

git --version - Nos dice que version tenemos instalada de git

git config --global user.name / user. email - Configura el nombre y mail de GitHub en toda la pc, pero también podemos consultarlo.

git init - Inicia el repositorio en la carpeta donde ponemos el comando, la rama principal se llama master y se crea de forma automática.

git status - Nos fijamos el estado de la carpeta, de los cambios, es nuestro mejor amigo, lo podemos usar siempre y vemos como esta nuestro repositorio local, nos ayuda a recordar los cambios que venimos haciendo.

git add - espacio y . es para agregar todos los cambios que están en rojo.

git add nombreDeArchivo – Pasa los cambios de working directory al staging área solo de un archivo.

git commit -m "proyecto-100 mensaje explicando los cambios" – Sirve para confirmar los cambios que están en verde en el staging área y los sube al repositorio local, los cambios del staging área pasan todos al repo.

git log - Ver commit, historial de versiones, quien lo hizo, el día.

git branch - Ver las ramas existentes localmente, la rama en la que estamos parados se marca de verde y con un asterisco.

git branch nombreRama - ¡Nos deja parados en la rama desde donde creamos la nueva rama ojo!, se pueden crear ramas desde cualquier rama, en nuestro flujo de trabajo creamos las ramas para los desafíos desde la rama master local.

git checkout rama – Me muevo de rama, ojo al movernos entre ramas sí tenemos cambios realizados en la rama donde estamos parados, y no están confirmados, es decir no tienen git add o git commit, los arrastramos a la rama donde nos vamos.)

git restore . - Vuelve atrás todos los cambios en color rojo, git restore y el nombre del archivo, vuelve atrás los cambios en ese archivo.

git branch -D nombreDeRamaABorrar - Tenemos que estar parados en cualquier rama que no sea la que queremos borrar y podemos ejecutar ese comando, elimina la rama completamente.

git checkout nroCommit/nombreRama - Me muevo a un commit o a una rama

git checkout -b nombreRama - ¡¡crea una rama nueva y te envía a esa rama- usar este!!

git merge nombreRama - Agrega los nuevos cambios-borra-agrega-modifica dentro de un archivo o archivos.

git remote add origin nombreRepoRemoto – Enlaza el repositorio local con un repositorio remoto creado en GitHub o cualquier otro Bitbucket, termina con .git

git push origin master – Sube los cambios de la rama que indicamos al final al repositorio remoto enlazado, si no existe en GitHub este comando crea la rama.

git reset --SOFT/--HARD nroCommitAlCualQueremosVolver - Vuelve atrás un merge erróneo o un commit, dependiendo a donde queremos volver. --soft --hard, investigar que comando

utilizaremos dependiendo si queremos estar en el commit anterior, pero con los cambios que hicimos sin agregar al staging área, o si no queremos tener ningún cambio de los que hicimos luego de ese commit.

En el caso de que subamos algo al repo desde GitHub, pero no lo hagamos desde nuestro repo local debemos antes en nuestro repo local hacer:

git pull origin NombreRamaRemota – La rama remota tiene cambios que la rama local no tiene, se ejecuta este comando para traer los cambios remotos a nuestra rama local, es posible que a veces este comando nos pueda generar algún conflicto de mergeo (porque internamente realiza eso entre las dos ramas, se puede solucionar a mano, recuerden instalar la herramienta plugin de visual studio code – Git History que es de mucha ayuda para marcarnos los conflictos y nos pregunta con que cambios nos quedamos si los existentes en la rama(current) o los que son de la rama remota incoming.)

EN NUESTRA SITUACION DONDE TRABAJAMOS SOLOS Y SIEMPRE HACEMOS UN TRABAJO UNIDERECCIONAL, ES DECIR TRABAJAMOS LOCAL Y ACTUALIZAMOS LA RAMA DEL REPOSITORIO REMOTO, PERO NO SUBIMOS NADA A REMOTO DE FORMA INDEPENDIENTE, SOLO LO HACEMOS DESDE EL PUSH DE LA TERMINAL, NO ES NECESARIO QUE USEMOS GIT PULL ORIGIN NOMBREDERAMA.

Git no termina acá, no dejen de investigar y aprender, les dejo algunos enlaces a artículos realizados por el programador FrontEnd Certificado por Google Miguel Ángel Durán García

<https://midu.dev/buenas-practicas-escribir-commits-git/>

<https://midu.dev/como-encuentro-commit-culpable-error/>

<https://midu.dev/como-deshacer-el-ultimo-commit-git/>

<https://leanpub.com/aprendiendo-git>