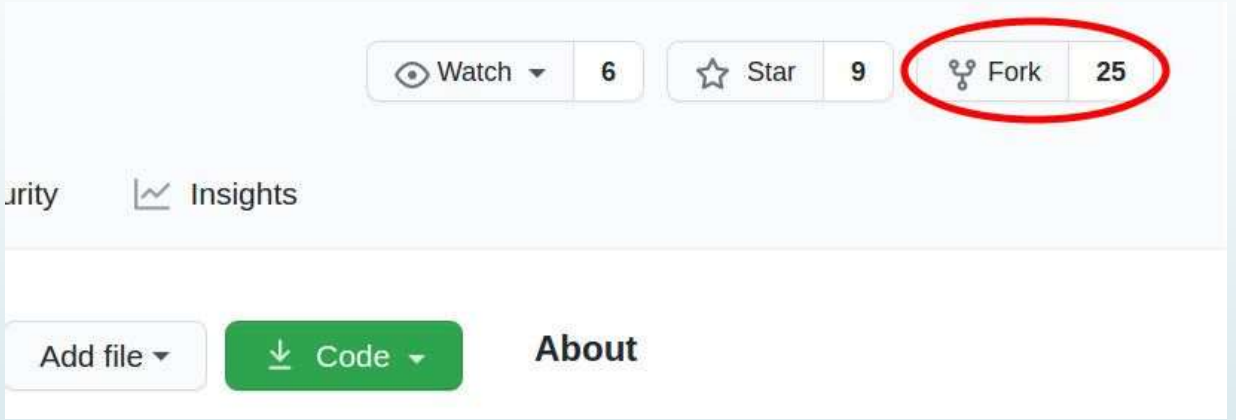
GIT

PULL REQUEST

- Son la forma de contribuir a un proyecto, por ejemplo:
- N usuario llamado Jonathan realiza un fork de un repositorio de Araceli y le efectúa algunos cambios. Ahora Jonathan puede hacer un pull request a Araceli, pero dependerá de ella aceptar o declinarlo. Es como decir: "Araceli, ¿podrías por favor extraer (pull) mis cambios?"

- Primero se envía ese cambio a su repositorio público para que otros lo vean
- `git push https://git.ko.xz/project master`
- `git request-pull v1.0 https://git.ko.xz/project master`

FORK



Para realizar pull request, se realiza un Fork del repositorio.

Esto creará una instancia del repositorio completo en tu cuenta.

REBASE

- Realizar un rebase a una rama (branch) en Git es una forma de mover la totalidad de una rama a otro punto del árbol. El ejemplo más simple es mover una rama más arriba en el árbol.
- Digamos que tenemos una rama que se separó de la rama master en el punto A:

```
        /o-----o---o--o-----o--------- branch
--o-o--A--o---o---o---o----o--o-o-o--- master
```

- Cuando se realiza rebase se puede mover así:

```
                                      /o-----o---o--o-----o------ branch
--o-o--A--o---o---o---o----o--o-o-o master
```

STASH

- Permite almacenar temporalmente una captura de tus cambios sin enviarlos al repositorio.
- Esta funcionalidad es útil cuando has hecho cambios en una rama que no estás listo para realizarle commit, pero necesitas cambiar a otra rama.
- Para guardar cambios en el stash:

```
git stash save "mensaje opcional"
```

- Ver los cambios guardados en el stash:

```
git stash list
```

- Recuperar los cambios guardados en el stash:

`git stash apply` NOMBRE-DEL-STASH

- Borrar los cambios guardados:

`git stash drop` NOMBRE-DEL-STASH

CLEAN

- Se usa para limpiar los archivos sin seguimiento en el repositorio . Si queremos eliminar los archivos no deseados, podemos usar el comando de limpieza en Git.

git clean –n: Muestra los archivos a eliminar sin hacerlo.

git clean –f: Elimina los archivos sin seguimiento del directorio actual, excepto las carpetas sin seguimiento o los archivos especificados con gitignore.

git clean –x: Actuará sobre todos los archivos ignorados.

git clean –xf: Eliminará los archivos sin seguimiento del directorio actual, así como cualquier archivo que Git generalmente ignora.

BRANCHES

Los desarrolladores que trabajan en un proyecto pueden trabajar en diferentes ramas antes de fusionar sus cambios con el código original o la rama principal.

git branch: Para ver los nombres de las sucursales locales.

git Branch –r: Para ver los nombres de las sucursales remotas.

git Branch –a: Para ver todas las sucursales locales y remotas.

MERGE

- Fusionará cualquier cambio que se haya hecho en la base de código en una rama separada de tu rama actual como un nuevo commit.

- La sintaxis es la siguiente:

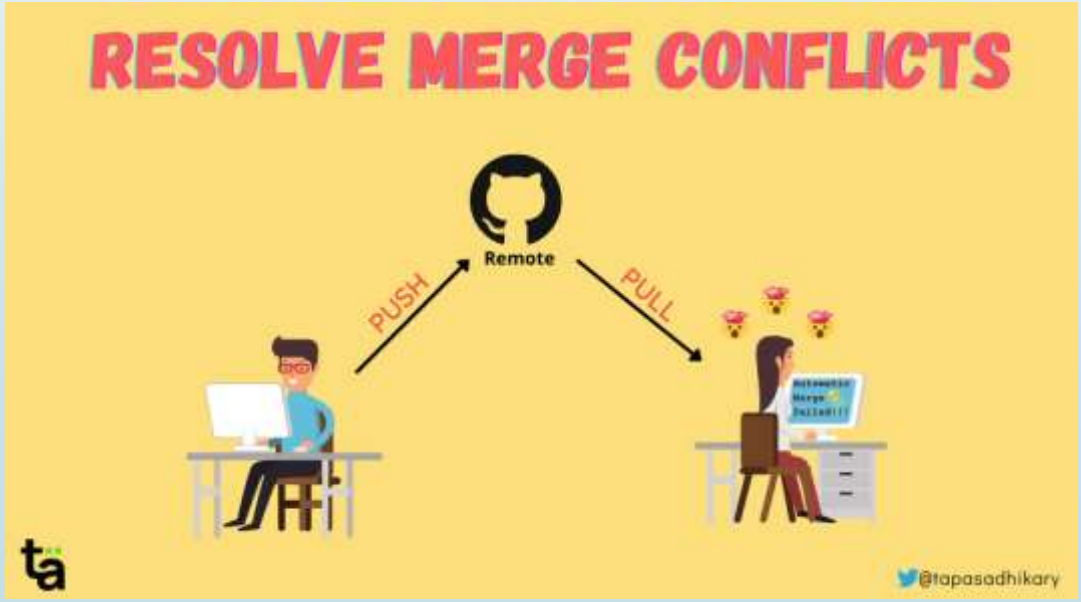
git merge Nombre de Rama.

Por ejemplo, si estás trabajando actualmente en una rama llamada N y deseas fusionar los nuevos cambios que se hayan realizado en una rama llamada X ejecutarías el siguiente comando:

git merge X

CONFLICTS

- Son problemas causados, principalmente cuando 2 desarrolladores o más están editando los mismos archivo o directorios
- Estos conflictos forzosamente se deben resolver de manera manual
- Por lo regular se hacen presentes cuando hacemos un pull, rebase o merge.



CHERRYPICK

- Básicamente es una técnica mediante la cual, podemos traer commits específicos desde otra rama, pero sin la necesidad de unir sus cambios o fusionar ramas en 1 sola.
- Es usado a menudo cuando surgen errores en ramas base alternas como DEVELOP o producción.
- Es usada en casos muy específicos, pero igual depende del equipo de desarrollo.