

Patrón Cache Hit para Terminales Zero-Noise

Esterería: DevSecOps e Infraestructura

Subtema: Automatización

mercedev.es — 2026-05-23 | Epic 5 - Fase 1

El Desafío (Síntoma)

El extractor Data-Driven de métricas SRE (`merci-extract-metrics.py`) imprimía un bloque extenso de texto en la terminal cada vez que se ejecutaba el orquestador maestro (`merci total`), detallando la actualización del DOM. Dado que las auditorías externas de PageSpeed son puntuales, el 99% de las veces el script procesaba exactamente el mismo JSON, generando ruido visual innecesario y degradando la *Developer Experience* (DX).

La Maniobra (Lógica)

Se implementó un patrón de *Cache Hit* basado en el disco duro. El script guarda el nombre del último reporte procesado en un archivo minúsculo y oculto (`.metrics_cache`). En ejecuciones posteriores, si el JSON más reciente de la carpeta coincide con el nombre almacenado en la caché, el script interrumpe el flujo y finaliza silenciosamente (`sys.exit(0)`), emitiendo únicamente una notificación en una sola línea.

El Aprendizaje / Deuda Técnica

En la filosofía *DevSecOps*, el ruido constante en la terminal provoca "Ceguera de Taller": el desarrollador se acostumbra a ignorar los logs por costumbre y termina pasando por alto errores críticos reales. El paradigma *Zero Maintenance* exige que las herramientas automatizadas solo "hablen" cuando tengan una novedad arquitectónica o un error que reportar.

En resumen

El programa que lee el rendimiento de la web llenaba la pantalla de texto cada vez que se guardaban los cambios, a pesar de que el rendimiento de la web era exactamente el mismo que hacía un segundo. Para solucionarlo, se le añadió una memoria a corto plazo para que recuerde cuál fue el último archivo que procesó. Ahora, si ve que nada ha cambiado, se mantiene en silencio, dejando la pantalla limpia y sin distracciones.