



Git y Github



Git

Git

Cuando escribimos código muchas veces es una tarea que desarrollamos partes, etapas o con otras personas y suelen haber cambios/correcciones que se realizan.

GIT es un sistema de control de versiones que nos ayuda a manejar esos cambios.

Git

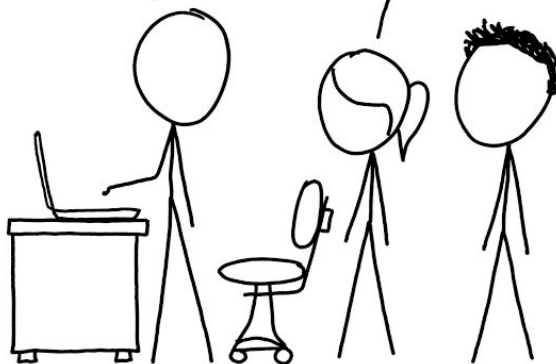
Para hacernos una idea:

Por ejemplo cuando hacemos una monografía o documento, no pasa que nos quedamos con muchas versiones del archivo según lo vamos modificando?

THIS IS GIT. IT TRACKS COLLABORATIVE WORK
ON PROJECTS THROUGH A BEAUTIFUL
DISTRIBUTED GRAPH THEORY TREE MODEL.

COOL. HOW DO WE USE IT?

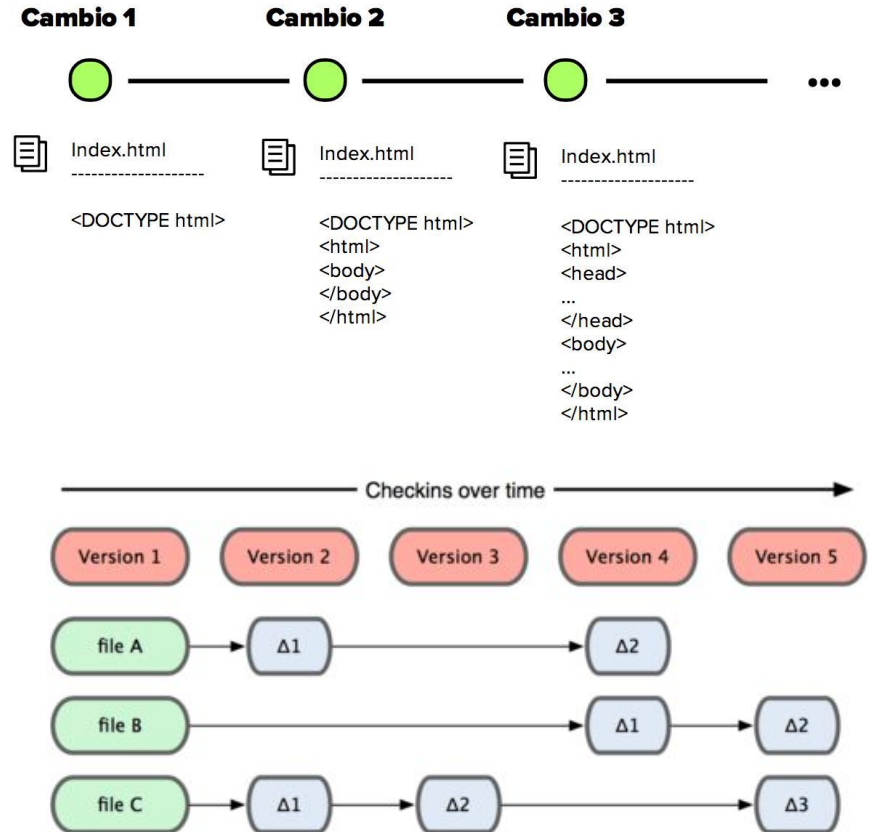
NO IDEA. JUST MEMORIZE THESE SHELL
COMMANDS AND TYPE THEM TO SYNC UP.
IF YOU GET ERRORS, SAVE YOUR WORK
ELSEWHERE, DELETE THE PROJECT,
AND DOWNLOAD A FRESH COPY.



Git

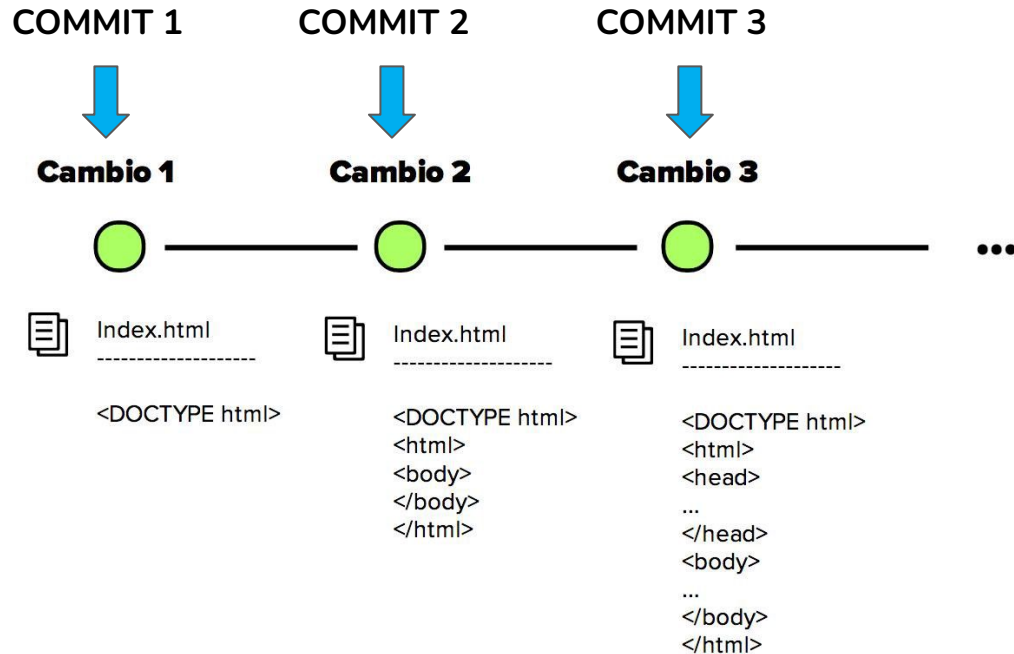
Para hacernos una idea:

Utilizando GIT en vez de crear varios archivos para cada cambio, mantendremos un solo archivo con todos los cambios y podremos <- retroceder / avanzar -> a través de esos cambios, juntarlos, crear una línea donde se guarden solo unos cambios en específico (ramas) haciendo que no nos llenemos de archivos y sea más sencillo manejarlo.



Commit

Git trabaja con commits, un commit es un punto en el tiempo con correcciones o nuevos cambios agregados a nuestro archivo(s).



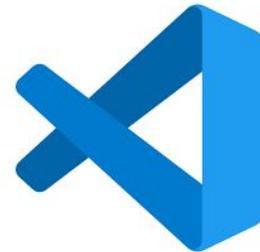
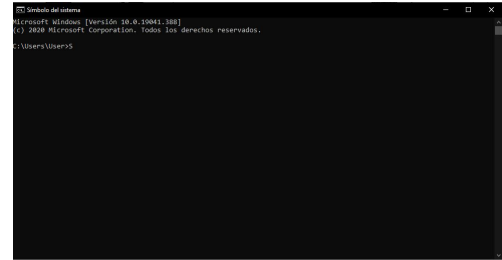
Que vamos a requerir para usar Git?

Instalar Git: lo podemos instalar desde:

<https://git-scm.com/>

Una terminal: en Windows podemos utilizar el símbolo del sistema.

Un editor de texto: nos sirve desde notepad, hasta VSCode, ya que git trabaja con texto plano (. txt , sin formato), preferentemente usaremos VSCode.



Configurar Git

Git necesita saber quienes somos antes de empezar a trabajar tenemos que indicar nuestro correo y contraseña

```
git config --list
```

```
git config --global user.name "nombre"
```

```
git config --global user.email "mail"
```

Run

```
git config --global user.email "you@example.com"  
git config --global user.name "Your Name"
```

to set your account's default identity.
Omit --global to set the identity only in this repository.

fatal: no email was given and auto-detection is disabled

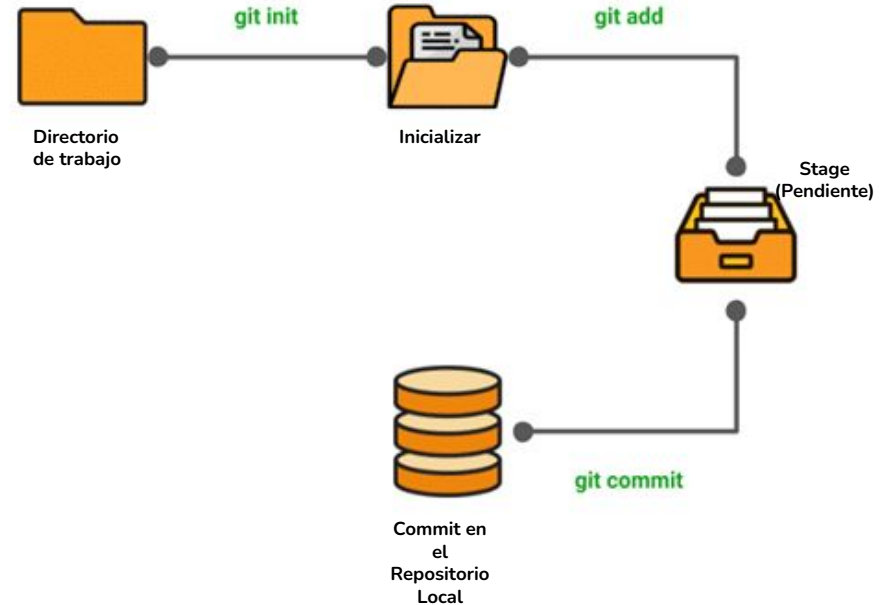
error An unexpected error occurred: "Command failed.
Exit code: 1
Command: sh
Arguments: -c react-scripts build && gh-pages -d build
Directory: /Users/panda/dev/react/straydrop.github.io
Output:
".

Ciclo de vida Git

Por eso Git maneja un ciclo de vida para los cambios:

1. Proyecto no Inicializado
2. Proyecto inicializado.
3. Se añade archivo(s) al Stage(RAM).
4. Se hace un commit (un punto de tiempo) al archivo(s)
5. ???????

Vamos a ver cómo se realiza este proceso:



Ejercicio

Resumen

Crear una carpeta aparte y en esa carpeta crear un archivo Biografia.txt, agregar una biografía, inicializar git, registrar 3 commit cada uno con sus cambios, verificando el estado en cada paso y regresar a un commit anterior. - 10 min.

git init - inicializar

git status - verificar el estado del repositorio

git add <archivo> - agregar al stage

git commit -m "mensaje - crear commit

git log - verificar commits

git config

git config --list

git config --global user.<name - email>

git show <archivo> - cambios en especifico

git diff <token1> <token2> - comparar cambios entre commits

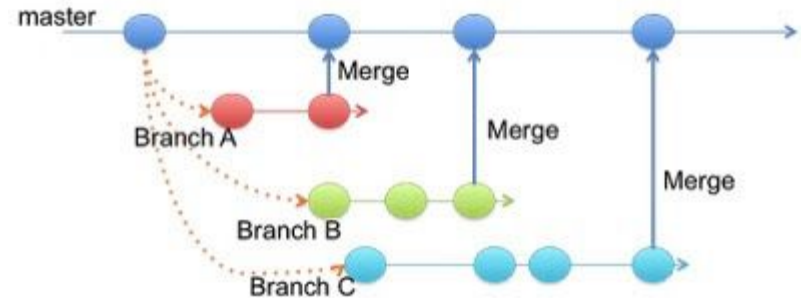
git reset <token> <--hard --soft> - regresar a commit anterior

Ramas



Ramas

En git tenemos lo que son ramas, como dice su nombre si en algún momento deseamos a partir de un commit realizar cambios sin afectar a nuestra rama principal (master), podemos crear ramas adicionales para en algún momento juntarlas, es algo muy común en desarrollo.



Ramas en Git

Para hacernos una idea:

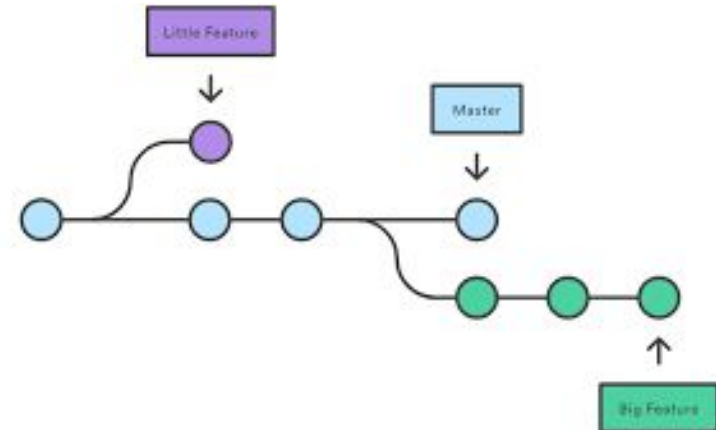
Algo así como esto:

----->



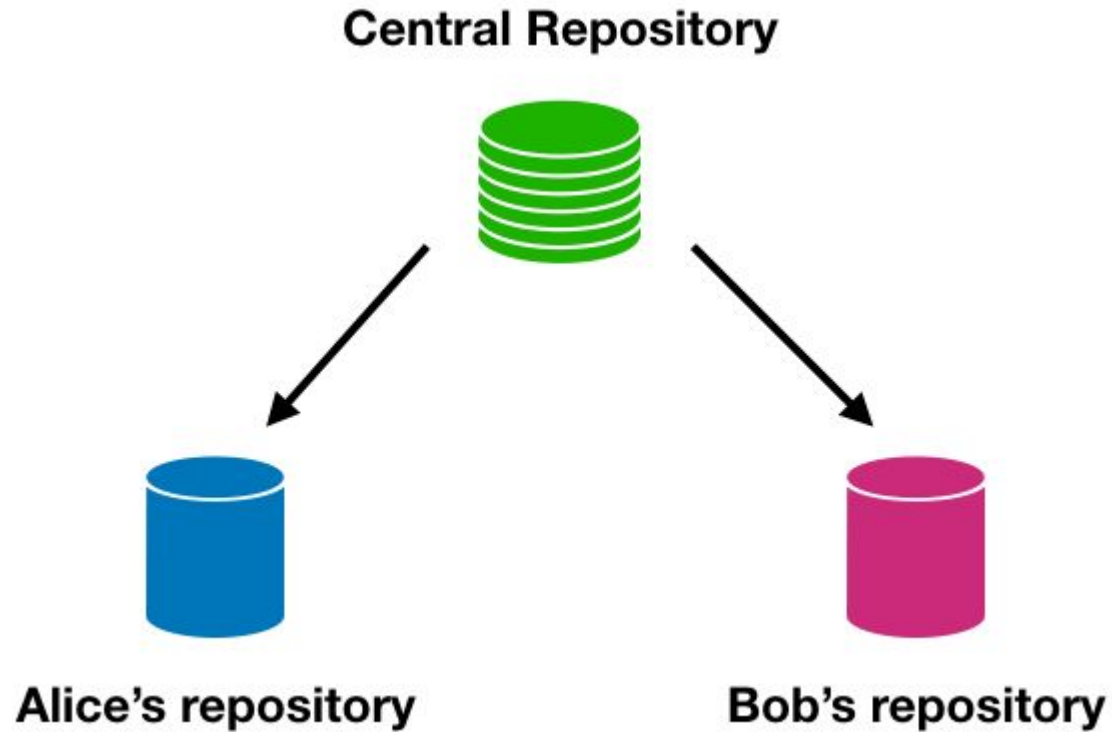
Pero que visto por los momentos que guardamos esos cambios seria algo así:

----->

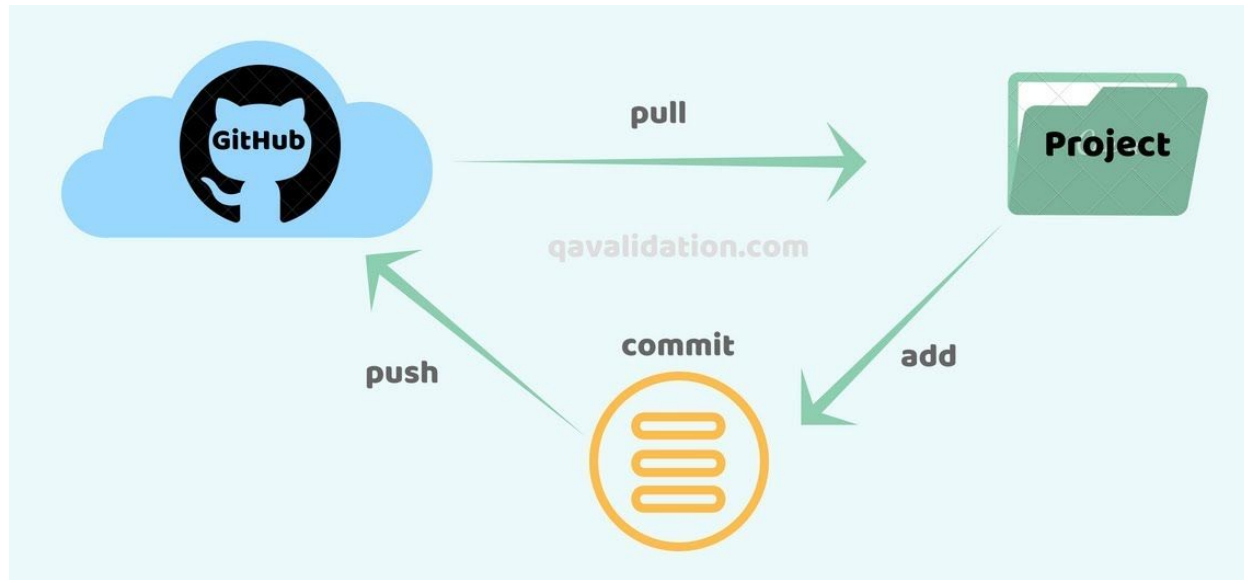


Cómo trabajamos en equipo?

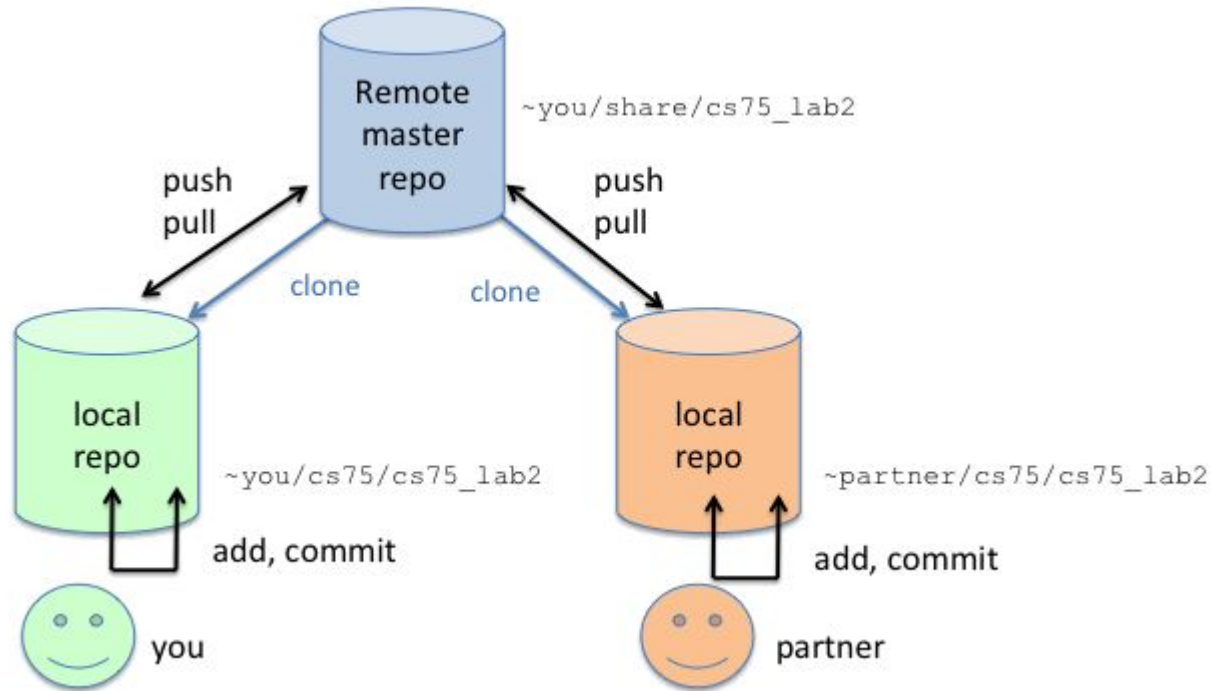
Ciclo de vida con origen remoto



Ciclo de vida con origen remoto



Ciclo de vida con origen remoto

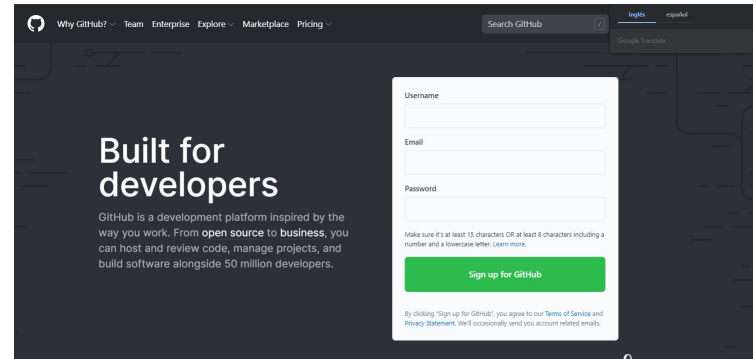


Github

Git es muy útil, pero funciona de forma local (en nuestra pc) así que colaborar es complicado.

Para eso existe Github, github es una plataforma donde podemos subir nuestros proyectos, colaborar, comentar, como en una red social, pero de proyectos hechos con código:

Para eso vamos a tener que crearnos una cuenta en **github.com**

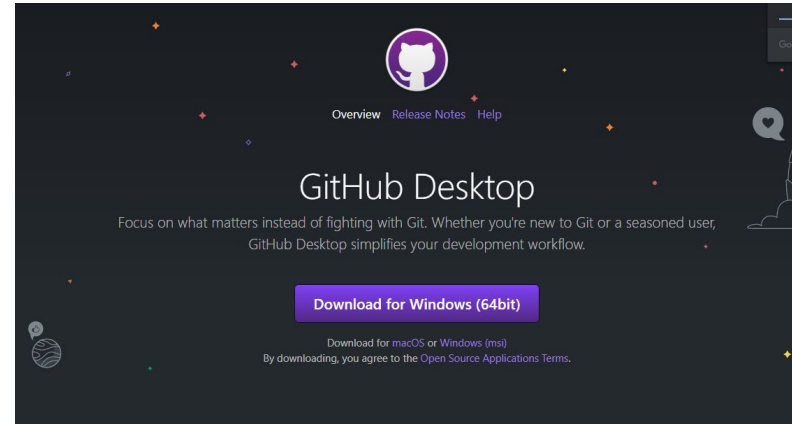


Github Desktop

Github Desktop es un programa que nos permite utilizar git y conectarnos con Github mediante un programa instalado en nuestra PC, muy útil y sencillo si nos liamos un poco con los comandos.

Lo podemos conseguir en:

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



Contenido Extra:

Comandos para empezar con Git:

<https://viviryaprenderweb.com/10-comandos-git-esenciales-para-saber-por-donde-empezar/>

Git Book - Oficial y en Español

<https://git-scm.com/book/es/v2>

Aprender git y ver como es en vivo

<https://learngitbranching.js.org/>

Cheatsheet Git

https://github.github.com/training-kit/downloads/es_ES/github-b-git-cheat-sheet/

