

Manuel de uso Git.

Cristian Andrés Galeano

Profesor: William Alexander Matallana Cortes

Universidad de Cundinamarca

Ingeniería de Sistemas y Computación

Chía

2025

INTRODUCCION

En este trabajo vamos a observar y aprender como se usan los diferentes comandos básicos de Git que vimos durante las clases para un correcto uso de estos mismos y así entender como se va a trabajar durante el semestre y en futuros usos en la vida laboral.

OBJETIVOS

- **Comprender los conceptos fundamentales de Git:** ¿Qué es un repositorio, un commit, una rama?
- **Configurar Git:** Establecer la información de usuario y personalizar las opciones de configuración.
- **Inicializar y clonar repositorios:** Crear nuevos repositorios y obtener copias de repositorios existentes.

Lista todas las configuraciones de Git, incluyendo las globales, locales y de sistema. Útil para verificar tu configuración actual.

```
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
user.name=C-Galeano
user.email=cristianagaleano@ucundinamarca.edu.co
user.email=cristianagaleano@ucundinamarca.edu.co
core.repositoryformatversion=0
core.filemode=false
core.bare=false
core.logallrefupdates=true
core.symlinks=false
core.ignorecase=true
remote.origin.url=https://github.com/C-Galeano/Ejerciciopracticorevert.git
```

Git config – -global user.email

Establece la dirección de correo electrónico global para todos los repositorios Git.

git config --global user.name "Nombre":

Establece tu nombre de usuario global para todas tus repositorios Git.

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

Elimina la configuración global de la dirección de correo electrónico.

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

Elimina la configuración global del nombre de usuario

GIT INIT:

Empezamos escribiendo en nuestra consola de JAVA el comando “git init”, esto para crear un nuevo repositorio, ese comando solo se usa al inicio, no hay necesidad de volverlo a escribir nuevamente.

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

Instale la versión más reciente de PowerShell para obtener nuevas característ

PS C:\Users\cgale\IdeaProjects\GIT> git init
Initialized empty Git repository in C:/Users/cgale/IdeaProjects/GIT/.git/
PS C:\Users\cgale\IdeaProjects\GIT> 
```

git clone <url_repositorio>:

Clona un repositorio Git existente desde una URL (puede ser remota o local) a una nueva carpeta

GIT STATUS:

Al introducir este comando, vemos los cambios que se han realizado en nuestro repositorio, este comando lo usamos con bastante frecuencia cuando estamos realizando ajustes a nuestros codigos, lo hacemos para verificar que los cambios que estemos realizando sean en la rama correcta.

```
PS C:\Users\cgale\IdeaProjects\GIT> git status
On branch master

No commits yet

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

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

Git add .

Justo despues de verificar que hemos relaizado los cambios correctos, en la rama correspondiente, procedemos a usar git add . Para que todos los cambios que realizamos se agreguen correctamente al repositorio.

Git commit -m "Aca se agrega un comentario"

Con este comando, le agregaremos un comentario a nuestros cambios realizados en el repositorio que acabamos de subir.

```
PS C:\Users\cgale\IdeaProjects\GIT> git add .
PS C:\Users\cgale\IdeaProjects\GIT> git commit -m "Lista la tarea"
[main fbb4497] Lista la tarea
1 file changed, 1 insertion(+)
```

Git push origin

En este comando lo que hacemos es enviar todos nuestros cambios realizados en el repositorio local a un repositorio remoto, aca despues del origin tenemos que escribir el nombre de la rama en la cual estamos

```
PS C:\Users\cgale\IdeaProjects\GIT> git push origin main
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 8 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (4/4), 408 bytes | 408.00 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/C-Galeano/GIT.git
402a7c7..fbb4497 main -> main
```

Y como podemos ver, ya se ha enviado todo a nuestro repositorio remoto



```
Code Blame 11 lines (10 loc) · 493 Bytes Code 55% faster with GitHub Copilot
1 //TIP To <b>Run</b> code, press <shortcut actionId="Run"/> or
2 // click the <icon src="AllIcons.Actions.Execute"/> icon in the gutter.
3 public class Main {
4     public static void main(String[] args) {
5         //TIP Press <shortcut actionId="ShowIntentionActions"/> with your caret at the highlighted text
6         // to see how IntelliJ IDEA suggests fixing it.
7         System.out.printf("HOLA, mi nombre es Cristian");
8         System.out.printf(" Este es mi trabajo del manual de git");
9     }
10 }
11 }
```

git pull origin <rama>:

Descarga los cambios de la rama especificada del repositorio remoto origin y los fusiona con tu rama local actual.

Git switch -c <nombre de la rama>

Con este comando lo que haremos es crear una nueva rama en nuestro repositorio local

```
PS C:\Users\cgale\IdeaProjects\GIT> git switch -c semestre
Switched to a new branch 'semestre'
```

Git switch <nombre de la rama>

Este comando nos sirve para cambiar entre ramas, hay que tener cuidado ya que es similar al anterior comando, para diferenciarlo el de crear rama lleva “-c”.

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

Para publicar la rama en el repositorio remoto usamos el comando git push origin, nuevamente.

Git branch -D

Con este comando borraremos la rama creada

```
PS C:\Users\cgale\IdeaProjects\GIT> git branch -D Semestre
Deleted branch Semestre (was fbb4497).
```

Git pull origin <nombre de la rama>
Aca borraremos los cambios realizados

```
PS C:\Users\cgale\IdeaProjects\GIT> git pull origin semestre
from https://github.com/C-Galeano/GIT
* branch               semestre -> FETCH_HEAD
```

git fetch --all:

Descarga todas las ramas y commits de todos los repositorios remotos, pero no los fusiona con tus ramas locales. Útil para ver los cambios en el servidor sin integrarlos inmediatamente.

git branch -r:

Lista todas las ramas remotas

git log:

Muestra el historial de commits. Puedes usar varias opciones para formatear la salida.

```
$ git log
commit 8987bd0a24b9570858d2aa9b138605a9549278c0 (HEAD -> main, origin/main)
Author: C-Galeano <cristianagaleano@ucundinamarca.edu.co>
Date: Thu Feb 20 16:55:14 2025 -0500

    first commit
```

git reflog:

Muestra un registro de las acciones que has realizado en tu repositorio local, incluyendo cambios de rama, commits, etc. Útil para recuperar commits perdidos.

```
$ git reflog
8987bd0 (HEAD -> main, origin/main) HEAD@{0}: Branch: renamed refs/heads/master to refs/heads/main
8987bd0 (HEAD -> main, origin/main) HEAD@{2}: commit (initial): first commit

cristianagale@Cristian MINGW64 ~/OneDrive/Escritorio/COMANDOS GIT (main)
$ git reflog
8987bd0 (HEAD -> main, origin/main) HEAD@{0}: Branch: renamed refs/heads/master to refs/heads/main
8987bd0 (HEAD -> main, origin/main) HEAD@{2}: commit (initial): first commit
```

git log --oneline:

Muestra el historial de commits en una línea por commit, mostrando el hash abreviado y el mensaje del commit.

```
$ git log --oneline
8987bd0 (HEAD -> main, origin/main) first commit
```

git merge <rama>:

Fusiona los cambios de la rama especificada (<rama>) en tu rama actual.

git merge --rebase <rama>:

Similar a git merge, pero reescribe el historial de commits para que parezca que los cambios se integraron en una sola línea. Útil para mantener un historial más limpio, pero puede ser más complicado.

git revert <commit>:

Crea un nuevo commit que deshace los cambios introducidos por el commit especificado (<commit>)

git push origin --delete <rama>:

Elimina la rama especificada (<rama>) del repositorio remoto origin

CONCLUSIONES

Con este manual he logrado entender mejor el uso de lenguaje Git para uso de repositorios dentro de git hub, esto para realizar los trabajos durante el semestre y así mismo para futuros empleos que pueda obtener como desarrollador

Referencias

- Git SCM. (s.f.). *Git documentation*. Recuperado el 2 de junio de 2024, de <https://git-scm.com/>
- Gemini. (IA de google)