

WordPress

Daniel Julià Lundgren

PID_00168317



Los textos e imágenes publicados en esta obra están sujetos –excepto que se indique lo contrario– a una licencia de Reconocimiento-Compartir igual (BY-SA) v.3.0 España de Creative Commons. Se puede modificar la obra, reproducirla, distribuirla o comunicarla públicamente siempre que se cite el autor y la fuente (FUOC. Fundació per a la Universitat Oberta de Catalunya), y siempre que la obra derivada quede sujeta a la misma licencia que el material original. La licencia completa se puede consultar en: <http://creativecommons.org/licenses/by-sa/3.0/es/legalcode.ca>

Índice

1. Introducción.....	5
2. Instalación de WordPress.....	7
3. Introducción al escritorio (<i>dashboard</i> o panel) de WordPress.....	12
3.1. Sobre los <i>feeds</i> automáticos	12
3.2. Apartados principales del escritorio	13
3.2.1. Escritorio	13
3.2.2. Entradas	14
3.2.3. Categorías y etiquetas	15
3.2.4. Multimedia	16
3.2.5. Enlaces (<i>links</i>)	17
3.2.6. Páginas (<i>pages</i>)	17
3.2.7. Comentarios	19
3.2.8. Apariencia (<i>appearance</i>)	20
3.2.9. <i>Plugins</i>	25
3.2.10. Usuarios	26
3.2.11. Herramientas (<i>tools</i>)	27
3.2.12. Ajustes (<i>settings</i>)	27
4. Configuración y seguridad.....	29
5. Introducción a los temas en WordPress.....	31
5.1. Algunos editores de código	31
5.2. Creación de un tema de WordPress desde cero	31
5.2.1. Anatomía de un tema	34
5.3. Creación de temas hijos (<i>child themes</i>)	36
5.3.1. Plantillas de páginas	37
5.3.2. Más sobre el <i>loop</i>	38
5.4. Temas <i>framework</i>	40
5.4.1. Derivación de un tema usando Toolbox como base	40
5.5. Ejercicio práctico. Creación de plantilla para portafolios	44
5.6. <i>Checklist</i> para crear un tema de WordPress	47
5.6.1. Algunos problemas típicos y su solución	49
5.7. Adaptar un tema de WordPress para diferentes dispositivos	50
5.8. <i>Shortcodes</i>	51
5.8.1. Funcionamiento del <i>shortcode</i> “Caption”	52
5.9. Desarrollo de un <i>plugin</i> en WordPress	53
5.9.1. Acciones	54
5.9.2. Filtros	55

1. Introducción

WordPress es probablemente el **gestor de blogs** más utilizado en estos momentos, y no solo para la creación de blogs, sino también para sitios web más genéricos (aunque posiblemente sin llegar a la flexibilidad de otros CMS más complejos como Drupal).

Ved también

Veremos este CMS en el módulo “Drupal”.

WordPress (<http://www.wordpress.com>) dispone de un servicio *online* donde cualquier usuario se puede dar de alta y crear un blog en pocos momentos. Al mismo tiempo, WordPress es un **proyecto de código abierto** con licencia GPL, de manera que cualquier persona se puede descargar el código de la aplicación –que funciona sobre la plataforma Apache+PHP+MySQL– e instalarlo en su propio servidor, con la flexibilidad que esto proporciona para, por ejemplo, permitirnos activar cualquier tema o *plugin* creado por terceras personas, y también, y sobre todo, para personalizar el código y crear temas y *plugins* nosotros mismos.

La base de esta plataforma de código abierto es <http://www.wordpress.org>. El proyecto está extensamente documentado en el llamado **codex** (http://codex.wordpress.org/es>Main_Page), el repositorio principal de documentación sobre WordPress.

Página web

La versión inglesa del codex acostumbra a estar más actualizada y a contener más información.

Existe una gran comunidad de desarrolladores que se encargan de mantener el proyecto y proporcionar infinidad de *plugins*, extensiones que permiten aumentar las prestaciones que WordPress tiene por defecto, además de temas (*themes*), que permiten adaptar la presentación y el *layout* para cada necesidad.

WordPress es una herramienta pensada desde el primer momento para la creación y administración de blogs. Por tanto, son las características asociadas a un blog las que se han tenido en cuenta en primer lugar.

Las **funcionalidades básicas** de WordPress son:

- Publicación de artículos (entradas) que se muestran desde el más reciente al más antiguo en la portada.
- Publicación de páginas “estáticas” que aparecen por defecto en el menú principal.
- Posibilidad de que los visitantes puedan añadir comentarios, con herramientas de moderación en el panel de administración.

- Publicación automática de RSS en formatos RSS 1.0, RSS 2.0 y Atom.
- *Ping* automático a servidores de notificación de cambios.
- Categorías y *tags* para etiquetar los contenidos.
- Posibilidad de cambiar fácilmente la presentación mediante *themes* (temas).
- Posibilidad de aumentar las funcionalidades instalando *plugins*.
- Gestión de *widgets*.

Desde la versión 3, se han unificado las versiones de WordPress y WordPress MU (MultiUser), ofreciendo nuevas funcionalidades como la posibilidad de crear tipos de contenidos o de actualizar varios *plugins* y temas simultáneamente, entre otras.

A continuación vamos a ver el proceso de instalación, las características principales del panel de administración y, finalmente, las posibilidades de ampliación vía creación de *plugins* y temas.

2. Instalación de WordPress

La instalación de WordPress es bastante rápida y fácil, y a veces se presenta como la “famosa instalación de 5 minutos”.

Para empezar, debemos disponer de un servidor web capaz de ejecutar PHP y de un servidor de bases de datos con MySQL (y una base de datos o la posibilidad de crear una).

Antes de instalar WordPress en nuestro servidor local o remoto, debemos asegurarnos de que el servidor cumple los requisitos mínimos (versiones de PHP y MySQL).

Requisitos (para la versión 3.3.2):

- PHP 5.2.4 o superior.
- MySQL 5.0 o superior.
- El módulo “mod_rewrite” de Apache (muy recomendable).

Una vez instalada una versión de WordPress, deberemos revisar continuamente si existen versiones nuevas. De hecho, el mismo panel de WordPress nos avisará si han aparecido. En este caso, siempre es recomendable actualizarla, por temas de seguridad pero también para poder disfrutar de nuevas funcionalidades.

Observación

Podemos instalar WordPress en el idioma que queramos (siempre que exista una traducción). Aquí lo haremos en castellano. Por otro lado, el idioma del núcleo de WordPress es independiente del texto que pueda existir en los temas o *plugins*: aunque instalamos WordPress en castellano, luego deberemos conseguir las traducciones de temas y *plugins* aparte.

Ejercicio

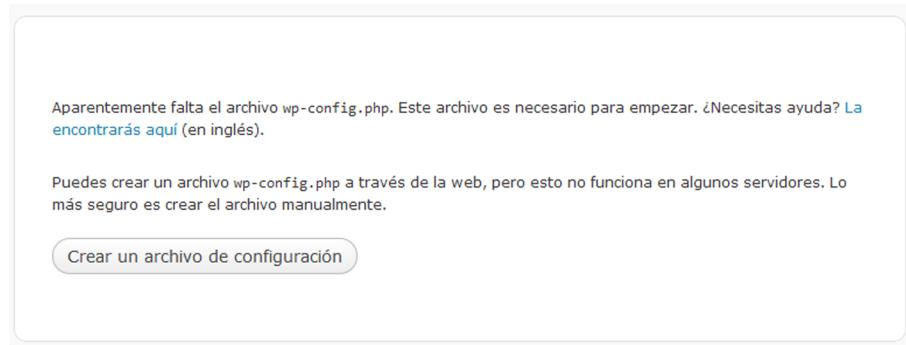
Consultar cuál es la versión actual de WordPress y si sus requisitos mínimos han cambiado desde la versión comentada en estos apuntes.

Una vez hemos comprobado los requisitos mínimos, descargamos los archivos de la página <http://wordpress.org/download/> o <http://es.wordpress.org/> y seguimos las siguientes instrucciones:

Pasos de la instalación:

- Descargar y descomprimir el zip de WordPress en el idioma que queramos instalar (aunque siempre podemos cambiar el idioma posteriormente).

- Crear una base de datos vacía para WordPress (si no disponemos ya de una) en el servidor web, asegurarse de tener los datos (usuario y *password*) de la base de datos (diferente a la cuenta del ftp), recordar el nombre que le hemos dado a esta base de datos.
- Copiar los archivos en el servidor (por ejemplo, <http://www.ejemplo.com>). Es recomendable hacerlo dentro de un directorio, por ejemplo, llamado “blog”. Después, si queremos, podemos hacer que aparezca en la raíz de nuestro dominio, tal como veremos más adelante.
- Entrar utilizando el navegador en <http://www.ejemplo.com/blog> y seguir las instrucciones.
- En el primer paso nos pedirá permiso para crear el archivo “config.php” automáticamente y en el siguiente paso, los datos de conexión con la base de datos.
- Si la instalación se ha realizado correctamente, nos llevará a una pantalla donde se nos pide el título del blog, la dirección de e-mail, el nombre de usuario y la contraseña del administrador. Es muy importante apuntar y guardar estos datos, ya que los necesitaremos cada vez que queramos acceder al panel de administración de nuestro blog.



Mensaje que nos encontramos al entrar por primera vez en la dirección de nuestro WordPress.



WORDPRESS

A continuación deberás introducir los detalles de conexión con tu base de datos. Si no estás seguro de cuales son contacta con tu proveedor de alojamiento.

Nombre de la base de datos	<input type="text" value="wordpress"/>	El nombre de la base de datos en la que quieras que se ejecute WP.
Nombre de usuario	<input type="text" value="username"/>	Tu nombre de usuario de MySQL
Contraseña	<input type="text" value="password"/>	...y la contraseña de MySQL.
Host de la base de datos	<input type="text" value="localhost"/>	Si no funciona localhost tendrás que contactar con tu proveedor de alojamiento para que te diga cual es.
Prefijo de tabla	<input type="text" value="wp_"/>	Si quieres ejecutar varias instalaciones de WordPress en una sola base de datos cambia esto.

Detalles de conexión con la base de datos.



WORDPRESS

Bienvenido

¡Bienvenido al famoso proceso de instalación de WordPress de cinco minutos! Tal vez quieras leer tranquilamente la [Documentación del archivo Léeme](#). En caso contrario, rellena los datos más abajo y en seguida estarás utilizando la plataforma de publicación personal más potente y extensible del mundo.

Información necesaria

Por favor, debes facilitarnos los siguientes datos. No te preocupes, siempre podrás cambiar estos ajustes más tarde.

Título del sitio	<input type="text" value="Mi blog"/>
Nombre de usuario	<input type="text" value="admin"/>
Los nombres de usuario sólo pueden tener caracteres alfanuméricos, espacios, guiones bajos, guiones, puntos y el símbolo @.	
Password, dos veces	<input type="text" value="....."/>
Se generará un password automático si lo dejas en blanco.	
<input type="button" value="Seguridad de la contraseña"/>	
Tu contraseña debe tener al menos siete caracteres. Para que tu contraseña sea segura, usa mayúsculas, minúsculas, números y símbolos como ! " ? \$ % ^ & .	
Tu correo electrónico	<input type="text"/>
Comprueba bien tu dirección de correo electrónico antes de continuar.	
Privacidad	<input checked="" type="checkbox"/> Permitir que mi sitio aparezca en motores de búsqueda como Google y Technorati.

Información general de nuestro sitio web y creación de la cuenta de administración.



Entrando por primera vez como administrador de nuestra plataforma WordPress.

Es posible que encontremos problemas en la instalación: por ejemplo, que WordPress no pueda crear el archivo “config.php”. En este caso, lo tendremos que editar a mano.

Observación

Si necesitamos editar el archivo “config.php”, nunca debemos usar como editor de texto aplicaciones como Microsoft Word, ya que se estos introducen caracteres indeseados. Para editar el texto de archivos de configuración, es mucho mejor usar un buen editor de código como los que se comentarán más adelante, o, en su defecto, un editor de texto simple como Notepad, en Windows, o SimpleText, en Mac.

Ejercicio

Descargar los archivos de WordPress, crear una base de datos en un servidor local e instalar WordPress siguiendo los pasos anteriores. Como servidor local puedes utilizar XAMPP, una distribución que contiene Apache, MySQL, PHP y Perl, pensada para que sea fácil de instalar y de usar en cualquier plataforma. También existen otras distribuciones como Wamp, para Windows, y Mamp, para Mac.

Haciendo clic en *login* accederemos a la pantalla de *login*, donde introduciremos los datos de la cuenta de usuario e iremos directamente al panel de administración. En esta pantalla aparece nuestro **escritorio (dashboard**, en inglés) o **panel de control**, donde tenemos acceso directo a las funcionalidades principales del CMS.

En WordPress la diferenciación entre el panel (*dashboard*) y el sitio web es muy evidente. Normalmente el *dashboard* se encontrará en la dirección wp-admin y el sitio web en la raíz, en nuestro caso en <http://www.ejemplo.com/blog/wp-admin> y <http://www.ejemplo.com/blog/>.

A la izquierda de la siguiente imagen vemos todas las opciones del panel de administración.

The screenshot shows the WordPress dashboard (Escritorio) with the following sections:

- Bienvenido a tu nuevo sitio de WordPress!**: Welcome message.
- Ajustes básicos**: Basic settings section with links to privacy, timezone, comments, and profile.
- Añade contenido real**: Content creation section with links to view example page and entry, delete example page and entry, create new page, and write first post.
- Personaliza tu sitio**: Site personalization section with links to choose theme, set background color, change header image, and add widgets.
- Ahora mismo**: Dashboard summary showing 1 Entrada, 1 Página, 1 Categoría, and 0 Etiquetas. It also mentions the Twenty Eleven theme and blocked search engines.
- Publicación rápida**: Quick publish form for a new post, including fields for Title, Content, and Tags, with Save Draft and Publish buttons.

El escritorio (dashboard) de WordPress.

The screenshot shows the WordPress website (Mi blog) using the Twenty Eleven theme, featuring:

- Mi blog**: Site title and description.
- Inicio**: Home page with a large featured image of a plant and navigation links.
- ¡Hola mundo!**: First post by "Sr WordPress" published on April 11, 2012, with 1 response.
- ENTRADAS RECIENTES**: Recent posts: ¡Hola mundo!, Sr WordPress en ¡Hola mundo!.
- COMENTARIOS RECIENTES**: Recent comments: Sr WordPress en ¡Hola mundo!.
- ARCHIVOS**: Archives: abril 2012.
- CATEGORÍAS**: Categories: Sin categoría.

Aspecto del sitio web de WordPress por defecto (con el tema "Twenty Eleven").

3. Introducción al escritorio (**dashboard** o panel) de WordPress

En WordPress, la usabilidad del panel de administración está muy cuidada y enseguida aprenderemos su funcionamiento básico, basado en entradas, páginas, categorías y *tags* (etiquetas).

Ejercicio

Una vez instalado WordPress, proponemos al alumno crear una entrada y una página de contacto. Observaremos que la entrada aparece en la portada, mientras que la página crea automáticamente una nueva opción en el menú. Además, las entradas tienen la opción de definir categorías y etiquetas, al contrario que las páginas.

La estructura básica de contenidos en WordPress está basada en:

- **Entradas.** Los artículos del blog que se pueden vincular con categorías y etiquetas.
- **Páginas.** Que aparecen automáticamente en las entradas del menú principal.
- **Categorías.** Vinculadas a las entradas. Pueden tener estructura jerárquica.
- **Etiquetas (*tags*).** Para etiquetar más específicamente las entradas.

Por supuesto, cada página o entrada tiene su URL única. La dirección por defecto de las entradas tienen el siguiente aspecto: <http://www.ejemplo.com/blog?p=xx>, donde “xx” es el id de la entrada. En el caso de las páginas, la URL es del tipo “page_id=xx”.

3.1. Sobre los *feeds* automáticos

Dado que WordPress es una plataforma de administración de blogs, uno de los contenidos que genera, además de las páginas únicas, es un *feed* de entradas y otro de comentarios. Estos *feeds* se publican en los dos formatos más utilizados, RSS y *Atom*.

RSS (*really simple syndication*) es uno de los formatos existentes de **sindicación de contenidos**. Se trata de un archivo en formato XML que se utiliza para difundir (sindicar) contenidos de manera que los lectores puedan “consumir” esos contenidos sin necesidad de utilizar un navegador: en lugar de navegador utilizan un agregador (como, por ejemplo, Google Reader).

Un **agregador** es un programa especial que lee estas fuentes (*feeds*) periódicamente, las organiza y notifica de alguna manera al usuario cuando se publican contenidos nuevos en las fuentes a las cuales está suscrito.

Además, un RSS permite sindicar los contenidos de una fuente en otro lugar.

Así, por ejemplo, en nuestro blog podríamos tener un apartado con contenidos que tienen origen en otra fuente. En WordPress lo podemos conseguir instalando algún *plugin* al respecto como FeedWordPress u otros.

Existen diferentes versiones, como RSS 0.9, RSS 1.0 y RSS 2.0.

En cuanto a **Atom**, es un formato que sirve para el mismo propósito. Se trata también de un XML, aunque con algunos de los nombres de etiquetas diferentes.

Cabe señalar que a menudo se habla genéricamente de RSS, sin tener en cuenta si se trata realmente de un archivo en formato RSS o Atom.

3.2. Apartados principales del escritorio

Vamos a analizar con más en detalle cada uno de los apartados del panel de administración (o escritorio) de WordPress:

- **Escrítorio.**
- **Entradas.**
- **Multimedia.**
- **Enlaces (links).**
- **Páginas.**
- **Comentarios** (para qué sirve Akismet).
- **Apariencia.**
- **Plugins.**
- **Usuarios** (roles).
- **Herramientas.**
- **Ajustes.**

3.2.1. Escritorio

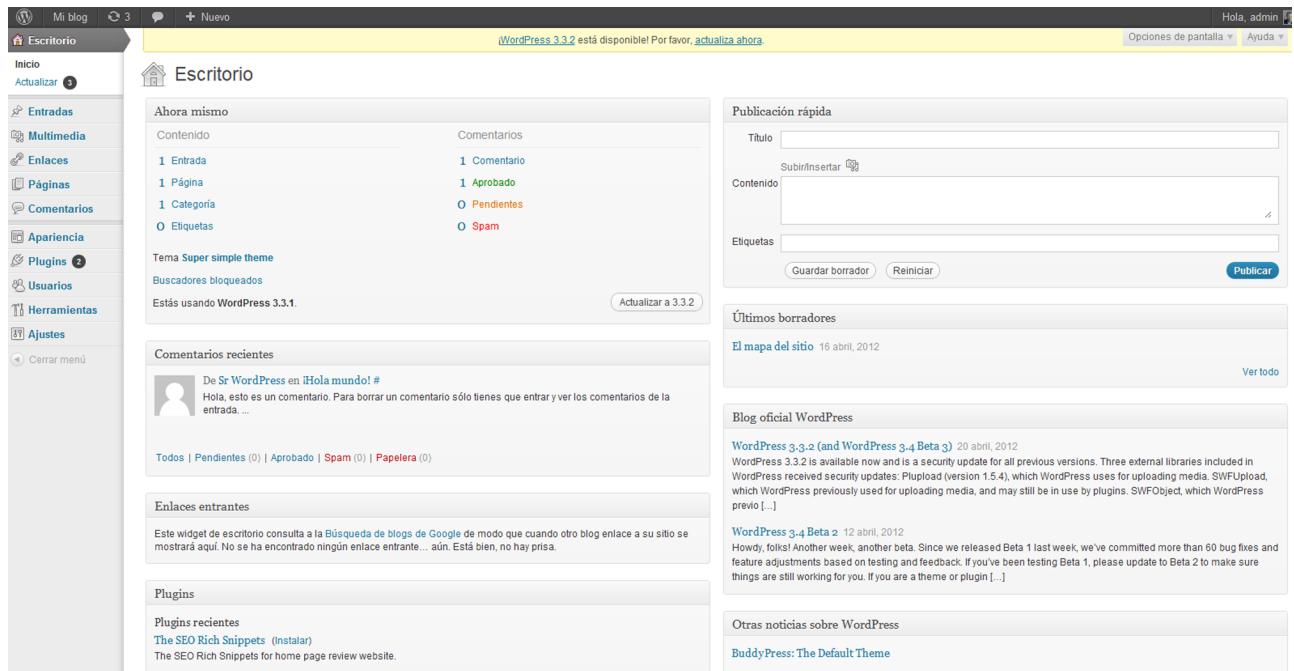
La primera opción, “Escrítorio”, nos da una **visión general** del estado de nuestro blog en cada momento –por ejemplo, cuántos artículos hemos publicado y los comentarios que hemos recibido–, además de la opción de escribir rápidamente un borrador de artículo en “Publicación rápida” (“QuickPress”).

Página web

Para más información sobre Atom, véase el *codex* en:
http://codex.wordpress.org/WordPress_Feeds.

Página web

Se puede encontrar más información sobre el tema en:
<http://en.wikipedia.org/wiki/RSS>.



El escritorio (*dashboard*) de WordPress.

El escritorio está organizado en bloques que agrupan funcionalidades, como “Comentarios recientes” o “Publicación rápida”, que se llaman ***widgets***. Estos bloques funcionales se pueden arrastrar y soltar en determinadas zonas de la pantalla para cambiarlos de orden. Podemos personalizar totalmente el escritorio arrastrando los *widgets* donde queramos.

Ved también

El mismo sistema se utiliza para construir las zonas “estáticas” de nuestro sitio web, como veremos más adelante en el apartado “Widgets”.

Además, en la opción “Opciones de pantalla” (pestana superior derecha) podemos decidir qué *widgets* se muestran en pantalla y cuántas columnas tiene el panel de administración.

3.2.2. Entradas

Es el apartado principal en que se nos muestran todas las entradas (también denominadas *posts* o artículos) que se han escrito.

Título	Autor	Categorías	Etiquetas	Fecha
El mapa del sitio - Borrador	admin	Sin categoría	Sin etiquetas	16/04/2012 Última modificación
¡Hola mundo!	admin	Sin categoría	Sin etiquetas	11/04/2012 Publicada
Título	Autor	Categorías	Etiquetas	Fecha

Apartado “Entradas” del panel de administración.

Observación

Al entrar en este apartado por primera vez aparece una entrada llamada “Hola Mundo” (o *Hello World!* en la versión inglesa). Se trata de una entrada generada automáticamente por WP como muestra y que, por supuesto, podemos borrar cuando queramos.

En el apartado “Entradas” tenemos cuatro subapartados:

- Entradas. Donde vemos todos los *posts* actuales.
- Añadir nueva (*Add New*). La opción para crear una entrada nueva.
- Categorías (*Categories*). Editar las categorías vinculadas a los *posts*.
- Etiquetas de las entradas (*Post tags*). Editar las etiquetas vinculadas a los *posts*.

Al publicar una entrada tenemos un editor de texto que nos permite dar formato al texto. Además, existe la posibilidad de subir fotos, vídeos o sonidos fácilmente utilizando la opción “Subir” (“Insert”). El editor también permite editar directamente el código HTML activando la pestaña superior derecha, “HTML”.

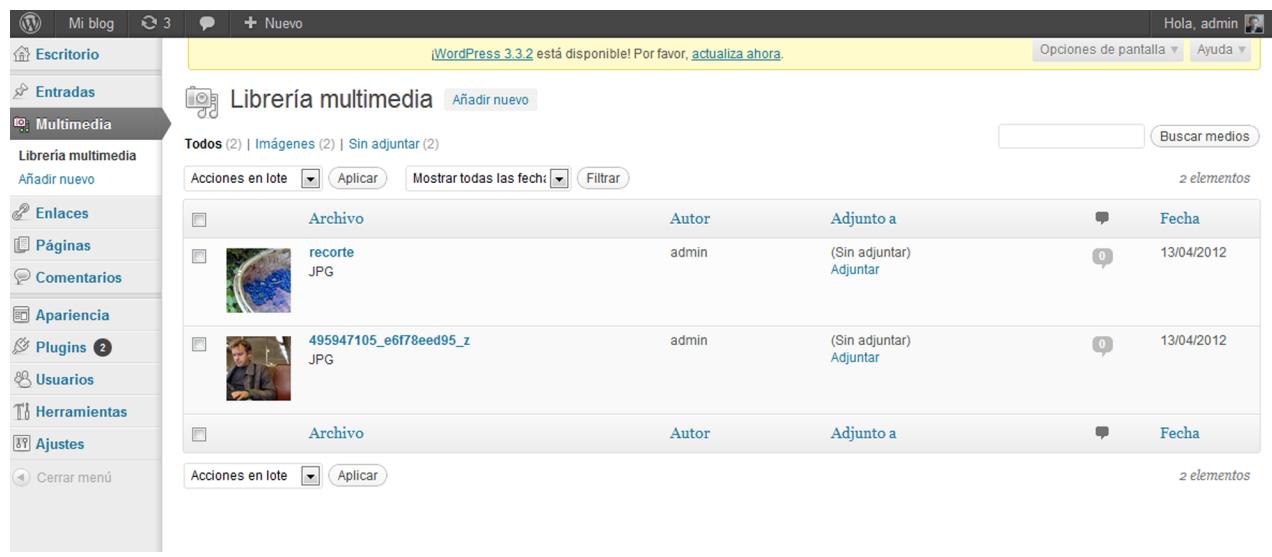
3.2.3. Categorías y etiquetas

Las **categorías** responden a las secciones generales de nuestro blog, mientras que las **etiquetas** (*tags*) son libres y nos permiten etiquetar individualmente los contenidos. Estas etiquetas y categorías ayudarán luego a los visitantes del sitio a navegar por el contenido. Además, son importantes para la indexación en los buscadores.

Siempre será una buena idea asignar una categoría y varias etiquetas a cada uno de los artículos. De esta manera, los visitantes de la web podrán navegar por el blog enlazando estas palabras y encontrar contenidos relacionados,

3.2.4. Multimedia

En “Librería multimedia” podemos ver todos los archivos que hemos subido previamente al publicar artículos (sean imágenes o cualquier otro tipo de archivos). De esta manera, los podemos editar en cualquier momento. También podemos subir más archivos (en “Añadir nuevo”) para tener recursos en el momento de publicar nuevos artículos. Así, tenemos la opción de subir directamente imágenes o archivos a la librería que luego podemos usar en nuestra entrada o, lo más práctico, en la mayoría de las situaciones, hacerlo en el momento de crear la entrada.



Acciones en lote	Archivo	Autor	Adjunto a	Fecha
<input type="checkbox"/>	 recorte JPG	admin	(Sin adjuntar) Adjuntar	13/04/2012
<input type="checkbox"/>	 495947105_e6f78eed95_z JPG	admin	(Sin adjuntar) Adjuntar	13/04/2012

Librería multimedia.

Ejercicio

Publicar un *post* con una imagen. Posteriormente, ir al apartado “Librería multimedia” y comprobar que la imagen aparece en el listado.

En el listado podemos filtrar los archivos por fecha o ver solo los que todavía no están asignados a ningún artículo.

Una de las ventajas de la librería es que se generan automáticamente versiones de la imagen en diversos tamaños –por ejemplo, en formato miniatura o en un tamaño apto para el ancho de las columnas de la entrada. De esta manera, el editor no debe preocuparse del tamaño original de la imagen. En “Ajustes” podemos cambiar los tamaños predefinidos.

Además WordPress tiene la funcionalidad de poder mostrar imágenes en formato galería. En el momento en que insertamos varias imágenes en el mismo artículo, nos ofrecerá esta opción. Aunque se puede utilizar visualmente, en realidad internamente se crea un *shortcode* “[gallery]”. Incluso es posible mostrar la galería de otra entrada introduciendo su id, “[gallery id=“5”]” por ejemplo.

Véase también

Hablaremos con mayor profundidad de los *shortcodes*, un sistema para personalizar contenidos en las entradas, en el apartado “Shortcodes”.

3.2.5. Enlaces (*links*)

Este apartado es sencillamente un listado de enlaces que podemos modificar. Cada lista de *links* está agrupada en una categoría; también podemos ampliar estas categorías: por defecto, solo existe “Sitios de interés” (“Blogroll”).

Nombre	URL	Categorías	Relación con el enlace (XFN)	Visible	Valoración
Documentation	codex.wordpress.org	Sitios de interés		Sí	0
Plugins	wordpress.org/extend/plugins	Sitios de interés		Sí	0
Suggest Ideas	wordpress.org/extend/ideas	Sitios de interés		Sí	0
Support Forum	wordpress.org/support	Sitios de interés		Sí	0
Themes	wordpress.org/extend/themes	Sitios de interés		Sí	0
WordPress Blog	wordpress.org/news	Sitios de interés		Sí	0
WordPress Planet	planet.wordpress.org	Sitios de interés		Sí	0
Nombre	URL	Categorías	Relación con el enlace (XFN)	Visible	Valoración

Apartado “Enlaces”.

Al igual que el resto de apartados, tiene contenidos de prueba que probablemente deberemos eliminar al personalizar nuestro sitio y añadir nuestros propios enlaces.

Los visitantes de nuestro blog verán este listado de enlaces en uno de los *widgets* que tenemos disponibles para construir nuestro sitio, como veremos más adelante.

3.2.6. Páginas (*pages*)

En WordPress existen de manera predefinida **dos tipos de contenidos**, las **entradas**, que ya hemos visto y que corresponden con las entradas del blog en sí mismo, y las **páginas**.

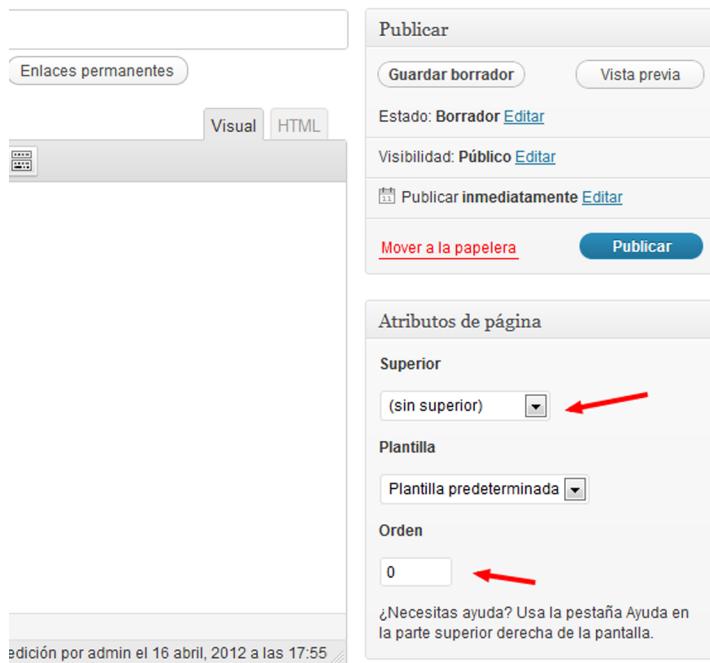
Título	Autor	Fecha
El mapa del sitio - Borrador	admin	16/04/2012 Última modificación
Página de ejemplo	admin	11/04/2012 Publicada
Título	Autor	Fecha

Apartado “Páginas”.

El apartado “Páginas” funciona exactamente igual que el apartado “Entradas”. Este tipo de contenido está pensado para páginas de contenido estático, como podría ser una sección de contacto, una página de presentación de nuestro blog, un producto, etc.

Una de las diferencias más importantes entre entradas y páginas es que estas últimas pueden tener una **estructura jerárquica** en forma de secciones y subsecciones (lo que podría servir, por ejemplo, para crear un catálogo de productos).

Para crear esta estructura jerárquica en el *widget* utilizaremos los “Atributos de la página”, que tenemos en la barra lateral derecha, para decidir en qué sección o subsección aparece cada página. Cada sección o subsección, a su vez, también es una página.



Organización jerárquica de las páginas.

En el desplegable “Superior” seleccionamos a qué página (sección) pertenece la página actual. Automáticamente, estas secciones y subsecciones aparecerán en el menú principal con menús desplegables. El campo “Orden” nos sirve para decidir el orden de las páginas en el menú.

Otra diferencia respecto a las entradas es que las páginas no tienen categorías ni etiquetas.

Ejercicio

Crear una página en el blog. Añadir esta página en el menú. Crear páginas en forma de subsecciones.

3.2.7. Comentarios

Una de las características más importantes de cualquier blog es la posibilidad de añadir comentarios por parte de los visitantes del sitio.

Apartado "Comentarios".

En este apartado podemos ver todos los comentarios del sitio y decidir cuáles van a ser visibles y cuáles queremos eliminar (por ejemplo, en el caso que se trate de *spam*).

En el subapartado "Ajustes de comentarios" del apartado "Ajustes" podemos cambiar la mecánica de aprobación. Una posibilidad será que los comentarios no se publiquen hasta que el administrador no los apruebe, pero existen otros procedimientos, como dejar publicar a los usuarios registrados, etc.

WordPress incluye por defecto un *plugin* muy recomendable llamado Akismet que, una vez activado, marca automáticamente como *spam* los comentarios maliciosos y permite la aprobación automática del resto. Existen muchos otros *plugins* que pueden ayudar en esta tarea, como por ejemplo Bad behaviour (<http://wordpress.org/extend/plugins/bad-behavior/>).

3.2.8. Apariencia (*appearance*)

Este es uno de los apartados más importantes del panel de administración o escritorio y está dividido, como mínimo, en tres subapartados:

- Temas.
- Widgets.
- Editor.

Según sea la implementación del tema que tengamos activada, aquí pueden aparecer otras opciones (que afectarán el aspecto de menús, fondo, cabecera, etc.).

Temas

Como buen gestor de contenidos, WP separa totalmente el contenido de la presentación.

La presentación es tarea de los **temas (themes)**.

Tema actual

Twenty Eleven 1.3 por [the WordPress team](#)

The 2011 theme for WordPress is sophisticated, lightweight, and adaptable. Make it yours with a custom menu, header image, and background — then go with available theme options for light or dark color scheme, custom link colors, and three layout choices. Twenty Eleven comes equipped with a Showcase template that transforms your front page into a showcase to show off your best content, widget support galore (sidebar, three footer areas, and a Showcases widget area), and a custom "Ephemera" widget to display your Aside, Link, Quote, or Status posts. Included are styles for print and for the admin editor, subject to featured images (as custom header images on posts and pages and as large images on featured "sticky" posts), and special styles for six different post formats.

OPCIONES: [Widgets](#) | [Menús](#) | [Opciones del tema](#) | [Fondo](#) | [Cabecera](#)

Etiquetas: dark, light, white, black, gray, one-column, two-columns, left-sidebar, right-sidebar, fixed-width, flexible-width, custom-background, custom-colors, custom-header, custom-menu, editor-style, featured-image-header, featured-images, full-width-template, microformats, post-formats, rtl-language-support, sticky-post, theme-options, translation-ready

Temas disponibles

Esplanade 1.0.5 por One Designs
A stylish, modern and flexible theme with responsive layout. Includes several custom templates, layouts and color schemes to choose from, 5 widget-ready areas and a user friendly options page to keep everything in control.
[Activar](#) | [Vista previa](#) | [Borrar](#)
Todos los archivos de este tema se encuentran en [/themes/esplanade](#).
Etiquetas: blue, brown, gray, green, tan, white, light, one-column, two-columns

Mi tema derivado de Toolbox por Dani
Este es un tema de ejemplo derivado de Toolbox.
[Activar](#) | [Vista previa](#) | [Borrar](#)
Los archivos de la plantilla están situados en [/themes/toolbox](#). Los archivos de la hoja de estilos están situados en [/themes/mitema](#). Mi tema derivado de Toolbox utiliza plantillas de Toolbox. Los cambios que se hagan a las plantillas afectarán a ambos temas.

Super simple theme 1 por Tu nombre
Explicación sobre mi tema
[Activar](#) | [Vista previa](#) | [Borrar](#)
Todos los archivos de este tema se encuentran en [/themes/simple](#).
Etiquetas: chulo, super, guay, azul

Apartado "Temas" del escritorio de WordPress.

Básicamente, existen tres maneras de instalar temas en WP:

- Autoinstalador. La manera más sencilla. Simplemente activar la pestaña "Instalar temas", buscar el tema (mediante palabras clave o características) y activarlo.
- Subir zip. También tenemos la opción de subir un archivo zip que contenga todos los archivos del tema. WP descomprimirá y copiará automáticamente el tema en el directorio correspondiente.
- Manual mediante FTP. Subir por FTP el tema en el directorio "wp-content/themes".

En caso de que no tengamos acceso al tema que queremos mediante el autoinstalador, siempre podemos descargar los temas que queramos desde diferentes repositorios disponibles en la web y usar la segunda o tercera opción para instalarlos.

En cualquier caso, el resultado será siempre el mismo: una vez instalado un tema tendremos un directorio en “wp-content/themes” con los archivos correspondientes a ese tema. Estos archivos en gran parte son plantillas pensadas para presentar el contenido de WordPress.

Por supuesto, podemos instalar cualquier tema, no solo necesariamente los disponibles en el sitio oficial (aunque solo los temas catalogados aquí están disponibles al hacer búsquedas en el autoinstalador de WP). En Internet encontraremos diversos repositorios de temas, como <http://spyrestudios.com/wordpress-powered-websites/>. En algunos de ellos, incluso se comercializan temas y, en contrapartida, se ofrece soporte técnico para su mantenimiento.

Una vez hemos encontrado el tema que queremos utilizar, el procedimiento será:

- Descargar el zip.
- Subir el zip directamente o descomprimir los archivos y copiar la carpeta del tema en “wp-content/themes”.
- Ir al panel de administración, en el apartado “Apariencia”, y activarlo.

Ejercicio

Instalar y activar un tema descargado a elección del alumno.

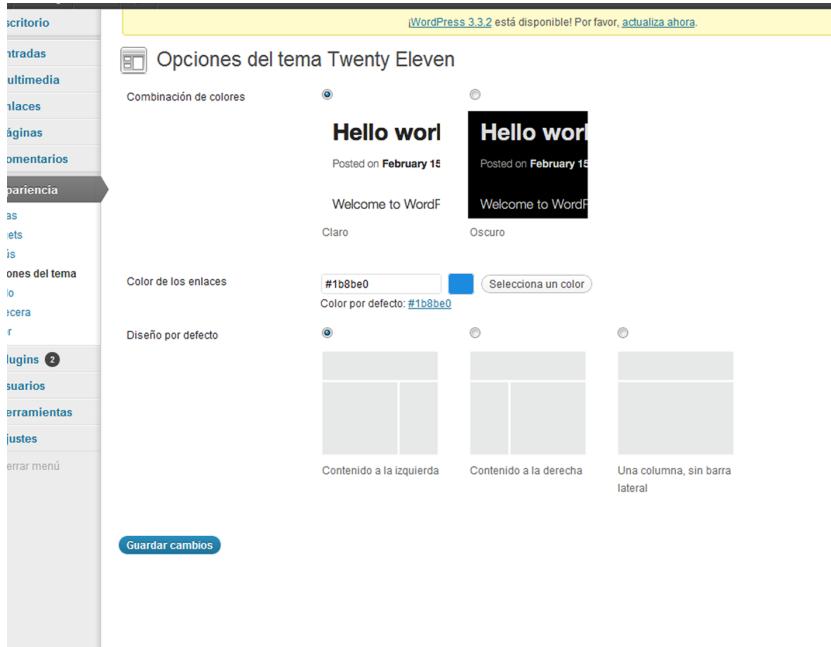
Finalmente, en este apartado es posible que encontremos opciones específicas de los temas que ya tengamos activados. Un tema puede ser tan complicado como podamos imaginar e incluir diversas opciones para personalizarlo.

Personalización del tema “Twenty Eleven”

“Twenty Eleven” es uno de los temas que vienen por defecto instalados en la distribución de WordPress. Es un tema que podemos utilizar como base para personalizar el nuestro. Tiene diferentes opciones que podemos cambiar desde el escritorio de WordPress.

Ved también

Veremos cómo podemos crear nuestros propios temas o personalizarlos a partir de un tema base en el apartado “Introducción a los temas en WordPress”.



Opciones de personalización del tema “Twenty Eleven”.

Por ejemplo podemos cambiar la combinación de colores (fondo texto), el color de los enlaces e incluso el *layout* básico.

Ejercicio

Activar el tema “Twenty Eleven”, cambiar las opciones de manera que el fondo del tema sea negro y escoger un color diferente para los enlaces. Comprobar el resultado.

Widgets

Los **widgets** son una manera sencilla e intuitiva de cambiar fácilmente la apariencia (y estructura) de nuestro sitio web. En el apartado “Widgets” podemos arrastrar y soltar (*drag & drop*) diferentes bloques en las regiones que tengamos definidas en el tema actual.

WordPress tiene de partida una serie de *widgets* predeterminados, como por ejemplo “Buscar”, “Comentarios recientes”, “Entradas recientes”, etc., y al activar determinados *plugins* pueden aparecer más.

Ejercicio

Arrastrar el *widget* “Calendario”, que muestra un calendario con la fecha de las entradas, y comprobar el resultado en la vista del sitio.

Cada tema tiene definidas una serie de regiones donde podemos arrastrar estos *widgets*. En el caso de “Twenty Eleven”, por ejemplo, existen cinco zonas (incluidas las tres columnas del *footer*). En otros temas podemos tener más o, incluso, solo una barra lateral en algún tema simple.

La definición de estas zonas se hace en el archivo functions.php del tema utilizando la función “register_sidebar” del *codex* (http://codex.wordpress.org/Function_Reference/register_sidebar).

Una de las novedades de la versión 3 de WP es la posibilidad de crear tantos menús personalizados como queramos y asignarlos a las regiones utilizando el widget “Menú personalizado”; incluso podemos utilizarlos en lugar del menú principal predeterminado.

Por otro lado, y según el tema que tengamos activado, podemos tener el subapartado “Opciones del tema”. Es el caso del tema por defecto (“Twenty Eleven”), donde podemos cambiar la configuración básica de colores y el *layout*.

Observación

Sin embargo en el anterior tema por defecto de WP, “Twenty Ten”, estas opciones no existen.

Si el tema lo permite, podemos, por ejemplo, cambiar fácilmente la imagen de fondo o de la cabecera. En realidad, todas estas opciones se definen en el archivo “functions.php” de cada tema.

Por ejemplo, para activar la opción de configuración de la imagen de cabecera tendríamos:

```
add_theme_support( 'custom-header' );
```

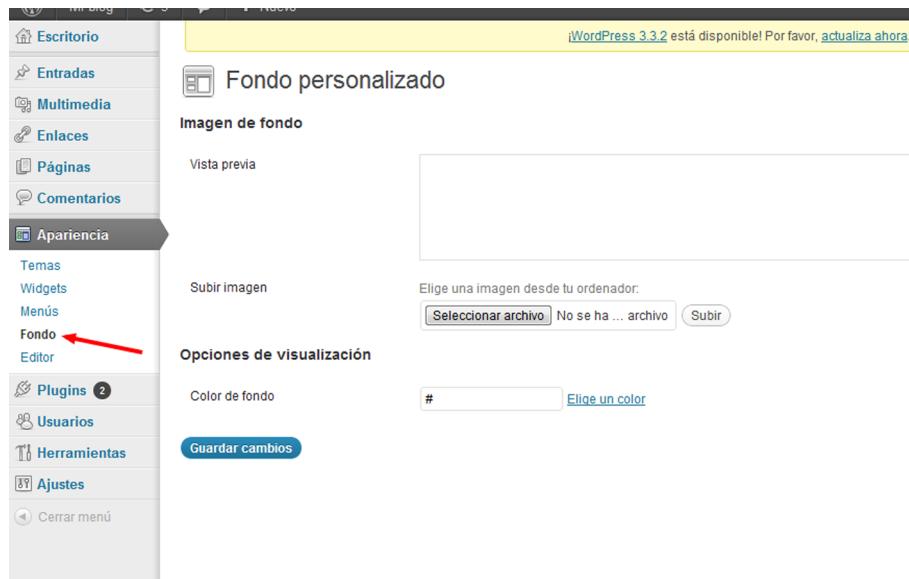
Página web

Para más información, véase el *codex* (http://codex.wordpress.org/Custom_Headers).

Mientras que la instrucción

```
add_theme_support( 'custom-background' );
```

activaría, por una parte, la opción “Fondo” en el panel de administración que permitiría cambiar la imagen, y, por otra, asociaría la imagen como estilo “background-image” en el *body* del tema.



Opción “Apariencia/Fondo” una vez activada en “functions.php”.

Editor

En el apartado “Editor” podemos editar todos los archivos que componen el tema, lo que permite la personalización avanzada de los temas. Por este motivo, esta opción (igual que otras avanzadas) solo es visible para el administrador pero no, por ejemplo, para el rol “Editor”.

Es una herramienta pensada para incluir fragmentos de código HTML o PHP rápidamente o para modificar el CSS, pero en la mayoría de los casos **no es una opción recomendable**. Siempre es mucho mejor utilizar un buen editor que nos ayudará en la sintaxis y evitará cometer errores.

3.2.9. Plugins

Los *plugins* son extensiones que nos permiten añadir funcionalidades a WP, como crear una tienda con su compleja administración, añadir estadísticas, mejorar el sistema de comentarios, implementar una red social, etc.

Existen infinidad de *plugins* para WordPress. En la página web <http://wordpress.org/extend/plugins/> podemos encontrar más de 10.000.

Igual que los temas, se pueden instalar directamente desde el escritorio de WP o manualmente en el directorio “wp-content/plugins”.

Algunos *plugins* recomendados:

Ved también

Veremos cómo funcionan los roles en WordPress en el apartado “Usuarios”.

Observación

Hay que tener mucho cuidado al editar estos archivos, ya que si borramos o alteramos algo importante es posible que el tema deje de funcionar.

Ved también

Veremos estos editores en el apartado “Algunos editores de código”.

- Google XML sitemaps (<http://wordpress.org/extend/plugins/google-site-map-generator/>). Crea un archivo “sitemap”.
- Database manager (<http://wordpress.org/extend/plugins/wp-dbmanager/>). Gestiona copias de seguridad en la base de datos.
- Feedburner Feedsmith (<http://wordpress.org/extend/plugins/feedburner-setting/>). Integra WP con Feedburner.
- W3 Total Caché (<http://wordpress.org/extend/plugins/w3-total-cache/>). Mejora la caché.
- Custom posts per page (<http://wordpress.org/extend/plugins/custom-posts-per-page/>). Permite modificar el número de entradas en diferentes situaciones.
- All in one SEO pack (<http://wordpress.org/extend/plugins/all-in-one-seo-pack/>). Optimización para buscadores.
- Google Analytics (<http://wordpress.org/extend/plugins/google-analytics-for-wordpress/>). Integra Google Analytics.
- Facebook (<http://wordpress.org/extend/plugins/simple-facebook-connect/>). Integra Facebook.

Ved también

Veremos los *plugins* con mayor profundidad en el apartado “Desarrollo de un *plugin* en WordPress”.

3.2.10. Usuarios

WordPress es un **sistema multiusuario**. En este sistema cada usuario tiene asignado un rol determinado. Un **rol** define la lista de permisos que cada tipo de usuario tiene activados. Un **permiso** puede ser, por ejemplo, “Editar una entrada”, “Publicar una entrada”, “Moderar los comentarios”, etc.

El rol más básico, implícito, es el usuario anónimo: un usuario que no se ha identificado. Muchas veces se habla del usuario anónimo de la web, ya que en muchos casos suele ser el usuario más habitual.

Una manera de comprobar el comportamiento de nuestro sitio según el rol que tenga el usuario, es usar dos navegadores diferentes (por ejemplo, Chrome y Firefox u Opera), de manera que en uno de los navegadores podemos conservar el usuario administrador y en otro estar en una sesión como usuario anónimo o con otro rol diferente.

WordPress define una serie de roles por defecto que, ordenados de menos a más permisos, son los siguientes:

- **Subscriptor.** Se ha registrado en el sitio. Puede escribir comentarios utilizando su cuenta, pero solo tiene permisos de lectura para el resto.
- **Colaborador.** Tiene acceso a algunas funciones de administración del sitio, como los comentarios, páginas, etc.
- **Autor.** Puede crear entradas y publicarlas.
- **Editor.** Puede crear entradas pero no publicarlas.
- **Administrador.** Tiene permisos para hacer cualquier cosa.
- **Super Administrador.** En el caso de instalaciones WordPress *multi-site*, existe este nivel superior.

Imaginemos, por ejemplo, el caso de un blog con varios autores: probablemente existirán varios usuarios con el rol “Autor”, pero también al menos un usuario con el rol “Administrador”.

En “Usuarios” tenemos la opción de editar la cuenta de los usuarios ya existentes o añadir nuevos con diferentes roles. Los permisos asignados a cada uno de los roles están perfectamente documentados en el *codex* (http://codex.wordpress.org/Roles_and_Capabilities).

3.2.11. Herramientas (tools)

Además de la herramienta “Publicar esto” que podemos arrastrar a la barra de favoritos de nuestro navegador, tenemos la opción de importar/exportar datos a o desde otros CMS. Al instalar *plugins* es posible que aquí aparezcan más opciones.

3.2.12. Ajustes (settings)

En este apartado podemos encontrar todas las opciones de configuración. Resaltaremos las más importantes:

- **Generales.** El título del sitio y las opciones generales que podemos cambiar en cualquier momento.
- En **Escritura (Writing)** tenemos opciones que nos permiten publicar en el blog remotamente, desde otros sistemas o servicios (como flickr) o desde correo electrónico. En este apartado también vemos al final la opción “Update services”, en la que tenemos una lista de los servidores a los cuales WP avisa cada vez que publicamos contenido nuevo en nuestro blog.

- En **Lectura** (*Reading*) tenemos varias opciones relativas a cómo se muestran los artículos en nuestro sitio
- En **Comentarios** tenemos diversas opciones relativas al flujo de publicación de las entradas y comentarios, y maneras de mostrar los avatares de los autores.
- En **Multimedia** (*Media*) tenemos opciones relativas a cómo se recortan las imágenes cuando las subimos.
- **Privacidad.** Permite dar acceso a los motores de búsqueda (como Google) o bloquearlo.
- En **Enlaces permanentes** (*Permalinks*) podemos definir cómo se construyen las rutas de las páginas de nuestro blog (es recomendable activarlo).

Permalinks y “.htaccess”

Los enlaces a las páginas en WP por defecto tienen el siguiente aspecto “/?p=123”, donde el número representa el id de la entrada (o de la página). En el caso del resto de páginas también se pasan parámetros por la URL, por ejemplo “/?cat=4” corresponde con el listado de entradas de la categoría con id = 4.

Sin embargo, tanto a nivel de posicionamiento web como de usabilidad de cara al usuario es mucho mejor utilizar enlaces permanentes del tipo “/pagina_ejemplo”. De esta manera, al compartir el enlace el usuario tiene una ligera idea del contenido.

De hecho es posible elegir en “Ajustes/Enlaces permanentes” diferentes estructuras para los enlaces combinando el título del contenido con la fecha. Para que esto funcione WordPress genera un archivo “.htaccess” que permite al servidor web Apache “reescribir” las URL. Los archivos “.htaccess” permiten activar opciones del servidor web Apache, en este caso se trata de redirecciones, pero podría ser cualquier otra opción.

Página web

Para más información sobre enlaces permanentes y “.htaccess”, véase el *codex* en: http://codex.wordpress.org/Using_Permalinks.

```
# BEGIN WordPress
<IfModule mod_rewrite.c>
RewriteEngine On
RewriteBase /sandbox/wordpress/
RewriteRule ^index\.php$ - [L]
RewriteCond %{REQUEST_FILENAME} !-f
RewriteRule . /sandbox/wordpress/index.php [L]
</IfModule>
# END WordPress
```

4. Configuración y seguridad

A pesar de las grandes ventajas de las aplicaciones de código abierto, también debemos ser conscientes de los posibles problemas de seguridad y estar pendientes siempre de las actualizaciones.

En este sentido se recomienda:

- **Actualizar a menudo.** Actualizar siempre que aparezcan actualizaciones de seguridad importantes. Además, tendremos la precaución de hacer una copia de seguridad de la base de datos previamente, por si algo pudiera ir mal. Por otro lado, las actualizaciones nos permitirán disfrutar de posibles nuevas funcionalidades.
- **Cambiar la cuenta del administrador.** Por seguridad es recomendable no usar “admin” como usuario administrador. Es preferible usar un nombre arbitrario y un *password* suficientemente robusto. También es recomendable cambiar la contraseña de vez en cuando.
- **Servidor de pruebas y de producción.** Es recomendable, especialmente si la instalación está muy personalizada, tener un servidor de pruebas y otro de producción. Antes de subir los cambios al servidor de producción nos aseguraremos de que todo está correcto en el servidor de pruebas. En el caso de que estemos desarrollando un tema o un *plugin* también sería recomendable utilizar algún sistema de control de versiones, como git (<http://git-scm.com/>).
- **Cambiar la ubicación aparente de WordPress.** Al esconder la ubicación real de WordPress conseguiremos que sea mucho más difícil un posible acceso no autorizado, además de tener los archivos más ordenados en el caso de que tengamos otras aplicaciones en el servidor. Por ejemplo, podemos apuntar WordPress a la raíz de nuestro servidor aunque realmente los archivos estén en el directorio “Blog”. El proceso consiste en dos pasos, primero mover el archivo index.php a la ubicación donde queremos que se muestre (por ejemplo, en la raíz) y después editar la inclusión del archivo wp-blog-header.php para que la ruta sea correcta. Además, previamente debemos modificar la información del sitio en “Ajustes”.
- **Filosofía *Don't hack core*.** Siempre que queramos modificar algo, debemos hacerlo solo en los archivos de nuestro tema, y si tenemos una necesidad muy específica, desarrollar un tema o *plugin* a medida. **Nunca debemos modificar los archivos principales de WordPress** (o de cualquier otro CMS). De esta manera, al actualizarlo no perderemos los cambios. Además,

Página web

El proceso está explicado en detalle en la siguiente página web:

<http://www.optiniche.com/blog/145/wordpress-tutorial-install-wordpress-in-a-different-directory/>.

modificar los archivos principales de WordPress también podría implicar riesgos de seguridad.

5. Introducción a los temas en WordPress

Un **tema** de WordPress consiste en un conjunto de **plantillas** php que contienen la estructura y se encargan de generar el contenido dinámico y el CSS que se encarga de la presentación.

Vamos a ver en detalle cuáles son estas plantillas y hasta qué punto las podemos modificar.

5.1. Algunos editores de código

Para crear nuestro propio tema de WordPress, necesitaremos un buen editor de HTML, CSS y PHP. A continuación proponemos algunos:

- **Dreamweaver** (<http://www.adobe.com/products/dreamweaver.html>). Un potente editor de Adobe con funciones avanzadas como el MultiScreen View. Para plataformas Windows y Mac.
- **Notepad ++** (<http://notepad-plus-plus.org/>). Editor de código fuente. Para plataforma Windows.
- **Smultron** (<http://www.peterborgapps.com/smultron/>). Editor web sencillo para la plataforma Mac. Se pueden encontrar en la Mac App Store.
- **Aptana Studio** (<http://www.aptana.com/>). Es una evolución del editor de código abierto Eclipse adaptado al desarrollo web. Preparado para editar HTML, CSS y JavaScript, además de PHP. Para plataformas Windows, Mac y Linux,
- **Netbeans** (<http://netbeans.org/>). Editor avanzado para PHP y otros lenguajes. Para plataformas Windows, Linux y Mac OS X.

5.2. Creación de un tema de WordPress desde cero

Aunque el proceso de creación de un tema desde cero puede ser laborioso, y en la mayoría de los casos es preferible reutilizar un tema ya existente como se comentará más adelante, se trata de un buen ejemplo para comprender el sistema de gestión de temas en WordPress.

En realidad el procedimiento para crear un tema simple es muy sencillo. Los pasos son los siguientes:

- Crear una carpeta dentro de “wp-content/themes” con el nombre que queramos.
- Crear un archivo “screenshot.png” de 300px × 225px y copiarlo dentro de esta carpeta (opcional, esta es la imagen que aparece de muestra en el administrador).
- Crear un archivo css llamado “style.css” con el siguiente contenido:

```
/*
Theme Name: El nombre de mi tema
Theme URI: http://lapaginademitema.com
Description: Explicación sobre mi tema
Version: 1
Author: Tu nombre
Author URI: http://mipaginaweb.com
Tags: doscolumnas, azul, arial
*/
```

- Finalmente, crear un archivo “index.php”, que será la plantilla principal de nuestro tema. Dentro de “index.php” deberemos escribir unas instrucciones de php para que el contenido sea dinámico.

Estas instrucciones específicas de WordPress tienen el nombre ***The loop*** (el bucle) y son la **base del funcionamiento de los temas en WP**. El *loop* se basa en un bucle que busca los contenidos que deben aparecer en cada página (según estemos en la portada, en la página única de una entrada, en el listado de una categoría, etc.) y muestra los campos que necesitemos. Los campos empiezan siempre por el artículo inglés *the*:

- `the_permalink()`. Muestra el enlace a la página única del contenido.
- `the_title()`. Muestra el título del contenido.
- `the_content()`. Muestra el contenido.
- Etc.

Un ejemplo de plantilla muy sencilla podría ser:

```
<?php get_header(); ?> <!-- incluye la cabecera -->
<?php if ( have_posts() ) : ?>
<!-- si hay contenido en esta página -->
<?php while ( have_posts() ) : the_post(); ?>
<!-- mientras haya contenidos -->
<a href="<?php the_permalink() ?>">
<!-- crea un enlace con el permalink -->
<?php the_title(); ?></a> <!-- el título del contenido -->
<?php the_content(); ?> <!-- el contenido -->
<?php endwhile; ?>
<?php endif; ?>
<?php get_sidebar(); ?> <!-- incluye el sidebar -->
<?php get_footer(); ?> <!-- incluye el footer -->
```

Observación

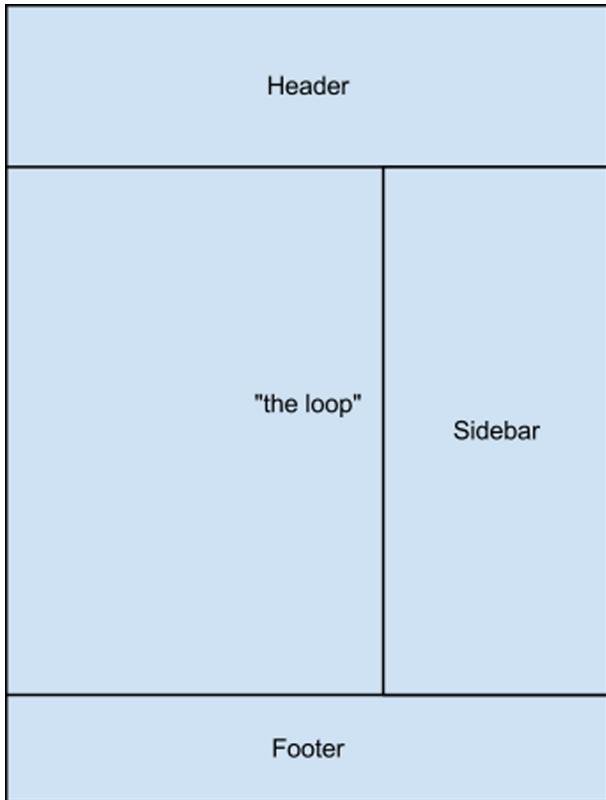
Como solo hemos definido una única plantilla “index.php”, todas las páginas se verán con esta estructura, es decir, que **todos los contenidos utilizarán esta plantilla**. Además, como no hemos añadido ningún código CSS tampoco existirá estilo ni *layout*. Sin embargo, el sitio web es totalmente funcional y se puede navegar por todos los contenidos; WordPress, “mágicamente”, se encarga de reproducir los contenidos dinámicamente.

Ejercicio

Seguir los pasos anteriores para crear un tema y activarlo.

En un tema más elaborado encontraríamos plantillas específicas para las páginas (page.php), entradas (single.php), categorías (category.php), etc. Si quisieramos evolucionar más el tema deberíamos añadir estas plantillas adicionales. Probablemente, en la plantilla de una página no nos interesa mostrar la fecha y el autor, en cambio en la plantilla de una entrada, sí.

En la mayoría de plantillas principales se incluyen subplantillas. Por ejemplo, cuando WordPress encuentra la instrucción “get_header”, en realidad mira si tenemos definido un archivo “header.php”; si no lo encuentra, utiliza un *header* por defecto. Lo mismo ocurre con el *sidebar* y el *footer*. Para personalizar realmente nuestro tema deberíamos definir estas subplantillas, “header.php”, “footer.php” y “sidebar.php”.



Esquema de una plantilla principal.

El problema de crear un tema desde cero es que, aunque tenemos flexibilidad total, **deberemos testear el tema a fondo** antes de darlo como bueno. Al menos deberíamos comprobar:

- Que el HTML y CSS sean válidos.
- Que no existen errores de JavaScript.
- Testearlo en diferentes navegadores, como IE7, IE8, IE9, Safari, Chrome, Opera y Firefox, en el escritorio, más una buena selección de navegadores en dispositivos.
- Eliminar comentarios innecesarios que podamos haber dejado en los archivos.

Estas comprobaciones no son tan necesarias si utilizamos el procedimiento de crear un tema hijo derivando uno ya existente que enseguida veremos.

5.2.1. Anatomía de un tema

Más en detalle, las **principales plantillas** que forman parte de un tema son las que se pueden ver en el siguiente gráfico marcadas en naranja:

Commonly Used WordPress Theme Files

STANDARD (used in most themes)	SPECIAL (optional additions)	CORE (required)	JUNK (legacy, don't use)
404.php	Error page, served up when someone goes to a URL on your site that doesn't exist		
archive.php	Page that displays posts in one particular day, month, year, category, tag, or author		
archives.php	Page template that includes search form, category list, and monthly archives (requires page using it)		
comments-popup.php	If you enable popup comments (obscure function), the comments link will use this template		
comments.php	This file delivers all the comments, pingbacks, trackbacks, and the comment form when called		
footer.php	Included at the bottom of every page. Closes off all sections. (Copyright, analytics, etc)		
functions.php	File to include special behavior for your theme.		
header.php	Included at the top of every page. (DOCTYPE, head section, navigation, etc)		
image.php	If you wish to have unique pages for each of the image on your site (for credits, copyright)		
images	FOLDER - Keeps all the images that make up your theme in one place		
index.php	This is typically the "homepage" of your blog, but also the default should any other views be missing		
links.php	Special page template for a home for your blogroll		
page.php	Template for Pages, the WordPress version of static-style/non-blog content		
rtl.css	A special CSS file for your optional inclusion to accommodate "right to left" languages		
screenshot.png	This is the image thumbnail of your theme, for help distinguishing it in the Appearance picker		
search.php	The search results page template		
sidebar.php	Included on pages where/when/if you want a sidebar		
single.php	This file displays a single Post in full (the Posts permalink), typically with comments		
style.css	The styling information for your theme, required for your theme to work, even if you don't use it		

Fuente: <http://jaredheinrichs.com/commonly-used-wordpress-theme-files.html>.

En esencia cualquiera de las plantillas principales de WordPress está compuesta de **subplantillas**. “**index.php**”, por ejemplo, incluye las siguientes subplantillas:

- header.php (la cabecera).
- sidebar.php (el *sidebar*).
- footer.php (el *footer*).

Además, todas las plantillas tienen las instrucciones del *loop* que permiten a WordPress obtener el contenido. La instrucción

```
<?php get_header(); ?>
```

obtiene en realidad el archivo “header.php” y lo incluye. Entonces, por qué, por ejemplo, no utilizar directamente

```
require "header.php"
```

La respuesta es que esta llamada de WordPress toma decisiones, como por ejemplo incluir el *header* del tema padre (si no está definido en el tema actual) o incluir una cabecera por defecto si no está incluida tampoco en el padre, además de realizar otras operaciones internas de WordPress.

Aunque index.php es la plantilla principal, existen **otras plantillas importantes**:

- **single.php:** Si está definida, es la plantilla para las entradas.
- **page.php:** Si está definida, es la plantilla para las páginas.
- **archive.php:** Es la plantilla de todos los listados de archivo (listado por fechas, por categorías, por *tags*, etc.).
- **category.php y tags.php:** Estas son plantillas más específicas. Si el listado que se muestra es de una categoría o un *tag* y están definidas, se utilizarán estas en lugar de archive.php.

El sistema de plantillas funciona de forma jerárquica. Dada una dirección concreta en la estructura de contenidos de nuestro WordPress, por ejemplo el listado de la categoría “Dibujo” que tenemos definida, WordPress seguirá la siguiente secuencia: en primer lugar buscará si existe el archivo “category-dibujo.php”; en caso contrario, buscará “category-xx.php” (donde “xx” es el id de la categoría); después buscará “category.php”, y si no la ha encontrado rastreará “archive.php”. Incluso en este último caso, si no ha encontrado archive.php, utilizará “index.php” como plantilla.

Este sistema de jerarquía permite una gran flexibilidad para personalizar el tema según los contenidos que esté mostrando.

5.3. Creación de temas hijos (*child themes*)

A no ser que queramos un tema con una estructura o funcionalidad muy específica, una buena opción es crear un **tema derivado de un tema ya existente (*child theme*)**. De esta manera, ahorraremos el trabajo que supone crear las plantillas y tendremos libertad para poder personalizar la presentación.

Los pasos para crear un tema derivado de otro son muy simples:

- Crear un directorio dentro de *themes* con el nombre que queramos.
- Crear un archivo “style.css” y a parte de los campos que ya conocíamos (ver apartado anterior) añadir:

Template: foo

Lectura recomendada

Para más información se puede ver el siguiente buen resumen sobre el funcionamiento de los temas en WordPress:
Joost de Valk (2011).
“Anatomy of a WordPress theme” (<http://yoast.com/wordpress-theme-anatomy/>).

Página web

Para más información, se puede consultar el esquema completo de jerarquías en el *codex* (http://codex.wordpress.org/images/1/18/Template_Hierarchy.png).

Observación

En el caso de un tema derivado, no es imprescindible crear un index.php: simplemente con el archivo style.css ya podríamos crear un tema personalizado mediante el CSS.

Esta es la línea importante que especifica que este tema está derivado de otro (el que se llama foo y por tanto se encuentra en “Themes/Foo”)

- Además en el CSS de nuestro tema debemos importar el css del tema “padre” (siempre imaginando que el tema padre se llama “foo”):

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

Ejercicio

Partir de un tema padre y modificar el css para adaptarlo a nuestro diseño, por ejemplo:

- Cambiar la tipografía.
- Cambiar algunos colores del texto.
- Cambiar algunas imágenes.
- Cambiar los bordes y los márgenes.

Si no necesitamos hacer cambios en la estructura de contenidos (campos de contenido que se presentan), tendremos suficiente personalizando el CSS, pero si queremos cambiar la estructura deberemos modificar también las plantillas php correspondientes.

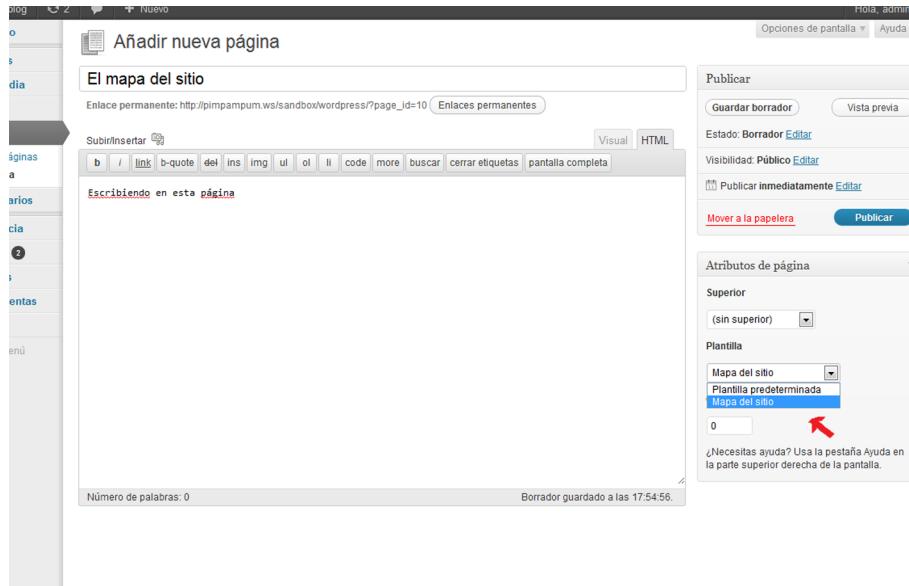
5.3.1. Plantillas de páginas

En realidad, existe en WordPress un mecanismo especial, más sencillo, para la creación de plantillas, que no utiliza el mecanismo de jerarquía. Es un método mucho más simple que se puede utilizar en casos puntuales. Es el mecanismo de los “nombres de plantilla” o “*template names*” en inglés.

Si creamos un archivo php (con el nombre que queramos) dentro de nuestro tema, simplemente escribiendo el siguiente comentario de php al principio:

```
<?php
/*
Template Name: Mapa del sitio
*/
?>
```

podremos seleccionar la plantilla en el momento de crear una nueva página de contenidos, como se ve en la captura siguiente.



Al crear una nueva página podemos seleccionar plantillas personalizadas usando el método “*template name*”.

Por supuesto, aparte del comentario es necesario que exista el resto del código php (incluir cabecera, *footer*, *sidebar* y el *loop*).

La ventaja de este sistema es que podemos personalizar la plantilla como queramos y añadir código html o php que no exista en el resto.

Por ejemplo, podríamos crear una página donde, además de los contenidos normales extraídos del *loop*, se mostrara un listado de todas las páginas existentes con la siguiente instrucción:

```
<?php wp_list_pages('title_li='); ?>
```

5.3.2. Más sobre el *loop*

Cualquier plantilla debe (o puede) utilizar el *loop*, que muestra los contenidos que corresponden a la pantalla actual.

Un *loop* muy simple podría ser el siguiente:

```
<?php get_header(); ?>
<!-- Empieza el loop. --&gt;
&lt;?php
while ( have_posts() ) : the_post(); ?&gt;
<!-- Mostrar el título vinculado con el permalink --&gt;
&lt;h2&gt;&lt;a href="&lt;?php the_permalink() ?&gt;"&gt;&lt;?php the_title(); ?&gt;&lt;/a&gt;&lt;/h2&gt;
<!-- Mostrar el contenido de la entrada dentro de un div --&gt;
&lt;div class="entry"&gt;
&lt;?php the_content(); ?&gt;
&lt;/div&gt;
&lt;?php endwhile; ?&gt;
<!-- Acaba el loop. --&gt;</pre>

```

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

Pero podemos ir añadiendo más cosas; a continuación, el código de la plantilla con un *loop* más completo:

```
<?php get_header(); ?>
<!-- Empieza el loop -->
<?php if ( have_posts() ) : while ( have_posts() ) : the_post(); ?>

<!-- el código comprueba si el contenido pertenece a la categoría con id=3 -->
<!-- si es así le asignamos la clase "post-cat-3". -->
<!-- en caso contrario la clase será "post". -->
<?php if ( in_category('3') ) { ?>
    <div class="post-cat-3">
<?php } else { ?>
    <div class="post">
<?php } ?>

<!-- Mostrar el título enlazado con el permalink -->
<h2><a href=<?php the_permalink() ?>" rel="bookmark" title="Enlace permanente a
<?php the_title_attribute(); ?>"><?php the_title(); ?></a></h2>

<!-- Mostrar la fecha en formato f/m/Y y un enlace al resto de entradas del autor-->
<small><?php the_time('d/m/Y') ?> by <?php the_author_posts_link() ?></small>
<!-- Mostrar el contenido de la entrada en un contenedor -->
<div class="entry">
    <?php the_content(); ?>
</div>

<!-- Mostrar una lista separada por comas de categorías -->
<p class="postmetadata">Publicado en <?php the_category(', ', ?></p>
</div> <!-- cerrar el primer contenedor-->

<!-- acabar el loop, y abrir una entrada else, para el caso de que no haya contenidos -->
<?php endwhile; else: ?>
<!-- En el caso de que no hay contenidos -->
<p>Lo siento, no se encontraron entradas</p>
<!-- Cerrar el loop definitivamente. -->
<?php endif; ?>
<!-- Incluir el sidebar y el footer -->
<?php get_sidebar(); ?>
<?php get_footer(); ?>
```

En algunos casos necesitaremos personalizar el *loop* para filtrar determinados tipos de contenido.

Por ejemplo, podríamos excluir algunas categorías al añadir antes del bucle:

```
<?php query_posts($query_string . '&cat=-3,-8'); ?>
```

En este ejemplo se excluyen las categorías con id 3 y 8 (notad el signo - que precede a los id).

5.4. Temas *framework*

Una de las mejores opciones para crear temas personalizados en WordPress es partir de un tema *framework*. Se trata de temas donde la estructura está muy bien definida y que son configurables, de manera que “solo” queda por definir la parte de presentación (básicamente en el CSS), es decir, tipografías, colores, imágenes, etc.

De hecho, podemos utilizar como tema base cualquier tema de WordPress, pero los temas *framework* están pensados especialmente para ello.

Existen varios temas *framework*, e incluso el tema “Twenty Eleven” integrado en WordPress se puede utilizar como tema de partida.

Algunos temas *framework*:

- **Thematic** (<http://wordpress.org/extend/themes/thematic>).
- **Toolbox** (<http://wordpress.org/extend/themes/toolbox>).
- **Foundation** (<https://github.com/drewsymo/Foundation>).
- **_s** (<http://themeshaper.com/2012/02/13/introducing-the-underscores-theme/>).
- **Skeleton** (<http://www.getskeleton.com/>).

Página web

Para más información, se puede consultar el *codex* (http://codex.wordpress.org/The_Loop).

Página web

Se puede ver más información sobre Skeleton en la página: <http://demos.simplethemes.com/skeleton/>.

Página web

Más información sobre temas *framework* en el *codex* (http://codex.wordpress.org/Theme_Frameworks#Base.2F_Starter_Theme).

5.4.1. Derivación de un tema usando Toolbox como base

Como ejemplo vamos a crear un tema utilizando Toolbox como tema base. Toolbox es un *framework* para crear temas construido en HTML5 y creado por la misma empresa desarrolladora de WordPress, Automattic.

Pasos que vamos a seguir:

- Crear un tema hijo editando el “style.css”.

- Activar el tema.
- ¿Podemos cambiar el *layout*?
- ¿Cuántas zonas tenemos para arrastrar *widgets*? ¿Dónde están definidas esas zonas?
- Crear en el CSS las tipografías más importantes.
- Imagen para la cabecera,
- Personalizar el pie de página.
- ¿Una *home* especial?

Instalación del tema:

Descargaremos los archivos y lo instalaremos de la manera habitual.

Crear un tema hijo y activarlo:

Para ello hay que crear una carpeta vacía (con el nombre que queramos) dentro de “Themes” y crear un archivo style.css con un contenido similar a este:

```
/*
Theme Name: Mi tema derivado de Toolbox
Theme URI: http://www.ejemplo.com
Description: Probando como derivar un tema de toolbox
Author: dani
Version: 1
Tags: toolbox,simple
Template: toolbox
*/
/* importamos el css del tema base */
@import url('../toolbox/style.css');
```

Observación

Aunque en CSS los comentarios se ignoran, WordPress los lee y extrae de ellos información sobre el tema. Por lo que respecta al CSS, lo único que importa es la instrucción “import”.

Explorad con Firebug, o el inspector de HTML/CSS del navegador que uséis habitualmente, los elementos “div” que tiene el tema. Observar los id y clases que nos permitirán adaptar nuestro diseño a la plantilla.

Añadir la posibilidad de tener un *background* editable:

Para ello tenemos que crear nuestro archivo “functions.php” y escribir dentro:

```
<?php  
add_custom_background(); ?>
```

Observación

En varios temas de WordPress esta opción ya está activa, pero no en el caso de Toolbox.

Abrid el administrador de WordPress y observad cómo ha aparecido una nueva opción dentro de “Apariencia” que permite cambiar el fondo. Cambiadlo y observad el resultado en el blog.

Crear un color de fondo para los “div” principales (contenido, *sidebar*), ajustar márgenes y *paddings*. Para ello debemos modificar nuestro “style.css”.

También podemos cambiar color tipografía, fuente, tamaño, etc.

```
#primary{  
background-color:#ffffff;  
}
```

Cambiar el estilo del título del blog (h1) y del subtítulo (h2):

```
h1{  
..  
}
```

Cambiar el estilo de la opción de menú seleccionada:

Descubrid con Firebug qué clase tiene la opción de menú seleccionada y editad el CSS para remarcarlo de alguna manera.

Crear una plantilla especial para las páginas (page.php) y para los artículos (single.php).

Crear un *page template* (plantillas especiales para páginas).

Ideas:

```
/**  
Page Templatge: Mi nombre  
*/
```

Una plantilla especial para la página de contacto, añadiendo un formulario de docs.google.com o similar, una plantilla para el *sitemap* del sitio, añadiendo funciones del *codex* de WordPress para listar páginas y entradas, etc.

Crear un *footer* y hacerlo dinámico :

Para ello necesitamos, por una parte, añadir el siguiente código en “functions.php”:

```
register_sidebar( array(
    'name' => __( 'Footer', 'footer' ),
    'id' => 'myfooter',
    'description' => __( 'Footer', 'footer' ),
    'before_widget' => '<li id="%1$s" class="widget-container %2$s">',
    'after_widget' => '</li>',
    'before_title' => '<h3 class="widget-title">',
    'after_title' => '</h3>',
) );
```

Y luego en nuestra plantilla del *footer*:

```
<?php dynamic_sidebar( 'footer' ) ?>
```

Visualizar las imágenes destacadas de los artículos (si las tienen; si no, mostrar una por defecto).

En “functions.php”:

```
add_theme_support( 'post-thumbnails' );
```

Una vez activado en nuestras plantillas podemos usar:

```
has_post_thumbnail()
the_post_thumbnail();
```

etc.

5.5. Ejercicio práctico. Creación de plantilla para portafolios

En este apartado nos plantearemos un ejercicio que tiene como objetivo la personalización de un tema para crear un portafolio que muestre solo los contenidos pertenecientes a la categoría “Portfolio” y las imágenes destacadas asociadas.

Instrucciones paso a paso:

- Partid de un tema derivado de Toolbox (“Child” u otro tema base diferente si se desea). Para ello, consultad las instrucciones del apartado anterior.
- Activad la opción de imágenes destacadas (o *Post thumbnails*) en “functions.php”:

```
add_theme_support( 'post-thumbnails' );
```

Esta instrucción añade la opción de vincular una imagen destacada para cada entrada.

- Cread varias entradas de prueba (como mínimo cinco), todas ellas con la categoría “Portfolio” y con “Imagen destacada”.
- Cread una plantilla dentro de nuestro tema derivado llamada “category-portfolio.php”. De esta manera WordPress utilizará esta plantilla cuando entremos en “/category/portfolio” y nos mostrará un listado de las entradas correspondientes.
- El contenido inicial de esta plantilla, muy simplificado, puede ser el siguiente:

```
<?php get_header(); ?>

<?php if ( have_posts() ) : ?>
<?php while ( have_posts() ) : the_post(); ?>

<!-- contenido de una entrada del portfolio -->
<a href="php the_permalink() ?&gt;"&gt;
    &lt;?php the_post_thumbnail('thumbnail');?&gt;&lt;/a&gt;
&lt;!-- fin del contenido de una entrada &amp;#8594;

&lt;?php endwhile; ?&gt;
&lt;?php endif; ?&gt;</pre
```

Lectura recomendada

Encontraréis más información sobre imágenes destacadas en el siguiente artículo:

Mark Jaquith (2009). “New in WordPress 2.9: Post Thumbnail Images” (<http://markjaquith.wordpress.com/2009/12/23/new-in-wordpress-2-9-post-thumbnail-images/>).

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

“the_post_thumbnail(‘thumbnail’)” escribe en la plantilla la imagen destacada en formato pequeño.

- También queremos cambiar el aspecto de la **página única** de las entradas del “Portfolio”, que corresponde a la plantilla “single.php”. Podríamos partir del “single.php” del tema padre, y añadir al menos esta línea que es la que muestra la imagen en formato grande:

```
<?php the_post_thumbnail('large'); ?>
```

Sin embargo esta solución **no nos permite tener una plantilla independiente** para las páginas únicas del “Portfolio”. Si queremos tener una plantilla independiente, como esto no está previsto en el sistema de plantillas de WordPress, deberemos *hackear* un poco. Para ello una solución sería editar la plantilla “single” y añadir al principio:

```
if ( in_category( 'portfolio' ) ) { //si el post es de esta categoría  
    get_template_part('single-portfolio');  
    //utiliza otra plantilla  
  
} else{  
  
    .. resto de la plantilla single.php
```

y al final:

```
<?php  
} //cerrar el if  
?>
```

Esta plantilla “single” decide utilizar otra plantilla diferente llamada “single-portfolio.php” en función de si el contenido tiene la categoría “Portfolio”.

Observación

También podemos definir “Post Types”, es decir, nuevos tipos de contenido. Una posible solución en nuestro caso sería crear un tipo de contenido “Portfolio” en lugar de utilizar una categoría en el tipo de contenido “Entrada” preexistente.

Siguiendo con el ejercicio, en nuestra plantilla especial para el “Portfolio”, “single-portfolio.php”, tendríamos lo mismo que en “single.php” pero sin la condición inicial y, además, podemos añadir:

```
<?php the_post_thumbnail('large'); ?>
```

que mostrará la imagen destacada en tamaño grande. Además, en esta plantilla podemos añadir cualquier otro cambio que queramos en la estructura. Todo lo que pongamos en esta plantilla solo afectará al listado de entradas de la categoría “Portfolio” (en la ruta “categories/portfolio”).

En este punto puede ser que nos preguntemos si es posible tener varias imágenes en la misma entrada. Las galerías de WordPress sí permiten asociar varias imágenes a una entrada, pero esto no tiene nada que ver con las imágenes destacadas (*thumbnails*). Solo es posible asociar un *thumbnail* para cada una de las entradas.

A continuación vamos a ver un ejemplo de personalización de la plantilla “category-portfolio.php”:

- Para empezar y poder hacer pruebas necesitaremos añadir contenidos (al menos diez entradas de la categoría).
- Si queremos añadir la opción de paginación en “category-portfolio.php”, tenemos que insertar al final o donde queremos que aparezca:

```
<?php  
next_posts_link();  
previous_posts_link();  
?>
```

- Añadir un contenedor exterior a las fotos (*thumbnails* cuadrados) escribiendo los “div” en la plantilla (quizás un “div” para la imagen y otro para el título) y modificar el CSS para que se muestre como una rejilla de 3x3.
- Modificar el número de posts a nueve en lugar de diez (en “Ajustes/Lectura”).
- Si queremos añadir un “div” con el título sobre las imágenes:

```
the_title()
```

- Crear un contenedor para los *thumbnails* y otro para los títulos, modificar el css para modificar el aspecto.
- Incluso podríamos añadir un *script* en jQuery para mostrar los títulos al hacer *rollover*:

```
<script>
var $j = jQuery.noConflict();
$j(document).ready(function(){
    $('.thumb').mouseover(function(){
        $j(this).find('.thumb-title').show("fast");
    });
    $('.thumb').mouseout(function(){
        $j(this).find('.thumb-title').hide();
    });
});

</script>
```

Material

Encontraréis los archivos del tema una vez acabados en:
http://www.kiwoo.org/recursos/ejemplos/drupal/tema_portfolio.zip

5.6. Checklist para crear un tema de WordPress

La creación de un tema en WordPress (o en otros CMS) requiere tomar decisiones para optimizar el desarrollo. La solución óptima puede variar según las especificaciones del proyecto. En este apartado veremos un *checklist* que permite escoger la mejor opción según estas especificaciones.

Como hemos visto, básicamente tenemos tres maneras de crear un tema:

- Crear un tema *child* de cualquier otro ya existente que se adapte a nuestro diseño y modificar (solo) el CSS.
- Crear un tema *child* de un tema *framework*, como puede ser Toolbox, y modificar el CSS y las plantillas que necesitemos redefinir.
- Crear un tema desde cero. No derivar el tema de ningún otro y crear todas las plantillas desde cero (para expertos).

Nuestro diseño define una serie de pantallas con un *layout* y contenidos determinados. Primero es necesario tener clara la estructura de contenidos (tipos de contenido, etiquetas, categorías).

Las preguntas que nos debemos hacer a partir de este momento son:

¿Existe algún tema disponible con el mismo *layout* que mi diseño?

- Sí: en este caso, crear un tema *child* y personalizar el CSS como queramos. **Fin.**
- No: en caso contrario, pasar al siguiente punto.

¿Existe algún *framework* que se pueda adaptar a la estructura de mi diseño? Un *framework* es un tema que solo tiene *layout* pero no tiene presentación: está en blanco. Cada uno define regiones dinámicas diferentes (para arrastrar *widgets*) y además algunos permiten configurar opciones de la presentación directamente desde el panel de WP.

- Sí: crear un tema *child* y personalizar el CSS. **Fin.**
- No: pasar al siguiente punto.

¿Es posible conseguir el *layout* creando algunas plantillas especiales? Una plantilla especial nos permite añadir contenidos que no teníamos y necesitamos, y añadir “div” o elementos HTML que nos permitan modificar el aspecto mediante CSS. Para tener una plantilla especial creamos un archivo con cualquier nombre y añadimos un comentario dentro del PHP:

```
<?php
/*
Template Name: Mi plantilla
*/
?>
.. resto de código
```

Dentro de la plantilla podemos poner lo que queramos aparte del *loop*.

¿Tenemos suficiente con la utilización de plantillas personalizadas?

- Sí: crear las plantillas especiales que queramos. **Fin.**
- No: pasar al siguiente punto.

¿Es posible conseguir el *layout* que queremos modificando algunas plantillas de la jerarquía? El mecanismo de plantillas de WordPress define una cadena de decisiones a partir de la cual se escoge la plantilla necesaria en cada momento. De esta manera podemos definir una plantilla especial para una página en concreto “page-\$id.php”, para el listado de una etiqueta en concreto “tag-\$slug.php”, y para muchos otros casos. Estas plantillas especiales están documentadas en el *codex*, por ejemplo, las plantillas de categoría *Template*. Las plantillas de la jerarquía se utilizan automáticamente si existen.

- Sí: crear las plantillas necesarias. **Fin.**
- No: pasar al siguiente punto.

Atención: si existen en la plantilla madre, deberemos copiar primero esa plantilla a nuestro “subtema”.

¿Podemos añadir funciones especiales en “functions.php”, o código especial dentro de alguna plantilla para solucionarlo? ¿O incluso algún plugin? Algunas llamadas a la API de WordPress nos permiten añadir funcionalidades como zonas activas, activación de funcionalidades de WP o funciones propias para lo que queramos (se trata de desarrollo más avanzado: aquí sí necesitaremos PHP para ello).

- Sí: personalizar el código de “functions.php” o de alguna de las plantillas.
Fin.
- No: pasar al siguiente punto.

Como opción final, la que nos da más flexibilidad, podemos crear un tema desde cero.

Requisitos para ser un buen themer

- Conocimientos de HTML. En las plantillas propias deberemos añadir el HTML que creamos conveniente.
- Conocimientos de CSS. Es la herramienta principal que usaremos para modificar el aspecto de nuestra plantilla.
- Es necesario conocer suficientemente PHP, lo necesario al menos para poder utilizar las variables y funciones de WP.

5.6.1. Algunos problemas típicos y su solución

La plantilla personalizada no funciona. Verificar que el contenido realmente está usando esa plantilla, por ejemplo añadiendo un mensaje de texto dentro como:

```
<strong>plantilla blah</strong>
```

Si no aparece este mensaje en el navegador es porque WP no está usando la plantilla que suponemos. Siempre hay que tener en cuenta la jerarquía de plantillas. Al ser un mecanismo automático, puede crear confusiones

Error de PHP. Observar en qué archivo y en qué línea especifica está el mensaje de error. Puede ser debido a que los delimitadores de PHP no están bien cerrados (por ejemplo, abrir “<?php” sin cerrarlo con “?>” dará un error). O puede que exista una condición “if “sin cerrar, o que haya una función no definida, por citar solo algunos ejemplos.

Página web

Más información sobre la jerarquía de plantillas de WordPress en el *codex* (http://codex.wordpress.org/Template_Hierarchy).

Error 500. Muy a menudo se debe a errores en el servidor causados por el “.htaccess” (archivo que se encarga de reescribir las URL “limpias”), entre otras cosas.

5.7. Adaptar un tema de WordPress para diferentes dispositivos

En el panorama actual se puede decir que se usan básicamente **tres tipos de dispositivos**:

- **Navegador web** de escritorio. La resolución suele ser alta o muy alta y se interactúa a través de un ratón o un *touchpad*.
- **Tablet.** Aquí el condicionante, más que la resolución, es el hecho de que las pantallas son táctiles y, por tanto, las zonas sensibles deben estar pensadas para ello.
- **Móviles** (*smartphones*). Con pantallas mucho más pequeñas.

Las posibles soluciones para afrontar un diseño multidispositivo son:

- **Crear un tema especial para móviles y otro para escritorio y/o tablet.** Para ello debemos activar un *plugin* que detecte el dispositivo y nos permita seleccionar un tema para nuestra web en función de ello.
- Utilizar un tema ***responsive*** que se adapte a la medida de la pantalla

Ejemplo: <http://demos.simplethemes.com/skeleton/>.

Mobble (<https://wordpress.org/extend/plugins/mobble/faq/>) es un interesante *plugin* que nos puede ayudar en la tarea de adaptar el diseño a diferentes dispositivos. Una vez activado nos permitirá utilizar instrucciones especiales en nuestras plantillas:

```
<?php  
is_handheld(); // any handheld device (phone, tablet, Nintendo)  
is_mobile(); // any type of mobile phone (iPhone, Android, Nokia etc)  
is_tablet(); // any tablet device (currently iPad, Galaxy Tab)  
is_ios(); // any Apple device (iPhone, iPad, iPod)  
?>
```

Ejemplo:

```
<?php
if (is_mobile()) {
    //no incluimos el sidebar
} else{
    if(is_tablet()){
        get_sidebar("tablet");

        //sidebar especial para tablet
    } else{
        get_sidebar();
        //sidebar para navegador de escritorio
    }
}
?>
```

Cómo testear nuestro diseño en diferentes dispositivos:

Idealmente deberíamos probar en todos los dispositivos reales, pero dado que esto no es posible tenemos los siguientes recursos:

- Utilizar un navegador normal y variar el tamaño de la ventana (mejor si utilizamos Chrome o Safari, que utilizan Webkit igual que los navegadores por defecto de iPhone y Android).
- Utilizar un simulador de navegador móvil, como el de Opera, Opera Mobile Emulator (<http://www.opera.com/developer/tools/mobile/>).
- Utilizar un emulador del kit de desarrollo de iPhone, Android, etc.
- Utilizar un emulador *on-line*.

Consejo

Si estamos desarrollando en local, podemos usar nuestro móvil o *tablet* conectándonos por la red WiFi y testear utilizando la IP del servidor.

5.8. **Shortcodes**

A partir de la versión 2.5 WordPress introduce una API para *shortcodes*.

Un **shortcode** es un código entre corchetes escrito directamente en el texto de una entrada. Es un mecanismo sencillo que permite que el código de un *plugin*, o directamente “functions.php”, traduzca el *shortcode* por otro contenido determinado por el contenido de la entrada o por los parámetros.

Por ejemplo, al introducir una galería de fotos dentro de una entrada, en realidad WP escribe un *shortcode* “[gallery]” que posteriormente es traducido al HTML necesario para mostrar la galería.

La API de *shortcodes* permite a los desarrolladores de *plugins* y temas crear código para que los usuarios puedan utilizar determinados *shortcodes* en sus entradas. Los *shortcodes* pueden tener múltiples atributos, por ejemplo:

```
[gallery id="123" size="medium"]
```

El desarrollo del código necesario para implementar un *shortcode* no es complicado. Veamos un ejemplo sencillo.

En el archivo “functions.php” de nuestro tema introducimos el siguiente código para definir y registrar una entrada de *shortcode*:

```
// [ejemplo]
function ejemplo_func( $atts ){
    return "probando shortcodes";
}
add_shortcode( 'test', 'ejemplo_func' );
```

En este caso, el código “[test]” se convertiría en el texto “probando *shortcodes*” cuando vemos la entrada o página resultante.

El desarrollo de un *shortcode* puede ser útil en casos en los que no necesitemos la complejidad que implica el desarrollo completo de un *plugin*.

5.8.1. Funcionamiento del *shortcode* “Caption”

Uno de los *shortcodes* integrados en WP es el “Caption” de las imágenes.

El campo “Leyenda” de la imagen se traduce como *shortcode* en WP.

Al insertar una imagen, aparte de los campos “Título” (“Title”) y “Texto alternativo” (“Alt”), disponemos del campo “Leyenda”. Si utilizamos este campo, la leyenda aparecerá en un marco envolviendo la foto.

WP implementa esta funcionalidad utilizando un *shortcode* especial llamado “Caption”. De hecho, si analizamos el código HTML generado por WP nos encontraremos con esto:

```
[caption id="attachment_60" align="alignnone" width="300" caption="Nenúfares"]

!\[\]\(5f5af4a3e3206e8486b585c4d9a19527\_img.jpg\)

```

Se trata de un *shortcode* con diversos parámetros. WP traduce este *shortcode* y sus parámetros en el HTML necesario para mostrar un marco gris alrededor de la foto.

En el caso anterior el resultado en la entrada publicada es:

```
<div id="attachment_60" class="wp-caption alignnone" style="width: 310px">
<a href="http://ejemplo.com//wordpress/wp-content/uploads/Nenúfares1.jpg"></a><p class="wp-caption-text">Nenúfares</p></div>
```

5.9. Desarrollo de un *plugin* en WordPress

Como ya hemos explicado, un *plugin* es un programa que permite añadir funcionalidades a WordPress. Estos programas se desarrollan utilizando la API de desarrollo de *plugins*.

El desarrollo de un *plugin* puede ser tan complejo como queramos. Obviamente, antes de desarrollar un *plugin* nos debemos preguntar si ya existe alguno similar entre los miles que ya están disponibles, y en ese caso utilizarlo o incluso, si la licencia lo permite, modificarlo o crear una nueva versión.

La API de plugins de WP está basado en 2 tipos de *hooks*:

- **Acciones.** Nuestro *plugin* puede activarse en determinados “eventos” de WordPress. Por ejemplo, en el momento de publicarse una nueva entrada, al cambiar de tema, al añadir un nuevo comentario, etc. En este momento

Página web

Más información sobre *shortcodes* en el *codex* en: http://codex.wordpress.org/Shortcode_API.

Página web

Más información sobre la API de desarrollo de *plugins* en el *codex* en: http://codex.wordpress.org/Plugin_API.

podemos capturar el “evento” y modificar el contenido, enviar un email o cualquier otra acción que queramos hacer.

- **Filtros.** Se llaman antes de guardar información en la base de datos o al mostrarse en el navegador.

A nivel básico podemos desarrollar un *plugin* a base de acciones y filtros, aunque probablemente necesitamos otras cosas, como acceso a la base de datos, modificar parámetros a través del escritorio de WP, etc. Sin embargo, esto queda fuera del dominio de estos apuntes y vamos a ver solo las bases.

5.9.1. Acciones

Existen cientos de acciones, todas ellas perfectamente documentadas en el *codex* (http://codex.wordpress.org/Plugin_API/Action_Reference). Vamos a ilustrarlo con un pequeño ejemplo consistente en un *plugin* que envía un correo electrónico cada vez que se publica una nueva entrada.

Para ello debemos utilizar la acción “publish_post”, que se llama cada vez que se dispara este “evento”. Utilizando “add_action” podemos vincular esta acción con una función definida por nuestro *plugin*.

Por ejemplo:

```
add_action('publish_post', enviar_email);

Por supuesto faltaría la definición de nuestra función enviar_email
function enviar_email($post_id) {
    mail("info@ejemplo.com", "blog actualizado", "Existe una nueva entrada
    en el blog");
}
```

Notad que la acción envía como parámetro el id del *post*. Esto puede ser útil si queremos recuperar el contenido para cualquier propósito.

Utilizar directamente una función no es una buena idea porque es fácil que existan conflictos con otras funciones con el mismo nombre. Para solucionar esto podemos encapsular la función como método de una clase y llamarlo de la siguiente manera:

```
class emailer {
    function enviar_email($post_ID)  {
        mail("info@ejemplo.com", "blog actualizado", "Existe una nueva entrada
        en el blog");
        return $post_ID;
    }
}
```

```
}

add_action('publish_post', array('emailer', 'enviar_email'));
```

5.9.2. Filtros

Los filtros son funciones a las que WordPress pasa datos justo antes de hacer algo con estos datos, como por ejemplo guardarlos en la base de datos o mostrarlos en el navegador. También existen cientos de filtros y están documentados en el *codex* (http://codex.wordpress.org/Plugin_API/Filter_Reference).

Un filtro recibe como entrada datos sin modificar y devuelve los datos modificados. En general la llamada es:

```
add_filter( 'nombre_del_hook', 'nombre_del_filtro', [prioridad], [otros_parámetros] );
```

Así, por ejemplo:

```
<?php

add_filter( 'the_content', 'modificar_contenido' );

function modificar_contenido($contenido) {
    return '*' . $contenido . '*';
}

?>
```

añadirá un asterisco antes y después de todos los contenidos.

El código de nuestro *plugin* se podría añadir directamente al “functions.php” de nuestro tema, pero normalmente lo encapsulamos en un directorio aparte. Hacer esto es relativamente sencillo. En primer lugar debemos elegir un nombre que no se preste a confusión con otros *plugins* y usarlo como nombre del directorio, donde por lo menos deberemos tener dos archivos.

El archivo principal del *plugin* (con el nombre que queramos, pero procurando que sea lo más descriptivo posible) contiene el código y un comentario al principio que proporciona información a WP:

```
/*
Plugin Name: Test para los apuntes
Plugin URI: http://la_url_del_plugin_con_actualizaciones
Description: Descripción del plugin
Version: La versión del plugin
Author: El autor
Author URI: http://URI_Of_The_Plugin_Author
License: GPL2
```

*/

Parte de esta información es la que el usuario ve en el panel de WordPress. Por ejemplo, en el caso del *plugin* Akismet, la información del *plugin* es la siguiente:

```
/*
Plugin Name: Akismet
Plugin URI: http://akismet.com/
Description: Used by millions, Akismet is quite possibly the best way in the world to
<strong>protect your blog from comment and trackback spam</strong>. It
keeps your site protected from spam even while you sleep. To get started:
1) Click the "Activate" link to the left of this description,
2) <a href="http://akismet.com/get/?return=true">Sign up for an Akismet
API key</a>, and 3) Go to your <a href="plugins.php?page=akismet-key-config">Akismet
configuration</a> page, and save your API key.
Version: 2.5.3
Author: Automattic
Author URI: http://automattic.com/wordpress-plugins/
License: GPLv2 or later
*/
```

Si queremos subir el *plugin* a la comunidad WordPress es necesario añadir un archivo “readme.txt” siguiendo las indicaciones del *codex* (http://codex.wordpress.org/Writing_a_Plugin#Readme_File).

Por supuesto, cuanto más complejo sea el *plugin*, más archivos podemos necesitar. Un *plugin* puede utilizar, además de filtros y acciones, cualquier llamada a la API de WordPress o utilizar código PHP de otras librerías, por ejemplo. Las posibilidades son ilimitadas.

Recomendación

Una buena fuente de inspiración para la creación de un *plugin* particular puede ser el análisis de otros *plugins* ya existentes.

Una de las muchas opciones posibles es desarrollar un *plugin* que sirva para activar un nuevo *widget* personalizado.

Página web

Más información sobre desarrollo de plugins en el *codex* en: http://codex.wordpress.org/Writing_a_Plugin.

Lectura recomendada

En el siguiente artículo podemos ver un ejemplo realizado paso por paso:

Dave Clements (2011).
“How to create a WordPress Widget from Scratch”: <http://www.doitwithwp.com/how-to-create-wordpress-widget-step-by-step/>.