Comandos avanzados Git

Stash

El comando git stash se utiliza en Git para guardar temporalmente todos los cambios en el directorio de trabajo que aún no están listos para ser confirmados. Esto te permite cambiar de rama sin perder esos cambios.

Para qué se usa:

- Para guardar cambios en el directorio de trabajo de manera temporal para poder cambiar de rama sin perderlos.
- Para limpiar el directorio de trabajo temporalmente, permitiéndote trabajar en un estado diferente.

Cómo se usa:

- Para guardar los cambios temporalmente, ejecutas: git stash
- Puedes listar los stashes guardados con: git stash list
- Para aplicar los cambios guardados y eliminarlos del stash, usas: git stash pop
- Para aplicar los cambios guardados sin eliminarlos del stash, usas: git stash apply

Clean

El comando git clean se utiliza para eliminar archivos no rastreados del directorio de trabajo en Git. Esto incluye archivos que fueron creados en el directorio de trabajo pero que aún no han sido agregados al índice con git add.

Para qué se usa:

- Para limpiar el directorio de trabajo eliminando archivos que no están siendo rastreados por Git, ayudando a mantener un entorno de trabajo limpio.

Cómo se usa:

- Para ver qué archivos serán eliminados sin realizar la acción, puedes usar: git clean -n o git clean --dry-run
- Para eliminar archivos no rastreados, ejecutas: git clean -f
- Si también deseas eliminar directorios no rastreados, usas: git clean -fd.

Cherry – pick

El comando git cherry-pick se utiliza para aplicar los cambios introducidos por algunos commits existentes en otra rama a la rama actual.

Para qué se usa:

- Para aplicar un cambio específico de una rama a otra sin tener que fusionar o rebasear toda la rama.
- Útil para aplicar correcciones rápidas o trasladar características específicas entre ramas sin traer todo el historial.

Cómo se usa:

- Primero, debes estar en la rama donde quieres aplicar el cambio.
- Luego, utilizas el comando git cherry-pick <commit-hash> donde <commit-hash> es el identificador del commit que deseas aplicar a tu rama actual.

Recuerda que al usar estos comandos, especialmente git clean y git cherry-pick, debes proceder con precaución ya que pueden alterar significativamente tu historial de trabajo y los archivos en tu directorio de trabajo.