

Manejo de Git usando GitHub

Creación de un repositorio local (en tu computadora)

git init

Este comando te permite inicializar un repositorio en local. Al hacerlo en la carpeta donde se ejecutó se creara un nuevo directorio (.git)

Guardar cambios en del archivo en la RAM

git add .

Usado para guardar los cambios de los archivos, Es recomendable que siempre que se agregue un nuevo archivo usarlo para que se guarde y evitar inconvenientes. No olvidar que luego del add hay que dar un espacio y poner (un punto)

Verificación del estado del repositorio

git status

Con este comando se puede ver el estado actual de los directorios y archivos, muy útil en caso de que se quiera verificar si se están agregando los cambios

Guardar los cambios

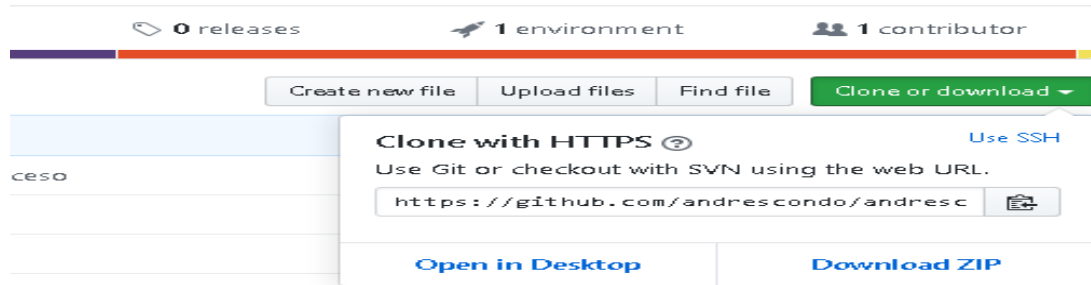
git commit -am' (mensaje) '

Guarda los cambios del último guardado del archivo en el editor. Este guardado sirve almacenar los cambios que se quieren mandar al repositorio remoto. Estos cambios no necesitan necesariamente el mensaje, pero se lo aplica como buena práctica para dar una idea de los cambios que se están enviando

Unir un repositorio remoto al local

git remote add origin (URL)

El URL/link del repositorio remoto se lo encuentra en el repositorio creado, es un botón verde llamado |Clone or download | para este comando se usa el HTTPS



Ejecutado este comando te da acceso enviar los cambios a ese repositorio remoto que has creado

Traer cambios por primera vez

git pull origin master --allow-unrelated-histories

Este comando permite fusionar el repositorio remoto y el repositorio local(donde estas trabajando ahora).

Este comando te mandara a una pantalla un poco extraña pero para salir hay dos formas

- 1) Pulsando las teclas |Esc | + |Shit| + |z | + |z|
- 2) Escribiendo | :q | y ENTER

***Nota:** Este comando solo se lo necesita hacer una vez por repositorio local

Traer los cambios

git pull origin master

Trae todos los cambios que se han realizado en el repositorio remoto. Es una buena práctica hacerlo desde que se está aprendiendo para acostumbrarse a traer los cambios que se les vaya aplicando. Ya que al estar en equipos de trabajo se realizan cambios constantemente.

Enviar cambios

git push origin master

Envía todos los cambios que se guardaron con el 'commit'

Ver las configuraciones

git config

Permite ver todas las configuraciones de que se puedan ejecutar

Configuraciones

git config --list

Permite ver todas las configuraciones que están guardadas

Añadir/Cambiar las configuraciones

git config --global user.name (Nombre o Nick)

git config --global user.email (direccion de correo)

Estos comandos permiten guardar los cambios que se les hace a las configuraciones de manera global. Todos estos cambios que se les va aplicando al GIT del dispositivo usado, en caso de usar más de un ordenador, se recomienda hacerle todos estos cambios

Trabajar con repositorios públicos

git clone (URL)

Permite traer una copia del master de un repositorio público/privado al directorio donde se ejecute el comando.

Es recomendable aplicar los comando de guardado en la RAM, y el commit, para luego enviarlo con el comando git push (sintaxis completa ya explicada arriba).

Generar Ramas

git branch (nombre_de_la_rama)

Las ramas son copias de la rama principal donde se le pueden aplicar cambios sin afectar a la rama principal, estos cambios se aplican a un nuevo archivo que solo existe en la rama en cuestion. Es muy útil para trabajar en paralelo.

Moverse entre ramas

`git checkout (nombre de la rama)`

Permite moverse entre ramas con el. En caso de que se quiera ver cuantas ramas existen se lo puede hacer ingresando el mismo comando pero sin ningún nombre

Unir ramas

`git merge (nombre_de_la_rama)`

Funciona dos ramas, aplicando los cambios de la rama sobre la rama donde se aplique. Es recomendable que se aplique este comando desde la rama principal.
