

# ALGUNOS CONCEPTOS PARA COMENZAR A APRENDER GIT



## FORK

Es una función de git que permite crear una copia de un determinado repositorio en nuestra cuenta de usuario.

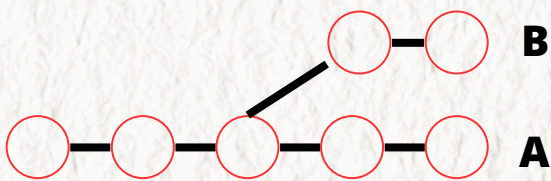
Suele utilizarse para proponer modificaciones en el proyecto de un tercero sin permitir la edición directa, o bien, para usar el proyecto de un tercero como base para uno propio.

## PULL REQUEST

Como se mencionó, hacer un fork, permite que seamos partícipes de los proyectos de otras personas. Pero todos los cambios que hagamos los haremos dentro de una copia del proyecto original. Para poder proponer aplicar los cambios que hagamos, necesitaremos hacer un pull request.

Un pull request, es una solicitud para que el código modificado pueda fusionarse con el código del repositorio original. Los cambios solo podrán verse reflejados una vez que la solicitud haya sido aprobada.

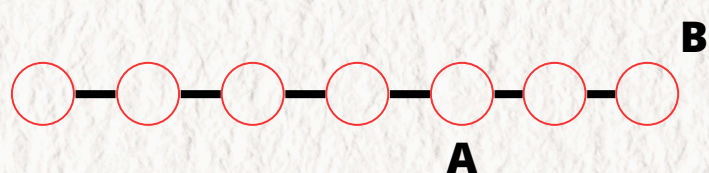
## REBASE



Si deseas incorporar los cambios de la rama A a la rama B. Usando el comando `git rebase <rama-base>`, donde la rama base será la rama en la que deseas incorporar los cambios, lograras que el historial de tu rama B se mueva directo al último commit de tu rama principal.

Git rebase nos permite reorganizar el historial de commits en una rama de forma lineal.

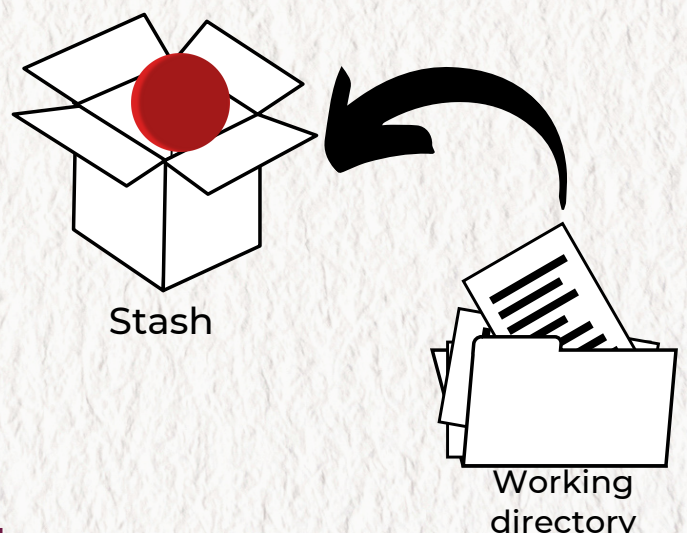
Supongamos que tienes una rama (B) separada de tu rama A.



## STASH

El stash es un área de git donde se puede almacenar temporalmente una captura de los cambios sin enviarlos al repositorio.

Cuando ejecutas el comando `git stash` se guardarán los cambios y se revertirá el directorio a como se veía en el último commit.



## DIFERENTES COMANDOS CON STASH

El stash tiene diferentes comandos para trabajar con el:

- `git stash list`: nos permite ver la lista de modificaciones guardadas en el stash.
- `git stash show`: muestra un resumen del stash.
- `git stash pop`: Elimina o desecha el último cambio guardado en el stash.
- `git stash apply`: permite seleccionar y sacar algún cambio del stash y aplicarlo al repositorio local.



# ALGUNOS CONCEPTOS PARA COMENZAR A APRENDER

# GIT



## CLEAN

El comando `git clean` elimina de manera recursiva los archivos que no están bajo control de versiones en el directorio actual.

## DIFERENTES OPCIONES CON GIT CLEAN

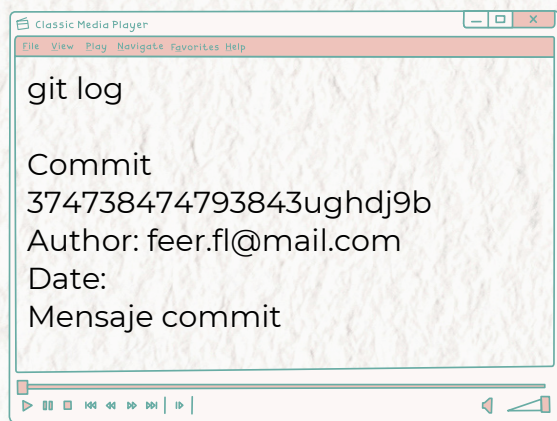


- `git clean -n`: realizara una ejecucuón en seco, es decir, mostrará qué archivos se eliminarán, sin eliminarlos realmente. Permite saber qué es lo que se eliminará.
- `git clean -f`: Cuando git se niega a eliminar archivos, se puede forzar esta aliminación a través de este comando.

## CHERRY-PICK

`git cherry-pick` es un comando en git que permite seleccionar commits arbitrarios de alguna rama y aplicarlos en otra.

Es de mucha utilidad este comando por ejemplo cuando se desean corregir errores, supongamos que se hizo un commit en la rama equivocada, puedes tomar ese commit y moverlo a la rama correcta.



Para poder ejecutar un cherry-pick, necesitaremos los identificadores de los commits que vamos a aplicar. Para obtenerlos, revisamos el log.

Nos posicionamos en la rama donde queremos aplicar el commit y ejecutamos:

```
git cherry-pick 374738474793843ughdj9b
```

Si se llegan a encontrar conflictos, podemos ejecutar el comando:

```
git cherry-pick --abort
```