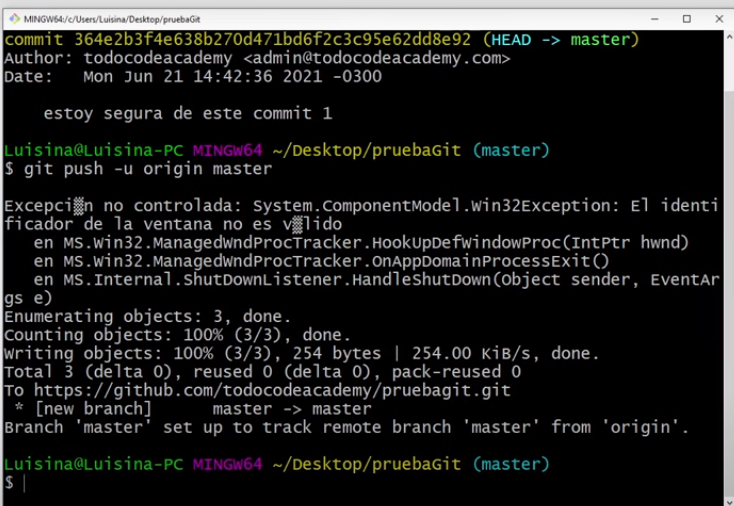
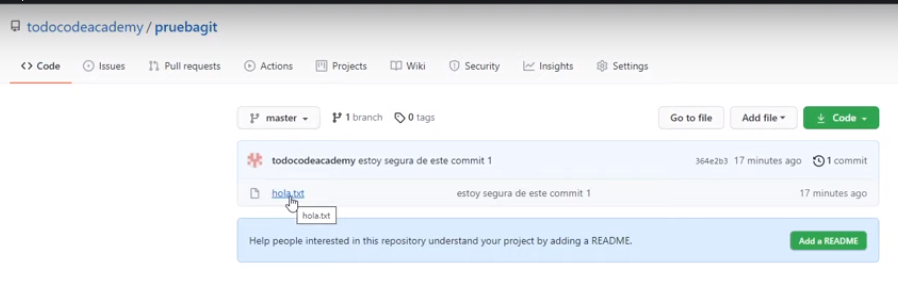
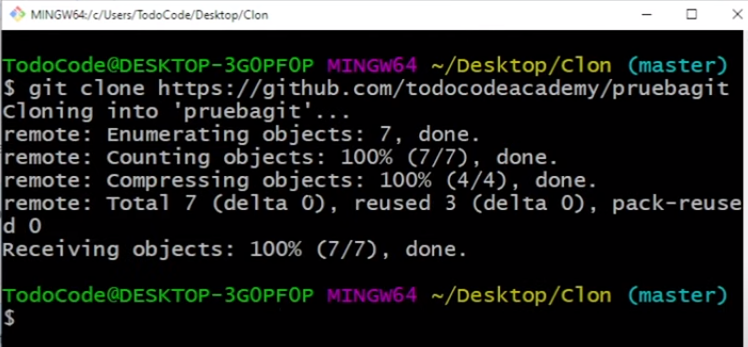
Git y github

mkdir nombreDelRepositorio: para crear una carpeta desde consola

* + Cd url del repositorio
  + git init,
  + git config –global user.name “usuario de github”
  + git config –global user.email “email del usuario”
  + git config –global color.ui auto:
  + git config list  
    ls -la --------sirve para ver el contenido de un repositorio
  + git remote add origin “Url de nuestro repositorio de github”  
      
    cat > índice.txt ------------crear un archivo txt con información adentro  
    Capitulo 1: Introduccion a Git  
    Capítulo 2 : Flujo de trabajo básico  
    Capitulo 3 : Repositorios remotos

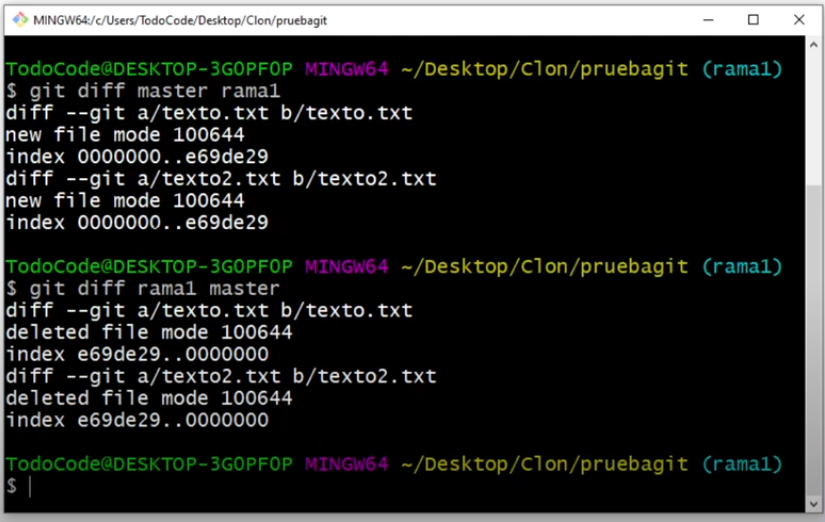
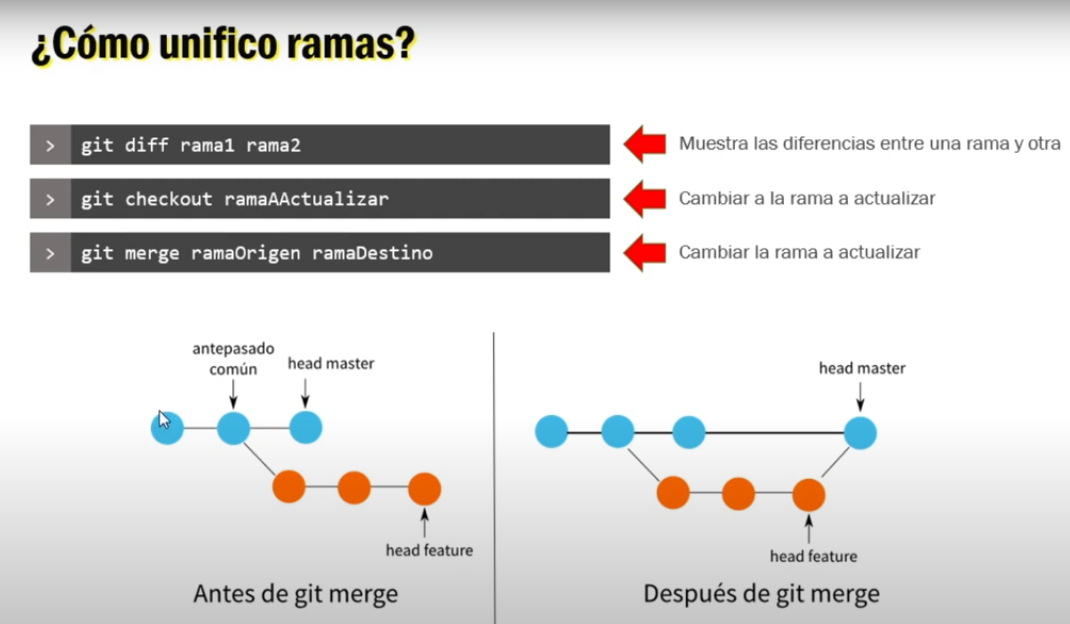
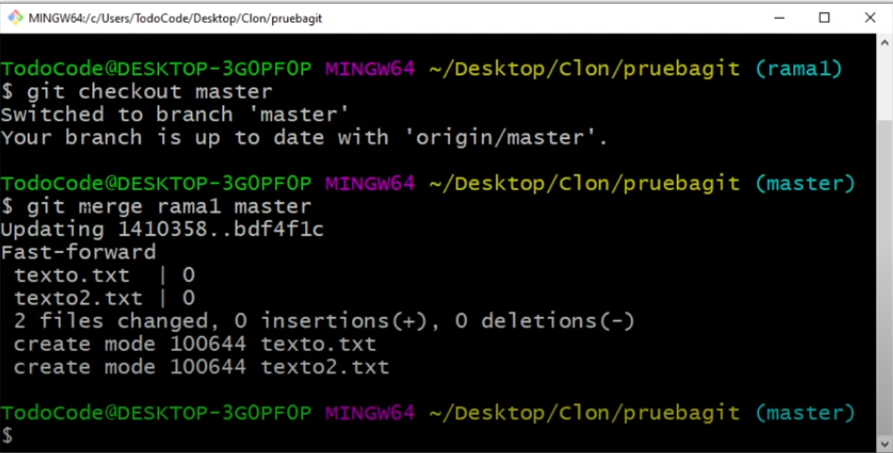
* + git status
  + git add hola.Txt
  + git status
  + git commit -m “Estoy seguro de este commit 1”
  + git push -u origin master ---------aquí tenemos que colocar el nombre de nuestro repositorio remoto
  + Por defecto git nos crea una rama llamada master o main
  + 
  + 



* + Clonar o descargar archivo desde un repositorio remoto:
  + git clone url del repositorio
  + 
  + Copiar solo un cambio y no tener que clonar todo el repositorio
  + git pull origin master
  + git branch “para saber en qué rama estamos parados actualmente”
  + Crear nueva rama:
  + git branch rama1
  + git branch nos mostrara que tenemos dos ramas creadas “master y rama1”
  + cambiarle el nombre a la rama creada: de primeras el nombre de la rama y después por el que la quiero cambiar
  + git branch -m rama1 ramita1
  + Cambiar de rama:
  + git checkout ramita1  
    git branch
  + Eliminar rama: para borrar una rama, no puede estar parado sobre la rama que se va a borrar, toca cambiarse de rama y luego borrarla.  
    git branch -d ramita1
  + git branch

crear archivos en mi rama desde comandos:

touch “texto.txt”

* + Confirmar que los cambios son de la rama que quiero:
  + git add . --------------para que agregue todos los archivos que estamos trabajando en esta rama
  + git commit “Un mensaje” ------------------para confimar, le estamos diciendo que los cambios realizados son exclusivos de la rama1  
      
    desvincular la rama master de la rama remota origin/master que ya no existe
  + git branch --unset-upstream
  + Luego, vincula tu rama master local a la nueva rama remota Master:  
    git branch -u origin/Master master  
      
    git push origin -d rama1 --------para eliminar una rama del remoto
  + git diff master rama1 “Es para conocer la diferencia que pueden existir entre dos ramas, coloco la primera rama con la que quiero comparar, luego la segunda, y nos devuelve cuales son las diferencias entre una y otra”
  + Ahora, esto dependiendo del orden en que pongamos las ramas, van a ver como el mensaje que recibimos de diferencia con este comando van a ser totalmente diferente.
  + 
  + Unificar 2 ramas en una sola, es decir: si yo tengo en este caso diferencias entre la rama1 y la rama master, que ambas tengan los mismos archivos y no tengan estas diferencias.
  + 
  + Se debe primero de parar en la rama que va a recibir los cambios en ese momento, este caso, si yo quiero recibir los cambios en master voy a tener que hacer git checkout master.
  + git merge ramaOrigen ramaDestino
  + (git merge rama1 master)
  + esta diciendo que se agregaron 2 nuevos archivos, texto y texto2,   
    
  + ahora que ya estamos parados en master, ya tenemos este cambio, tranquilamentente podemos hacer:
  + git add . “para que nos agregue absolutamente todos los cambios”  
    y un git commit -m “mensaje de los cambios”  
      
    git checkout rama1 ----------aparacen sus archivos
  + git checkout master -----------tiene los mismos archivos