente.

```

PS D:\Users\Lenovo\programacion II\clonados github\practica> git switch main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.

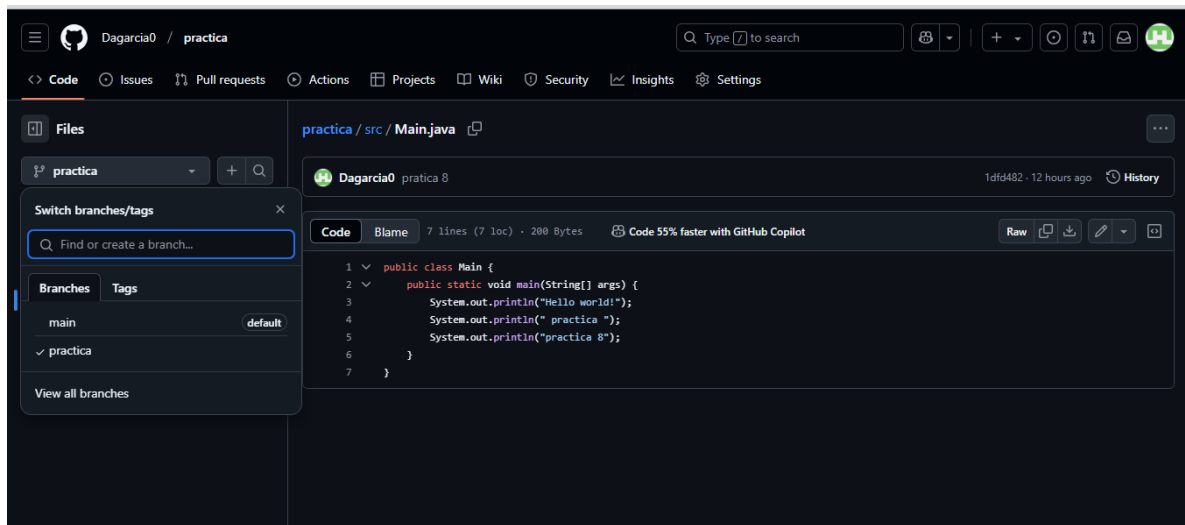
```



## git push origin rama

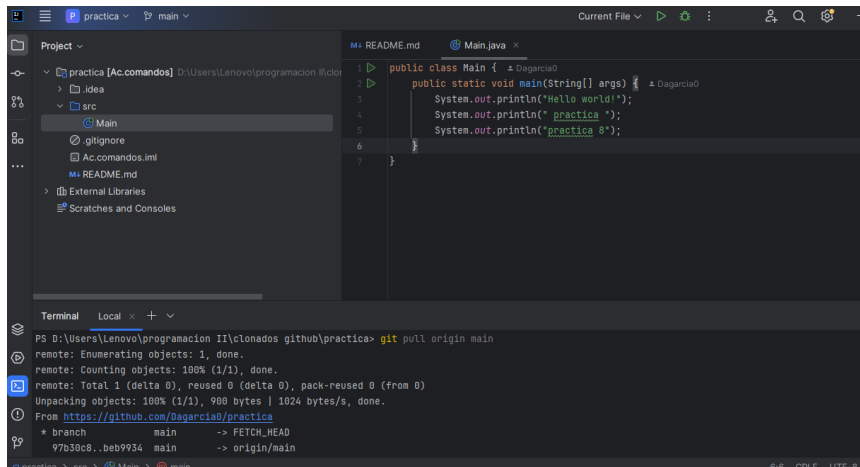
Sube cambios de la rama al remoto

```
PS D:\Users\Lenovo\programacion II\clonados github\practica> git push origin practica
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
remote:
remote: Create a pull request for 'practica' on GitHub by visiting:
remote:   https://github.com/Dagarcia0/practica/pull/new/practica
remote:
To https://github.com/Dagarcia0/practica.git
* [new branch]   practica -> practica
```



## git pull origin main

Descarga y fusiona cambios de la rama remota, actualizando nuestra rama MAIN con los cambios mas recientes que se hicieron remotamente, pocas palabras se une las ramas que tenias con lo cambio a la rama original MAIN finalizando con esta etapa de manera exitosa.



## git branch -D rama

este comando borra la rama local que habíamos creado de manera “forzada”. En caso de hacer este mismo comando pero con la letra “d” minúscula se elimina la rama local solo si ya ha sido fusionada con otra rama si de lo contrario tiene cambios sin fusionar, git mostrara un error.

```
PS D:\Users\Lenovo\programacion II\clonados github\practica> git branch -D practica
error: cannot delete branch 'practica' used by worktree at 'D:/Users/Lenovo/programacion II/clonados github/practica'
```

## git fetch --all

Descarga todas las ramas remotas sin fusionar y utilizamos “git Branch” para rectificar que si se allá descargado la rama remota.

```
PS D:\Users\Lenovo\programacion II\clonados github\practica> git fetch --all
PS D:\Users\Lenovo\programacion II\clonados github\practica> git branch
main
* practica
```

## git branch -r

Muestra las ramas remotas disponibles

```
PS D:\Users\Lenovo\programacion II\clonados github\practica> git branch -r
origin/HEAD -> origin/main
origin/main
origin/practica
origin/rama_practica
```

## git merge rama

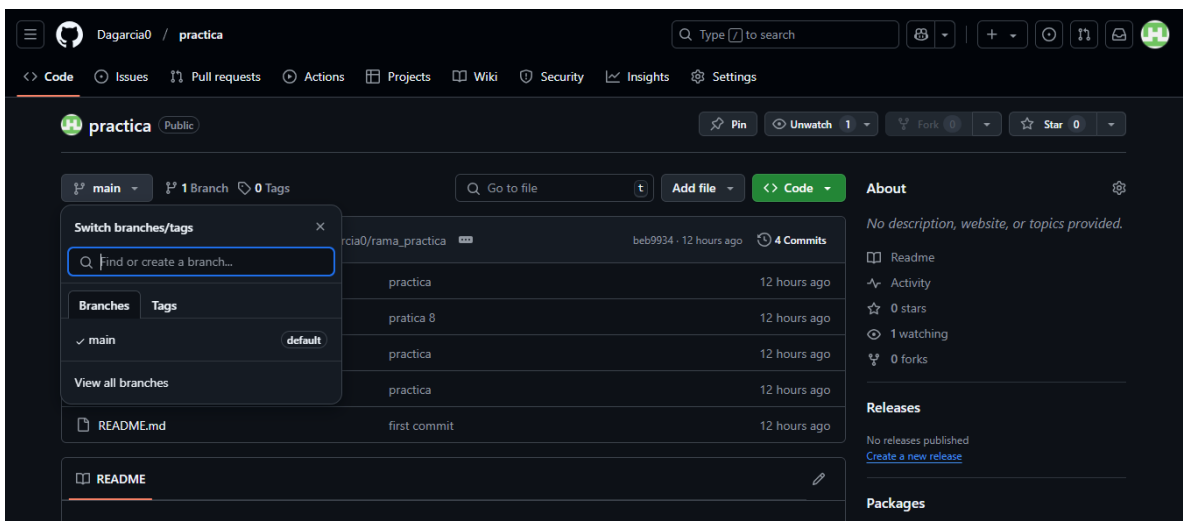
Fusiona una rama en la actual

```
PS D:\Users\Lenovo\programacion II\clonados github\practica> git merge practica
Already up to date.
```

## git push origin --delete rama

Elimina una rama en el repositorio remoto

```
PS D:\Users\Lenovo\programacion II\clonados github\practica> git push origin --delete practica
To https://github.com/Dagarcia0/practica.git
- [deleted]          practica
PS D:\Users\Lenovo\programacion II\clonados github\practica>
PS D:\Users\Lenovo\programacion II\clonados github\practica>
```



Efectivamente se borró la rama desde la consola sin necesidad de entrar a la plataforma de GitHub.

## Git config

### git config --global user.email

Para usar Git config email para configurar su correo electrónico en la terminal

### git config --global user.name

Para configurar su nombre de usuario, navegue a una terminal y ejecute

```
PS D:\Users\Lenovo\programacion II\clonados github\practica> git config --global user.email pablito@gmail.com
PS D:\Users\Lenovo\programacion II\clonados github\practica> git config --global user.name "Pablo Toto"
```

Son cadenas delimitadas que se componen por una “sección y una clave”. en función de su jerarquía, donde Gmail es una propiedad secundaria del bloque. La marca --global usa para que se defina este valor para el usuario actual del sistema operativo.

### git config --list

Nos devuelve en el mismo orden jerárquico que cubrimos anteriormente, mostrando los valores asociados a sus confirmaciones.

```
PS D:\Users\Lenovo\programacion II\clonados github\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
init.defaultbranch=master
gui.recentrepo=C:/ProgramJava/PROGRAMACION-2
user.name=Pablo Toto
user.email=pablito@gmail.com
core.repositoryformatversion=0
core.filemode=false
core.bare=false
core.logallrefupdates=true
core.symlinks=false
core.ignorecase=true
remote.origin.url=https://github.com/Dagarcia0/practica.git
```

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

Este comando elimina la configuración global del nombre de usuario en Git.

**git config --global --unset user.gmail**

este comando elimina la configuración global del Gmail de usuario en Git

```
PS D:\Users\Lenovo\programacion II\clonados github\practica> git config --unset user.name
PS D:\Users\Lenovo\programacion II\clonados github\practica> git config --global --unset user.name
PS D:\Users\Lenovo\programacion II\clonados github\practica> git config --global user.name
PS D:\Users\Lenovo\programacion II\clonados github\practica> git config --global --unset user.gmail
PS D:\Users\Lenovo\programacion II\clonados github\practica> git config --global user.gmail
```

En la imagen se muestra como se utilizo los comandos para borrar el usuario y correo que se habían creado anteriormente, comprobamos el borrado con el comando correspondiente para cada uno que seria “git config --global user.name y git config --global user.gmail”. mostrándonos el correcto borrado de estas.

Volvemos a utilizar el comando “git config --global user.email y git config --global user.name” para reintegrar nuestro usuario principal y correo.

```
user.gmail=dagarciaolaya@ucundinamarca.edu.co
user.name=Dagarcia0
```

## Historial

### git log

Muestra el historial detallado de commits

```
PS D:\Users\Lenovo\programacion II\clonados github\practica> git log
commit 7a8a103260db33986ac6339c41086cc0bc9522be (HEAD -> practica)
Author: Dagarcia0 <dagarciaolaya@ucundinamarca.edu.co>
Date:   Fri Feb 21 01:54:37 2025 -0500

    pequeña actualizacion

commit beb99340727c3a29051f99f3e3552c88cce73e8b (origin/main, origin/HEAD, main)
Merge: 97b30c8 1dfd482
Author: Dagarcia0 <dagarciaolaya@ucundinamarca.edu.co>
Date:   Thu Feb 20 13:27:13 2025 -0500

    Merge pull request #1 from Dagarcia0/rama_practica

    practica 8

commit 1dfd482d5abb833b1dd813da90f54aee85ab8019 (origin/rama_practica)
```

### git log --oneline

Muestra el historial de commits en una línea por commit

```
PS D:\Users\Lenovo\programacion II\clonados github\practica> git log --oneline
7a8a103 (HEAD -> practica) pequeña actualizacion
beb9934 (origin/main, origin/HEAD, main) Merge pull request #1 from Dagarcia0/rama_practica
1dfd482 (origin/rama_practica) practica 8
97b30c8 practica
0a949ef first commit
```

### git reflog

Muestra un historial de todas las acciones en Git

```
PS D:\Users\Lenovo\programacion II\clonados github\practica> git reflog
7a8a103 (HEAD -> practica) HEAD@{0}: commit: pequeña actualizacion
beb9934 (origin/main, origin/HEAD, main) HEAD@{1}: checkout: moving from main to practica
beb9934 (origin/main, origin/HEAD, main) HEAD@{2}: checkout: moving from practica to main
beb9934 (origin/main, origin/HEAD, main) HEAD@{3}: checkout: moving from main to practica
beb9934 (origin/main, origin/HEAD, main) HEAD@{4}: pull origin main: Fast-forward
97b30c8 HEAD@{5}: checkout: moving from rama_practica to main
1dfd482 (origin/rama_practica) HEAD@{6}: commit: practica 8
97b30c8 HEAD@{7}: checkout: moving from main to rama_practica
97b30c8 HEAD@{8}: clone: from https://github.com/Dagarcia0/practica.git
```

## Git reverse

Se creo una nueva rama para poder hacer el comando revert

```
PS D:\Users\Lenovo\programacion II\clonados github\practica> git branch
main
practica
prueba_revere
* revert_segundo
PS D:\Users\Lenovo\programacion II\clonados github\practica> git switch -m "comando reve
```

Subimos los cambios con el comando git push origin revert\_segundo

```
PS D:\Users\Lenovo\programacion II\clonados github\practica> git push origin revert_segundo
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), 368 bytes | 368.00 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.
remote:
remote: Create a pull request for 'revert_segundo' on GitHub by visiting:
remote:   https://github.com/Dagarcia0/practica/pull/new/revert\_segundo
remote:
To https://github.com/Dagarcia0/practica.git
* [new branch]      revert_segundo -> revert_segundo
```

Utilizamos nuevamente git log --oneline para mostrar en orden cronológico los commit

```
PS D:\Users\Lenovo\programacion II\clonados github\practica> git log --oneline
ccadb05 (HEAD -> revert_segundo, origin/revert_segundo) comando revert
7a8a103 (practica) pequeña actualizacion
beb9934 (origin/main, origin/HEAD, main) Merge pull request #1 from Dagarcia0/rama_practica
1dfd482 (origin/rama_practica) practica 8
97b30c8 practica
0a949ef first commit
```

## Git revert rama

