





git

¿Qué es Git?

Git es un sistema de control de versiones que originalmente fue diseñado para operar en un entorno Linux. Actualmente Git es multiplataforma, es decir, que ahora ya no solamente es compatible con Linux, sino también con MacOS y Windows.

¿Qué es un sistema de control de versiones?

Un sistema de control de versiones es el conjunto de herramientas que tenemos disponibles para poder gestionar de una forma eficiente las distintas versiones de nuestros proyectos.

Características importantes de Git

- Git se diferencia de otros sistemas de control de versiones en la forma en la que modela sus datos. Usualmente otros sistemas almacenan la información en una lista de cambios en archivos, mientras Git lo hace como un conjunto de archivos.
- La integridad con la que cuenta es bastante seria. No existen cambios, corrupción en archivos o cualquier alteración sin que Git lo sepa. Esto funciona gracias a una verificación con la que cuenta mediante un checksum, que es básicamente una suma de comprobación que se hace previo al almacenamiento de información.
- Casi todo en Git es local. Es difícil que se necesiten recursos o información externos, basta con los recursos locales con los que cuenta.
- Git cuenta con 3 estados en los que podemos localizar nuestros archivos:

Committed

Es la parte en la que nuestra información está segura alojada en nuestras bases de datos.

Modified

En esta parte hemos realizado cambios en nuestros archivos, pero aún no se ven reflejados en nuestra base de datos.

Staged

En esta parte marcamos nuestros archivos modificados dejándolos listos para confirmarlos.



git

Tipos de Sistemas de Control de Versiones

Local

Este tipo de sistema solo está disponible en el computador en uso y no cuenta con respaldos para recuperar la información en caso de perderla por algún problema.

Centralizado

Este sistema depende de un server en el que este almacenado el repositorio. Si el server sufre alguna falla o es destruido, el repositorio se perderá.

Distribuido

Cada participante del repositorio tiene una copia local y no afecta el trabajo del resto. No perdemos nuestro trabajo si nuestro se computador se daña, simplemente pedimos una copia a cualquier servidor donde esté almacenada la copia.



git

Beneficios de usar Git

Velocidad

Puedes trabajar fluidamente desde tu computador.

Diseño Sencillo

Herramientas necesarias para tomar acción en el VCS a través de código robusto.

Fuerte apoyo en el desarrollo no lineal

Una de las características más increíble de Git. No es necesario seguir un camino predefinido para trabajar. Es decir, puedes crear otras cosas que no estaban definidas. Al final, en algún punto las podrás agregar a tu proyecto.

Completamente distribuido

Esto da la ventaja de que cada quien tenga su propio repositorio. Haciendo que el repositorio principal sea casi imposible perderlo.

Capaz de manejar grandes proyectos.

Puede manejar proyectos tan grandes como Linux.



git

¿Qué diferencia a Git de los otros VCS?

Mientras otros VCS solo almacenan los cambios existentes dentro de los archivos. Git lo que hace además de lo anterior es que si no le hiciste un cambio a un archivo te crea un enlace a ese archivo en lugar de guardarlo otra vez. Esto hace a Git muy rápido para moverse en el tiempo.

Casi cualquier trabajo que hagas en Git es local. Es decir que puedes trabajar de modo offline sin miedo. Esto porque tienes un repositorio en tu computador. Cuando tengas conexión puedes guardar los cambios (Commits en Git).

Git tiene integridad. No puedes perder información durante su transmisión o sufrir corrupción de archivos sin que Git lo detecte.

¿Cómo almacena Git los cambios?

En un SHA-1. Esto es un conjunto de 40 caracteres hexadecimales que harán una referencia a uno de esos cambios en nuestro proyecto.



git

Los tres estados de Git

Working Directory

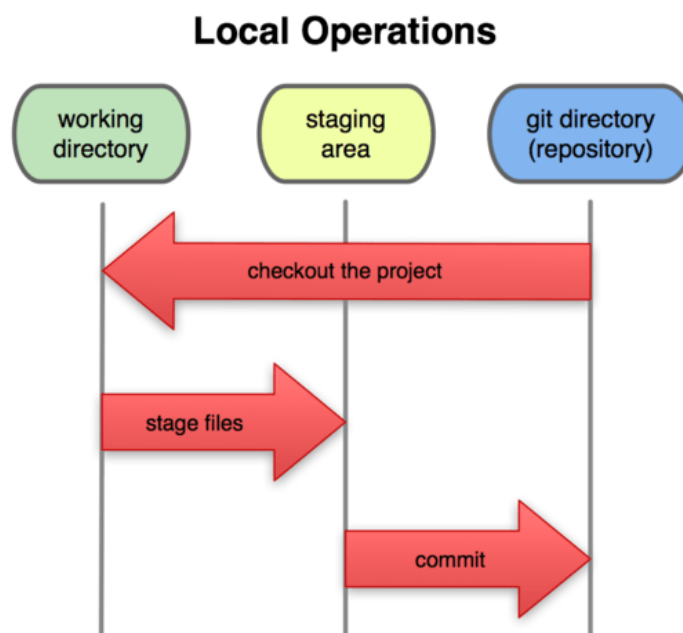
En este estado los archivos se encuentran en el repositorio, pero aún no han sido incluidos en el mismo.

Staging Area

Se preparan los archivos para ser incluidos al repositorio.

Git Directory (repository)

Los archivos son incluidos al repositorio con el uso de commits.



Comandos Terminal



git

Entrar en una carpeta

MINGW32:/c/Users/Luis Falcon/desktop/repositorio

```
Luis Falcon@DESKTOP-FJGJMA6 MINGW32 ~/desktop
$ cd repositorio

Luis Falcon@DESKTOP-FJGJMA6 MINGW32 ~/desktop/repositorio
$
```

Salir de una carpeta

MINGW32:/c/Users/Luis Falcon/desktop

```
Luis Falcon@DESKTOP-FJGJMA6 MINGW32 ~/desktop/repositorio
$ cd ..

Luis Falcon@DESKTOP-FJGJMA6 MINGW32 ~/desktop
$
```

Crear una carpeta

MINGW32:/c/Users/Luis Falcon/desktop

```
Luis Falcon@DESKTOP-FJGJMA6 MINGW32 ~/desktop
$ mkdir carpeta
```

Crear un archivo

MINGW32:/c/Users/Luis Falcon/desktop/repositorio

```
Luis Falcon@DESKTOP-FJGJMA6 MINGW32 ~/desktop/repositorio
$ touch file.txt
```




git

Editar un archivo

MINGW32:/c/Users/Luis Falcon/desktop/repositorio

```
Luis Falcon@DESKTOP-FJGJMA6 MINGW32 ~/desktop/repositorio
$ vim file.txt
```

MINGW32:/c/Users/Luis Falcon/desktop/repositorio

```
Este es un archivo de prueba y esta siendo editado...|
```

```
file.txt[+] [unix] (18:13 10/08/2018)
```

```
1,54 Todo
```

```
-- INSERTAR --
```

Para **editar** el archivo de texto, pulsar la tecla **i**

A continuación, presionar la tecla "**Esc**" y digitar alguna de las opciones.

Guardar y Salir :wq

Salir sin Guardar :q!

Mostrar carpetas o archivos creados

MINGW32:/c/Users/Luis Falcon/desktop/repositorio

```
Luis Falcon@DESKTOP-FJGJMA6 MINGW32 ~/desktop/repositorio
$ ls
carpeta/
```

Borrar carpetas o archivos

MINGW32:/c/Users/Luis Falcon/desktop/repositorio

```
Luis Falcon@DESKTOP-FJGJMA6 MINGW32 ~/desktop/repositorio
$ rm file.txt
```



git

Forzar el borrado de carpetas o archivos

MINGW32:/c/Users/Luis Falcon/desktop/repositorio

```
Luis Falcon@DESKTOP-FJGJMA6 MINGW32 ~/desktop/repositorio  
$ rm -rf subcarpeta
```

Limpiar la consola

MINGW32:/c/Users/Luis Falcon/desktop/repositorio

```
Luis Falcon@DESKTOP-FJGJMA6 MINGW32 ~/desktop/repositorio  
$ clear
```

Comandos Git



git

Mostrar la versión de Git

MINGW32:/c/Users/Luis Falcon

```
Luis Falcon@DESKTOP-FJGJMA6 MINGW32 ~  
$ git --version  
git version 2.17.1.windows.2
```

Configurar tu nombre globalmente

MINGW32:/c/Users/Luis Falcon/desktop/repositorio

```
Luis Falcon@DESKTOP-FJGJMA6 MINGW32 ~/desktop/repositorio  
$ git config --global user.name Luis
```

Habilitar los colores para todos los elementos

MINGW32:/c/Users/Luis Falcon/desktop/repositorio

```
Luis Falcon@DESKTOP-FJGJMA6 MINGW32 ~/desktop/repositorio  
$ git config --global color.ui true
```

Crear un Repositorio

MINGW32:/c/Users/Luis Falcon/desktop/Proyecto

```
Luis Falcon@DESKTOP-FJGJMA6 MINGW32 ~/desktop/Proyecto  
$ git init repositorio  
Initialized empty Git repository in C:/Users/Luis Falcon/Desktop/Proyecto/repositorio/.git/
```



git

Verificar el estatus de los archivos

MINGW32:/c/Users/Luis Falcon/desktop/Proyecto/repositorio

```
Luis Falcon@DESKTOP-FJGJMA6 MINGW32 ~/desktop/Proyecto/repositorio (master)
$ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)

        file.txt

nothing added to commit but untracked files present (use "git add" to track)
```

Agregar los archivos a Staging Area

MINGW32:/c/Users/Luis Falcon/desktop/Proyecto/repositorio

```
Luis Falcon@DESKTOP-FJGJMA6 MINGW32 ~/desktop/Proyecto/repositorio (master)
$ git add -A

Luis Falcon@DESKTOP-FJGJMA6 MINGW32 ~/desktop/Proyecto/repositorio (master)
$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)

        new file:   file.txt
```

-A Agrega todos los archivos que se encuentran en working directory.

git add file.txt Agrega solo el archivo mencionado que se encuentra en working directory.



git

Cancelar los archivos del Staging Area

MINGW32:/c/Users/Luis Falcon/desktop/Proyecto/repositorio

```
Luis Falcon@DESKTOP-FJGJMA6 MINGW32 ~/desktop/Proyecto/repositorio (master)
$ git rm --cached file.txt
rm 'file.txt'

Luis Falcon@DESKTOP-FJGJMA6 MINGW32 ~/desktop/Proyecto/repositorio (master)
$ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)

        file.txt

nothing added to commit but untracked files present (use "git add" to track)
```

Confirmar los archivos (Commit)

MINGW32:/c/Users/Luis Falcon/desktop/Proyecto/repositorio

```
Luis Falcon@DESKTOP-FJGJMA6 MINGW32 ~/desktop/Proyecto/repositorio (master)
$ git commit -m "Se agrego un archivo"
[master (root-commit) 6444d23] Se agrego un archivo
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 file.txt
```

-m "Mensaje" Se le asignara un mensaje al commit. El mensaje es obligatorio para realizar el commit.

Concatenar un Commit nuevo con el previo

MINGW32:/c/Users/Luis Falcon/desktop/Proyecto/repositorio

```
Luis Falcon@DESKTOP-FJGJMA6 MINGW32 ~/desktop/Proyecto/repositorio (master)
$ git commit --amend
[master 0ad386a] Se agregaron varios archivos
Date: Sat Aug 11 19:50:37 2018 -0500
2 files changed, 0 insertions(+), 0 deletions(-)
create mode 100644 document.txt
create mode 100644 file.txt
```

-m "Nuevo Mensaje" Se puede sustituir el mensaje del anterior commit. Accion Opcional.



git

Mostrar historial de todos los commits realizados

MINGW32:/c/Users/Luis Falcon/desktop/Proyecto/repositorio

```
Luis Falcon@DESKTOP-FJGJMA6 MINGW32 ~/desktop/Proyecto/repositorio (master)
$ git log
commit 0ad386a5928d3dce08810217ad6792f3656bb98f (HEAD -> master)
Author: Luis <luisdark935@hotmail.com>
Date: Sat Aug 11 19:50:37 2018 -0500

    Se agregaron varios archivos
```

Etiquetar el commit previo

MINGW32:/c/Users/Luis Falcon/desktop/Proyecto/repositorio

```
Luis Falcon@DESKTOP-FJGJMA6 MINGW32 ~/desktop/Proyecto/repositorio (master)
$ git tag -a 0.1 -m "Version estable del proyecto"
```

Etiquetar un commit específico

MINGW32:/c/Users/Luis Falcon/desktop/Proyecto/repositorio

```
Luis Falcon@DESKTOP-FJGJMA6 MINGW32 ~/desktop/Proyecto/repositorio (master)
$ git log
commit 0ad386a5928d3dce08810217ad6792f3656bb98f (HEAD -> master, tag: 0.1)
Author: Luis <luisdark935@hotmail.com>
Date: Sat Aug 11 19:50:37 2018 -0500

    Se agregaron varios archivos

Luis Falcon@DESKTOP-FJGJMA6 MINGW32 ~/desktop/Proyecto/repositorio (master)
$ git tag -a 0.2 -m "Version estable del proyecto" 0ad386a5928d3dce08810217ad6792f3656bb98f
```

- a Agrega una anotación
- m Agrega mensaje
- l Muestra una lista de todas las etiquetas
- f Renombra una etiqueta
- d Borra una etiqueta



git

Mostrar una lista resumida de todos los commits

MINGW32:/c/Users/Luis Falcon/desktop/Proyecto/repositorio

```
Luis Falcon@DESKTOP-FJGJMA6 MINGW32 ~/desktop/Proyecto/repositorio (master)
$ git log --oneline
0ad386a (HEAD -> master, tag: 0.1) Se agregaron varios archivos

Luis Falcon@DESKTOP-FJGJMA6 MINGW32 ~/desktop/Proyecto/repositorio (master)
$
```

Mostrar un gráfico de la lista de todos los commits (incluye alternos)

MINGW32:/c/Users/Luis Falcon/desktop/Proyecto/repositorio

```
Luis Falcon@DESKTOP-FJGJMA6 MINGW32 ~/desktop/Proyecto/repositorio (master)
$ git log --oneline --graph
* 0ad386a (HEAD -> master, tag: 0.1) Se agregaron varios archivos
```

Mostrar una lista resumida de una cantidad determinada de commits

MINGW32:/c/Users/Luis Falcon/desktop/Proyecto/repositorio

```
Luis Falcon@DESKTOP-FJGJMA6 MINGW32 ~/desktop/Proyecto/repositorio (master)
$ git log --oneline -2
f74e816 (HEAD -> master, tag: 0.3) Se modifico el contenido del archivo file.txt
e44cda2 (tag: 0.2) Se almaceno un nuevo archivo

Luis Falcon@DESKTOP-FJGJMA6 MINGW32 ~/desktop/Proyecto/repositorio (master)
$ git log --oneline
f74e816 (HEAD -> master, tag: 0.3) Se modifico el contenido del archivo file.txt
e44cda2 (tag: 0.2) Se almaceno un nuevo archivo
0ad386a (tag: 0.1) Se agregaron varios archivos
```




git

Mostrar las diferencias entre un commit y otro

MINGW32:/c/Users/Luis Falcon/desktop/Proyecto/repositorio

```
git
Luis Falcon@DESKTOP-FJGJMA6 MINGW32 ~/desktop/Proyecto/repositorio (master)
$ git log --oneline
f74e816 (HEAD -> master, tag: 0.3) Se modifico el contenido del archivo file.txt
e44cda2 (tag: 0.2) Se almaceno un nuevo archivo
0ad386a (tag: 0.1) Se agregaron varios archivos

Luis Falcon@DESKTOP-FJGJMA6 MINGW32 ~/desktop/Proyecto/repositorio (master)
$ git diff 0ad386a
diff --git a/file.txt b/file.txt
index e69de29..4a3ffaf 100644
--- a/file.txt
+++ b/file.txt
@@ -0,0 +1 @@
+mensaje de prueba de cambios
\ No newline at end of file
diff --git a/otherfile.txt b/otherfile.txt
new file mode 100644
index 0000000..e69de29
```

Resetear el proyecto desde un commit específico (soft)

MINGW32:/c/Users/Luis Falcon/desktop/Proyecto/repositorio

```
Luis Falcon@DESKTOP-FJGJMA6 MINGW32 ~/desktop/Proyecto/repositorio (master)
$ git log --oneline
f74e816 (HEAD -> master, tag: 0.3) Se modifico el contenido del archivo file.txt
e44cda2 (tag: 0.2) Se almaceno un nuevo archivo
0ad386a (tag: 0.1) Se agregaron varios archivos

Luis Falcon@DESKTOP-FJGJMA6 MINGW32 ~/desktop/Proyecto/repositorio (master)
$ git reset --soft e44cda2

Luis Falcon@DESKTOP-FJGJMA6 MINGW32 ~/desktop/Proyecto/repositorio (master)
$ git log --oneline
e44cda2 (HEAD -> master, tag: 0.2) Se almaceno un nuevo archivo
0ad386a (tag: 0.1) Se agregaron varios archivos
```

Nota: los archivos incluidos en los commits afectados serán trasladados al staging área del commit en uso.



Reseteo el proyecto desde un commit específico (mixed)

```
MINGW32/c/Users/Luis Falcon/desktop/Proyecto/repositorio

Luis Falcon@DESKTOP-FJGJMA6 MINGW32 ~/desktop/Proyecto/repositorio (master)
$ git log --oneline
f4cbd95 (HEAD -> master) Archivo agregado
e44cda2 (tag: 0.2) Se almaceno un nuevo archivo
0ad386a (tag: 0.1) Se agregaron varios archivos

Luis Falcon@DESKTOP-FJGJMA6 MINGW32 ~/desktop/Proyecto/repositorio (master)
$ git reset --mixed e44cda2

Luis Falcon@DESKTOP-FJGJMA6 MINGW32 ~/desktop/Proyecto/repositorio (master)
$ git status
On branch master
Untracked files:
  (use "git add <file>..." to include in what will be committed)

        archivo.txt

nothing added to commit but untracked files present (use "git add" to track)
```

Nota: los archivos incluidos en los commits afectados serán trasladados al working directory área del commit en uso.

Reseteo el proyecto desde un commit específico (hard)

```
MINGW32/c/Users/Luis Falcon/desktop/Proyecto/repositorio

Luis Falcon@DESKTOP-FJGJMA6 MINGW32 ~/desktop/Proyecto/repositorio (master)
$ git log --oneline
d8fc431 (HEAD -> master) Se recupero el archivo file.txt
e44cda2 (tag: 0.2) Se almaceno un nuevo archivo
0ad386a (tag: 0.1) Se agregaron varios archivos

Luis Falcon@DESKTOP-FJGJMA6 MINGW32 ~/desktop/Proyecto/repositorio (master)
$ git reset --hard e44cda2
HEAD is now at e44cda2 Se almaceno un nuevo archivo

Luis Falcon@DESKTOP-FJGJMA6 MINGW32 ~/desktop/Proyecto/repositorio (master)
$ git log --oneline
e44cda2 (HEAD -> master, tag: 0.2) Se almaceno un nuevo archivo
0ad386a (tag: 0.1) Se agregaron varios archivos
```

Nota: los archivos incluidos en los commits afectados serán eliminados completamente.



git

Crear una rama

MINGW32:/c/Users/Luis Falcon/desktop/Proyecto/repositorio

```
Luis Falcon@DESKTOP-FJGJMA6 MINGW32 ~/desktop/Proyecto/repositorio (master)
$ git branch responsive
```

Borrar una rama

MINGW32:/c/Users/Luis Falcon/desktop/Proyecto/repositorio

```
Luis Falcon@DESKTOP-FJGJMA6 MINGW32 ~/desktop/Proyecto/repositorio (master)
$ git branch -d responsive
Deleted branch responsive (was 0ababc8).
```

Forzar el borrado de una rama

MINGW32:/c/Users/Luis Falcon/desktop/Proyecto/repositorio

```
Luis Falcon@DESKTOP-FJGJMA6 MINGW32 ~/desktop/Proyecto/repositorio (master)
$ git branch -D responsive
Deleted branch responsive (was 0ababc8).
```

Mostrar lista de todas las ramas

MINGW32:/c/Users/Luis Falcon/desktop/Proyecto/repositorio

```
Luis Falcon@DESKTOP-FJGJMA6 MINGW32 ~/desktop/Proyecto/repositorio (master)
$ git branch -l
* master
responsive
```



git

Renombrar una rama

MINGW32:/c/Users/Luis Falcon/desktop/Proyecto/repositorio

```
Luis Falcon@DESKTOP-FJGJMA6 MINGW32 ~/desktop/Proyecto/repositorio (master)
$ git branch -m responsive ResponsiveDesign

Luis Falcon@DESKTOP-FJGJMA6 MINGW32 ~/desktop/Proyecto/repositorio (master)
$ git branch -l
  ResponsiveDesign
* master
```

Moverse entre ramas

MINGW32:/c/Users/Luis Falcon/desktop/Proyecto/repositorio

```
Luis Falcon@DESKTOP-FJGJMA6 MINGW32 ~/desktop/Proyecto/repositorio (master)
$ git checkout ResponsiveDesign
Switched to branch 'ResponsiveDesign'

Luis Falcon@DESKTOP-FJGJMA6 MINGW32 ~/desktop/Proyecto/repositorio (ResponsiveDesign)
$ |
```

Crear una rama y moverse a ella

MINGW32:/c/Users/Luis Falcon/desktop/Proyecto/repositorio

```
Luis Falcon@DESKTOP-FJGJMA6 MINGW32 ~/desktop/Proyecto/repositorio (master)
$ git checkout -b hotfix
Switched to a new branch 'hotfix'

Luis Falcon@DESKTOP-FJGJMA6 MINGW32 ~/desktop/Proyecto/repositorio (hotfix)
$
```



git

Moverse entre commits

MINGW32:/c/Users/Luis Falcon/desktop/Proyecto/repositorio

```
Luis Falcon@DESKTOP-FJGJMA6 MINGW32 ~/desktop/Proyecto/repositorio (ResponsiveDesign)
$ git log --oneline
0ababc8 (HEAD -> ResponsiveDesign, master) Archivo recuperado
e44cda2 (tag: 0.2) Se almaceno un nuevo archivo
0ad386a (tag: 0.1) Se agregaron varios archivos
```

```
Luis Falcon@DESKTOP-FJGJMA6 MINGW32 ~/desktop/Proyecto/repositorio (ResponsiveDesign)
$ git checkout e44cda2
Note: checking out 'e44cda2'.
```

You are in 'detached HEAD' state. You can look around, make experimental changes and commit them, and you can discard any commits you make in this state without impacting any branches by performing another checkout.

If you want to create a new branch to retain commits you create, you may do so (now or later) by using -b with the checkout command again. Example:

```
git checkout -b <new-branch-name>
```

HEAD is now at e44cda2 Se almaceno un nuevo archivo

```
Luis Falcon@DESKTOP-FJGJMA6 MINGW32 ~/desktop/Proyecto/repositorio ((0.2))
$
```



git

Mezclar cambios entre ramas

MINGW32:/c/Users/Luis Falcon/desktop/Proyecto/repositorio

```
Luis Falcon@DESKTOP-FJGJMA6 MINGW32 ~/desktop/Proyecto/repositorio (master)
$ git merge hotfix
Updating 0ababc8..4b647c4
Fast-forward
 licencia.txt | 0
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 licencia.txt

Luis Falcon@DESKTOP-FJGJMA6 MINGW32 ~/desktop/Proyecto/repositorio (master)
$ git branch -l
  ResponsiveDesign
  hotfix
* master
```

Guardar temporalmente archivos o cambios

MINGW32:/c/Users/Luis Falcon/desktop/Proyecto/repositorio

```
Luis Falcon@DESKTOP-FJGJMA6 MINGW32 ~/desktop/Proyecto/repositorio (master)
$ git status
On branch master
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

    new file:   archivo.txt
    new file:   licencia.txt

Luis Falcon@DESKTOP-FJGJMA6 MINGW32 ~/desktop/Proyecto/repositorio (master)
$ git stash
Saved working directory and index state WIP on master: e44cda2 Se almaceno un nuevo archivo
```



git

Mostrar lista de archivos guardados temporalmente

MINGW32:/c/Users/Luis Falcon/desktop/Proyecto/repositorio

```
Luis Falcon@DESKTOP-FJGJMA6 MINGW32 ~/desktop/Proyecto/repositorio (master)
$ git stash list
stash@{0}: WIP on master: e44cda2 Se almaceno un nuevo archivo
```

Mostrar lista de archivos guardados temporalmente

MINGW32:/c/Users/Luis Falcon/desktop/Proyecto/repositorio

```
Luis Falcon@DESKTOP-FJGJMA6 MINGW32 ~/desktop/Proyecto/repositorio (master)
$ git stash list
stash@{0}: WIP on master: e44cda2 Se almaceno un nuevo archivo

Luis Falcon@DESKTOP-FJGJMA6 MINGW32 ~/desktop/Proyecto/repositorio (master)
$ git stash drop stash@{0}
Dropped stash@{0} (0cdd6b969a0c9034af7edf66c76b2c77b2fa6683)
```

Aplicar cambios de los archivos guardados temporalmente

MINGW32:/c/Users/Luis Falcon/desktop/Proyecto/repositorio

```
Luis Falcon@DESKTOP-FJGJMA6 MINGW32 ~/desktop/Proyecto/repositorio (master)
$ git stash list
stash@{0}: WIP on master: e44cda2 Se almaceno un nuevo archivo

Luis Falcon@DESKTOP-FJGJMA6 MINGW32 ~/desktop/Proyecto/repositorio (master)
$ git stash apply stash@{0}
Removing otherfile.txt
Removing document.txt
On branch master
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

    new file:   licencia.txt

Changes not staged for commit:
  (use "git add/rm <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

    deleted:    document.txt
    deleted:    otherfile.txt
```



git

Mover un commit a una rama de desarrollo

MINGW32:/c/Users/Luis Falcon/desktop/Proyecto/repositorio

```
Luis Falcon@DESKTOP-FJGJMA6 MINGW32 ~/desktop/Proyecto/repositorio (hotfix)
$ git cherry-pick ec18494
[hotfix ec9416e] Licencia del proyecto establecida
Date: Sun Aug 12 20:33:22 2018 -0500
2 files changed, 0 insertions(+), 0 deletions(-)
delete mode 100644 document.txt
delete mode 100644 otherfile.txt

Luis Falcon@DESKTOP-FJGJMA6 MINGW32 ~/desktop/Proyecto/repositorio (hotfix)
$ git log --oneline
ec9416e (HEAD -> hotfix) Licencia del proyecto establecida
4b647c4 Licencia del proyecto establecida
0ababc8 (ResponsiveDesign) Archivo recuperado
e44cda2 (tag: 0.2) Se almaceno un nuevo archivo
0ad386a (tag: 0.1) Se agregaron varios archivos
```




Realizado por
Luis Falcon