

CÓMO ACELERAR SU SITIO DE WORDPRESS



Brian Jackson

KINSTA

Hosting WordPress Administrado

Cómo acelerar su sitio de WordPress

(LA MEJOR GUÍA)



Para ver la versión siempre actualizada de esta guía puede escanear el código QR o hacer click en el siguiente enlace:

<https://kinsta.com/es/aprender/acelerar-wordpress/>

Publicado por Kinsta

Traducido por Óscar Martín

Libro maquetado por Maja Szakadat



Sobre el autor – Brian Jackson

Brian es el Director de Marketing de Kinsta. Se enfoca en todo, desde el desarrollo de nuevas estrategias de crecimiento online, creación de contenido, SEO técnico, y alcance dentro de la comunidad. Le gusta escribir artículos realmente completos y ha publicado más de 2.500 artículos en su blog.

Una de las mayores pasiones de Brian es WordPress, que ha utilizado durante más de una década. Habiendo trabajado tanto en la industria de redes de entrega de contenido, como en el sector de alojamiento de alto rendimiento, se especializa en la optimización del rendimiento web y en buscar estrategias para acelerar los sitios web.

Cómo acelerar su sitio de WordPress

Capítulos

10

Tipos de Sitios de
WordPress:
Estático o Dinámico

13

Elija un Hosting
de WordPress de Alto
Rendimiento

25

Elija un Servidor Cerca
de sus Visitantes

34

Un DNS Premium
es Mucho Mejor que
un DNS Gratuito

38

El Tema
de WordPress
Es Importante

47

Todo Sobre los
Plugins de WordPress

52

Configuración
Óptima
de WordPress

58

¿Por Qué el Caché
Es Tan Importante?

68

La Optimización
de Imágenes
Es Necesaria

77

Optimizando Su
Base de Datos

95

Utilizando un Content
Delivery Network (CDN)

105

Descargue Archivos
Multimedia y Email
Cuando lo Necesite

110

¿Cómo Encontrar Cuellos
de Botella y Plugins Lentos?

129

Recomendaciones sobre la
Optimización del Back-End

144

Consejos sobre
la Optimización
del Front-End y los
Servicios Externos

172

Siempre Optimice
con los Dispositivos
Móviles en Mente

A lo largo de los años, hemos publicado muchos tutoriales con formas de optimizar y acelerar WordPress. Pero a veces puede ser confuso tratar de encontrar todo lo que necesita en un solo lugar. Por eso, hoy vamos a compartir con usted todo lo que sabemos sobre como turbo acelerar WordPress, más de 15 años de experiencia y duras lecciones aprendidas, todo en esta guía definitiva. Ya sea que esté empezando a usar WordPress o sea un desarrollador experimentado, ¡le prometemos que encontrará algo útil en esta guía!

Más de un tercio de la web está impulsada por WordPress. aunque es impresionante, también significa que hay miles de temas, plugins y tecnologías diferentes que tienen que coexistir. Para el usuario de WordPress, esto puede convertirse en una pesadilla cuando su sitio comienza a tener cuellos de botella y no saben por qué o ni siquiera por dónde empezar a solucionar los problemas.

Hoy vamos a profundizar en los pasos aplicables que puede dar ahora mismo para mejorar sus propios sitios de WordPress. También compartiremos algunos recursos con un valor incalculable.

CAPÍTULO 01:

Tipos de Sitios de WordPress: Estático o Dinámico



Antes de empezar con las optimizaciones, es importante entender primero que no todos los sitios de WordPress son iguales. Es por eso, que muchos usuarios tienen problemas, ya que no podemos resolver cada problema de la misma forma. Siempre damos a los sitios de WordPress una clasificación: estática o dinámica. Así que primero exploremos las diferencias entre estos dos tipos de sitios.

Sitios Principalmente Estáticos

Los estáticos, suelen ser sitios como los blogs, webs de pequeñas empresas, sitios de noticias pequeños, sitios personales, de fotografía, etc. Por estático, queremos decir que los datos en estos sitios de WordPress, **no cambian a menudo** (quizás un par de veces al día). Incluso la mayor parte de nuestro sitio de Kinsta, podría ser considerado un sitio estático.

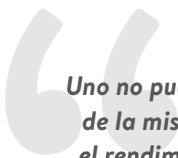
Esto es muy importante ya que muchas de las peticiones, pueden ser atendidas directamente desde la memoria caché en el servidor a velocidades increíbles. Pero no te preocupes; hablaremos mucho más sobre el tema de la caché más adelante. Esto quiere decir que tendremos menos consultas de la base de datos y no se requerirán tantos recursos para alcanzar un buen rendimiento en Google.

Sitios Altamente Dinámicos

Por otro lado, también tenemos sitios altamente dinámicos. Estos son sitios como los de eCommerce (WooCommerce o Easy Digital Downloads), de comunidad, de membresía, foros (bbPress o BuddyPress) y sistemas de administración de aprendizaje (learning management systems LMS). Con dinámico, queremos decir, que los datos en estos sitios de WordPress están **cambiando frecuentemente** (hay transacciones de servidor cada pocos minutos o incluso segundos). Esto quiere decir, que no todas las peticiones al servidor pueden ser servidas directamente desde la caché y requiere recursos de servidor adicionales, además de peticiones de la base de datos.

Estos sitios suelen tener un **gran número de visitantes y sesiones concurrentes**. En un sitio informativo o corporativo de WordPress, que normalmente es estático, un visitante podría permanecer por cinco o 10 minutos, hasta que encuentre lo que necesitaba (y esto es un número muy alto, normalmente el porcentaje de rebote es mucho mayor). En sitios dinámicos, sucederá lo opuesto. Los visitantes normalmente vienen al sitio para encontrarse con algo o alguien. Si están haciendo un curso online, no es raro que se queden varias horas.

Ya puedes ver a dónde estamos yendo con esto. El número de visitantes simultáneos conectados a su host de WordPress se irá acumulando rápidamente. Para empeorar aún más las cosas, tendrá un gran número de visitantes concurrentes, además de un problema de “contenido que no puede ser almacenado en la caché”.



Uno no puede tratar todos los sitios de WordPress de la misma forma cuando se trata de mejorar el rendimiento. ¡Los sitios estáticos y dinámicos son dos bestias muy distintas! .

CAPÍTULO 02:

Elija un Hosting de WordPress de Alto Rendimiento



Un host de WordPress, es una compañía que almacena todos los datos de su sitio web. Usted se inscribe a un plan y todas sus imágenes, contenido, videos, etc., serán almacenados en un servidor que se encuentra en el centro de datos del host. El host de WordPress le da un acceso fácil a estos datos, para poder administrarlos, y enviarlos a sus visitantes. Simple, ¿no? Bueno, no exactamente.

Hay tres tipos distintos de hosts de WordPress con los que se encontrará en internet. Veamos las ventajas y desventajas de cada uno. Es importante elegir el host correcto desde el principio, de otra manera, a la larga, le causará dolores de cabeza y tiempo perdido.

1. Hosting Compartido de WordPress

El primer y el más popular tipo de hosting de WordPress es al que llamamos “hosting compartido.” Estos incluyen los hosts más grandes en la industria, compañías como EIG, Bluehost y HostGator, también proveedores como Siteground, GoDaddy, y InMotion Hosting. Normalmente estos hosting utilizan cPanel y el cliente medio paga normalmente entre \$3 y \$25 al mes.

Cualquier persona que utilice este tipo de hosting descubrirá tarde o temprano, lentitud, es cuestión de tiempo. ¿Por qué? Porque **los hosts compartidos tienden a abarrotar sus servidores**, lo cual lleva a causar un impacto en el rendimiento de su sitio. Las

suspensiones de su sitio o los errores 500 continuos son cosas comunes que uno experimentará, ya que tienen limitaciones de espacio en casi todo, y consolidan recursos para sobrevivir. O aún peor, que su sitio quede temporalmente inactivo. A pesar de que aún no lo sabe, su sitio WordPress comparte espacio con 200 personas o más en su servidor. Cualquier problema que surja con otros sitios podrá afectar también al suyo.



Hosting Compartido de WordPress

No importa cómo haga las cuentas, después de los gastos, \$3 al mes no genera suficientes ingresos a la compañía de hosting. Especialmente cuando se agrega el soporte al servicio. En cuanto reciben un ticket de soporte ya están en números rojos. La forma en que ganan la gran mayoría de su dinero, es con ventas adicionales y cargos ocultos. Estas sobreventas, incluyen cosas como migraciones, registros de dominio, certificados SSL, etc. Otra táctica común es proveer grandes descuentos de registro. Pero una vez que llega la renovación del contrato, tendrás la factura real.

La mayoría de estos hosts ofrecen algo llamado “Recursos ilimitados”. Probablemente ya lo hayas visto. Bueno, en el mundo real, no existe tal cosa como “recursos ilimitados.” Lo que hacen los hosts, entre bambalinas es limitar a los clientes que estén utilizando demasiados recursos. Esto, en su lugar, termina causando que muchos clientes molestos se vayan, dejando espacio a clientes que no usan tantos recursos. Al final, se produce un círculo vicioso, en el que la compañía de hosting saca planes baratos, esperando a que la gran mayoría de sus clientes no usen demasiados recursos y terminen comprando los complementos.

El servicio y soporte al cliente con programas de hosting compartido, son de baja calidad, debido al alto volumen de sitios frente a los prestadores de soporte. Los hosts compartidos tienen que expandirse de tal forma, que sólo dejen lo suficiente para obtener los beneficios necesarios, y esto normalmente termina siendo una experiencia negativa para el cliente.

“Cuando se trata de hosting compartido, normalmente uno obtiene lo que paga.”

2. Hosting de WordPress con un VPS (Servidor Virtual Privado) Hecho por Usted (DIY)

El segundo tipo de hosting de WordPress es el DIY VPS, o “Hágalo usted mismo en un Servidor Virtual Privado.” Este grupo, normalmente está hecho de startups y usuarios con un poco más de experiencia en desarrollo y administración de servidores de WordPress. Estos son el grupo de DIY. Estas personas, generalmente, intentan ahorrar dinero, pero también les preocupa el rendimiento y se dan cuenta de lo importante que es para el éxito de su negocio. Las configuraciones comunes pueden incluir proveedores VPS como Digital Ocean, Linode, o Vultr; junto con una herramienta como ServerPilot, para administrar todo de forma más sencilla.

El precio de un pequeño VPS de DigitalOcean comienza en \$5 por mes y el plan más popular de ServerPilot empieza en \$10 al mes. Por lo que, dependiendo de la configuración, podría estar teniendo un gasto entre \$5 y \$15 al mes, quizás un poco más. El enfoque al DIY, puede ayudar a reducir los gastos, pero también significa que tendrá que hacerse responsable si algo se rompe, y tendrá que optimizar su servidor para obtener un buen rendimiento.

Este enfoque de DIY puede ser fantástico, pero también le puede causar problemas si no tiene cuidado. No elija este camino si no eres muy bueno con la tecnología o le gusta experimentar. Su tiempo es dinero, y debería usarlo para hacer crecer su negocio.

3. Hosting Administrado para WordPress

El tercer tipo de hosting es lo que ofrecemos en Kinsta y es el hosting administrado para WordPress. Este tipo de hosts pueden lidiar con todo lo relacionado con el servidor y back-end, y al mismo tiempo brindamos soporte cuando lo necesite. Hecho a la medida para trabajar con WordPress, con opciones como, entorno de pruebas y copias de seguridad automáticas. Sus equipos de soporte tendrán mayor conocimiento del CMS, ya que están enfocados en la misma plataforma (WordPress) todos los días.

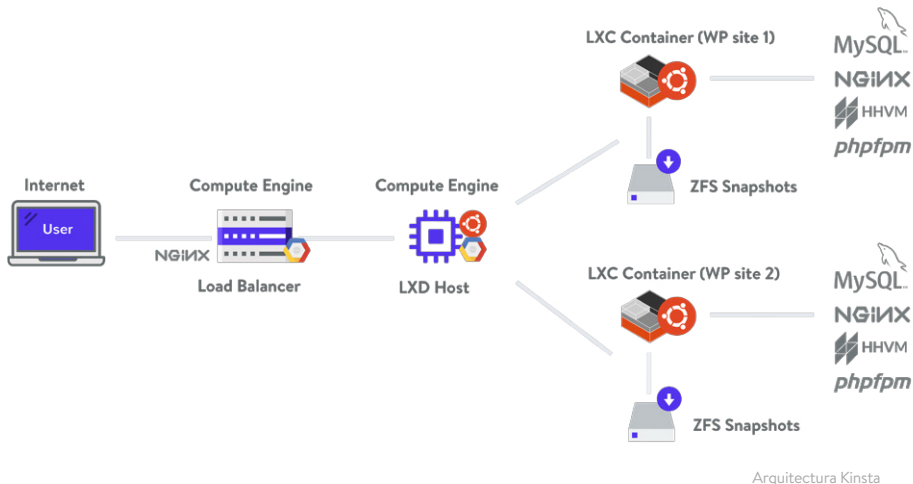
¡Si quiere ahorrar tiempo, el hosting administrado de WordPress es la mejor opción!

Los planes por un hosting administrado normalmente van desde los \$25 hasta los \$150 al mes, dependiendo del tamaño de su sitio y sus necesidades. Las grandes compañías como jQuery, Intuit, Plesk, Dyn, NGINX, e incluso la Casa Blanca utilizan WordPress para hospedar sus sitios web. Algunos hosts administrados de WordPress que seguramente ya conozca o quizás ya esté utilizando son: WP Engine, Flywheel, Pressable, Media Temple, Pressidium y Pagely.

Kinsta Toma un Enfoque Diferente

Kinsta por otro lado, lleva el hosting administrado de WordPress al siguiente nivel. Nuestra plataforma de hosting no entra en ninguna categoría de hosting tradicionales. Nuestra infraestructura está construida sobre Google Cloud Platform y es diferente a la infraestructura compartida, VPS o dedicadas tradicionales.

Cada Sitio de WordPress, en nuestra plataforma, funciona en un contenedor de software aislado que contiene todos los recursos necesarios para ejecutar el sitio (Linux, NGINX, PHP, MySQL). Esto quiere decir que el software que ejecuta cada sitio es completamente privado y no es compartido ni entre sus propios sitios.



Cada contenedor de sitio funciona en máquinas virtuales en uno de múltiples centros de datos de Google Cloud. Cada máquina tiene hasta 96 CPUs y cientos de GB de RAM. Los recursos de hardware (RAM/CPU) son asignados automáticamente a cada contenedor del sitio web por nuestras máquinas virtuales, cada vez que sea necesario (una excelente opción a la que llamamos autoescalado).

Cada año, Review Signal lanza sus evaluaciones de rendimiento de hosting de WordPress, y nos enorgullece decir que, por cinco años consecutivos, ¡Kinsta ha demostrado ser la mejor compañía en todos los niveles! Y no sólo para uno o dos de nuestros planes, sino para todos planes, desde el Básico, hasta el de nivel Empresarial.

“Kinsta ha hecho prácticamente perfectas las pruebas de LoadStorm y Blitz. Tampoco han tenido fallos en otras pruebas. No se me ocurre qué más decir sobre su excelente rendimiento.”



Kevin Ohashi
Fundador y consultor WP, ReviewSignal

Tampoco tenemos agentes de soporte de nivel 1 o nivel 2. Nuestro equipo de soporte completo está compuesto de desarrolladores de WordPress e ingenieros de hosting de Linux, muchos de los cuales han administrado sus propios servidores,

han creado temas y plugins, y han contribuido al núcleo de WordPress. Esto asegura que reciba soporte experto de alguien que utiliza y desarrolla activamente con WordPress.

Puede conversar con los mismos miembros del equipo de soporte que respaldan clientes Fortune 500 y grandes empresas. Somos muy quisquillosos sobre la calidad de nuestro equipo de soporte, así que sólo contratamos a menos del 1% de la gente que aspira al puesto. ¡No encontrarás mejor soporte en ningún lado!

Para aprender más sobre por qué debe elegir Kinsta para su hosting administrado de WordPress, visite nuestra web en kinsta.com. Pero independientemente de a quién elija como proveedor de hosting, siempre debe buscar estas características de servidor, para asegurarse que su web funcione lo más rápido posible.

“Con WP Engine, los problemas básicos generalmente se resuelven rápido. Sin embargo, para cualquier problema que sea complejo, la solución tardará un poco más, y tendrá que interactuar bastante con el equipo de soporte. Esto es un problema cuando tenemos un sitio de WordPress de alto nivel, y hay un problema urgente que tiene que resolverse pronto. Si me pregunta cuál de estos dos recomiendo, en mi opinión, Kinsta es la mejor opción. Ellos ofrecen mucho más de lo que prometen. No tendrá que preocuparse si su sitio va lento, o se encuentre inactivo, obtendrá soporte de calidad a cualquier problema relacionado con el hosting.”

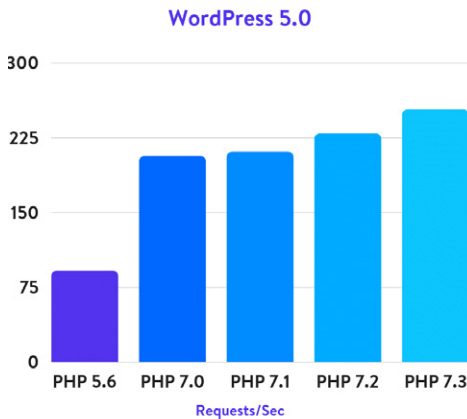


Harsh Agrawal
Blogger premiado, ShoutMeLoud

PHP 7 en Adelante para el Mejor Rendimiento

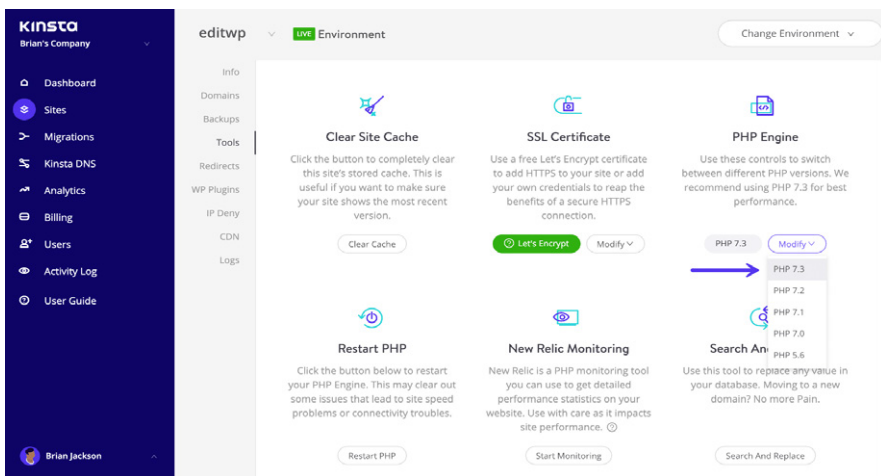
PHP es un lenguaje de código abierto, de scripting y programación de servidor que es usado principalmente para el desarrollo web. La mayor parte del software principal de WordPress está desarrollado en PHP, junto con sus plugins y temas, que hace de PHP un lenguaje muy importante para la comunidad de WordPress. Debe asegurarse de que su host de WordPress ofrezca por lo menos PHP 7 o versiones superiores.

Hay diferentes versiones de PHP que su host le proporcionará en su servidor, el nuevo PHP 7.3 ofrece enormes mejoras de rendimiento.



De hecho, en nuestras recientes pruebas de rendimiento de PHP, si compara PHP 7.3 con PHP 5.6, el primero puede manejar 3 veces más solicitudes (transacciones) por segundo. PHP 7.3 también es en promedio un 9% más rápido que PHP 7.2. Esto también puede afectar la capacidad de respuesta de su escritorio de WordPress.

Velocidades más rápidas y seguridad mejorada, es la razón por la que Kinsta siempre ofrece las versiones más recientes de PHP. Puede cambiar las versiones de PHP con un solo clic.



Desconfíe de cualquier host de WordPress que ofrezca HHVM como una alternativa a PHP. HHVM ya no es una solución viable para el hosting de WordPress ya que no admite oficialmente PHP.

Elija un Host que Utilice **NGINX**

Detrás de escenas, cada host de WordPress utiliza un servidor web, para que sus sitios de WordPress funcionen. Las opciones más comunes son NGINX y Apache.

Le recomendamos elegir un host que utilice NGINX por tener raíces en la optimización del rendimiento a escala. NGINX normalmente supera a otros servidores populares en pruebas de rendimiento, especialmente en situaciones de contenido estático o cuando se tiene un alto número de peticiones concurrentes, por eso, Kinsta utiliza NGINX.

Algunas compañías importantes utilizan NGINX, incluyendo Autodesk, Atlassian, Intuit, T-Mobile, GitLab, DuckDuckGo, Microsoft, IBM, Google, Adobe, Salesforce, VMWare, Xerox, LinkedIn, Cisco, Facebook, Target, Citrix Systems, Twitter, Apple, Intel, y muchos más. (fuente: [Sifter.com](https://sifter.com))

Según W3Techs, Apache es usado por un 44.0% de los sitios web, haciéndolo la opción más usada. Pero si ve cuál es el servidor web más popular entre los sitios de alto tráfico (top 10,000), verá que NGINX es usado por un 41.9%, de estos sitios mientras que Apache solo impulsa el 18.1%. Es usado por algunos de los sitios que más recursos consumen, incluyendo Netflix, NASA e incluso el mismo WordPress.com.

La Red Usada por Su Host Importa

Al elegir un host de WordPress podría pensar que no tendría que pedir o investigar qué red están utilizando, pero eso es lo que tiene que hacer. La red tiene un alto impacto en el rendimiento de su sitio web e incluso la rapidez del escritorio de su WordPress. Muchos hosts dejarán esto fuera de su marketing, ya que preferirán elegir la red más barata para recortar gastos.

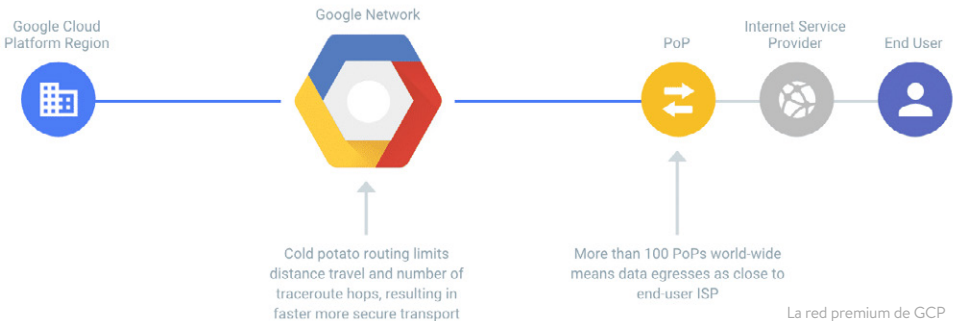
Aquí tenemos algunas preguntas que debería hacer:

- **¿Qué redes utiliza para transmitir datos?** ¿La mayoría es a través de redes ISP públicas o infraestructuras privadas como Google o Microsoft? Estos grandes proveedores tienen redes que están construidas y optimizadas para conseguir baja latencia y alta velocidad. ¡Incluso tienen sus propios cables de internet en las profundidades del océano!
- **¿Acaso las redes que están utilizando son redundantes?** ¿Qué sucede si un cable es cortado accidentalmente? Esto sucede más comúnmente de lo que piensa.

En el 2017 Google anunció su red estándar, que es una red más lenta, pero más barata. En Kinsta utilizamos **la red premium** para todos nuestros planes de hosting. A pesar de que esto nos cuesta más a nosotros, asegura que usted tendrá velocidades más rápidas.

Kinsta utiliza la red premium de Google Cloud Platform porque su sitio merece lo mejor.

De acuerdo con Google, la red premium logra un desempeño de red mejorada, al reducir la duración del viaje en el internet público; los paquetes entran (y se van) a la red de Google lo más cerca posible al usuario y luego viajan en la “columna” de Google antes de entrar a la máquina virtual. El nivel estándar, ofrece tráfico saliendo desde el GCP al proveedor de internet sobre las redes de transporte público (ISP) en lugar de la red de Google.



En pocas palabras, para que sea más fácil de entender:

- **Los paquetes premium pasan más tiempo en la red de Google**, con menos rebotes, y así funcionan mucho mejor (pero cuestan más).
- Los paquetes estándar pasan menos tiempo en la red de Google, y más tiempo en redes públicas y por eso, tienen un peor desempeño (pero cuestan mucho menos).

¿Qué impacto tiene esto? Bueno, para los datos que viajan a través de los continentes, su red de nivel premium es aproximadamente un **41% más rápida**, en promedio, que la red de nivel estándar. Para los datos viajando a una región cercana (mismo continente), la red premium es un **8% más rápida**. Mientras que las redes sólo toman una fracción de su tiempo de la carga total, ¡cada segundo se acumula!

Para los datos que atraviesan continentes, la red premium de Google Cloud es en promedio, ¡un 41% más rápida!

La redundancia también es clave, y es por eso que Google utiliza por lo menos tres caminos independientes (N+2 de redundancia) entre dos ubicaciones en la red de Google, asegurando que el tráfico siga fluyendo entre dos ubicaciones, incluso en caso de una interrupción.

Como podrá ver ahora, mucho de esto pasa detrás del escenario, cuando se trata de las redes. Asegúrese de que su host de WordPress esté utilizando una red prestigiosa y no esté eligiendo las de menor categoría para ahorrar dinero.

HTTP/2 es Necesario

HTTP/2 es un protocolo web lanzado en 2015, el cual fue diseñado para acelerar la entrega de los sitios web. Debido al soporte del navegador, este requiere HTTPS (SSL).

Si su host de WordPress no soporta HTTP/2, usted debe de buscar un nuevo proveedor. Con todo internet mudándose a HTTPS, esto ya no es solo una buena opción; es una necesidad.

La mejora en el desempeño con HTTP/2 es debido a una variedad de razones, como un mejor soporte a la multiplexación, paralelismo, compresión HPACK con decodificación Huffman, la extensión ALPN, y empuje del servidor. Solía haber un poco de demora por la TLS cuando se trataba de correr sobre HTTPS, pero esto ya no pasa tan seguido gracias a HTTP/2 y TLS 1.3.

Otro gran logro con HTTP/2 es que con la mayoría de sitios de WordPress, usted no tendrá que preocuparse sobre concatenación (combinar archivos) o fragmentación de dominios. Estas ya son optimizaciones obsoletas.

CAPÍTULO 03:

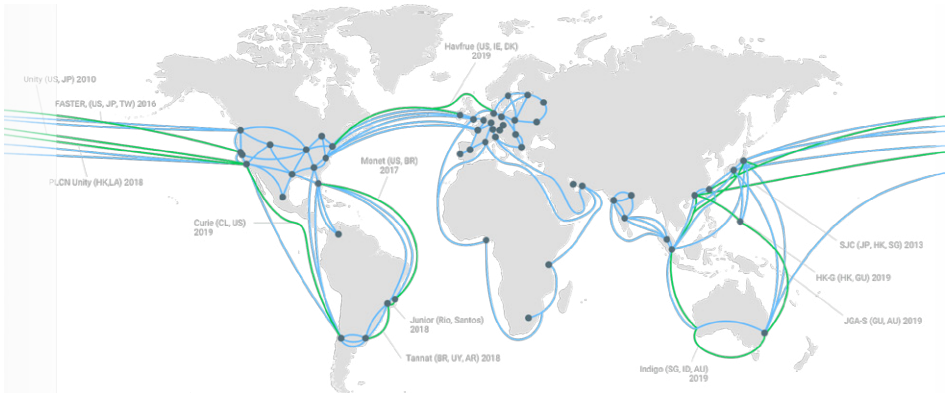
Elija un Servidor Cerca de sus Visitantes



Una de las primeras cosas que debería hacer al hospedar su sitio de WordPress es determinar de dónde viene la mayoría de sus visitantes o clientes. ¿Por qué es importante esto? Porque la ubicación del host de su sitio web juega una parte importante en determinar la latencia total de su red y el TTFB. También impacta en las velocidades de SFTP y el tiempo de respuesta del escritorio de admin de su WordPress.

Latencia de la Red: Se refiere al tiempo y/o retraso que sucede en la transmisión de datos sobre una red. En otras palabras, cuánto tiempo le toma a un paquete de datos ir de un punto a otro. Hoy día, esto es comúnmente medido en milisegundos; sin embargo, podrían ser segundos dependiendo de la red. Cuanto más cercano a cero esté, mejor será.

En otras palabras, cuanto mayor sea la distancia, mayor será la latencia, lo que equivale a un retardo mayor. No se puede superar la velocidad de la luz cuando se trata de datos que viajan por cables de fibra óptica. Por ejemplo, a continuación se muestra un mapa de la red Google Cloud Platform. Tienen miles de kilómetros de cable que van por debajo del océano y que atraviesan todo el mundo. Esta es una de las razones por las que es una de las redes más rápidas del mundo. Incluso con esta extensa red de última generación, siempre habrá retrasos de latencia.



Hay un par de factores que contribuyen a los retrasos de latencia de la red, entre los que se incluyen:

- **Retraso en la transmisión:** Los diferentes tipos de medios, como las conexiones inalámbricas o de fibra óptica, introducen algún tipo de retardo, ya que sólo pueden expulsar una cantidad limitada de bits. Por ejemplo, una conexión de fibra óptica tendrá un retardo de transmisión menor que una línea T1. También debe tener en cuenta el tamaño del paquete.
- **Retraso en la propagación:** Este es el tiempo que tarda el paquete de datos en viajar. Esto podría incluir la distancia, los retrasos en la red, etc.
- **Retraso en la cola:** Si se excede el ancho de banda, puede haber retrasos en la cola en los que los datos tienen que esperar en el host o en el router. Esto puede verse afectado por la congestión de la red.


Algunos podrían argumentar que la latencia de la red no es muy importante; sin embargo, puede serlo en lo que respecta al lugar donde usted elige alojar su sitio de WordPress. Para los medios y sus archivos (como imágenes, JavaScript, CSS, Video), una CDN puede corregir gran parte de la latencia adicional involucrada, al entregar una copia desde un servidor más cercano. Sin embargo, en la mayoría de las configuraciones, todavía tiene que consultar a su servidor de alojamiento para la carga inicial de DOC. Y es por eso que la latencia es importante!

La latencia es especialmente importante cuando se trata de negocios que atienden a visitantes en una ubicación geográfica específica. Por ejemplo, digamos que usted tiene una tienda de comercio electrónico en Sydney, y el 90% de sus clientes son de

Australia. Su empresa se beneficiaría de colocar su sitio en un servidor en Australia, en lugar de alojarlo en Europa o en los Estados Unidos

TTFB: Es el acrónimo para time to first byte (tiempo para el primer byte). Dicho de forma más sencilla, muestra cuánto tiempo tiene que esperar el navegador antes de recibir el primer byte de dato desde un servidor. Entre más tiempo pase para obtener el dato, más tiempo tomará mostrarlo en la página. De nuevo, cuanto más cercano esté a cero mucho mejor.

Un error común es calcularlo después de los tiempos de búsqueda de DNS, sin embargo, el cálculo original de TTFB en la red siempre incluye la latencia de la red. Esto implica un proceso de 3 pasos y los retrasos y la latencia pueden ocurrir en cualquier punto intermedio, lo que se suma al TTFB total.

| Connection Setup | | TIME |
|-----------------------------|---|------------------|
| Queueing | | 4.56 ms |
| Stalled | | 1.35 ms |
| Request/Response | | TIME |
| Request sent | | 0.11 ms |
| Waiting (TTFB) |  | 291.39 ms |
| Content Download | | 1.47 ms |
| Explanation | | 298.87 ms |

1. Solicitud al servidor Cuando alguien visita su sitio web, lo primero que sucede es que se envía una petición HTTP desde el cliente (navegador) al servidor. En este paso, hay una variedad de factores que pueden introducir retrasos. Los tiempos de búsqueda lentos de DNS podrían contribuir a aumentar el tiempo de la solicitud. Si el servidor está localizado geográficamente lejos, esto puede introducir latencia por la distancia que los datos tienen que viajar. Además, si tiene reglas de cortafuegos complejas, esto podría aumentar el tiempo de enrutamiento. Y no olvide la velocidad de Internet del cliente.

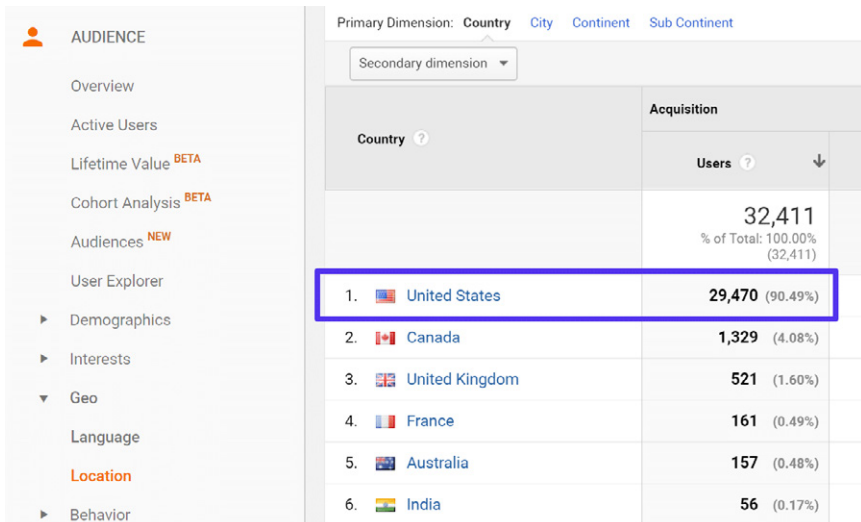
2. Procesamiento del Servidor Una vez enviada la solicitud, el servidor debe procesarla y generar una respuesta. Esto podría introducir una serie de retrasos diferentes, como llamadas lentas a la base de datos, demasiados scripts de terceros, no almacenar en caché su primera respuesta, código mal optimizado o tema de WordPress y recursos de servidor ineficientes como E/S de disco o memoria.

3. Respuesta al Cliente Después de que el servidor procesa la petición, tiene que devolverla al cliente (o mejor dicho, devolver el primer byte). Esto se ve muy afectado tanto por la velocidad de la red del servidor como por la del cliente. Si el cliente tiene Internet lento desde un hotspot Wi-Fi, esto va a reflejarse en el TTFB.

Generalmente, **cualquier cosa por debajo de 100 ms es un buen TTFB**. Google Page-Speed Insights recomienda menos de 200 ms para el tiempo de respuesta del servidor. Si usted está en el rango de 300-500 ms, está en el rango más común. Y si tiene más de 600 ms, es posible que tenga algo mal configurado en su servidor o que sea el momento de actualizar a un mejor Web Stack. O siga las siguientes recomendaciones sobre cómo reducir su TTFB. Y recuerde que la negociación SSL/TLS también puede ser un factor.

No le aburriremos con todos los detalles técnicos en esta publicación, todo lo que necesita saber es que **necesita que la latencia de su red y el TTFB sean lo más bajos posible**. Una de las maneras más fáciles de lograrlo es elegir el servidor más cercano a sus visitantes. Puede determinar la mejor ubicación siguiendo los consejos que enumeramos a continuación.

Consejo 1 – Revise la Geolocalización de sus Visitantes en Google Analytics



The screenshot shows the Google Analytics interface for an Audience report. The primary dimension is set to 'Country'. The table displays the top 6 countries by user count. The United States is highlighted with a blue box, showing 29,470 users (90.49% of the total).

| Primary Dimension: Country | | Acquisition | |
|----------------------------|----------------|-------------|------------------------------|
| Country | | Users | |
| | | 32,411 | % of Total: 100.00% (32,411) |
| 1. | United States | 29,470 | (90.49%) |
| 2. | Canada | 1,329 | (4.08%) |
| 3. | United Kingdom | 521 | (1.60%) |
| 4. | France | 161 | (0.49%) |
| 5. | Australia | 157 | (0.48%) |
| 6. | India | 56 | (0.17%) |

Una de las primeras comprobaciones que tiene que hacer es ver la geolocalización de sus visitantes en Google Analytics. Puede encontrarlo en “Audiencia → Información Geográfica → Ubicación.”

En el ejemplo siguiente, puede ver que más del 90% del tráfico viene de los Estados Unidos. Así que, en la mayoría de los casos, usted querrá ubicar su sitio de WordPress en un servidor en los Estados Unidos. También puede filtrar los datos por ciudad. Esto es especialmente importante si usted es una compañía local. Pero normalmente recomendamos una ubicación central como Iowa, USA.

Consejo 2 – Consulte los Datos de Ecommerce

Si tiene una tienda de eCommerce, asegúrese de revisar **y observar de dónde vienen sus clientes**. Así es como uno genera ingresos, así que estos son sus visitantes más importantes. Esto debería coincidir con el tráfico de arriba; sin embargo, esto no siempre es así. Si tiene los datos configurados de su eCommerce o metas en Google Analytics, usted puede fácilmente agregar esa información sobre los datos de geolocalización, para tomar una decisión más precisa. O consulte la información de ubicación almacenada en las bases de datos de su plataforma de eCommerce.

Consejo 3 – Haga una Prueba de Latencia

Hay muchas herramientas útiles para medir la latencia de su ubicación actual para distintos proveedores de la nube. Esta información puede ayudarle a evaluar qué región sería la mejor opción para su sitio.

- **gcping.com**: (mide latencia a regiones de Google Cloud Platform, incluyendo los servidores de Kinsta)
- **cloudping.info**: (mide la latencia de las regiones de Amazon Web Services)
- **azurespeed.com**: (mide la latencia de las regiones de Azure)

En el siguiente ejemplo, podemos ver que Oregon, USA (us-west1) es la región más rápida. Sin embargo, si está sirviendo a clientes en todas partes de Estados Unidos, podría ser mejor elegir Iowa, USA (us-central1) para asegurar una baja latencia para ambos lugares, visitantes de la costa oeste y este.

Mida su latencia a las regiones de los GCP

| REGION | MEDIAN LATENCY |
|-------------------------------------|----------------|
| Los Angeles, USA us-west2 | 28ms |
| Global HTTP Load Balancer global | 31ms |
| Oregon, USA us-west1 | 53ms |
| Northern Virginia, USA us-east4 | 82ms |
| Iowa, USA us-central1 | 83ms |
| South Carolina, USA us-east1 | 88ms |

En Kinsta, ofrecemos 20 centros de datos distintos alrededor del mundo. ¡Podrá elegir fácilmente un sitio que tendrá baja latencia y TTFB! Esto también ayudará a reducir los saltos de la red.



Formas Adicionales para Reducir la Latencia y el TTFB

Además de elegir un servidor con una ubicación más cercana, hay otras formas para reducir la latencia.

- Implementar una caché en su sitio de WordPress. ¡En nuestras pruebas la caché redujo nuestro TTFB por un sorprendente 90%!
- Utilice una red de entrega de contenido (CDN) para servir archivos almacenados en la caché de POPs alrededor del mundo. Esto ayudará a reducir la latencia de la red para visitantes que no se encuentran cerca del servidor de su host.
- Tome ventaja del protocolo HTTP/2 para minimizar el número de viajes en círculo, gracias al proceso de paralelización. HTTP/2 está habilitado en todos los servidores de Kinsta.
- Reduzca el número de peticiones HTTP. Ya que cada una puede tener su propia latencia, basada en la ubicación de su servidor.
- El DNS juega una parte importante en el TTFB, así que debería utilizar un proveedor de DNS premium con tiempos de búsqueda veloces.
- Utilice precarga y previsualización para llevar a cabo tareas detrás de escenas mientras la página carga.

No se preocupe, nosotros cubriremos todas las recomendaciones mencionadas anteriormente a continuación.

Velocidades SFTP y el Panel de Control de WordPress

Sus visitantes y clientes siempre deberían ser su prioridad. Pero otro aspecto del rendimiento del que muchos no hablan es **cómo algunas de estas decisiones afectan a su trabajo diario**. La ubicación del centro de datos que elija tiene un impacto la velocidad de sus descargas en SFTP y las velocidades de subida (transfiriendo datos con un cliente FTP), al igual que la capacidad de respuesta del escritorio de WordPress.

Mientras se asegura y elige una ubicación que sea la mejor para sus visitantes, también tenga en mente que esto puede afectar al mantenimiento de su sitio. Las tareas, tal como subir archivos a la biblioteca de archivos multimedia de WordPress, serán más rápidas cuando su sitio tenga como host un centro de datos cercano a usted.

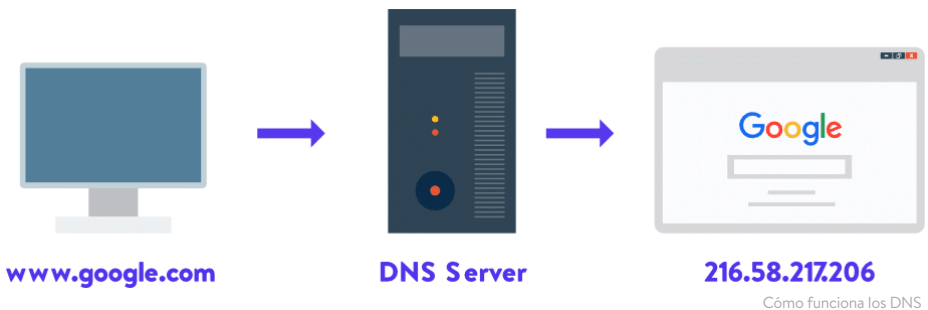
Consistentemente escuchamos a nuestros clientes en Kinsta que les sorprende lo rápido que responde su escritorio de administración con nosotros. Hay una multitud de factores que influyen en ello, pero tener 20 centros de datos distintos es una razón muy grande. Elija una ubicación que funcione para usted y sus visitantes. Después de todo, usted será el que pase miles de horas trabajando en su sitio web.

CAPÍTULO 04:

Un DNS Premium es Mucho Mejor que un DNS Gratuito



DNS, acrónimo para Domain Name System, es uno de los componentes más comunes y al mismo tiempo más incomprensidos del entorno de la red. Dicho de forma sencilla, el DNS ayuda a dirigir el tráfico en internet, al conectar nombres de dominio con servidores de la red. Esencialmente, toma una petición humana – el nombre de un dominio como `kinsta.com` – y lo traduce a una dirección IP de servidor amigable para computadoras – como `216.58-217.206`.



Puede encontrar DNS gratuitos y Premium. Todos los clientes de Kinsta obtienen acceso a un DNS premium a través de Amazon Route 53. Nosotros creemos que tener un DNS premium es una necesidad en el mundo moderno.

Una gran razón por la que uno debe elegir un DNS premium **es la velocidad y confiabilidad**. Buscar registros DNS y dirigir tráfico toma tiempo, incluso si es cuestión de milisegundos.

Normalmente, La velocidad de un DNS gratuito obteniendo un registro de nombre de dominio es realmente lento, mientras que un DNS premium ofrece un mejor rendimiento. Por ejemplo, en nuestras pruebas, descubrimos que el **DNS de NameCheap es un 33% más lento** que Amazon Route 53, siendo Amazon Route 53 un DNS premium. Además, un DNS premium puede ofrecer una mejor seguridad y disponibilidad, especialmente cuando se encuentra bajo un ataque DDoS.

Puede utilizar una herramienta como SolveDNS speed test para revisar los tiempos de búsqueda de su DNS. DNSPerf también proporciona excelentes datos de rendimiento en los principales proveedores de DNS.

Un buen término medio entre un proveedor de DNS gratuito y un DNS premium es Cloudflare, Cloudflare es un servicio gratuito que ofrece muchos de los beneficios de un DNS premium. Y son muy rápidos, con un promedio debajo de 20ms de tiempo de respuesta alrededor del mundo (como se puede observar abajo).

| Name Server | Result | Average (milliseconds) | Standard Deviation | 95% Confidence Interval |
|----------------------------------|---|------------------------|--------------------|-------------------------|
| algin.ns.cloudflare.com | <ul style="list-style-type: none"> ✔ Los Angeles: 5.65 ms ✔ Dallas: 6.04 ms ✔ New York: 3.47 ms ✔ Singapore: 3.29 ms ✔ London: 3.81 ms ✔ Amsterdam: 4.04 ms ✔ San Francisco: 12.4 ms ✔ Sydney: 6.57 ms ✔ Tokyo: 4.02 ms | 5.48 ms | 2.69 | [3.49, 7.47] |
| heather.ns.cloudflare.com | <ul style="list-style-type: none"> ✔ Los Angeles: 1.55 ms ✔ Dallas: 3.98 ms ✔ New York: 2.06 ms ✔ Singapore: 2.11 ms ✔ London: 2.95 ms ✔ Amsterdam: 2.17 ms ✔ San Francisco: 12.04 ms ✔ Sydney: 1.51 ms ✔ Tokyo: 2.62 ms | 3.44 ms | 3.12 | [1.13, 5.75] |

Sin embargo, algo que tenemos que advertirle de Cloudflare es que también tiene mayor tiempo de inactividad que muchos de los otros proveedores. Si usted está sirviendo principalmente a visitantes en los Estados Unidos, DNS Made Easy es otro proveedor de DNS premium que podría revisar. Tienen la reputación de brindar uno de los mejores tiempos de respuesta de cualquier DNS en la última década

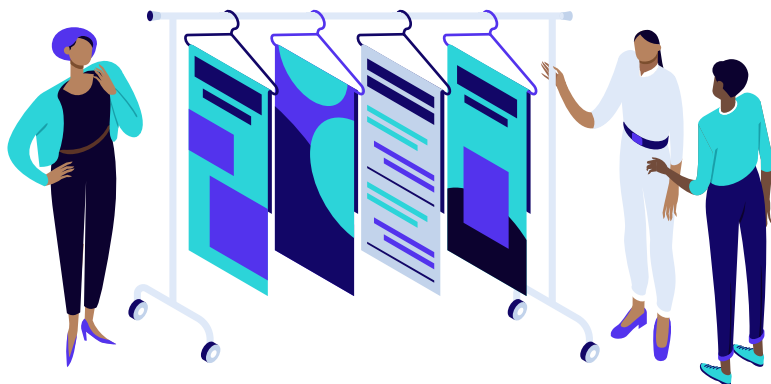
En los últimos 30 días, DNSPerf muestra el siguiente tiempo de actividad de estos proveedores:

- DNS Made Easy: 99.99% el cual equivale a 4m 23.0s de inactividad al mes.
- Amazon Route 53: 99.88% lo cual equivale a 52m 35.7s de inactividad al mes.
- Cloudflare: 99.85% el cual equivale a 1h 5m 44.6s de inactividad al mes.

¿Es importante el tiempo de inactividad para los proveedores de DNS? La respuesta es, sí y no. El DNS normalmente se encuentra almacenado en la cache del ISPs, utilizando el tiempo de vida (TTL) en el registro DNS. Por lo tanto, si un proveedor de DNS se encuentra inactivo por 10 minutos, probablemente no lo notará. El tiempo de inactividad importa si hay tiempos de inactividad constantes de parte del proveedor, o si los registros de su ISP y DNS muestran bajos valores de TTL.

CAPÍTULO 05:

El Tema de WordPress es Importante



Todos aman un nuevo tema de WordPress, pero tenga cuidado antes de elegir elegir uno con todas las nuevas y deslumbrantes opciones. Primero, debería revisar nuestro artículo sobre las diferencias entre temas gratuitos y de pago. Tratándose del rendimiento, cada elemento que usted ve en un tema, tiene algo de impacto en la velocidad total de su sitio web. Y desafortunadamente, con miles de temas a elegir, habrá buenas y malas elecciones.

Su tema de WordPress es importante para el rendimiento. Elija el correcto desde el principio.

¿Cómo se supone que sabemos cuál elegir? Recomendamos elegir una de estas dos opciones:

- Un tema rápido y liviano de WordPress que este hecho sólo con las opciones que usted necesita, y nada más.
- Un tema más completo de WordPress, pero donde pueda desactivar las opciones que no utilice.

Cosas como Google Fonts, Font Awesome icons, sliders, galerías, video y parallax scripts, etc. Estas son algunas de tantas cosas que debería poder apagar si no las está utilizando. No querrá tener que arreglarlas manualmente en otro momento. Y no le mostraremos 50 formas diferentes para hacer las cosas más sencillas. En su lugar, usted debería cambiarse a un tema de WordPress más liviano desde el principio o uno que le de estas opciones.

A continuación, le mostraremos un par de temas de WordPress que recomendamos. Confíe en nosotros, nos lo agradecerá en el futuro.

Cada tema mencionado a continuación, es completamente compatible con WooCommerce y Easy Digital Downloads, WPML, BuddyPress, y bbPress. Hicimos algunas pruebas de velocidad con cada tema, utilizando la siguiente configuración:

- Alojado en Kinsta, funcionando con WordPress 5.0
- PHP 7.3 y SSL (HTTPS)
- Kinsta CDN
- Imagify fue utilizando para comprimir automáticamente las imágenes

GeneratePress

GeneratePress es un tema de WordPress rápido, liviano (menos de 1MB en zip), móvil, hecho teniendo en cuenta la velocidad, SEO y usabilidad. Hecho por Tom Usborne, un desarrollador de Canadá quien activamente lo actualiza y bien soportado. Incluso algunos de los miembros del equipo de Kinsta utilizan GeneratePress para sus proyectos.

Hay versión gratuita y premium. Si ve su repositorio de WordPress, la versión gratuita actualmente tiene más de 200.000 instalaciones activas, y más de 2 millones de descargas, y una impresionante calificación de 5 de 5 (con más de 850 personas calificando con 5 estrellas).

Una de las cosas geniales sobre GeneratePress es que todas las opciones utilizan el personalizador de WordPress, de esta forma podrá ver todos cambios que haga de forma instantánea antes de presionar el botón de publicar. Esto también quiere decir que no tendrá que aprender un nuevo panel de control de temas.

The perfect **lightweight** theme for your next project.

[Install](#)[Premium](#)

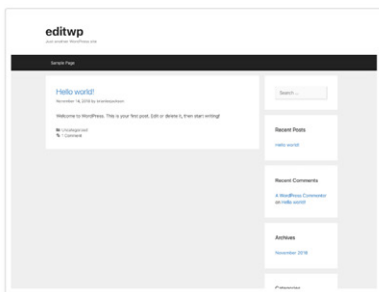
1,643,956+
Downloads

700+
★★★★★

100,000+
Active websites

40,000+
Happy customers

¿Qué tan rápido es? Hicimos una instalación nueva de GeneratePress, corrimos cinco pruebas en Pingdom y tomamos el promedio. El tiempo total de carga fue de **305 ms** con un **tamaño total de página de tan solo 16.8 KB**. Siempre es bueno tener una prueba base para ver lo que puede hacer un tema en términos de rendimiento puro.



Performance grade

A 97

Page size

16.8 KB

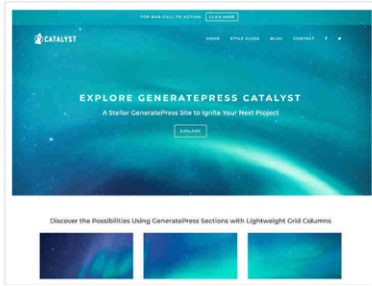
Load time

305 ms

Requests

7

Luego hicimos otro conjunto de pruebas, con uno de los temas preconstruídos de la biblioteca del sitio de GeneratePress. Este contiene imágenes, fondos, nuevas secciones, etc. Una ventaja que tiene GeneratePress es que tiene muchos temas preconstruídos que no requieren un plugin de construcción de páginas. Puede ver que aún así, se mantiene por debajo de **400 ms**.



Performance grade
B 88

Page size
837.5 KB

Load time
396 ms

Requests
36

Por supuesto, en un entorno de la vida real, podría tener otras cosas activas, como Google Analytics, Facebook remarketing pixel, Hotjar, etc. Pero usted debería aspirar a conseguir una resultado por debajo de 1 segundo.

A continuación, le mostraremos más formas de optimizar y acelerar WordPress.

OceanWP

OceanWP. Demos Extensions Blog Support Download Sign In / Sign Up

Create Beautiful Websites

With 899,712 downloads, OceanWP is the fastest growing theme for WordPress. It's the favorite choice of thousands of developers and hobby-users.

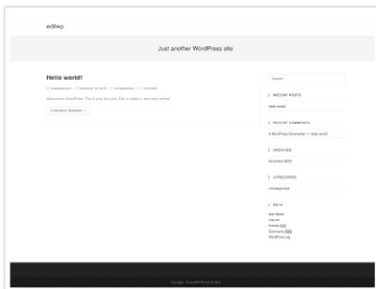
FREE DOWNLOAD

DEVELOPER **HOBBYIST**

El tema de OceanWP es liviano y altamente extensible. Le permite crear casi cualquier tipo de sitio web, desde un blog, un portafolio, un sitio de un negocio o un escaparate de WooCommerce, con un diseño hermoso y profesional. Creado por Nicolas Lococq, es actualizado y soportado de forma activa.

Como con GeneratePress, hay versiones gratuitas y premium. Si ve su repositorio de WordPress, la versión gratuita actualmente tiene más de 400.000 instalaciones activas, y otra impresionante calificación de 5 de 5 (con más de 2.600 personas que le han otorgado 5 estrellas).

¿Qué tan rápido es? Hicimos una instalación nueva de OceanWP, le hicimos cinco pruebas de velocidad en Pingdom, y tomamos el promedio. El tiempo total de carga fue de **389 ms** con un **tamaño total de página de tan solo 230.8 KB**. Los scripts en OceanWP son un poco más grandes, pero nada fuera de lo común.



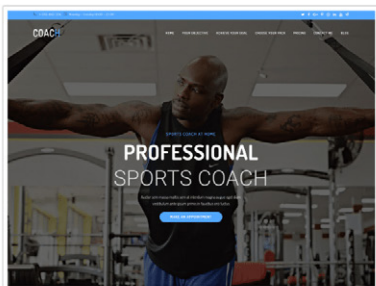
Performance grade
A 92

Page size
230.5 KB

Load time
389 ms

Requests
16

Luego hicimos otra serie de pruebas con uno de los temas demo de la biblioteca de OceanWP. Este contiene imágenes, fondos, nuevas secciones y el plugin constructor de páginas requerido, Elementor. Podrá ver que sigue estando por debajo de **600 ms**.



Performance grade
B 84

Page size
941.1 KB

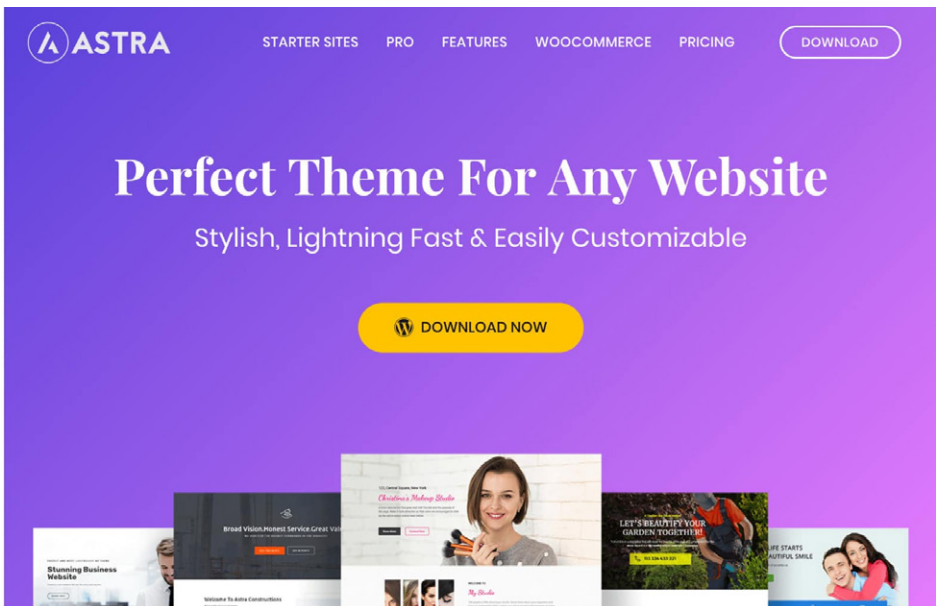
Load time
548 ms

Requests
48

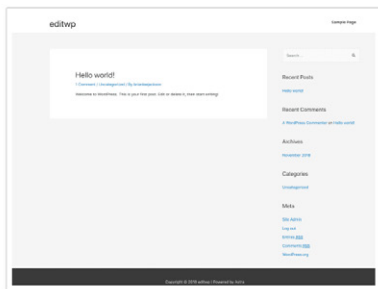
Astra

Astra es un rápido, completamente configurable y bello, perfecto para blogs, portafolios personales, sitios de negocios, y tiendas de WooCommerce. Es muy liviano (pesando menos de 50 KB en el front-end) y ofrece una velocidad inigualable. Hecho por el equipo de Brainstorm Force, es activamente actualizado y soportado. Podrá reconocerlos como los creadores del popular plugin All In One Schema Rich Snippets que lleva online desde hace varios años.

Como con GeneratePress y OceanWP, hay versiones gratuitas y premium. Si ve su repositorio de WordPress, la versión gratuita actualmente tiene más de 400.000 instalaciones activas, más de 1.6 millones de descargas, y otra impresionante calificación de 5 de 5 (con más de 2500 personas que le han otorgado 5 estrellas).



¿Qué tan rápido es? Hicimos una instalación nueva de Astra, le hicimos cinco pruebas de velocidad en Pingdom, y tomamos el promedio. El tiempo total de carga fue de **243 ms** con un **tamaño total de página de tan solo 26.6 KB**.



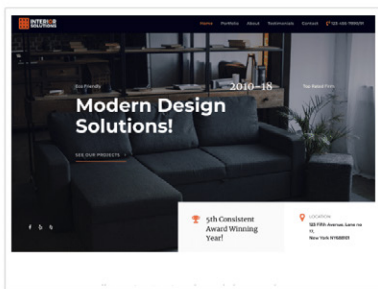
Performance grade
A 98

Page size
26.6 KB

Load time
243 ms

Requests
5

Luego, hicimos otro conjunto de pruebas con uno de los temas demo del Kit Básico de Astra en la biblioteca de su sitio. Este contiene imágenes, fondos, nuevas secciones, se requiere el plugin constructor de páginas Elementor. Puede ver que aún así estará por debajo de los 700 ms. Nota: las imágenes en esta demo están completamente comprimidas, pero ellos eligieron unas de alta resolución desde el principio.



Performance grade
B 86

Page size
1.6 MB

Load time
652 ms

Requests
54

Es importante ser un poco escéptico al tomar las diferencias entre pruebas de velocidad con estos tres temas. El problema es que es casi imposible hacer pruebas precisas para su comparación. Lo importante que queríamos mostrarle, es que todos estos temas de WordPress son extraordinariamente rápidos, ¡en sus versiones completas y en sus demos!

Advertencias Sobre los Constructores de Páginas

Como ya se habrá dado cuenta, OceanWP y Astra requieren constructores de páginas para utilizar los temas de la biblioteca de su sitio. Aquí hay algunas cosas que debe tener en mente al utilizar un plugin constructor de páginas:

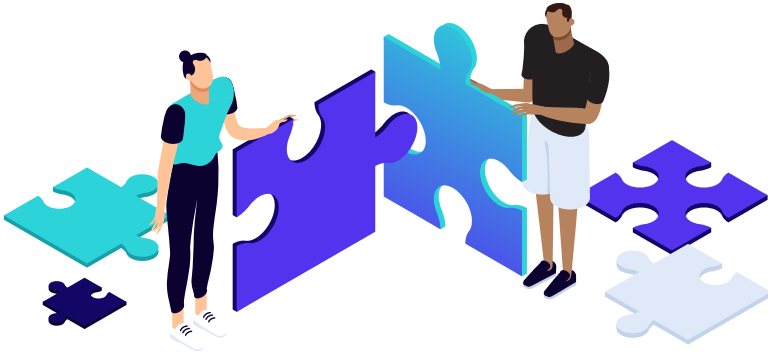
- Algunos constructores de páginas podrían incrementar el tiempo de carga en su sitio. Esto es porque tienen que cargar adicionalmente el CSS y JS para hacer que las cosas funcionen. ¡Así es como sucede la magia! Siempre recomendamos hacer pruebas de velocidad en su sitio de WordPress antes y después de instalar un constructor de páginas.
- Usted se está comprometiendo y encadenándose a ese constructor de páginas para el diseño. Asegúrese de elegir uno que sea actualizado regularmente y tenga todo lo que necesita a largo plazo.

Habiendo dicho esto, aún así, seguimos siendo fanáticos de los constructores de páginas como Elementor y Beaver Builder. Están desarrollados con el rendimiento en mente y tan sólo agregan un poco de carga extra. Para muchos, la funcionalidad y usabilidad valen la pena, ¡ya que estos plugins le permiten crear cualquier cosa con la que haya soñado! También podrían ser más rápidos en algunos casos, ya que podrían reemplazar hasta más de 5 plugins distintos que tendría que haber usado en su lugar.

Sin embargo, si no necesita un plugin constructor de páginas, no tiene porqué instalarlo. También será interesante ver como el nuevo editor Gutenberg jugará un rol en el diseño de sitios en los próximos años.

CAPÍTULO 06:

Todo Sobre los Plugins de WordPress



Ahora hablemos sobre los plugins de WordPress. Es posible que le hayan dicho que no debe de instalar demasiados plugins ya que podrían causar que su sitio de WordPress tenga un pero rendimiento y velocidad. Mientras que en algunas ocasiones esto es cierto, no es el factor más crítico. El número de plugins no es importante, sino la calidad de estos. Listo, ya lo dijimos.

Al igual que con los temas, importa si el plugin fue desarrollado y construido con el rendimiento en mente. Tenemos muchos clientes de Kinsta que utilizan 30 a 40 plugins en sus sitios y cargan en un segundo.

Mientras que es divertido agregar código a su sitio, no siempre es práctico por las siguientes razones:

- 1.** Tiene que mantener el código usted mismo y mantenerlo actualizado mientras los estándares cambian. La gente está ocupada, ¿Por qué no depender de desarrolladores fantásticos que conocen mejor que nadie estos estándares?
- 2.** La mayoría del tiempo, un plugin bien programado no introducirá mucha carga más que el mismo código.
- 3.** Tiene que recordar que la mayoría de la comunidad de WordPress no es tan experta en tecnología como lo sería un grupo de desarrolladores. Los plugins son soluciones que ayudan a resolver problemas.

Por supuesto también podemos encontrar plugins que no son tan buenos, y lo mejor sería alejarse de estos. Confíe en nosotros; hemos visto lo peor de lo peor en Kinsta. Muchos, no todos, de los plugins que hemos prohibido usar en Kinsta fue porque notamos de primera mano problemas de rendimiento al ser usados.

Sin embargo, no podemos ignorar que una de las cosas que la gente ama de WordPress es su enorme biblioteca de plugins. Pero con más de 56,000 plugins gratuitos listados en WordPress.org, y miles más en otros lados, puede ser difícil encontrar el plugin que necesita. ¡Es como encontrar una aguja en un pajar!

Sólo compartimos las cosas que utilizamos a diario. Y si, utilizamos plugins de WordPress en nuestro sitio como todos los demás. Incluso, muchos de los miembros del equipo en Kinsta han desarrollado y puesto a la venta sus propios plugins.

Un Gran Problema con los Plugins de WordPress

Un gran problema con los plugins de WordPress es el proceso de desinstalación. Sin importar que esté instalando un plugin o tema de WordPress, este almacena los datos en la base de datos. El problema es que cuando uno intenta borrar un plugin, usando uno de los métodos estándar, este, normalmente deja atrás tablas y filas en su base de datos. Con el tiempo, esto puede ir apilándose y hacer que su sitio empiece a estar más lento. En nuestro ejemplo, desinstalamos el plugin de seguridad Wordfence, y dejó atrás 24 tablas en nuestra base de datos (como puede ver a continuación).

Todavía es mucho peor si dejan atrás datos en su tabla de `wp_options`.

| | | | | | | | | | | | | | |
|--------------------------|--------------------|---|--------|-----------|--------|---|--------|-------|------|---------|--------|-------------------|----------|
| <input type="checkbox"/> | wp_wfBadLeechers | ★ | Browse | Structure | Search | 4 | Insert | Empty | Drop | 0 | InnoDB | latin1_swedish_ci | 16 K1B |
| <input type="checkbox"/> | wp_wfBlockedIPLog | ★ | Browse | Structure | Search | 4 | Insert | Empty | Drop | 35 | InnoDB | utf8_general_ci | 16 K1B |
| <input type="checkbox"/> | wp_wfBlocks | ★ | Browse | Structure | Search | 4 | Insert | Empty | Drop | 32 | InnoDB | utf8_general_ci | 32 K1B |
| <input type="checkbox"/> | wp_wfBlocksAdv | ★ | Browse | Structure | Search | 4 | Insert | Empty | Drop | 0 | InnoDB | utf8_general_ci | 16 K1B |
| <input type="checkbox"/> | wp_wfConfig | ★ | Browse | Structure | Search | 4 | Insert | Empty | Drop | 156 | InnoDB | utf8_general_ci | 16 K1B |
| <input type="checkbox"/> | wp_wfCrawlers | ★ | Browse | Structure | Search | 4 | Insert | Empty | Drop | 10 | InnoDB | latin1_swedish_ci | 16 K1B |
| <input type="checkbox"/> | wp_wfFileMods | ★ | Browse | Structure | Search | 4 | Insert | Empty | Drop | 0 | InnoDB | utf8_general_ci | 16 K1B |
| <input type="checkbox"/> | wp_wfHits | ★ | Browse | Structure | Search | 4 | Insert | Empty | Drop | 1,907 | InnoDB | latin1_swedish_ci | 1.3 M1B |
| <input type="checkbox"/> | wp_wfHoover | ★ | Browse | Structure | Search | 4 | Insert | Empty | Drop | 0 | InnoDB | utf8_general_ci | 32 K1B |
| <input type="checkbox"/> | wp_wfIssues | ★ | Browse | Structure | Search | 4 | Insert | Empty | Drop | 0 | InnoDB | utf8_general_ci | 16 K1B |
| <input type="checkbox"/> | wp_wfKnownFileList | ★ | Browse | Structure | Search | 4 | Insert | Empty | Drop | 0 | InnoDB | utf8_general_ci | 16 K1B |
| <input type="checkbox"/> | wp_wfLeechers | ★ | Browse | Structure | Search | 4 | Insert | Empty | Drop | 156,922 | InnoDB | latin1_swedish_ci | 13.5 M1B |
| <input type="checkbox"/> | wp_wfLockedOut | ★ | Browse | Structure | Search | 4 | Insert | Empty | Drop | 0 | InnoDB | utf8_general_ci | 16 K1B |
| <input type="checkbox"/> | wp_wfLocs | ★ | Browse | Structure | Search | 4 | Insert | Empty | Drop | 4 | InnoDB | utf8_general_ci | 16 K1B |

| | | | | | | | | | | | | | |
|--------------------------|--------------------|---|--------|-----------|--------|--------|-------|------|-----|--------|-------------------|----|-----|
| <input type="checkbox"/> | wp_wfLogins | ★ | Browse | Structure | Search | Insert | Empty | Drop | 5 | InnoDB | utf8_general_ci | 48 | KiB |
| <input type="checkbox"/> | wp_wfNet404s | ★ | Browse | Structure | Search | Insert | Empty | Drop | 1 | InnoDB | utf8_general_ci | 32 | KiB |
| <input type="checkbox"/> | wp_wfNotifications | ★ | Browse | Structure | Search | Insert | Empty | Drop | 2 | InnoDB | utf8_general_ci | 16 | KiB |
| <input type="checkbox"/> | wp_wfPendingIssues | ★ | Browse | Structure | Search | Insert | Empty | Drop | 0 | InnoDB | utf8_general_ci | 16 | KiB |
| <input type="checkbox"/> | wp_wfReverseCache | ★ | Browse | Structure | Search | Insert | Empty | Drop | 2 | InnoDB | latin1_swedish_ci | 16 | KiB |
| <input type="checkbox"/> | wp_wfScanners | ★ | Browse | Structure | Search | Insert | Empty | Drop | 618 | InnoDB | latin1_swedish_ci | 64 | KiB |
| <input type="checkbox"/> | wp_wfSNIPCache | ★ | Browse | Structure | Search | Insert | Empty | Drop | 0 | InnoDB | utf8_general_ci | 64 | KiB |
| <input type="checkbox"/> | wp_wfStatus | ★ | Browse | Structure | Search | Insert | Empty | Drop | 40 | InnoDB | utf8_general_ci | 48 | KiB |
| <input type="checkbox"/> | wp_wfThrottleLog | ★ | Browse | Structure | Search | Insert | Empty | Drop | 0 | InnoDB | utf8_general_ci | 32 | KiB |
| <input type="checkbox"/> | wp_wfVulnScanners | ★ | Browse | Structure | Search | Insert | Empty | Drop | 0 | InnoDB | latin1_swedish_ci | 16 | KiB |

Tablas de WordFence

Además de la base de datos, muchos de los plugins dejan atrás carpetas y archivos adicionales. En nuestra experiencia, esto es práctica habitual en los plugins de seguridad y de caché, los cuales han creado directorios adicionales para iniciar sesión. Por ejemplo, después de haber borrado el plugin de Wordfence, nos dejó atrás una carpeta llamada “wflogs” en nuestro directorio de contenido. Nuestra intención no es hablar mal de Wordfence, la gran mayoría de plugins y temas en el mercado funcionan de esta manera.

| Remote site: /www/ /public/wp-content/wflogs | | | |
|--|---------|---------------|-----------------------|
| | ? | upgrade | |
| | ? | uploads | |
| | | wflogs | |
| Filename | File... | Filetype | Last modified |
| .. | | | |
| .htaccess | 133 | HTACCESS File | 7/24/2017 10:54:49 AM |
| attack-data.php | 40,0... | PHP File | 7/24/2017 10:54:49 AM |
| config.php | 594,... | PHP File | 7/24/2017 10:57:30 AM |
| ips.php | 51 | PHP File | 7/24/2017 10:54:50 AM |
| rules.php | 101,... | PHP File | 7/24/2017 10:54:59 AM |
| wafRules.rules | 44,2... | RULES File | 7/24/2017 10:54:59 AM |

Logs de WordFence

¿Por Qué Hacen esto los Desarrolladores?

Se estará preguntando, ¿por qué los desarrolladores no agregan opciones de auto depuración cuando se desinstala y borra un plugin? Bueno, sí lo hacen. Pero, hay algunas razones que no son tan obvias al principio para no borrar todo.

- 1. Desean conservar las opciones para el usuario.** Si borra un plugin y decide usarlo de nuevo en otro momento, todas sus configuraciones y datos seguirán ahí. Mientras que esto es super conveniente, no es la forma más eficiente de hacerlo.
- 2. No les interesa su rendimiento.** Algunos desarrolladores podrían decir que dejar en su sitio web las tablas no causa un impacto en el rendimiento. Pero imagínese un sitio, después de diez años, habiendo usado cientos de plugins, que han generado miles de filas o tablas. Las peticiones de la base de datos tienen un impacto importante en el rendimiento de su sitio de WordPress, y los plugins pueden hacer muchas de estas peticiones si el desarrollador no tuvo cuidado. Generalmente, un plugin bien escrito sólo debería hacer una consulta a las tablas o filas a las cuales estaba vinculada, sin embargo, esto no siempre es el caso, lo hemos visto de primera mano en Kinsta, largas queries de base de datos haciendo que un sitio, vaya super lento, debido a datos auto cargados de forma innecesaria en la tabla wp_options, las cuales han sido dejadas atrás.
- 3. Han cometido un error.** El manual de plugins de WordPress dice que “algunos desarrolladores, no tan experimentados, en algunas ocasiones cometerán el error de utilizar un hook de desactivación para este propósito.”

¿Las buenas noticias? Hay formas de limpiar y deshacerse de un plugin de forma apropiada. En el siguiente tutorial de nuestra web puede ver como hacerlo: <https://kinsta.com/es/aprender/acelerar-wordpress/>

- Cómo Desinstalar un Plugin de WordPress (de forma apropiada)
- Cómo Limpiar las Tablas Dejadas Atrás

CAPÍTULO 07:

Configuración Óptima de WordPress



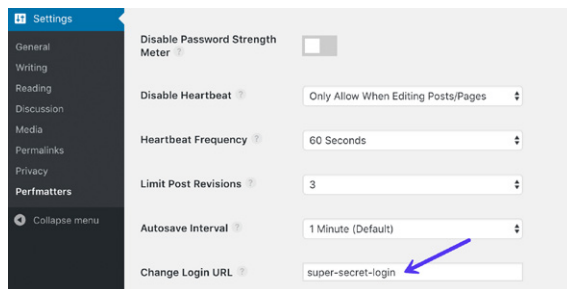
Ahora, es momento de pasar a la configuración óptima de WordPress. Hay un par de cambios que puede hacer para acelerar su sitio de WordPress. Muchos de estos cambios son sutiles, ¡pero puede ser de gran ayuda!

Cambie la URL de Inicio de Sesión de WordPress

Por defecto la URL de inicio de sesión de su sitio de WordPress es `dominio.com/wp-admin/`. Uno de los problemas de ello, es que todos los bots, hackers, y scripts también lo saben. Al cambiar la URL, ya no será un objetivo (o por lo menos ya no tanto), mejor protéjase de intentos de entrar por fuerza bruta, y reduzca el ancho de banda usado por bots que llegan a su URL de forma constante.

Cambiar la URL de inicio de sesión de WordPress puede ayudar a prevenir errores comunes, como el “429 Demasiadas Peticiones.” Esto no es una solución para todo, es un simple truco que puede ayudarle a proteger y reducir la carga en su sitio.

Para cambiar la URL de inicio de sesión de WordPress, recomendamos utilizar uno de los siguientes plugins:



Cambiar la URL de iniciar sesión de WordPress en Perfmatters

- **WPS Hide Login** (Gratuito, disponible en WordPress.org)
- **Perfmatters** (Premium, pero incluye otras opciones de optimización de rendimiento. Desarrollado por un miembro del equipo de Kinsta y disponible en perfmatters.io)

Deshabilite o Configure las Actualizaciones de los Plugins y Temas

Los escritorios de admin lentos de WordPress pueden ser influidos por la red, ubicación de los centros de datos, e incluso versiones de PHP. Pero otro factor del que no hablan muchos es la comprobación automática de actualizaciones de WordPress que funciona en segundo plano. Si dispone de muchos plugins y temas de WordPress podría afectarle. WeFoster tiene un excelente artículo sobre ello, donde la frase clave es “Síndrome de comprobación de Actualizaciones de Plugins de Terceras Partes” o por sus siglas en inglés TPPUCS (Third Party Plugin Update Check Syndrome).

Esencialmente el problema es que el comprobador de actualizaciones de WordPress, es el que hace una petición GET externa detrás de las escenas (<https://third-party-plugin/update-check.php>). En algunas ocasiones esto puede ser periódico o muy frecuente. Si está pasando todo el tiempo, esto podría dejar completamente inservible su escritorio de administración.

El problema es debido con la forma en el que el comprobador de actualizaciones de WordPress fue construido. Si está sufriendo de un tiempo de carga lenta en el escritorio de WordPress, podría querer intentar deshabilitar las actualizaciones automáticas. Advertencia: Haga esto sólo si pretende hacer las actualizaciones manualmente. Muchas actualizaciones incluyen mejoras de fallos o correcciones de seguridad.

Para desactivar las actualizaciones, le recomendamos utilizar uno de los siguientes plugins:

- **Disable All WordPress Updates:** Completamente gratuito sin opciones. Hace lo que hace muy bien.
- **Easy Updates Manager:** Brinda más control sobre la selección de actualizaciones. La versión básica es gratuita.

Fácilmente podría poner un recordatorio en su calendario, para deshabilitar el plugin una vez por semana, comprobar todas las actualizaciones y reactivarlo.

Desactivando Pingbacks

Un pingback es un comentario automatizado que es creado cuando otro blog tiene un enlace al suyo. También puede haber pingbacks propios que son creados cuando uno enlaza a un artículo propio en su blog.

Le recomendamos simplemente desactivarlos, ya que generan peticiones innecesarias y spam adicional en su sitio. Recuerde, entre menos llamadas a su sitio de WordPress haga, mejor, especialmente en sitios de alto tráfico. Sin mencionar el hecho de que un pingback en su sitio web es algo molesto. Siga estos pasos para desactivar los pingbacks.

Paso 1 – Desactive los Pingbacks de Otros Blogs

En su escritorio de WordPress, haga clic en “Ajustes → Comentarios.” Debajo de la sección de Discusión, desactive la opción de “Permitir notificación por enlaces de otro blog (pingbacks y trackbacks) en nuevos artículos.”

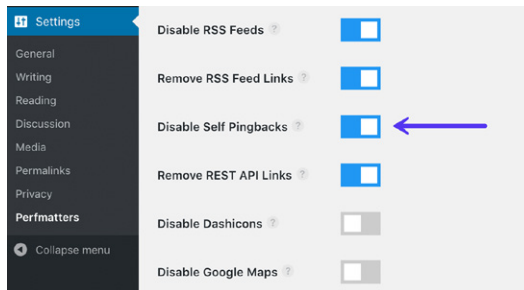
Discussion Settings

Default article settings

- Attempt to notify any blogs linked to from the article
-  Allow link notifications from other blogs (pingbacks and trackbacks) on new articles
- Allow people to post comments on new articles

(These settings may be overridden for individual articles.)

Paso 2 – Desactivando los Pingbacks Propios



Deshabilitar los Pingbacks automáticos con Perfmatters

Cuando se trata de desactivar los pingbacks propios, tiene un par de opciones. Puede utilizar el plugin gratuito No Self Pings (disponible en WordPress.org). O utilizar un plugin premium como perfmatters (disponible en perfmatters.io).

De forma alternativa, también podría desactivar los pingbacks propios agregando el siguiente código al archivo `functions.php` de su tema de WordPress. Pero cuidado, editar el código de un tema de WordPress podría deshabilitar su sitio si no se hace de la forma correcta. Un consejo, usted puede agregar fragmentos de PHP como este con el plugin gratuito Code Snippets (disponible en WordPress.org). De esta forma no tendrá que tocar su tema.

```
function wpsites_disable_self_pingbacks( &$links ) {  
    foreach ( $links as $l => $link )  
        if ( 0 === strpos( $link, get_option( 'home' ) ) )  
            unset($links[$l]);  
}  
  
add_action( 'pre_ping', 'wpsites_disable_self_pingbacks' );
```

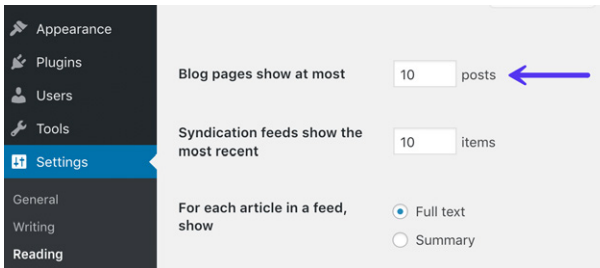
Limitar Publicaciones en el Contenido de su Blog

Si su blog está configurado como su página principal o utiliza otra página de su sitio, usted no necesita 50 thumbnails cargando al mismo tiempo. Para aquellos que tengan un blog con alto tráfico, su página de inicio es la página más importante de su sitio, y usted querrá que cargue rápido. Entre menos peticiones y archivos multimedia tenga, será mejor en términos de rendimiento.

También, esta es la razón por la cual se inventó la paginación (como se puede ver a continuación). La paginación es lo que usted ve al final del contenido de un blog que permite que pueda navegar a la siguiente página. Normalmente se usan números, o podrían utilizar “siguiente/anterior”. Su tema de WordPress probablemente tenga un sistema de paginación incluido.



WordPress por defecto establece un límite de últimas publicaciones de 10, pero hemos visto este valor cambiar tanto en nuestros clientes que ya perdimos la cuenta. Asegúrese de comprobar dos veces qué valor está usando. Nosotros le recomendamos un valor entre 8 y 12. Si tiene curiosidad, estamos utilizando 12 en la página de inicio de nuestro blog de Kinsta.



Podrá encontrar esta opción en el escritorio de administración de WordPress en “Ajustes → Lectura.” Luego, puede cambiar el valor por “Máximo número de páginas mostrado.”

CAPÍTULO 08:

¿Por Qué el Caché Es Tan Importante?



¡El caché es sin duda alguna una de las formas más importantes y sencillas para acelerar WordPress! Pero antes de enseñarle cómo usar el caché, es esencial entender cómo funciona y los distintos tipos de caché disponibles.

¿Qué es el Caché?

En pocas palabras, cada página web que uno visita en su sitio de WordPress requiere una petición al servidor, ser procesada por ese servidor (incluyendo los queries de la base de datos), y luego enviar un resultado final desde el servidor hasta el navegador del usuario. El resultado es su sitio web, completo, con todos los archivos y elementos que hacen que luzca como es.

Por ejemplo, podría tener un encabezado, imágenes, un menú, un blog. Ya que el servidor tiene que procesar todas esas peticiones, toma un poco de tiempo para que se entregue al usuario la página web completa – especialmente con sitios web desordenados o más grandes.

¡Aquí es cuando entran en juego los plugins de caché de WordPress! El caché insta al servidor a almacenar algunos archivos en disco o en la RAM, dependiendo de la

configuración. Así, puede recordar y duplicar el mismo contenido que ha estado sirviendo en el pasado. Básicamente, reduce la cantidad de trabajo requerido para generar una vista de la página. Como resultado, **sus páginas web cargarán mucho más rápidas, directamente desde el caché.**

Algunos beneficios del caché incluyen:

- **Su servidor utiliza menos recursos** – Esto está vinculado a la velocidad, ya que menos recursos hacen que un sitio sea más rápido. También pone menos estrés en su servidor. Esto es muy importante cuando se trata de sitios altamente dinámicos, como los sitios de membresía, también es importante determinar lo que uno puede servir y no servir del caché.
- **Usted notará un TTFB menor** – El caché es una de las formas más sencillas para reducir su TTFB. De hecho, ¡en nuestras pruebas el caché normalmente reduce el TTFB hasta en un 90%!

Tipos de Caché

Cuando se trata de tipos de caché, hay dos enfoques distintos que son normalmente usados:

1. Caché a Nivel Servidor
2. Caché con un Plugin

1. Caché a Nivel Servidor

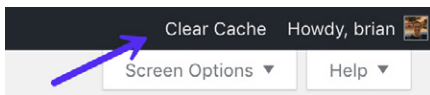
El caché a nivel servidor es uno de los enfoques más sencillos para el usuario final. Esto quiere decir que el proveedor de hosting de WordPress se encarga de ello por usted. En Kinsta, nosotros utilizamos los siguientes cuatro tipos de caché, los cuales son automáticamente usados en el software o a nivel servidor:

- Caché Bytecode
- Caché de Objeto
- Caché de Página
- Caché CDN

Esto quiere decir que no tendrá que preocuparse de tener que lidiar con plugins de caché complicados y confusos. Puede dejar de usar Google, buscando “el mejor plugin de caché en 2019” y enfocarse más en tareas productivas.

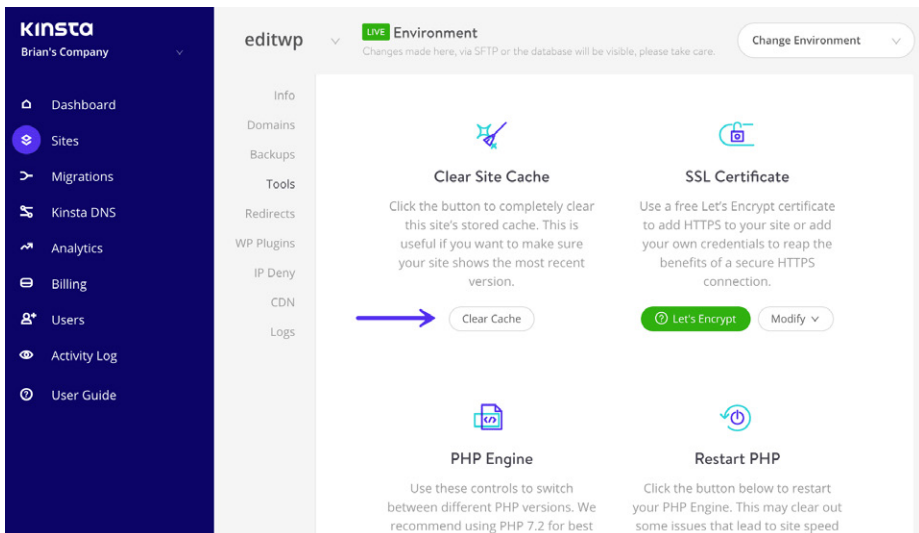
El caché de la página está configurado para funcionar desde el principio con el WordPress por defecto. ¡No tendrá que hacer nada! Simplemente abra su sitio de WordPress y el caché de la página empezará a funcionar.

También tenemos reglas de caché para sitios de ecommerce, como WooCommerce y Easy Digital Downloads. Por defecto, ciertas páginas no deberían ser puestas en el caché, como el carrito de compras, mi cuenta y la página de pago, están excluidas del caché. Los usuarios automáticamente se saltarán el caché cuando la cookie `woocommerce_items_in_cart` o la de `edd_items_in_cart` sean detectadas para asegurar un proceso de sincronización de pago correcto.



De forma sencilla puede borrar el caché de su sitio de WordPress en cualquier momento desde la barra de herramientas de admin.

También está integrada en nuestro escritorio de MyKinsta. Sólo un clic en Herramientas y en “Limpiar Caché.”



2. Caché con un Plugin

Si su proveedor de hosting no ofrece un caché, usted puede utilizar un plugin externo de caché para WordPress. Basándonos en nuestra experiencia.

Recomendamos uno de los siguientes:

- **WP Rocket** (premium, disponible en wp-rocket.me)
- **Cache Enabler** (gratis, disponible en WordPress.org)
- **W3 Total Cache** (gratis, disponible en WordPress.org)

¡También ofrecemos soporte completo para WP Rocket en Kinsta! Normalmente no permitimos plugins de caché en nuestro entorno porque pueden entrar en conflicto con nuestra solución de caché interna. Sin embargo, a partir de la versión 3.0 de WP Rocket, su función de caché de página será automáticamente desactivada al utilizar servidores de Kinsta.

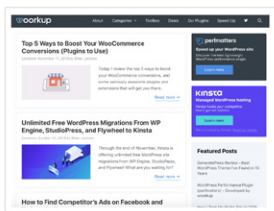
Esto permite a los clientes de Kinsta utilizar nuestro caché a nivel servidor súper rápido, pero obtener la ventaja de las fantásticas opciones de optimización que WP Rocket ofrece.

Sin Caché vs con Caché

¿Cuánto sirve el caché? La prueba está en el pudín

Hicimos algunas pruebas de velocidad con el caché a nivel servidor de Kinsta, para que pueda ver las diferencia, en términos de velocidad general y TTFB.

Sin Caché

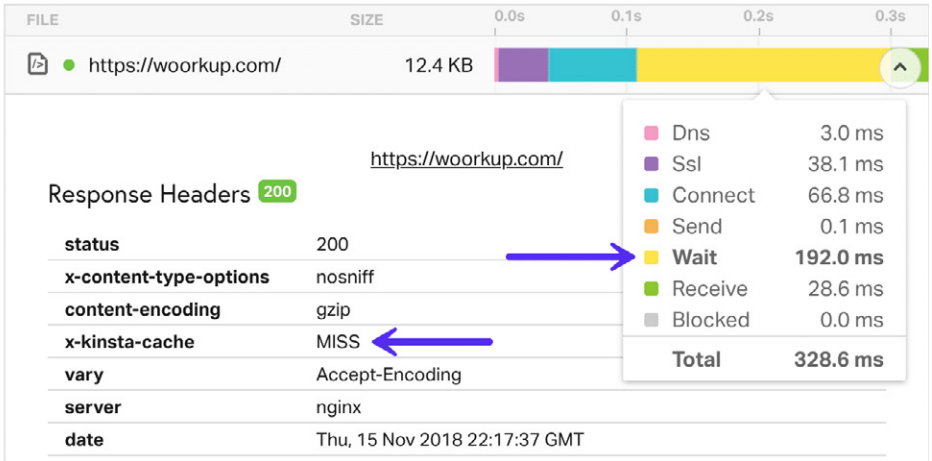


| | |
|----------------------------------|------------------------------|
| Performance grade B 89 | Page size 230.6 KB |
| Load time 507 ms | Requests 40 |

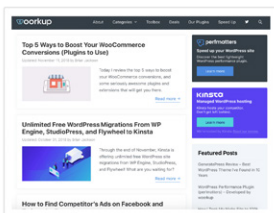
Primero hicimos cinco pruebas en Pingdom sin el caché habilitado y tomamos el promedio.

TTFB sin Caché

También es importante tener en cuenta la diferencia en el TTFB sin y con caché. El TTFB en Pingdom está representado por una barra de “espera” amarilla. Como puede ver el TTFB sin caché es de 192 ms. Podrá ver que no está sirviendo desde el caché ya que el encabezado `x-kinsta-cache` está mostrando un `MISS` (Faltante).



Con Caché Habilitado



Performance grade
B 89

Load time
351 ms

Page size
230.6 KB

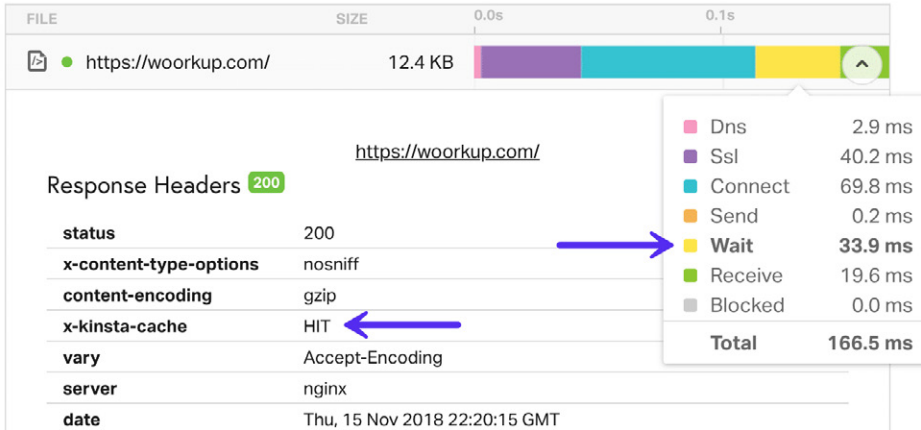
Requests
40

Luego habilitamos el caché a nivel del servidor e hicimos cinco pruebas en Pingdom y tomamos el promedio.

¡Como puede ver el caché a nivel servidor disminuyó el tiempo de carga de nuestra página por un 33.77%! y esto es sin el trabajo extra requerido. Este sitio que pusimos a prueba estaba bien optimizado, así que algunos sitios grandes sin optimizar verán diferencias mayores.

TTFB con el Caché Habilitado

Si vemos el TTFB con el caché habilitado, podemos ver que se encuentra por debajo de 35 ms. Podrá ver que no está sirviendo desde el caché ya que el encabezado `x-kinsta-cache` está mostrando un `HIT`.



El caché de CDN es igual de importante que el caché de su host de WordPress. Hablaremos más detalladamente sobre las CDNs más adelante.

¡El uso de caché en WordPress puede reducir fácilmente el tiempo de carga de su página hasta por un 33%!

Problemas con el Caché y los Sitios de Membresía

Los sitios de membresía contienen mucho contenido que no puede ser alojado en el caché, tiene páginas que se encuentran continuamente cambiando. Las cosas como la página de inicio de sesión para los miembros de la comunidad (la cual estaría recibiendo hits constantes dependiendo del tamaño del sitio), las páginas de pago para artículos o cursos digitales y los foros de discusión son culpables comunes y puntos de dolor, ya que normalmente estas no pueden estar en el caché.

Sin embargo, esto no termina aquí. En sitios de WordPress estándar, el escritorio de WordPress tampoco está almacenado en el caché para los usuarios “que ya iniciaron sesión”. Esto está bien cuando tiene algunos autores y administradores, pero cuando tiene miles de miembros utilizando el escritorio, puede causar problemas inmediatamente de rendimiento ya que no puede servirse del caché en el servidor. Esto quiere decir que usted necesitará potencia y arquitectura detrás de escenas para respaldarlo. Los proveedores de hosting compartido normalmente colapsan bajo estas circunstancias.

Caché de Objeto para Sitios Altamente Dinámicos

Cuando se trata de sitios de membresías de WordPress, las configuraciones comunes de caché no son suficientes ya que no toman ventaja total de ellas. Aquí es donde **entra en juego el caché de objeto**.

El caché de objeto almacena los resultados de los queries de la base de datos para que la próxima vez que un pedazo particular de dato es requerido, este pueda ser entregado desde el caché, sin la necesidad de hacer consulta a la base de datos (una petición). Esto acelera los tiempos de ejecución de PHP y reduce la carga en su base de datos. ¡Esto es extremadamente importante en sitios de membresía! con WordPress, puede implementar caché de objeto de formas distintas:

1. Una solución de caché como W3 Total Cache
2. **Redis (recomendada)**
3. Memcached

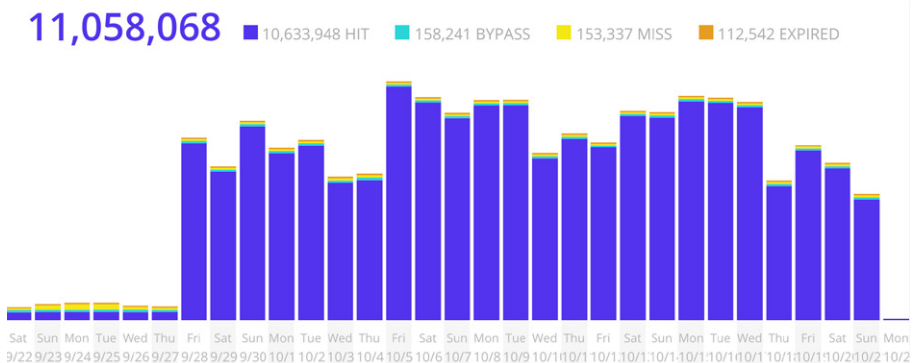
Nosotros ofrecemos Redis como un complemento en Kinsta, para que pueda tomar total ventaja de caché de objeto persistente para sus sitios de membresía.

Caché de Análisis

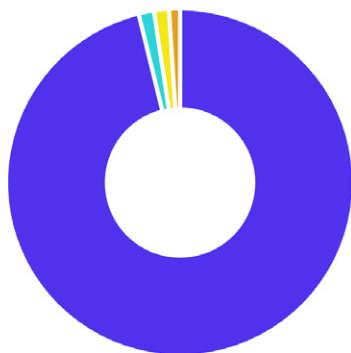
¿Recuerda el encabezado `x-kinsta-cache` que hemos mencionado anteriormente? Dependiendo del proveedor de host o solución de caché, el encabezado podría ser llamado de forma un poco distinta. Cada vez que se hace una petición, desde su sitio de WordPress, ese encabezado tiene un valor, como HIT, BYPASS, MISS y EXPIRED. Esto le permite ver cómo se está desempeñando su caché.

Incrementar el rango de hit del caché de su sitio de WordPress es importante porque usted necesita que la gran mayoría de su sitio esté servida desde el caché si es posible. En Kinsta usted puede analizar los datos en nuestro MyKinsta analytics tool y los logs del cache en Kinsta para determinar si hay caché que esté BYPASSing (Traspasando) peticiones GET que podrían estar alojadas en el caché o peticiones POST que podrían ser eliminadas.

El stack de componentes del caché (como se muestra abajo) le permite ver el estado de cada petición, sea un HIT, BYPASS, MISS o EXPIRED. Usted puede filtrar los datos de las últimas 24 horas, 7 días o 30 días.



La gráfica de componentes de caché le da un pequeño vistazo al rango de su caché. Entre más peticiones usted sirva desde el caché, será mucho mejor. Como puede ver en el ejemplo de abajo, este sitio de WordPress se encuentra en un rango de caché de 96.2% HIT. ¡Eso es bueno!



Total
11,058,068

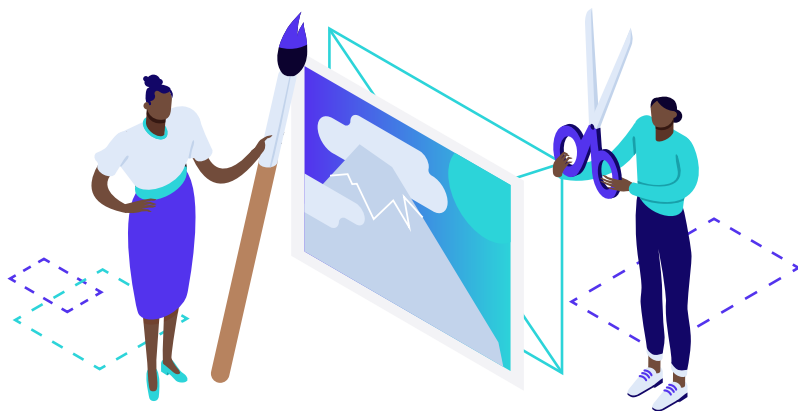
- HIT 10,633,948 (96.2%)
- BYPASS 158,241 (1.4%)
- MISS 153,337 (1.4%)
- EXPIRED 112,542 (1.0%)

La sección de los traspasos más altos del caché le permite ver qué peticiones no están siendo servidas desde el caché. Normalmente estas podrían incluir CRON jobs, peticiones admin-ajax, páginas de pago de ecommerce, query strings, y parámetros UTM, etc.

| PATH | REQUESTS |
|--|----------|
| /wp-cron.php?server_triggered_cronjob | 28,977 |
| /wp-admin/admin-ajax.php?action=edd_jilt_get_cart_data | 22,279 |
| ?ref=1 | 796 |
| / | 667 |
| /sitemap_index.xml | 490 |
| ?ajax=1&ht-kb-search=1&s= | 435 |
| /account/ | 372 |
| /checkout/ | 329 |
| ?ref=19&campaign=bw | 292 |

CAPÍTULO 09:

La Optimización de Imágenes es Necesaria

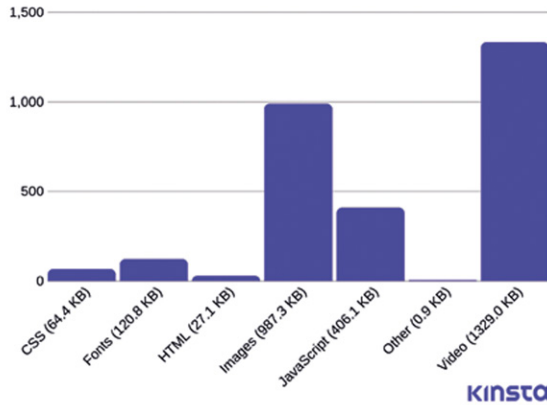


La optimización de imágenes es otra cosa directa que usted puede hacer, que tiene un alto impacto en el tiempo de carga total de su página. Esto no es opcional; ¡todo sitio debería estar haciendo esto!

Las grandes imágenes disminuyen la velocidad de sus sitios webs, lo cual no crea una experiencia de usuario óptima. La optimización de imágenes es el proceso de reducir el tamaño de su archivo, utilizando un plugin o un script, lo cual terminará acelerando el tiempo de carga de la página. Compresión lossy y lossless son dos métodos normalmente utilizados.

De acuerdo a HTTP Archive, desde Agosto de 2019, **las imágenes ocupan de media un 34%** del peso total de la página web. Así que, después de los videos, los cuales son mucho más difíciles de optimizar, ¡las imágenes son por mucho el primer lugar donde debería comenzar! Es más importante que JavaScript, CSS y Fonts. E irónicamente, un buen flujo de trabajo de optimización de imágenes es una de las cosas más sencillas de implementar, aún así, la mayoría de los dueños de sitios webs lo pasan por alto.

Media de Bytes Por Página



Las imágenes ocupaban de media un 54% del peso total de las páginas en el período de diciembre de 2017. Así que, aparentemente, ¡se está haciendo un mejor trabajo optimizando las imágenes! Pero un 34% sigue siendo un número que no puede ser ignorado. Si no tiene contenido en vídeo en su sitio web, las imágenes aún seguirán siendo el punto crítico #1 para el peso de la página.

Las imágenes ocupan en promedio un 34% del peso total de la página web. ¡Tiene que optimizarlas!

Encontrando el Equilibrio (Tamaño y Calidad del Archivo)

La meta principal del formato de sus imágenes es encontrar el **equilibrio correcto entre el tamaño del archivo más pequeño y una calidad aceptable**. Hay más de una forma de llevar a cabo la mayoría de estas optimizaciones. Una de las formas más básicas es comprimirlas antes de empezar a subirlas a WordPress. Normalmente, esto se puede hacer mediante una herramienta como Adobe Photoshop o Affinity Photo. O utilizando la nueva app online de Squoosh de Google. Sin embargo, estas tareas también pueden ser desempeñadas de forma automática, utilizando plugins, de los que hablaremos más adelante.

Las dos cosas principales a considerar son el **formato del archivo** y el **tipo de compresión** que utiliza. Al elegir la combinación correcta del formato del archivo y tipo de compresión, usted puede reducir el tamaño de la imagen hasta 5 veces más. Usted tendrá que experimentar con cada imagen o formato de archivo para ver cuál le funciona mejor.

Antes de empezar a modificar sus imágenes, asegúrese de que haya elegido el mejor tipo de archivo. Hay varios tipos de archivos que puede utilizar:

- **PNG** – produce imágenes de más alta calidad, pero también tiene un mayor tamaño de archivo. Fue creada como un formato de imagen sin pérdida, a pesar de que también puede ser con pérdida.
- **JPEG** – utiliza optimización con o sin pérdida. Puede ajustar la calidad para un buen balance de calidad y tamaño de archivo.

Idealmente, usted debería utilizar JPEG (o JPG) para imágenes con muchos colores y PNG para imágenes más simples.

¿Y los GIFs? Los GIFs animados siempre serán divertidos, pero estos matarán el rendimiento de su sitio. Muchos GIFs están por encima de 1 MB de tamaño. Recomendamos mantener estos para las redes sociales y Slack. Si hay alguno sin el cual no pueda vivir en su blog, entonces revise cómo puede comprimir GIFs animados.

Calidad de Compresión vs Tamaño

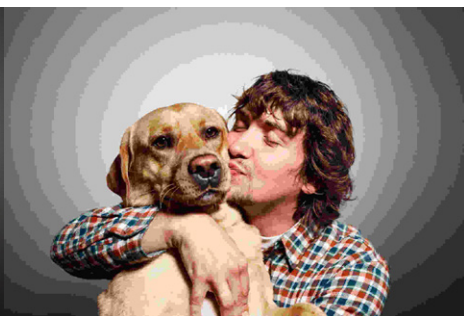
Aquí tenemos un ejemplo de lo que puede pasar cuando uno comprime demasiado una imagen. El primero está utilizando un rango de compresión bajo, lo que resulta en calidad más alta (pero un tamaño de archivo más grande). El segundo está utilizando un rango de compresión más alto, lo que da como resultado una imagen de menor calidad (pero de menor tamaño). Nota: la imagen original sin modificar pesa 2.06 MB.

Como puede observar la primera imagen rebasa los 590 KB. ¡Esto es muy grande para una sola foto! Lo mejor es mantener el peso total de su página por debajo de 1 o 2 MB. 590 KB sería un cuarto de ese total. La segunda imagen se ve fatal, pero pesa tan solo

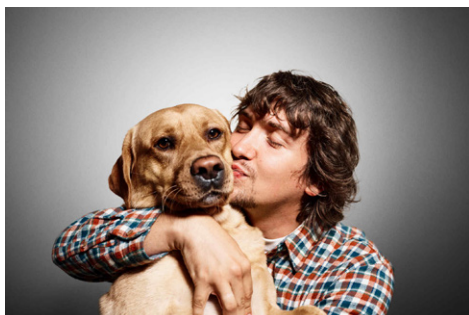
68 KB. Lo que necesita lograr, es encontrar un bonito equilibrio, entre el rango de compresión (calidad) y el tamaño del archivo.



Baja compresión (alta calidad) JPG – 590 KB



Alta compresión (baja calidad) JPG – 68 KB



Compresión media (excelente calidad) JPG – 151 KB

Ahora comprimimos la imagen de nuevo a un rango de compresión media, como podrá observar abajo, la calidad luce bien ahora, y el tamaño del archivo es de 151 KB, el cual es aceptable para una foto de alta resolución. Tratamos de mantener la mayoría de nuestras imágenes debajo de 100 KB para tener mejor rendimiento.

Optimización Lossy vs Lossless (con o sin Pérdida)

También es importante entender que hay dos tipos de compresión que puede utilizar, lossy y lossless.

La compresión lossy **elimina información de los datos** en su imagen. Como resultado, usted podría ver una degradación (reducción en la calidad o a lo que muchos llaman una imagen pixelada). Así que tiene que tener cuidado cuando reduce su imagen. No sólo por la calidad, sino porque no podrá revertir el proceso. Por supuesto, uno de los grandes beneficios de la compresión lossy y la razón por la que es uno de los métodos de compresión más populares es que se **puede reducir el tamaño del archivo de forma considerable**.



La compresión sin pérdida, a diferencia del lossy, **no reduce la calidad** de la imagen. ¿Cómo es esto posible? Normalmente se consigue al eliminar metadatos innecesarios (datos automáticamente generados producidos por el dispositivo que captura la imagen). Sin embargo, la desventaja más grande de este método es que **no podrá ver una reducción de tamaño de archivo significativa**. En otras palabras, a largo plazo usará mucho espacio de disco.

Querrá experimentar con lo que funcione mejor en su caso. Pero para la mayoría de los usuarios, **recomendamos utilizar compresión lossy** debido a que podrá comprimir fácilmente una imagen hasta un 70% (incluso hasta 90%) sin perder mucha calidad. Multiplique esto por 15 imágenes en una página, y jugará un rol importante en la reducción del tiempo de carga de su sitio.

Plugins de Compresión de Imágenes

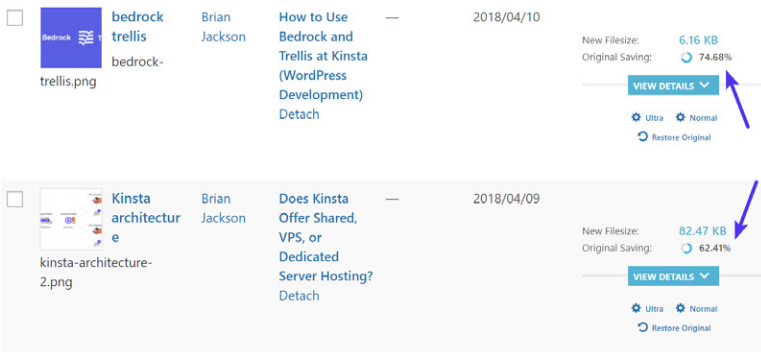
Las buenas noticias es que hay algunos plugins de compresión de imágenes sorprendentes para WordPress que puede utilizar para automatizar el proceso entero. Aquí tenemos algunos plugins que le recomendamos: (disponibles en WordPress.org):

- **Imagify** (lossy y lossless – optimiza imágenes externamente)
- **WP Smush** (lossy y lossless – optimiza imágenes externamente)
- **Optimole** (lossy y lossless- optimiza imágenes externamente)
- **EWWW Cloud** (lossy y lossless- optimiza imágenes externamente)
- **ShortPixel** (lossy y lossless- optimiza imágenes externamente)

Lo más importante al elegir un plugin de optimización de imagen es **utilizar uno que comprima y optimice las imágenes externamente** de sus servidores. Esto, causa una reducción en la carga de su sitio. Todos los plugin mencionadas anteriormente lo hacen.

Si tiene curiosidad, nosotros utilizamos el plugin de Imagify en el sitio de Kinsta. Este automáticamente comprime imágenes cuando las subimos a la biblioteca multimedia en WordPress. Así que jamás tenemos que preocuparnos. Con el tiempo usted sabrá qué nivel de compresión de imagen desea. Ofrece modo Normal, Agresivo y Ultra.

Nosotros usamos el modo Agresivo en Kinsta y normalmente vemos un **ahorro del 60-70%** dependiendo de la imagen. Nota: utilizamos muchos más PNGs que JPEGs debido a que la mayoría de nuestras imágenes son iconos e ilustraciones, no utilizamos fotos.



¿Qué tan rápido podría quedar su sitio de WordPress al usar un compresor de imágenes? Depende del tamaño de sus imágenes originales y cómo quedan después de la compresión. Sin embargo, ¡hicimos unas pruebas de velocidad y descubrimos que una solución de calidad de compresión de imagen puede **reducir el tiempo de carga de una página hasta un 80%!**

Lazy Loading

Si tiene muchas imágenes, podría considerar hacerles carga diferida (lazy loading). Esta es una técnica de optimización que carga contenido visible, pero retrasa la carga y la visualización del contenido que aparece debajo del primer pliegue de la web.

Lazy loading funciona de la siguiente forma:

- El navegador construye la página web DOM sin descargar imágenes ni precargar vídeos.
- JavaScript se utiliza para determinar qué imágenes descargar y qué videos cargar previamente en función del contenido que se ve inicialmente cuando se carga la página. Esas imágenes y videos se descargan y precargan según corresponda.
- La descarga y la reproducción de videos adicionales se retrasan hasta que un visitante del sitio se desplace por la página y aparece contenido adicional.

El resultado final es que las imágenes no se descargan y los vídeos no se cargan hasta que se necesitan realmente. Esto puede proporcionar un aumento significativo en el rendimiento de los sitios que incluyen muchas imágenes de alta resolución y vídeos incrustados.

Afortunadamente hay muchos plugins de WordPress gratuitos que hacen que la lazy loading sea muy fácil (disponibles en WordPress.org):

- **a3 Lazy Load**
- **Lazy Load by WP Rocket**
- **Lazy Load for Videos**

Consejos Adicionales de Optimización de Imágenes

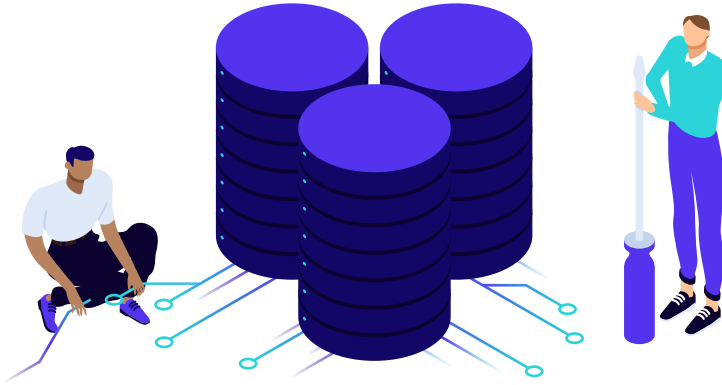
Aquí hay algunos consejos finales de optimizaciones de imágenes para tener en mente.

- Los días de subir imágenes escaladas al ancho de la columna o DIV han terminado. Las imágenes responsivas funcionan desde el principio en WordPress (desde la versión 4.4) y automáticamente mostrarán tamaños de imagen mucho más pequeñas para usuarios móviles.

- SVGs pueden ser otra forma alternativa sorprendente de usar imágenes. Todas las ilustraciones hechas a mano que ve en el sitio de Kinsta son SVGs (vectores). Los SVGs son normalmente mucho más pequeños en tamaño de archivo, aunque no siempre.
- Use fuentes de iconos en lugar de poner texto dentro de las imágenes – se ven mejor y ocupan menos espacio. Y si está usando un generador de fuentes puede optimizarlas aún más. Aprenda cómo disminuimos el tamaño de la fuente del icono de nuestro archivo por un 97,59% usando un generador de fuentes.
- También es posible que desee buscar en WebP, un nuevo formato de imagen de Google que está ganando terreno rápidamente.

CAPÍTULO 10:

Optimizando Su Base de Datos



Ahora le dejaremos algunos consejos sobre cómo optimizar la base de datos de su WordPress. Al igual que un coche, su base de datos necesita mantenimiento, ya que con el tiempo puede hincharse.

Los sitios de membresía son especialmente complejos, ya que normalmente generan más consulta complejas, las cuales agregan latencia adicional al sacar la información de la base de datos de MySQL. Mucho de esto es debido a las piezas móviles adicionales y grandes cantidades de datos que tienen. Esto podría ser causado por sitios que dependen mucho de los queries de búsqueda para la navegación o por utilizar `WP_Query`.

Sin mencionar que también tendrá grandes cantidades de usuarios concurrentes continuamente haciendo consultas a su base de datos.

Utilice el Motor de Almacenamiento de InnoDB MySQL

Muchos de los sitios antiguos aún utilizan el motor de almacenamiento de MyISAM en su base de datos. InnoDB, ha demostrado un mejor rendimiento y una mayor confiabilidad.

InnoDB es como aceite sintético, mientras que MyISAM es como elegir el regular.

Aquí le dejamos algunas ventajas de InnoDB vs MyISAM:

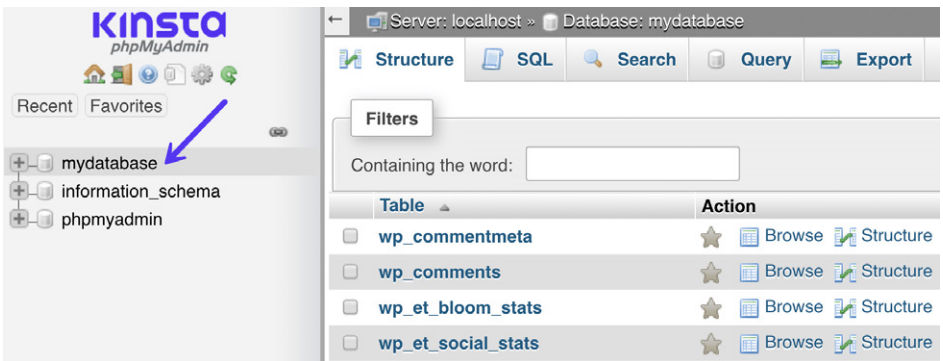
- InnoDB tiene **un seguro a nivel fila**. MyISAM sólo tiene un seguro a nivel tabla. Esto le permite a sus queries procesar más rápido.
- InnoDB tiene lo que llamamos integridad referencial, la cual involucra soportar **llaves foráneas** (RDBMS) y limitaciones de relación, MyISAM no hace esto (DMBS).
- InnoDB soporta transacciones, lo cual quiere decir que puede comprometerse y regresar a como estaba antes. MyISAM no puede hacer esto.
- InnoDB es más confiable, ya que utiliza registros transaccionales para auto recuperación. MyISAM no.

Así que podría estar preguntando, ¿está utilizando InnoDB o MyISAM? Si está usando un sitio relativamente nuevo de WordPress, es posible que ya esté utilizando el motor de almacenado InnoDB MySQL. Pero con sitios más antiguos de WordPress podría querer revisarlo. Algunos sitios podrían tener una mezcla de tablas MyISAM y de InnoDB, y podría ver mejoras al convertirlas todas a InnoDB.

Siga los siguientes pasos para revisarlo.

Paso 1

Inicie sesión a phpMyAdmin y haga clic en la base de datos de MySQL.



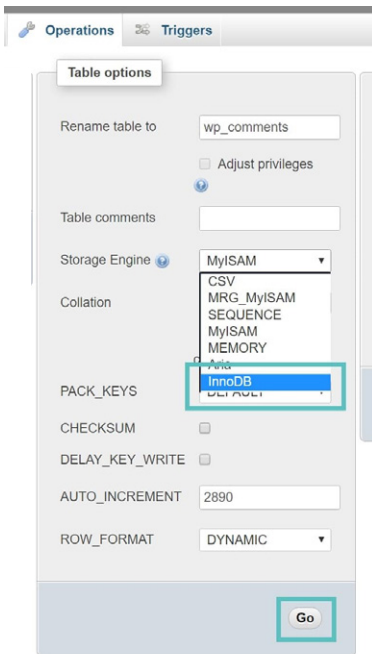
The screenshot shows the phpMyAdmin interface. On the left sidebar, the database 'mydatabase' is selected, indicated by a blue arrow. The main panel displays the table structure for 'mydatabase' with the following tables and actions:

| Table | Action |
|--------------------|--------------------|
| wp_commentmeta | ★ Browse Structure |
| wp_comments | ★ Browse Structure |
| wp_et_bloom_stats | ★ Browse Structure |
| wp_et_social_stats | ★ Browse Structure |

Paso 2

Haga un escaneo rápido y ordene por la columna “Tipo”, y podrá ver qué tipos de Motor de Almacenamiento están utilizando sus tablas. En el ejemplo de abajo, puede ver que dos de los tipos de las tablas siguen utilizando MyISAM.

| Icon | Structure | Search | Insert | Empty | Drop | Count | Engine | Table Name | Size |
|------|-----------|--------|--------|-------|------|-------|--------|--------------------|---------|
| 📄 | 🔍 | 🔍 | ➕ | 🗑️ | 🚫 | 0 | InnoDB | utf8_unicode_ci | 64 KiB |
| 📄 | 🔍 | 🔍 | ➕ | 🗑️ | 🚫 | 0 | InnoDB | utf8_unicode_ci | 48 KiB |
| 📄 | 🔍 | 🔍 | ➕ | 🗑️ | 🚫 | 0 | InnoDB | utf8_unicode_ci | 64 KiB |
| 📄 | 🔍 | 🔍 | ➕ | 🗑️ | 🚫 | 0 | InnoDB | utf8_unicode_ci | 80 KiB |
| 📄 | 🔍 | 🔍 | ➕ | 🗑️ | 🚫 | 0 | InnoDB | utf8_unicode_ci | 48 KiB |
| 📄 | 🔍 | 🔍 | ➕ | 🗑️ | 🚫 | 0 | InnoDB | utf8mb4_unicode_ci | 32 KiB |
| 📄 | 🔍 | 🔍 | ➕ | 🗑️ | 🚫 | 113 | MyISAM | utf8mb4_unicode_ci | 80 KiB |
| 📄 | 🔍 | 🔍 | ➕ | 🗑️ | 🚫 | 471 | MyISAM | utf8mb4_unicode_ci | 208 KiB |
| 📄 | 🔍 | 🔍 | ➕ | 🗑️ | 🚫 | 0 | InnoDB | utf8mb4_unicode_ci | 48 KiB |
| 📄 | 🔍 | 🔍 | ➕ | 🗑️ | 🚫 | 1 | InnoDB | utf8mb4_unicode_ci | 16 KiB |



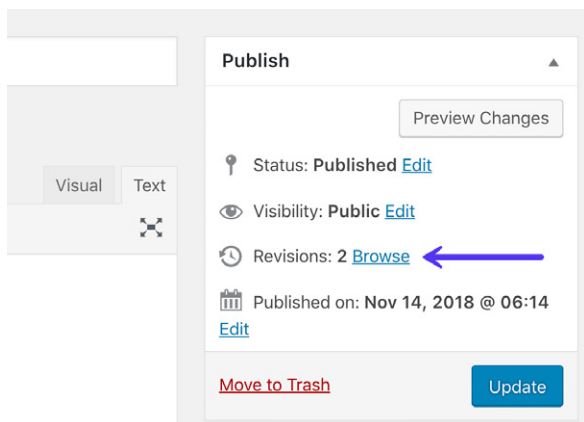
Si encuentra algunas tablas MyISAM, probablemente es momento de pasarlas a InnoDB. Siempre recomendamos contactar a su host y preguntar si pueden hacer esto por vosotros mismos. En Kinsta, las tablas de la base de datos de cada cliente automáticamente se convierten en InnoDB por nuestro equipo de migración

Pero siempre podrá convertirlas manualmente con phpMyAdmin. Simplemente haga clic en la tabla myISAM, clic en la pestaña “Operations” y cambie el motor de almacenamiento.

Si eres un desarrollador, otra alternativa sería convertirlos con WP-CLI.

Borrando, Límite de Página y Revisiones de Publicaciones

Cuando uno guarda una página o publicación en WordPress, esta crea lo que llamamos una revisión. Esto ocurre en los borradores y en publicaciones ya publicadas que son actualizadas. Las revisiones pueden ser útiles en caso de que necesite revertir a una versión previa de su contenido.

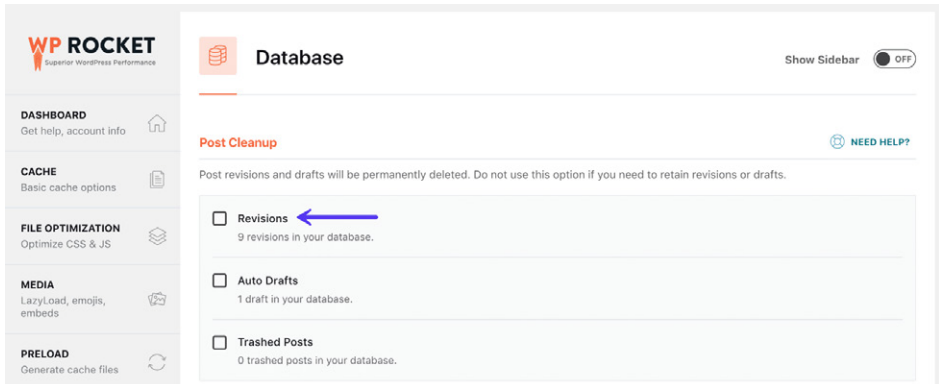


Sin embargo, las revisiones también pueden dañar el rendimiento de su sitio de WordPress. En sitios más grandes, esto puede irse acumulando muy rápido a miles de filas en su base de datos que no son necesarias. Y entre más filas tenga, más grande será el tamaño de la base de datos, la cual ocupa espacio en disco. Mientras que los índices fueron creados por este propósito, aún vemos cómo este problema afecta a los sitios de WordPress. Hay un par de cosas que puede hacer.

1. Borrando Viejas Revisiones

Si usted tiene un sitio antiguo de WordPress con muchas páginas y publicaciones, podría ser momento de hacer una rápida depuración y borrar esas viejas revisiones. Puede hacer esto con MySQL, pero con todos los malos fragmentos de código flotando alrededor de la red, le recomendamos hacer un backup de su sitio y utilizar un plugin gratuito como WP-Sweep.

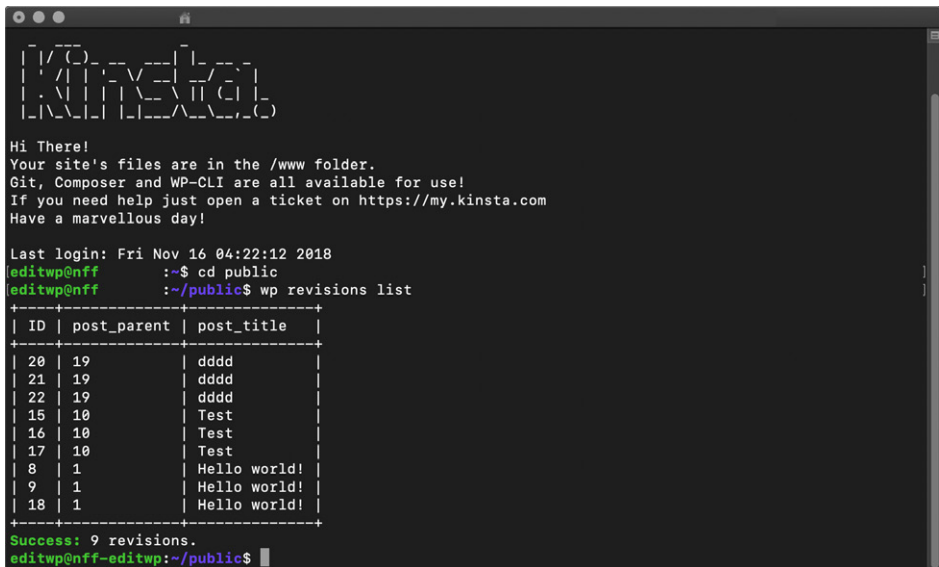
Otro de nuestros plugins favoritos, WP Rocket, también tiene una opción de optimización para bases de datos para poder depurar las revisiones.



Si usted sabe utilizar WP-CLI, hay un par de comandos que puede utilizar para ello.

Ingrese a su servidor a través de SSH y use el siguiente comando para obtener y ver el número de revisiones actuales en la base de datos

```
wp revisions list
```



Si usted obtiene un error, podría necesitar instalar primero el paquete `wp-revisions-cli` con el siguiente comando:

```
wp package install trepmal/wp-revisions-cli
```

Luego puede utilizar el siguiente comando para limpiar las revisiones:

```
wp revisions clean
```

2. Limitar Revisiones

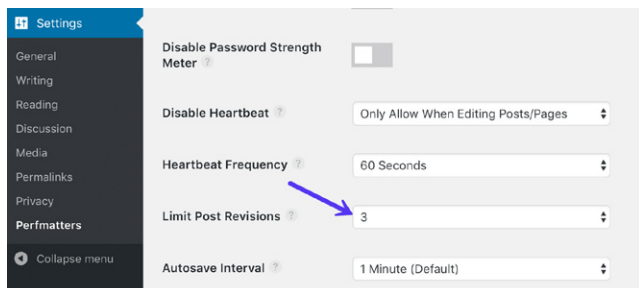
Otra buena estrategia y que utilizamos en Kinsta, es limitar el número de revisiones que pueden ser almacenadas por publicación o página. Incluso establecer un límite de diez, evitará que las revisiones se salgan de control, especialmente si hace muchas actualizaciones.

Para limitar revisiones, usted puede agregar el siguiente código a su archivo `wp-config.php`. El código siguiente necesita ser insertado por encima de `'ABSPATH'`, de otra forma, no funcionará. Puede cambiar el número al número de revisiones que usted quiera mantener almacenados en la base de datos.

```
define( 'WP_POST_REVISIONS', 10 );
```

```
59
60 /**
61  * WordPress Database Table prefix.
62  *
63  * You can have multiple installations in one database if you give each
64  * a unique prefix. Only numbers, letters, and underscores please!
65  */
66 $table_prefix = 'wp_';
67
68 define( 'WP_POST_REVISIONS', 10 ); ←
69
70 /* That's all, stop editing! Happy blogging. */
71
72 /** Absolute path to the WordPress directory. */
73 if ( ! defined( 'ABSPATH' ) )
74     define( 'ABSPATH', dirname( __FILE__ ) . '/' );
75
76 /** Sets up WordPress vars and included files. */
77 require_once ABSPATH . 'wp-settings.php';
78
```

Line 28, Column 1



Limitar revisiones con Perfmatters

O puede utilizar un plugin como Perfmatters para limitar revisiones.

3. Desactivando Revisiones

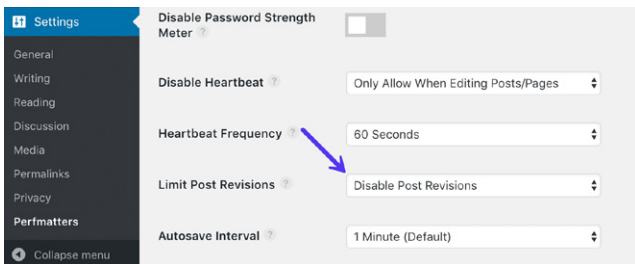
Y, por último, pero no por ello menos importante, también puede deshabilitar completamente las revisiones en su sitio. Si elige hacerlo, le recomendamos seguir la primera opción de arriba para borrar revisiones y luego deshabilitarlas. De esta forma su base de datos estará completamente libre de todas las viejas revisiones y no se agregaran nuevas de ahora en adelante.

Para deshabilitar revisiones, puede agregar el siguiente código a su archivo `wp-config.php`. El código siguiente necesita ser insertado por encima de `'ABSPATH'` de otra forma, no funcionará.

```
define('WP_POST_REVISIONS', false);
```

```
60  /**
61  * WordPress Database Table prefix.
62  *
63  * You can have multiple installations in one database if you give each
64  * a unique prefix. Only numbers, letters, and underscores please!
65  */
66  $table_prefix = 'wp_';
67
68  define('WP_POST_REVISIONS', false); ←
69
70  /* That's all, stop editing! Happy blogging. */
71
72  /** Absolute path to the WordPress directory. */
73  if ( ! defined( 'ABSPATH' ) )
74      define( 'ABSPATH', dirname( __FILE__ ) . '/' );
75
76  /** Sets up WordPress vars and included files. */
77  require_once ABSPATH . 'wp-settings.php';
78
```

Line 33, Column 34



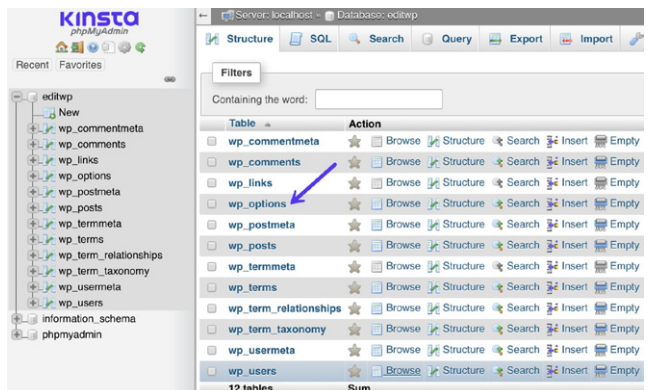
Desactivar revisiones con Perfmatters

O puede utilizar un plugin como Perfmatters para deshabilitar revisiones.

Limpie su Tabla de wp_options y Datos Autocargados

La tabla de wp_options normalmente no es tomada en cuenta cuando se trata en general WordPress y el rendimiento de la base de datos. Especialmente en sitios antiguos y grandes, este puede ser el culpable de tiempos de consultas lentas en su sitio debido a datos autocargados que son dejados atrás por plugins y temas. Confíe en nosotros; vemos esto todos los días.

La tabla wp_options contiene todo tipo de datos para su sitio de WordPress como:



- URL del sitio, URL de página de inicio, email del admin, categoría por defecto, publicaciones por página, formato de tiempo, etc.
- Configuración para plugins, temas y widgets
- Datos temporalmente almacenados en el caché

Esta tabla contiene los siguientes campos (columnas):

- option_id
- option_name
- option_value
- **autoload** (este es el que nos importa cuando se trata de desempeño)

+ Options

| | option_id | option_name | option_value | autoload |
|---|-----------|-----------------|-----------------------------|----------|
| <input type="checkbox"/> Edit Copy Delete | 1 | siteurl | https://editwp.com | yes |
| <input type="checkbox"/> Edit Copy Delete | 2 | home | https://editwp.com | yes |
| <input type="checkbox"/> Edit Copy Delete | 3 | blogname | editwp | yes |
| <input type="checkbox"/> Edit Copy Delete | 4 | blogdescription | Just another WordPress site | yes |

Una de las cosas más importantes de entender sobre la tabla `wp_options` es el campo de **autocargado**. Este contiene un valor de sí o de no (flag). Esencialmente controla si ha sido cargado o no por la función `wp_load_alloptions()`. Los datos autocargados son datos que son cargados en cada página de su sitio de WordPress. Ya le mostramos cómo **deshabilitar que ciertos scripts** sean cargados en todo el sitio, la misma idea funciona aquí. El atributo de autocargado está en “sí” por defecto para los desarrolladores, pero teóricamente, no todos los plugins deberían cargar sus datos en cada página.

Un problema que los sitios de WordPress podrían encontrarse, es cuando hay una gran cantidad de datos autocargados en la tabla `wp_options`. Esto es el culpable del siguiente resultado:

- Los datos están siendo autocargados por un plugin cuando debería estar establecido como “no.” Un buen ejemplo de esto sería un plugin de formulario de contacto. ¿Necesita cargar datos en cada página o sólo la página de contacto?
- Los plugins o temas han sido eliminados del sitio de WordPress, pero sus opciones siguen estando en la tabla de `wp_options`. Esto podría significar que los datos autocargados innecesarios siguen siendo consultados en cada petición.
- Los desarrolladores de plugins y temas están cargando datos en la tabla de `wp_options` en lugar de utilizar sus propias tablas. Hay argumentos a favor y en contra en este tema, ya que algunos desarrolladores prefieren plugins que no

creen tablas adicionales. Sin embargo, la tabla de `wp_options` no fue diseñada para retener miles de filas.

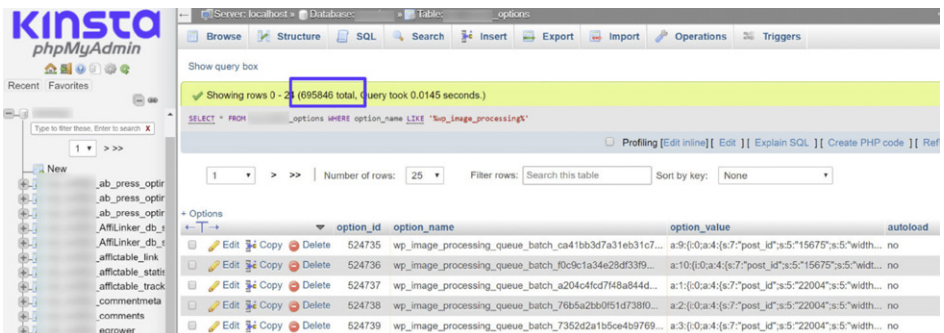
¿Cuál es el límite de los datos autocargados? Esto puede variar, pero idealmente, usted tiene que tener valores entre 300 KB a 1 MB. Una vez que empiece a acercarse al rango de 3-5 MB o más, hay probablemente cosas que pueden ser optimizadas o eliminadas de ser autocargadas. Y cualquier valor por encima de 10 MB debería ser resuelta de inmediato. Esto no siempre quiere decir que causará un problema, pero es un buen lugar por donde empezar.

La limpieza de datos de carga automática es algo con lo que ayudamos a los clientes todo el tiempo en Kinsta.

¿Cuándo fue la última vez que limpió su wp_options table? Sí... eso creímos. ¡Tiene que hacerlo ya!

Limpiando Transitorios

Al menos que esté utilizando un caché de objeto, WordPress almacena registros transitorios en la tabla de `wp_options`. Normalmente estos reciben un tiempo de expiración y deberían desaparecer con el tiempo. Sin embargo, esto no siempre es así. Hemos visto algunas bases de datos donde hay miles de viejos registros transitorios. De hecho, en un sitio, hemos tenido que lidiar con algunos registros transitorios corruptos en donde más de 695,000 filas fueron generadas en la tabla `wp_options`. ¡WoW!



The screenshot shows the phpMyAdmin interface. On the left is the navigation pane with a tree view of databases and tables. The main area displays the 'wp_options' table structure and a query result. The query is: `SELECT * FROM wp_options WHERE option_name LIKE 'wp_image_processing%'`. The result shows 25 rows, with a highlighted message: "Showing rows 0 - 25 (695046 total, query took 0.0145 seconds.)". Below the message is a table with columns: `option_id`, `option_name`, `option_value`, and `autoload`. The table contains several rows of image processing queue batch data.

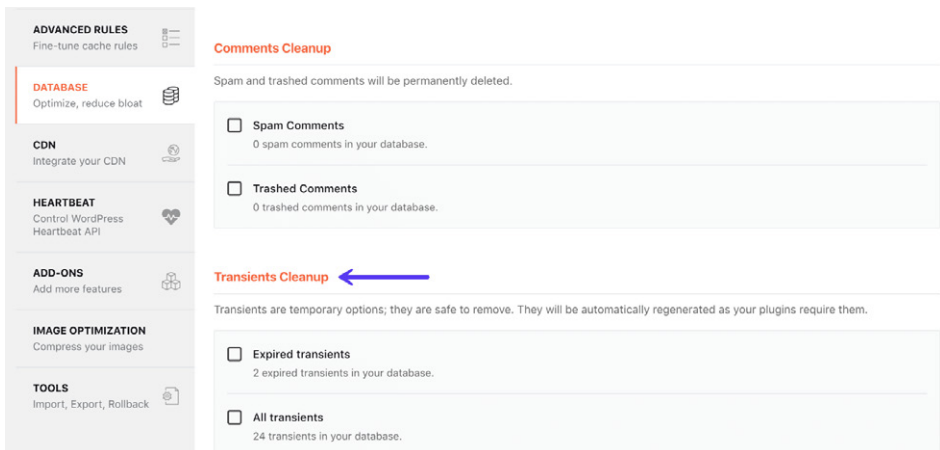
| option_id | option_name | option_value | autoload |
|-----------|---|---|----------|
| 524735 | wp_image_processing_queue_batch_ca41bb3d7a31eb31c7... | a:9:{0:a:4:{s:7:"post_id";s:5:"15675";s:5:"width...} | no |
| 524736 | wp_image_processing_queue_batch_f0c9c1a34e29df339... | a:10:{0:a:4:{s:7:"post_id";s:5:"15675";s:5:"width...} | no |
| 524737 | wp_image_processing_queue_batch_a204c4fd748a844d... | a:1:{0:a:4:{s:7:"post_id";s:5:"22004";s:5:"width...} | no |
| 524738 | wp_image_processing_queue_batch_76b5a2bb0f51d738f0... | a:2:{0:a:4:{s:7:"post_id";s:5:"22004";s:5:"width...} | no |
| 524739 | wp_image_processing_queue_batch_7352d2a1b50e4b9769... | a:3:{0:a:4:{s:7:"post_id";s:5:"22004";s:5:"width...} | no |

Es importante tener en cuenta que los transitorios no son autocargados por defecto. Podría utilizar un query como el de abajo para ver si hay datos transitorios autocargados.

```
SELECT *
FROM `wp_options`
WHERE `autoload` = 'yes'
AND `option_name` LIKE '%transient%'
```

Una mejor y más segura opción sería utilizar un plugin gratuito como Transient Cleaner o Delete Expired Transients que pueden ayudarle a limpiar solo los transitorios expirados de su tabla `wp_options`. Sin embargo, aparentemente ahora hay una función en WordPress que fue agregada en 4.9, que limpia transitorios expirados. Así que esperemos, que esto suceda automáticamente en su sitio.

WP Rocket también tiene la habilidad de limpiar transitorios en sus opciones de optimización de base de datos.



Limpiando Sesiones de WordPress

Otro problema común que hemos visto es que, en algunas ocasiones, los cron jobs dejan de estar sincronizados o no se activan de forma apropiada, y por eso, las sesiones no son depuradas. Podría terminar obteniendo un montón de filas `_wp_session_` en

su base de datos. En el ejemplo siguiente el sitio terminó con más de 3 millones de filas en su tabla de `wp_options`. Y la tabla ha crecido a más de 600 MB en tamaño.

| Table | Rows | Engine | Collation | Charset | Size |
|-------------|------------|--------|--------------------|---------|-----------|
| wp_links | | InnoDB | utf8mb4_unicode_ci | | 32 K B |
| wp_options | ~3,419,339 | InnoDB | utf8mb4_unicode_ci | | 620.9 MiB |
| wp_postmeta | 12,986 | InnoDB | utf8mb4_unicode_ci | | 9.1 MiB |
| wp_posts | 2,367 | InnoDB | utf8mb4_unicode_ci | | 4 MiB |

Podría utilizar un query como la siguiente para ver si tiene este problema:

```
SELECT *
FROM `wp_options`
WHERE `option_name` LIKE '_wp_session_%'
```

<< < 6491 > >> | Number of rows: 500 | Filter rows: Search this table | Sort by key: None

| option_id | option_name | option_value | autoload |
|-----------|---|---------------------------------|----------|
| 3301835 | _wp_session_254a6d16b4400d5aabf3c48b41b72d2b | a:1:{s:10:"edd_errors";s:0:"";} | no |
| 3301836 | _wp_session_expires_254a6d16b4400d5aabf3c48b41b72d... | 1520801306 | no |
| 3301837 | _wp_session_747c86aefae52149d1068a60f240249f | a:1:{s:10:"edd_errors";s:0:"";} | no |
| 3301838 | _wp_session_expires_747c86aefae52149d1068a60f24024... | 1520801306 | no |
| 3301839 | _wp_session_17aa53659273873116d83993bae7cfd8 | a:1:{s:10:"edd_errors";s:0:"";} | no |
| 3301840 | _wp_session_expires_17aa53659273873116d83993bae7cf... | 1520801306 | no |
| 3301841 | _wp_session_f3f29f822eec97951504e4d66afcd081 | a:1:{s:10:"edd_errors";s:0:"";} | no |
| 3301842 | _wp_session_expires_f3f29f822eec97951504e4d66afcd0... | 1520801306 | no |
| 3301843 | _wp_session_2d4be94abb1e766ee229506e1c26e60e | a:1:{s:10:"edd_errors";s:0:"";} | no |

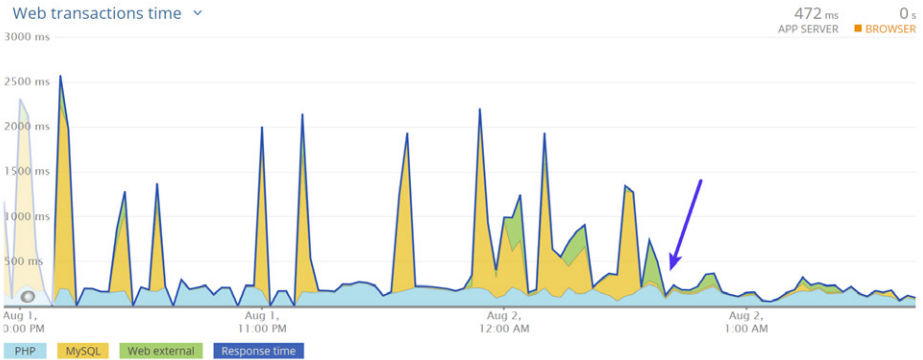
En la mayoría de los casos puede eliminarlos de forma segura (debería configurar un cron job) utilizando el siguiente comando:

```
DELETE FROM `wp_options`
WHERE `option_name` LIKE '_wp_session_%'
```

Después de limpiar todas las filas restantes de `_wp_session_` rows la tabla tenía menos de 1,000 filas y se redujo a 11 MB de tamaño.

| Table | Rows | Engine | Collation | Charset | Size |
|-------------|--------|--------|--------------------|---------|----------|
| wp_links | | InnoDB | utf8mb4_unicode_ci | | 32 K B |
| wp_options | 672 | InnoDB | utf8mb4_unicode_ci | | 11.5 MiB |
| wp_postmeta | 12,986 | InnoDB | utf8mb4_unicode_ci | | 9.1 MiB |
| wp_posts | 2,367 | InnoDB | utf8mb4_unicode_ci | | 4 MiB |

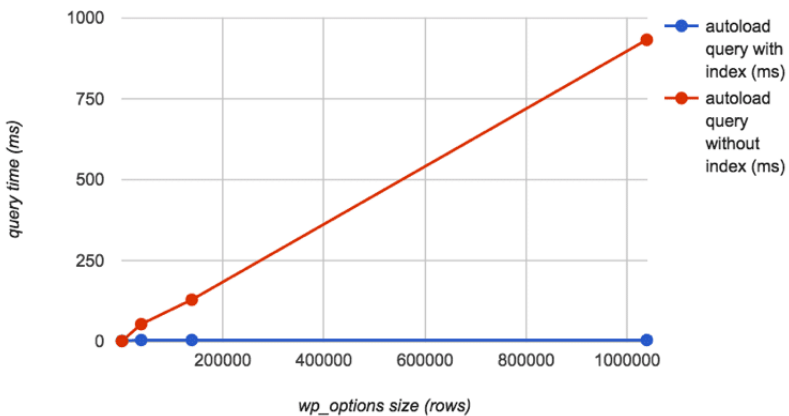
También reparó los picos que el sitio estaba obteniendo en MySQL.



Agregue un Índice para Autocargar

Limpiar la tabla de `wp_options` no fue suficiente, podría intentar agregar un “índice” en el campo de autocarga. Esto básicamente ayuda a que pueda buscar, de forma más eficiente. El sorprendente equipo de 10up llevó a cabo algunos escenarios de prueba en la tabla `wp_options` con un número típico de registros autocargados para mostrar cómo el agregar un índice de autocarga a las queries de `wp_options` puede mejorar el rendimiento.

Tamaño de `wp_options` vs tiempo de consulta



Fuente de la imagen: 10up

Utilice Redis como un Caché de Objeto Persistente para WordPress

Redis es un almacén de estructura de datos en memoria de código abierto. En el contexto de WordPress, Redis puede ser usado para almacenar los valores generados por el caché de objeto nativo de WordPress persistentemente para que los objetos en el caché puedan ser reusados entre las cargas de página.

Utilizar un caché de objeto persistente como Redis, permite que se puedan **reusar los objetos en el caché** en



lugar de requerir que la base de datos de MySQL sea consultada por segunda vez por el mismo objeto. El resultado es que Redis puede reutilizar la carga en la base de datos de MySQL en un sitio web, simultáneamente reduciendo el tiempo de respuesta del sitio e incrementando la habilidad de escalar del sitio y el poder lidiar con tráfico adicional.

Sitios altamente dinámicos (WooCommerce, sitios de membresías, foros de discusión, blogs con sistema de comentarios extremadamente activo) que no pueden aprovechar de forma adecuada el uso del caché de la página son candidatos potenciales para una opción de caché de objeto persistente como Redis.

Si usted es cliente de Kinsta, ofrecemos un complemento de Redis.

Utilice Elasticsearch para Acelerar la Búsqueda de WordPress

Elasticsearch es un motor de búsqueda de texto completo de software libre. Es usado para indexar datos y buscar los datos increíblemente rápido.

En el contexto de WordPress, Elasticsearch puede ser utilizado para **acelerar la consulta de una base de datos de WordPress**. Esto se hace al construir un índice del contenido de la base de datos de su sitio y luego utilizar Elasticsearch para buscar este índice mucho más rápido de lo que es capaz una consulta de MySQL para hacer lo mismo.



Si tiene el tiempo y la habilidad. Elasticsearch puede ser integrado con un sitio de WordPress por un desarrollador altamente conocedor de WordPress y Elasticsearch. Si su sitio hace un uso relativamente estándar de `WP_Query`, Elasticsearch también puede ser integrado al instalar ElasticPress, un plugin gratuito de WordPress de 10Up, disponible de WordPress.org, el cual automáticamente se integra con el objeto `WP_Query` para generar resultados de query con Elasticsearch en lugar de hacerlo con MySQL.

Cualquier sitio que utilice bastante `WP_Query` puede beneficiarse de Elasticsearch. Aquí tenemos unos ejemplos de sitios que pueden beneficiarse de Elasticsearch:

- Sitios en donde la búsqueda es la principal forma de navegación.
- Sitios de WooCommerce con un gran número de pedidos donde los administradores del sitio necesitan poder buscar la lista de pedidos regularmente.
- Cualquier sitio con un gran número de publicaciones donde los queries de MySQL están produciendo bajos resultados.

Como con Redis, también tenemos un complemento de Elasticsearch.

Deshabilitar Opciones No-Críticas que Utilizan Mucho la Base de Datos

Esto puede ser un poco obvio, pero puede haber un mundo de diferencia si usted deshabilita las opciones de plugins que no son críticas y que requieren usar bastante la base de datos.

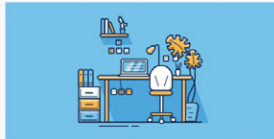
- Los widgets y plugins de publicaciones populares o relacionadas son terribles. Estos normalmente tienen queries pesados a nivel del sitio.
- Plugins de optimización de imágenes que comprimen imágenes utilizando su servidor. Siempre debería utilizar un plugin de optimización de imagen que optimice las imágenes de forma externa.

Si usted visita el blog de Kinsta y llega al final de la publicación, verá que tenemos lo que llamamos artículos relacionados “seleccionados.” Estos artículos son seleccionados manualmente por nosotros y asignados a la publicación. Esto reduce el query a casi nada y no dañará el rendimiento del sitio entero. ¿Toma más trabajo? Sí, pero la experiencia de usuario es mejor si usted elige qué quiere que vean sus lectores.

Hand-picked related articles



Working Remotely – Everything You Need to Know Beforehand



The 53 Best Tools for Freelancers to Scale a (Real) Business



Where and How to Hire a WordPress Developer

Así que, ¿cómo logramos esto? Utilizamos el sorprendente plugin de Advanced Custom Fields (disponible en WordPress.org) y luego asignamos estos campos a nuestro tipo de publicación de blog. Esto nos permite buscar y asignar cualquier contenido relacionado que queramos a cada una de nuestras publicaciones del blog (como podrá observar a continuación).

Post Settings

Related Posts

Choose hand-picked related posts here. If none are chosen (or fewer than needed for the display) we will fill the rest up with posts from within the same category/tags.

Search...

10 Best WordPress Directory Plugins to Organize Your Listings (draft)

10 Best WordPress Security Plugins to Lock out the Bad Guys

10 High-converting WordPress Landing Page Plugins With Beauty and Optimization Combined (draft)

10 Stunning Places to Find Free Images For WordPress

10 Things Not To Do In PHP 7

How to Score 100/100 in Google PageSpeed Insights with WordPress

10 Unique WordPress Ecommerce Strategies to Skyrocket Sales

Conversion Rate Optimization Tips – 12 Easy Ways to Boost Sales

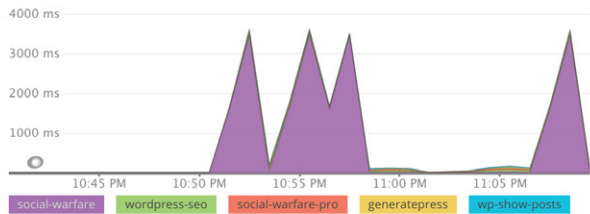
También recomendamos alejarse de plugins que agreguen un contador de vistas/publicaciones a su sitio, al menos que realmente lo necesite. Por ejemplo, evite cosas como “792 publicaciones” a un lado del avatar del usuario en las publicaciones de foro o “5,243 vistas” al listar publicaciones de foro. Cuando uno tiene una larga discusión, estos contadores serán una carga pesada para su base de datos. En general, minimice el uso de contadores y sólo utilícelos si es necesario.

Esto también es cierto para muchos contadores sociales. Por ejemplo, en este sitio, en la parte de abajo, podrá ver el tiempo de respuesta del popular plugin Social Warfare, este tiene 30 veces más que el siguiente plugin justo debajo de él. El caché está habilitado, pero obviamente, este plugin tiene un límite considerable de rendimiento. Después de deshabilitar el plugin en el sitio, los tiempos de carga disminuyeron instantáneamente y el tiempo de respuesta del escritorio de administración de WordPress mejorarán.

Sort by

| | |
|--------------------|----------|
| social-warfare | 18,9 sec |
| wordpress-seo | 624 ms |
| social-warfare-pro | 449 ms |
| generatepress | 425 ms |
| wp-show-posts | 389 ms |
| aawp | 356 ms |
| gp-premium | 229 ms |
| nogluten | 160 ms |
| imagify | 115 ms |
| wp-review | 107 ms |

Top 5 plugins and themes by response time



CAPÍTULO 11:

Utilizando un Content Delivery Network (CDN)



CDN es acrónimo de content delivery network. Lo que es lo mismo, una red de servidores (también conocidos como POPs) ubicados alrededor del mundo. Están diseñados para hospedar y entregar copias de su sitio de contenido estático de WordPress (y en algunas ocasiones dinámico), como imágenes, CSS, JavaScript, y video streams.

Antes que nada, usted no querrá confundir una CDN con un host de WordPress. Estos son dos servicios completamente distintos. Una CDN no es un reemplazo de su proveedor de hosting, sino que es una forma adicional de incrementar la velocidad de su sitio. Mientras que nuestro hosting aquí es Kinsta es súper rápido, una CDN puede hacer el sitio aún más rápido.

¿Cómo Funciona una CDN?

¿Cómo realmente funciona un CDN? Bueno, por ejemplo, cuando usted hospeda su sitio web con Kinsta, usted tiene que elegir una ubicación de centro de datos, como puede ser USA, Europa, Asia-Pacífico, o América del Sur.

Digamos que elige US Central. Esto quiere decir que su sitio web estará físicamente ubicado en un “servidor host” en Council Bluffs, Iowa. Cuando la gente en Europa visite su sitio web, tomará más tiempo para que se cargue, comparado con alguien que vive en Dallas, TX.

¿Por qué? Porque los datos tienen que viajar distancias más largas. Esto es a lo que llamamos latencia. La latencia se refiere al tiempo y/o retraso que sucede durante la transmisión de datos a través de una red. Entre más lejano, mayor será la latencia.

Tipos de CDN

Hay dos distintos tipos de content delivery networks:

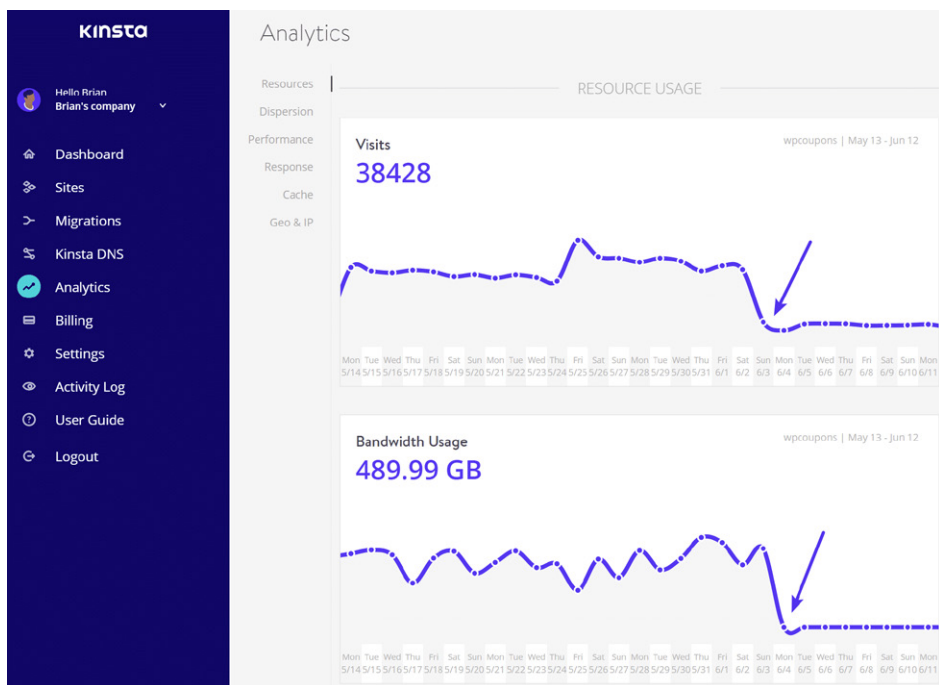
1. Traditional Pull CDN
2. Reverse Proxy CDN

Los Traditional pull CDNs hacen caché a una copia de todo su contenido y archivos multimedia, pero una petición del cliente aún se hace directamente a su proveedor de hosting. KeyCDN y CDN77 son ejemplos de CDNs tradicionales.

Un reverse proxy CDN es un poco diferente. Mientras que sigue actuando como una CDN, este intercepta todas las peticiones entrantes y actúa como un servidor intermediario entre el cliente y su host. Cloudflare y Sucuri son ejemplos de reverse proxy CDNs. Esta es una razón por la que uno tiene que apuntar su DNS directo a estos proveedores en lugar de a su host.

El beneficio de esto es porque actúan como un servidor intermediario, pueden brindar apps de firewalls potentes, los cuales pueden ayudar a bloquear y evitar que el tráfico dañino llegue a su sitio de WordPress y/o proveedor de hosting. Una cosa negativa de esto es que vienen con una pequeña carga adicional en términos de rendimiento, comparado a un traditional pull CDN. Pero con un desempeño adicional y opciones de seguridad, esto podría ser considerado despreciable.

A continuación tenemos un ejemplo de lo que pasó después de habilitar Sucuri en el sitio de un cliente. Como puede ver, tuvo un impacto dramático en la cantidad de tráfico negativo que entrante. Al final, este tipo de servicios pueden ayudarle a ahorrar coste de hosting.



Pruebas de Velocidad de CDN

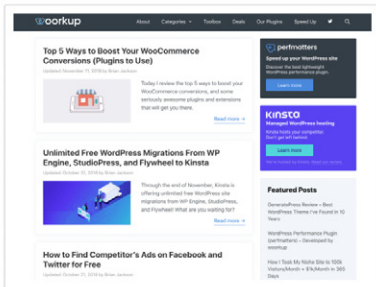
Anteriormente hablamos sobre los grandes beneficios del caché de WordPress. Bueno, el caché de CDN también es muy poderoso. Esto es porque normalmente las CDNs tienen más ubicaciones de servidores que los proveedores de hosting. Esto significa que puede cachear todos sus archivos (imágenes, JS, CSS) en un lugar más cercano a sus visitantes y servirlos en tiempo record.

Hagamos unas rápidas pruebas para ver qué tan rápido puede llegar a ser su sitio con una CDN.

Sin CDN

Nuestro sitio de prueba tiene como host a Kinsta y está físicamente ubicado en el centro de datos de Iowa, USA. Primero hicimos cinco pruebas de velocidad en Pingdom

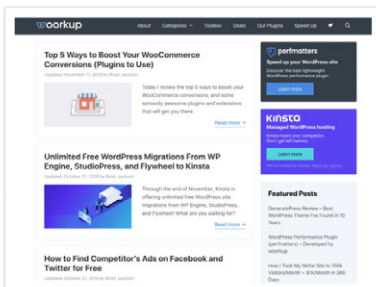
(sin la CDN habilitada), y tomamos el promedio. Importante: Estamos utilizando la ubicación de Europa-Reino Unido-Londres en Pingdom para demostrar el verdadero poder de una CDN. El tiempo de carga total fue de **1.03 s**.



| | |
|----------------------------------|------------------------------|
| Performance grade B 89 | Page size 231.0 KB |
| Load time 1.03 s | Requests 40 |

Con CDN

Luego habilitamos nuestra CDN e hicimos cinco pruebas adicionales de velocidad en Pingdom. Nuestro tiempo de carga total es ahora de **585 ms** desde la ubicación de prueba de Pingdom en Europa – Reino Unido – Londres. Así que al usar una CDN, **podimos reducir el tiempo de carga de nuestras páginas por un 43.2%**. Esto es bastante.

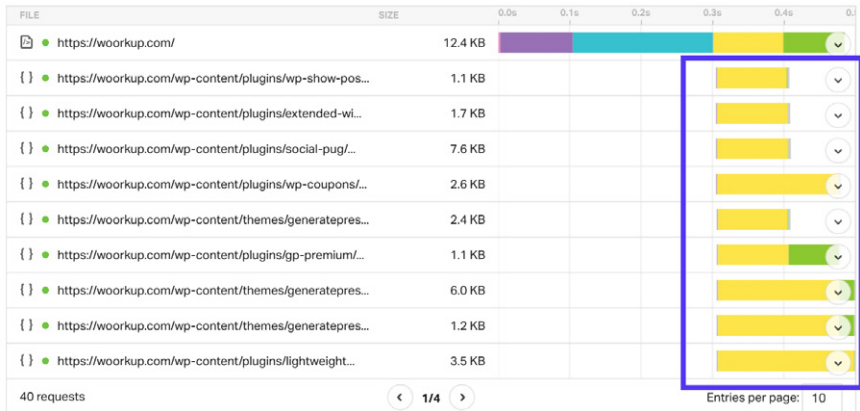


| | |
|----------------------------------|------------------------------|
| Performance grade B 89 | Page size 230.6 KB |
| Load time 585 ms | Requests 40 |

La razón de esta drástica diferencia es porque la CDN tiene un centro de datos en Londres. Esto quiere decir que todos los archivos son almacenados en el caché en esa ubicación y listas para ser servidas con una mínima latencia.

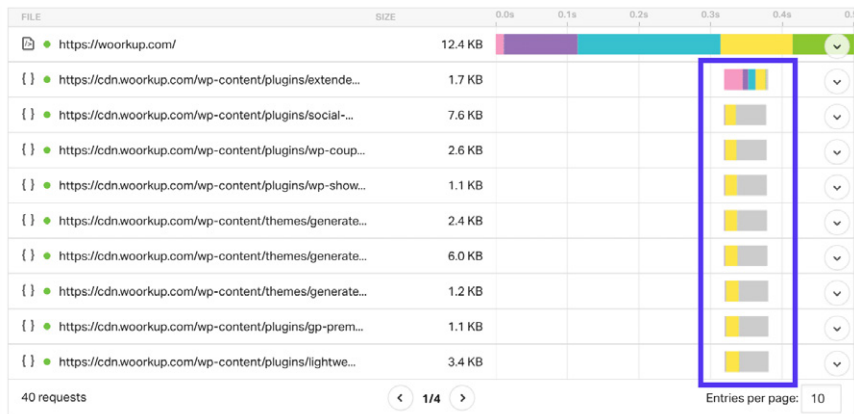
TTFB sin CDN

Recuerde que esa barra amarilla en Pingdom es el tiempo de espera, la cual es el tiempo del primer byte (TTFB). En nuestras pruebas de velocidad corriendo sin CDN, el TTFB promedio en archivos fue de **98 ms**.



TTFB con CDN

Una vez que habilitamos el CDN, el TTFB promedio en los archivos se redujo a un promedio de 15 ms. Al utilizar una CDN nuestro TTFB promedio se redujo por un 84.69%. Esto es principalmente porque los archivos están siendo servidos directamente desde el caché de la CDN.



¿Una CDN disminuye los tiempos de carga de nuestra página por un 43.2%! Revise la razones por la que debería usar una.

¿Cómo Habilitar una CDN?

Habilitar un CDN en su sitio de WordPress no tiene que ser difícil, y de hecho es bastante sencillo! Sólo siga los siguientes pasos.

Paso 1

Seleccione un proveedor de CDN y suscríbese al servicio. Normalmente cobran por mes o por uso de datos. La mayoría de los proveedores tendrán una calculadora para estimar sus costos.

- Cada proveedor de CDN debería tener documentación para ayudarle a empezar.
- Tenemos tutoriales detallados sobre cómo instalar Cloudflare y cómo instalar Sucuri que están disponibles en nuestra web kinsta.com.

Paso 2

Si está utilizando una traditional pull CDN, puede utilizar un plugin gratuito como CDN Enabler, WP Rocket, o Perfmatters para integrarlo con su sitio de WordPress. Estos plugins automáticamente enlazarán sus archivos a la CDN. No tendrá que hacer nada para lograr que esté su contenido en la CDN; ¡todo sin tocar nada! El Reverse Proxy CDN normalmente no requiere ni un plugin, aunque en algunas ocasiones lo tienen para habilitar opciones adicionales.

- Dashboard
- K Kinsta Cache
- Posts
- Media
- Forms
- Pages
- Downloads
- Jilt
- Knowledge Base
- Appearance
- Plugins

Perfmatters Settings

Options
CDN
Google Analytics
Extras
License
Support

CDN

CDN options that allow you to rewrite your site URLs with your CDN URLs.

Enable CDN Rewrite ?

CDN URL ?

Included Directories ?

CDN Exclusions ?

Activar el CDN en WordPress con Perfmatters

¿Cómo Habilitar la Kinsta CDN?

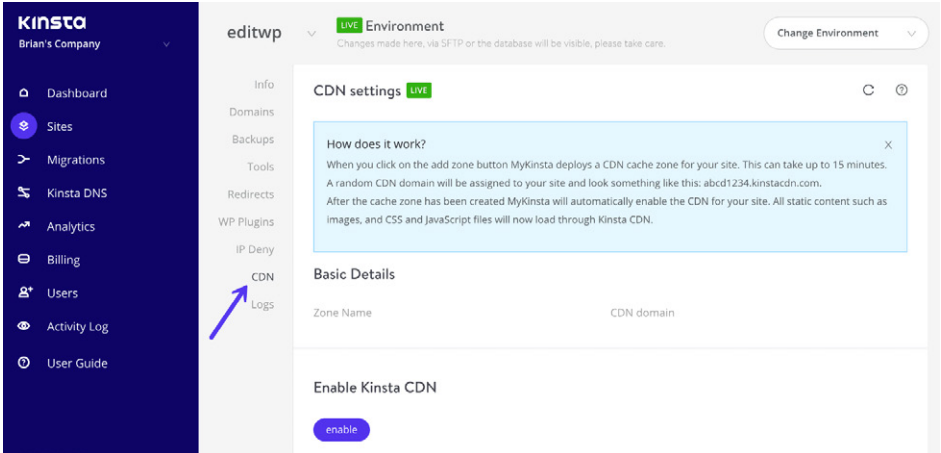
¿Le han gustado las pruebas de velocidad de CDN anteriores? Estuvimos utilizando KeyCDN en esas pruebas. Las buenas noticias es que nuestra Kinsta CDN utiliza KeyCDN. Es un content delivery network HTTP/2, con 34 ubicaciones, para cargar rapidísimo sus ficheros y archivos multimedia alrededor del mundo. Actualmente, las regiones a las que sirven incluyen América, América del Sur, Europa, África y Australia.



Si usted es un cliente de Kinsta, incluimos ancho de banda de CDN gratuito en todos nuestros planes de hosting. Puede habilitar la CDN Kinsta en dos simples pasos.

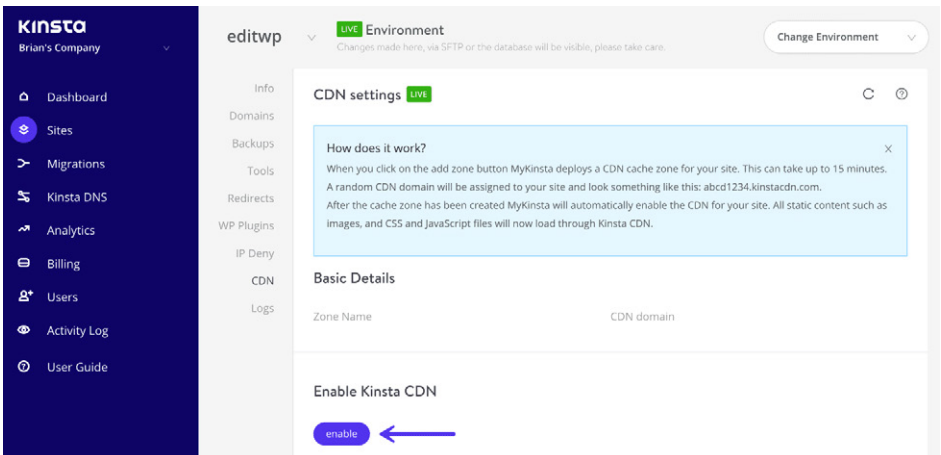
Paso 1

Primero, entre en su escritorio de MyKinsta. Haga clic en su sitio y luego en la pestaña Kinsta CDN.



Paso 2

Luego haga clic en “Habilitar Kinsta CDN.” Después de unos minutos, el CDN automáticamente será lanzado, y sus recursos serán servidos desde el caché alrededor del mundo. Eso es todo.



Optimizaciones adicionales de CDN

Aquí tenemos algunas optimizaciones de CDN adicionales que podría querer revisar o considerar.

- Si tiene muchos comentarios, los avatares pueden generar muchas peticiones. Estos se cargan desde `secure.gravatar.com`. Consulte este tutorial sobre como cargar gravatar desde su CDN. Nosotros lo hacemos en nuestro sitio de Kinsta.
- Puede hospedar sus propias fuentes personalizadas desde su CDN o incluso de Google Fonts en su CDN. Los enlaces a los tutoriales sobre cómo hacerlo están disponibles en nuestro sitio web en <https://kinsta.com/es/aprender/acelerar-wordpress/>

CAPÍTULO 12:

Descargue Archivos Multimedia y Email Cuando lo Necesite



Todo lo que genera una petición tiene un impacto en el rendimiento de su sitio de una forma y otra. Para sitios que alojan cientos de miles de archivos o grandes archivos multimedia, podría ser muy buena idea descargar todo esto por completo. La descarga es diferente al servirlo a través de una CDN. Con una CDN los datos originales aún siguen estando en su host, la CDN simplemente tiene múltiples copias de este.

Cuando el caché expira en los activos de su CDN este pide la consulta de nuevo a su host para las últimas copias de los archivos. Las CDNs están hechas para cachear archivos por largos períodos de tiempo. Pero debido al hecho de que tienen demasiados POPs, podría haber muchas consultas a medida que el caché expira en diferentes regiones.

Cuando uno descarga archivos multimedia o archivos normales, quiere decir que actualmente, están moviendo la ubicación física original de su proveedor de hosting. Así que mientras parece que sus archivos están siendo servidos de su sitio, en realidad se encuentran ubicados en otro lugar completamente distinto. Además de reducir las consultas adicionales de vuelta al host, la razón número uno, es también ahorrar espacio en el disco.

Descargando Archivos Multimedia a Amazon S3

Una de las soluciones de descarga más populares es Amazon S3. Amazon S3 es una solución de almacenamiento, y parte de los muchos productos de Amazon Web Services.

Normalmente solución es usada por sitios grandes que necesitan backups adicionales o están sirviendo archivos grandes (descargas, software, videos, juegos, archivos de audio, PDFs, etc.) Amazon es muy confiable, y por su infraestructura masiva, puede ofrecer bastante almacenamiento a precios muy bajos. Algunos de los clientes de S3 incluyen Netflix, Airbnb, SmugMug, Nasdaq, etc.



Porque normalmente lidian con almacenamiento en masa, puede tener por garantizado que sus precios serán menores que su host de WordPress. Descargar archivos multimedia a AWS puede ser una gran forma de ahorrar dinero y es gratuito por el primer año (hasta 5 GB de almacenamiento). También, porque las peticiones para sus archivos multimedia están servidas directamente desde

Amazon, Esto pone una menor carga a su sitio de WordPress, queriendo decir, tiempos de carga más rápidos.

Recomendamos usar el plugin WP Offload Media Lite para Amazon S3 (disponible en WordPress.org) para esto. El plugin copia los archivos de su sitio de WordPress a Amazon S3 automáticamente a medida que se cargan en la biblioteca de medios. También puede usar un CDN con los medios descargados para conseguir lo mejor de ambos mundos.

Descargar Archivos Multimedia a Google Cloud Storage



Otra solución de descarga popular es el Google Cloud Storage. Ya que en Kinsta utilizamos el Google Cloud Platform, somos grandes fanáticos de su tecnología e infraestructura. Debido a la infraestructura masiva de Google y por el hecho de que normalmente lidian con almacenamiento en masa, ellos también pueden ofrecer precios de almacenamiento muy bajos. Algunos de sus clientes incluyen a Spotify, Vimeo, Coca-Cola, Philips, Evernote, y Motorola.

Recomendamos usar el plugin gratuito WP-Stateless (disponible en WordPress.org) para ello. El plugin copia automáticamente los archivos de su sitio WordPress a Google Cloud Storage a medida que se cargan en la biblioteca multimedia. A continuación, ofrece diferentes modos, como realizar una copia de seguridad de los medios o incluso eliminar los medios de WordPress y servirlos desde Google Cloud Storage.

Descargar Emails Transaccionales y de Marketing

Lo crea o no, los emails tienen un impacto en su servidor y recursos de servidor. Con algunos hosts, especialmente hosts compartidos, abusar de esto podría causar que lo suspendan. Esto se convierte en un problema para aquellos que envían correos de forma masiva. Esta es la razón por la que los proveedores de emails transaccionales existen y la razón por la que muchos proveedores de hosting bloquean el envío de emails en puertos estándares desde el principio. **Nosotros nunca recomendamos el servicio de email de su proveedor de hosting.**

Si está enviando newsletters o bulk emails, siempre recomendamos las siguientes alternativas para obtener los mejores resultados:

- Utilice software de email marketing profesional que no sea parte de WordPress.
- Utilice un proveedor de servicio de email transaccional (HTTP API o SMTP) junto con WordPress.

Otras ventajas de utilizar servicios externos incluyen:

- Mejor envío de emails. ¡Deje que los proveedores de email hagan lo que mejor saben hacer!
- Menos probabilidad de ser enviado a una lista negra.
- Podría no ser posible establecer registros DMARC con sus proveedores de hosting.

Herramientas de Email Marketing

Algunos ejemplos de email marketing, incluye newsletters, anuncios de productos o servicios, ventas, invitaciones a eventos, recordatorios, etc. Aquí tenemos algunas de las herramientas de email marketing que recomendamos:

- **MailChimp** – Usamos MailChimp en Kinsta.
- **MailerLite**
- **Drip**

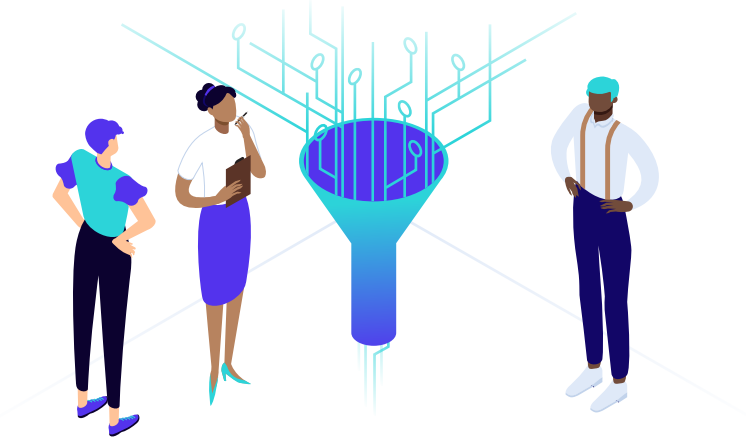
Servicios de Email Transaccionales

Algunos ejemplos de emails transaccionales incluyen recibos de compra de WooCommerce o EDD, notificaciones de creación de cuenta, notificaciones de envío, mensajes de error de apps, restablecimiento de contraseñas, etc. Si usted es cliente de Kinsta, nosotros dependemos de un proveedor externo de SMTP para asegurar un alto nivel de capacidad de entrega. Pero dependiendo de su volumen, siempre recomendamos moverlo fuera del sitio. Aquí tenemos algunos servicios de email transaccional que recomendamos:

- **SendGrid** – Nosotros usamos SendGrid en Kinsta.
- **Mailgun**
- **SparkPost**

CAPÍTULO 13:

¿Cómo Encontrar Cuellos de Botella y Plugins Lentos?



Ahora veremos algunos consejos sobre cómo encontrar cuellos de botella en su sitio de WordPress y qué puede hacer sobre esto.

Utilice New Relic para Identificar Plugins Lentos y Queries de Base de Datos

Hay varias herramientas excelentes en el mercado que le pueden ayudar a encontrar e identificar queries y plugins lentos en la base de datos, aquellos que estén consumiendo mucho tiempo. Nosotros somos grandes fanáticos de New Relic y lo usamos a diario. New Relic es una herramienta monitorea de PHP que puede utilizar para obtener estadísticas detalladas sobre su sitio.

Si usted es cliente de Kinsta, puede agregar su propia llave de licencia de New Relic en el escritorio de su MyKinsta.

The screenshot shows the Kinsta dashboard for 'Brian's Company'. The left sidebar contains navigation options: Dashboard, Sites, Migrations, Kinsta DNS, Analytics, Billing, Users, Activity Log, and User Guide. The main content area is titled 'editwp Environment' and includes a warning: 'Changes made here, via SFTP or the database will be visible, please take care.' Below this are several tool cards: 'Clear Site Cache', 'SSL Certificate', 'PHP Engine', 'Restart PHP', 'New Relic Monitoring', and 'Search And Replace'. The 'New Relic Monitoring' card shows 'Monitoring Disabled' and an 'Enable Monitoring' button, which is highlighted with a blue arrow.

Sin embargo, utilice New Relic con cuidado, ya que este impacta al desempeño del sitio. Este agrega JavaScript a su sitio. Nosotros recomendamos habilitarlo cuando necesite resolver problemas de rendimiento, y luego desactivarlo de nuevo.

Encontrando Plugins Lentos

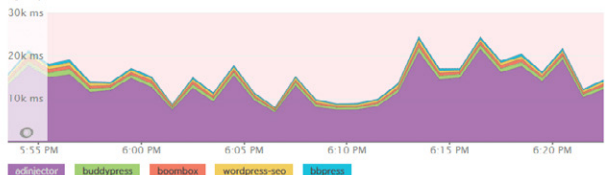
Cuando un plugin de WordPress está causando lentitud generalizada, los síntomas variarán dependiendo de la actividad en la que el plugin se esté desempeñando.

TIME PICKER
Last 30 minutes ending now

Sort by Most time consuming

| | |
|----------------------------|----------|
| adinjector | 390 sec |
| buddypress | 24.4 sec |
| boombox | 17.6 sec |
| wordpress-seo | 15.6 sec |
| bbpress | 12.6 sec |
| easy-social-share-buttons3 | 4.34 sec |
| zombily | 4.22 sec |

Top 5 plugins and themes
by response time



Sin embargo, en muchos casos, descubrirá que un plugin lento afectará cada página de un sitio de WordPress. En el caso del sitio cuyos datos podrá ver en la imagen de abajo, una lentitud generalizada fue observada en cada página de front-end en el sitio. Esto es lo que demostró New Relic tratándose del rendimiento de los plugins del sitio.

Inmediatamente podrá ver que el plugin de adinjector está consumiendo más de 15 veces la cantidad de tiempo que el siguiente plugin más lento.

Cuando uno ve datos como estos, puede ser atractivo pensar inmediatamente que el plugin está mal escrito o que es inefectivo. Mientras que esto puede ser cierto, no siempre es el caso. Una mala configuración del plugin, lentitud de la base de datos, o recursos externos que tarden en responder podrían causar que un plugin consuma más tiempo de lo esperado.

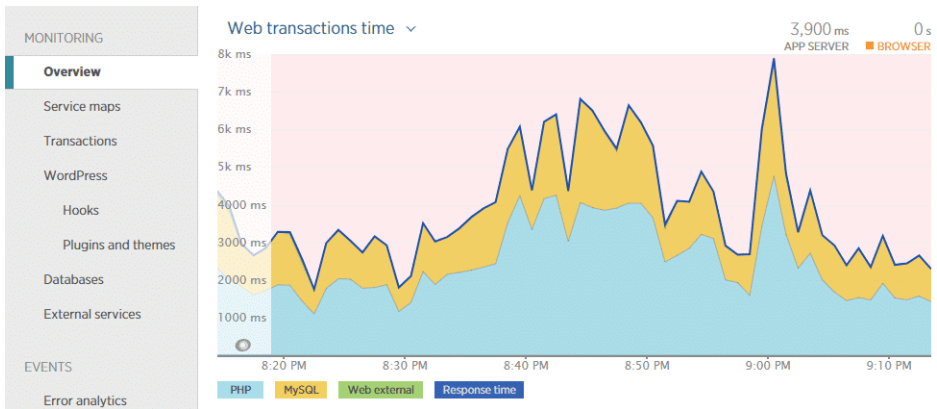
Cuando ve un plugin que está respondiendo lentamente, es buena idea revisar otras pantallas en New Relic para encontrar información adicional. Las transacciones, las bases de datos y recursos externos deberían ser revisados antes de decidir si desactivar el plugin es la mejor opción, o incluso la única opción.

Una Lentitud Generalizada Causada por una Base de Datos Sobrecargada

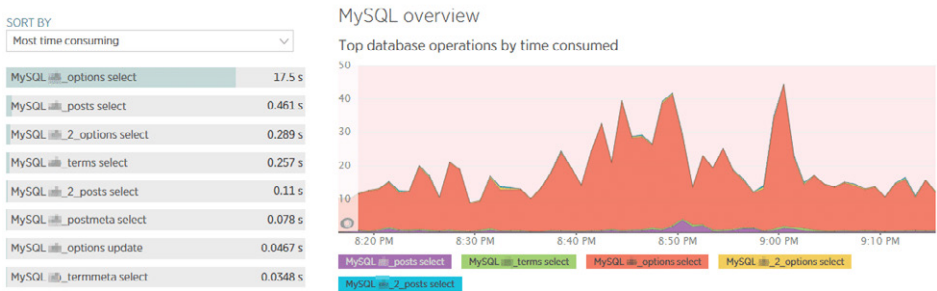
Una base de datos mal optimizada puede causar una lentitud generalizada en un sitio de WordPress. Anteriormente, hablamos sobre todas las cosas que podría hacer para arreglar esto. En New Relic, esta lentitud relacionada con la base de datos probablemente aparecerá en dos lugares:

- Primero, usted verá una gran cantidad de actividad MySQL en la vista general.
- Segundo, usted verá una o más tablas consumiendo mucho tiempo en la pestaña de la base de datos.

Empezando con la pantalla de vista general, un sitio con una base de datos sobrecargada podría verse de la siguiente forma:



Para entender un poco mejor qué tabla de la base de datos o query es el que está causando el problema, vaya a la pestaña de base de datos.



La pestaña de la base de datos apuntará a la tabla y el tipo de query que está consumiendo la mayor parte del tiempo. Si usted selecciona una de estas entradas en la lista, podrá ver más detalles, incluyendo algunos ejemplos de queries.

SORT BY
Most time consuming

| | |
|---------------------------|----------|
| MySQL wp_options select | 17.5 s |
| MySQL wp_posts select | 0.461 s |
| MySQL wp_2_options select | 0.289 s |
| MySQL wp_terms select | 0.257 s |
| MySQL wp_2_posts select | 0.11 s |
| MySQL wp_postmeta select | 0.078 s |
| MySQL wp_options update | 0.0467 s |
| MySQL wp_termmeta select | 0.0348 s |
| MySQL wp_2_options update | 0.0309 s |
| MySQL other | 0.0281 s |
| MySQL wp_sitemeta select | 0.0115 s |

MySQL wp_options select

Slow query trace 2,510 ms /archive-product

9:14 PM MAX TIME ACTION

Query
SELECT option_name, option_value FROM wp_options WHERE autoload = ?

Database instance
H2O

Database name
H2O

Stack trace

```

mysql_query called at /www/html/wp-content/themes/2014/public/wp-includes/wp-db.php (1857)
db::do_query called at /www/html/wp-content/themes/2014/public/wp-includes/wp-db.php (1775)
in wpdb::query called at /www/html/wp-content/themes/2014/public/wp-includes/wp-db.php (2358)
iget_results called at /www/html/wp-content/themes/2014/public/wp-includes/option.php (189)
lloptions called at /www/html/wp-content/themes/2014/public/wp-includes/functions.php (1364)
blog_installed called at /www/html/wp-content/themes/2014/public/wp-includes/load.php (583)
in wp_not_installed called at /www/html/wp-content/themes/2014/public/wp-settings.php (140)
in require_once called at /www/html/wp-content/themes/2014/public/wp-config.php (65)
in require_once called at /www/html/wp-content/themes/2014/public/wp-load.php (37)
in require_once called at /www/html/wp-content/themes/2014/public/wp-blog-header.php (13)
in require called at /www/html/wp-content/themes/2014/public/index.php (17)

```

En este caso, los datos apuntan hacia los datos autocargados en la tabla de `wp_options`. Recuerde, ya hemos repasado esto. Y sin duda alguna, un rápido análisis de la tabla `wp_options` confirma que casi 250 MB de datos son autocargados de esta tabla, haciendo a este sitio un candidato obvio para recibir mantenimiento y optimización de base de datos.

Asegúrese de consultar nuestro detallado tutorial sobre cómo utilizar New Relic para resolver los problemas de rendimiento en su sitio de WordPress.

Utilice el Plugin Gratuito Query Monitor

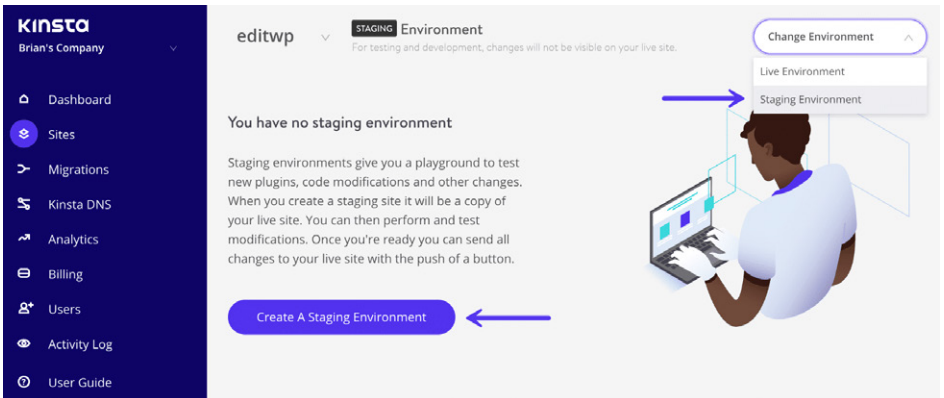
También puede utilizar un plugin de WordPress gratuito como Query Monitor. Utilícelo para identificar y reparar consultas a de base de datos lentas, llamadas AJAX, peticiones de REST API y mucho más. Además de esto, el plugin reporta detalles sobre el sitio, cosas como las dependencias de script y dependientes, los hooks de WordPress que fueron lanzados durante la generación de páginas, detalles del entorno de hosting, etiquetas condicionales de consultas encontradas en la página actual, y mucho más.

| Query Monitor | | | | |
|---|---|-------------------------------|------------------|---|
| Page generation time | Peak memory usage | Database query time | Database queries | Object cache |
| 0.0321 0.0% of 0s limit | 2,048 kB 0.0% of 16,777,216 kB limit | 0.0022 | SELECT: 3 | 99.9% hit rate External object cache: true |
| \$wpdb Queries | | | | |
| Query | Caller | Component | Rows | Time |
| 1 SELECT user_id, meta_key, meta_value FROM wp_usermeta WHERE user_id IN (1) ORDER BY umeta_id ASC | update_meta_cache() wp-includes/meta.php:826 | + Plugin: coming-soon-builder | 39 | 0.0004 |
| 2 SELECT wp_posts.* FROM wp_posts WHERE 1=1 AND (wp_posts.ID = '2') AND wp_posts.post_type = 'page' ORDER BY wp_posts.post_date DESC | WP_Query->get_posts() wp-includes/query.php:3655 | + Core | 1 | 0.0017 |
| 3 SELECT * FROM wp_posts WHERE (post_type = 'page') AND post_status = 'publish') AND post_parent = 2 ORDER BY wp_posts.post_title ASC LIMIT 0,1 | get_pages() wp-includes/post.php:4598 | + Core | 0 | 0.0001 |
| Total Queries: 3 | | | | 0.0022 |

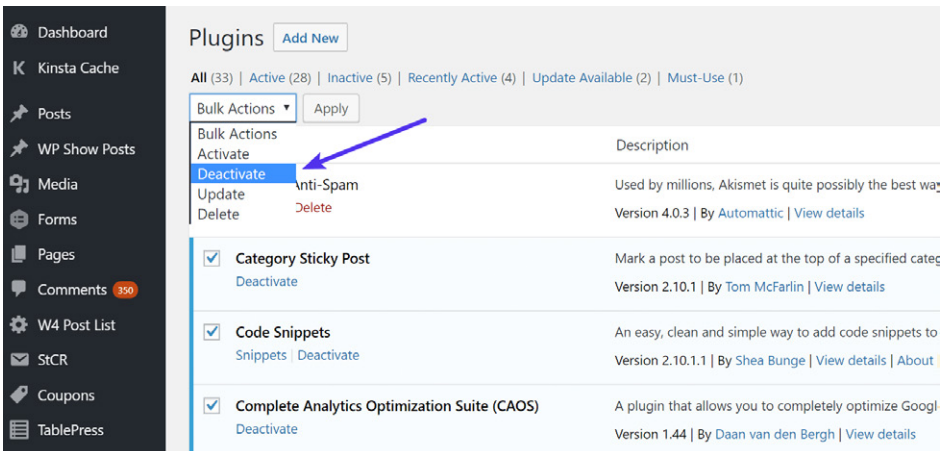
El plugin fue desarrollado por John Blackbourn, un comisionado de WordPress que actualmente es desarrollador en Human Made y previamente trabajaba con WordPress.com VIP – en otras palabras, alguien que conoce muy bien WordPress. Query Monitor fue agregado al directorio de plugin de WordPress en 2013 y actualmente cuenta con más de 10,000 instalaciones activas – una suma impresionante para un plugin de desarrollo. La calificación de cinco estrellas de cinco ayuda a explicar la razón de su popularidad entre los desarrolladores.

Utilizando Sitios de Prueba Sin Tocar el Sitio en Producción

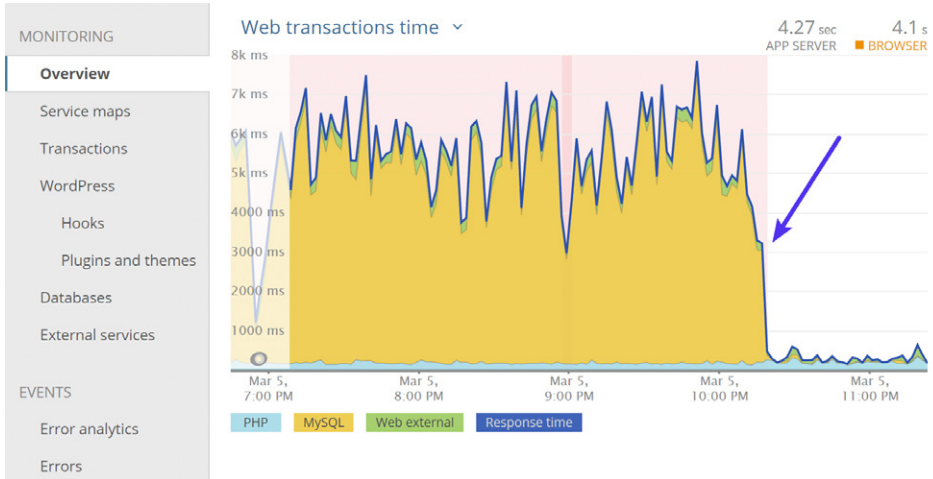
No sabemos qué haríamos sin los entornos de prueba. Estos pueden ser de un valor incalculable cuando se trata de resolver problemas de rendimiento. Afortunadamente, Kinsta tiene entornos de desarrollo con tan sólo dar un clic. Si su host de WordPress no le ofrece entornos de staging. Podría utilizar un plugin como WP Staging, aunque no es algo sencillo.



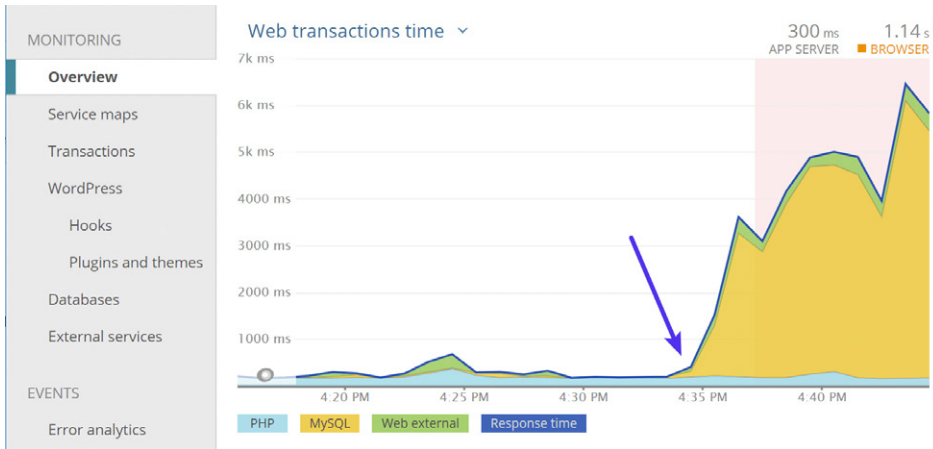
Después de tener un sitio de prueba funcionando, lo primero que puede hacer es deshabilitar todos los plugins. Ya que esto es una copia de su sitio en vivo, no tendrá que preocuparse de romper algo. Sin duda alguna, es una de las formas más sencillas para identificar problemas. Simplemente vaya a Plugins, seleccione todos y elija “Desactivar” en las acciones en lote.



Después de hacer esto, puede monitorear los tiempos de respuesta en New Relic o Query Monitor y ver qué pasa. En este ejemplo de abajo los tiempos de respuesta inmediatamente bajaron a niveles normales en el sitio, así que sabíamos que era uno de los plugins el que estaba causando un problema. Luego puede reactivarlos uno por uno, repitiendo el mismo proceso hasta encontrar al culpable.



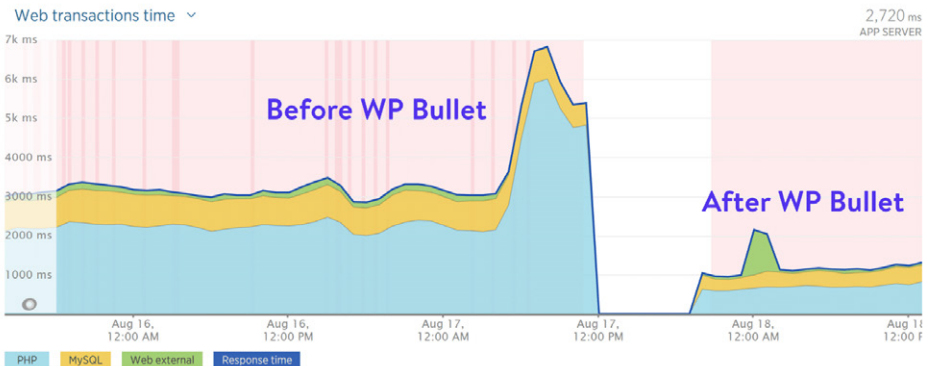
Aquí hay un ejemplo de qué pasó cuando habilitamos el plugin que estaba causando problemas. Los tiempos de carga (tiempos de transacción en la red) inmediatamente subieron.



¿Qué hacer después de encontrar el plugin que está causando problemas? Este es nuestro consejo:

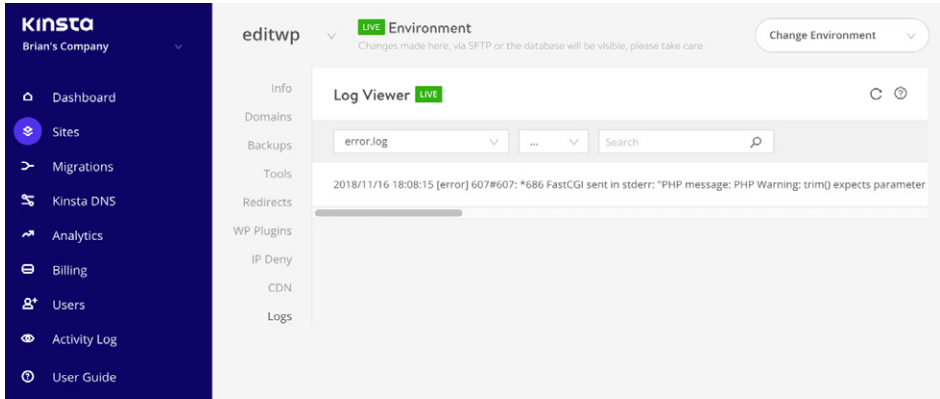
1. Actualice sus plugins y temas, a las últimas versiones, si es que aún no lo ha hecho.
2. Comuníquese con el desarrollador del plugin o tema y pida asistencia.
3. Encuentre un plugin alternativo que pueda ofrecer la misma funcionalidad.
4. Quizás su versión de PHP esté causando un problema. Cambie el motor de su PHP a una versión anterior y vea si el plugin o tema aún funciona.

También puede contratar a un desarrollador de WordPress para que arregle este problema. Si está relacionado con el rendimiento, tenemos que mencionar a Mike Andreason de wp-bullet.com. Es un desarrollador de tiempo completo en Codeable, que se especializa en optimización de rendimiento, él ha ayudado a muchos clientes de Kinsta con instalaciones complejas y llevar sus sitios al siguiente nivel.

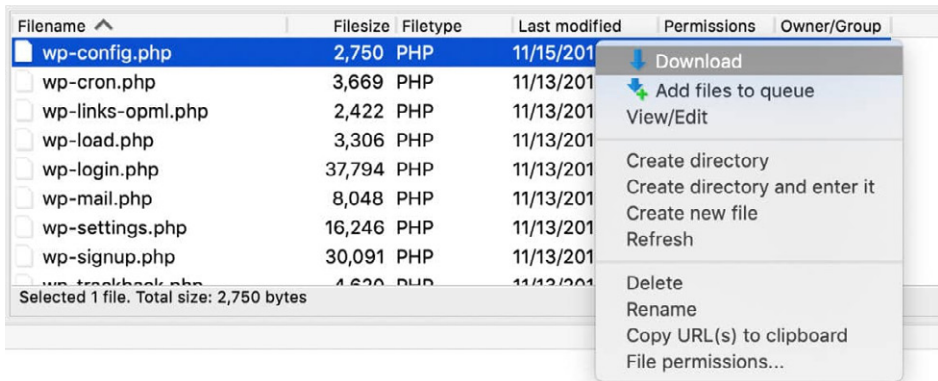


Revise sus Registros de Error (Error Logs)

Revisar los registros de error nunca es divertido, pero puede revelar mucho sobre los problemas de rendimiento de los plugins de WordPress. Si usted es cliente de Kinsta, puede ver fácilmente sus registros de errores, de caché y de acceso desde el escritorio de MyKinsta.



También puede habilitar el registro de errores agregando un poco de código en el archivo de `wp-config.php`. Primero, querrá conectarse a su sitio a través de SFTP. Luego, descargue su `wp-config.php` para que pueda editarlo.



Nota: ¡Siempre tenga listo un backup de este archivo primero!

Encuentre la línea que dice `/* That's all, stop editing! Happy blogging. */` y justo antes de esta, agregue lo siguiente (como se muestra abajo):

```
define( 'WP_DEBUG', true );
```

```

37
38
39
40 define( 'WP_DEBUG', true );
41
42 /* That's all, stop editing! Happy blogging. */
43
44 /** Absolute path to the WordPress directory. */
45 if ( !defined('ABSPATH') )
46     define('ABSPATH', dirname(__FILE__) . '/');
47
48 /** Sets up WordPress vars and included files. */
49 require_once(ABSPATH . 'wp-settings.php');

```

Si el código anterior ya existía en su archivo de `wp-config.php` pero estaba seleccionado como “false,” simplemente cámbielo a “true.” Esto habilitará el modo de debug. Nota: también verá las advertencias o errores en su administración de WordPress si es que existen.

Luego puede habilitar el registro de debugs para enviar todos los errores a un archivo agregando el siguiente código justo después de la línea `WP_DEBUG` (como se ve abajo):

```
define( 'WP_DEBUG_LOG', true );
```

```

38
39
40 define( 'WP_DEBUG', true );
41 define( 'WP_DEBUG_LOG', true );
42
43 /* That's all, stop editing! Happy blogging. */
44
45 /** Absolute path to the WordPress directory. */
46 if ( !defined('ABSPATH') )
47     define('ABSPATH', dirname(__FILE__) . '/');
48
49 /** Sets up WordPress vars and included files. */
50 require_once(ABSPATH . 'wp-settings.php');

```

Guarde los cambios y suba de nuevo el archivo a su servidor. Los errores luego serán registrados en el archivo `debug.log` dentro de su carpeta `/wp-content/`. Si por alguna razón no ve este archivo, siempre podrá crear uno.

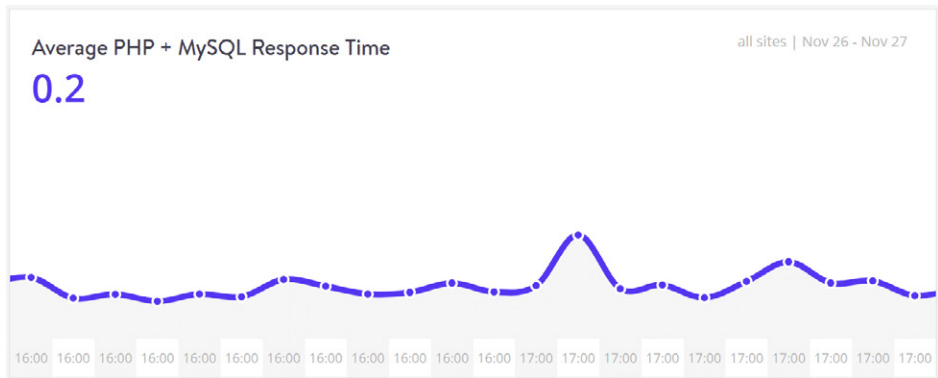
Utilice MyKinsta Analytics

Si usted es cliente de Kinsta, puede tomar ventaja de las funcionalidades de desempeño que hemos construido en nuestra herramienta de MyKinsta Analytics.

Bajo la sección de monitoreo de desempeño, usted podrá ver el tiempo de respuesta promedio de PHP + MySQL, rendimiento de PHP, uso de AJAX, tiempo de subida promedio, y el tiempo máximo de subida.

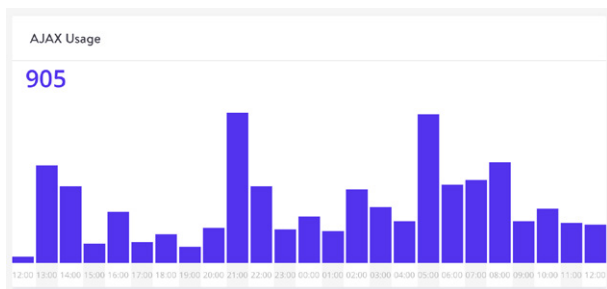
Tiempo Promedio de PHP + Tiempo de Respuesta de MySQL

Al visitar su sitio de WordPress, PHP y MySQL son usados para compilar y hacer consultas a los datos que ve en la página. Esta gráfica le muestra el tiempo de respuesta promedio del motor PHP y el motor MySQL para cada petición dinámica en el caché. Conocer estos tiempos de respuesta podrán ayudarle a resolver los problemas de velocidad.



El Rendimiento de PHP

El rendimiento indica el número de transacciones por segundo que una aplicación puede manejar, y en este reporte, se refiere al rendimiento de PHP desde su sitio de WordPress. En otras palabras, muestra cuántas veces ha sido solicitado un activo de PHP.



El Mejor Tiempo de Respuesta Promedio de PHP + MySQL

Esta lista muestra el mejor tiempo de respuesta promedio de PHP y MySQL. Estos números pueden ser picos de una sola vez, así que se sugiere comparar esta lista con “El Mejor Tiempo de Subida Máximo.”

| Top Average PHP + MySQL Response Time | | |
|--|----------|--------|
| PATH | REQUESTS | TIME |
| /wp-admin/update.php?action=upload-plugin | 1 | 2.19 s |
| /wp-cron.php?server_triggered_cronjob | 93 | 1.97 s |
| /wp-admin/plugins.php?action=activate&plugin=wp-rocket%2Fwp-rocket.php&plugin_status=all&paged=1&s&_wpnonce=70f7d2faca | 1 | 1.47 s |
| /wp-admin/plugins.php?activate=true&plugin_status=all&paged=1&s= | 2 | 1.37 s |
| /wp-admin/plugins.php?action=deactivate&plugin=wp-rocket%2Fwp-rocket.php&_wpnonce=729c628974 | 1 | 1.07 s |
| /wp-admin/plugins.php?action=activate&plugin=wp-rocket%2Fwp-rocket.php&_wpnonce=70f7d2faca | 1 | 0.91 s |
| /wp-cron.php?doing_wp_cron=1542348999.8110280036926269531250 | 1 | 0.67 s |
| /wp-admin/plugins.php?action=deactivate&plugin=wp-rocket%2Fwp-rocket.php&_wpnonce=113bcb7711 | 1 | 0.53 s |
| /wp-admin/plugins.php | 3 | 0.49 s |
| /wp-admin/plugin-install.php | 1 | 0.48 s |

El mejor Tiempo de Subida Máximo

El tiempo de subida es el tiempo total usado por NGINX (y servidores de subida) para procesar una petición y enviar una respuesta. El tiempo es medido en segundos, con milisegundos de resolución.

| PATH | REQUESTS | TIME |
|--|----------|----------|
| /wp-cron.php?server_triggered_cronjob | 93 | 180.00 s |
| /wp-admin/plugins.php?activate=true&plugin_status=all&paged=1&s= | 2 | 2.60 s |
| /wp-admin/update.php?action=upload-plugin | 1 | 2.19 s |
| /wp-admin/plugins.php?action=activate&plugin=wp-rocket%2Fwp-rocket.php&plugin_status=all&paged=1&s&_wpnonce=70f7d2faca | 1 | 1.47 s |
| /hello-world/ | 5 | 1.46 s |
| /wp-admin/plugins.php?action=deactivate&plugin=wp-rocket%2Fwp-rocket.php&_wpnonce=729c628974 | 1 | 1.07 s |
| / | 83 | 0.93 s |
| /wp-admin/plugins.php?action=activate&plugin=wp-rocket%2Fwp-rocket.php&_wpnonce=70f7d2faca | 1 | 0.91 s |
| /wp-admin/plugins.php | 3 | 0.85 s |
| /wp-cron.php?doing_wp_cron=1542348999.8110280036926269531250 | 1 | 0.67 s |

Su Sitio Podría Estar Hackeado

Si está teniendo problemas rastreando un problema de rendimiento, podría ser que su sitio esté siendo hackeado, siendo infectado con malware, o es objetivo de un ataque DDoS. Esto puede impactar en la velocidad de su sitio e incluso el tiempo de respuesta de su escritorio de administración de WordPress. En estos casos recomendamos lo siguiente:

1. Implemente un servidor proxy y WAF como Cloudflare o Sucuri
2. Bloquee las direcciones de IP malas utilizando los servicios de arriba o si es cliente de Kinsta, también puede bloquear direcciones de IP desde el escritorio de MyKinsta.

3. También puede implementar geo-bloqueo. Algunos países son muy malos cuando se trata de la calidad del tráfico que generan. Si usted está bajo ataque, podría necesitar bloquear un país entero, sea temporal o permanentemente.

Resolviendo Problemas con Códigos de Error (Códigos de Estado HTTP)

Los códigos de estado de HTTP son como breves notas del servidor de la red que es añadida sobre la parte superior de la página web. No es parte de la página. En su lugar, es un mensaje desde el servidor haciéndole saber cómo sucedieron las cosas cuando la petición para ver la página fue recibida por el servidor. ¡Estos pueden ser invaluable cuando se trata de resolver problemas!

Mientras que hay más de 40 códigos de estado distintos, a continuación están con los que vemos que los usuarios de WordPress comúnmente tienen problemas. Tenemos tutoriales disponibles en kinsta.com con instrucciones para arreglar cada uno de los siguientes códigos de error:

429: “Demasiadas Peticiones.” Generadas por el servidor cuando el usuario ha enviado demasiadas peticiones en una cierta cantidad de tiempo (limitando el rango). Esto puede ocurrir en algunas ocasiones si bots o scripts están intentando acceder a su sitio.

En este caso, podría querer intentar cambiar la URL de Inicio de sesión de su WordPress.

429 Too Many Requests

nginx

500: “Hubo un error en el servidor y la petición no pudo ser completada.”

Un código genérico que simplemente significa “error de servidor interno.” Algo salió mal en el servidor, y el recurso pedido no fue entregado. Este código normalmente es generado por plugins de otras compañías, PHP defectuoso, o incluso cuando se rompe la conexión a la base de datos.

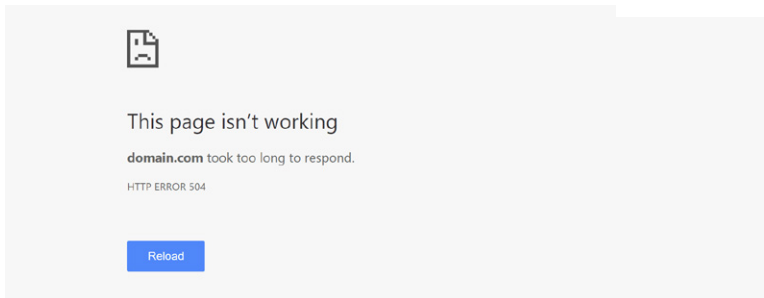
Error establishing a database connection

502: “Puerto de Enlace no Válido.” Este error de código normalmente significa que un servidor ha recibido una respuesta inválida de otro. Algunas veces un query o una petición tomará mucho tiempo, así que es cancelada o eliminada por el servidor y la conexión a la base de datos se rompe.



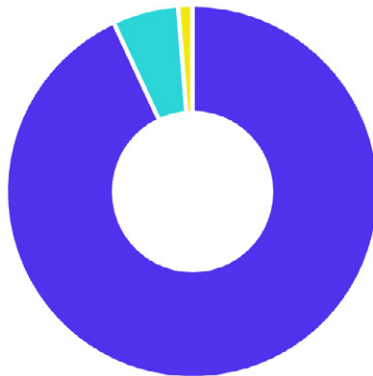
503: “El servidor no se encuentra disponible para hacerse cargo de esta petición por el momento.” La petición no puede ser completada ahora. Este código puede ser regresado por un servidor sobrecargado que no puede lidiar con peticiones adicionales.

504: “el servidor, actuando como una puerta de enlace, agotó el tiempo de espera para que el otro servidor responda” Recibimos este código cuando hay más de dos servidores involucrados en el proceso de una petición, y el primer servidor se queda sin tiempo de espera al esperar que responda el segundo servidor.



También puede adentrarse en estos códigos de respuesta de HTTP en nuestra herramienta de MyKinsta Analytics. Nuestro informe de desglose de los códigos de respuesta le permitirá ver una vista general de la distribución de los códigos de estado de HTTP servidos para los recursos pedidos.

Desglose de Códigos de Respuesta



Total
2,163,563

| | | |
|-----|-----------|---------|
| 200 | 2,012,973 | (93.0%) |
| 300 | 123,643 | (5.7%) |
| 400 | 26,442 | (1.2%) |
| 500 | 505 | (0.0%) |

Estadísticas de Respuesta

El informe de estadísticas de respuesta le permite ver el número total de redirecciones sucediendo en este momento, número de éxitos, y rango de error. Cada sitio de WordPress normalmente tiene un rango pequeño de error; esto es completamente normal.

Luego hay informes por cada tipo de código de error, como los errores 500, los errores 40 y las redirecciones, etc.

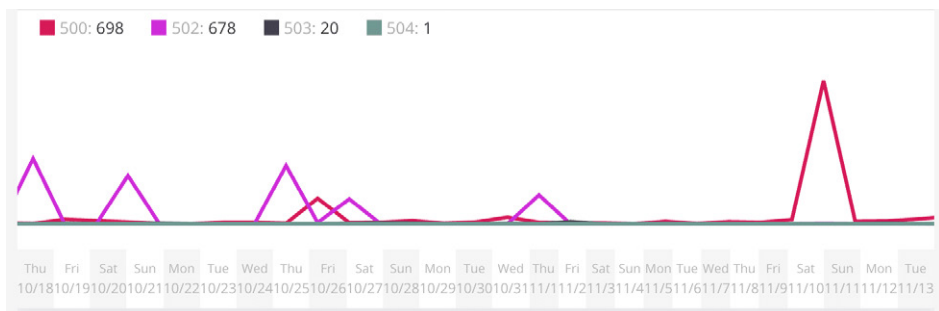
513,037
Redirects

119,796
Errors

99.3%
Success Rate

0.7%
Error Ratio

Desglose de Error 500



CAPÍTULO 14:

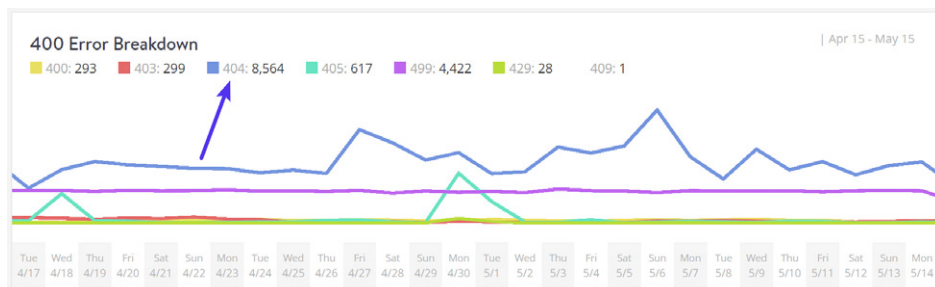
Recomendaciones sobre la Optimización del Back-End



Ahora nos adentraremos a las formas en las que puede acelerar WordPress al optimizar el back-end. El backend normalmente involucra a cualquier cosa que sea manejada completamente por el servidor, como PHP, HTTP, encabezados de caché, compresiones GZIP, etc.

Cree una Página 404 Ligera

Hemos visto de primera mano que los sitios altamente dinámicos normalmente generan muchos errores 404. ¿Su sitio web podría generar más de los que se imagina! Nuestra herramienta de analytics de MyKinsta puede ayudarle a determinar el número exacto (como se puede ver abajo).



La razón por la que estos errores son malos, es que muchas páginas 404 utilizan muchos recursos. Para sitios altamente dinámicos de WordPress, usted querrá evitar una página llena de 404. Cree una plantilla simple de 404 que evite hacer consulta a la base de datos lo más que se pueda. Y por su puesto, tómese el tiempo de arreglar los errores 404 ya que no sólo ocupan recursos, es simplemente una mala experiencia de usuario.

Incremente los PHP Workers

Los PHP workers podrían ser un término que jamás había escuchado, pero estos son de gran ayuda para muchos hosts, incluyendo a Kinsta, para poder controlar y limitar las peticiones (en lugar de limitarlo por CPU o RAM, lo que normalmente hacen los proveedores de hosting compartidos).

Los PHP workers determinan **cuántas peticiones simultáneas puede aguantar su sitio al mismo tiempo**. En pocas palabras, cada petición que no esté en el caché para su sitio web es controlado por un PHP worker. Por ejemplo, si tiene 4 peticiones que llegan a su sitio al mismo tiempo y su sitio tiene 2 PHP workers, dos de estas peticiones serán procesadas, mientras que las otras dos tendrán que esperar en la fila hasta que las primeras dos hayan terminado de procesar.

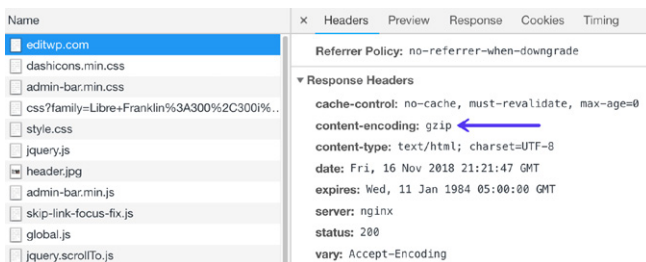
¿Recuerda cuando hablamos sobre uno de los mayores problemas de los sitios de membresías de WordPress con todas esas peticiones sin almacenar en el caché? Es por eso que los PHP workers son tan importantes, ya que ellos hacen el trabajo por cada petición. Por lo tanto, estos sitios normalmente requerirán PHP workers adicionales para asegurar que cada petición sea procesada sin retrasos y que sean completadas con éxito.

¿Qué pasa si continuamente pone hasta el tope a sus PHP workers? Básicamente, la fila empezará a empujar viejas peticiones las cuales podrían resultar en errores 500 en su sitio. Cada plan de hosting de Kinsta incluye un número predefinido de PHP workers. Si tiene problemas estimando lo que necesita su sitio, siempre podrá conversar con nuestro equipo de ventas o de soporte.

Utilice Compresión GZIP

GZIP es un formato de archivo y una aplicación de software utilizada para la compresión y descompresión de archivos. La compresión GZIP está habilitada a nivel del servidor, y permite una mayor reducción en el tamaño de su HTML, hojas de estilos, y archivos de JavaScript.

Cuando un navegador visita un sitio, este revisa si el servidor web tiene el GZIP habilitado revisando si existe el encabezado HTTP `content-encoding: gzip`. Si se detecta el encabezado, este sirve los archivos comprimidos y más pequeños. Si no es así, este sirve los archivos sin comprimir. Si no tiene habilitado el GZIP, probablemente verá advertencias y errores en las herramientas de prueba de velocidad como Google PageSpeed Insights y GTmetrix.



Habilitando la compresión GZIP puede ayudar a reducir el tamaño de su página web, el cual puede reducir significativamente la cantidad de tiempo para descargar un recurso, reducir el uso de datos para el cliente y mejorar el tiempo para la primera visualización de sus páginas. Esto es lo estándar en la mayoría de los proveedores de hosting, pero ya nada nos sorprende a estas alturas.

Habilitando Protección contra Hotlink

Este concepto de hotlinking es bastante simple. Usted encuentra una imagen en algún lugar de internet y utiliza la URL de la imagen directamente en su sitio. Esta imagen será mostrada en su sitio web, pero será servida desde la ubicación original. Esto es

muy conveniente para el que hace el hotlink, pero en realidad esto es robo, ya que está utilizando los recursos del sitio al que hizo hotlink. Es como si estuviéramos usando nuestro coche y conducir con la gasolina del coche que va a nuestro lado.

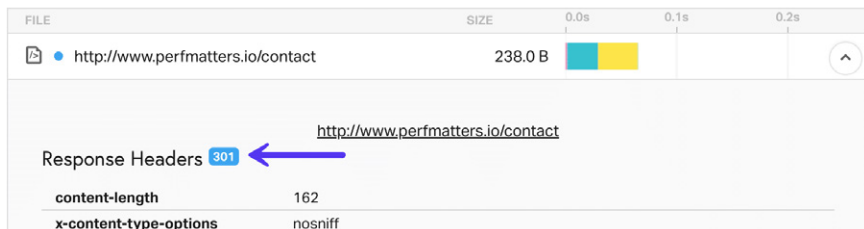
Hacer hotlinking es como conducir un coche utilizando la gasolina que le robó al coche del vecino.

Hotlinking puede **ser una gran fuga de recursos para el servidor objetivo**. Imagínese si usted está en un host compartido de WordPress y Huffington Post de repente pone un enlace a sus imágenes. Usted podría pasar de cientos de consultas por hora en su sitio a varios cientos de miles. Esto podría resultar en una suspensión de su cuenta de hosting. Esta es solo una razón para usar hosts de alto desempeño (los cuales puedan lidiar con dificultades como esta), pero también permitir protección de hotlink, para que esto no suceda.


Minimice las Redirecciones y Agréguelas a Nivel del Servidor

Uno siempre debe tener cuidado cuando tiene demasiadas redirecciones. Simples redirecciones como una simple redirección 301 HTTP a HTTPS, o www a no www (y viceversa) están bien. Y en muchas ocasiones estas son necesarias en ciertas áreas de su sitio web. Sin embargo, cada una tiene un costo en el rendimiento de su sitio. Y si empieza dirigiendo una sobre otra, es importante darse cuenta cómo impactarán a su sitio. Esto se aplica a páginas y direcciones de publicaciones, redirecciones de imagen y todo.

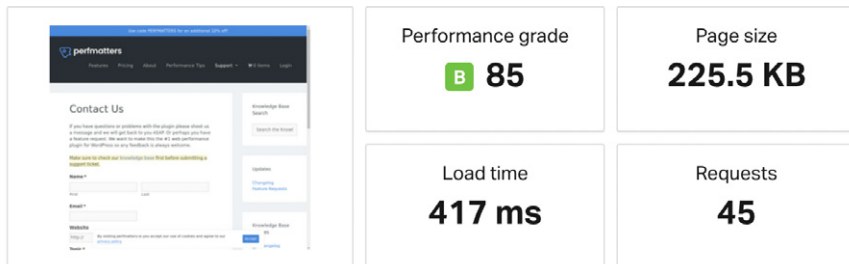
Una redirección generará una respuesta 301 o 302 en el estado de respuesta del encabezado.



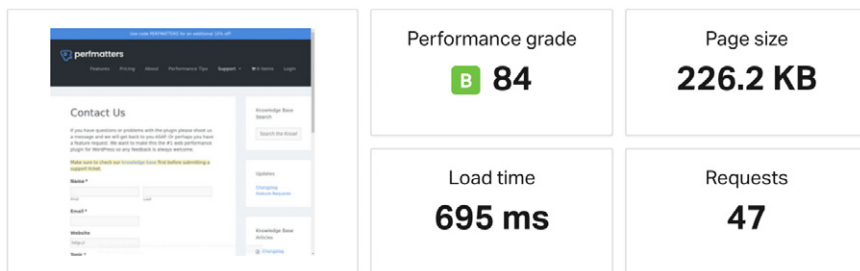
The screenshot shows a browser's developer tools interface. At the top, the address bar displays 'http://www.perfmatters.io/contact' with a size of '238.0 B'. Below the address bar, the 'Response Headers' section is expanded, showing a '301' status code with a blue arrow pointing to it. The headers listed are 'content-length: 162' and 'x-content-type-options: nosniff'. The URL 'http://www.perfmatters.io/contact' is also visible above the headers.

| FILE | SIZE | 0.0s | 0.1s | 0.2s |
|---|---------|--|------|------|
|  http://www.perfmatters.io/contact | 238.0 B | <div style="width: 100%; height: 10px; background-color: #00aaff; background-image: linear-gradient(to right, #00aaff, #ffff00);"></div> | | |
| Response Headers 301 ← | | | | |
| content-length | 162 | | | |
| x-content-type-options | nosniff | | | |

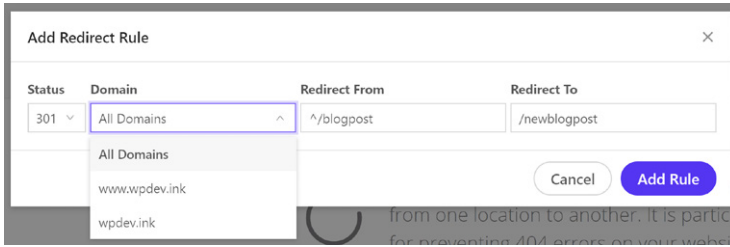
¿Cuánto impactan estas redirecciones a su sitio? Hagamos una pequeña prueba. Primero, hagamos una prueba de velocidad en nuestra página de contacto: <https://perfmatters.io/contact/>. Como puede ver abajo, obtuvimos un tiempo total de carga de **417 ms**.



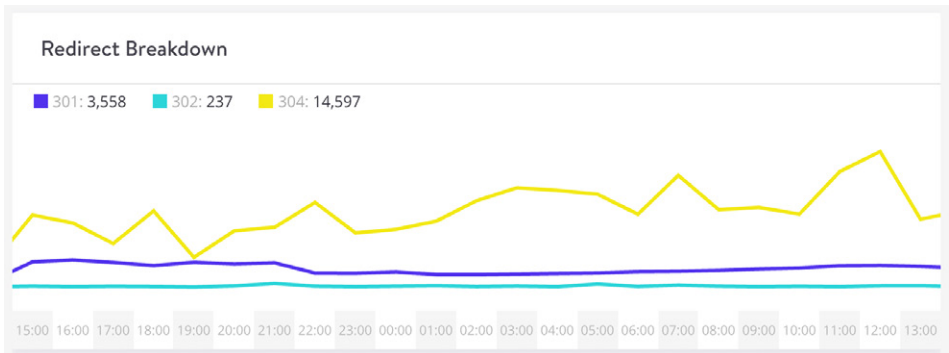
Luego, modificamos un poco la URL y hacemos otra prueba de velocidad para ver el impacto de múltiples redirecciones. <http://www.perfmatters.io/contact>. Como puede ver, la misma página ahora toma **695 ms** para cargar. Eso es un incremento de 66%. ¡Wow!



El utilizar plugins gratuitos de WordPress para implementar redirecciones en algunas ocasiones puede causar problemas de desempeño ya que la mayoría utilizan la función `wp_redirect`, la cual requiere ejecución de códigos y recursos adicionales. Algunos de estos agregan datos autocargados a su tabla de `wp_options`, la que incrementa la carga de la base de datos. Agregarlo a nivel del servidor es donde se deberían hacer. Nosotros le permitimos hacer eso en MyKinsta, con nuestra herramienta de reglas de redirección.



También puede ver un desglose completo de cuantas redirecciones están pasando en sus sitios en nuestra herramienta de MyKinsta analytics. Vea el número total de 301, 302 y 304.



No Permita que los Cron Jobs se Salgan de Control

Los CRON jobs (WP-CRON) son usados para agendar tareas repetitivas para su sitio de WordPress. Sin embargo, con el tiempo, estos se pueden salir de control y causar problemas de rendimiento. Puede utilizar el plugin gratuito de WP Croncontrol para tener el control de todos los Cron jobs que se están ejecutando en su sitio.

También hemos visto problemas de rendimiento con el controlador de Cron de WordPress: WP-Cron. Si un sitio no tiene suficientes PHP workers, en algunas ocasiones una petición entrará, WordPress creará el cron, pero el cron tiene que esperar por el worker, y seguirá esperando, Un enfoque mejor es deshabilitar WP-Cron y usar el cron del sistema en su lugar. Esto incluso se recomienda en la guía oficial del Plugin.

Para deshabilitar el WP-CRON, agregue el siguiente código a su archivo `wp-config.php`, justo antes de la línea que dice “That’s all, stop editing! Happy blogging.” Nota: Esto deshabilita que se active durante la carga de la página, no cuando lo llama directamente a través de `wp-cron.php`.

```
define('DISABLE_WP_CRON', true);
```

```
60  /**
61  * WordPress Database Table prefix.
62  *
63  * You can have multiple installations in one database if you give each
64  * a unique prefix. Only numbers, letters, and underscores please!
65  */
66  $table_prefix = 'wp_';
67
68  define('DISABLE_WP_CRON', true); ←
69
70  /* That's all, stop editing! Happy blogging. */
71
72  /** Absolute path to the WordPress directory. */
73  if ( ! defined( 'ABSPATH' ) )
74      define( 'ABSPATH', dirname( __FILE__ ) . '/' );
75
76  /** Sets up WordPress vars and included files. */
77  require_once ABSPATH . 'wp-settings.php';
78
```

Line 68, Column 33

Necesitará lanzar `wp-cron.php` de su servidor. Si es cliente de Kinsta, los sistemas crons ya estarán habilitados y se activarán cada 15 minutos por defecto. También puede incrementar la frecuencia al ponerse en contacto con el equipo de soporte. Si está familiarizado con SSH, usted podrá administrar los crons del servidor desde la línea de comandos.

Agregue Control de Caché y Encabezados que Expiran (Determine la Duración del Caché)

Cada script en su sitio de WordPress necesita tener un encabezado de caché HTTP adjunto a este (o por lo menos debería). Esto **determina cuando expira el caché en el archivo**. Para arreglar esto, asegúrese de que el host de WordPress tenga la configuración de encabezados `cache-control` y `expires` apropiada. Si no es así, probablemente verá advertencias sobre necesitar agregar encabezados de expiración o hacer leverage browser caching en las herramientas de prueba de velocidad.

Mientras que el encabezado de `cache-control` activa el caché desde el lado del cliente y establece la edad máxima de un recurso, el encabezado `expires` es usado para especificar un punto específico en el tiempo en que el recurso ya no es válido. Mientras que ambos encabezados pueden ser usados al mismo tiempo, no tiene que agregar necesariamente ambos. El `Cache-control` es el más reciente y usualmente el método más recomendado.

Kinsta automáticamente agrega encabezados de caché HTTP en todas las peticiones del servidor, y si está utilizando una CDN, probablemente agregaran estos encabezados por usted.

| Name | Headers | Preview | Response | Timing |
|---|---------|---------|----------|--------|
| editwp.com | | | | |
| admin-bar.min.js?ver=5.0-beta3-43867 | | | | |
| skip-link-focus-fix.js?ver=20151215 | | | | |
| wp-embed.min.js?ver=5.0-beta3-43867 | | | | |
| be9a4e9519964b933dea6723d480c95a?s=52&d=mm&r=g | | | | |
| be9a4e9519964b933dea6723d480c95a?s=128&d=mm&r=g | | | | |
| wp-emoji-release.min.js?ver=5.0-beta3-43867 | | | | |
| data:application/fo... | | | | |
| url-metrics | | | | |
| tpc-check.html | | | | |
| postmessage.js | | | | |
| tpc-check-add-cookie.js | | | | |

12 requests | 6.5 KB transferred | Finish: 1.19 s | DOMContentLoaded: 452 ms | ...

▼ Response Headers

- access-control-allow-origin: *
- cache-control: max-age=315360000 ←
- content-encoding: gzip
- content-type: application/javascript; charset=UTF-8
- date: Wed, 07 Nov 2018 03:17:34 GMT
- etag: W/"5bdc7194-2efa"
- expires: Thu, 31 Dec 2037 23:55:55 GMT ←
- last-modified: Fri, 02 Nov 2018 15:47:32 GMT
- server: nginx
- status: 200
- vary: Accept-Encoding
- x-content-type-options: nosniff

Si a su servidor le faltan estos encabezados, puede agregarlos manualmente.

Agregando Encabezado de Cache-Control en Nginx

Puede agregar encabezados de `cache-control` en Nginx al agregar el siguiente código o bloque de configuración en su servidor.

```
location ~* \.(js|css|png|jpg|jpeg|gif|svg|ico)$ {
    expires 30d;
    add_header Cache-Control "public, no-transform";
}
```


Agregando Encabezado Expires en Nginx

Puede agregar encabezados `expires` en Nginx al agregar el siguiente código en su servidor. En este ejemplo, usted podrá ver cómo especificar diferentes tiempos de expiración basados en los tipos de archivos.

```
location ~* \.(jpg|jpeg|gif|png|svg)$ {
    expires 365d;
}

location ~* \.(pdf|css|html|js|swf)$ {
    expires 2d;
}
```

Agregando Encabezado Cache-Control en Apache

Usted puede agregar encabezados `cache-control` en Apache al agregar lo siguiente a su archivo de `.htaccess`. Los fragmentos de código pueden ser agregados en la parte superior o inferior del archivo (antes de `#BEGIN WordPress` o después de `#END WordPress`).

```
<filesMatch ".(ico|pdf|flv|jpg|jpeg|png|gif|svg|js|css|swf)$">
Header set Cache-Control "max-age=84600, public"
</filesMatch>
```

Agregando Encabezados Expires en Apache

Puede agregar encabezados `expires` en Apache al agregar el siguiente archivo de `.htaccess`.

```
## EXPIRES HEADER CACHING ##
<IfModule mod_expires.c>
ExpiresActive On
ExpiresByType image/jpg "access 1 year"
```

```
ExpiresByType image/jpeg "access 1 year"
ExpiresByType image/gif "access 1 year"
ExpiresByType image/png "access 1 year"
ExpiresByType image/svg "access 1 year"
ExpiresByType text/css "access 1 month"
ExpiresByType application/pdf "access 1 month"
ExpiresByType application/javascript "access 1 month"
ExpiresByType application/x-javascript "access 1 month"
ExpiresByType application/x-shockwave-flash "access 1 month"
ExpiresByType image/x-icon "access 1 year"
ExpiresDefault "access 2 days"
</IfModule>

## EXPIRES HEADER CACHING ##
```

También es importante tener en cuenta que usted sólo puede agregar encabezados de caché de HTTP en su servidor. Si usted está recibiendo advertencias sobre eso, quizás usted necesite hacer leverage browser caching en una petición externa, no hay nada que pueda hacer, ya que usted no tiene una petición sobre su servidor. Los culpables más comunes incluyen el script de Google Analytics y píxeles de marketing, como Facebook y Twitter.

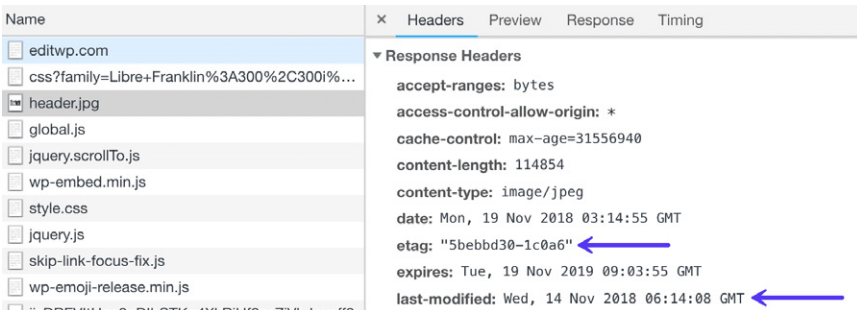
Si está intentando arreglar esto con el script de Google Analytics, usted puede hospedarlo localmente o en su CDN (aunque esto no es oficialmente soportado) con un plugin como Perfmatters (disponible en perfmatters.io) o WP Rocket (disponible en wp-rocket.me).

Agregue Encabezados de Último Modificado (Last-Modified) y ETag (Validando Caché)

Tenemos dos sets de encabezados, `last-modified` y `etag`.

Mientras que los encabezados de `cache-control` y `expires` ayudan a que el navegador determine si el archivo ha cambiado desde la última vez que fue requerido. (validan

el caché). Los encabezados de `last-modified` y `etag` validan y establecen la longitud del caché y deben incluirse en cada respuesta del servidor de origen. Si estos no son establecidos apropiadamente usted podría ver una advertencia diciendo «Especifique un validador de caché.»



Si los encabezados no son encontrados, este generará una nueva petición para el recurso todo el tiempo, lo que incrementa la carga en su servidor. Al utilizar encabezados de caché, asegurará que las peticiones posteriores no tengan que ser cargadas desde el servidor, así ahorrando ancho de banda y mejorando el desempeño para el usuario.

Kinsta automáticamente agrega los encabezados anteriores en todas las peticiones del servidor, y si está utilizando una CDN, probablemente también terminen agregando estos encabezados por usted. Al igual que con el `cache-control` y `expires`, y usted no podrá establecer manualmente estos encabezados HTTP en los recursos externos.

Encabezado Last-Modified

El encabezado `last-modified` es generalmente enviado de forma automática desde el servidor. Este es un encabezado **que generalmente no tendrá que agregar manualmente**. Este es enviado para ver si el archivo en el caché de su navegador ha sido modificado desde la última vez que fue pedido. Usted puede ver la petición del encabezado en Pingdom o utilizar Chrome DevTools para ver el valor del último encabezado `last-modified`.

Encabezado ETag

El encabezado **ETag** es similar al encabezado `last-modified`. También es usado para validar el caché de un archivo. Si usted está usando Apache 2.4 o mayor, el encabezado ETAG es agregado automáticamente, usando la directiva FileETag. En el caso de NGINX, el encabezado ETag ha sido habilitado por defecto desde el 2016.

Puede habilitar el encabezado ETag de forma manual en NGinx utilizando el siguiente código.

```
etag on
```

Agregando Encabezado Vary: Accept-Encoding

El encabezado `vary: Accept-Encoding` debería ser incluido en cada respuesta de servidor origen, ya que le dice al navegador si el cliente puede manejar o no las versiones comprimidas del contenido. Si no es establecido correctamente, podría ver una advertencia diciendo que usted necesita «Especificar un Encabezado Vary: Accept-Encoding.»

Por ejemplo, digamos que tiene un viejo navegador sin compresión GZIP y un navegador moderno con este. Si no utiliza el encabezado `vary: Accept-Encoding` su servidor web o CDN podría almacenar en el caché la versión sin compresión y entregarlo a la versión moderna del navegador por accidente, lo cual causaría daños al desempeño de su sitio de WordPress. Al utilizar el encabezado puede asegurar que su servidor web y/o CDN entregue la versión apropiada.

| Name | × | Headers | Preview | Response | Timing |
|--|---|---------|---------|----------|--------|
| editwp.com | | | | | |
| css?family=Libre+Franklin%3A300%2C300i%... | | | | | |
| header.jpg | | | | | |
| global.js | | | | | |
| jquery.scrollTo.js | | | | | |
| wp-embed.min.js | | | | | |
| style.css | | | | | |
| jquery.js | | | | | |
| skip-link-focus-fix.js | | | | | |

| ▼ Response Headers | |
|--------------------------------|-------------------------------|
| content-encoding: | gzip |
| content-type: | text/html; charset=UTF-8 |
| date: | Mon, 19 Nov 2018 03:14:57 GMT |
| server: | nginx |
| status: | 200 |
| vary: | Accept-Encoding ← |
| x-content-type-options: | nosniff |
| x-kinsta-cache: | HIT |

Kinsta automáticamente agrega los encabezados anteriores en todas las peticiones de servidor, y si usted está utilizando una CDN, probablemente terminen agregando estos encabezados por usted. Como con los otros encabezados de caché de los que hemos hablado anteriormente, usted no puede establecer manualmente este encabezado en recursos externos.

Agregar Encabezado Vary: Accept-Encoding en Apache

Puede agregar el encabezado vary: Accept-Encoding en Apache al agregar lo siguiente a su archivo `.htaccess`.

```
<IfModule mod_headers.c>
  <FilesMatch "(.js|css|xml|gz|html)$">
    Header append Vary: Accept-Encoding
  </FilesMatch>
</IfModule>
```

Agregar Encabezado Vary: Accept-Encoding en Nginx

Puede agregar el encabezado vary: Accept-Encoding en Nginx al agregar el siguiente código a su archivo de configuración. Todos los archivos de configuración Nginx están ubicados en el directorio `/etc/nginx/`. El archivo principal de configuración es `/etc/nginx/nginx.conf`.

```
gzip_vary on
```

Cambiando el límite de Memoria de WordPress en wp-config.php

Como se declara en el Codex de WordPress, con la Versión 2.5 de WordPress, la opción `WP_MEMORY_LIMIT` le permite especificar el número máximo de memoria que puede ser consumido por PHP. Esta opción podría ser necesaria si recibe un mensaje como el siguiente “tamaño permitido de memoria de xxxxxx bytes se ha agotado.”

Por defecto, WordPress intentará incrementar la memoria asignada a PHP a 40 MB para un solo sitio y 64MB para un multisitio. Ellos definen los límites de memoria en el archivo `./wp-includes/default-constants.php`, en las líneas 32-44.

También puede tener PHP `memory_limit` en el servidor por su proveedor de hosting. Estas son dos cosas distintas. En Kinsta nosotros establecemos `memory_limit` por defecto a 256M. Si usted se está encontrando con el error de “se ha agotado el tamaño de la memoria” puede intentar incrementar el límite de memoria de PHP en WordPress.

Agregue lo siguiente a su archivo `wp-config.php` file, antes de la línea que dice “That’s all, stop editing! Happy blogging.”

```
define( 'WP_MEMORY_LIMIT', '256M' );
```

Jan Reilink también tiene un excelente artículo que describe el problema del límite de memoria de WordPress detalladamente. También da una variante del código que podría utilizar. En lugar de establecer la cantidad manualmente, usted puede establecerla al valor de `memory_limit` de PHP.

```
define( 'WP_MEMORY_LIMIT', ini_get( 'memory_limit' ) );
```

CAPÍTULO 15:

Consejos sobre la Optimización del Front-End y los Servicios Externos



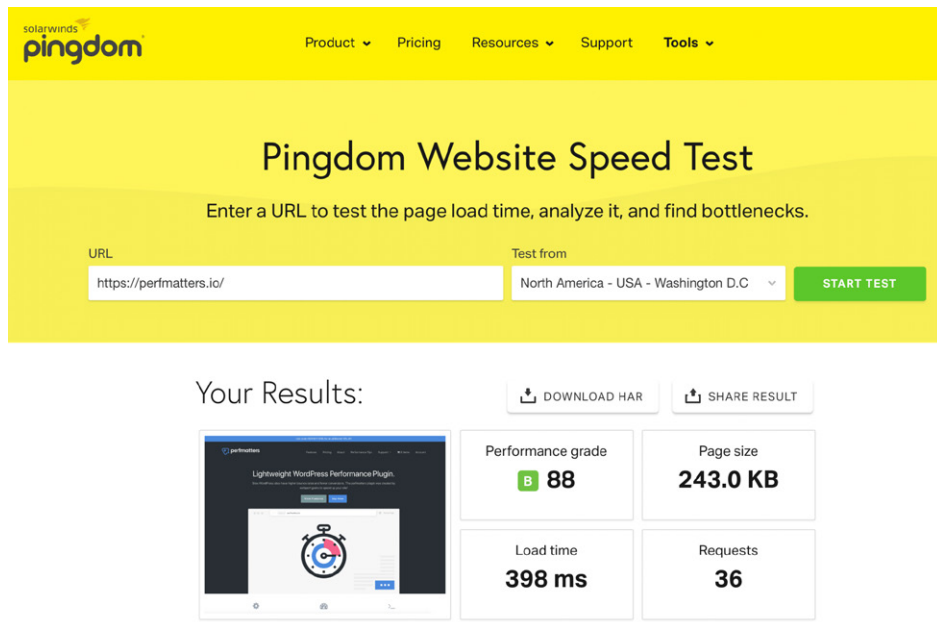
Ahora hablaremos de algunas formas para acelerar WordPress optimizando el front-end. El front-end normalmente involucra cualquier cosa que ha sido gestionada completamente por el navegador del lado del cliente, como CSS, JavaScript, Imágenes, etc. Esto también incluye analizar los servicios externos que están cargando en su sitio y cómo estos impactan en el tiempo de carga total.

Dos de los objetivos más importantes que debería tener cuando se trata de optimizaciones de front-end son:

- **Reducir el tamaño general de la página web.** El tamaño de su CSS, JavaScript, imágenes es muy importante. Un sitio web de 4 MB normalmente cargará mucho más lento que un sitio web de 1 MB. Sin embargo, Paul Calvano tiene un excelente artículo sobre el impacto del peso de la página, cuando se trata del tiempo de carga, y cómo esto es importante para asegurarse de que no sea la única cosa que usted está revisando, ya que en algunas ocasiones esto podría ser confuso.
- **Reduciendo peticiones HTTP y servicios externos.** Con HTTP/2, se pueden enviar múltiples peticiones y respuestas al mismo tiempo, utilizando una sola conexión TCP. Aunque esto es sorprendente, tratándose del rendimiento, reducir las peticiones HTTP puede ayudarle a acelerar su sitio de WordPress. Esto también incluye reducir el número total de peticiones externas.. Cada una de estas peticiones agregan retrasos adicionales, como búsquedas de DNS, conexiones TLS, y latencia de red.

Haga Pruebas de Velocidad a Su Sitio de WordPress para Obtener una Referencia

Cuando se trata de optimizar el front-end de su sitio, siempre es bueno empezar con una referencia. Esto significa hacer una prueba de velocidad. Hay multitud de formas en las que puede hacerlo. Tenemos guías en profundidad de cómo usar apropiadamente Pingdom y GTmetrix disponibles en nuestro sitio kinsta.com.



The screenshot shows the Pingdom Website Speed Test interface. At the top, there is a navigation bar with the Pingdom logo and links for Product, Pricing, Resources, Support, and Tools. The main heading is "Pingdom Website Speed Test" with the instruction "Enter a URL to test the page load time, analyze it, and find bottlenecks." Below this, there is a form with a URL input field containing "https://perfmatters.io/" and a "Test from" dropdown menu set to "North America - USA - Washington D.C.". A green "START TEST" button is to the right. Below the form, the results are displayed under the heading "Your Results:". There are two buttons: "DOWNLOAD HAR" and "SHARE RESULT". The results are shown in a grid of four boxes: Performance grade (B 88), Page size (243.0 KB), Load time (398 ms), and Requests (36). A small thumbnail image of the website being tested is also visible.

| Your Results: | |
|-------------------|-----------------|
| Performance grade | Page size |
| B 88 | 243.0 KB |
| Load time | Requests |
| 398 ms | 36 |

Aquí hay un par de cosas a tener en cuenta cuando haga pruebas:

1. Elija Uno y Siga con Ese

Somos fanáticos de Pingdom, GTmetrix, WebPageTest, PageSpeed Insights y Chrome DevTools. Sin embargo, no importa tanto qué herramienta de prueba de velocidad elija, lo más importante es que sea consistente. Todas tienen formas distintas de medir y cuantificar la velocidad, así que elija una herramienta y siga usando esa para todas sus pruebas y optimizaciones. Incluso Google le dice que elija una.

La herramienta de prueba de velocidad que elija no es muy importante, lo importante es que cuando elija una, la siga usando para todas sus pruebas.

2. No Se Obsesione Tratando de Obtener una Calificación Perfecta

Muchas de las herramientas como Google PageSpeed Insights tienen un tipo de calificación de velocidad y desempeño. Es importante recordar que la calificación no siempre importa tanto, ya que la velocidad de su sitio web y el rendimiento percibido por el usuario siempre será distinto. La calificación existe para ayudarle a medir qué tan bien le está yendo. Pero obsesionarse por conseguir esa perfecta calificación de 100/100 o una A, en algunos casos podría ser una pérdida de tiempo. Los sitios más grandes, con muchos scripts externos y anuncios jamás obtendrán una calificación perfecta, y esto es normal.

3. La Ubicación de Sus Pruebas Importan

La ubicación que elija cuando haga pruebas de velocidad importa bastante. Como hemos mencionado en secciones anteriores, la razón es que, todo esto es relativo a la ubicación del centro de datos que elija. TTFB, latencia de la red, todo entra en juego. Así que haga pruebas desde una ubicación que se encuentre cercana a su centro de datos y otras pruebas desde una ubicación lejana. Esto también le ayudará a ver el impacto que tiene la CDN en su sitio de WordPress

4. Haga Varias Pruebas por el Caché

Como ya hemos hablado anteriormente en la sección sobre el caché, si el caché ha sido depurado recientemente o ha expirado en su host de WordPress o CDN, este registrará un “MISS” en el encabezado de HTTP. Esto quiere decir que su sitio web o recursos no están siendo servidos desde el caché.

| FILE | SIZE | 0.0s | 0.1s | 0.2s |
|--|---|------|------|------|
| https://perfmatters.io/ | 9.6 KB | | | |
| https://perfmatters.io/ | | | | |
| Response Headers 200 | | | | |
| status | 200 | | | |
| x-content-type-options | nosniff | | | |
| content-encoding | gzip | | | |
| set-cookie | PHPSESSID=d33c7f3bfa8f6c0a9ba55f3a6804e1... | | | |
| x-kinsta-cache | MISS ← | | | |
| expires | Thu, 19 Nov 1981 08:52:00 GMT | | | |

Para poder ver de forma apropiada la velocidad de su sitio entero, necesita ver que todo cargue desde el caché, es decir que su página inicial y que todos sus activos registren un “HIT.” Esto en algunas ocasiones requiere que haga pruebas de velocidad adicionales. Luego podrá sacar el promedio.

| FILE | SIZE | 0.0s | 0.1s | 0.2s |
|--|---|------|------|------|
| https://perfmatters.io/ | 9.6 KB | | | |
| https://perfmatters.io/ | | | | |
| Response Headers 200 | | | | |
| status | 200 | | | |
| x-content-type-options | nosniff | | | |
| content-encoding | gzip | | | |
| set-cookie | PHPSESSID=d33c7f3bfa8f6c0a9ba55f3a6804e1... | | | |
| x-kinsta-cache | HIT ← | | | |
| expires | Thu, 19 Nov 1981 08:52:00 GMT | | | |

Ahora hablaremos de algunas optimizaciones de front-end que podrá hacer en su sitio de WordPress.

Eliminando Query Strings

Una advertencia o recomendación muy común que se ve en las herramientas de pruebas de velocidad, es que uno remueva las cadenas de consulta (query strings). ¿De qué se trata todo esto? Bueno, básicamente, es que sus archivos de CSS

y JavaScript, normalmente tienen la versión del archivo en sus URLs, como `https://domain.com/file.min.css?ver=4.5.3`. Algunos servidores y servidores proxy no podrán cachear la query strings. Así que, al eliminarlos, en algunas ocasiones podrá mejorar su caché.










Puede agregar manualmente el siguiente código al archivo `functions.php` de su tema

```
function remove_query_strings() {  
    if(!is_admin()) {  
        add_filter('script_loader_src', 'remove_query_strings_split', 15);  
        add_filter('style_loader_src', 'remove_query_strings_split', 15);  
    }  
}  
  
function remove_query_strings_split($src){  
    $output = preg_split("/(&ver|\?ver)/", $src);  
    return $output[0];  
}  
  
add_action('init', 'remove_query_strings');
```










O puede usar un plugin premium como Perfmatters que tiene una opción fácil de un solo clic para eliminar las cadenas de consulta.

Con Query Strings

Aquí hay un ejemplo de scripts cargando con query strings.

| Name | Status |
|---|--------|
|  editwp.com | 200 |
|  css?family=Libre+Franklin%3A300%2C300i%2C400%2C400...C600%2C600i... | 200 |
|  style.css?ver=4.9.8 | 200 |
|  jquery.js?ver=1.12.4 | 200 |
|  header.jpg | 200 |
|  skip-link-focus-fix.js?ver=1.0 | 200 |
|  global.js?ver=1.0 | 200 |
|  jquery.scrollTo.js?ver=2.1.2 | 200 |
|  jizDREVIthGc8qDLbSTKq4XkRiUf2zcZiVbJ.woff2 | 200 |

Sin Query Strings (Después del código)

| Name | Status |
|---|--------|
|  editwp.com | 200 |
|  css?family=Libre+Franklin%3A300%2C300i%2C400%2C400...C600%2C600i... | 200 |
|  style.css ← | 200 |
|  jquery.js | 200 |
|  header.jpg | 200 |
|  skip-link-focus-fix.js | 200 |
|  global.js | 200 |
|  jquery.scrollTo.js | 200 |
|  jizDREVIthGc8qDIbSTKq4XkRiUf2zcZiVbJ.woff2 | 200 |

Aquí hay un ejemplo de scripts después de haber eliminado los query strings.

Sin embargo, antes de que vaya y elimine query strings en su sitio, es importante saber la razón por la que los query strings son usados. Las versiones en los archivos son normalmente utilizadas por los desarrolladores de WordPress para resolver problemas de caché.

Por ejemplo, si un desarrollador de plugin saca una actualización y cambia `style.css` de `?ver=4.6` a `?ver=4.7`, este será tratado como una URL completamente nueva y no podrá estar en el caché. Si usted elimina los query strings y actualiza un plugin, esto podría causar que la versión en el caché siga sirviendo. En algunos casos, esto podría romper la apariencia de su sitio hasta que el recurso en el caché expire o el caché haya sido depurado por completo.

También, algunos CDNs pueden almacenar query strings en el caché. La Kinsta CDN puede y lo hará por defecto. Así que, si usted es cliente de Kinsta, los query strings ya se encuentran en el caché de sus activos.

Elimine Render-Blocking JavaScript y CSS

Podría aparecer una advertencia sobre render-blocking JavaScript y CSS cuando tiene archivos previniendo que la página cargue lo más rápido posible. A veces JS y CSS específicos son condicionales, es decir, estos no requieren mostrar el contenido above-the-fold. Usted puede prevenir que se conviertan en bloqueadores de visualización al utilizar atributos `async` y `defer`.

| Opportunity | Estimated Savings |
|---------------------------------------|--|
| 1 Eliminate render-blocking resources |  2.24 s ^ |

Resources are blocking the first paint of your page. Consider delivering critical JS/CSS inline and deferring all non-critical JS/styles. [Learn more](#).

Para eliminar render-blocking JavaScript y CSS usted necesita hacer lo siguiente:

Limpie el JS del Camino Crítico de Visualización

Mover JavaScript del camino crítico de visualización se hace al agregar el atributo `defer` o `async` a los elementos HTML `script` que llaman a los recursos de JavaScript.

- El **atributo `async`** le dice al servidor que empiece a descargar el recurso inmediatamente, sin hacer más lenta la fragmentación del HTML. Una vez que esté disponible el recurso, la fragmentación HTML será pausada para que el recurso pueda ser cargado.
- El **atributo `defer`** le dice al navegador que deje de descargar el recurso por un momento hasta que la fragmentación HTML se haya completado. Una vez que el servidor haya terminado con HTML, luego descargará y visualizará a todos los `scripts` diferidos en el orden en que aparecieron en el documento.

Optimice la Entrega de Recursos CSS

Dicho de forma sencilla, optimizar la entrega de CSS quiere decir que usted necesita descubrir cómo hacer que se bloquee la visualización.

- Identifique los estilos que son requeridos para visualizar a contenido `above-the-fold` y entregar esos estilos en línea con HTML.
- Utilizar CSS condicionalmente en dispositivos, sólo cuando es necesario.
- Cargar el CSS restante de forma asíncrona.

Hacer todo esto puede llegar a ser complejo y sin duda alguna requiere de algunos arreglos basados en los scripts que tenga cargando en su sitio. Aquí tenemos algunos plugins de WordPress que podrían ser de ayuda:

- Autoptimize (Gratis, disponible en WordPress.org)
- Async JavaScript (Gratis, disponible en WordPress.org)
- Hummingbird (Gratis, disponible en WordPress.org)

Para ver una explicación más detallada y una guía, le recomendamos revisar nuestro artículo sobre cómo eliminar JavaScript y CSS bloqueadores de visualización.







Combine CSS y JavaScript Externos en WordPress

La advertencia de combinar CSS externos es común verla al usar una CDN porque usted está alojando sus archivos CSS en un dominio externo, como `cdn.domain.com`. En el pasado, una forma rápida de arreglarlo era concatenar sus archivos CSS, o combinarlos para que carguen en una sola petición.

Sin embargo, si está utilizando HTTP con un proveedor que soporta HTTP/2, esta advertencia ya no es tan relevante como solía ser. Con HTTP/2 múltiples archivos CSS pueden ser cargados en paralelo sobre una sola conexión. Y más del 86% de navegadores soportan HTTP/2.

Pero esto no quiere decir que esta optimización esté completamente muerta. En algunas ocasiones, hemos visto que esto acelera los sitios de WordPress. Depende del tamaño de los archivos y cuántos de estos hay. Así que, esta es una optimización que recomendamos que pruebe en su sitio.

Una de las formas más fáciles de combinar sus archivos de CSS y JavaScript externos es el plugin gratuito Autoptimize. Después de combinarlos, usted verá un archivo “`autoptimize_xxxxx.css`” o “`autoptimize_xxxxx.js`” También soporta la carga desde su CDN. También puede hacer eso con su plugin de WP Rocket. (Disponible en wp-rocket.me).

| Name | Status | Domain |
|--|--------|--------------------|
|  editwp.com | 200 | editwp.com |
|  autooptimize_23d90d8a1eaacef4363d9e02c50ae363.css ← | 200 | mk0editwpn6... |
|  css?family=Libre+Franklin%3A300%2C300i%2C400%2C400...C600... | 200 | fonts.googlea... |
|  jquery.js?ver=1.12.4 | 200 | mk0editwpn6... |
|  header.jpg | 200 | mk0editwpn6... |
|  autooptimize_3fde0f2c0341cdc5afd7cc0c0e403f92.js ← | 200 | mk0editwpn6... |
| <input type="checkbox"/> jizDREVIthGc8qDlbSTKq4XkRiUf2zcZiVbJ.woff2 | 200 | fonts.gstatic.c... |
| <input type="checkbox"/> jizAREVIthGc8qDlbSTKq4XkRi24_Sl0q1vjitOh.woff2 | 200 | fonts.gstatic.c... |
| <input type="checkbox"/> jizAREVIthGc8qDlbSTKq4XkRi20-Sl0q1vjitOh.woff2 | 200 | fonts.gstatic.c... |

Utilice Minificación en HTML, CSS y JavaScript

Podemos reducir la cantidad de datos que el navegador tiene que descargar al minificar recursos HTML, CSS y de JavaScript. La minificación es el proceso de remover caracteres innecesarios como los comentarios y espaciones en blanco del código fuente. Estos caracteres son extremadamente útiles en el desarrollo, pero son inútiles para el navegador para visualizar la página.

HTML sin Minificar

Aquí le dejamos un ejemplo de código HTML sin minificar.

```

1 <!DOCTYPE html>
2 <html lang="en-US" class="no-js no-svg">
3 <head>
4 <meta charset="UTF-8">
5 <meta name="viewport" content="width=device-width, initial-scale=1">
6 <link rel="profile" href="http://gmpg.org/xfn/11">
7
8 <script>(function(html){html.className = html.className.replace(/\bno-
9 js\b/, 'js')})(document.documentElement);</script>
10 <title>editwp &#8211; Just another WordPress site</title>
<link rel='dns-prefetch' href='//fonts.googleapis.com' />
```


HTML Minificado

Aquí le dejamos un ejemplo de código HTML después de minificación

```
1 <!DOCTYPE html><html lang="en-US" class="no-js no-svg"><head><meta
  charset="UTF-8"><meta name="viewport" content="width=device-width,
  initial-scale=1"><link rel="profile" href="http://gmpg.org/xfn/11">
  <script>(function(html){html.className = html.className.replace(/\bno-
  js\b/, 'js')})(document.documentElement);</script> <link type="text/css"
  media="all" href="https://mk0editwpm6qbfrow5o9.kinstacdn.com/wp-
  content/cache/autoptimize/css/autoptimize_499850c34eb7b4e7cbd9d8b41b2a5
  7d6.css" rel="stylesheet" /><style type="text/css"
  media="print">#wpadminbar{display:none}</style><style type="text/css"
  media="screen">html{margin-top:32px !important}* html body{margin-
```

Puede utilizar el plugin gratuito de Autooptimize o el plugin premium WP Rocket para minificar fácilmente sus archivos.

Utilice Dominios Libres de Cookies

Generalmente, cuando uno sirve contenido como imágenes, JavaScript, CSS, no hay razón para que una cookie HTTP lo acompañe, ya que crea una carga adicional. Una vez que el servidor establece una cookie para un dominio en particular, todas las peticiones posteriores de HTTP para ese dominio deben incluir una cookie. Esta advertencia es típicamente vista en sitios con grandes números de peticiones.

Muchas veces, uno puede ignorar esta advertencia, ya que nuevos protocolos, como HTTP/2, lo hacen menos importante. El precio de una nueva conexión es normalmente más costoso que hacer streaming a todo sobre esa misma conexión.

Una forma sencilla de arreglar esta advertencia es utilizar un proveedor de CDN que pueda ignorar cookies, y pueda eliminar cookies, de esta forma se podría prevenir que el cliente reciba el encabezado de respuesta de la cookie establecida. KeyCDN es un proveedor de CDN que ofrece esta opción. Por defecto, podrá ver que las siguientes dos opciones estarán habilitadas. Esta es una alternativa sencilla sin tener que lidiar con el proceso de mover y configurar su sitio para entregar activos estáticos en subdominios separados.

Cache Cookies *

By default, files with cookies are not cacheable. However, enabling this option will ignore the presence of cookies and therefore force the edge servers to cache these files.

Strip Cookies *

This feature strips the cookies received from the origin server. The client will not receive the `Set-Cookie` response header. **It is highly recommended to enable this directive if you enable Cache Cookies.**

Si usted está usando Cloudflare, no puede deshabilitar las cookies en los recursos servidos a través de su red. CloudFlare incluye su propia cookie en el encabezado. De nuevo, estas cookies son muy pequeñas y las implicaciones en el rendimiento son extremadamente mínimas. Pero si usted utiliza Cloudflare, no hay forma de evitar esta advertencia.

Una segunda forma de lidiar con esto es re-configurando su sitio de WordPress para entregar activos estáticos desde un nuevo dominio o subdominio.

Deshabilitar Embebidos en WordPress

Cuando lanzaron WordPress 4.4, añadieron la opción de oEmbed al core. Esto permite a los usuarios incrustar videos de YouTube, tweets y muchos otros recursos en sus sitios simplemente con pegar una URL, de esta forma WordPress, automáticamente, la convierte en un embed y provee una vista previa en vivo en su editor visual. Con esta actualización, WordPress se convierte en el proveedor de oEmbed.

Esta opción es útil para muchas personas, y es mejor mantenerla activa.

Sin embargo también genera peticiones de HTTP adicionales en su sitio de WordPress para cargar el archivo `wp-embed.min.js`.

| Name | Status | Domain |
|---|--------|-----------------|
| <input type="checkbox"/> editwp.com | 200 | editwp.com |
| <input type="checkbox"/> global.js | 200 | mk0editwpn... |
| <input type="checkbox"/> css?family=Libre+Franklin%3A300%2C300%2C400%2C400... | 200 | fonts.google... |
| <input type="checkbox"/> jquery.scrollTo.js | 200 | mk0editwpn... |
| <input type="checkbox"/> wp-embed.min.js ← | 200 | mk0editwpn... |
| <input type="checkbox"/> style.css | 200 | mk0editwpn... |
| <input type="checkbox"/> jquery.js | 200 | mk0editwpn... |
| header.jpg | 200 | mk0editwpn... |
| <input type="checkbox"/> skip-link-focus-fix.js | 200 | mk0editwpn... |
| <input type="checkbox"/> wp-emoji-release.min.js | 200 | editwp.com |

Y este se carga en todo el sitio. Mientras que este archivo tan solo pesa 1.7 KB, más funcionalidades del estilo se van acumulando con el tiempo. La petición en algunas ocasiones es mucho más importante que el tamaño del contenido descargado.

Fácilmente podría lograr que este archivo no se cargue. Aquí tenemos algunas opciones:












- Deshabilitar Embeds con un Plugin (Gratis, disponible en WordPress.org)
- Perfmatters (Disponible en perfmatters.io)

O también puede deshabilitarlo por código. Puede ver todo el proceso en nuestra web <https://kinsta.com/es/base-de-conocimiento/desactivar-embebidos-wordpress/>

Deshabilitar Emojis en WordPress

Similar a los embeds, en WordPress 4.2, agregaron soporte para emojis en el core para navegadores antiguos. El gran problema con esto es que genera peticiones HTTP adicionales en su sitio de WordPress para cargar el archivo `wp-emoji-release.min.js`.

Y esto se carga en todo el sitio. Mientras que este archivo tan sólo pesa unos 10.5 KB, es inútil si no utiliza emojis en su sitio.

| Name | Status | Domain |
|--|--------|------------------|
|  editwp.com | 200 | editwp.com |
|  global.js | 200 | mk0editwpn... |
|  css?family=Libre+Franklin%3A300%2C300%2C400%2C400... | 200 | fonts.google... |
|  jquery.scrollTo.js | 200 | mk0editwpn... |
|  wp-embed.min.js | 200 | mk0editwpn... |
|  style.css | 200 | mk0editwpn... |
|  jquery.js | 200 | mk0editwpn... |
|  header.jpg | 200 | mk0editwpn... |
|  skip-link-focus-fix.js | 200 | mk0editwpn... |
|  wp-emoji-release.min.js ← | 200 | editwp.com |
|  jizDREVItHgc8qDlbSTKq4XkRiUf2zcZiVbJ.woff2 | 200 | fonts.gstatic... |

Hay un par de formas distintas para deshabilitar los Emojis en WordPress. Puedes hacerlo con los siguientes plugins:

- Deshabilitar Emojis (Gratis, disponible en WordPress.org)
- Perfmatters (Disponible en perfmatters.io)

○ lo puedes deshabilitar por código. Puedes ver como hacerlo en nuestra web en el siguiente enlace: <https://kinsta.com/knowledgebase/disable-emojis-wordpress/>

Cómo Acelerar los Comentarios de WordPress o Deshabilitarlos

Una sección de comentarios repleta en un sitio puede causar muchos problemas de rendimiento. Sólo piense en los recursos que se usan para hacer que funcionen los comentarios:

- Es necesaria una base de datos para mostrar los comentarios.
- Las entradas de la base de datos son creadas para cada comentario.
- Los comentarios y los metadatos de los comentarios son recibidos y procesados por el navegador del visitante.
- Los recursos externos, como los Gravatars, son solicitados, descargados y cargados (requiriendo una búsqueda de DNS separada).
- En muchos casos, tenemos que descargar y procesar grandes recursos de JavaScript y jQuery para lograr que funcione el sistema de comentarios de la forma en que se supone que deba funcionar.

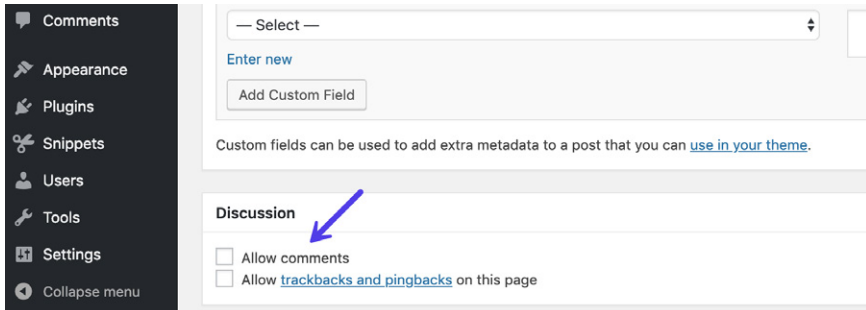
Aquí tenemos cuatro opciones distintas para acelerar los comentarios en WordPress:

Opción 1 – Deshabilitar los Comentarios

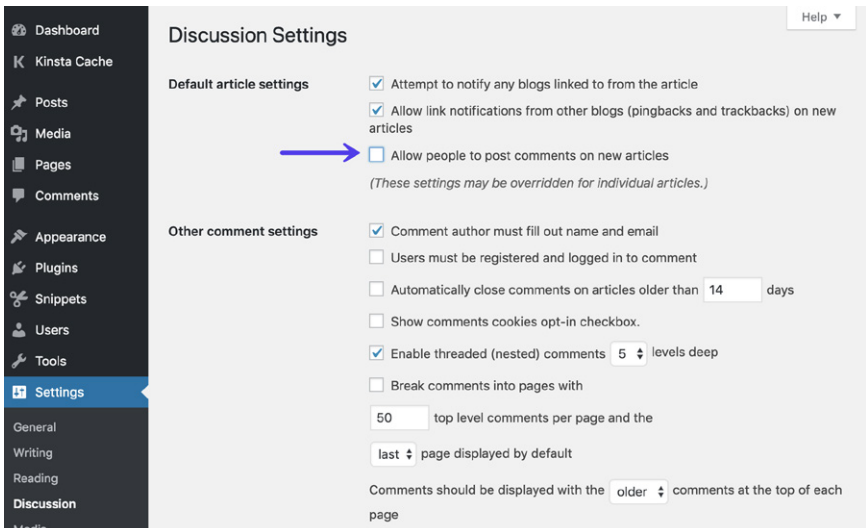
Si su sitio no está teniendo muchos comentarios y no cree que aporte gran valor, podría ser mejor deshabilitar el sistema de comentarios. Recuerde, los comentarios pueden impactar en el SEO ya que Google normalmente pondrá estos como contenido adicional de la página, así que sólo debe aprobar comentarios de alta calidad. Revise estas tres formas sencillas para deshabilitar los comentarios:

1. Deshabilitar Comentarios dentro de las Opciones de WordPress

Puede deshabilitar los comentarios individualmente en una página o entrada editándolos y desactivando la opción “Permitir comentarios” en la sección “Comentarios”. Si no ve esta sección, es posible que deba habilitarla en “Opciones de pantalla” (Se encuentra arriba a la derecha).



También puede deshabilitar globalmente los comentarios. Puede hacerlo en “Comentarios” dentro de los ajustes en su panel de WordPress y luego deseleccionando la opción “Permite que se publiquen comentarios en los artículos nuevos “. El inconveniente de este método es que no afecta a los artículos antiguos, solo a los nuevos.



Si necesita más control o simplemente desea deshabilitar globalmente todos los comentarios a la vez, le recomendamos que utilice uno de los siguientes métodos:

2. Deshabilitar Comentarios con un Plugin

Una de las formas más fáciles de deshabilitar comentarios es usar un plugin gratuito como Deshabilitar comentarios (disponible en WordPress.org). Puede deshabilitar comentarios en cualquier lugar con un solo clic o deshabilitarlos en entradas, páginas o medios.

3. Deshabilitar Comentarios con Código

También puede deshabilitar los comentarios con código simplemente eliminando la sección de comentarios para que no aparezca en su tema de WordPress con simples comentarios HTML.

En el archivo `page.php`, simplemente reemplace el siguiente código:

```
<?php comments_template( '', true ); ?>
```

Con este código:

```
<!-- Begin Comment  
<?php comments_template( '', true ); ?>  
End Comment -->
```

Este código elimina los comentarios de tus páginas. Y luego, para eliminarlo de las entradas de su blog, vaya al archivo `single.php` y reemplace el siguiente código:

```
<?php comments_template(); ?>
```

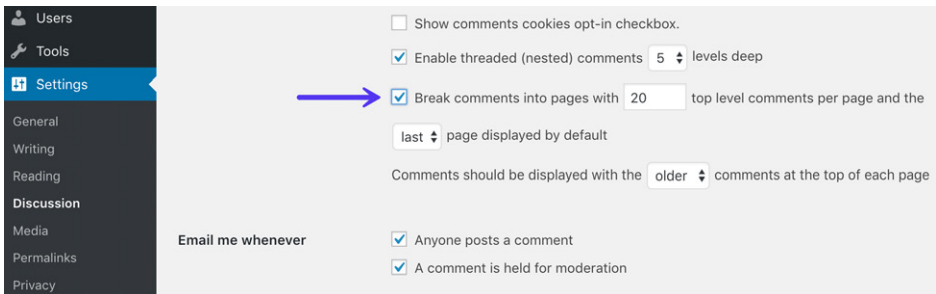
Con este código:

```
<!-- Begin Comment  
<?php comments_template(); ?>  
End Comment →
```

Opción 2 – Optimizar los Comentarios Nativos de WordPress

Su segunda opción sería optimizar el sistema de comentarios nativo de WordPress. Una forma sería reducir el número de comentarios que son cargados inicialmente en la página.

1. Vaya a Opciones → Discusión en el área de admin de WordPress.
2. Busque la sección de otras opciones de comentarios.
3. Seleccione la cajita a un lado de Separar comentarios en páginas y agregar valor por el número de comentarios que quiera mostrar en la carga inicial de la página.



Otra opción que tiene es alojar los Gravatars en su CDN. Este es el enfoque que elegimos en Kinsta.

Por defecto, cuando los comentarios de WordPress son cargados, cada Gravatar único requiere una petición de HTTP. Así que si una página cargada con comentarios de 50 participantes distintos, se requerirán 50 peticiones de HTTP para descargar todos esos Gravatars. Como se podrá imaginar, esto puede impactar la velocidad de su página. Sin mencionar el hecho de que hemos visto que la búsqueda externa de DNS a gravatar.com también puede tardar mucho y en algunos casos desconectarse por completo.

Si usted ve los Gravatars en el blog de Kinsta, podrá ver que se están cargando de Kinsta.com (incluyendo nuestra CDN).

Siempre asegúrese de hacer pruebas de velocidad al usar sistemas de comentarios externos. Vea todas estas peticiones separadas que Disqus genera (como se muestra a continuación). Mientras que muchas de estas peticiones están siendo cargadas de forma asíncrona, aún así notará algo de tiempo de carga adicional si está utilizando Disqus.

Opción 4 – Hacer Carga Diferida a los Comentarios

Su cuarta opción es hacer lazy load a los comentarios, para que no haga más lenta la carga durante la visualización inicial de la página. Aquí tenemos un par de plugins que podría revisar:

- **Lazy Load for Comments** (Gratis, disponible en WordPress.org): Este plugin le permite hacer carga diferida a los comentarios nativos de WordPress.
- **Disqus Conditional Load** (Gratis, disponible en WordPress.org): Si quiere utilizar el sistema de comentarios de Disqus, este es un plugin necesario para hacer lazy load a los comentarios.

Deshabilitar WordPress RSS Feeds

Si no está utilizando la función de blogging de WordPress en su sitio, usted podría deshabilitar los RSS feeds de WordPress. Probablemente esto no cause un gran impacto en el rendimiento, pero todo puede ser útil. Es una cosa menos de la que preocuparse.

Consulte estas dos formas para deshabilitar los RSS feeds en WordPress:

1. Deshabilitar RSS Feed con un Plugin

La primera forma de deshabilitar un feed RSS de WordPress es usar un plugin gratuito como Disable Feeds (disponible en WordPress.org). Este plugin desactiva todas las feed RSS / Atom en su sitio web de WordPress al redirigir todas las solicitudes.

También puede usar un plugin premium como Perfmatters (desarrollado por un miembro del equipo de Kinsta, disponible en perfmatters.io), que le permite tanto deshabilitar las feeds RSS como deshabilitar los enlaces de las feeds RSS, junto con otras optimizaciones de su sitio WordPress.

2. Deshabilitar RSS Feed con Código

Si no desea utilizar un plugin, puede desactivar las feeds RSS con código. Coloque el siguiente código en el archivo `functions.php` de su tema.

```
function itsme_disable_feed() {  
    wp_die( __( 'No feed available, please visit the <a href="'. esc_url(  
        home_url( '/' ) ) .'">homepage</a>!' ) );  
}  
  
add_action( 'do_feed', 'itsme_disable_feed', 1 );  
add_action( 'do_feed_rdf', 'itsme_disable_feed', 1 );  
add_action( 'do_feed_rss', 'itsme_disable_feed', 1 );  
add_action( 'do_feed_rss2', 'itsme_disable_feed', 1 );  
add_action( 'do_feed_atom', 'itsme_disable_feed', 1 );  
add_action( 'do_feed_rss2_comments', 'itsme_disable_feed', 1 );  
add_action( 'do_feed_atom_comments', 'itsme_disable_feed', 1 );
```

WordPress también genera enlaces a las feeds RSS dentro del encabezado de su página web. Puede ir un paso más allá y eliminar estos enlaces dentro del código HTML de sus páginas. Coloque el siguiente código en el archivo `functions.php` de su tema.

```
remove_action( 'wp_head', 'feed_links_extra', 3 );  
remove_action( 'wp_head', 'feed_links', 2 );
```

Utilice Prefetch y Preconnect

Las sugerencias y directivas de recursos como prefetch y preconnect pueden ser una gran forma para acelerar WordPress detrás de escena.

Prefetch

El DNS prefetch le permitirá resolver los nombres de dominio (llevar a cabo búsquedas DNS en background) antes de que un usuario haga clic en el enlace, lo que causará una mejora en el rendimiento. Esto se hace al agregar una etiqueta `rel="dns-prefetch"` en el encabezado de su sitio de WordPress.

```
<link rel="dns-prefetch" href="//domain.com">
```

Se suele usar el prefetching de DNS para la URL de su CDN, Google fonts, Google Analytics, etc.

```
<link rel="dns-prefetch" href="//cdn.domain.com/">  
<link rel="dns-prefetch" href="//fonts.googleapis.com/">  
<link rel="dns-prefetch" href="//www.google-analytics.com">
```

El prefetch también es soportado por la mayoría de los navegadores modernos. Aquí algunas formas de implementarlo.

1. Puede agregar el código a su header usando un plugin gratuito como Head, Footer y Post Injections (disponible en WordPress.org).
2. Puede agregar el código manualmente al archivo `header.php` de su tema.
3. Implementando un DNS prefetch utilizando un plugin como Perfmatters. Simplemente haga clic en la pestaña de "Extras" en el plugin de Perfmatters y agregue dominios. Formato: `//domain.tld` (uno por línea)

The screenshot shows the 'Perfmatters Settings' interface. On the left is a dark sidebar with navigation options: Dashboard, Kinsta Cache, Posts, Media, Pages, Comments, Appearance, Plugins, Users, Tools, and Settings (highlighted in blue). Below the sidebar is a 'General' section. The main content area is titled 'Perfmatters Settings' and has several tabs: Options, CDN, Google Analytics, Extras (selected), License, and Support. Under the 'Extras' tab, there is a sub-header 'Extras' with the text 'Extra options that pertain to Perfmatters plugin functionality.' Below this, there is a 'Script Manager' toggle switch which is currently turned off. Underneath is a 'DNS Prefetch' section with a text input field. A blue arrow points to this field, which contains the following text:


```
//cdn.domain.com/
//fonts.googleapis.com/
//www.google-analytics.com/
```

Preconnect

Preconnect le permite al navegador establecer conexiones anticipadas antes de que suceda una petición de HTTP, eliminando la latencia del viaje completo y ahorrando tiempo a los usuarios.

Preconnect es una herramienta importante en su caja de herramientas de optimización... este puede eliminar muchos viajes de ida y vuelta – y en algunos casos, reducir la latencia de petición en cientos e incluso miles de milisegundos.



Iya Grigorik

Se hace al agregar una etiqueta `rel="preconnect"` en el encabezado de su sitio de WordPress.

```
<link rel="preconnect" href="//domain.com">
```

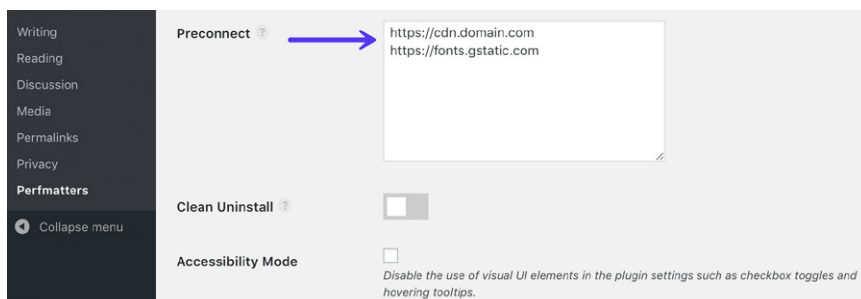
Algunos ejemplos de cosas en las que podría utilizar es en la URL de su CDN o Google Fonts.

```
<link rel="preconnect" href="https://cdn.domain.com">
```

```
<link rel="preconnect" href="https://fonts.gstatic.com">
```

Preconnect es soportado por la gran mayoría de navegadores modernos, con la excepción de Internet Explorer, Safari, IOS Safari, y Opera Mini. Aquí algunas formas de implementarlo:

1. Puede agregar el código a su header usando un plugin gratuito como Head, Footer y Post Injections (disponible en WordPress.org).
2. Puede agregar el código manualmente al archivo `header.php` de su tema.
3. Implementando preconnect, utilizando un plugin como Perfmatters. Simplemente haga clic en la pestaña de “Extras” en el plugin de Perfmatters y agregá dominios. Formato: `scheme://domain.tld` (uno por línea).



Deshabilite Scripts por Página/Entrada

Otra poderosa forma de acelerar WordPress es revisar cada petición que está cargando en sus páginas y publicaciones. Probablemente termine encontrando scripts que están cargando en todo el sitio, cuando no deberían hacerlo.

Puede utilizar un plugin premium como Perfmatters, tiene una opción de “Script manager”. Esto le permitirá deshabilitar scripts (CSS y JavaScript) por página/publicación, o incluso en todo el sitio con un sólo clic. Com decíamos anteriormente, este plugin está desarrollado por un miembro del equipo de Kinsta.

Algunos ejemplos de cosas que se pueden hacer con esto:

- El popular plugin Contact Form 7 se carga a sí mismo en cada página y publicación. Fácilmente puede deshabilitarlo en todos lados con un clic y habilitarlo solo para su página de contacto.
- Los plugins para compartir en las redes sociales solo deberían cargarse en sus publicaciones. Fácilmente puede deshabilitarlo en todos lados y sólo permitir que cargue en tipos de publicaciones o tipos de publicaciones personalizados.
- El plugin de Tabla de Contenidos (TOC) es cargado en cada página y publicación. Con el administrador de scripts, fácilmente puede controlar en dónde se cargará.
- Si ha actualizado a WordPress 5.0 y no está usando el editor de bloque de Gutenberg, quizá esté usando el editor clásico u otro editor de terceros, hay dos scripts que se agregan en todo el sitio y que puede deshabilitar:
`/wp-includes/css/dist/block-library/style.min.css` y `/wp-includes/css/dist/block-library/theme.min.css`.

¿Por qué Algunos Plugins Están Programados de esta Forma?

Podría estar preguntándose ¿por qué todos los desarrolladores de plugins no cargan sus scripts sólo cuando el plugin es detectado en la página? Bueno, es un poco más complicado de lo que parece. Por ejemplo, si tiene un plugin como Contact Form 7, este también tiene shortcodes que le permitirán colocarlo en cualquier lado. Esto incluye colocarlo en un widget. Con WordPress, es mucho más difícil hacer consultas a datos de ellos cuando uno quita los scripts, en lugar de hacer consulta a datos de la publicación o metadatos de la página.

Por lo tanto, en muchas ocasiones esto es debido a problemas de usabilidad. Entre menos oportunidad tenga para que se rompa un plugin, menor serán los tickets y soporte que tendrán. Sin embargo, con tantos plugins en el mercado, hay formas de lidiar con esto y programar para obtener un buen rendimiento si así lo desean. Desafortunadamente, algunas veces, el gran número de descargas y usuarios hacen que el programar por mejorar la usabilidad sea una prioridad.

Conociendo el Administrador de Scripts

Le daremos un pequeño tour del Administrador de Scripts. Después de hacer clic en su barra de herramientas, verá todos los scripts que son cargados en la URL actual, archivos de JavaScript y CSS. Luego, usted tendrá las siguientes opciones:

- 1. Estado Encendido:** (Opción por defecto)
- 2. Estado Apagado:** Deshabilitar en todos lados (más tarde puede elegir qué tipos de publicaciones quiere tenerlo habilitado, junto con la URL actual)
- 3. Estado Apagado:** Deshabilitar sólo en la URL actual (es muy útil para la página de inicio)
- 4. Estado Apagado:** Excepciones (URL actual, tipo de publicación, o archivo)

perfmatters

Script Manager

Global View

Settings

Script Manager

Manage scripts loading on the current page.

Plugins

Extended Widget Options ON

| Status | Script | Type | Size |
|-----------------|--|------|---------|
| ON | jquery-widgetopts /wp-content/plugins/extended-widget-options/assets/js/jquery.widgetopts.min.js | JS | 6.1 KB |
| ON | widgetopts-styles /wp-content/plugins/extended-widget-options/assets/css/widget-options.css | CSS | 13.6 KB |

Social Warfare ON

| Status | Script | Type | Size |
|--|---|------|---------|
| OFF | social_warfare_script /wp-content/plugins/social-warfare/js/script.min.js | JS | 10 KB |
| Disabled ⊙ Everywhere ⊙ Current URL | | | |
| Exceptions ⊙ Current URL | | | |
| Post Types: ☑ Posts ⊙ Pages ⊙ Lists ⊙ Coupons | | | |
| OFF | social_warfare /wp-content/plugins/social-warfare/css/style.min.css | CSS | 45.2 KB |
| Disabled ⊙ Everywhere ⊙ Current URL | | | |
| Exceptions ⊙ Current URL | | | |

Todo está **agrupado por el plugin o el nombre del tema**. Esto hace que sea súper sencillo deshabilitar el plugin entero. Normalmente un plugin de WordPress tendrá un archivo de JavaScript y de CSS. Un tema de WordPress podría tener más de 10 archivos.

Después de seleccionar y/o modificar las opciones, asegúrese de hacer clic en “Guardar” en el parte de abajo. Luego podrá probarlo con una herramienta de prueba de velocidad, para asegurarse de que los scripts no estén siendo cargados en la página o publicación. ¡Asegúrese de limpiar primero el caché! Y si la web se ve mal, siempre podrá dejar las opciones por defecto para volver a la normalidad.

En una prueba de velocidad por woorkup.com, pudieron **disminuir el tiempo total de carga en un 20.2%**. En su página de inicio, pudieron reducir el número de peticiones HTTP de 46 a 30. El tamaño de su página también se hizo más pequeño, de 506.3 KB a 451.6 KB.

Analizando el Rendimiento de Terceros

Básicamente, cualquier cosa que llame externamente desde su sitio, tiene consecuencias en los tiempos de carga. Lo peor de este problema es que algunos de estos es que son lentos de forma intermitente, haciendo que sea muy difícil identificar el problema.

Un servicio externo podría ser considerado cualquier cosa que se comunica con su sitio de WordPress desde fuera de su servidor. Aquí tenemos algunos ejemplos con los que nos hemos encontrado de forma regular:

- Plataformas de redes sociales, como Twitter, Facebook e Instagram (widgets o pixels de conversión)
- Redes externas de publicidad como Google AdSense, Media.net, BuySellAds, Amazon Associates.
- Analítica de sitios web y scripts de rastreo como Google Analytics, Crazy Egg, Hotjar, AdRoll

- Herramientas de Prueba A/B como Optimizely, VWO, Unbounce
- Sistemas de comentarios de WordPress como Disqus, Jetpack, Facebook comments
- Herramientas de backups y de seguridad como VaultPress, Sucuri y CodeGuard
- Herramientas para compartir en redes sociales como SumoMe, HelloBar
- Redes CDN como KeyCDN, Amazon CloudFront, CDN77, y StackPath
- JavaScript con un host externo

¿Cuántos de estos rastreadores impactan en el rendimiento? En nuestro propio caso de estudio, notamos que los scripts externos **incrementaron los tiempos de carga en un 86.08%**.

Ghostery también midió los 500 mejores dominios de EUA en Alexa, y los resultados fueron asombrosos, pero para nosotros, no fue realmente una sorpresa. Los sitios web eran 2x veces más lentos cuando no se habían bloqueado a estos rastreadores. Lo que quiere decir que los scripts de rastreo externos son uno de los principales contribuidores para reducir las velocidades de carga en la red.

AVERAGE PAGE LOAD TIME WITH vs. WITHOUT TRACKERS

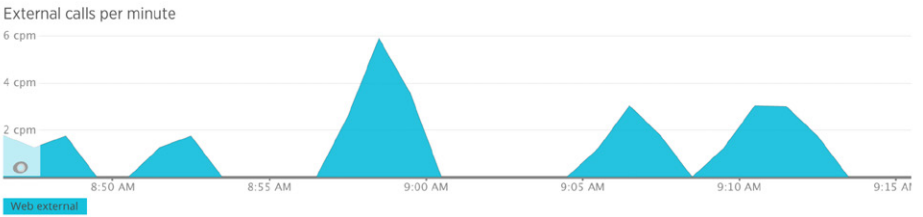
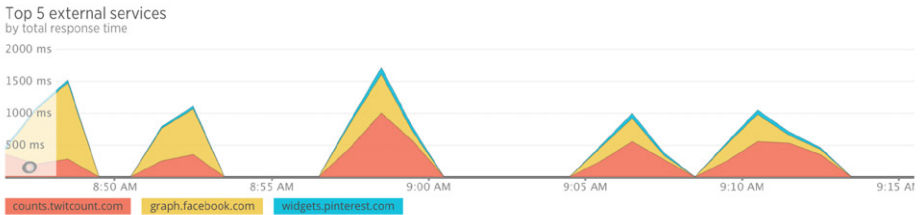


Average page load time (seconds) measured by Ghostery, of the top 500 US domains according to Alexa.

Fuente de la imagen: Ghostery

Tiene que tener mucho cuidado en su sitio de WordPress. ¡Tener tan sólo una llamada de un API externa podría causar que su sitio entero se caiga! Sí, no debería ser así, pero en muchos casos, así es. Lo hemos visto más veces de las que podemos contar.

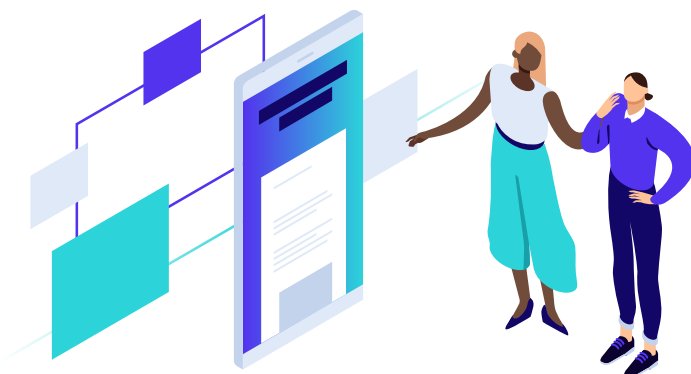
New Relic brinda una excelente y sencilla forma para monitorear servicios externos en el tiempo. En este ejemplo de abajo, podemos ver llamadas externas que se hacen a [twitcount.com](https://counts.twitcount.com), graph.facebook.com y widgets.pinterest.com



Es importante que cuando agregue una nueva opción o plugin a su sitio, primero investigue los recursos externos que se están cargando de este. ¡Entre menos mejor!

CAPÍTULO 16:

Siempre Optimice con los Dispositivos Móviles en Mente



Google empezó a sacar su índice de mobile-first el 26 de marzo del 2018. Previamente los sistemas de crawling, indexación y de posicionamiento de Google, usaban la versión de escritorio de los sitios web. La indexación centrada en móviles significa que el Googlebot ahora usará la versión móvil de su sitio de WordPress para indexar y posicionar. Esto ayuda a mejorar la experiencia para usuarios móviles.

Cuando se trata de optimizar su sitio para dispositivos móviles, **la velocidad es uno de los factores más importantes a tener en cuenta.** La velocidad juega un rol importante en todo, desde usabilidad al porcentaje de rebote y para determinar si los compradores potenciales regresarán o no a su sitio. De hecho, la velocidad es un factor importante para posicionarse en el buscador de Google y en los anuncios para las búsquedas en dispositivos móviles.

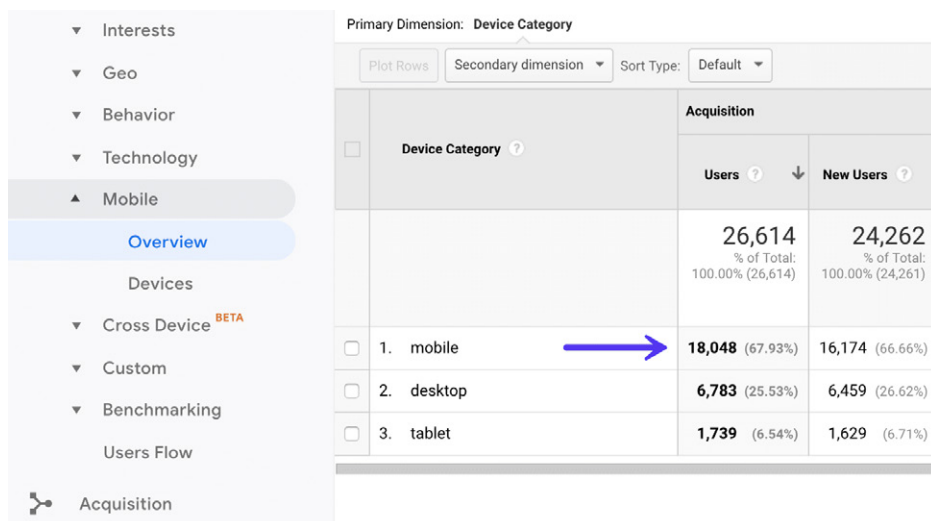
Las malas experiencias móviles llevarán a la mayoría de los usuarios a no regresar nunca. De acuerdo al último reporte de velocidad de página de Google, el tiempo promedio en que un sitio móvil tarda en cargar en 2018 es de 15 segundos. ¿Se puede imaginar esperar tanto tiempo para cargar una sola página? Sorprendente.

Los usuarios demandan (y merecen) algo mejor. De acuerdo al mismo informe de velocidad de página, **53% de los visitantes de sitios móviles dejan de esperar si la página tarda más de 3 segundos en cargar.**

Las experiencias lentas en dispositivos móviles están matando las conversiones. Estas previenen que usted obtenga la oportunidad de convertir clientes potenciales desde un principio. A medida que incrementen los tiempos de carga por unos segundos, la probabilidad de que alguien se vaya de su web aumenta exponencialmente. Aquí hay algunas cosas a considerar cuando uno optimiza para dispositivos móviles.

Revise Su Tráfico Móvil

Siempre es importante tomarse el tiempo de ver cuánto tráfico móvil está recibiendo, ya que esto podría cambiar un poco sus prioridades. Puede ver cuántos dispositivos móviles están visitando su sitio en Google Analytics bajo “Audiencia → Móvil → Vista general.” Como puede ver en este sitio, más del 67% de este tráfico viene de móviles. ¡Eso es muchísimo!



Primary Dimension: Device Category

Plot Rows Secondary dimension Sort Type: Default

| Device Category ? | Acquisition | |
|-------------------------------------|---|---|
| | Users ? ↓ | New Users ? |
| | 26,614 % of Total: 100.00% (26,614) | 24,262 % of Total: 100.00% (24,261) |
| <input type="checkbox"/> 1. mobile | 18,048 (67.93%) | 16,174 (66.66%) |
| <input type="checkbox"/> 2. desktop | 6,783 (25.53%) | 6,459 (26.62%) |
| <input type="checkbox"/> 3. tablet | 1,739 (6.54%) | 1,629 (6.71%) |

Si usted es cliente de Kinsta, también puede revisar el tráfico de los dispositivos móvil vs computadoras en MyKinsta Analytics. Como puede ver en este sitio, más del 88% del tráfico viene de las ordenadores. Siempre es importante verificar y no sólo asumir. Simplemente porque todos dicen que el tráfico móvil es superior, no quiere decir que pase lo mismo con su sitio. Vea los datos.

Mobile vs. Desktop

88.9% Desktop 11.1% Mobile



Sat 10/20/2018 Sun 10/21/2018 Mon 10/22/2018 Tue 10/23/2018 Wed 10/24/2018 Thu 10/25/2018 Fri 10/26/2018 Sat 10/27/2018 Sun 10/28/2018 Mon 10/29/2018 Tue 10/30/2018 Wed 10/31/2018 Thu 11/01/2018 Fri 11/02/2018 Sat 11/03/2018 Sun 11/04/2018 Mon 11/05/2018 Tue 11/06/2018 Wed 11/07/2018 Thu 11/08/2018 Fri 11/09/2018 Sat 11/10/2018 Sun 11/11/2018

Asegúrese de que Su Sitio Sea Responsive

Tested on: Nov 18, 2018 at 11:13 PM

Page is mobile friendly

This page is easy to use on a mobile device

Additional resources

- Open site-wide mobile usability report
- Learn more about mobile-friendly pages
- Post comments or questions to our discussion group

SCREENSHOT HTML

En 2019, ¡Más vale que su sitio sea responsive! Esto quiere decir que utiliza media queries para escalar las cosas de forma automática a dispositivos móviles. Si aún no lo ha hecho, entonces ya está detrás de su competencia. Todos los temas de WordPress que hemos mencionado en este artículo funcionan y se ven geniales en todos los dispositivos.

Utilice la herramienta Google Mobile-Friendly para hacer pruebas y asegurarse de que su sitio pase todos los requerimientos.

Revise Varias Veces para Asegurarse de que srcset Funcione

En el pasado, era muy importante cargar las imágenes para escalar y no permitir que CSS altere su tamaño. Sin embargo, esto ya no es importante ya que WordPress 4.4 ahora soporta imágenes responsivas (no escaladas por CSS). WordPress automáticamente crea varios tamaños distintos de cada imagen subida a la biblioteca de medios. Al incluir los tamaños disponibles de una imagen en un atributo srcset los navegadores ahora pueden elegir descargar el tamaño más apropiado e ignorar los otros. Vea este ejemplo de como se verá su código.

```

```

Debido a plugins externos de imágenes disponibles, hemos visto, en varias ocasiones, que esto no funcione de forma correcta. Por lo tanto, es importante revisar varias veces para verificar que sus imágenes estén recibiendo de forma apropiada el atributo srcset agregado con versiones distintas para diferentes tamaños de pantalla. La optimización de imagen es importante de ahora en adelante.

Google AMP Podría Ser la Solución Para Usted

Google AMP (Accelerated Mobile Pages Project) fue originalmente lanzado en octubre de 2015. El proyecto dependía de un AMP HTML, un nuevo framework abierto, construido en su totalidad de tecnologías existentes de la red, que permite a los sitios web, construir páginas livianas. En pocas palabras, ofrece una forma de servir una versión básica de la página web actual.

Tenemos una relación de amor/odio con Google AMP, y al parecer gran parte de la comunidad piensa lo mismo. Lo hemos probado y no obtuvimos buenos resultados. Sin embargo, esto no quiere decir que no funcione. Cada sitio web es distinto, y Google AMP está siendo constantemente mejorado.

Puede empezar rápidamente con Google AMP en su sitio de WordPress con uno de los siguientes plugins:

- AMP para WordPress (Gratis, disponible en WordPress.org)
- AMP para WP (Gratis, disponible en WordPress.org)

Resumen

Cómo ya se habrá dado cuenta, estamos obsesionados con todos los tipos de formas en las que usted puede acelerar su WordPress. Tener un sitio rápido es necesario para incrementar su posicionamiento, mejorar el rastreo de los motores de búsqueda, mejorar las tasas de conversión, incrementar el tiempo que un usuario está en el sitio y reduce la tasa de rebote. ¡Sin mencionar el hecho de que todos aman visitar un sitio web veloz!

Esperamos que esta guía haya sido útil y de su agrado, y que haya memorizado algunas cosas y las aplique en su sitio de WordPress.

Asegúrese de consultar también nuestro programa de afiliados de hosting de WordPress en <https://kinsta.com/affiliates>. Gane hasta \$ 500 por cada recomendación + 10% de comisiones mensuales recurrentes de por vida. Esta es la mejor oferta de afiliados que encontrará en la industria. ¡Vamos a acelerar juntos la web!

Para obtener más información sobre Kinsta, visítenos en <https://kinsta.com>.

¡Gracias por leer!



Notas

La velocidad de tu sitio web de WordPress es más importante de lo que piensas. No importa el tipo de sitio que tengas; ya sea una tienda WooCommerce o un blog, los tiempos de carga de la página están directamente correlacionados con:

- Mejores posiciones de palabras clave (la velocidad es un factor que tiene en cuenta Google).
- Aumento de las tasas de conversión.
- Mejora de la capacidad de rastreo de los motores de búsqueda como Google y Bing.
- Mayor tiempo de permanencia en su sitio.
- Disminución de las tasas de rebote.

Sin mencionar que tener un sitio web que carga rápido proporciona una mejor experiencia para sus visitantes.

En la actualidad, ¡más de un tercio de todos los sitios web en Internet funcionan bajo WordPress! Aunque es impresionante, también significa que hay miles de temas, plugins y tecnologías diferentes que tienen que coexistir. Para el usuario de WordPress, esto puede convertirse en una pesadilla cuando su web comienza a congestionarse. Muchos no saben por qué o ni siquiera por dónde empezar a solucionar los problemas.

El equipo de Kinsta se ocupa de este tipo de problemas a diario, ¡y lo ha visto todo! En esta guía, profundizamos en los pasos aplicables que puedes seguir ahora mismo para acelerar tus sitios de WordPress. Desde consejos para principiantes hasta estrategias avanzadas, seguro que sea cual sea tu nivel encontrarás algo útil. Contiene más de 15 años de experiencia y duras lecciones que hemos aprendido.

KINSTA