Git

Git init: crear el ~~área de ensayo y el~~ repositorio local

Git add: lleva el archivo desde el directorio de trabajo al área de ensayo

Git commit: lleva el archivo desde el área de ensayo al repositorio local –

Podemos agregar una descripción

Git commit – m “descripcion” : crea un respaldo del archivo

Podemos obtener el respaldo del archivo al llevarlo al repositorio

Git status -s : entrega un estatus de todos los archivos y directorios en el momento

que no está en el repositorio

??: Significa que ninguno esta bajo seguimiento(git add)

Luego al usar git add: el archivo aparecerá con una “A” de agregado

Si se ha modificado el archivo, aparecerá con una “M” roja, queriendo decir que no esta respaldado

Para agregarlo al área de ensayo o preparar el archivo para respaldarlo(actualizarlo) : git add

Git commit -m “descripcion nueva” : y hace el cambio

Git log --oneline : entrega un listado de las copias con su código y descripción

Para restaurar el archivo :

Git reset –hard 0931d4 (codigodecambio)

Git add .

Agrega(hace seguimiento) todos los archivos disponibles al repositorio “A” verde

Si queremos hacer git add y git commit –m “x” podemos unificarlo en el siguiente comando:

Git commit -am “x”

Para editar una descripción hecha en commit:

Git commit –amend y abrirá el editor (Visual studio) ahí editamos, guardamos y cerramos y estará actualizado el cambio

Subir repositorio a github

Primero se crea un nuevo repositorio

Luego esto:

echo "# cursoGit" >> README.md

git init

git add README.md

git commit -m "first commit"

git branch -M main

git remote add origin https://github.com/cerberuuss/cursoGit.git

git push -u origin main

git push -u origin main =Volver a subir archivos

Se puede realizar cambios desde la web github

Git pull = si realizamos cambios desde la web con esto se actualizaran nuestros archivos locales

Tag

Git tag nombre -m “descripción” = decimos que tenemos la v1

Git push –tags =Se suben las tag al repositorio

Git clone “url” =Podemos clonar el archivo, en caso de perdida o para descargarlo

Branch o ramas

Linea de tiempo de lo que se ha hecho con el proyecto

Se le llama Master(main), nombre que le da git (se le puede cambiar pero suele usarse por defecto)

Podemos volver atrás a alguna parte de esta línea de tiempo

Se puede crear otras ramas paralelas(copia)

Se puede unir la rama adicional a la master(merch) fusionando las dos ramas

Git Branch nombre =crea una nueva rama

Git Branch = permite ver las ramas del proyecto y en que rama me encuentro(verde y asterisco)

Git checkout nombre = Moverse a otra rama para editar en esa rama

Git merge nombreramaafusionar = permite fusionar las ramas pero debe ser llamado desde rama main/master

Git Branch -d nombrerama = para eliminar una rama ya habiendo realizado el merge