

ISFDyT N°70
Tecnicatura Superior en Análisis de Sistemas
Ingeniería de Software II

- Trabajo Práctico 1 – Control de versiones con Git y GitHub
- Eguia Juan Manuel; Falabella Valentino
- 3° año
- **Nombre de la profesora:** Marina Caseres
- **Fecha de entrega:** 11/07/2025
- Enlace al repositorio GitHub del grupo:
- juanma.egui@gmail.com ; valentinofala@gmail.com

Trabajo Practico 1 – Control de versiones con Git y GitHub

Parte 1: Iniciación a Git y configuración básica

```
MINGW64:/c/Users/Educacion
Educacion@DESKTOP-PPICMMB MINGW64 ~
$ git config --global user.name Valentino Falabella

Educacion@DESKTOP-PPICMMB MINGW64 ~
$ git config --global user.email valentinoofala@gmail.com

Educacion@DESKTOP-PPICMMB 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=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=Valentino
user.email=valentinoofala@gmail.com

Educacion@DESKTOP-PPICMMB MINGW64 ~
$
```

```
Educacion@DESKTOP-PPICMMB MINGW64 ~
$ cd ~/Documents

Educacion@DESKTOP-PPICMMB MINGW64 ~/Documents
$ mkdir practica-git

Educacion@DESKTOP-PPICMMB MINGW64 ~/Documents
$ cd practica-git

Educacion@DESKTOP-PPICMMB MINGW64 ~/Documents/practica-git
$ git init
Initialized empty Git repository in C:/Users/Educacion/Documents/practica-git/.git/

Educacion@DESKTOP-PPICMMB MINGW64 ~/Documents/practica-git (master)
$ echo "# Proyecto Git" > README.md

Educacion@DESKTOP-PPICMMB 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)
```

1-¿Dónde queda guardado el control de versiones? ¿En la nube o en tu equipo?

El control de versiones queda almacenado localmente en el equipo, dentro de la carpeta del proyecto (en este caso, en la ubicación Documentos). Git crea un directorio oculto llamado .git, donde guarda todo el historial de cambios y configuración del repositorio.

2- ¿Qué información te brinda git status? ¿Por qué es útil antes de hacer un commit?

El comando git status proporciona información sobre el estado actual del repositorio, incluyendo los archivos modificados, los que están listos para ser confirmados (staged) y los que no están siendo seguidos por Git. Es una herramienta fundamental antes de realizar un commit, ya que permite verificar exactamente qué cambios se registrarán, ayudando a evitar errores o confirmaciones involuntarias.

3- ¿Cuál es la diferencia entre agregar (add) y guardar un cambio (commit)?

¿Podés hacer un commit sin hacer un add antes?

¿Podés hacer un commit sin hacer un add antes?

El comando git add se utiliza para preparar (o "staging") los archivos que se desean incluir en el próximo commit. Por su parte, git commit registra de forma permanente esos cambios en el historial del repositorio. No es posible realizar un commit sobre archivos nuevos o modificados si previamente no fueron agregados mediante git add.

```
Educacion@DESKTOP-PPICMM8 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

Educacion@DESKTOP-PPICMM8 MINGW64 ~/Documents/practica-git (master)
$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   README.md

Educacion@DESKTOP-PPICMM8 MINGW64 ~/Documents/practica-git (master)
$ git commit -m "Primer commit: agrega README.md"
[master (root-commit) 0c7c7cc] Primer commit: agrega README.md
 1 file changed, 1 insertion(+)
 create mode 100644 README.md

Educacion@DESKTOP-PPICMM8 MINGW64 ~/Documents/practica-git (master)
$
$ git log
commit 0c7c7ccd067348830601a7da7f5b60150974fb7e (HEAD -> master)
Author: Valentino <valentinofala@gmail.com>
Date:   Fri Jul 4 20:36:26 2025 -0300

    Primer commit: agrega README.md
```

4- ¿Qué información muestra este comando?

El comando git log muestra el historial completo de commits realizados en el repositorio. Incluye información detallada como el identificador único (hash) de cada commit, el autor, la fecha y el mensaje asociado al cambio. Es útil para revisar la evolución del proyecto y rastrear modificaciones específicas.

Parte 2: Repositorio remoto y sincronización

```
Educacion@DESKTOP-PPICMMB MINGW64 ~/Documents/practica-git (master)
$ https://github.com/ValentinoFala/practica-git.git
bash: https://github.com/ValentinoFala/practica-git.git: No such file or directory

Educacion@DESKTOP-PPICMMB MINGW64 ~/Documents/practica-git (master)
$ git remote add origin https://github.com/ValentinoFala/practica-git.git

Educacion@DESKTOP-PPICMMB MINGW64 ~/Documents/practica-git (master)
$ git branch -M main

Educacion@DESKTOP-PPICMMB MINGW64 ~/Documents/practica-git (main)
$ git push -u origin main
info: please complete authentication in your browser...
```

5- ¿Qué significa vincular tu repositorio local con uno remoto? ¿Podés trabajar sin conexión a internet después de hacer esto?)

Vincular el repositorio local con uno remoto implica establecer una conexión entre el proyecto que se encuentra en tu computadora y su copia alojada en una plataforma como GitHub. Esto permite sincronizar los cambios entre ambas versiones. Una vez vinculado, es posible seguir trabajando de forma local sin conexión a internet; los cambios se podrán subir al repositorio remoto cuando haya acceso a la red.

6- ¿Qué estás logrando al hacer un push? ¿Por qué hay que subir los cambios al repositorio remoto si ya están en tu computadora?)

Al ejecutar un git push, se envían los commits realizados en el repositorio local hacia el repositorio remoto, como por ejemplo en GitHub. Este proceso actualiza la versión en línea del proyecto, permitiendo que otros colaboradores puedan acceder a los últimos cambios. Aunque los archivos ya estén en la computadora, subirlos al repositorio remoto es esencial para compartir el trabajo, mantener una copia de respaldo y facilitar la colaboración en equipo.

Parte 3: Trabajo con ramas

```
MINGW64:/c/Users/Educacion/Documents/practica-git

Educacion@DESKTOP-PPICMMB 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), 243 bytes | 243.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/ValentinoFala/practica-git.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.

Educacion@DESKTOP-PPICMMB MINGW64 ~/Documents/practica-git (main)
$ git pull origin main
From https://github.com/ValentinoFala/practica-git
 * branch            main       -> FETCH_HEAD
Already up to date.

Educacion@DESKTOP-PPICMMB MINGW64 ~/Documents/practica-git (main)
$
```

```
MINGW64:/c/Users/Educacion/Documents/practica-git

Educacion@DESKTOP-PPICMMB MINGW64 ~/Documents/practica-git (main)
$ git checkout -b desarrollo
Switched to a new branch 'desarrollo'

Educacion@DESKTOP-PPICMMB MINGW64 ~/Documents/practica-git (desarrollo)
$ echo "Este archivo está en desarrollo." > desarrollo.txt

Educacion@DESKTOP-PPICMMB 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

Educacion@DESKTOP-PPICMMB MINGW64 ~/Documents/practica-git (desarrollo)
$ git commit -m "Agrega archivo desarrollo.txt en rama desarrollo"
[desarrollo 63b7407] Agrega archivo desarrollo.txt en rama desarrollo
1 file changed, 1 insertion(+)
create mode 100644 desarrollo.txt

Educacion@DESKTOP-PPICMMB MINGW64 ~/Documents/practica-git (desarrollo)
$ git branch
* desarrollo
  main

Educacion@DESKTOP-PPICMMB MINGW64 ~/Documents/practica-git (desarrollo)
$ git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.
```

7- ¿Para qué sirve crear una nueva rama? ¿Qué ventajas tiene trabajar en ramas distintas en lugar de hacerlo siempre en la principal?

Crear una nueva rama permite desarrollar nuevas funcionalidades, realizar pruebas o experimentar sin modificar directamente el código principal del proyecto. Trabajar en ramas separadas ayuda a prevenir errores en la rama principal (main), facilita la colaboración entre varios desarrolladores y permite revisar, probar y fusionar los cambios de forma más ordenada y segura.


```
Educacion@DESKTOP-PPICMMB MINGW64 ~/Documents/practica-git (main)
$ git merge desarrollo
Updating 0c7c7cc..63b7407
Fast-forward
 desarrollo.txt | 1 +
 1 file changed, 1 insertion(+)
 create mode 100644 desarrollo.txt

Educacion@DESKTOP-PPICMMB MINGW64 ~/Documents/practica-git (main)
$ git push
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 12 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 337 bytes | 168.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/ValentinoFala/practica-git.git
 0c7c7cc..63b7407  main -> main

Educacion@DESKTOP-PPICMMB MINGW64 ~/Documents/practica-git (main)
$
```

8- ¿Qué pasa si hay diferencias entre las ramas al hacer un merge?

¿Podés revertir un merge si algo sale mal?

Si hay diferencias en las mismas líneas, Git genera un conflicto que debe resolverse manualmente. Si el merge falla, se puede revertir con `git reset` o `git revert`, siempre que haya un commit de respaldo previo.