Instalar Git

GitHub proporciona clientes de escritorio que incluyen una interfaz gráfica de usuario para las acciones más comunes que se pueden realizar en un repositorio y una edición de Git en la línea de comandos actualizada automáticamente para escenarios más avanzados.

GitHub para Windows

windows.github.com

GitHub para Mac

mac.github.com

Distribuciones de Git para sistemas Linux y POSIX se encuentran disponibles en el sitio web oficial Git SCM.

Git para todas las plataformas

git-scm.com

Efectuar cambios

Revisa cambios y crea un commit

```
$ git status
```

Enumera todos los archivos nuevos o modificados de los cuales se van a guardar cambios

```
$ git diff
```

Muestra las diferencias entre archivos que no se han enviado aún al área de espera

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

Guarda el estado del archivo en preparación para realizar un commit

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

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

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

Configurar herramientas

Configura la información del usuario para todos los respositorios locales

```
$ git config --global user.name "[name]"
```

Establece el nombre que estará asociado a tus commits

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

Establece el e-mail que estará asociado a sus commits

Crear repositorios

Inicializa 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]
```

Descarga un proyecto y toda su historial de versiones

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

```
$ git commit -m"[descriptive message]"
```

Registra los cambios del archivo permanentemente en el historial de versiones

Cambios grupales

Nombra una serie de commits y combina esfuerzos ya completados

```
$ git branch
```

Enumera todas las ramas en el repositorio actual

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

Crea una nueva rama

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

Cambia a la rama especificada y actualiza el directorio activo

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

Combina el historial de la rama especificada con la rama actual

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

Borra la rama especificada

Refactorización de archivos

Reubica y retira los archivos de los cuales se tiene una versión

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

Borra el archivo del directorio activo y lo pone en el área de espera en un estado de eliminación

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

Retira el archivo del historial de 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 ser guardado

Suprimir el seguimiento de cambios

Excluye los archivos temporales y las rutas

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

Repasar historial

Navega e inspecciona la evolución de los archivos de proyecto

```
$ git log
```

Enumera el historial de versiones para la rama actual

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

Enumera el historial de versiones 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

Borra errores y elabora un historial de reemplazo

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

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

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

Enumera todos los archivos ignorados en este proyecto

Guardar fragmentos

Almacena y restaura cambios incompletos

```
$ git stash
```

Almacena temporalmente todos los archivos modificados de los cuales se tiene al menos una versión guardada

```
$ git stash pop
```

Restaura los archivos guardados más recientemente

```
$ git stash list
```

Enumera todos los grupos de cambios que estan guardados temporalmente

```
$ git stash drop
```

Elimina el grupo de cambios más reciente que se encuentra guardado temporalmente

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 para un repositorio e intercambiar historial de versiones

```
$ 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]
```

Sube todos los commits de la rama local a GitHub

```
$ git pull
```

Descarga el historial del marcador e incorpora cambios