GIT

Git init (crea repositorio GIT)

Git config –global user.name “Daniel Prieto”

Git config –global user.email [Daniel.ric.prieto@gmail.com](mailto:Daniel.ric.prieto@gmail.com)

Git config –list (me informa los datos de user)

Git config user.name (me dice lo grabado en esta variable)

Git status (me muestra el estado de las 3 areas)

WOKING DIRECTOY

STAGING AREA

LOCAL REPOSITORY

Git log (me dice todo lo comiteado de la rama en la que este parado)

Cuando trabajo con git y creo archivos los mismos deben pasar por estas etapas :

WORKING DIRECTORY 🡪 add 🡪 STAGING AREA 🡪 commit 🡪 LOCAL REPOSITORY

Cada cambio que haga en algún archivo debo pasarlo al staging área y después el local repository.

Git add archivo.css

Git add –all Si quiero pasar todos los archivos al staging área

Git commit –m “COMENTARIO” si no coloco el comentario se abre un archivo para colocarle un comentario y después pide grabar el mismo

Git clean (ELIMNA LOS ARCHIVOS DEL WORKING DIRECTORY)

GITHUB es un repositorio remoto(se debe crear uno)

Logeandose al github

Por convención se nombra el nombre del repositorio como origin o en algunos casos upstream

Git remote add origin <https://github.com/prueba>

Git remote –v (me dice si se cargaron los archivos en el github)

Git push origin master (pasa los archivos al repositorio remoto del github)

Git pull origin master (trae los archivos desde el github)

WORKING AREA COMO VOLVER ATRÁS CAMBIOS

Git clean –f \*\*\* OJO AL UTILIZARLO \*\*\* elimina los archivos del working directory SI SOLO SON NO TRACEABLES

Git clean –n \*\*\* NO BORRA \*\*\* SOLO me dice que arhivos están para eliminar

Para eliminar un archivo es mejor :

Git checkout archivo (esta opción descarta los cambios realizados del working área y si no es trackeable lo elimina)

EN VISUAL STUDIO ES PRECIONANDO LA FECHA HACIA ATRAS

SI ME EQUIVOQUE EN BORRAR UN COMMIT Y QUIERO QUE LOS CAMBIOS VUELVAN TODOS COLOCAR

Git checkout master (De la rama en donde hice los commit)

Para moverme entre ramas colocar

Git checkout –b rama\_nueva2

Git merge mezcla la información de las ramas

Ej.

Git checkout –b rama\_nueva

Si hago cambios en la rama nueva de un determinado archivo y lo quiero agregar a la rama master o principal

Debo ir al a la rama master y luego

Git merge rama\_nueva

STAGING AREA COMO VOLVER CAMBIOS ATRÁS ( PASA DE STAGE A WORK)

Git reset HEAD nombeArchivo o en VISUAL STUDIO con el signo meno

COMMIT AREA O LOCAL REPOSITORY COMO VOLVER A STAGE

(NO HAY OPCION DESDE VISUAL STUDIO)

Vuelve atrás un commit especifico para ello debo pasar el hash code

Git revert HASHCODE

Git mv archivo\_viejo a archivo\_nuevo es para renombar archivos (este es un atajo por si en visual studio renombre el archivo después debo realizar el add y el commit para que tome los cambios) el cambio lo deja en el staging area y luego hace falta el commit como siempre.

Git branch me dice en que rama estoy master seria producción puede llamarse producción o el nombre que quiera.

Se puede crear ramas

Git branch nombre\_de\_la\_rama

Ej .

\*Master (el asterisco me dice en que rama estoy de momento)

Git branch –d nombre\_de\_la\_rama (Elimina una rama)

Git branch –m nombre\_viejo nombre\_nuevo (Cambia el nombre de la rama)

Git –d nombre\_de\_la\_rama (elimina una rama)

Git –D nombre\_de\_la\_rama(Fuerza la eliminación de una rama sin fusionar)

PASOS CUANDO SE TRABAJA EN GIT

* Crear y cambiar a una nueva rama

git checkout -b feature/nueva-funcionalidad

* Realizar Cambios y hacer comits

git add archivo-modificado.js

git commit -m "Añadir nueva funcionalidad"

* Volver a la rama principal y fusionar los cambios

git checkout main

git merge feature/nueva-funcionalidad

* Eliminar la rama de caracteristicas

git branch -d feature/nueva-funcionalidad

El stash es como una pila para guardar datos

Git stash (Sirve para guardar en la rama los cambios que estaba realizando ,sin control – s ni con gaurdar para después recuperarlos de ser necesario)

Git stash pop (recupero los cambios no guardados con control-s o guardar como)

Git stash list me dice los cambios realizados en la pila

Git stash clear elimina todos los cambios o los procesos de la pila

Git tag sirve para etiquetar los commits y etiquetar versiones realizadas si no coloco el nro de commit es que estoy etiquetando el ultimo commit realizado en la rama de la cual estoy parado.

Git tag –a versión0.0 –m “comienzo” hashcode

Git tag (me da la lista de todos los establecidos)

Git show version0.0 (Me informa el nro de commit y la información asosiada a ese tag)

Git tag –d version0.0 ( elimina el tag)