- git add

- git branch

- git checkout

- git clean

- git clone

- git commit

- git commit -- amend

- git config

- git fetch

- git init

- git log

- git merge

- git pull

- git push

- git rebase

- git reflog

- git remote

- git reset

- git revert

- git status

**Hook**

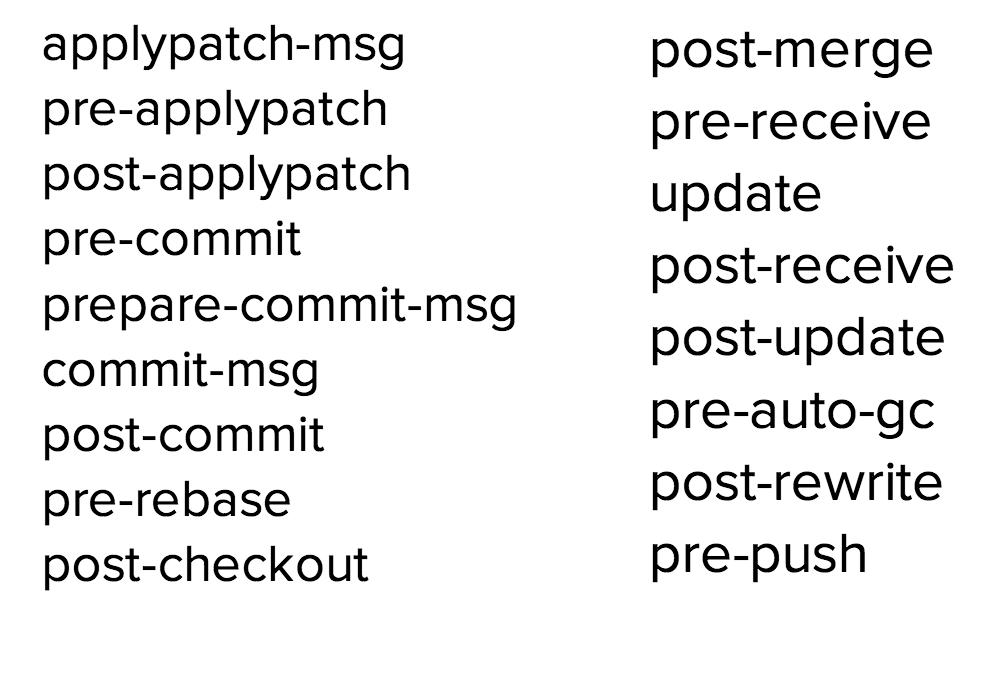
Es un script que corre automáticamente cada vez que sucede un evento particular en un repositorio.

Los “Hooks” te permiten personalizar el comportamiento interno de GIT y ejecutar acciones codificadas por ti en ciertos puntos del ciclo de vida del desarrollo.

Para entrar a ellos, los puedes localizar dentro de tu carpeta de trabajo, entrando en:

cd .git/hooks

Existen cerca de 17 hooks:



Cada uno se activa conforme al tipo de acción que realices en terminal.

Si activas y automatizas, por ejemplo, post-commit, y dentro de este archivo lo llenas con:

git status

git log

Entonces, después de crear el commit, automáticamente correrá esos comandos que están dentro de ese archivo.

Puedes conocer cada una de las acciones en:

<https://github.com/git/git/blob/master/Documentation/githooks.txt>

**Tag**

Una referencia típicamente usada para marcar un punto particular en una cadena de commits.

En contraste con HEAD, un tag no es actualizado por el comando de commit. Se debe de agregar con el comando de tag.

git tag -a v1.4 -m "my version 1.4"

El "-a”  coloca la versión del tag.El “-m" te permite ponerle una descripción sobre lo que trata ese tag.



Si colocas:

git tag

Te muestra la lista de los tags que tienes.

Si colocas:

# git show [la versión del tag]git show v1.0

Entonces obtendrás una descripción del tag:

tag v1.0

Tagger: Miguel Nieva <m@platzi.com>

Date:   Sat May 18 21:12:13 2016 -0700  
my version 1.4

commit ca82a6dff817ec66f44342007202690a93763949

Author: Miguel Nieva <m@platzi.com>Date:   Mon Mar 10 14:20:12 2016 -0700  
    The nav tab is added.

**git alias** (junto a config)

Git alias es un comando que te permite crear tus propios comandos de GIT, a partir de otros, con sus parámetros.

Está vinculado con el comando “config” y la fórmula es:

git config --global alias.[nombre del alias que quieres crear] ‘[comandos con parámetros que quieres que se ejecuten con el alias]

Un ejemplo, para la síntesis de un “git log” con ciertos parámetros que necesitas y no lo quieres escribir a cada momento, es:

git config --global alias.nicelog 'log --oneline --graph --all'

**Revisar carpetas ocultas de git :**

Ls -la

**Si desea reiniciar el proyecto:**

Git tiene el archivo .git que es el que rastrea todo lo del proyecto, el siguiente comando elimina toda las carpetas y sub directorios

Rm -rf .git

Se comprueba con git status y arroja el mensaje:

fatal: Not a git repository (or any of the parent directories): .git

ya solo debe eliminar todos los archivos de forma manual.

**Para marcar un proyecto como funcional o no, es una herramienta aparte que funciona muy bien con github**

<https://travis-ci.org/> es común mente usado por los tester

Atajos

* Crear una carpeta: mkdir +[nombre de la carpeta ] ejemplo: mkdir exposiciones
* Crear un archivo: touch +[nombre del archivo con su extención] ejemplo: touch archivo1.txt
* Editar el archivo: vim +[nombre del archivo ] ejemplo: vim archivo1.txt,precionas la letra (O) para empezar a editar al finalizar marcas x para guardar
* Chmod 0400 crear un archivo que solo lo leeo yo
* Mv comando de linux tambien usado en windows para mover un archivo: mv +[nombre del archivo] +[nombre de la carpeta donde lo mueve]

Servicios que se pueden incluir en github

<https://github.com/marketplace>

