**Comandos Básicos de Git**

Fuente: http://jonas.nitro.dk/git/quick-reference.html

a) Buscando Ayuda:

1. git help comando ó git comando --help

Muestra la ayuda para ese comando

b) Creación de un repositorio:

2. git init

Crea un repositorio en el directorio actual

3. git clone url

Clona un repositorio remoto dentro de un directorio

c) Operaciones sobre Archivos:

4. git add path

Adiciona un archivo o un directorio de manera recursiva

5. git rm ruta

Remueve un archivo o directorio del árbol de trabajo

-f : Fuerza la eliminación de un archivo del repositorio

6. git mv origen destino

Mueve el archivo o directorio a una nueva ruta

-f : Sobre-escribe los archivos existentes en la ruta destino

7. git checkout [rev] archivo

Recupera un archivo desde la rama o revisión actual

-f : Sobre-escribe los cambios locales no guardados

d) Trabajando sobre el código:

8. git status

Imprime un reporte del estado actual del árbol de trabajo local

9. git diff [ruta]

Muestra la diferencia entre los cambios en el árbol de trabajo local

10. git diff HEAD ruta

Muestra las diferencias entre los cambios registrados y los no registrados

11. git add path

Selecciona el archivo para que sea incluido en el próximo commit

12. git reset HEAD ruta

Marca el archivo para que no sea incluido en el próximo commit

13. git commit

Realiza el commit de los archivos que han sido registrados (con git-add)

-a : Automáticamente registra todos los archivos modificados

14. git reset --soft HEAD^

Deshace commit & conserva los cambios en el árbol de trabajo local

15. git reset --hard HEAD^

Restablece el árbol de trabajo local a la versión del ultimo commit

16. git clean

Elimina archivos desconocidos del árbol de trabajo local

e) Examinando el histórico:

17. git log [ruta]

Muestra el log del commit, opcionalmente de la ruta especifica

18. git log [desde [..hasta]]

Muestra el log del commit para un rango de revisiones dado

--stat : Lista el reporte de diferencias de cada revisión

-S'pattern' : Busca el historial de cambios que concuerden con el patrón de búsqueda

19. git blame [archivo]

Muestra el archivo relacionado con las modificaciones realizadas

f) Repositorios remotos:

20. git fetch [remote]

Trae los cambios desde un repositorio remoto

21. git pull [remote]

Descarga y guarda los cambios realizados desde un repositorio remoto

22. git push [remote]

Guarda los cambios en un repositorio remoto

23. git remote

Lista los repositorios remotos

24. git remote add remote url

Añade un repositorio remoto a la lista de repositorios registrados

g) Ramas:

25. git checkout rama

Cambia el árbol de trabajo local a la rama indicada

-b rama : Crea la rama antes de cambiar el árbol de trabajo local a dicha rama

26. git branch

Lista las ramas locales

27. git branch -f rama rev

Sobre-escribe la rama existente y comienza desde la revisión

28. git merge rama

Guarda los cambios desde la rama

h) Exportando e importando:

29. git apply - < archivo

Aplica el parche desde consola (stdin)

30. git format-patch desde [..hasta]

Formatea un parche con un mensaje de log y un reporte de diferencias (diffstat)

31. git archive rev > archivo

Exporta resumen de la revisión (snapshot) a un archivo

--prefix=dir/ : Anida todos los archivos del snapshot en el directorio

--format=[tar|zip] : Especifica el formato de archivo a utilizar: tar or zip

i) Etiquetas:

32. git tag name [revision]

Crea una etiqueta para la revisión referida

-s : Firma la etiqueta con su llave privada usando GPG

-l [patrón] : Imprime etiquetas y opcionalmente los registros que concuerden con el patrón de busqueda

j) Banderas de Estado de los Archivos:

M (modified) : El archivo ha sido modificado

C (copy-edit) : El archivo ha sido copiado y modificado

R (rename-edit) : El archivo ha sido renombrado y modificado

A (added) : El archivo ha sido añadido

D (deleted) : El archivo ha sido eliminado

U (unmerged) : El archivo presenta conflictos después de ser guardado en el servidor (merge)