

Drupal

Daniel Julià Lundgren

PID_00168318



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	6
2.1. Cambiar el idioma	9
2.2. Estructura de contenidos de Drupal	9
2.3. Tipos de contenidos y taxonomías	11
3. Apartados del panel de administración	12
3.1. Contenido (<i>content</i>)	12
3.2. Estructura (<i>structure</i>)	13
3.3. Apariencia (<i>appearance</i>)	15
3.4. Personas (<i>people</i>)	15
3.5. Módulos (<i>modules</i>)	17
3.6. Configuración (<i>configuration</i>)	18
3.7. Informes (<i>reports</i>)	19
3.8. Ayuda (<i>help</i>)	20
3.9. Definir tipos de contenido	20
3.10. Definir taxonomías	22
4. Ejemplo práctico. Implementación de una publicación electrónica en Drupal	25
4.1. <i>Briefing</i> del proyecto	25
4.2. Creación de la estructura de contenidos	25
5. Instalar módulos	32
6. Tabla de correspondencias entre Drupal y WordPress	34
7. Integración con redes sociales	36
8. Creación de temas (<i>themes</i>) en Drupal	39
8.1. Modificar un tema ya existente	39
8.2. Utilizar un tema <i>framework</i> como base	40
8.3. Crear un tema desde cero	41
8.3.1. Creando un tema “minimalista”	41
8.3.2. Creando un tema más completo	45
8.3.3. Ejercicio. Creación de un tema desde cero a partir de una plantilla estática (HTML + CSS)	46

1. Introducción

Drupal es uno de los gestores de contenidos más utilizados debido a su flexibilidad para adaptarse a diferentes tipos de proyectos (sitios corporativos, tiendas *on-line*, publicaciones electrónicas, comunidades, etc.).

El hecho de que Drupal sea un programa de código abierto y muy utilizado ofrece una gran ventaja: **una comunidad muy grande de desarrolladores** que nos puede solucionar muchas dudas y nos garantiza el mantenimiento del proyecto. En la página oficial de Drupal tenemos acceso a esta comunidad, además de a los archivos, módulos y temas que podemos descargar fácilmente.

Curiosidad

Existen miles de sitios web construidos sobre Drupal, algunos de ellos tan importantes como el de la Casa Blanca.

Sitio oficial [drupal.org](http://www.drupal.org).

Observación

Por supuesto, la página de Drupal también está hecha en Drupal. Casi siempre las páginas oficiales de cualquier CMS están construidas con el propio CMS: deben dar ejemplo.

El código fuente de Drupal se puede descargar en la página <http://www.drupal.org>. Los requerimientos son parecidos a otros CMS –básicamente PHP y MySQL–, aunque para saber las versiones mínimas de cada uno es necesario consultar en la página de Drupal las especificaciones mínimas.

En Drupal podemos ampliar las funcionalidades de la aplicación añadiendo **módulos** (que corresponden a lo que serían los *plugins* en WordPress). Cualquier persona de la comunidad puede desarrollar nuevos módulos para cubrir sus necesidades y lo normal es que luego los dejen a disposición de toda la comunidad para que los usen si los necesitan. La versión base de Drupal (que se denomina *core*) viene con algunos módulos básicos (muchos de ellos desactivados), pero para ampliar funcionalidades, según las necesidades que se tengan, muchas veces es necesario descargar otros módulos adicionales.

Ved también

Existen muchos puntos en común, conceptualmente, entre WordPress y Drupal. En el apartado “Tabla de correspondencias entre Drupal y WordPress” se sistematizan estas coincidencias.

2. Instalación

Para instalar Drupal necesitamos lo siguiente:

- Los archivos del *core* descargados de drupal.org.
- Un servidor con PHP+MySQL (local o remoto).
- Una base de datos MySQL y los datos de acceso (usuario y contraseña). Si se trata de un servidor remoto, acceso por FTP al servidor (y el correspondiente usuario y contraseña).
- Si queremos instalarlo en un idioma diferente al inglés, los archivos de traducción; por ejemplo, en castellano en: <http://localize.drupal.org/translate/languages/es>, o en catalán en: <http://localize.drupal.org/translate/languages/ca>.

Observación

Drupal también permite la instalación con SQLite, un sistema de base de datos que funciona como extensión de PHP. En ese caso, por tanto, no necesitaríamos servidor MySQL.

Pasos de la instalación:

- Descomprimir los archivos descargados de la web de Drupal. Podemos cambiar el nombre de la carpeta por el que queramos: si inicialmente se llama “drupal7.14” lo podemos cambiar por “revista”, por ejemplo.
- Copiar los archivos en el servidor (local o remoto) en el directorio adecuado (“www” o “htdocs”, dependiendo del servidor), de manera que podamos acceder a nuestro sitio, por ejemplo, en <http://localhost/revista> o <http://www.miservidoremoto.com/revista> (suponiendo que le hemos dado el nombre “revista” al directorio).
- Copiar y pegar el archivo “site/default/default.settings.php” como “settings.php”.
- Crear una base de datos vacía desde nuestro gestor de bases de datos (por ejemplo, “phpMyAdmin”, probablemente en la dirección <http://localhost/phpmyadmin> o donde sea apropiado en nuestro servidor local o remoto). Para nuestro ejemplo, imaginemos que hemos llamado a la base de datos también “revista”.
- Entrar mediante el navegador a la dirección <http://localhost/revista>.
- Seleccionar la opción “Install Drupal in English”¹.

⁽¹⁾Instrucciones para instalar Drupal en cualquier idioma inicialmente: <http://drupal.org/localize>.

- Aparecerá una pantalla donde se nos pedirá el nombre de la base de datos y el usuario y contraseña para acceder a ella. Rellenamos los datos y seguimos.
- Si hemos escrito los datos correctos, ahora nos pedirán los datos de la configuración del sitio y los datos de la cuenta del administrador de Drupal (para probar podemos poner, si estamos instalando en “localhost”, “admin/admin”, aunque eso, desde luego, no es nada seguro en un sitio real). Es importante apuntar ese usuario y contraseña, ya que los tendremos que usar para volver a entrar a Drupal posteriormente como administradores.
- Seguimos y si todo va bien tendremos el sitio funcionando.

Select an installation profile



► Choose profile

Choose language

Verify requirements

Set up database

Install profile

Configure site

Finished

Standard

Install with commonly used features pre-configured.

Minimal

Start with only a few modules enabled.

Save and continue

Installing Drupal



Completed 6 of 28. 21%
Installed *Text* module.

✓ Choose profile

✓ Choose language

✓ Verify requirements


✓ Set up database

► Install profile

Configure site

Finished

Configure site



- ✓ Choose profile
- ✓ Choose language
- ✓ Verify requirements
- ✓ Set up database
- ✓ Install profile
- ▶ **Configure site**

Finished

SITE INFORMATION

Site name *

Site e-mail address *

Automated e-mails, such as registration information, will be sent from this address. Use an address ending in your site's domain to help prevent these e-mails from being flagged as spam.

SITE MAINTENANCE ACCOUNT

Username *

Spaces are allowed; punctuation is not allowed except for periods, hyphens, and underscores.

E-mail address *

Password *

Password strength: █ **Weak**

Confirm password *

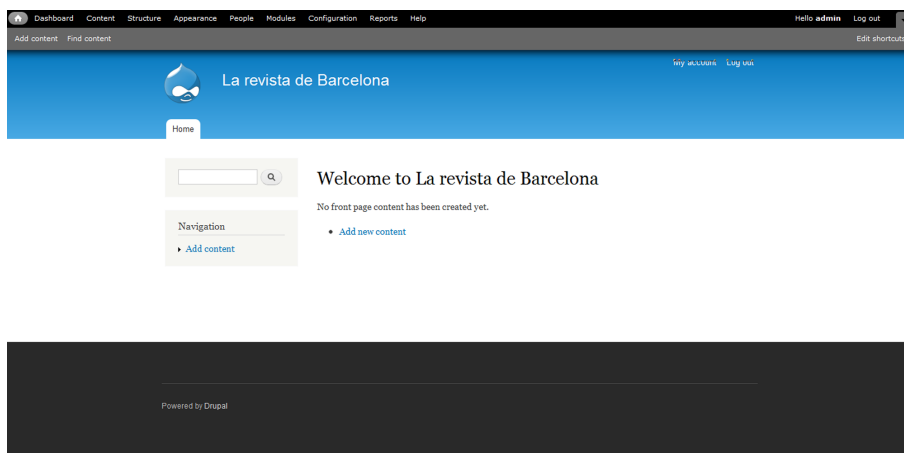
Passwords match: yes

To make your password stronger:

- Make it at least 6 characters
- Add uppercase letters
- Add numbers
- Add punctuation
- Make it different from your username

SERVER SETTINGS

Default country



The screenshot shows a Drupal website with a blue header. The site title is "La revista de Barcelona". The navigation menu includes "Home", "Add content", and "Find content". A search bar is present. The main content area displays "Welcome to La revista de Barcelona" and a message: "No front page content has been created yet." with a link to "Add new content". The footer indicates "Powered by Drupal".

Observación

Si hemos hecho la instalación en un servidor local, es posible que nos aparezca un mensaje (*warning*) en este punto avisando de que no se ha podido enviar un e-mail. No debemos preocuparnos: esto se debe simplemente al hecho de que muchos servidores locales no disponen de servidor de correo, pero esto no afecta al funcionamiento del resto de funcionalidades.

Una vez instalado todo correctamente, ya deberíamos poder entrar al sitio tanto en el rol de "Visitante anónimo" como en el de "Administrador". La dirección de nuestro sitio será `http://localhost/revista/` (o el nombre que le hayamos dado nosotros).

2.1. Cambiar el idioma

Si queremos cambiar la interfaz de Drupal a otro idioma *a posteriori*, siempre podemos instalar el módulo “Localization Update” y seguir los siguientes pasos:

- Activar el módulo “core locale” (este módulo se encarga de activar la funcionalidad multiidioma).
- Ir a la sección de configuración de idioma (en “admin/config/regional/language”) y seleccionar los idiomas que queramos.
- Bajar e instalar el módulo “localization update”.
- A partir de este momento los textos se actualizarán a todos los idiomas que tengamos activos.

Página web

Estos pasos se encuentran descritos en la página oficial de Drupal: <http://drupal.org/node/21145>.

2.2. Estructura de contenidos de Drupal

En Drupal cada unidad de contenido es un **nodo**.

Estos nodos pueden ser de diferentes tipos de contenido (tipos que podemos definir en el administrador). Al instalar Drupal tenemos dos definidos: “**Basic page**” (“Página”) y “**Article**” (similares a los tipos “Entry” y “Page” de WordPress).

Podemos crear un contenido nuevo en la opción “Agregar contenido” (“*Create content*”). Entonces deberemos elegir qué tipo de contenido de los dos existentes queremos crear. Probemos a crear contenidos de cualquiera de ellos.



Observación

Hay pocas diferencias entre los dos tipos de contenidos predefinidos, “Article” y “Basicpage”.

Al seleccionar “Agregar contenido” tenemos la opción de escoger entre los tipos que tengamos definidos.

Cualquier tipo de contenido tiene dos campos por defecto; *“Title”* (“Título”) y *“Body”* (“Texto”). El título es obligatorio, ya que es la manera de poder identificarlo; el campo *“Body”* es opcional.

Prestemos atención sobre todo a la opción que aparece debajo de esta pantalla, “Opciones de publicación” (“Publishing options”). Si la desplegamos veremos que existen tres opciones:

- “Publicado” (*“Published”*). Si desmarcamos esta opción el nodo se guardará pero no será visible para los visitantes de nuestro sitio.
- “Promovido en la portada” (*“Promoted to front page”*). Si marcamos esta opción el nodo aparecerá en la portada del sitio.
- “Fijo al comienzo de las listas” (*“Sticky at top of lists”*). En el caso de que este contenido aparezca en una lista, siempre será el primero de la lista.

Seleccionamos *“Promoted to front page”* y publicamos un contenido de prueba. Si vamos a la portada de nuestra web (por ejemplo, en localhost/revista), veremos que el nodo aparece en la portada. Si hacemos clic en el título, veremos que nos lleva a una página única con la dirección **“node/1”**.

En Drupal tenemos básicamente **dos tipos de páginas** (o **plantillas**):

- La portada o *“Home page”*, donde se muestran todos aquellos contenidos (nodos) promovidos en la portada.
- Las páginas únicas para cada uno de los contenidos (nodos), por defecto con la ruta **“node/xx”**, donde “xx” es el identificador del contenido.

Si tuviéramos muchos contenidos promovidos en la portada, aparecería una lista con todos ellos, paginada si fuera necesario (con las opciones anterior/siguiente), y haciendo clic en cualquiera de los títulos llegaríamos a las página del nodo correspondiente (por ejemplo, *“node/1”*, *“node/2”*, etc.).

Cuando un nodo aparece en una lista (por ejemplo en la portada, pero también en el listado de una categoría o etiqueta), se muestra solo un resumen del mismo: es lo que se denomina el modo resumido (*“teaser”*) del nodo.

Observación

Las rutas son por defecto del tipo *“node/1”* si se han activado las “Rutas limpias” (*“Clean urls”*). En caso contrario, todas las rutas son del tipo *“index.php?q=node/1”*. Para poder activar las “Rutas limpias” se debe tener activado el módulo **“mod_rewrite”** de Apache o del servidor web que se esté utilizando.

Además del concepto de nodo, Drupal introduce en la versión 7 el concepto de **entidades** (*entities*). Una entidad es una unidad de contenido más general: puede ser un nodo de información, pero también usuarios, comentarios o taxonomías. El hecho de tratar todos estos elementos de una manera común permite una gran flexibilidad al sistema.

Página web

Hay más información sobre las entidades en la página oficial de Drupal: <http://drupal.org/node/1261744>.

2.3. Tipos de contenidos y taxonomías

Antes de empezar a introducir nuestros contenidos, es necesario tener claro cuál va a ser la estructura de nuestro proyecto. En el CMS vamos a definir esta estructura. Aunque podemos cambiarla en el futuro, es mucho más difícil hacerlo después que desde el principio, por lo que es especialmente importante intentar anticipar todo lo que podamos.

Respecto a la estructura de contenidos, tenemos **dos aspectos principales** a definir:

- Los **tipos de contenidos**. Nos debemos plantear: ¿existen tipos de contenidos diferenciados? Por ejemplo, ¿necesitamos noticias y artículos de opinión? Debemos valorar la posibilidad de definir tipos diferentes en el momento en que se trate de contenidos que hay que mostrar de una manera totalmente diferente, o que tienen campos diferentes. Para diferenciar las noticias de diferentes categorías, en lugar de utilizar tipos de contenidos, usaremos categorías, ya que los campos y la manera de visualizar todas ellas son iguales. En general, definiremos pocos tipos de contenido y usaremos categorías y etiquetas para organizar el contenido de cada tipo.
- Las **taxonomías** (vocabularios de palabras clave que podemos asignar a los contenidos). Las taxonomías son muy útiles ya que permiten crear fácilmente categorías, además de poder etiquetar los contenidos libremente. Pueden funcionar de dos maneras: como categorías o como etiquetas. Una categoría es una lista “cerrada” de posibilidades, mientras que una etiqueta es libre y se pueden añadir nuevas en cualquier momento. A efectos internos, prácticamente no hay ninguna diferencia y Drupal las trata igual: la diferencia está más bien a nivel de usuario. Las categorías usan una lista desplegable y las etiquetas un campo de texto con “autocompletado”. Podemos separar las etiquetas mediante comas.

Para asociar las taxonomías a los tipos de contenido lo debemos hacer como “campos”. Cada tipo puede tener una o varias taxonomías asociadas. Más adelante veremos cómo se concreta esto en el panel de administración de Drupal.

3. Apartados del panel de administración

Para entrar en el panel de administración seleccionamos “*Administer*” en el menú de la izquierda o vamos directamente a “/revista/admin” (o el nombre que hayamos decidido dar a nuestro primer proyecto en Drupal).



Menú de administración de Drupal con sus opciones principales.

Veremos que hay nueve apartados principales:

- **Panel de control** (*Dashboard*). Nos da una visión global y podemos personalizarlo.
- **Contenido** (*Content*). Todo lo relacionado con los contenidos.
- **Estructura** (*Structure*). Todo lo relacionado con la estructura del sitio (bloques, tipos de contenido, menús y taxonomía),
- **Apariencia** (*Appearance*). Apariencia (básicamente, los *themes*).
- **Personas** (*People*). Gestión de los usuarios del sitio.
- **Módulos** (*Modules*). Lista de módulos disponibles.
- **Configuración** (*Configuration*). Configuración del sitio.
- **Informes** (*Reports*).
- **Ayuda** (*Help*).

3.1. Contenido (*content*)

Aquí tenemos todo lo relacionado con los contenidos.

Opción “Contenido” con sus dos apartados principales, “Contenido” y “Comentarios”.

Tenemos dos pestañas principales:

- **Contenido (*content*)**. Gestión de los contenidos (nodos). Contiene un listado con todos los contenidos que se pueden filtrar por tipo o por estatus (publicado, no publicado, etc.). Por supuesto, también nos permite añadir nuevos contenidos.
- **Comentarios (*comments*)**. Gestión de los comentarios que han escrito los visitantes del sitio. En este apartado puede que nos encontremos comentarios en estado de moderación, que deben ser aprobados por el administrador para ser visibles (de manera similar a WordPress).

3.2. Estructura (*structure*)

Subopciones del menú “Estructura” (la opción disparadores se activa con el módulo correspondiente).

a) Bloques (*blocks*)

Este apartado sirve para asignar bloques a regiones de la pantalla. Cada plantilla tiene unas regiones definidas. Podemos asignar los bloques a las regiones que queramos y ordenarlas (simplemente moviéndolas usando la cruz). Los bloques son preexistentes pero se pueden añadir nuevos bloques en la opción “Add block” de la pestaña superior. Es totalmente equivalente a la opción “Widget” de WordPress.

Ejercicio

Practicar en este apartado y asociar algunos de estos bloques a las zonas disponibles en el tema actual.

b) Menús

Aquí podemos configurar y crear menús (también son bloques, de manera que se deben asignar en el apartado “Blocks”). Existen cuatro menús predefinidos y “Navigation” es el menú del administrador. Cada menú contiene ítems que son realmente las opciones del menú. Estas opciones se pueden reordenar con *drag & drop*. Cada opción del menú es básicamente un texto y un enlace que puede ser interno (del tipo “node/1”) o externo (por ejemplo, <http://www.uoc.edu>).

c) Taxonomy

Las taxonomías son categorías y etiquetas que se pueden asignar a los tipos de contenidos.

d) Tipos de contenido (*content types*)

Muestra los tipos de contenido que tenemos definidos. Podemos añadir fácilmente nuevos tipos si los necesitamos. Tenemos dos opciones principales para cada tipo de contenido:

- “Gestionar campos” (“*manage fields*”), para modificar los campos asociados.
- “Gestionar presentación” (“*manage display*”) para modificar la manera en que se ven estos campos dependiendo del “modo de presentación”

Los nodos tienen básicamente dos modos de presentación: el modo “Resumen” (“*teaser*”) y el modo “Predeterminado”. En los listados, por ejemplo el de la portada, los nodos aparecen por defecto en modo “Resumen”, de manera que si el texto es muy largo se recortará automáticamente para mostrar solo un fragmento.

Ved también

Veremos este aspecto con mayor profundidad en el apartado “Definir taxonomías”.

La opción “Gestionar presentación” nos permite una gran flexibilidad, ya que podemos decidir si se visualizan (y cómo se visualizan) los campos de cada tipo de contenidos según estas dos situaciones.

Ejercicio

Publicar una página nueva de contenido (“*Create content*”) y añadir un texto muy largo, proveniente, por ejemplo, de la página de generación de *Lorem ipsum* (<http://www.lipsum.com/feed/html>).

Asegurarse de que el nodo se publica en la portada. Una vez publicado, ir a la portada (clicando en el logo de Drupal o en el tema que tengamos activo, por ejemplo). Ver cómo aparece el texto recortado automáticamente (nodo en formato “Teaser”) y además aparece un enlace “Read More”.

3.3. Apariencia (*appearance*)

Este es el apartado que hace referencia al aspecto del sitio. Como buen gestor de contenidos, el contenido y la presentación deberían estar siempre lo más separados posible. Esto se consigue gracias a los **temas** (*themes*). Así, en esta sección podemos activar cualquier tema y cambiar opciones sobre la presentación.

The screenshot shows the Drupal 'Apariencia' (Appearance) configuration page. At the top, there's a breadcrumb 'Inicio > Administración' and a sub-header 'Apariencia'. Below that, a description states: 'Seleccionar y configurar el tema predeterminado para su sitio web. Hay temas alternativos disponibles.' There's a '+ Instalar nuevo tema' link. The main section is titled 'TEMAS ACTIVADOS' and lists two themes:

- Bartik 7.14 (tema predeterminado)**: Un tema flexible con muchas regiones al que se le pueden cambiar los colores. Includes a link to 'Opciones'.
- Seven 7.14**: Un tema visual para administración de una columna, sin tablas, de anchura variable. Includes links for 'Opciones', 'Desactivar', and 'Establecer como predeterminado'.

Opción “Apariencia” con el listado de temas disponible.

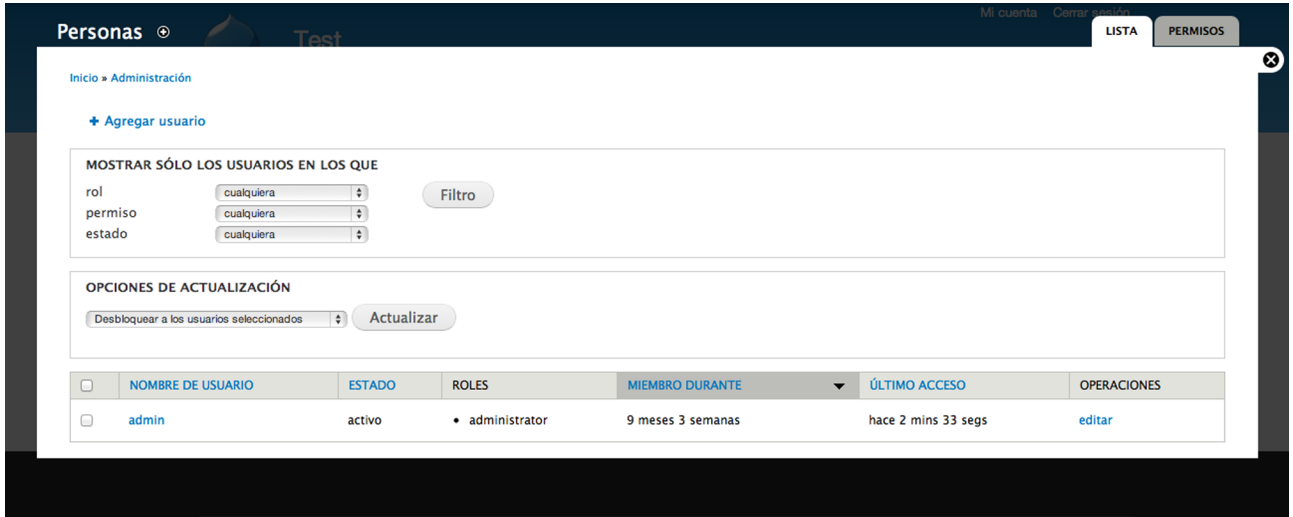
Ejercicio

Descargar y activar el siguiente tema: <http://drupal.org/project/journalcrunch>.

Comprobar el resultado en la portada y el resto de secciones.

3.4. Personas (*people*)

Drupal es bastante potente en la gestión de usuarios. En el apartado "Personas" podremos ver el listado de todos los usuarios, crear roles y permisos para cada uno de ellos, o definir opciones de configuración relativas a los usuarios como, por ejemplo, la manera de registrarse en el sitio.



Personas ⊕ Test Mi cuenta Cerrar sesión LISTA PERMISOS

Inicio • Administración

+ Agregar usuario

MOSTRAR SÓLO LOS USUARIOS EN LOS QUE

rol

permiso

estado

OPCIONES DE ACTUALIZACIÓN

<input type="checkbox"/>	NOMBRE DE USUARIO	ESTADO	ROLES	MIEMBRO DURANTE	ÚLTIMO ACCESO	OPERACIONES
<input type="checkbox"/>	admin	activo	• administrator	9 meses 3 semanas	hace 2 mins 33 segs	editar

Listado de usuarios del sitio web en el apartado "Personas".

Disponemos de 2 pestañas principales: Lista y Permisos.

- En **Lista** podemos ver la lista de todos los usuarios registrados en el sitio, que podemos filtrar por "rol", "permiso" o "estado". Además, podemos editar cada uno de ellos usando la opción "Edit".
- A su vez en la pestaña **Permisos** disponemos de 2 subpestañas: "Permisos" y "Roles".
 - En "Permisos" podemos activar o desactivar los permisos para cada uno de los roles definidos en nuestra aplicación.
 - En "Roles" vemos un listado de los roles disponibles. Por defecto existen solo 3 roles: usuario anónimo, usuario registrado y administrador. Podemos añadir más roles según nuestras necesidades (por ejemplo, editor, corrector, etc.), a los cuales deberemos activar los permisos correspondientes.

Personas ⊕ Test Mi cuenta Cerrar sesión LISTA PERMISOS

Inicio » Administración » Personas Permisos Roles

Los permisos le dan el control sobre lo que los usuarios pueden hacer y ver en su sitio. Usted puede definir un grupo específico de permisos para cada rol. (Vea la página de [Roles](#) para crear un rol). Dos roles importantes a considerar son usuarios registrados y administradores. Cualquier permiso otorgado al rol de usuarios registrados se dará a cualquier usuario que pueda iniciar sesión en su sitio. Puede convertir a cualquier rol en el rol administrador del sitio, lo que significa que todos los nuevos permisos se le asignarán automáticamente. Puede hacer esto en la página de [Opciones de Usuario](#). Debe tener cuidado y asegurarse que sólo los usuarios de confianza obtengan este acceso y este nivel de control del sitio.

Ocultar las descripciones

PERMISO	USUARIO ANÓNIMO	USUARIO REGISTRADO	ADMINISTRATOR
Block			
Administrar bloques	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Comment			
Administrar comentarios y opciones de comentarios	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Ver comentarios	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Publicar comentarios	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Omitir aprobación de comentario	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Editar comentarios propios	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Contextual links			
Usar enlaces contextuales	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>

Lista de permisos para cada uno de los roles.

3.5. Módulos (*modules*)

En este apartado se muestran todos los módulos (corresponde a lo que serían los *plugins* en WordPress), Podemos activar los módulos que queramos. En el listado podemos ver que muchos de los módulos no están activados por defecto.

Módulos ⊕ Test Mi cuenta Cerrar sesión LISTA ACTUALIZAR DESINSTALAR

Inicio » Administración

Descargár [módulos contribuidos](#) para ampliar el funcionamiento de Drupal.

Revise e instale regularmente [las actualizaciones disponibles](#) para mantener un sitio seguro y actualizado. Ejecute el [script de actualización](#) siempre que un módulo es actualizado.

+ Instalar nuevo módulo

▼ NÚCLEO

ACTIVADO	NOMBRE	VERSIÓN	DESCRIPCIÓN	OPERACIONES
<input type="checkbox"/>	Aggregator	7.14	Integra contenido sindicado (canales de noticias RSS, RDF y Atom).	
<input checked="" type="checkbox"/>	Block	7.14	Controla el montaje visual de los bloques con los que se construye una página. Los bloques son cajas de contenido que se representan en una zona o región de una página web. Necesitado por: Dashboard (activado)	Ayuda Permisos Configurar
<input type="checkbox"/>	Blog	7.14	Activa los blogs multi usuario.	
<input type="checkbox"/>	Book	7.14	Permite a los usuarios crear y organizar los contenidos relacionados en un índice.	
<input checked="" type="checkbox"/>	Color	7.14	Permite a los administradores cambiar el esquema de color de los temas compatibles con él..	Ayuda
<input checked="" type="checkbox"/>	Comment	7.14	Permite a los usuarios comentar y discutir contenido publicado. Necesita: Text (activado), Field (activado), Field SQL storage (activado) Necesitado por: Forum (desactivado), Tracker (desactivado)	Ayuda Permisos Configurar
<input type="checkbox"/>	Contact	7.14	Habilita el uso de formularios de contacto personal y a nivel sitio.	
<input type="checkbox"/>	Content translation	7.14	Permite que el contenido sea traducido a diferentes idiomas. Necesita: Locale (activado)	

Opción "Módulos" con el listado de módulos disponibles y los activos.

3.6. Configuración (*configuration*)

Aquí aparecen todo tipo de opciones relacionadas con la configuración del sistema. En general, son opciones que solo modificaremos durante el desarrollo del proyecto y que en principio no sería necesario editar una vez el sitio está ya publicado. En cualquier caso, estas son opciones a las cuales solo debería tener acceso el administrador del sitio.



Opciones de configuración.

Todas las opciones están organizadas por categorías.

Hay que destacar que al instalar cualquier módulo, es posible que en este apartado aparezcan otras opciones relacionadas con él.

Por defecto, podremos ver las siguientes categorías:

- **Personas.** Opciones relacionadas con los usuarios, por ejemplo, la manera de registrarse.
- **Autoría del contenido.** Formatos de texto. Drupal dispone de tres formatos de texto diferentes:
 - *Plain text.* Solo texto.
 - *Filtered HTML.* La opción por defecto, permite añadir algunas etiquetas básicas de HTML, como cabeceras (h1,h2), *strong*, etc. Podemos modificar estas etiquetas permitidas como queramos.
 - *Full HTML.* Cuando activamos esta opción al editar el texto se admite cualquier etiqueta de HTML.

- **Búsqueda y metadatos.** Opciones relativas a la búsqueda y cómo se muestran las URL de los contenidos
- **Medios.** Aquí podemos definir los tamaños predefinidos para las imágenes. Drupal construye por defecto tres tamaños diferentes para cada imagen que subamos al sistema. Podemos modificar la manera de crear estas imágenes o incluso crear nuevos tipos.
- **Regional e idioma.** Las opciones relativas al idioma, las traducciones y la manera de mostrar las fechas. Este apartado es muy importante en cualquier sitio multiidioma.
- **Sistema.** La información básica de nuestro sitio (nombre, lema, etc.). Número de contenidos publicados en la portada (por defecto son 10).
- **Interfaz de usuario.** En este apartado podemos definir atajos: accesos rápidos en el panel de administración.
- **Desarrollo.** Opciones para desarrolladores y para mejorar el rendimiento del sitio. Una vez publicado el proyecto, deberíamos activar la caché de páginas y bloques, y unificar en un solo archivo todos los documentos CSS y JavaScript de nuestro tema.
- **Servicios web.** Opciones relativas a la publicación automática de RSS.

VACIAR CACHÉ

Vaciar todas las cachés

CACHÉ

Caché de páginas para usuarios anónimos

Caché de bloques

Mínimo de permanencia en caché

<ninguno>

Las páginas en caché no serán recreadas hasta que no haya transcurrido al menos este periodo.

Caducidad de las páginas en caché

<ninguno>

El tiempo máximo que una caché externa puede utilizar una versión antigua de una página.

OPTIMIZACIÓN DE ANCHO DE BANDA

Los recursos externos pueden ser optimizados automáticamente, lo que reducirá tanto el tamaño como el número de peticiones a su sitio web.

Reunir y comprimir archivos CSS.

Juntar los archivos JavaScript.

Guardar configuración

Opciones del apartado "Desarrollo/Rendimiento".

3.7. Informes (reports)

Informes internos relacionados con el sitio como:

- Recibir un informe del estado funcional del sitio y de cualquier problema detectado.
- Actualizaciones disponibles para los módulos y temas instalados.
- Mensajes recientes del registro.
- Eventos registrados recientemente.
- Frases principales en las búsquedas.
- Información general de los campos en todos los tipos de contenido.
- Principales errores de "acceso denegado".
- Principales errores de "página no encontrada".

3.8. Ayuda (*help*)

Este apartado no es más que un listado de documentos de ayuda destinado a los usuarios de la herramienta de administración de Drupal.



Apartado "Ayuda".

3.9. Definir tipos de contenido

Una vez vistas las opciones principales del panel de administración, vamos a ver cómo configurar la estructura de contenidos de un proyecto básico.

Drupal dispone por defecto de dos tipos de contenido, “Página” y “Artículo”. Sin embargo, en muchos proyectos tendremos la necesidad de definir nuevos tipos.

Por ejemplo, si quisiéramos crear una tienda, probablemente crearíamos el tipo de contenido “Producto”, que tendría una serie de campos especiales (precio, referencia, etc.) o, en el caso de querer crear un *portfolio*, podríamos crear un tipo de contenido “Proyecto”, que tuviera como campos especiales varias imágenes y una categoría para cada proyecto, etc.

Definir tipos de contenido en Drupal es muy fácil. Solo hay que ir al apartado “Estructura/Tipos de contenido” (“*Structure/Types*”) y activar la opción “Añadir tipo de contenido” (“*Add content type*”).

Así, para añadir un tipo nuevo que se llame “Noticia”, en el campo “*Type*” deberemos escribir un código interno para este tipo (que podría ser también “Noticia”) y añadir una descripción de ayuda que aparecerá cuando alguien cree un contenido de este tipo.

Cada tipo de contenido tiene unas opciones predeterminadas. Estas opciones las podemos ver en la parte inferior de esta pantalla. Por ejemplo, podemos decidir si se publica por defecto o no en la portada (“*Promoted to front page*”) y podemos activar o desactivar los comentarios sobre este tipo de contenido (“*Comment settings*”). Los dos tipos de contenido predefinidos en Drupal tienen un comportamiento por defecto diferente. Los artículos se publican en la portada y admiten comentarios, mientras que las “Páginas” hacen lo contrario.

Una vez creado un nuevo tipo de contenido, este aparecerá en la lista de tipos de contenidos y podremos crear contenidos de este tipo en la opción “*Create content*”.

Los tipos de contenido individuales pueden tener diferentes campos, comportamientos y permisos asignados.

Nombre *

Producto Nombre de sistema: producto [Editar]

El nombre legible para humanos de este tipo de contenido. Este texto será mostrado como parte del listado en la página *Agregar nuevo contenido*. Es recomendable que este nombre comience con una letra mayúscula y contenga solamente letras, números y espacios. Este nombre debe ser único.

Descripción

Un producto de mi tienda

Describir este tipo de contenido. Este texto es lo que se muestra en la página *Añadir nuevo contenido*.

<p>Opciones del formulario de envío</p> <p>Nombre</p>	<p>Opciones predefinidas</p> <p><input type="checkbox"/> Publicado</p> <p><input checked="" type="checkbox"/> Colocado en la página principal</p> <p><input type="checkbox"/> Fijo al comienzo de las listas</p> <p><input type="checkbox"/> Crear nueva revisión</p> <p><small>Usuarios con el permiso de <i>Administrar contenido</i> podrán sobrescribir estas opciones.</small></p>
<p>Opciones de publicación</p> <p>No publicado, Colocado en la página principal</p>	<p>Funcionamiento multi idioma</p> <p><input checked="" type="radio"/> Desactivado</p> <p><input type="radio"/> Activado</p> <p><small>Active el funcionamiento multi idioma para este tipo de contenido. Si está habilitado, se agregará al formulario de edición un campo selector de idioma, lo que le permitirá elegir entre uno de los idiomas habilitados. Si está inhabilitado, los mensajes nuevos se salvan en el idioma predeterminado. El contenido existente no se verá afectado por cambios en esta opción.</small></p>
<p>Opciones de presentación</p> <p>No mostrar información del envío</p>	
<p>Opciones de comentarios</p> <p>Abrir, Hilos de comentarios, 50 comentarios por página</p>	
<p>Opciones del menú</p>	

Creando un nuevo tipo de contenido “producto” en “Estructura/Tipos de contenido”.

Ejercicio

Crear un tipo de contenido nuevo en el administrador y publicar un nodo de ejemplo de este tipo de contenido.

3.10. Definir taxonomías

El uso de **taxonomías** es una herramienta básica para poder estructurar los contenidos, ya que, entre otras cosas, permiten:

- Definir categorías en el contenido y, por tanto, definir la estructura relacionada.
- Etiquetar los contenidos como queramos.
- Listar (automáticamente) los contenidos a partir de estas categorías o etiquetas.

Vamos a imaginar un caso concreto: para el tipo de contenido “Noticia” anterior, queremos la posibilidad de diferenciar dos tipos de noticia: “Deportes” y “Política”. Para ello deberemos ir al apartado “Estructura/Taxonomía”, activar la opción “Agregar vocabulario” y dar un nombre a nuestro vocabulario (por ejemplo, “Tipo de noticia”) y una descripción si la creemos necesaria. En este momento ya tendremos un vocabulario, pero vacío. En cualquier momento podemos agregar términos a este vocabulario (en la opción “Agregar términos”).

Añadiremos los términos “Política” y “Deporte” a nuestro vocabulario.

Nombre *
 Nombre de sistema: tipo_de_noticia [\[Editar\]](#)

Descripción

Creación de un nuevo vocabulario “Tipo de noticia” en “Estructura/Taxonomía”.

Nombre *

Descripción
La categoría economía

Formato de texto [Más información sobre los formatos de texto](#)

- Las direcciones de las páginas web y las de correo se convierten en enlaces automáticamente.
- Etiquetas HTML permitidas: <a> <cite> <blockquote> <code> <dl> <dt> <dd>

- Saltos automáticos de líneas y de párrafos.

Alias de URL

Especifique de forma opcional una URL alternativa a mediante la que acceder a este término. Use una ruta relativa y no añada la barra final o el alias de URL no funcionará.

RELACIONES

Términos padres

Peso *

Los términos se muestran en orden ascendente por su peso.

Una vez creado el vocabulario podemos agregar términos.

Por último, para que este vocabulario esté vinculado con los contenidos debemos añadirlo como campo al tipo de contenido. Así, si queremos vincularlo al tipo de contenido “Noticias”, deberemos seguir los siguientes pasos:

- Ir a “Estructura/Tipos de contenido/Gestionar campos”.
- Agregar un nuevo campo con la etiqueta “Tipo de noticia” y nombre interno “field_tipo_noticia”.
- Como tipo de campo debemos seleccionar “Referencia de término” para vincular el campo con un vocabulario de la taxonomía. En “Control” podemos elegir cómo funcionará: en forma de lista desplegable, autocompletado (etiquetado libre) o como casillas de selección.
- Guardar.
- En el siguiente paso, seleccionaremos el vocabulario que queremos vincular. En nuestro ejemplo, “Tipo de noticia”.
- Finalmente podemos cambiar las opciones por defecto del campo, como la etiqueta que se mostrará al usuario, si es un campo obligatorio o no, o el número de valores (se podrían vincular varios valores simultáneamente si fuera necesario).

The screenshot shows the Drupal administration interface for configuring a content type field. The breadcrumb trail is: Inicio » Administración » Estructura » Tipos de contenido » Noticia. The main content area displays a table of fields for the 'Noticia' content type.

ETIQUETA	NOMBRE	CAMPO	CONTROL	OPERACIONES
+	Title	title	Elemento del módulo Nodo	
+	Body	body	Texto largo y resumen	editar eliminar
+	foto	field_foto	Imagen	editar eliminar
+	Lugar	field_lugar	Texto	editar eliminar

Below the table, there are two sections for adding new fields:

- Agregar nuevo campo:** Includes a text input for 'Tipo de noticia', a dropdown for 'Referencia de término', and a dropdown for 'Lista de selección'.
- Agregar campo existente:** Includes a dropdown for '- Seleccione un campo existente -' and a dropdown for '- Seleccione un control -'.

A 'Guardar' button is located at the bottom of the configuration area.

Vinculando el vocabulario "Tipo de noticia" al tipo de contenido "Noticia".

A partir de este momento, al añadir nuevos contenidos de tipo "Noticia" tendremos el "Tipo de noticia" como campo en forma de lista desplegable. De la misma manera, podemos añadir otros campos que podamos necesitar a este o a cualquier otro tipo de contenido.

Ejercicio

Crear el tipo de contenido "Noticia" y el vocabulario "Tipo de noticia". Añadir un contenido de prueba al menos para cada una de las categorías.

4. Ejemplo práctico. Implementación de una publicación electrónica en Drupal

Para poner en práctica los conocimientos básicos de Drupal, vamos a imaginar un ejercicio concreto. Se trata de la implementación de una publicación electrónica con determinados requerimientos. Partiremos de un *briefing* o descripción básica del objetivo de la publicación y a continuación veremos los pasos necesarios para construirla.

4.1. Briefing del proyecto

La revista electrónica será una publicación de actualidad muy simple con los requisitos que detallamos a continuación:

- La sección principal mostrará **noticias** que podrán ser, por ejemplo, de los siguientes tipos: deportes, política, economía, gente, tiempo e internacional.
- Queremos crear un apartado de **opinión** donde los editores puedan escribir artículos como si se tratara de un blog.
- También tendremos una sección que muestre solo **fotos** de actualidad.
- También necesitamos crear una página de **contacto** con una presentación de la revista.
- Tendremos siempre visible un **menú** con las secciones (categorías) además de los apartados de fotos, opinión y contacto.
- Todas las páginas tendrán la opción de **compartir** en redes sociales.

El objetivo será conseguir tener una versión básica de esta publicación funcionando y, además, crear un nuevo rol llamado “Editor” con derecho para publicar contenidos, pero no para cambiar opciones dentro del panel de administración.

4.2. Creación de la estructura de contenidos

A partir de los requisitos, deducimos que necesitaremos crear dos tipos de contenido, a parte de los ya existentes por defecto (entrada y página):

- Noticia.
- Opinión.

Observación

Asumimos que ya tenemos Drupal instalado tal como se ha explicado y que conocemos la dinámica de funcionamiento del administrador de Drupal.

En cuanto a contenido, las noticias tendrán un campo de foto y, además, las categorías que hemos decidido. El contenido del apartado “Opinión” también tendrá un campo de foto, pero no necesita categorías de momento.

1) Creación de un vocabulario “Categoría” para las noticias

Vamos a empezar creando las categorías. Para ello, iremos al apartado “Estructura/Taxonomía” y crearemos un vocabulario nuevo llamado “Categoría”. Una vez creado este vocabulario, añadiremos el listado de términos con deportes, política, etc.

En Drupal podemos crear tantos vocabularios como queramos y pueden funcionar como “tags” (etiquetas libres) o “categorías” (listado limitado de posibilidades).

2) Creación del tipo de contenido “Noticia”

A continuación vamos a crear el tipo de contenido “Noticia”, con dos campos especiales: uno será la categoría que hemos creado anteriormente y el otro un campo de imagen que llamaremos “Foto”. Para ello, vamos al apartado “Estructura/Tipos de contenido” y seleccionamos “Añadir tipo de contenido”. En este momento veremos toda una serie de propiedades que se pueden ajustar a este tipo de contenido por defecto. Podemos dejar las opciones tal y como están por defecto. Si es necesario, ya las cambiaremos después.

A continuación añadiremos los dos campos que necesitamos. En el apartado “Gestionar campos” añadiremos primero la categoría, dándole el nombre que queramos a este campo –por ejemplo, “Categoría” – y seleccionando sobre todo que se trata de un campo “Referencia de término”.

En el paso siguiente indicaremos que queremos utilizar el vocabulario “Categoría”. Seguidamente, indicaremos otros datos, como el texto de ayuda, si se trata o no de un campo obligatorio, el número de valores que puede tener, etc. En nuestro caso, también dejaremos todas las opciones en sus valores por defecto (en caso necesario lo podemos cambiar en otro momento).

En este momento nuestro tipo de contenido “Noticia” tiene tres campos:

- El título.
- El *body*.
- La categoría.

Finalmente, añadiremos otro campo de imagen que llamaremos “Foto”, también con todas las opciones por defecto.

En este momento ya tenemos creado nuestro tipo de contenido “Noticia”.

Ejercicio

Crear al menos dos contenidos de este tipo, añadiendo en cada caso una foto y una categoría. Ver el resultado en la portada de nuestro sitio web.

Nos falta un paso importante: decidir cómo se muestran los campos en las dos situaciones principales donde se pueden mostrar los contenidos. Para ello hay que ir al apartado “Gestionar presentación” de cada tipo de contenido.

Tenemos dos modos de presentación por defecto:

- Predeterminado: Corresponde a la página única del contenido.
- Resumen (*teaser*): Corresponde al modo resumido que aparece en los listados (por ejemplo en la portada, o en el listado de una categoría o *tag*).

Modificando estas opciones podemos decidir, entre otras cosas,

- El orden de los campos.
- Si aparece o no la etiqueta de cada campo.
- En qué tamaño aparece la imagen.
- Etc.

3) Creación del tipo de contenido “Opinión”

Seguiremos los mismos pasos que en el caso anterior para crear un tipo de contenido “Opinión”, pero en este caso sin añadirle una categoría, solo el campo foto.

4) Creación del tipo de contenido “Foto”

También crearemos un tipo de contenido “Foto”. En este tipo de contenido eliminaremos el campo de texto (*body*) y añadiremos un campo foto de imagen.

Llegados a este punto tendremos cinco tipos de contenido definidos:

- “Article”.
- “Basic Page”.
- “Noticia”.
- “Opinión”.
- “Foto”.

5) Creación de contenidos de prueba con el módulo Devel

Cuando estamos creando un sitio web muchas veces nos interesa tener contenidos de prueba. El módulo Devel nos permite crear contenidos ficticios para probar cómo aparecen según la estructura que hemos generado.

Instalaremos el módulo <http://drupal.org/project/devel> y activaremos “Devel generate”. Una vez activado este módulo, en el apartado “Configuración/Desarrollo de la opción” nos aparecerá “Crear contenido”.

Ejercicio

Crear 50 contenidos de tipo noticia y comprobar el resultado.

6) Apartado “Contacto”

Crearemos el apartado “Contacto” como una página normal de tipo de contenido “Basic page”, por lo que no es necesario crear ningún tipo de contenido nuevo para este apartado. Al crear la página, nos aseguraremos de añadirla al menú principal.

7) Instalación de un editor de textos más potente para el administrador

En un proyecto real, en este punto nos daremos cuenta de que probablemente necesitamos un editor de texto más potente (como el que WordPress tiene por defecto). Para solucionar este problema podemos instalar módulos como FCKEditor o TinyMCE, o incluso mejor el meta-módulo <http://drupal.org/project/wysiwyg>. Se trata de un meta-módulo en el sentido de que es una base para instalar diversos editores de texto y activarlos, pero para ello deberemos descargar adicionalmente los archivos del editor que queramos utilizar (siguiendo las instrucciones).

8) Creación de entradas en el menú para cada una de las categorías

Para conseguir las entradas para cada una de las secciones (categorías) que hemos definido, iremos al apartado “Estructura/Menús” y editaremos los enlaces de “Main menu” (el menú principal). Los menús consisten en dos campos, el texto que se muestra al usuario y el enlace que activa.

De hecho, cualquier término de cualquier vocabulario tiene un id y la ruta “/taxonomy/id” nos llevará siempre al listado de todos los contenidos que incluyen ese término.

Para saber el enlace que activa cada categoría, podemos hacer clic en esa categoría en alguna página y copiar la ruta del enlace del navegador, pero debemos tener en cuenta que las rutas siempre deben ser relativas. Por tanto, en el menú siempre introduciremos la ruta interna.

Suponiendo que nuestra ruta absoluta fuera (incluye el código del idioma actual) <http://localhost/revista/es/taxonomy/term/5>, la ruta relativa que debemos introducir en el menú sería: “taxonomy/term/5”.

En caso contrario no podríamos trasladar el contenido del sitio a otro servidor, ya que las rutas serían distintas.

Es importante tener en cuenta que, al introducir los términos de un vocabulario, podemos asignar en el campo “Alias de url” la dirección que queremos que aparezca en el navegador, por ejemplo “Deportes” en lugar “de taxonomy/term/27”. Esto es mucho mejor tanto para el usuario como para los buscadores de Internet.

9) Definir roles específicos para el editor

En un proyecto real también necesitaremos crear roles especiales para el personal que se encargará de administrar el sitio web.

Para crear un rol que pueda, por ejemplo, crear y editar contenido sin acceso al resto de opciones de Drupal, seguiremos los siguientes pasos:

- En el apartado “Personas/Permisos/Roles” crearemos un rol nuevo llamado “Editor”.
- Posteriormente, iremos al apartado “Permisos” y activaremos para este rol solo los permisos relativos al apartado “Node access” para crear, editar y borrar contenidos.
- Finalmente, crearemos un usuario en el apartado “Personas/Lista” y le asignaremos el rol “Editor”.

Ejercicio

Probar que efectivamente funciona, creando el rol “Editor” con permisos específicos, un usuario con este rol y, posteriormente, entrando al sitio en otro navegador diferente (para que no mantenga la sesión de Administrador), utilizando un usuario con rol “Editor” y comprobando que podemos crear y editar contenido pero sin tener acceso al resto de las opciones del panel de administración.

10) Listado de fotos

En nuestra propuesta habíamos dicho que existiría un apartado “Foto” que mostraría fotos de temas de actualidad. En Drupal **no existe una manera inmediata de mostrar un listado de contenidos de un tipo determinado**. Para solucionar este problema podemos utilizar el módulo “Vistas” (“Views”).

Atención: Este módulo depende de otro, <http://drupal.org/project/ctools>, que habrá que instalar previamente (si no lo hacemos, veremos un aviso al respecto al instalar Views).

“Vistas” es uno de los módulos más útiles de Drupal, ya que nos permite crear fácilmente páginas o bloques personalizados filtrando y ordenando contenidos como queramos.

Ejercicio

Instalar el módulo “Vistas” y activarlo.

Para crear una página especial con el listado de fotos seguiremos los siguientes pasos una vez instalado el módulo “Views”:

- Ir a “Estructura/vistas”.
- Agregar nueva vista.
- Darle el nombre “Fotos” y filtrar por tipo de contenido “Foto”.
- Dar el nombre que queramos a la dirección de la página (por ejemplo “Fotos”).
- Esta será la dirección que tendremos que utilizar si queremos añadirla al menú principal.

Utilizando el panel de la vista podemos fácilmente crear también un bloque con fotos, personalizar los campos que queremos que aparezcan, etc.

11) Listado de artículos de opinión

De la misma manera, crearemos también el apartado donde se podrán ver los artículos de opinión. Crearemos una vista “Opinión” que liste todas las entradas de este tipo y una entrada en el menú para esta sección. También crearemos un bloque con los títulos de los últimos cinco artículos de opinión y los asignaremos a la barra lateral.

Ya hemos creado la estructura de contenidos que necesitamos (tipos de contenido, taxonomías y vistas). El siguiente paso será crear el tema que mostrará estos contenidos al usuario. Exactamente igual que haríamos con WordPress, una manera ágil de conseguir un tema personalizado es crear un tema derivado de otro y personalizar el CSS.

12) Personalizar un tema base

Si utilizamos un tema base, lo mejor (aunque no imprescindible) es utilizar algún tema *framework*, como por ejemplo Zen.

Un **tema *framework*** es aquel que define el *layout* y los estilos correspondientes, pero que no describe la parte de presentación.

Los pasos para crear un tema derivado de Zen se pueden encontrar en el “readme.txt” del tema. Básicamente, consisten en copiar la subcarpeta “Starterkit” como carpeta de nuestro tema y cambiar todas las referencias a “Starterkit”.

13) Alternativa: instalar un tema *framework responsive*

Otra posibilidad sería utilizar como base un tema *responsive* que se adapte a pantallas de diferentes tamaños.

Algunos de los mejores que podemos encontrar para Drupal son Omega (<http://drupal.org/project/omega>) y Adaptive Theme (<http://drupal.org/project/adaptivetheme>).

Estos temas, aparte de adaptarse a las diferentes resoluciones y dispositivos, también optimizan las imágenes para diferentes resoluciones y tienen muchas opciones para los desarrolladores.

Página web

Podemos encontrar el tema Zen en la web oficial: <http://drupal.org/project/zen>.

Vídeo

Screencast sobre cómo personalizar Zen:

<http://drupalonlinetraining.com/youtube/linwclark/drupal-7-basic-theming-zen-and-css>

Página web

Más información sobre cómo crear sub-temas en la página de Drupal en “Creating a sub-theme”: <http://drupal.org/node/225125>.

5. Instalar módulos

Al instalar Drupal disponemos de partida de una serie de módulos que podemos activar en caso de necesidad. Estos módulos que forman parte de la instalación inicial son los denominados del núcleo o *core*.

Además de los módulos del núcleo, podemos instalar centenares de módulos adicionales creados por la comunidad Drupal. En la página oficial podemos ver más de 10.000, organizados por temas.

Recomendaciones sobre el uso de módulos:

- Asegurarse de que el módulo es compatible con la versión de Drupal que usamos.
- Asegurarse de que es estable. Esto se puede comprobar en el listado de módulos de drupal.org porque aparece en color verde; si está en desarrollo (rojo) probablemente funcione pero no tendremos fiabilidad total, por lo que no es recomendable en un proyecto comercial.
- Leer a fondo la documentación del módulo antes de instalarlo para asegurarnos de que cumple las especificaciones que queremos.
- En cualquier caso, siempre deberíamos instalarlo en un servidor de pruebas antes de hacerlo en el servidor de producción. De esta manera, podremos detectar problemas antes de darlo por definitivo. Una vez comprobado, ya lo podremos instalar en el servidor final.

Downloads

Recommended releases

Version	Downloads	Date	Links
7.x-3.3	tar.gz (1.51 MB) zip (1.72 MB)	2012-Feb-22	Notes
6.x-2.16	tar.gz (1.21 MB) zip (1.35 MB)	2011-Nov-14	Notes

Other releases

Version	Downloads	Date	Links
6.x-3.0	tar.gz (1.13 MB) zip (1.31 MB)	2012-Jan-04	Notes

Development releases

Version	Downloads	Date	Links
7.x-3.x-dev	tar.gz (1.54 MB) zip (1.76 MB)	2012-Jun-09	Notes
6.x-3.x-dev	tar.gz (1.1 MB) zip (1.26 MB)	2012-May-19	Notes

Diferentes versiones y código de colores en la página del módulo Views.

Como hemos dicho, existen infinidad de módulos, con funcionalidades totalmente diversas, algunos de ellos tan completos que requerirían varios capítulos.

Ejercicio

Instalar y probar alguno de estos módulos en un servidor de pruebas:

- XML Sitemap (<http://drupal.org/project/xmlsitemap>). Genera automáticamente un archivo "sitemap.xml" compatible con Google en la raíz del servidor.
- Pathauto (<http://drupal.org/project/pathauto>). Genera nombres de direcciones para las páginas según los títulos del contenido (por ejemplo "/unejemplo en lugar de /node/24").
- Ubercart (<http://drupal.org/project/ubercart>). Módulo de comercio electrónico de gran complejidad y con muchísimas opciones. En realidad, se trata de un grupo extenso de módulos.
- Simplenews (<http://drupal.org/project/simplenews>). Sistema completo de gestión de *newsletters*.
- FiveStar (<http://drupal.org/project/fivestar>). Sistema de votación/valoración de los usuarios.
- Organic Groups (<http://drupal.org/project/og>). Gestión de grupos de usuarios.

6. Tabla de correspondencias entre Drupal y WordPress

Llegados a este punto, es muy probable que hayamos encontrado similitudes entre las plataformas WordPress y Drupal. De hecho, muchos conceptos en ambas plataformas coinciden pero tienen nombres diferentes.

Veamos a continuación una tabla de equivalencias de algunos de estos conceptos:

Wordpress	Drupal	Comentarios
<i>Plugins</i>	Módulos	En otras plataformas se llaman extensiones. Drupal parte inicialmente con muchos más módulos que WP.
<i>Widgets</i>	Bloques	En WP existe además <i>drag & drop</i> .
Tema	Tema	En ambos casos los temas están dentro de la carpeta "Themes" del servidor.
Escritorio	Escritorio	Visión general del panel de administración.
Entradas y páginas	Artículos y páginas	Los dos tipos de contenido que existen por defecto en ambas plataformas.
.php	.tpl.php	Extensión de las plantillas del tema.
functions.php	template.php	Archivo del tema para añadir instrucciones PHP de cada sistema.
style.css	nombre.info	Archivo donde se declaran las opciones del tema.

A primera vista podríamos considerar Drupal como una ampliación de WordPress, en el sentido de que tiene más funcionalidades por defecto, sin necesidad de instalar módulos.

Algunas de estas funcionalidades que no existen por defecto en WordPress (a no ser que instalemos *plugins*) son las siguientes:

- Posibilidad de definir nuevos tipos de contenido.
- Posibilidad de definir campos en los tipos de contenido y de decidir el orden o cómo se visualizan.
- Gestión de contenidos en múltiples idiomas.
- Soporte para RDF.

- Drupal no dispone de un editor de texto “WYSIWYG” preinstalado (Word-Press sí, TinyMCE), pero permite instalar una diversidad de ellos.
- Sistema de gestión de usuarios, roles y permisos mucho más completo.

Como contrapartida, la curva de aprendizaje de Drupal es más larga que la de WP debido a su complejidad.

Drupal es probablemente una opción para sitios web complejos con una estructura de contenidos que deba personalizarse a fondo según los requisitos.

7. Integración con redes sociales

En la actualidad una aplicación web no se puede considerar completa si no implementa funcionalidades que la vinculen con redes sociales. Si los contenidos que se publican en nuestro sitio se publican además en otras redes, aumentaremos mucho su visibilidad y esta visibilidad es una puerta de entrada a nuevas visitas.

Además de usar las redes como medio de publicación, también podemos **integrar componentes de red social** en nuestro sitio siguiendo algunas normas elementales:

- Que los usuarios de la web puedan generar contenidos (creando tipos de contenido con los permisos apropiados para usuarios registrados o anónimos).
- Que los usuarios queden identificados con su nombre y su avatar (imagen de perfil).
- Que puedan crear grupos, contactos y puedan enviarse mensajes entre ellos. Estas funcionalidades no existen por defecto pero se pueden proporcionar mediante módulos adicionales.
- Además, podemos facilitar el acceso mediante el uso de cuentas externas:
 - Mediante OpenId (módulo del *core*).
 - Facebook Connect (módulo adicional).
 - Twitter (módulo adicional).

PERSONALIZACIÓN

Permitir firmas.

Activar imágenes de usuario.

Directorio de imágenes

Subdirectorio de la carpeta de archivos subidos en el que se guardarán las imágenes.

Imagen predeterminada

URL de la imagen que se debe mostrar para los usuarios que no hayan personalizado su foto. Déjelo vacío para no usar

Estilo de presentación de imagen

El estilo seleccionado se usará en la presentación y la imagen original se mantendrá. Los estilos se pueden configurar en

En "Configuración/Opciones de la cuenta" podemos activar las imágenes de perfil de los usuarios e incluso asociar una imagen predeterminada.

Existen varios módulos adicionales (no incluidos en el *core*) que facilitan el uso del CMS como red social:

- **Organic Groups** (<http://drupal.org/project/og>). Permite a los usuarios crear y administrar sus propios grupos. Cada grupo puede tener suscriptores y mantener una página principal del grupo en la que los suscritos se comunican entre sí.
- **User Relationships** (http://drupal.org/project/user_relationships). Permite a los usuarios crear relaciones entre ellos. Es un componente básico para una red social o cualquier otro sitio donde los usuarios deban comunicarse entre ellos.
- **PrivateMsg** (<http://drupal.org/project/privatemsg>). Permite a los visitantes del sitio web enviarse mensajes privados entre sí. Perfecto para sitios orientados a la comunidad, como los foros y sitios de redes sociales.
- **Invite** (<http://drupal.org/project/invite>). Las invitaciones son importantes para crear efectos de red y el crecimiento exponencial de una comunidad de intereses. En este módulo se añade una función “Invitar a un amigo”.
- **Contact Importer** (http://drupal.org/project/contact_importer). Permite importar usuarios de otras redes sociales.
- **Fivestar** (<http://drupal.org/project/fivestar>) (depende de la Voting api). Sistema de votación para los contenidos.
- **Radioactivity** (<http://drupal.org/project/radioactivity>). Sistema para priorizar información según la popularidad, lo más votado, comentado, visto, etc.
- **Activity** (<http://drupal.org/project/activity>). Muestra información de la actividad de cada usuario (tal y como hace Facebook, por ejemplo).
- **Feeds** (<http://drupal.org/project/feeds>). Permite integrar *feeds* (fuentes de información en RSS) que generan automáticamente contenido (nodos).
- **Twitter** (<http://drupal.org/project/twitter>). Integra la cuenta de Drupal con la de Twitter, de manera que se publican automáticamente en Twitter los contenidos que publicamos en nuestro sitio.
- **Gravatar** (<http://drupal.org/project/gravatar>). Integra Gravatar como imagen de perfil de los usuarios. Gravatar es un sistema global de imágenes de perfil asociadas a cuentas de correo. En realidad, Gravatar es un servicio que permite obtener la imagen de perfil de cualquier usuario a partir de su

Página web

Para más información se puede consultar la página web de Gravatar: <http://es.gravatar.com/>.

dirección de correo. Al activar este módulo, la imagen de perfil de los usuarios vendrá del servicio Gravatar en lugar del sistema interno de Drupal.

Otros módulos que pueden ser interesantes son:

- Subscriptions (<http://drupal.org/project/subscriptions>). Permite a los usuarios suscribirse a los contenidos.
- CAPTCHA (<http://drupal.org/project/captcha>). Seguridad anti *bots*.

Lectura recomendada

Para saber más:
"Community building and social networking modules" (<http://drupal.org/node/206724>).

8. Creación de temas (*themes*) en Drupal

La **personalización** del aspecto de nuestro sitio desarrollado en Drupal para los usuarios se realiza siempre a través de un **theme (tema)**.

En este sentido, es totalmente equivalente a WordPress.

Un **tema** corresponde con una **serie de archivos (plantillas)** dentro de una carpeta común, que normalmente se sitúa en la ruta *"Sites/All/Themes"* (de manera análoga, los módulos se sitúan en *"Sites/All/Modules"*).

En la práctica (igual que en WordPress), también existen tres posibilidades para personalizar un tema:

- Utilizar un tema derivado de uno ya existente (*child theme*).
- Utilizar un tema derivado de un tema *framework*.
- Crear un tema desde cero.

8.1. Modificar un tema ya existente

La opción más sencilla, sin duda, es descargar un tema ya existente, por ejemplo de la página de temas oficial de drupal.org, copiarlo en el directorio correspondiente (*"Sites/All/Themes"*) y activarlo desde el panel de administración.

Una vez hemos activado el tema, siempre podremos modificar el CSS para personalizarlo según nuestras necesidades, aunque no es la manera correcta de hacerlo, ya que no podríamos actualizar el tema a nuevas versiones sin el riesgo de perder nuestros cambios.

La opción correcta es **crear un tema derivado**. Para ello debemos seguir los siguientes pasos:

- Crear una carpeta nueva en *"Sites/All/Themes"* con el nombre de nuestro tema, por ejemplo *"/ejemplo"*.
- En esta carpeta crear un archivo de texto, con el mismo nombre que la carpeta, *"ejemplo.info"* donde estarán los datos principales del tema (corresponde con *"style.css"* de WordPress).

Observación

El panel de Drupal ofrece también la posibilidad de subir el archivo directamente o copiar la URL del archivo *"gzip"* del tema, pero a efectos prácticos el resultado es el mismo: copiar los archivos en un directorio de *"Sites/All/Themes"*.

- En el archivo “info” definiremos la información que define el tema.

```
;el nombre del tema
name = Ejemplo

description = Tema derivado de otro

; obligatorio, la versión del "core"
core = 7.x

; hoja de estilos, al menos definimos una nueva, deriva las del padre
stylesheets[all][] = ejemplo.css

;necesitamos definir todas las regiones, aunque estén en el padre
;obligatoriamente debe existir la región 'content'
regions[content] = Contenido

;el tema del cual deriva, debe existir en sites/all/themes
base theme=otrotema
```

Ejercicio

Visitar la página de temas de Drupal, escoger uno de ellos, descargarlo y activarlo en nuestra instalación local de Drupal. Crear un tema derivado y activarlo. Personalizar algún detalle del CSS y observar el resultado.

8.2. Utilizar un tema *framework* como base

Utilizar un tema base cualquiera no es, a menudo, la mejor opción, ya que probablemente tengan una estructura fija difícil de modificar.

Existen un tipo de temas pensados especialmente para crear temas derivados, los denominados **temas *framework***.

Entre ellos, tenemos **Zen** (<http://drupal.org/project/zen>). Zen es un tema especial de Drupal en el cual están definidas todas las etiquetas en el CSS, pero están “vacías”. Es un tema *framework* cuyo objetivo es facilitarnos al máximo la creación de un tema totalmente personalizado, pero haciendo al mismo tiempo parte del trabajo por nosotros.

Las etiquetas del CSS se corresponden con las etiquetas de la plantilla, de manera que, básicamente, lo único que debemos hacer en muchos casos es modificar el archivo CSS para crear nuestra personalización.

Zen nos facilita el trabajo de crear un tema derivado, ya que nos proporciona una plantilla básica para ello; se trata de la carpeta “Starterkit”. Simplemente debemos copiarla en “*Sites/All/Themes*” y hacer una serie de cambios documentados en: <http://drupal.org/node/1549668>.

El tema Zen no solo está pensado para modificar el CSS, sino que permite modificar también las plantillas “*tpl.php*”, lo que da una gran flexibilidad para crear temas a medida con una estructura diferente.

Existen muchos otros temas *framework* para Drupal, entre otros podemos destacar:

- Omega.
- AdaptativeTheme.
- Fusion.

8.3. Crear un tema desde cero

Como tercera opción vamos a ver cómo podemos crear nuestra propio tema partiendo desde cero. Este es el sistema que elegiríamos en el caso de querer adaptar un diseño creado en HTML y CSS a una plantilla dinámica de Drupal. Es el **método** que ofrece **más flexibilidad a nivel de diseño y estructura de navegación**, pero también, y en consecuencia, **la mayor complejidad y coste en tiempo**.

Las plantillas del tema definen tanto la estructura de navegación del sitio, como la presentación (básicamente definida por el o los archivos CSS).

Debemos tener identificadas las zonas de la página donde queremos que el contenido sea dinámico, que se transformarán en las regiones. Convencionalmente, estas regiones son la cabecera, una barra lateral y el *footer*, pero puede ser que necesitemos otras.

Antes de tratar cómo implementar un tema completo, vamos a ver un caso extremo: el tema más simple que podemos desarrollar.

8.3.1. Creando un tema “minimalista”

Intentaremos crear el tema más simple posible que permita la navegación por los contenidos de Drupal. Primero debemos pensar un nombre para el tema, y pondremos ese nombre a la carpeta donde guardaremos los archivos. En nuestro caso vamos a imaginar que nuestro primer tema se llama *minimalista*. Por tanto, la ruta de la carpeta será: “*Sites/All/Themes/minimalista*”.

Para crear nuestra plantilla (*theme*) a medida necesitamos al menos **dos archivos**, los más importantes:

Observación

Siempre es importante fijarse en que la versión de los temas o módulos que bajamos de drupal.org u otro sitio sean compatibles con la versión del *core* que estemos usando.

Página web

Listado completo de temas *framework* o *starter*: <http://drupal.org/node/323993>.

- **minimalista.info**: La configuración de nuestra plantilla.
- **page.tpl.php**: La plantilla que utilizarán todas las páginas.

En el archivo “**minimalista.info**”, que no es más que un archivo de texto, definimos el nombre, las hojas de estilo, los archivos JavaScript y las regiones de nuestro tema.

En nuestro caso **minimalista** podríamos tener lo siguiente:

```
name = Minimalista ;el nombre del tema
description = Tema muy muy simple de ejemplo
engine = phptemplate ; obligatorio
stylesheets[all][] = style.css ; hoja de estilos, podemos añadir más
regions[content] = Contenido ;esta región es obligatoria!
regions[mi_region] = Una sola región extra ;
core = "7.x" ;obligatorio, qué versión de Drupal se requiere
```

Una vez hemos creado este archivo de texto y lo hemos guardado como “**minimalista.info**”, nos queda crear la plantilla principal del tema, llamada “**page.tpl.php**”. La plantilla “**page.tpl.php**” se corresponde con el contenido del `<body>`, de manera que esta etiqueta HTML **no** debe estar en la plantilla.

Como estamos creando un tema **minimalista**, el contenido podría ser el siguiente:

```
<h1>Mi plantilla</h1>

<?php if ($page['content']) : ?>
<div class="content">
<?php print render ($page['content']); ?> <!-- mostrar el contenido de
la región content -->
</div>
<?php endif; ?>

<?php if ($page['mi_region']) : ?>
<div class="mi_region">
<?php print render ($page['mi_region']); ?> <!-- mostrar contenido de
la región mi_region -->
</div>
<?php endif; ?>
```

Vemos que la plantilla tiene algunas instrucciones de PHP incrustadas. Estas instrucciones se encargan de escribir dentro del documento el contenido asignado a las regiones que tenemos definidas.

La primera línea significa “si hay contenido definido en la región *content*”. A continuación, la instrucción

```
print render ($page['content']);"
```

imprime el contenido de esta región en la página, y lo mismo para la región “*mi_region*”.

Pero, ¿cómo se traduce esto en el administrador de Drupal?

Para verlo en funcionamiento, lo que tenemos que hacer en primer lugar es activar este tema minimalista en el panel de administración, en la opción “Apariencia” (“*Appearance*”).

El “*content*” de una página de Drupal es el contenido que se muestra en la página. Si estamos en la portada, será la lista de nodos publicados en la portada en formato “*teaser*”; si estamos en la página única de un nodo (“*node/xx*”), será el contenido completo de un nodo, o serán los resultados de una búsqueda, etc. Por tanto, habrá muy pocas circunstancias en que “*\$content*” no tenga contenido.

Haciendo un paralelismo con WP,

```
print render ($page['content']);
```

equivale al *loop* de WP.

En este sentido, la región “*content*” actúa de una manera especial. El resto de regiones mostrará solo los bloques que le hayamos asignado en el panel de administración en el apartado “Estructura/Bloques”. En este caso corresponde a los *sidebars* de WP.

En nuestro ejemplo, “*mi_region*” será una región a la cual deberemos asignar bloques mediante el panel de administración. Al entrar en “Estructura/Bloques” veremos que podemos asignar los bloques a la lista de regiones definidas en info, en nuestro caso solo “*content*” y “*mi_region*”:

```
<d7_regiones_minimalista.jpg>
```

Blocks MINIMALISTA

Home » Administration » Structure

This page provides a drag-and-drop interface for assigning a block to a region, and for controlling the order of blocks within regions. Since not all themes implement the same regions, or display regions in are positioned on a per-theme basis. Remember that your changes will not be saved until you click the *Save blocks* button at the bottom of the page. Click the *configure* link next to each block to configure its visibility settings.

Demonstrate block regions (Minimalista)

[+ Add block](#)

BLOCK	REGION	OPERATIONS
Contenido		
+ Search form	Contenido	configure
+ Main page content	Contenido	configure
+ Navigation	Contenido	configure
+ System help	Contenido	configure
+ User login	Contenido	configure
+ Powered by Drupal	Contenido	configure
Una sola región extra		
No blocks in this region		
Disabled		
+ Main menu	- None -	configure

Asignando bloques a las regiones definidas en nuestro "archivo .info".

Si hemos vinculado los bloques del formulario de búsqueda y el "user login" a la región "mi_region", la magia de Drupal hará que en nuestra plantilla, la instrucción

```
<?php print render ($page['mi_region']); ?>
```

escriba el contenido de todos estos bloques automáticamente en su lugar.

Una vez activado nuestro tema "minimalista", podemos navegar por todos los contenidos de nuestro sitio. Por supuesto faltan detalles importantes, como por ejemplo la hoja de estilos. Vamos a crear una **hoja de estilos** también minimalista. Para ello editaremos un archivo llamado "style.css" y añadiremos los estilos asignados a los selectores que hemos introducido en la plantilla "page.tpl.php":

```
.mycontent{
  background-color: #ffeccc;
width:80%;
  float:left;
}
.mi_region{
background-color: #ccccff;
width:20%;
  float:right;
}
```

De esta manera podremos diferenciar visualmente las dos regiones que tenemos definidas en nuestra plantilla.



Resultado visual al aplicar el anterior css al tema.

Observad que si navegamos por el sitio, lo que cambia es el “*content*”, mientras que el contenido de la región “*mi_region*” es siempre el mismo. En Drupal, por defecto, los bloques asignados a las regiones (excepto “*content*”) se ven en todas las páginas del sitio. Exactamente igual que en WordPress con los *widgets*, que por defecto se ven en todas las páginas.

Observación

Si analizamos el código fuente de nuestra página, veremos que Drupal ha añadido muchas cosas al HTML que tenemos en nuestro tema: entre otras cosas, el código de la cabecera de la página, que incluye el nombre, la inclusión de hojas de estilo y JavaScripts, etc. También añade el tag “*body*” con clases específicas para cada nodo o sección. Más adelante veremos que existe otra plantilla más general, que también podemos editar, que es la que engloba “*page.tpl.php*”, llamada “*html.tpl.php*”. En esta plantilla tenemos la parte del documento externa al *body* (*head*, etc.).

8.3.2. Creando un tema más completo

Por supuesto, podemos definir un tema tan en profundidad como queramos. Todas las partes de una página en Drupal provienen de alguna plantilla que se puede redefinir, por ejemplo, los bloques, los comentarios, los nodos, etc.

Para crear una plantilla siempre hay que partir de la original. La plantilla original es la que se utiliza por defecto si no hemos especificado ninguna. En este enlace, por ejemplo, tenemos el original de la plantilla *page.tpl.php* y las variables predefinidas por Drupal que podemos utilizar. El resto de plantillas las podemos encontrar en la página de documentación de Drupal.

Siempre que no definamos alguna de las plantillas, Drupal usará la que existe por defecto. La idea siempre es redefinir lo ya existente.

Algunos ejemplos de plantillas:

Observación

A diferencia de WordPress, en Drupal las variables predefinidas son distintas según la plantilla que utilicemos. Además, aquí no existe el concepto de *loop*.

- **html.tpl.php**: Como hemos comentado anteriormente, es la que incluye la estructura del documento html (excepto el contenido de *body* que corresponde con “page.tpl.php”).
- **node.tpl.php**: La plantilla de un nodo se utiliza para mostrar cualquier contenido de cualquier tipo de contenido. Podemos definir plantillas específicas para cada tipo de contenido utilizando su nombre, por ejemplo, “node--noticia.tpl.php”, suponiendo que tenemos un tipo de contenido “Noticia”. Corresponde con el contenido del *loop* en WP.
- **field.tpl.php**: Dado que los tipos de contenido pueden tener campos específicos, cada campo puede tener una plantilla especial. “field.tpl.php” define la plantilla para todos los campos por defecto, pero podríamos definir la plantilla “field-name-field_foto” en el caso de un campo llamado “field_foto”.

Imaginando que ya tenemos un diseño maquetado en HTML + CSS, con una sola estructura de página, los pasos que deberíamos seguir para crear un tema funcional serían:

- Copiar el HTML y CSS de nuestra plantilla de la maquetación en la carpeta del tema y cambiar el nombre del archivo HTML por “page.tpl.php” (este será la plantilla principal de nuestro tema).
- Crear el archivo “.info”.
- Copiarlo en “Sites/All/Themes” y activarlo.

8.3.3. Ejercicio. Creación de un tema desde cero a partir de una plantilla estática (HTML + CSS)

Vamos a partir de una maquetación estática en HTML y CSS.

En este ejercicio partiremos de los siguientes archivos:

- index.html. La plantilla principal.
- nasa_estils_responsive.css. La hoja de estilos.
- Archivos de imagen en /img.

El objetivo es conseguir que esta plantilla estática se convierta en dinámica y se integre con Drupal. Para ello los pasos que debemos seguir son los siguientes. Atención, es importante borrar la caché de Drupal cada vez que hagamos cambios en nuestro tema, ya que si no, no podremos ver los cambios.

Material

Los archivos se pueden obtener en http://www.kiwoo.org/recursos/ejemplos/drupal/maquetacion_nasa_responsive.zip

1) Copiar la plantilla estática como carpeta dentro de “*Sites/All/Themes*” (por ejemplo “*Sites/All/Themes/nasa*”).

2) Añadir un archivo de texto llamado “*nasa.info*”. En este archivo debemos indicar la información básica y las hojas de estilo que utilizamos.

```
name = Nasa
description = Un tema basado en una plantilla estática previa
core = 7.x
project = "nasa"
regions[content] = Content
regions[sidebar] = Sidebar
regions[footer] = Footer
stylesheets[all][] = nasa_estils_responsive.css
```

3) Hacer una copia de “*index.html*” como “*page.tpl.php*”. Eliminar el contenido del principio hasta `<body>` incluido, y al final eliminar también `</body></html>`. La plantilla “*page.tpl.php*” es solo del contenido de la página (lo que hay dentro de `<body>`).

4) A partir de este momento ya podríamos activar el tema (para probarlo), pero aparecerían algunos problemas:

- El contenido es estático (naturalmente).
- Si hay imágenes en el HTML no aparecen (dado que Drupal no encuentra la ruta).

Para solucionar el problema de la ruta de las imágenes debemos sustituirlas por rutas dinámicas. Por ejemplo, “*img/badge_001.png*” debe convertirse en:

```
<?=$base_path.$directory?>/img/badge_001.png.
```

Consejo

Podemos hacer un *replace all* en nuestro editor.

5) ¿Dónde aparece el contenido? Esta instrucción muestra el contenido de la región “*content*”:

```
<?= render($page['content']);?>
```

La región “*content*” es la más importante, ya que muestra el contenido que debe ir en cada página: sin esta región, todas las páginas serían iguales.

Para el resto de regiones debemos cambiar el nombre “*content*” por el nombre de la región correspondiente. Las regiones que se deben utilizar en “*page.tpl.php*” son las definidas en el “.*info*”, pero también existen otras internas de Drupal (como “*help*”, “*action_links*”, etc.).

6) Podemos consultar todas las variables que se pueden utilizar en “*page.tpl.php*”, incluidas las regiones por defecto en: <http://api.drupal.org/api/drupal/modules%21system%21page.tpl.php/7>.

7) En este ejercicio solo estamos creando una plantilla, la principal (“*page.tpl.php*”), pero podríamos definir muchas otras. Podemos encontrar una lista de todas las plantillas posibles en: <http://drupal.org/node/190815>.

8) Menú. En “*page.tpl.php*” existe una variable que contiene las entradas del menú. Podríamos comprobarlo escribiendo en la plantilla:

```
<?php
print_r($main_menu);
?>
```

Sin embargo, necesitamos el menú en formato lista desordenada. Afortunadamente, Drupal proporciona una función que nos ayuda a hacerlo:

```
<?php print theme_links( array('links' => $main_menu, 'attributes' =>
array('class' => array('moclase'))); ?>
```

(“*\$main_menu*” es el nombre del menú principal. Existen muchos parámetros que se pueden utilizar. También podemos optar por copiar este código de otro tema).

9) *Footer* dinámico. No es más que otra región (definida en el archivo “.*info*”) y, por tanto, para hacerla aparecer:

```
<?php
print render($page['footer']);
?>
```

10) Barra lateral. De la misma manera, donde queramos que aparezca el código, haremos:

```
<?php
print render($page['sidebar']);
```

Lectura recomendada

En relación con estos parámetros se puede consultar: “*Theme Links*” (http://api.drupal.org/api/drupal/includes%21theme.inc/function/theme_links/7).


```
?>
```

11) Es necesario mostrar información interna del sistema como *tabs* o mensajes. “\$messages” son los mensajes internos de Drupal que aparecen, por ejemplo, al enviar correctamente los campos de un formulario, etc. Para que podamos ver estos mensajes en nuestro tema, debemos añadir siempre:

```
<?php print $messages; ?>
```

Los *tabs* son pestañas que añade Drupal en algunas páginas, normalmente destinadas al administrador de la web.

Un caso típico es la opción de poder editar un contenido cuando estamos “loginados” como administrador (*tabs*; ver, *edit*):

```
<?php if ($tabs): ?><?php print render($tabs); ?><?php endif; ?>
```

12) Podemos definir otras plantillas. Por ejemplo, si necesitamos una estructura diferente especial para la *home* modificaremos:

```
page--front.tpl.php
```

13) De hecho, incluso podríamos crear plantillas para un nodo en concreto:

```
page--node--xx.tpl.php
```

donde “xx” es el id del nodo.

Esto ofrece bastante flexibilidad en la implementación de nuestro diseño.

14) Una de las plantillas más importantes es: “node.tpl.php”. Esta es **la plantilla de cada uno de los contenidos**.

La plantilla base de “node.tpl.php” se encuentra en “Modules/Node”. Podemos utilizarla como base y modificarla.

```
<div id="node-<?php print $node->nid; ?>" class="<?php print $classes; ?>
clearfix"<?php print $attributes; ?>>
<?php print $user_picture; ?>
<?php print render($title_prefix); ?>
<?php if (!$page): ?>
<h2<?php print $title_attributes; ?><a href="<?php print $node_url; ?>">
```

```
<?php print $title; ?></a></h2>
<?php endif; ?>
<?php print render($title_suffix); ?>
<?php if ($display_submitted): ?>
<div class="submitted"> <?php print $submitted; ?> </div>
<?php endif; ?>
<div class="content"><?php print $content_attributes; ?>>
<?php // We hide the comments and links now so that we can render them later.
hide($content['comments']);
hide($content['links']);
print render($content); ?>
</div>
<?php print render($content['links']); ?>
<?php print render($content['comments']); ?>
</div>
```

Si lo comparamos con WordPress, “node.tpl.php” sería el interior del *loop*. Ya hemos comentado que en Drupal no existe el concepto de *loop*.

Cuando en la plantilla principal escribimos:

```
<?php print render($page['content'])?>
```

Drupal incluye tantas veces como haga falta las plantillas “node.tpl.php” según los contenidos que existan en aquella página.

15) Podemos definir plantillas para cada tipo de contenido. Por ejemplo, para el tipo de contenido “Noticia” crearemos “node--noticia.tpl.php” (cuidado: ¡son 2 guiones!).

16) Además, podemos tener plantillas para los campos especiales, como por ejemplo “field--field_audio.tpl.php”. Atención, **al hacer cambios en nuestro tema es importante borrar la caché de Drupal antes de probar el resultado.**

En nuestra plantilla especial para el tipo de contenido podemos mostrar los campos donde queramos, primero escondiéndolos y luego mostrándolos donde queramos. Por ejemplo:

```
<?php
hide($content['field_foto']);
print render($content);?>
<div id="foto">
<?php print render($content['field_foto']);?>
</div>
```

Atención, en el panel de administración, en la opción “Tipos de contenido”, podemos cambiar si se muestran o no los campos (en modo “*Teaser*” y en modo “Página”) y cómo se muestran de una manera mucho más fácil. Modificaremos la plantilla solo si no hay otra opción.

17) En nuestras plantillas de nodo podemos utilizar variables para mostrar la información de manera diferente según dónde nos encontremos. Algunos ejemplos:

“\$teaser”: Nos indica si el nodo está en modo “*Teaser*” o no.

“\$is_front”: Nos indica si el nodo está en la portada.

\$logged_in: Indica que el usuario está “loginado”, etc.

Existen muchas otras variables documentadas en Drupal.org.

Resultado del ejercicio

Maquetación HTML+CSS convertida en tema de Drupal (a nivel muy básico):

http://www.kiwoo.org/recursos/ejemplos/drupal/maquetacion_nasa_responsive.zip

