YOORshop SAS
RCS 817 466 147 — 254 Rue Vendôme, 69003 Lyon, France — yoorhosting.com
`[datos de la empresa por completar]`

Ref. informe: YH-DEMO-2026-02
Fecha de análisis: 30 de octubre de 2026

DIAGNÓSTICO Y SOLUCIONES CLAVE

INFORME DE INGENIERÍA APLICATIVA Y BASE DE DATOS · ENTREGABLE N.º 2

Optimización del rendimiento de un sitio web

Análisis de picos de CPU al 70 %, contención de caché del menú y reescritura SQL de un escaneo de 787 504 filas.

CLIENTE DESTINATARIO

Sociedad: SARL Moto Pièces Express (Ficticio)
Tienda en línea: https://boutique-moto.example (PrestaShop 8.2.3)
Servidor dedicado: srv-demo.yoorhosting.com (Xeon 16 núcleos, 64 GB RAM)
Entorno: PHP-FPM pool motodemo_prod (48 workers) • MariaDB

CARGA DE CPU OBSERVADA
70 % a 1 270 %
45 picos registrados en importaciones

CONSULTAS LENTAS PHP
540 / 976
Bloqueadas en preg_grep del menú

ACELERACIÓN SQL MEDIDA
x 4 000
De 1,60 s a 0,0004 s por llamada

TIEMPO FINAL NUEVOS PRODUCTOS
0,0004 s
Por comparación directa indexada

SOLICITUD DEL CLIENTE

«Al observar importantes ralentizaciones durante la actualización periódica de tarifas y stock mediante módulos de importación automatizados, con saturaciones recurrentes del procesador (picos de hasta 1 270 % de CPU acumulada en 16 núcleos), el cliente encargó a YOORshop **identificar el origen exacto de los bloqueos** y prescribir un **plan de optimización sin impacto negativo en el catálogo**.»

MECANISMO COMÚN DE SOBRECARGA IDENTIFICADO

El análisis cruzado del registro de consultas lentas PHP (*PHP slowlog*) y de las marcas de tiempo de modificación en base de datos (`date_upd`) reveló un cuello de botella estructural: **cada guardado de un producto vacía simultáneamente dos cachés aplicativas pesadas**, que se recalculan en bucle durante los flujos de sincronización:

Correlación temporal demostrada: La modificación de 22 productos en una hora coincide al segundo con la explosión de consultas lentas (20:44, 20:47, 20:49, 20:50 y de 20:52 a 20:56). En total se registraron 45 picos de carga disparados por peticiones POST de administración desde la IP del operador `198.51.100.22` .

1 Problema N.º 1: Regeneración abusiva del menú (Módulo `ps_mainmenu`)

PUNTO DE ANÁLISIS	CONSTATACIÓN TÉCNICA Y SOLUCIÓN VALIDADA
Anclaje excesivo	El módulo está enganchado innecesariamente a `actionObjectProductAddAfter` , `actionObjectProductUpdateAfter` y `actionObjectProductDeleteAfter` (tabla `psdemo_hook_module`). Estos hooks fuerzan la llamada a `clearMenuCache()` (ps_mainmenu.php:1007), destruyendo toda la caché JSON.
Coste de reconstrucción	**540 de las 976 consultas lentas** PHP se detienen en `preg_grep` (ps_mainmenu.php:796). La reconstrucción compara **9 788 categorías con 26 968 imágenes**. Dos procesos concurrentes recalculan el mismo archivo en paralelo (colisión de escritura a las 21:02:01 y 21:02:05).
Remediación inmediata	**Desenganchar `ps_mainmenu` de los tres hooks de producto** desde el back-office de PrestaShop (*Diseño* > *Posiciones*). Modificación 100 % sin riesgo: el menú no contiene ningún producto individual (configuración `MOD_BLOCKTOPMENU_ITEMS = CAT77,CAT71,LNK1,LNK3`). Los hooks de categorías y páginas CMS permanecen activos.

2 Problema N.º 2: Consulta no indexada del bloque «Nuevos productos» (`ps_newproducts`)

El módulo `ps_newproducts` ejecuta tras cada actualización el método `Product::getNewProducts()`. La consulta original aplica un cálculo directo sobre la columna indexada, impidiendo que MariaDB utilice el índice:

Consulta original (No sargable) — **1,60 segundos**

```
WHERE DATEDIFF(date_add, NOW()) < 20
```

Consecuencia en MariaDB: El índice `date_add` queda desactivado. *Full Table Scan* de **787 504 filas** en `psdemo_category_product`, tabla temporal en disco y ordenación pesada.

✓ Consulta optimizada (Sobrecarga override) — **0,0004 segundos**

```
WHERE date_add > '2026-10-07'
```

Consecuencia en MariaDB: Uso inmediato del índice B-Tree. Lectura restringida a los 8 registros requeridos. **Ganancia de velocidad medida: factor 4 000**.

3 Dimensionamiento de capacidad del pool PHP-FPM

Para absorber los flujos simultáneos sin rechazos HTTP 503, nuestro equipo dimensionó a medida el pool `motodemo_prod`:

PROCESOS MÁX. (CHILDREN)
48 workers
Estándar compartido: 15

MEMORIA ASIGNADA
1 024 MB / worker
Absorbe importaciones de catálogo

CARGA CRON MEDIA
57 % de 1 núcleo
Régimen estabilizado por minuto

RESTAURACIÓN DE CACHÉS
Inmediata
Fin de conflictos entre procesos

4 Síntesis de acciones y mejoras obtenidas

#	ACCIÓN RECOMENDADA	PROCEDIMIENTO	IMPACTO MEDIDO
1	Desenganche del módulo `ps_mainmenu`	Back-office (Diseño > Posiciones)	**-540 consultas lentas** Elimina la contención en `preg_grep`.
2	Despliegue override SQL `getNewProducts`	Archivo en `override/classes/Product.php`	**Tiempo dividido por 4 000** Fin del escaneo de 787 504 filas.
3	Purga de sondas de monitorización	Limpieza de `/root/cpuwatch/`	**Completado** Sondas detenidas el 28/10 a las 12:44.

Validación técnica de ingeniería:
División de Ingeniería y Rendimiento Web — YOORshop SAS

✓ Plan de Optimización Validado