

Introducción a git y GitHub



ÍNDICE

- ¿Qué es git?
- ¿Qué es GitHub?
- Crear una cuenta de GitHub
- GitHub Desktop
- Comandos básicos de git

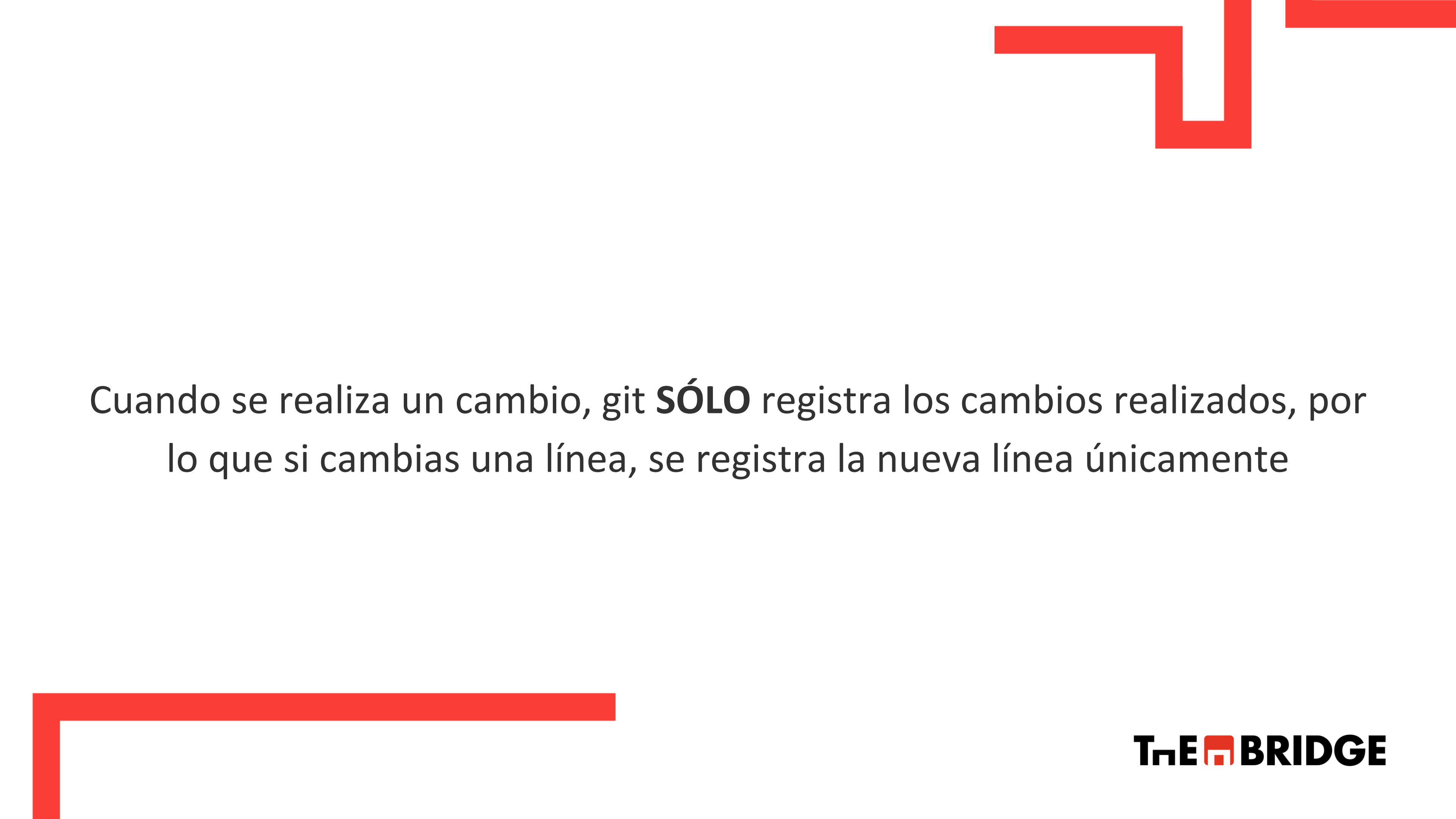


¿Qué es git?

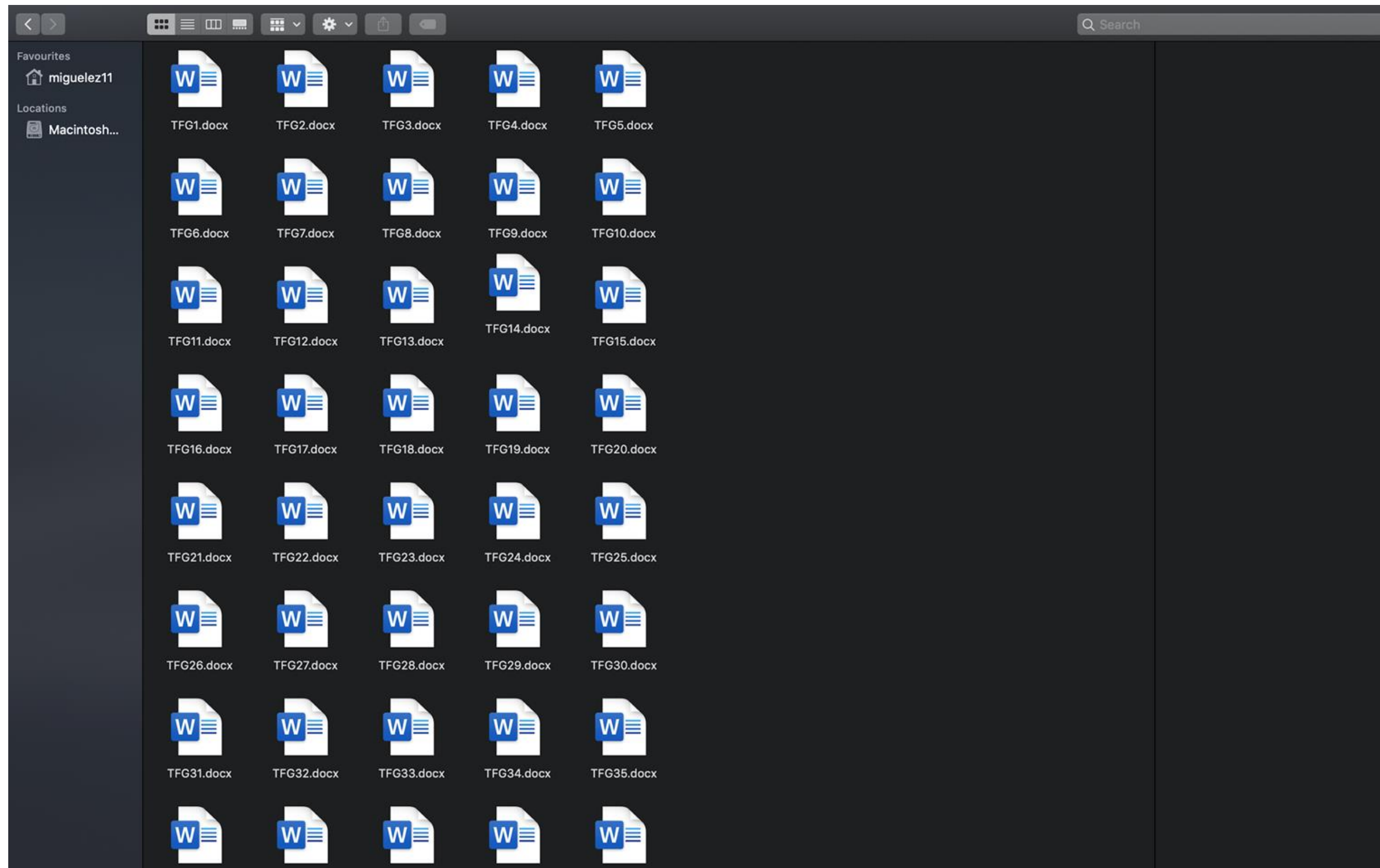




Git es un sistema de control de versiones/código que permite guardar y mantener un proyecto de código desarrollado por una o varias personas.



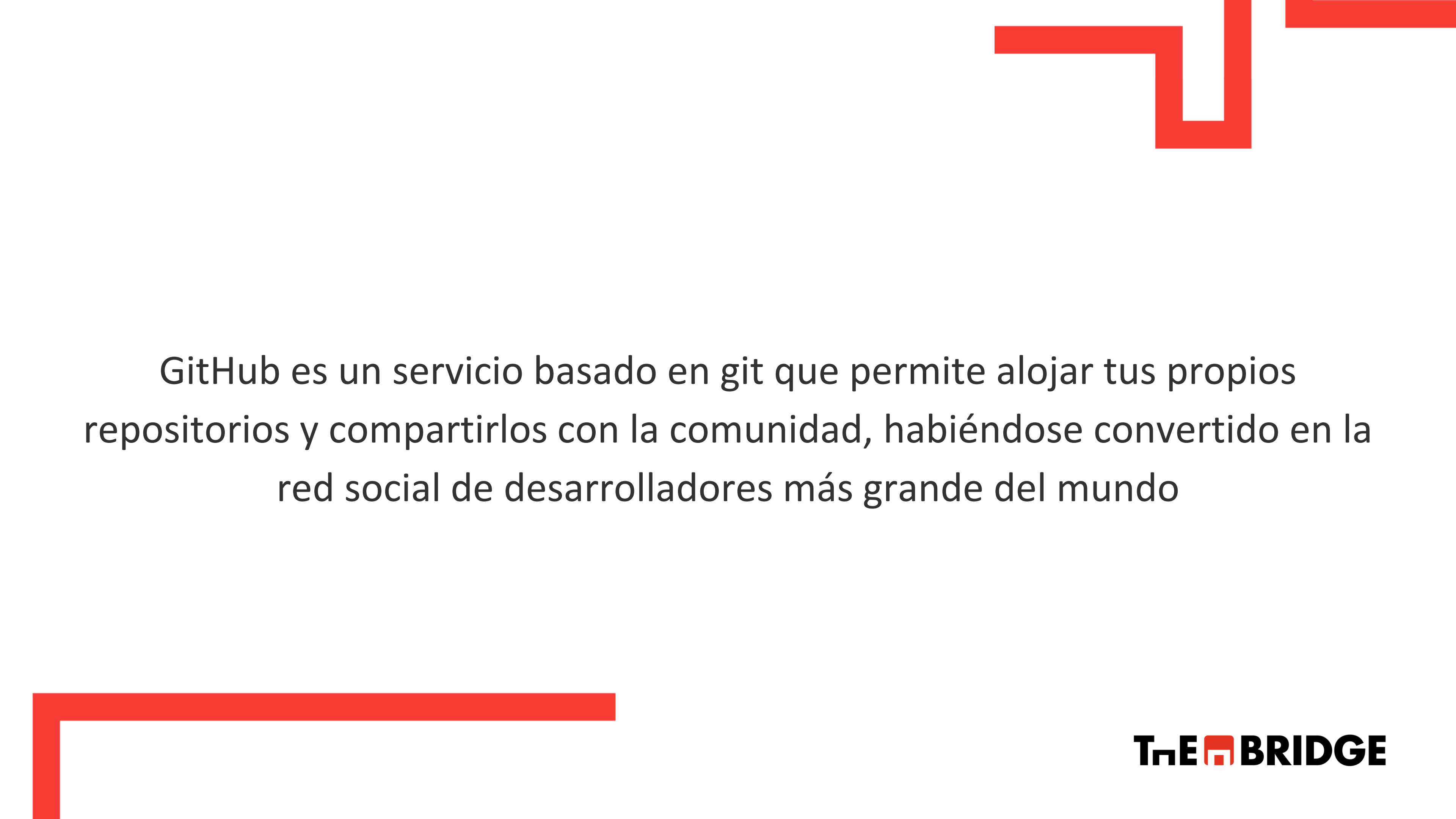
Cuando se realiza un cambio, git **SÓLO** registra los cambios realizados, por lo que si cambias una línea, se registra la nueva línea únicamente



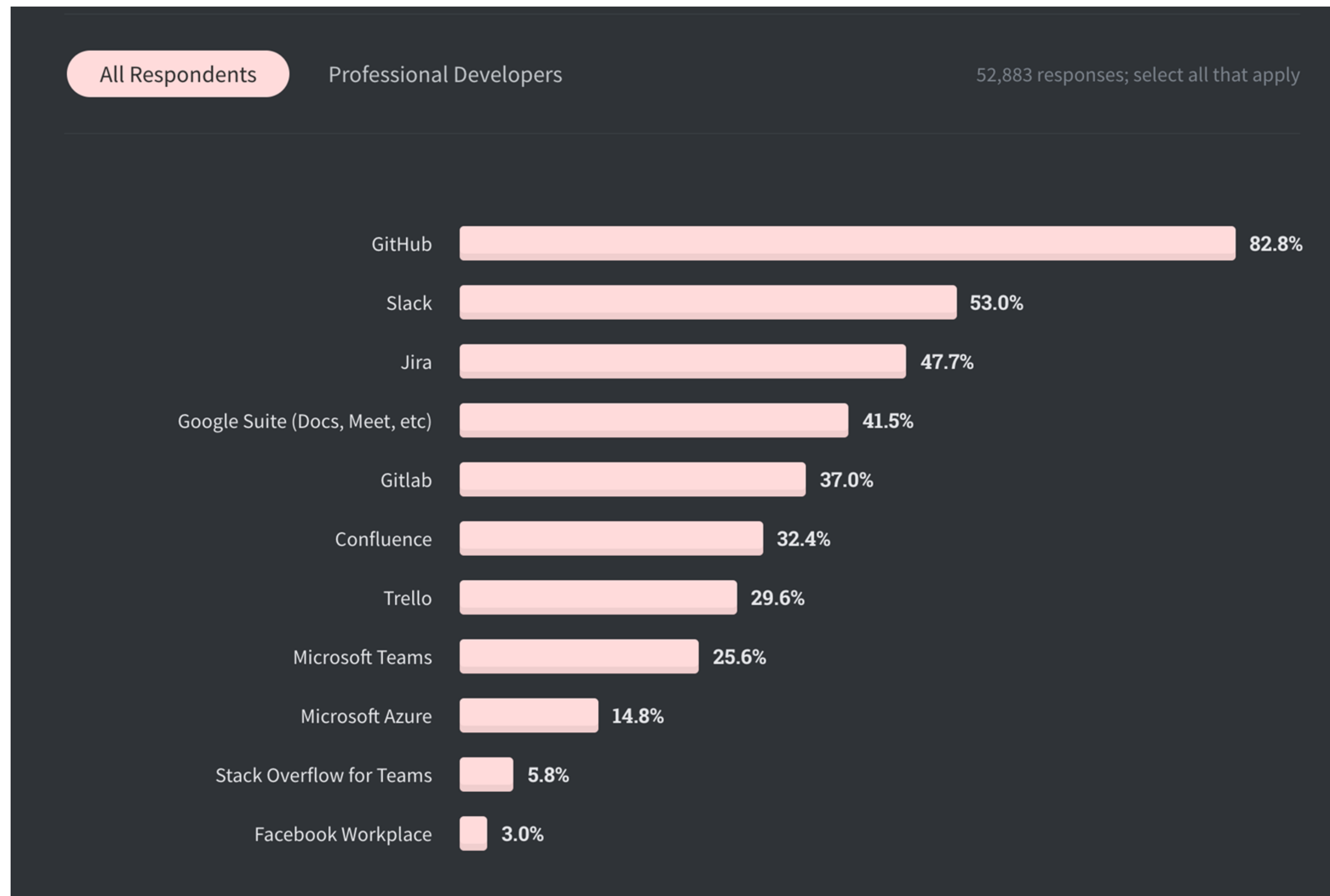
Ejemplo de un proyecto con muchas versiones del mismo

¿Qué es GitHub?





GitHub es un servicio basado en git que permite alojar tus propios repositorios y compartirlos con la comunidad, habiéndose convertido en la red social de desarrolladores más grande del mundo



Encuesta Stack Overflow 2020 - Herramientas de colaboración

Crear una cuenta en GitHub



<https://github.com/>

Join GitHub

Create your account

Username *

rafazamb



Email address *

rafael@thebridgeschool.es



Password *

.....



Make sure it's at least 15 characters OR at least 8 characters including a number and a lowercase letter.

[Learn more.](#)



Please verify your email address

Before you can contribute on GitHub, we need you to verify your email address.

An email containing verification instructions was sent to rafael@thebridgeschool.es.

[Resend verification email](#)

[Change your email settings](#)



GitHub Desktop



GitHub Desktop

<https://desktop.github.com/>



GitHub Desktop

Welcome to GitHub Desktop

GitHub Desktop is a seamless way to contribute to projects on GitHub and GitHub Enterprise. Sign in below to get started with your existing projects.

New to GitHub? [Create your free account.](#)

Sign in to GitHub.com

Sign in to GitHub Enterprise

Skip this step

Authorize GitHub Desktop



GitHub Desktop by desktop

wants to access your rafazamb account



Repositories

Public and private



Personal user data

Full access



Workflow

Update GitHub Action Workflow files.



Cancel

Authorize desktop

GitHub Desktop

Configure Git

This is used to identify the commits you create. Anyone will be able to see this information if you publish commits.

Name

rafazamb

Email

77900746+rafazamb@users.noreply.github.com

Continue

Cancel

Example commit

Fix all the things

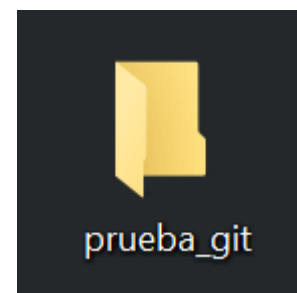
 rafazamb • 30m

Crear un repositorio



U Crear un repositorio

1. Elige una ruta de tu ordenador donde vayas a crear tu primer repositorio. En mi caso: “C:\Users\rzambrano\Desktop\prueba_git” (Carpeta vacía)



2. Desde GitHub Desktop vamos a *File > New Repository*
Le ponemos un nombre al repositorio y en el Local Path, la ruta que hemos elegido

NOTA: Si quieres crear un repositorio a partir de una ruta donde ya tienes archivos, en vez de *File > New Repository*, tendrías que ir a *File > Add local repository*

Create a new repository

Name

repo_taller

Description

Local path

C:\Users\rzambrano\Desktop\prueba_git

Choose...

☐ Initialize this repository with a README

Git ignore

None

License

None

Create repository

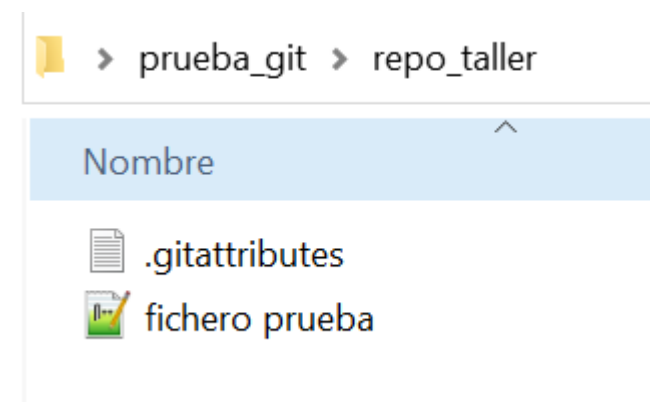
Cancel

**Guarda versiones en tu
repositorio**



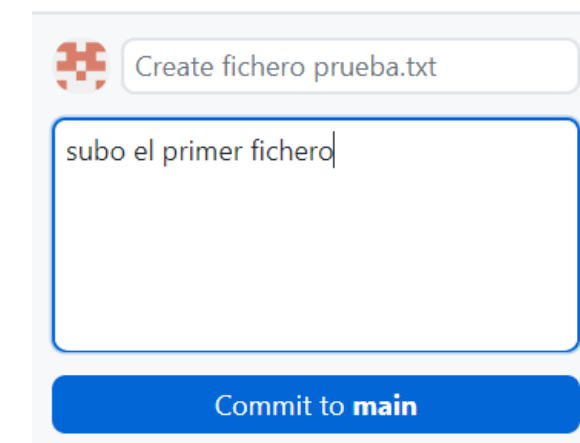
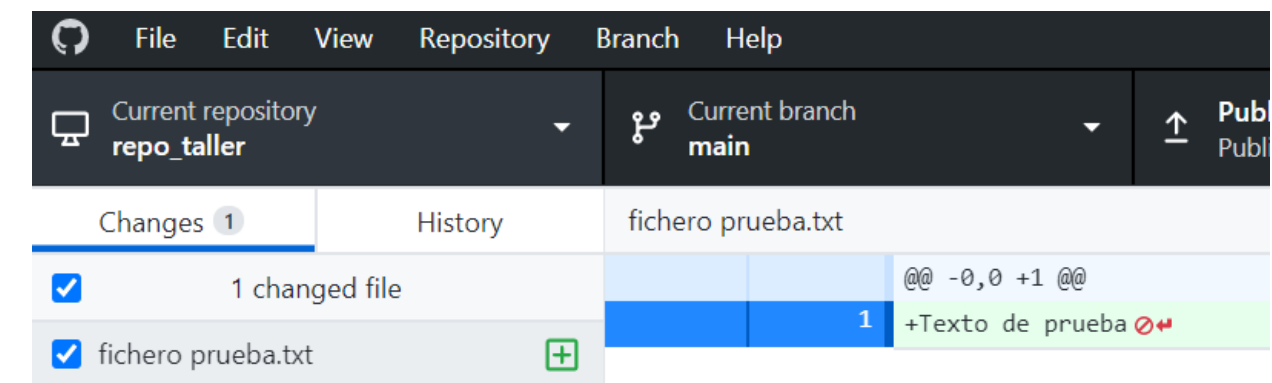
Guarda versiones en tu repositorio

Añade un fichero de texto cualquiera a tu repositorio



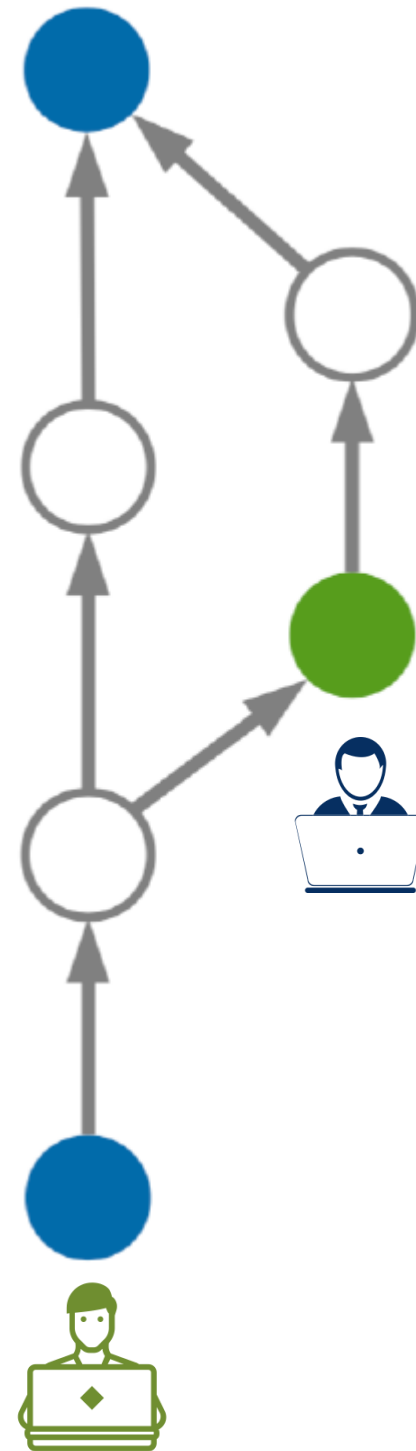
Ahora ve a GitHub Desktop y verás que te dice que hay cambios en un archivo. Introduce una descripción y realiza el primer **commit**

NOTA: Git tiene la opción de dividir el desarrollo en varias ramas (o *branches*). La principal se llama main, y durante este taller únicamente trabajaremos con esta rama principal.



Guarda versiones en tu repositorio

Git Branches



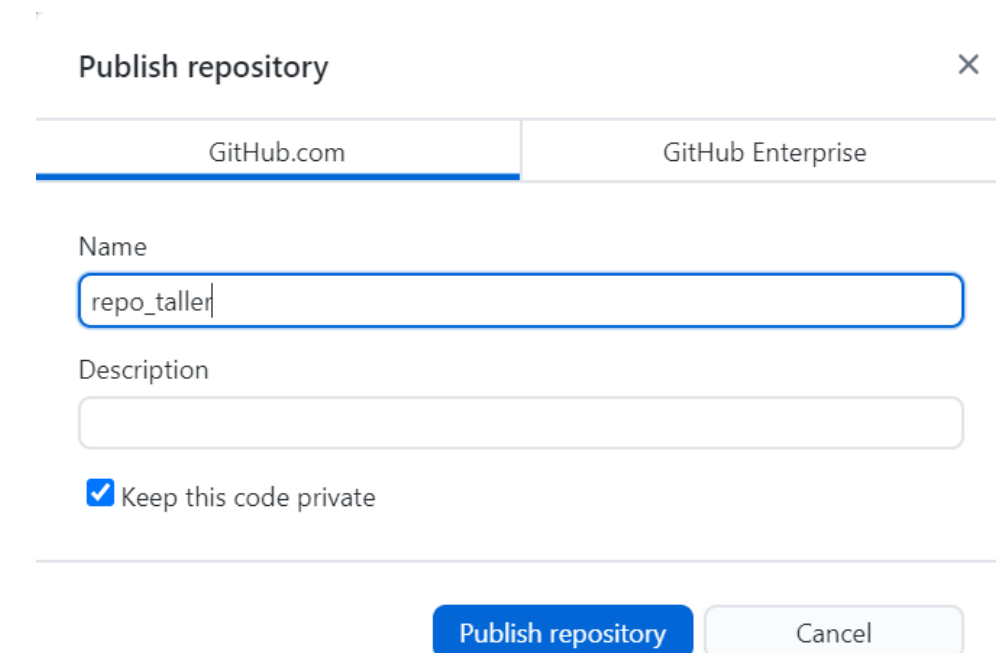
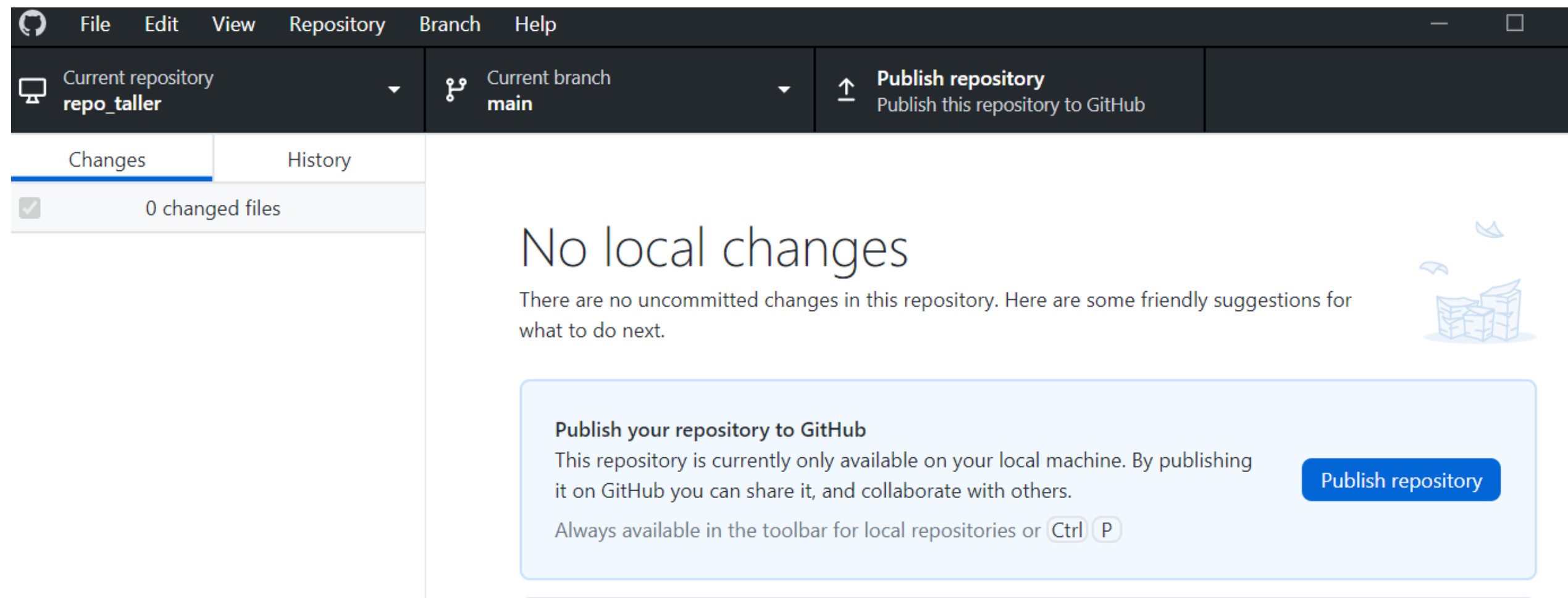
Publicar tu repositorio en GitHub



Publicar tu repositorio en GitHub

Ya tenemos una primera versión de nuestro repositorio **en local**

Vamos a trasladar esta lógica de versiones a GitHub (la “nube”)



Accede a [GitHub](https://github.com) y comprueba que ya tienes tu primer repositorio


Publicar tu repositorio en GitHub



← → ↻ github.com/rafazamb/repo_taller

🔒 rafazamb / repo_taller Private Unwatch

[Code](#) [Issues](#) [Pull requests](#) [Actions](#) [Projects](#) [Security](#) [Insights](#) [Settings](#)

[main](#) 1 branch 0 tags [Go to file](#) [Add file](#) [Code](#)

 rafazamb Create fichero prueba.txt [...](#) 7b0c7ab 12 minutes ago ⌚ 2 commits

 .gitattributes	Initial commit	24 minutes ago
 fichero prueba.txt	Create fichero prueba.txt	12 minutes ago

Add a README with an overview of your project. [Add a README](#)

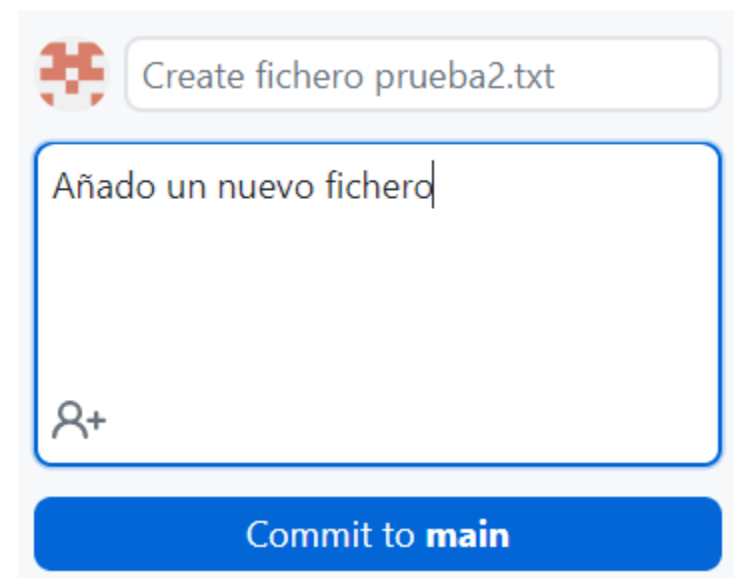
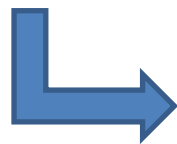
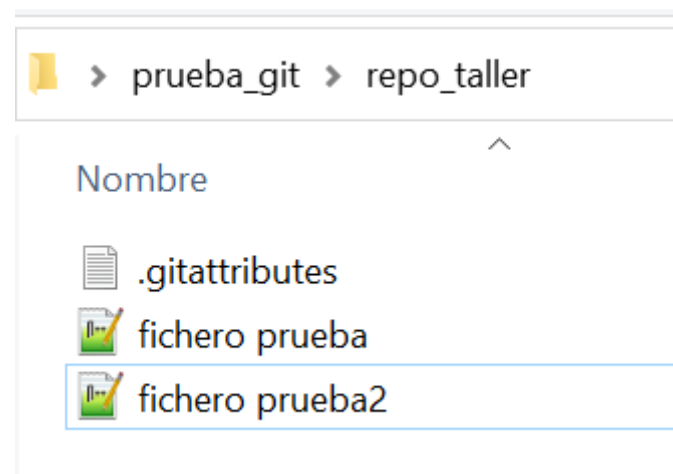
Incorporar nuevas versiones



Incorporar nuevas versiones

Añade otro archivo o modifica el actual y haz otro *commit*

Ahora ya no publicaremos el repositorio, sino que subiremos los cambios (*push*)



Push commits to the origin remote

You have 1 local commit waiting to be pushed to GitHub.


Always available in the toolbar when there are local commits waiting to be pushed or


Ctrl P


Push origin

Incorporar nuevas versiones

En GitHub, dentro commits podemos ver el histórico de versiones


 main ▾


 1 branch


 0 tags




Go to file

Add file ▾

 Code ▾

 rafazamb Create fichero prueba2.txt ...


0cf6f61 2 minutes ago  3 commits

 .gitattributes	Initial commit	33 minutes ago
 fichero prueba.txt	Create fichero prueba.txt	22 minutes ago
 fichero prueba2.txt	Create fichero prueba2.txt	2 minutes ago


Create fichero prueba2.txt ...

 rafazamb committed 3 minutes ago

Create fichero prueba.txt ...

 rafazamb committed 23 minutes ago

Initial commit

 rafazamb committed 34 minutes ago

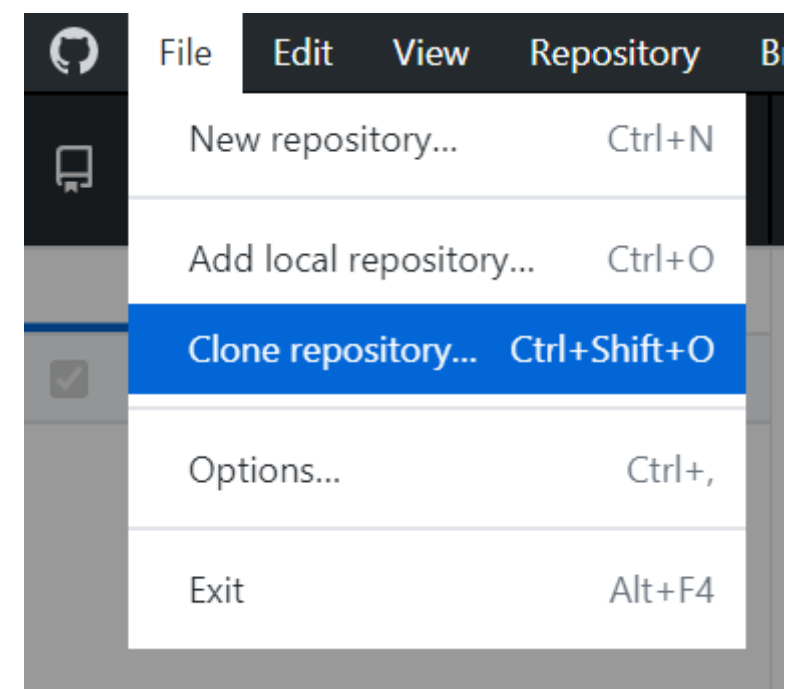
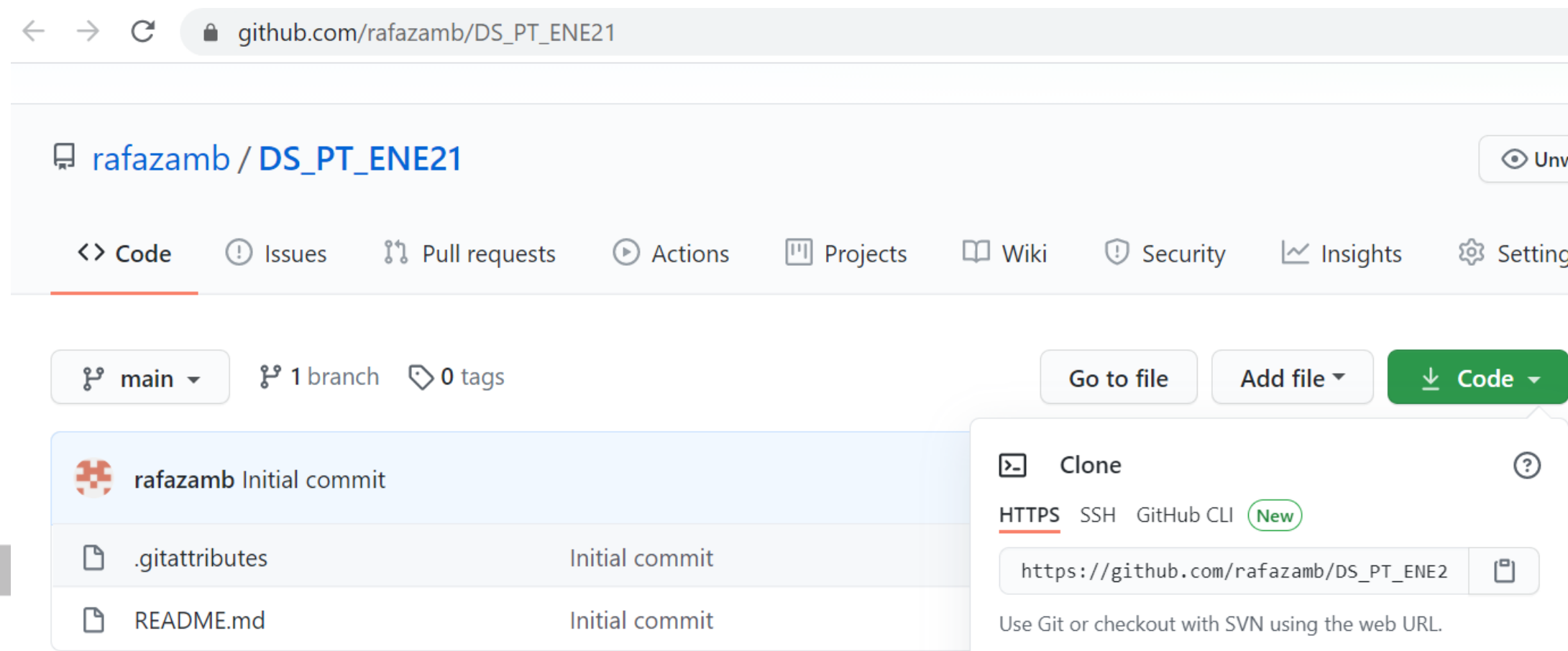
Clonar un repositorio



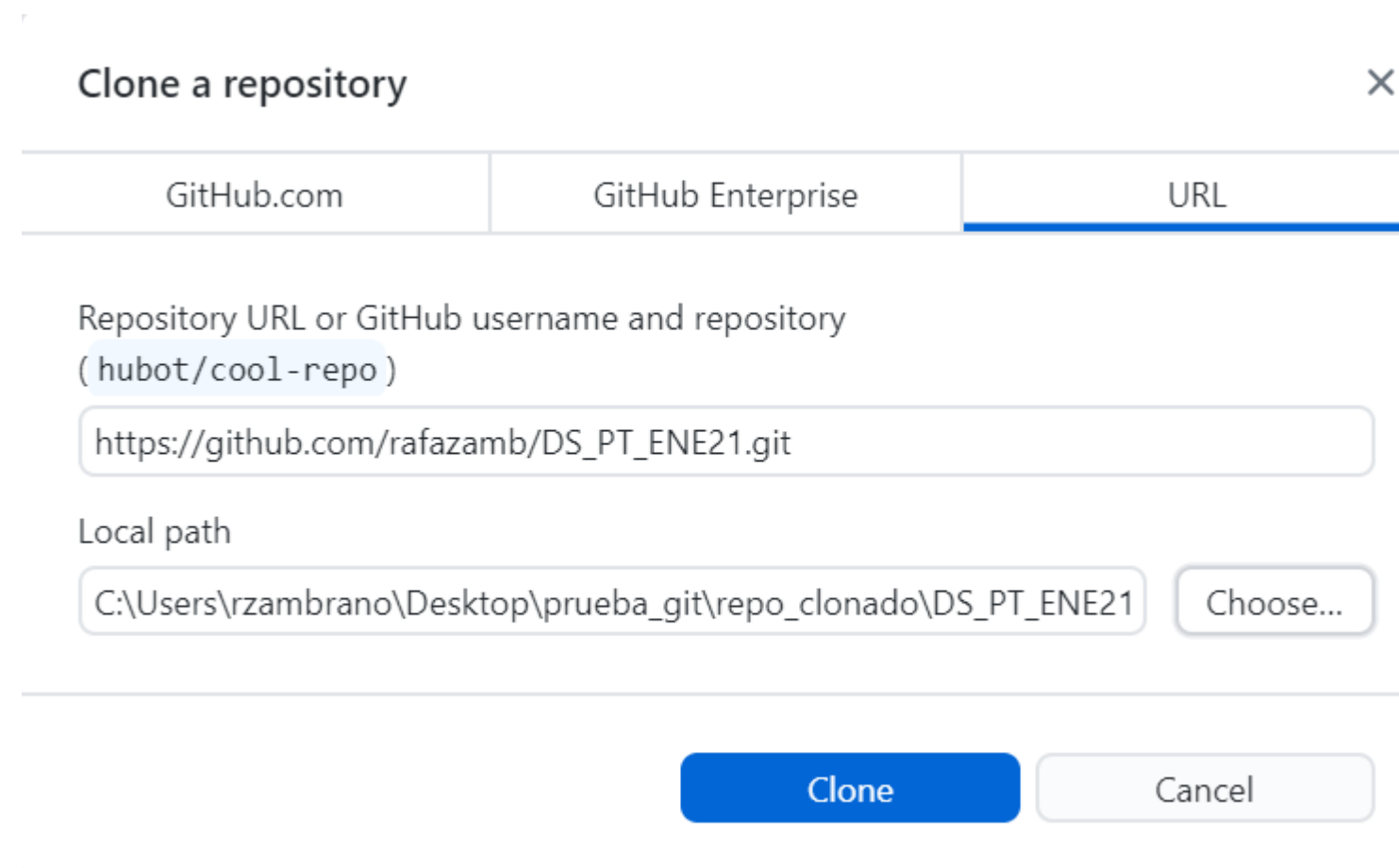
Clonar un repositorio

Para clonar un repositorio, necesitamos:

- 1) Ruta del ordenador donde queramos clonar el repositorio
- 2) Tener acceso al repositorio
- 3) Link de GitHub del repositorio



Clonar un repositorio



Clone a repository

GitHub.com GitHub Enterprise URL

Repository URL or GitHub username and repository
(hubot/cool-repo)

https://github.com/rafazamb/DS_PT_ENE21.git

Local path

C:\Users\rzambrano\Desktop\prueba_git\repo_clonado\DS_PT_ENE21 Choose...

Clone Cancel

De esta forma tendremos un nuevo repositorio local con todos los archivos del curso.

IMPORTANTE. Se recomienda ir actualizado esta carpeta para bajarse todos los materiales del curso, pero **NO** usarla para guardar tu propio material, ya que vas a tener conflictos en las versiones de los archivos. Lo recomendable es que desarrolles tus propios notebooks en otra ruta y que dejes esta únicamente para que se actualice con los últimos cambios que haya en el repositorio de los profesores.

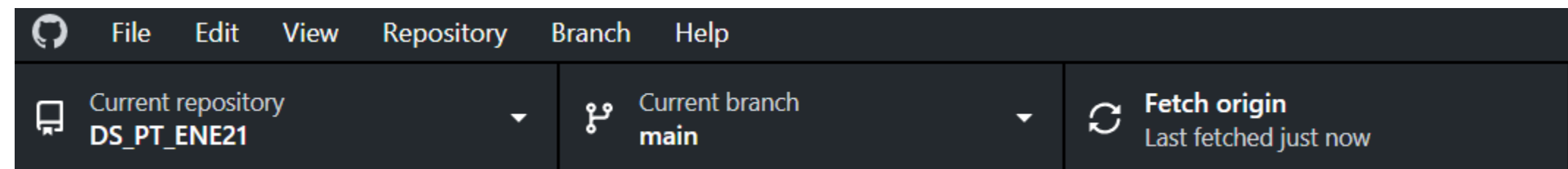
**Descargar los cambios del
repositorio remoto**



U Descargar cambios del repositorio remoto

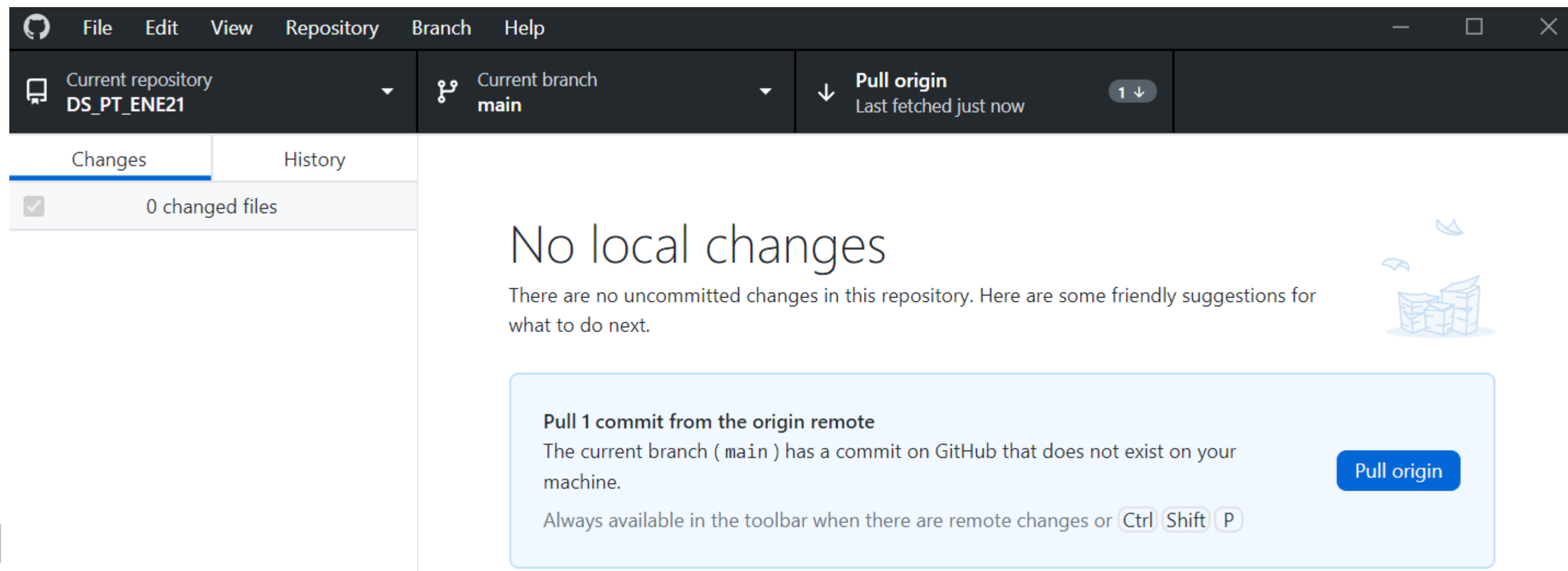
Solo nos queda por ver cómo se actualiza el repositorio en nuestro ordenador con los últimos cambios.

¿Cómo sé si hay cambios? Desde GitHub Desktop haz clic en *Fetch origin* para comprobar si hay una nueva versión

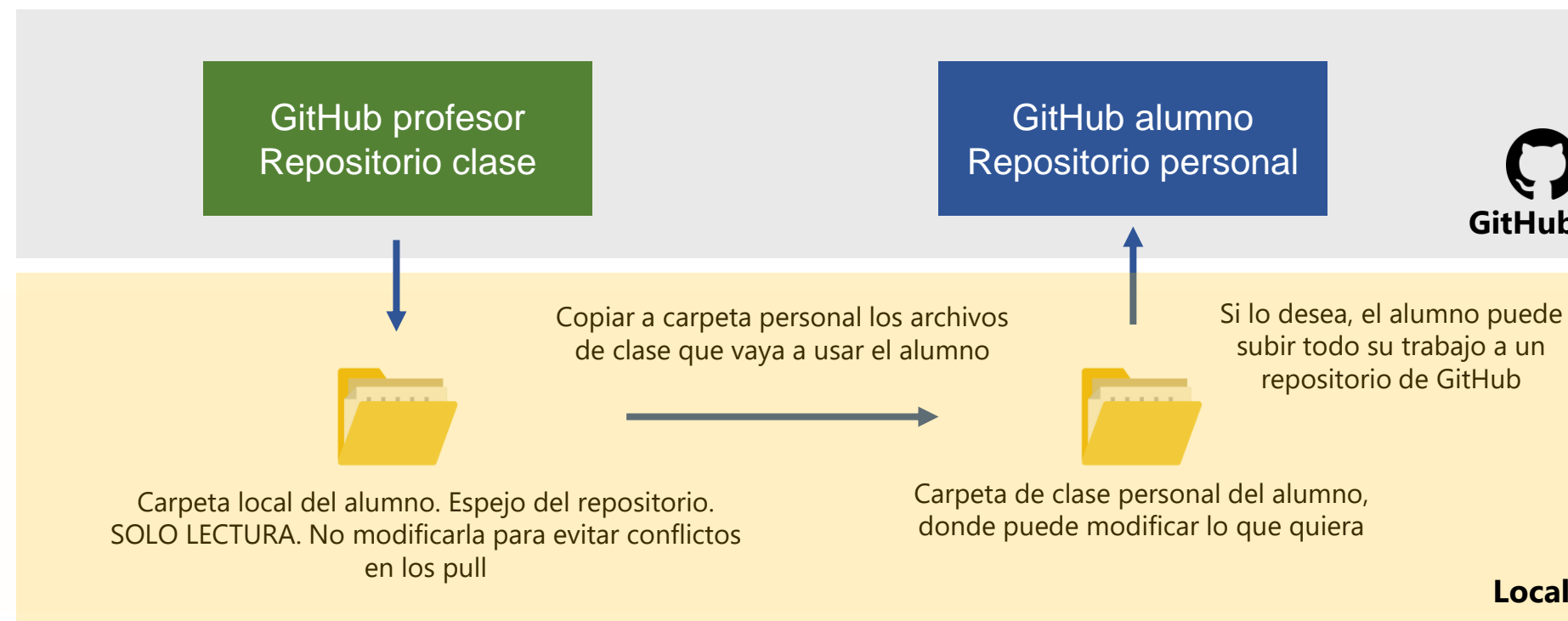
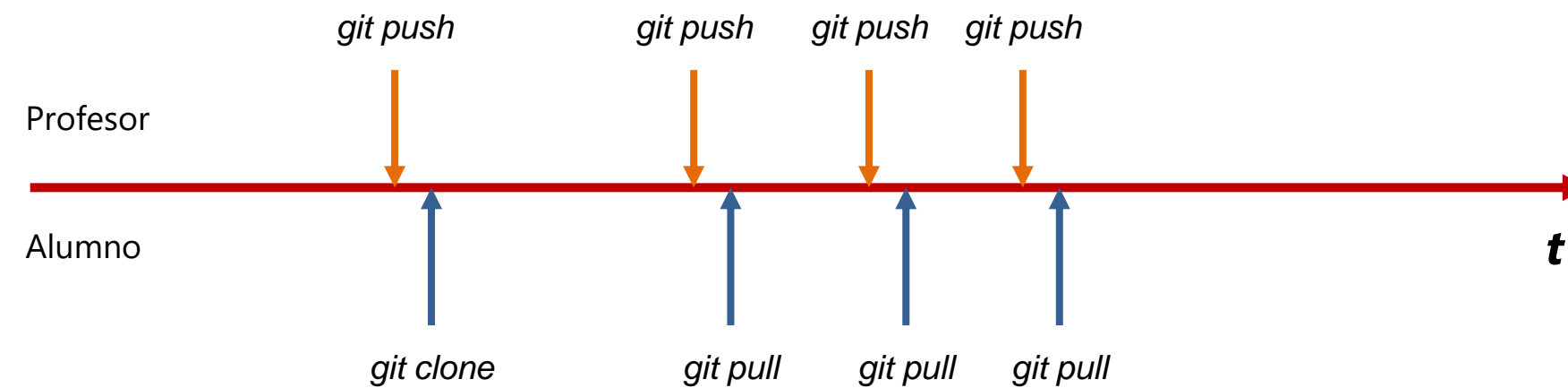


U Descargar cambios del repositorio remoto

En caso de existir cambios, nos sugerirá que hagamos un *pull* para descargarlos. Al hacer clic en *pull origin* el repositorio estará actualizado

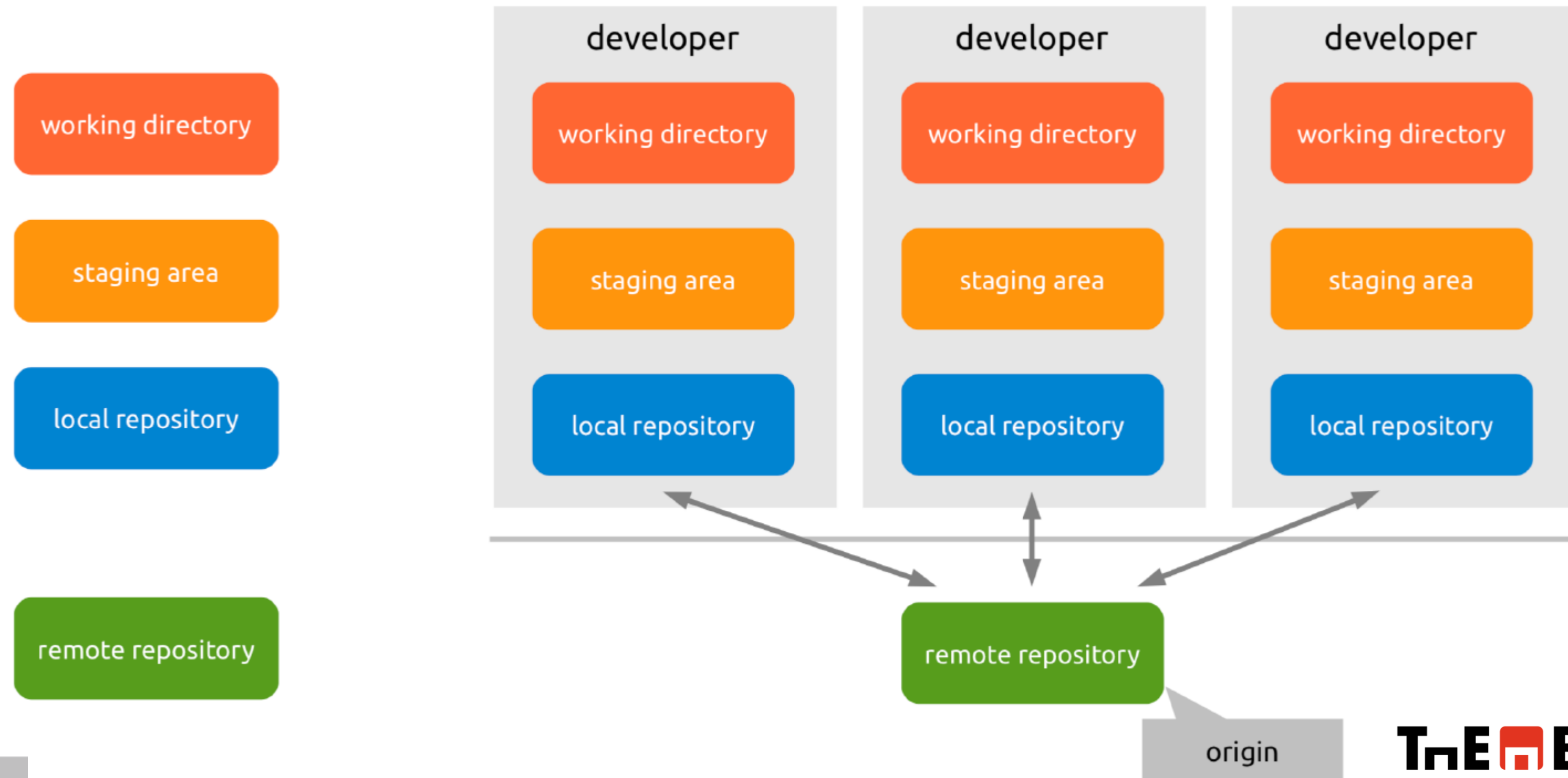


U Git en clase



U Secciones Git

Un proyecto GIT consta de tres secciones principales:



Funcionalidades básicas de Git

- ***git init***: Crear un repositorio
- ***git commit***: Guarda diferentes versiones de tu repositorio
- ***git push***: Versiona de nuevo y sube los cambios
- ***git clone***: Clonar un repositorio
- ***git pull***: Descarga los cambios del repositorio remoto



1. git commit



2. git push



3. git out!

Uso de git en la terminal de comandos

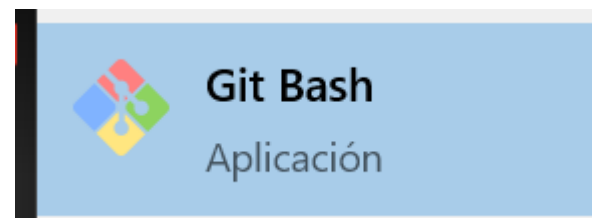


U Descargar e instalar Git

Git también puede utilizarse mediante línea de comandos en una terminal

Instalación de Git: <https://git-scm.com/>

Windows: Una vez instalado, abrir la aplicación “Git Bash” para acceder a la terminal



Mac: Comprueba si ya lo tienes instalado desde la Terminal, escribiendo **git --version**

Comandos básicos git



git config



git config: Nos permite cambiar la configuración de git, tanto global como de un repositorio



git config

Configuración de usuario

git config --global user.name [NOMBRE DE USUARIO]

git config --global user.email [EMAIL]

git config --global credential.helper store

git clone [url repo]



git clone [url repo]: Crea una copia del repositorio en nuestro ordenador

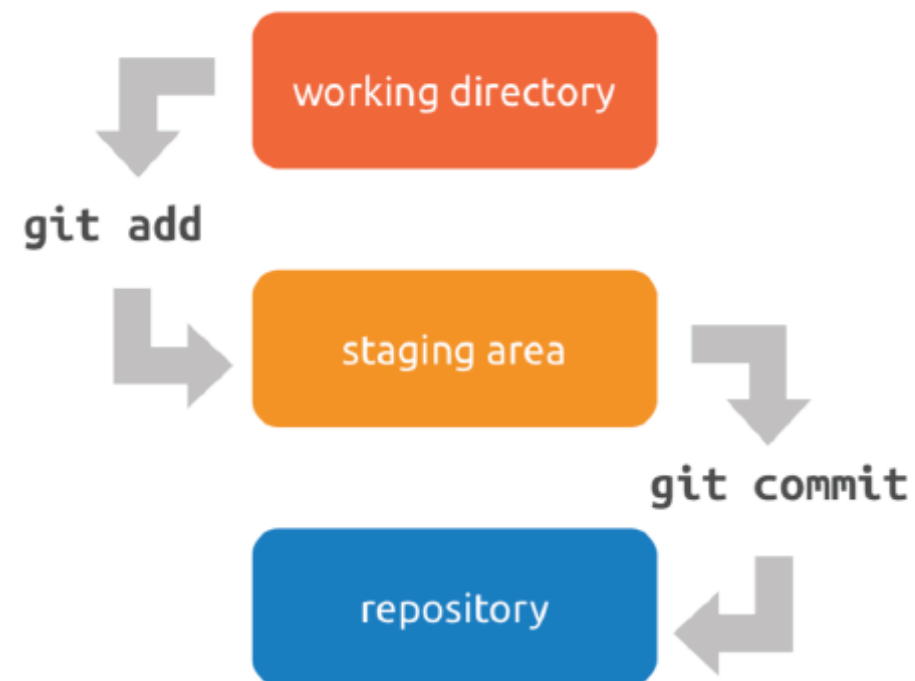


git status

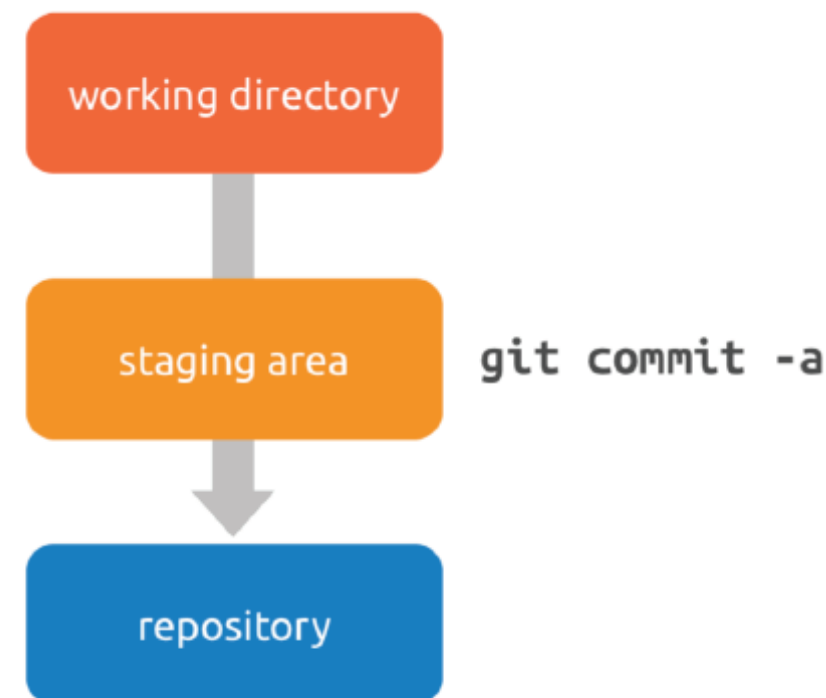
git status: Sirve para ver el estado actual de nuestro repositorio

git add [archivo]

git add [archivo]: Añade los cambios realizados al staging area



Al incluir un fichero en el directorio o cambiarlo, Git nos advertirá, pero tendremos que realizar las acciones de “add” y “commit” para incluirlo



También puede no hacerse el paso intermedio

`git commit -m “descripción cambios”`

git commit: Crea una nueva versión con los cambios en el staging area

git log

git log: Muestra el histórico de versiones

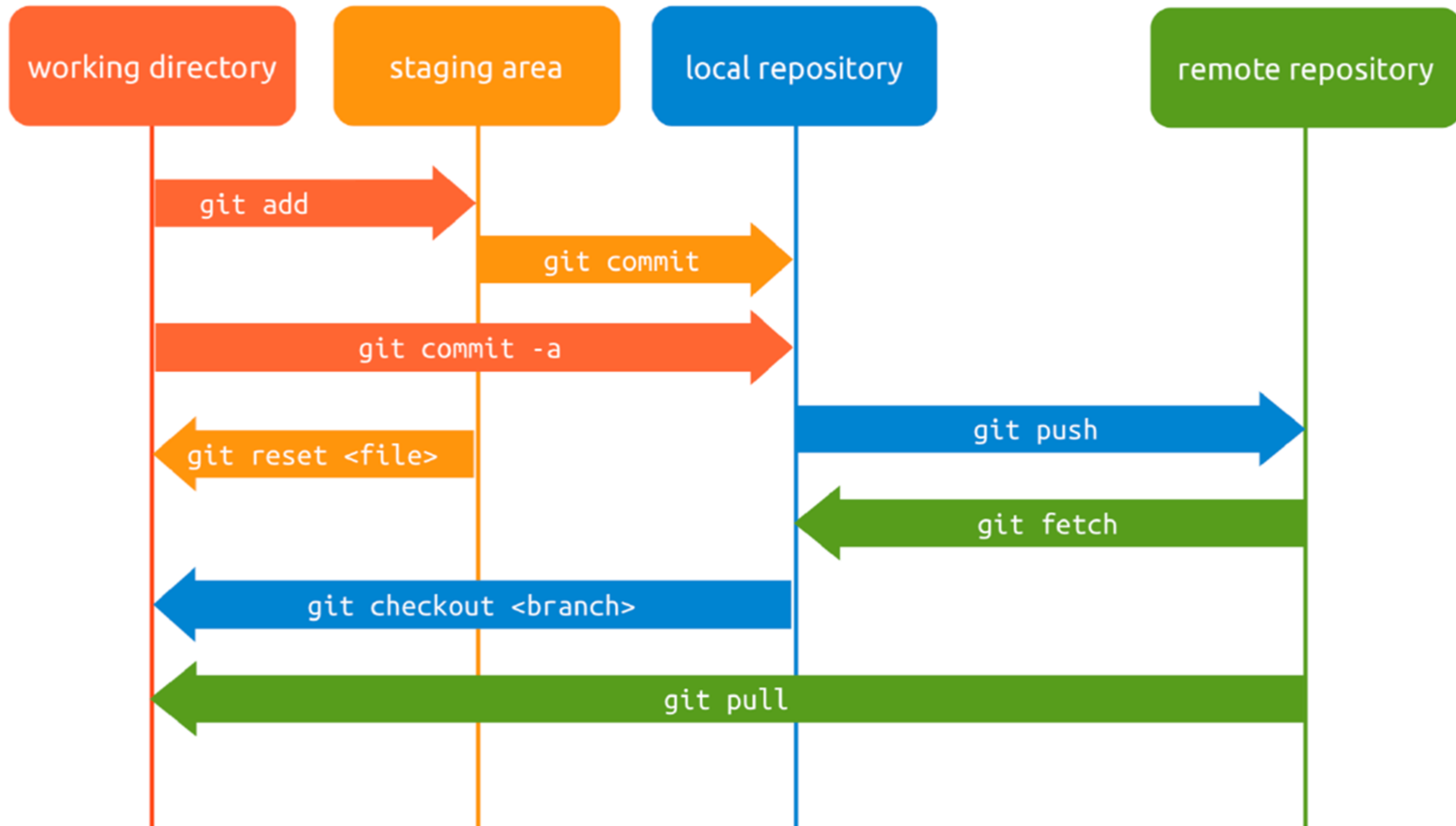
git pull

git pull: Descarga todos los commits del servidor

git push

git push: Sube todos los commits al servidor

Esquema de funcionamiento



LA PEQUEÑA INFOGRAFÍA DE GIT!

¡A Dios pongo por testigo, que nunca más tendré 10 copias del mismo trabajo!



GIT CLONE

Descargar el proyecto a nuestro ordenador.



GIT STATUS

¿Qué está pasando?



GIT ADD

"Empaqueta" los cambios o archivos nuevos.



GIT COMMIT

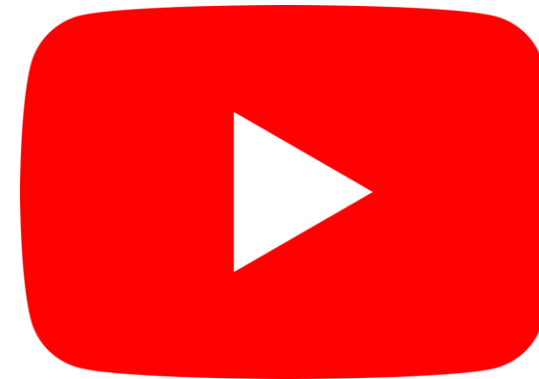
"Etiqueta" el paquete donde hemos ido metiendo los cambios y archivos nuevos.



GIT PUSH ORIGIN MASTER

Envía todos los paquetes al ciberespacio.

Taller de introducción a GIT



<https://youtu.be/Mkd8idNUoIM>

Practicar con GIT

https://learngitbranching.js.org/?locale=es_ES

Gracias

Rafa Zambrano

rafael@thebridgeschool.es