



COMANDOS GIT

BRENDA ANABEL ALVARADO TREJO



Comando para crear un nuevo repositorio git

```
$ git init
```

Crea una copia local del repositorio ejecutando

```
$ git clone /path/to/repository
```

```
$ git clone username@host:/path/to/repository
```

- Clona un repositorio remoto dentro de un directorio

```
$ git clone url
```

Adiciona un archivo o un directorio de manera recursiva

```
$ git rm ruta
```

Establece el nombre que desea esté anexado a sus transacciones.

```
$ git config -global user.email "[email  
addres]"
```

Establece el e-mail que desea esté anexado a sus transacciones de commit

```
$ git config -global color.ui auto
```

Habilita la útil colorización del producto de la línea de comando



CREAR REPOSITORIOS

Inicia un nuevo repositorio u obtiene uno de una URL existente.

```
$ git init [Project-name]
```

Crea un nuevo repositorio local con el nombre especificado

```
$ git clone [url]
```

EFFECTUAR CAMBIOS

Revista las ediciones y elabora una transmisión de commit

```
$ git status
```

Enumera todos los archivos nuevos o modificados que se deben confirmar.

```
$ git diff
```

Muestra las diferencias del archivo que no se han enviado aún al área de espera.

```
$ git add [file]
```

Toma una instantánea del archivo para preparar la versión

```
$ git diff --staged
```

Muestra las diferencias del archivo entre el área de espera y la última versión del archivo.

```
$ git reset [file]
```



Mueve el archivo del área de espera, pero preserva su contenido.

```
$git commit -m "[mensaje]"
```

Registra las instantáneas del archivo permanente en el historial de la versión.

CAMBIOS GRUPALES

Nombra una serie de commits y combina esfuerzos ya culminados

```
$ git branch
```

Enumera todas las ramas en el repositorio actual.

```
$ git branch [branch-name]
```

Crea una nueva rama

```
$ git checkout [branch-name]
```

Cambia a la rama especificada y actualizada el directorio activo.

```
$ git merge [branch]
```

Combina el historial de la rama especificada con la rama actual

```
$ git branch -d [branch-name]
```

Borra la rama especificada.

```
$ git rm [file]
```

Borra el archivo del directorio activo y pone en el área de espera el archivo borrado



```
$ git rm --cached [file]
```

Retira el archivo del control de versiones, pero preserva el archivo a nivel local

```
$ git mv [file-original] [file-renamed]
```

Cambia el nombre del archivo y lo prepara para el commit

REPASAR EL HISTORIAL

Navega e inspecciona la evolución de los archivos de proyecto

```
$ git log
```

Enumera el historial de la versión para la rama actual

```
$ git log --follow [file]
```

Enumera el historial de versión para el archivo, incluidos los cambios de nombre

```
$ git diff [first-branch]...[second-branch]
```

Muestra las diferencias de contenido entre dos ramas.

```
$ git show [commit]
```

Produce metadatos y cambios de contenido del commit especificado.

REHACER COMMITS

Borre errores y elabora historial de reemplazo



```
$git reset [commit]
```

Deshace todos los commits después de [commit], preservando los cambios localmente.

```
$ git reset --hard [commit]
```

Desecha todo el historial y regresa al commit especificado.

SINCRONIZAR CAMBIOS.

Registrar un marcador de repositorio e intercambiar historial de versión

```
$ git fetch [bookmark]
```

Descargar todo el historial del marcador del repositorio.

```
$ git merge [bookmark] / [branch]
```

Combina la rama del marcador con la rama local actual

```
$ git push [alias] [branch]
```

Carga todos los commits de la rama local al GitHub

```
$ git pull
```

Decarga el historial del marcador e incorpora cambios.