

CONTROL DE VERSIONES CON GIT Y GITHUB

Trabajo practico 1

Instituto superior de formación docente y técnico

n°70 Carrera: técnico superior en análisis de sistema

Materia: ingeniería de software II

Fecha de entrega: 11-07-2025

Profesora: Marina Caseres

Alumnos de tercer año

Barragán Betina – Pinto

Hernán

Betinabarragan13@gmail.co
m

Tabla de contenido

Parte 1: Iniciación con git y configuración básica.	
2 Consigna 1: Configurar git en tu equipo.	
2 Consigna 2: Crear un repositorio local y realizar tu primer commit.	2
Parte 2: Repositorio remoto y sincronización.....	
3 Consigna 3: Crear un repositorio remoto y vincularlo.....	
3 Parte 3: Trabajo con ramas.....	
4 Consigna 4: crear y trabajar con una nueva rama.	

4 Consigna 5: Fusionar ramas.	5
Parte 4: colaboración y trabajo en equipo.....	
6 Consigna 6: Agregar un colaborador.....	6
Capturas adicionales.....	6
conclusión.....	6

Parte 1: Iniciación con git y configuración

básica. Consigna 1: Configurar git en tu equipo.

En esta primera parte, tuve inconvenientes ya que no leí bien como había que poner el comando para poder configurar mis datos junto a mi usuario de git y comencé a poner espacios donde no los iba, lo cual al dar enter no me dejaba configurar mis datos de usuarios, no me leía la información y no sabía porque era hasta que busqué en la inteligencia artificial que quería decir esa línea y me respondió que era un error en la sintaxis y había un espacio de más.

En la captura de pantalla no se llega a ver pero me decía algo parecido como “fatal error”.

```
MINGW64/C:/Users/empan/Documents/practica-git
empan@LAPTOP-P07K0Q94 MINGW64 ~
$ git config --global user.name "HernanCristian Pinto"
empan@LAPTOP-P07K0Q94 MINGW64 ~
$ git config --list
error: key does not contain a section: list
empan@LAPTOP-P07K0Q94 MINGW64 ~
$ 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=openssl
core.autocrlf=true
core.fsmonitor=true
core.symlinks=false
pull.rebase=false
credential.helper=manager
credential.helper=manager
credential.helper=manager
init.defaultbranch=master
user.name=HernanCristian Pinto
user.email=hermanpinto.1t@gmail.com
empan@LAPTOP-P07K0Q94 MINGW64 ~
$ cd ~/Documents
empan@LAPTOP-P07K0Q94 MINGW64 ~/Documents
$ mkdir practica-git
empan@LAPTOP-P07K0Q94 MINGW64 ~/Documents
$ cd practica-git
empan@LAPTOP-P07K0Q94 MINGW64 ~/Documents/practica-git
$ git init
Initialized empty Git repository in C:/Users/empan/Documents/practica-git/.git/
empan@LAPTOP-P07K0Q94 MINGW64 ~/Documents/practica-git (master)
$ echo " Proyecto Git" > README.md
empan@LAPTOP-P07K0Q94 MINGW64 ~/Documents/practica-git (master)
$ git status
On branch master
No commits yet
Untracked files:
  (use "git add <file>..." to include in what will be committed)
    README.md
nothing added to commit but untracked files present (use "git add" to track)
empan@LAPTOP-P07K0Q94 MINGW64 ~/Documents/practica-git (master)
$ git add README.md
warning: in the working copy of 'README.md', LF will be replaced by CRLF the next time Git touches it
empan@LAPTOP-P07K0Q94 MINGW64 ~/Documents/practica-git (master)
$ git commit -m "Primer commit: agrega README.md"
[master (root-commit) 4223d11] Primer commit: agrega README.md
1 file changed, 1 insertion(+)
create mode 100644 README.md
```

Consigna 2: Crear un repositorio local y realizar tu primer commit.

En la consigna dos me sucedió algo pareció a lo ocurrido en la consigna 1, pero viceversa, ya que esta vez no deje espacio donde los iba y me volvió a advertir con un error en la sintaxis y al ya haber consultado anteriormente en la inteligencia en la consigna 1 me puse a chequear los espacios si estaban ubicados correctamente, luego de corregir eso pude seguir normalmente con las consignas.

```

MINGW64/c/Users/Betina/Documents/practica-git
$ cd ^[[200~/Documents
bash: cd: $'\E[200~/Documents': No such file or directory

Betina@LAPTOP-HD8L858R MINGW64 ~
$ cd ~/Documents

Betina@LAPTOP-HD8L858R MINGW64 ~/Documents
$ mkdir practica-git

Betina@LAPTOP-HD8L858R MINGW64 ~/Documents
$ cd practica-git

Betina@LAPTOP-HD8L858R MINGW64 ~/Documents/practica-git
$ git init
Initialized empty Git repository in C:/Users/Betina/Documents/practica
-git/.git/

Betina@LAPTOP-HD8L858R MINGW64 ~/Documents/practica-git (master)
$ echo "# proyecto git" > README.md

Betina@LAPTOP-HD8L858R MINGW64 ~/Documents/practica-git (master)
$ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
      README.md

nothing added to commit but untracked files present (use "git add" to
track)

Betina@LAPTOP-HD8L858R MINGW64 ~/Documents/practica-git (master)
$ git add README.md
warning: in the working copy of 'README.md', LF will be replaced by CR
LF the next time git touches it

Betina@LAPTOP-HD8L858R MINGW64 ~/Documents/practica-git (master)
$ git commit -m "Primer commit: agrega README.md"
[master (root-commit) 6cc32f4] Primer commit: agrega README.md
1 file changed, 1 insertion(+)
 create mode 100644 README.md

Betina@LAPTOP-HD8L858R MINGW64 ~/Documents/practica-git (master)
$ |

```

Parte 2: Repositorio remoto y sincronización.

Consigna 3: Crear un repositorio remoto y vincularlo.

En esta consigna ya al tener un poco mas de conocimientos y practica luego de haber realizado las demás consignas se hizo más fácil por decirlo de alguna manera.

Pero de igual manera surgieron imprevistos, a la hora de completar la autenticación en el browser ya que había olvidado la contraseña de mi git y tuve que hacer un par de cosas para la autenticación y confirmar de que era yo, lo cual se tardó más y al consultarle a la inteligencia artificial si era normal que se tarde tanto la autenticación me dijo que volviera a poner los comandos porque quizás no los lee bien por ese problema que había tenido y así fue, volví a comenzar de nuevo con esta consigna y ahí si pude completar la tarea.

Por eso hay dos capturas de pantallas que corresponden a esta consigna.

Parte 3: Trabajo con ramas.

Consigna 4: crear y trabajar con una nueva rama.

```
MINGW64/c/Users/empan/Documents/practica-git
empan@APTOP-P07K0Q94 MINGW64 ~/Documents/practica-git (master)
$ git log
commit 4223d110532dfb847764d534421230b304bcb9a (HEAD -> master)
Author: HernanCristianPinto <hernanpinto.1@gmail.com>
Date: Mon Jul 7 10:16:53 2025 -0300

    Primer commit: agrega README.md

empan@APTOP-P07K0Q94 MINGW64 ~/Documents/practica-git (master)
$ git remote add origin https://github.com/HernanCristianPinto/practica-git.git
empan@APTOP-P07K0Q94 MINGW64 ~/Documents/practica-git (master)
$ git branch -M main
empan@APTOP-P07K0Q94 MINGW64 ~/Documents/practica-git (main)
$ git push -u origin main
info: please complete authentication in your browser...
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 258 bytes | 258.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/HernanCristianPinto/practica-git.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.

empan@APTOP-P07K0Q94 MINGW64 ~/Documents/practica-git (main)
$ git pull origin main
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
Unpacking objects: 100% (3/3), 912 bytes | 41.00 KiB/s, done.
From https://github.com/HernanCristianPinto/practica-git
 * branch            4223d11..9222534 main -> origin/main
Updating 4223d11..9222534
  fast-forward
   README.md | 3 +++
   1 file changed, 3 insertions(+)

empan@APTOP-P07K0Q94 MINGW64 ~/Documents/practica-git (main)
$ git checkout -b desarrollo
Switched to a new branch 'desarrollo'

empan@APTOP-P07K0Q94 MINGW64 ~/Documents/practica-git (desarrollo)
$ echo "Este archivo está en desarrollo." > desarrollo.txt

empan@APTOP-P07K0Q94 MINGW64 ~/Documents/practica-git (desarrollo)
$ git add desarrollo.txt
warning: in the working copy of 'desarrollo.txt', LF will be replaced by CRLF the next time Git touches it

empan@APTOP-P07K0Q94 MINGW64 ~/Documents/practica-git (desarrollo)
$ git commit -m "Agrega archivo desarrollo.txt en rama desarrollo"
[desarrollo 8fe4bb] Agrega archivo desarrollo.txt en rama desarrollo
1 file changed, 1 insertion(+)
 create mode 100644 desarrollo.txt

empan@APTOP-P07K0Q94 MINGW64 ~/Documents/practica-git (desarrollo)
$ git branch
* desarrollo
  main

empan@APTOP-P07K0Q94 MINGW64 ~/Documents/practica-git (desarrollo)
$ git checkout main
```

En esta consigna se me dificultó un poco más que las demás, ya que había dejado pasar un día y lo poco que recordaba no alcanzaba para resolver esta consigna.

Busqué que problemas podrían ser lo que me había marcado esa línea y era que no me encontraba dentro de la carpeta que tenía el repositorio, por ende, no se iba a crear la rama porque era el lugar incorrecto así que, decidí entrar nuevamente a documentos y a la carpeta llamada practica-git y ahí sí pude crear la rama ya que era donde estaba el repositorio.

También me hizo una advertencia que al chequear en la inteligencia artificial me dijo que no afectaba en nada mi funcionamiento ni mi trabajo, ya que era cuando guarde el archivo o cuando lo vuelva a abrir se puede cambiar el formato del mismo.

```
MINGW64/c/Users/Betina/Documents/practica-git
Betina@LAPTOP-HD8L858R MINGW64 ~
$ git checkout -b desarrollo
Fatal: not a git repository (or any of the parent directories): .git

Betina@LAPTOP-HD8L858R MINGW64 ~
$ cd ~/Documents/practica-git

Betina@LAPTOP-HD8L858R MINGW64 ~/Documents/practica-git (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean

Betina@LAPTOP-HD8L858R MINGW64 ~/Documents/practica-git (main)
$ git checkout -b desarrollo
Switched to a new branch 'desarrollo'

Betina@LAPTOP-HD8L858R MINGW64 ~/Documents/practica-git (desarrollo)
$ echo "este archivo esta en desarrollo." > desarrollo.txt

Betina@LAPTOP-HD8L858R MINGW64 ~/Documents/practica-git (desarrollo)
$ git add desarrollo.txt
warning: in the working copy of 'desarrollo.txt', LF will be replaced by CRLF the next time Git touches it

Betina@LAPTOP-HD8L858R MINGW64 ~/Documents/practica-git (desarrollo)
$ git commit -m "agrega archivo desarrollo.txt en rama desarrollo"
[desarrollo 398c365] agrega archivo desarrollo.txt en rama desarrollo
1 file changed, 1 insertion(+)
create mode 100644 desarrollo.txt

Betina@LAPTOP-HD8L858R MINGW64 ~/Documents/practica-git (desarrollo)
$ |
```

Consigna 5: Fusionar ramas.

Esta consigna siguiendo los pasos en el trabajo practico de los comandos, me resulto fácil, no me devolvió ningún error ni nada.

```
MINGW64/c/Users/Betina/Documents/practica-git
Betina@LAPTOP-HD8L858R MINGW64 ~/Documents/practica-git (desarrollo)
$ git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.

Betina@LAPTOP-HD8L858R MINGW64 ~/Documents/practica-git (main)
$ git merge desarrollo
Updating 6cc32f4..398c365
Fast-forward
 desarrollo.txt | 1 +
 1 file changed, 1 insertion(+)
 create mode 100644 desarrollo.txt

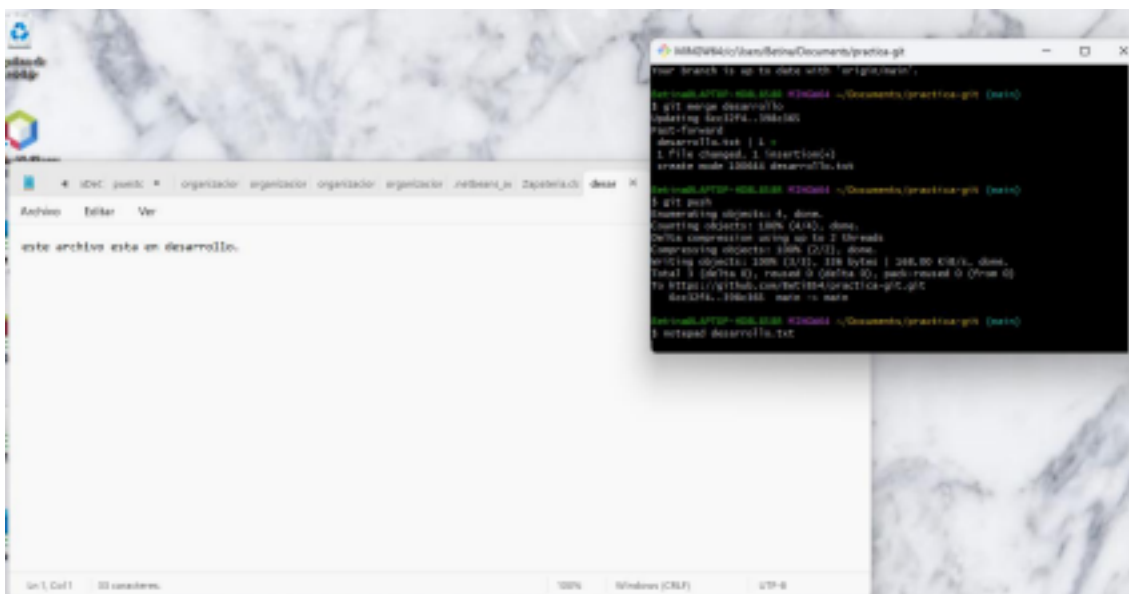
Betina@LAPTOP-HD8L858R MINGW64 ~/Documents/practica-git (main)
$ git push
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 2 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 336 bytes | 168.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/BetiB54/practica-git
 6cc32f4..398c365  main -> main
```

Parte 4: colaboración y trabajo en

equipo. Consigna 6: Agregar un colaborador.

Capturas adicionales.

Esta captura muestra que en git bash, coloque ese comando para que abra el bloq de notas y nos muestre lo que esta adentro del archivo desarrollo.txt.



Consigna 7: Clonar un repositorio

¿En qué se diferencia clonar un repositorio de crear uno nuevo? ¿Por qué es importante clonar en lugar de descargar un ZIP?)

- Clonar sirve para trabajar de forma completa y colaborativa con un proyecto Git.
- Descargar como ZIP solo te sirve si querés ver el código, pero no trabajar con Git.
- Si vas a programar o hacer cambios, **cloná** el repositorio. Si solo querés mirar, podés descargar el

ZIP.

conclusión.

Este trabajo práctico resultó muy útil para familiarizarme con los principales comandos de Git, una herramienta fundamental para la gestión de proyectos. Gracias a esta experiencia, ahora me siento más preparado para aplicarlos en futuras materias, especialmente en trabajos grupales donde la organización y la colaboración son claves para obtener buenos resultados.