GIT Y GITHUB

https://bluuweb.github.io/tutorial-github/01-fundamentos/

¿Qué es GIT?	¿Qué es GitHub?
Es un software de control de versiones, su propósito es llevar registro de los cambios en archivos de computadora y coordinar el trabajo que varias personas realizan sobre archivos compartidos.	Es una plataforma de desarrollo colaborativo para alojar proyectos (en la nube) utilizando el sistema de control de versiones Git, Además cuenta con una herramienta muy útil que es GitHub Pages donde podemos publicar nuestros proyectos estáticos (HTML, CSS y JS) gratis.

GIT

Versión de git

// Conocer la versión de git instalada

git version

Registrar nuevo usuario asociado a git:

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

git config --global user.email "myemail@example.com"

Ayuda

// Ayuda sobre los comandos

git help

Mi primer repositorio

// Iniciar un nuevo repositorio // Crear la carpeta oculta .git

git init

// Ver que archivos no han sido registrados

git status

// Agregar todos los archivos para que esté pendiente de los cambios

git add .

// Crear commit (fotografía del proyecto en ese momento)

git commit -m "primer commit"

// Muestra la lista de commit del más reciente al más antigüo

git log

Trucos

// Muestra en una línea los commit realizados

git log --oneline

// Muestra en una línea los commit realizados pero más elegante

git log --oneline --decorate --all --graph

// Solo muestra los archivos modificados

git status -s

// Vemos información de la rama maestra

git status -s -b

git status -sb //Hace lo mismo que el comando anterior

Diferencias entre -- y -

– decorate hace referencia a una palabra
-s hace referencia al comando o a varios comandos, -sa serían dos comandos diferentes

Creando alias globales

// Guardamos el alias "lg" que ejecutará todo lo que está entre comillas

git config --global alias.lg "log --oneline --decorate --all --graph"

// Para ver el archivo config con los alias creados

git config --global -e

Renombrar archivos

// Cambiar nombre

git mv nombreOriginal.vue nombreNuevo.vue

Eliminar archivos

// Cambiar nombre

git rm nombreArchivo.vue

Ignorando Archivos

Para no hacer seguimiento de carpetas o archivos, debemos crear el siguiente archivo:

.gitignore

Su estructura de ejemplo sería así:

arhivo.js // Ignora el archivo en cuestión

*.js // Ignora todos los arhivos con extensión .js

node_modules/ //Ignora toda la carpeta

Ramas o branch

// Crea una nueva rama

git branch nombreRama

// Nos muestra en qué rama estamos

git branch

// Nos movemos a la nueva rama

git checkout nombreRama

- **Atajos**

Podemos utilizar **git checkout -b nuevaRama** para crear la nuevaRama y viajar a ella.

Podemos unir la rama master con la nueva, para eso tenemos que estar en la master para ejecutar el siguiente comando:

// Nos movemos a la nueva rama

git merge nombreRama

// Eliminar una rama

git branch -d nombreRama

Tags

Con los tags podemos hacer versiones de nuestro proyecto.

// Crear un tags

git tag versionAlpha -m "versión alpha"

// Listar tags

git tag

// Borrar tags

git tag -d nombreTags

// Hacer una versión en un commit anterior ej: f52f3da

git tag -a nombreTag f52f3da -m "version alpha"

// Mostrar información del tag

git show nombreTag

GITHUB

Crear una nuevo repositorio

Para subir nuestro proyecto debemos crear un nuevo repositorio, al momento de la creación nos mostrará una serie de comandos para subir el proyecto.

git remote add origin https://github.com/bluuweb/tutorial-github.git

git push -u origin master

Al ejecutar estas líneas de comando te pedirá el usuario y contraseña de tu cuenta de github.

// Nos muestra en qué repositorio estamos enlazados remotamente.

git remote -v

Subir los tags

Por defecto si creaste un proyecto con diferentes versiones no subirá los tags, para eso tenemos el siguiente comando.

git push --tags

Push

Al ejecutar el comando git push estaremos subiendo todos los cambios locales al servidor remoto de github, ten en cuenta que tienes que estar enlazado con tu repositorio, para eso puedes utilizar **git remote -v** luego ejecuta:

git push

Pull

Cuando realizamos cambios directamente en github pero no de forma local, es esencial realizar un pull, donde descargaremos los cambios realizados para seguir trabajando normalmente.

Es importante estar enlazados remotamente, puedes verificar con: **git remote -v,** luego ejecuta:

git pull

Fetch

Este comando hace la comparación de nuestros archivos locales con los del servidor, si existiera alguna diferencia nos pediría realizar un git pull para realizar un match de nuestros arhivos locales.

git fetch

Clonar repositorio

Para descargar un repositorio completo basta con tomar la url ej: https://github.com/bluuweb/tutorial-github.git y ejecutar el siguiente comando en alguna carpeta de su computadora.

git clone https://github.com/bluuweb/tutorial-github.git nombreCarpeta