**Curso de Git**

Configuración de inicio de Git bash una vez instalado git.

Ingresar nuestro nombre

* **git config --global user.name “luis**”

Ingresar nuestro correo el cual debe ser el mismo de git hub.

* **git config --global user.email “j.luisfranco.cucei@gmail.com”**

si queremos cambiar el nombre o email que ingresamos basta con ejecutar el mismo comando solo cambiando el nombre a ingresar.

Poner de colores los distintos resultados que nos da git cada vez que ejecutamos algún comando.

* **git config --global color.ui true**

Observar la lista de las configuraciones que hemos realizado.

* **git config --global --list**
* **git config -l**

Para limpiar la consola de git bash.

* **Clear**

Salir de una ventana de git

* **q**

Salir de git bash.

* **exit**

Ayuda

* **git help**

Ayuda especifica de algún comando

* **git help “palabra clave”**

Una vez posicionados en el directorio donde se encuentran los archivos a manejar ingresamos el siguiente comando para indicarle a git que monitoree todos los cambios realizados, este comando solo se usa al inicio del proyecto y solo se usa una sola vez.

* **git init**

Para ver el estado de nuestros archivos.

* **git status**

Agregar archivos listos para el siguiente paso.

* **git add “nombre archivo”**

Para agregar todo lo que está dentro del directorio.

* **git add -A**

Guardar los cambios que se hicieron agregando un mensaje descriptivo de lo realizado.

* **git commit -m “mensaje descriptivo………”**

Observar los commits que se han realizado.

* **git** **log**

Observar los commits incluido los cambios en el archivo.

* **git** **log -p**

Observar los commits en una sola línea.

* **git log --oneline**

Observar los commits en una sola línea con grafica de las ramas.

* **git log --graph --oneline**

Crear un txt con todos los commits que hemos realizado.

* **git log > nombreArchivo.txt**

Podemos movernos entre commits para visualizar los cambios que se han realizado o ver errores, para ello se ejecuta el siguiente comando utilizando el código sha que se nos muestra cuando ejecutamos un git log.

* **git** **checkout “código sha del git log”**

Colocarse en el último commit que se ha realizado.

* **git checkout master**

Las dos maneras de eliminar commits son las siguientes:

Eliminar un commit sin que se altere nuestro código.

* **git reset --soft “código sha del git log”**

Eliminar un commit con todo y el código que se modificó en ese commit.

* **git reset --hard “código sha del git log”**

**Nota: la rama head es el commit donde nos encontramos actualmente y la rama master es la que se genera al inicio del proyecto cuando ejecutamos el git init.**

Las ramas son muy útiles ya que sirven para no alterar directamente el proyecto original y arreglar errores o hacer cambios.

Ver las ramas existentes.

* **git branch**

Crear una rama.

* **git branch “nombre rama”**

Movernos entre ramas.

* **git checkout “nombre rama”**

Crear una rama y al mismo tiempo movernos a ella.

* **git checkout -b “nombre rama”**

Para fusionar ramas lo primero que hay que hacer es situarse en la rama que absorberá a la otra es decir **git checkout rama** y después ejecutar el siguiente comando.

* **git merge “rama que se absorvera”**

Al realizar merge existen 2 tipos de fusiones fast-forward y manual merge, donde fast-forward solo hace la fusión sin preguntar nada y manual merge antes de realizar la fusión tiene que pasar por nosotros.

Para eliminar una rama.

* **git branch -D “nombre rama”**

**USO DE GITHUB**

Si se requiere cambiar las credenciales para autentificarse en github se ejecuta el siguiente código.

* **git config credential.username --global “nombre”**

Después de eso cuando se quiera conectar con github se pedirá ingresar la contraseña.

Si solo quiere cambiar las credenciales dentro de un solo repositorio.

* **git config credential.username “nombre”**

Para copiar un repositorio de git hub.

* **git clone “url del repositorio en github”**

Comprobar que nuestro repositorio esta sincronizado con github.

* **git remote -v**

Eliminar conexión con el repositorio que se sincronizo.

* **git remote remove origin**

Subir los commits al repositorio remoto de github.

* **git push origin master**
* master es el nombre de la rama que se subirá al repositorio por lo que puede ser otra.