

El WordPress

Daniel Julià Lundgren

PID_00168278



Els textos i imatges publicats en aquesta obra estan subjectes –llevat que s'indiqui el contrari– a una llicència de Reconeixement-Compartir igual (BY-SA) v.3.0 Espanya de Creative Commons. Podeu modificar l'obra, reproduir-la, distribuir-la o comunicar-la públicament sempre que en citeu l'autor i la font (FUOC. Fundació per a la Universitat Oberta de Catalunya), i sempre que l'obra derivada quedi subjecta a la mateixa llicència que el material original. La llicència completa es pot consultar a <http://creativecommons.org/licenses/by-sa/3.0/es/legalcode.ca>

Índex

1. Introducció	5
2. Instal·lació de WordPress	7
3. Introducció a l'escriptori (<i>dashboard</i> o panell) de WordPress	12
3.1. Sobre els <i>feeds</i> automàtics	12
3.2. Apartats principals de l'escriptori	13
3.2.1. Escriptori	13
3.2.2. Entrades	14
3.2.3. Categories i etiquetes	15
3.2.4. Multimèdia	16
3.2.5. Enllaços (<i>links</i>)	17
3.2.6. Pàgines (<i>pages</i>)	17
3.2.7. Comentaris	19
3.2.8. Aparença (<i>appearance</i>)	20
3.2.9. <i>Plug-ins</i>	25
3.2.10. Usuaris	26
3.2.11. Eines (<i>tools</i>)	27
3.2.12. Ajustaments (<i>settings</i>)	27
4. Configuració i seguretat	29
5. Introducció als temes en WordPress	30
5.1. Alguns editors de codi	30
5.2. Creació d'un tema de WordPress des de zero	30
5.2.1. Anatomia d'un tema	33
5.3. Creació de temes fills (<i>child themes</i>)	35
5.3.1. Plantilles de pàgines	36
5.3.2. Més sobre el <i>loop</i>	37
5.4. Temes <i>framework</i>	39
5.4.1. Derivació d'un tema usant Toolbox com a base	39
5.5. Exercici pràctic. Creació d'una plantilla per a <i>portfolios</i>	43
5.6. Llista de control per a crear un tema de WordPress	46
5.6.1. Alguns problemes típics i la seva solució	48
5.7. Adaptar un tema de WordPress per a diferents dispositius	49
5.8. <i>Shortcodes</i>	50
5.8.1. Funcionament del <i>shortcode</i> "Caption"	51
5.9. Desenvolupament d'un <i>plug-in</i> en WordPress	52
5.9.1. Accions	53
5.9.2. Filtres	54

1. Introducció

El **WordPress** és probablement el **gestor de blogs** més utilitzat en aquests moments, i no solament per a la creació de blogs, sinó també per a llocs web més genèrics (encara que possiblement sense arribar a la flexibilitat d'altres CMS més complexos com Drupal).

El WordPress (<http://www.wordpress.com/>) disposa d'un servei en línia en què qualsevol usuari es pot donar d'alta i crear un blog en un moment. Al mateix temps, el WordPress és un **projecte de codi obert** amb llicència GPL, de manera que qualsevol persona es pot descarregar el codi de l'aplicació –que funciona sobre la plataforma Apache+PHP+MySQL– i instal·lar-lo en el propi servidor, amb la flexibilitat que això proporciona, per exemple, per a permetre'ns activar qualsevol tema o *plug-in* creat per terceres persones, i també i sobretot, per a personalitzar el codi i crear temes i *plug-ins* nosaltres mateixos.

La base d'aquesta plataforma de codi obert és <http://www.wordpress.org>. El projecte està extensament documentat en l'anomenat **Codex** (http://codex.wordpress.org/es:Main_Page), el repositori principal de documentació sobre WordPress.

Hi ha una gran comunitat de desenvolupadors que s'encarreguen de mantenir el projecte i de proporcionar infinitat de *plug-ins*, extensions que permeten augmentar les prestacions que el WordPress té per defecte, a més de temes (*themes*), que permeten adaptar la presentació i el *layout* per a cada necessitat.

El WordPress és una eina pensada des del primer moment per a la creació i administració de blogs. Per tant, són les característiques associades a un blog les que s'han tingut en compte en primer lloc.

Les **funcionalitats bàsiques** de WordPress són:

- Publicació d'articles (entrades) que es mostren des del més recent al més antic en la portada.
- Publicació de pàgines “estàtiques” que apareixen per defecte en el menú principal.
- Possibilitat que els visitants afegeixin comentaris, amb eines de moderació en el panell d'administració.

Vegeu també

Veurem aquest CMS en el mòdul “El Drupal”.

Pàgina web

La versió anglesa del Codex acostuma a estar més actualitzada i a contenir més informació.

- Publicació automàtica d’RSS en formats RSS 1.0, RSS 2.0 i Atom.
- Ping automàtic a servidors de notificació de canvis.
- Categories i etiquetes (*tags*) per a etiquetar els continguts.
- Possibilitat de canviar fàcilment la presentació mitjançant temes (*themes*).
- Possibilitat d'augmentar les funcionalitats instal·lant *plug-ins*.
- Gestió de ginys (*widgets*).

Des de la versió 3, s'han unificat les versions de WordPress i El WordPress MU (MultiUser), oferint noves funcionalitats com la possibilitat de crear tipus de continguts o d'actualitzar diversos *plug-ins* i temes simultàniament, entre altres.

A continuació veurem el procés d'instal·lació, les característiques principals del panell d'administració i, finalment, les possibilitats d'ampliació mitjançant la creació de *plug-ins* i temes.

2. Instal·lació de WordPress

La instal·lació de WordPress és bastant ràpida i fàcil, i de vegades es presenta com la “famosa instal·lació de cinc minuts”.

Per començar, hem de disposar d'un servidor web capaç d'executar PHP i d'un servidor de bases de dades amb MySQL (i una base de dades o la possibilitat de crear-ne una).

Abans d'instal·lar el WordPress en el nostre servidor local o remot, ens hem d'assegurar que el servidor compleix els requisits mínims (versions de PHP i MySQL).

Requisits (per a la versió 3.3.2):

- PHP 5.2.4 o superior
- MySQL 5.0 o superior
- El mòdul “mod_rewrite” d'Apache (molt recomanable)

Una vegada instal·lada una versió de WordPress, haurem de revisar contínuament si hi ha versions noves. De fet, el mateix panell de WordPress ens avisarà si han aparegut. En aquest cas, sempre és recomanable actualitzar-la, per temes de seguretat però també per a poder gaudir de noves funcionalitats.

Observació

Podem instal·lar el WordPress en l'idioma que vulguem (sempre que hi hagi una traducció). Aquí ho farem en castellà. D'altra banda, l'idioma del nucli de WordPress és independent del text que hi pugui haver en els temes o *plug-ins*: encara que instal·lem el WordPress en castellà, després haurem d'aconseguir les traduccions de temes i *plug-ins* a part.

Exercici

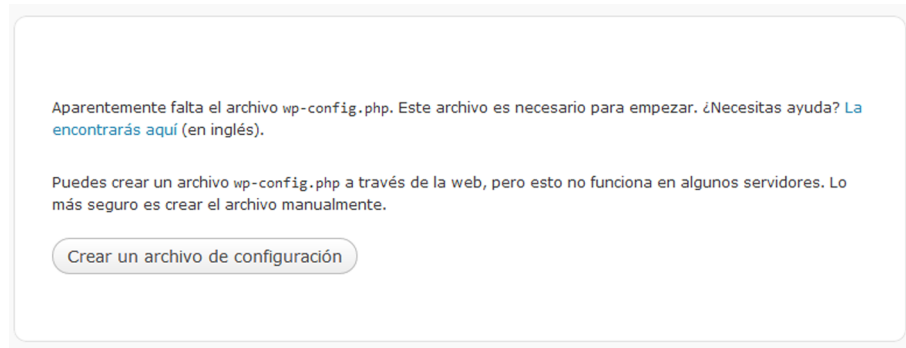
Consulteu com és la versió actual de WordPress i si els seus requisits mínims han canviat des de la versió comentada en aquestes anotacions.

Una vegada hem comprovat els requisits mínims, descarreguem els arxius de la pàgina <http://wordpress.org/download/> o <http://es.wordpress.org/> i seguim les instruccions següents:

Passos de la instal·lació:

- Descarregar i descomprimir el zip de WordPress en l'idioma en què el vulguem instal·lar (encara que posteriorment sempre podem canviar l'idioma).

- Crear una base de dades buida per a WordPress (si encara no en tenim una) en el servidor web, assegurar-se de tenir les dades (usuari i contrasenya o *password*) de la base de dades (diferent que el compte de l'FTP), recordar el nom que hem donat a aquesta base de dades.
- Copiar els arxius en el servidor (per exemple <http://www.exemple.com>). És recomanable fer-ho dins d'un directori anomenat, per exemple, *blog*. Després, si volem, podem fer que aparegui en l'arrel del nostre domini, tal com veurem més endavant.
- Entrar utilitzant el navegador a <http://www.exemple.com/blog> i seguir les instruccions.
- En el primer pas ens demanarà permís per a crear l'arxiu `config.php` automàticament i en el pas següent, les dades de connexió amb la base de dades.
- Si la instal·lació s'ha fet correctament, ens portarà a una pantalla en què se'ns demana el títol del blog, l'adreça de correu electrònic, el nom d'usuari i la contrasenya de l'administrador. És molt important apuntar i conservar aquestes dades, ja que les necessitarem cada vegada que vulguem accedir al panell d'administració del nostre blog.



Missatge que ens trobem en entrar per primera vegada en l'adreça del nostre 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 quieres 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.

Detalls de connexió amb la base de dades lloc



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="password" value="••••"/>
Se generará un password automático si lo dejas en blanco.	<input type="text"/>
	<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.

Informació general de nostre lloc web i creació del compte d'administració



Entrada per primera vegada com a administrador de la nostra plataforma WordPress

És possible que trobem problemes en la instal·lació: per exemple, que el WordPress no pugui crear l'arxiu `config.php`. En aquest cas, l'haurem d'editar a mà.

Observació

Si necessitem editar l'arxiu `config.php`, mai no hem d'usar com a editor de textos aplicacions com Microsoft Word, ja que introdueixen caràcters indesitjats. Per a editar el text d'arxius de configuració és molt millor usar un bon editor de codi com els que es comentaran més endavant, o, en defecte d'això, un editor de textos simple com Notepad, en Windows, o SimpleText, en Mac.

Exercici

Descarregueu els arxius de WordPress, creeu una base de dades en un servidor local i instal·leu el WordPress seguint els passos anteriors. Com a servidor local podeu utilitzar XAMPP, una distribució que conté Apache, MySQL, PHP i Perl, pensada perquè sigui fàcil d'instal·lar i d'usar en qualsevol plataforma. També hi ha altres distribucions com Wamp, per a Windows, i Mamp, per a Mac.

Fent clic a *login* accedirem a la pantalla de *login* en què introduïrem les dades del compte d'usuari i anirem directament al panell d'administració. En aquesta pantalla apareix el nostre **escriptori** (*dashboard*, en anglès) o **tauler de control**, on tenim accés directe a les funcionalitats principals del CMS.

En WordPress la diferenciació entre el tauler (*dashboard*) i el lloc web és molt evident. Normalment el *dashboard* es trobarà en l'adreça `wp-admin`, i el lloc web en l'arrel, en el nostre cas a `http://www.exemple.com/blog/wp-admin` i `http://www.exemple.com/blog/`.

A l'esquerra de la imatge següent veiem totes les opcions del panell d'administració.

Escritorio

¡Bienvenido a tu nuevo sitio de WordPress!

Si necesitas ayuda para empezar, mira nuestra documentación sobre [Primeros Pasos con WordPress](#). Si prefieres entrar en materia, aquí tienes algunas de las cosas que la mayoría de la gente hace al configurar su nuevo sitio WordPress. Si necesitas ayuda, por favor utiliza las pestañas de Ayuda en la parte superior derecha para obtener información acerca de cómo utilizar tu pantalla actual y dónde ir para conseguir más ayuda.

Ajustes básicos

Aquí hay algunas cosas fáciles que puedes hacer para modificar cosas. Asegúrate de hacer clic en Guardar en cada pantalla de Configuración.

- [Selecciona la configuración de privacidad](#)
- [Selecciona tu zona horaria](#)
- [Activa o desactiva los comentarios](#)
- [Rellena tu perfil](#)

Añade contenido real

Mira los editores con la página y el post de ejemplo para ver cómo funcionan. Después, borra el contenido por defecto y jémpieza a escribir!

- Ver la [página](#) y la [entrada](#) de ejemplo
- Borrar la [página](#) y la [entrada](#) de ejemplo
- [Crea una página sobre ti](#)
- [Escribe la primera entrada](#)

Personaliza tu sitio

Usa el tema actual — Twenty Eleven — o [selecciona uno nuevo](#). Si nos atenemos a Twenty Eleven, aquí tienes algunas maneras de hacer que tu sitio sea único.

- [Selecciona claro u oscuro](#)
- [Establecer un color de fondo](#)
- [Selecciona una nueva imagen para la cabecera](#)
- [Añade algunos widgets](#)

¿Sabes lo que estás haciendo? [Descarta este mensaje](#)

Ahora mismo

Contenido	Comentarios
1 Entrada	1 Comentario
1 Página	1 Aprobado
1 Categoría	0 Pendientes
0 Etiquetas	0 Spam

Tema **Twenty Eleven** con 6 widgets

Buscadores bloqueados

Publicación rápida

Título

Subir/Insertar

Contenido

Etiquetas

L'escritori (*dashboard*) de WordPress

Mi blog

Otro sitio realizado con WordPress

Inicio [Página de ejemplo](#)

¡Hola mundo!

Publicado el **11 abril, 2012**

Bienvenido a WordPress. Esta es tu primera entrada. Edítala o bórrala, ¡y comienza a publicar!

Publicado en [Sin categoría](#) | [1 Respuesta](#)

Entradas recientes:

- [¡Hola mundo!](#)

Comentarios recientes:

- [Sr WordPress](#) en [¡Hola mundo!](#)

Archivos:

- [abril 2012](#)

Categorías:

- [Sin categoría](#)

Aspecte del lloc web de WordPress per defecte (amb el tema "Twenty Eleven")

3. Introducció a l'escriptori (*dashboard* o panell) de WordPress

En WordPress, la usabilitat del panell d'administració està molt cuidada i de seguida n'aprendrem el funcionament bàsic, basat en entrades, pàgines, categories i etiquetes (*tags*).

Exercici

Una vegada instal·lat el WordPress, proposem a l'alumne crear una entrada i una pàgina de contacte. Observarem que l'entrada apareix en la portada, mentre que la pàgina crea automàticament una nova opció en el menú. A més, les entrades tenen l'opció de definir categories i etiquetes, al contrari que les pàgines.

L'estructura bàsica de continguts en WordPress està basada en:

- **Entrades.** Els articles del blog que es poden vincular amb categories i etiquetes.
- **Pàgines.** Que apareixen automàticament en les entrades del menú principal.
- **Categories.** Vinculades a les entrades. Poden tenir una estructura jeràrquica.
- **Etiquetes (*tags*).** Per a etiquetar més específicament les entrades.

Per descomptat, cada pàgina o entrada té el seu URL únic. L'adreça per defecte de les entrades té l'aspecte següent: <http://www.exemple.com/blog?p=xx>, en què "xx" és l'ID de l'entrada. En el cas de les pàgines, l'URL és del tipus "page_id=xx".

3.1. Sobre els *feeds* automàtics

Atès que el WordPress és una plataforma d'administració de blogs, un dels continguts que genera, a més de les pàgines úniques, és un *feed* d'entrades i un altre de comentaris. Aquests *feeds* es publiquen en els dos formats més utilitzats, RSS i Atom.

RSS (*really simple syndication*) és un dels formats existents de **sindicació de continguts**. Es tracta d'un arxiu en format XML que s'utilitza per a difondre (sindicar) continguts de manera que els lectors els puguin "consumir" sense necessitat d'utilitzar un navegador: en lloc de navegador utilitzen un agregador (com, per exemple, Google Reader).

Un **agregador** és un programa especial que llegeix aquestes fonts (*feeds*) periòdicament, les organitza, i notifica d'alguna manera a l'usuari quan es publiquen continguts nous en les fonts a les quals està subscrit.

A més, un RSS permet syndicar els continguts d'una font en un altre lloc.

Així, per exemple, en el nostre blog podríem tenir un apartat amb continguts que tenen origen en una altra font. En WordPress el podem aconseguir instal·lant algun *plug-in* sobre aquest tema com FeedWordPress o d'altres.

Hi ha diferents versions, com RSS 0.9, RSS 1.0 i RSS 2.0.

Quant a **Atom**, és un format que serveix per al mateix propòsit. Es tracta també d'un XML, encara que amb alguns dels noms d'etiquetes diferents.

Cal assenyalar que sovint es parla genèricament d'RSS, sense tenir en compte si es tracta realment d'un arxiu en format RSS o Atom.

3.2. Apartats principals de l'escriptori

A continuació analitzarem amb més detall cadascun dels apartats del panell d'administració (o escriptori) de WordPress:

- **Escriptori.**
- **Entrades.**
- **Multimèdia.**
- **Enllaços (*links*).**
- **Pàgines.**
- **Comentaris** (per a què serveix Akismet).
- **Aparença.**
- **Plug-ins.**
- **Usuaris** (rols).
- **Eines.**
- **Ajustaments.**

3.2.1. Escriptori

La primera opció, Escriptori, ens dona una **visió general** de l'estat del nostre blog a cada moment –per exemple, quants articles hem publicat i els comentaris que hem rebut–, a més de l'opció d'escriure ràpidament un esborrany d'article a publicació ràpida (*QuickPress*).

Pàgina web

Per a més informació sobre Atom, vegeu el Codex a: http://codex.wordpress.org/WordPress_Feeds.

Pàgina web

Es pot trobar més informació sobre el tema a <http://en.wikipedia.org/wiki/RSS>.

The screenshot shows the WordPress dashboard interface. On the left is a sidebar menu with options like 'Inicio', 'Actualizar', 'Entradas', 'Multimedia', 'Enlaces', 'Páginas', 'Comentarios', 'Apariencia', 'Plugins', 'Usuarios', 'Herramientas', and 'Ajustes'. The main area is titled 'Escritorio' and contains several widgets: 'Ahora mismo' (Now) showing content and comment counts; 'Comentarios recientes' (Recent comments) with a comment preview; 'Enlaces entrantes' (Incoming links); 'Plugins recientes' (Recent plugins); 'Publicación rápida' (Quick post) form; 'Últimos borradores' (Latest drafts); 'Blog oficial WordPress' (Official WordPress blog) with news about version 3.3.2; and 'Otras noticias sobre WordPress' (Other news about WordPress).

L'escritori (*dashboard*) de WordPress

L'escritori està organitzat en blocs que agrupen funcionalitats, com “Comentaris recents” o “Publicació ràpida”, que es diuen *ginys* (*wid-gets*). Aquests blocs funcionals es poden arrossegar i deixar anar en determinades zones de la pantalla per a canviar-los d'ordre. Podem personalitzar totalment l'escritori arrossegant els ginys on vulguem.

Vegeu també

Per a construir les zones estàtiques del nostre lloc web s'utilitza el mateix sistema, com veurem més endavant en l'apartat “Ginys”.

A més, en l'opció Opcions de pantalla (pestanya superior dreta) podem decidir quins ginys es mostren en pantalla i quantes columnes té el panell d'administració.

3.2.2. Entrades

És l'apartat principal en què se'ns mostren totes les entrades (també denominades *posts* o *articles*) que s'han escrit.

Apartat Entrades del panell d'administració

Observació

En entrar en aquest apartat per primera vegada apareix una entrada anomenada *Hola Món* (o *Hello World!* en la versió anglesa). Es tracta d'una entrada generada automàticament per WordPress com a mostra i que, per descomptat, podem esborrar quan vulguem.

En l'apartat “Entrades” tenim quatre subapartats:

- Entrades. Hi veiem tots els *posts* actuals.
- Afegir nova (*Add new*). L'opció per a crear una entrada nova.
- Categories (*Categories*). Editar les categories vinculades als *posts*.
- Etiquetes de les entrades (*Post tags*). Editar les etiquetes vinculades als *posts*.

En publicar una entrada tenim un editor de textos que ens permet donar format al text. A més, hi ha la possibilitat de pujar fotos, vídeos o sons fàcilment utilitzant l'opció Pujar (*Insert*). L'editor també permet editar directament el codi HTML activant la pestanya superior dreta, HTML.

3.2.3. Categories i etiquetes

Les **categories** responen a les **seccions** generals del nostre blog, mentre que les **etiquetes** (*tags*) són lliures i ens permeten etiquetar individualment els continguts. Aquestes etiquetes i categories ajudaran després els visitants del lloc a navegar pel contingut. A més, són importants per a la indexació en els cercadors.

Sempre serà una bona idea assignar una categoria i diverses etiquetes a cadascun dels articles. D'aquesta manera, després, els visitants del web podran navegar pel blog enllaçant aquestes paraules i trobar continguts relacionats.

3.2.4. Multimèdia

A Biblioteca multimèdia, podem veure tots els arxius que hem pujat prèviament en publicar articles (imatges o qualsevol altre tipus d'arxius). D'aquesta manera, els podem editar en qualsevol moment. També podem pujar més arxius (a Afegir nou) per a tenir recursos en el moment de publicar nous articles. Així, tenim l'opció de pujar directament imatges o arxius a la biblioteca que després podem usar en les nostres entrades o, el més pràctic en la majoria de les situacions, fer-ho en el moment de crear l'entrada.

Biblioteca multimèdia

Exercici

Publiqueu un *post* amb una imatge. A continuació, aneu a l'apartat "Biblioteca multimèdia" i comproveu que la imatge apareix en la llista.

En la llista podem filtrar els arxius per data o veure només els que encara no estan assignats a cap article.

Un dels avantatges de la biblioteca és que es generen automàticament versions de la imatge en diverses mides –per exemple, en format miniatura o en una mida apta per a l'amplada de les columnes de l'entrada. D'aquesta manera, l'editor no s'ha de preocupar de la mida original de la imatge. A Ajustaments podem canviar les mides predefinides.

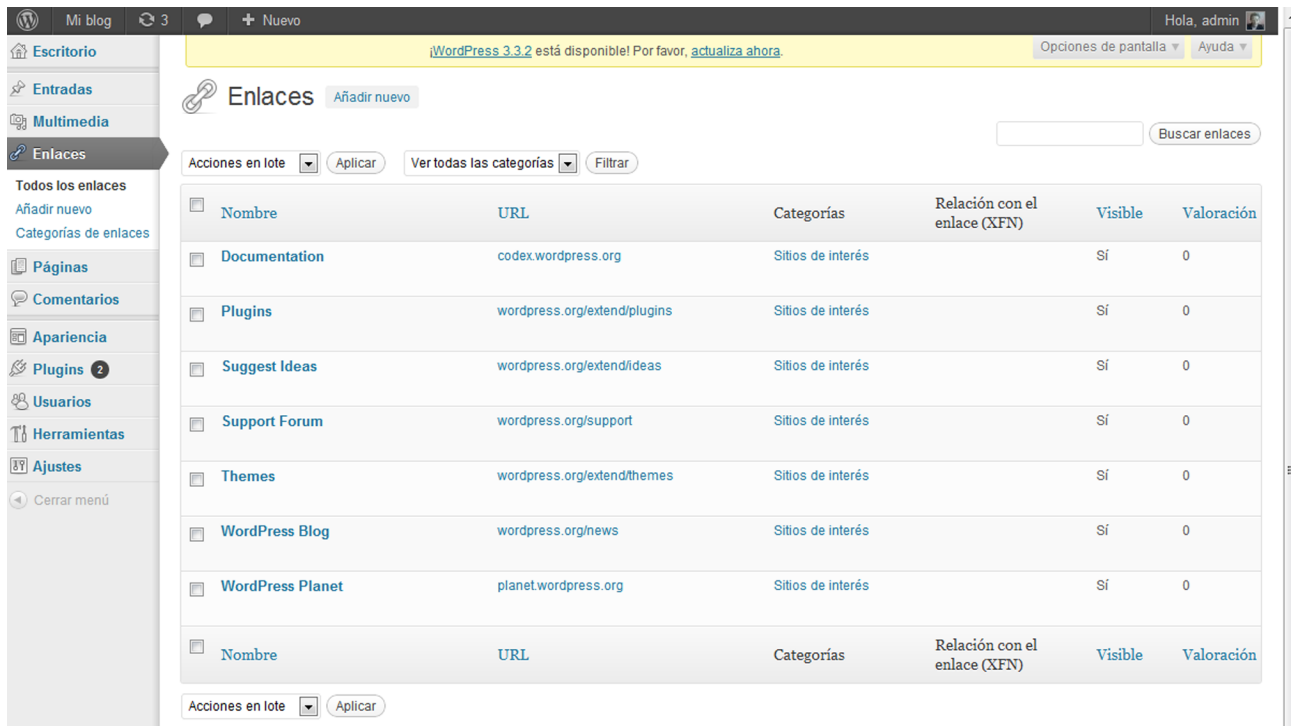
A més el WordPress té la funcionalitat de poder mostrar imatges en format galeria. En el moment en què inserim diverses imatges en el mateix article, ens oferirà aquesta opció. Encara que es pot utilitzar visualment, en realitat internament es crea un *shortcode* "[gallery]". Fins i tot és possible mostrar la galeria d'una altra entrada introduint el seu ID, "[gallery id="5"]" per exemple.

Vegeu també

Parlarem amb més detall dels *shortcodes*, un sistema per a personalitzar continguts en les entrades, en l'apartat "*Shortcodes*".

3.2.5. Enllaços (*links*)

Aquest apartat és senzillament una llista d'enllaços que podem modificar. Cada llista d'enllaços està agrupada en una categoria; també podem ampliar aquestes categories: per defecte, només hi ha “Llocs d'interès” (“Blogroll”).



The screenshot shows the WordPress 'Enlaces' (Links) management interface. The page title is 'Enlaces' and it includes a 'Añadir nuevo' (Add new) button. Below the title, there are controls for 'Acciones en lote' (Batch actions) and 'Aplicar' (Apply), and a dropdown for 'Ver todas las categorías' (View all categories) with a 'Filtrar' (Filter) button. A search box labeled 'Buscar enlaces' (Search links) is also present. The main content is a table of links with the following columns: 'Nombre' (Name), 'URL', 'Categorías' (Categories), 'Relación con el enlace (XFN)' (Relationship with the link (XFN)), 'Visible' (Visible), and 'Valoración' (Rating). The table lists several links, all categorized as 'Sitios de interés' (Sites of interest), with a rating of 0 and visible status set to 'Sí' (Yes).

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

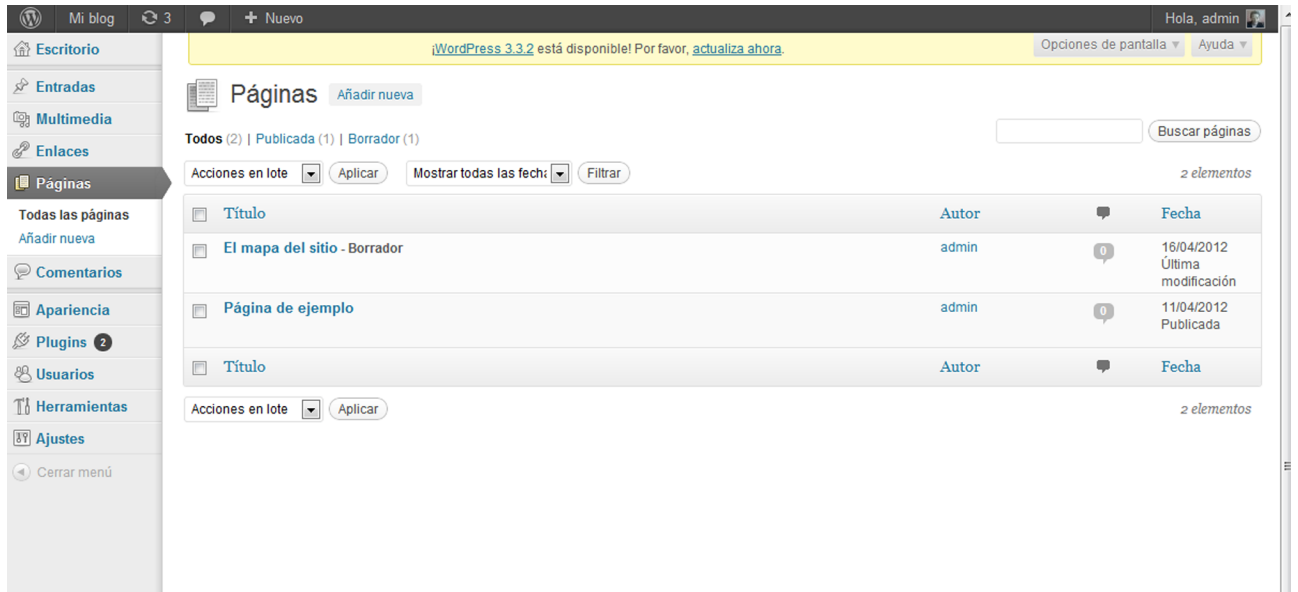
Apartat “Enllaços”

Igual que la resta d'apartats, té continguts de prova que probablement haurem d'eliminar en personalitzar el nostre lloc i afegir els nostres propis enllaços.

Els visitants del nostre blog veuran aquesta llista d'enllaços en un dels ginys que tenim disponibles per a construir el nostre lloc, com veurem més endavant.

3.2.6. Pàgines (*pages*)

En WordPress hi ha de manera predefinida **dos tipus de continguts**, les **entrades**, que ja hem vist i que es corresponen amb les entrades del blog en si mateix, i les **pàgines**.



WordPress 3.3.2 está disponible! Por favor, [actualiza ahora](#)

Páginas [Añadir nueva](#)

Todos (2) | Publicada (1) | Borrador (1)

Acciones en lote Mostrar todas las fechas 2 elementos

Título	Autor	Fecha
<input type="checkbox"/> El mapa del sitio - Borrador	admin	16/04/2012 Última modificación
<input type="checkbox"/> Página de ejemplo	admin	11/04/2012 Publicada

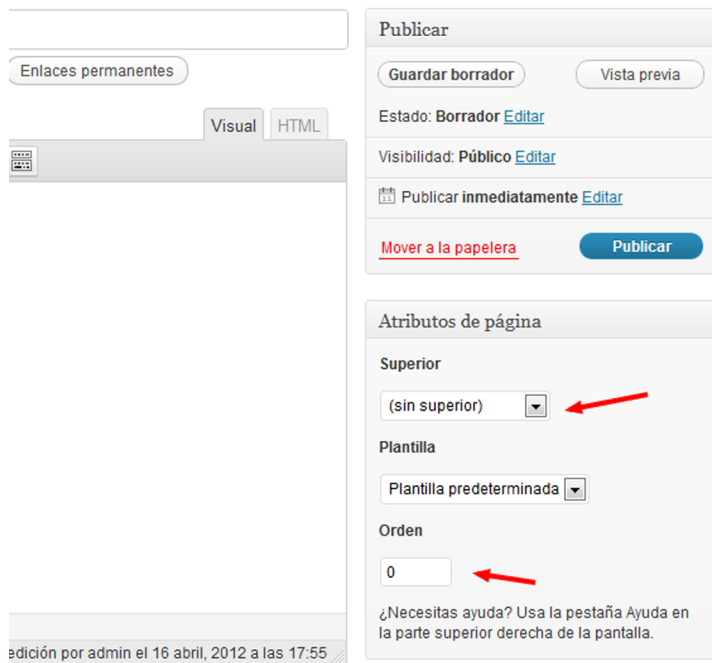
Acciones en lote 2 elementos

Apartat "Páginas"

L'apartat "Páginas" funciona exactament igual que l'apartat "Entrades". Aquest tipus de contingut està pensat per a pàgines de contingut estàtic, com podria ser una secció de contacte, una pàgina de presentació del nostre blog, un producte, etc.

Una de les diferències més importants entre entrades i pàgines és que aquestes últimes poden tenir una **estructura jeràrquica** en forma de seccions i subseccions (la qual cosa podria servir, per exemple, per a crear un catàleg de productes).

Per a crear aquesta estructura jeràrquica en el giny utilitzarem els atributs de la pàgina, que tenim en la barra lateral dreta, per decidir en quina secció o subsecció apareix cada pàgina. Al seu torn, cada secció o subsecció també és una pàgina.



The image shows a screenshot of the WordPress 'Publicar' (Publish) sidebar. The sidebar is divided into two main sections: 'Publicar' and 'Atributos de página' (Page Attributes). In the 'Publicar' section, there are buttons for 'Guardar borrador' (Save Draft) and 'Vista previa' (Preview). Below these, the status is 'Estado: Borrador' with an 'Editar' (Edit) link. The visibility is set to 'Público' with an 'Editar' link. There is also a 'Publicar inmediatamente' button with an 'Editar' link. At the bottom of this section are 'Mover a la papelera' (Move to Trash) and 'Publicar' buttons. The 'Atributos de página' section has three fields: 'Superior' (Parent) set to '(sin superior)' with a dropdown arrow and a red arrow pointing to it; 'Plantilla' (Template) set to 'Plantilla predeterminada' with a dropdown arrow; and 'Orden' (Order) set to '0' with a red arrow pointing to it. At the bottom of the sidebar, there is a small text box that says '¿Necesitas ayuda? Usa la pestaña Ayuda en la parte superior derecha de la pantalla.'

Organització jeràrquica de les pàgines

En el desplegable superior seleccionem a quina pàgina (secció) pertany la pàgina actual. Automàticament, aquestes seccions i subseccions apareixeran en el menú principal amb menús desplegable. El camp "Ordre" ens serveix per a decidir l'ordre de les pàgines en el menú.

Una altra diferència respecte a les entrades és que les pàgines no tenen ni categories ni etiquetes.

Exercici

Creeu una pàgina en el blog. Afegiu aquesta pàgina en el menú. Creeu pàgines en forma de subseccions.

3.2.7. Comentaris

Una de les característiques més importants de qualsevol blog és la possibilitat d'afegir comentaris per part dels visitants del lloc.



Apartat "Comentarios"

En aquest apartat podem veure tots els comentaris del lloc i decidir quins seran visibles i quins volem eliminar (per exemple, en cas que es tracti d'*spam*).

En el subapartat "Ajustaments de comentaris" de l'apartat "Ajustaments" podem canviar la mecànica d'aprovació. Una possibilitat seria que els comentaris no es publiquin fins que l'administrador no els aprovi, però hi ha altres procediments com deixar publicar als usuaris registrats, etc.

El WordPress inclou per defecte un *plug-in* molt recomanable anomenat Akismet, que, una vegada activat, marca automàticament com *a spam* els comentaris maliciosos i permet l'aprovació automàtica de la resta. Hi ha molts altres *plug-ins* que poden ajudar en aquesta tasca com, per exemple, Bad behaviour (<http://wordpress.org/extend/plugins/bad-behavior/>).

3.2.8. Aparença (*appearance*)

Aquest és un dels apartats més importants del panell d'administració o escriptori i està dividit, com a mínim, en tres subapartats:

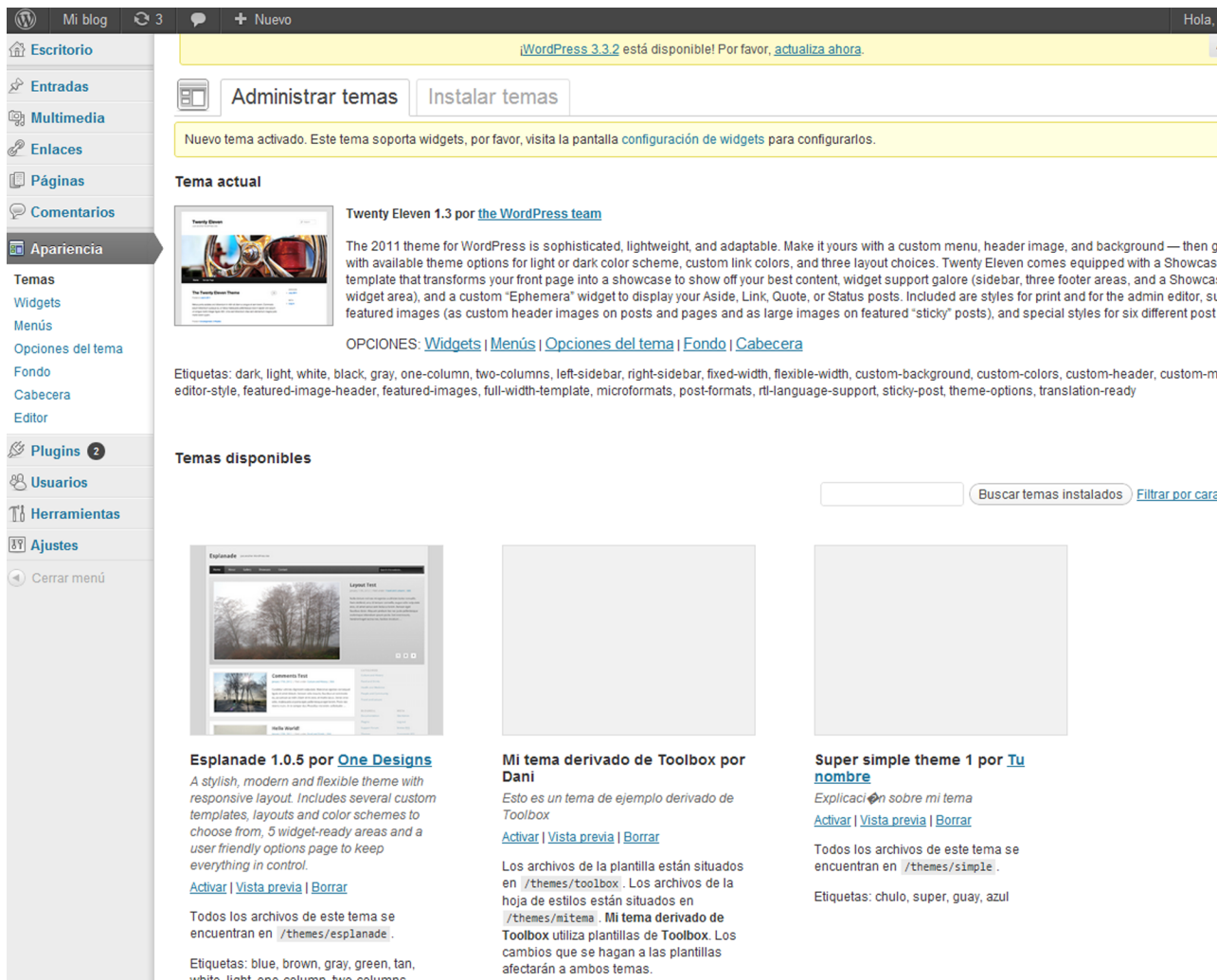
- Temes
- Ginys
- Editor

Segons la implementació del tema que tinguem activada, aquí poden aparèixer altres opcions (que afecten l'aspecte de menús, fons, capçalera, etc.).

Temes

Com a bon gestor de continguts, el WordPress separa totalment el contingut de la presentació.

La presentació és tasca dels **temes** (*themes*).



Apartat "Temas" de l'escriptori de WordPress

Bàsicament, hi ha tres maneres d'instal·lar temes en WordPress:

- Autoinstal·lador. La manera més senzilla. Simplement cal activar la pestanya Instal·lar temes, buscar el tema (amb paraules clau o característiques) i activar-lo.
- Pujar zip. També tenim l'opció de pujar un arxiu zip que contingui tots els arxius del tema. El WordPress descomprimirà i copiarà automàticament el tema en el directori corresponent.
- Manual mitjançant FTP. Pujar per FTP el tema en el directori "wp-content/themes".

En cas que no tinguem accés al tema que volem mitjançant l'instal·lador automàtic, sempre podem descarregar els temes que vulguem des de diferents repositoris disponibles en el web i usar la segona o tercera opció per a instal·lar-los.

En qualsevol cas, el resultat serà sempre el mateix: una vegada instal·lat un tema tindrem un directori a `wp-content/themes` amb els arxius corresponents a aquest tema. Aquests arxius són en gran part plantilles pensades per a presentar el contingut de WordPress.

Per descomptat, podem instal·lar qualsevol tema, no sols necessàriament els disponibles en el lloc oficial (encara que només els temes catalogats aquí estan disponibles en fer cerques en l'autoinstal·lador de WordPress). A Internet trobarem diversos repositoris de temes, com <http://spyrestudios.com/wordpress-powered-websites/>. En alguns, fins i tot es comercialitzen temes i, en contrapartida, s'ofereix suport tècnic per al seu manteniment.

Una vegada hem trobat el tema que volem utilitzar, el procediment serà:

- Descarregar el zip.
- Pujar el zip directament o descomprimir els arxius i copiar la carpeta del tema a `wp-content/themes`.
- Anar al panell d'administració, en l'apartat "Aparença", i activar-lo.

Exercici

Instal·leu i activeu el tema descarregat que vulgueu.

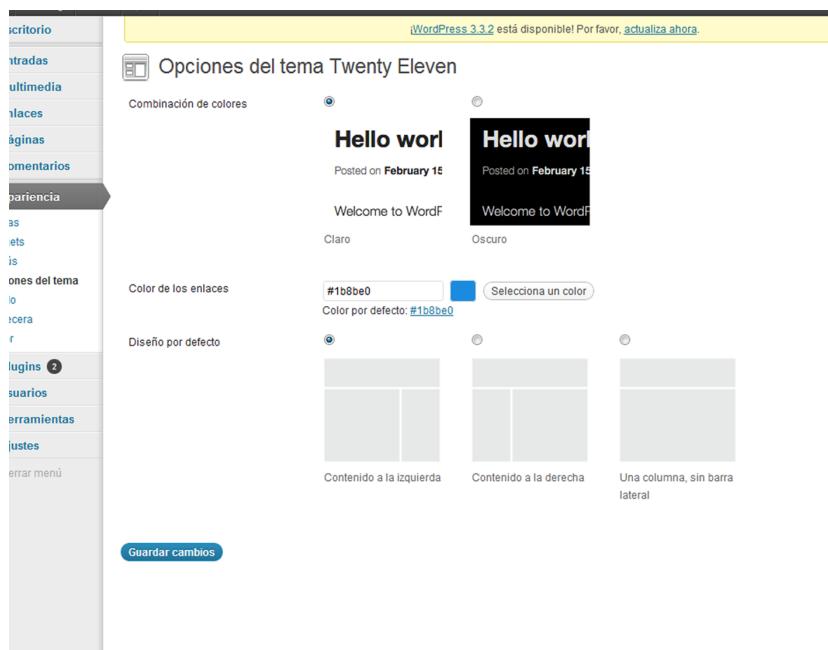
Finalment, en aquest apartat és possible que trobem opcions específiques dels temes que ja tinguem activats. Un tema pot ser tan complicat com puguem imaginar i incloure diverses opcions per a personalitzar-lo.

Personalització del tema "Twenty Eleven"

"Twenty Eleven" és un dels temes instal·lats per defecte en la distribució de WordPress. És un tema que podem utilitzar com a base per a personalitzar el nostre. Té diferents opcions que podem canviar des de l'escriptori de WordPress.

Vegeu també

Veurem com podem crear els nostres propis temes o personalitzar-los a partir d'un tema base en l'apartat "Introducció als temes en WordPress".



Opcions de personalització del tema "Twenty Eleven"

Per exemple, podem canviar la combinació de colors (fons text), el color dels enllaços i fins i tot el *layout* bàsic.

Exercici

Activeu el tema "Twenty Eleven", canvieu les opcions de manera que el fons del tema sigui negre i escolliu un color diferent per als enllaços. Comproveu el resultat.

Ginys

Els **ginys** són una manera senzilla i intuïtiva de canviar fàcilment l'aparença (i l'estructura) del nostre lloc web. En l'apartat "Ginys" podem arrossegar i deixar anar (*drag-and-drop*) diferents blocs en les regions que tinguem definides en el tema actual.

El WordPress té de partida una sèrie de ginys predeterminats, com per exemple "Cercar", "Comentaris recents", "Entrades recents", etc., i en activar determinats *plug-ins* en poden aparèixer més.

Exercici

Arrossegeu el giny "Calendari", que mostra un calendari amb la data de les entrades, i comproveu el resultat en la vista del lloc.

Cada tema té definides una sèrie de regions on podem arrossegar aquests ginys. En el cas de "Twenty Eleven", per exemple, hi ha cinc zones (incloses les tres columnes del *footer*). En altres temes en podem tenir més o, fins i tot, tenir només una barra lateral en algun tema simple.

La definició d'aquestes zones es fa en l'arxiu `functions.php` del tema utilitzant la funció `register_sidebar` del Codex (http://codex.wordpress.org/Function_Reference/register_sidebar).

Una de les novetats de la versió 3 de WordPress és la possibilitat de crear tants **menús** personalitzats com vulguem i assignar-los a les regions utilitzant el giny Menú personalitzat; fins i tot els podem utilitzar en lloc del menú principal predeterminat.

D'altra banda, i segons el tema que tinguem activat, podem tenir el subapartat "Opcions del tema". És el cas del tema per defecte ("Twenty Eleven"), en què podem canviar la configuració bàsica de colors i el *layout*.

Si el tema ho permet, podem, per exemple, canviar fàcilment la imatge de fons o de la capçalera. En realitat, totes aquestes opcions es defineixen en l'arxiu `functions.php` de cada tema.

Per exemple, per a activar l'opció de configuració de la imatge de capçalera tindriem:

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

Pàgina web

Per a més informació, vegeu el Codex (<http://codex.wordpress.org/Custom-Headers>).

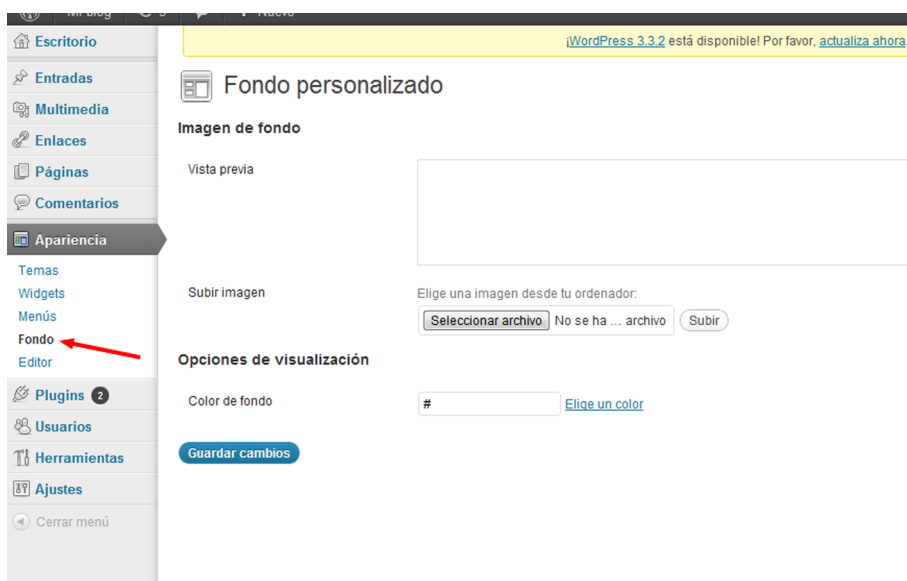
Mentre que la instrucció

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

activaria, d'una banda, l'opció Fons en el panell d'administració que permetria canviar la imatge, i, d'una altra, associaria la imatge com a estil *background-image* en el cos (*body*) del tema.

Observació

No obstant això, en el tema anterior per defecte de WordPress, "Twenty Ten", aquestes opcions no existeixen.



Opció Aparença/fons una vegada activada a funcions.php

Editor

En l'apartat “Editor” podem editar tots els arxius que componen el tema, la qual cosa permet la personalització avançada dels temes. Per aquest motiu, aquesta opció (igual que altres d’avançades) només és visible per a l'administrador però no, per exemple, per al rol “Editor”.

És una eina pensada per a incloure fragments de codi HTML o PHP ràpidament o per a modificar el CSS, però en la majoria dels casos **no és una opció recomanable**. Sempre és molt millor utilitzar un bon editor que ens ajudarà en la sintaxi i evitarà cometre errors.

3.2.9. *Plug-ins*

Els *plug-ins* són extensions que ens permeten afegir funcionalitats al WordPress, com crear una tenda amb la seva complexa administració, afegir estadístiques, millorar el sistema de comentaris, implementar una xarxa social, etc.

Hi ha infinitat de *plug-ins* per al WordPress. En la pàgina web <http://wordpress.org/extend/plugins/> en podem trobar més de 10.000.

Igual que els temes, es poden instal·lar directament des de l'escriptori de WordPress o manualment en el directori wp-content/plugins.

Alguns *plug-ins* recomanats:

Vegeu també

Veurem com funcionen els rols en WordPress en l'apartat “Usuaris”.

Observació

Cal tenir molta cura en editar aquests arxius, ja que si esborrem o alterem alguna cosa important és possible que el tema deixi de funcionar.

Vegeu també

Veurem aquests editors en l'apartat “Alguns editors de codi”.

- Google XML sitemaps (<http://wordpress.org/extend/plugins/google-sitemap-generator/>). Crea un arxiu *sitemap*.
- Database manager (<http://wordpress.org/extend/plugins/wp-dbmanager/>). Gestiona còpies de seguretat en la base de dades.
- Feedburner Feedsmith (<http://wordpress.org/extend/plugins/feedburner-setting/>). Integra el WordPress amb el Feedburner.
- W3 Total Caché (<http://wordpress.org/extend/plugins/w3-total-cache/>). Millora la *caché*.
- Custom posts per page (<http://wordpress.org/extend/plugins/custom-posts-per-page/>). Permet modificar el nombre d'entrades en diferents situacions.
- All in one SEO pack (<http://wordpress.org/extend/plugins/all-in-one-seo-pack/>). Optimització per a cercadors.
- 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.

Vegeu també

Veurem els *plug-ins* amb més profunditat en l'apartat "Desenvolupament d'un *plug-in* en WordPress".

3.2.10. Usuaris

El WordPress és un **sistema multiusuari**. En aquest sistema cada usuari té assignat un rol determinat. Un **rol** defineix la llista de permisos que cada tipus d'usuari té activats. Un **permís** pot ser, per exemple, "Editar una entrada", "Publicar una entrada", "Moderar els comentaris", etc.

El rol més bàsic, implícit, és l'usuari anònim: un usuari que no s'ha identificat. Moltes vegades es parla de l'usuari anònim del web, ja que en molts casos és l'usuari més habitual.

Una manera de comprovar el comportament del nostre lloc segons el rol que tingui l'usuari és usar dos navegadors diferents (per exemple, Chrome i Firefox o Opera), de manera que en un dels navegadors podem conservar l'usuari administrador i en l'altre estar en una sessió com a usuari anònim o amb un altre rol diferent.

El WordPress defineix una sèrie de rols per defecte que, ordenats de menys a més permisos, són els següents:

- **Subscriptor.** S'ha registrat en el lloc. Pot escriure comentaris utilitzant el seu compte, però només té permisos de lectura per a la resta.
- **Col·laborador.** Té accés a algunes funcions d'administració del lloc, com els comentaris, pàgines, etc.
- **Autor.** Pot crear entrades i publicar-les.
- **Editor.** Pot crear entrades però no publicar-les.
- **Administrador.** Té permisos per a fer qualsevol cosa.
- **Superadministrador.** En el cas d'instal·lacions WordPress *multi-site*, hi ha aquest nivell superior.

Imaginem, per exemple, el cas d'un blog amb diversos autors: probablement hi haurà diversos usuaris amb el rol "Autor", però també almenys un usuari amb el rol "Administrador".

A Usuaris tenim l'opció d'editar el compte dels usuaris que ja hi ha o d'afegir-ne de nous amb diferents rols. Els permisos assignats a cadascun dels rols estan perfectament documentats en el Codex (http://codex.wordpress.org/Roles_and_Capabilities).

3.2.11. Eines (*tools*)

A més de l'eina "Publicar això", que podem arrossegar a la barra de favorits del nostre navegador, tenim l'opció d'importar/exportar dades a d'altres CMS.

En instal·lar *plug-ins* és possible que aquí apareguin més opcions.

3.2.12. Ajustaments (*settings*)

En aquest apartat podem trobar totes les opcions de configuració. Destacarem les més importants:

- **Generals.** El títol del lloc i les opcions generals que podem canviar en qualsevol moment.
- **Esriptura (*writing*).** Hi tenim opcions que ens permeten publicar en el blog remotament, des d'altres sistemes o serveis (com Flickr) o des del correu electrònic. En aquest apartat també veiem al final l'opció *Update services*, en la qual tenim una llista dels servidors als quals el WordPress avisa cada vegada que publiquem contingut nou en el nostre blog.

- **Lectura** (*reading*). Hi tenim diverses opcions relatives a com es mostren els articles en el nostre lloc.
- **Comentaris**. Hi tenim diverses opcions relatives al flux de publicació de les entrades i comentaris, i maneres de mostrar els avatars dels autors.
- **Multimèdia** (*media*). Hi tenim opcions relatives a com es retallen les imatges quan les pugem.
- **Privadesa**. Permet donar accés als motors de cerca (com *Google*) o bloquejar-lo.
- **Enllaços permanents** (*permalinks*). Hi podem definir com es construeixen les rutes de les pàgines del nostre blog (és recomanable activar-lo).

Enllaços permanents i “.htaccess”

Els enllaços a les pàgines en WordPress per defecte tenen l'aspecte següent: `?p=123`, en què el número representa l'ID de l'entrada (o de la pàgina). En el cas de la resta de pàgines també es passen paràmetres per l'URL, per exemple `?cat=4` correspon a la llista d'entrades de la categoria amb ID = 4.

No obstant això, tant pel que fa al posicionament web com a la usabilitat per a l'usuari, és molt millor utilitzar enllaços permanents del tipus `/pagina_exemple`. D'aquesta manera, en compartir l'enllaç l'usuari té una lleugera idea del contingut.

De fet, a Ajustaments/enllaços permanents, és possible triar diferents estructures per als enllaços combinant el títol del contingut amb la data. Perquè això funcioni, el WordPress genera un arxiu `.htaccess` que permet al servidor web Apache “reescriure” els URL. Els arxius `.htaccess` permeten activar opcions del servidor web Apache, en aquest cas es tracta de readreçaments, però podria ser qualsevol altra opció.

Pàgina web

Per a més informació sobre enllaços permanents i `.htaccess`, vegeu el Codex a: 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
RewriteCond %{REQUEST_FILENAME} !-d
RewriteRule . /sandbox/wordpress/index.php [L]
</IfModule>
# END WordPress
```

4. Configuració i seguretat

Malgrat els grans avantatges de les aplicacions de codi obert, també hem de ser conscients dels possibles problemes de seguretat i estar sempre pendents de les actualitzacions.

En aquest sentit es recomana:

- **Actualitzar sovint.** Actualitzar sempre que apareguin actualitzacions de seguretat importants. A més, hem de tenir la precaució de fer una còpia de seguretat de la base de dades prèviament, per si alguna cosa va malament. D'altra banda, les actualitzacions ens permeten gaudir de possibles noves funcionalitats.
- **Canviar el compte de l'administrador.** Per seguretat és recomanable no usar *admin* com a usuari administrador. És preferible usar un nom arbitrari i una contrasenya prou robusta. També és recomanable canviar la contrasenya de tant en tant.
- **Servidor de proves i de producció.** És recomanable, especialment si la instal·lació està molt personalitzada, tenir un servidor de proves i un altre de producció. Abans de pujar els canvis al servidor de producció ens hem d'assegurar que tot està correcte en el servidor de proves. En cas que estiguem desenvolupant un tema o un *plug-in*, també seria recomanable utilitzar algun sistema de control de versions, com git (<http://git-scm.com/>).
- **Canviar la ubicació aparent de WordPress.** En amagar la ubicació real de WordPress aconseguirem que un accés no autoritzat sigui molt més difícil, a més de tenir els arxius més ordenats en cas que tinguem altres aplicacions en el servidor. Per exemple, podem apuntar WordPress a l'arrel del nostre servidor encara que realment els arxius siguin en el directori Blog. El procés consisteix en dos passos, primer moure l'arxiu *index.php* a la ubicació en què volem que es mostri (per exemple, en l'arrel) i després editar la inclusió de l'arxiu *wp-blog-header.php* perquè la ruta sigui correcta. A més, prèviament hem de modificar la informació del lloc a "Ajustaments".
- **Filosofia *don't hack core*.** Sempre que vulguem modificar alguna cosa, ho hem de fer només en els arxius del nostre tema, i si tenim una necessitat molt específica, desenvolupar un tema o *plug-in* a mida. **Mai no hem de modificar els arxius principals de WordPress** (o de qualsevol altre CMS). D'aquesta manera, en actualitzar-lo no perdrem els canvis. A més, modificar els arxius principals de WordPress també podria implicar riscos de seguretat.

Pàgina web

El procés s'explica detalladament en la pàgina web següent:

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

5. Introducció als temes en WordPress

Un **tema** de WordPress consisteix en un conjunt de **plantilles** PHP que contenen l'estructura i s'encarreguen de generar el contingut dinàmic i el CSS que s'encarrega de la presentació.

Vegem detalladament quines són aquestes plantilles i fins a quin punt les podem modificar.

5.1. Alguns editors de codi

Per a crear el nostre propi tema de WordPress, necessitarem un bon editor d'HTML, CSS i PHP. A continuació en proposem alguns:

- **Dreamweaver** (<http://www.adobe.com/products/dreamweaver.html>). Un potent editor d'Adobe amb funcions avançades com el MultiScreen View. Per a plataformes Windows i Mac.
- **Notepad ++** (<http://notepad-plus-plus.org/>). Editor de codi font. Per a plataforma Windows.
- **Smultron** (<http://www.peterborgapps.com/smultron/>). Editor web senzill per a la plataforma Mac. Es poden trobar en la Mac App Store.
- **Aptana Studio** (<http://www.aptana.com/>). És una evolució de l'editor de codi obert Eclipse adaptat al desenvolupament web. Preparat per a editar HTML, CSS i JavaScript, a més de PHP. Per a plataformes Windows, Mac i Linux,
- **Netbeans** (<http://netbeans.org/>). Editor avançat per a PHP i altres llenguatges. Per a plataformes Windows, Linux i Mac OS X.

5.2. Creació d'un tema de WordPress des de zero

Encara que el procés de creació d'un tema des de zero pot ser laboriós, i en la majoria dels casos és preferible reutilitzar un tema ja existent, com es comentarà més endavant, es tracta d'un bon exemple per a comprendre el sistema de gestió de temes en WordPress.

En realitat, el procediment per a crear un tema simple és molt senzill. Els passos són els següents:

- Crear una carpeta dins de wp-content/themes amb el nom que vulguem.
- Crear un arxiu “screenshot.png” de 300px × 225px i copiar-lo dins d'aquesta carpeta (opcional, aquesta és la imatge que apareix de mostra en l'administrador).
- Crear un arxiu CSS anomenat *style.css* amb el contingut següent:

```
/*
Theme Name: El nom del meu tema
Theme URI: http://lapaginadelmeutema.com
Description: Explicació sobre el meu tema
Version: 1
Author: El teu nom
Author URI: http://lamevapaginaweb.com
Tags: duescolumnes, blau, arial
*/
```

- Finalment, crear un arxiu *index.php*, que serà la plantilla principal del nostre tema. Dins d'*index.php* haurem d'escriure unes instruccions de PHP perquè el contingut sigui dinàmic.

Aquestes instruccions específiques de WordPress s'anomenen ***the loop*** (el bucle) i són la **base del funcionament dels temes en WordPress**. El *loop* es basa en un bucle que busca els continguts que han d'aparèixer a cada pàgina (segons si som en la portada, en la pàgina única d'una entrada, en la llista d'una categoria, etc.) i mostra els camps que necessitem. Els camps sempre comencen sempre per l'article anglès *the*:

- `the_permalink()`. Mostra l'enllaç a la pàgina única del contingut
- `the_title()`. Mostra el títol del contingut
- `the_content()`. Mostra el contingut.
- Etc.

Un exemple de plantilla molt senzilla podria ser:

```
<?php get_header(); ?> <!-- inclou la capçalera -->
<?php if ( have_posts() ) : ?>
  <!-- si hi ha contingut en aquesta pàgina -->
  <?php while ( have_posts() ) : the_post(); ?>
  <!-- mentre hi hagi continguts -->
  <a href="<?php the_permalink()?">">
  <!-- crea un enllaç amb el permalink -->
  <?php the_title();?></a> <!-- el títol del contingut -->
  <?php the_content();?> <!-- el contingut -->
  <?php endwhile; ?>
<?php endif; ?>
<?php get_sidebar(); ?> <!-- inclou el sidebar -->
<?php get_footer(); ?> <!-- inclou el footer -->
```

Observació

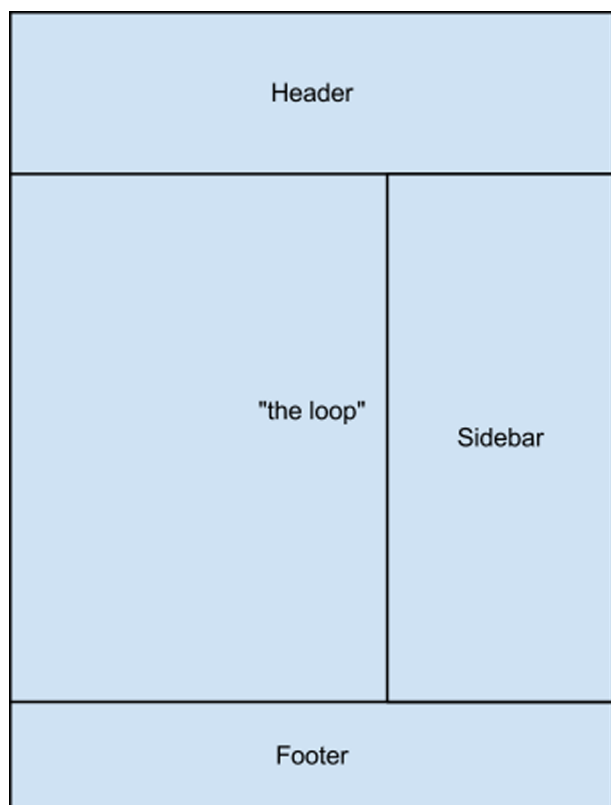
Com que només hem definit una única plantilla `index.php`, totes les pàgines es veuran amb aquesta estructura, és a dir, que **tots els continguts utilitzaran aquesta plantilla**. A més, com que no hem afegit cap codi CSS, tampoc no hi haurà ni estil ni *layout*. No obstant això, el lloc web és totalment funcional i es pot navegar per tots els continguts; WordPress, "màgicament", s'encarrega de reproduir els continguts dinàmicament.

Exercici

Seguiu els passos anteriors per a crear un tema i activar-lo.

En un tema més elaborat trobaríem plantilles específiques per a les pàgines (`page.php`), entrades (`single.php`), categories (`category.php`), etc. Si volguéssim fer evolucionar més el tema hauríem d'afegir aquestes plantilles addicionals. Probablement, en la plantilla d'una pàgina no ens interessa mostrar la data i l'autor, en canvi en la plantilla d'una entrada, sí.

En la majoria de plantilles principals s'inclouen subplantilles. Per exemple, quan el WordPress troba la instrucció `get_header`, en realitat mira si tenim definit un arxiu `header.php`; si no el troba, utilitza una capçalera (*header*) per defecte. Passa el mateix amb el *sidebar* i el *footer*. Per a personalitzar realment el nostre tema hauríem de definir aquestes subplantilles, `header.php`, `footer.php` i `sidebar.php`.



Esquema d'una plantilla principal

El problema de crear un tema des de zero és que, encara que tenim flexibilitat total, **haurem de verificar el tema a fons** abans de donar-lo com a bo. Almenys hauríem de comprovar el següent:

- Que l'HTML i el CSS siguin vàlids.
- Que no hi hagi errors de JavaScript.
- Verificar-lo en diferents navegadors, com IE7, IE8, IE9, Safari, Chrome, Opera i Firefox, en l'escriptori, més una bona selecció de navegadors en dispositius.
- Eliminar comentaris innecessaris que puguem haver deixat en els arxius.

Aquestes comprovacions no són tan necessàries si utilitzem el procediment de crear un tema fill derivant-ne un de ja existent que de seguida veurem.

5.2.1. Anatomia d'un tema

Més detalladament, les **plantilles principals** que formen part d'un tema són les que es poden veure en el gràfic següent marcades en taronja:

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 is 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

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

En essència qualsevol de les plantilles principals de WordPress està composta de **subplantilles**. **index.php**, per exemple, inclou les subplantilles següents:

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

A més, totes les plantilles tenen les instruccions del *loop* que permeten al WordPress obtenir el contingut.

La instrucció

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

obté en realitat l'arxiu header.php i l'inclou. Llavors, ens podem preguntar per què no utilitzem directament, per exemple,

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

La resposta és que aquesta crida de WordPress pren decisions, com per exemple incloure la capçalera del tema “pare” (si no està definit en el tema actual) o incloure una capçalera per defecte si tampoc no està inclosa en el “pare”, a més de fer altres operacions internes de WordPress.

Encara que `index.php` és la plantilla principal, hi ha **altres plantilles importants**:

- **single.php**. Si està definida, és la plantilla per a les entrades.
- **page.php**. Si està definida, és la plantilla per a les pàgines.
- **archive.php**. És la plantilla de totes les llistes d'arxiu (llista per dates, per categories, per etiquetes, etc.).
- **category.php** i **tags.php**. Aquestes són plantilles més específiques. Si la llista que es mostra és d'una categoria o una etiqueta i està definida, en lloc d'`archive.php` s'han d'utilitzar aquestes.

El sistema de plantilles funciona de manera jeràrquica. A partir d'una adreça concreta en l'estructura de continguts del nostre WordPress, per exemple la llista de la categoria “Dibuix” que tenim definida, el WordPress seguirà aquesta seqüència: en primer lloc buscarà si existeix l'arxiu **category-dibuix.php**; en cas contrari, buscarà **category-xx.php** (en què `xx` és l'ID de la categoria); després buscarà **category.php**, i si no l'ha trobat rastrejarà **archive.php**. Fins i tot en aquest últim cas, si no ha trobat `archive.php`, utilitzarà **index.php** com a plantilla.

Aquest sistema de jerarquia permet una gran flexibilitat per a personalitzar el tema segons els continguts que estigui mostrant.

5.3. Creació de temes fills (*child themes*)

Tret que vulguem un tema amb una estructura o funcionalitat molt específica, una bona opció és crear un **tema derivat d'un tema que ja existeix** (*child theme*). D'aquesta manera, estalviem la feina que representa crear les plantilles i tindrem llibertat per a poder personalitzar la presentació.

Els passos per a crear un tema derivat d'un altre són molt simples:

Lectura recomanada

Per a més informació es pot veure el bon resum següent sobre el funcionament dels temes en WordPress:

Joost de Valk (2011). “Anatomy of a WordPress theme” <http://yoast.com/wordpress-theme-anatomy/>.

Pàgina web

Per a més informació, es pot consultar l'esquema complet de jerarquies en el Codex: http://codex.wordpress.org/images/1/18/Template_Hierarchy.png.

- Crear un directori dins de temes amb el nom que vulguem.
- Crear un arxiu `style.css` i, a part dels camps que ja coneixíem (vegeu l'apartat anterior), afegir:

```
Template: foo
```

Aquesta és la línia important que especifica que aquest tema està derivat d'un altre (el que es diu *foo* i, per tant, es troba a *Themes/Foo*).

- A més en el CSS del nostre tema hem d'importar el CSS del tema “pare” (sempre imaginant que el tema pare es diu *foo*):

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

Exercici

Partiu d'un tema pare i modifiqueu el CSS per adaptar-lo al nostre disseny, per exemple:

- Canviar la tipografia.
- Canviar alguns colors del text.
- Canviar algunes imatges.
- Canviar les vores i els marges.

Si no necessitem fer canvis en l'estructura de continguts (camps de contingut que es presenten), serà suficient personalitzar el CSS, però si volem canviar l'estructura també haurem de modificar les plantilles PHP corresponents.

5.3.1. Plantilles de pàgines

En realitat, en WordPress hi ha un mecanisme especial, més senzill, per a crear plantilles que no utilitza el mecanisme de la jerarquia. És un mètode molt més simple que es pot utilitzar en casos puntuals. És el mecanisme dels noms de plantilla o *template names* en anglès.

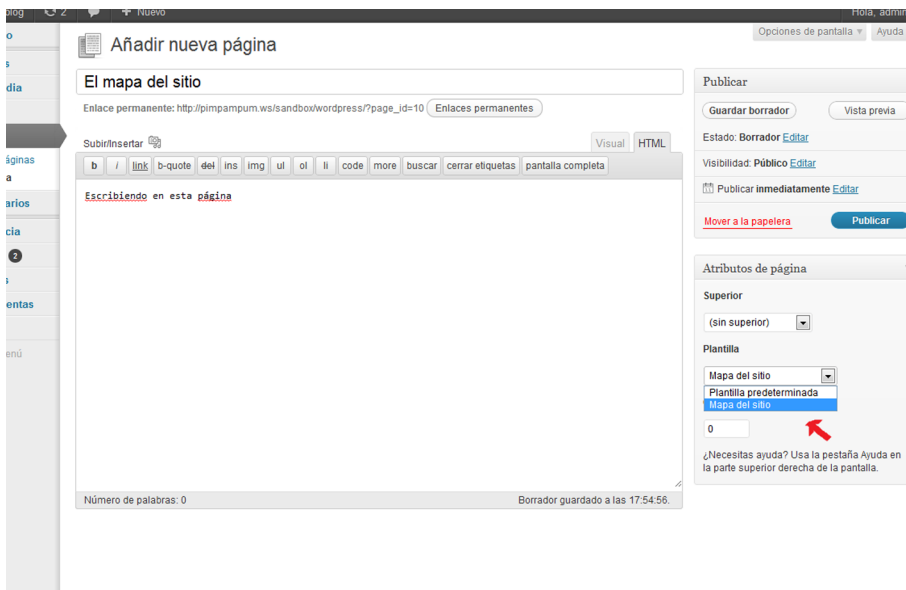
Si creem un arxiu PHP (amb el nom que vulguem) dins del nostre tema, simplement escrivint el comentari següent de PHP al principi,

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

podrem seleccionar la plantilla en el moment de crear una nova pàgina de continguts, com es veu en la captura següent.

Observació

En el cas d'un tema derivat, no és imprescindible crear un `index.php`: simplement amb l'arxiu `style.css` ja podríem crear un tema personalitzat mitjançant el CSS.



En crear una nova pàgina podem seleccionar plantilles personalitzades usant el mètode de noms de plantilla.

Per descomptat, a part del comentari és necessari que hi hagi la resta del codi PHP (s'ha d'incloure la capçalera, el *footer*, el *sidebar* i el *loop*).

L'avantatge d'aquest sistema és que podem personalitzar la plantilla com vulguem i afegir codi HTML o PHP que no existeixi en la resta.

Per exemple, podríem crear una pàgina en què, a més dels continguts normals extrets del *loop*, es mostrés una llista de totes les pàgines amb la instrucció següent:

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

5.3.2. Més sobre el *loop*

Qualsevol plantilla ha de (o pot) utilitzar el *loop*, que mostra els continguts que corresponen a la pantalla actual.

Un *loop* molt simple podria ser el següent:

```
<?php get_header(); ?>
<!-- Comença el loop. -->
<?php
while ( have_posts() ) : the_post(); ?>
<!-- Mostrar el títol vinculat amb el permalink -->
<h2><a href="<?php the_permalink() ?>" ><?php the_title();?></a></h2>
<!-- Mostrar el contingut de l'entrada dins d'un div -->
<div class="entry">
<?php the_content(); ?>
</div>
<?php endwhile; ?>
<!-- Acaba el loop. -->
```

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

Però podem anar afegint més coses; a continuació, el codi de la plantilla amb un *loop* més complet:

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

<!-- el codi comprova si el contingut pertany a la categoria amb id=3 -->
<!-- si és així hi assignem la classe "post-cat-3". -->
<!-- en cas contrari la classe serà "post". -->
<?php if ( in_category('3') ) { ?>
<div class="post-cat-3">
<?php } else { ?>
<div class="post">
<?php } ?>

<!-- Mostrar el títol enllaçat amb el permalink -->
<h2><a href="<?php the_permalink() ?>" rel="bookmark" title="Enllaç permanent a
<?php the_title_attribute(); ?>"><?php the_title();?></a></h2>

<!-- Mostrar la data en format f/m/Y i un enllaç a la resta d'entrades de l'autor-->
<small><?php the_time('d/m/Y') ?> by <?php the_author_posts_link() ?></small>
<!-- Mostrar el contingut de l'entrada en un contenidor -->
<div class="entry">
<?php the_content(); ?>
</div>

<!-- Mostrar una llista separada per comes de categories -->
<p class="postmetadata">Publicat a <?php the_category(', '); ?></p>
</div> <!-- tancar el primer contenidor-->
<!-- acabar el loop, i obrir una entrada else, per si no hi ha continguts -->
<?php endwhile; else: ?>
<!-- Si no hi ha continguts -->
<p>Ho sento, no es van trobar entrades</p>
<!-- Tancar el loop definitivament. -->
<?php endif; ?>

<!-- Incloure el sidebar i el footer -->
<?php get_sidebar(); ?>
<?php get_footer(); ?>
```

En alguns casos necessitarem personalitzar el *loop* per a filtrar determinats tipus de contingut.

Per exemple, podríem excloure algunes categories en afegir abans del bucle:

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

En aquest exemple s'exclouen les categories amb ID 3 i 8 (fixeu-vos en el signe – que precedeix els ID).

5.4. Temes *framework*

Una de les millors opcions per a crear temes personalitzats en WordPress és partir d'un tema *framework*. Es tracta de temes en què l'estructura està molt ben definida i que són configurables, de manera que “només” queda per definir la part de presentació (bàsicament en el CSS), és a dir tipografies, colors, imatges, etc.

De fet, com tema base podem utilitzar qualsevol tema de WordPress, però els temes *framework* estan pensats especialment per a això.

Hi ha diversos temes *framework*, i fins i tot el tema “Twenty Eleven” integrat en WordPress es pot utilitzar com a tema de partida.

Alguns temes *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.getskelton.com/>).

5.4.1. Derivació d'un tema usant Toolbox com a base

Com a exemple crearem un tema utilitzant Toolbox com a tema base. Toolbox és un *framework* per a crear temes construït en HTML5 i creat per la mateixa empresa desenvolupadora de WordPress, Automattic.

Passos que seguirem:

- Crear un tema fill editant l'“`style.css`”.

Pàgina web

Per a més informació, es pot consultar el Codex a: http://codex.wordpress.org/The_Loop.

Pàgina web

Es pot veure més informació sobre Skeleton en la pàgina <http://demos.simplethemes.com/skeleton/>.

Pàgina web

Més informació sobre temes *framework* en el Codex (http://codex.wordpress.org/Theme_Frameworks#Base.2FStarter_Theme).

- Activar el tema.
- Podem canviar el *layout*?
- Quantes zones tenim per a arrossegar ginys? On estan definides aquestes zones?
- Crear en el CSS les tipografies més importants.
- Imatge per a la capçalera,
- Personalitzar el peu de pàgina.
- Una *home* especial?

Instal·lació del tema:

Descarregarem els arxius i l'instal·larem de la manera habitual.

Crear un tema fill i activar-lo:

Per a això cal crear una carpeta buida (amb el nom que vulguem) dins de *Themes* i crear un arxiu *style.css* amb un contingut similar a aquest:

```
/*
Theme Name: El meu tema derivat de Toolbox
Theme URI: http://www.exemple.com
Description: Provant com derivar un tema de toolbox
Author: dani
Version: 1
Tags: toolbox,simple
Template: toolbox
*/
/* importem el css del tema base */
@import url('../toolbox/style.css');
```

Observació

Encara que en CSS els comentaris s'ignoren, el WordPress els llegeix i extreu informació sobre el tema. Pel que fa al CSS, l'única cosa que importa és la instrucció *import*.

Exploreu amb Firebug, o l'inspector d'HTML/CSS del navegador que useu habitualment, els elements "div" que té el tema. Observeu els ID i classes que ens permetran adaptar el nostre disseny a la plantilla.

Afegir la possibilitat de tenir un *background* editable :

Per a això hem de crear el nostre arxiu `functions.php` i escriure dins:

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

Observació

En diversos temes de WordPress aquesta opció ja està activa, però no en el cas de Toolbox.

Obriu l'administrador de WordPress i observeu com ha aparegut una nova opció dins d'Aparença que permet canviar el fons. Canvieu-lo i observeu el resultat en el blog.

Crear un color de fons per als “div” principals (contingut, *sidebar*), ajustar marges i *padding*s. Per a això hem de modificar el nostre `style.css`. També podem canviar el color la tipografia, la font, la mida, etc.

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

Canviar l'estil del títol del blog (h1) i del subtítol (h2):

```
h1{
..
}
```

Canviar l'estil de l'opció de menú seleccionada:

Descobriu amb Firebug quina classe té l'opció de menú seleccionada i editeu el CSS per a remarcar-ho d'alguna manera.

Crear una plantilla especial per a les pàgines (`page.php`) i per als articles (`single.php`).

Crear una *page template* (plantilla especial per a pàgines).

Idees:

```
/**
Page Template: El meu nom
*/
```

Una plantilla especial per a la pàgina de contacte, afegint un formulari de docs.google.com o similar, una plantilla per al *sitemap* del lloc, afegint funcions del Codex de WordPress per a fer llistes de pàgines i entrades, etc.

Crear un *footer* i fer-lo dinàmic :

Per a això necessitem, d'una banda, afegir el següent codi a `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>',
) );
```

I després en la nostra plantilla del *footer*:

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

Visualitzar les imatges destacades dels articles (si les tenen; si no, mostrar-ne una per defecte).

A `functions.php`:

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

Una vegada activat en les nostres plantilles podem usar:

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

etc.

5.5. Exercici pràctic. Creació d'una plantilla per a *portfolios*

En aquest apartat ens plantejarem un exercici que té com a objectiu la personalització d'un tema per a crear un *portfolio* que només mostri els continguts que pertanyen a la categoria "Portfolio" i les imatges destacades associades.

Instruccions pas a pas:

- Partiu d'un tema derivat de Toolbox ("Child" o, si es vol, un altre tema base diferent). Per a això, consulteu les instruccions de l'apartat anterior.
- Activeu l'opció d'imatges destacades (o "Post thumbnails") a `functions.php`:

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

Aquesta instrucció afegeix l'opció de vincular una imatge destacada per a cada entrada.

- Creeu diverses entrades de prova (com a mínim cinc), totes amb la categoria "Portfolio" i amb "Imatge destacada".
- Creeu una plantilla dins del nostre tema derivat anomenada "category-portfolio.php". D'aquesta manera el WordPress utilitzarà aquesta plantilla quan entrem a `/category/portfolio` i ens mostrarà una llista de les entrades corresponents.
- El contingut inicial d'aquesta plantilla, molt simplificat, pot ser el següent:

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

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

<!-- contingut d'una entrada del portfolio -->
<a href="<?php the_permalink() ?>">
<?php the_post_thumbnail('thumbnail'); ?></a>
<!-- fi del contingut d'una entrada -->

<?php endwhile; ?>
<?php endif; ?>

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

Lectura recomanada

Trobareu més informació sobre imatges destacades en l'article següent:

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/>.

Observació

També podem definir post types, és a dir, nous tipus de contingut. En el nostre cas, una possible solució seria crear un tipus de contingut "Portfolio" en lloc d'utilitzar una categoria en el tipus de contingut "Entrada" preexistent.

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

“the_post_thumbnail(“thumbnail”)” escriu en la plantilla la imatge destacada en format petit.

- També volem canviar l'aspecte de la **pàgina única** de les entrades del “Portfolio”, que correspon a la plantilla “single.php”. Podríem partir del “single.php” del tema pare, i afegir almenys aquesta línia, que és la que mostra la imatge en format gran:

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

No obstant això aquesta solució **no ens permet tenir una plantilla independent** per a les pàgines úniques del “Portfolio”. Si volem tenir una plantilla independent, com que això no està previst en el sistema de plantilles de WordPress, haurem de “hackejar” una mica. Per a això una solució seria editar la plantilla “single” i afegir al principi

```
if ( in_category( 'portfolio' ) ) { //si el post és d'aquesta categoria
    get_template_part('single-portfolio');
    //utilitza una altra plantilla

}else{

.. resta de la plantilla single.php
```

i al final

```
<?php
} //tancar l'if
?>
```

Aquesta plantilla “single” decideix utilitzar una altra plantilla diferent anomenada “single-portfolio.php” en funció de si el contingut té la categoria “Portfolio”.

Seguint amb l'exercici, en la nostra plantilla especial per al “Portfolio”, “single-portfolio.php”, tindríem el mateix que a “single.php” però sense la condició inicial i, a més, podem afegir:

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

que mostrarà la imatge destacada en un mida gran. A més, en aquesta plantilla podem afegir qualsevol altre canvi que vulguem en l'estructura. Tot el que posem en aquesta plantilla només afectarà la llista d'entrades de la categoria "Portfolio" (en la ruta *categories/portfolio*).

En aquest punt pot ser que ens preguntem si és possible tenir diverses imatges en la mateixa entrada. Les galeries de WordPress sí permeten associar diverses imatges a una entrada, però això no té res a veure amb les imatges destacades (*thumbnails*). Només és possible associar un *thumbnail* per a cadascuna de les entrades.

A continuació veurem un exemple de personalització de la plantilla "category-portfolio.php":

- Per a començar i poder fer proves necessitarem afegir continguts (almenys deu entrades de la categoria).
- Si volem afegir l'opció de paginació a "category-portfolio.php", hem d'inserir al final o on volem que aparegui:

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

- Afegirem un contenidor exterior a les fotos (*thumbnails* quadrats) escrivint els "div" en la plantilla (potser un "div" per a la imatge i un altre per al títol) i modificarem el CSS perquè es mostri com una reixeta de 3x3.
- Modificarem el nombre de *posts* a nou en comptes de deu (a Ajustaments/Lectura).
- Si volem afegir un "div" amb el títol sobre les imatges:

```
the_title()
```

- Crearem un contenidor per als *thumbnails* i un altre per als títols, modificarem el CSS per a modificar l'aspecte.
- Fins i tot podríem afegir un *script* jQuery per a mostrar els títols en fer *rollover*:

```
<script>
var $j = jQuery.noConflict();
```

Material

Trobareu els arxius del tema una vegada acabats a:
http://www.kiwoo.org/recursos/ejemplos/drupal/tema_portfolio.zip

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

5.6. Llista de control per a crear un tema de WordPress

La creació d'un tema en WordPress (o en altres CMS) requereix prendre decisions per a optimitzar el desenvolupament. La solució òptima pot variar segons les especificacions del projecte. En aquest apartat veurem una llista de control (*checklist*) que permet escollir la millor opció segons aquestes especificacions.

Com hem vist, bàsicament tenim tres maneres de crear un tema:

- Crear un tema *child* de qualsevol altre ja existent que s'adapti al nostre disseny i modificar (només) el CSS.
- Crear un tema *child* d'un tema *framework*, com pot ser Toolbox, i modificar el CSS i les plantilles que necessitem redefinir.
- Crear un tema des de zero. No derivar el tema de cap altre i crear totes les plantilles des de zero (per a experts).

El nostre disseny defineix una sèrie de pantalles amb un *layout* i continguts determinats. Primer és necessari tenir clara l'estructura de continguts (tipus de contingut, etiquetes, categories).

Les preguntes que ens hem de fer a partir d'aquest moment són:

Hi ha algun tema disponible amb el mateix *layout* que el nostre disseny?

- Sí: en aquest cas, s'ha de crear un tema *child* i personalitzar el CSS com vulguem. **Fi**.
- No: en cas contrari, s'ha de passar al punt següent.

Hi ha algun *framework* que es pugui adaptar a l'estructura del nostre disseny? Un *framework* és un tema que només té *layout* però no té presentació: està en blanc. Cadascun defineix regions dinàmiques diferents (per a arrossegar ginys) i a més alguns permeten configurar opcions de la presentació directament des del panell de WordPress.

- Sí: s'ha de crear un tema *child* i personalitzar el CSS. **Fi.**
- No: s'ha de passar al punt següent.

És possible aconseguir el *layout* creant algunes plantilles especials? Una plantilla especial ens permet afegir continguts que no teníem i que necessitem, i afegir “div” o elements HTML que ens permetin modificar l'aspecte mitjançant CSS. Per tenir una plantilla especial vam crear un arxiu amb qualsevol nom i vam afegir un comentari dins del PHP:

```
<?php
/*
Template Name: La meva plantilla
*/
?>
.. resta de codi
```

Dins de la plantilla podem posar el que vulguem a part del *loop*.

En tenim prou amb la utilització de plantilles personalitzades?

- Sí: s'han de crear les plantilles especials que vulguem. **Fi.**
- No: s'ha de passar al punt següent.

És possible aconseguir el *layout* que volem modificant algunes plantilles de la jerarquia? El mecanisme de plantilles de WordPress defineix una cadena de decisions a partir de la qual s'escull la plantilla necessària en cada moment. D'aquesta manera podem definir una plantilla especial per a una pàgina en concret “page-\$id.php”, per a la llista d'una etiqueta en concret “tag-\$slug.php”, i per a molts altres casos. Aquestes plantilles especials estan documentades en el Codex, per exemple, les plantilles de categoria *Template*. Les plantilles de la jerarquia s'utilitzen automàticament si n'hi ha.

- Sí: s'han de crear les plantilles necessàries. **Fi.**
- No: s'ha de passar al punt següent.

Atenció: si existeixen en la plantilla mare, haurem de copiar primer aquesta plantilla en el nostre subtema.

Podem afegir funcions especials a “functions.php”, o codi especial dins d'alguna plantilla per a solucionar-ho? O fins i tot algun *plug-in*? Algunes crides a l'API de WordPress ens permeten afegir funcionalitats com zones actives, activació de funcionalitats de WordPress i funcions pròpies per al que vulguem (es tracta de desenvolupament més avançat: aquí sí necessitarem PHP per a això).

- Sí: s'ha de personalitzar el codi de “functions.php” o d'alguna de les plantilles. **Fi.**
- No: s'ha de passar al punt següent.

Com a opció final, la que ens dóna més flexibilitat, podem crear un tema des de zero.

Requisits per ser un bon *themer*

- Tenir coneixements d'HTML. En les plantilles pròpies haurem d'afegir l'HTML que considerem convenient.
- Tenir coneixements de CSS. És l'eina principal que usarem per a modificar l'aspecte de la nostra plantilla.
- Conèixer prou el PHP, el necessari almenys per a poder utilitzar les variables i les funcions de WordPress.

5.6.1. Alguns problemes típics i la seva solució

La plantilla personalitzada no funciona. S'ha de verificar que el contingut està usant realment aquesta plantilla, per exemple afegint un missatge de text a dins com:

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

Si no apareix aquest missatge en el navegador és perquè el WordPress no està usant la plantilla que suposem. Sempre cal tenir en compte la jerarquia de les plantilles. En ser un mecanisme automàtic, pot crear confusions.

Error de PHP. S'ha de veure en quin arxiu i en quina línia específica hi ha el missatge d'error. Pot ser perquè els delimitadors de PHP no estan ben tancats (per exemple, obrir `<?php` sense tancar-lo amb `?>` donarà un error). O pot ser que hi hagi una condició *if* sense tancar, o que hi hagi una funció no definida, per esmentar només alguns exemples.

Error 500. Molt sovint és degut a errors en el servidor causats pel `.htaccess` (arxiu que s'encarrega de reescriure els URL "nets"), entre altres coses.

Pàgina web

Podeu trobar més informació sobre la jerarquia de les plantilles de WordPress en el Codex a: http://codex.wordpress.org/Template_Hierarchy.

5.7. Adaptar un tema de WordPress per a diferents dispositius

En el panorama actual es pot dir que s'usen bàsicament **tres tipus de dispositius**:

- **Navegador web** d'escriptori. La resolució sol ser alta o molt alta i s'interactua amb un ratolí o un *touchpad*.
- **Tauleta**. Aquí el condicionant, més que la resolució, és que les pantalles són tàctils i, per tant, les zones sensibles han d'estar pensades per a això.
- **Mòbils** (telèfons intel·ligents o *smartphones*). Amb pantalles molt més petites.

Les possibles solucions per a tractar amb un disseny multidispositiu són:

- **Crear un tema especial per a mòbils i un altre per a l'escriptori i/o tauleta**. Per a això hem d'activar un *plug-in* que detecti el dispositiu i ens permeti seleccionar un tema per al nostre web en funció d'això.
- Utilitzar un tema **responsive** que s'adapti a la mida de la pantalla.

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

Mobble (<https://wordpress.org/extend/plugins/mobble/faq/>) és un *plug-in* interessant que ens pot ajudar en la tasca d'adaptar el disseny a diferents dispositius. Una vegada activat ens permetrà utilitzar instruccions especials en les nostres plantilles:

```
<?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)
?>
```

Exemple:

```
<?php
if (is_mobile()) {
    //no incloem el sidebar
}else{
    if(is_tablet()){
        get_sidebar("tablet");
        //sidebar especial per a tauleta
    }else{
        get_sidebar();
        //sidebar per a navegador d'escriptori
    }
}
?>
```

Com verificar el nostre disseny en diferents dispositius:

Idealment els hauríem de provar en tots els dispositius reals, però atès que això no és possible tenim els recursos següents:

- Utilitzar un navegador normal i variar la mida de la finestra (millor si utilitzem Chrome o Safari, que utilitzen Webkit igual que els navegadors per defecte d'iPhone i Android).
- Utilitzar un simulador de navegador mòbil, com el d'Opera, Opera Mobile Emulator (<http://www.opera.com/developer/tools/mobile/>).
- Utilitzar un emulador del *kit* de desenvolupament d'iPhone, Android, etc.
- Utilitzar un emulador en línia.

Consell

Si desenvolupem en local, podem usar el nostre mòbil o tauleta connectant-nos per la xarxa Wi-Fi i fer tests utilitzant la IP del servidor.

5.8. Shortcodes

A partir de la versió 2.5, WordPress introdueix una API per a *shortcodes*.

Un *shortcode* és un codi entre claudàtors escrit directament en el text d'una entrada. És un mecanisme senzill que permet que el codi d'un *plug-in*, o directament "functions.php", tradueixi el *shortcode* per un altre contingut determinat pel contingut de l'entrada o pels paràmetres.

Per exemple, en introduir una galeria de fotos dins d'una entrada, en realitat el WordPress escriu un *shortcode* "[gallery]" que posteriorment és traduït a l'HTML necessari per a mostrar la galeria.

L'API de *shortcodes* permet als desenvolupadors de *plug-ins* i temes crear un codi perquè els usuaris puguin utilitzar determinats *shortcodes* en les seves entrades.

Els *shortcodes* poden tenir múltiples atributs, per exemple:

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

El desenvolupament del codi necessari per a implementar un *shortcode* no és complicat. Vegem-ne un exemple senzill.

En l'arxiu `functions.php` del nostre tema introduïm el codi següent per definir i registrar una entrada de *shortcode*:

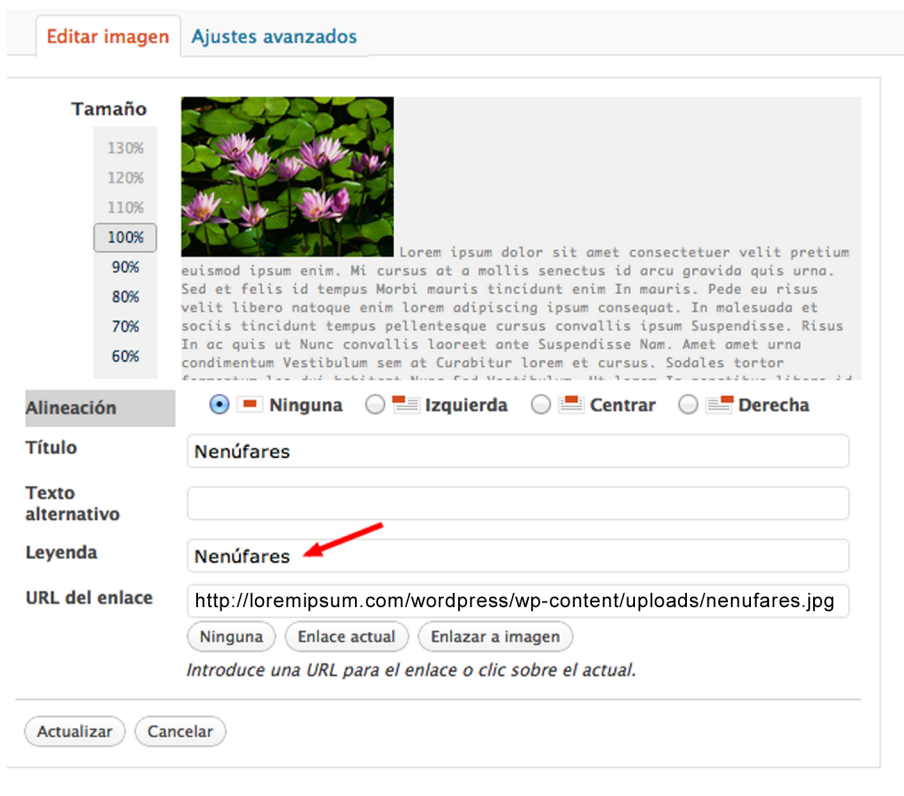
```
//[exemple]
function exemple_func( $atts ){
    return "provant shortcodes";
}
add_shortcode( 'test', 'exemple_func' );
```

En aquest cas, el codi “[test]” es convertiria en el text “provant shortcodes” quan veiem l'entrada o pàgina resultant.

El desenvolupament d'un *shortcode* pot ser útil en casos en què no necessitem la complexitat que implica el desenvolupament complet d'un *plug-in*.

5.8.1. Funcionament del *shortcode* “Caption”

Un dels *shortcodes* integrats en WordPress és el “Caption” de les imatges.



The screenshot shows the WordPress image editor interface. At the top, there are two tabs: "Editar imagen" (selected) and "Ajustes avanzados". Below the tabs, there is a preview area showing a photo of purple flowers (Nenúfares) with a text box containing Lorem Ipsum placeholder text. To the left of the preview, there is a "Tamaño" (Size) dropdown menu with options from 60% to 130%. Below the preview, there are several settings sections: "Alineación" (Alignment) with radio buttons for "Ninguna", "Izquierda", "Centrar", and "Derecha"; "Título" (Title) with a text input field containing "Nenúfares"; "Texto alternativo" (Alt text) with an empty text input field; "Leyenda" (Caption) with a text input field containing "Nenúfares" and a red arrow pointing to it; and "URL del enlace" (Link URL) with a text input field containing "http://loremipsum.com/wordpress/wp-content/uploads/nenufares.jpg" and three buttons: "Ninguna", "Enlace actual", and "Enlazar a imagen". At the bottom, there are two buttons: "Actualizar" (Update) and "Cancelar" (Cancel).

El camp “Leyenda” de la imatge es tradueix com a *shortcode* en WordPress.

En inserir una imatge, a part dels camps “Títol” (“Title”) i “Text alternatiu” (“Alt”), disposem del camp “Llegenda”. Si utilitzem aquest camp, la llegenda apareixerà en un marc que envolta la foto.

El WordPress implementa aquesta funcionalitat utilitzant un *shortcode* especial anomenat “Caption”. De fet, si analitzem el codi HTML generat per WordPress ens trobarem amb això:

```
[caption id="attachment_60" align="alignnone" width="300" caption="Nenúfars"]
<a href="http://exemple.com /wordpress/
wp-content/uploads/Nenúfars1.jpg"></a>[/caption]
```

Es tracta d'un *shortcode* amb diversos paràmetres. El WordPress tradueix aquest *shortcode* i els seus paràmetres en l'HTML necessari per a mostrar un marc gris al voltant de la foto.

En el cas anterior el resultat en l'entrada publicada és:

```
<div id="attachment_60" class="wp-caption alignnone" style="width: 310px">
<a href="http://exemple.com //wordpress/
wp-content/uploads/nenúfars1.jpg"></a>
<p class="wp-caption-text">Nenúfars</p></div>
```

5.9. Desenvolupament d'un *plug-in* en WordPress

Com ja hem explicat, un *plug-in* és un programa que permet afegir funcionalitats al WordPress. Aquests programes es desenvolupen utilitzant l'API de desenvolupament de *plug-ins*.

El desenvolupament d'un *plug-in* pot ser tan complex com vulguem. Òbviament, abans de desenvolupar un *plug-in* ens hem de preguntar si ja n'hi ha algun de similar entre els milers que ja estan disponibles, i en aquest cas l'hem d'utilitzar o fins i tot, si la llicència ho permet, l'hem de modificar o crear una nova versió.

L'API de *plug-ins* de WordPress està basat en dos tipus de *hooks* :

- **Accions.** El nostre *plug-in* es pot activar en determinats “esdeveniments” de WordPress. Per exemple, en el moment de publicar-se una nova entra-

Pàgina web

Podeu trobar més informació sobre *shortcodes* en el Codex a: http://codex.wordpress.org/Shortcode_API.

Pàgina web

Podeu trobar més informació sobre l'API de desenvolupament de *plug-ins* en el Codex a: http://codex.wordpress.org/Plugin_API.

da, en canviar de tema, en afegir un nou comentari, etc. En aquest moment podem capturar l'“esdeveniment” i modificar el contingut, enviar un correu electrònic o qualsevol altra acció que vulguem fer.

- **Filtres.** Es criden abans de desar informació en la base de dades o en mostrar-se en el navegador.

A nivell bàsic podem desenvolupar un *plug-in* amb accions i filtres, encara que probablement necessitem altres coses, com accés a la base de dades, modificar paràmetres per mitjà de l'escriptori de WordPress, etc. Tanmateix, això queda fora del domini d'aquests apunts i només veurem les bases.

5.9.1. Accions

Hi ha centenars d'accions, totes perfectament documentades en el Codex (http://codex.wordpress.org/Plugin_API/Action_Reference).

Ho il·lustrarem amb un petit exemple consistent en *un plug-in* que envia un correu electrònic cada vegada que es publica una nova entrada.

Per a això hem d'utilitzar l'acció “publish_post”, que es crida cada vegada que es dispara aquest esdeveniment. Utilitzant “add_action” podem vincular aquesta acció amb una funció definida pel nostre *plug-in*.

Per exemple:

```
add_action('publish_post', enviar_email);  
Per descomptat faltaria la definició de la nostra funció enviar_email  
function enviar_email($post_id){  
    mail("info@exemple.com", "blog actualitzat", "Hi ha una nova entrada en el blog");  
}
```

Fixeu-vos que l'acció envia com a paràmetre l'ID del *post*. Això pot ser útil si volem recuperar el contingut per a qualsevol propòsit.

Utilitzar directament una funció no és una bona idea perquè és fàcil que hi hagi conflictes amb altres funcions amb el mateix nom. Per a solucionar això podem encapsular la funció com a mètode d'una classe i cridar-lo de la manera següent:

```
class emailer {  
    function enviar_email($post_ID) {  
        mail("info@exemple.com", "blog actualitzat", "Hi ha una nova entrada en el blog");  
        return $post_ID;  
    }  
}
```

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

5.9.2. Filtres

Els filtres són funcions a les quals el WordPress passa dades just abans de fer alguna cosa amb aquestes dades, com per exemple desar-les en la base de dades o mostrar-les en el navegador. També hi ha centenars de filtres i estan documentats en el Codex (http://codex.wordpress.org/Plugin_API/Filter_Reference).

Un filtre rep com a entrada dades sense modificar i retorna les dades modificades. En general la crida és:

```
add_filter ( 'nom_del_hook', 'nom_del_filtre', [prioritat], [altres_paràmetres] );
```

Així, per exemple,

```
<?php  
add_filter( 'the_content', 'modificar_contingut' );  
function modificar_contingut($contingut){  
    return '*' . $contingut . '*';  
}  
?>
```

afegirà un asterisc abans i després de tots els continguts.

El codi del nostre *plug-in* es podria afegir directament al `functions.php` del nostre tema, però normalment l'encapsulem en un directori a part. Fer això és relativament senzill. En primer lloc, hem de triar un nom que no es presti a confusió amb altres *plug-ins* i usar-lo com a nom del directori, en el qual haurem de tenir almenys dos arxius.

L'arxiu principal del *plug-in* (amb el nom que vulguem, però procurant que sigui el més descriptiu possible) conté el codi i un comentari al principi que proporciona informació al WordPress:

```
/*  
Plugin Name: Test per a les anotacions  
Plugin URI: http://l'_url_del_plug-in_amb_actualitzacions  
Description: Descripció del plug-in  
Version: La versió del plug-in  
Author: L'autor  
Author URI: http://uri_Of_The_Plugin_Author  
License: GPL2
```

```
*/
```

Part d'aquesta informació és la que l'usuari veu en el panell de WordPress. Per exemple, en el cas del *plug-in* Akismet, la informació del *plug-in* és la següent:

```
/*
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 volem pujar el *plug-in* a la comunitat WordPress és necessari afegir un arxiu `readme.txt` seguint les indicacions del Codex (http://codex.wordpress.org/writing_a_Plugin#Readme_File).

Per descomptat, com més complex sigui el *plug-in*, més arxius podem necessitar. A més de filtres i accions, un *plug-in* pot utilitzar qualsevol crida a l'API de WordPress o utilitzar codi PHP d'altres biblioteques, per exemple. Les possibilitats són il·limitades.

Recomanació

Una bona font d'inspiració per a crear un *plug-in* particular pot ser l'anàlisi d'altres *plug-ins* que ja existeixen.

Una de les moltes opcions possibles és desenvolupar un *plug-in* que serveixi per a activar un nou giny personalitzat .

Pàgina web

Podeu trobar més informació sobre desenvolupament de *plug-ins* en el Codex (http://codex.wordpress.org/Writing_a_Plugin).

Lectura recomanada

En l'article següent podem veure un exemple fet pas a pas:

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

