

Git

Git extensions herramienta visual gratis

Git Kraken herramienta visual muy chula y gratis^^

Principales ramas

- Master, Código en producción
- Development, rama donde se integra la nueva funcionalidad y se prueba antes de pasar a master, se crea a partir de la master
- Features Ramas que contienen funcionalidades. Se crean a partir de development
- Hotfix para corregir errores en producción

## **RAMAS**

Crear rama git checkout -b nombreRama crea y nos cambia a esa rama

Git branch nombreRama crea la rama, no nos mueve a ella

Cambiar nombre a rama ya creada git branch -m nombreRama nombreNuevo

Para eliminar git branch -d nombreRama Si hay commit y archivos en esa rama primero lo mergeamos o la eliminamos con git branch -D nombreRama Ojo que se pierde todo

Para eliminarla del repositorio remoto la rama git push origin --delete fx

## **STATUS, COMMIT, DIFF, STASH**

Git status para ver si falta stagear o comitear, git log --oneline , git log mas info

Descartar unos cambios en vez de comitearlos = git checkout -- archivoAlgnorar

Descartar todos los cambios = git checkout -- . Todos los archivos modificados descartados nLos cambios descartados no se pueden rescatar

Git add . Preparar para commit. el punto significa todos los archivos de esa carpeta

Se puede sacar de esa zona de staged --> git reset HEAD nombreArchivo o . Para todos

Mismo efecto --> git rm --cached nombreArchivo

Para comitear --> Git commit -m "texto explicativo"

Si no esta configurado git pedira git config --global user.mail "you@example.com"

Y git config --global user.name "you Name"

Para subirlos a remoto git push origin nombreRama

Si la rama no existe en el repositorio remoto git push --set-upstream origin nombreRama

En el directorio principal de nuestro proyecto nano .gitignore se nos abra el nano y ignoraremos todos los archivos que no queramos q se suban al repositorio ejemplo \*.txt ignorara los txt ctrl+o para guardar ctrl+o para salir. Si pusieramos /\*txt ignoraria los txt solo en el directorio raiz

Para bajar todos los cambios del repositorio git pull

Git fetch para consultar si hay algo nuevo en el repositorio en remoto

Git diff nombreArchivo para ver que cambios de han modificado

Git stash para guardar en la pila

Git stash list para ver todo lo guardado en la pila

Para sacarlo de la pila git stash apply nombreArchivo que sale detallado en git stash list

Git stash drop para eliminar el ultimo cambio que hicimos (el primero de la lista)

O Git stash drop nombreArchivo para eliminar uno en concreto

## **ERROR COMMIT**

Error commit en rama que no toca!! Estando en la rama equivocada hacemos un git log cogemos el código del commit equivocado, cambiamos a la rama que queriamos hacer el commit en primer lugar, una vez en la rama git cherry-pick códigoDelCommmmit y movemos el commit a nuestra rama y tendríamos que borrarlo en la otra rama.

Git reset head~1 ultimo, cambio ~2 penultimo cambio etc

Se elimina el commit pero no los cambios realizados, si queremos eliminar tambien los cambios

Git reset --hard head~1 eliminaria el código y el commit

## **MERGE**

### **Hay tres posibles merges**

- Fast-forward lo ha mergeado sin problemas
- CONFLICT (content) : pone la rama donde esta el conflicto tenemos que fixear para poder mergear, habra que ir al archivo y coger lo q se quiera desechar lo demas y comitear
- auto-merging. Ejemplo =Creamos desde la rama f4 la rama f6 en la rama f6 tenemos todos los archivos de f4, modificamos un archivo por ejemplo ultima linea , añadimos stage y comiteamos. Volvemos a la rama f4 y ahcemos un cambio en ese mismo archivo en la primera línea añadimos en el stage y comiteamos. Ahora mergeamos nos saltara un editor para que comentemos lo que hicimos en el código, y para salir :wq como no ha interferido un commit con el otro no hay problema alguno ya que las líneas estan en el mismo archivo pero separas

En la rama donde queramos mergear --> git merge nombreRama

Tambien podriamos abortarlo con la instruccion -->git merge --abort

## **TAGS**

Git tag -a nombreTag -m"mensaje"

Git tag -l para ver los tags creados

Si queremos añadirle a un commit un tag, buscamos con git log y git tag nombreTag códigocommit

Borrar --> git tag -d nombreTag

Para subir tag a origin --> git push origin nombreTag para subir topes -->git push --tags

## **REPOSITORIO NUEVO EMPEZANDO DESDE LOCAL**

Creamos los archivos que deseemos git init para inicializarlo, git add . Para agregarlos todos al staged git commit -m "mensaje" para comitear acto seguido conectar con github o cualquiera

Git remote add origin nombreRepositorio , git push para subir la rama ya que no estara en el repositorio remoto git push --set-upstream origin nombreRama nos pedira el login y password si es la primera vez que nos conectamos y listo

### **REPOSITORIO NUEVO EMPEZANDO DESDE REPOSITORIO NUEVO**

1 cread la carpeta donde va a estar el proyecto

2 la primera linea sera esta = git clone nombreProyecto

3 entramos en la carpeta proyecto = cd nombreCarpetaBajadaOProyecto/

4 para obtener la información de todas las ramas del repositorio remoto = git fetch origin

5 comprobamos las ramas (estarán en rojo es normal ) = git branch -a

6 Cambiaros de rama para ver si la informacion en el main es diferente = git checkout main

7 Escribid algo en vuestra rama en el index.html por ejemplo = <p>ultima prueba</p> ctrl + s (guardad los cambios) git add . > git commit -m "mensaje" > git push

8 comprobadlo en el github!

### **PUEDE CAMBIAR DE SERVIDOR EL REPOSITORIO**

- Primero debemos desconectarlo del antiguo git remote remove origin
- Para luego conectarlo con el nuevo (Se vera en github u otros)