

Arquitectura y Gobernanza: merci-total.py como SSOT

Compendio | Vol. 1

mercedev.es — 2026-05-07 | Fase 7

Arquitectura y Gobernanza: merci-total.py como SSOT

El Desafío (Síntoma)

A medida que el ecosistema Merci crecía, la cantidad de scripts independientes (optimizador de imágenes, compilador SASS, orquestador SSG, sincronizador CMS, auditor estático y rastreador dinámico) se multiplicó. Ejecutar estas herramientas individualmente generaba fricción operativa y un alto riesgo de omisión humana. Por ejemplo, compilar el mapa del sitio antes de generar los HTML, o hacer un commit sin pasar la auditoría de accesibilidad WAI-ARIA (Web Accessibility Initiative - Accessible Rich Internet Applications). Esta fragmentación amenazaba la integridad del principio de Única Fuente de Verdad (SSOT - Single Source of Truth).

La Maniobra (Lógica)

Se implementó `merci-total.py` como el orquestador maestro del pipeline CI/CD (Continuous Integration / Continuous Deployment) local.

El script impone un orden de ejecución innegociable que separa la fase de Construcción (Build) de la fase de Aseguramiento de Calidad (QA):

1. **Build:** `merci-optimizer` -> `merci-styles` -> `merci-publish` -> `merci-wp` -> `merci-sync-pages` .
2. **QA & SEO:** `merci-sitemap` -> `merci-audit` -> `merci-linkcheck` .

Se dotó al pipeline del patrón **Fail-Fast**: si cualquier subprocesso devuelve un código de error (ej. un enlace roto o un secreto expuesto), el orquestador se detiene inmediatamente. Simultáneamente, se aplicó el patrón de **Degradación Elegante (Graceful Degradation)** para dependencias externas: si falta la API Key

de IA o la librería `weasyprint` para PDFs, el pipeline advierte pero finaliza con éxito, protegiendo la experiencia "Out-of-the-Box" del Boilerplate.

El Aprendizaje / Deuda Técnica

Pipeline as Code. La automatización es inútil si depende de la memoria del desarrollador. Encapsular la cadena de suministro en un único comando (`merci total`) transforma una arquitectura compleja en un flujo de trabajo de fricción cero. Además, esta centralización es el pilar que permite la Regla 14 de gobernanza: cualquier clon del Boilerplate hereda este flujo exacto, previniendo la Deriva de Configuración (Configuration Drift).

Cuadernillos de Referencia (Profundización táctica)

Para estudiar los desafíos técnicos concretos que cimentaron esta arquitectura, consulta:

- [Release Pipeline Iterativo \(Agile\) en despliegue de plantillas](#)
- [Documentación como Código: El README.md como Única Fuente de Verdad](#)
- [Silenciando al Guardián: Excepciones explícitas en Linters Estáticos](#)