Nancy Araceli Cano Cerritos.

Programación Web.

**COMANDOS USADOS EN GIT.**

|  |  |
| --- | --- |
| **Descripción** | **Comando** |
| Crea un repositorio en el directorio actual. | **git unit** |
| Clona un repositorio remoto dentro de un directorio. | **git clone** *url* |
| Adiciona un archivo o un directorio de manera recursiva. | **git add path** |
| Remueve un archivo o directorio del árbol de trabajo.  Fuerza la eliminación de un archivo del repositorio. | **git rm** *la ruta*  **git rm -f** *la ruta* |
| Mueve el archivo o directorio a una nueva ruta.  Sobre-escribe los archivos existentes en la ruta destino. | **git mv** *origen destino*  **git mv -f** *origen destino* |
| Recupera un archivo desde la rama o revisión actual.  Sobre-escribe los cambios locales no guardados. | **git checkout** *[rev] archivo*  **git checkout -f** *[rev] archivo* |
| Imprime un reporte del estado actual del árbol de trabajo local. | **git status** |
| Muestra la diferencia entre los cambios en el árbol de trabajo local. | **git diff +**  *[ruta]* |
| Muestra las diferencias entre los cambios registrados y los no registrados. | **git diff HEAD +** *ruta* |
| Selecciona el archivo para que sea incluido en el próximo commit. | **git add** *path* |
| Marca el archivo para que no sea incluido en el próximo commit. | **git reset HEAD** *ruta* |
| Realiza el commit de los archivos que han sido registrados (con git-add)  Automáticamente registra todos los archivos modificados. | **git commit**  **git commit -a** |
| Modificar el mensaje del último commit | **git commit --amend** |
| Deshace commit y conserva los cambios en el árbol de trabajo local. | **git reset** --soft HEAD^ |
| Restablece el árbol de trabajo local a la versión del ultimo commit. | **git reset** --hard HEAD^ |
| Elimina archivos desconocidos del árbol de trabajo local. | **git clean** |
| Muestra el log del commit, opcionalmente de la ruta especifica. | **git log** [ruta] |
| Muestra el log del commit para un rango de revisiones dado.  Lista el reporte de diferencias de cada revisión. | **git log** [desde [..hasta]]  --stat |
| Muestra el archivo relacionado con las modificaciones realizadas. | **git blame** [archivo] |
| Trae los cambios desde un repositorio remoto. | **git fetch** [remote] |
| Descarga y guarda los cambios realizados desde un repositorio remoto. | **git pull** [remote] |
| Guarda los cambios en un repositorio remoto. | **git push** [remote] |
| Lista los repositorios remotos | **git remote** |
| Añade un repositorio remoto a la lista de repositorios registrados. | **git remote add** *remote url* |
| 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. | **git checkout** *rama* |
| Lista las ramas locales. | **git branch** |
| Guarda los cambios desde la rama. | **git merge** rama |
| Aplica el parche desde consola (stdin). | **git apply** - < archivo |
| Formatea un parche con un mensaje de log y un reporte de diferencias (diffstat). | **git format-patch** *desde [..hasta]* |
| 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 tag** *name**[revision]* |
| Guarda el estado en una pila y limpia el directorio para poder cambiar de rama. | **git stash** |
| Muestra la pila | **git stash list** |
| Vuelve al estado original del dir. Stash{n} especifica uno concreto Y --index reaplica los cambios stagged. | **git stash apply** |
| Elimina el primero en la pila. | **git stash pop** |