



Notas del video 1

GitHub es un controlador de versiones que nos permite administrar los cambios en nuestros documentos: qué cambiamos y cuándo lo cambiamos.

Términos comunes

- Terminal
- CLI: Command Line Interface
- cd: Change Directory
- Code Editor
- Repository: Proyecto, lugar para guardar los archivos de código.
- Git
- GitHub es un sitio web de repositorios en línea

Git Commands

Clone: Importar un repositorio a tu máquina

add: Seguir archivos y cambios en git

commit: Guardar cambios en Git

push: Subir cambios a un repositorio remoto (GitHub)

pull: Descargar los cambios del repositorio remoto.

Para usar desde GitHub

Podemos crear un repositorio localmente (en la computadora) o directamente desde GitHub.

Un repositorio debe tener un archivo README en el que se describe sobre qué es el proyecto y qué contiene. GitHub muestra el archivo de manera automática en la página principal del repositorio.

Cada commit tiene un id y muestra los cambios específicos realizados en el repositorio.

Para usar localmente

Debemos instalar primero Git.

A pesar de que es posible escribir código en cualquier lado (archivos de texto plano). es útil tener un editor de código.

- **git clone:**

Para "clonar" un repositorio remoto.

Git crea carpetas ocultas.

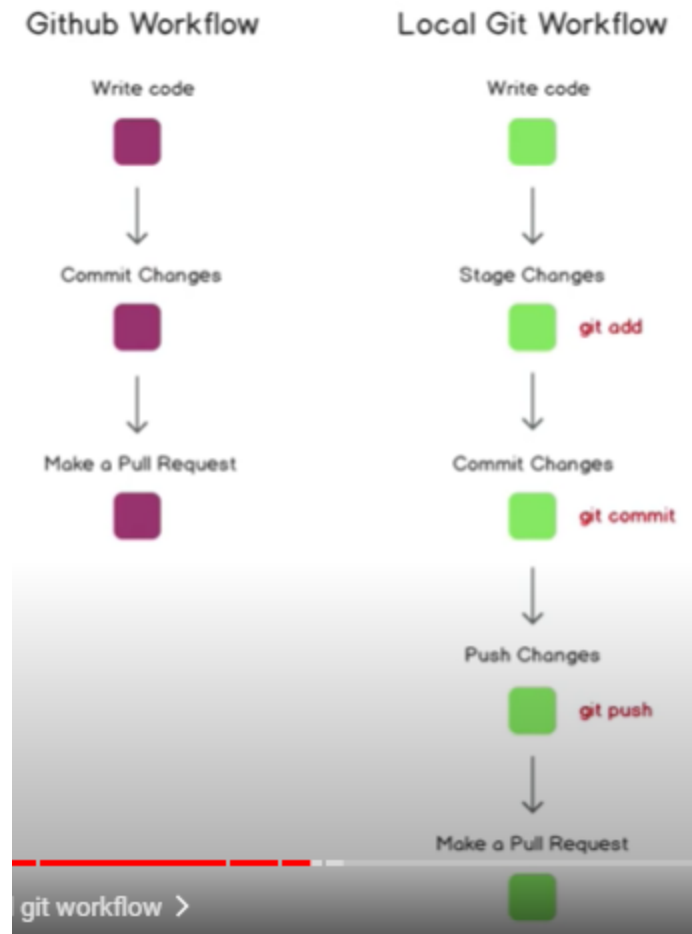
- **git status:** Cambios hechos pero en los que no se ha hecho commit.
- **git add:** Para añadir un archivo que seguir, llevar el registro de cambios. Con git add . añadimos todos los archivos del repositorio.
- **git commit -m:** Commit y mensaje que explique los cambios que estamos registrando. Se puede añadir más de un comentario.
- **git push:** Para subir los cambios al repositorio remoto.

Conectar tu máquina local con GitHub

SSH keys: Para probar que eres quien está trabajando. [min 20:32](#)

- git push origin master
- **git init:** Inicializar un repositorio con una carpeta que ya tenemos creada. Creamos un repositorio vacío en GitHub para conectarlo con el de nuestra computadora.
- **git remote:** Para añadir una referencia al repositorio en línea
git remote add origin link del repositorio vacío que creamos en GitHub.
- **git push -u origin master:** para fijar como default origin master y no tener que escribirlo cada vez.

Git Workflow



Git Branching

Master branch: Rama principal del repositorio. Si solo nosotros estamos trabajando en el repositorio, es esta se guardan todos los cambios.

Al crear una nueva branch, al principio ambas ramas contienen los mismos datos, pero al guardar cambios podemos hacerlos en solo una rama. Es útil cuando creamos nuevas características en nuestro código que estamos probando sin afectar el código que ya teníamos y agregándolas cuando estamos seguros de que funcionan correctamente.

- **git branch:** Muestra las ramas que tenemos y en cuál estamos posicionados.
- **git checkout:** Para cambiar entre ramas o crear una nueva. Para crear una nueva: `git checkout -b nueva-rama`
- **git diff:** Diferencias entre los archivos.

- **git merge nueva rama:** Para mezclar la rama master y la nueva rama.

min 44

Pull request (pr)

Requerimiento para tener tu código en otra rama. Después de hacer un pr podemos actualizar el código haciendo commit. Una vez hecho el pr regresamos a la rama master y se borra la rama de la que hicimos el pr.

Podemos abrir un pr en GitHub y comparar la información de la rama principal y la rama de la que haremos el pr. Se pueden agregar comentarios en cada línea en la que se ha hecho un cambio. También es posible mantener una conversación sobre los cambios en el pr.

Una vez decidido que queremos agregar los cambios en la rama master, hacemos merge. Para actualizar los cambios localmente tenemos que hacer pull.

- **git branch -d branchname:** Para borrar una rama que ya no utilizaremos.

Es posible que varios usuarios cambien archivos al mismo tiempo, por lo que se originan problemas al hacer merge en la rama master.

- **git commit -am "message":** Para hacer add y commit al mismo tiempo, pero solo con archivos que ya han sido creados.

stash: temporary holding place

Es importante mantener tu rama actualizada con la rama master.

- **Merge conflict:** Cuando hay cambios diferentes en la rama que estamos bajando del repositorio remoto y la que estamos trabajando localmente. Después de resolver los conflictos, hacemos otro commit en la rama que estamos trabajando.

Undoing in Git

Podemos deshacer cambios que ya tenemos en el staging area con **git reset**.

Para deshacer el último commit, usamos **git reset HEAD~1**

- **git log:** Para ver el registro de commits que hemos hecho y su correspondiente hash.

Para regresar a antes de un commit específico, usamos el hash del commit que queremos y usar git reset hash.

Forking

Para copiar un repositorio remoto existente en GitHub. Lo agregamos como un repositorio nuevo en nuestra cuenta. Los cambios que hagamos no formaran parte del repositorio original. Si quisiéramos agregarlos al original, tenemos que hacer un pr.