Fusión de Ramas

Una rama viene a ser una división de tu trabajo actual con el objetivo de poder realizar en esa nueva división quizás algunos cambios que todavía no está seguro que formen parte de la rama principal.

Merge fast-forward: Se hace cuando la rama secundaria no interfiere con el contenido de la rama principal. Según la teoría indica, este tipo de merge al detectar que no hay ningún cambio sobre nuestra rama principal de forma transparente puede hacer la unión de la rama secundaria con la rama inicial.

Parados sobre nuestro proyecto (por ej.: index.html, estilos.css y app.js), creamos un nuevo repositorio local.

git init

Mando a seguir y hago primer commit (en rama master) de todos los archivos de mi proyecto.

git add .

git commit -m "Primer commit"

Luego, agrego un nuevo archivo: nuevaFuncionalidad.html veo que todavía git no le hace ningún seguimiento.

git s

Para crear una nueva rama (para hacer cambios sin afectar lo q veníamos haciendo).

git branch nuevaFuncionalidad

Para ver de forma resumida y visual todas las ramas que tengo, en q rama estoy parado con el HEAD y los mensajes de los commit hechos.

git log --oneline --decorate --all –graph

Cambiar a la nueva rama

git checkout nuevaFuncionalidad

Repito el git log para ver q ahora el HEAD apunta a la nueva rama.

git log --oneline --decorate --all --graph

Mando a seguimiento(o sea, al área temporal) a todos los archivos de mi proyecto (ahora se incluye otro.html)

git add .

Mando los cambios al repo local

git commit -m "Primer commit de esta rama"

Ahora vuelvo a cambiar a la rama master

git checkout main

Veo q desaparece otro.html

Supongamos ahora q la nueva rama funciona muy bien y queremos integrarla a la rama principal. Para eso hacemos un merge (fusión) de ambas ramas.

git merge nuevaFuncionalidad

Ahora ya podemos eliminar la rama nuevaFuncionalidad ya que no la usamos más.

git branch -d nuevaFuncionalidad

Verificamos con

git log --oneline --decorate --all –graph

Creamos ahora un nuevo archivo nuevaOpcion.html.

Ahora vamos a crear una nueva rama y que se dirija a ella automáticamente

git checkout -b ramaOpcion

Verificamos

git log --oneline --decorate --all –graph

Vuelvo a la rama master

git checkout master/main

Fusiono las ramas

git merge ramaOpcion

Muestra el mensaje *“Already-up-to-date”* porque no hicimos commit del nuevo archivo creado

Entonces, voy a la rama opción y modifico el archivo nuevaOpcion.html (pongo el texto "html")

git checkout ramaOpcion

Mando los cambios al repo local

git add .

git commit -m "Nueva opcion"

Ahora vuelvo a hacer el merge de ambas ramas

git checkout master

git merge ramaOpcion

Verificamos

git log --oneline --decorate --all –graph

Elimino la rama que ya no uso

git branch -d ramaOpcion

Conflictos: Ocurre cuando lo que estamos modificando

Tanto en la rama principal como en la rama secundaria afectan a la misma parte de un mismo archivo. Al momento de querer hacer un merge vamos a tener un conflicto en el cual debemos resolver antes de seguir prosiguiendo con git.

git init

Estando en el archivo nuevaOpcion.html, le agrego un body al html que ya traía de antes.

Luego, estando parado en la rama master, hago:

git add .

git commit -m "se agrego body"

Creo una nueva rama y voy a ella

git checkout -b nuevaRama

Modifico ese mismo archivo para agregarle un head y footer y mando los cambios al repo local

git add .

git commit -m "se agrego head y footer"

Cambiamos a la rama principal

git checkout master

Voy a modificar el mismo archivo para agregarle title y mando los cambios al repo local

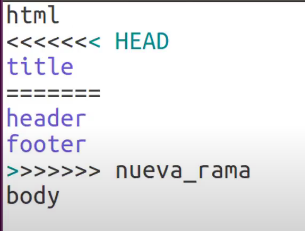
git add .

git commit -m "se agrego title"

Ahora intento hacer un merge entre las ramas

git merge nuevaRama

Nos tira un conflicto!



Entonces vamos al archivo y dejamos la estructura final que queremos.

*html*

*title*

*header*

*body*

*footer*

Luego hacemos la fusión

git merge nuevaRama

vemos el mensaje del ultimo commit

git l

Verificamos

git log --oneline --decorate --all –graph