Automatización Extendida: Compilador Data-Driven para la Biblioteca

Estantería: DevSecOps e Infraestructura

Subtema: Gobernanza

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

El Desafío (Síntoma)

A medida que el Agente Autónomo del glosario extraía e inyectaba nuevas definiciones técnicas (DevSecOps) directamente en el archivo Markdown (`glosario-tecnico.md`), se hicieron evidentes dos problemas estructurales: 1. **Fragilidad de Formato:** La Inteligencia Artificial, a pesar de usar Prompts estrictos, ocasionalmente sufría "alucinaciones de formato" (devolviendo `**Término**` en vez de `### Término`). Un simple error tipográfico de la IA rompía la visualización del documento. 2. **Desalineación Filosófica:** El glosario Markdown estaba siendo tratado como una base de datos cuando, en realidad, en nuestro ecosistema SSG, la *Única Fuente de Verdad* es la bitácora de documentación. El glosario no debe ser un documento de trabajo, sino un *Art de Coté* (artefacto compilado).

La Maniobra (Lógica)

Se refactorizó el script `merci-glosario.py` abandonando la inyección directa en Markdown en favor de un enfoque **Data-Driven (Basado en JSON)**:

1. **Estado JSON Maestro:** Se extrajo el histórico completo hacia un nuevo archivo `glosario-tecnico.json` . Este JSON almacena los términos, definiciones, el registro de archivos origen, y una *lista negra* de términos descartados.
2. **IA Determinista:** Se configuró el cliente HTTP para requerir explícitamente `format: "json"` a la API de Ollama. La IA ahora no devuelve texto libre, sino un objeto estructurado predecible e imposible de romper visualmente.
3. **Compilación Dinámica (Build-Time):** El script ahora opera como un compilador. Tras sincronizar los nuevos términos con el JSON, genera y sobrescribe el archivo `glosario-tecnico.md` desde cero. Esto garantiza que el artefacto resultante esté matemáticamente ordenado (alfabéticamente) y formateado de forma impecable en cada ejecución del pipeline (`merci-total.py`).

El Aprendizaje / Deuda Técnica

- **JSON sobre Markdown para Agentes:** Cuando un Agente Autónomo tiene permisos de escritura continua, el formato de almacenamiento de estado debe ser estricto (JSON o Base de Datos). Dejar que un LLM manipule código Markdown interactivo o en bruto introduce Deuda Técnica impredecible.
- **Aislamiento del Origen:** Consolidar el paradigma de que "los archivos .md finales son de solo lectura (compilados)" simplifica el mantenimiento. Si el Markdown se corrompe por acción humana, la próxima ejecución del Build lo restaurará instantáneamente desde el JSON.

En resumen

Antes, nuestra Inteligencia Artificial escribía el diccionario técnico directamente sobre la página final, lo que a veces causaba errores visuales si la IA se equivocaba con el formato o la puntuación. Para solucionarlo, ahora la IA guarda las definiciones en un "cajón de datos" estructurado e invisible. Luego, nuestro sistema actúa como una imprenta mecánica: coge esos datos seguros y fabrica la página perfecta y ordenada alfabéticamente desde cero cada vez. Así nos aseguramos de que el resultado nunca se rompa visualmente, por mucho que la IA se equivoque.