***COMANDOS DE GITHUB***

**PARA ENLAZAR UNA PC CON LOS REPOSITORIOS REMOTOS**

**git config --global user.email “**[**coremanrique@gmail.com**](mailto:coremanrique@gmail.com)**”**

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

* Para verificar el usuario y correo que enlaza esta pc con el repo remoto

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

**git config --global --unset-all user.name 🡪** con esto eliminamos user.name de git

**git config --global --unset-all user.email 🡪** esto elimina user.email de git

**PARA GUARDAR PROYECTO EN REPOSITORIO LOCAL**

**git init** 🡪 esto crea la rama principal y el repositorio local

**git status** 🡪 seguimiento de los archivos pendientes o no guardados

**git add .** 🡪 solo se agregarían los archivos nuevos y modificados área temporal

git add --all 🡪 incluye todos los cambios, incluyendo nuevos archivos, modificaciones y eliminaciones

**git commit -m “comentario”**  🡪 envía todos los archivos a repositorio local

**ENLAZA REPOSITORIO LOCAL CON REPOSITORIO REMOTO (ENVIA ARCHIVOS DE OTRA RAMA)**

**git remote add origin https://enlace del repositorio remoto**

**git remote -v** 🡪 confirma si esta enlazado los repos locales y remotos con push y fetch

**git push -u origin master** 🡪 con esto se envía los archivos locales al repo remoto

**git push –force origin main** 🡪 Esto forzara el envió de los archivos a la rama principal

**CREACION DE NUEVAS RAMAS**

**git branch rama2** 🡪 se crea nueva rama

**git branch** 🡪 para ver todas las ramas

**git checkout (rama elegida)** 🡪 seleccionamos la rama en la que deseamos trabajar

**git branch -m master main** 🡪 para cambiar de nombre de master a main

**RECUPERAR CUALQUIER RAMA CLONADA QUE TENGA NUESTRO REPOSITIO REMOTO**

**git fetch** 🡪 con esto se descarga y trae al repo local todas las ramas pero no se pueden ver

**git branch -r** 🡪 podremos ver las ramas existentes y trabajar en cualquiera de ellas

**ELIMINAR RAMAS**

**git branch -d (rama a eliminar)** 🡪 eliminara rama del repo local

**git push origin --delete** (rama a eliminar) 🡪 eliminara esta rama del repo remoto

**ELIMINAR ARCHIVO DE CUALQUIER RAMA**

**git rm -r (archivo a borrar)**

**CLONAR**

**git clone** [**https://enlace**](https://enlace) **del repositorio a clonar**

**git pull origin main** 🡪 actualizaremos todos los archivos de la rama padre en nuestro repo local

**git push origin main** 🡪 Se enviara todos los cambios del repo local al repo remoto

**git push –force origin main** 🡪 Esto forzara a subir los cambios locales al repositorio remoto

**FUSIONAR RAMAS SECUNDARIAS CON RAMA PRINCIPAL O MASTER**

**git checkout master** 🡪 primero debemos estar en la rama principal

**git merge rama2** 🡪 elegir la rama que se quiera fusionar con la principal

**git merge --no-ff rama2** 🡪 esta opción fusionara cambios pero conservara ambas ramas y commits

**TAG**

**git tag -a 0.0.1 fn1v3bn4 -m “release 0.0.1”** 🡪 (0.0.1)=nombre del tag; (fn1v3bn4)= el commit

**git tag 1.0.0 -m “reléase 1.0.0”**

**git push --tags** 🡪 recién se podrá subir los tags al repo remoto

**git tag -d nombreTag** 🡪 Borrar tag

**REGRESAR O RESETEAR A UN PUNTO ANTES DE UN MERGE O FUSION**

**git checkout c19b41d** 🡪 con esto iremos a commit donde queremos volver

**git checkout master** 🡪 con esto regresamos a la rama padre

**git reset --hard c19b41d** 🡪 volverá al commit seleccionado o su estado anterior al cambio

**VER COMMITS EN NUMEROS Y EN FORMA GRAFICA**

**git log --oneline** 🡪 se vera los commits en números y letras

**git log --oneline --all --graph** 🡪 se vera los commits en grafico

**VER ARCHIVOS DE CADA RAMA**

**dir**

**CAMBIAR DE NOMBRE MASTER A MAIN**

**git branch -m master main**

**git config –global init.defaultBranch main 🡪** Con esto se pondrá por defecto a main como rama principal

**ELIMINAR RAMA HEAD/MAIN**

git branch -r -d origin/head

INICIAR TODOS LOS DIAS CON LA SICRONIZACION PARA EVITAR CONFLICTOS

1 git checkout main (branch a sincronizar) va a mover a la branch que finalmente el código hará merge

2 git pull –rebase (este comando va a traer todo el código desde el repo remoto hacia el repositorio local)

3 git checkout develop (branch local a la que deseamos sincronizar)

4 git rebase main ( el nombre de la branch a la cual va hacer merge tu código, este comando pone tus commits encima de todo el código sincronizado desde el repo remoto)