

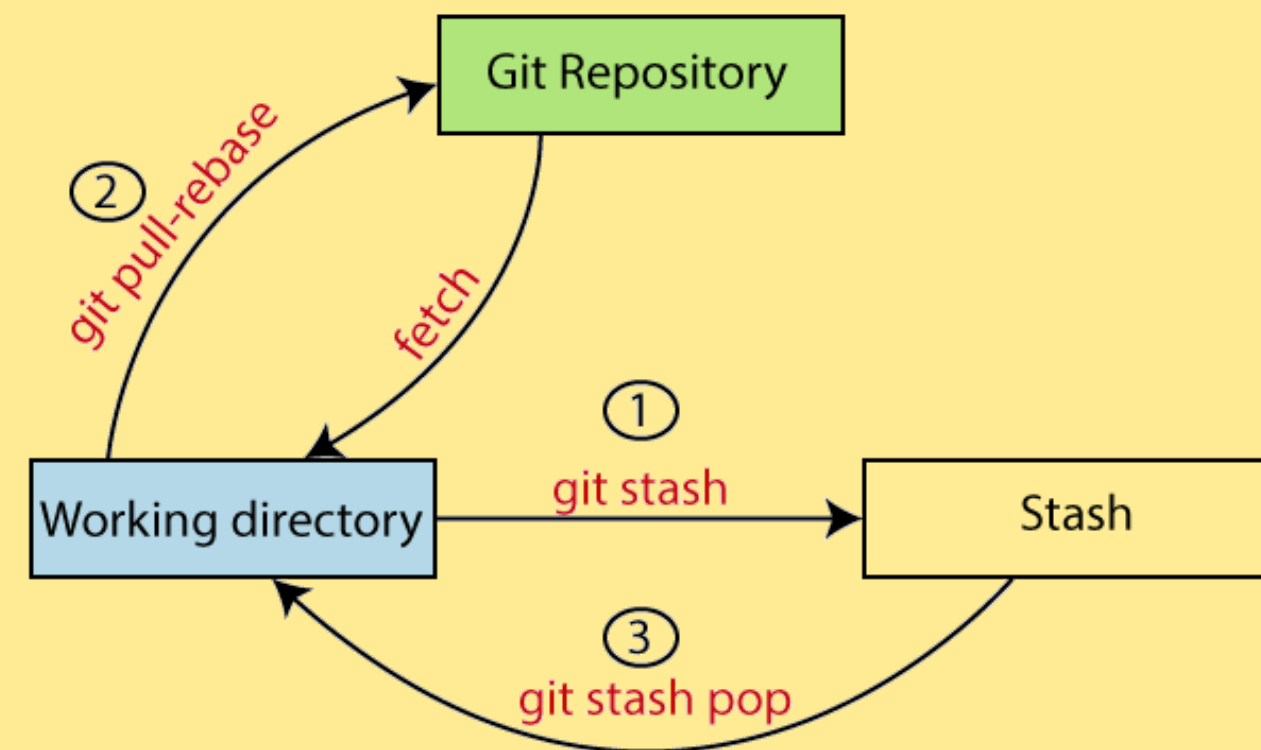
GIT

Danna Valeria Morales Aguilar

STASH

Almacena temporalmente los cambios que hayas hechi en el código en el que se esta trabajando para poder trabajar en otra cosa, y despues regresar a aplicar cambios.

Guardar los cambios en stashes es practico si se tiene que trabajar en otra cosa pero se esta en medio de un cambio de código pero no esta listo para confirmar los cambios.



¿CÓMO GUARDAR EL TRABAJO EN UN STASH?

El comando `git stash` toma los cambios sin confirmar y los guarda para utilizarlos después y luego los deshace en el código en el que se está trabajando.

```
$ git status
On branch main
Changes to be committed:

  new file:   style.css

Changes not staged for commit:

  modified:   index.html

$ git stash
Saved working directory and index state
HEAD is now at 5002d47 our new homepage

$ git status
On branch main
nothing to commit, working tree clean
```

GIT STASH -U

Tiene en cuenta los archivos que no han sido rastreados.

GIT STASH -A

Tiene en cuenta los archivos que fueron ignorados.

GIT STASH -P

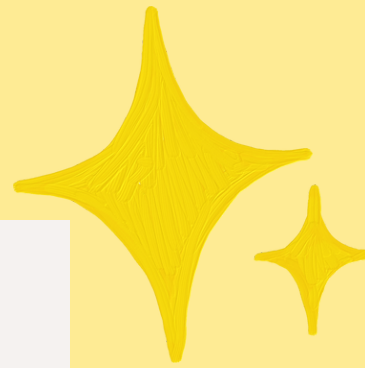
Permite seleccionar y almacenar solo determinados archivos.
Lista de stashes

VARIANTES DE STASH



VARIANTES DE STASH

```
git stash list
```



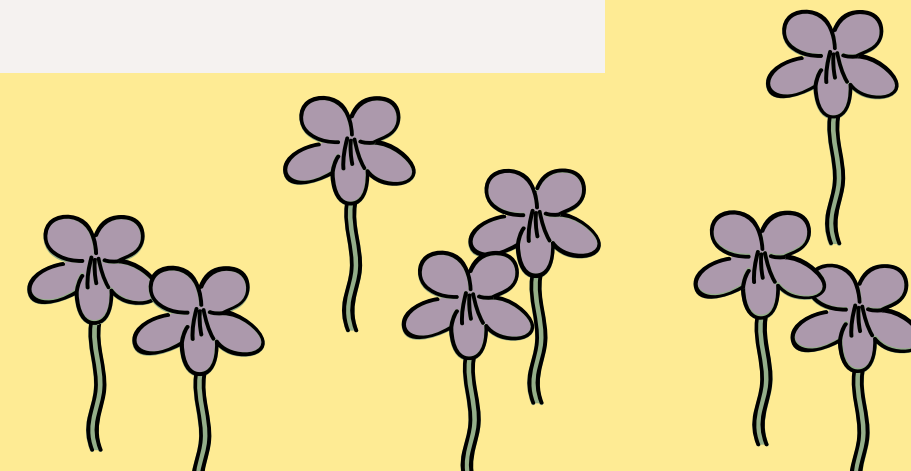
GIT STASH POP

Selecciona el último git stash y se muestra de nuevo.

GIT STASH LIST

Se usa para hacer una lista de tus git stashes y así obtener una buena visión general de tu almacenamiento.

```
git stash pop stash@{1}
```





CLEAN

Se usa para eliminar archivos no rastreados del directorio de trabajo, Estos son archivos que no están bajo seguimiento de Git y no están en el área de staging.

Se debe usar con cuidado ya que los archivos eliminados no se pueden recuperar fácilmente a menos que estén respaldados en otro lugar.

```
git clean [-d] [-f] [-i] [-n] [-q] [-e <pattern>] [-x | -X] [--]  
[<pathspec>...]
```

OPCIONES DE CLEAN

GIT CLEAN -Q

Se ejecuta de manera silenciosa, no mostrará en la salida estándar los nombres de los archivos que están siendo eliminados.

GIT CLEAN -X

Se usa si se desea eliminar solo archivos y directorios no rastreados.

GIT CLEAN -E

Con esta opción se evita que ciertos archivos sean eliminados.


GIT CLEAN -N

Se usa para ver qué archivos se eliminarán sin eliminarlos.

OPCIONES DE CLEAN


GIT CLEAN -D

Se usa para eliminar directorios no rastreados.




GIT CLEAN -F

Es para eliminar archivos y directorios no rastreados de forma forzada sin confirmación.



GIT CLEAN -I

Elimina archivos no rastreados de forma interactiva, permite revisar la lista de archivos no rastreados y seleccionar cuales se desean eliminar.



CHERRY-PICK



Es utilizado para aplicar los cambios de uno o varios commits específicos de una rama a otra.

Permite seleccionar commits individuales de una rama y aplicarlos en otra rama diferente.

Es útil cuando se desea traer cambios específicos de una rama a otra sin tener que fusionar la rama completa.

```
git cherry-pick [--edit] [-n] [-m <parent-number>] [-s] [-x] [--ff]
                [-S[<keyid>]] <commit>...
git cherry-pick (--continue | --skip | --abort | --quit)
```

OPCIONES DE CHERRY-PICK

GIT CHERRY-PICK -E

Abre el mensaje del commit en un editor de texto para que pueda ser modificado antes de confirmar el cherry-pick.

GIT CHERRY-PICK -N

Realiza el cherry-pick pero no crea un nuevo commit, permite hacer modificaciones adicionales antes de confirmar los cambios.



OPCIONES DE CHERRY-PICK

`GIT CHERRY-PICK -X`

Inlcuye automáticamente un comentario en el mensaje del commit resultante que indica que commit original se ha cherry-picked.

`GIT CHERRY-PICK -S`

Agrega automáticamente el "signed-off-by" en el mensaje del commit resultante.

`GIT CHERRY-PICK -M PARENT-NUMBER`

Se utiliza para indicar cuál de los padres del commit debe ser considerado como el "padre" al aplicar los cambios.

