

WPO: La guía definitiva para mejorar la velocidad de tu web

WPO: **Web Performance Optimization**. La manera elegante de decir: “Hacer que una web cargue más rápido” (Ya saben que nos encanta inventar un montón de acrónimos porque sí y que nos encanta “optimizar” todo).

Siempre ha sido importante, si tu web tardaba mucho en cargar vendías menos que si cargaba rápido, pues los visitantes se iban con la competencia. Pero ahora ha tomado mucha fuerza desde que Google anunció que la velocidad de carga sería **una señal más para su(s) algoritmo(s)**.

Y sí es importante que tu web cargue rápido, pero no solo por eso. Ni siquiera principalmente por eso... Terminando esta guía sabrás exactamente que hacer para optimizar la velocidad de carga de tu sitio, garantizado 😊 Y también las verdaderas razones por la que es muy importante.

Primero una advertencia... **Dejémonos de tonterías**.

He visto muuuchos posts que hablan sobre el WPO, incluso algunos que se hacen llamar “guías definitivas”, que mencionan siempre los 3-5 puntos básicos y ya, no explican nada más. En serio, llegamos al punto donde alguien puede leer 3 o más de estos artículos y encontrará la misma información, nada más que con otras palabras.

Y para colmo... se quedará pensando que con optimizar la velocidad de carga de su web **automáticamente será el número 1 en Google!**

Pero no, no es así. Así que antes de decirte como mejorar la velocidad de tu web, te explicaré porque no es tan importante para el posicionamiento como dicen. Prometo que después me meto de lleno en las cosas que debes hacer! Hasta te dejo aquí la tabla de contenidos por si quieres saltarte esta pequeña explicación 😊 (pero te recomiendo que mejor la leas)

¿Crees que si haces que tu sitio web cargue en 3 segundos en lugar de 5

mejorarás tu posicionamiento? Mejor piénsalo otra vez... el impacto no es tan grande como les gusta decir por ahí.

La velocidad de carga de tu web no afecta tanto su posicionamiento

Pruébalo tu mismo con tus sitios. Yo lo he hecho ya con varios sitios y me consta que aunque la velocidad de carga se reduzca a la mitad no por eso ganan un montón de posiciones de la nada.

Primero, **la fuente oficial:**

Aunque la velocidad de un sitio es una nueva señal, no tiene tanto peso como la relevancia de una página. En estos momentos menos del 1% de las búsquedas están siendo afectados por esta señal.

Sí sí... es del 2010 y muy probablemente es algo que afecta ya a más del 1% de las búsquedas... pero también hay información más actual, **del 2013:**

No obtienes ninguna ayuda por tener un sitio rápido. Más bien, sitios atípicos que sean lentos, se posicionarán en puestos inferiores. En igualdad de condiciones, un sitio que es muy lento se posicionará debajo.

O sea que realmente no es un factor tan importante.

Si la página A y la página B tienen la misma probabilidad de posicionarse para una palabra clave X (es decir, están igual de optimizadas) pero B carga más lento que A... es más probable que A se posicione encima de B.

Y ya. Tan tan, eso es todo!

“Pero yo vi *en un estudio que publicaron en Moz* que realmente si importa y que lo más importante es el TTFB (Time To First Byte, o tiempo hasta el primer byte).”

Solo diré dos cosas:

- En el mismo estudio hay un disclaimer, que dice: “Aunque Google no ha sido claro, intencionalmente, acerca de que aspecto concreto de la velocidad de una página impacta en su posicionamiento, han sido muy claros en declarar que la relevancia del contenido es aún el rey. Así que, en otras palabras, aunque podemos demostrar cierta correlación (o falta de) entre algunas métricas de velocidad específicas y el posicionamiento, **nunca podremos probar una relación causal**, pues otros factores que no podemos medir están en juego también”. Pero mucha gente toma ese estudio como LA verdad!
- Cito a Enrico Altavilla, cuya respuesta me parece muy buena y acertada:

*Desafortunadamente, nadie nunca comprueba los resultados de ese tipo de “estudios”. Los lectores ya toman esos resultados como verdaderos, **creyendo que si un sitio es popular es garantía de que sus resultados son correctos.** En este caso en específico, si analizas los datos duros usados para ese estudio en Moz (los cuales publicaron para que las demás personas pudieran checar los resultados) resulta que muchas de las búsquedas utilizadas para esa prueba tenían a Wikipedia, Amazon y otros sitios con mucha autoridad en las primeras posiciones. Dado que esas compañías tienen servidores y una infraestructura tecnológica muy rápidos, su TTFB es bastante bajo. ¡En algunos casos es incluso cero! Como consecuencia, esos pocos sitios con TTFB muy bajos, casi siempre presentes en los primeros 5 resultados, influyen esa correlación, dando la impresión de que*

muchos sitios presentan estas características. Cuando en realidad son solo unos pocos sitios que bajan el promedio.

Enrico
Altavilla

No hay más que agregar.

Pero en fin... a pesar de que no afecta tanto el posicionamiento **sí que es importante que tu sitio cargue rápido**. Y te daré 3 razones. Un sitio que carga rápido:

- Vende más que un sitio que carga lento.
- Tiene más probabilidades de tener un porcentaje de rebote bajo. Si el sitio tarda mucho en cargar los visitantes simplemente se van.
- Tiene más visitantes recurrentes que uno lento.

Sobre todo si tomamos en cuenta que cada vez los usuarios son menos pacientes... esperan que un sitio cargue cada vez más rápido.

Pero en fin! Basta de teoría, ahora sí vamos a ver como optimizar la velocidad de carga de un sitio web 😊

Primer Paso: Mediciones iniciales



Foto cortesía de [bradhoc](#)

Lo primero que debes hacer para poder realizar un buen trabajo de WPO es tomar algunas mediciones iniciales antes de optimizar tu sitio. Además, algunas de estas herramientas te dirán qué cambios necesitas hacer porque probablemente no todo en esta guía te sirva.

Las 3 herramientas principales que yo utilizo para esto son:

- [GTMetrix](#)
- [Pingdom](#)
- [WebPage Test](#)

Toma un Excel y toma tu página inicial, analízala con estas tres herramientas y anota los resultados. Después toma algunas páginas importantes de tu sitio y analízalas también! **No solo te enfoques a la página de inicio.**

Por ejemplo, si tienes una tienda online puedes tomar una página de producto, una categoría y una página de tu blog para analizarlas y guardar los resultados también.

Este primer paso es muy sencillo, pero es básico. ¿Cómo sabrás si sirvieron todas tus optimizaciones si no sabes con qué comparar?

Segundo Paso: Optimizando la velocidad de carga

Aquí ya entramos de lleno en materia... Aquí tienes **más de 10 formas en las que puedes mejorar la velocidad de tu web**. Las he dividido de acuerdo a su importancia para que te sea más fácil navegarlas. En algunas he incluido plugins para WordPress que facilitan la tarea y también el código necesario para hacerla manualmente.

Eso sí, hay que tener en cuenta que no todas van a aplicar a tu sitio y habrá algunas que no puedas hacer.

En fin! Vamos con estos tips de WPO:

Importancia Alta

Activa la compresión GZip

Esta es la primera de las tareas que pueden hacerse en WordPress utilizando el plugin **W3 Total Cache** (de aquí en adelante será el plugin W3C). El problema es que es un poco “complicado” de configurar si no tienes idea de lo que estás haciendo. Así que **te dejo con este practico tutorial** (y al grano) de Andres Ledo sobre **como configurar el plugin W3C** 😊

Ahora vamos con la segunda opción, la manual. Si estás en apache (como la mayoría) solo tienes que poner este código en el archivo .htaccess y listo:

```
<ifmodule mod_deflate.c>
```

```
AddOutputFilterByType DEFLATE text/text text/html text/plain te
```

```
xt/xml text/css application/x-javascript application/javascript text
/javascript
</ifmodule>
```

Caché de navegador

Aprovechar la caché del navegador en servidores apache también es muy sencillo. De nuevo en el .htaccess, solo pon este código:

```
ExpiresActive On

ExpiresDefault A0
<FilesMatch “\.(pdf|ppt|doc)$”>
ExpiresDefault A9030400
</FilesMatch>
<FilesMatch “\.(jpg|jpeg|png|gif)$”>
ExpiresDefault A2592000
</FilesMatch>
<FilesMatch “\.(txt|xml|js|css|htm|html|php)$”>
ExpiresDefault A604800
</FilesMatch>
```

En ExpiresDefault, el tiempo después de la A se indica en segundos. 604800 corresponde a 7 días, por ejemplo (60 segundos*60 minutos*24 horas*7 días).

Esto igual puede configurarse con el plugin W3C.

Cargar JavaScript al final o asíncronamente

¿Por qué? Fácil. La carga y ejecución de Javascript pausa la renderización de la página (osea, la visualización) y si cargas mucho código puede que tus usuarios tarden varios segundos en ver la página. **Y se irán.**

Por eso conviene moverlos al final de tu página, así primero carga todo el texto y las imágenes de tu sitio y hasta después se cargará y ejecutará el

código Javascript. Y si no puedes ponerlos al final, de menos cargarlos asíncronamente.

Si es un sitio hecho “a mano” entonces esto es muy sencillo, solo basta con mover las etiquetas `<script></script>` antes de `</body>` y listo. Sin embargo, si usas algún CMS **la cosa se complica un poquito**. Así que aquí te explicaré tres cosas: Como cargar al final tus scripts en sitios que no usen un CMS, como aplazar la carga de tus scripts en un sitio con Wordpress y como cargarlos asíncronamente.

1 – Como cargar al final tus scripts si no estás usando un CMS

Para esto basta con añadir un poco de código Javascript dentro de `<head></head>`, antes de que se carguen todos los demás scripts. El código en cuestión es:

```
function cargarScript(src) {  
  var element = document.createElement("script");  
  element.src = src;  
  document.body.appendChild(element);  
}
```

```
function descargarScripts {  
  
  cargarScript("ruta/al/script1.js");  
  
  cargarScript("ruta/al/script2.js");  
  
  cargarScript("ruta/al/script3.js");  
  
}
```

```
if (window.addEventListener)
```



```
window.addEventListener("load", descargarScripts , false);

else if (window.attachEvent)

window.attachEvent("onload", descargarScripts );

else

window.onload = descargarScripts;
```

Una alternativa más sencilla es añadir el atributo defer en el script, quedando algo así:

```
<script src="ruta/al/script.js" defer="defer"></script>
```

Pero solo está soportado por IE de la versión 10 para arriba. Ah.. ese Internet Explorer.

2 – Como cargar el javascript al final si estás usando WordPress.

Primero, de la manera sencilla. Utilizando un plugin como [este](#).

Después, la manera un poco más avanzada, pero mejor, pues al no instalar otro plugin no cargas más tu sitio. Solo tienes que editar el archivo functions.php y añadir el siguiente código:

```
function aplazar_carga_js( $url ) {
if ( FALSE === strpos( $url, '.js' ) ) return $url;
if ( strpos( $url, 'jquery.js' ) ) return $url;
// return "$url' defer ";
return "$url' defer onload=";
}

add_filter( 'clean_url', 'defer_parsing_of_js', 11, 1 );
```

Al ser PHP debe estar entre las etiquetas<?php y ?>.

Y listo 😊

3 – Como cargar el javascript asíncronamente en WordPress.

Lo mismo, puede ser de forma fácil usando un plugin como [este](#).

Y la forma más recomendada, aunque parece un poquito más complicada, es meter mano al archivo functions.php. **Con copiar y pegar este código ya la hicieron** 😊

```
function comenzar_filtrado() {  
    ob_start();  
}
```

```
function reemplazar() {  
    $aux = ob_get_clean();  
    $aux = str_replace("src='%%'", "async='async' src=", $script_out);  
    print $aux;  
}
```

```
function cargar_scripts($url) {  
    $scripts= array (  
        'http://tusitio.com/ruta/script/script1.js',  
        'http://tusitio.com/ruta/script/script2.js',  
        'http://tusitio.com/ruta/script/script3.js'  
    );
```

```
    foreach($scripts as $script)  
        if (substr($url, 0, strlen($script)) === $script)  
            return "%%" . $url;  
    return $url;  
}
```

```
add_filter( 'clean_url', 'cargar_scripts', 11, 1);
```

```
add_action( 'wp_print_scripts', 'comenzar_filtrado');  
add_action( 'print_head_scripts', 'reemplazar');  
add_action( 'print_footer_scripts', 'reemplazar');
```

En la función “cargar_scripts” solo cambia los que puse de prueba por los scripts que quieras cargar de manera asíncrona y ese código lo que hará será agregarles el atributo async para que el navegador los cargue asíncronamente. Eso sí, **no todos los navegadores lo soportan** (Internet Explorer solo de la versión 10 en adelante).



Una advertencia muy importante al aplazar la carga de Javascript o volverla asíncrona:

Es muy probable que se fastidie tu sitio si aplicas esto a todos los scripts que tengas, sobre todo a JQuery. Vas a tener que experimentar para ver que scripts si puedes aplazar, o poner

asíncronos, sin romper nada. Esto es más común si tu sitio está usando algún CMS!

Así que cada que hagas algún cambio, **mejor revisa tu web para ver si no rompiste algo**. De preferencia en modo incógnito y cargando con Ctrl+F5 para cargar sin utilizar tu caché. Me pasó en este blog que después de hacer estos cambios ninguna imagen cargaba y yo no me había dado cuenta... olvide revisar sin utilizar la caché de mi navegador :S

Menos mal que alguien me avisó!

Usa un CDN

Este paso es tan sencillo, y mejora tanto el rendimiento de un sitio que me sorprende que **son pocas las personas que lo implementan**.

Pero primero, **¿Qué es un CDN?** Definición rápida: una CDN es un conjunto de servidores ubicados en distintas partes del mundo que guardan una copia de algunos contenidos de tu web (documentos, videos, imágenes, etc).

Cuando una persona accede a tu web se le envían los contenidos desde el servidor más cercano a su ubicación. **Así el sitio carga más rápido y evitas la sobrecarga de tu servidor.**

Usar un CDN es muy sencillo, y si apenas vas empezando puedes utilizar el plan gratuito de CloudFlare. Explicar como configurar alargaría mucho este post, así que te dejo con este artículo de Andres Ledo (sí, otra vez) donde te enseña **como configurar CloudFlare**. Lo único que te recomiendo, es que al llegar a la opción de “Rocket Loader” verifiques que no se haya roto nada en tu sitio! A mi me causó muchos problemas y termine deshabilitando esa opción.

Y si estás utilizando algún plugin para comprimir archivos, entonces no actives las opciones “Auto Minify”.

Carga las imágenes con Lazy Loading

Cargar imágenes usando la técnica de Lazy Loading puede hacer que un sitio con muchas imágenes en cada post **cargue muchísimo más rápido**. Esta técnica consiste en que la imagen se carga solo hasta que el usuario hace scroll hasta la imagen, no al momento de cargar la página.

Y es bastante sencilla de aplicar, te diré dos alternativas: Con algún plugin en WordPress o a manita.

Primero con plugin. He probado con varios plugins y el que mejor ha funcionado es este: **Image Lazy Load**. **Solo se instala, se activa y listo!** No hay nada más que hacer. Sencillo. Ahora, si no estás usando WordPress o no quieres cargar tu sitio con otro plugin más, está la alternativa manual.

Primero hay que descargar este script: **Unveil.js** y añadirlo en todas tus páginas. Después hay que modificar el código de cada imagen, poniendo algo así:

```

```

En el src en lugar de la imagen original, ponemos un “placeholder”. Una imagen temporal que cargará inicialmente en lugar de nuestra imagen. Basta con que sea una imagen minúscula, de 1×1 si quieren... **no tendría caso cargar una imagen grande si lo que queremos es cargar la página más rápido!**

Y para terminar, el siguiente código:

```
$(document).ready(function() {  
  $("img").unveil();  
});
```

(Obviamente hay que tener JQuery!)

Importancia Media

Comprimir CSS, JS y HTML

De nuevo, la solución manual y usando WordPress.

Si estás en WordPress puedes hacer esto con el mismo plugin, W3C.

Y si lo quieres hacer manual, solo tienes que usar alguna herramienta **como esta**. Bastante completa, te permite comprimir HTML, CSS, Javascript. Solo

pega el código que quieras comprimir (o sube el archivo), reemplaza el código original por el código comprimido y listo.

No olvides hacer un respaldo primero, sobre todo en Javascript. A veces deja de funcionar el código cuando se comprime!

Combinar ficheros

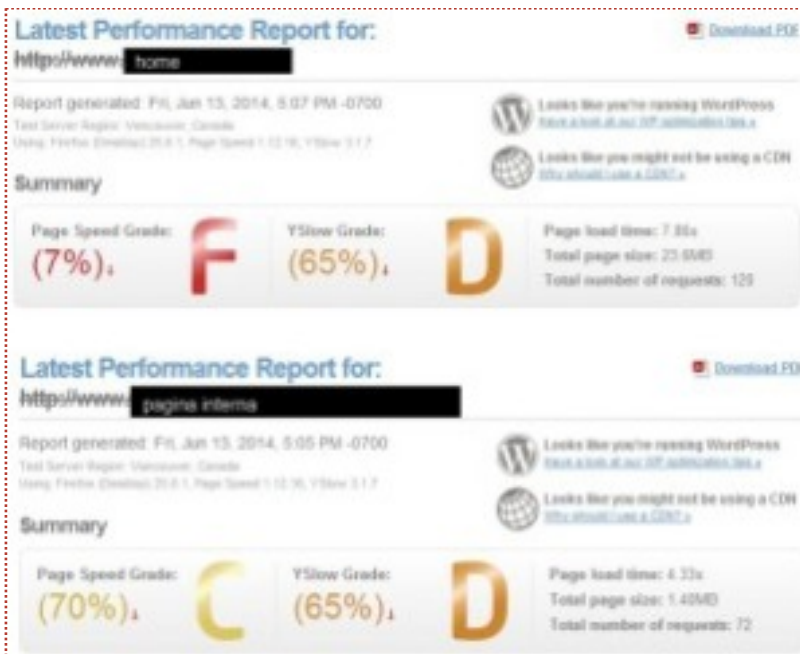
De nuevo, esto lo haces fácilmente con el W3C. Y si lo quieres hacer manual, es algo muy sencillo también. Solo debes tomar varios de tus archivos CSS y JS y combinarlos en uno solo.

De preferencia, combina todos los que pesen poco y algunos de los demás archivos. Después, solo manda llamar tu CSS o tu JS combinado en lugar de todos los demás. Y de paso comprímelos como te enseñé en el punto anterior!

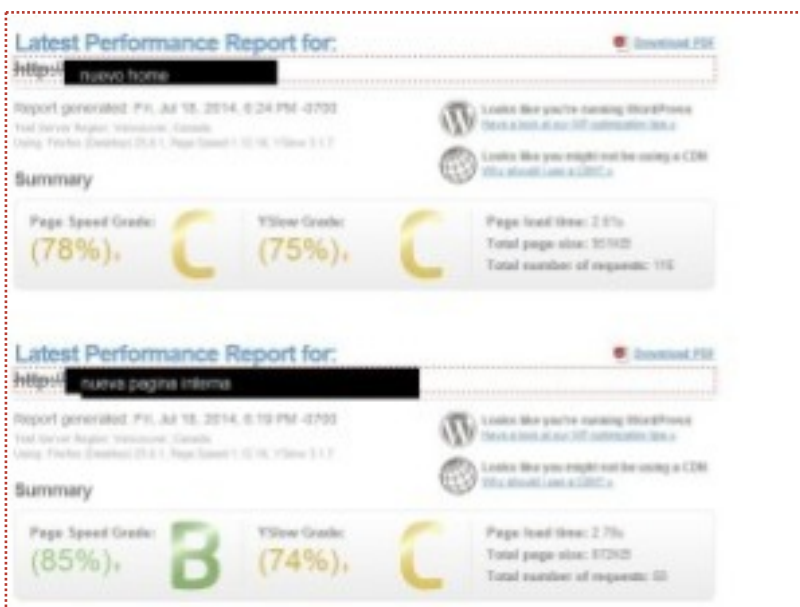
Comprimir imágenes

Esto es muy simple, antes de subir una imagen y usarla en tu sitio, pasala por [Smush.it](#). Así la comprimirás, haciendo que pese menos sin que pierda calidad. O si tienes WordPress, un plugin [como este](#) puede servirte 😊

En algunas páginas solo comprimir imágenes puede resultar en una mejora drástica de la velocidad. Aquí un ejemplo de un sitio con el que trabaje:



Antes de comprimir imágenes



Después de

Combina imágenes en Sprites

Sprites: Una imagen que se compone de un conjunto de imágenes agrupadas. Así en lugar de cargar varias imágenes cargas una sola y manipulas la posición de la imagen para que se muestre una imagen en específico. Para no hacer maaas largo este post, les dejo un [tutorial sobre CSS Sprites](#) muy bueno y sencillo que encontré por ahí 😊

Importancia Baja

Estas mejoras no tendrán un impacto significativo en el tiempo de carga de tu sitio, pero **al final todo suma** 😊 así que conviene hacerlas también

Optimiza la base de datos de tu sitio en WordPress

Esto se puede hacer fácilmente **con este plugin**. Primero respalda tu base de datos, después activa el plugin y limpia tu BD. Con este plugin igual puedes ejecutar comandos avanzados de limpieza directamente en la BD... **pero no recomiendo que te metas con eso al menos que sepas realmente lo que haces.**

Si después de limpiar tu base de datos tu sitio sigue funcionando bien, ya terminaste. Si hay algún problema tendrás que restaurar tu base de datos a la versión anterior...

Carga scripts comunes desde el CDN de Google

Primero la opción fácil y rápida para WordPress: **Use Google Libraries**. Instala, activa y ya terminaste.

Si lo harás manual, revisa que librerías puedes cargar desde Google **aquí**.

Tiene varias de las más comunes: JQuery, Script.aculo.us, SWFObject. Y al momento de llamarlas en lugar de cargar tu copia local, solo pon la URL que Google te facilita 😊

No redimensiones las imágenes desde HTML

Esto es fácil de implementar en sitios “tradicionales”. Pero si usas un CMS como WordPress, Drupal, Joomla y demás, se vuelve un poco más complicado...

Muchas veces las plantillas que usamos redimensionan imágenes desde HTML. Es decir, ponen algo como esto:


```

```

Y resulta que imagen.jpg mide 1500 x 2500!

Eso hace que cargues una imagen muy pesada y al final terminas mostrando una más pequeña. El objetivo es hacer mejor algo así:

```

```

Y así ya cargas una versión redimensionada de imagen.jpg que pesa mucho menos. Haciendo que la página sea más ligera y que cargue más rápido.

Tercer Paso: Comprueba los cambios

El tercer paso consiste simplemente en repetir el primero. En verdad!

Mide de nuevo las páginas elegidas al inicio y **compara los resultados de su velocidad**. Dependiendo de tantas fallas hayas tenido al inicio puedes experimentar una mejora considerable de velocidad en tu sitio.

Y una última cosa más... no olvides revisar tu sitio para ver si los cambios que has hecho no lo han dejado inservible. Sobre todo si le moviste algo al Javascript. De nuevo: Verifícalo en modo incógnito y sin usar tu caché (Ctrl + F5). Incluso puedes ir un paso más lejos y comprobarlo con algún proxy como el de Hide My Ass.

Extra: Sobre las plantillas 'optimizadas para SEO'

No pude evitar hacer un comentario cerca de estas plantillas que he visto que venden mucho, **según optimizadas para SEO**. Normalmente lo que quieren

decir es que cargan rápido, un cierto porcentaje en su **PageSpeed**, y ya.

El problema es “como” lo hacen. Normalmente no tienen mucho código Javascript ejecutándose, tienen pocos estilos (incluso a veces los tienen embebidos directamente en el HTML) y algunas de las pocas imágenes que suelen usar (sobre todo las pequeñas, como íconos) las tienen codificadas directamente en base64.

Ejemplo de una imagen codificada:

```
/>
```

Así se ahorran el tener que cargarla de manera externa. ¿Desventajas de poner así tus imágenes?

- Es un proceso engorroso tener que estar convirtiendo las imágenes. Aunque si sabes algún lenguaje como PHP puedes automatizarlo.
- Para algunas imágenes el resultado en texto es enorme. Por ejemplo, para esta imagen de 50×50



La codificación da como resultado más de 41000 caracteres!

- ¿Y si quieres modificar la imagen? Tienes que pasarla de la codificación a imagen, después editarla, volverla a codificar y reemplazar el código anterior.

Mucho trabajo.

Y como dije al inicio, el impacto de la velocidad de carga en el posicionamiento no es tan grande como muchos quieren creer. Entonces, estas plantillas de “SEO” no tienen mucho eh. Toda esa “velocidad”, ese 99% en el PageSpeed de Google, **se va a caer cuando comiences a meterle plugins, contenido, imágenes y demás.** Así que no te vayas con la finta creyendo que ya con esa plantilla “tendrás un mejor SEO”. Ya otra cosa es que tengan en promedio un buen CTR por ejemplo, ahí si puede ayudarte mucho, pero eso ya no es SEO.

Conclusión

Pfff **al fin terminaste con este tutorial..** ahora ya sabes porque es importante tener una web que cargue rápido (y porque no es solo el SEO, ni principalmente) y también has aprendido (espero!) a realizar las mejoras necesarias en tu sitio para optimizar su velocidad de carga, ya sea que uses WordPress o no.