Git y GitHub

Comandos en Git (Consola de git).

* git status:

Verifica el estado actual de los archivos que estoy trabajando con git. Muestra los archivos que han sido modificados o eliminados, de manera que informe al usuario lo que ha pasado.

* git add <archive>

Selecciona un archivo para realizar un commit

* git commit

Abre una ventana para escribir el título del commit

* git commit –m “message”

Agrega el título al commit directamente

* git reset --soft HEAD­­­~1

Deshace el último commit realizado.

* git push

Agrega un repositorio a la nube o aun repositorio remoto: Ej. GitHub

* git remote

Obtiene el nombre de la dirección del repositorio remoto

* git remote –v

Obtiene la Url del repositorio remoto

* git push origin master

Sube el repositorio a GitHub los parámetros son remoto=origin y el rama=master

* git pull

Descarga los archivos del remoto

* git pull origin master

Obtengo los datos que hay en el repositorio en la rama master remoto al pc

* git commit .am “message”

Seleccionó todos los archivos que he modificado localmente para realizarle un commit

* git branch

Obtiene todas las ramas creadas en el proyecto

* git branch <name>

Creo una nueva rama con el nombre que quiera

* git checkout <name>

Cambio de una rama a otra pasándole el nombre de la rama que quiero estar

* git checkout –b <name>

Creo nueva rama y me coloca en esa nueva rama

* git log –-graph –-decorate --oneline –all

Muestra de forma clara los commits realizados en cada rama, sin información extra, solo con el texto.

* git merge

Trae los cambios realizados de una rama a otra.

* git merge <name-rama>

Trae los datos de la rama con el nombre que le he dado a la rama actual en la que esté, creando una nueva mezcla

* git branch –d <name-branch>

Elimina una rama que no vaya a utilizar

-d comprueba que no vayas a perder el trabajo

-D git no comprueba las ramas a eliminar, no comprueba la mezcla que haya entre una rama secundaria y la rama principal

* git gc

Elimina todos los archivos basura, para liberar espacio en el disco.

push -> Envía

pull->Descarga

commit -> Anotación

issue -> Publicación, errores o problemas.

branch -> Rama Es una extensión del proyecto principal con sus propios commit y archivos, los puede utilizar para la creación de nuevas características del proyecto sin afectar el original

merge -> Mezclar, fusionar o Unir, ramas en git.

pull -> Jalar o traer un repositorio que esta remoto.

push -> Empujar, presionar o subir un repositorio trabajado localmente a un servidor remoto.

Al realizar commit puedo hacer referencia a issues colocando la almodilla o numeral

#1 junto con el número que hacer referencia al issue tratado

Para cerrar un issue, se puede hacer desde GitHub o desde la terminal cuando se realiza un commit colocando la palabra **Closes #1** en la descripción del commit

Con Fork, puedo traer un repositorio de otra persona para poder utilizarlo como propia y realizar cambios.

* git clone -> clona un reopsitorio en el computador
* $ git add –p <archive>

Preguntá que cambiós quiere que se vea en el historial der archivo, luego cuando se haga el commit