

Un segundo, vamos a ver qué tal...; dos segundos, a ver si carga...; tres segundos, ¿qué le pasa a esto?; cuatro segundos, me desespero; cinco segundos, bufff, no puedo más, abandono.

Así actuamos, así somos: demandamos inmediatez. Lo queremos todo ya, rápido, al momento. No queremos esperar. La espera nos pone nerviosos, nos enfada y altera. Todo tiene que ser fácil, cómodo y rápido. Muy rápido. Porque si no... nos vamos. Así de sencillo. Sin contemplaciones.

Para evitar situaciones semejantes y tener usuarios y/o clientes contentos y sonrientes navegando a través de tu sitio web, voy a darte una serie de pautas con las que espero hacer volar tu WordPress.

Importancia de la velocidad de carga

Por si alguien todavía tiene dudas y aún se cuestiona la necesidad de disponer de un sitio bien optimizado, aquí van tres razones de peso:

Mejora la experiencia de usuario Como he dicho anteriormente, a la gente le gusta los sitios rápidos y tiende a abandonar los lentos. El tiempo de carga es uno de los principales factores que contribuyen al abandono de un sitio web. Los usuarios esperan que una web cargue en una media de dos segundos, tras los cuales comienzan a impacientarse. Y si se prolonga, lo más seguro es que se vayan y, lo peor de todo, que no vuelvan jamás.

Va a favorecer tu posición en los resultados de búsqueda Desde 2010 Google contempla la velocidad de carga como uno de los factores de posicionamiento, que va a repercutir en que aparezcas en una posición u otra en los resultados de búsqueda.

Mayores índices de conversión Este aspecto tiene mucha relación con la experiencia de usuario: si un sitio web tarda mucho en cargar es muy probable que perdamos potenciales clientes. Sin embargo, si la web fluye, el usuario permanecerá más tiempo y tenemos muchas papeletas para lograr conversiones.

Análisis

Análisis general del sitio web

Lo primero que hay que hacer es realizar un testado de rendimiento del sitio web. Para ello disponemos de herramientas muy útiles y además gratuitas que nos van a permitir realizar una primera valoración del estado del sitio y proporcionarnos valiosos consejos de cómo mejorar aquellos aspectos que no están suficientemente depurados:

[Google PageSpeed Insights](#)

[Pingdom Website Speed Test](#)

[GTmetrix](#)

[WebPagetest](#)

[YSlow](#)

Análisis específico del rendimiento de WordPress

Como complemento a las herramientas anteriores con las que hemos realizado un test general, debemos ejecutar, además, un análisis más concreto de nuestra instalación de WordPress. Para ello, recomiendo la instalación del plugin [P3 \(Plugin Performance Profiler\)](#).

Gracias a este fantástico plugin vamos a ser capaces de diagnosticar problemas potenciales y comprobar qué plugins son los que están consumiendo excesivos recursos.

Optimización básica

Una vez realizado el análisis previo, disponemos de información más que suficiente para ponernos manos a la obra e iniciar el proceso de optimización.

Empresa de hosting

Es fundamental elegir un buen hosting profesional y que esté especializado en WordPress. De poco o nada servirá todo lo que hagamos por optimizar el rendimiento de nuestro sitio web, si el servidor en el que está alojado no cumple con unos mínimos:

Servidor Apache o Nginx.

Soporte para PHP 5.6 o superior, con un límite de memoria de al menos 128 MB.

MySQL 5.6 o superior.

Canal de soporte técnico rápido y eficaz, con el que sepas que puedes contactar, ya sea por teléfono, chat o mediante tickets.

Elige bien el tema que vayas a usar

En este caso, es indispensable que analices al detalle su ficha de características y asegúrate de que esté bien optimizado para SEO, velocidad, etc. Echa un vistazo a su demo, porque te dará una idea del impacto que puede suponer este tema sobre la velocidad de carga de tu sitio web. Por otro lado, si eres desarrollador y has programado tu propio tema, utiliza [Theme Check](#) para asegurarte de que has respetado [los estándares y buenas prácticas de WordPress](#). Te ahorrarás futuros quebraderos de cabeza, te lo aseguro.

Actualiza, actualiza y actualiza

Actualiza los plugins que tengas instalados, la versión de WordPress y el tema que estés usando, dado que en las actualizaciones siempre se busca arreglar fallos e incorporar mejoras de rendimiento y seguridad.

Desactiva y elimina todos los plugins que no estés

utilizando

Responde sinceramente: ¿de verdad necesitas todo ese arsenal de plugins, widgets y complementos que tienes actualmente instalados en tu WordPress? En unos casos, podrás desinstalar unos cuantos porque no los usas para nada; en otros, seguro que hay funcionalidades que pueden reemplazarse por una funcioncita en tu archivo functions.php. En definitiva, aplica el sentido común. De esta forma evitaremos estar ocupando espacio en el servidor, consumos excesivos por funcionalidades que ni empleamos y además tener plugins abandonados que, al no estar en uso, ni nos molestemos en actualizar, lo que puede suponer una enorme puerta de entrada para posibles ataques, infecciones con malware, etc.

Reorganiza los plugins activos

Ahora que te has deshecho de ciertas funcionalidades, mediante [Plugin Organizer](#) deberías reorganizar la carga de tus plugins activos y decidir cuándo y dónde deben cargarse. Hay veces que estamos empleando plugins que se cargan en todas las páginas sin necesidad. Y aquí, amigo mío, cada detalle cuenta.

Optimiza la base de datos

Utiliza [WP-Sweep](#) o [WP-Optimize](#) para eliminar revisiones, autoguardados, comentarios borrados, spam, metas huérfanos o duplicados y optimizar la base de datos. Es recomendable que realices un chequeo de la base de datos activando la herramienta de optimización integrada por defecto en WordPress, esto lo puedes hacer mediante la siguiente línea de código insertada en tu archivo wp-config.php:

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

Tras esto, podrás acceder a www.tudominio.com/wp-admin/maint/repair.php y ejecutar las tareas de optimización de la base de datos.

Elimina todos los temas que tengas instalados, pero que no estés usando

Tal y como sucede con los plugins, con los temas hay que hacer lo mismo: elimina lo que no uses, ya que conviene disponer de un sitio lo más limpio y organizado posible.

Optimiza tus imágenes

Por un lado, instala el plugin [EWWW Image Optimizer](#) o [WP](#)

[Smush](#) y ejecutalo para procesar de nuevo todas las imágenes que tengas actualmente alojadas en tu servidor. Mediante este proceso, lograrás imágenes más ligeras y reducir el peso general del sitio

web. Te recomiendo que las nuevas imágenes que vayas subir a tu sitio web las optimices previamente utilizando algún programa para ello como [RIOT](#), que ya mencionamos anteriormente cuando hablamos del [SEO en imágenes](#). Puedes utilizar herramientas como u online como [tinyPNG](#), que permite optimizar imágenes en formato .jpg y .png. A todo esto le podemos añadir la carga progresiva de imágenes. Mediante plugins del tipo [BJ Lazy Load](#), con el que las imágenes se irán cargando conforme el usuario vaya haciendo scroll.

Utiliza CSS Sprites

Consiste en agrupar todas las imágenes que componen tu sitio web (logotipo, fondos, iconos, etc) en una sola imagen, de tal forma que se realiza una petición HTTP por carga en lugar de tantas como elementos gráficos compongan tu página. Existen algunas herramientas muy útiles para ayudarte a componer los sprites:

[Instant Sprite](#)

[CSSSprites.com](#)

[CSS Sprite Generator](#)

[CSS Sprites](#)

Y si además necesitas una ayudita a la hora de generar la hoja de estilos, te recomiendo [SpriteBox](#).

Desactiva las revisiones

Las revisiones son una función de autoguardado que nos ofrece WordPress por defecto. Para desactivarlas, simplemente tenemos que añadir el siguiente código dentro del archivo wp-config.php:

```
define( 'WP_POST_REVISIONS', 0 );
```

Puede ocurrir que haya proyectos en los que necesitemos contar con el autoguardado, pero para no sobrecargar el sistema podemos limitar el número de revisiones que WordPress va a generar por página. Para ello, modificaríamos el código anterior, quedando de la siguiente manera:

```
define( 'WP_POST_REVISIONS', 3 );
```

Cambia el intervalo de autoguardado

Mientras estás editando una página o post, WordPress realiza un autoguardado de borradores. Esto es muy útil para recuperar tu trabajo en el caso de que pierdas conexión, el navegador se quede bloqueado o ante cualquier otro imprevisto. Este autoguardado se realiza cada minuto, pero podemos personalizarlo mediante la

inserción de una simple línea de código en nuestro fichero wp-config para así aumentar el intervalo:

```
define ( 'AUTOSAVE_INTERVAL', 120 );
```

En este caso, estaríamos aumentando el intervalo a 2 minutos.

Implementa un sistema de caché

Esta implementación se puede realizar mediante la instalación de servicios en el servidor como [Varnish Cache](#) o mediante la instalación y configuración de algún plugin de caché como [WP Super Cache](#) o [W3 Total Cache](#). Además, deberías activar la caché interna

de WordPress insertando la siguiente línea de código en tu archivo wp-config.php:

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

Minimiza tus archivos JS y CSS

Comprime estos archivos y agrúpalos en el menor número que sea posible. Puedes utilizar [Autoptimize](#) para facilitarte las cosas.

Lograremos reducir el número de peticiones al servidor y que las páginas carguen más rápido.

Carga las CSS al principio y el Javascript al final

Este consejo no va a reducir el tamaño de la página, pero si va a favorecer la sensación de que la web carga antes, ya que se verá el contenido al instante.

Deshabilita pingbacks y trackbacks

Los pingbacks y trackbacks permiten notificar a los autores de los posts cuándo un artículo suyo ha sido mencionado o enlazado en otros sitios web. Puedes desactivarlo desde Ajustes > Comentarios ya que suelen ser fuente de spam y de consumo de recursos.

Optimización avanzada

¿Qué tal va evolucionando tu sitio hasta ahora? ¿Vas notando mejoría? Si todavía no has logrado un aumento sustancial del rendimiento

Utiliza un CDN (Content Delivery Network)

Cada vez que un usuario visita un sitio web, necesita descargar todos los archivos del servidor en el que se encuentra alojada. El tiempo que tarda en recibirlos es lo que se conoce como tiempo de respuesta. Este tiempo será mayor dependiendo del número de archivos que tenga que descargar, así como de la distancia a la que

se encuentre el usuario respecto al lugar en el que se encuentre ubicado el servidor. El CDN lo que hace es copiar todos los archivos estáticos y alojarlos en uno de los muchos servidores que tienen alrededor del mundo. De esta forma, al usuario se le sirve el contenido ubicado en el CDN y no en tu servidor, que responderá más rápido al tener menos elementos que cargar. Y no sólo eso, sino que además se descargará los archivos que se encuentren ubicados en el servidor más cercano al usuario. Puedes utilizar algunos servicios como [Amazon CloudFront](#) o [CloudFlare](#). Para un sitio normalito, el plan gratuito de CloudFlare puede resultar suficiente.

Activar la compresión Gzip

Con esto logramos que el servidor comprima todo el código fuente, lo que va a reducir la cantidad de datos enviados y repercutirá positivamente en la velocidad de carga. Revisa si tienes activada la compresión Gzip. Para eso puedes hacer uso del servicio [checkgzipcompression.com](#). En el fichero `.htaccess`

```
<IfModule mod_deflate.c>
<IfModule mod_setenvif.c>
  <IfModule mod_headers.c>
    SetEnvIfNoCase ^(\Accept-EncodXng|X-cept-Encoding|X{15}|~{15}|-{15})$ ^((gzip|deflate)\s*,?
\s*)+|[X~-]{4,13}$ HAVE_Accept-Encoding
    RequestHeader append Accept-Encoding "gzip, deflate" env=HAVE_Accept-Encoding
  </IfModule>
</IfModule>
<IfModule mod_filter.c>
  AddOutputFilterByType DEFLATE application/atom+xml \
application/javascript \
application/json \
application/rss+xml \
application/vnd.ms-fontobject \
application/x-font-ttf \
application/xhtml+xml \
application/xml \
font/opentype \
image/svg+xml \
```

```
image/x-icon \
text/css \
text/html \
text/plain \
text/x-component \
text/xml

</IfModule>
</IfModule>
```

Especifica el Vary: Accept-Encoding header

Por si hubiera alguna incompatibilidad tras activar la compresión Gzip.

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

Especificar caché de navegador

Normalmente, los archivos estáticos como imágenes, CSS, librerías javascript no suelen cambiar con demasiada frecuencia, con lo cual resulta altamente recomendado especificar caché del navegador para que estos archivos se almacenen en la caché del navegador del usuario durante el tiempo que especifiquemos. Así, la próxima vez que el usuario acceda a nuestra web, cargará un poco más rápido al tener guardados estos ficheros en su caché local.

```
<IfModule mod_expires.c>
ExpiresActive On
ExpiresByType image/jpg "access plus 1 year"
ExpiresByType image/jpeg "access plus 1 year"
ExpiresByType image/gif "access plus 1 year"
ExpiresByType image/png "access plus 1 year"
ExpiresByType text/css "access plus 1 month"
ExpiresByType application/pdf "access plus 1 month"
ExpiresByType text/x-javascript "access plus 1 month"
ExpiresByType application/x-shockwave-flash "access plus 1 month"
ExpiresByType image/x-icon "access plus 1 year"
ExpiresDefault "access plus 2 days"
</IfModule>
```

Activa 'Keep-Alive'

De esta forma vamos a permitir que nuestro navegador realice múltiples peticiones al servidor sin necesidad de cerrar la conexión cada vez, lo que va a favorecer el tiempo de respuesta. Lo primero que tienes que hacer es comprobar si ya tienes activado Keep-Alive. Para ello puedes utilizar la herramienta [Keep Alive Test](#). En el caso de que no lo tengas activado, inserta la siguiente línea en tu fichero .htaccess:

```
<IfModule mod_headers.c>  
Header set Connection Keep-Alive  
</IfModule>
```

Desactiva las Etags

Las HTTP Etags son unas etiquetas que se asignan a los archivos y se utiliza para validar la caché web: se compara si la etiqueta de cada archivo que se encuentre en el caché del navegador coincide con aquella que está en el servidor. En caso afirmativo, no vuelve a descargar tal archivo. Este proceso de comparación suele ralentizar la carga, problema que podemos solucionar añadiendo la siguiente línea de código en nuestro .htaccess:

```
<ifModule mod_headers.c>  
Header unset ETag  
</ifModule>  
FileETag None
```

Deshabilita el cron WordPress

El cron de WordPress se encarga de lanzar diversos procesos como buscar actualizaciones para WordPress, plugins y temas instalados, publicar artículos en las fechas programadas, enviar pingbacks, etc. El problema reside en que cada vez que un usuario visita nuestro sitio web, se ejecuta el cron. Con lo que si tenemos un sitio con muchas visitas, este hecho puede convertirse en un problema serio de consumo de recursos del servidor. Así que en estos casos es recomendable deshabilitarlo. Para ello tenemos que realizar una serie de pasos:

1. Deshabilita el cron insertando el siguiente código en tu fichero wp-config.php, justo después de la información referente a la base de datos.

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

2. Una vez insertado, debemos acudir a nuestro panel de administración del servidor de alojamiento que hayamos contratado. Buscar la opción Avanzado > Tareas Cron y configurar una nueva tarea CRON. Tendrás que insertar en el campo de Comando lo siguiente:

```
wget -q -O - http://tusitioweb.com/wp-cron.php?doing_wp_cron >/dev/null 2>&1
```

Espero que todas las herramientas, consejos y recomendaciones, algunas más básicas y otras más avanzadas, te ayuden a mejorar el rendimiento de tu sitio web. 3,2, 1, ¡despegamos!