

Conflictos masivos en Git y la estrategia de fuerza bruta (--ours)

Cuadernillo | Vol. 1

mercedev.es — 2026-04-28 | Fase 8 (Expansión de Contenido)

El Desafío (Síntoma)

Tras una reescritura de historial local (`git reset --soft`), el servidor remoto conservó una línea de tiempo "fantasma". Al intentar sincronizar mediante `git pull` , se desató un conflicto de fusión (Merge Conflict) masivo. Git inyectó marcadores de conflicto (`<<<<<<<`) en archivos de texto e impidió la fusión de archivos binarios (PDFs), bloqueando por completo el ciclo de Integración Continua (CI - Continuous Integration).

La Maniobra (Lógica)

Dado que se tenía la certeza absoluta de que el entorno local era la Única Fuente de Verdad (SSOT - Single Source of Truth) y que el servidor remoto contenía basura obsoleta, se descartó la resolución manual archivo por archivo. Se ejecutó la estrategia de resolución por imposición local:

```
git checkout --ours .
git add .
git commit -m "Merge branch 'main' - Resolución masiva de conflictos"
```

El Aprendizaje / Deuda Técnica

Entrar en pánico o confiar en las interfaces gráficas (GUI - Graphical User Interface) de los editores de código ante conflictos masivos suele derivar en la pérdida de datos y en la desincronización de la bitácora. La terminal ofrece mecanismos de cirugía de alta precisión. Comprender el funcionamiento de estrategias como `--ours` (quedarse con la versión local) o `--theirs` (quedarse con la versión entrante) permite neutralizar colisiones catastróficas en apenas tres comandos, siempre y cuando la disciplina operativa (commits atómicos) se haya mantenido rigurosamente.