

## ACTIVIDAD INTERMEDIA

**NOMBRE: CLAUDIA ILARRAGORRI**

1. Crear un repositorio en Gitlab o Github y agregar los archivos de texto relacionados a la clase (actividades, resoluciones de las actividades, anotaciones que pueden ser públicas, imágenes, gráficos, datos, etc).

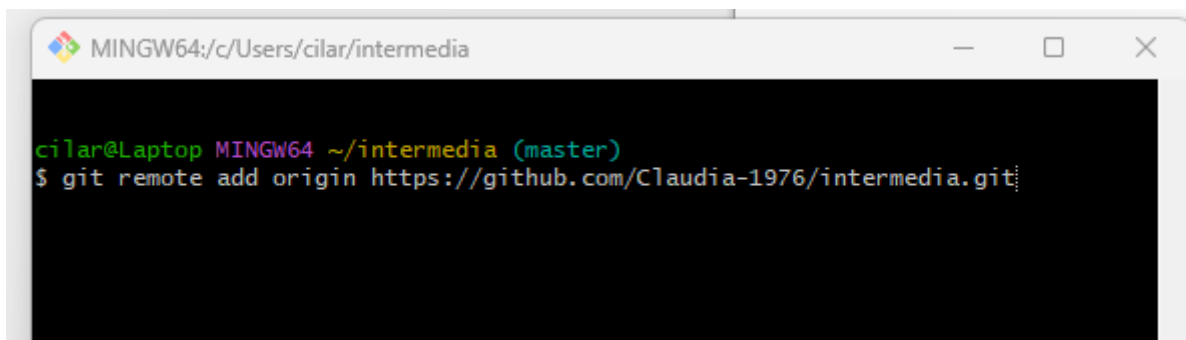
**Repositorio Github :** <https://github.com/Claudia-1976/intermedia.git>

**Herramienta:** Gitbush

```
cilar@Laptop MINGW64 ~/intermedia (main)
$ git init
Initialized empty Git repository in C:/Users/cilar/intermedia/.git/

cilar@Laptop MINGW64 ~/intermedia (master)
$ 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=Claudia Ilarragorri
user.email=cilarragorri@gmail.com
push.autosetupremote=true
core.repositoryformatversion=0
core.filemode=false
core.bare=false
core.logallrefupdates=true
core.symlinks=false
core.ignorecase=true
```

- Conectar al repositorio remoto



The screenshot shows a terminal window titled "MINGW64:/c/Users/cilar/intermedia". The prompt is "cilar@Laptop MINGW64 ~/intermedia (master)". The command entered is "\$ git remote add origin https://github.com/Claudia-1976/intermedia.git".

2. Crear al menos dos versiones (commits) distintas y obtener una captura con la salida del comando logs donde se vean esos commits.

```
cilar@Laptop MINGW64 ~/intermedia (master)
$ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
  Git- Actividad intermedia- PARMISANO.pdf
  imagenes/

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

cilar@Laptop MINGW64 ~/intermedia (master)
$ git add Git- Actividad intermedia- PARMISANO.pdf
fatal: pathspec 'Git-' did not match any files

cilar@Laptop MINGW64 ~/intermedia (master)
$ git add imagenes/

cilar@Laptop MINGW64 ~/intermedia (master)
$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
    new file:   imagenes/imagen.jpg
    new file:   imagenes/imagen2.jpg
    new file:   imagenes/imagen3.jpg

Untracked files:
  (use "git add <file>..." to include in what will be committed)
  Git- Actividad intermedia- PARMISANO.pdf

cilar@Laptop MINGW64 ~/intermedia (master)
```

```
cilar@Laptop MINGW64 ~/intermedia (master)
$ git commit -m "Primer commit"
[master (root-commit) 4f9b381] Primer commit
3 files changed, 0 insertions(+), 0 deletions(-)
create mode 100644 imagenes/imagen.jpg
create mode 100644 imagenes/imagen2.jpg
create mode 100644 imagenes/imagen3.jpg

cilar@Laptop MINGW64 ~/intermedia (master)
$
```

```
cilar@Laptop MINGW64 ~/intermedia (master)
$ git commit -m "Segundo commit"
[master abab75f] Segundo commit
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 actividad-intermedia.odt


cilar@Laptop MINGW64 ~/intermedia (master)
$
```

3. Pushear el proyecto y obtener una captura del directorio raíz del proyecto en la plataforma utilizada (Github o Gitlab).



```
cilar@Laptop MINGW64 ~/intermedia (master)
$ git push
Enumerating objects: 9, done.
Counting objects: 100% (9/9), done.
Delta compression using up to 8 threads
Compressing objects: 100% (8/8), done.
Writing objects: 100% (9/9), 256.47 KiB | 11.66 MiB/s, done.
Total 9 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
remote:
remote: Create a pull request for 'master' on GitHub by visiting:
remote:   https://github.com/Claudia-1976/intermedia/pull/new/master
remote:
To https://github.com/Claudia-1976/intermedia.git
 * [new branch]      master -> master
branch 'master' set up to track 'origin/master'.
```

- Captura del proyecto en Github

This branch is 2 commits ahead of, 1 commit behind `main`.


 Claudia-1976 Segundo commit

abab75f · 2 minutes ago 2 Commits


 imagenes	Primer commit	8 minutes ago
 actividad-intermedia.odt	Segundo commit	2 minutes ago

- Activity


modificacion archivo readme ...

 Claudia-1976 pushed 1 commit to `master` · abab75f...fd275b8 · 14 seconds ago

Segundo commit ...

 Claudia-1976 created `master` · abab75f · 20 minutes ago

Initial commit ...

 Claudia-1976 created `main` · 8881958 · 4 hours ago

## 4 - Principales diferencias entre Subversion y GIT

### Cuadro comparativo

<b>GIT</b>	<b>SUBVERSION</b>
Distribuido.	Centralizado.
Puede actuar como cliente y servidor	Necesita un cliente y un servidor por separado
La mayoría de sus funciones y características están disponible off-line	El trabajo off-line es limitado
Asume que cada usuario tiene los mismos permisos de acceso.	Provee una administración de usuarios centralizada.
Fácil de aprender (Curva mayor de aprendizaje que SVN)	Muy fácil de aprender
Es utilizado por la mayoría de los desarrolladores (72% de los repositorios)	Es utilizado por menos usuarios (23% de los repositorios)
Capacidades limitadas para manejar archivos binarios grandes	Manejo eficiente de archivos binarios grandes

### **GIT**

Git es un sistema de control de versiones distribuido, de código abierto, que se utiliza para rastrear cambios en archivos a lo largo del tiempo. Permite a los desarrolladores trabajar colaborativamente, administrar diferentes versiones de código y facilitar la colaboración en proyectos.

Permite guardar instantáneas del estado de los archivos en un punto específico del tiempo, lo que facilita revertir a versiones anteriores o comparar cambios.

Cada desarrollador tiene una copia completa del repositorio lo que facilita trabajar colaborativamente con otros usuarios. Tiene la funcionalidad de "ramas" o branches que posibilita desarrollar nuevas funcionalidades sin afectar el código principal hasta tanto este lista para integrarse.

### **SVN (SUBVERSION)**

Un repositorio Subversion (SVN) es una base de datos centralizada que almacena versiones de archivos y directorios, y que mantiene un historial completo de todos los cambios realizados. Es el corazón del sistema de control de versiones de Subversion, permitiendo a múltiples usuarios colaborar en un proyecto compartiendo y gestionando las modificaciones.

El repositorio SVN actúa como una base de datos central donde se guarda la información del proyecto, incluyendo el código fuente, documentos y otros archivos relevantes.

SVN registra todos los cambios realizados en el repositorio, creando una especie de "árbol de versiones" que permite acceder a versiones anteriores del proyecto.

Permite a múltiples usuarios trabajar en el mismo proyecto, compartiendo cambios y manteniendo una copia de trabajo local sincronizada con el repositorio central.

## 5- Enumerar y describir las principales diferencias entre Gitlab y Github.

GitHub se destaca por su comunidad de desarrolladores y su facilidad de uso, ideal para proyectos de código abierto y equipos pequeños. GitLab, por otro lado, ofrece una plataforma DevOps completa con funciones de Integración continua/Entrega continua (CI/CD) integradas, lo que lo convierte en una buena opción para entornos empresariales que buscan una gestión de proyectos más integral.

Github	GitLab
<b>Ventajas</b> <ul style="list-style-type: none"><li>• La configuracion es sencilla</li><li>• Interfaz de usuario potente</li><li>• Orientado a compartir codigo</li><li>• Favorece la colaboracion remota</li><li>• La comunidad es enorme</li></ul>	<b>Ventajas</b> <ul style="list-style-type: none"><li>• La version de la comunidad de GitLab es gratuita y de código abierto.</li><li>• Documentacion extensa para la importacion o exportacion de datos</li><li>• CI/CD integrado</li><li>• Constantemente agrega nueva funcionalidad</li><li>•Codigo facil de mantener</li><li>• Tiende a ser mas seguro</li></ul>
<b>Contras</b> <ul style="list-style-type: none"><li>• Github en si no es de codigo abierto</li><li>• Carece de CI/CD integrado</li><li>• Desarrollo de API integrado</li><li>• Desarrollo de Api menos refinado</li><li>• Funciones limitadas de para versiones gratuitas</li><li>• Solo los repositorios publicos son gratuitos</li></ul>	<b>Contras</b> <ul style="list-style-type: none"><li>• Comunidad mas pequeña que la de Github</li><li>• La actualizaciones pueden ser complejas</li><li>• Menos maduro que Github</li></ul>