

Recuperación de datos y el peligro de los comandos destructivos en Git

Cuadernillo | Vol. 1

mercedev.es — 2026-04-27 | Fase 8 (Expansión de Contenido)

El Desafío (Síntoma)

Durante un intento de purgar material multimedia pesado del historial profundo de Git (reescritura de historial), la carpeta objetivo desapareció repentinamente del explorador de archivos del editor de código (VS Code). Se generó una alerta crítica por posible pérdida de datos no versionados (archivos *untracked*), asumiendo que el comando de consola había arrasado con el disco duro físico.

La Maniobra (Lógica)

Se aplicó ingeniería forense para determinar el estado real de la máquina: 1. **Descartar la ilusión óptica:** Se utilizó la terminal nativa del sistema (`ls -la`) para confirmar si la desaparición era real o si la interfaz gráfica (GUI) estaba ocultando los archivos por estar listados en el `.gitignore` . 2. **Rastreo de operaciones:** Al confirmar la ausencia física, se dedujo que el comando `git rm --cached` no era el culpable (pues está diseñado para no tocar el disco). La causa raíz fue una manipulación accidental en la GUI (clic en "Descartar cambios"), la cual elimina archivos físicos sin red de seguridad. 3. **Rescate:** Se localizaron y restauraron los archivos directamente desde la Papelera de reciclaje del sistema operativo anfitrión (Ubuntu) y, como medida de mitigación futura, se programó un script Python para generar copias de seguridad locales (`merci-backup.py`).

El Aprendizaje / Deuda Técnica

Este incidente consolida tres reglas de arquitectura operativa innegociables en DevSecOps:

1. **Las interfaces gráficas mienten; la terminal es la única fuente de verdad.** Los editores de código modernos ocultan información por

conveniencia. Ante la duda, el diagnóstico siempre debe realizarse desde una terminal nativa (fuera del IDE).

2. **Git no protege lo que no conoce.** Operar con archivos *untracked* (no rastreados) es caminar por la cuerda floja. Si se borran, Git no tiene memoria de ellos para restaurarlos.
3. **Snapshots antes de la cirugía.** Ejecutar comandos de reescritura de historial (`filter-branch` , `reset --hard`) o purgas de caché es una operación "a corazón abierto". Es obligatorio generar un archivo `.zip` con el estado íntegro del proyecto antes de alterar el motor de control de versiones.

Asumir que las herramientas nos protegen de nosotros mismos es el primer paso hacia el desastre.