Taller de GIT

Sesión 2

Nixvoid

Recordando los Comandos básicos

```
$git init

$git status

$git add

$git commit

$git log

$git checkout
```

Arreglar Commits

```
$git commit --amend
```

Deshacer Commits

```
$git reset
$git reset --hard
$git reset --soft
```

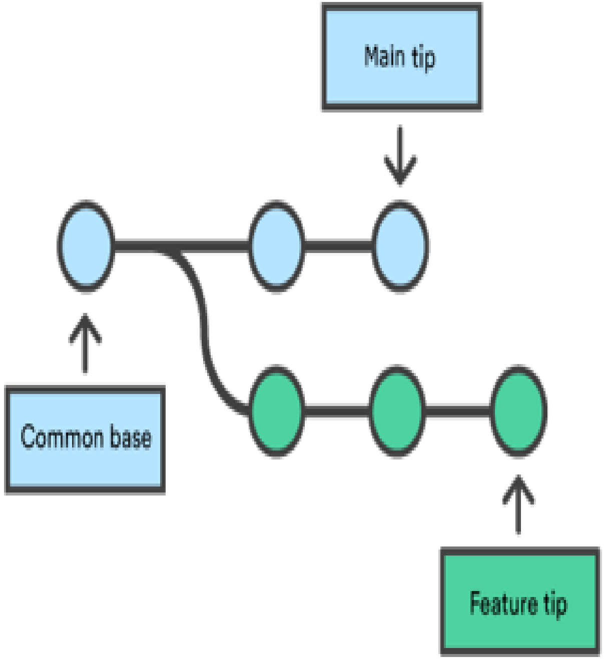
Git Diff

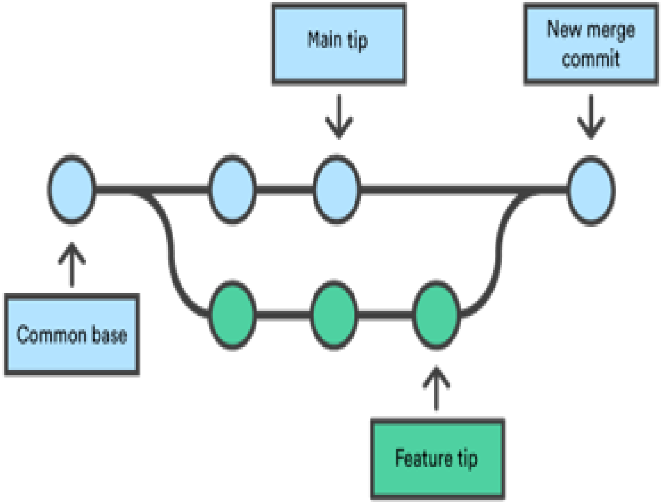
```
$git diff
```

Ejemplo

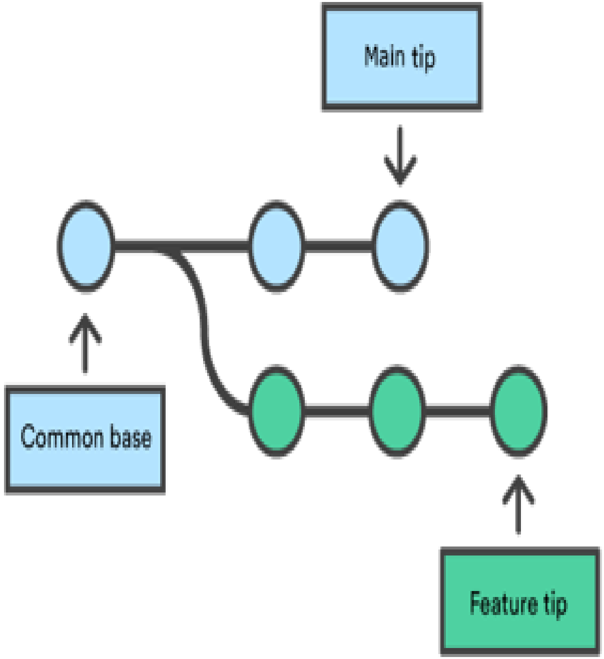
```
$git diff
diff --git a/first_file.txt b/first_file.txt
 index 0e3da6d..fe563b0 100644
 --- a/first_file.txt
 +++ b/first_file.txt
 @@ -1,4 +1,4 @@
 Este es mi primer mensaje:
 "Hola"
 Este es mi segundo mensaje:
 -"Versionar es muy divertido"
 +"Versionar es muy muy divertido"
```

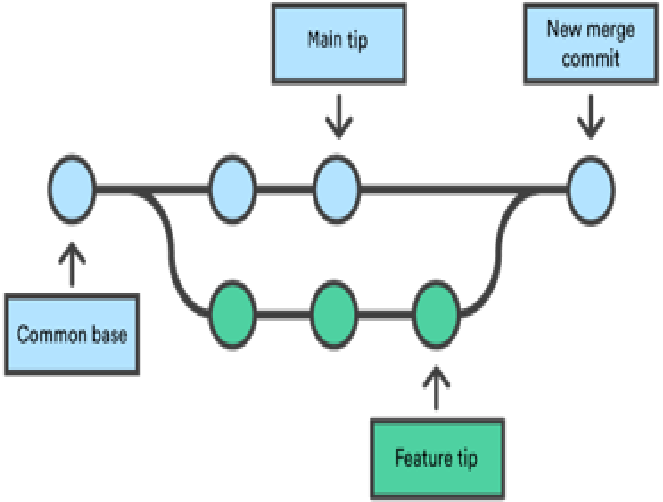
Git merge


Main tip
Common base
Feature tip


Main tip
New merge commit
Common base
Feature tip

Git merge

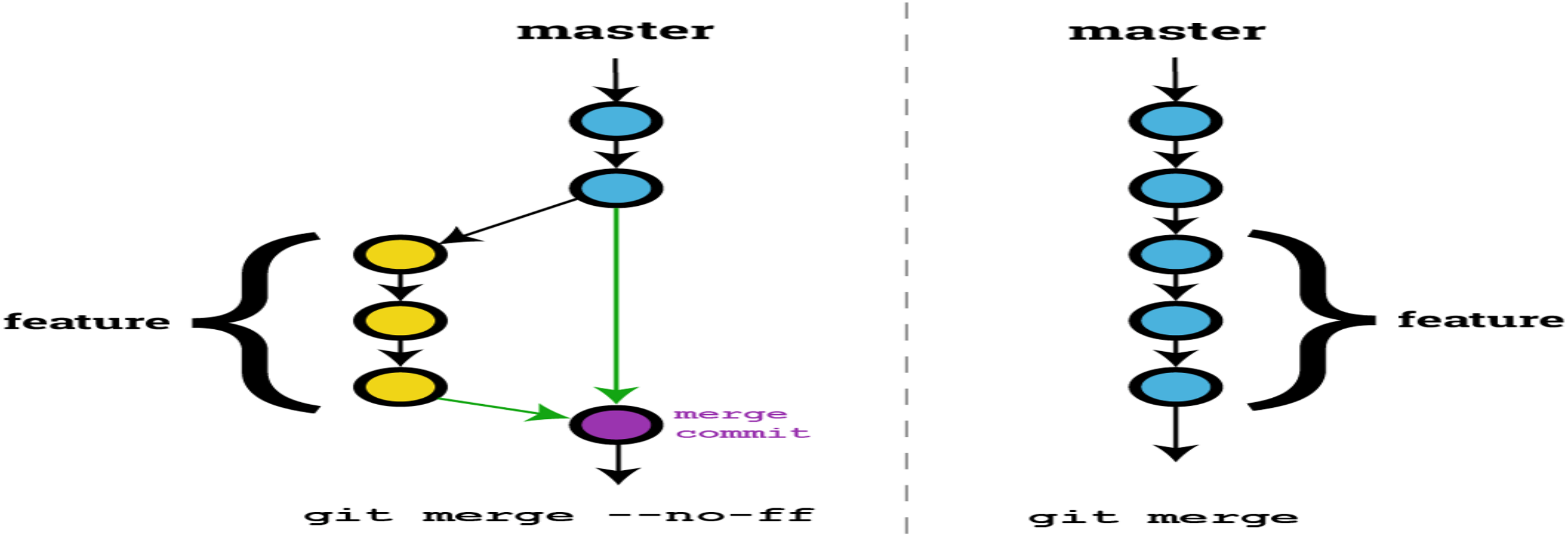

Main tip
Common base
Feature tip


Main tip
New merge commit
Common base
Feature tip

Tipos de Merge

- Fast Forward Merge

- 3-Way-Merge

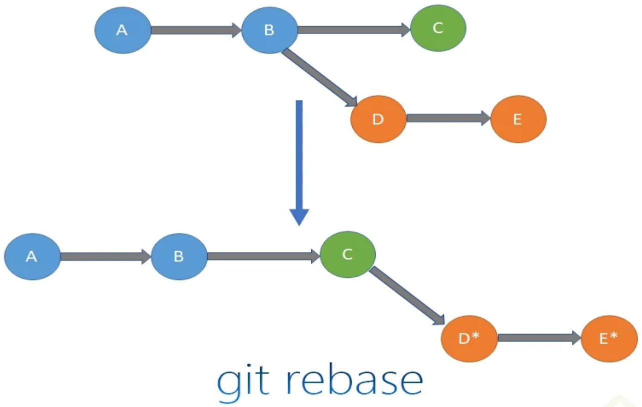


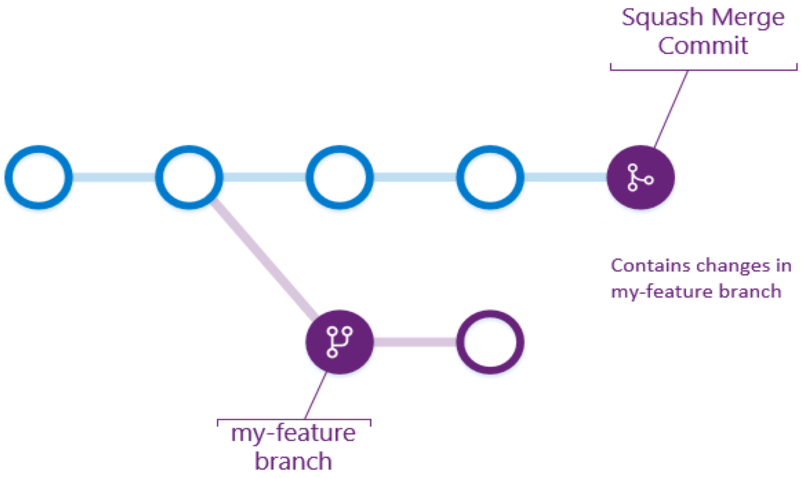
```
git merge --no-ff
```

```
git merge
```

Tipos de Merge

- Squash Merge
- Rebase & Merge





Comandos para hacer Merge

```
$git checkout master / ForMergeBranch

$git merge MergingBranch

$git branch -d MergingBranch

$git merge --abort

$git merge --no-ff

$git merge --squash
```

GitHub - GitLab - BitBucket

- GitHub Link
- GitLab Link
- BitBucket Link

Clonar un repositorio en GitHub

- Desde linea de comandos vía https

```
$git clone https://github.com/<userName>/<repoName>.git
```

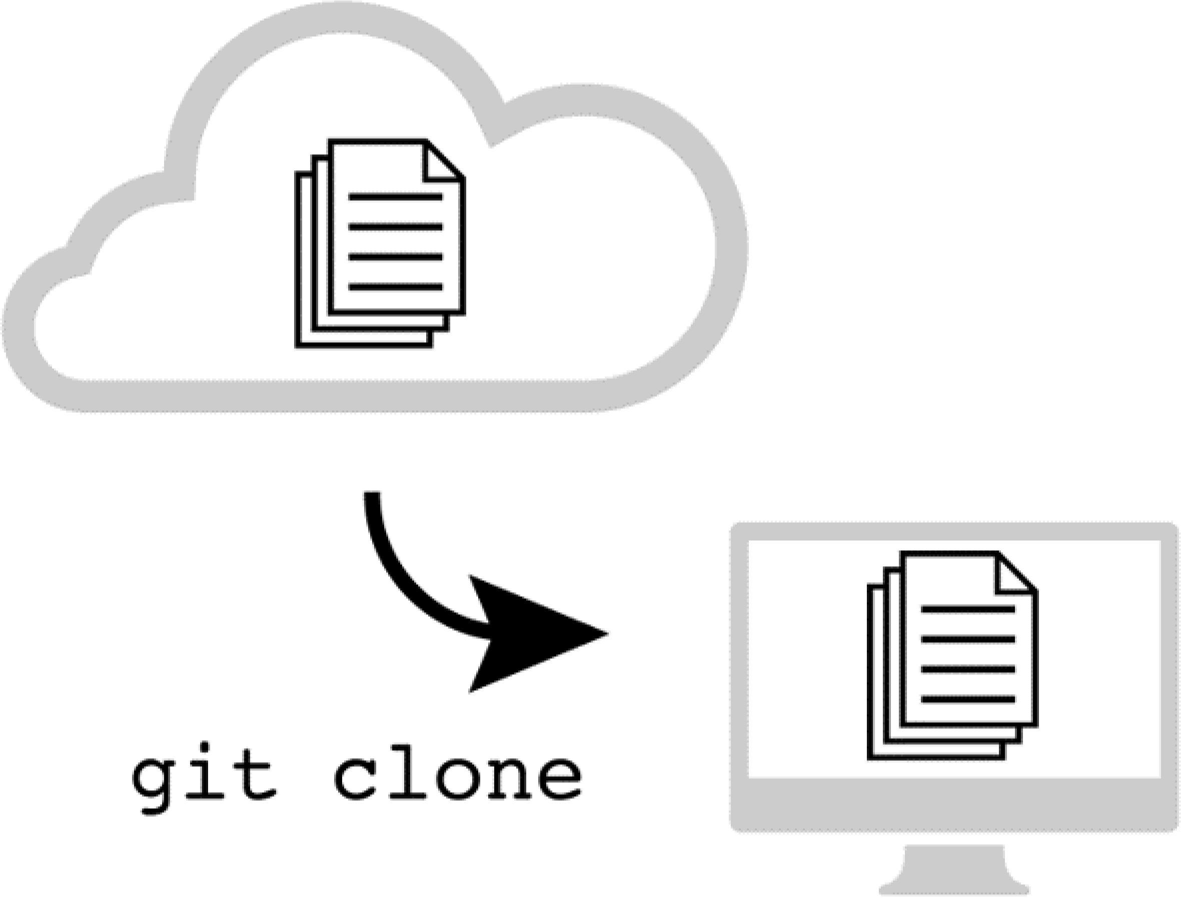
- Desde linea de comandos vía SSH

```
$git clone git@github.com:<userName>/<repoName>.git
```

- Usando githubDesktop

- Descargar el repositorio (comprimido)

- Usando GitHubCLI

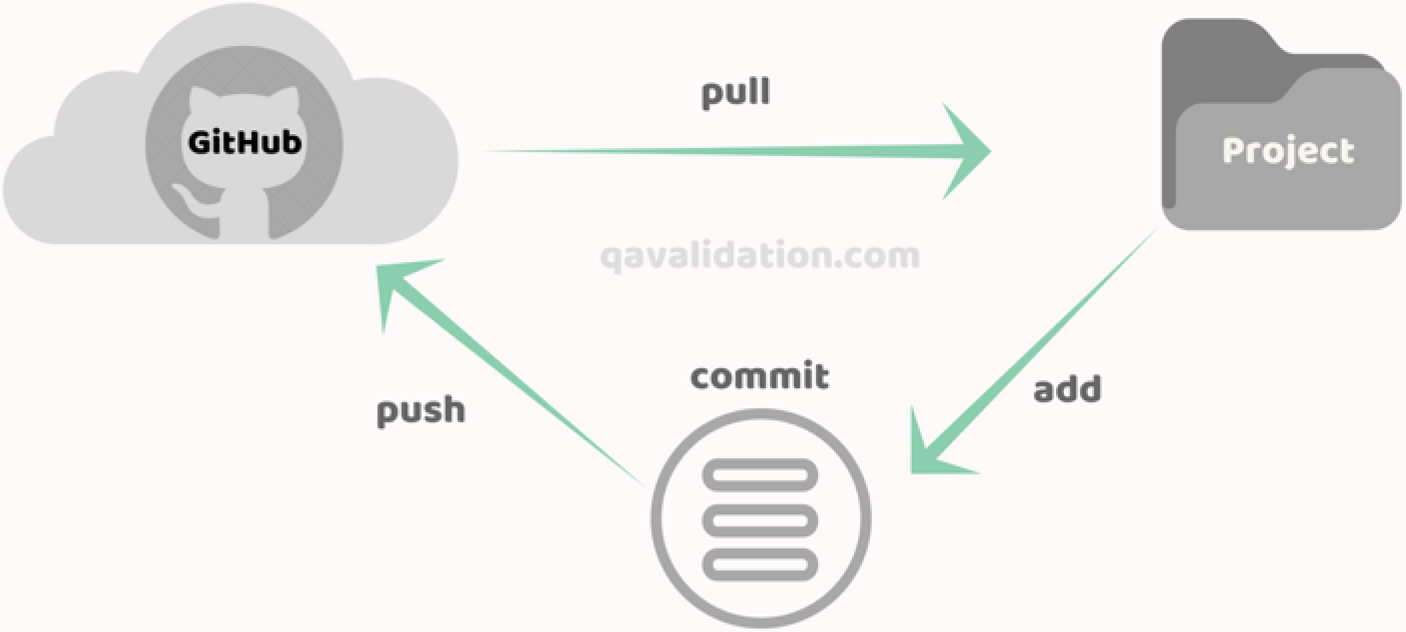


```
git clone
```

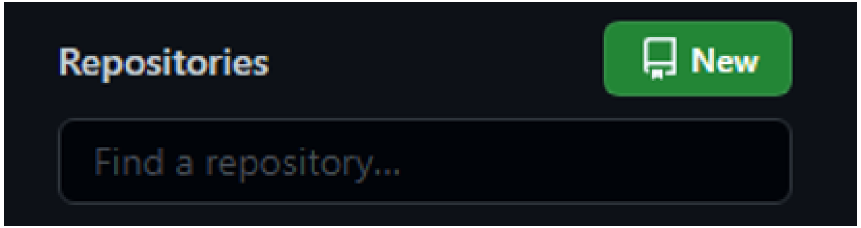
Comandos para trabajar con remotos

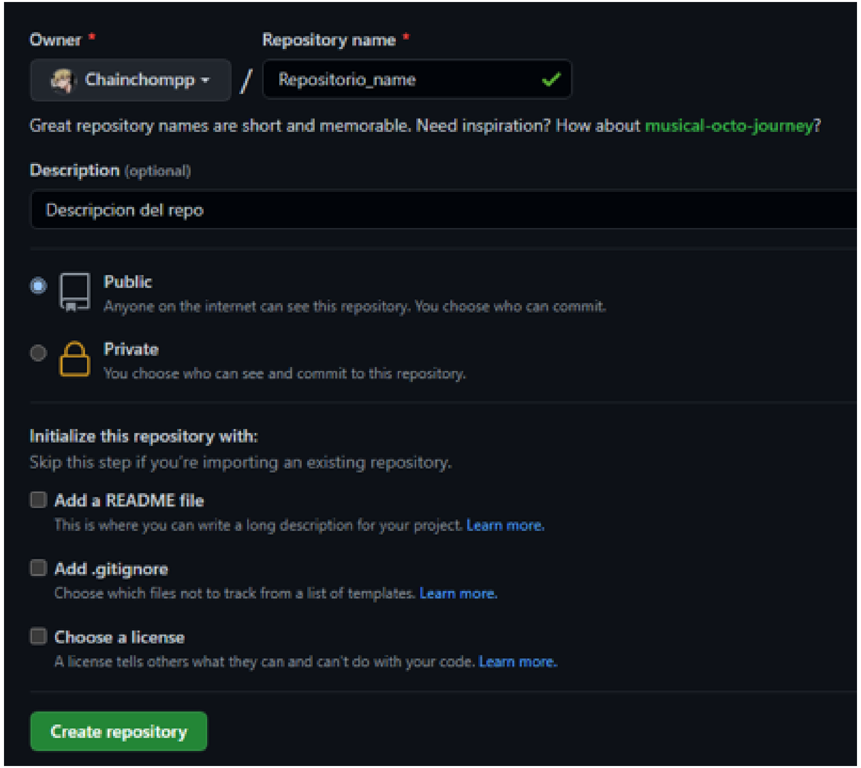
```
$git remote add <remoteName> <remoteUrl>

$git remote

$git remote remove <remoteName>

$git remote rename <oldName> <newName>

$git push <remoteName> <branchName>

$git pull

$git fetch
```

Git PUSH PULL



Crear repositorios en GitHub





Owner * Repository name *

Chainchompp / Repositorio_name ✓

Great repository names are short and memorable. Need inspiration? How about musical-octo-journey?

Description (optional)

Descripcion del repo

● Public
Anyone on the internet can see this repository. You choose who can commit.

○ Private
You choose who can see and commit to this repository.

Initialize this repository with:
Skip this step if you're importing an existing repository.

□ Add a README file
This is where you can write a long description for your project. Learn more.

□ Add .gitignore
Choose which files not to track from a list of templates. Learn more.

□ Choose a license
A license tells others what they can and can't do with your code. Learn more.

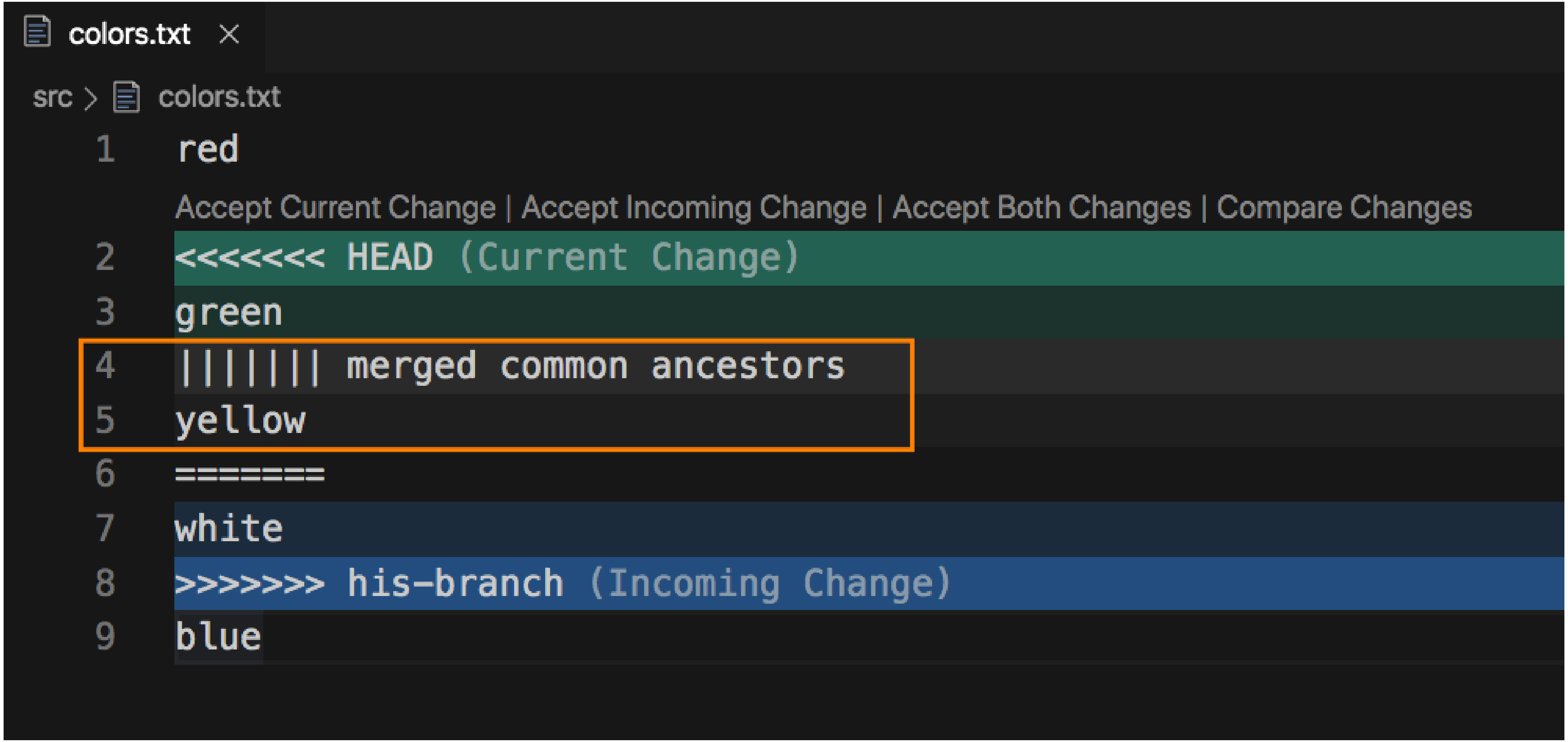
Cambiar ramas remotas

```
$git branch -m oldName newName

$git push --set-upstream remoteName newName

$git push origin --delete oldName
```

Arreglar commits remotos

```
$git revert
```

Git Merge Conflicts



colors.txt
src > colors.txt
Accept Current Change | Accept Incoming Change | Accept Both Changes | Compare Changes

```
red
<<<<<<< HEAD (Current Change)
green
||||||| merged common ancestors
yellow
=======
white
>>>>>>> his-branch (Incoming Change)
blue
```