

# Limpieza segura de la base de datos en WordPress

Paso a paso + checklist completa

**Menos bloat, misma web – en 15 minutos y con copia de seguridad**

Autor: Fabini | [@fabiniidc](https://twitter.com/fabiniidc)

Guía gratuita · [fabini.one](https://fabini.one)

# Antes de empezar: seguridad primero

Esta guía te ayudará a reducir el peso de tu base de datos WordPress y mejorar los tiempos de respuesta eliminando elementos innecesarios: transients caducados, tablas huérfanas de plugins desinstalados, espacio desperdiciado por fragmentación (Overhead) y revisiones acumuladas. Todo ello sin comprometer la estabilidad de tu sitio.

**Advertencia importante:** Trabajar directamente sobre la base de datos conlleva riesgos. Una acción mal ejecutada puede dejar tu web inaccesible o perder datos. Por eso, antes de tocar nada, debes completar estos requisitos obligatorios.



## Backup completo

Realiza una copia de seguridad completa: archivos del servidor (vía FTP o panel de hosting) y exportación de la base de datos (desde phpMyAdmin: Exportar → Método rápido → SQL). Guarda ambas copias en tu ordenador.



## Entorno de staging

Si tu hosting lo permite, trabaja primero en un entorno de pruebas (staging). Esto te permite experimentar sin afectar la web en producción. Si no dispones de staging, extrema las precauciones.



## Conoce tu prefijo

Identifica el prefijo de tus tablas WordPress. Por defecto es `wp_`, pero muchas instalaciones usan otro (por ejemplo: `wpxx_` o `prod_`). Compruébalo en phpMyAdmin o en el archivo `wp-config.php`.



**Nota ética y de responsabilidad:** Si no comprendes completamente algún paso de esta guía, detente y consulta con un profesional o desarrollador de confianza. Nunca ejecutes comandos SQL o acciones en la base de datos sin entender sus consecuencias. Esta guía asume conocimientos básicos-intermedios de administración WordPress (uso de FTP, phpMyAdmin, WP-CLI).

Una vez cumplidos estos tres requisitos y verificado que tienes acceso a restaurar el backup si algo sale mal, puedes continuar con confianza.

# Medición "antes": establece tu línea base

Antes de realizar cualquier limpieza, debes medir el estado actual de tu base de datos y rendimiento web. Esto te permitirá cuantificar las mejoras y detectar problemas si algo falla durante el proceso.

## Tamaño de la base de datos

Accede a phpMyAdmin, selecciona tu base de datos WordPress en el menú lateral izquierdo. En la pestaña "Estructura" verás el tamaño total en la parte inferior (por ejemplo: "Suma: 245,3 MiB"). Anota este valor.

## TTFB (Time To First Byte)

El TTFB mide cuánto tarda el servidor en responder. Abre tu navegador (Chrome/Firefox), pulsa F12 para abrir las DevTools, ve a la pestaña "Red" o "Network", recarga la página de inicio y busca el documento HTML principal. Verás "Waiting (TTFB)" en milisegundos. Anota este valor.

**Herramientas alternativas para medir TTFB:** Puedes usar servicios online como GTmetrix, WebPageTest o Pingdom Tools. Estas herramientas ofrecen informes detallados que incluyen el TTFB de tu servidor. Realiza 2-3 mediciones y calcula la media para mayor precisión.

Fecha	Tamaño BD (MB)	TTFB Home (ms)	Notas
[[DD/MM/AAAA]]	[[ej: 245.3]]	[[ej: 850]]	Antes de limpieza

Guarda estos valores iniciales. Los compararemos tras completar todos los pasos de limpieza para evaluar la mejora real conseguida. También verifica que el frontend (páginas públicas) y el panel de administración carguen sin errores antes de empezar.

# Paso 1: Eliminar transients caducados

Los transients son fragmentos de datos temporales que WordPress y sus plugins almacenan en la base de datos con una fecha de caducidad. Se usan para cachear información (por ejemplo, resultados de APIs, consultas pesadas) y mejorar rendimiento. Sin embargo, WordPress no siempre limpia automáticamente los transients caducados, y con el tiempo pueden acumularse miles de registros obsoletos ocupando espacio innecesario.

01

## Opción recomendada: WP-CLI

Si tienes acceso a WP-CLI (línea de comandos de WordPress), este es el método más seguro y rápido. Conéctate por SSH a tu servidor y navega hasta el directorio raíz de WordPress.

02

## Ejecuta el comando

Escribe el siguiente comando y pulsa Enter. Este borra únicamente los transients cuya fecha de expiración ya pasó, sin afectar los activos.

03

## Verificación

WP-CLI mostrará cuántos transients eliminó (por ejemplo: "Deleted 1847 expired transients"). Anota este número y verifica que la web sigue funcionando correctamente.

```
# Eliminar solo transients caducados (recomendado)
wp transient delete --expired
```

```
# Opcional (úsalo CON CUIDADO): eliminar TODOS los transients
# Solo si sabes que ningún proceso crítico depende de ellos
# wp transient delete --all
```

**Advertencia:** El comando `wp transient delete --all` borra todos los transients, incluyendo los válidos. Esto puede ralentizar temporalmente tu web hasta que se regeneren. Úsalo solo si comprendes las implicaciones o si estás seguro de que no hay transients críticos activos.

**Sin WP-CLI:** Si no dispones de acceso SSH o WP-CLI, puedes usar un plugin de confianza como "WP-Optimize" o "Advanced Database Cleaner" que incluyen opciones para limpiar transients caducados desde el panel de administración. Asegúrate de leer la documentación del plugin antes de usarlo.

- Comando ejecutado sin errores
- Número de transients eliminados anotado
- Frontend y panel de administración verificados

# Paso 2: Optimizar tablas (reducir Overhead)

El "Overhead" es espacio desperdiciado dentro de las tablas de la base de datos. Cuando WordPress borra o actualiza registros, no siempre recupera ese espacio inmediatamente, dejando fragmentos vacíos que aumentan el tamaño de las tablas sin aportar datos útiles. Optimizar las tablas reorganiza su estructura interna, liberando ese espacio y mejorando la eficiencia de las consultas.

## Método visual: phpMyAdmin

Accede a phpMyAdmin desde tu panel de hosting. Selecciona tu base de datos WordPress en el menú lateral. Verás la lista de todas las tablas con una columna "Overhead" (fragmentación). Las tablas con overhead superior a 0 necesitan optimización.

1. Marca la casilla "Seleccionar todo" en la parte inferior
2. En el desplegable "Para los elementos seleccionados:" elige "Optimizar tabla"
3. Confirma la acción y espera a que se complete

## Método CLI: WP-CLI

Si prefieres la línea de comandos, WP-CLI ofrece un comando específico para optimizar toda la base de datos de WordPress en un solo paso. Es rápido y seguro.

```
# Optimizar base de datos completa  
wp db optimize
```

El comando mostrará el progreso y confirmará cuándo finalice. No requiere confirmación adicional.

**¿Qué hace la optimización?** La operación OPTIMIZE TABLE (MySQL/MariaDB) reescribe la tabla eliminando el espacio desperdiciado, desfragmenta los índices y mejora el rendimiento de las consultas. Es similar a la desfragmentación de un disco duro: reorganiza los datos para que ocupen menos espacio y sean más rápidos de leer.

**¿Cuándo optimizar?** Es recomendable ejecutar esta optimización cada 1-3 meses, o después de desinstalar plugins pesados, limpiar grandes cantidades de datos o notar ralentizaciones en el panel de administración. La operación es segura y no requiere backup específico (aunque siempre debes tener uno reciente).

**Nota técnica:** Durante la optimización, las tablas pueden bloquearse temporalmente (segundos), lo que puede causar pequeños retrasos en las peticiones concurrentes. En sitios con mucho tráfico, considera ejecutar esta operación en horarios de baja actividad.

- Todas las tablas optimizadas correctamente
- Overhead reducido a 0 o mínimo
- Web funciona sin errores tras optimización

# Paso 3: Identificar y eliminar tablas huérfanas

Cuando desinstalas un plugin de WordPress, especialmente aquellos complejos como WooCommerce, bbPress, BuddyPress o formularios avanzados, muchas veces las tablas personalizadas que crearon permanecen en la base de datos. Estas "tablas huérfanas" no tienen utilidad sin el plugin activo, pero continúan ocupando espacio y añadiendo complejidad innecesaria.

## Detectar tablas sospechosas

En phpMyAdmin, examina la lista de tablas de tu base de datos WordPress. Las tablas huérfanas suelen incluir el nombre del plugin en su identificador. Ejemplos comunes:

- `wp_woocommerce_*` si desinstalaste WooCommerce
- `wp_edd_*` si desinstalaste Easy Digital Downloads
- `wp_bbp_*` si desinstalaste bbPress
- `wp_formmaker_*` si desinstalaste Form Maker

## Verificación obligatoria antes de borrar

Nunca elimines una tabla sin completar estas tres verificaciones. Un error aquí puede dejar tu web inaccesible.

1. **El plugin/servicio ya no está instalado:** Comprueba en Plugins → Plugins instalados que efectivamente lo desinstalaste
2. **La tabla no cambia al navegar:** Anota el número de registros de la tabla sospechosa, navega por tu web durante 5 minutos (frontend y admin), vuelve a phpMyAdmin y verifica que el número no cambió
3. **Hay backup reciente:** Confirma que tienes una copia de seguridad de menos de 24 horas

## Eliminación segura

Si las tres verificaciones anteriores son positivas y estás 100% seguro de que la tabla es huérfana, procede así:

1. En phpMyAdmin, marca solo la tabla sospechosa (no "Seleccionar todo")
2. Haz clic en "Exportar" y guarda un dump SQL de esa tabla específica en tu ordenador (por si acaso)
3. Vuelve a la estructura, marca la tabla, elige "Eliminar" (DROP) en el desplegable
4. Confirma la acción
5. Inmediatamente verifica que la web carga correctamente

 **Política conservadora:** Si tienes dudas sobre una tabla, NO la elimines. Es preferible conservar 10 MB de tablas innecesarias que romper funcionalidades importantes. Algunos plugins avanzados usan tablas con nombres poco descriptivos. En caso de duda extrema, consulta documentación del plugin o desarrolladores.

- Tablas huérfanas identificadas y listadas
- Verificaciones de seguridad completadas al 100%
- Tablas eliminadas con dump de respaldo guardado

# Paso 4: Controlar revisiones y autoguardados

WordPress guarda automáticamente cada cambio que haces en entradas y páginas como "revisiones", permitiéndote restaurar versiones anteriores. Aunque útil, esta función puede generar miles de registros en la base de datos si no se limita. Un artículo editado 50 veces genera 50 revisiones. Multiplica esto por cientos de entradas y obtienes decenas de miles de filas innecesarias ocupando espacio y ralentizando consultas.

1

2

3

## Limitar revisiones futuras

Edita el archivo `wp-config.php` desde FTP o administrador de archivos. Busca la línea `/* That's all, stop editing! */` y ANTES de ella añade las siguientes líneas. Esto limita las revisiones a 10 por entrada (razonable para la mayoría de sitios).

## Ajustar autoguardado

Por defecto, WordPress autoguarda cada 60 segundos mientras editas. Puedes aumentar este intervalo a 120 segundos (2 minutos) para reducir la frecuencia sin perder seguridad. Añade esta línea también en `wp-config.php`.

## Limpiar revisiones existentes

Las configuraciones anteriores solo afectan nuevas entradas. Para limpiar revisiones antiguas acumuladas, tienes dos opciones: usar un plugin de confianza (WP-Optimize, Advanced Database Cleaner) o ejecutar una consulta SQL manual (avanzado, requiere conocimiento).

```
/* Configuración de revisiones y autoguardado */
/* Limitar revisiones a un número razonable */
define('WP_POST_REVISIONS', 10);

/* Autoguardado cada 120 segundos (opcional) */
define('AUTOSAVE_INTERVAL', 120);

/* That's all, stop editing! Happy publishing. */
```

**Criterios para limpiar revisiones antiguas:** Si decides usar un plugin o consulta SQL para eliminar revisiones existentes, sé selectivo. No borres todas las revisiones: conserva las últimas 5-10 de cada entrada (especialmente entradas importantes). Algunos plugins permiten configurar "mantener X revisiones por entrada" o "eliminar revisiones anteriores a X días". Elige opciones conservadoras: mejor limpiar poco que limpiar demasiado.

□ **Advertencia sobre desactivación total:** Puedes desactivar completamente las revisiones con `define('WP_POST_REVISIONS', false);`, pero esto NO es recomendable. Las revisiones son una red de seguridad valiosa ante errores humanos o actualizaciones problemáticas. Limítalas, no las elimines.

## Antes de editar `wp-config.php`

- Descarga una copia de seguridad del archivo actual
- Usa un editor de texto plano (Notepad++, Sublime, VS Code), nunca procesadores de texto
- Respeta la codificación UTF-8 sin BOM
- No añadas espacios o saltos de línea extra

## Después de editar `wp-config.php`

- Sube el archivo editado vía FTP en modo ASCII o binario
- Verifica inmediatamente que la web carga
- Si aparece error, restaura el archivo original
- Verifica permisos: `wp-config.php` debe tener permisos 440 o 400 (seguridad)

# Re-medición y validación: compara resultados

Has completado todos los pasos de limpieza. Ahora es el momento de medir nuevamente el tamaño de tu base de datos y el rendimiento web, comparar con los valores iniciales y verificar que todo funciona correctamente. Esta fase de validación es crítica para confirmar que la limpieza fue exitosa y no introdujo problemas.

01

## Medir tamaño de BD actualizado

Accede nuevamente a phpMyAdmin, selecciona tu base de datos y revisa el tamaño total en la pestaña Estructura (parte inferior). Anótalo en la tabla "Después".

02

## Medir TTFB actualizado

Repite la medición del Time To First Byte usando el mismo método que en la medición inicial (DevTools del navegador o herramienta online). Realiza 2-3 mediciones y calcula la media para mayor precisión.

03

## Verificación funcional completa

Navega por tu sitio web durante 10 minutos. Comprueba: página de inicio, entradas de blog, páginas estáticas, formularios de contacto, proceso de compra (si es tienda), login de usuarios, panel de administración, editor de entradas, y cualquier funcionalidad crítica de tu negocio.

Momento	Tamaño BD (MB)	TTFB Home (ms)	Diferencia
Antes	[[ej: 245.3]]	[[ej: 850]]	—
Después	[[ej: 198.7]]	[[ej: 680]]	-46.6 MB / -170 ms

**Interpretación de resultados:** Una limpieza exitosa típicamente reduce el tamaño de la base de datos entre un 10% y 40% (dependiendo del nivel de acumulación previo) y mejora el TTFB entre 50-300 ms. Si tus resultados están en ese rango, excelente. Si la mejora es menor, puede deberse a que tu base de datos ya estaba relativamente limpia, o que el cuello de botella de rendimiento está en otro lugar (servidor, plugins pesados, caché mal configurada).



Reducción media de tamaño

En bases de datos con mantenimiento irregular



Mejora en TTFB  
Típica tras optimización completa



Errores detectados

Si seguiaste todos los pasos con cuidado

- ☐ **Si algo no funciona:** Si tras la limpieza detectas errores, páginas que no cargan o funcionalidades rotas, detén cualquier acción adicional y consulta la sección de Troubleshooting (siguiente card). No entres en pánico: tienes el backup completo preparado al inicio.

# Checklist imprimible + Troubleshooting

## Checklist de verificación completa (1 minuto)

Antes de dar por finalizado el proceso, verifica estos puntos:

- Backup completo realizado y verificado
- Transients caducados eliminados sin errores
- Tablas optimizadas (Overhead ↓ a cero o mínimo)
- Tablas huérfanas revisadas y eliminadas con seguridad
- Revisiones limitadas en wp-config.php
- Autoguardado ajustado (opcional)
- Frontend carga correctamente (todas las páginas)
- Panel de administración funciona sin errores
- Métricas "después" anotadas y comparadas
- Documentación del proceso completada

## Troubleshooting: soluciones rápidas

**Error al optimizar tablas:** Si phpMyAdmin o WP-CLI devuelven error al optimizar una tabla específica, puede indicar corrupción. Solución: restaura esa tabla desde tu backup. En phpMyAdmin: Importar → seleccionar archivo SQL del backup → marca "Continuar si se encuentran errores" → Ejecutar. Verifica que la web funcione después.

**Borré una tabla en uso por error:** Si tras eliminar una tabla aparecen errores de base de datos ("Table doesn't exist"), necesitas restaurar inmediatamente. Pasos: 1) Accede a phpMyAdmin. 2) Ve a Importar. 3) Selecciona tu archivo de backup SQL completo. 4) Marca solo la tabla que necesitas restaurar (si tu backup lo permite) o importa todo el backup (sobrescribirá toda la BD). 5) Verifica funcionalidad.

**No tengo WP-CLI:** Si tu hosting no proporciona acceso SSH o WP-CLI, todos los pasos descritos pueden realizarse alternativamente desde phpMyAdmin (más lento pero igual de efectivo) o mediante plugins de optimización de base de datos. Plugins recomendados: WP-Optimize, Advanced Database Cleaner. Lee su documentación antes de usarlos.

**La web está más lenta tras limpiar:** Si el rendimiento empeoró en lugar de mejorar, posibles causas: 1) Borraste transients activos (se regenerarán en minutos/horas). 2) Borraste una tabla que un plugin necesita (restaura backup). 3) Problema no relacionado con la BD (caché desactivada, servidor sobrecargado). Verifica logs de error PHP y WordPress (wp-content/debug.log si WP\_DEBUG está activo).

**Cuándo pedir ayuda profesional:** Si tras restaurar el backup persisten errores, si no puedes acceder al panel de administración, o si la web muestra "Error estableciendo conexión con la base de datos", detente y contacta con tu proveedor de hosting o un desarrollador WordPress. No intentes soluciones avanzadas sin conocimientos técnicos sólidos.

# Seguimiento continuo y cierre

La limpieza de base de datos no es una tarea única, sino parte de un mantenimiento preventivo regular. Establecer un calendario de revisión te ayudará a mantener tu WordPress ágil y evitar acumulaciones futuras. Esta última sección te proporciona una hoja de seguimiento y las conclusiones finales.

## Hoja de seguimiento mensual (próximos 30 días)

Anota estos valores cada semana para identificar tendencias y anticipar necesidades de limpieza. Si el tamaño de BD crece más de un 10% mensual sin añadir contenido sustancial, es momento de repetir la limpieza.

Semana	Fecha	Tamaño BD (MB)	TTFB (ms)	Notas
1	[[DD/MM]]	[[ej: 198.7]]	[[ej: 680]]	Post-limpieza
2	[[DD/MM]]			
3	[[DD/MM]]			
4	[[DD/MM]]			

### Frecuencia recomendada

- Transients:** Cada 2-4 semanas (automatizable con WP-CLI + cron)
- Optimización de tablas:** Cada 1-3 meses o tras desinstalar plugins grandes
- Revisión de tablas huérfanas:** Cada 6 meses o tras limpieza de plugins
- Verificación de revisiones:** Anual (la configuración en wp-config.php es permanente)

### Próximos pasos

- Implementa caché de objeto si tienes más de 5.000 visitas/día (Redis/Memcached)
- Considera migrar tablas muy grandes (> 1GB) a almacenamiento externo
- Revisa y desactiva plugins que no uses activamente (cada plugin añade consultas)
- Monitoriza consultas lentas con Query Monitor (plugin) para identificar cuellos de botella

## Conclusión: menos bloat, más respuesta

Has completado una limpieza integral de tu base de datos WordPress siguiendo prácticas conservadoras y seguras. Tu sitio ahora responde más rápido, tu base de datos ocupa menos espacio, y has establecido límites preventivos para evitar acumulaciones futuras. Recuerda: mide, limpia y vuelve a medir. El mantenimiento preventivo es siempre más eficaz que la corrección reactiva.

**Comparte tu experiencia:** Si esta guía te ayudó a optimizar tu WordPress, etiqueta **@ilfabini** en redes sociales y comparte tus resultados. ¿Cuántos MB ahorraste? ¿Cuánto mejoró tu TTFB? Tu feedback ayuda a mejorar futuras versiones de esta guía.

[Más recursos gratuitos](#)

[Descarga esta guía en PDF](#)

▢ **Nota legal:** Esta guía se proporciona "tal cual" sin garantías de ningún tipo. El autor no se responsabiliza de pérdidas de datos, interrupciones de servicio o cualquier daño derivado del uso de esta información. Siempre realiza backups antes de modificar tu base de datos y, si no estás seguro, consulta con profesionales cualificados. El uso de esta guía implica aceptación de estos términos.