Comandos Git

Git stash

El comando git stash almacena temporalmente (o guarda en un stash) los cambios que hayas efectuado en el código en el que estás trabajando para que puedas trabajar en otra cosa y, más tarde, regresar y volver a aplicar los cambios más tarde. Esto es útil cuando necesitas cambiar a otra rama o hacer otro trabajo sin querer perder tus cambios actuales. Puedes recuperarlos más tarde usando git stash apply o git stash pop. Es una forma conveniente de mantener limpias las ramas mientras trabajas en varias cosas simultáneamente. Por ejemplo:

```
# Guardar cambios temporales
$ git stash
# Cambiarse a la rama deseada
$ git checkout <otra-rama>
# Realizar algún trabajo allí
# Volver a la rama original y restaurar los cambios guardados
$ git checkout <mi-rama>
$ git stash apply
```

Git clean

El comando git clean se utiliza para eliminar archivos no deseados de tu directorio de trabajo. Esto podría incluir la eliminación de artefactos de construcción temporal o la fusión de archivos en conflicto. Existen opciones adicionales como -n, -d, y -x para personalizar su comportamiento. La opción -n muestra qué archivos serían borrados sin realmente eliminarlos; -d también borra directorios vacíos; y -x elimina incluso los archivos ignorados por .gitignore. Un ejemplo de uso podría ser:

```
# Borrar todos los archivos no controlados por Git
$ git clean -fdx
```

Git cherry-pick

A menudo se confunde con git cherry, pero son comandos distintos con propósitos ligeramente diferentes. Mientras git cherry compara commits según sus contenidos, cherry-pick aplica commits específicos a tu rama actual mediante su SHA-1 completo o range. La ejecución de cherry-pick es el acto de elegir una confirmación de una rama y aplicarla a otra. git cherry-pick puede ser útil para deshacer cambios. Por ejemplo, supongamos que una confirmación se aplica accidentalmente en la rama equivocada. Puedes cambiar a la rama correcta y ejecutar cherry-pick en la confirmación para aplicarla a donde debería estar.

git cherry-pick es una herramienta útil, pero no siempre es una práctica recomendada. Ejecutar cherry-pick puede generar confirmaciones duplicadas y, en muchos casos en los que su ejecución sí funcionaría, son preferibles las fusiones tradicionales.

```
git cherry-pick <commit-hash>
```