El mejor tutorial que he visto en años:

<https://www.youtube.com/watch?v=RGOj5yH7evk&t=1496s&ab_channel=freeCodeCamp.org>

Está en inglés. Se los voy a traducir.

# Objetivos

* Actualizar mi repositorio local con el repositorio remoto.
* Actualizar el repositorio remoto desde mi repositorio local.
* Información de cambios: quién, cuándo, qué.
* Control de versiones.
* Restaurar versiones anteriores.
* Trabajar con branches y forks.
* Control de accesos.

# Control de versiones

El control de versiones es la gestión de los cambios en un sistema cualquiera, en el código que desarrollamos. Trabajo en mi PC en un sistema. De vez en cuando, por ejemplo cada noche, le digo a Git que me guarde una foto (snapshot) de mi proyecto. Git la guarda, y nunca más la pierde ni se la olvida, de modo que yo siempre puedo recuperar versiones anteriores de mi proyecto.

# Repositorio o directorio o carpeta

El lugar, la carpeta o directorio, donde mi proyecto está guardado. Un repositorio es básicamente lo mismo que un proyecto. Adentro de esa carpeta y sus subcarpetas, guardo el código, las imágenes, los documentos de desarrollo y cualquier recurso. Puedo poner todos los repositorios o proyectos individuales dentro de una carpeta contenedora. Esta carpeta contenedora no necesita ser un repositorio Git, o sea que no necesito inicializarla.

# Git y GitHub

Uno puede crear o editar cualquier archivo localmente en Git, o en la nube en GitHub.

# Git

El sistema más popular de control de versiones. Es libre y gratuito. Es la herramienta que en mi PC local rastrea todos los cambios de los proyectos que yo le dije que vigile.

# Comandos de Git Bash y de Git

## cd

Change directory. OJO poner \\.

## ctrl + c

Interrumpe la ejecución de cualquier comando.

## copiar de la consola al clipboard

Simplemente seleccionar el texto deseado en la consola. Listo. Ya está.

## exit

Sale de Git Bash.

## shift + insert o simplemente insert

Pega en la consola el contenido del clipboard de Windoes.

## git add

Cada vez que agrego, edito o borro algo en mi proyecto, le digo a Git que quiero que me vigile eso: un directorio o un file. Puedo poner add ., o sea el dot, este directorio. Si no uso add, Git no sabe que yo lo quiero trackear.

## git clone URL\_remote\_repo

Le digo a Git que me copie en mi PC local un repositorio remoto, que está hosteado por ejemplo en GitHub. De este modo, yo trabajo en mi proyecto localmente, en mi PC local. Hay que poner la URL del repositorio remoto.

## git commit -m "a" -m "some description"

Es como save o grabar, para que Git grabe los cambios que hice desde el último commit. Le digo a Git que me saque una foto (snapshot) de mi proyecto en este momento. La m es por message. Debe haber un mensaje, pero puede ser un carácter y no significar nada.

## git config –list

Me muestra la configuración global. El archivo es C:\Program Files\Git\etc\ gitconfig, es texto plano y se puede editar con Notepad++.

## git pull

Lo contrario de push. Descargo el proyecto desde el repositorio remoto a mi PC local.

## git push origin master (o main)

Upload los commits de Git (mi PC local) a un repositorio remoto, como por ejemplo GitHub.

***origin*** es la carpeta donde está el repositorio que quiero publicar.

***master*** (o main)es la rama que quiero publicar.

## git push -u origin branch

-u significa creame un ***upstream***. Graba los parámetros que le pongo.

***origin*** es el repositorio local.

***branch*** es la rama que quiero publicar.

Una vez hecho esto, uso git puch origin

## git remote add <shortname\_del\_repo\_remoto> <URL\_repositorio\_remoto>

Agrega una referencia de un repositorio local a un repositorio remoto. Para agregar un nuevo repositorio Git remoto como un nombre corto al que puede hacer referencia fácilmente, ejecute git remote add <shortname> <url>. La expresión "repositorio Git remoto" está en el manual oficial de Git, y yo pienso que puede resultar confusa. Los repositorios remotos "están físicamente hosteados" en GitHub. Git corre en mi PC local, y solo en ella. Todos los repositorios que Git maneja directamente son locales, están en mi PC. sin embargo, Git también puede operar sobre los repositorios remotos, que nativamente son de GitHub, SUPUESTO que le doy los privilegios suficientes.

## git remote --v

Me muestra todos los repositorios remotos que estén conectados al repositorio local en el cual estoy parado.

## git status

Me muestra todos los files que fueron creados, editados o deleteados. Ojo que solo los files que hayan sido added serán trackeados. Los files modificados que no hayan sido added, se mostrarán como untracked. Antes de poder commit ese cambio, lo tengo queadd que que Git lo trackee. Una vez que están added, entonces sí están listos para ser committed.

## git version

Muestra en Git Bash o la consola que uno use la versión instalada de Git.

## pwd

Print working diretory. Me dice en qué directorio estoy.

# GitHub

Un sitio web para hostear mis repositorios en la nube. Lo que yo hago localmente en mi PC, y localmente Git lo mantiene vigilado porque yo le dije, GitHub lo sincroniza a la nube. Esto facilita el trabajo colaborativo remoto.

## Comandos de GitHub

### pull request

# README.md

La extensión md es por mark down. Es en principio el primer file que uno crea en un proyecto. Contiene texto que explica qué es proyecto, qué hace, y cualquier otra información relevante. Se puede crear on line o localmente.