

CONFIGURAR HERRAMIENTAS

Configura la información del usuario para todos los repositorios locales

<code>\$ git config --global user.name "[name]"</code>
Establece el nombre que desea esté anexado a sus transacciones de commit
<code>\$ git config --global user.email "[email address]"</code>
Establece el e-mail que desea esté anexado a sus transacciones de commit
<code>\$ git config --global color.ui auto</code>
Habilita la útil colorización del producto de la línea de comando

CREAR REPOSITORIOS

Inicia un nuevo repositorio u obtiene uno de una URL existente

<code>\$ git init [project-name]</code>
Crea un nuevo repositorio local con el nombre especificado
<code>\$ git clone [url]</code>
Descarga un proyecto y toda su historia de versión

<code>\$ git log</code>
Enumera el historial de la versión para la rama actual
<code>\$ git log --follow [file]</code>
Enumera el historial de versión para el archivo, incluidos los cambios de nombre
<code>\$ git diff [first-branch]...[second-branch]</code>
Muestra las diferencias de contenido entre dos ramas
<code>\$ git show [commit]</code>
Produce metadatos y cambios de contenido del commit especificado

<code>\$ git rm [file]</code>
Borra el archivo del directorio activo y pone en el área de espera el archivo borrado
<code>\$ git rm --cached [file]</code>
Retira el archivo del control de versiones, pero preserva el archivo a nivel local
<code>\$ git mv [file-original] [file-renamed]</code>
Cambia el nombre del archivo y lo prepara para commit

Reubicay retira los archivos con versión

SUPRIMIR TRACKING REHACER COMMITS

Excluye los archivos temporales y las rutas

EFFECTUAR CAMBIOS

Revisa las ediciones y elabora una transacción de commit

<code>\$ git status</code>
Enumera todos los archivos nuevos o modificados que se deben confirmar
<code>\$ git diff</code>
Muestra las diferencias de archivos que no se han enviado aún al área de espera
<code>\$ git add [file]</code>
Toma una instantánea del archivo para preparar la versión
<code>\$ git diff --staged</code>
Muestra las diferencias del archivo entre el área de espera y la última versión del archivo
<code>\$ git reset [file]</code>
Mueve el archivo del área de espera, pero preserva su contenido
<code>\$ git commit -m "[descriptive message]"</code>
Registra las instantáneas del archivo permanentemente en el historial de versión

CAMBIOS GRUPALES

Nombra una serie de commits y combina esfuerzos ya culminados

<code>\$ git branch</code>
Enumera todas las ramas en el repositorio actual
<code>\$ git branch [branch-name]</code>
Crea una nueva rama
<code>\$ git checkout [branch-name]</code>
Cambia a la rama especificada y actualiza el directorio activo
<code>\$ git merge [branch]</code>
Combina el historial de la rama especificada con la rama actual
<code>\$ git branch -d [branch-name]</code>
Borra la rama especificada

NOMBRES DEL ARCHIVO DE REFACTORIZACIÓN

REPASAR HISTORIAL

Navega e inspecciona la evolución de los archivos de proyecto

***.log build/
temp-***

Un archivo de texto llamado `.gitignore` suprime la creación accidental de versiones de archivos y rutas que concuerdan con los patrones especificados

`$ git ls-files --other --ignored --exclude-standard`

Enumera todos los archivos ignorados en este proyecto

Almacena y restaura cambios incompletos

`$ git stash`

Almacena temporalmente todos los archivos tracked modificados

`$ git stash pop`

Restaura los archivos guardados más recientemente

`$ git stash list`

Enumera todos los sets de cambios en guardado rápido

`$ git stash drop`

Elimina el set de cambios en guardado rápido más reciente

Borra 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

GUARDAR FRAGMENTOS SINCRONIZAR CAMBIOS

Registrar un marcador de repositorio e intercambiar historial de versión

`$ git fetch [bookmark]`

Descarga 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`

Descarga el historial del marcador e incorpora cambios