Guía

**Git & Github**



**Curso** **GIT & GitHub.**

**Sección** Todos los comandos.

Uso.Git nos sirve para gestionar los cambios que cada persona hizo en el proyecto.

* Hablar de que git nos sirve para las branches
* Para hacer duplicados de sitios
* Generar versiones de nuestro codigo

Comandos.

* **git init** 
  + Nos ayuda a crear un repositorio.
* **git** 
  + Es el comando inicial.
* **git status**
  + Es para saber el estado de nuestras versiones.
  + Explicar que Onbranch master es la rama principal
  + No commits yets significa que aun no hay ninguna versión que UNTRACKED files es un archivo que no esta en el commit (version)
* **git add index.html**
  + Para agregar un archivo.
* **git commit –m “descripción de nueva versión”**
  + Para crear una versión
* **git add .**
  + Para agregar todos nuestros archivos antes de realizar el commit.

**NOTA:**

* **Debemos primero poner git add . antes de hacer nuestro nuevo commit.**
* **No debemos hacer versiones por cada cambio.**
* **git rm –catched index.html**
  + Para remover un archivo creado en una versión especifica.
* **git rm -- cached -f index.html**
  + Sirve para remover un archivo con error

Estadisticas de los commits.

* **git log**
  + Nos permite saber que usuario fue quien hizo algun cambio y nos deja ver su correo electrónico.
* **clear**
  + Limpia lo que hemos escrito.
* **git log –-stat**
  + Para ver las estadísticas.
* **git log –online**
  + para ver todo en una linea
* **git log –graph**
  + para poder poner todo con ramificaciones en los costados
* **git diff**
  + sirve para comparar el codigo
* **git diff id amigos.html**
  + para comparar como era un archivo
* **git checkout id**
  + Nos permite ver la id de algun commit realizado

Branches.

* **git branch**
  + para ver la rama en donde estamos
* **git checkout master // Puede ser la id de la branch o el nombre de la rama.**
  + para poder salir del commit (Y elminar dicho commit) en que estabamos y poder solo estar en la rama master la cual almacena la información.
* **git checkout -b nuevaRama**
  + para poder generar una nueva rama y copiar nuestro proyecto a dicha rama
* **git checkout rama**
  + para poder movernos entre ramas que ya tenemos
* **git merge rama**
  + rama para poder poner contenido de una rama a la rama actual (COPIAR)

Cuenta de Github. Para ello lo que requerimos es generar nuestra KEY y pegarla en nuestra interfaz de Github.

Comandos de Github. (Subir y bajar). Primero debemos generar nuestro nuevo repositorio en Github para poder subit nuestros archivos.

* **git remote add origin url** // la URL nos la daran en Nuestro repositorio o la podemos comprobar con remote -v
* **git push –u origin master** // nombre de la rama o la id // Nos sirve para subir dicha rama a Github.
* **git pull origin master** // sirve para actualizar los elementos de nuestro contenido en Github
* **git remote –v** // Nos sirve para ver donde es que se encuentra el contenido en github de lo que acabamos de subir.