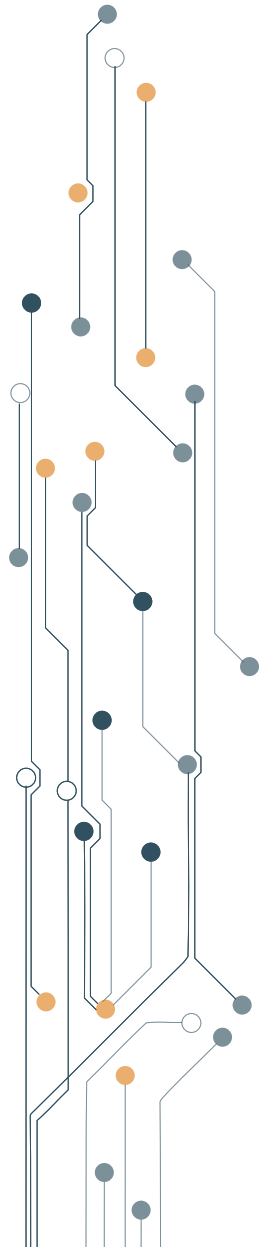
Themes

Telefónica
EDUCACIÓN DIGITAL

Índice



Themes

1 Cómo elegir un tema

Existen webs que se dedican a comercializar temas de Wordpress y cuentan con un extenso catálogo: ThemeForest[1], Creative Market[2], MojoThemes[3],... Estas webs permiten consultar y comparar muchos temas y nos ofrecen opiniones de usuarios. También hay desarrolladores de temas que comercializan directamente sus temas desde sus webs y, por supuesto, gran cantidad de desarrolladores capaces de desarrollar o adaptar temas a medida.

A la hora de escoger un tema tenemos varias opciones. Podemos limitarnos a buscar e instalar un tema (gratuito o de pago), podemos realizar modificaciones a partir de un tema, o podemos crear uno desde cero.

Para tomar esta decisión deberemos tener en cuenta varios factores:

Objetivo

No es lo mismo realizar una web con una finalidad concreta y un uso temporal (para anunciar un evento, por ejemplo) que una web que deba mantenerse en el tiempo y crecer sin que la plantilla suponga limitaciones. Además hay que tener en cuenta que el diseño que elijamos debe servir para mostrar de la forma más correcta el contenido de la web y para atraer al público y facilitarle encontrar lo que busca.

Presupuesto/Tiempo

La cantidad de dinero disponible y el plazo para el desarrollo son determinantes, aunque lo deseable es que estén alineados con los objetivos a cumplir.

Funcionalidades y flexibilidad

Podemos encontrar plantillas para blogs, portafolios, webs corporativas, tiendas online... y también temas lo suficientemente completos para dar cabida o todos estos tipos de proyectos. Estos últimos aportan evidentemente una mayor flexibilidad, pero en algunos casos pueden significar una complejidad extra innecesaria.

1 https://themeforest.net/category/wordpress
2 https://creativemarket.com/themes/wordpress
3 http://www.mojo-themes.com/categories/wordpress/

Personalización

Algunos temas están preparados para modificar su aspecto directamente desde el panel de administración de Wordpress sin necesidad de tener conocimientos de diseño o CSS. Estos temas permiten por ejemplo cambiar las tipografías o el esquema de colores de la web

Calidad

¿Cómo podemos garantizar que el código tiene la calidad deseable (rendimiento, SEO, etc) y que el tema funciona correctamente? Un buen punto de partido puede ser consultar la experiencia del desarrollador y las opiniones de otros usuarios.

Mantenimiento

También debemos consultar si el tema esta correctamente mantenido, es decir, si recibe actualizaciones en caso de detectarse errores o para adaptarse a nuevas versiones de Wordpress.

Soporte

Es importante, cuando usamos temas de terceros, contar con alguna vía de soporte a través de la cual poder hacer consultas al desarrollador en caso de que nos encontremos con algún problema.

Extras

Algunos temas incluyen plugins e integran plugins para, por ejemplo, construir las páginas "arrastrando y soltando" (drag and drop) elementos[4], o sliders para mostrar y animar diferentes contenidos en un mismo espacio. De nuevo, habrá que valorar si estos extras suponen un valor añadido o no para nuestro propósito.

Otras consideraciones

Dependiendo de nuestros objetivos, puede ser esencial encontrar un tema que funcione correctamente en cualquier tipo de dispositivo y tamaño de pantalla (algo que podemos considerar esencial en casi todos los casos actualmente), o contar con traducciones a diferentes idiomas.

4 Como Visual Composer (https://vc.wpbakery.com) de WPBakery, Divi Builder (https://www.elegantthemes.com/plugins/divi-builder/) de Elegant Themes, o Fusion Builder (http://avada.theme-fusion.com/fusion-builder-2/) de Avada

2 Punto de partida

Podemos crear un tema desde cero o partir de un tema ya existente. Lo habitual es modificar un tema cuando las modificaciones a realizar van a ser mínimas, pues de lo contrario puede ser más sencillo desarrollar íntegramente el tema por nosotros mismos.

Parent/Child themes

La forma habitual de crear un tema partiendo de otro es la de generar un tema hijo (child theme) que herede del tema a modificar pero le sobreescriba o modifique en aquello que queramos. Esta forma de actuar permitirá que podamos actualizar el tema padre en caso de que aparezca una nueva versión sin que eso signifique perder nuestras modificaciones específicas al mismo.

Crear un tema hijo a partir de otro será recomendable cuando queramos hacer pequeñas variaciones que el propio tema no nos permita realizar desde el área de administración, y siempre que no vayamos a cambiar de forma drástica su estructura y su funcionalidad.

Starter themes

Sin embargo, también existen temas a modo de "esqueleto" que nos facilitan el desarrollo de nuestros propios temas, puesto que cuentan con el código básico y con u diseño mínimo que nos servirá de punto de partida ahorrándonos tiempo. Algunos de los más populares son Underscores[5], Components[6], Bones[7], Slekeleton[8], FoundationPress[9]. También existen otros como Sage[10], que está pensados para ofrecer un flujo de trabajo más profesional al desarrollador (usando herramientas como Gulp/Bower, SASS).

Una simple búsqueda de "starter themes" para WordPress nos ofrecerá más opciones y nos permitirá consultar el estado de cada uno de los proyectos para elegir el que más nos convenga.

---

5 http://underscores.me/
También llamado "_s". Desarrollado por Automattic.

6 http://components.underscores.me/
También de Automattic., Ofrece puntos de partida diferentes según el tipo de web a desarrollar

7 http://themble.com/bones/

8 http://themes.simplethemes.com/skeleton/

9 https://foundationpress.olefredrik.com/
Hace uso del framework front-end Foundation 6.

10 https://roots.io/sage/

Theme frameworks

Además de los citados starter themes, contamos también con los llamados "theme frameworks" en los que el tema padre incluye toda la parte de programación (archivos .php y .js) para relegar al tema hijo solo lo referente al estilo (CSS y áreas de widgets), aunque también contaremos con el archivo functions.php en caso de necesitarlo.

Por eso, en caso de querer modificar solo el diseño, estos frameworks pueden ser una buena opción (generalmente de pago) como punto de partida. Además, podremos cambiar nuestro tema por otro desarrollado para el mismo framework sin temor a perder ninguna funcionalidad.

Entre los frameworks más usados podemos encontrar: Genesis Framework[11], Thesis[12], o PageLines[13]

11 http://my.studiopress.com/themes/genesis/
12 http://diythemes.com/
13 https://www.pagelines.com/

3 Creando un tema

Para crear un tema solo es necesario tener un buen conocimiento de HTML y CSS. Conocer PHP nos permitirá ampliar y configurar ciertas funcionalidades, pero la mayoría de las lineas que tendremos que escribir en PHP cuando creemos temas son llamadas a funciones bien documentadas[14].

La documentación de WordPress tiene mucha información sobre el desarrollo de temas:

- Manual del Desarrollador de Temas
  https://developer.wordpress.org/themes/
- Guía de desarrollo de temas en el Codex
  https://codex.wordpress.org/Theme_Development

En las siguientes páginas veremos las claves básicas para el desarrollo de temas. Crearemos un tema partiendo de otro, aunque el proceso es similar para crear un tema desde cero. Solo cambia el tiempo que nos llevaría su desarrollo y el hecho de que, en este caso, nos podremos beneficiar de las posibles actualizaciones del tema padre.

Lo primero es instalar el tema padre en caso de no estar ya instalado. Para hacerlo podemos seguir acceder a wp-admin > Apariencia > Temas y pulsar el botón "Añadir nuevo" para buscar el tema que queramos e instalarlo directamente. En caso de que el tema que queramos instalar no esté disponible en el listado ofrecido por Wordpress, bastará con descargarlo de la web correspondiente y subirlo (comprimido en .zip) mediante el botón de "Subir tema", o vía FTP (descomprimido) a nuestro directorio wp-content/themes, de forma que quede junto a los temas de ejemplo y con una estructura interna similar.

En nuestro caso vamos a usar como padre uno de los temas provistos con la instalación de Wordpress (Twenty Sixteen) por lo que no será necesario instalar un tema nuevo

Nuestro tema hijo, deberá estar junto con los demás, por lo que crearemos una carpeta junto al resto de temas en el directorio wp-content/ a la que llamaremos twentysixteen_child

14 En el caso de los child themes podremos modificar el aspecto aprovechándonos del código ya existente y limitando la presencia de PHP a los bucles y las llamadas que "pintarán" nuestro contenido.

3.1 Hojas de estilo

El único archivo necesario para que WordPress reconozca nuestro child theme es la hoja de estilos que deberá llamarse style.css en el que definiremos unos parámetros básicos e importaremos el archivo style.css del tema padre.

\wp-content\themes\twentysixteen_child\style.css

```
/*
Theme Name: Twenty Sixteen Child Theme
Theme URI: http://wordpress.org/themes/
twentysixteen
Description: Example of Twenty Sixteen Child Theme
Author: Curso WP
Author URI: http://talentumempleo.com/
Template: twentysixteen
Version: 1.0.0
*/

@import url("../twentysixteen/style.css");
```

Como vemos hemos incluido la información básica sobre nuestro tema[15], en la que debemos incluir junto a la palabra "Template" el nombre del tema del que heredaremos[16]. Esa referencia a la plantilla ("template") y la linea final que importa sus estilos son fundamentales para que nuestro tema sea su "hijo".

Otras etiquetas que podríamos añadir en la información son "License" para añadir el nombre del tipo de licencia, "License URI" indicando url para ampliar información sobre dicha licencia, o "Tags" señalando etiquetas a modo de palabras clave.

Solo con este archivo situado correctamente, ya debemos tener el tema disponible en wp-admin > Apariencia > Temas. Y si hacemos clic sobre él, se nos mostrará la información que incluimos en el comentario del inicio del archivo. Basta con activar el tema para comenzar a usarlo aunque, al no tener aún ninguna modificación, el resultado será el mismo que al usar el tema padre. Por ahora.

Todo el código que añadamos en nuestro archivo style.css se ejecutará después del código de la hoja de estilo del tema padre, por lo que será muy fácil sobreescribir el diseño original.

15 Esta información no debe coincidir con la de ningún otro tema.

16 Date cuenta de que usamos el nombre que tiene el directorio de la carpeta, no el que aparece con mayúsculas y espacios en el panel de administración.

3.2 Plantillas

Las plantillas son los archivos que generarán las páginas a mostrar a los visitantes en función de su petición. Estos archivos incluyen la estructura HTML, código PHP y las llamadas **etiquetas de plantilla (template tags)**[17] que son funciones que nos facilitan obtener elementos o información de nuestro wordpress.

Al estar creando un tema hijo, ya contamos con las plantillas del tema padre, que podremos reescribir o ampliar. En caso de estar creando un tema completamente nuevo, es preciso crear al menos una plantilla llamada index,php que sería la que se usara para mostrar todas las páginas de nuestro sitio.

Lo normal es que queramos usar plantillas diferentes para diferentes páginas de nuestra web.

- index.php
  Plantilla principal. Es la única cuya presencia es obligada.
- comments.php
  Plantilla para comentarios.
- front-page.php
  Plantilla para la página principal.
- home.php
  Plantilla para la página que muestra las últimas entradas.

17 https://codex.wordpress.org/Template_Tags

Wordpress indica que esta página sea la principal de la web si no se indica lo contrario en el wp-admin

- single.php
  Plantilla para mostrar cada entrada de forma individual.
- single-{post-type}.php
  Plantilla para mostrar cada entrada del tipo definido[18], de forma individual. Por ejemplo single-book.php se usará para mostrar de forma de forma individual las entradas del tipo “book”.
- page.php
  Plantilla para mostrar cada página de forma individual.
- category.php
  Plantilla a mostrar cuando se solicita una categoría determinada.
- tag.php
  Plantilla a mostrar cuando se solicita una etiqueta determinada.
- taxonomy.php
  Plantilla a mostrar cuando se solicita un término determinado de una taxonomía.
- author.php
  Plantilla a mostrar cuando se solicita información sobre un

18 Veremos como crear diferentes tipos de entradas en el próximo capítulo.

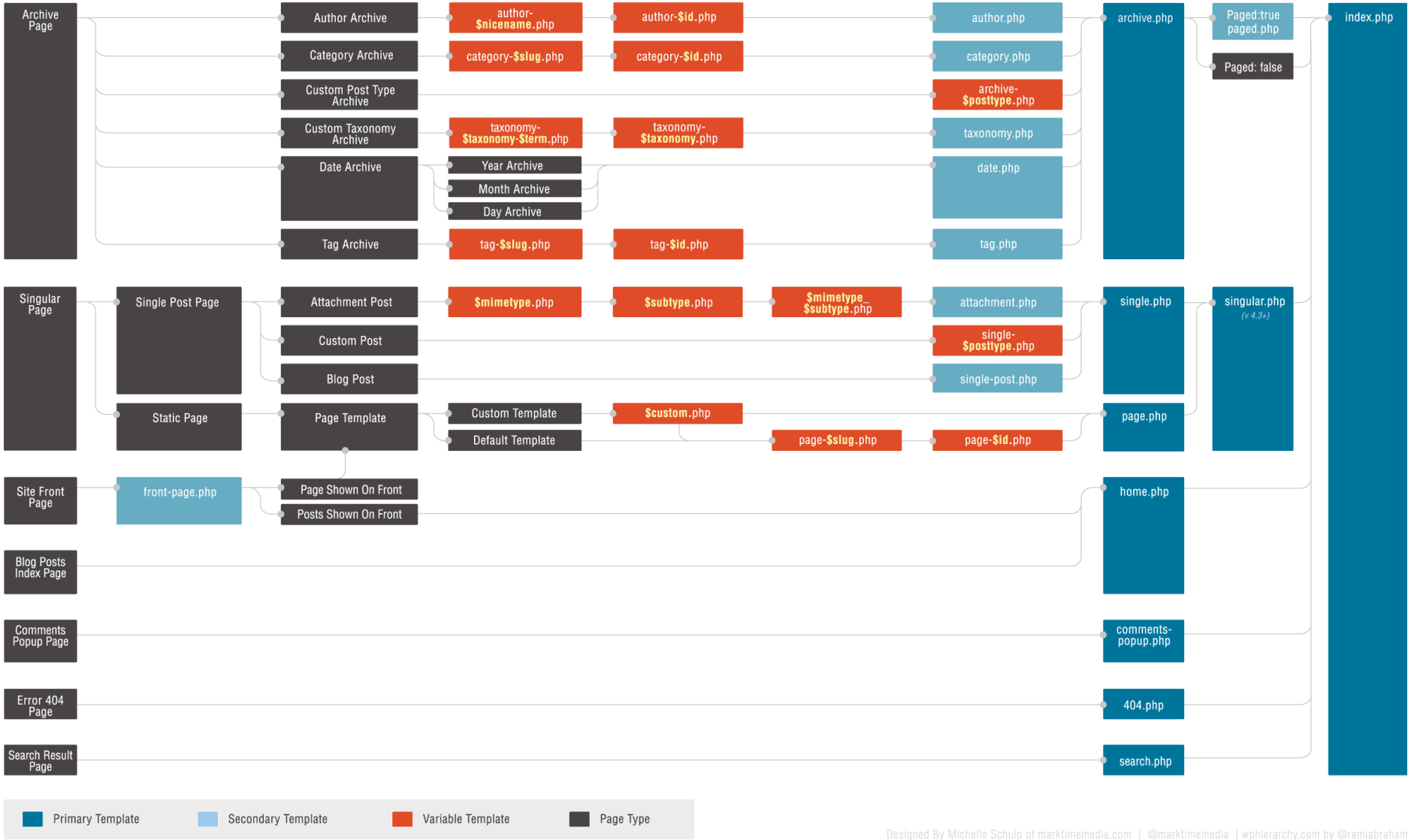
autor.

- date.php
  Se usa esta plantilla cuando un solicitan entradas de una fecha u hora determinada.
- archive.php
  La plantilla de archivo de usa para mostrar las entradas filtradas por categoría, autor o fecha
- search.php
  Esta plantilla se usa tras la búsqueda de un usuario, para mostrar los resultados coincidentes.
- attachment.php
  Plantilla a mostrar cuando se solicita un archivo adjunto
- image.php
  Plantilla a mostrar cuando se solicita un archivo adjunto que sea un imagen.
- 404.php
  Es la plantilla que se motrará cuando Wordpress no pueda encontrar la página/entrada solcitada por el usuario.

Vemos que algunas de estas plantillas parecen coincidir en los casos de uso. Por ejemplo, el caso de uso de archive.php coincide con los de category.php, author.php o date.php. ¿Entonces en qué caso se usa una plantilla o las otras? Dado que un tema no tiene por qué contar con todas estas plantillas, Wordpress usará siempre la plantilla más específica posible. En caso de que el usuario elija expresamente una plantilla para un contenido, esta prevalecerá.

La jerarquía queda clara en el diagrama que WordPress ofrece en su documentación[19] y que reproducimos en la siguiente página.

19 https://developer.wordpress.org/themes/basics/template-hierarchy/


Archive Page
Author Archive
Category Archive
Custom Post Type Archive
Custom Taxonomy Archive
Date Archive
Year Archive
Month Archive
Day Archive
Tag Archive
author-$nicename.php
category-$slug.php
taxonomy-$taxonomy-$term.php
tag-$slug.php
author-$id.php
category-$id.php
taxonomy-$taxonomy.php
tag-$id.php
author.php
category.php
archive-$posttype.php
taxonomy.php
date.php
tag.php
archive.php
Paged:true paged.php
Paged: false
index.php
Singular Page
Single Post Page
Static Page
Attachment Post
Custom Post
Blog Post
Page Template
Custom Template
Default Template
$mimetype.php
$subtype.php
$mimetype_$subtype.php
$custom.php
page-$slug.php
attachment.php
single-$posttype.php
single-post.php
page-$id.php
single.php
page.php
singular.php (v 4.3+)
Site Front Page
front-page.php
Page Shown On Front
Posts Shown On Front
home.php
Blog Posts Index Page
Comments Popup Page
comments-popup.php
Error 404 Page
404.php
Search Result Page
search.php
Primary Template
Secondary Template
Variable Template
Page Type

3.3 Plantillas de página personalizadas (Custom Page Templates)

Además de poder crear plantillas específicas para un determinado post, categoría, etc como vimos en el diagrama de jerarquía; podemos crear plantillas que aparecerán disponibles al crear/editar una página.

Para crear estas plantillas bastará con añadir un DocBlock con la etiqueta "Template Name"

```
<?php
/**
 * Template Name: Full Width Page
 *
 * @package TelWP Theme
 */

// A continuación el código de la plantilla
```

Veamos un ejemplo. Copiaremos la plantilla por defecto para las páginas (page.php) del tema "Twenty Sixteen" y lo copiaremos en el directorio raíz de nuestro tema hijo con otro nombre[20]. Después procederemos a hacer tres sencillo cambios.

Primero modificaremos el comentario de la cabecera por el código del ejemplo anterior. Después añadiremos una clase extra al principal. Por último eliminaremos el *sidebar*.

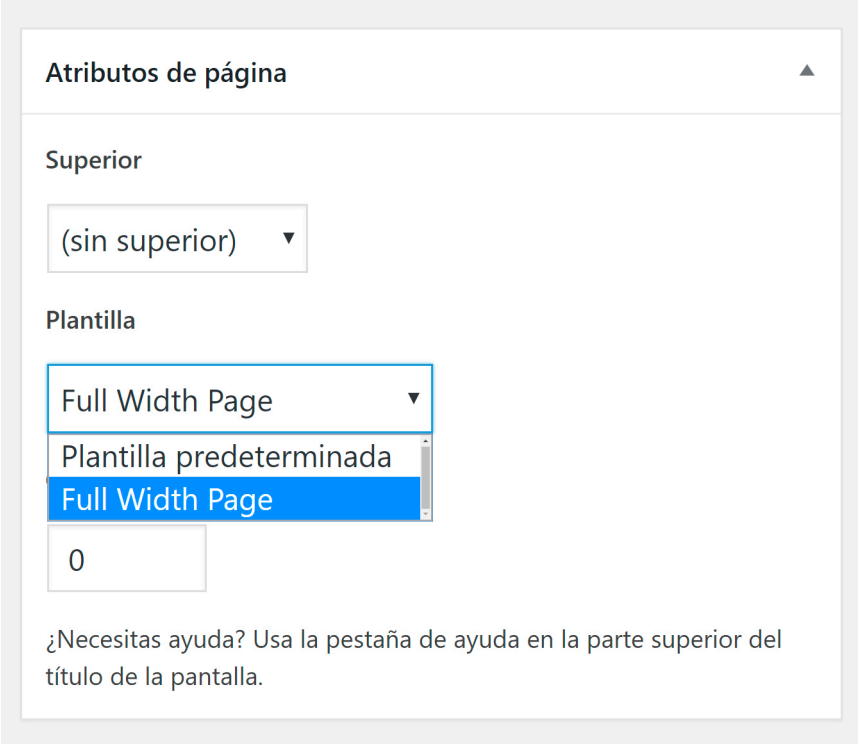
wp-content/themes/twentysixteen_child/page_full-width.php

```
<?php
/**
 * Template Name: Full Width Page
 *
 * @package TelWP Theme
 */

...
<div id="primary" class="content-area full-width">

...
<?php get_sidebar(); ?>
<?php get_footer(); ?>
```

20 Evitando los comunes. https://developer.wordpress.org/themes/basics/template-files/#common-wordpress-template-files

Tras esta sencilla operación, podremos acudir al WP-Admin y cambiar la plantilla por defecto de cualquiera de nuestras páginas por nuestra nueva plantilla de ancho completo y sin *sidebar*.

Bastará con desplegar las opciones de plantilla, realizar el cambio y guardar la modificación.



3.4 functions.php

Podemos añadir un archivo llamado functions.php en el directorio raíz de nuestro tema para crear cualquier función a la que queramos llamar en las plantillas de nuestro tema.

Aunque no es obligatorio, si es recomendable comenzar este archivo con un DocBlock con información del tema:

```
<?php
/**
* TelWP Theme custom functions.
*
* @package TelWP Theme
*/


function sayHello(){
    echo "¡Hola!";
}
```

Además, podemos asignar nuestras funciones a *hooks* como veremos a continuación.

Una buena forma de aprender a desarrollar temas es estudiar el código de otros temas, especialmente si vamos a crear un tema hijo. En las siguientes páginas vamos a ver un fragmento del archivo functions.php del tema "Twenty Sixteen", incluido por defecto en las instalaciones de WordPress.

Este fragmento hace uso de funciones como add_theme_support[21] que es usada varias veces para:

- Usar como título y logotipo de la página los que se configuren en WP-Admin.
- Habilitar las imágenes destacadas en entradas y páginas (su tamaño se configura después mediante otra función).
- Listar los formatos de post soportados

Otra importante función que aparece es register_nav_menus que permite designar los menús que usará nuestra plantilla indicando el nombre interno y el que se mostrará al usuario[22].

21 https://developer.wordpress.org/reference/functions/add_theme_support/

22 El uso de la función especial __() sirve para permitir la traducción de cadenas. El primer argumento es la cadena a traducir y el segundo el dominio o ámbito de esa traducción, que en este caso se ha configurado unas lineas más arriba mediante la función load_theme_textdomain().

wp-content/themes/twentysixteen/functions.php

```
...
if ( ! function_exists( ‘twentysixteen_setup’ ) ) :
/**
 * Sets up theme defaults and registers support for various WordPress features.
 *
 * Note that this function is hooked into the after_setup_theme hook, which
 * runs before the init hook. The init hook is too late for some features, such
 * as indicating support for post thumbnails.
 *
 * Create your own twentysixteen_setup() function to override in a child theme.
 *
 * @since Twenty Sixteen 1.0
 */
function twentysixteen_setup() {
    /*
     * Make theme available for translation.
     * Translations can be filed at WordPress.org. See: https://translate.wordpress.org/projects/wp-
themes/twentysixteen
```

```
 * If you're building a theme based on Twenty Sixteen, use a find and replace
 * to change 'twentysixteen' to the name of your theme in all the template files
 */
load_theme_textdomain( 'twentysixteen' );

...

/*
 * Let WordPress manage the document title.
 * By adding theme support, we declare that this theme does not use a
 * hard-coded <title> tag in the document head, and expect WordPress to
 * provide it for us.
 */
add_theme_support( 'title-tag' );

/*
 * Enable support for custom logo.
 *
 *  @since Twenty Sixteen 1.2
 */
```

```
add_theme_support( ‘custom-logo’, array(
      ‘height’      => 240,
      ‘width’       => 240,
      ‘flex-height’ => true,
) );

/*
 * Enable support for Post Thumbnails on posts and pages.
 *
 * @link http://codex.wordpress.org/Function_Reference/add_theme_support#Post_Thumbnails
 */
add_theme_support( ‘post-thumbnails’ );
set_post_thumbnail_size( 1200, 9999 );

// This theme uses wp_nav_menu() in two locations.
register_nav_menus( array(
      ‘primary’ => __( ‘Primary Menu’, ‘twentysixteen’ ),
      ‘social’  => __( ‘Social Links Menu’, ‘twentysixteen’ ),
) );
```

```
    ...

    /*
     * Enable support for Post Formats.
     *
     * See: https://codex.wordpress.org/Post_Formats
     */
    add_theme_support( 'post-formats', array(
        'aside', 'image', 'video', 'quote', 'link', 'gallery', 'status', 'audio', 'chat',
    ) );

    ...
}
endif; // twentysixteen_setup
add_action( 'after_setup_theme', 'twentysixteen_setup' );
```

La última linea asocia la función anterior al hook 'after_setup_theme'[23] que es el evento que se dispara nada más cargar el tema activo.

Como indica el comentario del inicio de la función, las funciones de los temas hijos se cargan antes que las de su padre. Por eso se incluye la función dentro de un condicional que comprueba que no se haya definido la función previamente (por el tema hijo).

23 https://developer.wordpress.org/reference/hooks/after_setup_theme/

Telefónica

EDUCACIÓN DIGITAL