Trabajo en equipo

* Se supone que ya nos dieron un link del repo o acceso a un repo en el caso de un proyecto donde somos colaboradores.
* Vamos a explicar todo paso a paso por mas que ya lo sepan pero no quiero despues quilombos.
* Cuando ya nos dieron el link o la invitacion nos llevara al repositorio original que es el repositorio del administrador, si se fijan a¿en el margen superior izquierdo deberian ver el nombre de usuario del administrador seguido de / y el nombre del repositorio.
* Eso signific aue estamos dentro del repo del administrador, si miramos al margen superior derecho veremos unas pestañas donde se encuentra la pestaña fork.
* Hacemos clic y creamos un nuevo fork . Un fork es una copia exacta del repositorio .
* Al hacer clic nos dirige automáticamente a nuestro github , si observamos al margen superior izquierdo veremos nuestro usuario/nombre del proyecto y si vemos debajo nos indica de que cuenta hicimos el fork ,eso nos indica que hicimos una copia exacta y que ese repositorio es nuestra copia.
* Lo siguiente es clonar esta copia de repo nuestra, para ello en el boton clonar damos clic y copiamos la url en https no en ssh.
* Abrimos nuestra terminal, creamos una carpeta con el comando mkdir mas el nombre que querramos usar.
* Procederemos a clonar lo que copiamos anteriormente para ello ejecutamos el comando git clone y pegamos la url copiada.
* Esto crea una carpeta con el nombre del repositorio y descarga todo su historial.
* Hacemos un ls y vemos que este esa carpeta del repo
* Entramos a la carpeta cd mas nombre del repo
* Por defecto el comando git clone ha creado el remoto origin ( si no recuerdan que es el remoto en el primer tuto esta explicado)
* Ingresamos git remote –v y vemos que el el remoto origin apunta al fork con nuestro nombre de usuario.
* Pero nos interesa también tener un enlace al remoto original del administrador con nuestro fork para luego poder pedirle que agregue nuestros cambios al proyecto que va a terminar en producción para ello haremos lo siguiente.
* Lo primero es renombrar nuestro fork y cambiar la palabra origin por fork ( usamos fork para saber que ese remoto es nuestra copia o lo que forkeamos , pero podemos poner el nombre que querramos) para ello ejecutamos git remote rename indicando el nombre actual y el nombre nuevo que tendrá como parámetros ejem git rename remote origin fork ( aca indicamos que la palabra origin de nuestra copia la cambiaremos por fork ) le damos enter.
* Hacemos un git remote –v y veremos que la palabra origin que estaba al comienzo se reemplazo por fork.
* Ahora vamos a enlazar el repositorio del administrador ( acá hay que tener mucho cuidado ,hacerlo paso a paso)
* Entramos a github donde dejamos abierto que sería nuestra copia del repo para corroborar miramos que sea nuestro usuario/nombre del repo y si vemos debajo nos indicaria de donde forkeamos y el usuario del administrador/nombre del repo original del administrador hacemos clic ahí nos cambiara de ruta ,si nos fijamos en el margen superior izquierdo nos deberia aparecer el nombre del administrador/nombre del repo. Eso nos indica que estamos dentro del repo del administrador, una vez alli haremos clic en clonar repositorio y copiamos la url https
* Volvemos a la terminal, como debemos agregar un nuevo remoto lo hacemos con el comando git remote add origin ( origin es para identificar , podemos poner administrador, o lo que querramos , lo importante es entender que es el repo original y no el fork o copia) por convencion se usa origin , pegamos al url copiada y le damos enter
* Ejem git remote origin + url ttps
* Hacemos git remote –v y veremos que ahora nuestro fork esta ya enlazado a el remoto del administrador
* Abrimos el proyecto con code .
* En un flujo de trabajo ideal nosotros no deberíamos hacer modificaciones en la rama main, en su lugar debemos crear nuevas ramas en las cuales trabajar para ello ejecutamos el comando git checkout –b mas el nombre que le querramos poner a la rama generalmente se usa features/seguido del nombre de quien trabaja en caso que no sean muchos , en caso de ser muchos se usa features/seguido de el trabajo que estamos haciendo o correccion en la que trabajamos por ejem features/cambio-de-funciones-de-js de esa manera queda claro que trabajo ha hecho esa rama. En las empresa como es el caso de esta gente puede pasarte un codigo ejem features/SS-45245-s eso dependera de las reglas de cada empresa.
* Para crear ramas el comando es git checkout –b y para cambiar de rama se usa el mismo comando pero sin la –b es decir git checkout mas el nombre de la rama a la queramos ir.
* Para probar que hicimos bien nuestro cambio hacemos git branch y nos deberia aparecer la rama main y la rama creada, en la que estemos posicionado aparecera en verde y con un asterisco.
* Entramos a nuestro visual hacemos algun cmabio y volvemos a la terminal para ver si esta tomando esos cambios, hacemos git status y vemos que se hayan puesto en rojo los cambios a agregar . Hacemos un git add mas el nombre del archivo o un punto si queres subir todo lo que esta en rojo y enter
* Hacemos el git commit –m “xxxxx” y enter y veremos que no tenemos mas para hacer.
* Ejecutamos el comando git log –oneline y vemos que aparece nuestra rama
* Ahora vamos a pedirle al administrador que acepte nuestros cambios
* Primero debemos enviar la rama local al fork nuestro para eso hacemo el comando git push el remoto al que vamos a enviar los cambios y luego la rama del remoto que los va a recibir ejem git push fork ramax( ramax es nuestra rama features/xxxx)
* Recordar que los cambios que se enviaran corresponden a la rama en la que estamos ubicados al moneto de ejecutar este comando en este caso seria rama features/xxx
* Como hemos indicado que el repositorio remoto recibirá los cambios en una rama llamada feature/xxx el repositorio remoto, se encargara de crear esa rama en caso de no existir , la salida del comando nos dice que la nueva rama ha sido creada en el repositorio nuestro es decir en nuestra copia fork.
* Si vamos a nuestro GitHub ( corroborar en margen superior izquierdo que estamos en nuestro usuario/nombre del repo), vemos que existen dos ramas en nuestro repositorio en la pestaña que dice branch , vemos que evisten dos ramas , main y features/xxxx
* Cuando vemos el ultimo historial de commits que no recuerdo donde esta , no vemos nuestro ultimo comit porque estamos en la rama main y para verlo entramos a la branch feature/xxx y nos mostrara todos los commits incluido el ultimo.
* Hacemos click si queremos ver los cambios que hicimos, volvemos atrás donde están las dos branch y no se donde no recuerdo buscamos la opcion para hacer el pull request. Si alguno se acuerda por favor déjelo indicado porque no pienso abrir todo de nuevo jajajaaja. Bueno eso era facil hacen clic y con eso solicitamos al administrador que acepte nuestros cambios y los incluyan en la rama principal, si recuerdan indican que cambios hicieron una breve descripción . Si vemos las pestañas hay unos campos ya completados.
* En el campo base fork nos muestra el repositorio original donde queremos que nuestros cambios se mezclen, en el campo base la rama donde queremos que nuestros cambios se mezclen, en el campo head fork indica el repositorio en donde están los cambios que queremos enviar y en el campo compare esta la rama que tiene los cambios que queremos enviar. ( esto creo que cambio como esta distribuida la info en la ventana al menos me cambio en la nueva version de github a como lo hacia antes pero los campos osn esos y significan eso amen de donde esten ubicados)
* Indicamos titulo y descripcion breve para convencer de que acepeten los cambios. Hacemos clic en create pull request o crear pull request
* Github notifica al administrador y el administrador vera si acepta o no el merge y si lo acepta llega la notificación que fue aceptado el cambio.
* Como se dijo al principio , nuestro fork o copia contenia todo el historial al momento del crear nuestro fork o copia por lo tanto no tendremos la actualizacion que hizo el administrador para ello debemos traernos esas actualizaciones para eso volvemos a la terminal
* Corroboramos en que rama estamos ubicados con git branch , si estamnos en la rama feature/xxx nos pasamos a la rama main para ello ejecutamos git checkout main.
* Recuerden que el proyecto esta conectado a dos remotos nuestro fork o copia y el origin o original.
* Hacemos un git pull origin main +enter
* Esto nos traera todo el nuevo historial de la rama main del repositorio original del administrador , a la rama en la que estamos ubicados en local ( nuestra rama main)
* Ejecutamos git log y vemos que aprecen nuevos commits que llegaron del repositorio original del administrador
* El siguiente paso es enviar esos nuevos commits a nuestro fork o copia trucha jajajaja asi tendremos esos cambios para eso ejecutamos lo siguiente.
* - git push fork main
* Asi tendremos todo lo nuevo en nuestra copia o fork
* Ahora ya nuestra rama feature/xx cumplio su función por lo tanton la vamos a eleminar para ello ejecutamos git branch –d feature/xxxx ( importante ver que no estemos parados en la rama que queremos eliminar porque nos dara error)
* Para mantener las cosas en orden también vamos a eliminar la rama featrure/xxxx del repositorio remoto fork para eso vamos a ejecutar
* Para eliminar una rama remota debemos ejecutar git push indicamos el remoto dejamos un espacio y ponemos –delete espacio y luego el nombre de la rama remota que queremos eliminar por ejem git push fork –delete feature/xxxx
* Ahora el historial esta limpio y sincronizado para volver a trabajar en otra rama de ser necesario.
* Si vamos a nuestro github vemos que ya nos aparecen los nuevos commits tal como esta en repositorio original del administrador.

REPASO RAPIDO

Creamos una nueva rama git checkout –b rama/nueva

Hacemos algun cambio en el vs code

Git status detecta el cambio

Hacemos un git add + archivos o en el caso de que queramos subir todo con pone un punto y enter bastara

Hace un git commit –m “ya terminamos esto “

Hacemos push al fork nuestro git push fork rama/nueva

Vamos al nuestro giyhub clic en la pestaña code , luego click en branches , buscamos el boton pull request que vaya a saber dios donde esta jajajajaa investiguen . Clic en new pull request , caundo es el primer pull request aparece al lado, luego hay que buscarlos por eso no recuerdo , aunqu es raro que hagmaos mas de un pull request.

Verificamos que los campos esten correctamente llenados, escribimos un nombre y reseña clic en create pull request

Esperamos si nos aceptan los cambios

Nos movemos a la rama main git checkout main

Un ves dentro ejecutamos un git log –oneline –all –decorate –graph hechamos un vistaso a ver que todo este correcto el historial

- git remote –v

Traemos los cambios del repositorio original

- git pull origin main

Enviamos el nuevo historial al fork nuestro con

* git push fork main

Hechamos otro vistazo al historial git log –oneline –all –decorate –graph

No se si les aparece a todos lo mimos enm eso que es medio grafico pero se deberia ver que main. Origin main, for main apuntan todos al mismo commits lo cual nos indica que estamos bien sincronizados

Elminamos la rama/nueva

.. git branch –d rama/nueva

Elimanmos del for con

..git push fork –delete rama/nueva

En varios tramos de este paso a paso delante dit aparecen puntos o guiones etc, es porque sino me ponía la g en mayúscula pero no lleva nada delante ningún comando git , siempre inician con la palabra git en minúscula