Domando a WooCommerce: Jerarquía de plantillas en temas custom

Cuadernillo | Vol. 1

mercedev.es — 2026-04-28 | Fase 8 (Expansión de Contenido)

El Desafío (Síntoma)

Al integrar la tienda en la arquitectura híbrida, WooCommerce destruyó por completo el diseño estructural del proyecto. La página de la tienda carecía de cabecera (`<header>`), pie de página (`<footer>`) y CSS, rompiendo la experiencia de usuario. El plugin ignoraba el archivo `index.php` del Child Theme y escupía un HTML desnudo.

La Maniobra (Lógica)

Se requirieron dos acciones simultáneas para recuperar el control arquitectónico: 1. **Declaración de soporte:** Se inyectó `add_theme_support('woocommerce')` en el archivo `functions.php` . Esto desactiva el mecanismo de autodefensa del plugin. 2. **Envoltorio explícito:** Se creó el archivo `woocommerce.php` en la raíz del tema, replicando la estructura HTML monolítica del proyecto (`<body><header>...</header><main>`) e inyectando la función `woocommerce_content()` en el interior del bloque deseado.

El Aprendizaje / Deuda Técnica

Los sistemas dinámicos complejos o CMS (Content Management System - Sistema de Gestión de Contenidos) como WordPress poseen jerarquías de plantillas (Template Hierarchy) estrictas. Si un tema *custom* o un Boilerplate minimalista no declara sus intenciones explícitamente mediante *hooks*, los plugins asumen que el tema es incapaz de renderizarlos y ejecutan rutinas de *fallback* invasivas.

Mantener el control de la capa visual en ecosistemas híbridos exige dominar y neutralizar estas inyecciones por defecto, como también se demostró al purgar el CSS basura inyectado por los bloques de WooCommerce

(`wp_dequeue_style('wc-blocks-style')`) para proteger las métricas de rendimiento 100/100.