

# Desarrollando templates para Joomla! 1.5



# Joomla!™

...because open source matters

## Guía paso a paso

Por Comunidad Joomla!

[comunidadjoomla.org](http://comunidadjoomla.org)

# Índice

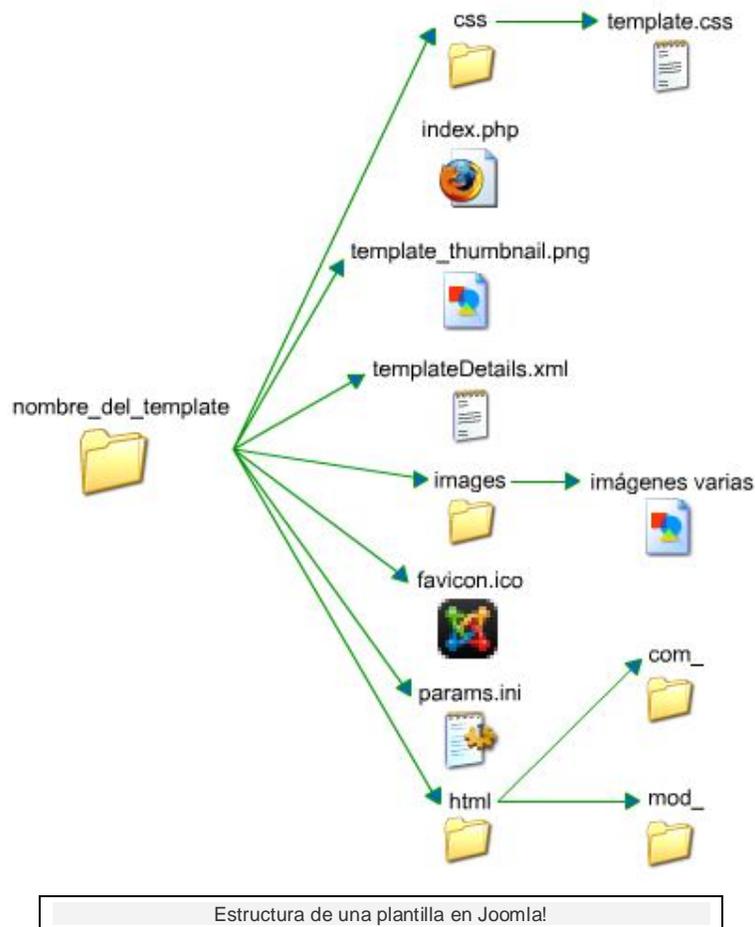
Por Comunidad Joomla!	1
Índice	2
Antes de empezar	3
Estructura básica de una plantilla para Joomla! 1.5.x	3
Carpetas	4
css	4
html	4
images	4
Archivos	4
index.php	4
templateDetails.xml	4
template_thumbnail.png	4
params.ini	4
template.css	4
favicon.ico	4
imágenes varias	5
Creación de la carpeta para nuestro template	5
Creación de templateDetails.xml	5
Datos	5
Archivos	6
Posiciones	6
Parámetros	6
Creación de index.php	8
<jdoc:include type="head" />	9
\$this->template	9
\$mainframe->getCfg('sitename')	9
<jdoc:include type="modules" name="posición" style="estilo" />	9
<jdoc:include type="component" />	10
Edición de template.css	12
Clases pertenecientes a contenido	15
componentheading	15
contentheading	15
buttonheading	15
small	15
createdate	16
readon	16
article_separator	16
Clases pertenecientes a módulos	16
moduletable	16
h3	17
active	17
item	17
parent	17
Otras clases CSS importantes	17
button	17
inputbox	17
pagenav	17
modifydate	17
sectiontableentry1 / sectiontableentry2	18
Manejo de parámetros en la plantilla	22
Creación de template_thumbnail.png	24
Apuntes finales	24
Recursos para el futuro	25

## Antes de empezar

- Necesariamente se debe tener:
  - Conocimiento básico de HTML, CSS y PHP;
  - Conocimiento del entorno de Joomla! y conceptos básicos (que es un modulo, que es un componente, etc.)
  - Un servidor local o remoto con Joomla! 1.5 instalado;
  - Un editor de programación (por ejemplo Notepad++ para Windows);

## Estructura básica de una plantilla para Joomla! 1.5.x

Básicamente un template puede estar formado por las siguientes carpetas y archivos:



# Carpetas

## **css**

Es la carpeta que contiene los estilos en cascada del template.

## **html**

Esta carpeta posee los archivos que funcionaran como plantilla especifica para módulos y componentes. Su uso no es prioritario pero es una funcionalidad a explotar si deseamos un código personalizado para componentes/módulos.

## **images**

Contendrá las imágenes usadas para el diseño.

# Archivos

## **index.php**

El archivo principal, contendrá todo el HTML y directivas PHP de Joomla! para cargar el contenido (artículos, componentes, módulos, etc.)

## **templateDetails.xml**

Determinante para que la plantilla sea reconocida en el back-end del CMS. Posee diversos datos como licencia, autor, e-mail, etc. En Joomla! 1.5 la plantilla podrá tener parámetros manejables desde el back-end para cambiar la visualización del template. En este archivo se agregaran los parámetros y las posiciones de disponibles para mostrar los módulos.

## **template\_thumbnail.png**

Es la imagen que actuará como previsualización del template en la administración.

## **params.ini**

Contendrá los valores para configurar los parámetros del template. Si la plantilla no posee parámetros para configurar, este archivo no será necesario.

## **template.css**

Es el estilo en cascada principal que se encuentra dentro de la carpeta 'css'. En la rama de Joomla! 1.0.x este archivo tenía el nombre de template\_css.css

## **favicon.ico**

Es el favicon que usaremos en nuestro sitio.

## imágenes varias

Conformaran el diseño de la plantilla.

Dentro de la carpeta 'html', como decía más arriba, estarán archivos y carpetas específicos para los módulos y componentes del sitio. Con esto podremos manejar a nuestro gusto la apariencia y el output de dichos componentes y módulos. La idea es que la plantilla prevalezca por encima del output HTML que genera Joomla! De esta forma, por ejemplo, podremos reemplazar todas las tablas que genera tal componente y reemplazarlas por DIVs.

## Creación de la carpeta para nuestro template

Lo primero que haremos será crear la carpeta en donde se alojaran los archivos de nuestra plantilla. Para eso vamos al directorio 'templates' de Joomla! y creamos una carpeta con el nombre que queramos, en minúscula, y en caso de ser varias palabras, sin espacios. Luego podemos crear las subcarpetas necesarias: css, images y/o HTML



Nombre de la carpeta que alojará nuestro template

## Creación de templateDetails.xml

Podemos dividir en cuatro partes el contenido de templateDetails.xml: Datos, Archivos, Posiciones y Parámetros.

## Datos

Aquí, entre cada etiqueta pondremos los datos específicos de la plantilla como su versión, autor, email, página web, año, licencia y descripción:

```
XML
<?xml version="1.0" encoding="utf-8"?>
<install version="1.5" type="template">
<name>Nombre del template</name>
<creationDate>Fecha de creación</creationDate>
<author>Nombre del autor</author>
<copyright>Licencia</copyright>
<authorEmail>E-mail del autor</authorEmail>
<authorUrl>Web del autor</authorUrl>
<version>Versión</version>
<description>
<![CDATA[
Aquí va la descripción, puede usarse HTML o texto plano.
]]>
</description>
```

## Archivos

Cuando hayamos terminado nuestro trabajo, aquí deberemos especificar todos los archivos que hemos utilizado para el diseño, para luego comprimir todo en ZIP y tener un instalador de la plantilla. Como ahora no necesitamos un instalador (ya que estamos creando el template dentro del mismo Joomla!) podemos dejar estos campos incompletos, pero es muy recomendable que al terminar el trabajo lo completemos debidamente:

```
XML
<files>
<filename>index.php</filename>
<filename>templateDetails.xml</filename>
<filename>template_thumbnail.png</filename>
<filename>favicon.ico</filename>
<filename>css/template.css</filename>
<filename>images/imagen.png</filename>
</files>
```

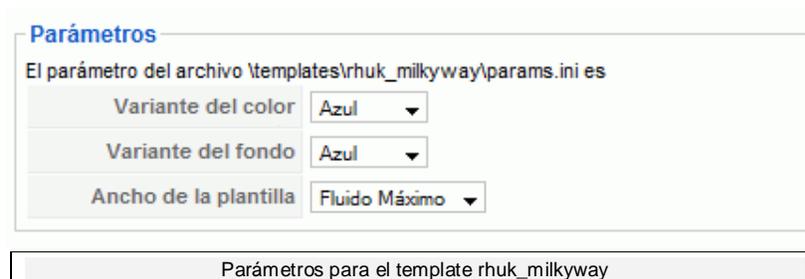
## Posiciones

Es importante saber con anticipación que posiciones usaremos y cuantas en nuestro sitio para planificar en donde mostrarlos en la plantilla.

```
XML
<positions>
<position>left</position>
<position>right</position>
<position>user3</position>
</positions>
```

## Parámetros

Si deseamos realizar una plantilla más versátil, podemos agregarle parámetros para ser manejados desde el Back-end. Por ejemplo, si nuestra plantilla cuenta con la posibilidad de configurar los colores de fondo, esta opción estará habilitada para ser activada desde la administración de Joomla!. En Joomla! 1.0.x, si añadíamos un parámetro, este solo podía ser modificado tocando el index.php de la plantilla. Podemos encontrar un ejemplo de parámetros de template accediendo a **Extensiones --> Gestor de plantillas --> rhuk\_milkyway**. En la columna derecha veremos los parámetros para dicho template:



## XML

```
<params>
<param name="color" type="list" default="blanco" label="Color de la
tipografia" description="Color de la tipografia">
<option value="azul">Azul</option>
<option value="rojo">Rojo</option>
<option value="verde">Verde</option>
</param>
</params>
</install>
```

Los valores por default y cada vez que modificamos un parámetro desde el back-end, son guardados en el archivo **params.ini**. Corre por nuestra cuenta si nosotros queremos o no añadir este tipo de posibilidad a nuestro diseño. Por lo general son las plantillas comerciales las que lo implementan, ya que le otorga mucha flexibilidad al producto. Más adelante veremos un ejemplo de utilización de parámetros.

Entonces templateDetails.xml puede tener la siguiente apariencia:

## XML

```
<?xml version="1.0" encoding="utf-8"?>
<install version="1.5" type="template">
<name>Nombre del template</name>
<creationDate>Fecha de creacion</creationDate>
<author>Nombre del autor</author>
<copyright>Licencia</copyright>
<authorEmail>E-mail del autor</authorEmail>
<authorUrl>Web del autor</authorUrl>
<version>Version</version>
<description>
<![CDATA[
Aqui va la descripcion, puede usarse HTML o texto plano.
]]>
</description>
<files>
<filename>index.php</filename>
<filename>templateDetails.xml</filename>
<filename>template_thumbnail.png</filename>
<filename>favicon.ico</filename>
<filename>css/template.css</filename>
<filename>images/imagen.png</filename>
</files>
<positions>
<position>left</position>
<position>right</position>
<position>user3</position>
</positions>
<params>
<param name="color" type="list" default="azul" label="Color de la tipografia"
description="Color de la tipografia">
<option value="azul">Azul</option>
<option value="rojo">Rojo</option>
<option value="verde">Verde</option>
</param>
</params>
</install>
```

## Creación de index.php

Para la creación del index.php, lo primero que haremos será insertar el HTML que formará la maquetación de la plantilla. Para eso partiremos desde un layout básico de tres columnas, barra de navegación, header y footer. Entonces en index.php creamos el HTML:

```
PHP
<!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">
<head>
</head>

<body>

<!-- Contenedor -->
<div id="contenedor">

<!-- Comienzo header -->
<div id="header">

</div>
<!-- Fin header -->

<!-- Comienzo navegacion-->
<div id="navigacion">

</div>
<!-- Fin navegacion -->

<!-- Comienzo columna izquierda -->
<div id="izquierda">

</div>
<!-- Fin columna izquierda -->

<!-- Comienzo contenido -->
<div id="contenido">

</div>
<!-- Fin contenido -->

<!-- Comienzo columna derecha -->
<div id="derecha">

</div>
<!-- Fin columna derecha -->

<!-- Comienzo footer -->
<div id="footer">

</div>
<!-- Fin footer -->

</div>
<!-- Fin contenedor -->

</body>
</html>
```

Lo siguiente será ir incorporando las directivas que cargaran toda la información para que la plantilla funcione. Veamos las básicas:

## **<jdoc:include type="head" />**

Esta directiva irá dentro de la cabecera HTML. Cargará el título de la página en cuestión, metatags, feed, y framework javascript. Joomla! a partir de la versión 1.5 decide usar como framework javascript a Mootools. Y esto lo hace de forma nativa, es decir que no nos dá la posibilidad de elegir entre cargarlo o no (a menos que editemos algún archivo fuente). Obviamente no hay que tomarlo para mal, sino al revés, entre líneas el equipo de Joomla! nos esta diciendo que echemos rienda suelta a nuestra imaginación y utilicemos a Mootools para añadir efectos, nuevas propiedades o lo que queramos a la plantilla. En otras palabras, que explotemos todo su potencial. Otro hecho es que para mostrar nuestro favicon, tendremos que añadir las líneas manualmente para mostrarlo.

## **\$this->template**

Obtendremos el nombre de la carpeta contenedora de nuestra plantilla. Se puede utilizar para cargar el estilo en cascada, favicon y cualquier archivo que queramos y se encuentre dentro de esta carpeta.

Con esto ya podemos crear la cabecera HTML de la plantilla:

```
PHP
<head>
<jdoc:include type="head" />
<link rel="stylesheet" href="templates/<?php echo $this->template
?>/css/template.css" type="text/css" />
</head>
```

Por otro lado, para el cuerpo utilizaremos:

## **\$mainframe->getCfg('sitename')**

Obtendremos el nombre del sitio en cuestión.

## **<jdoc:include type="modules" name="posición" style="estilo" />**

Con esta directiva estamos indicando la carga de un módulo en particular y de una manera específica. Esto se establecerá a partir de los parámetros 'name' y 'style'. En el caso del primero, escribiremos la posición de módulo que queremos cargar. Por ejemplo, si deseamos todos los

módulos derechos, el parámetro será 'name="right"'. El segundo parámetro posee algunas variantes, el uso de cada una va a depender de nuestras necesidades. Las variedades de estilo pueden ser **'table', 'horz', 'xhtml', 'rounded' y 'raw'**. Tanto con 'table' como con 'horz' el contenido del módulo será cargado a través de tablas, la diferencia es que el primer parámetro lo hará de forma vertical (a través de una columna) y el segundo lo hará de forma horizontal (a través de una fila). Si utilizamos 'xhtml' el contenido en vez de ser cargado en tablas será insertado en DIVs. El parámetro 'rounded' es similar a 'xhtml' con la diferencia que carga el contenido en varios DIVs anidados. Este último se suele utilizar para realizar bordes redondeados a través de imágenes. Finalmente el parámetro 'raw' cargará el contenido neto, sin contenedores ni título.

**Ejemplo:** Queremos insertar en DIVs todos los módulos izquierdos en determinada parte del sitio, la directiva quedará conformada por:

#### PHP

```
<jdoc:include type="modules" name="left" style="xhtml" />
```

**Ejemplo 2:** Queremos cargar los módulos que se encuentran en la posición 'top' de manera horizontal:

#### PHP

```
<jdoc:include type="modules" name="top" style="horz" />
```

En caso de no especificar el estilo, por default el valor será 'table'.

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

Con ella cargaremos el contenido principal del sitio (artículos, secciones, categorías, etc.). A diferencia de la inclusión de módulos, esta directiva no posee parámetros para manejar en donde cargar el contenido.

Aprendiendo esto ya estaremos resolviendo el 80% de index.php, quedando:

## PHP

```
<!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">
<head>
<jdoc:include type="head" />
<link rel="stylesheet" href="templates/<?php echo $this->template
?>/css/template.css" type="text/css" />
</head>

<body>

<!-- Contenedor -->
<div id="contenedor">

<!-- Comienzo header -->
<div id="header">
<h1><a href="index.php"><?php echo $mainframe->getCfg('sitename');?></a></h1>
</div>
<!-- Fin header -->

<!-- Comienzo navegacion-->
<div id="navigacion">
<jdoc:include type="modules" name="user3" style="xhtml" />
</div>
<!-- Fin navegacion -->

<!-- Comienzo columna izquierda -->
<div id="izquierda">
<jdoc:include type="modules" name="left" style="xhtml" />
</div>
<!-- Fin columna izquierda -->

<!-- Comienzo contenido -->
<div id="contenido">
<jdoc:include type="component" />
</div>
<!-- Fin contenido -->

<!-- Comienzo columna derecha -->
<div id="derecha">
<jdoc:include type="modules" name="right" style="xhtml" />
</div>
<!-- Fin columna derecha -->

<!-- Comienzo footer -->
<div id="footer">
<!--Cargo el footer-->
<?php include_once('includes/footer.php'); ?>
</div>
<!-- Fin footer -->

</div>
<!-- Fin contenedor -->

</body>
</html>
```

En el DIV 'header' se cargó una etiqueta H1 que muestra el nombre del sitio. Por otro lado, en el footer se carga el archivo footer.php que trae un texto predeterminado para mostrar. Este se encuentra en includes/footer.php

Si en el backend, configuramos para que nuestro template sea el predeterminado, al ingresar al sitio veremos:



Lo cual no está mal, ya que aún no hemos editado el CSS, por el contrario está muy bien, porque hemos podido cargar el contenido del sitio en nuestra plantilla.

## Edición de template.css

Vamos a la carpeta 'css' y creamos el archivo template.css. Lo primero que haremos será insertar los estilos para la maquetación de la plantilla:

## CSS

```
* {
padding: 0;
margin: 0;
}

body {
font-size: 62.5%;
font-family: 'Lucida Grande', Verdana, Arial, Sans-Serif;
color: #333;
background-color:white;
}

#contenedor {
margin: 0 auto;
width: 922px;
}

#header {
width: 900px;
float: left;
padding: 10px;
margin: 10px 0px 5px 0px;
}

#navegacion {
float: left;
width: 900px;
padding: 10px;
margin: 0px 0px 5px 0px;
}

#izquierda {
margin: 0px 5px 5px 0px;
padding: 10px;
width: 195px;
float: left;
}

#contenido{
float: left;
margin: 0px 5px 5px 0px;
padding: 10px;
width: 456px;
display: inline;
}

#derecha {
color: #333;
margin: 0px 0px 5px 0px;
padding: 10px;
width: 195px;
float: left;
}

#footer {
width: 900px;
clear: both;
margin: 0px 0px 10px 0px;
padding: 10px;
}
```

## Joomla! 1.5

- [About Joomla!](#)
- [Features](#)
- [News](#)
- [The Community](#)

### Main Menu

- [Home](#)
- [Joomla! Overview](#)
- [Joomla! License](#)
- [More about Joomla!](#)
- [FAQ](#)
- [The News](#)
- [Web Links](#)
- [News Feeds](#)

### Resources

- [Joomla! Home](#)
- [Joomla! Forums](#)
- [Joomla! Help](#)
- [OSM Home](#)
- [Administration](#)

### Key Concepts

- [Extensions](#)
- [Content Layouts](#)
- [Example Pages](#)

### Login Form

Username

Password

Remember Me

- [Forgot your password?](#)
- [Forgot your username?](#)
- [Create an account](#)

Welcome to the Frontpage

Welcome to Joomla!

Written by Administrator

Thursday, 12 October 2006 10:00

**Joomla! is a free open source framework and content publishing system designed for quickly creating highly interactive multi-language Web sites, online communities, media portals, blogs and eCommerce applications.**



Joomla! provides an easy-to-use graphical user interface that simplifies the management and publishing of large volumes of content including HTML, documents, and rich media. Joomla! is used by organisations of all sizes for Public Web sites, Intranets, and Extranets and is supported by a community of thousands of users.

Last Updated ( Thursday, 08 November 2007 12:06 )

[Read more...](#)

Joomla Community Portal

Online

Written by Administrator

Wednesday, 07 July 2004 09:54

The new [Joomla Community Portal](#) is now online. The Joomla

Community Portal is a new web site

for the Joomla community. There you

will find a constant source of

information about the activities

powering the Joomla Project, learn

about [Joomla Events](#) worldwide, and

see if there is a [Joomla User](#)

[Group](#) nearby. A new,

monthly [Joomla Community](#)

[Magazine](#) promises an interesting

overview of project activities,

community accomplishments, and

learning material.

Last Updated ( Sunday, 31 August

2008 04:21 )

[Read more...](#)

The New Joomla Security

Strike Team

Written by Administrator

Wednesday, 07 July 2004 09:54

The Joomla Project has assembled a

top-notch team of experts to form

the new Joomla Security Strike

Team. This new team will solely

focus on investigating and resolving

security issues. Instead of working in

relative secrecy, the JSST will have a

strong public-facing presence at the

new [Joomla Security Center](#).

Last Updated ( Sunday, 31 August

2008 04:07 )

[Read more...](#)

What is the FTP layer

for?

Written by Administrator

Friday, 06 October 2006 21:27

The FTP Layer allows file operations

(such as installing Extensions or

updating the main configuration file)

without having to make all the folders

and files writable. This has been an

issue on Linux and other Unix based

platforms in respect of file

permissions. This makes the site

admin's life a lot easier and increases

security of the site.

You can check the write status of

relevant folders by going to

"Help->System Info" and then in the

sub-menu to "Directory

Permissions". With the FTP Layer

enabled even if all directories are

red, Joomla! will operate smoothly.

NOTE: the FTP layer is not required

on a Windows host/server.

Last Updated ( Thursday, 03 May

2007 09:19 )

Only one edit window!

How do I create "Read

more...?"

Written by Administrator

Friday, 06 October 2006 19:29

This is now implemented by inserting

a [Read more...](#) tag (the button is

located below the editor area) a

dotted line appears in the edited text

showing the split location for the

[Read more...](#). A new Plugin takes

care of the rest.

It is worth mentioning that this does

not have a negative effect on

migrated data from older sites. The

new implementation is fully

backward compatible.

Last Updated ( Thursday, 08

November 2007 11:03 )

### Polls

Joomla! is used for?

- Community Sites
- Public Brand Sites
- eCommerce
- Blogs
- Intranets
- Photo and Media Sites
- All of the Above!

### Who's Online

We have 1 guest online

### Advertisement

Featured Links:

[Joomla!](#)

Joomla! The most popular and widely used Open Source CMS Project in the world.

[JoomlaCode](#)

JoomlaCode, development and distribution made easy.

[Joomla! Extensions](#)

Joomla! Components, Modules,

Plugin and Languages by the

bucket load.

[Joomla! Shop](#)

For all your Joomla!

merchandise.

[Ads by Joomla!](#)

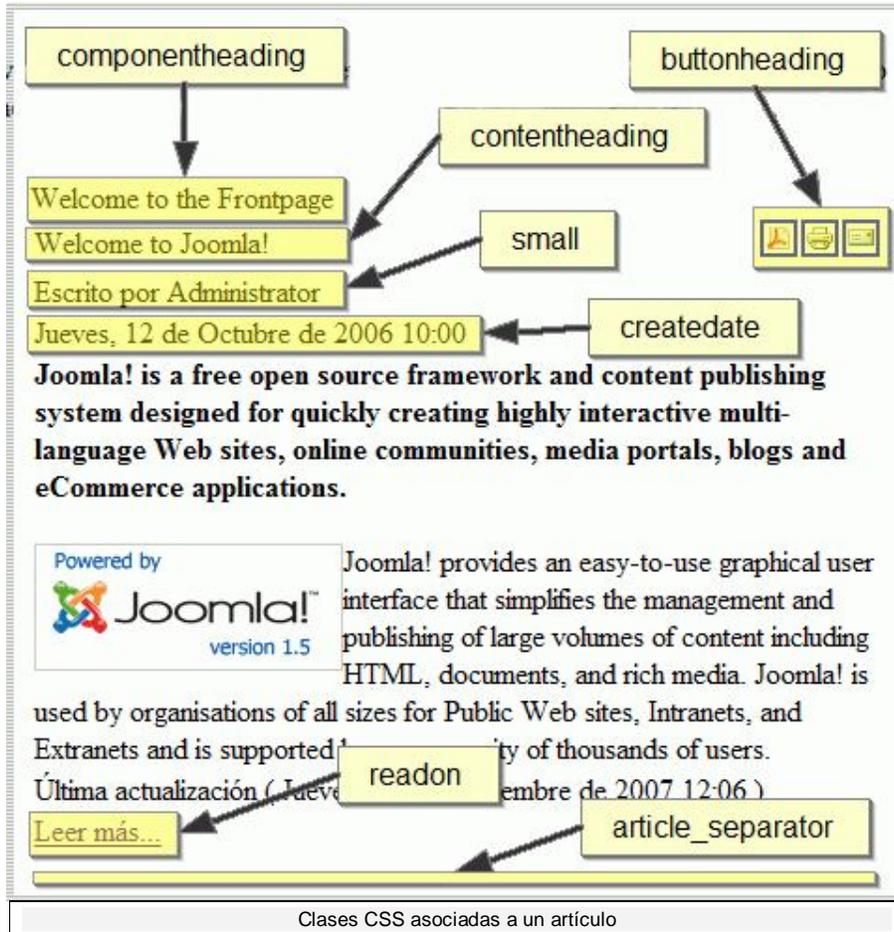
© 2008 Joomla! 1.5

Joomla! is Free Software released under the GNU General Public License.

Apariencia de nuestro template con estilos de maquetación

Ahora somos libres de añadir los estilos que queramos, pero para eso primero debemos conocer algunas de las clases que Joomla! carga por default en sus elementos.

## Clases pertenecientes a contenido



### componentheading

Nuestra el título del componente en cuestión. La página de inicio, al ser un componente (com\_frontpage) tiene posibilidad de mostrar un título.

### contentheading

Es la clase perteneciente a la celda que muestra el título de los artículos.

### buttonheading

Pertece a las celdas que muestran los iconos de PDF, Imprimir y Enviar a un amigo.

### small

Es usado por varios elementos, entre ellos el que muestra el autor de un artículo.

## createdate

Es usado por la celda que muestra la fecha de creación de un artículo.

## readon

Utilizada por el link "Leer más..."

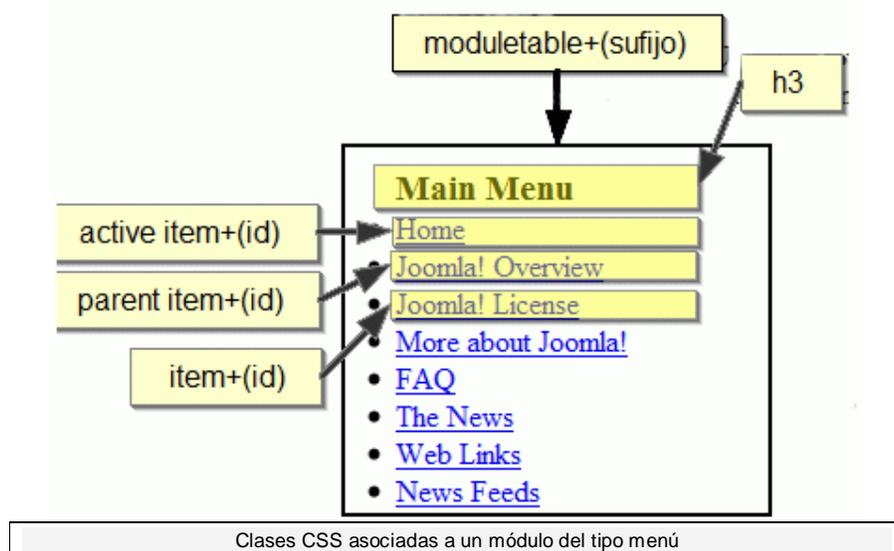
## article\_separator

Utilizada por la etiqueta <span> que separa a los artículos entre sí.

## Clases pertenecientes a módulos

Por lo general, las clases asociadas a módulos dependen de varias cosas, desde la forma en que nosotros seleccionamos el estilo de carga del módulo, hasta si esta posee un sufijo de la clase de módulo. Además muchos módulos creados por terceros poseen sus propias clases propias.

Tomemos como ejemplo, un módulo del tipo menú:



## moduletable

Arriba contábamos que los módulos pueden cargarse de diferentes formas, dentro de tablas ó en DIVs. Para nuestro template nosotros hemos especificado que los módulos se carguen en un solo DIV a través del parámetro `style="xhtml"`. Para este caso, el DIV contenedor posee la clase CSS asociada **moduletable**

Si nosotros entramos a la edición del módulo, podremos añadir un sufijo específico para **moduletable**:

Sufijo de la clase del módulo	<input type="text" value="_menu"/>
Parámetro para añadir sufijo CSS	

### h3

Los títulos de los módulos se cargan con la etiqueta H3.

### active

Para el caso del menú, cuando nos encontramos dentro de una página existente como item, se agrega la clase **active**.

### item

A su vez cada elemento del menú posee una clase unica, la cual es item más el ID que posea dicho elemento. Es importante esta clase cuando deseamos un diseño único por cada elemento del menú.

### parent

Si hemos creado un elemento que a su vez posee sub-items, al elemento padre se le añade la clase **parent**.

## Otras clases CSS importantes

### button

Es una clase asociada a elementos del tipo botón.

### inputbox

Suele estar asociada a elementos text-input.

### pagenav

Clase asociada a la paginación de artículos.

### modifydate

Clase asociada a la fecha de modificación de un artículo.

## sectiontableentry1 / sectiontableentry2

Suele estar asociada cuando se muestran datos en tablas.

Ahora que conocemos las clases mas importantes, podemos ir aplicando un poco de estilo al template y acomodar los módulos que vamos a usar:

```
CSS

* {
padding: 0;
margin: 0;
}

body {
font-size: 62.5%;
font-family: 'Lucida Grande', Verdana, Arial, Sans-Serif;
color: #333;
}

#contenedor {
margin: 0 auto;
width: 922px;
}

#header {
width: 900px;
float: left;
padding: 10px;
margin: 10px 0px 5px 0px;
}

#navegacion {
float: left;
width: 900px;
padding: 10px;
margin: 0px 0px 5px 0px;
}

#izquierda {
margin: 0px 5px 5px 0px;
padding: 10px;
width: 195px;
float: left;
}

#contenido {
float: left;
margin: 0px 5px 5px 0px;
padding: 10px;
width: 456px;
display: inline;
}

#derecha {
color: #333;
margin: 0px 0px 5px 0px;
padding: 10px;
width: 195px;
float: left;
}

#footer {
width: 900px;
clear: both;
margin: 0px 0px 10px 0px;
padding: 10px;
}

/* Color
----- */
```

```

a{
text-decoration:none;
color:#3B6EA5;
}

#header a {
font-size:1.8em;
margin-left:230px;
color:#3B6EA5;
}

/* Elementos
----- */

a:hover {
color: #147;
text-decoration: underline;
}

a img{
border:0px;
}

h1, h2, h3 {
font-family: 'Trebuchet MS', 'Lucida Grande', Verdana, Arial, Sans-Serif;
font-weight: bold;
}

h1 {
font-size: 2em;
}

h2 {
font-size: 1.6em;
}

h3 {
font-size: 1.3em;
}

h1, h1 a, h1 a:hover, h1 a:visited, .description {
text-decoration: none;
}

h2, h2 a, h2 a:visited, h3, h3 a, h3 a:visited {
color: #333;
}

h2, h2 a, h2 a:hover, h2 a:visited, h3, h3 a, h3 a:hover, h3 a:visited{
text-decoration: none;
}

/* Modulos
----- */

#izquierda .moduletable_menu{
text-align:right;
margin-bottom:15px;
}

#derecha .moduletable_menu, #derecha .moduletable{
text-align:left;
margin-bottom:15px;
}

#izquierda .moduletable_menu li{
margin:3px 0 0;
padding:0;
list-style-image:none;
list-style-type:none;
}

#izquierda .moduletable_menu li:before{
content: "\00BB \0020";
}

```

```

.moduletable_menu h3, .moduletable h3, .moduletable_text h3{
margin:5px 0 0;
padding:0;
text-transform:uppercase;
}

/* Contenido
----- */

.contentheading {
font-size:1.4em;
margin:0;
color:#333333;
font-weight:bold;
}

.contentheading:hover {
color:#3B6EA5;
}

.componentheading{
font-size:1.4em;
margin:0;
color:#777777;
font-weight:bold;
padding-left:10px;
}

.contentpaneopen{
line-height:1.4em;
font-size:1.2em;
text-align:left;
padding: 0px 10px 0px 10px;
}

.contentpaneopen img{
padding: 5px 10px;
}

.readon{
font-size:0.8em;
text-align:left;
}

.contenttoc, .pagenavcounter{
margin-bottom:10px;
}

.article_separator{
display:block;
background:#474747;
height:1px;
margin:10px 60px 10px 10px;
}

small, .createdate, .modifydate{
font-family:Arial,Helvetica,Sans-Serif;
font-size:0.9em;
line-height:1.5em;
color:#777777;
padding-bottom:10px;
}

.banneritem_text, .bannerfooter_text{
padding:5px;
}

/* Menu superior
----- */

#navegacion #mainlevel-nav{
list-style:none;
position:relative;
padding:0px;
float:left;
padding-left:100px;
}

```

```

#navegacion #mainlevel-nav li {
float:left;
height:20px;
}

#navegacion #mainlevel-nav li a {
float:left;
text-decoration:none;
margin:0px;
padding-right:10px;
}

#navegacion #mainlevel-nav li a:hover {
text-decoration:underline;
}

#navegacion #mainlevel-nav li a:before{
content: "\00BB \0020";
}

```



Template con diseño

El diseño está basado en **White as Milk** <http://azeemazeez.com/misc/whiteasmilk/> , template para Wordpress, creado por **Azeem Azeez** <http://www.azeemazeez.com/blogs/white-as-milk/>

## Manejo de parámetros en la plantilla

Como vimos al principio, los parámetros de un template se deben especificar en **templateDetails.xml**. Allí se debe poner el tipo de parámetro, las opciones que posee y cual es el parámetro por default. A su vez, los valores de los parámetros son guardados en el archivo **params.ini** que como vimos, se encuentra en la carpeta de la plantilla.

Veamos por ejemplo, la utilización de parámetros para cambiar el color de la tipografía del template. En **templateDetails.xml** nosotros ya hemos especificado la opción. Lo siguiente será crear un archivo **params.ini** con el valor por default que debe tener el parámetro, en este caso queremos que el color predeterminado de la tipografía sea azul:

```
color=azul
```

En caso de tener más parámetros, estos deben ir uno por línea con su correspondiente valor:

```
nombre_del_parametro_1=valor
nombre_del_parametro_2=valor
nombre_del_parametro_3=valor
```

Lo siguiente será ir al **index.php** y agregar la directiva que llamará al valor del parámetro. Este valor lo utilizaremos para crear un identificador CSS y así especificar cuando cargar un color y cuando otro. Entonces en el **index.php**, en la etiqueta **<body>** agregamos:

```
PHP
<body id="php echo $this-&gt;params-&gt;get('color'); ?">
```

De esta forma, dependiendo del valor que tenga el parámetro podremos tener:

Si seleccionamos como valor azul:

```
HTML
<body id="azul">
```

Si seleccionamos como valor verde:

```
HTML
<body id="verde">
```

Si seleccionamos como valor rojo:

#### HTML

```
<body id="rojo">
```

Nos quedaría modificar el CSS del template para que dependiendo del identificador, se cambien los colores. En el apartado Color podemos poner:

#### CSS

```
/* Color
----- */

#header a {
font-size:1.8em;
margin-left:230px;
}

a{
text-decoration:none;
}

#azul a{
color:#3B6EA5;
}

#azul #header a {
color:#3B6EA5;
}

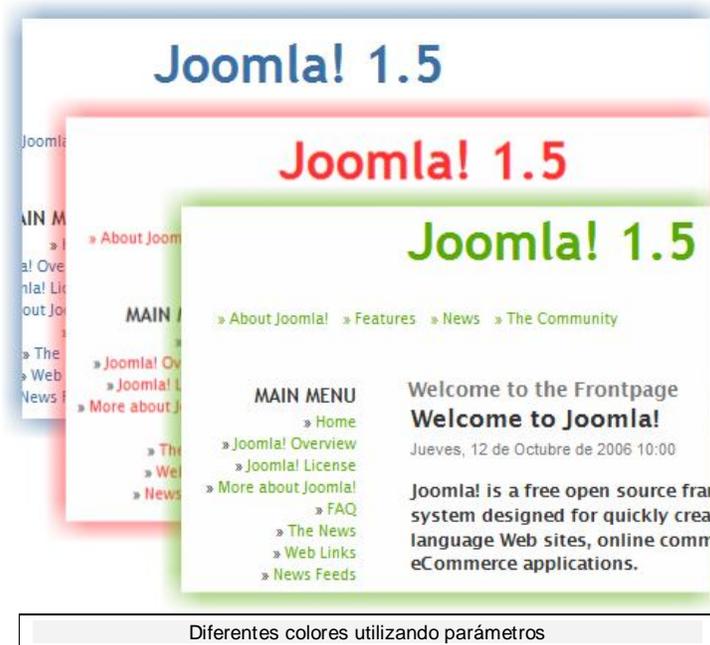
#verde a{
color:#59AC03;
}

#verde #header a {
color:#59AC03;
}

#rojo a{
color:#FF3131;
}

#rojo #header a {
color:#FF3131;
}
```

Entrando a la configuración del template podremos cambiar el color de la tipografía.



## Creación de template\_thumbnail.png

Lo último que nos quedaría para terminar nuestro template básico, es la creación de `template_thumbnail.png`, la imagen que actuará a modo de previsualización del template en el back-end del sitio. No está establecido el tamaño estándar de la imagen, pero por lo general es de 200px de ancho y 150px de alto.

## Apuntes finales

- Si deseamos crear un instalador de nuestro template, recordemos de especificar en **templateDetails.xml** todos los archivos que se utilizaron. Esto es importante ya que si no lo hacemos, al momento de subir el instalador Joomla! nos entregará un mensaje de error.
- Una forma fácil para descubrir las diferentes clases asociadas que genera el output de Joomla! es utilizando **Firebug** <http://getfirebug.com>, una extensión para Firefox que permite inspeccionar el DOM de un sitio y visualizar como está conformado.
- En la web podemos encontrar varios recursos para continuar con el aprendizaje de la creación de templates para Joomla!, el más importante se encuentra en la documentación oficial del proyecto: **Template reference material** [http://docs.joomla.org/Template\\_reference\\_material](http://docs.joomla.org/Template_reference_material)
- No nos olvidemos de Mootools! En su página oficial <http://mootools.net> se puede encontrar documentación y ejemplos para añadir efectos a la plantilla.
- Por seguridad, es conveniente crear un archivo `index.html` sin contenido y alojarlo en la carpeta de nuestro template y subcarpetas.

## Recursos para el futuro

Tanto si estás interesado en desarrollar sitios para terceros, como si lo estás en desarrollar sitios para tí, tras este manual paso a paso, te queda todo un mundo abierto de posibilidades y de recursos para seguir profundizando y adquiriendo mayor experiencia con Joomla!

Si te decides a trabajar más con Joomla!, estos enlaces generales pueden ayudarte en tus venideros desarrollos:

- <http://forum.joomla.org> – Regístrate, busca y pregunta. ¡Siempre ayuda a otros si te es posible!
- <http://help.joomla.org> – Encuentra más y más documentación.
- <http://extensions.joomla.org> – Encuentra un montón de extensiones!.
- <http://www.joomla.org> – Estate al día y !encuentra el modo de devolver tu gratitud a la comunidad en la medida de tus posibilidades!
- <http://www.comunidadjoomla.org> – ¡Visítanos y participa con nosotros si te apetece!, aparte de adaptar guías y otro material de ayuda para Joomla! del inglés al español, dedicamos nuestras energías a:
  - Mantener y actualizar los packs de Joomla! en nuestro idioma.
  - Generar material de ayuda en nuestro idioma para Joomla!
  - Traducir noticias Joomla! del inglés al español
  - Traducir extensiones para Joomla! que sean GNU/GPL
  - Dar soporte para Joomla!, en nuestro idioma, desde nuestro Foro <http://foro.comunidadjoomla.org/>
  - Tenemos una wiki para documentación
  - ¡Y un blog para charlar sobre temas de interés!

Todo esto y algunas cosas más, con el objetivo principal de fomentar y promover la colaboración entre usuarios, sin ánimo de lucro; brindando el espacio y un entorno apropiado que lo permitan, a través de nuestro proyecto Comunidad Joomla!.