

COMUNIDAD JOOMLA
BIBLIOTECA COMUNIDAD JOOMLA

Desarrollo de plantillas para Joomla!

Autor:
Leandro D'ONOFRIO

Correcciones:
Rafael GÓMEZ
Gustavo Raúl ARAGÓN

BAJO LICENCIA CREATIVE COMMONS

Agosto 2011

Índice general

1. Introducción a las plantillas en Joomla!®	2
1.1. Bienvenido/a	2
1.2. Obtener el material de aprendizaje	2
1.3. Requisitos	2
1.4. Software	3
1.5. Las plantillas en Joomla!	4
1.5.1. Tipos de plantillas	4
1.5.2. Encontrar las plantillas disponibles en la administración del CMS	6
1.5.3. Estructura básica de una plantilla	7
2. Desarrollo de plantillas para Joomla!	11
2.1. Creación de los archivos de la plantilla	11
2.1.1. Análisis previo	11
2.1.2. Utilizar la plantilla ‘atomic’ como base	12
2.2. El archivo templateDetails.xml	16
2.2.1. Sección 1: Información asociada	18
2.2.2. Sección 2: Carpetas y archivos	18
2.2.3. Sección 3: Posiciones de módulos	18
2.2.4. Sección 4: Definición de lenguajes	19
2.2.5. Descubrir la plantilla en la administración de CMS	20
2.3. El archivo index.php	21
2.3.1. Sección 1: Sesión y contexto del CMS	24
2.3.2. Sección 2: Framework JavaScript	24
2.3.3. Sección 3: Referencia a JApplication	25
2.3.4. Sección 4: Tipo de documento, codificación y estilo de escritura	25
2.3.5. Sección 5: Cabecera del documento	25
2.3.6. Sección 6: Archivos CSS	26

2.3.7. Sección 7: Archivos CSS para otras direcciones de escritura	26
2.3.8. Sección 8: Funcionalidades JavaScript	27
2.3.9. Sección 9: Nombre del sitio	27
2.3.10. Sección 10: Llamado a distintos módulos	27
2.3.11. Sección 11: Mensajes del sistema	29
2.3.12. Sección 12: Contenido de los componentes	29
2.3.13. Sección 13: Año presente y nombre del sitio	29
2.4. El sistema de grillas de BluePrint	30
2.5. Maquetación del diseño	32
2.5.1. Modificaciones dentro de la etiqueta head	32
2.5.2. Maquetación de la sección superior	32
2.5.3. Maquetación del texto superior, video y caja destacada	33
2.5.4. Maquetación de las columnas del medio	35
2.5.5. Maquetación del texto destacado inferior y slideshow	38
2.5.6. Maquetación del footer y zona inferior	39
2.6. El archivo template.css	42
2.6.1. Estilos generales	43
2.6.2. Estilos para la sección superior	44
2.6.3. Estilos para el destacado superior, video y caja media	46
2.6.4. Estilos para las columnas centrales	50
2.6.5. Estilos para el texto destacado inferior y slideshow	52
2.6.6. Estilos para el footer y zona inferior	53
2.6.7. Estilos para elementos y mensajes del sistema	55
2.6.8. Estilos completos	57
2.7. Finalizando la plantilla	65
3. Temas avanzados	67
3.1. Estilos de plantillas, parámetros de configuración y maquetaciones diferentes por secciones	67
3.1.1. Cambiar la imagen del logo	68
3.1.2. Cambiar el texto del eslogan	72
3.2. Personalización de módulos y componentes	79
3.2.1. Extensiones desarrolladas en MVC	79
3.2.2. Módulos personalizados	82
3.3. Componentes personalizados	89
3.3.1. Conclusiones	94

3.4. Incorporación de jQuery	94
3.4.1. Creación de un slideshow con jQuery	97
3.5. Personalización de las páginas de errores	99
3.5.1. Sección 1: Parámetros de la plantilla	105
3.5.2. Sección 2: Tipo de documento, título y archivos CSS	105
3.5.3. Sección 3: Logotipo	106
3.5.4. Sección 4: Mensajes de error	107
3.5.5. Sección 5: Footer	108
4. Apéndice	111
4.1. Evitar que Mootools cargue de forma predeterminada	111
4.2. Estilos determinados para diferentes navegadores	111
4.3. Detectar la página principal del sitio	112
4.4. Detectar si un usuario está logueado en el sitio	112
4.5. Detectar un componente determinado	113

Capítulo 1

Introducción a las plantillas en Joomla!®

1.1. Bienvenido/a

En el momento de construir un sitio web utilizando Joomla!®, la creación de plantillas (o *templates*) suele ser uno de los trabajos más comunes que emprenden diseñadores y desarrolladores web.

Este manual pretende ser un documento fundamental para que, tanto usuarios noveles como avanzados, aprendan a realizar una plantilla funcional para Joomla! 1.6 y versiones posteriores.

Al finalizar el documento, deberá ser capaz de:

- Entender la estructura básica de una plantilla;
- Conocer los archivos que la componen y su funcionalidad;
- Poder agregar nuevas características y estilos personalizados;
- Conocer cómo modificar correctamente plantillas ya creadas;
- Saber crear plantillas desde cero.

1.2. Obtener el material de aprendizaje

El material de aprendizaje y el código fuente de los ejemplos que se utilizan en el libro están hospedados en [un repositorio de Github](#). Desde allí es posible descargar un archivo .zip o .tar con todo al material.

También esta invitado a seguir mejorando el material, ya sea [reportando errores o añadiendo nuevos materiales](#).

1.3. Requisitos

Para comenzar es necesario contar con los siguientes conocimientos mínimos:

- HTML/XHTML:
 - Conocer la estructura básica de un documento HTML/XHTML;
 - Saber reconocer las etiquetas básicas.
- CSS:
 - Reconocer la manera de incorporar estilos dentro de un documento;
 - Saber crear estilos básicos en archivos *.css*;
 - Entender el funcionamiento de maquetaciones simples con elementos flotantes.
- JavaScript:
 - Reconocer la manera de incorporar código JavaScript en un documento.
- PHP:
 - Reconocer las etiquetas de apertura y cierre de código PHP;
 - Entender las funciones básicas de flujos de control (**if**, **if else**, **else**).
- Joomla!:
 - Conocimientos básicos del CMS: administración de contenidos, instalación de extensiones, manejo de componentes y módulos.

1.4. Software

Para trabajar con los contenidos de la guía, son necesarias las siguientes herramientas:

- El navegador web **Firefox** y la extensión **Firebug**.
- Un editor de textos planos o un IDE (Entorno de Desarrollo Integrado):
 - **Notepad++** - Windows;
 - **Gedit/Kate** - Linux;
 - **TextMate** - Mac OS X;
 - **Aptana** - todas las plataformas;
 - **Netbeans PHP** - todas las plataformas.
- Un servidor local (**XAMPP** / **MAMP**) o un cliente FTP/SSH/SFTP para acceder a un servidor remoto (**Filezilla** / **WinSCP** / **Putty**);
- **Joomla! 1.6 o posterior** instalado en el servidor local o remoto.

1.5. Las plantillas en Joomla!

Antes de comenzar, puede que se pregunte ¿qué es una plantilla y para qué sirve?. Las plantillas son archivos que permiten controlar el diseño y la disposición de los elementos a lo largo de un sitio. Las mismas incluyen:

- El marcado HTML y la maquetación base del sitio;
- Los espacios donde se mostrarán los contenidos (artículos, módulos, componentes);
- El diseño (a través de estilos en cascada e imágenes);
- Y funcionalidades especificadas que dependerán de cada plantilla.

En particular, Joomla posee un excelente sistema de plantillas que permite adaptar cualquier sitio a los requerimientos deseados. Además, a medida que el *CMS* (Sistema de gestión de contenidos) evoluciona, el mismo va incorporando nuevas características que permiten mejorar la experiencia al momento de crear una plantilla.

A partir de la versión 1.6, el CMS incorpora varias características nuevas relacionadas al desarrollo de plantillas:

- Tres nuevos *templates* listos para utilizar y adaptar a las necesidades del sitio;
- El código generado por la aplicación es 100 % XHTML estricto;
- Incorporación de estilos en plantillas, una funcionalidad que permite dar más flexibilidad y personalización al sitio, permitiendo mostrar, por ejemplo, un estilo único en secciones determinadas;
- Detección integrada de navegadores, para poder cargar un estilo diferente para cada tipo de navegador.

Nota

Si desea conocer en mayor detalle otras características de Joomla! 1.6, puede visitar el sitio oficial dedicado a dicha versión: <http://www.joomla.org/16/>

1.5.1. Tipos de plantillas

Joomla cuenta con dos tipos de plantillas:

- **Plantillas para utilizar en la administración (*backend*):** Son creadas especialmente para la parte de administración, las cuales pueden incorporar funcionalidades que faciliten el manejo de los contenidos en el sitio.
- **Plantillas para utilizar en el sitio público (*frontend*):** Son creadas para ser visualizadas por los visitantes. Pueden ser simples o complejas, todo dependerá del tipo de sitio y hacia quienes está orientado. Durante este manual aprenderá a crear este tipo de plantillas.

Administración

Joomla!

Sitio Usuarios Menús Contenido Componentes Extensiones Ayuda

Ninguno en el sitio 2 en la administración Sin mensajes Ver el sitio Desconectar

Gestor de plantillas: Plantillas

Opciones Ayuda

Estilos Plantillas

Filtro: Buscar Limpiar - Seleccionar localización -

	Plantilla	Localización	Versión	Fecha	Autor
	Detalles de la plantilla atomic Sin previsualización disponible	Sitio	1.6.0	10/10/09	Ron Severdia contact@kontentdesign.com http://www.kontentdesign.com
	Detalles de la plantilla beeze_25 Sin previsualización disponible	Sitio	1.6.0	21 May 2010	Angie Radtke a.radtke@der-auftritt.de http://www.der-auftritt.de
	Detalles de la plantilla beeze_20 Sin previsualización disponible	Sitio	1.6.0	25 November 2009	Angie Radtke a.radtke@der-auftritt.de http://www.der-auftritt.de
	Detalles de la plantilla bluestork La previsualización de plantillas de la administración no está disponible	Administrador	1.6.0	07/02/09	Ron Severdia contact@kontentdesign.com http://www.kontentdesign.com
	Detalles de la plantilla hathor La previsualización de plantillas de la administración no está disponible	Administrador	1.6.0	May 2010	Andrea Tarr hathor@tarconsulting.com http://www.tarconsulting.com

Cantidad 20

Pack traducido y readaptado por Comunidad Joomla! | Joomla!® es software libre liberado bajo la GNU General Public License. Versión 1.6.1

Figura 1.1: La sección ‘Plantillas’ en la administración del CMS

1.5.2. Encontrar las plantillas disponibles en la administración del CMS

Las plantillas disponibles para utilizar se encuentran, dentro de la administración, yendo a: **Extensión > Gestor de plantillas > Plantillas**.

A través del filtro por ubicación es posible mostrar las plantillas pertenecientes al sitio o a la administración. Al hacer clic sobre la imagen correspondiente, se mostrará una vista alargada del diseño de la plantilla. Además, al clickear en el enlace de detalle, podrá acceder a los archivos principales que componen a la plantilla, así como editarlos desde la misma interfaz.

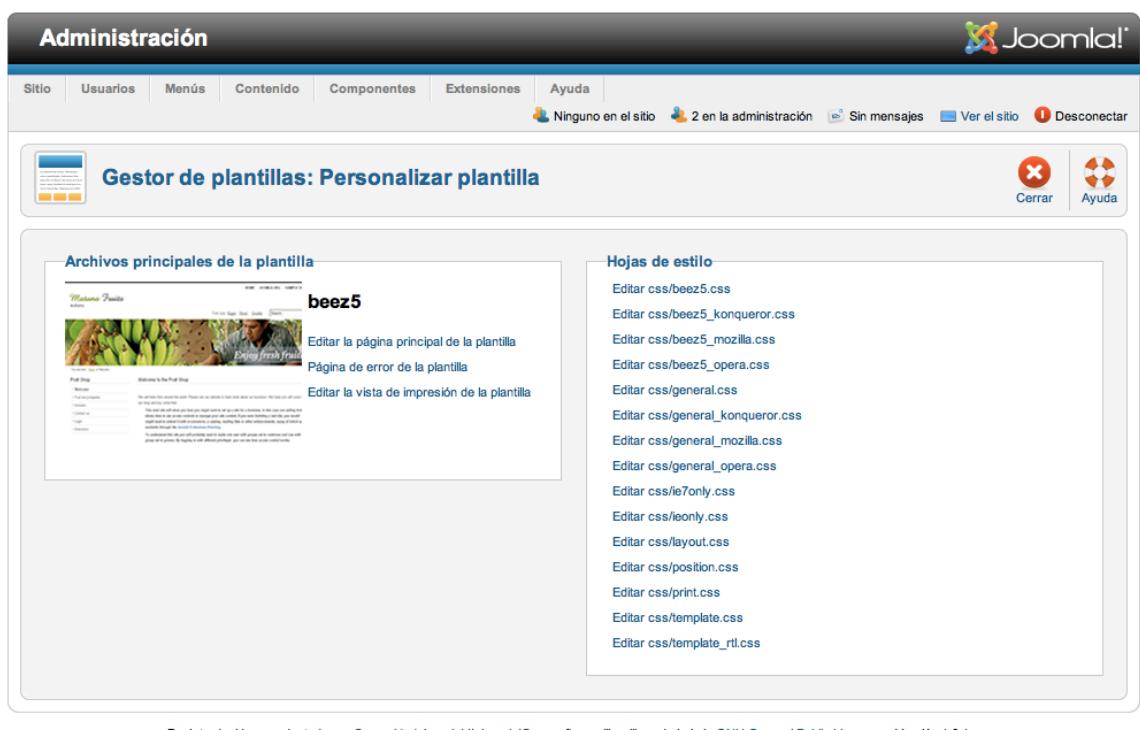


Figura 1.2: La sección ‘Personalizar plantilla’ en la administración del CMS

Por otro lado, al acceder al apartado estilos, como se mencionó antes, podrá navegar por una de las nuevas características introducidas en Joomla 1.6: Los estilos en plantillas.

Esta funcionalidad lo que permite es, utilizando un mismo *template* otorgar un diseño diferente a determinadas secciones. Por ejemplo: En gran parte del sitio se desea utilizar la plantilla con un diseño predeterminado, pero en la sección *contacto* se requiere que el color de fondo sea otro y que la estructura sea más angosta.

Nota

Las características de personalización dependerán de cada plantilla. Existirán plantillas que otorgarán muchas opciones y flexibilidad mientras que otras otorgarán pocas o ninguna.

Al acceder a uno de los ítems enlistados, es posible acceder a las opciones de personalización y especificar a qué secciones se desea asignar el estilo.

El nivel de personalización puede ser variado: Tamaño de estructuras, cambios de imágenes y textos, posición de elementos y diseño.

Estilo	Localización	Plantilla	Predeterminado	Asignado	ID
Atomic - Default	Sitio	atomic	★	✓	3
Beez5 - Default-Fruit Shop	Sitio	beez5	★	✓	6
Beez2 - Parks Site	Sitio	beez_20	★	✓	114
Beez2 - Default	Sitio	beez_20	★	✓	4
Bluestork - Default	Administrador	bluestork	★		2
Hathor - Default	Administrador	hathor	★		5

Figura 1.3: La sección ‘Estilos’ en la administración del CMS

1.5.3. Estructura básica de una plantilla

Las plantillas en Joomla! van alojadas dentro de la carpeta `templates`, en el directorio raíz del CMS. Allí dentro, cada plantilla esta separada por carpetas.

Todas las plantillas poseen la siguiente estructura básica de archivos y carpetas.

Los archivos y carpetas dentro de la estructura pueden dividirse en obligatorios y no obligatorios:

Carpetas y archivos obligatorios

Son obligatorios, ya que si no están presentes, la plantilla no funcionará:

- **Carpeta raíz:** Es la carpeta que contiene todos los archivos de la plantilla. Su nombre no debe poseer espacios intermedios. Su ubicación está dentro de la carpeta `templates`, que se encuentra en el directorio raíz;
- **index.php:** Archivo principal de la plantilla. Contiene la estructura básica, así como también las llamadas a archivos CSS y JavaScript. Su contenido está formado por etiquetas HTML y PHP;
- **templateDetails.xml:** Archivo XML que sirve para instalar la plantilla en el CMS. Posee la siguiente información:
 - Datos relacionados con la plantilla: nombre de la plantilla, autor, licencia, versión, etc;
 - Estructura de carpetas, archivos y lenguajes;
 - Nombres de las posiciones de módulos que la plantilla admite;
 - Opciones de configuración de la plantilla.

Administración

Joomla!

Sitio Usuarios Menús Contenido Componentes Extensiones Ayuda Ninguno en el sitio 1 en la administración Sin mensajes Ver el sitio Desconectar

Gestor de plantillas: Editar estilo

Guardar Guardar y cerrar Guardar como copia Cerrar Ayuda

Detalles

Nombre del estilo * Beez2 – Default

Plantilla beeze_20

Sitio

Predeterminado Todos

ID 4

Descripción de la plantilla Plantilla accesible para Joomla 1.6

Opciones avanzadas

Altura de plegado (%) 53

Altura de expansión (%) 72

Logotipo images/joomla_blae Seleccionar Limpiar

Título del sitio Joomla!

Descripción del sitio Open Source Content

Posición de navegación antes del contenido

Color de la plantilla Personal

Asignación a los menús

Selección de elementos del menú: Cambiar selección

Menú de usuario

- Su perfil
- Enviar un artículo
- Enviar un enlace web

Menú principal

- Inicio
- Mapa del sitio
- Artículos
- Enlaces web
- Contactos
- Conectarse
- Sitios de ejemplo
- Parques
- Tienda
- Administración del sitio
- Página de ejemplo

Superior

- Inicio
- Sitios de ejemplo
- Joomla.org

Tienda de frutas

- Bienvenido
- Enciclopedia de frutas
- Cultivos
- Contáctanos
- Conectarse
- Visítanos
- Añadir una receta
- Recetas

Pack traducido y readaptado por Comunidad Joomla! | Joomla!® es software libre liberado bajo la GNU General Public License. Versión 1.6.1

Figura 1.4: La sección ‘Editar estilo’ en la administración del CMS

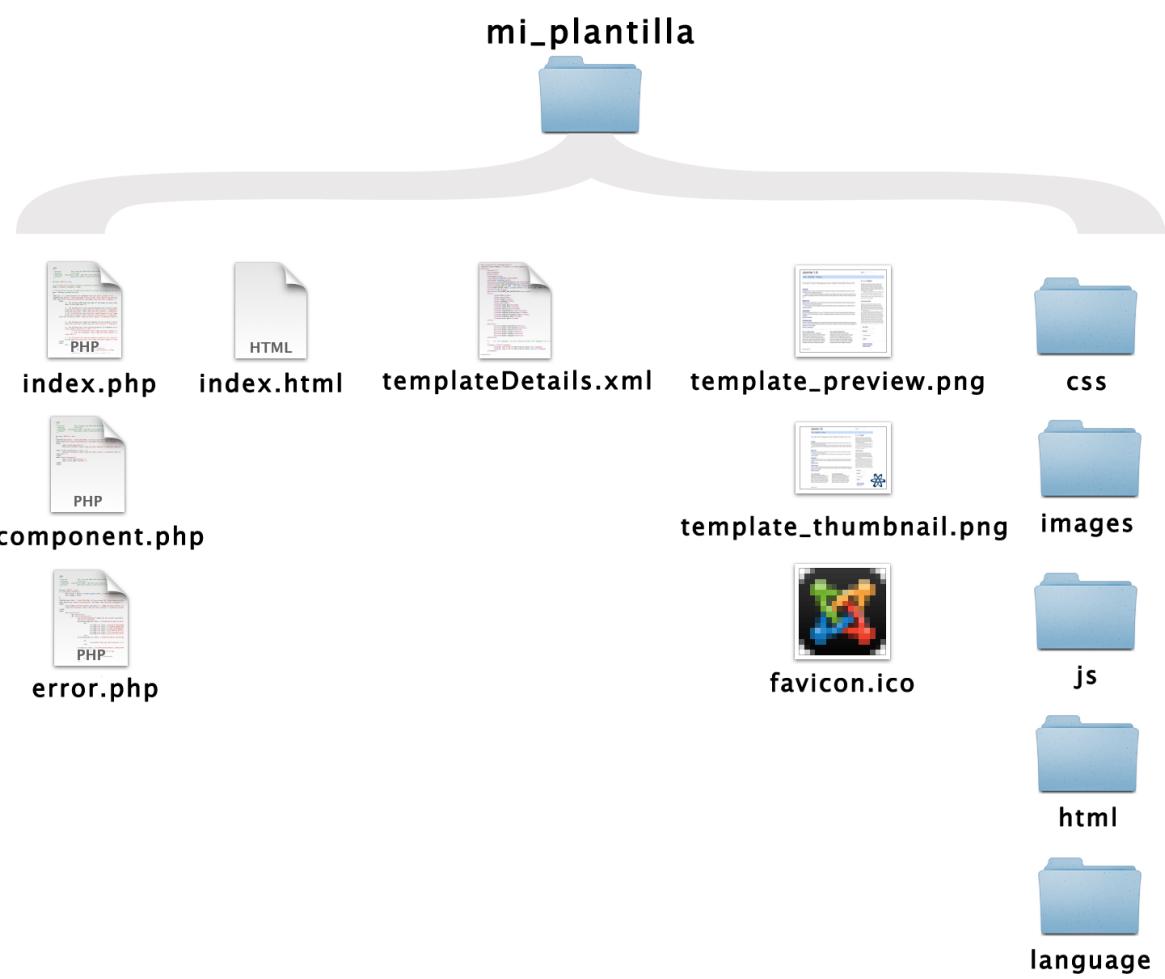


Figura 1.5: Esquema de la estructura básica de una plantilla

- **template_preview.png**: Imagen con una captura completa de la plantilla en funcionamiento. Sirve como vista previa al momento de seleccionar una plantilla en el panel de administración;
- **template_thumbnail.png**: Pequeña imagen que también actúa como vista previa al momento de seleccionar una plantilla en el panel de administración;
- **index.html**: Archivo HTML en blanco (sin etiquetas). Sirve como método de seguridad en servidores que permiten explorar directorios de archivos desde el navegador. Es recomendable crear un archivo **index.html** por cada carpeta que tenga la plantilla;
- **css**: Carpeta que contendrá los estilos CSS que utilizará la plantilla.

Carpetas y archivos no obligatorios

Son archivos que agregan funcionalidades o características a la plantilla, pero que no es obligatorio incluirlos. Crearlos o no, dependerá de las necesidades al momento de crear la plantilla:

- **component.php**: Archivo PHP que permite crear una versión reducida de la plantilla. Por lo general, se usa como vista previa al querer imprimir un artículo o enviarlo por correo electrónico. En caso de no existir, se utiliza el archivo **component.php** ubicado en la carpeta `/templates/system/`;
- **error.php**: Archivo PHP utilizado para mostrarse cuando ocurre algún error en el CMS o cuando una página no ha sido encontrada;
- **favicon.ico**: Imagen que se utilizará como ícono de la página. Joomla comprueba que el archivo exista y, en caso positivo, se incorporan automáticamente las etiquetas apropiadas para mostrarlo;
- **images**: Carpeta con imágenes a utilizar en el diseño de la plantilla;
- **js**: Carpeta con archivos JavaScript para utilizar en la plantilla;
- **html**: Los archivos alojados en esta carpeta permiten *sobrescribir* la salida HTML que imprime el CMS de forma predeterminada en componentes y módulos. Es decir, si desea personalizar el contenido HTML de una extensión, los archivos que lo harán serán ubicados en esta carpeta;
- **language**: Contendrá archivos del idioma `.ini`, los cuales servirán para mostrar textos diferentes dependiendo del idioma configurado para el sitio y la administración. Cada idioma está separado por su correspondiente carpeta.

Capítulo 2

Desarrollo de plantillas para Joomla!

2.1. Creación de los archivos de la plantilla

2.1.1. Análisis previo

Antes de comenzar a crear los archivos pertenecientes a la plantilla, es muy importante trabajar un aspecto fundamental en toda pieza de software: el análisis previo.

A continuación se enumeran algunas cuestiones que deberá tener en cuenta al momento del análisis:

- **Definiciones gráficas:**

- Contar con un diseño gráfico en el cual se basará la plantilla. El diseño deberá contemplar diferentes aspectos que permitan al usuario estar cómodo y además satisface las necesidades del sitio: Gama de colores, disposición de los elementos, técnicas gráficas utilizadas, usabilidad y accesibilidad.

- **Definiciones funcionales:**

- Dependiendo de los requerimientos y necesidades, el sitio contará con una cantidad definida de funcionalidades. Para implementarlas, deberá analizar de qué forma realizarlas: Utilizando las características que el CMS trae por defecto, instalando extensiones de terceros o desarrollando una solución personalizada.
- Debe tener en cuenta que la plantilla sea lo suficientemente extensible y flexible para adaptarse a distintas situaciones. Una manera de hacerlo es a través de la implementación de parámetros de configuración en la plantilla.

- **Definiciones técnicas:**

- Si ya ha trabajado en el mundo del diseño y desarrollo web, sabrá que no siempre es posible realizar un sitio que luzca exactamente igual en todos los navegadores, versiones y dispositivos. Particularmente, Internet Explorer suele ser el navegador más problemático. Por lo tanto deberá tomar como determinación qué navegadores, versiones y dispositivos serán soportados por la plantilla.

- Nombre de la plantilla e información asociada: Deberá elegir un nombre representativo para la plantilla y si lo desea, una licencia;
- Posiciones de módulos: Esta es una definición que dependerá de los elementos que compongan a la plantilla y su disposición. Para hacerlo, deberá delimitar el diseño en “zonas funcionales”, por ejemplo: zona perteneciente al menú, zona perteneciente a la caja de búsqueda, zona perteneciente a elementos izquierdos, etc. A partir de esta definición, le será fácil definir las posiciones de módulos y sus nombres.

Nota

Tenga en cuenta que no siempre se podrá contar con anticipación con todos los puntos nombrados, sin embargo, cuanto antes los tenga, mejor.

En el caso de este manual, se utilizará el siguiente diseño para la realización de la plantilla.

Al realizar el análisis, es posible definir las siguientes posiciones de módulos y funcionalidades:

- Tres espacios para módulos de menús (dos en la parte superior, uno en la parte inferior);
- Un espacio para un texto destacado;
- Un espacio para un video destacado;
- Un espacio para una texto resumido destacado;
- Un espacio central para tres artículos;
- Un espacio para un texto explicativo;
- Un espacio para un slideshow de imágenes;

Además la plantilla contará con las siguientes definiciones:

- Nombre de la plantilla: InterPoint
- Licencia: GNU General Public License version 2
- Navegadores soportados: Internet Explorer 8 o superior, Firefox 3.6 o superior, Google Chrome, Safari 5 o superior.

Una vez obtenidos los resultados del análisis es posible proseguir con la creación de los archivos.

2.1.2. Utilizar la plantilla ‘atomic’ como base

A partir de la versión 1.6, Joomla incorpora una nueva plantilla llamada *atomic*, la cual tiene como objetivo que pueda ser utilizada como base para la creación de nuevas plantillas. Su principal característica es la de utilizar *Blueprint*, un *framework* CSS que permite agilizar la creación de estilos y maquetaciones.

Para comenzar, lo primero será ir a la carpeta `templates` para copiar y pegar el directorio `atomic`. Luego se renombra la carpeta copiada con el nombre que se ha seleccionado para la plantilla.

Nota

El nombre debe estar en minúsculas y con guiones bajos (en caso de tener espacios de por medio).

Lo siguiente será ir por cada uno de los archivos para modificarlos en base a las necesidades solicitadas.



Sign-Up Login

Home About Us Client Services Features Blog Sliders Contact Us

Developing Business Solutions

Etiam vitae augue nunc. Quisque tincidunt feugiat vestibulum. Vestibulum rhoncus neque ligula. Proin eu erat est, in placerat mauris. Nam justo erat, pellentesque a auctor quis, elementum at urna. Phasellus varius, arcu eget semper ultrices, mauris enim molestie nulla, a sollicitudin

[Watch The Demo](#)



Sign-up for our free newsletter for all of our updates at InterPoint

Email Address

Submit



Solutions Made Easy

Suspendisse vulputate fringilla arcu et porttitor. Mauris placerat mattis tellus id sodales. Morbi vitae ipsum lectus, ullamcorper eleifend orci.

Vivamus pulvinar hendrerit dolor, vel scelerisque sapien dignissim non. In hac habitasse platea dictumst.

[>> Read More](#)



We Listen To You

Suspendisse vulputate fringilla arcu et porttitor. Mauris placerat mattis tellus id sodales. Morbi vitae ipsum lectus, ullamcorper eleifend orci.

Vivamus pulvinar hendrerit dolor, vel scelerisque sapien dignissim non. In hac habitasse platea dictumst.

[>> Read More](#)



Monitoring your Output

Suspendisse vulputate fringilla arcu et porttitor. Mauris placerat mattis tellus id sodales. Morbi vitae ipsum lectus, ullamcorper eleifend orci.

Vivamus pulvinar hendrerit dolor, vel scelerisque sapien dignissim non. In hac habitasse platea dictumst.

[>> Read More](#)

Simple Ideas For Simple Business Plans

Ut fermentum aliquam luctus. Vestibulum et dictum augue. Integer felis purus, bibendum at mollis non, eleifend vitae lacus. Etiam non nulla mi, a pharetra erat. Vivamus aliquam elit laoreet sem euismod ultrices.

Aenean nibh dolor, laoreet et sollicitudin imperdiet, pulvinar ut augue. Mauris tempor interdum elit. Etiam nec mauris a ante facilisis iaculis. Integer in tortor ut urna dapibus ultrices et nec mi.

[Read More](#)

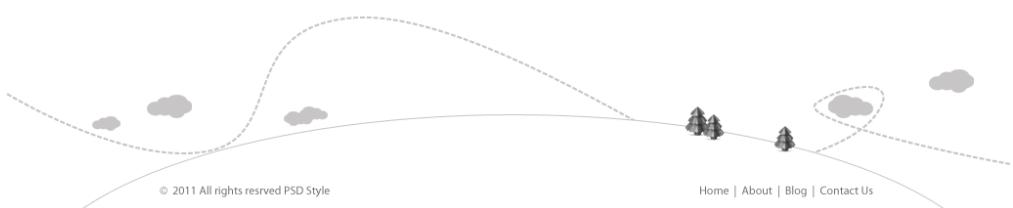


Figura 2.1: El diseño a realizar. Créditos: PSD Style - <http://psdstyle.net/>

InterPoint
In The Business of Business

Sign-Up Login

Home About Us Client Services Features Blog Sliders Contact Us

Developing Business Solutions

Etiam vitae augue nunc. Quisque tincidunt feugiat vestibulum. Vestibulum rhoncus neque ligula. Proin eu erat est, in placerat mauris. Nam justo erat, pellentesque a auctor quis, elementum at urna. Phasellus varius, arcu eget semper ultrices, mauris enim molestie nulla, a sollicitudin

[Watch The Demo](#)

Sign-up for our free newsletter for all of our updates at InterPoint

Email Address Submit

Solutions Made Easy

Suspendisse vulputate fringilla arcu et porttitor. Mauris placerat mattis tellus id sodales. Morbi vitae ipsum lectus, ullamcorper eleifend orci.

Vivamus pulvinar hendrerit dolor, vel scelerisque sapien dignissim non. In hac habitasse platea dictumst.

[>> Read More](#)

We Listen To You

Suspendisse vulputate fringilla arcu et porttitor. Mauris placerat mattis tellus id sodales. Morbi vitae ipsum lectus, ullamcorper eleifend orci.

Vivamus pulvinar hendrerit dolor, vel scelerisque sapien dignissim non. In hac habitasse platea dictumst.

[>> Read More](#)

Monitoring your Output

Suspendisse vulputate fringilla arcu et porttitor. Mauris placerat mattis tellus id sodales. Morbi vitae ipsum lectus, ullamcorper eleifend orci.

Vivamus pulvinar hendrerit dolor, vel scelerisque sapien dignissim non. In hac habitasse platea dictumst.

[>> Read More](#)

Simple Ideas For Simple Business Plans

Ut fermentum aliquam luctus. Vestibulum et dictum augue. Integer felis purus, bibendum at mollis non, eleifend vitae lacus. Etiam non nulla mi, a pharetra erat. Vivamus aliquam elit laoreet sem euismod ultrices.

Aenean nibh dolor, laoreet et sollicitudin imperdiet, pulvinar ut augue. Mauris tempor interdum elit. Etiam nec mauris a ante facilisis iaculis. Integer in tortor ut urna dapibus ultrices et nec mi.

[Read More](#)

© 2011 All rights reserved PSD Style

Home | About | Blog | Contact Us

Figura 2.2: Definición de las posiciones de módulos y funcionalidades

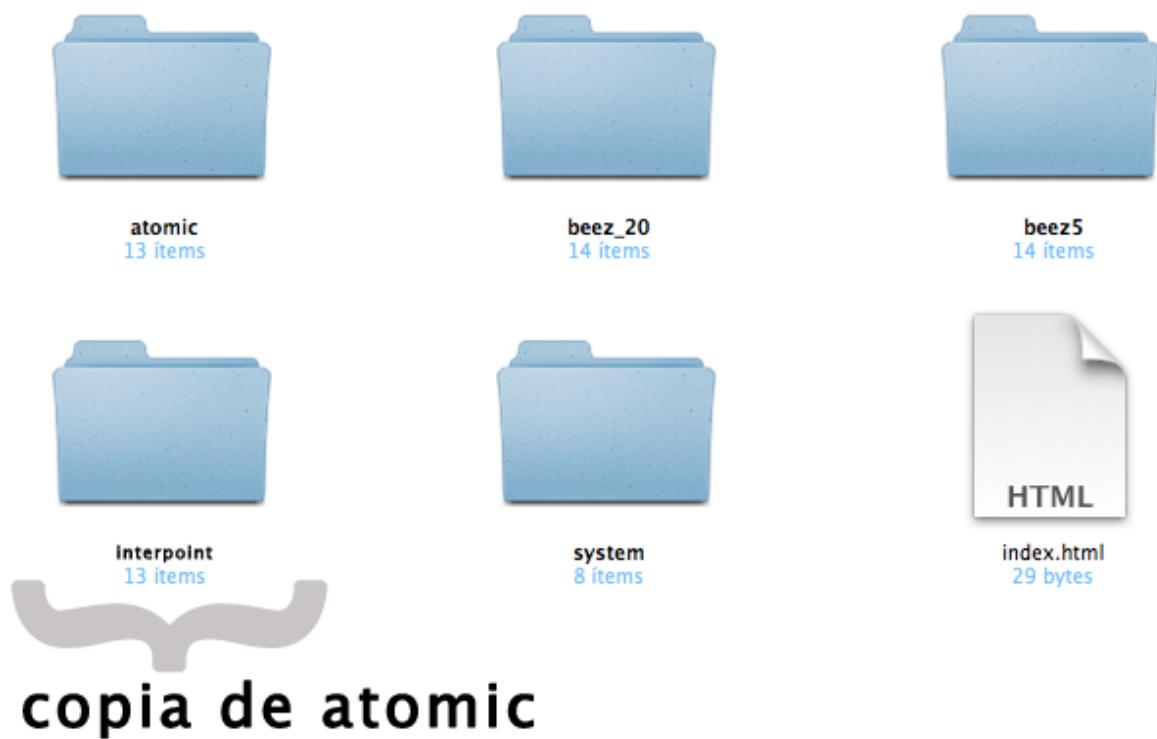


Figura 2.3: Copia de la carpeta 'atomic' dentro de 'templates'

2.2. El archivo templateDetails.xml

Como se comentó anteriormente, `templateDetails` es un archivo XML que posee toda la información asociada de la plantilla. Su importancia es debido a que es el archivo que lee Joomla al momento de instalar la plantilla y mostrar su información asociada en el panel de administración.

Muchas de las definiciones que se realizaron en el análisis previo van a ir expuestas en este archivo.

El archivo esta conformado de la siguiente manera:

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE install PUBLIC "-//Joomla! 1.6//DTD template 1.0//EN" "http://www.joomla.org/xml/dtd/1.6/template-install.dtd">
<extension
    version="1.6"
    type="template"
    client="site">
    <name>atomic</name>
    <creationDate>10/10/09</creationDate>
    <author>Ron Severdia</author>
    <authorEmail>contact@kontentdesign.com</authorEmail>
    <authorUrl>http://www.kontentdesign.com</authorUrl>
    <copyright>Copyright (C) 2005 - 2011 Open Source Matters, Inc. All rights reserved.</copyright>
    <license>GNU General Public License version 2 or later; see LICENSE.txt</license>
    <version>1.6.0</version>
    <description>TPL_ATOMIC_XML_DESCRIPTION</description>
    <files>
        <folder>html</folder>
        <folder>css</folder>
        <folder>images</folder>
        <folder>language</folder>
        <folder>js</folder>
        <filename>index.php</filename>
        <filename>index.html</filename>
        <filename>favicon.ico</filename>
        <filename>templateDetails.xml</filename>
        <filename>template_preview.png</filename>
        <filename>template_thumbnail.png</filename>
        <filename>component.php</filename>
        <filename>error.php</filename>
    </files>
    <positions>
        <position>atomic-bottomleft</position>
        <position>atomic-bottommiddle</position>
        <position>atomic-search</position>
        <position>atomic-sidebar</position>
        <position>atomic-topmenu</position>
        <position>atomic-topquote</position>
    </positions>
    <!--      For core templates, we also install/uninstall the language files in
            the core language folders.
    -->
    <languages folder="language">
        <language tag="en-GB">en-GB/en-GB.tpl_atomic.ini</language>
        <language tag="en-GB">en-GB/en-GB.tpl_atomic.sys.ini</language>
```

```

</languages>
</extension>
```

Con la información recopilada, el mismo quedará:

```

<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE install PUBLIC "-//Joomla! 1.6//DTD template 1.0//EN" "http://www.
    joomla.org/xml/dtd/1.6/template-install.dtd">
<extension
    version="1.6"
    type="template"
    client="site">
    <name>InterPoint</name>
    <creationDate>2011</creationDate>
    <author>Comunidad Joomla</author>
    <authorEmail>contacto@comunidadjoomla.org</authorEmail>
    <authorUrl>http://www.comunidadjoomla.org/</authorUrl>
    <copyright>Comunidad Joomla 2011</copyright>
    <license>GNU General Public License version 2</license>
    <version>1.0</version>
    <description>TPL_INTERPOINT_XML_DESCRIPTION</description>
    <files>
        <folder>html</folder>
        <folder>css</folder>
        <folder>images</folder>
        <folder>language</folder>
        <folder>js</folder>
        <filename>index.php</filename>
        <filename>index.html</filename>
        <filename>favicon.ico</filename>
        <filename>templateDetails.xml</filename>
        <filename>template_preview.png</filename>
        <filename>template_thumbnail.png</filename>
        <filename>component.php</filename>
        <filename>error.php</filename>
    </files>
    <positions>
        <position>interpoint-menu_ingresar</position>
        <position>interpoint-menu_superior</position>
        <position>interpoint-destacado_superior</position>
        <position>interpoint-video</position>
        <position>interpoint-caja_medio</position>
        <position>interpoint-destacado_inferior</position>
        <position>interpoint-slideshow_inferior</position>
        <position>interpoint-menu_inferior</position>
    </positions>
    <!--          For core templates, we also install/uninstall the language files
        in the core language folders.
    -->
    <languages folder="language">
        <language tag="es-ES">es-ES/es-ES.tpl_interpoint.ini</language>
        <language tag="es-ES">es-ES/es-ES.tpl_interpoint.sys.ini</language>
    </languages>
</extension>
```

Para entender lo modificado, a continuación se muestra un detalle de cada sección correspondiente al archivo:

2.2.1. Sección 1: Información asociada

```
<extension
    version="1.6"
    type="template"
    client="site">
<name>InterPoint</name>
<creationDate>2011</creationDate>
<author>Comunidad Joomla</author>
<authorEmail>contacto@comunidadjoomla.org</authorEmail>
<authorUrl>http://www.comunidadjoomla.org/</authorUrl>
<copyright>Comunidad Joomla 2011</copyright>
<license>GNU General Public License version 2</license>
<version>1.0</version>
<description>TPL_INTERPOINT_XML_DESCRIPTION</description>
```

Las primeras etiquetas corresponden a la información asociada a la plantilla: Nombre, fecha de creación, autor, contacto, licencia, versión y descripción.

Nota

Note que en la descripción se ha puesto TPL_INTERPOINT_XML_DESCRIPTION, que es una referencia que se incorporará en el archivo de lenguaje es-ES.tpl_interpoint.ini. La idea es que todos los textos localizados estén situados en el archivo del idioma, por si en un futuro la plantilla se utilizada en más idiomas.

2.2.2. Sección 2: Carpetas y archivos

```
<files>
    <folder>html</folder>
    <folder>css</folder>
    <folder>images</folder>
    <folder>language</folder>
    <folder>js</folder>
    <filename>index.php</filename>
    <filename>index.html</filename>
    <filename>favicon.ico</filename>
    <filename>templateDetails.xml</filename>
    <filename>template_preview.png</filename>
    <filename>template_thumbnail.png</filename>
    <filename>component.php</filename>
    <filename>error.php</filename>
</files>
```

Luego se especifican los archivos y carpetas que compondrán a la plantilla. Es muy importante, a medida que se avanza con el desarrollo de la plantilla, ir especificando en esta sección los archivos nuevos creados. En caso contrario, si no existe referencia a un archivo, cuando se quiera instalar la plantilla el CMS mostrará un error.

2.2.3. Sección 3: Posiciones de módulos

```
<positions>
    <position>interpoint-menu_ingresar</position>
    <position>interpoint-menu_superior</position>
    <position>interpoint-destacado_superior</position>
```

```

<position>interpoint-video</position>
<position>interpoint-caja_medio</position>
<position>interpoint-destacado_inferior</position>
<position>interpoint-slideshow_inferior</position>
<position>interpoint-menu_inferior</position>
</positions>

```

A continuación se especifican las posiciones admitidas por los módulos en la plantilla. El nombre de cada una suele ser un breve resumen de la posición o funcionalidad específica de dicha zona. Luego, en el archivo del idioma `es-ES.tpl_interpoint.sys.ini` es posible especificar una descripción más explicativa de cada posición.

2.2.4. Sección 4: Definición de lenguajes

```

<languages folder="language">
    <language tag="es-ES">es-ES/es-ES.tpl_interpoint.ini</language>
    <language tag="es-ES">es-ES/es-ES.tpl_interpoint.sys.ini</language>
</languages>

```

Por último se añaden los archivos del idioma, los cuales tendrán la descripción y textos explicativos de las posiciones de módulos y parámetros de la plantilla. El archivo `es-ES.tpl_interpoint.ini` contendrá todas las definiciones que se utilizarán en la pantalla de edición de estilos de plantillas. Mientras que `es-ES.tpl_interpoint.sys.ini` contendrá las explicaciones correspondientes a cada posición de módulos (estos textos aparecen de ayuda, en el momento de asignar un módulo a una posición). Para poder utilizar estos archivos del idioma, lo que se hará es:

- En la carpeta de la plantilla, ir al directorio `language`;
- Allí dentro existe otra carpeta llamada `en-GB`. Cambiarle el nombre por `es-ES`;
- Dentro de esta carpeta existen dos archivos `.ini`: `en-GB.tpl_atomic.ini` y `en-GB.tpl_atomic.sys.ini`. Respectivamente, se los renombra a `es-ES.tpl_interpoint.ini` y `es-ES.tpl_interpoint.sys.ini`;
- Abrir `es-ES.tpl_interpoint.ini`, borrar el contenido y luego escribir:

```
TPL_INTERPOINT_XML_DESCRIPTION="Template de ejemplo para Joomla!"
```

Note que `TPL_INTERPOINT_XML_DESCRIPTION` hace referencia al término incorporado en la etiqueta `<description>` en el archivo `templateDetails.xml`. El texto de descripción se puede cambiar a gusto.

- Guardar `es-ES.tpl_interpoint.ini`;
- Abrir `es-ES.tpl_interpoint.sys.ini`, borrar el contenido y escribir:

```

TPL_INTERPOINT_XML_DESCRIPTION="Template de ejemplo para Joomla!"
TPL_INTERPOINT_POSITION_INTERPOINT-MENU_INGRESAR="Menú superior de ingreso y registro"
TPL_INTERPOINT_POSITION_INTERPOINT-MENU_SUPERIOR="Menú superior principal"
TPL_INTERPOINT_POSITION_INTERPOINT-DESTACADO_SUPERIOR="Texto destacado superior"
TPL_INTERPOINT_POSITION_INTERPOINT-VIDEO="Caja de video"
TPL_INTERPOINT_POSITION_INTERPOINT-CAJA_MEDIO="Caja del medio"
TPL_INTERPOINT_POSITION_INTERPOINT-DESTACADO_INFERIOR="Texto destacado inferior"
TPL_INTERPOINT_POSITION_INTERPOINT-SLIDESHOW_INFERIOR="Slideshow inferior"
TPL_INTERPOINT_POSITION_INTERPOINT-MENU_INFERIOR="Menú inferior"

```

- Guardar `es-ES.tpl_interpoint.sys.ini`.

Nota

Note que cada referencia de descripción de módulos esta compuesta por `TPL_(nombre de la plantilla)_POSITION_(nombre de la posición del módulo)`. También se ha agregado nuevamente la descripción de la plantilla con la misma referencia que se usó anteriormente. Los textos de descripción de cada posición se pueden cambiar al gusto.

Nota

En el momento de instalar la plantilla, los archivos `es-ES.tpl_interpoint.ini` y `es-ES.tpl_interpoint.sys.ini` se copiarán dentro de la carpeta `language/es-ES`, en el directorio raíz del CMS.

Nota

En el archivo `templateDetails.xml` también se especifican las opciones de personalización de la plantilla. Más adelante se explicará como añadir esta característica y de que forma.

Con esto listo, es posible continuar con la visualización del template en la administración de Joomla.

2.2.5. Descubrir la plantilla en la administración de CMS

Antes de continuar, es importante que Joomla reconozca la copia realizada de la plantilla, ya que será necesario poder previsualizar el trabajo a medida que se va avanzando.

Para esto, en la administración, vaya a **Extensiones > Gestor de extensiones > Descubrir > Presionar el botón Descubrir**. Si todo sale bien, debería aparecer la plantilla enlistada.

Nombre	Tipo	Versión	Fecha	Carpeta	Cliente	Autor	ID
InterPoint	Plantilla	1.0	2011	No aplicable	Sitio	Comunidad Joomla	10010

Este apartado le permite descubrir extensiones que no han sido agregadas usando el proceso normal de instalación.
Por ejemplo, algunas extensiones son demasiado grandes para ser subidas usando la interfaz web, debido a las restricciones impuestas por quien sirve el hospedaje. Usando esta característica podrá subir los archivos de la extensión directamente al directorio de su servidor web usando otros medios del tipo FTP o SFTP, para poner en el lugar apropiado los archivos de esa extensión.
Por lo tanto, puede usar esta característica para encontrar alguna extensión subida recientemente y activarla en su instalación de Joomla!
Usando esta utilidad también se pueden descubrir e instalar múltiples extensiones al mismo tiempo.

Figura 2.4: Descubrir la plantilla en la administración de CMS

Lo siguiente será seleccionar el ítem de la plantilla y presionar el botón **Instalar**. Si todo salió bien debe aparecer el texto “**La instalación de la extensión descubierta se ha realizado correctamente**” y más abajo la descripción de la plantilla.

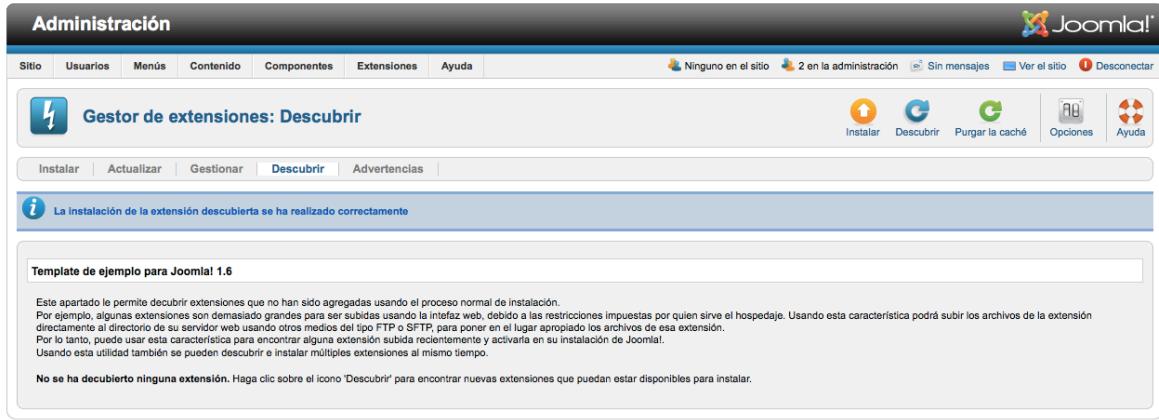


Figura 2.5: Pantalla de instalación correcta

Finalmente, para poder visualizar la plantilla en el sitio, vaya a **Extensiones > Gestor de plantillas** (Figura 2.6).

Seleccionar el ítem de la plantilla nueva y presionar el botón **Predeterminada**. De esta forma se logrará que al ingresar al sitio se cargue la plantilla.

2.3. El archivo index.php

Al ser uno de los archivos más importantes, es fundamental entender las etiquetas que conforman al `index.php`. A continuación se mostrará el archivo completo para luego ir viendo en detalle cada sección que lo conforma.

```
<?php
/**
 * @version      $Id: index.php 20196 2011-01-09 02:40:25Z ian $
 * @package      Joomla.Site
 * @copyright    Copyright (C) 2005 - 2011 Open Source Matters, Inc. All rights reserved.
 * @license      GNU General Public License version 2 or later; see LICENSE.txt
 */

defined('_JEXEC') or die;

/* The following line loads the MooTools JavaScript Library */
JHTML::_('behavior.framework', true);

/* The following line gets the application object for things like displaying the site name */
$app = JFactory::getApplication();
```

Estilo	Localización	Plantilla	Predeterminado	Asignado	ID
Atomic - Default	Sitio	atomic	★	✓	3
Beez5 - Default-Fruit Shop	Sitio	beez5	★	✓	6
Beez2 - Parks Site	Sitio	beez_20	★	✓	114
Beez2 - Default	Sitio	beez_20	★	✓	4
Bluestork - Default	Administrador	bluestork	★		2
Hathor - Default	Administrador	hathor	★		5
InterPoint - Predeterminado	Sitio	InterPoint	★		116

Figura 2.6: La nueva plantilla aparece enlistada

```

?>
<?php echo '<?' ; ?>xml version="1.0" encoding="<?php echo $this->_charset ?>"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/
xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="<?php echo $this->language;
?>" lang="<?php echo $this->language; ?>" dir="<?php echo $this->direction;
?>" >
<head>
<!-- The following JDOC Head tag loads all the header and meta information
from your site config and content. --&gt;
&lt;jdoc:include type="head" /&gt;

<!-- The following five lines load the Blueprint CSS Framework (http://
blueprintcss.org). If you don't want to use this framework, delete these
lines. --&gt;
&lt;link rel="stylesheet" href="&lt;?php echo $this-&gt;baseurl ?&gt;/templates/&lt;?php
echo $this-&gt;template ?&gt;/css/blueprint/screen.css" type="text/css" media="
screen, projection" /&gt;
&lt;link rel="stylesheet" href="&lt;?php echo $this-&gt;baseurl ?&gt;/templates/&lt;?php
echo $this-&gt;template ?&gt;/css/blueprint/print.css" type="text/css" media="
print" /&gt;
&lt;!--[if lt IE 8]&gt;&lt;link rel="stylesheet" href="blueprint/ie.css" type="text/
css" media="screen, projection"&gt;&lt;![endif]--&gt;
&lt;link rel="stylesheet" href="&lt;?php echo $this-&gt;baseurl ?&gt;/templates/&lt;?php
echo $this-&gt;template ?&gt;/css/blueprint/plugins/fancy-type/screen.css" type
="text/css" media="screen, projection" /&gt;
&lt;link rel="stylesheet" href="&lt;?php echo $this-&gt;baseurl ?&gt;/templates/&lt;?php
echo $this-&gt;template ?&gt;/css/blueprint/plugins/joomla-nav/screen.css"
type="text/css" media="screen" /&gt;
</pre>

```

```

<!-- The following line loads the template CSS file located in the template
    folder. -->
<link rel="stylesheet" href="php echo $this-&gt;baseurl ?&gt;/templates/<?php
    echo $this-&gt;template ?&gt;/css/template.css" type="text/css" /&gt;

&lt;!-- The following four lines load the Blueprint CSS Framework and the
    template CSS file for right-to-left languages. If you don't want to use
    these, delete these lines. --&gt;
&lt;?php if($this-&gt;direction == 'rtl') : ?&gt;
    &lt;!--link rel="stylesheet" href="<?php echo $this-&gt;baseurl ?&gt;/templates/<?php
        echo $this-&gt;template ?&gt;/css/blueprint/plugins/rtl/screen.css" type
        ="text/css" /&gt;
    &lt;!--link rel="stylesheet" href="<?php echo $this-&gt;baseurl ?&gt;/templates/<?php
        echo $this-&gt;template ?&gt;/css/template rtl.css" rel="stylesheet" type
        ="text/css" /&gt;
&lt;?php endif; ?&gt;

&lt;!-- The following line loads the template JavaScript file located in the
    template folder. It's blank by default. --&gt;
&lt;script type="text/javascript" src="<?php echo $this-&gt;baseurl ?&gt;/templates/<?php
    echo $this-&gt;template ?&gt;/js/template.js"&gt;&lt;/script&gt;
&lt;/head&gt;
&lt;body&gt;
    &lt;div class="container"&gt;
        &lt;hr class="space" /&gt;
        &lt;div class="joomla-header span-16 append-1"&gt;
            &lt;h1&gt;&lt;?php echo $app-&gt;getCfg('sitename'); ?&gt;&lt;/h1&gt;
        &lt;/div&gt;
        &lt;?php if($this-&gt;countModules('atomic-search')) : ?&gt;
            &lt;div class="joomla-search span-7 last"&gt;
                &lt;jdoc:include type="modules" name="atomic-search" style="none" /&gt;
            &lt;/div&gt;
        &lt;?php endif; ?&gt;
    &lt;/div&gt;
    &lt;?php if($this-&gt;countModules('atomic-topmenu')) : ?&gt;
        &lt;jdoc:include type="modules" name="atomic-topmenu" style="container" /&gt;
    &lt;?php endif; ?&gt;

    &lt;div class="container"&gt;
        &lt;div class="span-16 append-1"&gt;
            &lt;?php if($this-&gt;countModules('atomic-topquote')) : ?&gt;
                &lt;jdoc:include type="modules" name="atomic-topquote" style="none" /&gt;
            &lt;?php endif; ?&gt;
                &lt;jdoc:include type="message" /&gt;
                &lt;jdoc:include type="component" /&gt;
                &lt;hr /&gt;
            &lt;?php if($this-&gt;countModules('atomic-bottomleft')) : ?&gt;
                &lt;div class="span-7 colborder"&gt;
                    &lt;jdoc:include type="modules" name="atomic-bottomleft" style="
                        bottommodule" /&gt;
                &lt;/div&gt;
            &lt;?php endif; ?&gt;

            &lt;?php if($this-&gt;countModules('atomic-bottommiddle')) : ?&gt;
        &lt;div class="span-7 last"&gt;
</pre

```

```

<jdoc:include type="modules" name="atomic-bottommiddle" style="bottommodule" />
</div>
<?php endif; ?>
</div>
<?php if($this->countModules('atomic-sidebar')) : ?>
<div class="span-7 last">
<jdoc:include type="modules" name="atomic-sidebar" style="sidebar" />
</div>
<?php endif; ?>

<div class="joomla-footer span-16 append-1">
<hr />
&copy;<?php echo date('Y'); ?> <?php echo $app->getCfg('sitename'); ?>
</div>
</div>
</body>
</html>

```

2.3.1. Sección 1: Sesión y contexto del CMS

```
defined('_JEXEC') or die;
```

La línea indicada podrá encontrarla en muchos archivos fuente de Joomla. Se incorpora por motivos de seguridad, su objetivo es asegurar que el archivo PHP está siendo ejecutado dentro de la sesión y contexto del CMS (de forma tal que si un usuario quiere vulnerar el sitio ejecutando un archivo en particular, le sea difícil realizarlo).

Nota

Más información sobre JEXEC: <http://docs.joomla.org/JEXEC>

2.3.2. Sección 2: Framework JavaScript

```
JHTML::_('behavior.framework', true);
```

Para crear varias funcionalidades JavaScript, Joomla utiliza el *framework Mootools*. Dicho framework está compuesto por varios archivos .js que se cargan de forma predeterminada en la cabecera de la plantilla (<head />). Al incorporar la línea mostrada con el argumento **true**, se especifica que también se cargue en la plantilla el archivo **mootools-more.js**, el cual posee diversas funciones que potencian aún más al framework JavaScript.

Nota

Note que si quita la línea de código en **index.php**, Mootools seguirá cargándose en la plantilla.

Nota

Más información sobre Mootools y Mootools More: <http://mootools.net/> y <http://mootools.net/more/>

2.3.3. Sección 3: Referencia a JApplication

```
$app = JFactory::getApplication();
```

Esta línea guarda en la variable \$app una referencia al objeto PHP de Joomla JApplication. Sin entrar en detalles técnicos, utilizando esta línea es posible obtener información del sitio para mostrar dentro de la plantilla (por ejemplo: nombre del sitio, título de la página, nombre del *template*, etc).

Nota

Puede consultar la información relacionada con JApplication en <http://api.joomla.org/Joomla-Framework/Application/JApplication.html>

2.3.4. Sección 4: Tipo de documento, codificación y estilo de escritura

```
<?php echo '<?' ; ?>xml version="1.0" encoding="<?php echo $this->_charset ?>"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/
xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="<?php echo $this->language;
?>" lang="<?php echo $this->language; ?>" dir="<?php echo $this->direction;
?>" >
```

El siguiente bloque de código declara el **tipo de documento** (**Doctype**) de la plantilla, así como también la codificación, idioma y **dirección de escritura** (de izquierda a derecha o viceversa). \$this es una referencia al objeto PHP de Joomla JDocumentHTML, el cual posee información necesaria para crear el documento HTML (archivos cargados, información de etiquetas meta, idioma, etc).

Nota

Puede consultar la información relacionada a JDocumentHTML en <http://docs.joomla.org/JDocumentHTML>

2.3.5. Sección 5: Cabecera del documento

```
<jdoc:include type="head" />
```

El código mostrado crea e incorpora las etiquetas necesarias para llenar la cabecera de la plantilla:

- Etiqueta <base />;
- Etiquetas <meta />:
 - content-type;
 - robots;
 - keywords;
 - rights;
 - language;
 - generator;
- Etiqueta <title />;
- Etiquetas <link /> para cargar estilos CSS predeterminados, *favicon* y *RSS*;
- Etiquetas <script /> para cargar archivos y funciones JavaScript predeterminados.

A algunas etiquetas meta, Joomla permite desactivarlas o editarlas, mientras que a otras no. Por otro lado, dependiendo si se está utilizando una extensión, Joomla puede cargar más archivos CSS y JavaScript dentro de la cabecera.

2.3.6. Sección 6: Archivos CSS

```
<link rel="stylesheet" href="php echo $this-&gt;baseurl ?&gt;/templates/&lt;?php echo<br/$this->template ?>/css/blueprint/screen.css" type="text/css" media="screen,  
projection" />  
<link rel="stylesheet" href="php echo $this-&gt;baseurl ?&gt;/templates/&lt;?php echo<br/$this->template ?>/css/blueprint/print.css" type="text/css" media="print" />  
<!--[if lt IE 8]><link rel="stylesheet" href="blueprint/ie.css" type="text/css"  
media="screen, projection"><![endif]-->  
<link rel="stylesheet" href="php echo $this-&gt;baseurl ?&gt;/templates/&lt;?php echo<br/$this->template ?>/css/blueprint/plugins/fancy-type/screen.css" type="text/  
css" media="screen, projection" />  
<link rel="stylesheet" href="php echo $this-&gt;baseurl ?&gt;/templates/&lt;?php echo<br/$this->template ?>/css/blueprint/plugins/joomla-nav/screen.css" type="text/  
css" media="screen" />
```

A continuación se incorporan los archivos pertenecientes al framework CSS *Blueprint*:

- **screen.css**: Posee estilos de reseteo, tipográficos, para formularios y para crear maquetaciones a través de grillas. Más adelante se explicará el funcionamiento de las grillas utilizando *Blueprint*;
- **print.css**: Estilos que mejoran el formateo del sitio al momento de imprimir;
- **ie.css**: Estilos específicos para *Internet Explorer*. La etiqueta `[if lt IE 8]` significa que el estilo se cargará únicamente para versiones anteriores a *Internet Explorer 8*;
- **/plugins/fancy-type/screen.css**: Incorpora algunos estilos extras para la manipulación de tipografías en textos;
- **/plugins/joomla-nav/screen.css**: Estilos específicos para el menú de navegación horizontal del sitio.

Nota

Más adelante se explicará la utilización de Blueprint

En esta parte del código, `$this` se utiliza para completar el valor del atributo `href` hacia los archivos CSS: primero obteniendo la URL del sitio (`$this->baseurl`) y luego el nombre de la plantilla configurada para utilizar (`$this->template`).

2.3.7. Sección 7: Archivos CSS para otras direcciones de escritura

```
<?php if($this->direction == 'rtl') : ?>  
    <link rel="stylesheet" href="php echo $this-&gt;baseurl ?&gt;/templates/&lt;?php<br/    echo $this->template ?>/css/blueprint/plugins/rtl/screen.css" type="text/  
    css" />  
    <link rel="stylesheet" href="php echo $this-&gt;baseurl ?&gt;/templates/&lt;?php<br/    echo $this->template ?>/css/template rtl.css" rel="stylesheet" type="text/  
    css" />  
<?php endif; ?>
```

El siguiente trozo de código tiene un solo objetivo: Leer la configuración del CMS y determinar la dirección de lectura. En caso que el valor sea `rtl` (de derecha a izquierda o *right to left* en inglés) se cargan los estilos apropiados. Si el sitio tendrá un sentido de lectura de izquierda a derecha, este trozo de código se puede eliminar.

Nota

Note que el bloque condicional `if` no tiene la sintaxis clásica que se suele utilizar:

```
if () {  
...  
} else {  
...  
}
```

Esto es porque se utiliza una sintaxis alternativa que permite mejorar la lectura del código y su funcionamiento es exactamente el mismo que la manera clásica. Más información: <http://www.php.net/manual/es/control-structures.alternative-syntax.php>

2.3.8. Sección 8: Funcionalidades JavaScript

```
<script type="text/javascript" src="php echo $this-&gt;baseurl ?&gt;/templates/<?php<br/echo $this->template ?>/js/template.js"></script>
```

La última línea de código incorporada dentro de la cabecera del sitio es la llamada hacia el archivo `template.js`. Si se revisa el archivo es posible notar que el mismo está vacío. La idea es que todas las funciones JavaScript que se quieran incorporar dentro de la plantilla sean incorporadas allí dentro.

Al igual que con las llamadas a los archivos CSS, se utiliza `$this` para completar el atributo `src`, primero obteniendo la URL del sitio y luego el nombre de la plantilla utilizada.

2.3.9. Sección 9: Nombre del sitio

```
<div class="joomla-header span-16 append-1">  
    <h1><?php echo $app->getCfg('sitename'); ?></h1>  
</div>
```

Ya dentro del cuerpo del documento, entre etiquetas H1 se utiliza `$app` para mostrar el nombre del sitio. Dicho nombre es obtenido de la configuración del CMS.

2.3.10. Sección 10: Llamado a distintos módulos

```
<?php if($this->countModules('atomic-search')) : ?>  
    <div class="joomla-search span-7 last">  
        <jdoc:include type="modules" name="atomic-search" style="none" />  
    </div>  
<?php endif; ?>
```

El siguiente trozo de código se repite varias veces en el cuerpo del documento, por lo tanto es importante entender su funcionamiento:

- `$this->countModules()`: Esta función lo que hace es contar la cantidad de módulos publicados en una determinada posición de la plantilla. En el ejemplo mostrado, se pregunta cuantos módulos están publicados en la posición `atomic-search`, en caso de ser igual o mayor a 1, se ejecuta el código siguiente. Esto se suele hacer para que no se muestre código HTML vacío en caso que un módulo no esté publicado (como es el caso de las etiquetas `<div />` en el código que se muestra).

Nota

Más información sobre `countModules()`: <http://docs.joomla.org/JDocumentHTML/countModules>

- `<jdoc:include type="modules" name="" style="" />`: Representa una de las partes más importantes de la plantilla. Con esta directiva se indica la carga de un módulo en particular y de una manera específica. Esto se establecerá a partir de dos atributos:

- *name*: En este atributo se debe especificar el nombre de una posición determinada de la plantilla. Estas posiciones son especificadas en el archivo `templateDetails.xml` y su nombre debe ser único, sin espacios y lo suficiente descriptivo para entender que tipo de módulo debería ir en esa zona. Por ejemplo, el `atomic-search` indica que esa zona de la plantilla es adecuada para mostrar el buscador del sitio (sin embargo, esto no es obligatorio, es posible poner el módulo que se desee en dicha posición).
- *style*: En este atributo de especifica el estilo con que se quiere mostrar un módulo determinado. Dicho estilo no se refiere a CSS, sino a qué tipos de etiquetas HTML encerrarán al módulo y de qué manera. De forma predeterminada, Joomla! cuenta con seis estilos para mostrar un módulo:
 - `xhtml`: El módulo estará encerrado por solo una etiqueta `<div />` la cual tendrá como atributo la clase `moduletable+sufijo_del_modulo`. El sufijo es un parametro que se puede especificar en cada módulo dentro del panel de administración y sirve basicamente para diferenciar en estilos CSS a los diferentes módulos.
 - `rounded`: El módulo estará encerrado dentro de 4 etiquetas `<div />`, las cuales permiten (añadiendo luego estilos CSS) un diseño con bordes redondeados al módulo; El primer `<div />` padre tendrá como atributo la clase `module+sufijo_del_modulo`.
 - `table` y `horz`: Utilizando cualquiera de estas dos formas, el módulo estará encerrado dentro de una tabla HTML.
 - `none`: El módulo aparecerá “en crudo”, es decir, sin ningún tipo de etiquetas encerrándolo, solo el HTML que compone al mismo;
 - `outline`: Este estilo se suele utilizar para funciones de depuración. El mismo permite mostrar una linea punteada delimitando el tamaño del módulo para notar su tamaño y espacio ocupado en la plantilla.

Nota

Puede encontrar más información sobre los diferentes estilos de presentación de módulos en http://docs.joomla.org/What_is_module_chrome%3F

A su vez, Joomla! permite incorporar estilos personalizados para mostrar módulos. Estos estilos deben ir especificados en el archivo `modules.php` dentro de la carpeta `html` de la plantilla.

Nota

Más adelante se explicará como añadir diferentes estilos de presentación de módulos.

Los módulos no suelen ocupar posiciones principales de la plantilla, sino más bien son elementos funcionales al sitio: Menús de navegación, cajas de encuestas, formularios de ingreso, etc. La posición principal de la plantilla, por lo general, es ocupada por el contenido de los distintos componentes del CMS (artículos de noticias, secciones y categorías, etc).

Otros trozos de código que incorporan posiciones de módulos en `index.php`:

```

<?php if($this->countModules('atomic-topmenu')) : ?>
    <jdoc:include type="modules" name="atomic-topmenu" style="container" />
<?php endif; ?>
<?php if($this->countModules('atomic-topquote')) : ?>
    <jdoc:include type="modules" name="atomic-topquote" style="none" />
<?php endif; ?>
<?php if($this->countModules('atomic-bottomleft')) : ?>
    <div class="span-7 colborder">
        <jdoc:include type="modules" name="atomic-bottomleft" style="bottommodule"
            " />
    </div>
<?php endif; ?>
<?php if($this->countModules('atomic-bottommiddle')) : ?>
    <div class="span-7 last">
        <jdoc:include type="modules" name="atomic-bottommiddle" style=""
            bottommodule" />
    </div>
<?php endif; ?>
<?php if($this->countModules('atomic-sidebar')) : ?>
    <div class="span-7 last">
        <jdoc:include type="modules" name="atomic-sidebar" style="sidebar" />
    </div>
<?php endif; ?>

```

Nota

Note que varios trozos de código utilizan estilos personalizados: container, bottommodule y sidebar. Más adelante se explicará la creación de estilos personalizados de módulos.

2.3.11. Sección 11: Mensajes del sistema

```
<jdoc:include type="message" />
```

Esta directiva debe aparecer sólo una vez en la plantilla. Se utiliza para mostrar diferentes mensajes del CMS (por ejemplo: cuando un usuario se registra en el sitio, el CMS le informa que el registro fué satisfactorio o erróneo). Dichos mensajes aparecen en la zona en donde se incorpore la directiva.

2.3.12. Sección 12: Contenido de los componentes

```
<jdoc:include type="component" />
```

Como se comentó anteriormente, el contenido de los componentes del CMS suelen ocupar la zona principal de la plantilla. Esta línea justamente tiene esa finalidad: permite mostrar el contenido del componente en que se este situado (artículos, categorías, formulario de registro, etc).

2.3.13. Sección 13: Año presente y nombre del sitio

```
<div class="joomla-footer span-16 append-1">
    <hr />
    &copy;<?php echo date('Y'); ?> <?php echo $app->getCfg('sitename'); ?>
</div>
```

Finalmente, llegando al final del archivo, se incorporan una serie de etiquetas para mostrar el año presente y el nombre del sitio, formando un texto similar a: **® 2011 Mi Sitio Web.**

Lo siguiente a realizar es la modificación de `index.php` para crear una maquetación en base a las necesidades requeridas. Como se utilizará a **BluePrint** como herramienta, es necesario primero entender su funcionamiento.

2.4. El sistema de grillas de BluePrint

BluePrint es una colección de archivos `.css`, los cuales poseen variados estilos predefinidos para ahorrar tiempo en diferentes tareas del diseño web: Maquetaciones, reseteo de estilos, diseño de formularios, tipografías, etc. Para la creación de la maquetación de la plantilla se utilizará su sistema de grillas.

Nota

Puede obtener más información sobre **BluePrint** en su sitio web: <http://www.blueprintcss.org/>

El sistema de grillas de **BluePrint** es muy fácil de utilizar, permitiendo agilizar de forma severa el proceso de maquetación. Su mecanismo permite crear un máximo de 24 columnas (o grillas) las cuales se pueden combinar dependiendo de las necesidades.

La clave de su utilización es añadir una serie de clases predefinidas en los elementos del documento para poder crear las columnas y el ancho necesario.

Por ejemplo, si se desea crear dos columnas y que cada una utilice el espacio de ocho grillas, se deberá incorporar en el cuerpo del documento:

```
<body>
  <div class="container">

    <div class="span-12">
      ...
    </div>

    <div class="span-12 last">
      ...
    </div>
  </div>
</body>
```

El `<div />` con la clase `container` será el elemento que contendrá todas las columnas de la maquetación. Al agregar dicha clase, se hará que el documento tenga un ancho de `950px` y aparezca centrada horizontalmente. Luego, cada `<div />` con la clase `span-12` especifica que ocupen el espacio de 12 grillas, o sea, la mitad del espacio disponible (recordar que como máximo se dispone de 24 grillas). De la misma forma, si se desea que cada columna ocupe un tercio del espacio, lo único que hay que hacer es cambiar la clase por `span-8`. Finalmente note que el último elemento (es decir, la última columna) debe tener la clase `last`, caso contrario la maquetación se visualizará de forma incorrecta.

También existen otras clases que permiten modificar la disposición de las columnas: `prepend-x` y `append-x`. La primera permite mover una columna más hacia la derecha, mientras que la otra más hacia la izquierda, todo dependerá del valor que se le asigne a `x` (por ejemplo: `prepend-4` / `append-4`).

2.5. Maquetación del diseño

Ya explicado el sistema de grillas de BluePrint, se comenzará la modificación de `index.php` para agregar la maquetación del diseño.

2.5.1. Modificaciones dentro de la etiqueta head

El `<header />` quedará prácticamente igual a como estaba antes, con algunas modificaciones:

- Se eliminan algunos comentarios y se agregan otros;
- Se quitan las llamadas a los estilos `/plugins/fancy-type/screen.css` y `/plugins/joomla-nav/screen.css` ya que no se utilizarán;
- También se eliminan las llamadas a los estilos utilizados para adaptar los textos al sentido de lectura de derecha a izquierda (*RTL*).

```
<?php
defined('_JEXEC') or die;
JHTML::_('behavior.framework', true);
$app = JFactory::getApplication();
?>
<?php echo '<?' : ?>xml version="1.0" encoding="<?php echo $this->_charset ?>"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/
    xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="<?php echo $this->language;
    ?>" lang="<?php echo $this->language; ?>" dir="<?php echo $this->direction;
    ?>">
<head>
    <jdoc:include type="head" />
    <link rel="stylesheet" href="<?php echo $this->baseurl ?>/templates/<?php
        echo $this->template ?>/css/blueprint/screen.css" type="text/css" media="
            screen, projection" />
    <link rel="stylesheet" href="<?php echo $this->baseurl ?>/templates/<?php
        echo $this->template ?>/css/blueprint/print.css" type="text/css" media="
            print" />
    <!--[if lt IE 8]><link rel="stylesheet" href="blueprint/ie.css" type="text/
        css" media="screen, projection"><![endif]-->
    <link rel="stylesheet" href="<?php echo $this->baseurl ?>/templates/<?php
        echo $this->template ?>/css/template.css" type="text/css" />
    <script type="text/javascript" src="<?php echo $this->baseurl ?>/templates/<?
        php echo $this->template ?>/js/template.js"></script>
</head>
```

2.5.2. Maquetación de la sección superior

Dentro del cuerpo del documento, la sección superior se dividirá en dos columnas: una dedicada al logo y otra dedicada a los dos menús:

```
<body>
    <div class="container">
        <!-- Comienzo header -->
```

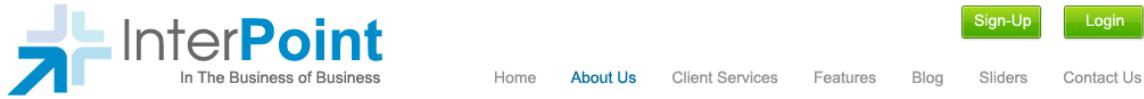


Figura 2.8: Diseño de la sección superior

```

<div id="header" class="span-24">

    <!-- Comienzo Logo -->
    <div id="logo" class="span-9">
        <a href=<?php echo $this->baseurl ?>" title=<?php echo $app->
            getCfg('sitename'); ?>">
            <h1><?php echo $app->getCfg('sitename'); ?></h1>
        </a>
        <h2 id="eslogan">Lorem ipsum dolor sit amet</h2>
    </div>
    <!-- Fin Logo -->

    <!-- Comienzo Menús Superiores -->
    <div id="menus_superiores" class="span-15 last">

        <?php if($this->countModules('interpoint-menu_ingresar')) : ?>
            <div id="menu_ingresar">
                <jdoc:include type="modules" name="interpoint-
                    menu_ingresar" style="none" />
            </div>
        <?php endif; ?>

        <?php if($this->countModules('interpoint-menu_superior')) : ?>
            <div id="menu_superior">
                <jdoc:include type="modules" name="interpoint-
                    menu_superior" style="none" />
            </div>
        <?php endif; ?>

    </div>
    <!-- Fin Menús superiores -->

</div>
<!-- Fin Header -->
</div>
</body>

```

2.5.3. Maquetación del texto superior, video y caja destacada

En esta sección se tendrá en primer lugar dos columnas y luego una columna ocupando el total del espacio:

```

<!-- Mensajes del CMS -->
<jdoc:include type="message" />

<!-- Comienzo Destacado superior -->
<?php if($this->countModules('interpoint-destacado_superior')) : ?>


```

Developing Business Solutions

Etiam vitae augue nunc. Quisque tincidunt feugiat vestibulum. Vestibulum rhoncus neque ligula. Proin eu erat est, in placerat mauris. Nam justo erat, pellentesque a auctor quis, elementum at urna. Phasellus varius, arcu eget semper ultrices, mauris enim molestie nulla, a sollicitudin

[Watch The Demo](#)



Sign-up for our free newsletter for all of our updates at InterPoint

Email Address

Submit

Figura 2.9: Diseño del texto superior, video y caja destacada

```
<div id="destacado_superior" class="span-10">
    <jdoc:include type="modules" name="interpoint-destacado_superior" style="xhtml" />
</div>
<?php endif; ?>
<!-- Fin Destacado superior -->

<!-- Comienzo Video -->
<?php if($this->countModules('interpoint-video')) : ?>
    <div id="video" class="span-14 last">
        <div id="reproductor">
            <jdoc:include type="modules" name="interpoint-video" style="none" />
        </div>
    </div>
<?php endif; ?>
<!-- Fin Video -->

<!-- Comienzo Caja medio -->
<?php if($this->countModules('interpoint-caja_medio')) : ?>
    <div id="caja_medio" class="span-24">
        <jdoc:include type="modules" name="interpoint-caja_medio" style="none" />
    </div>
<?php endif; ?>
<!-- Fin Caja medio -->
```

Nota

Note que se ha agregado `<jdoc:include type="message" />` para que aparezcan los mensajes del sistema debajo de cabecera del diseño.



Solutions Made Easy

Suspendisse vulputate fringilla arcu et porttitor. Mauris placerat mattis tellus id sodales. Morbi vitae ipsum lectus, ullamcorper eleifend orci.

Vivamus pulvinar hendrerit dolor, vel scelerisque sapien dignissim non. In hac habitasse platea dictumst.

[>> Read More](#)



We Listen To You

Suspendisse vulputate fringilla arcu et porttitor. Mauris placerat mattis tellus id sodales. Morbi vitae ipsum lectus, ullamcorper eleifend orci.

Vivamus pulvinar hendrerit dolor, vel scelerisque sapien dignissim non. In hac habitasse platea dictumst.

[>> Read More](#)



Monitoring your Output

Suspendisse vulputate fringilla arcu et porttitor. Mauris placerat mattis tellus id sodales. Morbi vitae ipsum lectus, ullamcorper eleifend orci.

Vivamus pulvinar hendrerit dolor, vel scelerisque sapien dignissim non. In hac habitasse platea dictumst.

[>> Read More](#)

Figura 2.10: Diseño de las columnas del medio

2.5.4. Maquetación de las columnas del medio

Las columnas del medio cargarán tres artículos publicados en la página principal del sitio. Para eso se utilizará la etiqueta `<jdoc:include type="component" />`:

```
<!-- Comienzo Contenido principal -->
<div id="contenido" class="span-24">
    <jdoc:include type="component" />
</div>
<!-- Fin Contenido principal -->
```

Para hacer que cada artículo ocupe una columna, en la administración del CMS, ir a **Menús > Menú principal** y luego acceder al ítem que se tenga marcado como página de inicio.

En la columna derecha, en el apartado **Opciones de presentación** establecer los siguientes valores.

De esta forma se hará que se muestren, como máximo, tres artículos, separados cada uno en columnas. Al mostrar dichos artículos, el HTML generado por el CMS será:

```
<div class="span-24" id="contenido">

    <div class="blog-featured">

        <div class="items-leading">

            <div class="leading-0">

                <h2>
                    <a href="#link">Título del artículo 1</a>
                </h2>

                <p>Texto del artículo 1...</p>

                <p class="readmore">
                    <a href="#link">Leer más...</a>
                </p>

                <div class="item-separator"></div>

            </div>

        </div>

    </div>

</div>
```

Administración

Sitio Usuarios Menús Contenido Componentes Extensiones Ayuda Ninguno en el sitio 2 en la administración Sin mensajes Ver el sitio Desconectar

Gestor de menús: Editar elemento del menú

Detalles

Tipo de elemento del menú * Mostrar todos los artículos destacados Seleccionar

Título del elemento del menú * Inicio

Alias homepage

Nota

Enlace index.php?option=com_content&view=featu

Estado Publicado

Acceso Público

Menú * Menú principal

Elemento principal Elemento de menú raíz

Abrir en Misma ventana

Página de inicio Sí No

Idioma Todos

Estilo de la plantilla - Usar la predeterminada -

ID 435

Opciones de presentación

Si se deja algún campo en blanco se usará lo asignado desde la configuración

Cantidad de artículos de introducción 3

Cantidad de artículos después de los de introducción 0

Cantidad de columnas 3

Cantidad de enlaces 0

Orden en columnas múltiples Usar el valor global

Orden de las categorías Usar el valor global

Orden del artículo Usar el valor global

Ordenar por fecha Usar el valor global

Paginación Ocultar

Resultados de la paginación Usar el valor global

Opciones del artículo

Opciones de integración

Opciones del tipo de enlace

Opciones de visionado en la página

Opciones de los metadatos

Asignación de módulos para este elemento del menú

Pack traducido y readaptado por Comunidad Joomla! | Joomla!® es software libre liberado bajo la GNU General Public License. Versión 1.6.1

Figura 2.11: Sección de edición de elemento del menú

▼ Opciones de presentación

Si se deja algún campo en blanco se usará lo asignado desde la co

Cantidad de artículos de introducción

Cantidad de artículos después de los de
introducción

Cantidad de columnas

Cantidad de enlaces

Orden en columnas múltiples ▾

Orden de las categorías ▾

Orden del artículo ▾

Ordenar por fecha ▾

Paginación ▾

Resultados de la paginación ▾

Figura 2.12: *Opciones de presentación*

```

<div class="leading-1">

    <h2>
        <a href="#link">Título del artículo 2</a>
    </h2>

    <p>Texto del artículo 2...</p>

    <p class="readmore">
        <a href="#link">Leer más...</a>
    </p>

    <div class="item-separator"></div>

</div>

<div class="leading-2">

    <h2>
        <a href="#link">Título del artículo 3</a>
    </h2>

    <p>Texto del artículo 3...</p>

    <p class="readmore">
        <a href="#link">Leer más...</a>
    </p>

    <div class="item-separator"></div>

</div>

</div>

</div>

```

Por lo tanto, Joomla se encarga de generar el HTML que necesitamos para poder lograr las columnas del medio.

Nota

Otra solución para generar tres columnas sería utilizar tres módulos separados y crear por nuestra cuenta el HTML de cada columna.

2.5.5. Maquetación del texto destacado inferior y slideshow

Para la siguiente parte del diseño, se crearán dos columnas, una para el texto destacado y otra para la zona del slideshow.

```

<!-- Comienzo Destacado inferior -->
<div id="destacado_inferior" class="span-15 append-1">
    <jdoc:include type="modules" name="interpoint-destacado_inferior" style="
        xhtml" />

```

Simple Ideas For Simple Business Plans

Ut fermentum aliquam luctus. Vestibulum et dictum augue. Integer felis purus, bibendum at mollis non, eleifend vitae lacus. Etiam non nulla mi, a pharetra erat. Vivamus aliquam elit laoreet sem euismod ultrices.

Aenean nibh dolor, laoreet et sollicitudin imperdiet, pulvinar ut augue. Mauris tempor interdum elit. Etiam nec mauris a ante facilisis iaculis. Integer in tortor ut urna dapibus ultrices et nec mi.

[Read More](#)



Figura 2.13: Diseño del texto destacado inferior y slideshow

```
</div>
<!-- Fin Destacado inferior -->

<!-- Comienzo Slideshow inferior -->
<div id="slideshow_inferior" class="span-8 last">
    <?php if($this->countModules('interpoint-slideshow_inferior')) : ?>
        <div id="slideshow_contenedor">
            <jdoc:include type="modules" name="interpoint-slideshow_inferior"
                style="none" />
        </div>
        <span id="slideshow_der" class="slideshow_nav"></span>
        <span id="slideshow_izq" class="slideshow_nav"></span>
    <?php endif; ?>
</div>
<!-- Fin Slideshow inferior -->
```

Dentro del `<div>` que va a contener el slideshow, se han agregado dos etiquetas `` que servirán para construir la navegación entre imágenes.

2.5.6. Maquetación del footer y zona inferior

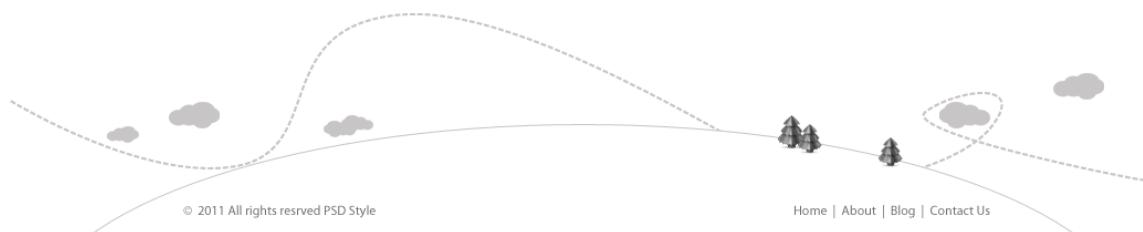


Figura 2.14: Diseño del footer y zona inferior

Finalmente para esta zona también se utilizarán dos columnas, una para el texto legal y otra para el menú inferior:

```
<!-- Comienzo Footer -->
<div id="footer" class="span-24">
    <div id="legal" class="prepend-3 span-9">
        &copy; <?php echo date('Y'); ?> <?php echo $app->getCfg('sitename'); ?>
```

```

</div>
<div id="menu_inferior" class="prepend-3 span-9 last">
    <jdoc:include type="modules" name="interpoint-menu_inferior" style="none" />
</div>
</div>
<!-- Fin Footer -->

```

Nota

Note que además de las clases `span-8` se utilizó `prepend-3` para mover el contenido de las columnas hacia la derecha en cada columna.

Si se juntan todas las partes, el archivo `index.php` quedará conformado de la siguiente forma:

```

<?php
defined('_JEXEC') or die;
JHTML::_('behavior.framework', true);
$app = JFactory::getApplication();
?>
<?php echo '<?' ; ?>xml version="1.0" encoding="<?php echo $this->_charset ?>"?
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/
    xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="<?php echo $this->language;
?>" lang="<?php echo $this->language; ?>" dir="<?php echo $this->direction;
?>" >
<head>
    <jdoc:include type="head" />
    <link rel="stylesheet" href="<?php echo $this->baseurl ?>/templates/<?php
        echo $this->template ?>/css/blueprint/screen.css" type="text/css"
        media="screen, projection" />
    <link rel="stylesheet" href="<?php echo $this->baseurl ?>/templates/<?php
        echo $this->template ?>/css/blueprint/print.css" type="text/css"
        media="print" />
    <!--[if lt IE 8]><link rel="stylesheet" href="blueprint/ie.css" type="text/css" media="screen, projection"><![endif]-->
    <link rel="stylesheet" href="<?php echo $this->baseurl ?>/templates/<?php
        echo $this->template ?>/css/template.css" type="text/css" />
    <script type="text/javascript" src="<?php echo $this->baseurl ?>/
        templates/<?php echo $this->template ?>/js/template.js"></script>
</head>
<body>
    <div class="container">
        <!-- Comienzo header -->
        <div id="header" class="span-24">
            <!-- Comienzo Logo -->
            <div id="logo" class="span-9">
                <a href="<?php echo $this->baseurl ?>" title="<?php echo $app->
                    getCfg('sitename'); ?>">
                    <h1><?php echo $app->getCfg('sitename'); ?></h1>
                </a>
                <h2 id="eslogan">Lorem ipsum dolor sit amet</h2>
            </div>

```

```

<!-- Fin Logo -->

<!-- Comienzo Menús Superiores -->
<div id="menus_superiores" class="span-15 last">

    <?php if($this->countModules('interpoint-menu_ingresar')) : ?>
        <div id="menu_ingresar">
            <jdoc:include type="modules" name="interpoint-
                menu_ingresar" style="none" />
        </div>
    <?php endif; ?>

    <?php if($this->countModules('interpoint-menu_superior')) : ?>
        <div id="menu_superior">
            <jdoc:include type="modules" name="interpoint-
                menu_superior" style="none" />
        </div>
    <?php endif; ?>

    </div>
    <!-- Fin Menús superiores -->

</div>
<!-- Fin Header -->

<!-- Mensajes del CMS -->
<jdoc:include type="message" />

<!-- Comienzo Destacado superior -->
<?php if($this->countModules('interpoint-destacado_superior')) : ?>
    <div id="destacado_superior" class="span-10">
        <jdoc:include type="modules" name="interpoint-destacado_superior"
            style="xhtml" />
    </div>
<?php endif; ?>
<!-- Fin Destacado superior -->

<!-- Comienzo Video -->
<?php if($this->countModules('interpoint-video')) : ?>
    <div id="video" class="span-14 last">
        <div id="reproductor">
            <jdoc:include type="modules" name="interpoint-video" style="
                none" />
        </div>
    </div>
<?php endif; ?>
<!-- Fin Video -->

<!-- Comienzo Caja medio -->
<?php if($this->countModules('interpoint-caja_medio')) : ?>
    <div id="caja_medio" class="span-24">
        <jdoc:include type="modules" name="interpoint-caja_medio" style="
            none" />
    </div>
<?php endif; ?>
<!-- Fin Caja medio -->

```

```

<!-- Comienzo Contenido principal -->
<div id="contenido" class="span-24">
    <jdoc:include type="component" />
</div>
<!-- Fin Contenido principal -->

<!-- Comienzo Destacado inferior -->
<div id="destacado_inferior" class="span-15 append-1">
    <jdoc:include type="modules" name="interpoint-destacado_inferior"
        style="xhtml" />
</div>
<!-- Fin Destacado inferior -->

<!-- Comienzo Slideshow inferior -->
<div id="slideshow_inferior" class="span-8 last">
    <?php if($this->countModules('interpoint-slideshow_inferior')) : ?>
        <div id="slideshow_contenedor">
            <jdoc:include type="modules" name="interpoint-
                slideshow_inferior" style="none" />
        </div>
        <span id="slideshow_der" class="slideshow_nav"></span>
        <span id="slideshow_izq" class="slideshow_nav"></span>
    <?php endif; ?>
</div>
<!-- Fin Slideshow inferior -->

<!-- Comienzo Footer -->
<div id="footer" class="span-24">
    <div id="legal" class="prepend-3 span-9">
        &copy;<?php echo date('Y'); ?> <?php echo $app->getCfg('sitename
        '); ?>
    </div>
    <div id="menu_inferior" class="prepend-3 span-9 last">
        <jdoc:include type="modules" name="interpoint-menu_inferior"
            style="none" />
    </div>
</div>
<!-- Fin Footer -->
</div>
</body>
</html>

```

Al ingresar al sitio se visualizará la estructura de la plantilla, aún sin estilos.

En caso de no tener ningún artículo publicado en la página de inicio, la sección central aparecerá vacía.

2.6. El archivo template.css

Llegó el momento de añadir diseño a la plantilla. Para eso se editará el archivo `template.css` que se encuentra dentro de la carpeta `css` de la plantilla. Note que ya existe contenido dentro del archivo, lo que se hará es borrarlo para luego incorporar los estilos apropiados.

Mi sitio web

Lorem ipsum dolor sit amet

Joomla!

¡Felicitaciones! Ahora ya tiene un sitio Joomla! en funcionamiento.

Joomla! hace que construir su sitio así como lo desea y mantenerlo actualizado sea fácil.

Joomla! es una plataforma flexible y poderosa, tanto si se está construyendo un pequeño sitio, como si se trata de un sitio enorme, con cientos de miles de visitantes. Joomla! está hecho con código abierto, lo cual quiere decir que lo puede modificar para hacerlo funcionar del modo que usted desee.

Principiantes

Si se trata de su primer sitio hecho con Joomla!, está en el sitio correcto. Joomla! le ayudará a levantar su sitio web y a hacerlo funcionar rápida y fácilmente.

Use la cuenta del administrador que creó al instalar Joomla!, para acceder al panel de control de la administración de Joomla!.

[Leer más: Principiantes](#)

Profesionales

Joomla! 1.6 continúa el desarrollo del entorno de trabajo y del CMS de un modo en que la flexibilidad y potencia de su visión de la web, se pueda traer a la realidad. Ahora, con la administración totalmente siguiendo el estilo de MVC, la habilidad para controlar su aspecto y la gestión de extensiones es completa.

[Leer más: Profesionales](#)

©2011 Mi sitio web

Figura 2.15: El sitio con la estructura HTML, aún sin estilos

Nota

Antes de comenzar a escribir los estilos, es muy importante tener preparados los recortes de las diferentes imágenes que componen el diseño. Dichas imágenes deben ir guardadas en la carpeta `images` de la plantilla.

Nota

Si no sabe recortar imágenes con Photoshop, puede consultar el siguiente artículo: http://www.todo-photoshop.com/tutorial-photoshop/basicos/recortando_imagenes_photoshop.html

Nota

A medida que se van aplicando los estilos es conveniente ir rellenando el sitio con contenidos (menús, artículos, etc.) para poder visualizar de mejor manera el trabajo.

Nota

*El nivel de comprensión de los estilos mostrados a continuación dependerán de los conocimientos de CSS que posea. Si desea introducirse más en profundidad con los estilos en cascada, puede consultar la colección referente a este tema en **Libros Web**: <http://www.librosweb.es/>. Aún así, los estilos descritos son de ejemplo, los cuales pueden mejorarse o implementarse de otra forma.*

2.6.1. Estilos generales

Los primeros estilos corresponden para elementos generales del sitio: Tipografías, colores de enlaces, etc:

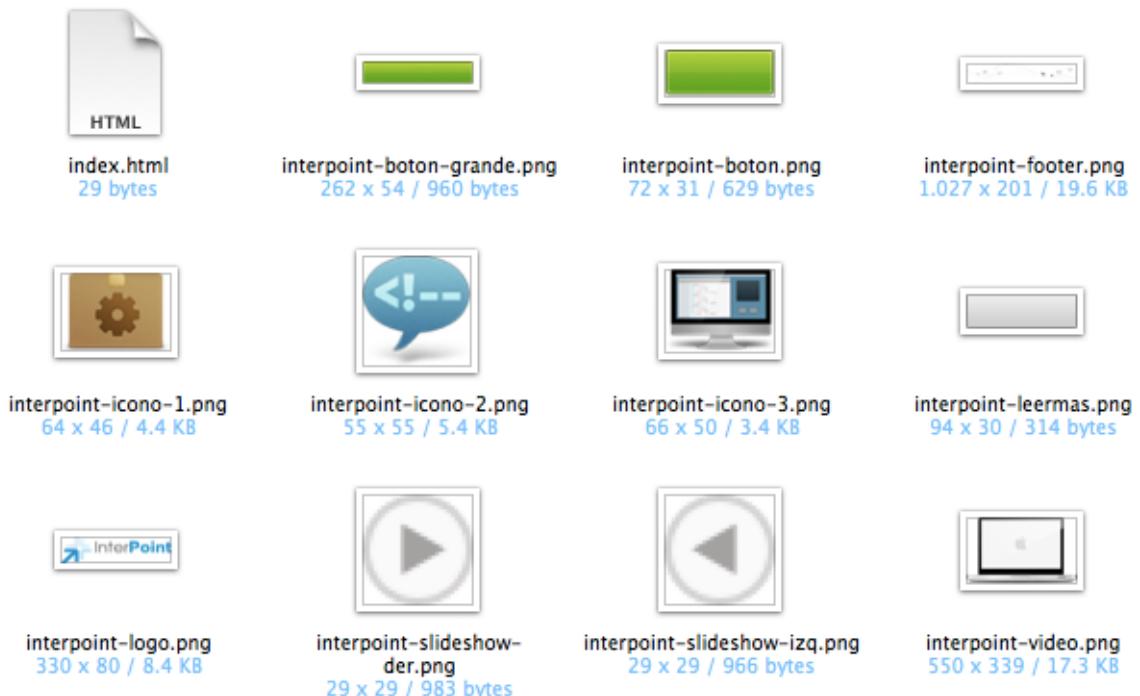


Figura 2.16: Imágenes recortadas que conformarán el diseño

```
/* Estilos generales
-----
body{
    background: #FFF;
    color: #5f5d5d;
    font-family: Arial, Helvetica, sans-serif;
    font-size: 14px;
}
```

2.6.2. Estilos para la sección superior

Se continua con los estilos para la sección superior:

```
/* Parte superior
-----
#header{
    padding: 15px 0 20px 0;
    border-bottom: 1px solid #bcb9b9;
}

/* Logo y eslogan
-----
#logo{
    position: relative;
}
```

```

#logo h1{
    width: 330px;
    height: 80px;
    background: transparent url(..../images/interpoint-logo.png) no-repeat left top;
    margin: 0;
    text-indent: -3000px;
}

#logo a{
    display: block;
}

#eslogan{
    color: #696767;
    font-size: 14px;
    font-weight: normal;
    margin: 0;
    position: absolute;
    right: 20px;
    bottom: 0px;
}

/* Menú ingresar
-----*/
#menu_ingresar ul{
    margin: 0;
    overflow: hidden;
}

#menu_ingresar li{
    width: 72px;
    float: right;
    list-style: none;
    margin: 0 0 0 20px;
}

#menu_ingresar li a{
    display: block;
    width: 72px;
    height: 31px;
    line-height: 30px;
    text-align: center;
    text-decoration: none;
    color: #FFF;
    background: transparent url(..../images/interpoint-botón.png) no-repeat center
        top;
}

#menu_ingresar li a:hover{
    text-decoration: underline;
}

/* Menú superior principal
-----*/

```

```

#menu_superior ul{
    margin: 20px 0 0 30px;
    overflow: hidden;
    padding: 0;
}

#menu_superior li{
    width: auto;
    float: left;
    list-style: none;
    margin: 0 0 0 20px;
}

#menu_superior li a{
    display: block;
    text-align: center;
    text-decoration: none;
    color: #959393;
}

#menu_superior li a:hover{
    text-decoration: underline;
}

#menu_superior .selected a{
    color: #036e9e;
}

```

Para completar los espacios superiores, se utilizaron dos módulos del tipo **Menú**.

2.6.3. Estilos para el destacado superior, video y caja media

Luego se continúa con los estilos para el destacado superior, video y caja del medio:

```

/* Destacado superior
-----*/
#destacado_superior{
    margin: 40px 0 0 0;
}

#destacado_superior h3{
    color: #035f88;
    font-size: 28px;
    font-family: "Myriad Pro", Arial, Helvetica, sans-serif;
    font-weight: 600;
}

#destacado_superior p{
    font-size: 18px;
    color: #5f5d5d;
}

#destacado_superior a{
    display: block;
    width: 262px;
    height: 64px;
}

```



Joomla!

¡Felicitaciones! Ahora ya tiene un sitio Joomla! en funcionamiento.

Joomla! hace que construir su sitio así como lo desea y mantenerlo actualizado sea fácil.

Joomla! es una plataforma flexible y poderosa, tanto si se está construyendo un pequeño sitio, como si se trata de un sitio enorme, con cientos de miles de visitantes. Joomla! está hecho con código abierto, lo cual quiere decir que lo puede modificar para hacerlo funcionar del modo que usted desee.

Principiantes

Si se trata de su primer sitio hecho con Joomla!, está en el sitio correcto. Joomla! le ayudará a levantar su sitio web y a hacerlo funcionar rápida y fácilmente.

Use la cuenta del administrador que creó al instalar Joomla!, para acceder al panel de control de la administración de Joomla!.

[Leer más: Principiantes](#)

Profesionales

Joomla! 1.6 continúa el desarrollo del entorno de trabajo y del CMS de un modo en que la flexibilidad y potencia de su visión de la web, se pueda traer a la realidad. Ahora, con la administración totalmente siguiendo el estilo de MVC, la habilidad para controlar su aspecto y la gestión de extensiones es completa.

[Leer más: Profesionales](#)

©2011 Mi sitio web

Figura 2.17: Sección superior con los estilos aplicados

```

color: #FFF;
font-size: 28px;
font-family: "Myriad Pro", Arial, Helvetica, sans-serif;
font-weight: 600;
line-height: 55px;
text-decoration: none;
text-align: center;
text-shadow: 1px 1px 3px #000;
margin: 0 auto;
background: transparent url(..../images/interpoint-botón-grande.png) no-repeat
            top center;
}

#destacado_superior a:hover{
    text-decoration: underline;
}

/* Video
-----*/
#video{
    height: 355px;
    margin: 40px 0 0 0;
    background: transparent url(..../images/interpoint-video.png) no-repeat top
                center;
    position: relative;
}

#reproductor{
    position: absolute;
    left: 72px;
    top: 17px;
}

/* Caja medio
-----*/
#caja_medio{
    background: #d9eef7;
    margin: 50px 0 0 0;
}

#caja_medio p{
    padding: 10px;
    color: #181818;
    font-size: 18px;
    font-family: "Myriad Pro", Arial, Helvetica, sans-serif;
    margin: 0;
}

```

Para completar los espacios se utilizaron tres módulos del tipo **HTML personalizado** (uno con el texto destacado/botón, otro con el video y otro con un pequeño texto para la caja media).



Aliquam tincidunt mauris

Sed egestas, ante et vulputate volutpat, eros pede semper est, vitae luctus metus libero eu augue. Morbi purus libero, faucibus adipiscing, commodo quis, gravida id, est. Sed lectus. Praesent elementum hendrerit tortor. Sed semper lorem at felis. Vestibulum volutpat.

Lore ipsum ad



Morbi in sem quis dui placerat ornare. Pellentesque pharetra a, ultricies in, diam. Sed arcu. Cras consequat.

Joomla!

¡Felicitaciones! Ahora ya tiene un sitio Joomla! en funcionamiento.

Joomla! hace que construir su sitio así como lo desea y mantenerlo actualizado sea fácil.

Joomla! es una plataforma flexible y poderosa, tanto si se está construyendo un pequeño sitio, como si se trata de un sitio enorme, con cientos de miles de visitantes. Joomla! está hecho con código abierto, lo cual quiere decir que lo puede modificar para hacerlo funcionar del modo que usted desee.

Principiantes

Si se trata de su primer sitio hecho con Joomla!, está en el sitio correcto. Joomla! le ayudará a levantar su sitio web y a hacerlo funcionar rápida y fácilmente.

Use la cuenta del administrador que creó al instalar Joomla!, para acceder al panel de control de la administración de Joomla!.

[Leer más: Principiantes](#)

Profesionales

Joomla! 1.6 continúa el desarrollo del entorno de trabajo y del CMS de un modo en que la flexibilidad y potencia de su visión de la web, se pueda traer a la realidad. Ahora, con la administración totalmente siguiendo el estilo de MVC, la habilidad para controlar su aspecto y la gestión de extensiones es completa.

[Leer más: Profesionales](#)

©2011 Mi sitio web

Figura 2.18: Destacado superior, video y caja media con los estilos aplicados

2.6.4. Estilos para las columnas centrales

Luego se incorporan los estilos para las columnas medias:

```
/* Contenido / Columnas centrales
-----
#contenido{
    margin: 50px 0 0 0;
}

#contenido .leading-0, #contenido .leading-1, #contenido .leading-2{
    width: 280px;
    float: left;
}

#contenido h2{
    line-height: 55px;
    margin: 0 0 10px 0;
}

#contenido h2 a{
    color: #035f88;
    font-size: 20px;
    font-weight: 600;
    font-family: "Myriad Pro", Arial, Helvetica, sans-serif;
    text-decoration: none;
}

#contenido h2 a:hover{
    text-decoration: underline;
}

#contenido .leading-0{
    margin: 0 50px 0 0;
}

#contenido .leading-1{
    margin: 0 50px 0 0;
}

#contenido .leading-2{
    margin: 0;
}

#contenido .leading-0 h2{
    padding: 0 0 0 69px;
    background: transparent url(../images/interpoint-icono-1.png) no-repeat left
                center;
}

#contenido .leading-1 h2{
    padding: 0 0 0 60px;
    background: transparent url(../images/interpoint-icono-2.png) no-repeat left
                center;
}

#contenido .leading-2 h2{
```

```

padding: 0 0 0 71px;
background: transparent url(..../images/interpoint-icono-3.png) no-repeat left
            center;
}

#contenido .readmore a{
    color: #3c8203;
    text-decoration: none;
}

#contenido .readmore a:hover{
    text-decoration: underline;
}

#contenido .readmore a:before {
    content: ">> ";
}

```

The screenshot shows a website layout with a header containing the logo 'InterPoint' and navigation links for Inicio, Sobre nosotros, Clientes, Servicios, Blog, Características, and Contacto. Below the header, there is a section titled 'Aliquam tincidunt mauris' with some placeholder text ('Lorem ipsum dolor sit amet'). To the right of this text is a video player displaying a video titled 'Dogboarding' from 'DANIELS'. Below the video player is a green button labeled 'Lorem ipsum ad'. At the bottom of the page, there is a light blue footer bar with the text 'Morbi in sem quis dui placerat ornare. Pellentesque pharetra a, ultricies in, diam. Sed arcu. Cras consequat.'



Joomla! es una plataforma flexible y poderosa, tanto si se está construyendo un pequeño sitio, como si se trata de un sitio enorme, con cientos de miles de visitantes. Joomla! está hecho con código abierto, lo cual quiere decir que lo puede modificar para hacerlo funcionar del modo que usted desee.

[>> Leer más: Joomla!](#)



Principiantes

Si se trata de su primer sitio hecho con Joomla!, está en el sitio correcto. Joomla! le ayudará a levantar su sitio web y a hacerlo funcionar rápida y fácilmente.

Use la cuenta del administrador que creó al instalar Joomla!, para acceder al panel de control de la administración de Joomla!.

[>> Leer más: Principiantes](#)



Profesionales

Joomla! 1.6 continúa el desarrollo del entorno de trabajo y del CMS de un modo en que la flexibilidad y potencia de su visión de la web, se pueda traer a la realidad. Ahora, con la administración totalmente siguiendo el estilo de MVC, la habilidad para controlar su aspecto y la gestión de extensiones es completa.

[>> Leer más: Profesionales](#)

©2011 Mi sitio web

Figura 2.19: Columnas centrales con los estilos aplicados

Como se explicó anteriormente, los contenidos de las columnas están conformados por tres artículos seleccionados para ser publicados en la página de inicio.

2.6.5. Estilos para el texto destacado inferior y slideshow

```
/* Destacado inferior
-----
#destacado_inferior{
    margin: 50px 0 0 0;
}

#destacado_inferior .newsflash-title{
    color: #035f88;
    font-size: 20px;
    font-weight: 600;
    font-family: "Myriad Pro", Arial, Helvetica, sans-serif;
}

#destacado_inferior .readmore{
    display: block;
    width: 94px;
    height: 30px;
    line-height: 30px;
    background: transparent url(..../images/interpoint-leermas.png) no-repeat center
                center;
    color: #5f5d5d;
    text-decoration: none;
    text-align: center;
}

#destacado_inferior .readmore:hover{
    text-decoration: underline;
}

/* Slideshow inferior
-----
#slideshow_inferior{
    margin: 50px 0 0;
    position: relative;
}

#slideshow_contenedor{
    border: 10px solid #e0e0e0;
    height: 195px;
    width: 290px;
    float: right;
}

#slideshow_inferior .slideshow_nav{
    width: 29px;
    height: 29px;
    position: absolute;
    top: 90px;
    cursor: pointer;
```

```

    z-index: 100;
}

#slideshow_izq{
    background: transparent url(../images/interpoint-slideshow-izq.png) no-repeat
                center center;
    left: -10px;
}

#slideshow_der{
    background: transparent url(../images/interpoint-slideshow-der.png) no-repeat
                center center;
    right: -10px;
}

```

Para los mostrar el artículo se utilizó un módulo del tipo **Artículos - Noticias de actualidad**. Se lo configuró para mostrar un solo artículo, de una categoría definida, ordenado por fecha. De esta forma, en esta sección pueden ir, por ejemplo, las últimas noticias del sitio y cada vez que se carga una noticia nueva, esta aparece en dicho lugar.

Para la sección de slideshow, se utilizó un módulo de **HTML personalizado**. Se incorporó una sola imagen en modo de demostración.

Nota

El slideshow aún no es funcional. Más adelante se explicará como agregar la funcionalidad a la plantilla utilizando JavaScript.

2.6.6. Estilos para el footer y zona inferior

```

/* Footer y menú inferior
-----*/
#footer{
    height: 200px;
    background: transparent url(../images/interpoint-footer.png) no-repeat top
                center;
}

#legal{
    color: #6e6d6d;
    font-size: 12px;
    font-family: "Myriad Pro", Arial, Helvetica, sans-serif;
    margin: 175px 0 0 0;
    line-height: 12px;
}

#menu_inferior{
    margin: 175px 0 0 0;
}

#menu_inferior ul{
    overflow: hidden;
    padding: 0;
    margin: 0;
}

```



Aliquam tincidunt mauris

Sed egestas, ante et vulputate volutpat, eros pede semper est, vitae luctus metus libero eu augue. Morbi purus libero, faucibus adipiscing, commodo quis, gravida id, est. Sed lectus. Praesent elementum hendrerit tortor. Sed semper lorem at felis. Vestibulum volutpat.

Lorem ipsum ad



Morbi in sem quis dui placerat ornare. Pellentesque pharetra a, ultricies in, diam. Sed arcu. Cras consequat.



Joomla! es una plataforma flexible y poderosa, tanto si se está construyendo un pequeño sitio, como si se trata de un sitio enorme, con cientos de miles de visitantes. Joomla! está hecho con código abierto, lo cual quiere decir que lo puede modificar para hacerlo funcionar del modo que usted desee.

[>> Leer más: Joomla!](#)



Principiantes

Si se trata de su primer sitio hecho con Joomla!, está en el sitio correcto. Joomla! le ayudará a levantar su sitio web y a hacerlo funcionar rápida y fácilmente.

Use la cuenta del administrador que creó al instalar Joomla!, para acceder al panel de control de la administración de Joomla!.

[>> Leer más: Principiantes](#)



Profesionales

Joomla! 1.6 continúa el desarrollo del entorno de trabajo y del CMS de un modo en que la flexibilidad y potencia de su visión de la web, se pueda traer a la realidad. Ahora, con la administración totalmente siguiendo el estilo de MVC, la habilidad para controlar su aspecto y la gestión de extensiones es completa.

[>> Leer más: Profesionales](#)

Parques australianos

Este es un sitio básico sobre los bellos y fascinantes parques de Australia.

En este sitio puede leer todo sobre mis viajes a los distintos parques. Mire las fotos, y encuentre enlaces a los sitios de parques.

Este sitio es un ejemplo del uso del núcleo de Joomla! para crear sitios básicos o "borradores de sitio," un blog personal, o una manera de presentar información sobre algún tema que le interese.

[Leer más...](#)

©2011 Mi sitio web



Figura 2.20: Texto destacado inferior y slideshow con los estilos aplicados

```

#menu_inferior li{
    width: auto;
    float: left;
    list-style: none;
    margin: 0 0 0 10px;
}

#menu_inferior li a{
    display: block;
    text-align: center;
    text-decoration: none;
    color: #6e6d6d;
    font-size: 12px;
    font-family: "Myriad Pro", Arial, Helvetica, sans-serif;
    border-left: 1px solid #6e6d6d;
    padding: 0 0 0 10px;
    line-height: 12px;
}

#menu_inferior li:first-child a{
    border: 0px;
}

#menu_inferior li a:hover{
    text-decoration: underline;
}

#menu_inferior .selected a{
    color: #036e9e;
}

```

Para completar las zonas inferiores se usó un solo módulo **Menú** con distintos enlaces hacia secciones del sitio.

2.6.7. Estilos para elementos y mensajes del sistema

Por último se agregan estilos pertenecientes a elementos comunes en el CMS: Mensajes del sistema, iconos de acciones y tooltips.

```

/* Estilos pertenecientes al CMS
-----*/
/* Mensajes del sistema
-----*/
#system-message{
    display: block;
    clear: both;
    padding: 15px 0 0 0;
}

#system-message dt, #system-message dd{
    margin: 5px 0;
}

#system-message ul{
    margin: 0;
}

```



Aliquam tincidunt mauris

Sed egestas, ante et vulputate volutpat, eros pede semper est, vitae luctus metus libero eu augue. Morbi purus libero, faucibus adipiscing, commodo quis, gravida id, est. Sed lectus. Praesent elementum hendrerit tortor. Sed semper lorem at felis. Vestibulum volutpat.

Lore ipsum ad



Morbi in sem quis dui placerat ornare. Pellentesque pharetra a, ultricies in, diam. Sed arcu. Cras consequat.



Joomla!

Joomla! es una plataforma flexible y poderosa, tanto si se está construyendo un pequeño sitio, como si se trata de un sitio enorme, con cientos de miles de visitantes. Joomla! está hecho con código abierto, lo cual quiere decir que lo puede modificar para hacerlo funcionar del modo que usted desee.

[>> Leer más: Joomla!](#)



Principiantes

Si se trata de su primer sitio hecho con Joomla!, está en el sitio correcto. Joomla! le ayudará a levantar su sitio web y a hacerlo funcionar rápida y fácilmente.

Use la cuenta del administrador que creó al instalar Joomla!, para acceder al panel de control de la administración de Joomla!.

[>> Leer más: Principiantes](#)



Profesionales

Joomla! 1.6 continúa el desarrollo del entorno de trabajo y del CMS de un modo en que la flexibilidad y potencia de su visión de la web, se pueda traer a la realidad. Ahora, con la administración totalmente siguiendo el estilo de MVC, la habilidad para controlar su aspecto y la gestión de extensiones es completa.

[>> Leer más: Profesionales](#)

Parques australianos

Este es un sitio básico sobre los bellos y fascinantes parques de Australia.

En este sitio puede leer todo sobre mis viajes a los distintos parques. Mire las fotos, y encuentre enlaces a los sitios de parques.

Este sitio es un ejemplo del uso del núcleo de Joomla! para crear sitios básicos o "borradores de sitio," un blog personal, o una manera de presentar información sobre algún tema que le interese.

[Leer más...](#)



Figura 2.21: Footer y zona inferior con los estilos aplicados

```

/* Acciones
-----
ul.actions {
    clear:both;
    margin-top: -50px;
    float:right;
}

ul.actions li {
    list-style-type: none;
    float:right;
    margin-left: 10px;
}

/* Tooltips
-----
.tip-wrap{
    background: #D9EEF7;
    padding: 5px;
    font-size: 12px;
}

```

Estos elementos aparecen en ciertas situaciones, por ejemplo: cuando un usuario quiere ingresar al sitio y la acción es rechazada, el CMS muestra un mensaje. Los iconos de acciones y tooltips suelen aparecer cuando se ingresa al sitio con permisos de edición de artículos.

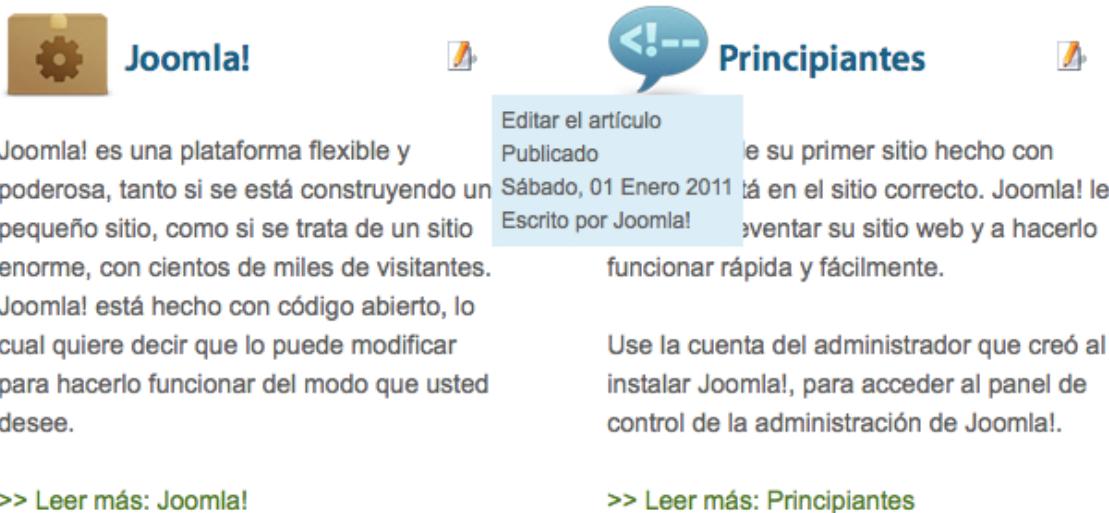


Figura 2.22: Acciones y tooltips con los estilos aplicados

2.6.8. Estilos completos

Al juntar todos los estilos, el archivo `template.css` conformado de la siguiente forma:

```

/* Estilos generales
-----*/
body{
    background: #FFF;
    color: #5f5d5d;
    font-family: Arial, Helvetica, sans-serif;
    font-size: 14px;
}

/* Parte superior
-----*/
#header{
    padding: 15px 0 20px 0;
    border-bottom: 1px solid #bcb9b9;
}

/* Logo y eslogan
-----*/
#logo{
    position: relative;
}

#logo h1{
    width: 330px;
    height: 80px;
    background: transparent url(../images/interpoint-logo.png) no-repeat left top;
    margin: 0;
    text-indent: -3000px;
}

#logo a{
    display: block;
}

#eslogan{
    color: #696767;
    font-size: 14px;
    font-weight: normal;
    margin: 0;
    position: absolute;
    right: 20px;
    bottom: 0px;
}

/* Menú ingresar
-----*/
#menu_ingresar ul{
    margin: 0;
    overflow: hidden;
}

#menu_ingresar li{
    width: 72px;
}

```

```

        float: right;
        list-style: none;
        margin: 0 0 0 20px;
    }

    #menu_ingresar li a{
        display: block;
        width: 72px;
        height: 31px;
        line-height: 30px;
        text-align: center;
        text-decoration: none;
        color: #FFF;
        background: transparent url(../images/interpoint-botón.png) no-repeat center
                    top;
    }

    #menu_ingresar li a:hover{
        text-decoration: underline;
    }

/* Menú superior principal
-----*/
#menu_superior ul{
    margin: 20px 0 0 30px;
    overflow: hidden;
    padding: 0;
}

#menu_superior li{
    width: auto;
    float: left;
    list-style: none;
    margin: 0 0 0 20px;
}

#menu_superior li a{
    display: block;
    text-align: center;
    text-decoration: none;
    color: #959393;
}

#menu_superior li a:hover{
    text-decoration: underline;
}

#menu_superior .selected a{
    color: #036e9e;
}

/* Destacado superior
-----*/
#destacado_superior{

```

```

        margin: 40px 0 0 0;
    }

#destacado_superior h3{
    color: #035f88;
    font-size: 28px;
    font-family: "Myriad Pro", Arial, Helvetica, sans-serif;
    font-weight: 600;
}

#destacado_superior p{
    font-size: 18px;
    color: #5f5d5d;
}

#destacado_superior a{
    display: block;
    width: 262px;
    height: 64px;
    color: #FFF;
    font-size: 28px;
    font-family: "Myriad Pro", Arial, Helvetica, sans-serif;
    font-weight: 600;
    line-height: 55px;
    text-decoration: none;
    text-align: center;
    text-shadow: 1px 1px 3px #000;
    margin: 0 auto;
    background: transparent url(..../images/interpoint-botón-grande.png) no-repeat
                top center;
}

#destacado_superior a:hover{
    text-decoration: underline;
}

/* Video
-----*/
#video{
    height: 355px;
    margin: 40px 0 0 0;
    background: transparent url(..../images/interpoint-video.png) no-repeat top
                center;
    position: relative;
}

#reproductor{
    position: absolute;
    left: 72px;
    top: 17px;
}

/* Caja medio
-----*/

```

```

#caja_medio{
    background: #d9eef7;
    margin: 50px 0 0 0;
}

#caja_medio p{
    padding: 10px;
    color: #181818;
    font-size: 18px;
    font-family: "Myriad Pro", Arial, Helvetica, sans-serif;
    margin: 0;
}

/* Contenido / Columnas centrales
-----*/
#contenido{
    margin: 50px 0 0 0;
}

#contenido .leading-0, #contenido .leading-1, #contenido .leading-2{
    width: 280px;
    float: left;
}

#contenido h2{
    line-height: 55px;
    margin: 0 0 10px 0;
}

#contenido h2 a{
    color: #035f88;
    font-size: 20px;
    font-weight: 600;
    font-family: "Myriad Pro", Arial, Helvetica, sans-serif;
    text-decoration: none;
}

#contenido h2 a:hover{
    text-decoration: underline;
}

#contenido .leading-0{
    margin: 0 50px 0 0;
}

#contenido .leading-1{
    margin: 0 50px 0 0;
}

#contenido .leading-2{
    margin: 0;
}

#contenido .leading-0 h2{

```

```

padding: 0 0 0 69px;
background: transparent url(..../images/interpoint-icono-1.png) no-repeat left
            center;
}

#contenido .leading-1 h2{
    padding: 0 0 0 60px;
    background: transparent url(..../images/interpoint-icono-2.png) no-repeat left
            center;
}

#contenido .leading-2 h2{
    padding: 0 0 0 71px;
    background: transparent url(..../images/interpoint-icono-3.png) no-repeat left
            center;
}

#contenido .readmore a{
    color: #3c8203;
    text-decoration: none;
}

#contenido .readmore a:hover{
    text-decoration: underline;
}

#contenido .readmore a:before {
    content: ">> ";
}

/* Destacado inferior
-----*/
#destacado_inferior{
    margin: 50px 0 0 0;
}

#destacado_inferior .newsflash-title{
    color: #035f88;
    font-size: 20px;
    font-weight: 600;
    font-family: "Myriad Pro", Arial, Helvetica, sans-serif;
}

#destacado_inferior .readmore{
    display: block;
    width: 94px;
    height: 30px;
    line-height: 30px;
    background: transparent url(..../images/interpoint-leermas.png) no-repeat center
            center;
    color: #5f5d5d;
    text-decoration: none;
    text-align: center;
}

```

```

#destacado_inferior .readmore:hover{
    text-decoration: underline;
}

/* Slideshow inferior
-----*/
#slideshow_inferior{
    margin: 50px 0 0;
    position: relative;
}

#slideshow_contenedor{
    border: 10px solid #e6e6e6;
    height: 195px;
    width: 290px;
    float: right;
}

#slideshow_inferior .slideshow_nav{
    width: 29px;
    height: 29px;
    position: absolute;
    top: 90px;
    cursor: pointer;
    z-index: 100;
}

#slideshow_izq{
    background: transparent url(../images/interpoint-slideshow-izq.png) no-repeat
                center center;
    left: -10px;
}

#slideshow_der{
    background: transparent url(../images/interpoint-slideshow-der.png) no-repeat
                center center;
    right: -10px;
}

/* Footer y menú inferior
-----*/
#footer{
    height: 200px;
    background: transparent url(../images/interpoint-footer.png) no-repeat top
                center;
}

#legal{
    color: #6e6d6d;
    font-size: 12px;
    font-family: "Myriad Pro", Arial, Helvetica, sans-serif;
    margin: 175px 0 0 0;
    line-height: 12px;
}

```

```

#menu_inferior{
    margin: 175px 0 0 0;
}

#menu_inferior ul{
    overflow: hidden;
    padding: 0;
    margin: 0;
}

#menu_inferior li{
    width: auto;
    float: left;
    list-style: none;
    margin: 0 0 0 10px;
}

#menu_inferior li a{
    display: block;
    text-align: center;
    text-decoration: none;
    color: #6e6d6d;
    font-size: 12px;
    font-family: "Myriad Pro", Arial, Helvetica, sans-serif;
    border-left: 1px solid #6e6d6d;
    padding: 0 0 0 10px;
    line-height: 12px;
}

#menu_inferior li:first-child a{
    border: 0px;
}

#menu_inferior li a:hover{
    text-decoration: underline;
}

#menu_inferior .selected a{
    color: #036e9e;
}

/* Estilos pertenecientes al CMS
-----*/
/* Mensajes del sistema
-----*/
#system-message{
    display: block;
    clear: both;
    padding: 15px 0 0 0;
}

#system-message dt, #system-message dd{
    margin: 5px 0;
}

```

```

#system-message ul{
    margin: 0;
}

/* Acciones
-----
ul.actions {
    clear:both;
    margin-top: -50px;
    float:right;
}

ul.actions li {
    list-style-type: none;
    float:right;
    margin-left: 10px;
}

/* Tooltips
-----
.tip-wrap{
    background: #D9EEF7;
    padding: 5px;
    font-size: 12px;
}

```

2.7. Finalizando la plantilla

El trabajo ya está casi listo, las últimas tareas a realizar son:

- Verificar que el diseño se visualiza correctamente en todos los navegadores soportados. En caso que exista algún problema de diseño con *Internet Explorer*, recuerde que existe el archivo `template_ie.css` en donde es posible añadir estilos que únicamente serán cargados por dicho navegador.
- Experimentar con diferentes tipos de contenidos para poder comprobar lo flexible que es la plantilla y en qué situaciones es posible encontrar un error.
- Crear dos capturas de pantallas: `template_preview.png` que será una imagen a tamaño real del `template` y `template_thumbnail.png` que será una imagen miniatura del diseño. Dichas imágenes van dentro de la carpeta raíz de la plantilla.
- Finalmente crear un instalador de la plantilla: es tan sencillo como seleccionar todos los archivos que componen la plantilla y comprimirlos en un archivo `.zip`. Ese archivo será el instalador para utilizar en múltiples sitios.

InterPoint

Aliquam tincidunt mauris.

Sed aptent, ante vel malesuada volutpat, enim pede semper est, ultricies felis libero augue. Morbi purus diam, feugiat adipiscing, commodo quis, gravida et, aenean sed lectus. Proin elementum hendrerit tortor. Sed semper lacinia at felis. Vestibulum eu ligula.

Lorem ipsum ad.

Watch our video about our services. Pellentesque pharetra a, ultricies in, aliquam sed, erat. Donec congue.

Business

InterPoint es una plataforma flexible y potente, destinada a las empresas que buscan una solución completa para la gestión de sus servicios, con control de todos los sistemas, conexión web directa con Google Adwords, etc.

Private

Si se trata de su propia vida familiar o profesional, InterPoint es la mejor opción para organizar y administrar su vida social y laboral. Funciones sociales y financieras.

Professional

InterPoint 1.0 combina el manejo de las finanzas personales con las necesidades del trabajo. Ofrece acceso directo a la redacción, correo, así como administración remota.

Detalles de la plantilla InterPoint

Sin previsualización disponible

Figura 2.23: Detalles de la plantilla

Capítulo 3

Temas avanzados

3.1. Estilos de plantillas, parámetros de configuración y maquetaciones diferentes por secciones

Como se comentó la principio del manual, a partir de Joomla! 1.6, el CMS incorpora una nueva funcionalidad llamada estilos de plantillas. Dichos estilos permiten, a través de parámetros de configuración, variar el diseño de la plantilla en varios aspectos: colores, imágenes, maquetación, tamaños, entre otras características. A su vez cada estilo puede ser asignado a una o varias secciones específicas.

▼ Opciones avanzadas

Altura de plegado (%)	<input type="text" value="53"/>
Altura de expansión (%)	<input type="text" value="72"/>
Logotipo	<input type="text" value="images/joomla_black.gif"/> <input type="button" value="Seleccionar"/> <input type="button" value="Limpiar"/>
Título del sitio	<input type="text" value="Joomla!"/>
Descripción del sitio	<input type="text" value="Open Source Content Management System"/>
Posición de navegación	<input type="text" value="antes del contenido"/> <input type="button" value=""/>
Color de la plantilla	<input type="text" value="Personal"/> <input type="button" value=""/>

Figura 3.1: Parámetros de configuración de la plantilla

Nota

Recuerde que el listado de estilos de plantillas se puede encontrar yendo, en la administración, a Extensiones > Gestor de plantillas. Al ingresar a alguno de los ítems enlistados se pueden visualizar los parámetros disponibles para configurar y a que secciones aplicarlos.

En el caso de la plantilla creada de ejemplo, se le añadirán varios parámetros de configuración para poder:

- Cambiar la imagen del logotipo;
- Cambiar el texto del eslogan;
- Aplicar una maquetación diferente.

3.1.1. Cambiar la imagen del logo

Modificación de templateDetails.xml

Los parámetros de configuración de la plantilla se agregan en el archivo `templateDetails.xml` añadiendo las etiquetas que se muestran a continuación:

```
<config>
  <fields name="params">
    <fieldset name="advanced">
      <field name="logo" type="media"
            label="TPL_INTERPOINT_LOGO_LABEL" description="
            TPL_INTERPOINT_LOGO_DESC" />
    </fieldset>
  </fields>
</config>
```

Como se muestra, la etiqueta más importante es `<field />`. Dicha etiqueta posee los siguientes atributos:

- **name**: Debe ser un nombre único, a través del cual se hará referencia al parámetro para obtener su valor;
- **type**: El tipo de campo, el cual puede ser de varios tipos: `text` (campo de texto), `list` (lista de opciones), `media` (campo con selección de archivo);
- **label**: Será el nombre resumido que acompaña al campo cuando sea mostrado desde la administración;
- **description**: La descripción será la explicación del campo.

Nota

Note que para los atributos `label` y `description` se utiliza `TPL_INTERPOINT_LOGO_LABEL` y `TPL_INTERPOINT_LOGO_DESC`. Estas dos referencias se deben agregar en el archivo `es-ES.tpl_interpoint.ini` junto a los textos correspondientes.

Por lo tanto, `templateDetails.xml` quedará de la siguiente manera:



Figura 3.2: Parámetros de configuración de la plantilla: Imagen del logo

```

<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE install PUBLIC "-//Joomla! 1.6//DTD template 1.0//EN" "http://www.
 joomla.org/xml/dtd/1.6/template-install.dtd">
<extension
  version="1.6"
  type="template"
  client="site">
<name>InterPoint</name>
<creationDate>2011</creationDate>
<author>Comunidad Joomla</author>
<authorEmail>contacto@comunidadjoomla.org</authorEmail>
<authorUrl>http://www.comunidadjoomla.org/</authorUrl>
<copyright>Comunidad Joomla 2011</copyright>
<license>GNU General Public License version 2</license>
<version>1.0</version>
<description>TPL_INTERPOINT_XML_DESCRIPTION</description>
<files>
    <folder>html</folder>
    <folder>css</folder>
    <folder>images</folder>
    <folder>language</folder>
    <folder>js</folder>
    <filename>index.php</filename>
    <filename>index.html</filename>
    <filename>favicon.ico</filename>
    <filename>templateDetails.xml</filename>
    <filename>template_preview.png</filename>
    <filename>template_thumbnail.png</filename>
    <filename>component.php</filename>
    <filename>error.php</filename>
</files>
<positions>
    <position>interpoint-menu_ingresar</position>
    <position>interpoint-menu_superior</position>
    <position>interpoint-destacado_superior</position>
    <position>interpoint-video</position>
    <position>interpoint-caja_medio</position>
    <position>interpoint-destacado_inferior</position>
    <position>interpoint-slideshow_inferior</position>
    <position>interpoint-menu_inferior</position>
</positions>
<languages folder="language">
    <language tag="es-ES">es-ES/es-ES.tpl_interpoint.ini</language>
    <language tag="es-ES">es-ES/es-ES.tpl_interpoint.sys.ini</language>

```

```

</languages>

<config>
    <fields name="params">
        <fieldset name="advanced">
            <field name="logo" type="media"
                label="TPL_INTERPOINT_LOGO_LABEL" description
                ="TPL_INTERPOINT_LOGO_DESC" />
        </fieldset>
    </fields>
</config>
</extension>

```

Modificación del index.php

En `index.php` lo primero que se hará es declarar una variable para guardar el valor del parámetro que queremos obtener. Esto es posible realizarlo dentro de las etiquetas PHP que se encuentran al principio del documento:

```

<?php

defined('_JEXEC') or die;

JHTML::_('behavior.framework', true);

$app = JFactory::getApplication();

//Parámetros de la plantilla
$imagenLogo = $this->params->get('logo');

?>

```

Como puede ver, los parámetros se obtienen a través de `$this->params->get()` pasando como argumento el nombre del parámetro a obtener (en este caso el parámetro `logo`). Luego, antes del cierre de la etiqueta `<head />` se incorpora:

```

<?php if($imagenLogo) : ?>

<style type="text/css">

    #logo h1{
        background: transparent url(<?php echo $imagenLogo; ?>) no-
                    repeat left top;
    }

</style>

<?php endif; ?>

```

De modo que, si se configuró el parámetro, se añade un estilo que sobrescriba el establecido en `template.css` y agregue la nueva imagen del logo en el diseño.

La cabecera del archivo `index.php` queda de la siguiente forma:

```
<?php
```



Figura 3.3: Inclusión de la nueva imagen para el logo

```

defined('_JEXEC') or die;

JHTML::_('behavior.framework', true);

$app = JFactory::getApplication();

//Parámetros de la plantilla
$imagenLogo = $this->params->get('logo');

?>

<?php echo '<?' ; ?>xml version="1.0" encoding="<?php echo $this->_charset ?>"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/
  xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="<?php echo $this->language;
?>" lang="<?php echo $this->language; ?>" dir="<?php echo $this->direction;
?>" >

<head>

<jdoc:include type="head" />

<link rel="stylesheet" href="<?php echo $this->baseurl ?>/templates/<?php
  echo $this->template ?>/css/blueprint/screen.css" type="text/css" media
  ="screen, projection" />
<link rel="stylesheet" href="<?php echo $this->baseurl ?>/templates/<?php
  echo $this->template ?>/css/blueprint/print.css" type="text/css" media="
  print" />
<!--[if lt IE 8]><link rel="stylesheet" href="blueprint/ie.css" type="text/
  css" media="screen, projection"><![endif]-->

<link rel="stylesheet" href="<?php echo $this->baseurl ?>/templates/<?php
  echo $this->template ?>/css/template.css" type="text/css" />

<script type="text/javascript" src="<?php echo $this->baseurl ?>/templates
  /<?php echo $this->template ?>/js/jquery.min.js"></script>
<script type="text/javascript" src="<?php echo $this->baseurl ?>/templates
  /<?php echo $this->template ?>/js/jquery.cycle.all.min.js"></script>
<script type="text/javascript" src="<?php echo $this->baseurl ?>/templates
  /<?php echo $this->template ?>/js/template.js"></script>

<?php if($imagenLogo) : ?>

<style type="text/css">

#logo h1{
  background: transparent url(<?php echo $this->baseurl ."/".
  $imagenLogo; ?>) no-repeat left top;

```

```

        }

    </style>

<?php endif; ?>

</head>

```

3.1.2. Cambiar el texto del eslogan

Siguiendo los mismos pasos anteriores, se modifica `templateDetails.xml` añadiendo un nuevo parámetro:

```

<config>
    <fields name="params">
        <fieldset name="advanced">
            <field name="logo" type="media"
                  label="TPL_INTERPOINT_LOGO_LABEL" description=""
                  TPL_INTERPOINT_LOGO_DESC" />

            <field name="eslogan" type="text" default=""
                  label="TPL_INTERPOINT_ESLOGAN_LABEL"
                  description="TPL_INTERPOINT_ESLOGAN_DESC"
                  filter="string" />
        </fieldset>
    </fields>
</config>

```

Nota

Note que se añadieron dos nuevos atributos: `* default`: Será el valor predeterminado en caso que el campo esté incompleto; `* filter`: Este permite especificar el tipo de información que se espera en el campo (`string`, `word`, `integer`);



Figura 3.4: Parámetros de configuración de la plantilla: Texto de eslogan

Luego, en `index.php`, se agrega una nueva llamada para obtener el parámetro `eslogan`:

```

<?php

defined('_JEXEC') or die;

```

```

JHTML::_('behavior.framework', true);

$app = JFactory::getApplication();

//Parámetros de la plantilla
$imagenLogo = $this->params->get('logo');
$eslogan = $this->params->get('eslogan');

?>

```

Y se modifica la maquetación correspondiente a la zona del logotipo con lo siguiente:

```

<!-- Comienzo Logo -->
<div id="logo" class="span-9">
    <a href="php echo $this-&gt;baseurl ?&gt;" title="<?php echo $app-&gt;getCfg('
        sitename'); ?&gt;"&gt;
        &lt;h1&gt;&lt;?php echo $app-&gt;getCfg('sitename'); ?&gt;&lt;/h1&gt;
    &lt;/a&gt;
    &lt;h2 id="eslogan"&gt;
        &lt;?php if($eslogan) : ?&gt;
            &lt;?php echo $eslogan; ?&gt;
        &lt;?php else : ?&gt;
            Lorem ipsum dolor sit amet
        &lt;?php endif; ?&gt;
    &lt;/h2&gt;
&lt;/div&gt;
&lt;!-- Fin Logo --&gt;
</pre

```

Si se configuró el parámetro `eslogan`, se muestra el texto correspondiente, caso contrario se muestra el predeterminado.



Figura 3.5: Diseño con nuevo eslogan configurable

Aplicar una maquetación diferente

La idea de esta configuración es poder predisponer de dos presentaciones distintas para mostrar en la parte central de la plantilla:

- Una columna, como es actualmente (recordar que en apariencia son tres columnas debido a que se configuró a Joomla para que muestre los artículos de esa forma, pero la maquetación sigue siendo de una sola columna);
- Dos columnas, para poder mostrar los contenidos en un formato del tipo blog: una columna central con los artículos y otra complementaria para mostrar módulos adicionales.

Por lo tanto, al igual que antes, se modifica `templateDetails.xml`:

```

<config>
    <fields name="params">
        <fieldset name="advanced">
            <field name="logo" type="media"
                label="TPL_INTERPOINT_LOGO_LABEL" description=""
                TPL_INTERPOINT_LOGO_DESC" />

            <field name="eslogan" type="text" default=""
                label="TPL_INTERPOINT_ESLOGAN_LABEL"
                description="TPL_INTERPOINT_ESLOGAN_DESC"
                filter="string" />
            <field name="maquetacion" type="list" default="1"
                label="TPL_INTERPOINT_MAQUETACION_LABEL"
                description="TPL_INTERPOINT_MAQUETACION_DESC"
                filter="integer"
            >
                <option value="1">TPL_INTERPOINT_OPCION_1COLUMN</
                    option>
                <option value="2">TPL_INTERPOINT_OPCION_2COLUMNAS</
                    option>
            </field>
        </fieldset>
    </fields>
</config>

```

Nota

Note que el parámetro, al ser del tipo lista, necesita tener definidas las opciones que se podrán seleccionar.

The screenshot shows the 'Opciones avanzadas' (Advanced Options) section of a Joomla template configuration. It includes fields for 'Imagen del logo' (Logo image), 'Texto de eslogan' (Slogan text), and 'Maquetación central' (Central layout). The 'Imagen del logo' field has a file path 'images/joomla_logc' and buttons for 'Seleccionar' (Select) and 'Limpiar' (Clear). The 'Texto de eslogan' field contains the text 'Mi sitio web y Jooml'. The 'Maquetación central' field is set to '1 columna' (1 column) with a dropdown arrow.

Figura 3.6: Parámetros de configuración de la plantilla: Maquetación central

En `index.php` se agrega nuevamente una variable que obtenga el valor del parámetro `maquetacion`:

```

<?php
defined('_JEXEC') or die;

JHTML::_('behavior.framework', true);

```

```

$app = JFactory::getApplication();

//Parámetros de la plantilla
$imagenLogo = $this->params->get('logo');
$eslogan = $this->params->get('eslogan');
$maquetacion = $this->params->get('maquetacion');

?>

Luego, dentro del cuerpo del documento se agrega una nueva maquetación y posición de módulos:

<!-- Maquetación a una columna -->
<?php if($maquetacion == 1) : ?>

<!-- Comienzo Contenido principal -->
<div id="contenido" class="span-24">
    <jdoc:include type="component" />
</div>
<!-- Fin Contenido principal -->

<!-- Maquetación a dos columnas -->
<?php else : ?>

<!-- Comienzo Contenido principal -->
<div id="columna_izquierda" class="span-17 append-1">
    <jdoc:include type="component" />
</div>
<!-- Fin Contenido principal -->

<!-- Comienzo Columna derecha -->
<div id="columna_derecha" class="span-6 last">
    <jdoc:include type="modules" name="interpoint-columna_derecha" style="
        personalizado" headerLevel="3" />
</div>
<!-- Fin Columna derecha -->

<?php endif; ?>

```

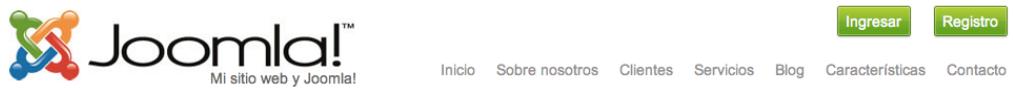
En caso que se haya seleccionado la opción “1 columna”, se muestra el contenido en una maquetación de una sola columna. Caso contrario, se muestra una maquetación de dos columnas.

Nota

Note que se agregó una nueva posición de módulo: interpoint-columna_derecha. Esta deber ser agregada además en templateDetails.xml y la referencia de su descripción en es-ES. tpl_interpoint.sys.ini.

Para poder aplicar la nueva maquetación en el sitio utilizando los estilos de plantillas, se debe:

- En la administración, ir al gestor de estilos de plantillas;
- Duplicar el estilo predeterminado para el sitio;
- Acceder a esta copia, cambiar la opción de maquetación y decidir a que ítems del menú desea ver el nuevo diseño.



Morbi in sem quis dui placerat ornare. Pellentesque pharetra a, ultricies in, diam. Sed arcu. Cras consequat.

Joomla!



Detalles

Categoría: [Joomla!](#)
Publicado el Sábado, 01 Enero 2011 00:00
Escrito por Joomla!
Visto: 20

Joomla! es una plataforma flexible y poderosa, tanto si se está construyendo un pequeño sitio, como si se trata de un sitio enorme, con cientos de miles de visitantes. Joomla! está hecho con código abierto, lo cual quiere decir que lo puede modificar para hacerlo funcionar del modo que usted desee.

Joomla! es una plataforma flexible y poderosa, tanto si se está construyendo un pequeño sitio, como si se trata de un sitio enorme, con cientos de miles de visitantes. Joomla! está hecho con código abierto, lo cual quiere decir que lo puede modificar para hacerlo funcionar del modo que usted desee.

- [< Anterior](#)
- [Siguiente >](#)

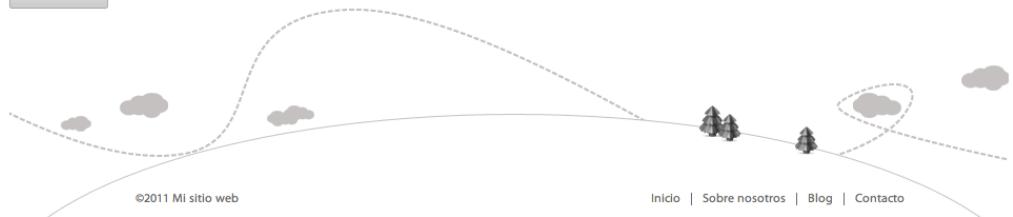
Parques australianos

Este es un sitio básico sobre los bellos y fascinantes parques de Australia.

En este sitio puede leer todo sobre mis viajes a los distintos parques. Mire las fotos, y encuentre enlaces a los sitios de parques.

Este sitio es un ejemplo del uso del núcleo de Joomla! para crear sitios básicos o "borradores de sitio," un blog personal, o una manera de presentar información sobre algún tema que le interese.

[Leer más...](#)



[Inicio](#) | [Sobre nosotros](#) | [Blog](#) | [Contacto](#)

Figura 3.7: Diseño desacomodado al ingresar a un artículo del sitio

Por ejemplo, si se selecciona asignar el nuevo diseño a una categoría dada, al ingresar a un artículo de esa categoría se visualizará la sección central desacomodada.

Una elección pueden ser despublicar los módulos superiores e inferiores y se asignar otros en la columna derecha, mejorando la situación.



Figura 3.8: Diseño sin los módulos superiores al ingresar a un artículo del sitio

Agregando algunos estilos en `template.css` es posible mejorar el diseño:

```
/* Columna izquierda
-----
#columna_izquierda{
    margin: 50px 0 0 0;
}

#columna_izquierda h2 a {
    color: #035F88;
    font-family: "Myriad Pro", Arial, Helvetica, sans-serif;
    font-size: 35px;
    font-weight: 600;
    text-decoration: none;
}
```

```

#columna_izquierda h2 a:hover{
    text-decoration: underline;
}

#columna_izquierda .article-info{
    background: #D9EEF7;
    color: #181818;
    font-size: 13px;
    margin: 0 0 20px 0;
    padding: 10px;
}

/* Columna derecha
-----*/
#columna_derecha{
    margin: 50px 0 0 0;
}

/* Módulos derechos
-----*/
#columna_derecha .moduledtable{
    margin: 0 0 30px 0;
}

#columna_derecha h3{
    color: #035F88;
    font-family: "Myriad Pro", Arial, Helvetica, sans-serif;
    font-size: 20px;
    font-weight: 600;
    margin: 0 0 10px 0;
}

#columna_derecha .moduledtable ul{
    list-style: none;
    padding: 0;
}

#columna_derecha .search .inputbox{
    font-size: 14px;
    padding: 5px;
}

#columna_derecha .mostread a{
    display: block;
    text-decoration: none;
    border-bottom: 1px solid #CCC;
    padding: 0 0 5px 0;
    color: #3C8203;
}

#columna_derecha .mostread a:hover{
    text-decoration: underline;
}

```

```

#columna_derecha #modlgn_username, #columna_derecha #modlgn_passwd{
    width: 100%;
}

/* Página
-----
.pagenav, .pagination{
    list-style: none;
    text-align: center;
    margin: 20px 0;
}

.pagenav li, .pagination li{
    display: inline-block;
    margin: 0 10px;
}

```

Como se puede ver, los estilos en plantillas son una opción interesante que le otorgan flexibilidad a la plantilla, permitiendo cambiar su diseño ante distintas situaciones.

3.2. Personalización de módulos y componentes

3.2.1. Extensiones desarrolladas en MVC

Desde la versión 1.5, Joomla permite la sobreescritura del HTML que poseen módulos y componentes, otorgando mucho más control sobre lo que se desea mostrar en el sitio, sin tener que modificar archivos del núcleo de dichas extensiones o del mismo CMS.

Para poder sobreescibir la estructura HTML que posea una extensión, es requisito que ésta esté creada utilizando el patrón de arquitectura MVC (Modelo-Vista-Controlador).

Desde el punto de vista del usuario común, no existe una forma automática de saber si una extensión utiliza MVC. La única manera de saberlo es yendo a la carpeta de dicha extensión y mirando su estructura de carpetas y archivos.

En donde existe una carpeta `controllers`, otra `models` y otra `views`. En ésta última carpeta se encuentran archivos que se utilizaran como base para personalizar el HTML del componente.

En este caso de los módulos, la carpeta `tmpl` posee archivos que se utilizaran para personalizar a gusto el HTML del mismo.

Nota

En Joomla, los archivos que componen cada componente pueden encontrarse dentro de la carpeta `components`, en el directorio raíz del CMS. Allí cada componente está separado por carpetas diferentes (`com_banners`, `com_contact`, etc). En el caso de los módulos, estos se encuentran en la carpeta `modules`, también en el directorio raíz del CMS. Y al igual que el caso anterior, cada módulo está separado en una carpeta diferente (`mod_articles_archive`, `mod_articles_categories`, etc).

Nota

Si desea conocer un poco más sobre MVC puede consultar:



Figura 3.9: Diseño al ingresar a un artículo del sitio con los estilos aplicados

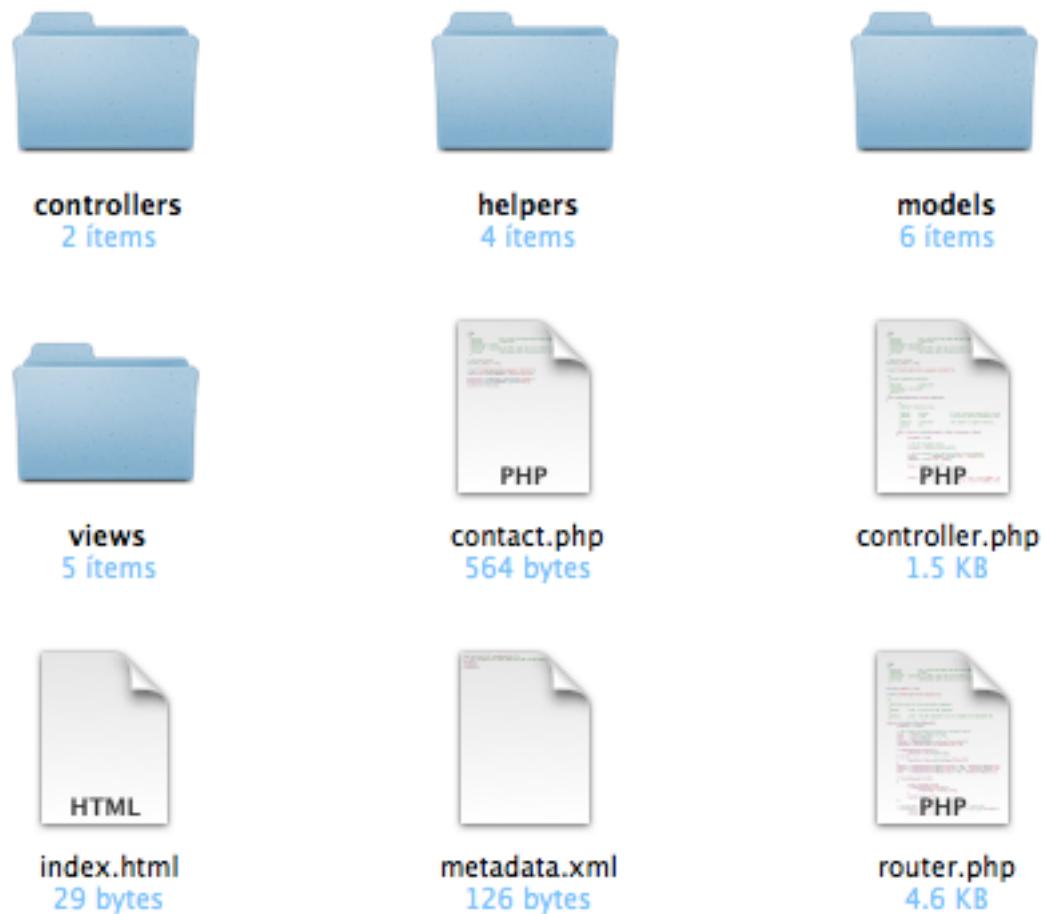


Figura 3.10: Estructura habitual de los componentes creados con el patrón MVC

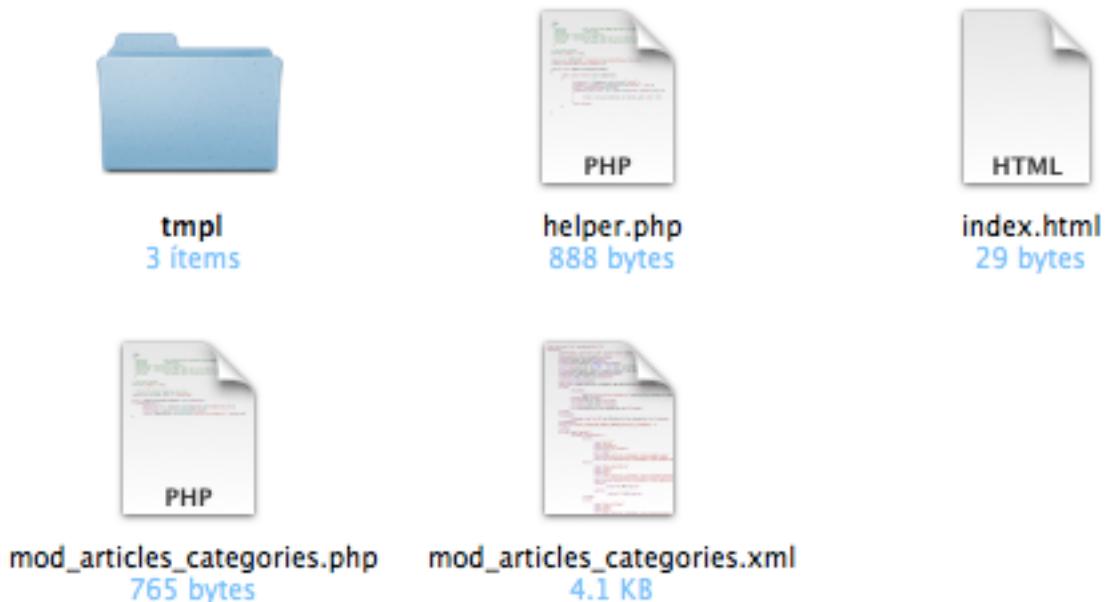


Figura 3.11: Estructura habitual de los módulos creados con el patrón MVC

- http://es.wikipedia.org/wiki/Modelo_Vista_Controlador
- <http://docs.joomla.org/MVC>

Tanto la personalización de módulos como de componentes, se trabajan dentro de la carpeta `html` de la plantilla.

3.2.2. Módulos personalizados

En este aspecto, es posible adaptar a los módulos en varios aspectos:

- La manera con que se muestra el módulo: Anteriormente, en el análisis de las secciones que componen a `index.php`, se explicó que los módulos se pueden cargar de diferentes formas cambiando el valor del atributo `style` en la directiva `<jdoc:include type="modules" name="" style="" />`. Sin embargo, existen situaciones en que se necesita que un módulo cargue dentro de una estructura determinada. Por eso mismo, Joomla permite crear estilos personalizados de visualización de módulos.
- El mismo HTML que compone un módulo: También es posible personalizar las etiquetas HTML que componen a uno o varios módulos en particular, otorgando la flexibilidad para adaptar un diseño al gusto.

Personalizar la forma con que muestran los módulos

Como se comentó anteriormente, el trabajo de personalización de módulos (y también de componentes) ocurre en la carpeta `html` de la plantilla.

Es el archivo `modules.php` quien se encarga de contener los diferentes estilos personalizados de visualización de módulos. Al abrir a dicho archivo, el mismo posee tres funciones definidas:

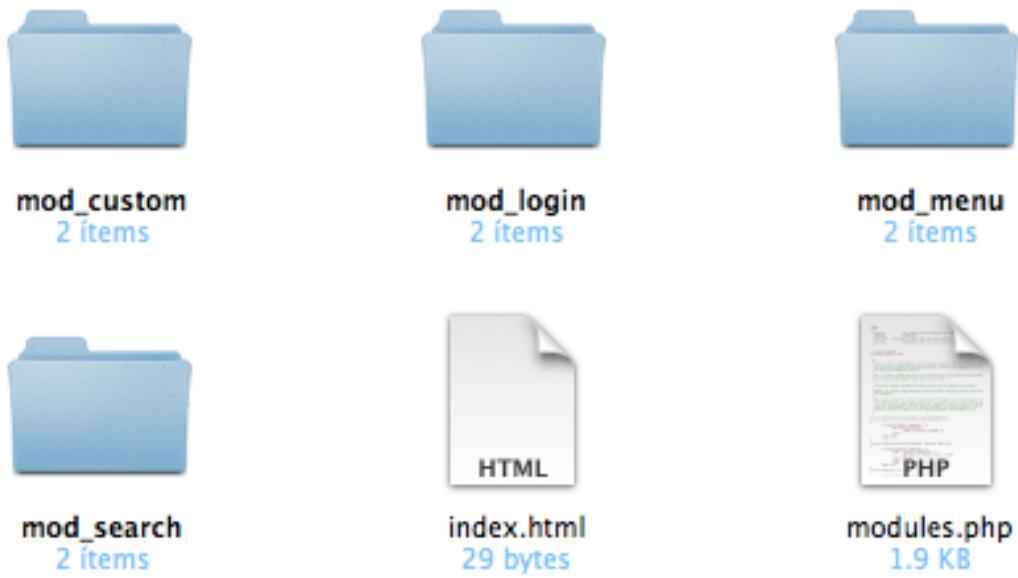


Figura 3.12: Visualización al ingresar a la carpeta 'html' de la plantilla creada

```

function modChrome_container($module, &$params, &$attribs)
{
    if (!empty ($module->content)) : ?>
        <div class="container">
            <?php echo $module->content; ?>
        </div>
    <?php endif;
}

function modChrome_bottommodule($module, &$params, &$attribs)
{
    if (!empty ($module->content)) : ?>
        <?php if ($module->showtitle) : ?>
            <h6><?php echo $module->title; ?></h6>
        <?php endif; ?>
        <?php echo $module->content; ?>
    <?php endif;
}

function modChrome_sidebar($module, &$params, &$attribs)
{
    if (!empty ($module->content)) : ?>
        <?php if ($module->showtitle) : ?>
            <h3><?php echo $module->title; ?></h3>
        <?php endif; ?>
        <?php echo $module->content; ?>
    <?php endif;
}

```

Cada función corresponde a una manera personalizada de mostrar un módulo (las cuales ya venían

definidas en la plantilla `atomic` a modo de ejemplo). Lo que se hará es crear una nueva función para definir un nuevo estilo de visualización de módulos:

```
<?php
function modChrome_personalizado($module, &$params, &$attribs)
{
    if (isset($attribs['headerLevel'])) :
        $headerLevel = $attribs['headerLevel'];
    else :
        $headerLevel = 3;
    endif; ?>

    <?php if (!empty ($module->content)) : ?>

        <?php if ($module->showtitle) : ?>

            <h<?php echo $headerLevel; ?>>
                <?php echo $module->title; ?>
            </h<?php echo $headerLevel; ?>>

        <?php endif; ?>

        <div id="module_<?php echo $module->id; ?>" class="moduledtable<?php
            echo $params->get('moduleclass_sfx'); ?>">
            <?php echo $module->content; ?>
        </div>

    <?php endif;
}

?>
```

Al descomponer la función en detalles, se obtiene un análisis más profundo:

```
<?php
function modChrome_personalizado($module, &$params, &$attribs) {
    ...
}
```

El nombre de la función debe estar conformado por `modChrome_(nombre_del_estilo)`. Entre paréntesis se pasan tres argumentos:

- `$module`: Es el objeto PHP perteneciente al módulo que se mostrará en la posición definida. Este objeto posee información relacionada al módulo: título (y si se debe mostrar), contenido, ID único;
- `&$params`: Contendrá información relacionada a los parámetros de configuración del módulo, los cuales se especifican al editarlo desde el panel de administración;
- `&$attribs`: Antes se señaló que la directiva `<jdoc:include type="modules" />` posee los atributos `type`, `name` y `style`. Sin embargo es posible agregar más atributos personalizados con el nombre y valor que se deseé. Luego, para poder obtener el valor de los nuevos atributos, se utiliza a `$attribs`. En el caso del estilo creado, se agregará un nuevo atributo llamado `headerLevel`, quedando la

directiva de esta forma: <jdoc:include type="modules" name="(nombre_de_la_posicion)" style="personalizado" headerLevel="3" />.

Nota

Notar que el atributo `style` para poder utilizar el estilo personalizado, posee el valor `personalizado` no `modChrome_personalizado` (que es el nombre de la función).

```
if (isset($attribs['headerLevel'])) :  
    $headerLevel = $attribs['headerLevel'];  
else :  
    $headerLevel = 4;  
endif;
```

En este bloque de código, se pregunta si el atributo `headerLevel` posee algún valor. Si lo posee se asigna a la variable `$headerLevel` dicho valor o en caso contrario se establece a 4 como valor predeterminado.

```
<?php if (!empty ($module->content)) : ?>  
...  
<?php endif;
```

Luego se pregunta si el módulo posee algún contenido HTML, ya que no tiene sentido mostrar un módulo vacío.

```
<?php if ($module->showtitle) : ?>  
  
    <h<?php echo $headerLevel; ?>>  
        <?php echo $module->title; ?>  
    </h<?php echo $headerLevel; ?>>  
  
<?php endif; ?>
```

Continuando, se pregunta si el módulo posee un título asignado. Si lo posee, se muestra dicho título entre etiquetas `<h />`, la cual podrá variar dependiendo del valor establecido en el atributo `headerLevel`.

```
<div id="module_<?php echo $module->id; ?>" class="moduledtable<?php echo $params  
->get('moduleclass_sfx'); ?>">  
    <?php echo $module->content; ?>  
</div>
```

Finalmente se muestra el contenido del módulo, el cual se encierra en una etiqueta `<div />` que posee el atributo `id` conformado por `module_(id_del_modulo)` y el atributo `class` por `moduledtable(parametro_moduleclass_sfx)` (recordar nuevamente que este parámetro se especifica en la configuración del módulo, en la administración del CMS. Más precisamente corresponde al parámetro *Clase CSS* del módulo).

El archivo `modules.php` quedará conformado de la siguiente forma:

```

function modChrome_container($module, &$params, &$attribs)
{
    if (!empty ($module->content)) : ?>
        <div class="container">
            <?php echo $module->content; ?>
        </div>
    <?php endif;
}

function modChrome_bottommodule($module, &$params, &$attribs)
{
    if (!empty ($module->content)) : ?>
        <?php if ($module->showtitle) : ?>
            <h6><?php echo $module->title; ?></h6>
        <?php endif; ?>
        <?php echo $module->content; ?>
    <?php endif;
}

function modChrome_sidebar($module, &$params, &$attribs)
{
    if (!empty ($module->content)) : ?>
        <?php if ($module->showtitle) : ?>
            <h3><?php echo $module->title; ?></h3>
        <?php endif; ?>
        <?php echo $module->content; ?>
    <?php endif;
}
?>
<?php
function modChrome_personalizado($module, &$params, &$attribs)
{
    if (isset($attribs['headerLevel'])) :
        $headerLevel = $attribs['headerLevel'];
    else :
        $headerLevel = 3;
    endif; ?>

    <?php if (!empty ($module->content)) : ?>

        <?php if ($module->showtitle) : ?>

            <h<?php echo $headerLevel; ?>>
                <?php echo $module->title; ?>
            </h<?php echo $headerLevel; ?>>

        <?php endif; ?>

        <div id="module_<?php echo $module->id; ?>" class="moduledtable<?php
            echo $params->get('moduleclass_sfx'); ?>">
            <?php echo $module->content; ?>
        </div>

    <?php endif;
}

?

```

Para mostrar en la plantilla los módulos de la nueva forma definida, tan solo es necesario agregar una

nueva directiva en algún lugar o reemplazar alguna por la nueva. Por ejemplo, para la sección de dos columnas, en la parte derecha, se incorpora el nuevo estilo:

```
<div id="columna_izquierda" class="span-17 append-1">
    <jdoc:include type="component" />
</div>

<div id="columna_derecha" class="span-6 last">
    <jdoc:include type="modules" name="interpoint-columna_derecha" style="personalizado" headerLevel="3" />
</div>
```

Si se verifica el código HTML generado en cada módulo derecho, se podrá notar una estructura de esta forma:

```
<h3>(título del módulo)</h3>
<div class="moduledtable(parametro moduleclass_sfx)" id="module_(id del módulo)">
    (contenido del módulo)
</div>
```

Por lo tanto, el nuevo estilo personalizado de módulo se ha aplicado correctamente.

Personalizar el HTML de un módulo en particular

Anteriormente se comentó que los módulos creados bajo el patrón MVC, poseen una subcarpeta llamada `tmp1`. Dicha carpeta contendrá uno o más archivos `.php` con las etiquetas HTML que conforman al módulo.

Por ejemplo, si se dirige a la carpeta `tmp1` del módulo `mod_stats` (módulo que muestra las estadísticas del sitio), encontrará un archivo llamado `default.php` que contiene:

```
<?php
/**
 * @version           $Id: default.php 20196 2011-01-09 02:40:25Z ian $
 * @package          Joomla.Site
 * @subpackage       mod_stats
 * @copyright        Copyright (C) 2005 - 2011 Open Source Matters, Inc. All rights reserved.
 * @license          GNU General Public License version 2 or later; see LICENSE.txt
 */
// no direct access
defined('_JEXEC') or die;
?>
<dl class="stats-module<?php echo $moduleclass_sfx ?>">
<?php foreach ($list as $item) : ?>
    <dt><?php echo $item->title;?></dt>
    <dd><?php echo $item->data;?></dd>
<?php endforeach; ?>
</dl>
```

Suponiendo el caso en el que se necesita que los datos de estadísticas del sitio, en lugar de mostrarse dentro de etiquetas `<dl />`, `<dt />` y `<dd />`, se muestren dentro de una lista desordenada (``, ``). Para realizarlo, utilizando la plantilla creada, se debe:

- En la carpeta `html` de la plantilla, crear una carpeta llamada `mod_stats`;
- Dentro de la carpeta creada, copiar los archivos `default.php` y `index.html` que estan en la carpeta `tmpl`;

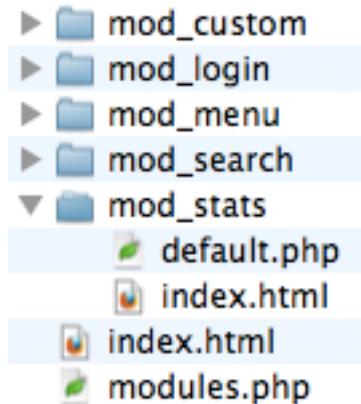


Figura 3.13: Estructura de la carpeta 'html'

Lo siguiente será abrir el archivo `default.php` copiado y modificarlo con las nuevas etiquetas:

```
<?php
// no direct access
defined('_JEXEC') or die;
?>
<ul class="stats-module<?php echo $moduleclass_sfx ?>">
<?php foreach ($list as $item) : ?>
    <li>
        <p><?php echo $item->title;?>: <?php echo $item->data;?></p>
    </li>
<?php endforeach; ?>
</ul>
```

Si se publica el módulo **estadísticas** en el sitio, se podrá ver que en lugar de utilizar las etiquetas HTML que posee el archivo `default.php` en la carpeta `tmpl`, se utilizaran las etiquetas del archivo creado en la carpeta `mod_stats` del directorio `html` de la plantilla.

En caso de querer personalizar otros módulos, se deben realizar los mismos pasos descritos.

Nota

Note que **atomic**, la plantilla utilizada como base para el trabajo, ya posee varios módulos personalizados dentro de la carpeta `html`:

- `mod_custom`
- `mod_login`
- `mod_menu`
- `mod_search`

Por lo tanto, si se necesita personalizar alguno de estos módulos, tan solo se debe modificar los archivos `.php` que estan dentro de cada directorio.

3.3. Componentes personalizados

Los componentes suelen ser extensiones más complejas que los módulos, ya que abarcan la parte principal de la página.

El mecanismo para personalizar el HTML de un componente determinado es exactamente igual que el descrito para los módulos, con algunas diferencias. Para demostrar su funcionamiento se personalizará el funcionamiento de la pantalla de resultados de búsqueda.

Para la personalización, se desea que las zonas “**Condiciones de búsqueda**” y “**Buscar solo en**” aparezcan ocultas y que éstas sean visibles al hacer clic en un enlace con el texto “**Búsqueda avanzada**”.

Para realizarlo, se hará lo siguiente:

- Ir a la carpeta del componente de búsqueda. En este caso `com_search` (la cual se encuentra dentro del directorio `components`);
- Allí dentro se encontrará a la carpeta `views`. Al entrar a ella, existirá otra carpeta con el nombre `search`.

Nota

Dependiendo del tipo de componente, es posible encontrar más de una carpeta dentro del directorio `views`. Por ejemplo, al ir a la carpeta `views` del componente `com_content` se encontrarán 6 directorios: `archive`, `article`, `categories`, `category`, `featured` y `form`. Cada carpeta es una funcionalidad distinta del componente.

- Al entrar a la carpeta `search`, se encontraran varios archivos y además un directorio con el nombre `tmpl`, el cual contiene (al igual que en el caso de los módulos) archivos `.php` con las etiquetas HTML que conforman al componente.

Nota

Note que existe más de un archivo `.php` dentro del directorio. Esto es debido a que, al ser los componentes más complejos, suelen dividir su HTML en varios archivos para reunirlos en uno solo (`default.php`).

- El archivo a copiar es `default_form.php`, ya que posee las etiquetas HTML que conforman las zonas a personalizar.
- Lo siguiente a realizar es crear una carpeta `com_search` dentro del directorio `html` de la plantilla. Allí dentro se creará otra carpeta llamada `search` para luego pegar el archivo `default_form.php` localizado en el paso anterior.

Nota

Note que se ha creado una carpeta `search`. Esta corresponde al nombre de la carpeta `search` que se encuentra dentro del directorio `views` del componente. En caso de que existan otras carpetas con otros nombres, se debe seguir la misma lógica.



- Total: encontrados 6 resultados.
- Condiciones de búsqueda:**
- Todas las palabras Cualquier palabra Frase exacta
 Orden:
- Buscar solo en:**
- Categorías Contactos Artículos Canales electrónicos Enlaces web
- Cantidad a mostrar
1. [Idiomas](#)
(Categoría)
 Joomla!, normalmente, se instala en inglés, pero podrá encontrar traducciones de la interfaz, datos de ejemplo, y pantallas de ayuda en docenas de **idiomas**. Ayuda Información sobre traducciones Si ...
 Creado el
 2. [Extensiones](#)
(Categoría)
 El gestor de contenidos Joomla! le permite crear webs de distinto tipo usando extensiones. Hay 5 tipos básicos de extensión: componentes, módulos, plantillas, **idiomas**, y plugins. Su sitio web incluye todas ...
 Creado el
 3. [Módulo selector de idioma](#)
(Módulos de utilidades)
 El módulo de selección del **idioma**, le permite aprovechar las ventajas que proporcionan las etiquetas del **idioma** al crear contenido, módulos, y menús. Este módulo muestra una lista de los **idiomas** de contenido ...
 Creado el 01 Enero 2011
 4. [Profesionales](#)
(Joomla!)
 ... Joomla! sea más ligero y rápido que nunca. Los archivos del **idioma**, ahora se pueden sobrescribir, sin tener que perder los cambios durante la actualización. Con el archivo XML apropiado, los usuarios pueden ...
 Creado el 01 Enero 2011
 5. [El proyecto Joomla!](#)
(Joomla!)
 ... por todo el mundo Software que es libre, seguro, y de calidad alta Una comunidad que sabe pasarlo bien y desea participar Gente de todo el mundo usando su **idioma** preferido Un proyecto que actúa anónimamente ...
 Creado el 01 Enero 2011
 6. [¿Qué novedades trae Joomla! 1.5?](#)
(Sin clasificar)
 ... Accesibilidad extendida para soportar internacionalización, caracteres de doble byte, soporte para escritura de derecha a izquierda para el **idioma** Árabe, Farsi, y Hebreo entre otros Integración extendida ...
 Creado el 01 Enero 2011

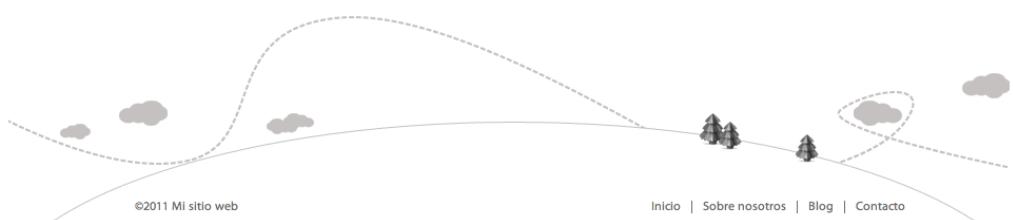


Figura 3.14: Diseño actual de la pantalla de resultados de búsqueda

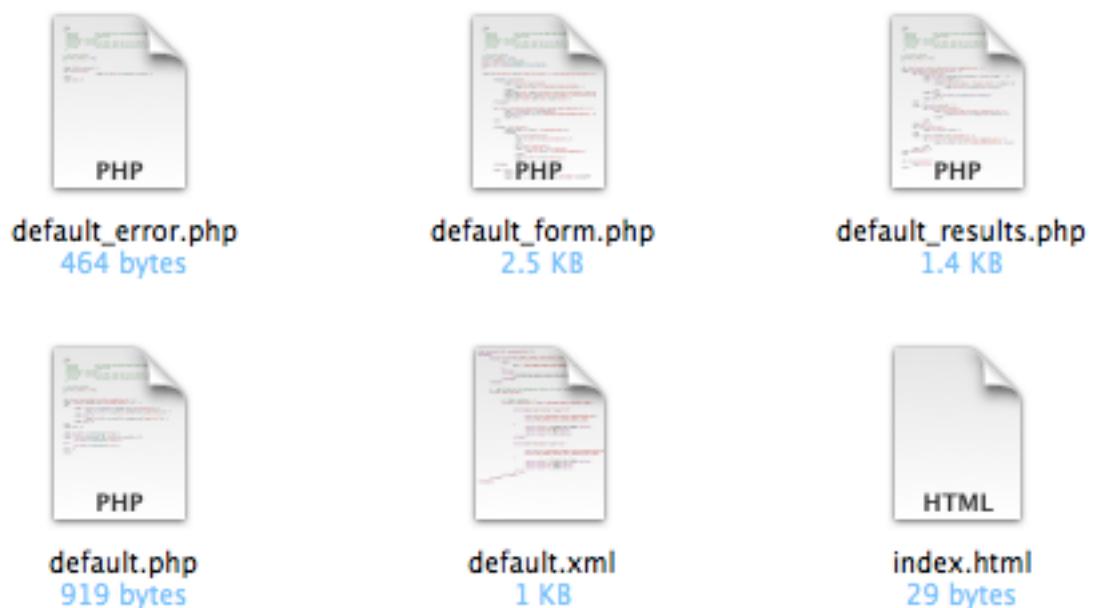


Figura 3.15: Disposición de archivos en la carpeta ‘tmpl’

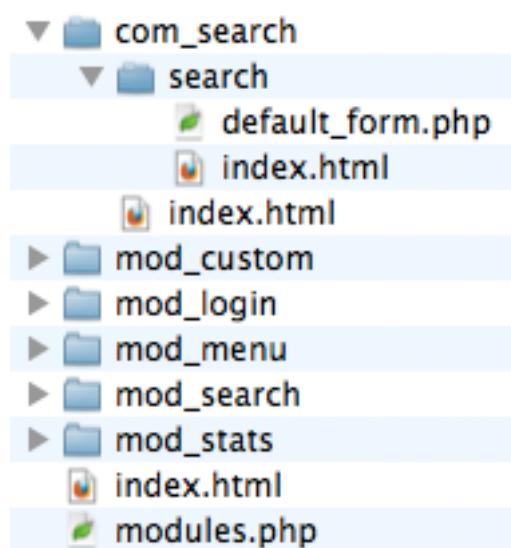


Figura 3.16: Estructura del directorio ‘html’ de la plantilla

Lo siguiente será abrir el archivo `default_form.php` copiado, y modificarlo al gusto.

Las etiquetas que corresponden a la zona a personalizar son:

```
<fieldset class="phrases">
    <legend><?php echo JText::_('COM_SEARCH_FOR');?>
    </legend>
        <div class="phrases-box">
            <?php echo $this->lists['searchphrase']; ?>
        </div>
        <div class="ordering-box">
            <label for="ordering" class="ordering">
                <?php echo JText::_('COM_SEARCH_ORDERING');?>
            </label>
            <?php echo $this->lists['ordering'];?>
        </div>
</fieldset>

<?php if ($this->params->get('search_areas', 1)) : ?>
    <fieldset class="only">
        <legend><?php echo JText::_('COM_SEARCH_SEARCH_ONLY');?></legend>
        <?php foreach ($this->searchareas['search'] as $val => $txt) :
            $checked = is_array($this->searchareas['active']) && in_array($val
                , $this->searchareas['active']) ? 'checked="checked"' : '';
        ?>
        <input type="checkbox" name="areas[]" value="<?php echo $val;?>" id="area
            -<?php echo $val;?>" <?php echo $checked;?> />
            <label for="area-<?php echo $val;?>">
                <?php echo JText::_($txt); ?>
            </label>
        <?php endforeach; ?>
    </fieldset>
<?php endif; ?>
```

Como se comentó, se desea ocultar estas zonas de manera predeterminada y agregar un enlace, que al hacerle clic, haga visibles las opciones de búsqueda.

Lo primero será crear el enlace y además encerrar las zonas que contienen las opciones en un `<div />`:

```
<p>
    <a href="#" id="search-advanced-link">Búsqueda avanzada</a>
</p>

<div id="search-advanced">

    <fieldset class="phrases">
        <legend><?php echo JText::_('COM_SEARCH_FOR');?>
        </legend>
            <div class="phrases-box">
                <?php echo $this->lists['searchphrase']; ?>
            </div>
            <div class="ordering-box">
                <label for="ordering" class="ordering">
                    <?php echo JText::_('COM_SEARCH_ORDERING');?>
                </label>
                <?php echo $this->lists['ordering'];?>
            </div>
    </fieldset>
```

```

<?php if ($this->params->get('search_areas', 1)) : ?>
    <fieldset class="only">
        <legend><?php echo JText::_('COM_SEARCH_SEARCH_ONLY');?></legend>
        <?php foreach ($this->searchareas['search'] as $val => $txt) :
            $checked = is_array($this->searchareas['active']) && in_array(
                $val, $this->searchareas['active']) ? 'checked="checked"' :
                '';
        ?>
        <input type="checkbox" name="areas[]" value="<?php echo $val;?>" id="area-<?php echo $val;?>" <?php echo $checked;?> />
        <label for="area-<?php echo $val;?>">
            <?php echo JText::_($txt); ?>
        </label>
        <?php endforeach; ?>
    </fieldset>
<?php endif; ?>

</div>

```

Luego, se incorporan unos estilos CSS para darle más presencia al enlace y una función JavaScript que realizará el trabajo de ocultar/mostrar las opciones de búsqueda:

```

<?php
$doc = JFactory::getDocument();
$js = "window.addEvent('domready', function() {
    var opciones = new Fx.Slide('search-advanced');

    opciones.hide();

    $('#search-advanced-link').addEvent('click', function(e){
        e = new Event(e);
        opciones.toggle();
        e.stop();
    });
});

$css = "#search-advanced-link{
font-weight: bold;
font-size: 16px;
}";

$doc->addStyleDeclaration ($css);
$doc->addScriptDeclaration ($js);
?>

```

Nota

Note que, para insertar el estilo CSS y la función JavaScript, ha utilizado `JFactory::getDocument()`, `addStyleDeclaration()` y `addScriptDeclaration()`. El beneficio de utilizar estas funciones es que Joomla se encarga de insertar todo el código dentro de las etiquetas `<head />` de la plantilla.

Más información sobre:

- `JDocument`: <http://docs.joomla.org/JDocument>
- `getDocument`: <http://docs.joomla.org/JFactory::getDocument/>

- `addScriptDeclaration`: <http://docs.joomla.org/JDocument/addScriptDeclaration>
- `addStyleDeclaration`: <http://docs.joomla.org/JDocument/addStyleDeclaration>

Otra manera de realizar el trabajo es utilizando las etiquetas `<style type="text/css" />` y `<script type="text/javascript" />`. El funcionamiento sería el mismo, pero todo el código quedaría dentro del `<body />` del documento.

Nota

Para realizar la funcionalidad JavaScript, se ha aprovechado que Joomla utiliza **Mootools** como framework JavaScript. Esta biblioteca posee una clase JavaScript llamada `Fx.Slide` que permite realizar el tipo de efecto se necesita. Más información sobre `Fx.Slide`: <http://mootools.net/docs/more/Fx/Fx.Slide>

Nota

Una buena práctica es incorporar el texto **Búsqueda avanzada** dentro de los archivos del idioma de la plantilla y no en el mismo archivo `.php`.

De esta manera ya se tiene terminada la personalización del componente. En caso que se quiera realizar lo mismo con otros componentes se deben realizar los mismos pasos descritos anteriormente.

3.3.1. Conclusiones

Se ha podido comprobar el potencial de las plantillas de Joomla para la personalización de componentes y módulos. Esta práctica posee varias ventajas importantes:

- Adaptación a las necesidades sin mucho esfuerzo;
- No se modificó ningún archivo del núcleo de la extensión permitiendo que, en caso de actualización, no se pierda el trabajo realizado, minimizando el tiempo de mantenimiento;
- Centralización del trabajo: el mismo transcurrió dentro de la carpeta `html` de la plantilla.

3.4. Incorporación de jQuery

Anteriormente se mostró el aprovechamiento de **Mootools** para facilitar el armado de funciones JavaScript. Sin embargo, puede suceder que se necesite utilizar a **jQuery**, otra popular biblioteca JavaScript para dotar de funcionalidades a la plantilla.

Nota

Más información sobre jQuery: <http://jquery.com/> Aprender a utilizar jQuery: <http://librojquery.com/>

 **Joomla!**TM
Mi sitio web y Joomla!

[Ingresar](#) [Registro](#)

Inicio Sobre nosotros Clientes Servicios Blog Características Contacto

Buscar palabras clave:

Total: encontrados 6 resultados.

[Búsqueda avanzada](#)

Cantidad a mostrar

- 1. [Idiomas](#)
(Categoría)
Joomla!, normalmente, se instala en inglés, pero podrá encontrar traducciones de la interfaz, datos de ejemplo, y pantallas de ayuda en docenas de **idiomas**. Ayuda Información sobre traducciones Si ...
Creado el
- 2. [Extensiones](#)
(Categoría)
El gestor de contenidos Joomla! le permite crear webs de distinto tipo usando extensiones. Hay 5 tipos básicos de extensión: componentes, módulos, plantillas, **idiomas**, y plugins. Su sitio web incluye todas ...
Creado el
- 3. [Módulo selector de idioma](#)
(Módulos de utilidades)
El módulo de selección del **idioma**, le permite aprovechar las ventajas que proporcionan las etiquetas del **idioma** al crear contenido, módulos, y menús. Este módulo muestra una lista de los **idiomas** de contenido ...
Creado el 01 Enero 2011
- 4. [Profesionales](#)
(Joomla!)
... Joomla! sea más ligero y rápido que nunca. Los archivos del **idioma**, ahora se pueden sobrescribir, sin tener que perder los cambios durante la actualización. Con el archivo XML apropiado, los usuarios pueden ...
Creado el 01 Enero 2011
- 5. [El proyecto Joomla!](#)
(Joomla!)
... por todo el mundo Software que es libre, seguro, y de calidad alta Una comunidad que sabe pasarlo bien y desea participar Gente de todo el mundo usando su **idioma** preferido Un proyecto que actúa anónimamente ...
Creado el 01 Enero 2011
- 6. [Qué novedades trae Joomla! 1.5?](#)
(Sin clasificar)
... Accesibilidad extendida para soportar internacionalización, caracteres de doble byte, soporte para escritura de derecha a izquierda para el **idioma** Árabe, Farsi, y Hebreo entre otros Integración extendida ...
Creado el 01 Enero 2011

Buscar...

Artículos más leídos

Parques australianos
Visitámos
Tienda de frutas

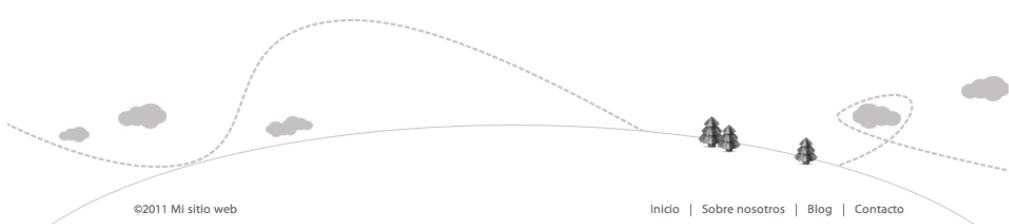
Datos de acceso

Usuario

Contraseña

Recuérdeme

[Recordar contraseña?](#)
[Recordar usuario?](#)
[Crear una cuenta](#)



©2011 Mi sitio web

Inicio | Sobre nosotros | Blog | Contacto

Figura 3.17: Diseño final de la página de resultados

Búsqueda avanzada

Condiciones de búsqueda:

Todas las palabras Cualquier palabra Frase exacta
Orden:

Buscar solo en:

Categorías Contactos Artículos Canales electrónicos Enlaces web

Figura 3.18: Diseño de la página de resultados al hacer clic en el enlace ‘Búsqueda avanzada’

Los pasos para incorporar jQuery en la plantilla son los siguientes:

- Descargar la biblioteca jQuery: <http://ajax.googleapis.com/ajax/libs/jquery/1/jquery.min.js>
- Guardar el archivo jquery.min.js dentro de la carpeta js de la plantilla;
- Abrir el index.php de la plantilla e insertar a la biblioteca dentro del <head />, después de la directiva <jdoc:include type="head" /> pero antes de la llamada al archivo template.js:

```
<jdoc:include type="head" />
<link rel="stylesheet" href="php echo $this-&gt;baseurl ?&gt;/templates/&lt;?php echo
    $this-&gt;template ?&gt;/css/blueprint/screen.css" type="text/css" media="screen,
    projection" /&gt;
&lt;link rel="stylesheet" href="<?php echo $this-&gt;baseurl ?&gt;/templates/&lt;?php echo
    $this-&gt;template ?&gt;/css/blueprint/print.css" type="text/css" media="print" /&gt;
&lt;!--[if lt IE 8]&gt;&lt;link rel="stylesheet" href="blueprint/ie.css" type="text/css"
    media="screen, projection"&gt;&lt;![endif]--&gt;

&lt;link rel="stylesheet" href="<?php echo $this-&gt;baseurl ?&gt;/templates/&lt;?php echo
    $this-&gt;template ?&gt;/css/template.css" type="text/css" /&gt;

&lt;script type="text/javascript" src="<?php echo $this-&gt;baseurl ?&gt;/templates/&lt;?php
    echo $this-&gt;template ?&gt;/js/jquery.min.js"&gt;&lt;/script&gt;
&lt;script type="text/javascript" src="<?php echo $this-&gt;baseurl ?&gt;/templates/&lt;?php
    echo $this-&gt;template ?&gt;/js/template.js"&gt;&lt;/script&gt;</pre
```

- Abrir template.js y escribir en la primera línea:

```
jQuery.noConflict();
```

Nota

La línea anterior permite que jQuery funcione sin problemas con otras bibliotecas JavaScript (como es el caso de Mootools). Hay que tener en cuenta que, cuando se escriban funciones jQuery, en lugar de utilizar el símbolo \$, se deberá utilizar la palabra `jQuery`. Por ejemplo, la siguiente función:

```
$("#footer").hide();
```

Pasa a ser:

```
jQuery("#footer").hide();
```

Más información sobre `jQuery.noConflict()`: <http://api.jquery.com/jQuery.noConflict/>

Y listo. De esta forma es posible utilizar jQuery sin problemas en la plantilla.

3.4.1. Creación de un slideshow con jQuery

Aprovechando que se incorporó a jQuery en la plantilla, se mostrará la manera de crear un slideshow de imágenes para completar la funcionalidad planeada en la página de inicio:



Figura 3.19: Diseño del slideshow

Para esto, se utilizará la extensión de la biblioteca **jQuery Cycle Plugin**: <http://jquery.malsup.com/cycle/>

Recordar que la zona que conforma al slideshow en la plantilla esta compuesta de la siguiente forma:

```
<div id="slideshow_inferior" class="span-8 last">
    <?php if($this->countModules('interpoint-slideshow_inferior')) : ?>
        <div id="slideshow_contenedor">
```

```

<jdoc:include type="modules" name="interpoint-slideshow_inferior
    " style="none" />
</div>
<span id="slideshow_der" class="slideshow_nav"></span>
<span id="slideshow_izq" class="slideshow_nav"></span>
<?php endif; ?>
</div>

```

La posición `interpoint-slideshow_inferior` mostrará a un módulo del tipo HTML personalizado, en donde se incorporarán varias imágenes para hacer funcionar al slideshow. Al insertar las imágenes, el HTML final queda de la siguiente manera:

```

<div id="slideshow_inferior" class="span-8 last">
    <div id="slideshow_contenedor">

        <p></p>
        <p></p>
        <p></p>
        <p></p>

    </div>
    <span id="slideshow_der" class="slideshow_nav"></span>
    <span id="slideshow_izq" class="slideshow_nav"></span>
</div>

```

Es probable que al hacerlo, todo el slideshow se vea mal. Esto por ahora no importa, ya que luego, al establecer las funciones relativas al slideshow el problema se habrá solucionado.

Lo siguiente a realizar será descargar **jQuery Cycle Plugin**: <http://jquery.malsup.com/cycle/download.html>.

Descomprimir el archivo .zip descargado y copiar el archivo `jquery.cycle.all.min.js` en la carpeta `js` de la plantilla, de forma que quede en conjunto con `jquery.min.js` y `template.js`:

Luego, insertar a `jquery.cycle.all.min.js` en la plantilla. La llamada al archivo debe quedar por debajo de `jquery.min.js` pero antes de `template.js`:

```

<script type="text/javascript" src="<?php echo $this->baseurl ?>/templates/<?php
    echo $this->template ?>/js/jquery.min.js"></script>
<script type="text/javascript" src="<?php echo $this->baseurl ?>/templates/<?php
    echo $this->template ?>/js/jquery.cycle.all.min.js"></script>
<script type="text/javascript" src="<?php echo $this->baseurl ?>/templates/<?php
    echo $this->template ?>/js/template.js"></script>

```

Por último, en `template.js` se establecen las funciones para hacer funcionar el slideshow. El archivo quedará de la siguiente forma:

```

jQuery.noConflict();
/*Funciones para el slideshow*/
jQuery(document).ready(function() {
    jQuery('#slideshow_contenedor').cycle({
        fx:          'scrollHorz',
        timeout:    3000,
        prev:        '#slideshow_izq',

```

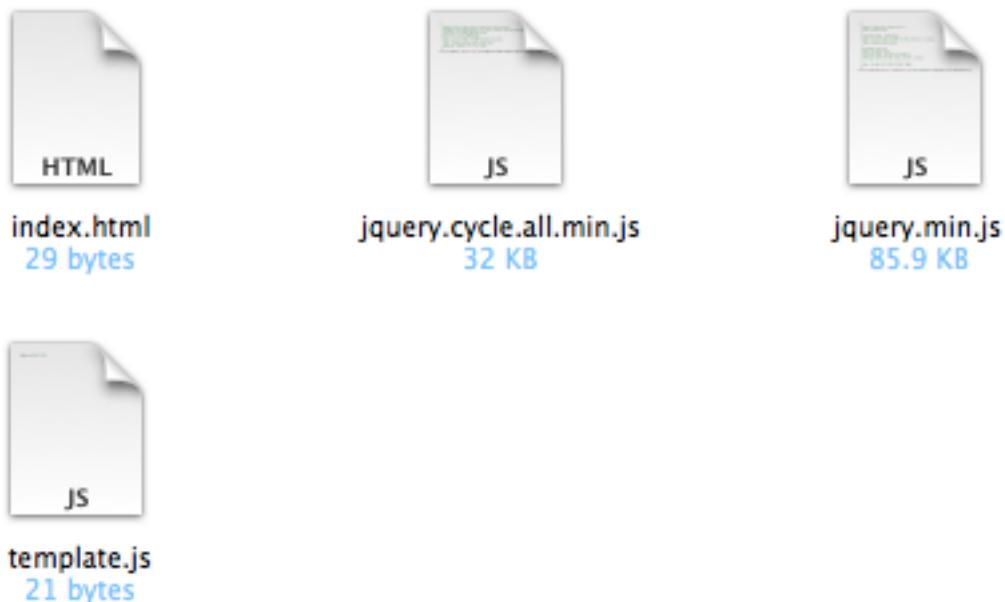


Figura 3.20: Archivos dentro de la carpeta ‘js’

```

        next:      '#slideshow_der'
    });
}

```

La creación del slideshow es muy fácil, tan solo se debe especificar:

- El ID del <div /> contenedor de las imágenes: `#slideshow_contenedor`;
- El efecto a utilizar entre transiciones: `scrollHorz`;
- El tiempo de transición entre imágenes: 3000 milisegundos;
- Los IDs de los elementos para pasar a la siguiente o anterior imagen: `#slideshow_izq` y `#slideshow_der`.

Más información y otras opciones sobre la utilización de jQuery Cycle Plugin: <http://jquery.malsup.com/cycle/>

3.5. Personalización de las páginas de errores

Una de los aspectos al que menos se suele prestar atención en el momento de crear un sitio son las páginas de errores. Éstas son muy importantes, ya que, en caso de presentarse un error ante el usuario, se lo debe incentivar a que no abandone la página y continúe en ella.

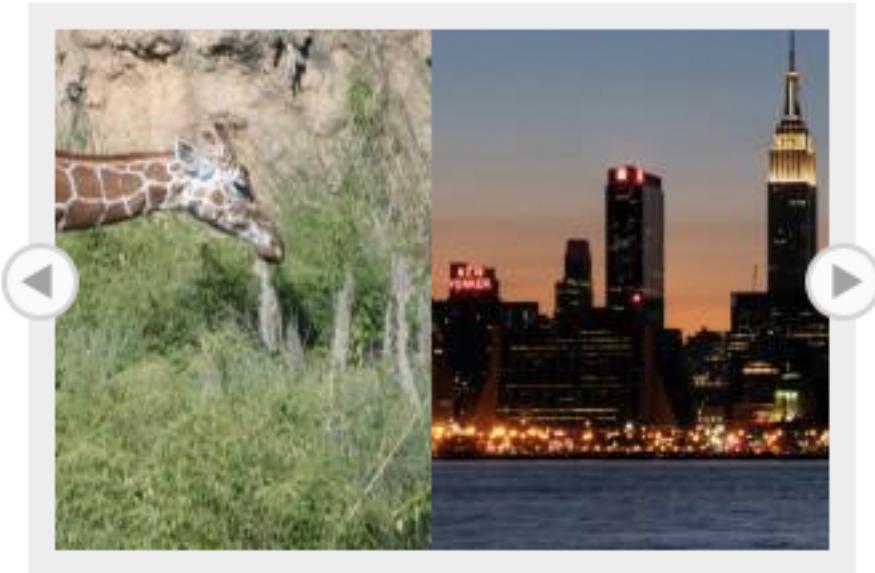


Figura 3.21: Slideshow en funcionamiento

Cada error siempre va acompañado por un código de estado HTTP. Los tipos de errores más comunes son:

- **404 - Página no encontrada:** Sucede cuando el usuario quiere acceder a una página que no existe más;
- **500 - Error interno:** Sucede al existir un error interno del servidor web.

Nota

Más información sobre los códigos de estado HTTP: http://es.wikipedia.org/wiki/Anexo:C%C3%B3digos_de_estado_HTTP

Como se puede observar, no posee un aspecto “muy amigable”.

Lo que se hará es modificar la página de errores para que herede el diseño del sitio. Para esto, hacerlo es necesario modificar el archivo `error.php` que se encuentra en la carpeta raíz de la plantilla. Al abrir el archivo se encontrará lo siguiente:

```
<?php
/**
 * @version           $Id: error.php 20196 2011-01-09 02:40:25Z ian $
 * @package          Joomla.Site
 * @copyright        Copyright (C) 2005 - 2011 Open Source Matters, Inc. All
 *                   rights reserved.
 * @license          GNU General Public License version 2 or later; see
 *                   LICENSE.txt
 */
defined('_JEXEC') or die;
```

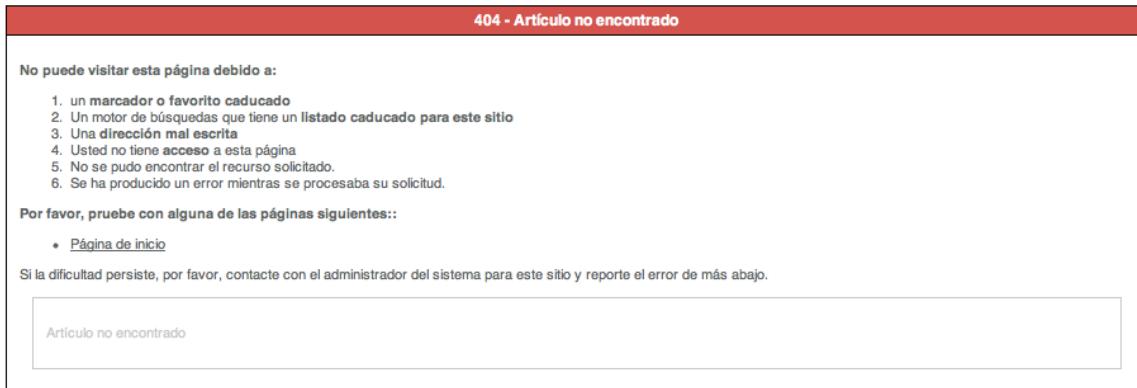


Figura 3.22: Diseño predeterminado de la página de error 404 en Joomla

```

if (!isset($this->error)) {
    $this->error = JError::raiseWarning(404, JText::_('JERROR_ALERTNOAUTHOR'))
    );
    $this->debug = false;
}
?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org
/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang=<?php echo $this->language;
?>" lang=<?php echo $this->language; ?>" dir=<?php echo $this->direction;
?>">
<head>
    <title><?php echo $this->error->getCode(); ?> - <?php echo $this->title;
    ?></title>
    <link rel="stylesheet" href=<?php echo $this->baseurl; ?>/templates/
        system/css/error.css" type="text/css" />
</head>
<body>
    <div class="error">
        <div id="outline">
            <div id="errorboxoutline">
                <div id="errorboxheader"><?php echo $this->error->getCode
                    (); ?> - <?php echo $this->error->getMessage(); ?></
                    div>
                <div id="errorboxbody">
                    <p><strong><?php echo JText::_('
                        JERROR_LAYOUT_NOT_ABLE_TO_VISIT'); ?></strong></p>
                    <ol>
                        <li><?php echo JText::_('
                            JERROR_LAYOUT_AN_OUT_OF_DATE_BOOKMARK_FAVOURITE
                            '); ?></li>
                        <li><?php echo JText::_('
                            JERROR_LAYOUT_SEARCH_ENGINE_OUT_OF_DATE_LISTING
                            '); ?></li>
                        <li><?php echo JText::_('
                            JERROR_LAYOUT_MIS_TYPED_ADDRESS');
                            ?></li>
                        <li><?php echo JText::_('

```

```

        JERROR_LAYOUT_YOU_HAVE_NO_ACCESS_TO_THIS_PAGE
        '); ?></li>
    <li><?php echo JText::_('
        JERROR_LAYOUT_REQUESTED_RESOURCE_WAS_NOT_FOUND
        '); ?></li>
    <li><?php echo JText::_('
        JERROR_LAYOUT_ERROR_HAS_OCCURRED_WHILE_PROCESSING_YOU
        '); ?></li>
</ol>
<p><strong><?php echo JText::_('
        JERROR_LAYOUT_PLEASE_TRY_ONE_OF_THE_FOLLOWING_PAGES'
        ; ?></strong></p>
<ul>
    <li><a href=<?php echo $this->baseurl;
        ?>/index.php" title=<?php echo JText
        ::_('
        JERROR_LAYOUT_GO_TO_THE_HOME_PAGE');
        ?>"><?php echo JText::_('
        JERROR_LAYOUT_HOME_PAGE'); ?></a></li
    >
</ul>
<p><?php echo JText::_('
        JERROR_LAYOUT_PLEASE_CONTACT_THE_SYSTEM_ADMINISTRATOR
        '); ?></p>
<div id="techinfo">
<p><?php echo $this->error->getMessage(); ?></p>
<p>
    <?php if ($this->debug) :
        echo $this->renderBacktrace();
    endif; ?>
</p>
</div>
</div>
</div>
</body>
</html>
```

Lo primero a hacer es borrar el contenido e incorporar una estructura similar a la creada en `index.php`.

Nota

Es importante entender que `error.php` no funciona de forma exactamente igual a `index.php`. Por ejemplo, los parámetros de la plantilla se acceden de diferente forma y no es posible insertar módulos y contenidos de la manera anteriormente vista (a través de la directiva `<jdoc:include />`).

```

<?php
defined('_JEXEC') or die;

$app = JFactory::getApplication();
$params = JFactory::getApplication()->getTemplate(true)->params;
```

```

//Parametros de la plantilla
$imagenLogo = $params->get('logo');
$eslogan = $params->get('eslogan');

?>

<?php echo '<?' ; ?>xml version="1.0" encoding="<?php echo $this->_charset ?>"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/
xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="<?php echo $this->language;
?>" lang="<?php echo $this->language; ?>" dir="<?php echo $this->direction;
?>" >

<head>
    <title><?php echo $this->error->getCode(); ?> - <?php echo $this->title;
    ?></title>

    <link rel="stylesheet" href="<?php echo $this->baseurl ?>/templates/<?php
        echo $this->template ?>/css/blueprint/screen.css" type="text/css" media
        ="screen, projection" />
    <link rel="stylesheet" href="<?php echo $this->baseurl ?>/templates/<?php
        echo $this->template ?>/css/blueprint/print.css" type="text/css" media="
        print" />
    <!--[if lt IE 8]><link rel="stylesheet" href="blueprint/ie.css" type="text/
        css" media="screen, projection"><![endif]-->

    <link rel="stylesheet" href="<?php echo $this->baseurl ?>/templates/<?php
        echo $this->template ?>/css/template.css" type="text/css" />

    <?php if($imagenLogo) : ?>

        <style type="text/css">

            #logo h1{
                background: transparent url(<?php echo $this->baseurl ."/".
                $imagenLogo; ?>) no-repeat left top;
            }

        </style>

    <?php endif; ?>

</head>

<body>
    <div class="container">

        <!-- Comienzo Header -->
        <div id="header" class="span-24">

            <!-- Comienzo Logo -->
            <div id="logo" class="span-9 prepend-8">
                <a href="<?php echo $this->baseurl ?>" title="<?php echo $app->getCfg
                    ('sitename'); ?>">
                    <h1><?php echo $app->getCfg('sitename'); ?></h1>
                </a>
            </div>
        </div>
    </div>

```

```

<h2 id="eslogan">
    <?php if($eslogan) : ?>
        <?php echo $eslogan; ?>
    <?php else : ?>
        Lorem ipsum dolor sit amet
        <?php endif; ?>
    </h2>
</div>

</div>
<!-- Fin Header -->

<!-- Comienzo Error -->
<div id="error" class="span-18 push-3">

    <!-- Error 404 -->
    <?php if ($this->error->getCode() == 404) : ?>

        <p>La página que esta buscando ya no se encuentra disponible.</p>
        <p>Lo invitamos a realizar un búsqueda...</p>

        <?php
            $module = JModuleHelper::getModule('search');
            echo JModuleHelper::renderModule($module);
        ?>

        <p>o ir a la <a href="php echo $this-&gt;baseurl ?&gt;" title="<?php echo
            $app-&gt;getCfg('sitename'); ?&gt;">página principal del sitio</a>.</p>
        >

    <!-- Error 500 -->
    <?php elseif ($this->error->getCode() == 500) : ?>

        <p>Un error desconocido ha ocurrido.</p>
        <p>Mientras solucionamos el asunto, lo invitamos a continuar por la <
            a href="php echo $this-&gt;baseurl ?&gt;" title="<?php echo $app-&gt;
            getCfg('sitename'); ?&gt;">página principal del sitio</a>.</p>

        <div id="error_descripcion">
            <p>Descripción del error:</p>
            <p><?php echo $this->error->getMessage(); ?></p>
        </div>

    <?php endif; ?>

</div>
<!-- Fin Error -->

<!-- Comienzo Footer -->
<div id="footer" class="span-24">
    <div id="legal" class="prepend-3 span-9">
        &copy; <?php echo date('Y'); ?> <?php echo $app->getCfg('sitename');
        ?>
    </div>
</div>
<!-- Fin Footer -->

```

```

</div>
</body>

</html>
```

Para entender de mejor manera lo incorporado se detallará cada sección:

3.5.1. Sección 1: Parámetros de la plantilla

```

<?php

defined('_JEXEC') or die;

$app = JFactory::getApplication();
$params = JFactory::getApplication()->getTemplate(true)->params;

//Parámetros de la plantilla
$imagenLogo = $params->get('logo');
$eslogan = $params->get('eslogan');

?>
```

A igual que en `index.php`, se necesita acceder a los valores de los parámetros de la plantilla. Sin embargo, aquí se realiza diferente forma, utilizando `getApplication()`, el cual posee el objeto PHP global de la aplicación.

Nota

Más información sobre `getApplication`: <http://docs.joomla.org/JFactory:getApplication>

3.5.2. Sección 2: Tipo de documento, título y archivos CSS

```

<?php echo '<?' : ?>xml version="1.0" encoding="<?php echo $this->_charset ?>"?
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/
xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="<?php echo $this->language;
?>" lang="<?php echo $this->language; ?>" dir="<?php echo $this->direction;
?>" >

<head>
    <title><?php echo $this->error->getCode(); ?> - <?php echo $this->title;
    ?></title>

    <link rel="stylesheet" href="<?php echo $this->baseurl ?>/templates/<?php
        echo $this->template ?>/css/blueprint/screen.css" type="text/css" media
        ="screen, projection" />
    <link rel="stylesheet" href="<?php echo $this->baseurl ?>/templates/<?php
        echo $this->template ?>/css/blueprint/print.css" type="text/css" media="print" />
```

```

<!--[if lt IE 8]><link rel="stylesheet" href="blueprint/ie.css" type="text/
css" media="screen, projection"><![endif]-->

<link rel="stylesheet" href="php echo $this-&gt;baseurl ?&gt;/templates/&lt;?php
echo $this-&gt;template ?&gt;/css/template.css" type="text/css" /&gt;

&lt;?php if($imagenLogo) : ?&gt;

&lt;style type="text/css"&gt;

#logo h1{
    background: transparent url(&lt;?php echo $this-&gt;baseurl ."/".
$imagenLogo; ?&gt;) no-repeat left top;
}

&lt;/style&gt;

&lt;?php endif; ?&gt;

&lt;/head&gt;
</pre

```

Luego, se crea el `Doctype` del documento, se inserta el título y los archivos CSS de la plantilla. Los archivos JavaScript no son necesarios, por lo tanto no se incorporan.

Nota

Note que se utiliza `$this->error->getCode()` para obtener el código del error actual de la página y así mostrarlo en el título de la página.

3.5.3. Sección 3: Logotipo

```

<body>
<div class="container">

<!-- Comienzo Header -->
<div id="header" class="span-24">

    <!-- Comienzo Logo -->
    <div id="logo" class="span-9 prepend-8">
        <a href="php echo $this-&gt;baseurl ?&gt;" title="<?php echo $app-&gt;getCfg
('sitename'); ?&gt;""
            <h1><?php echo $app->getCfg('sitename'); ?></h1>
        </a>
        <h2 id="eslogan">
            <?php if($eslogan) : ?>
                <?php echo $eslogan; ?>
            <?php else : ?>
                Lorem ipsum dolor sit amet
            <?php endif; ?>
        </h2>
    </div>

</div>
<!-- Fin Header -->

```

Para la cabecera, se mostrará el logotipo del sitio y se alinearán en el centro.

3.5.4. Sección 4: Mensajes de error

```
<!-- Comienzo Error -->
<div id="error" class="span-18 push-3">

<!-- Error 404 -->
<?php if ($this->error->getCode() == 404) : ?>

<p>La página que esta buscando ya no se encuentra disponible.</p>
<p>Lo invitamos a realizar un búsqueda...</p>

<?php
    $module = JModuleHelper::getModule('search');
    echo JModuleHelper::renderModule($module);
?>

<p>o ir a la <a href="php echo $this-&gt;baseurl ?&gt;" title="<?php echo
    $app-&gt;getCfg('sitename'); ?&gt;"página principal del sitio</a>.</p>

<!-- Error 500 -->
<?php elseif ($this->error->getCode() == 500) : ?>

<p>Un error desconocido ha ocurrido.</p>
<p>Mientras solucionamos el asunto, lo invitamos a continuar por la <a
    href="php echo $this-&gt;baseurl ?&gt;" title="<?php echo $app-&gt;getCfg(
        'sitename'); ?&gt;"página principal del sitio</a>.</p>

<div id="error_descripcion">
    <p>Descripción del error:</p>
    <p><?php echo $this->error->getMessage(); ?></p>
</div>

<?php endif; ?>

</div>
<!-- Fin Error -->
```

Esta sección es la más importante. En ella existen dos bloques condicionales:

- En caso que la página no sea encontrada (404), se mostrará un mensaje conveniente e se invitará al usuario a realizar una búsqueda o continuar por la página principal del sitio. Algo importante es que la caja de búsqueda se incorpora a través de la directiva `JModuleHelper::getModule()` en la cual entre parentesis se le indica el tipo de módulo a mostrar (en este caso `search`).

Nota

Más información sobre `getModule`: <http://docs.joomla.org/JModuleHelper::getModule>

- En caso que la página posea un error interno (500), también se mostrará un mensaje conveniente invitando al usuario a continuar por la página principal. Por otro lado, para tener de referencia, se mostrará el tipo de error que se produjo a través de la directiva `$this->error->getMessage()`.

Nota

Una buena práctica es no incorporar los textos en nuestro idioma dentro del archivo, sino utilizar los archivos del idioma de la plantilla. Esto se puede realizar una vez comprobado que los textos a mostrar son los correctos.

3.5.5. Sección 5: Footer

```
<!-- Comienzo Footer -->
<div id="footer" class="span-24">
    <div id="legal" class="prepend-3 span-9">
        &copy;<?php echo date('Y'); ?> <?php echo $app->getCfg('sitename');
    ?>
    </div>
</div>
<!-- Fin Footer -->

</div>
</body>

</html>
```

El documento se termina mostrando, en el pie, el nombre del sitio y el año corriente.

Por otro lado, es necesario escribir algunos estilos CSS para los elementos de la página de error. En `template.css` se incorpora:

```
/* Paginas de error
-----
#error {
    text-align: center;
    padding: 50px 0;
}

#error p{
    font-size: 25px;
    margin: 0;
}

#error input{
    font-size: 25px;
    border: 5px solid #ccc;
}

#error_descripcion{
    border: 1px solid #CCC;
    margin: 20px 0;
}

#error_descripcion p{
    font-size: 9px;
}
```



La página que esta buscando ya no se encuentra disponible.
Lo invitamos a realizar una búsqueda...

Buscar..

ó ir a la [página principal del sitio.](#)

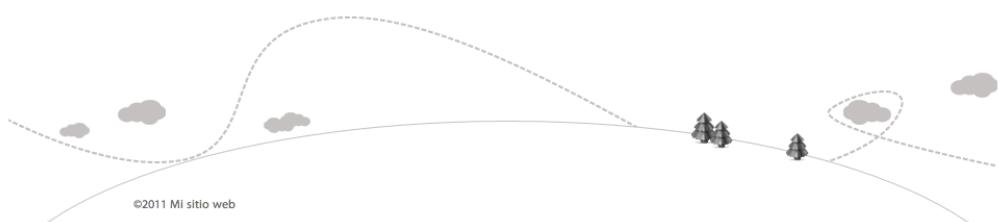


Figura 3.23: Diseño final de la página de error 404



Un error desconocido ha ocurrido.

Mientras solucionamos el asunto, lo invitamos a continuar por
[la página principal del sitio](#).

Descripción del error:
DatabaseMySQL:query: 1146 - Table 'joomla16.jos_content_frontpage' doesn't exist SQL=SELECT a.id, a.title, a.alias, a.title_alias, a.introtext, a.checked_out, a.checked_out_time, a.catid, a.created, a.created_by, a.created_by_alias, CASE WHEN a.modified = 0 THEN a.created ELSE a.modified END as modified, a.modified_by, uam.name as modified_by_name,CASE WHEN a.publish_up = 0 THEN a.created ELSE a.publish_up END as publish_up, a.publish_down, a.attribs, a.metadata, a.metakey, a.metadesc, a.access, a.hits, a.xreference, a.featured, LENGTH(a.fulltext) AS readmore, CASE WHEN badcats.id is not null THEN 0 ELSE a.state END AS state,c.title AS category_title, c.path AS category_route, c.access AS category_access, c.alias AS category_alias,CASE WHEN a.created_by_alias > '' THEN a.created_by_alias ELSE ua.name END AS author,ua.email AS author_email,contact.id as contactid,parent.title as parent_title, parent.id as parent_id, parent.path as parent_route, parent.alias as parent_alias,ROUND(v.rating_sum / v.rating_count) AS rating, v.rating_count as rating_count,c.published, CASE WHEN badcats.id = null THEN c.published ELSE 0 END AS parents_published FROM jos_content AS a LEFT JOIN jos_categories AS c ON c.id = a.catid LEFT JOIN jos_users AS ua ON ua.id = a.created_by LEFT JOIN jos_users AS um ON um.id = a.modified_by LEFT JOIN jos_contact_details AS contact on contact.user_id = a.created_by LEFT JOIN jos_categories as parent ON parent.id = c.parent_id LEFT JOIN jos_content_rating AS v ON a.id = v.content_id LEFT OUTER JOIN (SELECT cat.id AS id FROM jos_categories AS cat JOIN jos_categories AS parent ON cat.lft BETWEEN parent.lft AND parent.rgt WHERE parent.extension = 'com_content' AND parent.published != 1 GROUP BY cat.id) AS badcats ON badcats.id = c.id INNER JOIN jos_content_frontpage AS fp ON fp.content_id = a.id WHERE a.access IN (1,1) AND CASE WHEN badcats.id is null THEN a.state ELSE 0 END = 1 AND (a.publish_up = '0000-00-00 00:00:00' OR a.publish_up <='2011-04-24 02:01:17') AND (a.publish_down = '0000-00-00 00:00:00' OR a.publish_down >= '2011-04-24 02:01:17') ORDER BY c.path, a.ordering, a.created DESC LIMIT 0, 3



©2011 Mi sitio web

Figura 3.24: Diseño final de la página de error 500

Capítulo 4

Apéndice

4.1. Evitar que Mootools cargue de forma predeterminada

Existen ocasiones en que no se desea cargar a Mootools en la plantilla, ya sea porque se va a utilizar otra biblioteca JavaScript o porque se necesita agilizar la velocidad de la página reduciendo las peticiones a archivos que no se utilizan.

Para quitar cualquier archivo .js se debe realizar lo siguiente:

En `index.php`, antes de la directiva `<jdoc:include type="head" />` insertar:

```
<?php  
unset($this->_scripts[$this->baseurl."/media/system/js/mootools-core.js"]);  
unset($this->_scripts[$this->baseurl."/media/system/js/core.js"]);  
unset($this->_scripts[$this->baseurl."/media/system/js/caption.js"]);  
unset($this->_scripts[$this->baseurl."/media/system/js/mootools-more.js"]);  
?>
```

En este caso, se estará eliminando de la plantilla los archivos `mootools-core.js`, `core.js`, `caption.js`, `mootools-more.js`. Algo a tener en cuenta es que Joomla (y muchas extensiones) suelen hacer uso de Mootools, por lo que realizar esta práctica puede traer problemas posteriores.

4.2. Estilos determinados para diferentes navegadores

Un problema común al diseñar una página es tratar de hacer que el diseño sea compatible con la mayoría de los navegadores. A partir de Joomla! 1.6 el CMS incorpora una nueva característica que es la incorporación de estilos específicos dependiendo del tipo de navegador web.

Suponiendo que se desea mostrar estilos diferentes para *Internet Explorer* y *Mozilla Firefox*, lo primero a realizar será crear dos archivos: `template_msie.css` y `template_mozilla.css`. El primero será el archivo que cargará *Internet Explorer*, mientras que el otro será el que cargue *Mozilla Firefox*.

Luego, en el archivo `index.php` de la plantilla se incorpora:

```
<?php  
$archivos = JHtml::_('stylesheet','templates/interpoint/css/template.css',  
    null, false, true);  
if ($archivos):
```

```

        if (!is_array($archivos)):
            $archivos = array($archivos);
        endif;
        foreach($archivos as $archivo):
    ?>
        <link rel="stylesheet" href="<?php echo $archivo;?>" type="text/
            css" />
<?php
        endforeach;
    endif;
?>

```

Nota

Antes de cargar el archivo correspondiente, Joomla comprueba que éste exista en el directorio web.

`JHtml` es la función que se encargará de detectar el tipo de navegador y cargará los archivos necesarios. Algo interesante es que además es posible cargar un archivo para una determinada versión de un navegador. Por ejemplo, si se desea establecer un estilo para Internet Explorer 8, el nombre del archivo debe ser `template_msie_8.css`.

Nota

Es posible encontrar más información sobre las diferentes variantes para cargar archivos específicos para un navegador en los comentarios de la función `stylesheet()` en el archivo `/libraries/joomla/html/html.php`. También podemos cargar archivos `.js` para distintos navegadores, pero en lugar de poner `stylesheet` como parámetro de `JHtml` debemos cambiarlo por `script`.

4.3. Detectar la página principal del sitio

Puede existir una situación en que se necesita detectar cuando el usuario se encuentra en la página principal del sitio. A través del siguiente código es posible realizarlo:

```

<?php
$app      = JFactory::getApplication();
$menu     = $app->getMenu();
if (($menu->getActive() == $menu->getDefault()))
{
    //El usuario se encuentra en la pagina principal del sitio
}
?>

```

Nota

Más información sobre `getMenu()`: <http://docs.joomla.org/JApplication::getMenu>

4.4. Detectar si un usuario está logueado en el sitio

Si se desea detectar si un usuario está (o no) logueado en el sitio, es posible realizarlo con este código:

```

<?php
$usuario =& JFactory::getUser();
if ($usuario->guest == 1)
{
    //El usuario no esta logueado en el sitio
}
?>

```

Nota

Más información sobre `getUser()`: http://docs.joomla.org/Accessing_the_current_user_object

4.5. Detectar un componente determinado

Puede ocurrir que en algunas situaciones necesite detectar un componente determinado. Esto es posible realizarlo obteniendo el valor del parámetro `option` que se establece en la URL de la página:

```

<?php
$componente = JRequest::getVar('option');
if ($componente == "com_content")
{
    //Se encuentra en el componente com_content
}
?>

```

De la misma forma es posible obtener otros parámetros como son `ItemId`, `view`, etc.

Nota

Más información sobre la obtención de variables por `POST` y `GET`: http://docs.joomla.org/Retrieving_data_from_GET_and_POST_requests

Derechos de autor

Material bajo licencia [Creative Commons Reconocimiento-NoComercial-CompartirIgual 2.5 España](#): Es libre de copiar, distribuir, transmitir y modificar el trabajo, siempre y cuando haga referencia la autoría original a [Leandro D'Onofrio](#), para [Comunidad Joomla](#) y [Biblioteca Comunidad Joomla](#). Si altera, transforma o crea una obra derivada, deberá distribuir el resultado bajo una licencia igual, similar o compatible. Cualquiera de las condiciones mencionadas pueden no aplicarse si obtiene permisos del autor. Para cualquier reutilización o distribución, deberá dejar en claro la licencia, la mejor manera para hacerlo es a través de un enlace hacia la licencia [Creative Commons Reconocimiento-NoComercial-CompartirIgual 2.5 España](#).

COMUNIDADJOOMLA.ORG is not affiliated with or endorsed by the Joomla! Project or Open Source Matters. The Joomla! name and logo are used under a limited license granted by Open Source Matters the trademark holder in the United States and other countries.

Copyright ©2008 Open Source Matters. Todos los derechos reservados. Joomla! es un software libre distribuido bajo la licencia GNU/GPL.