

29 de marzo de 2022

# Git y GitHub

EQUIPO B

- **Herrera Godina Adriana Jocelyn**
- **Sanchez Torres Sergio Daniel**

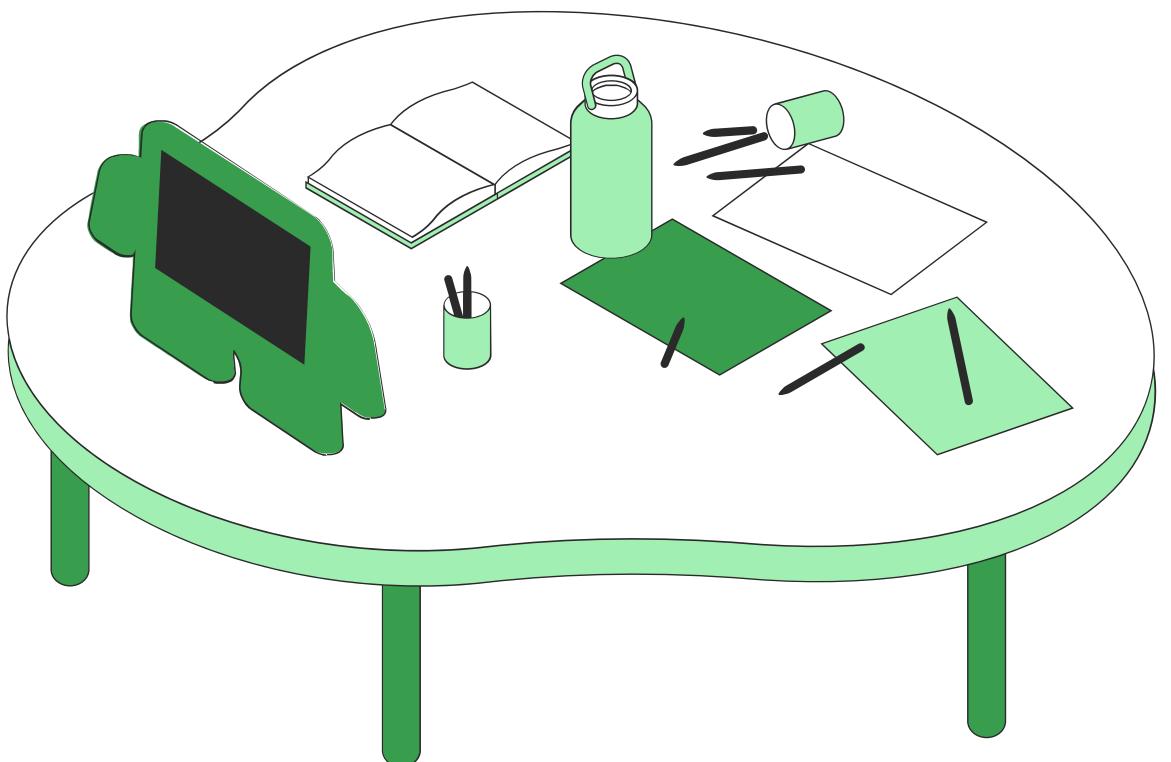


git



# Agenda

¿Qué abordaremos?



- Introducción
- 

- ¿Qué es Git?
- 

- Ventajas de usar Git
- 

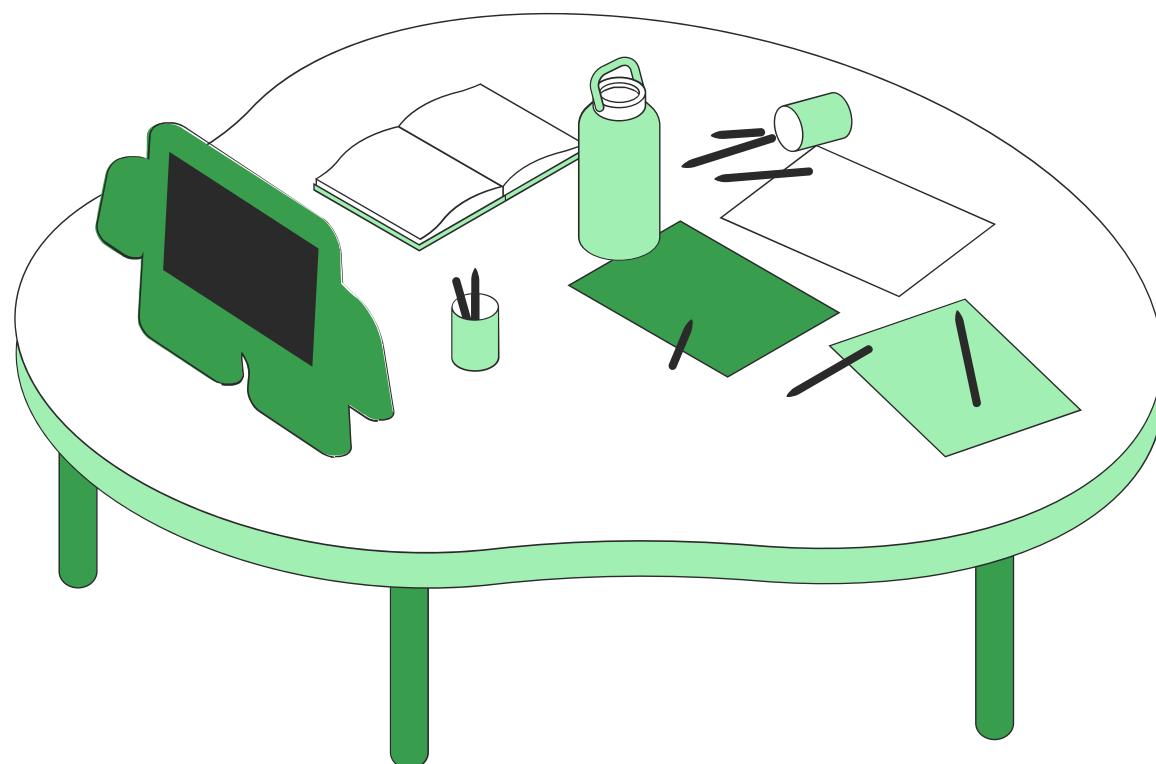
- ¿Cómo funciona Git?
- 

- ¿Qué es GitHub?
- 

- Primeros pasos usando Git y GitHub

# Agenda

¿Qué abordaremos?



- Actividad
- Conclusiones
- Q & A
- Críticas



Van a  
participar  
todos en la  
actividad.  
¡Pongan  
atención!



Nos van a  
ayudar a  
formular las  
conclusiones.  
¡Pongan  
atención!



# Introducción

- Sistema de Control de Versiones - SCV

# ¿Qué es Git?

- Es el sistema de control de versiones moderno más utilizado del mundo



```
isMounted() {
  isMounted = false;

  selectParams = _ => {
    C:\Users\CottonBro>d:
    (^ http://localhost:
      en(response => respo
      en(response => {
        D:\startup-dev\master>npm install
        npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@1.2.13 (node_modules\fsevents):
        npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@1.2.13: wanted {"os":"darwin","arch":"any"} (current: {"os":"win32","arch":"x64"})
        audited 962 packages in 9.118s
      tch(err => console.
      ok jhdjdfvkjdlddkl
      s = _ => {
        found 0 vulnerabilities
      (^ http://localhost:
        en(respo
        en(response => {
          D:\startup-dev\master>npm run dev
          > mesto@1.0.0 dev D:\startup-dev\master
          > webpack serve
          wds: Project is running at http://localhost:8080/
          wds: webpack output is served from
          wds: Content not from webpack is served from D:\startup-dev\master\dist
          wdm: wait until bundle finished: /
          Завершить выполнение пакетного файла [Y(да)/N(нет)]? y
        D:\startup-dev\master>npm run bui
      en(this.getCosts)
      tch(err => console.error(err))

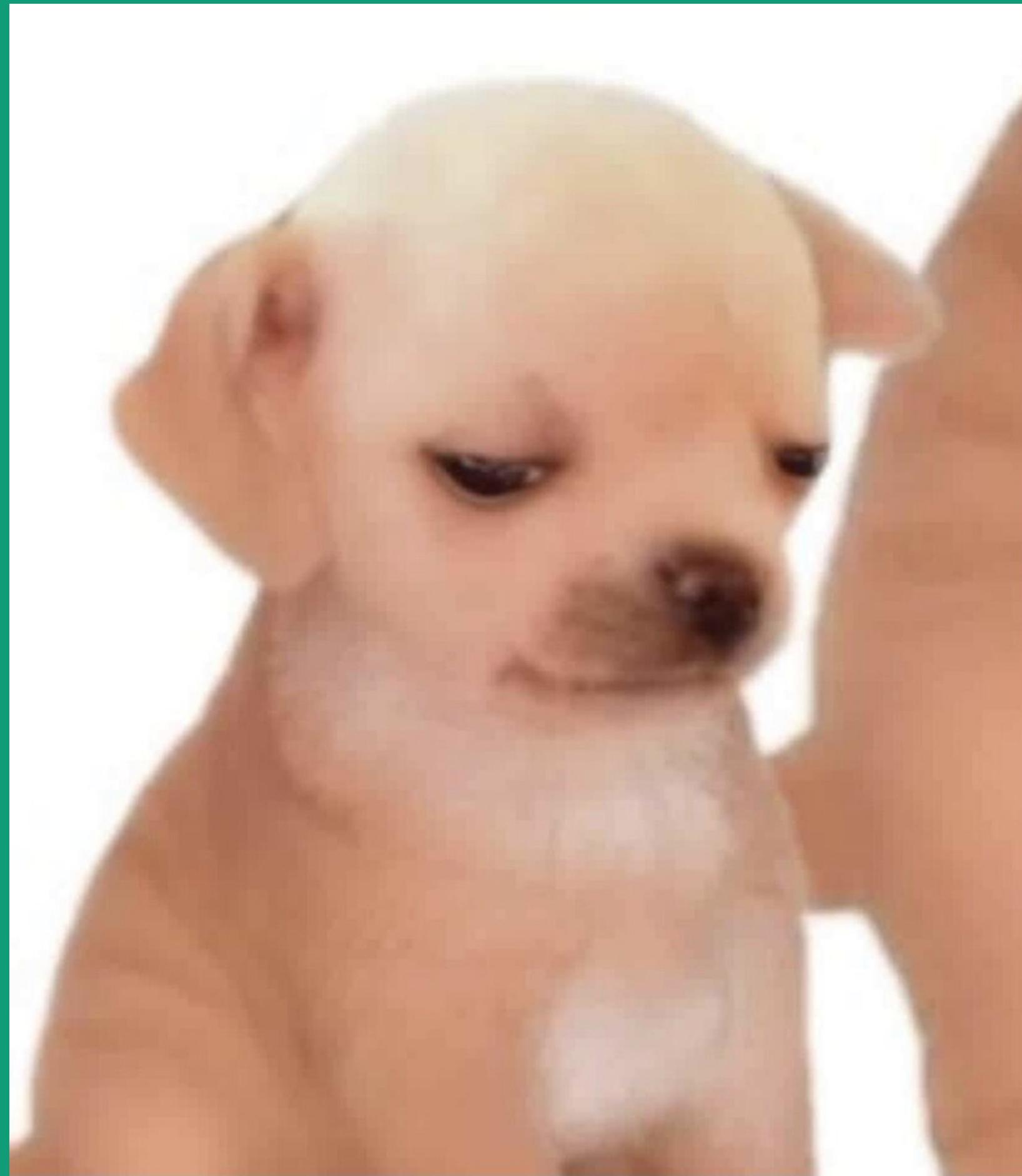
      st = (cost) => {
        ^ http://localhost:4000/project/${this.props.project_id}/costs/delete?cost_id=${cost.cost_id}`)
    
```

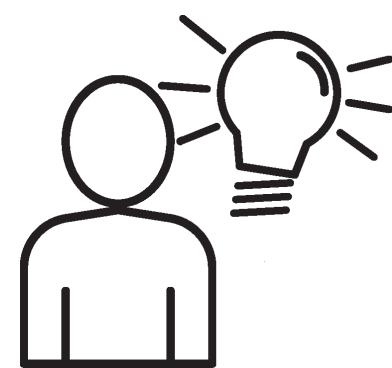


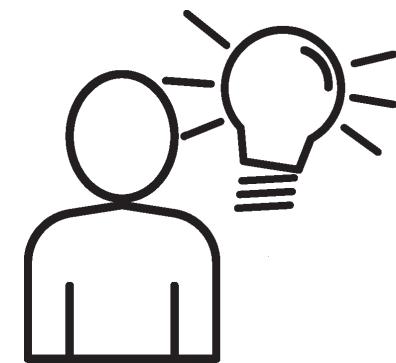
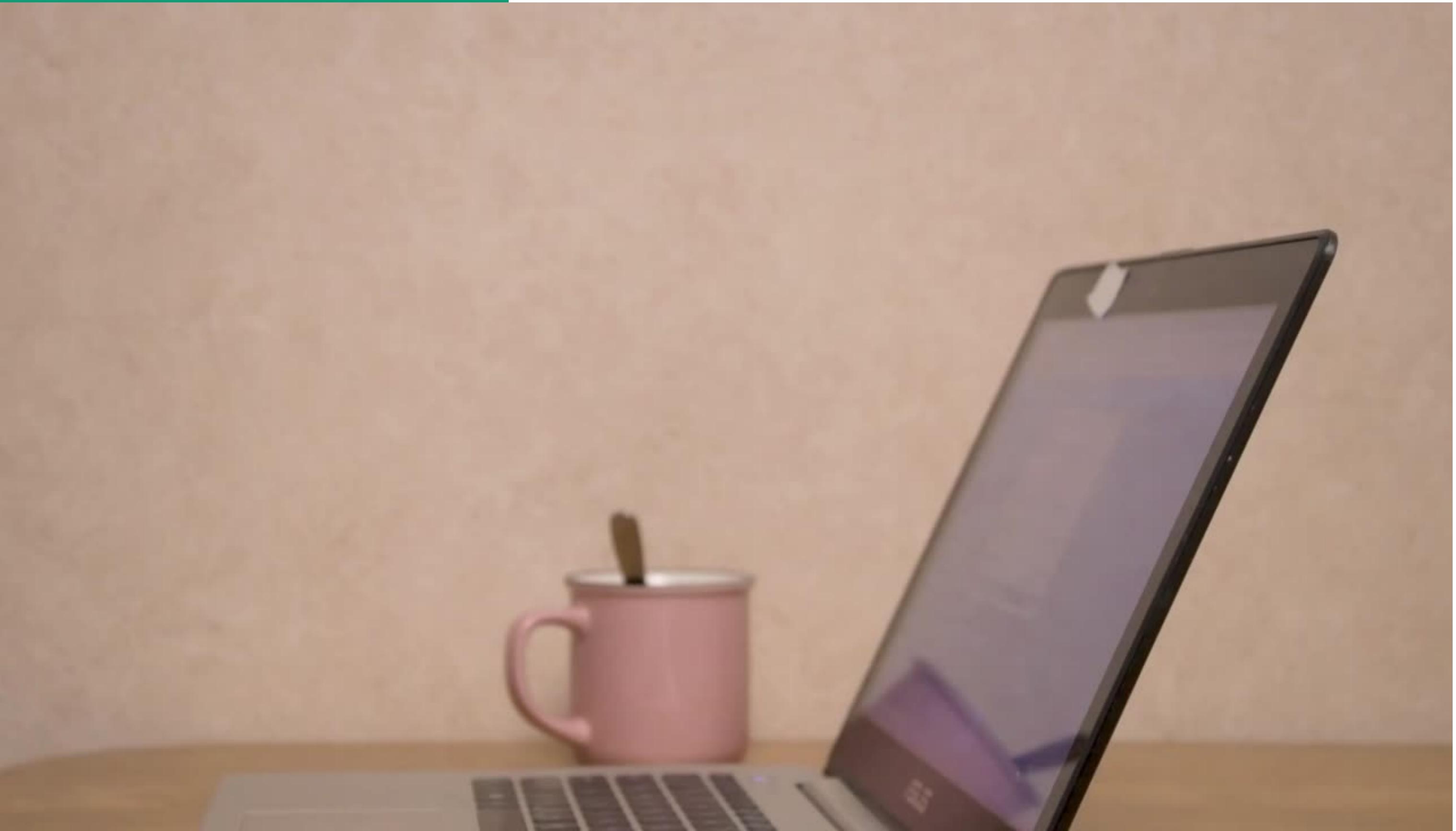


**BitKEEPER**  
Scalable Version Control









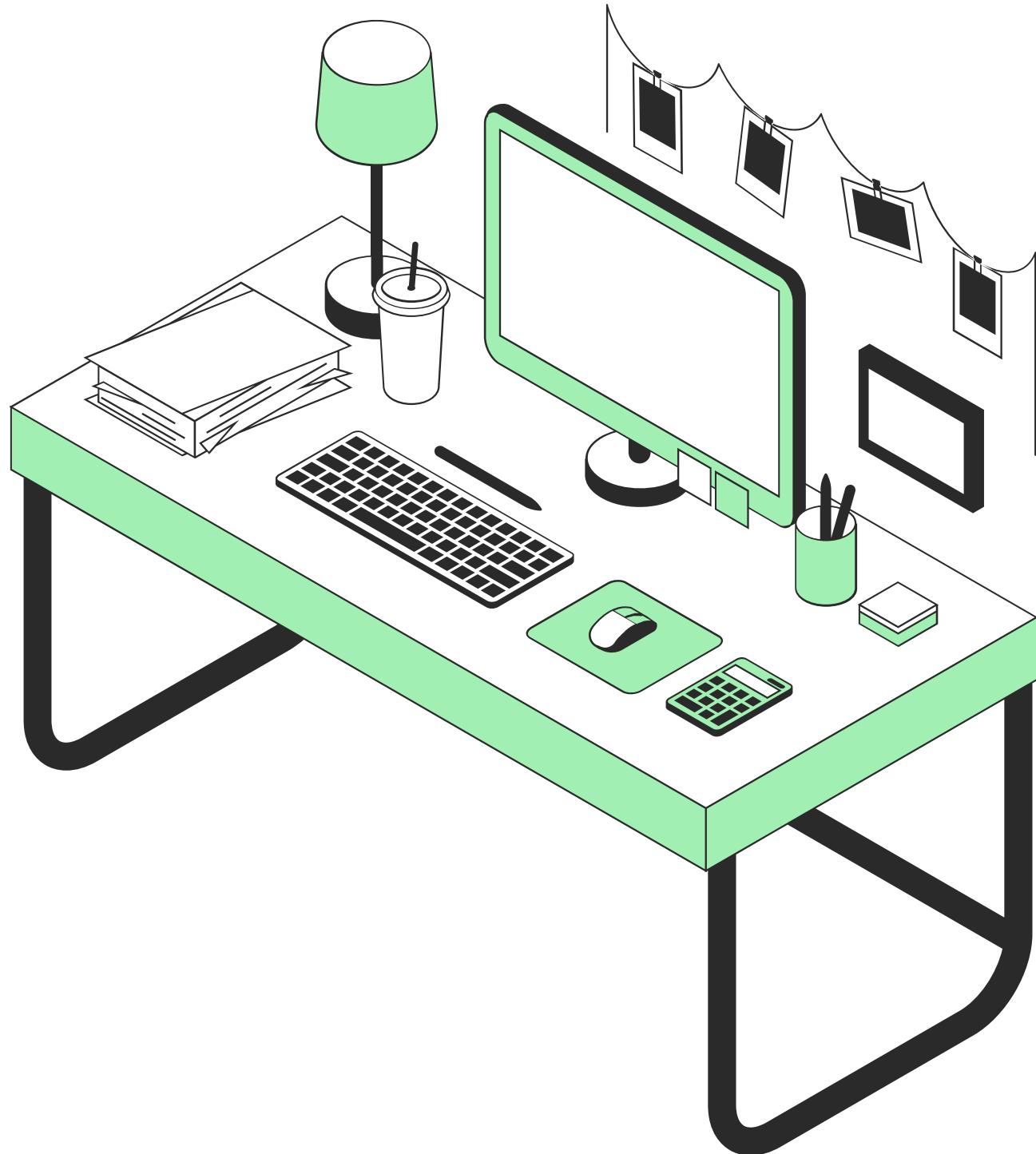


# Ventajas de usar Git



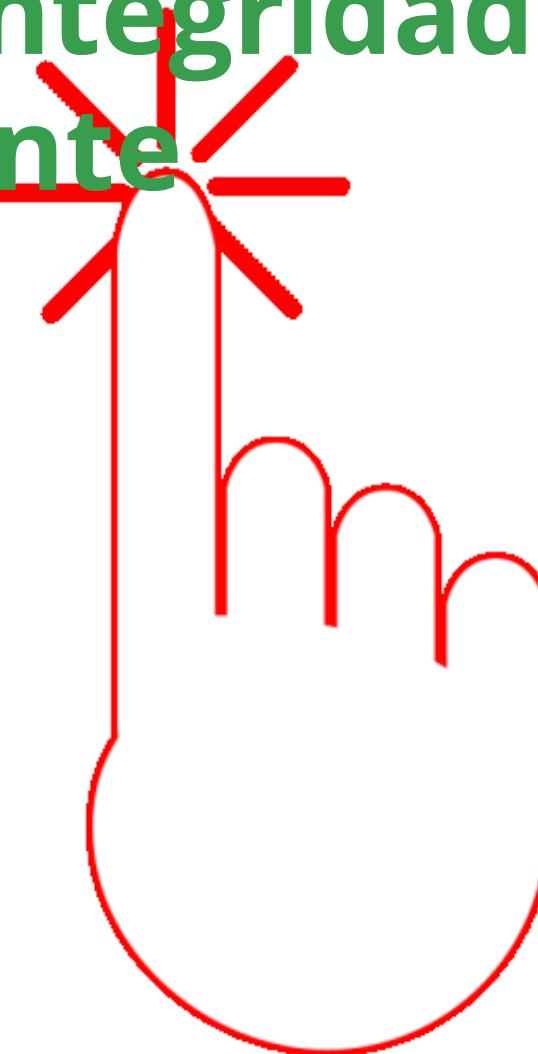
# Rendimiento

Su **objetivo** es centrarse en  
el **contenido** propio del  
archivo.



# Seguridad

Consevar la integridad  
del código fuente  
gestionado



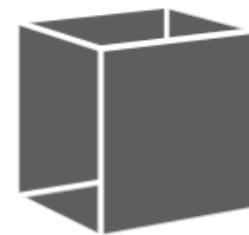
# Flexibilidad

Git se ha ideado para  
posibilitar la ramificación y  
el etiquetado como  
procesos de primera  
importancia



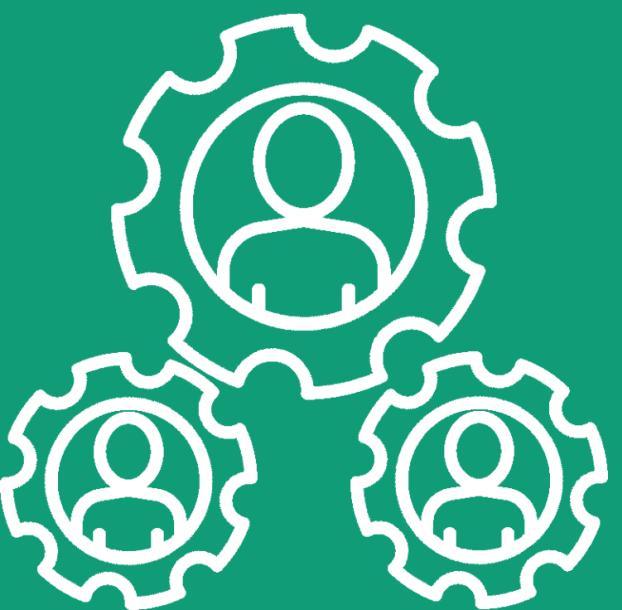
# Más ventajas

- Las operaciones en Git son locales
- Git no necesita del uso de internet.
- Git no borra
- Integridad

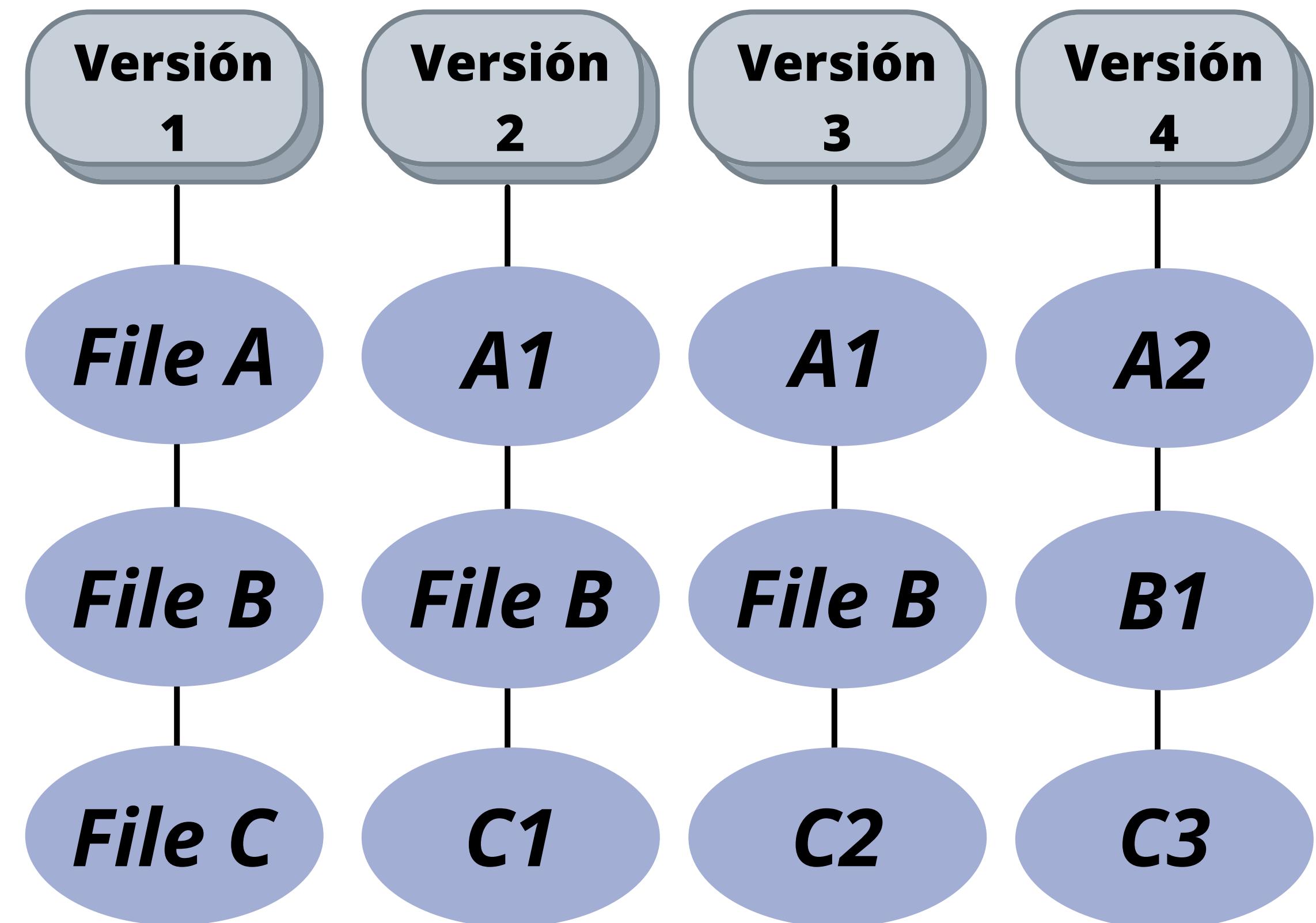


# ¿Cómo funciona Git?

"Fotos instantáneas"  
S snapshots



**SCV**



# Tenemos 3 estados...



- Trabajo cotidiano
  - Copia del proyecto
- Archivo que contiene información sobre la siguiente confirmación
- Lugar donde se almacenan los datos

# Otros conceptos dentro de los estados...



## Modified

- Working Directory
- El archivo ha sido modificado.



## Staged

- Staging Area
- Preparado
- El archivo está "staged"



## Committed

- Local Repository
- Confirmado



**ESO ES ES MUCHO  
MUY IMPORTANTE**

**Los 3  
estados  
anteriores...**

**SOLO OCURREN EN TU  
COMPUTADORA**

# Para confirmar...



git add

git commit

# Comandos básicos



# git config

- Configuración del nombre de usuario y correo
- Solo es necesario hacerlo una vez
- Pueden modificarse los valores



*git config --global user.name "username"*

---

*git config --global user.email "email"*

---

*git config --list*

# git init

- Inicializar o clonar
- Debemos estar en la ruta del proyecto
- Git crea un conjunto de archivos ocultos que NO debemos modificar

*git init*

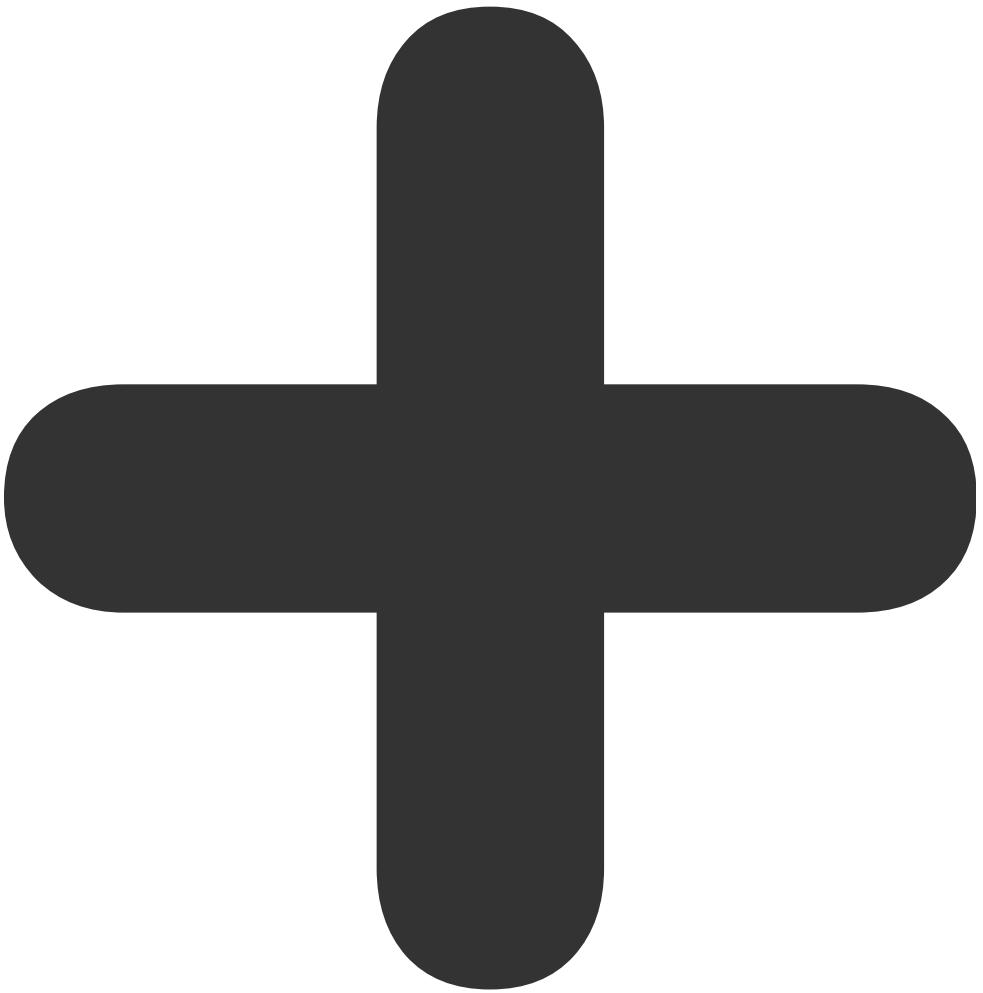


# git add

- Working Directory al Staging Area

*git add fileName.extension*

*git add --all*



# git status

- Ver el estado del proyecto

*git status*



# git commit

- Para pasar al Local Repository

*git commit*

---

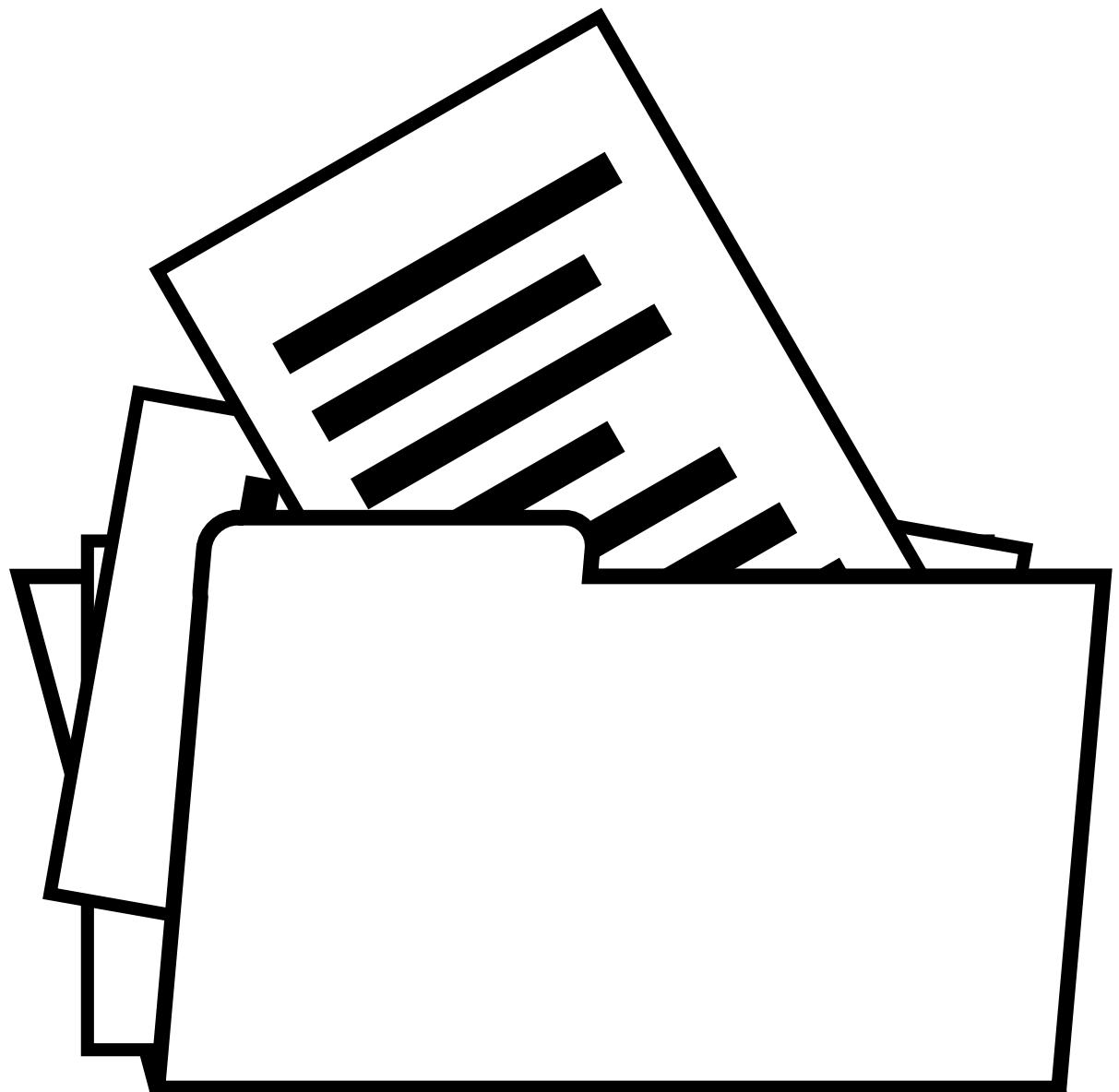
*git commit -m "Description"*



# git log

- Estado de los commits
- Muestra:
  - Código hash
  - Autor
  - Fecha
  - Mensaje

*git log*



# git remote

- Establecemos conexión con el repositorio remoto
- Podemos comprobar si dicha conexión fue establecida

*git remote add repositoryName repositoryAddress*

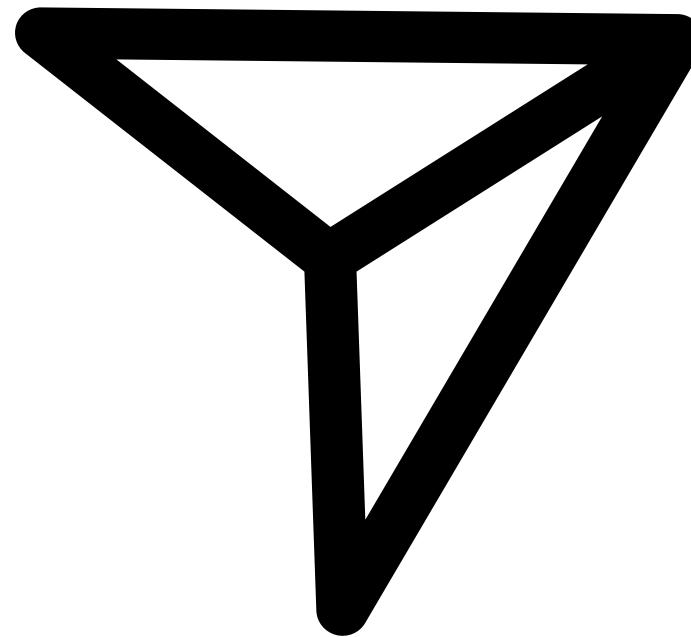
*git remote add origin repositoryAddress*

*git remote -v*



# git push

- Enviamos archivos del repositorio local al repositorio remoto



*git push repositoryName branch*

*git push origin master*

*git push origin*

# git pull

- "Jala" los nuevos cambios hechos
- Del repositorio remoto al repositorio local



*git push origin master*

# git help

- Muestra ayuda acerca de un comando en específico o sobre todos los comandos disponibles



**git help comando**

---

**git help -all**

---

**git help**

# Otros comandos...

- Comandos de reparación básica
- Comando de Branch y Merge
- Comandos de actualización de proyectos, comparación, etc.

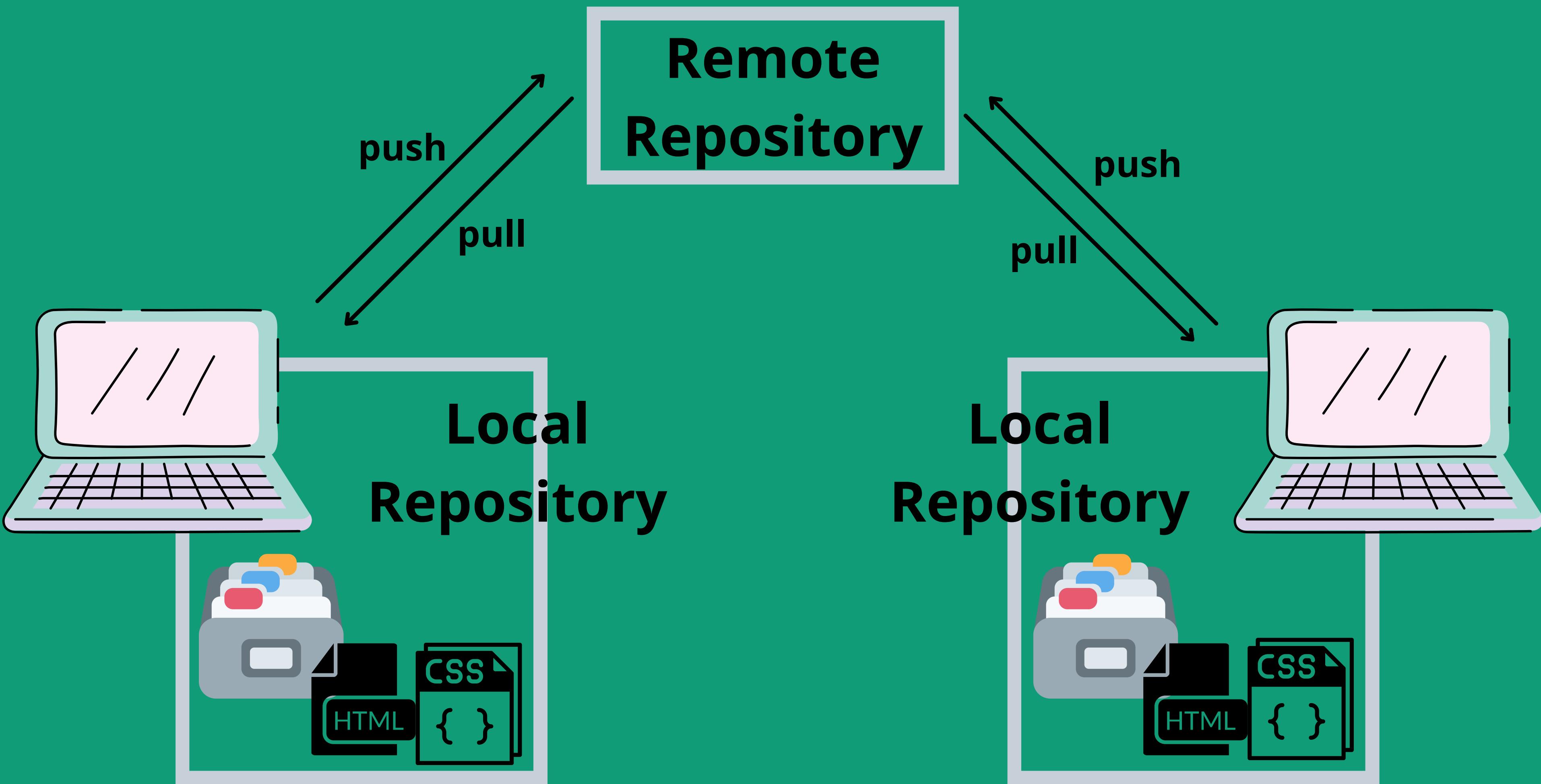




# ¿Qué es GitHub?

¿Para qué nos sirve?

- Repositorio remoto
- "Plataforma de hospedaje", según la documentación de GitHub
- Trabajo colaborativo
- Red social del código
- Interfaz gráfica más amigable



# Operaciones que podemos hacer en GitHub

- Crear repositorios
- 

- Commits
- 

- Branch
- 

- Pull request
- 

- Clone

# Operaciones que podemos hacer en GitHub

- Fork
- Notificaciones
- Organización y colaboradores

# Existen otras herramientas basadas en Git:

- GitLab



# Primeros pasos usando Git y GitHub





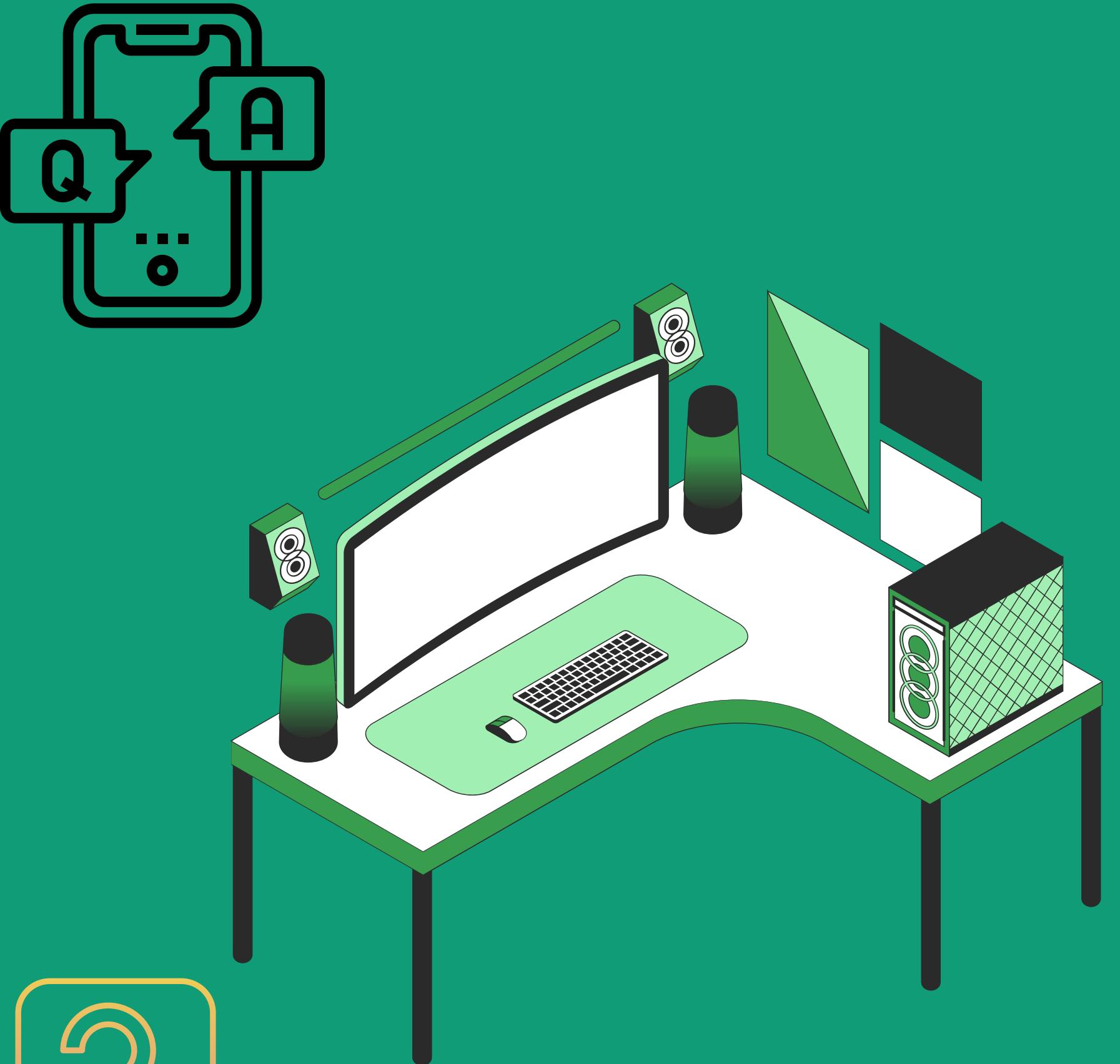
# ACTIVIDAD



# Conclusiones



Thanks a  
LOT!



# Q & A



# Referencias

- **Configurar Git - GitHub Docs.**  
Retrieved 28 March 2022, from  
<https://docs.github.com/es/get-started/quickstart/set-up-git>
- **Git y Github | Curso Práctico de Git y Github Desde Cero. (2018). [Video].**  
Retrieved from  
<https://www.youtube.com/watch?v=HiXLkL42tMU>
- **Colina, C. Curso de Git y Github: GIT PROFESIONAL desde cero + E-Book [Video]. Udemy.**

