# Arquitectura Zero Maintenance: Compilación Incremental y st_mtime

**Estantería:** DevSecOps e Infraestructura

**Subtema:** Gobernanza

**mercedev.es** — 2026-05-20 | Epic 3 - Fase 2

## El Desafío (Síntoma)

A medida que el ecosistema documental de mercedev.es crecía, el orquestador maestro (`merci-total.py`) comenzó a mostrar signos de fatiga, superando los 20 segundos de tiempo de ejecución.

La inyección del *Profiler* reveló que el 90% de este tiempo (aprox. 8.5 segundos) era consumido por el motor de Generación de Sitios Estáticos (SSG) al compilar PDFs mediante WeasyPrint. El sistema operaba bajo un patrón **Clean Build**: en cada ejecución, eliminaba a ciegas todos los archivos generados y los volvía a renderizar desde cero, independientemente de si el documento original (Markdown) había sido modificado o no.

Paralelamente, las herramientas de auditoría documental dependían de metadatos mantenidos manualmente por los humanos (fechas escritas en comentarios de texto) para calcular la deriva documental, provocando constantes falsos negativos por olvidos.

## La Maniobra (Lógica)

Se ejecutó una refactorización transversal hacia el paradigma **Zero Maintenance** (Cero Mantenimiento Humano) e **Incremental Build** (Construcción Incremental):

1. **Adopción de la verdad física (`st_mtime`):** Se erradicó la dependencia de las fechas manuales en los archivos de código. El detector de deriva y el motor SSG ahora interrogan directamente al Sistema Operativo para obtener la fecha inmutable de modificación física del archivo (`st_mtime`).
2. **Patrón Cache Hit:** El orquestador compara el `st_mtime` del PDF de salida frente al Markdown de origen. Si el origen no ha cambiado, se aborta la costosa llamada de renderizado (Cache Hit), reduciendo el tiempo de proceso de ese archivo a casi 0 milisegundos.
3. **Recolección de Basura (Mark & Sweep):** Para evitar la acumulación de archivos "zombis" (PDFs/HTMLs cuyos Markdowns originales han sido

borrados o renombrados), se implementó un *Garbage Collector* diferido. Durante la compilación, el script rastrea los archivos generados válidos en un `set()`. Al finalizar, itera sobre los directorios públicos y ejecuta un `unlink()` exclusivamente sobre los archivos huérfanos.

4. **Supply Chain Security:** Se endureció el Agente Auditor implementando el módulo nativo `ast` (Abstract Syntax Tree). En lugar de depender de expresiones regulares falibles, el auditor disecciona la gramática de los scripts `.py` para bloquear cualquier importación que no figure en una lista blanca estricta (Zero Trust).

## El Aprendizaje / Deuda Técnica

**El rendimiento no se optimiza con hardware, se optimiza con arquitectura.**

Migrar del *Clean Build* al *Incremental Build* redujo el tiempo de compilación del SSG de 8.46s a **0.41s** (una mejora del 95%), logrando que el pipeline maestro completo caiga por debajo de la barrera psicológica de los 10 segundos (Sub-10s).

Auditar la verdad física ( `st_mtime` ) y erradicar las cabeceras manuales demostró que el verdadero **DevSecOps** es aquel que elimina la carga cognitiva del desarrollador. Si una herramienta requiere que el humano introduzca datos redundantes para funcionar, la herramienta está mal diseñada.

Queda como conocimiento consolidado que la Inteligencia Artificial (SLMs), si bien es excepcional para razonar sobre código (como demostró el *Chaos Monkey*), añade una latencia inaceptable al ciclo crítico de Integración Continua (CI). Tareas de sincronización determinista, como la actualización del Roadmap o la recolección de métricas, deben delegarse incondicionalmente a código nativo (Python puro) para mantener la Experiencia de Desarrolladora (DX) intacta.

## En resumen

A medida que la web crecía, el sistema tardaba cada vez más en actualizarse porque borraba y volvía a construir todos los archivos desde cero ante cualquier

pequeño cambio, por mínimo que fuera. Le hemos enseñado al sistema a mirar la fecha física en la que se guardó cada archivo en el disco duro. Ahora, el sistema solo actualiza lo que realmente se ha tocado y hace una "limpieza de basura" automática al final. Hemos pasado de esperar casi 10 segundos a que los cambios sean casi instantáneos, sin que ninguna persona tenga que acordarse de anotar o modificar fechas a mano.