

Guía rápida con

Las bases del desarrollo de temas WordPress

Por Francisco Aguilera G.



Introducción

Hola, gracias por descargar esta guía.

Mi nombre es Francisco Aguilera, soy el autor del blog FranciscoAMK.com, y mi objetivo en este documento es enseñarte los puntos esenciales que deberías conocer antes de comenzar en el desarrollo de temas de WordPress, además de derribar algunos mitos que suelen asustar a aquellos que tienen interés de comenzar en este rubro.

Sin más *¡Vamos a ello!*

Lección 1

Por qué aprender a crear temas WordPress

Hoy en día crear un sitio web es más fácil que nunca, es solo cosa de contratar un hosting, instalar WordPress y comenzar a publicar contenido. Mientras que para definir su diseño, basta con conseguir un tema atractivo y configurarlo de acuerdo a tus necesidades.

¿Pero qué pasa cuando quieres pasar al siguiente nivel? ¿Cuando quieres que tu sitio web sea único? ¿Cuando quieres implementar un diseño propio que probablemente tú mismo o alguien dentro de tu equipo diseñó?

En esos casos, lo más probable es que un tema comprado, por muchas opciones que tenga, no te permita llegar a igualar al 100% esa idea que tienes en mente.

Ni hablar si estás usando un tema gratuito.

Ya sea que estés trabajando en un sitio propio o un sitio para un cliente, siempre será mucho más atractiva la posibilidad de tener algo único, con ese toque propio y saber que nadie más allí fuera tiene lo mismo.

Lo más común (y que probablemente a ti también te haya pasado) es verse maravillado por la versión demo de un tema, sin prever que éste luego vendría con cientos de archivos en un lenguaje difícil de entender para los “*no-desarrolladores*”, con un panel de opciones eterno y con todas las líneas de texto relevantes en inglés.

En un tema de este tipo, realizar un cambio aparentemente sencillo, como modificar una parte de la estructura HTML, se puede convertir en una verdadera pesadilla. Incluso usando un tema hijo.

Mientras que si el tema fuera tuyo, sabrías inmediatamente dónde ir y dónde intervenir para conseguir el resultado esperado.

Cuando el tema que usas en tu sitio lo desarrollas por tu propia cuenta, no solamente puedes controlar en un 100% el aspecto visual del sitio, sino que también conoces por completo el código fuente de éste, lo cual hará tu trabajo mucho más fácil en caso de que decidas hacer cambios a futuro, ya sea a nivel estético, estructural e incluso funcional.

A mí me ha tocado trabajar con varios sitios de clientes, resolviendo problemas que se hubieran ahorrado por completo si es que hubiesen desarrollado un tema propio desde un inicio.

Cuando sabes cómo crear tus propios temas de WordPress puedes tomar ventaja de esta capacidad de tres formas:

Primero

Podrás ofrecer a tus clientes (o utilizar para tu propio sitio) las bondades y la autenticidad de tus diseños exclusivos, lo cual sin duda alguna es un valor agregado en cualquier sitio y te permitirá aumentar la tarifa que cobras.

Segundo

Podrás diseñar y desarrollar temas propios para ofrecer de manera masiva, en alguno de los marketplaces existentes como themeforest o creativemarket y si lo deseas, también podrás crear tu propia tienda de temas en tu sitio.

Tercero

Si eventualmente te encontraras con la necesidad de modificar un tema desarrollado por un tercero, tendrás mayor conocimiento de los archivos de plantilla, de cómo funcionan y de cómo poder modificarlos.

En resumen

Aprender a crear tus propios temas puede resultarte de gran utilidad ya que te facilitará varias tareas pero sobre todo: Te permitirá elevar tus tarifas ya que estarás entregando mayor exclusividad a tus clientes y eventualmente podrás duplicar tu sueldo [tal como yo lo hice](#).

Lección 2

La estructura de archivos de un tema

Cuando descargas un tema de WordPress y miras dentro de la carpeta, seguramente te encontrarás con una enorme cantidad de archivos, lo cual puede ser verdaderamente intimidante si tienes la intención de aprender a crear tus propios temas.

Pero no debes dejar que esto te asuste, porque para poder funcionar correctamente, un tema de WordPress sólo necesita de 2 archivos: *index.php* y *style.css*.

Archivos básicos

Style.css es una hoja de estilos, por lo tanto, tendrá la misión de aplicar diseño al tema. Por ejemplo, desde este archivo vamos a poder definir cuál es el color de fondo del tema, el color de los enlaces, el tipo de letra que se va a usar, el tamaño en el que se van a mostrar las imágenes, etc.

Pero además, este archivo es el encargado de entregar a WordPress la información necesaria sobre el tema para que lo pueda identificar y presentar en

el área de administración, con un título, una descripción y la información del autor.

El otro archivo, tiene la misión de estructurar el contenido en la parte frontal del sitio, es decir, gracias a *index.php*, vamos a poder definir la estructura HTML dentro de la cual se van a mostrar los contenidos del sitio.

Con estos 2 archivos, WordPress ya puede reconocer y activar un tema, y éste puede funcionar bastante bien.

Es una buena idea también incluir un archivo llamado *screenshot.png*, el cual es una imagen de 600 x 450 píxeles que permitirá ver una miniatura del tema antes de activarlo desde el área de administración.



Jerarquía de plantillas

Dentro del contexto de los temas de WordPress existe un concepto llamado "jerarquía de plantillas", lo cual significa que dependiendo del tipo de conte-

nidos que corresponda mostrar en el navegador, WordPress buscará dentro de la carpeta de nuestro tema el archivo indicado para cada ocasión.

Si es que alguno de los archivos de plantilla que WordPress busca no está disponible, siempre llegará a *index.php*.

Nosotros podemos tomar ventaja de la jerarquía de plantillas creando diferentes archivos para presentar los contenidos de forma diferente en ciertas circunstancias.

Lo más común es desarrollar un archivo de plantilla llamado *single.php*, el cual será el encargado de presentar la pantalla de detalle de los artículos, y también es muy frecuente desarrollar el archivo *page.php*, el cual se encargará de definir la estructura que se utilizará al presentar las páginas estáticas del sitio.

Hay una serie de otros archivos que se puede desarrollar, como por ejemplo *category.php*, que aplica al listado de artículos dentro de una categoría o *front-page.php* que aplica a la página de inicio del sitio.

Si quieres saber más respecto a la jerarquía o conocer más respecto a los archivos de plantilla deberías mirar [este diagrama](#) del codex de WordPress que lo grafica muy bien.

Archivos parciales

En todos los archivos PHP que han sido nombrado hasta ahora, hay elementos que se repiten en cada uno de ellos, como por ejemplo: la cabecera, el pie de página o la barra lateral (si es que el diseño lo contempla así).

No hay necesidad de repetir estos elementos en cada uno de los archivos de plantilla, en su lugar se pueden crear archivos parciales para cada uno de estas partes.

Lo más frecuente es tener dentro de la carpeta de un tema los archivos:

- ***header.php***: Que contiene la cabecera del tema
- ***footer.php***: Que contiene el pie de página del tema
- ***sidebar.php***: Que contiene la barra lateral

De este modo, desde los archivos de plantilla de nuestro tema sólo deberemos hacer una llamada a estos archivos usando las funciones que WordPress nos provee que son respectivamente:

- `get_header();`
- `get_footer();`
- `get_sidebar();`

Otro archivo muy relevante en los temas de WordPress es *[functions.php](#)*, el cual nos va a permitir desarrollar diferentes funciones para aplicar dentro de nuestro tema.

En resumen

Como puedes ver, para crear un tema de WordPress no hace falta desarrollar una gran cantidad de archivos. Eventualmente podríamos usar solamente `index.php` y `style.css`, pero para darle algo de mayor versatilidad, lo más frecuente es que un tema tenga una estructura de archivo similar a lo siguiente:

- *header.php*
- *footer.php*
- *sidebar.php*
- *index.php*
- *page.php*
- *single.php*
- *front-page.php*
- *screenshot.png*

- *functions.php*
- *style.css*

Los archivos adicionales que decidas incluir en tu tema van a depender del diseño y complejidad que le hayas dado, y para decidir qué nombre poner a estos archivos, debes revisar la página sobre la jerarquía de plantillas del `codex`, de este modo sabrás cómo nombrarlos y en qué momento WordPress acudirá a ellos.

Lección 3

Cómo funciona WordPress

Una de las principales características de los sistemas de gestión de contenidos web, en especial de WordPress es que separa completamente el contenido del sitio de su diseño.

Dentro de este contexto los Temas cumplen la función de definir el aspecto visual que los sitios tendrán.

Aunque en realidad es más que eso

Los temas serán los encargados de establecer una estructura HTML para el sitio y decidir en qué parte de esta estructura y de qué manera se presentarán los contenidos.

Es por esto que WordPress no puede funcionar si no tiene al menos un tema instalado y activo.

También hay características específicas de WordPress que necesitan ser habilitadas desde el tema activo para poder funcionar como por ejemplo los menús dinámicos, las imágenes destacadas, las zonas para usar widgets o los formatos de entrada.

Todas estas funcionalidades se habilitan desde el archivo de funciones del tema, llamado *functions.php*.

Desde él podemos además, definir algunas características del sitio, cargar archivos CSS y Javascript y eventualmente crear cualquier tipo de función personalizada.

Sin embargo debemos ser cautos de sólo desarrollar funciones que tengan relación con la apariencia del sitio. Para otro tipo de funcionalidades existen los plugins.

Hace un tiempo escribí en mi blog un artículo dedicado a los [alcances del archivo *functions.php*](#) que te recomiendo visitar.

El loop

Ahora bien, volviendo a los aspectos más básicos de un tema de WordPress, nos encontraremos con la parte esencial de todo tema: “*El Loop*”.

Podemos entender el loop como el código que usamos en nuestros archivos de plantilla que llamará al contenido que corresponda mostrar en cada pantalla del sitio.

El loop luce más o menos así:

```
<?php if ( have_posts() ) : while ( have_posts() ) : the_post(); ?>
    El contenido
<?php endwhile; else:?>
    No hay contenidos
<?php endif; ?>
```

El loop se compone de 3 partes,

La primera parte se encarga de verificar si hay posts para mostrar en la página, si no los hay, nos permitirá mostrar un mensaje al visitante para dejarle saber que no hay contenidos disponibles:

```
<?php if ( have_posts() ) : while ( have_posts() ) : the_post(); ?>
    El contenido
<?php endwhile; else:?>
    No hay contenidos
<?php endif; ?>
```

Si es que hay contenidos para mostrar, el loop pasa a la segunda parte en donde revisa todos los post que correspondan a esta pantalla y por cada uno de ellos, pasa a la tercera parte.

```
<?php if ( have_posts() ) : while ( have_posts() ) : the_post(); ?>
    El contenido
<?php endwhile; else:?>
    No hay contenidos
<?php endif; ?>
```

Esta tercera parte consiste en traer la información del post y mostrarla en pantalla de acuerdo a la estructura que nosotros hayamos definido.

```
<?php if ( have_posts() ) : while ( have_posts() ) : the_post(); ?>
    El contenido
<?php endwhile; else:??>
    No hay contenidos
<?php endif; ?>
```

En este ejemplo, podemos ver como por cada post, se mostrará un div con la clase “articulo” y dentro de él se mostrará el título dentro de un elemento h2.

```
<?php if ( have_posts() ) : while ( have_posts() ) : the_post(); ?>
    <div class="articulo">
        <h2><?php the_title();?></h2>
    </div>
<?php endwhile; else:??>
    No hay contenidos
<?php endif; ?>
```

En resumen

Como pudiste apreciar en el ejemplo, implementar el loop en un tema es bastante sencillo, pero si aún así te parece algo complejo, en el próximo capítulo te mostraré por qué NO necesitas saber programación para comenzar a crear temas de WordPress.

Lección 4

No necesitas saber programar

Pese a que efectivamente sería una gran ayuda, no es necesario tener conocimientos avanzados de programación para comenzar a crear tus propios temas de WordPress.

Lo que más se necesitará son habilidades en cuanto a HTML y CSS, incluso un poco de jQuery estaría bien, pero más allá de eso, saber programación del lado del servidor no es un requisito para comenzar.

Si bien WordPress y sus temas están desarrollados en lenguaje PHP, las funciones que con mayor frecuencia tendrás que usar son bastante intuitivas.

Por ejemplo, cuando quieras mostrar el título de un post deberás usar la función ***the_title()***, mientras que cuando quieras mostrar el contenido tendrás que usar la función ***the_content()***.

Cuando digo que no necesitas saber programación no quiero decir que no deberías aprender, de hecho, es algo que absolutamente te recomendaría, porque en el futuro te puedes encontrar con situaciones en donde no haya una función nativa de WordPress que te permita solucionar un problema y

te veas en la necesidad de desarrollar algo de mediana o baja complejidad por tu cuenta.

Tampoco quiero decir que con WordPress los desarrolladores ya no sean necesarios. Su trabajo es absolutamente fundamental, pero creo que contratar a un desarrollador de alta capacidad sólo para mostrar el título de un post seguramente sería un trabajo muy aburrido ya que no plantea ningún desafío.

Mientras que si se delegara a un desarrollador la labor de crear un plugin específico para un sitio con necesidades especiales, sin duda sería algo mucho más interesante.

WordPress es un sistema que está muy bien documentado, en el codex podrás encontrar explicaciones a casi todas las funciones que necesitarás ocupar y si alguna de ellas no está bien documentada, la comunidad se encargará de ayudarte en los foros de soporte.

¿Quieres seguir aprendiendo?

Con las 4 lecciones recibidas hasta ahora ya tienes una base sólida para poder entender y disfrutar de la vasta gama de videos, tutoriales y artículos publicados a lo largo de todo internet.

Sin embargo, si quieres obtener resultados rápidos y aprender a desenvolverte como un profesional en el desarrollo de temas y sitios con WordPress lo mejor que puedes hacer es tomar el curso: “[Creación de temas WordPress Profesionales](#)”.

En este curso aprenderás a desarrollar temas de WordPress completamente funcionales con calidad profesional, con videos paso a paso, acceso a los códigos fuente y con la posibilidad de realizar preguntas en cada clase.

[*Ver temario y precios »*](#)

Yo seré el instructor de este curso y si tienes cualquier duda al respecto, solo escríbeme a francisco@apkestudio.com y te responderé encantado.

Espero verte dentro del curso ;)

Un abrazo,

@FranciscoAMK