

DOCUMENTACIÓN

Ejercicio 1 (ejercicios de creación y actualización de repositorios):

- Creamos un archivo indice.txt cun contenido.
- Comprobamos o estado do repositorio con git status, veremos os cambios en rojo.
- Con git add * ou git add indice.txt (* para añadir todos os arquivos, si solo queres un archivo ou carpeta indicas o nome ou path) añadimos os cambios a zona de intercambio temporal.
- Volvemos a facer git status e agora os cambios están en verde.
- Ahora para confirmar os cambios temos que facer un commit, neste caso git commit -m "Añadido índice del libro."
- Volvemos a facer git status e vemos como os cambios desaparecieron.
- Añadimos novo contenido a indice.txt
- Facemos outro git status para comprobar os cambios.
- Git add * para añadir todos os cambios (o tambien git add indice.txt).
- Para confirmar os cambios facemos git commit -m "Comentario".
- Para ver os cambios da última versión con respecto a versión anterior hacemos git show e aparecerános os cambios, os da anterior versión en rojo e os novos en verde.
- Hacemos git commit - -amend -m "Comentario", para refacer o último commit con un comentario novo.

Ejercicio 2 (ejercicios de manejo de historial de cambios):

- Mostraremos o historial de commits con git log (no meu caso git lg, porque o configurei cun comando anteriormente).
- Creamos a carpeta capitulos con un archivo capitulo1.txt e o seu contenido
- Facemos un git add * e git commit -m "Comentario" para añadir e confirmar os cambios.
- Facemos outro git log (git lg no meu caso) para comprobar que aparece o commit que acabamos de realizar.
- Creamos capitulo2.txt con su contenido en capitulos.
- Facemos un git add * e git commit -m "Comentario" para añadir e confirmar os cambios.
- Mostramos as diferencias entre a ultima e a antepenúltima version co comando git diff HEAD~2 . . HEAD
- Creamos capitulo3.txt con su contenido en capitulos.

- Facemos un `git add *` e `git commit -m "Comentario"` para añadir e confirmar os cambios.
- Facemos outro `git log` (`git lg` no meu caso) para comprobar que aparece o commit que acabamos de realizar.
- Mostramos as diferencias entre a primeira e a ultima version con `git diff <hash da primeira version>...HEAD`
- Añadimos contenido a `indice.txt`
- Facemos un `git add *` e `git commit -m "Comentario"` para añadir e confirmar os cambios.
- Vemos quen foi o que fixo os cambios con `git annotate indice.txt`

Ejercicio 3 (ejercicios de desfacer cambios):

- Eliminar a última linea de `indice.txt` comprobar o estado do repositorio con `git status`.
- Desfacemos os cambios con `git checkout -- indice.txt`. Volvemos a comprobar o estado do repositorio con `git status` e confirmamos que a última linea non foi eliminada.
- Volvemos a eliminar a última linea de `indice.txt`. Facemos un `git add indice.txt` para añadir os cambios a zona de intercambio temporal.
- Desfacemos os cambios con `git reset indice.txt` e despois `git checkout -- indice.txt`. Por ultimo confirmamos que a última linea non foi eliminada.
- Eliminar a última linea de `indice.txt` e o fichero `capitulo3.txt`. Facemos un `git add *` e `git commit -m "Comentario"` para añadir e confirmar os cambios.
- Facemos outro `git log` (`git lg` no meu caso) para comprobar que aparece o commit que acabamos de realizar.
- Para desfacer o commit utilizamos o comando `git reset --soft HEAD~1`
- Facemos un `git status` e podemos ver que os cambios estan en verde, e decir, estan añadidos a zona de intercambio temporal.
- Facemos outro commit con `git commit -m "Comentario"`. E comprobamos que se fixo o commit con `git status` e `git log` (`git lg` no meu caso).
- Volvemos a desfacer o commit con `git reset --hard HEAD~1` con esto volveriamos a version anterior.

Ejercicio 4 (gestión de ramas):

- Creamos unha nova rama con `git branch bibliografia` e mostramos todas as ramas existentes con `git branch -av`.
- Creamos o arquivo `capitulo4.txt` con seu contenido dentro da carpeta `capitulos`. Facemos un `git add *` e `git commit -m "Comentario"` para añadir e confirmar os cambios.

- Mostramos os commits de todas as ramas con `git log --graph --all --oneline`.
- Posicionamonos na rama con `git checkout bibliografia` e creamos o ficheiro `bibliografia.txt` co seu contenido.
- Facemos un `git add *` e `git commit -m "Comentario"` para añadir e confirmar os cambios.
- Mostramos os commits de todas as ramas con `git log --graph --all --oneline`.
- Para fusionar a rama `bibliografia` con `main` primeiro temos que posicionarnos en `main` con `git checkout main` e despois `git merge bibliografia`. Con isto a os cambios da rama `bibliografia` añádironse a `main`.
- Mostramos os commits de todas as ramas con `git log --graph --all --oneline`.
- Agora eliminamos a rama `bibliografia` con `git branch -d bibliografia`.
- Mostramos que foi eliminada con `git log --graph --all --oneline`.
- Volvemos a crear a rama `bibliografia` con `git branch bibliografía` e posicionamonos nela con `git checkout bibliografia`.
- Modificamos o ficheiro `bibliografia.txt` e facemos un `git add *` e `git commit -m "Comentario"` para añadir e confirmar os cambios.
- Agora cambiamos a rama `master` e modificamos o arquivo `bibliografia.txt` con un contenido diferente ao da rama `bibliografia` e facemos un `git add *` e `git commit -m "Comentario"` para añadir e confirmar os cambios.
- Agora fusionamos as ramas con `git merge bibliografia`, esto fara saltar un conflicto o cal teremos que resolver eliminando o texto que este igual para crear a mellor version.
- Unha vez solucionado o conflicto confirmamos a resolucion do conflicto con `git commit -m "Comentario"` e vemos os commits de todas as ramas con `git log --graph --all --oneline`.