**CLASE 7 GIT/GITHUB**

(22) 🚀GIT: DIFF y MERGE - Trabajando con BRANCHES (Ramas)🌳 2021 🤩 | Introducción a GIT y GITHUB #7 - YouTube

https://www.youtube.com/watch?v=W8rwULnu9nA

Transcript:

(00:01) y bienvenidos a un nuevo vídeo en donde en todo [Música] por gente en el vídeo de hoy vamos a continuar con esta serie de git ya terminando un poquito con lo último que vamos a ver en ella y vamos a ver un par de comandos que nos van a ayudar a identificar algunas diferencias que pueden haber entre las distintas ramas que tengamos que como nos hablamos anteriormente en el vídeo anterior si no lo viste te lo dejo acá arriba si hablamos de que cada rama puede tener versiones diferentes o archivos diferentes con respecto a otras vamos a

(00:44) ver el dif que es un comando que nos permite ver justamente estas diferencias entre ramas como si también vamos a ver el march que es un comando que nos va a permitir por unir el contenido de ramas que tengamos para que todos queden de una forma unificada por ejemplo cuando tenemos una rama que queremos unificar a nuestro en nuestro máster para que sea la versión oficial de nuestra aplicación así que basta hablar si todavía no te suscribiste por favor dale click a la campanita suscríbete dale a me gusta deja un comentario que nos ayuda

(01:18) muchísimo posicionamos en youtube y seguirnos en nuestras redes sociales así que vamos ahora a ver lo que tiene que ver con el merge el dif y algunos que otros consejitos para tener en cuenta a la hora de trabajar con estas ramas bien gente como pueden ver estoy acá en nuestro repositorio local prueba git que venimos trabajando en clases anteriores sino también es que para poder llegar acá y pueden comprender lo que estamos haciendo te recomiendo que mires las clases anteriores tenemos acá nuestra consola abierta y estamos posicionados

(01:50) en la rama 1 se acuerdan que teníamos rama 1 y master y dependiendo de si cambiamos entre una rama o la otra los archivos dentro del prueba git que era nuestro repositorio local cambiaban si aparecían o desaparecían en este caso estoy en rama 1 fíjense como tengo todos los archivos que habíamos creado de pruebas o la pruebita texto texto 2 pero fíjense que si hago git checkout master me cambio a master desaparecen texto un texto solo y texto 2 que eran los dos archivos extra que yo tenía si vuelvo nuevamente a rama 1 fíjense cómo vuelven

(02:32) a aparecer pero ahora imagínense que no tenemos sólo dos ramas y que vamos a hacer un chequeo rápido y ver esto sino que tenemos no sé cinco seis siete ramas un montón no vamos a estar cambiando rama por rama para ver cuál es la diferencia entre una y otra sino que podemos ocupar un comando que es el comando diff que viene de diferente en inglés que para conocer las diferencias que pueden existir entre dos ramas y en qué consiste o qué información nos da este comando vamos a probarlo fíjense si se colocó git diff tengo que poner luego

(03:07) la primera rama con la que quiero comparar y luego la segunda en este caso por ejemplo quiero comparar con respecto a rama 1 que lo que me va a devolver este comando fíjense apretamos enter y me voy a devolver cuáles son las diferencias entre una y otra fíjense que me dice que master con respecto a rama 1 en rama 1 existe un nuevo archivo fíjense que dice new file llamado texto txt de igual manera dice lo mismo con respecto al archivo texto dos puntos de xp que también nos dice que es un new file fíjense cómo nos va tirando la

(03:46) información y nos dice acá tenés una diferencia entre master y rama 1 y nos va alistando cuáles son las diferencias en este caso nos muestra los dos archivos diferentes que tenemos ahora esto dependiendo del orden en que pongamos las ramas es decir acá yo puse primero masters y luego rama 1 pero si yo invierto este orden van a ver como el mensaje que que recibimos de diferencia de este comando va a ser totalmente distinto fíjense voy a cambiar el orden voy a poner un rama 1 primero y luego master y fíjense cómo hace la

(04:23) comparativa al revés en lugar de decirme qué rama uno tiene un archivo nuevo con respecto a master que es con decimos acá la comparación me está diciendo que master tiene un archivo - es decir un delete file un archivo eliminado que se llama texto y texto 2 fíjense cómo me está haciendo lo contrario acá arriba me decía rama uno tiene dos archivos nuevos con respecto a master ya que quería acá me está diciendo rama uno tiene dos archivos demás con respecto a master como que los tendría que haber eliminado como que master los eliminó y no los

(05:00) tiene sí fíjense cómo es la comparación al revés con respecto al primer caso bien bueno qué vamos a hacer ahora vamos a borrar nuestra línea de comandos y una pregunta es que hay muchísimo sobre todo en el vídeo anterior también donde me consultaban bueno profe listo supongo que estamos trabajando entre muchos tengo varias ramas y quiero unificar el trabajo que acabo de hacer a la rama master que es como sabemos la rama principal de nuestro proyecto con el que estemos trabajando como lo hago bueno esto se hace mediante el comando

(05:33) commerce el comando merz lo que nos permite es unificar dos ramas en una sola sí es decir si yo tenga este caso diferencias entre la rama una rama master que ambas tengan exactamente los mismos archivos y no tengan estas diferencias bien fíjense que lo que hace el marx les voy a mostrar acá a continuación una imagen lo que hace el merge gente es unificar una rama que yo tenga fíjense yo puedo tener acá antes de un gamers yo tenía una rama master sí y en particular a esta rama master en algún momento yo creé una nueva rama por ejemplo nuestra

(06:16) rama 1 y fui creando así nuevos archivos que difieren de lo que más tarde nya en principio pero ahora yo quiero unificar y quiero que ambas tengan exactamente lo mismo en ese caso vamos a utilizar el mar como lo ven acá no tenemos que colocar la rama origen primero es decir de dónde van a ir los archivos o la actualización hacia luego la rama destino es decir la rama que va a recibir esos cambios en este caso tendremos que hacer desde rama 1 hacia nuestra rama más cómo quedaría esto si nos vemos gráficamente después

(06:55) del mar básicamente estos cambios que yo hice por ejemplo en rama 1 van a pasar ahora a estar también a unificarse ya formar parte de mi master de esta manera los dos van a tener exactamente lo mismo si estos cambios que yo hice un rama uno se van a incorporar ahora en mi rama master se entiende bien vamos a probar entonces a hacer esto ahora en la línea de comandos a ver si nos sale y si quedan exactamente igual y en gente como pueden ver ya estoy acá estoy nuevamente en la consola y estamos parados en nuestra rama 1 como lo sabemos porque

(07:30) tenemos acá el archivo texto y texto 2 que difieren con respecto a nuestra rama más fíjense estando parada son rama 1 si tiro el git march fíjense lo que me va a contestar la consola a ver qué pasa me está diciendo que ya está el día que no hay ningún cambio para hacer pero como si yo sé que master tienen los archivos menos que rama 1 y este resultado se produce porque yo estoy parada en la rama que está más actualizada bien yo me tengo que parar siempre cuando voy a hacer un mar en la rama que va a recibir los cambios en ese

(08:09) momento sí entonces en este caso si yo quiero recibir los cambios en master voy a tener que hacer check out master ahora sí ya estoy parada en master fíjense que desaparecieron texto y texto 2 y lo que voy a pasar a hacer es nuevamente el mar va a tener el mismo orden el origen va a ser mi rama 1 que tiene los archivos y que difiere de respecto a master y mi destino va a ser master fíjense en este caso se apretó enter empieza a hacer la actualización fíjense que dice ok team y luego acá abajo dice que se agregaron dos nuevos archivos

(08:52) texto y texto 2 ahora que ya estamos parados del master ya tenemos este cambio y tranquilamente podemos hacer get up y punto para que nos agrega absolutamente todos los cambios y un hit commit y poner un mensajito por ejemplo cambios master y dejar todo guardado bien en este caso ahora fíjense si hago gets check out y me voy nuevamente a rama 1 fíjense se mantienen los mismos archivos si pongo 10 check out y me voy ahora nuevamente a master como master ya tiene estos cambios también aparece exactamente lo mismo bien se va entendiendo hasta hay

(09:37) gente bueno gente eso fue todo por hoy espero que les haya gustado con este vídeo damos por finalizada otra serie en el canal en este caso la de git y git cap obviamente que todos estos fueron fundamentos y elementos básicos pero con esto ya puedes arreglarte aunque sea básicamente en un entorno de desarrollo o en un equipo de desarrollo clásico digamos donde estuvimos trabajando entre distintos compañeros haciendo push haciendo pull haciendo commit haciendo cambios y utilizando diferentes ramas para nuestras aplicaciones obviamente

(10:13) que esto es una base ahora te toca a vos practicar y empezar a meterle muchísima ficha y muchísima garra agarrarle la mano aquí lo máximo posible para poder mejorar y versionar lo mejor posible cada uno de tus proyectos si te gustó esta serie o quieres saber más de git déjanos en los comentarios que comando te gustaría que expliquemos qué extra le podemos agregar a esta serie o qué ideas tenéis para series futuras sobre este tema o sobre otros así que no te olvides que este curso probablemente dentro de poco esté en la todo tú academy lo dejo

(10:45) acá y seguramente vamos a visionar nuestras redes sociales para que estén al tanto y que puedan tener tanto como los otros cursos que ya tenemos anteriormente este con certificado que valide que efectivamente ustedes llegaron al final del curso así que espero que les haya gustado que lo hayan disfrutado y no se olviden de apoyar al canal con me gusta con suscripciones o también con cafecito desde argentina o paypal desde cualquier parte del mundo que nos ayuda un montón a seguir creciendo nos vemos gente en un próximo vídeo en

(11:14) donde en todo chau chau [Música]