# wordpressAprende a crear un tema de WordPress desde cero

Contenido

[Cómo crear un tema de WordPress desde cero 1](#_Toc504982076)

[Archivos de un tema de WordPress 2](#_Toc504982077)

[Paso 1: Creamos el archivo style.css 3](#_Toc504982078)

[Paso 2: Imágenes y JavaScript 3](#_Toc504982079)

[Paso 3: Creamos la cabecera del tema 3](#_Toc504982080)

[Paso 4: Creamos la barra lateral 6](#_Toc504982081)

[Paso 5: Creamos el footer 7](#_Toc504982082)

[Paso 6: Creamos la página principal 7](#_Toc504982083)

[Paso 7. El loop de WordPress 8](#_Toc504982084)

[Paso 8. Añadir thumbnails al loop 10](#_Toc504982085)

[Paso 9. Crear un campo de búsqueda 11](#_Toc504982086)

[Paso 10.Configurar la página que muestra la entrada 12](#_Toc504982087)

[Paso 11. Crear una nueva plantilla 15](#_Toc504982088)

[Paso 12: Página de categorías y etiquetas 16](#_Toc504982089)

[Para acabar 16](#_Toc504982090)

WordPress es un sistema gestor de contenidos muy potente con el que podemos crear sitios web muy potentes y versátiles. Todavía hay quien piensa que WordPress es sólo una plataforma para crear blogs, pero nada más lejos de la realidad, con WordPress podemos crear verdaderos portales web que incluyan incluso una tienda online.

Existen muchos sitios web donde adquirir [plantillas de WordPress premium](http://www.esandra.com/out/themeforest) de gran valor añadido, sin ir más lejos aquí he recopilado [50 de las mejores plantillas de WordPress](http://www.esandra.com/50-plantillas-premium-de-wordpress-que-no-te-puedes-perder). Pero a veces queremos ir un paso más allá y desarrollar nuestro propio tema, y las primeras veces tenemos la tendencia a tirar de [un framework como Thematic](http://www.esandra.com/wordpress-crea-tu-propio-tema-con-thematic) o de un tema básico pensado para desarrollo como [Bones](http://themble.com/bones/).

### Cómo crear un tema de WordPress desde cero

Para poder seguir este tutorial, vamos a necesitar tener WordPress instalado.

Suponiendo pues que tenemos WordPress instalado y en correcto funcionamiento, lo primero que tendremos que hacer es ir a la carpeta que contiene los temas de WordPress. Esta carpeta se encuentra en las siguiente ruta: carpetawordpress > wp-content > themes, siendo carpetawordpress la carpeta donde tenemos instalado WordPress. Dentro de themes veremos las carpetas de los temas de WordPress que contiene nuestra instalación. Crearemos una nueva carpeta con el nombre de nuestro nuevo tema.

Para este tutorial nuestro nuevo tema se llamará mitema, con lo que dentro de themes crearemos la carpeta mitema.

### Archivos de un tema de WordPress

Tal y como veremos, WordPress funciona por módulos, así por ejemplo para insertar el footer en una página habremos de llamar al archivo footer.php, en caso que exista. Y lo mismo con el header y la barra lateral del tema.

Asi, un tema de WordPress puede estar formado por muchos archivos, **pero para que un tema de WordPress funcione, sólo necesitará dos archivos: index.php y style.css**. Está claro que con sólo estos dos archivos tendremos un tema muy básico y de ir por casa, pero a final de cuentas, será considerado por WordPress un tema válido si estos dos archivos están bien configurados.

Con esto estamos también diciendo que para que un tema de WordPress sea reconocido por WordPress deberá incluir sí o sí estos dos archivos index.php y style.css. Con que falte la hoja de estilos o el archivo index.php WordPress no reconocerá la carpeta mitema como un tema de WordPress.

Aquí vamos a desarrollar un tema más completo que nos sirva de base para otros proyectos. Y raramente crearemos un tema de WordPress que no contenga una cabecera, un footer y una barra lateral, por no decir un buscador interno. Tampoco vamos aquí a desarrollar un tema complejo con Custom Post Types y funciones avanzadas, pero sí que queremos tener lo necesario para crear un tema decente con el que poder desarrollar proyectos más complejos.

Así, vamos a desarrollar un tema con la mayoría de archivos y carpetas y serán los siguientes:

* **/images**. Carpeta con las [imágenes](https://graphicriver.net/?ref=creativasfera&clickthrough_id=1179642349&redirect_back=true) del tema.
* **/js.** Carpeta con los archivos de JavaScript
* **style.css**. Hoja de estilos del tema. Obligatoria para que el tema funcione.
* **index.php**. Obligatorio para que el tema funcione. Por defecto será la página principal.
* **screenshot.png**. Muestra la imagen en miniatura que se verá en el panel de adminsitración en Apariencia > Temas.
* **favicon.ico**. La imagen que se verá en el navegador y al guardar el marcador.
* **header.php**. Módulo que contiene la cabecera del tema.
* **sidebar.php**. La barra lateral del tema. Nota: Si queremos más de una barra lateral, se habrán de configurar de modo manual.
* **footer.php**. Módulo que contiene el pie de página del tema.
* **single.php**. Este archivo indica cómo se ve una entrada de blog.
* **category.php**. Muestra cómo se ve la página de resultados de una categoría.
* **tag.php**. Muestra cómo se ve la página de resultados de una etiqueta.
* **search.php**. Muestra cómo se ve la página de resultados de búsqueda.
* **template.php**. [Plantilla](https://themeforest.net/?ref=creativasfera&clickthrough_id&redirect_back=true) de página que a título de ejemplo llamaremos template.
* **functions.php**. Archivo con las funciones de nuestro tema de WordPress.

A partir de aquí entrarán temas como diseño, capacidad de crear un buen código HTML y CSS, etc,

### Paso 1: Creamos el archivo style.css

El primer paso va a consistir en crear dentro de la carpeta mitema el archivo style.css. Dentro del archivo style.css, en la parte superior deberemos incluir esta información:

|  |  |
| --- | --- |
|  | /\* |
|  | Theme Name: Mitema | |

|  |  |
| --- | --- |
|  | Theme URI: <http://www.mitema.com> |
|  | Description: Mi primer tema de WP. | |

|  |  |
| --- | --- |
|  | Author: Tu nombre |
|  | Author URI: <http://www.tunombre.com> | |

|  |  |  |
| --- | --- | --- |
|  | Version: 1.0 | |
|  | \*/ |

Este fragmento de texto está comentado con /\* ... \*/. Todos los datos que ponemos aquí van a aparecer junto al tema en Apariencia > Temas.

A continuación, si vamos a utilizar HTML5, es conveniente insertar normalize.css. Lo podemos importar o bien incrustar aquí. Puedes descargarte aquí [una copia de normalize.css](http://necolas.github.io/normalize.css/).

A partir de aquí, podemos ya crear nuestro propio código de CSS.

### Paso 2: Imágenes y JavaScript

Este tema lo vamos a desarrollar con HTML5, por lo que para su correcto funcionamiento en todos los navegadores vamos a necesitar [Modernizr](http://modernizr.com/), y en caso que hagamos una web responsive, también necesitaremos [respond.js](https://github.com/scottjehl/Respond). Modernizr es una librería de JavaScript que detecta las capacidades de HTML5 y CSS3 del navegador. Respond es un polyfill que sirve para que Internet Explorer sepa interpretar los media queries de CSS3.

Así, crearemos las carpetas images y js dentro de la carpeta mitema, y en js pondremos una copia de Modernizr y Respond.

Ten en cuenta que el screenshot.png y el favicon.ico se han de ubicar en la raíz del tema y no dentro de la carpeta images.

### Paso 3: Creamos la cabecera del tema

El archivo header.php contiene la cabecera de nuestro tema. Es aquí donde pondremos nuestro logo, menú principal y aquello que aparezca en la parte superior de nuestro tema. Veamos el código de header.php:

|  |  |
| --- | --- |
|  | <!DOCTYPE html> |
|  | <html <?php language\_attributes(); ?>> | |

|  |  |
| --- | --- |
|  |  |
|  | <head> | |

|  |  |  |
| --- | --- | --- |
|  | <meta charset= <?php bloginfo( 'charset' ); ?>"> | |
|  | <title><?php wp\_title(); ?></title> |

|  |  |
| --- | --- |
|  |  |
|  | <!-- Definir viewport para dispositivos web móviles --> | |

|  |  |  |
| --- | --- | --- |
|  | <meta name="viewport" content="width=device-width, minimum-scale=1"> | |
|  |  |

|  |  |
| --- | --- |
|  | <link rel="shortcut icon" href="<?php echo get\_stylesheet\_directory\_uri(); ?>/favicon.ico" /> |
|  | <link rel="stylesheet" media="all" href="<?php bloginfo( 'stylesheet\_url' ); ?>" /> |

|  |  |
| --- | --- |
|  |  |
|  | <link rel="pingback" href="<?php bloginfo( 'pingback\_url' ); ?>" /> | |

|  |  |
| --- | --- |
|  |  |
|  | <?php wp\_head(); ?> | |

|  |  |
| --- | --- |
|  |  |
|  | </head> | |

|  |  |
| --- | --- |
|  | <body> |
|  | <div class="wrapper"> | |

|  |  |
| --- | --- |
|  | <header> |
|  | <h1><a href="<?php echo get\_option('home'); ?>"><?php bloginfo('name'); ?></a></h1> | |

|  |  |
| --- | --- |
|  | <hr> |
|  | <?php wp\_nav\_menu( array('menu' => 'Main', 'container' => 'nav' )); ?> | |

|  |  |
| --- | --- |
|  | </header> |

Lo primero que hemos hecho ha sido declarar el doctype, en este caso para HTML5. Seguidamente hemos abierto la etiqueta html indicando que seleccione los atributos de idioma de la instalación de WordPress. Dentro de head hemos usado bloginfo( 'charset' ); para que seleccione la codificación predeterminada de nuestra instalación de WordPress. La función wp\_title() selecciona y muestra el título especificado en Ajustes > General del panel de administración de WordPress.

En el caso de estar diseñando un tema responsive, especificaremos el viewport. Para mostrar la URL al favicon.ico le habremos de indicar la ruta a la carpeta mitema de la instalación de WordPress, y para ello llamaremos a la función de PHP get\_stylesheet\_directory\_uri();:

|  |  |
| --- | --- |
|  | link rel="shortcut icon" href="<?php echo get\_stylesheet\_directory\_uri(); ?>/favicon.ico" /> |

Otro modo de acceder a mitema será:

|  |  |
| --- | --- |
|  | <script src="<?php bloginfo('stylesheet\_directory'); ?>/js/respond.min.js"></script> |

Aquí hemos accedido al directorio de mitema con bloginfo('stylesheet\_directory');.

Finalmente incluiremos la función wp\_head de WordPress, necesaria para que muchos [plugins](http://www.esandra.com/in/plugins) y funcionalidades de WordPress funcionen. A partir de aquí cerramos la etiqueta head y empezamos con los contenidos de body.

La razón de por qué incluir el body en header.php y no en index.php. es que los contenidos de la cabecera del tema van a estar en todas las páginas, por lo que al incluirlo aquí, evitamos duplicar código y el mantenimiento del tema va a ser mucho más sencillo. Así, dentro de las etiquetas de header incluiremos nuestra cabecera. En este ejemplo, hemos puesto el nombre del sitio web con un enlace a la página de inicio:

|  |  |
| --- | --- |
|  | <h1><a href="<?php echo get\_option('home'); ?>"><?php bloginfo('name'); ?></a></h1> |

**Insertar el menú de navegación**

En la etiqueta header hemos escrito esta línea:

|  |  |
| --- | --- |
|  | <?php wp\_nav\_menu( array('menu' => 'Main', 'container' => 'nav' )); ?> |

Bien, para que esto funcione, primero deberemos añadir al archivo functions.php este código:

|  |  |
| --- | --- |
|  | <?php |
|  | // Registro del menú de WordPress | |

|  |  |
| --- | --- |
|  |  |
|  | add\_theme\_support( 'nav-menus' ); | |

|  |  |
| --- | --- |
|  |  |
|  | if ( function\_exists( 'register\_nav\_menus' ) ) { | |

|  |  |  |
| --- | --- | --- |
|  | register\_nav\_menus( | |
|  | array( |

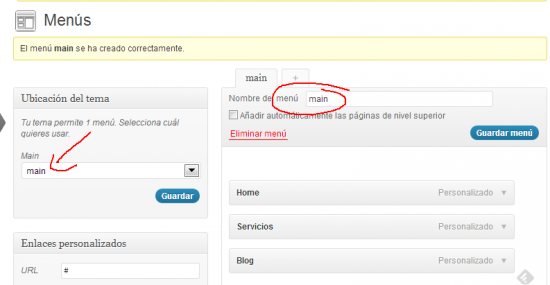
|  |  |  |
| --- | --- | --- |
|  | 'main' => 'Main' | |
|  | ) |

|  |  |  |
| --- | --- | --- |
|  | ); | |
|  | } |

|  |  |
| --- | --- |
|  | ?> |

Este código lo que hace es crear un menú llamado “Main” que nos aparecerá en el panel de administración de WordPress en Apariencia > Menús. Lo que esté código dice es que si existe una función llamada register\_nav\_menus, que entonces cree un nuevo menú de navegación llamado Main. Hemos añadido la función de WordPRess add\_theme\_support() para indicar que en este tema queremos poder crear menús dinámicos.

A partir de aquí, simplemente habremos de ir a Apariencia > Menús y crear nuestro menú y guardarlo:

[](http://www.esandra.com/file/2013/05/menu-wordpress.png)

Con esto ya tenemos la cabecera del tema creada junto con su menú de navegación. Recuerda que para añadir páginas al menú de WordPress, primero hay que crear las páginas o no podremos hacerlo.

### Paso 4: Creamos la barra lateral

La mayoría de temas de WordPress incluyen una barra lateral que no tiene por qué aparecer en todas las páginas del tema. En las barras laterales acostumbramos a poder añadir widgets, por lo que se van a tener que habilitar para widgets. Para ello, abriremos nuestro archivo functions.php e incluiremos esta función:

|  |  |
| --- | --- |
| <?php | |
|  |

|  |
| --- |
| //  Main Sidebar |
| if(function\_exists('register\_sidebar')) | |

|  |
| --- |
| register\_sidebar(array( |
| 'name' => 'Main Sidebar', | |

|  |  |
| --- | --- |
| 'before\_widget' => '<hr>', | |
| 'after\_widget' => '', |

|  |
| --- |
| 'before\_title' => '<h3>', |
| 'after\_title' => '</h3>', |

|  |  |
| --- | --- |
| )); | |
|  |

|  |
| --- |
| ?> |

Al hacer esto, si vamos a Apariencia > Widgets, veremos que nos aparece una nueva zona de widgets llamada Main Sidebar.

**El archivo sidebar.php**

Una vez creada la zona lista para widgets, vamos a tener que llamarla dentro del archivo sidebar.php. Creamos este archivo y escribimos:

|  |  |
| --- | --- |
| <section id="sidebar"> | |
|  |

|  |  |
| --- | --- |
| <?php if ( !function\_exists('dynamic\_sidebar') || !dynamic\_sidebar('Main Sidebar') ) : ?> | |
| <?php endif; ?> |

|  |
| --- |
|  |
|  |

|  |
| --- |
| </section> |

He utilizado un id teniendo en cuenta que en este tema sólo va a existir una barra lateral, pero si quieres crear más, quizás prefieras crear una clase. Con esto ya tenemos creada la barra lateral lista para widgets.

### Paso 5: Creamos el footer

Hasta aquí hemos creado la cabecera y la barra lateral. Ahora vamos a crear el footer para después integrarlo todo en index.php. Recordemos que WordPress funciona por módulos, con lo que lo que estamos haciendo ahora es crear los módulos necesarios para que nuestra página de inicio, en este caso index.php, se muestre correctamente.

**El archivo footer.php**

|  |
| --- |
| <footer> |
| <p>&amp;amp;amp;copy; <?php bloginfo('name'); ?>, <?=date('Y');?>. Mi primer tema de WP.</p> | |

|  |
| --- |
|  |
| </footer> | |

|  |
| --- |
|  |
| </div> <!-- Fin de wrapper --> | |

|  |
| --- |
|  |
| <?php wp\_footer(); ?> | |

|  |
| --- |
| </body> |
| </html> |

Wrapper engloba tanto el header como el footer, si queremos un header o un footer que se extienda toda la pantalla, deberemos mover las posiciones de los div que abren y cierran wrapper.

Aquí aparece la función wp\_footer, que al igual que pasaba con wp\_header, no debemos olvidarla. Estas funciones son lo que se llaman *hooks* y son necesarias para que funcionen los plugins que instalemos.

En el caso de wp\_footer va a ser además necesaria si queremos ver la barra de administración de WordPress. Lo comento más que nada porque no serás el primero que se vuelve loco buscando el por qué no aparece la barra de administración hasta que descubre que es porque no ha llamado a este *hook* de WordPress.

### Paso 6: Creamos la página principal

Como ya he comentado, en este tutorial la página principal se va a corresponder a los contenidos de index.php. Así, creamos este archivo tal y como indico a continuación.

**El archivo index.php**

|  |  |  |
| --- | --- | --- |
|  | <?php get\_header(); ?> | |
|  |  |

|  |  |
| --- | --- |
|  |  |
|  | <section id="main"> | |

|  |  |
| --- | --- |
|  |  |
|  |  |

|  |  |
| --- | --- |
|  |  |
|  | </section> <!-- Fin de main --> | |

|  |  |
| --- | --- |
|  |  |
|  | <?php  get\_sidebar()?> | |

|  |  |
| --- | --- |
|  |  |
|  |  |

|  |  |
| --- | --- |
|  | <?php get\_footer(); ?> |

Fíjate en estas tres funciones:

get\_header();

get\_sidebar();

get\_footer();

Deben ir siempre entre las etiquetas, o en otro caso no funcionará.

Lo que hace get\_header() es insertar los contenidos del archivo header.php en el lugar donde se encuentra esta función.

Lo mismo pasa con get\_sidebar() para los contenidos de sidebar.php y get\_footer() para los contenidos de footer.php. Es lo que se comentaba anteriormente de que WordPress funciona por módulos.

### Paso 7. El loop de WordPress

La potencia de WordPress reside en el loop. ¿Qué es el loop? Significa bucle en inglés y lo que hace es mostrar las entradas de la instalación de WordPress de la forma en que nosotros lo programemos. Es gracias al loop de WordPress que podemos ver un listado de entradas

Código de un loop:

|  |  |  |
| --- | --- | --- |
|  | <?php if (have\_posts()) :  while (have\_posts()) : the\_post(); ?> | |
|  |  |

|  |  |
| --- | --- |
|  | <h2><a href="<?php the\_permalink(); ?>"><?php the\_title(); ?></a></h2> |
|  | <small>Publicado el <?php the\_time('j/m/Y') ?> por <?php the\_author\_posts\_link() ? </small> |

|  |  |  |
| --- | --- | --- |
|  | <?php the\_excerpt(); ?> | |
|  |  |

|  |  |
| --- | --- |
|  |  |
|  | <?php endwhile; else: ?> | |

|  |  |  |
| --- | --- | --- |
|  | <p><?php \_e('No hay entradas .'); ?></p> | |
|  | <?php endif; ?> |

Vayamos paso a paso. En la primera línea tenemos:

|  |  |
| --- | --- |
|  | <?php if(have\_posts()) : ?><?php while(have\_posts()) : the\_post(); ?> |

Lo que aquí le estamos diciendo a WordPress es que mire si hay alguna entrada disponible, esto lo hacemos con if(have\_posts()). Lo siguiente que le decimos es que mientras haya entradas, esto es, while(have\_posts()), que seleccione cada uno de los posts: the\_post().

Una vez ha seleccionado el post, lo siguiente es que le vamos a indicar cómo ha de mostrar los contenidos del post.

Así pues, ahora mostraremos el título de la entrada con una enlace a la misma:

|  |  |
| --- | --- |
|  | <h3  class="title"><a href="<?php the\_permalink() ?>" rel="bookmark" title="Enlace permanente a <?php the\_title\_attribute(); ?>"><?php the\_title(); ?></a></h3> |

La función the\_permalink() se encarga de encontrar el enlace permanente a la entrada, para que podamos ir a la misma. Y la función the\_title() mostrará en pantalla el título de la entrada.

Si queremos que debajo del título se vean datos como la fecha de publicación, la categoría, el autor, etc., se lo tendremos que especificar a continuación:

|  |  |
| --- | --- |
|  | <small>Publicado el <?php the\_time('j/m/Y') ?> por <?php the\_author\_posts\_link() ?></small> |

La función the\_time() muestra la fecha de publicación y en el paréntesis podemos definir el formato de la fecha entre comillas. Con la función the\_author\_posts\_link() estamos mostrando el nombre del autor con un enlace a su página de autor.

Si queremos mostrar las etiquetas, lo haremos con la función the\_tags(), y si lo quisiéramos hacer es mostrar la categoría de la entrada, utilizaríamos la función the\_category().

Seguidamente, con la función the\_excerpt() mostraremos el resumen de la entrada. Si no hemos creado un resumen, se mostrará la primera parte de la entrada.

Para cerrar el loop, llamamos este código:

|  |  |
| --- | --- |
|  | <?php endwhile; else: ?> |
|  | <p><?php \_e('No hay entradas .'); ?></p> | |

|  |  |
| --- | --- |
|  | <?php endif; ?> |

En el caso que no hayan entradas, se mostrará el mensaje de “No hay entradas”.

### Paso 8. Añadir thumbnails al loop

Si queremos añadir thumbnails al loop, lo que haremos es lo siguiente: vamos al archivo functions.php y añadimos este código:

|  |  |
| --- | --- |
|  | <?php |
|  | //Habilitar thumbnails | |

|  |  |
| --- | --- |
|  | add\_theme\_support('post-thumbnails'); |
|  | set\_post\_thumbnail\_size(150, 150, true); | |

|  |  |
| --- | --- |
|  | ?> |

Con este código hemos habilitado los thumbnials o imágenes destacadas de nuestras entradas y hemos establecido un tamaño para las mismas de 150x150 píxeles. Una vez hecho esto, modificaremos nuestro loop como sigue:

|  |  |  |
| --- | --- | --- |
|  | <?php if (have\_posts()) :  while (have\_posts()) : the\_post(); ?> | |
|  |  |

|  |  |
| --- | --- |
|  | <h2><a href="<?php the\_permalink(); ?>"><?php the\_title(); ?></a></h2> |
|  | <small>Publicado el <?php the\_time('j/m/Y') ?> por <?php the\_author\_posts\_link() ?>  </small> |

|  |  |  |
| --- | --- | --- |
|  | <div class="thumbnail"> | |
|  | <?php |

|  |  |  |
| --- | --- | --- |
|  | if ( has\_post\_thumbnail() ) { | |
|  | the\_post\_thumbnail(); |

|  |  |  |
| --- | --- | --- |
|  | } | |
|  | ?> |

|  |  |
| --- | --- |
|  | </div> |
|  | <?php the\_excerpt(); ?> | |

|  |  |
| --- | --- |
|  |  |
|  | <?php endwhile; else: ?> | |

|  |  |  |
| --- | --- | --- |
|  | <p><?php \_e('No hay entradas .'); ?></p> | |
|  | <?php endif; ?> |

Fíjate qué hemos añadido:

|  |  |
| --- | --- |
|  | <?php |
|  | if ( has\_post\_thumbnail() ) { | |

|  |  |  |
| --- | --- | --- |
|  | the\_post\_thumbnail(); | |
|  | } |

|  |  |
| --- | --- |
|  | ?> |

if ( has\_post\_thumbnail() ) significa que compruebe si la entrada en cuestión tiene una imagen destacada asignada y en caso afirmativo, le decimos que la muestre con the\_post\_thumbnail();.

A partir de aquí, si queremos que el extracto quede en línea con la imagen destacada, lo haremos a través del CSS.

Con esto ya hemos creado nuestra página principal, con su cabecera, zona de contenidos con las últimas entradas, barra lateral y pie de página.

Para añadir una paginación de las entradas, leer [cómo crear la paginación de las entradas en WordPress](http://design.sparklette.net/teaches/how-to-add-wordpress-pagination-without-a-plugin/).

### Paso 9. Crear un campo de búsqueda

El archivo **search.php**

|  |  |  |
| --- | --- | --- |
|  | <?php get\_header(); ?> | |
|  |  |

|  |  |
| --- | --- |
|  | <section id="main"> |
|  | <h2>Resultados de la búsqueda</h2> | |

|  |  |  |
| --- | --- | --- |
|  | <?php if (have\_posts()) :  while (have\_posts()) : the\_post(); ?> | |
|  |  |

|  |  |
| --- | --- |
|  | <h2><a href="<?php the\_permalink(); ?>"><?php the\_title(); ?></a></h2> |
|  | <small>Publicado el <?php the\_time('j/m/Y') ?> por <?php the\_author\_posts\_link() ?>  </small> |

|  |  |  |
| --- | --- | --- |
|  | <div class="thumbnail"> | |
|  | <?php |

|  |  |  |
| --- | --- | --- |
|  | if ( has\_post\_thumbnail() ) { | |
|  | the\_post\_thumbnail(); |

|  |  |  |
| --- | --- | --- |
|  | } | |
|  | ?> |

|  |  |
| --- | --- |
|  | </div> |
|  | <?php the\_excerpt(); ?> | |

|  |  |
| --- | --- |
|  |  |
|  | <?php endwhile; ?> | |

|  |  |
| --- | --- |
|  | <?php if (function\_exists("pagination")) { |
|  | pagination($additional\_loop->max\_num\_pages); | |

|  |  |
| --- | --- |
|  | } ?> |
|  | <?php  else: ?> | |

|  |  |  |
| --- | --- | --- |
|  | <div class="entry"><?php \_e('Lo sentimos, no hay resultados con este término de búsqueda.'); ?></div> | |
|  | <?php endif; ?> |

|  |  |
| --- | --- |
|  |  |
|  | </section> <!-- Fin de main --> | |

|  |  |
| --- | --- |
|  |  |
|  | <?php  get\_sidebar()?> | |

|  |  |
| --- | --- |
|  |  |
|  | <?php get\_footer(); ?> | |

Se parece mucho a index.php. Esta página es la que va a mostrar los resultados de una búsqueda en cuestión. Así, vamos a necesitar crear un buscador. Vamos al archivo sidebar.php y justo antes de la función que habilita la zona de widgets en el sidebar, incluimos este código:

|  |  |
| --- | --- |
|  | <form method="get" id="searchform" action="<?php bloginfo('url'); ?>/"> |
|  | <input type="text" placeholder="Buscar…" value="<?php the\_search\_query(); ?>" name="s" id="s" /> |

|  |  |  |
| --- | --- | --- |
|  | <button type="submit" class="icon-only"  id="searchsubmit"></button> | |
|  | </form> |

Con esto tenemos un campo de búsqueda en nuestra barra lateral. Para que se visualice como queremos, simplemente hará falta crear el código CSS necesario.

### Paso 10.Configurar la página que muestra la entrada

La página del post es aquella donde vamos a ver la entrada en cuestión. Vamos a poder definir cómo se ve, incluir o no comentarios, etc.

**El archivo single.php**

|  |  |  |
| --- | --- | --- |
|  | <?php get\_header(); ?> | |
|  |  |

|  |  |
| --- | --- |
|  | <section id="main"> |
|  | <article id="single"> | |

|  |  |  |
| --- | --- | --- |
|  | <?php if(have\_posts()) : ?><?php while(have\_posts()) : the\_post(); ?> | |
|  |  |

|  |  |
| --- | --- |
|  | <h2><a href="<?php the\_permalink() ?>" rel="bookmark" title="Enlace permanente a <?php the\_title\_attribute(); ?>"><?php the\_title(); ?>. </a></h2> |
|  | <small> Publicado por <?php the\_author\_posts\_link() ?>  el <?php the\_time('j/m/Y') ?>. Categoría: <?php the\_category(', '); ?> </small><br> |

|  |  |
| --- | --- |
|  |  |
|  | <div class="post"> | |

|  |  |
| --- | --- |
|  |  |
|  | <?php the\_content(); ?> | |

|  |  |
| --- | --- |
|  |  |
|  | </div> | |

|  |  |
| --- | --- |
|  |  |
|  |  |

|  |  |  |
| --- | --- | --- |
|  | <?php endwhile; ?> | |
|  |  |

|  |  |  |
| --- | --- | --- |
|  | <?php endif; ?> | |
|  |  |

|  |  |  |
| --- | --- | --- |
|  | </article> <!-- Fin de single --> | |
|  |  |

|  |  |  |
| --- | --- | --- |
|  | </section> <!-- Fin de main --> | |
|  | <?php  get\_sidebar()?> |

|  |  |
| --- | --- |
|  |  |
|  | <?php get\_footer(); ?> | |

Faltaría añadir el poder poner comentarios en cadena y las entradas relacionadas. En el loop utilizamos the\_excerpt() para mostrar el extracto de la entrada, mientras que aquí utilizamos la función the\_content() para que se muestre el contenido de la entrada.

**Mostrar las entradas relacionadas en single.php**

Para mostrar las entradas relacionadas, añadiremos este código justo después de mostrar el contenido de la entrada:

|  |  |
| --- | --- |
|  | <?php |
|  | $tags = wp\_get\_post\_tags($post->ID); | |

|  |  |
| --- | --- |
|  | if ($tags) { |
|  | $tag\_ids = array(); | |

|  |  |  |
| --- | --- | --- |
|  | foreach($tags as $individual\_tag) $tag\_ids[] = $individual\_tag->term\_id; | |
|  | $args=array( |

|  |  |
| --- | --- |
|  | 'tag\_\_in' => $tag\_ids, |
|  | 'post\_\_not\_in' => array($post->ID), | |

|  |  |  |
| --- | --- | --- |
|  | 'showposts'=>5, // Number of related posts that will be shown. | |
|  | 'caller\_get\_posts'=>1 |

|  |  |
| --- | --- |
|  | ); |
|  | $my\_query = new wp\_query($args); | |

|  |  |
| --- | --- |
|  | if( $my\_query->have\_posts() ) { |
|  | echo '<h3>Artículos relacionados</h3><ul>'; | |

|  |  |  |
| --- | --- | --- |
|  | while ($my\_query->have\_posts()) { | |
|  | $my\_query->the\_post(); |

|  |  |
| --- | --- |
|  | ?> |
|  | <li><a href="<?php the\_permalink() ?>" rel="bookmark" title="Permanent Link to <?php the\_title\_attribute(); ?>"><?php the\_title(); ?></a></li> | |

|  |  |  |
| --- | --- | --- |
|  | <?php | |
|  | } |

|  |  |  |
| --- | --- | --- |
|  | echo '</ul>'; | |
|  | } |

|  |  |  |
| --- | --- | --- |
|  | wp\_reset\_query(); | |
|  | } |

|  |  |
| --- | --- |
|  | ?> |

Este código muestra las entradas relacionadas en base a las etiquetas de las entradas.

**Habilitar y mostrar los comentarios**

En una entrada va a ser necesario que podamos comentar y ver los comentarios de los otros.

Para permitir los comentarios encadenados, añadiremos este código a nuestro archivo functions.php:

|  |  |
| --- | --- |
|  | // Permitir comentarios encadenados |
|  | function enable\_threaded\_comments(){ | |

|  |  |  |
| --- | --- | --- |
|  | if (is\_singular() AND comments\_open() AND (get\_option('thread\_comments') == 1)) { | |
|  | wp\_enqueue\_script('comment-reply'); |

|  |  |  |
| --- | --- | --- |
|  | } | |
|  | } |

|  |  |
| --- | --- |
|  | add\_action('get\_header', 'enable\_threaded\_comments'); |
|  |  |

Y en single.php justo antes de la línea que contiene el endwhile añadiremos:

|  |  |
| --- | --- |
|  | <div class="comentarios"> |
|  | <?php comments\_template(); ?> | |

|  |  |
| --- | --- |
|  | </div> |

Es el CSS lo que visualmente nos va a permitir que se vean los comentarios encadenados, aquí el CSS que se utiliza para los comentarios:

|  |  |
| --- | --- |
|  | .commentlist li { |
|  | list-style: none !important; | |

|  |  |  |
| --- | --- | --- |
|  | } | |
|  |  |

|  |  |
| --- | --- |
|  |  |
|  | .comment-body { | |

|  |  |  |
| --- | --- | --- |
|  | padding: 15px; | |
|  | } |

|  |  |
| --- | --- |
|  |  |
|  | .comment-author { | |

|  |  |  |
| --- | --- | --- |
|  | margin-top: 5px; | |
|  | } |

|  |  |
| --- | --- |
|  |  |
|  | .reply,  .commentmetadata { | |

|  |  |  |
| --- | --- | --- |
|  | font-size: 0.9em; | |
|  | } |

|  |  |
| --- | --- |
|  |  |
|  |  |

|  |  |  |
| --- | --- | --- |
|  | .depth-1, .depth-2, .depth-3, .depth-4 { | |
|  | border: 1px solid #eaeaea; |

|  |  |  |
| --- | --- | --- |
|  | border-radius: 4px; | |
|  | } |

|  |  |
| --- | --- |
|  |  |
|  | .depth-1, .depth-3 { | |

|  |  |  |
| --- | --- | --- |
|  | background-color: #fafafa; | |
|  | } |

|  |  |
| --- | --- |
|  |  |
|  | .depth-2, .depth-4 { | |

|  |  |  |
| --- | --- | --- |
|  | background-color: #fff; | |
|  | } |

|  |  |
| --- | --- |
|  |  |
|  | .comment-author-eSandra { | |

|  |  |
| --- | --- |
|  | border: 3px solid #eaeaea; |
|  | border-left:3px solid #fbc356; | |

|  |  |  |
| --- | --- | --- |
|  | } | |
|  |  |

|  |  |
| --- | --- |
|  | .depth-1 { |
|  | margin: 20px 0; | |

|  |  |
| --- | --- |
|  | } |
|  | /\* Respuesta a comentarios\*/ | |

|  |  |
| --- | --- |
|  |  |
|  |  |

|  |  |
| --- | --- |
|  | .depth-2 { |
|  | margin:2% 5% !important; | |

|  |  |  |
| --- | --- | --- |
|  | } | |
|  |  |

|  |  |
| --- | --- |
|  | .depth-3 { |
|  | margin: 2% 7% !important; | |

|  |  |  |
| --- | --- | --- |
|  | } | |
|  |  |

|  |  |
| --- | --- |
|  |  |
|  | .depth-4 { | |

|  |  |  |
| --- | --- | --- |
|  | margin:2% 5% 4% 10% !important; | |
|  | }0 |

|  |  |
| --- | --- |
|  | 1 |
|  | + |

|  |  |  |
| --- | --- | --- |
|  | #comment, #author, #email { | |
|  | border-radius: 4px; |

|  |  |  |
| --- | --- | --- |
|  | border:2px solid #ccc; | |
|  | } |

Con todo esto, ya tenemos la página de la entrada.

### Paso 11. Crear una nueva plantilla

Para crear nuestras propias plantillas y poderlas seleccionar desde el panel de administración de WordPress cuando creemos o editemos una página, simplemente hemos de añadir en la parte superior este código:

|  |  |
| --- | --- |
|  | <?php |
|  | /\*Template Name: Nombre\_de\_la\_plantilla \*/ | |

|  |  |
| --- | --- |
|  | ?> |

A partir de aquí, le daremos el formato que queramos.

### Paso 12: Página de categorías y etiquetas

Las páginas de category.php y etiquetas.php son prácticamente iguales. Simplemente añadiremos antes del loop:

|  |  |
| --- | --- |
|  | <?php $post = $posts[0];  ?> |
|  | <?php  if (is\_category()) { ?> | |

|  |  |
| --- | --- |
|  | <?php } elseif( is\_tag() ) { ?> |
|  | <h2>Etiqueta &amp;amp;amp;#8216;<?php single\_tag\_title(); ?>&amp;amp;amp;#8217;</h2> | |

|  |  |
| --- | --- |
|  | <?php } elseif (is\_day()) { ?> |
|  | <h2>Archivo para <?php the\_time('F jS Y'); ?>:</h2> | |

|  |  |  |
| --- | --- | --- |
|  | <?php  } elseif (is\_month()) { ?><h2>Archivo para <?php the\_time('F, Y'); ?>:</h2> | |
|  | <?php } elseif (is\_year()) { ?> |

|  |  |  |
| --- | --- | --- |
|  | <h2>Archivo para <?php the\_time('Y'); ?>:</h2> | |
|  | <?php /} elseif (is\_author()) { ?> |

|  |  |
| --- | --- |
|  | <h2>Archivo del autor </h2> |
|  | <?php } elseif (isset($\_GET['paged']) &amp;amp;amp;&amp;amp;amp; !empty($\_GET['paged'])) { ?> | |

|  |  |  |
| --- | --- | --- |
|  | <h2>Archivos del blog</h2> | |
|  | <?php } ?> |

Haremos lo mismo si queremos crear una página con nuestros archivos.

### Para acabar

En este artículo hemos visto cómo crear un tema de WordPress desde cero. Se pueden añadir otros archivos, como author.php para mostrar la página de autor, o 404.php para crear nuestra propia página de error

Fuente:

* <http://www.esandra.com/crear-plantilla-wordpress-desde-cero>
* <https://franciscoamk.com/crear-tema-wordpress/>
* <https://franciscoamk.com/crear-tema-wordpress-2/>