

Optimización de Copias de Seguridad Locales y Zero Bloat

Cuadernillo | Vol. 1

mercedev.es — 2026-04-30 | Fase 8 (Expansión de Contenido)

1. El Desafío (Síntoma)

Durante el ciclo de desarrollo, la herramienta local de copias de seguridad (`merci-backup.py`) estaba generando archivos ZIP con un peso superior a los 50 MB. Para un repositorio compuesto casi exclusivamente por archivos de texto plano (Markdown, HTML, SASS, Python), este tamaño era una anomalía crítica que indicaba la presencia de código "polizón".

2. La Maniobra (Lógica)

Para descubrir qué se estaba comprimiendo bajo el capó, se implementó un modo de "Caja de Cristal" en el script mediante la bandera `--verbose` (o `-v`). Al ejecutarlo, la terminal reveló la causa raíz: se estaba empaquetando la instalación completa de WordPress (`public/blog`), los binarios autogenerados por el compilador Dart Sass (`scripts/merci/bin`) y un cementerio de PDFs de auditorías antiguas.

El fallo residía en la lógica de exclusión. El script ignoraba carpetas por su nombre genérico (ej. `backups` o `.git`), pero no se podía añadir la palabra `blog` a la lista negra porque eso habría destruido el respaldo de los artículos legítimos guardados en `laboratorio/blog/` .

La Solución Arquitectónica: Se refactorizó el script para abandonar la coincidencia por cadenas de texto y adoptar un filtrado por **rutas absolutas** usando la librería `pathlib` :

```
# Comprobación estricta y quirúrgica
dirs[:] = [d for d in dirs if (Path(root) / d) not in EXCLUDE_PATHS]
```

Esto permitió bloquear quirúrgicamente la ruta `public/blog` (el CMS pesado) salvando `laboratorio/blog` (nuestro código fuente).

3. El Aprendizaje / Deuda Técnica

Este incidente refuerza dos grandes lecciones del paradigma DevSecOps:

1. **Zero Bloat en Disaster Recovery:** Una copia de seguridad de código fuente debe contener exclusivamente el estado inmutable del proyecto. Las dependencias externas, los binarios (Sass) o el CMS (WordPress) se instalan, se regeneran o se gestionan aparte. Tras el parche, el backup se redujo a **0.35 MB (una optimización del 99.3%)**.
 2. **Transparencia de las herramientas CLI:** Toda herramienta de consola (CLI) que realice operaciones destructivas o de I/O masivo debe poseer un modo detallado (`--verbose`). Un script no debe ser nunca una caja negra; el desarrollador debe poder auditar su comportamiento en tiempo real a través de la salida estándar de la terminal.
-