**Comandos básicos para el uso de GIT.**

|  |  |
| --- | --- |
| Comando | Especificación |
| **git help** <comando> | Muestra ayuda específica del comando que se requiera. |
| **git clone** <uri> <nom> | Clona un directorio usando como nombre de directorio <nom> |
| **git init** | Crea un repositorio en el directorio actual |
| **git add** <dir> | Añade recursivamente todos los archivos de <dir> |
| **git diff --staged** | Compara los cambios organizados con el último **commit** |
| **git commit –a -m** | Organiza automáticamente archivos rastreados |
| **git –cached** <file or regexp> | No realiza un seguimiento del archivo, pero los deja en el directorio de trabajo |
| **git rm** <file> | Borra el archivo <file> |
| **git move** <source><target> | Mueve <source> a la ubicación <target>. |
| **git checkout** <rev> | Recupera un archivo desde la revisión actual.  El parámetro –f sobre escribe los cambios no guardados localmente. |
| **git status** | Muestra el estado del árbol de trabajo actual. |
| **git diff HEAD** <path> | Muestra las diferencias entre los cambios registrados y los no registrados. |
| **git reset HEAD** <path> | Marca <path> para que no sea incluido en el próximo commit. |
| **git commit** | Realiza el commit de los archivos que han sido registrados con “**git add**”.  El parámetro –a, registra automáticamente todos los archivos modificados. |
| **git reset --soft HEAD** | Deshace el **commit** y conserva los cambios en el árbol de trabajo actual. |
| **git reset --hard HEAD** | Restablece el árbol de trabajo local a la versión del último **commit.** |
| **git clean** | Elimina los archivos desconocidos del árbol de trabajo local. |
| **git log** <path> | Muestra el log del **commit**, opcionalmente de la ruta específica. |
| **git log**  <Rev 1…<RevN>> | Muestra el log del **commit** para un rango de revisiones dado.  --stat: Lista el reporte de diferencias para cada versión.  -S ‘pattern’:Busca el historial de cambios que concuerden con el historial de búsqueda. |
| **git blame** <file> | Muestra el archivo relacionado con las modificaciones realizadas. |
| **git fetch *<***remote***>*** | Trae los cambios desde un repositorio remoto |
| **git pull** | Trae e integra con otro repositorio o rama local. |
| **git remote add *<***remote url***>*** | Añade un repositorio a la lista de repositorios registrados. |
| **git branch** | Lista las ramas locales. |
| **git brach** –f <rama/rev> | Sobre escribe la rama existente y comienza desde la revisión. |
| **git merge** <rama> | Guarda los cambios desde la rama. |
| **git apply** <file> | Aplica el parche desde la consola (Standard input) |
| **git format-patch <**F1**…**FN **>** | Formatea un parche con un mensaje de log y un reporte de diferencias. |
| **git archive <**rev> | Exporta resumen de la revisión 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/zip. |
| **git tag** name [rev] | 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 búsqueda. |
| **git remote –v** | Lista los repositorios remotos. |
| **git push** | Actualiza referencias remotas junto con objetos asociados. |
| **git remote show *<***name***>*** | Lista la URL del repositorio remoto. |
| **git remote rename** <current> <change> | Renombra una referencia a un repositorio remoto. |
| **git config --global alias.”ch” <command>** | Crea un alias para autocompletar el comando <command>. |
| **git config --global alias.’commad’ 'reset HEAD --'** | Crea el comando ‘command’ para resumir ‘reset HEAD --’.  Haciendo equivalente la ejecución de estos después de habe ejecutado el comando git config –global alias.unstage ‘reset HEAD --’  $ git unstage fileA  $ git reset HEAD fileA |
| **git config –global user.email ‘email’** | Establece el email para el usuario actual. |
| **git clone ssh://’user’@’server/project.git’**  o bien  **git clone user@server:project.git** | Clona un repositorio de git, usando ssh. |