



ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO

FACULTAD DE INFORMÁTICA Y ELECTRÓNICA

ESCUELA DE INGENIERÍA EN SISTEMAS

**“ESTUDIO COMPARATIVO DEL DESARROLLO DE COMPONENTES EN
SISTEMAS DE ADMINISTRACIÓN DE CONTENIDOS PARA LA
IMPLEMENTACIÓN DEL PORTAL WEB DE LA LIBRERÍA MAJOS”**

TESIS DE GRADO

**Previa la obtención del título de
INGENIERO EN SISTEMAS INFORMÁTICOS**

Presentado por:

**CARLOS ARTURO JARA SANTILLÁN
VÍCTOR MANUEL OQUENDO CORONADO**

RIOBAMBA – ECUADOR

2010

El desarrollo del presente proyecto lleva la inmensa gratitud a la Escuela Superior Politécnica de Chimborazo, en especial a la Escuela de Ingeniería en Sistemas, por abrirnos las puertas hacia el conocimiento científico y facilitar todo el equipo tecnológico necesario para la realización de nuestra tesis.

A nuestros Maestros Ing. Ms.C. Danilo Pástor, Dr. Julio Santillán, quienes con humildad, sinceridad y responsabilidad, supieron guiarnos e impartir sus valiosos conocimientos.

Agradezco a DIOS y a mi Madre DOLOROSA por darme la fortaleza para enfrentar todo obstáculo ante mí presentado, la sabiduría para tomar la mejor decisión y el coraje para no desmayar jamás.

Al milagro más grande que me ha dado la vida mi abuelita ELIZABETH por transmitirme ese don de gente y la manera positiva en que toma la adversidad por más duras que esta sea y por ser la única persona que desde siempre en verdad a estado a mi lado.

A MERY y JOSÉ mis amados hermanos, por su apoyo incondicional sin importar la distancia ni el tiempo y por su ejemplo de superación y dedicación.

A RAFAELITA mi querida sobrina por llenar mi corazón de alegría con su dulzura y ocurrencias.

A MIS AMIGOS gracias por formar parte de mi vida, por estar junto a mí cuando más los he necesitado, por enseñarme que un verdadero amigo puede transformarse en un hermano.

CARLOS

Mi tesis la dedico con todo cariño:

A Dios que me dio la oportunidad de vivir y de regalarme una familia maravillosa.

Con mucho amor a mis padres que me dieron la vida y han estado en todo momento. Gracias por todo papá y mamá por darme una carrera para mi futuro y por creer en mí, aunque hemos pasado momentos difíciles siempre han estado apoyándome y brindándome todo su amor, por todo esto les agradezco de todo corazón el que estén conmigo y a mi lado.

A mi hermano gracias por estar conmigo y apoyarme siempre, te quiero mucho.

A todos mis amigos, muchas gracias por estar conmigo en todo este tiempo donde he vivido momentos felices y tristes, gracias por ayudarme a sobrellevar todo lo ante mi presentado.

VÍCTOR

NOMBRE

FIRMA

FECHA

Ing. Iván Menes

**DECANO DE LA FACULTAD
DE INFORMÁTICA Y
ELECTRÓNICA**

.....

.....

Ing. Raúl Rosero

**DIRECTOR DE LA
ESCUELA DE INGENIERÍA
EN SISTEMAS.**

.....

.....

Ing. Ms.C. Danilo Pástor

DIRECTOR DE TESIS

.....

.....

Dr. Julio Santillán

MIEMBRO DEL TRIBUNAL

.....

.....

Lcdo. Carlos Rodríguez

**DIRECTOR DPTO.
DOCUMENTACIÓN**

.....

.....

NOTA DE LA TESIS

.....

“Nosotros, **Carlos Arturo Jara Santillán y Víctor Manuel Oquendo Coronado** somos responsables de las ideas, doctrinas y resultados expuestos en esta tesis; y, el patrimonio intelectual de la Tesis de Grado pertenece a la ESCUELA SUPERIOR POLITECNICA DE CHIMBORAZO”

.....

Carlos Arturo Jara Santillán

.....

Víctor Manuel Oquendo Coronado

ÍNDICE DE ABREVIATURAS

ALT:	Alternative Text.
API:	Application Programming Interface
BBS:	Bulletin Board System
CMS:	content management system.
FTP:	file transfer protocol.
GNU/GPL:	General Public License
HTML:	hypertext markup language.
HTTP:	hypertext transport protocol.
ISP:	Internet Service Provider
LAMP:	Linux – apache - mysql–php.
LDAP:	Lightweight Directory Access Protocol
PDA:	personal digital assistant.
PDF:	portable document format.
PHP:	Hypertext Pre-processor (inicialmente PHP Tools, o, Personal Home Page Tools).
R&D:	Research and Development
SEO:	Search Engine Optimization.
SQL:	Structure Query Language
URL:	uniform resources locator.
XML:	Xtensible Markup Language.
WWW:	World Wide Web.
WAMP:	Windows – apache - mysql–php.
WYSIWYG:	What You See Is What You Get

ÍNDICE GENERAL

CAPÍTULO I

GENERALIDADES

1.1.	Justificación	28
1.1.1.	Teórica	28
1.1.2.	Práctica	30
1.2.	Antecedentes	31
1.3.	Objetivos	32
1.3.1.	General	32
1.3.2.	Específicos	33
1.4.	Hipótesis	33

CAPÍTULO II

FUNDAMENTO TEÓRICO

2.1.	La Web	34
2.1.1.	Páginas Web	35
2.1.2.	Estándares Web.....	36
2.1.3.	Tecnologías Web.....	37
2.1.3.1.	Navegadores Web	38
2.1.3.2.	Servidores Web	38
2.1.3.3.	Otras tecnologías	39
2.2.	¿Qué es un CMS?.....	39
2.2.1.	Introducción.....	39
2.2.2.	Precisión del Concepto de CMS	41
2.2.2.1.	El ciclo de publicación de contenidos	42
2.2.2.2.	Definición de CMS	43
2.2.2.3.	Flujo de trabajo en un CMS	44
2.2.2.4.	Operación de un CMS	45

2.2.3.	Estructura de un CMS.....	47
2.2.4.	Características que un CMS debe cumplir para ser "SEO friendly".....	48
2.3.	Clasificación de los CMS.....	51
2.3.1.	Gestores de contenido con código abierto	51
2.3.1.1.	Plataformas de gestión de contenidos.....	51
2.3.1.2.	CMS para Foros	54
2.3.1.3.	CMS para Blogs	55
2.3.1.4.	CMS para galerías.....	56
2.3.1.5.	CMS para Galerías de Arte	56
2.3.1.6.	CMS para Wikis	56
2.3.1.7.	CMS para ECommerce	56
2.3.1.8.	CMS para groupware.....	57
2.3.2.	Gestores de contenidos en código propietario	57
2.3.3.	Gestores con algún grado de estandarización	60
2.4.	PHP el lenguaje de programación en CMS 's	60
2.4.1.	Argumento.....	60
2.4.2.	Historia	61
2.4.2.1.	PHP 3	61
2.4.2.2.	PHP 4	62
2.4.2.3.	PHP 5	62
2.4.2.4.	PHP 6	63
2.4.3.	Páginas CMS PHP.....	63
2.4.4.	Características del lenguaje PHP para el desarrollo de aplicaciones web	65
2.5.	Extensiones de los CMS 's.....	68
2.5.1.	Descripción	68
2.5.2.	Tipos de Extensiones.....	69
2.5.2.1.	Componentes	69
2.5.2.2.	Módulos.....	70
2.5.2.3.	Plugins	70
2.5.2.4.	Idiomas.....	71

2.5.2.5.	Plantillas	71
2.5.3.	Patrón MVC	71
2.5.3.1.	Descripción	71
2.5.3.2.	Elementos.....	72
2.5.3.2.1.	Modelo	73
2.5.3.2.2.	Controlador.....	73
2.5.3.2.3.	Vista	74
2.5.3.3.	Diagrama de Secuencia	74

CAPÍTULO III

ESTUDIO COMPARATIVO DE CMS ´S EN EL DESARROLLO DE COMPONENTES

2.1	Delimitación	76
3.2.	Joomla!	82
3.2.1.	Argumento.....	82
3.2.2.	Características.....	83
3.2.3.	Versiones.....	86
3.2.4.	Historia del proyecto	87
3.2.5.	Estructura Organizativa/Asociativa	88
3.2.6.	Estado actual	90
3.2.7.	Plataformas compatibles	90
3.2.8.	Desarrollo de componentes	91
3.2.8.1.	Prefacio	91
3.2.8.2.	Elaboración	93
3.3.	Drupal	94
3.3.1.	Argumento.....	94
3.3.2.	Características.....	95
3.3.3.	Versiones.....	99
3.3.4.	Historia del Proyecto	99
3.3.5.	Estructura organizativa/asociativa.....	100
3.3.6.	Estado Actual	100

3.3.7.	Plataformas compatibles	101
3.3.8.	Desarrollo de componentes (Módulos)	101
3.4.	Typo 3.....	104
3.4.1.	Argumento.....	104
3.4.2.	Características.....	105
3.4.3.	Versiones.....	107
3.4.4.	Historia del Proyecto	107
3.4.5.	Estructura Organizativa/Asociativa	108
3.4.6.	Estado Actual	110
3.4.7.	Tecnología	111
3.4.8.	Seguridad	112
3.4.9.	Soporte	113
3.4.10.	Desarrollo de componentes	113
3.5.	Parámetros de Medición.....	115
3.5.1.	Introducción.....	115
3.5.2.	¿Cómo evaluar un componente en un CMS?	116
3.5.3.	Parámetros de evaluación	117
3.5.3.1.	Requerimientos del sistema	118
3.5.3.1.1.	Soporte	119
3.5.3.1.2.	Seguridad	119
3.5.3.1.3.	Plataforma	119
3.5.3.2.	Facilidad de uso.....	120
3.5.3.2.1.	Desarrollo de interfaz de usuario	120
3.5.3.2.2.	Acceso de datos.....	120
3.5.3.2.3.	Flexibilidad.....	121
3.5.3.3.	Gestión de componentes dentro del CMS	121
3.5.3.3.1.	Interfaz para la gestión	121
3.5.3.3.2.	Instalación / desinstalación de extensiones	121
3.5.3.3.3.	Soporte multilinguaje	122
3.5.3.4.	Estructura de las extensiones	122
3.5.3.4.1.	Uso del esquema MVC (modelo, vista, controlador)	122

3.5.3.4.2.	Distribución de los archivos de la extensión.....	123
3.5.3.4.3.	Localización específica de una parte de la extensión	123
3.6.	Descripción de los módulos de prueba	123
3.6.1.	Módulo 1	123
3.6.2.	Módulo 2	124
3.6.3.	Módulo 3	124
3.6.4.	Módulo 4	125
3.7.	Desarrollo de los Módulos de Prueba	125
3.7.1.	Desarrollo de los Módulos en la Herramienta Joomla! 1.5.....	125
3.7.1.1.	Módulo 1	125
3.7.1.2.	Módulo 2:	128
3.7.1.3.	Módulo 3:	135
3.7.1.4.	Módulo 4	146
3.7.2.	Desarrollo de los Módulos en la Herramienta TYPO3	148
3.7.2.1.	Módulo 1	148
3.7.2.2.	Módulo 2	150
3.7.2.3.	Módulo 3	155
3.7.2.4.	Módulo 4	159
3.7.3.	Desarrollo de los Módulos en la Herramienta Drupal 6.19.....	162
3.7.3.1.	Módulo 1	162
3.7.3.2.	Módulo 2:	167
3.7.3.3.	Módulo 3:	172
3.7.3.4.	Módulo 4:	181
3.8.	Análisis Comparativo	185
3.8.1.	Requerimientos para el desarrollo	187
3.8.1.1.	Determinación de Variables.....	187
3.8.1.2.	Valoraciones.....	187
3.8.1.3.	Interpretación de la Tabla III.III	188
3.8.1.4.	Calificación.....	189
3.8.2.	Facilidad de uso.....	190
3.8.2.1.	Determinación de Variables.....	190

3.8.2.2.	Valoraciones.....	190
3.8.2.3.	Interpretación de la Tabla III.IV	191
3.8.2.4.	Calificación.....	192
3.8.3.	Gestión de componentes dentro del CMS	193
3.8.3.1.	Determinación de Variables.....	193
3.8.3.2.	Valoraciones.....	193
3.8.3.3.	Interpretación de la Tabla III.V.....	194
3.8.3.4.	Calificación.....	195
3.8.4.	Estructura de las extensiones	195
3.8.4.1.	Determinación de Variables.....	196
3.8.4.2.	Valoraciones.....	196
3.8.4.3.	Interpretación de la Tabla III.VI	197
3.8.4.4.	Calificación.....	198
3.9.	Puntajes Alcanzados.....	199
3.10.	Interpretación de la Tabla III.VII y la Figura III.146.....	200
3.11.	Resultado del Análisis de la Tabla III.VII y la Figura III.146	200
3.12.	Conclusión	202
3.13.	Comprobación de la Hipótesis.....	202

CAPÍTULO IV

DESARROLLO DEL PORTAL WEB

4.1.	Parte Teórica.....	205
4.1.1.	Introducción.....	205
4.1.2.	Desarrollo	206
4.2.	Parte práctica	210
4.2.1.	Introducción.....	210
4.2.2.	Desarrollo	211
4.2.2.1.	Ingeniería de la Información	211
4.2.2.1.1.	Definición del Ámbito del Software.....	211
4.2.2.1.2.	Antecedentes Tecnológicos.....	212
4.2.2.1.2.1	Recursos Hardware.....	212

4.2.2.1.2.2 Recursos Software	212
4.2.2.1.2.3 Recursos Humanos	213
4.2.2.1.3. Definición del Problema	213
4.2.2.1.3.1 Definir Alternativas de Solución	213
4.2.2.2. Análisis Orientado a Objetos.....	214
4.2.2.2.1. Caso de Uso: Sistema Actual	214
4.2.2.2.2. Caso de Uso 2	216
4.2.2.2.2.1 Diagrama de Caso de Uso 2	216
4.2.2.2.3. Caso de Uso 3	217
4.2.2.2.3.1 Diagrama de Caso de Uso 3	217
4.2.2.2.4. Caso de Uso 4	218
4.2.2.2.4.1 Diagrama de Caso de Uso 4	218
4.2.2.2.5. Definir el Modelo Conceptual	219
4.2.2.2.6. Definir Diagrama de Secuencia	219
4.2.2.3. Diseño Orientado a Objetos.....	220
4.2.2.3.1. Diagrama de base de datos	220
4.2.2.3.2. Diagrama de Componentes	221
4.2.2.4. Implementación.....	221
4.2.2.5. Pruebas	229
4.2.2.6. Implantación del Portal Web.....	234

CONCLUSIONES

RECOMENDACIONES

RESUMEN

GLOSARIO

BIBLIOGRAFÍA

ÍNDICE DE FIGURAS

Figura II. 1: Concepto de CMS	40
Figura II. 2: Ciclo de publicación de contenidos.....	42
Figura II. 3: Operaciones de un CMS ¹²	46
Figura II. 4: Diagrama de Flujo del Patrón MVC	72
Figura II. 5: Diagrama de secuencia del patrón MVC ²⁶	74
Figura III. 6: Resultado encuesta pregunta nº 1.....	79
Figura III. 7: Resultado encuesta pregunta nº 2.....	80
Figura III. 8: Resultado encuesta pregunta nº 3.....	80
Figura III. 9: Resultado encuesta pregunta nº 4.....	80
Figura III. 10: Resultado encuesta pregunta nº 5.....	81
Figura III. 11: Estructura Organizativa de Joomla! ³⁰	89
Figura III. 12: Estructura del componente Prueba	94
Figura III. 13: Diagrama de Funcionamiento de Drupal	103
Figura III. 14: Visitas al Sitio Web.....	106
Figura III. 15: Estructura Organizativa de Typo3.....	110
Figura III. 16: Modelo de dominio de la extensión de ejemplo "BlogExample"	114
Figura III. 17: Ciclo simplificado a través de una extensión/componente.....	114
Figura III. 18: Creación de los directorios y archivos .php necesarios, para el desarrollo del componente com_libreria – Módulo 1 Herramienta Joomla! 1.5.....	125
Figura III. 19: Creación dentro de la carpeta models, el archivo articulo.php– Módulo 1 Herramienta Joomla! 1.5.....	126
Figura III. 20: Creación dentro de la carpeta views\articulo, el archivo view.html.php – Módulo 1 Herramienta Joomla! 1.5	126
Figura III. 21: Creación dentro de la carpeta views\articulo\tmpl, el archivo default.php – Módulo 1 Herramienta Joomla! 1.5	127
Figura III. 22: Ejecución de la aplicación – Módulo 1 Herramienta Joomla! 1.5.....	127
Figura III. 23: Creación de la tabla jos_libreria_articulo en la base de datos deMySQL – Módulo 2 Herramienta Joomla! 1.5.....	128

Figura III. 24: Creación del archivo articulo.php, dentro del directorio tables – Módulo 2 Herramienta Joomla! 1.5	128
Figura III. 25: Creación de la función save en el controlador controller.php – Módulo 2 Herramienta Joomla! 1.5	129
Figura III. 26: Creación dentro de la carpeta models, el archivo articulos.php– Módulo 2 Herramienta Joomla! 1.5	129
Figura III. 27: Creación dentro de la carpeta views\articulos, el archivo view.html.php - Módulo 2 Herramienta Joomla! 1.5	130
Figura III. 28: Creación dentro de la carpeta views\articulos\tmpl, el archivo default.php - Módulo 2 Herramienta Joomla! 1.5	130
Figura III. 29: Ejecución de la aplicación en el Back-End, ingreso de un artículo - Módulo 2 Herramienta Joomla! 1.5	131
Figura III. 30: Ejecución de la aplicación en el Back-End, listado de los artículos ingresados - Módulo 2 Herramienta Joomla! 1.5	131
Figura III. 31: Creación del componente com_libreria en el Front-End - Módulo 2 Herramienta Joomla! 1.5	132
Figura III. 32: Modelo 3, creación del archivo artículo.php dentro de la carpeta models en el Front-End - Módulo 2 Herramienta Joomla! 1.5	132
Figura III. 33: Creación del directorio y archivos necesarios para cada una de las vistas en el Front-End - Módulo 2 Herramienta Joomla! 1.5	133
Figura III. 34: Creación del archivo view.html.php en el directorio ..\views\articulo en el Front-End - Módulo 2 Herramienta Joomla! 1.5....	133
Figura III. 35: Creación del archivo default.php en el directorio ..\views\articulo\tmpl en el Front-End - Módulo 2 Herramienta Joomla! 1.5	134
Figura III. 36: Ejecución de la aplicación en el Front-End, datos de un artículo escogido - Módulo 2 Herramienta Joomla! 1.5	134
Figura III. 37: Creación de las nuevas tablas en la base de datos de MySQL – Módulo 3 Herramienta Joomla! 1.5	135
Figura III. 38: Nueva estructura de la tabla jos_libreria_articulo en la base de datos de MySQL – Módulo 3 Herramienta Joomla! 1.5.....	135

Figura III. 39: Creación de los archivos correspondientes a las tablas del componente en la base de datos, dentro del directorio tables – Módulo 3 Herramienta Joomla! 1.5.....	136
Figura III. 40: Creación del archivo categoria.php, dentro del directorio tables – Módulo 3 Herramienta Joomla! 1.5	136
Figura III. 41: Creación del patrón MVC, para el desarrollo del componente expandido com_libreria en el back-end – Módulo 3 Herramienta Joomla! 1.5	136
Figura. 42: Creación los archivos dentro de la carpeta models, modelos correspondientes para la obtención de datos específicos – Módulo 3 Herramienta Joomla! 1.5.....	137
Figura III. 43: Creación del archivo categorias.php, modelo para la obtención de los datos para el listado de las categorías existentes en el back-end – Módulo 3 Herramienta Joomla! 1.5	137
Figura III. 44: Creación del archivo categoria.php, modelo para la obtención de los datos para una categoría específica existente en el back-end – Módulo 3 Herramienta Joomla! 1.5	138
Figura III. 45: Modificación del archivo controller.php, controlador para la gestión de datos de las tablas existentes en el back-end – Módulo 3 Herramienta Joomla! 1.5.....	138
Figura III. 46: Creación dentro de la carpeta views\categorias, el archivo view.html.php en el back-end - Módulo 3 Herramienta Joomla! 1.5	139
Figura III. 47: Creación de la vista para el listado de categorias en el back-end - Módulo 3 Herramienta Joomla! 1.5.....	139
Figura III. 48: Creación dentro de la carpeta views\categorias, el archivo view.html.php en el back-end - Módulo 3 Herramienta Joomla! 1.5	140
Figura III. 49: Creación dentro de la carpeta views\categorias\tmpl, el archivo default.php en el back-end - Módulo 3 Herramienta Joomla! 1.5	140
Figura III. 50: Pantalla de visualización del listado de artículos existentes, en el back-end - Módulo 3 Herramienta Joomla! 1.5	141
Figura III. 51: Pantalla de visualización de un artículo específico, en el back-end - Módulo 3 Herramienta Joomla! 1.5.....	141

Figura III. 52: Creación del patrón MVC, para el desarrollo del componente expandido com_libreria en el front-end - Módulo 3 Herramienta Joomla! 1.5	142
Figura III. 53: Creación del archivo controller.php, controlador del componente expandido en el front-end - Módulo 3 Herramienta Joomla! 1.5	142
Figura III. 54: Creación de los archivos para cada modelo necesario en el componente en el front-end - Módulo 3 Herramienta Joomla! 1.5	143
Figura III. 55: Creación del archivo categorias.php, modelo para la obtención de los datos para el listado de las categorías existentes en el front-end - Módulo 3 Herramienta Joomla! 1.5	143
Figura III. 56: Creación de los directorios para cada vista correspondientes, el archivo en el front-end - Módulo 3 Herramienta Joomla! 1.5	144
Figura III. 57: Modificación dentro de la carpeta views\categorias, el archivo view.html.php en el front-end - Módulo 3 Herramienta Joomla! 1.5	144
Figura III. 58: Modificación dentro de la carpeta views\categorias\tmpl, el archivo default.php en el front-end - Módulo 3 Herramienta Joomla! 1.5	144
Figura III. 59: Visualización del listado de Categorías en el front-end - Módulo 3 Herramienta Joomla! 1.5	145
Figura III. 60: Visualización de un artículo específico en el front-end - Módulo 3 Herramienta Joomla! 1.5	145
Figura III. 61: Menú para la administración de extensiones - Módulo 4 Herramienta Joomla! 1.5.....	146
Figura III. 62: Instalación de componentes - Módulo 4 Herramienta Joomla! 1.5.....	146
Figura III. 63: Búsqueda del componente a instalar - Módulo 4 Herramienta Joomla! 1.5	146
Figura III. 64: Componente instalado correctamente - Módulo 4 Herramienta Joomla! 1.5.....	147
Figura III. 65: Gestión de componentes, desinstalación, habilitación, deshabilitación - Módulo 4 Herramienta Joomla! 1.5	147
Figura III. 66: Desinstalación exitosa de un componente - Módulo 4 Herramienta Joomla! 1.5.....	147

Figura III. 67: Creación de la extensión LibreriaArticulos, con la aplicación kickstarterwizard - Módulo 1 Herramienta Typo3 v. 4	148
Figura III. 68: Estructura de archivos de la extensión - Módulo 1 Herramienta Typo3 v. 4	148
Figura III. 69: Archivo tca.php con los datos para nuestra interfaz de usuario - Módulo 1 Herramienta Typo3	149
Figura III. 70: Añadir nuevo elemento de artículo de librería - Módulo 1 Herramienta Typo3.....	149
Figura III. 71: Formulario de ingreso de datos de un Artículo de librería - Módulo 1 Herramienta Typo3.....	150
Figura III. 72: Creación de los campos de la tabla tx_libreriaarticulos_articulo con la aplicación kickstarterwizard - Módulo 2 Herramienta Typo3.....	150
Figura III. 73: Tabla creada en la base de datos, con sus respectivos campos - Módulo 2 Herramienta Typo3	151
Figura. 74: Código necesario para el ingreso de datos de un artículo en la base de datos en el archivo ext_tables.php - Módulo 2 Herramienta Typo3	151
Figura III. 75: Ingreso de datos de un nuevo artículo de librería en el back-end - Módulo 2 Herramienta Typo3	152
Figura. 76: Visualización del artículo ingresado en el back-end de nuestra aplicación - Módulo 2 Herramienta Typo3.....	152
Figura. 77: Archivo class.tx_LibreriaArticulos_pi1.php, con el código para visualizar el listado de artículos en el front-end - Módulo 2 Herramienta Typo3.....	153
Figura III. 78: Visualización del listado de artículos en el Front-End - Módulo 2 Herramienta Typo3.....	153
Figura III. 79: Archivo class.tx_LibreriaArticulos_pi1.php, con el código para visualizar un artículo específico en el front-end - Módulo 2 Herramienta Typo3.....	154
Figura III. 80: Visualización de un artículo específico en el Front-End - Módulo 2 Herramienta Typo3.....	154

Figura III. 81: Creación de los campos de la tabla tx_libreriaarticulos_categoria con la aplicación kickstarterwizard - Módulo 3 Herramienta Typo3.....	155
Figura III. 82: Tabla creada en la base de datos, con sus respectivos campos - Módulo 3 Herramienta Typo3	156
Figura III. 83: Código necesario para el ingreso de datos de un artículo en la base de datos en el archivo ext_tables.php - Módulo 3 Herramienta Typo3	156
Figura III. 84: Ingreso de datos de una nueva categoría en el back-end - Módulo 3 Herramienta Typo3.....	157
Figura III. 85: Visualización de la categoría ingresada en el back-end de nuestra aplicación - Módulo 3 Herramienta Typo3	157
Figura. 86: Archivo class.tx_LibreriaArticulos_pi3.php, con el código para visualizar el listado de categorías en el front-end - Módulo 3 Herramienta Typo3.....	158
Figura III. 87: Visualización del listado de categorías en el Front-End - Módulo 3 Herramienta Typo3.....	158
Figura III. 88: Menú para la administración de extensiones - Módulo 4 Herramienta Typo3.....	159
Figura III. 89: Instalación de extensiones - Módulo 4 Herramienta Typo3.....	159
Figura. 90: Búsqueda del componente a instalar - Módulo 4 Herramienta Typo3.....	160
Figura III. 91: Extensión instalada correctamente- Módulo 4 Herramienta Typo3.....	160
Figura. 92: Gestión de componentes, desinstalación - Módulo 4 Herramienta Typo3.....	161
Figura III. 93: Desinstalación exitosa de una extensión - Módulo 4 Herramienta Typo3.....	161
Figura III. 94: Creación del directorio lib_mod1 – Módulo 1 Herramienta Drupal	162
Figura III. 95: Archivos necesarios para la creación del Módulo 1 Herramienta Drupal.....	162

Figura III. 96: Contenido del archivo lib_mod1.info – Módulo 1 Herramienta Drupal	163
Figura III. 97: Contenido del archivo lib_mod1.install – Módulo 1 Herramienta Drupal.....	163
Figura III. 98: Contenido de la tabla lib_mod1_datos – Módulo 1 Herramienta drupal	164
Figura III. 99: Contenido del archivo lib_mod1.module – Módulo 1 Herramienta drupal	164
Figura III. 100: Activación del Módulo 1 Herramienta Drupal	165
Figura III. 101: Lugar de la visualización del Módulo 1 Herramienta Drupal	165
Figura III. 102: Librería Módulo 1 - Módulo 1 Heramienta Drupal	166
Figura. 103: Formulario de ingreso de un nuevo artículo - Módulo 1 Herramienta Drupal.....	166
Figura III. 104: Archivos necesarios para la creación del Módulo 2 Herramienta Drupal.....	167
Figura III. 105: Librería Módulo 2 - Módulo 2 Herramienta Drupal	167
Figura. 106: Contenido del archivo lib_mod2.info - Módulo 2 Herramienta Drupal	167
Figura III. 107: Contenido del archivo lib_mod2.install - Módulo 2 Herramienta Drupal.....	168
Figura III. 108: Contenido de la tabla lib_mod2_datos - Módulo 2 Herramienta Drupal.....	169
Figura III. 109: Contenido del archivo lib_mod2.module - Módulo 2 Herramienta Drupal.....	169
Figura III. 110: Lugar de la visualización del módulo 2 Herramienta Drupal	170
Figura III. 111: Librería Módulo 2 - Módulo 2 Herramienta Drupal	170
Figura III. 112: Formulario de ingreso de un nuevo artículo - Módulo 2 Herramienta Drupal.....	171
Figura. 113: Listado de productos - Módulo 2 Herramienta Drupal.....	171
Figura. 114: Enlace del listado de productos - Módulo 2 Herramienta Drupal	172

Figura III. 115: Archivos necesarios para la creación del módulo 3 - Módulo 3 Herramienta Drupal	172
Figura III. 116: Contenido del archivo lib_mod3.info - Módulo 3 Herramienta Drupal.....	173
Figura III. 117: Contenido del archivo lib_mod3.install - Módulo 3 Herramienta Drupal.....	173
Figura III. 118: Contenido de la tabla lib_mod3_articulo - Módulo 3 Herramienta Drupal.....	174
Figura III. 119: Contenido de la tabla lib_mod3_articulo_proveedor - Módulo 3 Herramienta Drupal	174
Figura III. 120: Contenido de la tabla lib_mod3_categoria - Módulo 3 Herramienta Drupal.....	175
Figura III. 121: Contenido de la tabla lib_mod3_proveedor - Módulo 3 Herramienta Drupal.....	175
Figura III. 122: Contenido de la tabla lib_mod3_subcategoria - Módulo 3 Herramienta Drupal.....	175
Figura. 123: Contenido del archivo lib_mod3.module - Módulo 3 Herramienta Drupal.....	176
Figura III. 124: Activación del módulo 3 - Módulo 3 Herramienta Drupal.	177
Figura. 125: Lugar de la visualización del módulo 3 - Módulo 3 Herramienta Drupal	177
Figura III. 126: Librería Módulo 3 - Módulo 3 Herramienta Drupal	178
Figura III. 127: Formulario Agregar Categoría - Módulo 3 Herramienta Drupal	178
Figura III. 128: Formulario Agregar Subcategoría - Módulo3 Herramienta Drupal	178
Figura III. 129: Formulario Agregar Proveedor - Módulo 3 Herramienta Drupal	179
Figura III. 130: Formulario Agregar Artículo - Módulo 3 Herramienta Drupal	179
Figura III. 131: Listado de Productos - Módulo 3 Herramienta Drupal	180
Figura III. 132: Formulario Registrar Venta - Módulo 3 Herramienta Drupal	180

Figura III. 133: Listado de productos - Módulo 3 Herramienta Drupal	180
Figura III. 134: Descarga de módulos - Módulo 4 Herramienta Drupal	181
Figura III. 135: Ubicación del módulo descargado - Módulo 4 Herramienta Drupal	181
Figura III. 136: Activación del módulo descargado - Módulo 3 Herramienta Drupal	182
Figura. 137: Mensaje de configuración satisfactoria - Módulo 4 Herramienta Drupal	182
Figura III. 138: Asignación de permisos sobre el módulo descargado - Módulo 4 Herramienta Drupal	183
Figura III. 139: Desinstalación del módulo descargado - Módulo 4 Herramienta Drupal.....	183
Figura III. 140: Confirmación de desinstalación del módulo descargado - Módulo 4 Herramienta Drupal	184
Figura III. 141: Mensaje de desinstalación exitosa del módulo descargado - Módulo 4 Herramienta Drupal	184
Figura III. 142: Comparación de porcentajes parámetro Requerimientos para el desarrollo	189
Figura III. 143: Comparación de porcentajes parámetro Facilidad de uso	192
Figura III. 144: Comparación de porcentajes parámetro Facilidad de uso	195
Figura III. 145: Estructura de las Extensiones.....	198
Figura III. 146: Diagrama General de Resultados.....	200
Figura IV. 147: Desarrollo Iterativo de la Aplicación	206
Figura IV. 148: Diagrama de Caso de Uso 2	216
Figura IV. 149: Diagrama de Caso de Uso 3	217
Figura IV. 150: Diagrama de Caso de Uso 4	218
Figura. 151: Modelo Conceptual.....	219
Figura IV. 152: Diagrama de Secuencia	219
Figura IV. 153: Diagrama de Base de Datos	220
Figura IV. 154: Diagrama de Componentes	221
Figura IV. 155: Estructura del componente com_libreria, back-end.	222
Figura IV. 156: Pantalla inicial del componente com_libreria, back-end...	222

Figura IV. 157: Forma de visualización de los listados de datos, back-end	223
Figura IV. 158: Forma de visualización de formularios, back-end	223
Figura IV. 159: Estructura del componente com_libreria,front-end	224
Figura IV. 160: Forma de visualización de los listados, de los filtros a los que pertenecen los artículos de librería, front-end	224
Figura IV. 161: Forma de visualización de un artículo de librería, front-end	225
Figura IV. 162: Estructura del componente com_libreria_cart, back-end .	226
Figura IV. 163: Listado de los pedidos de artículos realizados, back-end .	226
Figura IV. 164: Detalles de un pedido escogido, back-end.....	227
Figura IV. 165: Estructura del componente com_libreria_cart, front-end .	227
Figura IV. 166: Botón para agregar al carrito de compras, front-end	228
Figura IV. 167: Visualización del Carrito de Compras, front-end	228
Figura IV. 168: Formulario datos cliente para el pedido, front-end	228
Figura IV. 169: Comprobante de la orden de pedido, front-end.....	229
Figura IV. 170: Porcentajes resultantes de los Intervalos de tiempo	230
Figura IV. 171: Porcentajes resultantes de los Intervalos de tiempo	232
Figura IV. 172: Porcentajes resultantes de los tiempos que se usan en las tareas de gestión de los artículos	233
Figura IV. 173: Tiempos resultantes del tiempo de demora	234

ÍNDICE DE TABLAS

Tabla III. I: Escala de Puntuación para calificación de Parámetros	185
Tabla III. II: Escalas de valoración cualitativa	185
Tabla III. III: Requerimientos para el desarrollo.....	188
Tabla III. IV: Facilidad de uso	191
Tabla III. V: Gestión de componentes dentro del CMS	194
Tabla III. VI: Estructura de las extensiones	197
Tabla III. VII: Tabla General de Resultados	199
Tabla III. VIII: Comparación Joomla! – Typo3	203
Tabla III. IX: Comparación Joomla! - Drupal.....	203
Tabla III. X: Comparación Typo3 - Drupal	203
Tabla III. XI: Mejor CMS (Sin comparación – Con Comparación)	204
Tabla IV. XII: Características Hardware para el Sistema.....	212
Tabla IV. XIII: Características Software para el Sistema.....	212
Tabla IV. XIV: Recursos Humanos	213
Tabla IV. XV: Caso de Uso 1: Sistema Actual.....	215
Tabla IV. XVI: Caso de Uso 2: Ingreso al Sistema	216
Tabla IV. XVII: Caso de Uso 3: Crear y Gestionar Campeonatos.....	217
Tabla IV. XVIII: Caso de Uso 4: Emitir Reportes	218
Tabla IV. XIX: Tiempo de búsqueda de un artículo	230
Tabla IV. XX: Tiempo de demora para el pedido de artículos.....	231
Tabla IV. XXI: Tiempo de gestión de Artículos.....	232
Tabla IV. XXII: Tiempo de demora para la comprobación de la orden	233

INTRODUCCIÓN

El uso y desarrollo del Internet ha dado lugar a la creación de comunidades de información y conocimiento, así como también al desarrollo de un conjunto de herramientas que permiten a las personas su desarrollo en el ámbito personal y profesional.

Las empresas se ven en la necesidad de promocionar sus productos y/o servicios en la web, pero las herramientas propietarias son costosas, es por tal motivo que hoy en día se opta por soluciones en Open Source.

En los últimos tiempos los gestores de administración de contenidos, CMS (Content Management System), han aparecido como una solución para la implementación de portales web para la pequeña, mediana y gran empresa.

Joomla!, Typo3, Drupal son considerados los CMS (Content Management System) más utilizados.

Existen componentes ya desarrollados que realizan tareas específicas, estos no tienen ningún formato establecido, por lo que resulta difícil su reutilización o adaptación.

En la librería "Majos" todas sus transacciones se realizan manualmente y en muchos casos los productos no son adecuadamente registrados provocando que estos se extravíen.

El propósito del estudio consistió en comparar el desarrollo de componentes en Joomla!, Typo3 y Drupal como sistemas de administración de contenido encontrando el mejor para la implementación del portal web de la Librería Majos.

Dentro del ámbito empresarial, lo importante es concretar negocios, realizar ventas, transacciones comerciales, entre otras, que enriquecen sustancialmente las oportunidades de crecimiento y desarrollo de las empresas, esto se debe realizar en el menor tiempo posible y abarcando el mayor mercado, para lograrlo es necesario que cada empresa por más pequeña que esta sea realice sus operaciones comerciales a través de la web.

El conocer la mejor herramienta para el desarrollo de componentes para el comercio dentro de los CMS's para la implementación de portales web, nos da una ventaja para el desempeño laboral y el saber administrar, desarrollar y reutilizar estos componentes nos abre las puertas el momento de incursionar en el competitivo mundo del campo profesional.

Para conocer el mejor CMS se determinó cuatro parámetros con tres variables de medición para cada uno y además se desarrollo cinco módulos tanto para Joomla! como para Drupal y Typo3.

En el primer capítulo se hizo una breve descripción del presente trabajo, en el segundo capítulo está todo el fundamento teórico necesario para el estudio, el tercer capítulo es el más importante ya que contiene la información de los tres CMS, los parámetros de medición, los módulos desarrollados para cada CMS, el análisis y el resultado. El cuarto capítulo tiene el desarrollo de la aplicación web para la Librería Majo's en el CMS ganador.

CAPÍTULO I

GENERALIDADES

1.1. Justificación

1.1.1. Teórica

Joomla!, Typo3 y Drupal son sistemas de gestor de contenidos dinámicos CMS (Content Management System) que permite crear sitios web de alta interactividad, profesionalidad y eficiencia. La administración de estos está enteramente basada en la gestión online de contenidos. Es gestión online porque todas las acciones que realizan los administradores de sitios, ya sea para modificar, agregar o eliminar contenidos se realiza exclusivamente mediante un navegador web (browser) conectado a Internet, es decir, a través del protocolo HTTP (Protocolo de transferencia de hipertexto).

Joomla!, Typo3 y Drupal hacen uso del lenguaje PHP (Hypertext Pre Processor) y SQL (Structure Query Language). Utiliza bases de datos relacionales, más específicamente MySQL. Tanto PHP como MySQL son programas Opensource de libre distribución y uso, convirtiéndoles de esta manera en herramientas de código libre, y al ser Joomla!, Typo3 y Drupal aplicaciones Web, funcionan en servidores de páginas web (HTTP Servers).

Esto convierte a Joomla!, Typo3 y Drupal en poderosas herramienta de difusión Información, de Marketing Online, de negocios por Internet, de administración de proyectos.

Los componentes en Joomla!, Typo3 y Drupal ayudan a personalizar las aplicaciones, ya que son herramientas de diseño de sitios web, que tienen la capacidad de lograr que una página web se ajuste a especificaciones propias, permitiendo llegar al completo potencial del sitio web. Los componentes en Joomla!, Typo3 y Drupal son la clave para la consecución de los objetivos del proyecto.

TYPO3.- Posiblemente se trate del más indicado para el mundo de las empresas. A nivel europeo creo que es el más desarrollado, siendo utilizado tanto por pymes como por grandes grupos empresariales. Typo3 ofrece una gama de funcionalidades muy completas y muy evolucionadas. Sus posibilidades de parametrización y su interfaz de fácil utilización le permiten ser utilizado por especialistas no informáticos. Es un CMS inevitable, evolutivo, potente y estable¹.

TYPO3 es una herramienta sobre todo de gestión de contenido muy completa².

¹ CMS: <http://www.aplicacionesempresariales.com/elijendo-el-mejor-open-source-para-tu-empresa.html>. (2009-05-10)

² Typo3: <http://www.typo3performance.com/typo3-cms/> (2009 -05 -10)

DRUPAL es un sistema de gestión de contenido híbrido y un poco atípico. Si se orienta a comunidades, no es sin embargo un portal y es ahí donde reside su gran fuerza.

DRUPAL permite administrar una información como un nudo, este nudo puede ser un foro, artículo, noticia, tutorial, comentario, libro cooperativo. La forma no está ya fija a una arquitectura modelo, es pues fácilmente modificable. Como contrapartida de esta libertad, es necesario pasar un poco de tiempo para familiarizarse con su lógica particular³.

Por segundo año consecutivo DRUPAL se llevó el premio al mejor CMS del 2008⁴.

JOOMLA supone la continuidad de Mambo, un CMS de fácil utilización, potente y ergonómico adaptado a lugares escaparate, comerciales, asociativos, institucionales o comunitarios y personales. Al menos eso es lo que dicen sus desarrolladores. Muchas personas aprecian su capacidad para autorizar al usuario a publicar el contenido "sin tener que ser un experto".

Joomla posee numerosas funcionalidades y extensiones. ¡La última versión Joomla! 1.5 quiere marcar la evolución más importante en la historia de Joomla!. Este esfuerzo es caracterizado por el hecho de que se está rehaciendo el código de Joomla e integra una nueva API que permite a los desarrolladores proporcionar aplicaciones más robustas e innovadoras³.

1.1.2.Práctica

Al desarrollar una aplicación web en la librería "Majo's", se automatizará el proceso de control de información además se permitirá el comercio electrónico proporcionado un valor agregado a la aplicación.

³ CMS: <http://www.aplicacionesempresariales.com/elijendo-el-mejor-open-source-para-tu-empresa.html>. (2009-05-10)

⁴ CMS: <http://www.planetacms.com/intereses-generales/actualidad/el-mejor-cms-del-2008> (2009 -05-10)

Todos los procesos que se realizaban manualmente, registrándolos en documentos físicos, se convertirán en información digital, llevando a las búsquedas de productos específicos, a utilizar tiempos mínimos mejorando la eficiencia en atención a los clientes.

Los inventarios de productos se controlaran de mejor manera, evitando pérdidas de los mismos, brindando alertas oportunas de escasez de algún producto para su pronto abastecimiento, evitando así la pérdida de ventas.

El portal web aparte de su publicidad online, poseerá dos componentes primordiales la gestión de los productos de librería automatizando sus procesos, y el segundo componente que contendrá el carrito de compras el que permitirá escoger todos los artículos necesarios al cliente y determinará el valor total que deberá cancelar.

La aplicación se realizará bajo el sistema operativo Windows XP para facilidad de nuestro usuario quien está familiarizado a este, se utilizara como gestor de base de datos mysql, y servidor Apache, por ser de libre distribución.

1.2. Antecedentes

A medida que el acceso y el uso del internet han ido creciendo en gran medida, las empresas se ven en la necesidad de promocionar sus productos y/o servicios en la web, pero las herramientas propietarias son costosas, es por tal motivo que las empresas de hoy en día optan por soluciones en Open Source.

En los últimos tiempos los gestores de administración de contenidos, CMS (Content Management System), se centran en el desarrollo de

aplicaciones web no tan dinámicas, lo que no permitían la adaptación de nuevos requerimientos específicos.

Joomla!, Typo3, Drupal son considerados los CMS (Content Management System) más utilizados, los cuales para lograr dinamismo en sus aplicaciones web, necesitan integrar complementos, componentes, módulos y plugins, permitiendo aplicaciones eficientes con gran interacción con los usuarios.

En la actualidad existen componentes desarrollados que realizan tareas específicas, pero se torna difícil adaptar o personalizar dichos componentes porque poseen una lógica de programación diferente.

En la librería "Majos" todas sus transacciones se realizan manualmente y en muchos casos los productos no son adecuadamente registrados provocando que estos se extravíen.

Toda la información de los productos se encuentra registrada en documentos físicos, por lo que resulta dificultoso y toma periodos de tiempo amplios en realizar búsquedas de productos específicos, malogrando la eficiencia en la atención, por lo que se desarrollará una aplicación web para solucionar estos inconvenientes.

1.3. Objetivos

1.3.1.General

Comparar el desarrollo de componentes en Joomla!, Typo3 y Drupal como sistemas de administración de contenido encontrando el mejor para la implementación del portal web de la Librería Majo's.

1.3.2.Específicos

- Establecer los parámetros de medición para la comparación del desarrollo de componentes en los CMS (content management system) Joomla!, Typo3 y Drupal.
- Estudio comparativo de Joomla!, Typo3 y Drupal en el desarrollo de componentes como CMS (content management system).
- Desarrollo e integración del componente específico para la gestión de los artículos de librería y el componente de carrito de compras que permite la reservación online a la aplicación web en el CMS más potente usando el patrón de diseño MVC (Modelo, vista, controlador).

1.4. Hipótesis

Comparar el desarrollo de componentes en sistemas de administración de contenido encontrando el mejor para la implementación del portal web de la Librería Majo´s.

CAPÍTULO II

FUNDAMENTO TEÓRICO

2.1. La Web

La *red* o *www* de *World Wide Web*, es básicamente un medio de comunicación de texto, gráficos y otros objetos multimedia a través de Internet, es decir, la web es un sistema de hipertexto que utiliza Internet como su mecanismo de transporte o desde otro punto de vista, una forma gráfica de explorar Internet.

La web fue creada en 1989 en un instituto de investigación de Suiza, la web se basa en buscadores y el protocolo de transporte de hipertexto (hypertexttransport protocol (http)). La mayoría de los documentos de la web se crean utilizando lenguaje HTML (hypertextmarkuplanguage).

Es importante saber que web o www no son sinónimo de Internet, la web es un subconjunto de Internet que consiste en páginas a las que se puede acceder usando un navegador. Internet es la red de redes donde reside

toda la información. Tanto el correo electrónico, como FTPs, juegos, etc. son parte de Internet, pero no de la Web.

Para buscar hipertexto se utilizan programas llamados buscadores web que recuperan trozos de información (llamados *documentos* o *páginas web*) de los servidores web y muestran en la pantalla del ordenador de la persona que está buscando la información gráfica, textual o video e incluso audio.

Después se pueden seguir enlaces o hyperlinks en cada página a otros documentos o incluso devolver información al servidor para interactuar con él. Al acto de seguir un enlace tras otro a veces se le llama navegar en Internet.

La web se ha convertido en un medio muy popular de publicar información en Internet, y con el desarrollo del protocolo de transferencia segura (secured server protocol (https)), la web es ahora un medio de comercio electrónico donde los consumidores pueden escoger sus productos on-line y realizar sus compras utilizando la información de sus tarjetas bancarias de forma segura⁵.

2.1.1. Páginas Web

Una página web es una fuente de información adaptada para la World Wide Web (WWW) y accesible mediante un navegador de Internet. Ésta información se presenta generalmente en formato HTML y puede contener hiperenlaces a otras páginas web, constituyendo la red enlazada de la World Wide Web.

Las páginas web pueden ser cargadas de un ordenador o computador local o remoto, llamado Servidor Web, el cual servirá de HOST. El servidor web

⁵ World Wide Web: <http://www.masadelante.com/fags/www> (2009-09-11)

puede restringir las páginas a una red privada, por ejemplo, una intranet, o puede publicar las páginas en el World Wide Web. Las páginas web son solicitadas y transferidas de los servidores usando el Protocolo de Transferencia de Hipertexto (HTTP - Hypertext Transfer Protocol). La acción del Servidor HOST de guardar la página web, se denomina "HOSTING".

Las páginas web pueden consistir en archivos de texto estático, o se pueden leer una serie de archivos con código que instruya al servidor cómo construir el HTML para cada página que es solicitada, a esto se le conoce como Página Web Dinámica.

Una página Web está compuesta por uno o varios documentos html relacionados entre sí mediante hipervínculos (enlaces). Además estos documentos pueden contener otros elementos como pueden ser imágenes, sonidos, animaciones multimedia y aplicaciones.

2.1.2. Estándares Web

Se destacan los siguientes estándares:

- el Identificador de Recurso Uniforme (URI), que es un sistema universal para referenciar recursos en la Web, como páginas web,
- el Protocolo de Transferencia de Hipertexto (HTTP), que especifica cómo se comunican el navegador y el servidor entre ellos,
- el Lenguaje de Marcado de Hipertexto (HTML), usado para definir la estructura y contenido de documentos de hipertexto.
- el Lenguaje de Marcado Extensible (XML), usado para describir la estructura de los documentos de texto.

Lo siguiente es una lista de los documentos que definen los tres estándares principales de la Web:

- **UniformResourceLocators (URL)** (en castellano, Localizador de Recursos Uniforme)
 - RFC 1738, Localizador de Recursos Uniforme (URL) (Diciembre de 1994)
 - RFC 3986, UniformResourceIdentifier (URI) (en castellano, Identificador de Recursos Uniforme): Sintaxis general (Enero de 2005)

- **Hypertext Transfer Protocol (HTTP)** (en castellano, Protocolo de Transferencia de Hipertexto)
 - RFC 1945, Especificación de HTTP/1.0 (Mayo de 1996)
 - RFC 2616, Especificación de HTTP/1.1 (Junio de 1999)
 - RFC 2617, Autenticación HTTP
 - HTTP/1.1 Especificación de errores de HTTP/1.1

- **Hypertext MarkupLanguage (HTML)** (en castellano, Lenguaje de Etiquetado de Hipertexto)
 - Internet Draft, HTML version 1
 - RFC 1866, HTML version 2.0
 - Referencia de la especificación HTML 3.2
 - Especificación de HTML 4.01
 - Especificación de HTML Extensible (XHTML)⁶

2.1.3. Tecnologías Web

Las tecnologías web implican un conjunto de herramientas que nos facilitarán lograr mejores resultados a la hora del desarrollo de un sitio web⁶.

⁶ World Wide Web: http://es.wikipedia.org/wiki/World_Wide_Web#Tecnolog.C3.ADas_web (2009-09-11)

2.1.3.1. Navegadores Web

- Amaya
- Epiphany
- Galeon
- Internet Explorer
- Konqueror sobre linux
- Lynx sobre linux
- Mozilla Firefox
- Netscape Navigator
- Opera
- Safari
- Seamonkey
- Shiira
- MaikNavigator
- Flock
- Google Chrome

2.1.3.2. Servidores Web

- CERN httpd
- Servidor HTTP Apache (libre, servidor más usado del mundo)
- Servidor HTTP Cherokee
- IIS
- Resin

2.1.3.3. Otras tecnologías

- OAI-PMH
- CFM Coldfusion
- DHTML
- PHP
- ASP
- CGI
- JSP (Tecnología Java)
- .NET⁷

2.2. ¿Qué es un CMS?

2.2.1.Introducción

El término CMS es el acrónimo de la frase en inglés: Content Managment System (Sistema de Gestión de Contenidos). Algunas definiciones se centran en los programas. Así en Wikipedia (<http://es.wikipedia.org/wiki/CMS>)se define:

“Un Sistema de gestión de contenidos es un programa que permite crear una estructura de soporte (framework, o marco de trabajo) para la creación y administración de contenidos por parte de los participantes principalmente en páginas Web”.

Esta definición se centra en los programas que permiten crear las estructuras de soporte de un CMS.

⁷ World Wide Web: http://es.wikipedia.org/wiki/World_Wide_Web#Tecnolog.C3.ADas_web (2009-09-11)

Otra posible definición de trabajo hace énfasis en un aspecto importante:

Un CMS soporta la creación, administración, distribución, publicación y descubrimiento de información corporativa.

Esta definición se centra más en el ciclo de vida de los contenidos. En ello debemos pensar que este ciclo ha cambiado y que ese cambio es una de las características fundamentales de la WEB 2.0. Si partimos de los antecedentes de Web 2.0 concluiríamos que en un inicio del desarrollo de Internet existían páginas Web estáticas, cuyos contenidos se cambiaban de forma centralizada, manteniéndose incluso desactualizados y que luego se pasó a páginas dinámicas y actualizadas periódicamente con una Base de Datos, regulada y manejada por alguien.

La Web 2.0 ha pasado a ser mucho más que eso y puede considerarse como una plataforma de servicios, con contenidos que son modificados y actualizados y creados por los propios usuarios y basada en un principio clave:

Un servicio mejora automáticamente cuanto más se utiliza.

De esta forma el concepto de CMS se centra en un ciclo de creación, administración y publicación de contenidos, ver Figura II. 1:

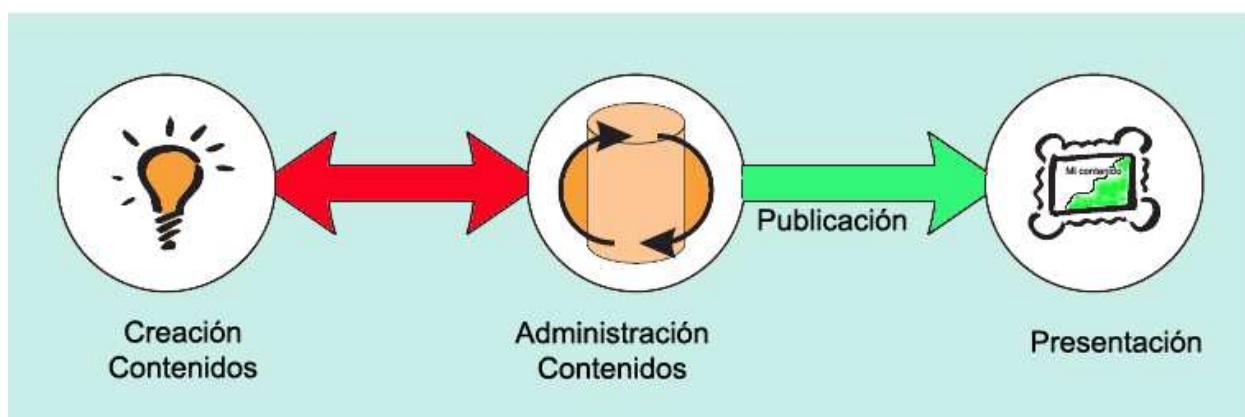


Figura II. 1: Concepto de CMS

El mismo concepto de CMS deja claro algunos beneficios decisivos en el manejo de los contenidos, en particular:

- Mejor control del proceso de autoría de contenidos
- Soporte a procesos de autoría descentralizados
- Tiempos mínimos en el cambio de páginas en la Web
- Mayor consistencia de los contenidos
- Facilidad de navegación
- Mayor flexibilidad
- Mejora de los niveles de seguridad
- Reduce la posibilidad de duplicación de información
- Posibilidad de crecimiento controlado
- Reducción de los costos de mantenimiento de sitios

Aparte de estos beneficios el empleo de los CMS permite a una empresa incrementar sus posibilidades de comunicación con clientes y público en general.

2.2.2.Precisión del Concepto de CMS

Para comprender el concepto de CMS debemos dejar claro que son las Estrategias de Comunicación las que realmente llevan a gestionar Contenidos de forma efectiva; los sistemas informáticos pueden a lo sumo proporcionar las herramientas necesarias para la publicación en línea. Por tanto al hablar de CMS (Content Management System) por lo general nos estamos refiriendo a sistemas de publicación de contenidos⁸.

⁸ CMS: <http://scrib.com/doc/9784583/Que-es-un-CMS> (2009-09-11)

2.2.2.1. El ciclo de publicación de contenidos

Si queremos entender y manejar un CMS debemos así tener presente que en la implementación de estos Sistemas subyace la definición de un ciclo de publicación de contenidos que pasa por los pasos siguientes:

- Autoría
- Edición
- Aprobación
- Planificación
- Publicación
- Vigencia

El que en muchas ocasiones en sitios personales y blogs estas tareas se realicen por el mismo individuo, no implica el que no tengamos que cumplirlas y muchos de los fracasos que podemos tener en términos de audiencia, actualización y vigencia de contenidos, obedecen a que de una forma u otra se improvisan los contenidos y se ignora la necesidad del cumplimiento de este ciclo. Por tanto el flujo de creación, y revisión para ser publicado es el siguiente: ver Figura II. 2:



Figura II. 2: Ciclo de publicación de contenidos

¿Que entendemos por contenido?

Podemos definir contenido como información que necesita ser publicada, distribuida a una audiencia y que puede encontrarse en múltiples formatos: textos, imágenes, hojas de cálculo, multimedia, PDF. Desde el punto de vista práctico la publicación puede entenderse cómo la presentación de contenidos de la forma deseada (más adecuada para una audiencia), mediante plantillas:

Las plantillas (templates) pueden definirse como las formas de presentación de los contenidos

Una característica básica de los CMS es estos Sistemas es que manejan el contenido de forma separada de la presentación, lo que posibilita la opción de distribuir el contenido en múltiples formatos: WEB, PDA, Celulares, Impresoras. Uno de los conceptos básicos de WEB 2.0 es que la misma imaginación de los usuarios define nuevas formas creativas de manejo de los contenidos, por lo que los CMS deben ofrecer gran flexibilidad, posibilidad de colaboración y facilitar una interface consistente que permita estandarización en la presentación de los contenidos⁹.

2.2.2.2. Definición de CMS

A partir de estas consideraciones podemos definir un CMS como:

Un Sistema que organiza y permite acceder desde una Base de Datos todo tipo de contenidos digitales y que se estructura en torno a cuatro aspectos esenciales:

- Autoría

⁹ CMS: <http://scrib.com/doc/9784583/Que-es-un-CMS> (2009-09-11)

- El mecanismo de colocar y dar formato al contenido digital en una Base de Datos
- Flujo de Trabajo
 - La ruta de aprobación de los contenidos
- Almacenamiento
 - La forma en que se manejan los datos en la Base correspondiente y la posibilidad de referenciar los mismos
- Publicación
 - Los mecanismos y opciones de presentación de los contenidos digitales desde la Base de Datos

Al diseñar un CMS es imprescindible partir de un análisis previo de los contenidos que se manejarán. En esta etapa debe:

- Producirse una lista de los temas e información que será incluida en el Sitio.
- Organizar la lista en grupos lógicos con una estructura jerárquica adecuada
- Crear el Mapa del Sitio¹⁰

2.2.2.3. Flujo de trabajo en un CMS

Los contenidos en un CMS se procesan a partir de una unidad básica: El Artículo. Los CMS establecen una taxonomía para clasificar los Artículos.

Todo Artículo se clasifica dentro de una Categoría que a la vez se subordina a una Sección.

¹⁰ CMS: <http://scrib.com/doc/9784583/Que-es-un-CMS> (2009-09-11)

Una de las Tareas centrales de diseñar un CMS es por tanto determinar la categorización de los contenidos dentro de la estructura jerárquica más adecuada.

En relación con el Artículo ejemplifiquemos el ciclo de vida y los niveles de seguridad que se establecen:

- Primer paso: El Artículo se crea y se envía a proceso de aprobación (Nivel de usuario: Autor)
- Segundo paso: El Artículo se edita y se chequea (Nivel de usuario: Editor)
- Tercer paso. Aprobación. Si se acepta el Artículo continua el flujo, sino se devuelve al autor para corrección (Nivel de usuario: Editor jefe).
- Cuarto paso. Planificación de publicación: Fija fecha de expiración del Artículo (Nivel de Usuario Planificador)

Tenemos así:

1. Autores (Authors),
2. Editores (Editors)
3. Editores jefes (Approvers)
4. Planificadores (Schedulers).

A estos niveles se agrega el de Administrador que puede desempeñar todas las funciones anteriores y otras (creación, aprobación y eliminación de archivos, Autores, etc.)¹¹

2.2.2.4. Operación de un CMS

Un sistema de administración de contenido funciona a menudo en el servidor del sitio Web.

¹¹ CMS: <http://scrib.com/doc/9784583/Que-es-un-CMS> (2009-09-11)

Este proceso de comunicación con el Servidor Web que aloja el CMS es generalmente mal interpretado por el usuario inexperto que no distingue claramente el proceso de subir y/o bajar contenidos, ver Figura II. 3:

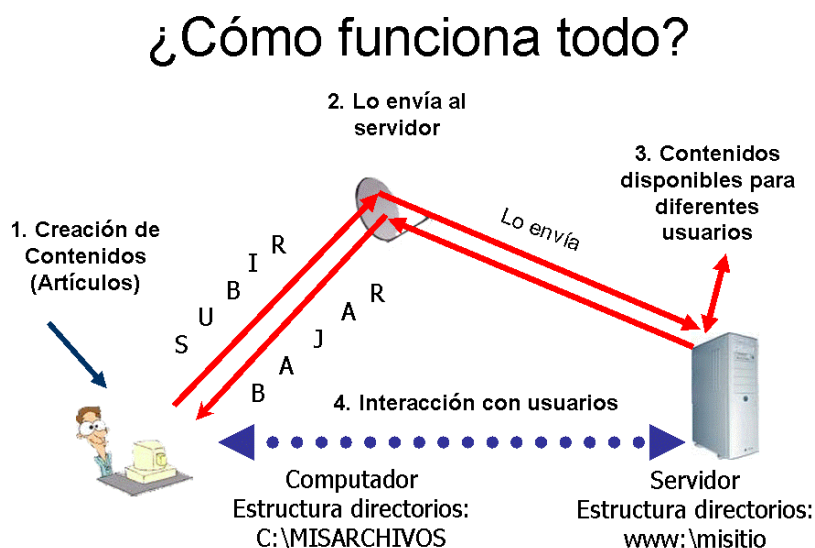


Figura II. 3: Operaciones de un CMS¹²

Los diferentes CMS proporcionan diferentes niveles de acceso: Administrador, Editor, autor. Un blog puede considerarse dentro de la clasificación general de CMS y en ese caso la misma persona realiza las diferentes funciones. De igual forma el CMS permite (si se tiene el nivel de acceso correspondiente) comunicarse con el módulo de Administración, que es el que permite las tareas generales, como:

- Clasificación de contenidos (Secciones, Categorías, Artículos)
- Administración de contenidos
- Selección de plantillas
- Determinación de estructuras de navegación

El acceso al CMS se realiza generalmente a través del navegador Web, y el proceso de subir contenidos, requiere el uso de FTP¹².

¹² CMS: <http://scrib.com/doc/9784583/Que-es-un-CMS> (2009-09-11)

2.2.3. Estructura de un CMS

Las diferentes funcionalidades ofrecidas por un CMS pueden ser separadas en diferentes categorías. Todas juntas constituyen la estructura de un CMS.

- Front-End y Back-End.

Un CMS consiste en un Front End, que es el sitio web (website), que los visitantes y los usuarios registrados pueden ver. Y un Back End, que contiene la parte de administración del sitio web, localizado en una diferente URL al sitio web; y es donde se realizan las tareas de configuración, mantenimiento, limpieza, creación de estadísticas, etc.

- Configurationsettings.

Las decisiones de configuración que se aplican al sitio web son especificados en "Configurationsettings". Esto incluye el título del sitio web, las palabras para los motores de búsqueda, opciones que permiten o prohíben darse de alta como usuario en el sitio web, y muchas otras funciones.

- Derechos de acceso (Access Rights).

En un CMS, los nombres de usuario son asignados a personas que están involucradas en el desarrollo del sitio web y su administración y poseen diferentes derechos de acceso. El rango de derechos va desde un usuario simplemente registrado, registrado como autor o editor, hasta la categoría de superadministrador, que tiene total control sobre el dominio. Según los derechos asignados, el sitio web muestra diferentes contenidos, o el usuario trabaja en diferentes partes de la administración aparte de el sitio web.

- Contenido (Content).

Un CMS permite casi cualquier tipo de contenido. Texto simple, imágenes, vínculos, música y archivos multimedia en general o una combinación de todos ellos. Para mantener una buena presentación de

los contenidos, son incluidos en estructuras jerarquizadas, siendo clasificados según secciones y categorías, que a su vez, deben ser administrados.

- Plantillas (Templates).

Una plantilla define colores, las fuentes y su tamaño, imágenes de fondo, espaciado y distribución de la página, es decir, todo lo que tiene que ver con la apariencia del sitio web.

- Extensiones. Componentes (Extensions. Components).

Todo sistema debe poder ser ampliable para poder crecer en función de los requerimientos que aparezcan. Las funcionalidades que pertenecen a determinado contexto son cubiertas por lo que conocemos como Componente. Por ejemplo, componentes típicos son una tienda online, un gestor de usuarios, un gestor de listas de correo, un foro, una galería de imágenes, un gestor de descargas. Los módulos que se necesitan integrar con los componentes son utilizados para integrar contenidos en la parte deseada dentro de la plantilla. Son ejemplos de módulos, el que permite ver los usuarios conectados en el momento, el que permite ver las estadísticas del sitio, el que muestra los artículos de contenido más reciente, etc.

- Rutina de trabajo (workflow).

Si hay varias personas trabajando con el CMS, una buena rutina de trabajo facilitará la gestión de los contenidos. Por ejemplo, si un editor revisa una lista de noticias enviadas al sitio para ser corregidas, y después de revisarlos indica que las noticias ya han sido corregidas, el publicador podrá directamente decidir si la noticia se publica en el Front Page (página principal del sitio web) o no¹³.

2.2.4. Características que un CMS debe cumplir para ser "SEO friendly"

¹³ CMS: <http://scrib.com/doc/9784583/Que-es-un-CMS> (2009-09-11)

Uno de los aspectos cruciales a la hora de decantarse por una u otra solución de Gestión de Contenidos Web, tanto si se trata de un producto Open Source como si es un software comercial, es la facilidad y la flexibilidad que ofrecen a la hora de implementar técnicas de SEO (SearchEngineOptimization).

De poco serviría que el esfuerzo de adaptación e implantación de un CMS fuera exitoso desde el punto de vista de los procedimientos de la empresa, de su adaptación a los conocimientos y experiencia de los usuarios, de su categorización e incluso de la apariencia visual si, finalmente, el contenido que genera acaba siendo mal indexado por los robots de búsqueda. Sería un éxito invisible (que es lo mismo que un fracaso).

- **Metadatos.-** Es imprescindible que el CMS que escojamos permita el control manual sobre los metadatos de nuestro sitio web, puesto que es parte esencial de cualquier campaña SEO. Aunque nuestro website pueda generar dinámicamente las etiquetas para el título de la página, la descripción y las keywords, el CMS debe facilitarnos paralelamente el pleno control manual de los mismos.
- **Gestión de URLs y categorías.-** Lo ideal es que el CMS nos ofrezca la posibilidad de generar URLs "amistosas" para los buscadores, compuestas, por ejemplo, con frases cuyas palabras estén separadas por guiones simples.

Es preferible: <http://www.martinalia.com/El-Ministerio-de-Educacion> que http://www.martinalia.com/article.php?id_article=201

Los buscadores tienen en cuenta las palabras que encuentran en la cadena que forma la URL.

El CMS también debe permitir que se pueda modificar la categoría que aparece en la URL sin que esto provoque enlaces rotos.

- **Texto de los enlaces.-** El texto que describe un enlace (link anchor text) es un aspecto importante a la hora de controlar nuestro posicionamiento. Casi todos los CMS ofrecen control sobre este aspecto dentro del texto, pero no todos ofrecen el control del mismo en la estructura de navegación.
- **Validación de enlaces.-** Un buen CMS controlará los cambios en cascada que deba realizar para evitar enlaces rotos cuando se renombre o reubique un directorio o una página en nuestro sitio web. Idealmente, debería avisarnos si se encuentran enlaces rotos (internos o externos a nuestro sitio).
- **Sitemap y XML.-** El CMS debe generar dinámicamente un Sitemap de nuestra web, pero permitiéndonos controlar en todo momento los textos de los enlaces, el Sitemap debe construirse en XML acorde con los estándares utilizados por los buscadores.
- **Atributos ALT.-** Es imprescindible que el CMS permita añadir atributos ALT (texto alternativo) a cualquier objeto incrustado como las imágenes. No sólo por cuestiones de posicionamiento, sino también de accesibilidad.
- **Redirecciones 301.-** El CMS debe ofrecer una utilidad para configurar redirecciones 301 (redirecciones permanentes) sin necesidad de tener que editar manualmente el archivo .htaccess.
- **Páginas 404 personalizadas.-** Cuando se cambia de sitio una página o se migra un sitio web completo, los buscadores seguirán apuntando hacia URLs que ya no existen, por lo que es importante disponer de páginas de error 404 que apunten hacia la dirección correcta para evitar perder tráfico en la página en cuestión.
- **Modelado del PageRank.-** El modelado del Google PageRank es la técnica que permite añadir la etiqueta "no follow" sobre los enlaces, con el objetivo de controlar manualmente el Google PageRank de un sitio web.

- **Contenido duplicado.-** El contenido duplicado es uno de los factores que más perjudica al posicionamiento de una página, por lo que el CMS debería ofrecer alguna utilidad para controlar este aspecto¹⁴.

2.3. Clasificación de los CMS

2.3.1. Gestores de contenido con código abierto

2.3.1.1. Plataformas de gestión de contenidos

- ActionApps (PHP)
- Apache Lenya (Java/XML)
- ASP Nuke (ASP) Un CMS basado en ASP de código libre.
- Blakord Portal (ASP) CMS en ASP con código libre y totalmente en español. Próximamente habrá nueva versión libre, Draco Portal.
- CMS10 (PHP, SWF y Ajax) Gestor de contenidos de nueva generación.
- CMS HYDRAportal
- CMS Contenido (PHP)
- CMSimple Un gestor simple para el mantenimiento rápido de pequeñas webs. Es simple, pequeño y rápido.
- cmsMadeSimple (PHP) CMS fácil de usar y con muchos "add-ons" para añadir.
- door108(PHP) CMS totalmente en español basado en e107. Incluye novedades como sistema de administración MultiTarea y MultiArea para la creación de cientos de sitios sin ocupar casi espacio físico.
- Dédalus (PHP) CMS que pretende ser una revolución en el mundo de los gestores de contenidos, centrándose en la seguridad, mejora de características tradicionales e incorporación de ideas innovadoras.

¹⁴ CMS: <http://martinalia.com/10-caracteristicas-que-todo-CMS> (2009-09-11)

- DotNetNuke (.NET) CMS desarrollado en .NET, gratis y con fuentes.
nota: Más que un CMS en sí es un Framework de .NET pensado para desarrollar CMS entre otras cosas.
- Dragonfly CMS (PHP) Portal que auna en su core: foros, galerías de fotos, descargas y noticias, entre sus modulos más representativos.
- Drupal (PHP) Poderoso CMS muy conocido por la calidad de su código y por la seguridad que brinda, es estable y de actualización continua, configuración sencilla, instalación ágil, importante cantidad de módulos y temas visuales, excepcional documentación y comunidad activa y muy amigable, gran concepto de nodo.
- Dynamicweb CMS (.NET) CMS desarrollado en .NET, solución con más de 60 módulos y una aplicación completa de eCommerce.
- Elgg (PHP) CMS muy completo y fácil de administrar y usar, ideal para usuarios con nuevos
- E107 (PHP) CMS muy completo y fácil de administrar y usar, ideal para usuarios con conocimientos generales acerca de estos sistemas. Sencillo sistema de instalación, amplia selección de temas visuales y módulos, muy flexible, backend muy bien ordenado, dropdownmenu agradable y organizado.
- eZPublish (PHP) CMS framework muy potente que sirve para páginas webs, intranets, comercio electrónico, extranets y portales.
- Gekko (PHP) CMS en español muy seguro, fácil de configurar y altamente escalable.
- Jaws(PHP) Framework y CMS amigable para el usuario y desarrollador.
- Joomla (PHP / MySQL) Versión surgida de Mambo independiente de la empresa que está detrás de Mambo. Instalación muy sencilla y con muchas extensiones y módulos, la documentación es exhaustiva y concisa, interfaz de la administración muy intuitiva y poderosa, backend muy utilizable y editor WYSIWYG, opciones de personalización, una gran comunidad de usuarios.
- Jupiter Content Manager
- Magnolia CMS Edición Comunitaria La Edición Comunitaria de Magnolia es un Sistema de Manejo de Contenidos Empresariales poderoso,

gratuito y fácil de usar. Está disponible bajo una licencia de Código Abierto, la versión 3 GPL. La Edición Comunitaria de Magnolia incluye una interface de navegador web intuitiva creada por AJAX, una interface de programación de aplicaciones o API (del inglés Application Programming Interface) clara y programable por medio de Java y una útil biblioteca personalizada para plantillas fáciles en JSP y Servlets. Puede utilizar cualquier repositorio de contenido JSR-170. Hay también una edición Empresarial la cual no es gratis y tiene soporte de parte del vendedor. Es uno de los pocos CMS open source basado en el estándar JSR-170. Muy fácil e intuitivo de usar, además de ser altamente escalable por su arquitectura de servidores distribuidos. Además de la versión comercial destinada a las empresas, tiene también una versión gratis comunitaria. Al estar basado en PHP y MySQL ofrece gran capacidad de adaptación en múltiples plataformas y flexibilidad para añadir modificaciones.

- Mambo (PHP) CMS muy fácil de usar, pero con posibilidades un poco limitadas.
- MemHT Portal (PHP)
- MODx PHP MODx es un derivado (Fork) de Etomite, resulta ser un CMS más versátil que otros demasiado estructurados.
- NukeET (PHP) CMS totalmente en español basado en el PHP-Nuke.
- Openflavor(PHP) Gestor de contenidos web en castellano.
- OpenCms (Java)
- PHP REGION Ñ (PHP) Un cms al estilo php-nuke pero desarrollado totalmente en español.
- Plone (Zope/Python) Muy flexible y poderoso, excelente interfaz de usuario, instalación muy limpia, buena cantidad de addons, impresionante grado de personalización, integración con LDAP u otros sistemas de login.
- PHP-Nuke (PHP)
- phpwcms (PHP) CMS orientado a la construcción de sitios web para profesionales y empresas.

- POC-CMS (PHP) CMS totalmente desarrollado en español basado en el PHP REGION Ñ
- Postnuke(PHP) Poderoso CMS/Web Framework modular con motor de temas visuales para una interfaz de usuario muy flexible y mantenible, con gran cantidad de módulos para toda necesidad, con un Network Operations Center para soportar una gran comunidad de desarrollo muy activa, y con un código fuente muy limpio y de alta calidad.
- SPIP (PHP) Gestor de Contenido de licencia libre
- Textpattern (PHP)
- Tiki CMS
- TYPO3 (PHP) herramienta CMS con estructura multinivel, motor de búsquedas, gestión de autoría y publicación de contenidos, mecanismo de uso de plantillas para la maquetación de páginas, multilingüaje,... Es también una herramienta portal: administra la personalización de las páginas según la identidad de los usuarios. Es enteramente extensible por módulos. Dispone de una comunidad muy activa.
- TYPOLight (PHP) potente CMS especializado en la accesibilidad. Utiliza XHTML y CSS para generar páginas que cumplen W3C/WAI. Desarrollado por Leo Feyer en 2004 bajo licencia GPL.
- WebGUI (Perl) Ocupa más de 40 MB, flexible, adaptable, multilingüe.
- Xaraya (PHP). Es un CMS bastante potente y general, aunque con una elevada curva de aprendizaje.
- XOOPS (PHP) CMS modular.Instalación sencilla, gran soporte comunitario, gran cantidad de módulos y temas visuales, mucha funcionalidad, sistema de permisos muy bueno.
- Webmaster CMS(PHP)¹⁵

2.3.1.2. CMS para Foros

¹⁵ CMS: http://es.wikipedia.org/wiki/Anexo:Sistemas_de_gesti%C3%B3n_de_contenidos (2009-09-11)

- bbPress (PHP/MySQL) Se integra con WordPress.
- phpBB (PHP/MySQL)
- punBB (PHP/MySQL)
- MyBB (PHP/MySQL)
- SMF (PHP/MySQL)
- Vanilla (Foros) (PHP/MySQL) Sistema de Foros de Lussumo.com, usando AJAX en gran parte del código basan su trabajo en una acepción del concepto "vanilla": software funcional, simple, limpio y bonito (*Short and sweet*). Poco soporte en español pero una gran cantidad de desarrolladores y complementos. Personalizable en extremo¹⁶.

2.3.1.3. CMS para Blogs

- WordPress (PHP/MySQL)
- b2evolution.net (PHP/MySQL)
- pMachine Pro (PHP/MySQL)
- bBlog (PHP)
- Simple PHP Blog (PHP)
- DotClear (PHP/MySQL)
- Serendipity (PHP/MySQL)
- BLOG:CMS (PHP/MySQL)
- Lifetype (PHP/MySQL)
- Webmaster CMS (PHP)
- Plone
- Post Revolution (PHP/MySQL)
- Nucleus CMS (PHP/MySQL)
- Textpattern (PHP)¹⁶

¹⁶ CMS: http://es.wikipedia.org/wiki/Anexo:Sistemas_de_gesti%C3%B3n_de_contenidos (2009-09-11)

2.3.1.4. CMS para galerías

- Gallery (PHP/MySQL)
- plogger (PHP/MySQL)
- coppermine (PHP/MySQL)
- FileBrowser (PHP/MySQL) Sistema de administración de archivos (sobre todo imágenes) de los creadores de Vanilla Lussumo.com y por lo tanto con la misma filosofía de trabajo.
- Pixelpost (PHP/MySQL) gesr de fotologs¹⁷

2.3.1.5. CMS para Galerías de Arte

- PyASC (Python/MySQL) Sistema de administración de contenidos para Galerías de Arte¹⁷

2.3.1.6. CMS para Wikis

- MediaWiki (PHP. Un CMS que permite que todos puedan modificar el contenido)
- TikiWiki (PHP)
- Dokuwiki (PHP)
- PmWiki (PHP)¹⁷

2.3.1.7. CMS para ECommerce

¹⁷ CMS: http://es.wikipedia.org/wiki/Anexo:Sistemas_de_gesti%C3%B3n_de_contenidos (2009-09-11)

- osCommerce (PHP/MySQL)
- Magento (PHP/MySQL)
- PrestaShop (PHP/MySQL)
- Zen Cart¹⁸

2.3.1.8. CMS para groupware

- Webcollab (PHP/MySQL)¹⁸

2.3.2. Gestores de contenidos en código propietario

- **ADSM Portal 2.0** Gestor de contenidos para PYMES de ADSM Solutions. Mediante plantillas, es posible adaptar por completo el sitio Web a las necesidades del cliente. Su sencillo panel de administración permite al cliente editar, añadir y eliminar contenidos. El sistema es totalmente escalable, pudiendo adaptarse a todo tipo de necesidades, desde pequeños sitios Web hasta completos portales de contenidos.
- **Portal Builder CMS** - Un nuevo e innovador gestor de contenidos. Portal Builder CMS es un nuevo producto de tercera generación que se lanzó al mercado a mediados del 2009 por la empresa **SOFTENG** - especialista en diseño y desarrollo de webs usando su gestor de contenidos. Desarrollado en sus etapas iniciales en colaboración con **Microsoft**, cubre necesidades exigentes en empresas medianas y grandes que buscan una solución profesional para gestionar su web y llevar a cabo su estrategia de marketing online, pero a un precio razonable comparado con los costes de licencia e implantación de los productos del cuadrante de Gartner. El video de su sistema de edición es

¹⁸ CMS: http://es.wikipedia.org/wiki/Anexo:Sistemas_de_gesti%C3%B3n_de_contenidos (2009-09-11)

una pequeña muestra de su potencia: **Video de la edición in site** del gestor de contenidos - Portal Builder CMS

- **AgrupaliaSkipper**. SkipperAgrupalia permite el control y administración de los contenidos web de forma fácil y flexible. Sin utilizar plantillas predefinidas, sin necesidad de infraestructura técnica y sin límites en el desarrollo de funcionalidades.
- **GlobalSys** - Gestor de portales web y contenidos
- **Aladetres**, Completo gestor CMS para empresas y administración local sobre tecnología LAMP. Implementación personalizada.
- **AST X-CMS** Un CMS hecho en ASP, almacena los datos en archivos XML, maneja múltiples formatos, foros, blogs.
- **Autoeditable**. Autoeditable está orientado a pequeñas empresas y profesionales que necesiten una web y poder autogestionar sus contenidos de forma fácil y rápida.
- **Hábitat Portal**. Hábitat es un CMS en español, Amigable, Personalizable, Orientado a servicios y Generador de comunidades.
- **Content-SORT**, clasificado como Sistema de Gestión de Portales y de Contenidos Web (CMS), soporta todos los estándares Web W3C y de accesibilidad definidos por la WAI. Orientado originalmente sobre tecnología LAMP, realmente es mutiplataforma (PHP), y se basa en una arquitectura de 3 capas: Bases de Datos, Aplicación, Presentación. Sitio oficial Content-SORT
- **Infodata** de dbysesystems (www.dbyse.com), es un gestor de contenidos avanzado para el mundo editorial, con capacidad para manejar más de 60 tipos distintos de ficheros, provenientes de casi cualquier canal informativo, homogeniza los contenidos en base de datos y los muestra a los clientes a través de una web publicada en la intranet
- **Eximius2 CMS**. Eximius2 CMS es un sistema de gestión de contenidos que permite la administración completa de un portal o sitio web, y es lo suficientemente flexible para crecer junto con cualquier organización desde muy pequeña hasta grandes productores de información. El sistema incluye Modelado de Contenidos, Gestión de Versiones, contenidos Multi-Idioma, Formularios dinámicos, etc.

- **Globalsys**, * El primer gestor de portales y contenidos que se comercializo en España. Actualmente más de 400 empresas lo utilizan.
- [[: **AWM** :AvantSite Web Management]] pertenece a la última generación de Flash CMS's o Flash Content Management Systems, creados para la web 2. Es uno de los pocos del mercado que permite a los web masters crear sitios enteramente flash, con contenidos multimedia embebidos (no emergentes) y gestión de contenidos. También da la posibilidad de que el usuario final tenga su propio site Flash CMS.
- **Civinext Groupware 2.0** Es una plataforma desarrollada exclusivamente para administrar de manera eficiente la gestión de la comunicación interna y externa en una organización. Se caracteriza por integrar diferentes sistemas en uno solo: los sistemas de gestión de contenidos (CMS) multimediales, los sistemas de postulaciones laborales (E-Recruitment), la administración de eventos, la gestión de blogs interactivos, la gestión de sistemas de encuestas y la gestión de empleados e internos.
- **Contendo CMS**. Permite al usuario actualizar la información de su sitio Web de una forma muy rápida y sencilla, sin necesidad de conocimientos técnicos. Desarrollado por Ensitech
- **VRContents** (Perl) Es un CMS desarrollado en Chile por VRWEB orientado a ser flexible y configurable para cada necesidad.
- **Prodigia Easy Site Manager** (Flash 8, PHP, MySQL, AS 2.0) Innovador y funcional Flash CMS. Implementa ingeniería del SW por capas. Permite crear portales, webs corporativas o tiendas virtuales con pasarela de pago 100% Flash. Fácil (cualquier usuario sin conocimientos técnicos lo gestiona), ágil e intuitivo. Gestión WYSIWYG, Gestor de perfiles de usuarios y de usuarios, creación de plantillas, multiidioma, menús multinivel, escalable. Todos los contenidos (video, audio, imágenes, documentos, ficheros, ...) integrados sin ventanas emergentes, flash fpt integrado para subir y gestionar ficheros, gestión de portada, dispone de soporte técnico videoconferencia integrado, gestión de secciones drag&drop, 5 años de desarrollo en continua mejora.

- **Jarimba**, CMS en Java desarrollado por KrugerCorporation <http://www.kruger.com.ec>, Su página oficial es: <http://www.jarimba.com>¹⁹.

2.3.3. Gestores con algún grado de estandarización

Algunos gestores que son compatibles con tecnologías PortletSpecification API JSR-168 y la Content Repository API JSR-170.

- IBM WebSphere
- Novell Extend
- Magnolia CMS, Vignette
- MetaSpace Portal
- JetSpeed
- Liferay
- JBoss Portal
- K-Dimension - Web Content Management System¹⁹

2.4. PHP el lenguaje de programación en CMS´s

2.4.1. Argumento

PHP es un lenguaje de programación interpretado, diseñado originalmente para la creación de páginas web dinámicas. Es usado principalmente en interpretación del lado del servidor (server-side scripting) pero actualmente puede ser utilizado desde una interfaz de línea de comandos o en la

¹⁹ CMS: http://es.wikipedia.org/wiki/Anexo:Sistemas_de_gesti%C3%B3n_de_contenidos (2009-09-11)

creación de otros tipos de programas incluyendo aplicaciones con interfaz gráfica usando las bibliotecas Qt o GTK+.

PHP es un acrónimo recursivo que significa PHP Hypertext Pre-processor (inicialmente PHP Tools, o, Personal Home Page Tools). Fue creado originalmente por RasmusLerdorf en 1994; sin embargo la implementación principal de PHP es producida ahora por The PHP Group y sirve como el estándar de facto para PHP al no haber una especificación formal. Publicado bajo la PHP License, la Free Software Foundation considera esta licencia como software libre²⁰.

2.4.2.Historia

Fue originalmente diseñado en Perl, en base a la escritura de un grupo de CGI binarios escritos en el lenguaje C por el programador danés-canadiense RasmusLerdorf en el año 1994 para mostrar su currículum vitae y guardar ciertos datos, como la cantidad de tráfico que su página web recibía. El 8 de junio de 1995 fue publicado "**P**ersonal **H**ome **P**age Tools" después de que Lerdorf lo combinara con su propio *FormInterpreter* para crear PHP/FI²⁰.

2.4.2.1. PHP 3

Dos programadores israelíes del Technion, ZeevSuraski y AndiGutmans, reescribieron el analizador sintáctico (*parser* en inglés) en el año 1997 y crearon la base del PHP3, cambiando el nombre del lenguaje a la forma actual. Inmediatamente comenzaron experimentaciones públicas de PHP3 y fue publicado oficialmente en junio del 1998.

²⁰ PHP: <http://es.wikipedia.org/wiki/PHP> (2009-09-11)

Para 1999, Suraski y Gutmans reescribieron el código de PHP, produciendo lo que hoy se conoce como motor Zend. También fundaron Zend Technologies en RamatGan, Israel²⁰.

2.4.2.2. PHP 4

En mayo de 2000 PHP 4 fue lanzado bajo el poder del motor ZendEngine 1.0. El día 13 de julio de 2007 se anunció la suspensión del soporte y desarrollo de la versión 4 de PHP, a pesar de lo anunciado se ha liberado una nueva versión con mejoras de seguridad, la 4.4.8 publicada el 13 de enero del 2008 y posteriormente la versión 4.4.9 publicada el 7 de agosto de 2008. Según esta noticia se dará soporte a fallos críticos hasta el 2008-08-08²¹.

2.4.2.3. PHP 5

El 13 de julio de 2004, fue lanzado PHP 5, utilizando el motor ZendEngine 2.0 (o ZendEngine 2). La versión más reciente de PHP es la 5.3.0 (30 de junio de 2009), que incluye todas las ventajas que provee el nuevo ZendEngine 2 como:

- Mejor soporte para la Programación Orientada a Objetos, que en versiones anteriores era extremadamente rudimentario, con PHP Data Objects.
- Mejoras de rendimiento.
- Mejor soporte para MySQL con extensión completamente reescrita.
- Mejor soporte a XML (XPath, DOM, etc.).

- Soporte nativo para SQLite.
- Soporte integrado para SOAP.
- Iteradores de datos.
- Manejo de excepciones²¹.

2.4.2.4. PHP 6

Está previsto el lanzamiento en breve de la rama 6 de PHP. Cuando se lance esta nueva versión quedarán solo dos ramas activas en desarrollo (PHP 5 y 6), pues se abandonó el desarrollo y soporte de PHP 4 el 13 de julio de 2007.

Las diferencias que encontraremos frente a PHP 5.* son:

- Soportará Unicode.
- Limpieza de funcionalidades obsoletas como `register_globals`, `safe_mode`, etc.;
- PECL.
- Mejoras en orientación a objetos²¹.

2.4.3. Páginas CMS PHP

PHP es un lenguaje de script diseñado originalmente para la producción de páginas web dinámicas. PHP es ampliamente utilizado con diferentes lenguajes de scripting que son especialmente adecuados para el desarrollo web, y es combinado con HTML. En general, se ejecuta en un servidor web, que está configurado para tener el código PHP como entrada, y crear contenido de la página web como salida (frontend y back end). Puede ser

²¹ PHP: <http://es.wikipedia.org/wiki/PHP> (2009-09-11)

desplegado en la mayoría de servidores web y en casi cualquier sistema operativo.

Las Ventajas de un CMS (Content managementsystem) su traducción en ingles, auto-manejo de contenido "o" PHP son: Descentralización de mantenimiento por parte de uno solo web master. La edición del contenido, puede ser hecha en cualquier momento y lugar, en cuestión de minutos. Las personas con conocimiento promedio de procesamiento de texto puede crear el contenido directamente. Para la edición principal de la pagina no se necesita conocimientos de HTML, action script o etc. El sistema se puede configurar con restricciones de acceso. A los usuarios que usted indique, se les asignan permisos, que les impiden la edición de contenidos, que no estén autorizados a cambiar. Si se hacen cambios en el contenido, la coherencia de diseño se conserva, dado que el contenido se almacena por separado. La navegación es ajustada y generada automáticamente. Los botones de menús pueden ser generados automáticamente, en cuestión de segundo, a partir de la base de datos de contenido. Ya que el contenido está almacenado en una base de datos, significa que este puede ser reutilizado en muchos lugares en la web, y el formato puede se desplegado en varios dispositivos (navegador web, teléfono móvil / WAP, PDA, impresión). PHP tiene la ventaja de contar con extensiones "plugings" o aplicaciones de contenido dinámico. Entre dichas extensiones o módulos se encuentran: foros, tiendas virtuales, encuestas, solicitudes de compras, búsqueda, gestión de noticias.

Un sistema de autogestión de contenidos da a los usuarios finales la capacidad de modificar fácilmente el contenido de su sitio web sin previa programación o experiencia en diseño web cuando usted lo decida. Hoy en día, ya no es tan necesario el diseño estático, de páginas web HTML, ya que en la mayoría de los casos no se actualizan constantemente. Con un CMS se da mantenimiento a la página en el instante, para que sea flexible y funcional. Si está familiarizado con procesadores de textos, tales como

Word, Pages, Open Office, será capaz de trabajar con un CMS de manera fácil y practica²².

2.4.4. Características del lenguaje PHP para el desarrollo de aplicaciones web

Libre, experimentado apoyo de 1000 comunidades PHP

En realidad, hay probablemente más de 1000 comunidades que le puede proporcionar ayuda en PHP. Hay literalmente cientos de foros dedicados al desarrollo Web con PHP y cientos más que tienen sub-foros de PHP. La ayuda para los principiantes en PHP está literalmente a un post (de foro) de distancia. En general encontrará una respuesta a su problema en un tiempo muy corto. Obtener ayuda de los programadores php con más experiencia es un gran placer, ya que puede eliminar una gran parte de la incertidumbre que acompaña el aprender algo nuevo.

El costo de una aplicación de código abierto no es nada

A diferencia de otros lenguajes de codificación propietarios (en particular aspx de Microsoft) usted no paga un centavo para empezar a programar en PHP. De hecho, no es necesario obtener o descargar ningún software PHP para empezar. Simplemente utilice una herramienta de edición de código fuente y empiece a programar. Suba su código a una cuenta de hospedaje barato y está listo. Usted puede obtener fácilmente editores de código abierto de una variedad de fuentes. Una búsqueda en Google de "editores php" devuelve cientos de posibilidades. A medida que avanza puede descargar paquetes como LAMP o WAMP para que pueda ver php en acción en un entorno de servidor en vivo.

²² PHP: <http://es.wikipedia.org/wiki/PHP> (2009-09-11)

El código abierto tiene mayor seguridad - porque todo el mundo sabe lo que hay en él

De nuevo, a diferencia de los sistemas web propietarios, todo el mundo puede ver de lo que está hecho php. Algunas personas argumentan que esto hace el PHP menos seguro porque todo el mundo conoce que vulnerabilidades específicas de PHP existen. Mi opinión es que la naturaleza abierta de php, literalmente, "forza" a los desarrolladores y la comunidad para mantenerse al corriente de las cuestiones de seguridad y responder a ellas tan pronto como sea posible. Tenga en cuenta sin embargo que es fácil cometer errores de seguridad simple con PHP - y todos los lenguajes sufren de esta característica - sólo tiene que hacer su investigación para mantener sus aplicaciones web seguras.

Existen cientos de ejemplos de código PHP

Además de los foros que ofrecen asesoramiento y experiencia de forma gratuita, también encontrará cientos de ejemplos que le ayudarán a empezar y avanzar como programador PHP. Desde código simple para conectarse a una base de datos, ejemplos más complejos que le permitan recoger datos de un sitio externo o un ejemplo de completo de AJAX y su integración con PHP es lo que hay para que estudie, desarme y construya.

Resultados de las plataformas de código existente

Las plataformas son un desarrollo relativamente reciente que puede ahorrarle un montón de tiempo dolores de cabeza. Las plataformas más simples son como casas prefabricadas. Un equipo de desarrolladores ha combinado sus esfuerzos para resolver el problema de la programación de

tareas repetitivas. Las plataformas permiten poner en marcha aplicaciones web simples y complicadas en cuestión de horas.

Hay decenas de plataformas de PHP para una amplia variedad de tareas. Una vez más una simple búsqueda en Google le proporcionará una gran variedad de plataformas para elegir y explorar.

Interoperabilidad con las aplicaciones Web 2.0

Las modernas aplicaciones web 2.0 son ampliamente caracterizadas por mezclas e interfaces de usuario tipo desktop. PHP es una excelente opción para interactuar con otros sitios web y proporcionar una experiencia de usuario rica. Los comandos simples PHP como curl o fopen permiten adquirir datos de otros sitios web con relativa facilidad.

PHP trabaja bien con JavaScript para que pueda ofrecer a sus usuarios finales modernas interfaces de respuesta que están más allá de las viejas interfaces estáticas del pasado.

Muchos sistemas de gestión de contenidos usan PHP

Si usted decide que quiere ser diseñador web o programador, entonces PHP es una excelente elección. Muchos de los modernos sistemas de gestión de contenidos (CMS) usan PHP uso. Wordpress, Drupal, Joomla - estos paquetes CMS PHP tan populares. Cada CMS por lo general tiene una vibrante comunidad de desarrolladores PHP.

Y, si usted se convierte en experto en uno o más CMS's basados en PHP, entonces tiene una oportunidad muy buena de encontrar trabajo en personalización de paquetes de CMS para clientes o proveedores web.

PHP es escalable a la aplicación web más grande

Algunos de los sitios web más visitados y populares en el planeta usan PHP.

PHP se escalará adecuadamente para cumplir los requisitos más exigentes para los sitios web tales como joomla.org y wordpress.com. Estos sitios web ofrecen una disponibilidad de 7x24 para miles de usuarios simultáneos sin perder el ritmo. Si usted tiene Facebook o MySpace en mente, entonces PHP es un lenguaje de desarrollo definitivamente vale la pena considerar.

La mayoría de empresas de hospedaje web soportan PHP

La mayoría de los proveedores de alojamiento de sitios web soporta PHP. Se trata de un componente estándar de la mayoría de las cuentas de hospedaje basadas en Linux. Dentro de LAMP y WAMP antes la 'P' significa PHP. Si desea que sus aplicaciones se ejecuten en un servidor web basado en Linux (o alguno basado en Windows), entonces PHP es una opción obvia²³.

2.5. Extensiones de los CMS´s

2.5.1. Descripción

Fuera del núcleo de los CMSs, se hace un gran trabajo de gestión para cubrir las necesidades de contenido que permitan hacer los sitios más personalizables. Para mucha gente el auténtico poder de los CMSs reside en aplicaciones para el entorno de trabajo llamadas Extensiones.

Una extensión se usa para agregar una capacitación a los CMSs que, por defecto y en su código base, no están disponibles.

²³ PHP: <http://www.sip.gob.mx/noticias-sobre-lenguajes-aplicaciones-y-herramientas/397-10-razones-para-el-desarrollo-de-aplicaciones-web-con-php> (2009-09-11)

2.5.2. Tipos de Extensiones

Hay cinco tipos de extensiones:

- Componentes
- Módulos
- Plantillas
- Plugins
- Idiomas

2.5.2.1. Componentes

Un componente es el más largo y complejo de los tipos de extensión que hay. Los componentes, son como mini-aplicaciones que renderizan el cuerpo principal de la página. Una analogía sencilla de entender podría ser como si un CMS se tratase de un libro y todos los componentes son capítulos de ese libro. El componente del núcleo para artículos (`com_content`), por ejemplo, es la mini-aplicación del núcleo que maneja la renderización de los artículos así como el componente del núcleo para registros (`com_user`) es la mini-aplicación que maneja el registro de usuarios.

Un componente puede manejar datos, mostrar configuraciones, proveer funciones, y en general puede llevar a cabo cualquier operación no contemplada por las funciones generales del código del núcleo. Los componentes trabajan mano a mano con módulos y plugins, en los CMSs que lo permitan, para proveer de una rica variedad en la funcionalidad y modo de mostrar las cosas. Ellos hacen posible que se pueda transformar completamente un CMS expandiendo ampliamente su capacitación y flexibilidad.

2.5.2.2. Módulos

Una más ligera y flexible extensión usada para la renderización de páginas son los módulos. Los módulos se usan para pequeñas partes de la página y generalmente son menos complejos y capaces de verse a través de los distintos componentes. Si un CMS fuese un libro, un módulo podría verse como un pie de página o bloque del encabezado, o quizá como una imagen/leyenda del libro que puede ser renderizado sobre algunas páginas en particular. Obviamente, puedes tener una nota al pie de página, pero no todas las páginas lo tienen. La nota al pie de página también podrían aparecer recordándote que capítulo estás leyendo o el componente que tienes cargado.

Los módulos son como mini-utilidades que pueden ubicarse en cualquier lugar de tu sitio. Los módulos a veces funcionan en conjunción con algún componente y otras veces son independientes. Los módulos se pueden asignar a una posición de módulo la cual haya sido definida en la plantilla y también dentro de la administración, usando el gestor de módulos y de posiciones disponibles. Por ejemplo, 'left' y 'right' son posiciones comunes para presentaciones a tres columnas.

2.5.2.3. Plugins

Una de las extensiones más avanzadas para un CMS es el Plugin. Un plugin es una sección de código que corre cuando ocurre algún evento predefinido dentro del CMS. Por ejemplo, los editores son plugins que se ejecutan cuando sucede un evento `onGetEditorArea`. Usar un plugin permite al desarrollador decidir la manera en que su código se comporta dependiendo de los plugins que haya instalados para reaccionar a los eventos.

2.5.2.4. Idiomas

Las más básicas y críticas extensiones son un idioma. Un CMS se libera con multitud de idiomas de instalación, pero el sitio y la administración están empaquetados en un solo idioma, idioma de origen del CMS, la mayoría está en inglés. El intentar incluir todos los idiomas disponibles actualmente podría hacer al paquete inmanejable a efectos del peso que alcanzaría para subirlo a un servidor. Los archivos del idioma habilitan todas las interfaces de usuario tanto para el front como para el backend de manera predeterminada en nuestro idioma local preferido.

2.5.2.5. Plantillas

Una plantilla de un CMS es una serie de archivos que controlan la presentación del contenido en el CMS. Una Plantilla no es un Sitio Web, ni tampoco puede ser considerado un diseño completo de un sitio Web. La Plantilla es el diseño básico y fundamental para visualizar tu sitio. Produce el efecto de un "Sitio Web Completo", las plantillas trabajan de la mano con el contenido almacenado en la Base de datos. La Plantilla determina el Estilo CSS de los contenidos, pero no solo eso, inclusive determina el estilo de los enlaces, menús, el path (ruta de navegación), el tamaño y color del texto, entre otros.

2.5.3. Patrón MVC

2.5.3.1. Descripción

Para el diseño de aplicaciones con sofisticados interfaces se utiliza el patrón de diseño Modelo-Vista-Controlador. La lógica de un interfaz de usuario cambia con más frecuencia que los almacenes de datos y la lógica de negocio. Si realizamos un diseño ofuscado, es decir, un pastiche que mezcle los componentes de interfaz y de negocio, entonces la consecuencia será que, cuando necesitemos cambiar el interfaz, tendremos que modificar trabajosamente los componentes de negocio. Mayor trabajo y más riesgo de error.

Se trata de realizar un diseño que desacople la vista del modelo, con la finalidad de mejorar la reusabilidad. De esta forma las modificaciones en las vistas impactan en menor medida en la lógica de negocio o de datos²⁴.

2.5.3.2. Elementos

Elementos del patrón:

- Modelo: datos y reglas de negocio
- Vista: muestra la información del modelo al usuario
- Controlador: gestiona las entradas del usuario

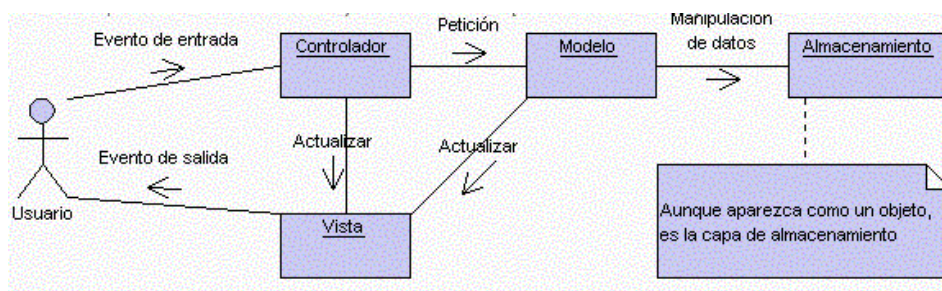


Figura II. 4: Diagrama de Flujo del Patrón MVC²⁵

²⁴Patrón Modelo-Vista-Controlador: <http://www.proactiva-calidad.com/java/patrones/mvc.html> (2009-09-12)

²⁵ Patrón Modelo-Vista-Controlador: <http://www.proactiva-calidad.com/java/patrones/mvc.html> (2009-09-12)

2.5.3.2.1. Modelo

El **modelo** es el responsable de:

- Acceder a la capa de almacenamiento de datos. Lo ideal es que el modelo sea independiente del sistema de almacenamiento.
- Define las reglas de negocio (la funcionalidad del sistema). Un ejemplo de regla puede ser: "Si la mercancía pedida no está en el almacén, consultar el tiempo de entrega estándar del proveedor".
- Lleva un registro de las vistas y controladores del sistema.
- Si estamos ante un modelo activo, notificará a las vistas los cambios que en los datos pueda producir un agente externo (por ejemplo, un fichero bath que actualiza los datos, un temporizador que desencadena una inserción).

2.5.3.2.2. Controlador

El **controlador** es responsable de:

- Recibe los eventos de entrada (un clic, un cambio en un campo de texto, etc.).
- Contiene reglas de gestión de eventos, del tipo "SI Evento Z, entonces Acción W". Estas acciones pueden suponer peticiones al modelo o a las vistas. Una de estas peticiones a las vistas puede ser una llamada al método "Actualizar()". Una petición al modelo puede ser "Obtener_tiempo_de_entrega(nueva_orden_de_venta)".

2.5.3.2.3. Vista

Las vistas son responsables de:

- Recibir los datos del modelo y mostrarlos al usuario.
- Tienen un registro de su controlador asociado (normalmente porque además lo instancia).
- Pueden dar el servicio de "Actualización()", para que sea invocado por el controlador o por el modelo (cuando es un modelo activo que informa de los cambios en los datos producidos por otros agentes)²⁶.

2.5.3.3. Diagrama de Secuencia

Un ejemplo de MVC con un modelo pasivo (aquel que no notifica cambios en los datos) es la navegación web, que responde a las entradas del usuario, pero no detecta los cambios en datos del servidor, ver Figura III. 5.

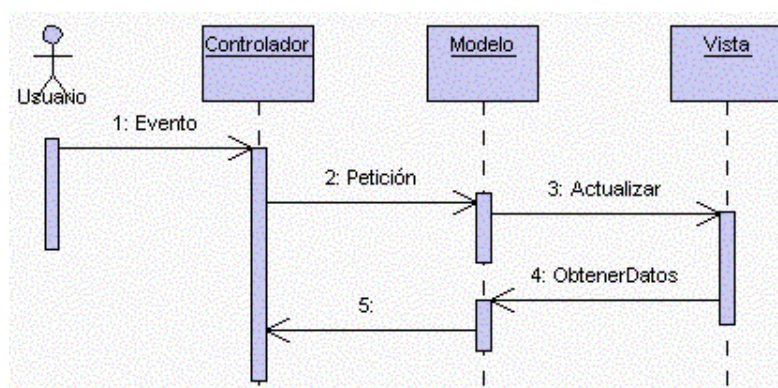


Figura II. 5: Diagrama de secuencia del patrón MVC²⁶

²⁶ Patrón Modelo-Vista-Controlador: <http://www.proactiva-calidad.com/java/patrones/mvc.html> (2009-09-12)

Pasos:

1. El usuario introduce el evento.
2. El Controlador recibe el evento y lo traduce en una petición al Modelo (aunque también puede llamar directamente a la vista).
3. El modelo (si es necesario) llama a la vista para su actualización.
4. Para cumplir con la actualización la Vista puede solicitar datos al Modelo.
5. El Controlador recibe el control.

CAPÍTULO III

ESTUDIO COMPARATIVO DE CMS´S EN EL DESARROLLO DE COMPONENTES

2.1 Delimitación

En el momento existen un gran número de CMS en el mercado, entre gratuitos y comerciales, para el desarrollo de aplicaciones web, todos estos tienen características propias que los definen en su popularidad y uso en el mundo por parte de los desarrolladores. De todas estas herramientas se han elegido 3 CMS por los motivos que se explican a continuación:

- En el sitio web de packtpublishing, <http://www.packtpub.com/award>

Se encuentran los ganadores a mejor CMS del 2009, valorados en diferentes categorías, los CMS ganadores son:

- WordPress
- MODx

- SilverStripe
 - Drupal
 - Joomla
 - ImpressCMS
 - Pixie
 - Plig
 - Plone
 - dotCMS
 - mojoPortal
- En la página web de “aplicaciones empresariales”
<http://www.aplicacionesempresariales.com/eligiendo-el-mejor-cms-open-source-para-tu-empresa.html>

Nos orienta como elegir el mejor CMS para la empresa, dependiendo de las necesidades, en los que destaca a varios CMS´s con sus características más relevantes. Los CMS son los siguientes:

- Typo3
 - EZPublish
 - Alfresco
 - Plone
 - Drupal
 - Joomla
- En la página web de packt pub
<http://www.packtpub.com/open-source-cms-award-previous-winners>

Encontramos los CMS ganadores desde el año 2006 al 2008, en los que encontramos a los siguientes CMS:

Año 2006

- Joomla
- Drupal

- Phone

Año 2007

- Drupal
- Joomla
- CMS Made Simple
- MODx
- TYPOlight
- dotCMS
- e107
- MojoPortal
- Plone
- Silva
- WordPress
- Elgg

AÑO 2008

- Drupal
- Joomla
- DotNetNuke
- SilverStripe
- CMS Made Simple
- ImpressCMS
- Plone
- dotCMS

- En la página web de AFERVE

<http://aferve.com/los-mejores-cms-de-2009/comment-page-1/>

Se desarrolló una encuesta, en la cual se preguntaba al usuario, que CMS consideraba el mejor, bajo varios parámetros de evaluación, los CMS destacados son:

- Drupal
- WordPress

- Joomla
 - DotClear
 - MODx
 - Plone
 - DotNetNuke
- En la revista alemana "PHP magazine", con su link de la página web http://www.phpmag.de/itr/online_artikel/show.php?id=161&nodeid=62

Se realizó un estudio de 40 CMS basadas en php, tras un detallado estudio se eligieron a cinco CMS como los mejores:

- Typo3
- SixCMS
- Aman_RedSYS
- Powerslave
- Astarte

NOTA: la página web del resultado del estudio, se encuentra en idioma alemán, para la traducción al idioma español se utilizó la herramienta traductora de google.

- Encuesta realizada a 30 personas vinculadas en el desarrollo de aplicaciones web. (ver anexo 2)

Conoce el significado de CMS?

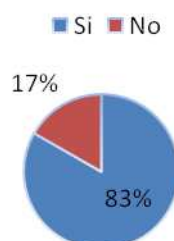


Figura III. 6: Resultado encuesta pregunta n° 1

¿De los CMS listados a continuación cual ha utilizado, o por lo menos ha escuchado?

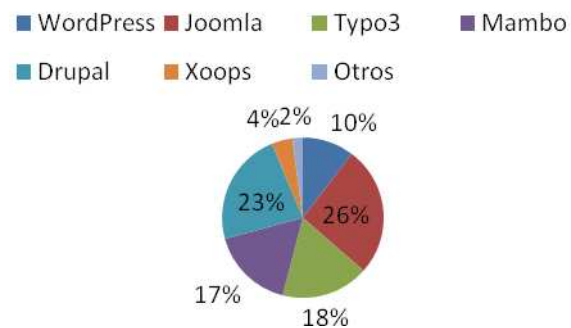


Figura III. 7: Resultado encuesta pregunta n° 2

¿Dentro de los CMS conoce el propósito de un componente?

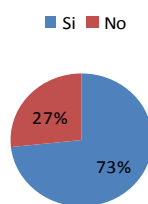


Figura III. 8: Resultado encuesta pregunta n° 3

¿En que CMS ha utilizado, personalizado o desarrollado un componente?

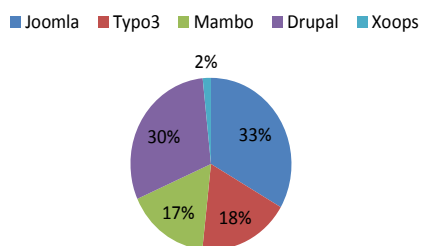


Figura III. 9: Resultado encuesta pregunta n° 4

¿A su criterio cual es el mejor CMS en la actualidad para el desarrollo de componentes?

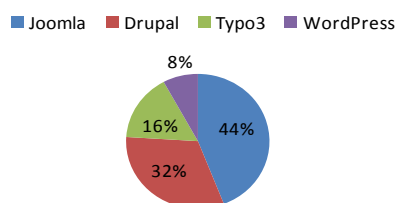


Figura III. 10: Resultado encuesta pregunta n° 5

Después de un detallado análisis de los motivos, se han seleccionado a las herramientas Typo3, Joomla! y Drupal, como los CMS que serán usados en el estudio comparativo, por los siguientes motivos:

- Estos 3 CMS son Open Source, y utilizan como lenguaje programación a PHP, y como gestor de base de datos a MySQL.
- Los CMS escogidos cuentan con crecientes comunidades para el apoyo y soporte de las aplicaciones desarrolladas en cada uno de estos.
- Joomla!, Drupal y Typo3 ofrecen gran flexibilidad para la personalización de las aplicaciones, mediante extensiones desarrolladas por uno mismo o por terceras personas, adaptándose a las necesidades requeridas.
- Los CMS optados son ganadores de premios a los mejores CMS entre sus competidores, destacándose su importancia para el estudio de los mismos.
- Después de la encuesta realizada (ver anexo 2), y el análisis de los resultados, se ha concluido que los CMS preferidos en la actualidad para el desarrollo de las aplicaciones web son: Joomla, Drupal, Typo3.

3.2. Joomla!

3.2.1. Argumento

Joomla! es un sistema de gestor de contenidos dinámicos (CMS) que permite crear sitios web de alta interactividad, profesionalidad y eficiencia. La administración de Joomla! está enteramente basada en la gestión online de contenidos.²⁷

Es gestión online porque todas las acciones que realizan los administradores de sitios Joomla!, ya sea para modificar, agregar o eliminar contenidos se realiza exclusivamente mediante un navegador web (browser) conectado a Internet, es decir, a través del protocolo HTTP (Protocolo de transferencia de hipertexto).

Sólo esto es necesario para que el usuario de Joomla! pueda publicar información en la Red Global, y mantenerla siempre actualizada. Esto convierte a Joomla! en una poderosa herramienta de difusión de Información, de Marketing Online, de negocios por Internet, de administración de proyectos en general, educativos en particular.

Con Joomla! podemos crear sitios web de noticias, sitios corporativos, sitios web de presencia, portales comunitarios, e incluso también puede crearse con Joomla! sistemas que funcionen en redes cerradas (Intranets) para gestionar información interna (comunicaciones, usuarios, etc) de compañías o empresas de negocios. Esto último significa que el ámbito de aplicación de Joomla! no es exclusivo de Internet.

La interfaz administrativa de Joomla! es tan sencilla y amigable, que cualquier persona puede administrar sus propios contenidos web sin la

²⁷GRAF, Hagen. 2008. Building Websites with Joomla! A step by step tutorial to getting your Joomla! CMS website up fast; p 5.

necesidad de poseer conocimientos técnicos, sin saber lenguaje HTML, y sin recurrir a un WebMaster cada vez que hay que actualizar tal o cual cosa en un sitio web. Todos los aspectos técnicos del sistema están ocultos al usuario común, y han sido resueltos por los programadores de Joomla!, haciendo las cosas sencillas para que los administradores y editores de contenidos puedan prescindir casi completamente de una capacitación técnica especializada.

Joomla! es programado en lenguaje PHP (Hypertext Pre Processor) y SQL (StructureQueryLanguage). Utiliza bases de datos relacionales, más específicamente MySQL. Tanto PHP como MySQL son programas OpenSource de libre distribución y uso, y al ser Joomla! una aplicación Web, funciona obviamente en servidores de páginas web (HTTP Servers). Estos servidores de páginas web pueden ser de pruebas (es decir, por ejemplo, Joomla! funcionando en un servidor instalado en la misma máquina del usuario que lo administra), o también servidores comerciales de producción (esto es, cuando publicamos un sitio web basado en Joomla! alojado en una empresa proveedora del servicio hosting).

El funcionamiento de Joomla! se lleva a cabo gracias a sus dos principales elementos:

- La base de datos MySQL: allí es donde se guarda toda la información y la mayor parte de la configuración del sistema, de una forma ordenada y en distintas tablas las cuales cada una de ellas almacena información específica y determinada.
- Los scripts PHP: son los que ejecutan las acciones de consulta y realizan modificaciones en la base de datos convirtiendo los datos en simples páginas web interpretables por los navegadores de Internet (Browsers) y perfectamente inteligibles para los usuarios y administradores.

3.2.2. Características

- Sistema de Administración sencillo y avanzado.

Con una interfaz de administración clara y concreta que le permitirá gestionar todos los aspectos fundamentales clave de un sitio web y sus contenidos.

- Organización del sitio web.

Joomla! está preparado para organizar eficientemente los contenidos de su sitio en secciones y categorías, lo que facilita la navegabilidad para los usuarios y permite crear una estructura sólida, ordenada y sencilla para los administradores. Desde el panel administrador de Joomla! usted podrá crear, editar y borrar las secciones y categorías de su sitio de la manera en que más le convenga.

- Publicación de contenidos.

Con Joomla! CMS se podrá crear páginas ilimitadas y editarlas desde un sencillo editor, basado en WYSIWYG (siglas WhatYouSeeIsWhatYouGet. Estos editores permiten estar viendo en todo momento el texto que se escribe tal y como es mostrado una vez impreso), que permite formatear los textos con los estilos e imágenes deseados. Los contenidos son totalmente editables y modificables.

- Escalabilidad e implementación de nuevas funcionalidades.

Joomla! ofrece la posibilidad de instalar, desinstalar y administrar componentes y módulos, que agregarán servicios de valor a los visitantes de su sitio web, por ejemplo: galerías de imágenes, foros, newsletters, clasificados, etc.

- Internacionalización.

La interfaz de Joomla! se encuentra traducida a muy diversos idiomas.

- Administración de usuarios.

Joomla! le permite almacenar datos de usuarios registrados y también la posibilidad de enviar e-mails masivos a todos los usuarios. La administración de usuarios es jerárquica, pues los distintos grupos de usuarios poseen diferentes niveles de facultades o permisos dentro de la gestión y administración del sitio.

- Diseño y aspecto estético del sitio.
Es posible cambiar todo el aspecto del sitio web tan solo con un par de clics, gracias al sistema gestor de plantillas (templates) que utiliza Joomla!
- Navegación y menús.
La administración de los menús permite tomar decisiones rápidas y sencillas sobre la navegación que queremos en nuestro sitio web.
- Administrador de archivos multimedia.
Joomla! posee una utilidad para subir imágenes, videos, archivos de sonido, al servidor para ser usados en todo el sitio.
- Disposición de módulos modificable.
En un sitio creado con Joomla!, la posición de módulos puede acomodarse como se prefiera.
- Encuestas.
Joomla! posee un sistema de votaciones y encuestas dinámicas con resultados en barras porcentuales.
- Feed de Noticias.
Joomla! trae incorporado un sistema de sindicación de noticias por RSS/XMS de generación automática.
- Publicidad.
Es posible hacer publicidad en el sitio usando el Administrador de Banners.
- Estadísticas de visitas.
Con información de navegador, sistema operativo y detalles de los documentos (páginas) más vistos.
- Automatización en la publicación.
Las páginas y documentos de Joomla! pueden programarse con fecha de publicación y fecha de caducidad. Es decir un documento puede programarse para que se publique automáticamente al llegar una determinada fecha, y luego des publicarse también de forma automática en otra fecha.

- Archivo e historial.
Las páginas viejas o publicaciones que hayan perdido vigencia pueden enviarse a un "archivo" de almacenamiento, sin necesidad de tener que borrarlas. Esto permite también dar la posibilidad a los navegantes de consultar artículos viejos o documentos anteriores en un historial.
- Formatos de lectura.
Cada documento es generado automáticamente por Joomla! en formato PDF, en versión imprimible, y en XML.
- Envío por E-mail.
Los usuarios del sitio Joomla! podrán enviar automáticamente a un amigo por email cada documento publicado.
- Valoración de contenidos.
Los visitantes del sitio podrán votar la calidad de lo publicado.
- Comentarios.
(Opcional) Los usuarios podrán comentar sus opiniones o expresar sus inquietudes en la misma página de contenidos.

3.2.3. Versiones

- 16 de septiembre de 2005: Joomla! 1.0.0
- 21 de septiembre de 2005: Joomla! 1.0.1
- 2 de octubre de 2005: Joomla! 1.0.2
- 14 de octubre de 2005: Joomla! 1.0.3
- 21 de noviembre de 2005: Joomla! 1.0.4
- 24 de diciembre de 2005: Joomla! 1.0.5
- 15 de enero de 2006: Joomla! 1.0.6
- 15 de enero de 2006: Joomla! 1.0.7
- 26 de febrero de 2006: Joomla! 1.0.8
- 5 de junio de 2006: Joomla! 1.0.9
- 25 de junio de 2006: Joomla! 1.0.10
- 29 de agosto de 2006: Joomla! 1.0.11
- 12 de octubre de 2006: Joomla! 1.5 [BETA] released.

- 12 de diciembre de 2006: Joomla! 1.0.12
- 21 de julio de 2007: Joomla! 1.0.13.
- 1 de septiembre de 2007: Joomla! 1.5 [RC2]
- 4 de octubre de 2007: Joomla! 1.5 [RC3]
- 18 de diciembre de 2007: Joomla! 1.5 [RC4]
- 14 de enero de 2008: Joomla! 1.0.14 RC1.
- 22 de enero de 2008: Joomla! 1.5 [Stable].
- 8 de febrero de 2008: Joomla! 1.5.1 [Stable].
- 23 de marzo de 2008: Joomla! 1.5.2 [Stable].
- 23 de abril de 2008: Joomla! 1.5.3 [Stable].
- 6 de julio de 2008: Joomla! 1.5.4 [Stable].
- 28 de julio de 2008: Joomla! 1.5.5 [Stable].
- 12 de agosto de 2008: Joomla! 1.5.6 [Stable].
- 9 de septiembre de 2008: Joomla! 1.5.7 [Stable].
- 10 de noviembre de 2008: Joomla! 1.5.8 [Stable].
- 12 de diciembre de 2008: Joomla! 1.5.9 [Stable].
- 28 de marzo de 2009: Joomla! 1.5.10 [Stable].
- 3 de junio de 2009: Joomla! 1.5.11 [Stable].
- 1 de julio de 2009: Joomla! 1.5.12 [Stable].
- 22 de julio de 2009: Joomla! 1.5.13 [Stable].
- 30 de julio de 2009: Joomla! 1.5.14 [Stable].
- 4 de noviembre de 2009: Joomla! 1.5.15 [Stable].²⁸

3.2.4. Historia del proyecto

Joomla surge como el resultado de una bifurcación o mejora de Mambo, de la corporación Miro de Australia, quien mantenía la marca del nombre Mambo en esa época y el grupo principal de desarrolladores. Joomla nace con esta división el 17 de agosto de 2005. La corporación Miro formó una organización sin ánimo de lucro con el propósito inicial de fundar el proyecto

²⁸ Joomla: http://es.wikipedia.org/wiki/Joomla!#Historia_del_proyecto (2009-10-2)

y protegerlo de pleitos. El grupo de desarrollo reclamó que muchas de las cláusulas de la estructura de la fundación fueron acuerdos previos hechos por el comité directivo de Mambo, el que no tiene la consultoría necesaria de quienes mantienen el proyecto e incluye cláusulas que violan los valores principales del código abierto. El grupo de desarrollo creó un sitio Web que se llamó OpenSourceMatters para distribuir información a los usuarios, desarrolladores, diseñadores Web y a la comunidad en general. En ese momento el líder Andrew Eddie, conocido como "MasterChief", escribió una carta abierta para la comunidad, que apareció en la sección de anuncios del foro público en mambo-server.com. Al siguiente día, 1000 personas ingresaron al sitio web opensourcematters.org para expresar su apoyo y estímulo por las acciones ejecutadas por el grupo de desarrollo. El sitio web recibió un aviso de temporalmente fuera de servicio debido al excesivo tráfico. Este evento apareció en newsforge.com, e-week.com, y ZDnet.com. Peter Lamont CEO de Miro dio una respuesta en el artículo titulado "The Mambo Open Source Controversy - 20 Questions with Miro". Durante ese periodo de tiempo, Joomla se hizo más grande.

El 1 de septiembre de 2005, el nuevo nombre, "Joomla", que es la pronunciación en inglés de la palabra Jumla que significa "todos juntos" o "en su conjunto".²⁹

3.2.5. Estructura Organizativa/Asociativa

La gestión de administración principal del proyecto esta delegada al grupo principal ("CoreTeam"). Todos los miembros de este grupo trabajan en conjunto como un solo grupo, comprometidos para guiar a Joomla! dentro del movimiento de código abierto. Este grupo está compuesto por diferentes perfiles, con variadas experiencias y una serie diversa de disciplinas.

²⁹Joomla: http://es.wikipedia.org/wiki/Joomla!#Historia_del_proyecto(2009-10-2)

Este grupo nace cuando se experimentó el surgimiento de Joomla en el 2005. El grupo principal es mucho más que una congregación de desarrolladores, su responsabilidad principal radica en la organización con respecto Joomla en su estructura funcional como organización y no únicamente en la programación del sistema de gestión de contenidos.

El proyecto se constituye de varios grupos que se han creado para enriquecer el conocimiento que la comunidad Joomla proporciona. Cada uno de los grupos se centra en un aspecto específico de Joomla! que es importante para la expansión y desarrollo, el grupo principal no puede estar en cada discusión de estos temas, por ello existe un líder y un cabecilla alterno en cada uno de los grupos que se encargan de comunicarse de forma directa con el Grupo Principal, ver Figura III. 11.

Los grupos de trabajo, suministran un canal de comunicación esencial entre la gran comunidad de Joomla! y el grupo principal de forma que traen inquietudes a la luz, mediación de cambios y extensión de información.³⁰



Figura III. 11: Estructura Organizativa de Joomla!³⁰

³⁰Joomla: http://es.wikipedia.org/wiki/Joomla!#Historia_del_proyecto(2009-10-2)

3.2.6. Estado actual

De forma similar a otros proyectos, Joomla mantiene dos versiones de la aplicación: una estable y otra Beta o en desarrollo. La versión estable es la que se considera para usuarios y a medida que aparecen errores se corrigen, esta versión se publica sin nuevas funcionalidades. La versión Beta, incluye nuevas funcionalidades y mejoras a los fallos reportados en versiones anteriores, de igual forma se tiene que en este tipo de versión se reflejan las directivas del proyecto para usuarios avanzados y desarrolladores.

El 22 de enero de 2008 se ha lanzado la versión 1.5 estable de Joomla, que incorpora notables mejoras en el área de seguridad, administración y cumplimiento con estándares W3C. Actualmente cuenta con una gran cantidad de componentes, módulos y plugins, aunque se pueden usar los de la versión anterior (1.0.X). Únicamente hay que activar un plugin incluido llamado System - Legacy que mejora considerablemente (no por completo) la compatibilidad con los mambots para Joomla! 1.0.X.

Hoy en día es el CMS más utilizado del mundo, el más seguro y con mayor crecimiento que cualquier otro.

3.2.7. Plataformas compatibles

- **En GNU/Linux:** Después de bajar el archivo de distribución en un directorio bajo el sitio del servidor web, ejecute la descompresión porque el archivo generalmente viene en formato .zip. Este programa no necesita recompilarse porque se basa en php que es un lenguaje interpretado. Por ejemplo: si el root de su sitio web es /var/www/html es posible crear un directorio que se llame /var/www/html/joomla donde quedarán todos los guiones del programa.

- **En Windows:** De forma similar a GNU/Linux, es necesario descomprimir el archivo dentro de un directorio en la raíz del servidor web por ejemplo si apache está instalado en c:\apache, el archivo se debe crear en c:\apache\htdocs\joomla.

Se asume que el usuario ha trabajado con el servidor Apache y el gestor de bases de datos MYSQL, por ello es necesario que antes de iniciar el proceso de instalación, tanto para Windows como para GNU/Linux se haya creado una base de datos en mysql con el correspondiente usuario y permisos; el siguiente paso a ejecutar es por medio del navegador, se debe entrar al directorio que se ha creado, siguiendo con nuestro ejemplo se debe ingresar al url, http://su_sitio.com/joomla de forma que se correrá un asistente que lo guiará hasta el final de la instalación.

- **En Mac OSX:** Se puede hacer de dos maneras. O bien activando el ordenador como servidor Apache mediante la función "compartir web" en Preferencias del Sistema, o instalando una aplicación denominada MAMP que a su vez instala MySQL, PHP 5 y Apache 2. De esta manera el contenido del archivo .zip de Joomla! debe ser colocado en la carpeta: Aplicaciones/MAMP/htdocs/ y luego iniciar la instalación.

3.2.8. Desarrollo de componentes

3.2.8.1. Prefacio

Desde la versión 1.5 de Joomla! Incluye la posibilidad de incorporar un patrón de diseño MVC en la elaboración de componentes, que a la larga facilitará el mantenimiento de nuestro componente, y lo hará más fácil de

extender al mismo tiempo que resultará más fácil de leer por otras personas.

El patrón de diseño MVC se lleva utilizando mucho tiempo en el ámbito del desarrollo web en marcos de trabajo como Jakartastruts de apache (java), Java Server Faces de Sun (java), Symphony (php), etc.

El marco de trabajo de Joomla! proporciona una implementación sin archivos de configuración, al contrario de lo que hacen otros marcos de trabajo como Struts o JSF. Esto puede ser una ventaja o una desventaja según como se mire. Joomla no proporciona este archivo de configuración, sin embargo tiene otra forma de mapear las acciones del controlador. Joomla le da vital importancia al nombre que se tiene que dar a cada fichero del componente, es importantísimo nombrar bien el controlador, las vistas y los modelos del componente, de lo contrario Joomla no sabrá encontrarlos.

Nuestro componente tendrá dos partes diferenciadas: el front-end y el back-end.

- Front-end: el front-end es la parte del componente que es visible para el usuario de nuestro sitio web. Se visualiza donde hayamos incluido la etiqueta `<jdoc:includetype="component" />` en el template. Su código fuente se encuentra en el directorio components dentro del directorio principal de Joomla!.
- Back-end: es la parte del componente que se mostrará cuando entremos al sitio como administradores. El componente se mostrará donde lo hayamos especificado en el template del back-end. El template para la parte de back-end y la parte de front-end es diferente. Su código fuente se encuentra en el directorio components dentro del directorio administrator dentro del directorio principal de Joomla!.

En la parte de back-end se realizara labores de mantenimiento y configuración para la aplicación. Después, en la parte de front-end se

obtendrá los parámetros que previamente hemos configurado en la parte de back-end.³¹

3.2.8.2. Elaboración

Al iniciar la elaboración del componente, Joomla impone la norma de: los componentes se deben de ubicar en un directorio cuyo nombre empiece por *com_* seguido del nombre del componente, por ejemplo para el componente prueba seria *com_prueba*. Este componente se debe crearlo dentro de la capeta *components*, ubicada en la carpeta principal de Joomla! en nuestro servidor web.

Al momento de llamar al componente, lo primero que hace Joomla es buscar el archivo php que hay dentro, con el mismo nombre que el componente, y lo ejecuta. Este es el punto de entrada del componente donde residirá una instancia del componente. Se establecerá el fichero *prueba.php* dentro del directorio del componente.

Siguiendo con el patrón MVC, a continuación crearemos el controlador de nuestro componente, *controller.php*, dentro del directorio. En el mismo directorio del componente se creará un directorio con nombre *models* y otro con nombre *views*.

Dentro del directorio *views*, hay que crear un directorio por cada vista y el directorio tiene que tener el nombre de la vista; se debe crear el directorio *prueba* dentro del directorio *views*.

Dentro del directorio *views/prueba* se debe crear el directorio *tmpl*, aquí es donde se guardarán los layouts de la vista. El layout es el fichero final que se mostrará por el navegador.

³¹Joomla: <http://www.nosolocodigo.com/tutorial-componentes-para-joomla-15-ii>(2009-10-2)

Se creará el fichero *views/prueba/view.html.php*, este fichero es muy necesario. Ahora se debe crear el fichero *prueba.php* dentro de *models*, el cual corresponde a nuestro modelo según el patrón MVC.³²

Así quedará la estructura básica de un componente (el componente prueba), usando el patrón MVC en Joomla!. Ver Figura III.12.

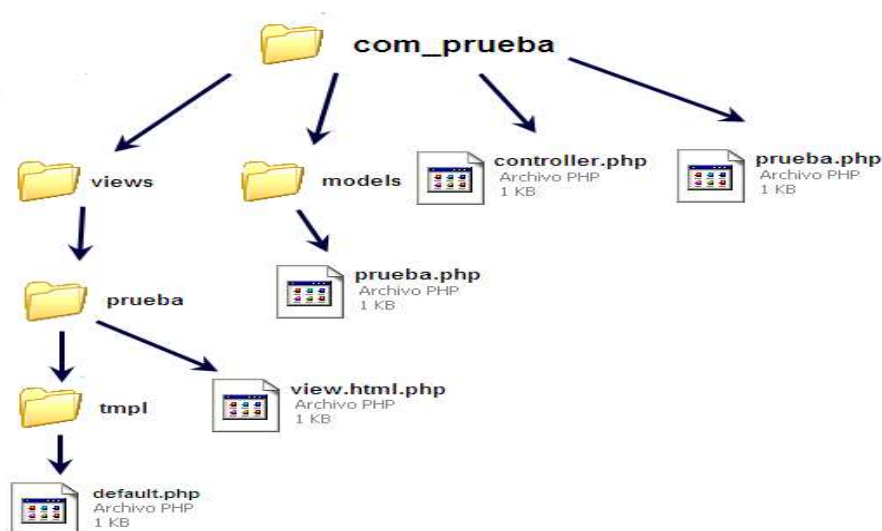


Figura III. 12: Estructura del componente Prueba

3.3. Drupal

3.3.1. Argumento

"Drupal es el equivalente fonético en inglés a la palabra neerlandesa³³ druppel que significa gota".

³²Joomla: <http://www.nosolocodigo.com/tutorial-componentes-para-joomla-15-ii>(2009-10-2)

³³neerlandés: Es lengua oficial en los Países Bajos y Bélgica

Drupal es un sistema de gestión de contenido modular multipropósito y muy configurable que permite publicar artículos, imágenes, u otros archivos y servicios añadidos como foros, encuestas, votaciones, blogs y administración de usuarios y permisos. Drupal es un sistema dinámico: en lugar de almacenar sus contenidos en archivos estáticos en el sistema de ficheros del servidor de forma fija, el contenido textual de las páginas y otras configuraciones son almacenados en una base de datos y se editan utilizando un entorno Web.³⁴

Es un programa de código abierto, con licencia GNU/GPL, escrito en PHP, desarrollado y mantenido por una activa comunidad de usuarios. Destaca por la calidad de su código y de las páginas generadas, el respeto de los estándares de la web, y un énfasis especial en la usabilidad y consistencia de todo el sistema

El diseño de Drupal es especialmente idóneo para construir y gestionar comunidades en Internet. No obstante, su flexibilidad y adaptabilidad, así como la gran cantidad de módulos adicionales disponibles, hace que sea adecuado para realizar muchos tipos diferentes de sitio web.

3.3.2. Características

Ayuda on-line.- Un robusto sistema de ayuda online y páginas de ayuda para los módulos del 'núcleo', tanto para usuarios como para administradores.

Búsqueda.- Todo el contenido en Drupal es totalmente indexado en tiempo real y se puede consultar en cualquier momento.

Código abierto.- El código fuente de Drupal está libremente disponible bajo los términos de la licencia GNU/GPL. Al contrario que otros sistemas de

³⁴Drupal: <http://es.wikipedia.org/wiki/Drupal>(2009-10-11)

'blogs' o de gestión de contenido propietarios, es posible extender o adaptar Drupal según las necesidades.

Módulos.- La comunidad de Drupal ha contribuido muchos módulos que proporcionan funcionalidades como 'página de categorías', autenticación mediante jabber, mensajes privados, bookmarks, etc.

Personalización.- Un robusto entorno de personalización está implementado en el núcleo de Drupal. Tanto el contenido como la presentación pueden ser individualizados de acuerdo las preferencias definidas por el usuario.

URL's amigables.- Drupal usa el mod_rewrite de Apache para crear URLs que son manejables por los usuarios y los motores de búsqueda.

Autenticación de usuarios.- Los usuarios se pueden registrar e iniciar sesión de forma local o utilizando un sistema de autenticación externo como Jabber, Blogger, LiveJournal o otro sitio Drupal. Para su uso en una intranet, Drupal se puede integrar con un servidor LDAP.

Permisos basados en roles.- Los administradores de Drupal no tienen que establecer permisos para cada usuario. En lugar de eso, pueden asignar permisos a un 'rol' y agrupar los usuarios por roles.

Control de versiones.- El sistema de control de versiones de Drupal permite seguir y auditar totalmente las sucesivas actualizaciones del contenido: qué se ha cambiado, la hora y la fecha, quién lo ha cambiado, y más. También permite mantener comentarios sobre los sucesivos cambios o deshacer los cambios recuperando una versión anterior.

Enlaces permanentes (Permalinks).- Todo el contenido creado en Drupal tiene un enlace permanente asociado a él para que pueda ser enlazado externamente sin temor de que el enlace falle en el futuro.

Objetos de Contenido (Nodos).- El contenido creado en Drupal es, funcionalmente, un objeto (Nodo). Esto permite un tratamiento uniforme de

la información, como una misma cola de moderación para envíos de diferentes tipos, promocionar cualquiera de estos objetos a la página principal o permitir comentarios -o no- sobre cada objeto.

Plantillas (Templates).- El sistema de temas de Drupal separa el contenido de la presentación permitiendo controlar o cambiar fácilmente el aspecto del sitio web. Se pueden crear plantillas con HTML y/o con PHP.

Sindicación del contenido.- Drupal exporta el contenido en formato RDF/RSS para ser utilizado por otros sitios web. Esto permite que cualquiera con un 'Agregador de Noticias', tal como NetNewsWire o Radio UserLand visualice el contenido publicado en la web desde el escritorio.

Agregador de noticias.- Drupal incluye un potente Agregador de Noticias para leer y publicar enlaces a noticias de otros sitios web. Incorpora un sistema de cache en la base de datos, con temporización configurable.

Soporte de Blogger API.- La API de Blogger permite que un sitio Drupal sea actualizado utilizando diversas herramientas, que puedan ser 'herramientas web' o 'herramientas de escritorio' que proporcionen un entorno de edición más manejable.

Independencia de la base de datos.- Aunque la mayor parte de las instalaciones de Drupal utilizan MySQL, existen otras opciones. Drupal incorpora una 'capa de abstracción de base de datos' que actualmente está implementada y mantenida para MySQL y PostgreSQL, aunque permite incorporar fácilmente soporte para otras bases de datos.

Multiplataforma.- Drupal ha sido diseñado desde el principio para ser multi-plataforma. Puede funcionar con Apache o Microsoft IIS como servidor web y en sistemas como Linux, BSD, Solaris, Windows y Mac OS X. Por otro lado, al estar implementado en PHP, es totalmente portable.

Múltiples idiomas y Localización.- Drupal está pensado para una audiencia internacional y proporciona opciones para crear un portal multilingüe. Todo el texto puede ser fácilmente traducido utilizando una

interfaz web, importando traducciones existentes o integrando otras herramientas de traducción como GNU ettext.

Administración vía Web.- La administración y configuración del sistema se puede realizar enteramente con un navegador y no precisa de ningún software adicional.

Análisis, Seguimiento y Estadísticas.- Drupal puede mostrar en las páginas web de administración informes sobre referrals (enlaces entrantes), popularidad del contenido, o de cómo los usuarios navegan por el sitio.

Registros e Informes Toda la actividad y los sucesos del sistema son capturados en un 'registro de eventos', que puede ser visualizado por un administrador.

Comentarios enlazados.- Drupal proporciona un potente modelo de comentarios enlazados que posibilita seguir y participar fácilmente en la discusión sobre el comentario publicado. Los comentarios son jerárquicos, como en un grupo de noticias o un foro.

Encuestas.- Drupal incluye un módulo que permite a los administradores y/o usuarios crear encuestas on-line totalmente configurables.

Foros de discusión.- Drupal incorpora foros de discusión para crear sitios comunitarios vivos y dinámicos.

Libro Colaborativo.- Esta característica es única de Drupal y permite crear un proyecto o "libro" a ser escrito y que otros usuarios contribuyan contenido. El contenido se organiza en páginas cómodamente navegables.

Control de congestión.- Drupal incorpora un mecanismo de control de congestión que permite habilitar y deshabilitar determinados módulos o bloques dependiendo de la carga del servidor. Este mecanismo es totalmente configurable y ajustable.

Sistema de Cache.- El mecanismo de cache elimina consultas a la base de datos incrementando el rendimiento y reduciendo la carga del servidor.

3.3.3. Versiones

Drupal hace su lanzamiento un primero de enero del 2001 con la versión 4

Drupal 4.X

Drupal 5.X

Drupal 6.X

En la actualidad la se utiliza la versión 6.19.

3.3.4. Historia del Proyecto

Drupal fue originalmente escrito por Dries Buytaert y es el software usado para impulsar por ejemplo a los sitios web DebianPlanet, Spread Firefox y KernelTrap, la fecha de su lanzamiento fue el primero de enero del 2001.

A pesar de que empezó como un pequeño BBS, Drupal ha llegado a ser mucho más que sólo un portal de noticias gracias a su arquitectura flexible. Drupal se compone de una infraestructura base y un conjunto de módulos que ofrecen un amplio conjunto de funciones, incluyendo sistemas de comercio electrónico, galerías de fotos, administración de listas de correo electrónico, e integración de CVS. Es posible añadir módulos de terceros para modificar el comportamiento de Drupal u ofrecer nuevas funciones.

Drupal se usa, entre otros, en intranets de compañías, enseñanza en línea, comunidades de arte y administración de proyectos. Muchos piensan que la relevancia de Drupal en las comunidades de usuarios es lo que lo hace destacarse de la competencia.

3.3.5. Estructura organizativa/asociativa

La estructura de Drupal se presenta en los equipos de drupal.org o llamados también Drupal.org Teams.

Administradores de Manuales (HandbookMaintainers).- encargados de los manuales y la documentación.

Administradores del sitio (AdministraSiteMaintainers).- encargados de la administración del sitio con tareas como la eliminación de Spam.

Web Masters.- encargados de la gestión del diseño. Del sistema del sitio CMS.

Administración de usuarios (UserAdmins).

Equipo de Infraestructura (InfraestructureTeam).

Administradores del Núcleo (CoreMaintainers).

Equipo de Seguridad (Security Team).- encargados de niveles de seguridad con claves, parches, etc³⁵

3.3.6. Estado Actual

La última versión estable hasta el momento es la 6.19.

³⁵Drupal: <http://drupal.org.es/caracteristicas>(2009-10-11)

3.3.7. Plataformas compatibles

Independencia de la base de datos Aunque la mayor parte de las instalaciones de Drupal utilizan MySQL, existen otras opciones. Drupal incorpora una 'capa de abstracción de base de datos' que actualmente está implementada y mantenida para MySQL y PostgreSQL, aunque permite incorporar fácilmente soporte para otras bases de datos.

Multiplataforma Drupal ha sido diseñado desde el principio para ser multiplataforma. Puede funcionar con Apache o Microsoft IIS como servidor web y en sistemas como Linux, BSD, Solaris, Windows y Mac OS X. Por otro lado, al estar implementado en PHP, es totalmente portable.

Múltiples idiomas y Localización Drupal está pensado para una audiencia internacional y proporciona opciones para crear un portal multilingüe. Todo el texto puede ser fácilmente traducido utilizando una interfaz web, importando traducciones existentes o integrando otras herramientas de traducción como GNU ettext.

3.3.8. Desarrollo de componentes (Módulos)

Para desarrollar un módulo en Drupal se crea tres archivos que van a estar ubicados dentro de drupal/sites/all/modules, estos tres archivos tendrán el mismo nombre del módulo pero con diferentes extensiones: .info, .install y .module.

En el archivo.info se encuentra la información del módulo, aquí se especifica el nombre con el que el módulo va a ser visualizado en la plantilla.

En el archivo `.install` se encuentra la información de la creación de tablas en la base de datos y algunas configuraciones iniciales para el funcionamiento del módulo.

El archivo `.module` Esta compuesto por dos grandes grupos de funciones. Los `hooks` y los `themes`. Los `hooks` son funciones que permiten la comunicación con todo el núcleo de Drupal, pueden ser eventos u operaciones con campos, usuarios, etc, los `themes` son funciones que permiten formatear la salida de las variables que se trabajan en los `hooks`.

“Drupal no se ajusta al patrón MVC”³⁶

Los módulos en Drupal son ficheros con extensión `.module` que contienen funciones escritas en PHP. Estas funciones actúan como *hooks* (funciones gancho), que son llamadas por Drupal durante sus procesos habituales de gestión de contenido y construcción de páginas web. Por ejemplo, cada vez que un nodo es creado, visualizado, modificado o borrado, Drupal llama a una de estos *hooks* pasándoles el contenido del nodo. De esta forma los módulos tienen la posibilidad de modificar y adaptar la información a visualizar en las páginas web antes de que se mande definitivamente al navegador.

Los *hooks* no sólo son llamados cuando se gestionan los nodos, también se llaman en muchas otras circunstancias. Por ejemplo, cuando se comprueba si el usuario tiene acceso a la información que está intentando consultar, cuando se empieza a crear o está a punto de terminarse una página web, cuando se están creando o modificando comentarios, cuando se están creando los menús, y así un largo etcétera.

Todas estas llamadas que realiza Drupal a los módulos posibilita que el sistema sea altamente "maleable", logrando que se pueda modificar prácticamente el funcionamiento entero del sistema sin tener que alterar en

³⁶Drupal: <http://www.slideshare.net/develCuy/el-incompredido-drupal>(2009-10-11)

ningún momento el núcleo (*core*) del mismo. De hecho, muchas de las características básicas de Drupal, como puede ser la posibilidad de organizar el contenido en categorías, son tratadas en la práctica como módulos.

La mayoría de los módulos en Drupal se instalan simplemente copiándolos en el directorio *modules*. Y aunque normalmente sólo se componen de un fichero *.module*, algunas veces pueden venir acompañados de archivos auxiliares, como hojas de estilo CSS por ejemplo.

Actualmente existe un número realmente grande de módulos disponibles. De temática diversa y de gran calidad. Antes de decidirse por un módulo u otro se deben revisar los requerimientos de los mismos. La mayoría de los módulos sólo piden que se instalen en una versión determinada de Drupal, pero otros pueden exigir la presencia de algún componente externo a Drupal, como la posibilidad de ejecutar código Perl, o quizás una librería de terceros. Incluso algunos requieren la modificación del esquema de la base de datos, mediante la adición de columnas a las tablas ya existentes o la creación de tablas nuevas.³⁷

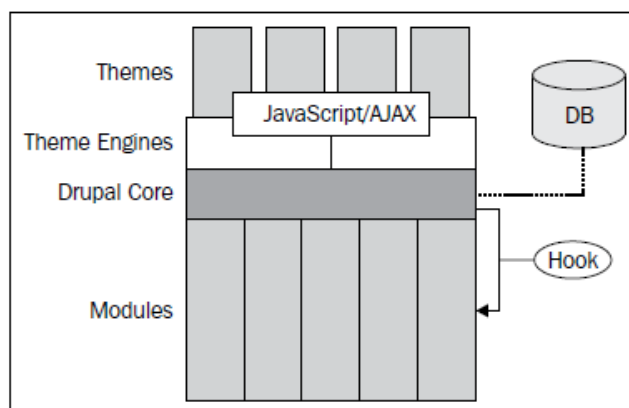


Figura III. 13: Diagrama de Funcionamiento de Drupal

³⁷Drupal: <http://www.inmensia.com/articulos/drupal/modulos.html>(2009-10-11)

3.4. Typo 3

3.4.1. Argumento

TYPO3 es un software libre de portal y gestión de contenidos, open-source, el nivel de la empresa del sistema de gestión de contenido fue creado por KasperSkårhøj (un desarrollador de Dinamarca) en el año 2000, con la participación de una comunidad muy activa que se desarrolló, en primer lugar en los países nórdicos y germánicos y después en Francia, Estados Unidos y Canadá. Se trata de una aplicación basada en Web que puede ejecutar en un servidor web de nuestra elección. Las características que conforman TYPO3 son modularizadas, separando todas las secciones principales y funciones.³⁸

Es una herramienta sobre todo de gestión de contenido muy completa. Permite realizar enteramente un sitio web de contenidos, con todo lo que eso implica: estructura multinivel, motor de búsquedas, gestión de autoría y publicación de contenidos, mecanismo de uso de plantillas para la maquetación de páginas, etc. Pero es también un portal. Administra, en particular, la personalización de las páginas según la identidad de los usuarios, es decir sabe integrar una selección de contenidos en una misma página, según los derechos del usuario identificado.

Es enteramente extensible por módulos, pueden ser módulos de gestión en el interfaz de administración - gestión básica MySQL, gestión de servidor LDAP, o módulos orientados al usuario, incluidos en el sitio- encuestas, foros, calendario, noticias, búsquedas.

Realizar un portal con Typo3 permite mezclar estrechamente contenido y módulos de personalización, integración de información y módulos externos,

³⁸Typo3: <http://es.wikipedia.org/wiki/TYPO3>(2009-10-11)

administrando al mismo tiempo de una única manera los derechos de edición y publicación.

3.4.2. Características

- Con TYPO3 se puede, rápida y fácilmente, crear y gestionar el contenido de nuestro sitio web.

Nuevas secciones pueden ser añadidas rápidamente y fácilmente cambiadas. Diferentes personas pueden tener permisos para editar las distintas secciones mejorando así la productividad y optimizando la colaboración en un entorno easy-to-use web-based (facilidad de uso basado en web).

- Podemos controlar que otros usuarios back-end pueden y no pueden tener acceso.

De modo que los editores de contenido sólo tienen acceso a las herramientas de edición de contenidos, y pueden utilizar estas herramientas sólo en las secciones de contenido que se les permite editar. Los cambios de contenido están versionadas, por lo que podemos ver qué cambios se han hecho, y volver a una versión anterior si es necesario.

- Podemos tener los borradores de las páginas.

Así que podremos trabajar en la mejora de una página o modificar sus detalles, y tener el contenido listo para cuando es necesario subirlo en el sitio web (cuando este puede bloquearse para sustituir el contenido existente, a un clic de botón). Además de las distintas versiones del contenido, TYPO3 también nos permite gestionar las distintas traducciones de contenido, proporcionando a nuestros visitantes un sitio multilingüe (siempre y cuando tengamos los editores que puede traducir nuestros contenidos).

- TYPO3 nos permite agregar rápidamente todo un conjunto de nuevas características a nuestro sitio web.

Utilizando extensiones a través de una interfaz conocida como el administrador de extensiones. Esto nos permite añadir características como los libros de visitas, foros, encuestas de voto o en el tecleo de un botón, sin necesidad de instalar y administrar otra aplicación de software.

- El contenido puede ser gestionado fácilmente.
A través editores de texto enriquecido y la intuitiva interfaz de usuario de TYPO3.
- Se puede administrar varios sitios web y dominios.
A través de la misma instalación de TYPO3. Esto es muy útil si tenemos un sitio de pequeñas empresas con diferentes dominios para diferentes productos o servicios.
- Si estamos tratando de modificar la misma sección o contenido que alguien más está editando, TYPO3 nos advierte de ello y nos previene de corromper el contenido que no se ha presentado todavía.

Todo esto puede estar sucediendo mientras que los visitantes siguen viendo la web. Ver Figura III.14.

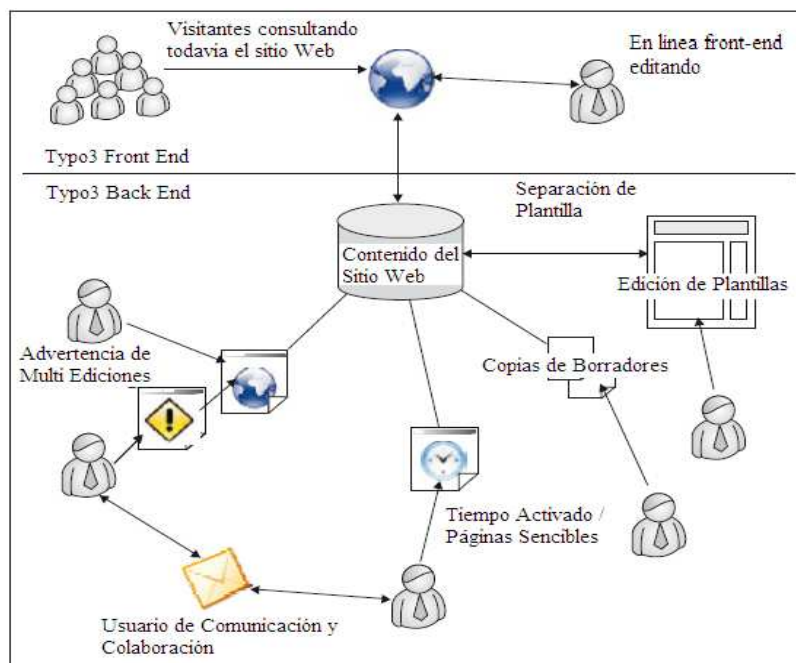


Figura III. 14: Visitas al Sitio Web

3.4.3. Versiones

Año 1998: versión 1.0

Año 1998: versión 1.0.1

Año 1998: versión 1.2.0

Año 1998: versión 1.3.0

Año 2000: version 2.0

24 de Mayo de 2002: versión 3.0

7 de Abril de 2006: versión 4.0

3.4.4. Historia del Proyecto

Typo3 fue uno de los primeros CMS en comenzar a desarrollarse. El proyecto se inicia en Dinamarca por parte de KasperSkårhøj a finales de 1997, en un momento en el cual el concepto "Gestor de Contenidos" era aún desconocido. En dichas circunstancias, los primeros pasos consistieron en identificar las necesidades de los usuarios, lo que dio lugar a tres versiones prototipo en 1998, cada una más avanzada que la anterior. En el verano de ese mismo año, el desarrollo de Typo3 se encomendó a la agencia superfish.com (esta empresa parece haber desaparecido en la actualidad, pues su dominio se encuentra vacío). Sin embargo, aproximadamente un año más tarde, y considerando que superfish.com estaba tomando una dirección distinta a la buscada, KasperSkårhøj llega a un acuerdo para desvincularse de dicha empresa y decide continuar el camino en solitario.

En Agosto de 2000 se hace pública la primera versión para evaluación (versión *beta*) de Typo3, dándose a conocer por primera vez al mundo del software libre. A finales de ese mismo año, Typo3 empezaba a contar con

una comunidad emergente, que comenzaba a suministrar nuevas e interesantes ideas para responder a las necesidades de los usuarios. Es por ello que se suele considerar que el nacimiento "oficial" del proyecto tiene lugar en estos momentos.

El año 2001 es muy importante para Typo3. En Abril se celebró en Suiza un taller de una semana de duración dirigido por KasperSkårhøj. Además, en Septiembre, una prestigiosa revista alemana para profesionales de las tecnologías de la información, denominada Ix, publicó un artículo con una comparativa entre Typo3 y Zope (otro conocido CMS), en la cual Typo3 salía globalmente beneficiado. Esto atrajo importantes reconocimientos y confirmó a Typo3 como uno de los CMS basados en PHP más importantes.

El 24 de Mayo de 2002 ve la luz la versión 3.0, la primera versión estable de esta herramienta. Sólo cinco días más tarde, otra reconocida publicación alemana denominada PHP Magazin publica un artículo donde se comparan 40 CMS basados en PHP. Tras un detallado estudio se seleccionan cinco plataformas, entre las cuales se encuentra Typo3, compitiendo en prestaciones con las otras cuatro herramientas elegidas, todas ellas comerciales.³⁹

3.4.5. Estructura Organizativa/Asociativa

El equipo central de TYPO3 se dedica a desarrollar y mantener la parte central de nuestro favorito CMS. Además, hay equipos sub para cada una de las ramas principales (V4 y V5), y equipos de proyecto de trabajo sobre las principales características de versiones específicas.

³⁹Typo3: <http://www.proyectoparque.net/index.php?id=87&L=2>(2009-10-11)

El equipo R&D (Research and Development) investigación y desarrollo, organiza la investigación y proyectos de desarrollo. Su objetivo principal es hacer valer un desarrollo centrado y coherente a largo plazo.⁴⁰

En detalle su trabajo consiste en estas tareas principales:

- Supervisa y coordina el desarrollo de TYPO3 y FLOW3
- Se toma la decisión final sobre las cuestiones estratégicas o controvertidas planteadas en el equipo principal.
- Es compatible con los proyectos de desarrollo, proporcionando los conocimientos y contactos.
- Apoya y supervisa los equipos impulsados por la comunidad centrada en la investigación y el desarrollo.
- Se da consejo en lo que respecta a la codificación de las directrices y la hoja de ruta.

ECT - esta es la forma corta de (ExtensionsCoordinationTeam) Equipo de Coordinación de Extensión.

La idea de la ECT es ayudar a todos los desarrolladores de extensión con los siguientes objetivos:

- Buena calidad en el código
- Buenos estándares e interfaces
- Agradable en el uso, características base (manejo de formularios, manejo de traducciones)

⁴⁰Typo3: <http://typo3.org/teams> (2009-10-11)

- Poner ideas juntas
- El uso de equipos de efectos de sinergia y conceptos más globales

Y existen otros equipos más, con la tarea de realizar su trabajo de forma óptima, para contribuir con la comunidad de Typo3. Ver Figura III.115.

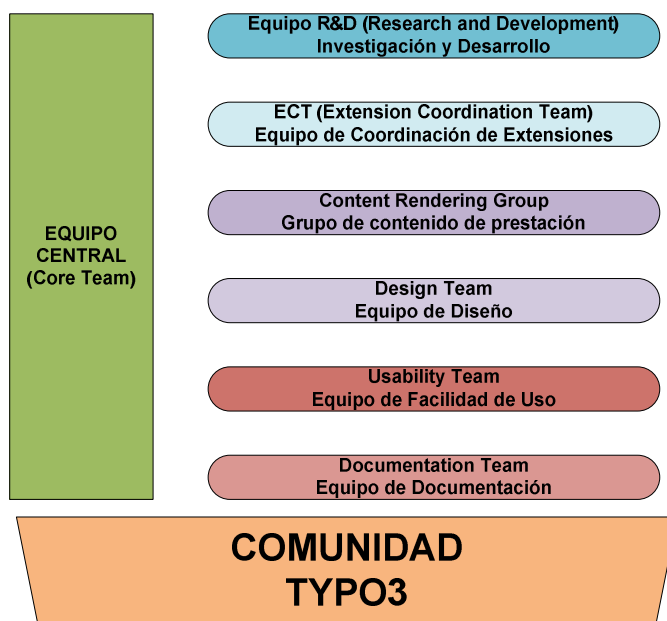


Figura III. 15: Estructura Organizativa de Typo3

3.4.6. Estado Actual

La última versión estable de este CMS es la 4.0, disponible desde el 7 de Abril de 2006. Ha supuesto un salto cualitativo importante, para dar respuesta al espectacular crecimiento del proyecto en los últimos años y a las nuevas necesidades que los usuarios están demandando. Desde entonces han aparecido las subversiones 4.0.1 y 4.0.2.

Actualmente el país que concentra la mayor parte de la actividad alrededor de Typo3es Alemania, por ser allí donde se encuentra el mayor número de

grupos de usuarios que contribuyen al proyecto, aunque también se trabaja para el mismo desde otros países, como Dinamarca, Suiza o Austria.

Typo3 está implantado en más de 122.000 servidores por todo el mundo. En su sitio oficial se tiene constancia de 5.739 sitios web que han adoptado este entorno hasta Octubre de 2009. No obstante, se sospecha que el número total puede ser hasta diez veces mayor. Un total de 1.133 consultorías en todo el mundo prestan asistencia y asesoramiento sobre Typo3, de las cuales trece se localizan en España. Ha sido traducido a 43 idiomas y está en constante desarrollo en más de 60 países por una comunidad de, aproximadamente, 27.000 personas.

Entre las referencias más conocidas de Typo3 están famosas empresas y entidades como BASF, EDS, Konica-Minolta, Daimler-Chrysler, Volkswagen, The New York Times, UNESCO y numerosas universidades y administraciones públicas. En Estados Unidos, Typo3 está protagonizando en la actualidad un auténtico "boom" en las universidades. Centros como la Universidad de Stanford, la Universidad John Hopkins y la Universidad de Rhode Island, entre otros, han migrado su sitio web a este CMS de Software Libre. Estos datos muestran claramente el impacto que ha tenido y está teniendo esta herramienta a nivel internacional, hasta el punto de que se ha convertido en una referencia y está siendo utilizada por prestigiosas organizaciones e instituciones en todos los sectores de actividad.⁴¹

3.4.7. Tecnología

- Basado en navegador

TYPO3 no requiere ningún software especial en el lado del usuario. Cualquier navegador actual con soporte de gráficos es válido. Se recomienda MS Internet Explorer 5+.

⁴¹Typo3: <http://www.proyectoparque.net/index.php?id=87&L=2>(2009-10-11)

- PHP4 / MySQL

Está basado en el muy popular y rápido lenguaje de script PHP4, combinado con MySQL. De modo que es posible realizar una instalación básica de TYPO3 por prácticamente cualquier ISP.

- Procesamiento de imágenes opcional

Prestaciones como el procesamiento de imágenes requiere de software adicional, pero generalmente disponible en el servidor como Gdlib, Freetype e ImageMagick.

- Código base encapsulado

Líneas maestras de proyecto muy exigentes, políticas de nombres y una completa visión de las variables globales del sistema proporciona un claro código de partida. Todo el código importante está encapsulado en clases fácilmente extendibles por un programador PHP.

- Base de datos relacional

El contenido se almacena en una base de datos relacional muy rápida, como MySQL, PostgreSQL y Oracle.

Se clasifica, por tanto, como software LAMP (Linux, Apache, MySQL y PHP), pero también se lo puede usar con la plataforma Windows.

3.4.8. Seguridad

Pueden configurarse descargas seguras de archivos. El envío y almacenamiento de contraseñas al panel de control trasero se encripta con md5. Para intranets, extranets y áreas de administración puede aplicarse filtrado IP. La notificación de login multi estado mediante correo electrónico

es una eficaz alarma antirrobo. Para la transmisión de datos encriptados se puede usar SSL.

3.4.9. Soporte

- Software de servidor

TYPO3 funciona óptimamente bajo el servidor web Apache, usando el motor de scripts PHP 4 y MySQL como motor de base de datos. Todos estos componentes están disponibles en varias plataformas (Linux, Windows, FreeBSD, etc.) y, por consiguiente, hacen de TYPO3 un CMS realmente multiplataforma.

No obstante, la plataforma de funcionamiento más común por su robustez y rendimiento es Linux.

- Documentación abundante

Está disponible gran cantidad de información detallada, incluyendo vídeos multimedia que cubren todos los niveles del producto, en particular los puntos técnicos más importantes.

- Comunidad de desarrollo mundial

Gran comunidad de desarrollo, muy activa y repartida por todo el mundo.

- Consultores profesionales

Consultores que trabajan localmente y red de asociados.

3.4.10. Desarrollo de componentes

El dominio se traduce en el software de objetos - el llamado "domainmodel (modelo de dominio)" - durante este mutuo proceso de aprendizaje. Los objetos que no pertenecen al modelo de dominio son todos aquellos que son responsables para el ahorro, la búsqueda, validación, de filtrado y visualización de datos. Estos son sacados de la ampliación casi por completo y de origen a Extbase.

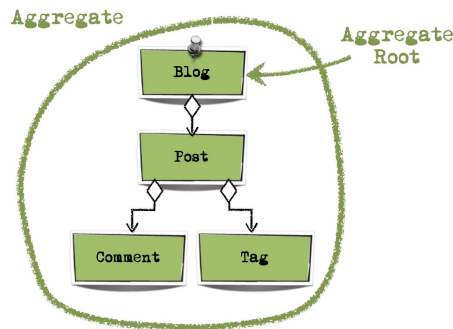


Figura III. 16: Modelo de dominio de la extensión de ejemplo "BlogExample"

Ciclo en una extensión/componente con TYPO3

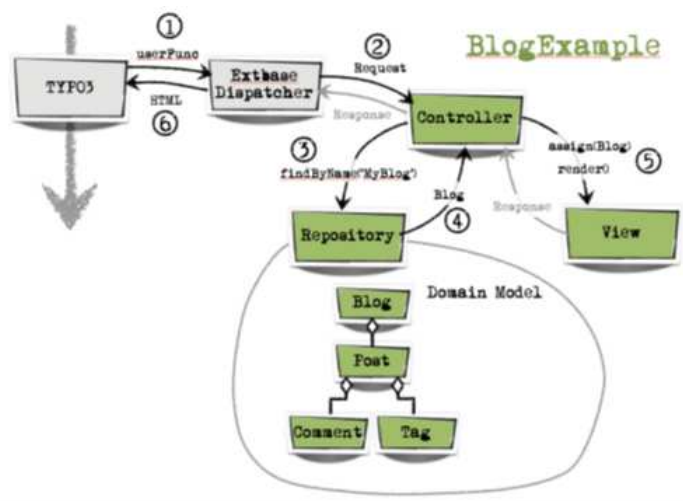


Figura III. 17: Ciclo simplificado a través de una extensión/componente

1. TYPO3 procesa y el contenido de la página "se encuentra" en el plugin de la página. No llama a la extensión directamente sino que pasa el control al dispatcher "de Extbase. El dispatcher decide la combinación de controlador y de la acción que debería hacerse cargo de la acción basada en la configuración transmitida por TYPO3 y los parámetros `$_GET` y `$_POST`.

2. El despachador pasa toda la información que sea necesaria para la el control del proceso en el controlador responsable. El controlador internamente ejecuta la acción ("Acción") en la forma de un método.
3. El repositorio responsable es solicitada en el método de acción. Esto ocurre con la ayuda del método "find".
4. El repositorio ofrece un objeto de dominio completamente construida. Esto es una instancia de la clase "blog" en nuestro ejemplo. Este objeto de blog ya contiene todos los asociados-post, comentario y objetos tag (a fin de calmar a todos los expertos; carga lenta será apoyado en el futuro).
5. El método de acción asigna el objeto blog para el responsable objeto de vista (asignar las solicitudes ()) y la creación de los contenidos en el formato deseado (render ()).
6. Ver devuelve el contenido creado en su respuesta a la operadora, y el despachador se lo pasa a la TYPO3 superior, proceso de renderizado.⁴²

3.5. Parámetros de Medición

3.5.1. Introducción

Hoy en día existen muchas opciones de gestores de contenido o CMS (Content Management Systems) en el mercado, es importante conocerlas para ampliar las posibilidades de trabajo. Cada una de las opciones cuenta con características similares y otras más amplias según las necesidades de tu proyecto.

⁴²Typo3: http://typo3.org/extensions/repository/view/doc_extbase/current/(2009-10-11)

Dentro de los CMS, existen elementos de suma importancia como los componentes, que son aplicaciones que crean funcionalidades extras, que se integran dentro del CMS. En cualquier momento, estos componentes pueden ser instalados directamente, personalizados o desarrollados desde cero para ser integrados perfectamente en la web.

Hay que tener en cuenta que no se encuentra en la actualidad algún tipo de estándar para medir todas las ventajas de los componentes, es por esto que iremos determinando los parámetros a partir de una investigación y los diferentes criterios por parte de usuarios y desarrolladores, a la vez que iremos proponiendo nuevos, para valorar de una manera más real el desarrollo de componentes.

3.5.2. ¿Cómo evaluar un componente en un CMS?

Para el análisis de un componente nos plantearemos una serie de preguntas que nos ayudaran a evaluarlo.

¿Se aprecia claramente el objetivo del componente?, ¿Está claro que información contiene dicho componente?

Se valora que el usuario/desarrollador pueda conocer de forma rápida y sencilla que tipo de información contiene el componente.

¿Es sencillo de entender?, ¿Se distingue claramente el objetivo de todos los elementos?

Se valora la sencillez, resulta muy útil que un usuario/desarrollador comprenda fácilmente los elementos que contiene el componente y sepa antes de hacer algún cambio la funcionalidad de cada elemento.

¿Existe algún tipo de categorización o de etiquetado?, ¿Resulta útil el etiquetado/categorización para componentes?, ¿Se muestran las etiquetas/categorías más populares?

La categorización de componentes crea una estructura jerárquica, ya que esta estructura queda definida por las categorías. El etiquetado es más flexible. Es el usuario/desarrollador quien etiqueta la información.

Hay que tener en cuenta también el nivel de accesibilidad.

La compartición de información entre diferentes aplicaciones es uno de los retos actuales. Los componentes ayudan en este sentido permitiendo que otros programas puedan consumir la información que contiene un determinado componente.

3.5.3. Parámetros de evaluación

Para establecer los parámetros que vamos a utilizar en la evaluación de los sistemas de gestión de contenidos con respecto al desarrollo de componentes en primera instancia será el de revisar la información disponible para cada CMS estudiado en la presente investigación para llegar a entender su funcionalidad.

Estos parámetros de evaluación serán utilizados en el análisis del desarrollo y la administración de un componente para cada uno de los CMS: Joomla!, Drupal y TYPO3 considerando todas las funcionalidades comunes y agrupando varios factores críticos, estos criterios funcionales son:

- Requerimientos del sistema
 - Soporte
 - Seguridad
 - Plataforma

- Facilidad de uso
 - Desarrollo de interfaz de usuario
 - Acceso a datos
 - Flexibilidad

- Gestión de extensiones dentro del CMS
 - Interfaz para la gestión
 - Instalación / desinstalación de extensiones
 - Soporte multilinguaje

- Estructura de las extensiones
 - Uso del esquema MVC (modelo, vista, controlador)
 - Organización de los archivos de la extensión
 - Localización específica de una parte de la extensión.

Al desarrollar el mismo componente en cada uno de los CMS estamos en condiciones de realizar un análisis cualitativo con respecto a estos criterios funcionales, teniendo en cuenta que para nuestra investigación necesitaremos datos cuantitativos por lo que en posteriores capítulos daremos una valoración numérica a los mismos. Cada uno de los parámetros de evaluación citados será desglosado en un conjunto de características que permitirán ser más específicos en el análisis y engloban su definición.

3.5.3.1. Requerimientos del sistema

Es conveniente que la definición de requerimientos del sistema por parte de los diferentes CMS, sea analizado por el programador, ya que facilitará el desarrollo y mantenimiento de los sistemas web a implementar, tomando en cuenta todas las prestaciones de cada uno de ellos además de que hablamos de código libre.

Aquí algunas características que tomaremos en consideración:

3.5.3.1.1. Soporte

El soporte para desarrollo de componentes se puede encontrar en varios lugares. El mejor lugar es empezar por el sitio oficial de cada sistema de administración de contenidos. Aquí se puede hallar la información que regularmente se va publicando y actualizando. Los miembros de las comunidades de cada CMS, publican regularmente en sus blogs sobre temas varios, como técnicas de programación, seguridad, eventos, etc. Existe la opción de suscripción, la cual permitirá conocer las nuevas publicaciones.

3.5.3.1.2. Seguridad

Cada vez se valora más la seguridad de un producto a la hora de tomar una elección para seleccionar en qué software se implementará una aplicación. En el mundo de los CMS, por su vocación de producto accesible desde internet, resulta aún más importante valorar cuál ofrece más garantías de no sufrir un desfase.

3.5.3.1.3. Plataforma

Igual que cualquier programa, todo CMS necesitará de un sistema operativo instalado previamente en el ordenador, vamos a tomar en cuenta que otros programas más necesitan para su funcionamiento y si en cada sitio web oficial existe el instalador para cada plataforma sea esta Windows o Linux.

3.5.3.2. Facilidad de uso

En la presente evaluación se va considerar también el factor facilidad de uso, para lo cual se utilizará varios métodos como la observación, cuestionarios y entrevistas, para conocer el nivel de satisfacción de los usuarios finales del componente desarrollado.

El control y personalización de componentes requiere mucha dedicación, organización, planificación, sin querer decir con esto que su desarrollo sea una tarea imposible.

Se recomiendo también el uso de la "Lógica" para entender cómo funcionan y sobre todo como reaccionan ante problemas o cambios.

3.5.3.2.1. Desarrollo de interfaz de usuario

La elaboración de interfaces de usuario es un aspecto primordial al momento del desarrollo de componentes, ya que es el medio de comunicación entre el usuario final y la aplicación, esta interacción debe ser creada de la forma más sencilla, llamativa y rápida.

3.5.3.2.2. Acceso de datos

El acceso a datos es un ámbito esencial dentro de los componentes de un cms, porque permite gestionar información almacenada en bases de datos, a través del uso de ingresos, consultas, modificaciones y eliminaciones, los mismos que deben ejecutarse de una forma que sea lo más rápida y segura posible.

3.5.3.2.3. Flexibilidad

La flexibilidad hace referencia a la amplitud con que pueden ser utilizados los componentes en diferentes aplicaciones web para distintas necesidades. Esta característica permite la reutilización total del componente, o parte del mismo con la necesidad o no de ampliar su funcionamiento para características propias del sitio web a aplicarse, se debe considerar el lenguaje de programación y la base de datos a utilizar en el CMS escogido.

3.5.3.3. Gestión de componentes dentro del CMS

Como desarrolladores de sitios web al utilizar sistemas de administración de contenidos, estamos en la responsabilidad del mantenimiento diario del mismos, específicamente refiriéndonos a la administración de componentes las actividades podrían incluir la instalación y desinstalación de los mismos, búsquedas de componentes específicos, habilitación o des habilitación de acuerdo a los requerimientos del sitio web, entre otros.

3.5.3.3.1. Interfaz para la gestión

Otro aspecto muy importante a ser tomado en cuenta debe ser el entorno de trabajo el momento de desarrollar específicamente la interfaz para la gestión. Un administrador debe tener la capacidad de personalizar los elementos de contenidos que se pueden ver y utilizar dentro de la interfaz, dejándole un control total o bien limitando dentro de las opciones disponibles para simplificar el trabajo de los diferentes tipos de usuarios.

3.5.3.3.2. Instalación / desinstalación de extensiones

La instalación y desinstalación de extensiones debe ser una de las principales fortalezas de los CMS's. Debido a la rapidez con que se desarrollan nuevas extensiones es muy importante la facilidad de este parámetro ya que con esto se dota al proyecto de una flexibilidad que le permite incorporar nuevas funcionalidades día a día

3.5.3.3.3. Soporte multilinguaje

Una aplicación informática, se dice que esta es multilinguaje cuando su Interfaz de usuario puede ser mostrada a elección del usuario en cualquiera de diferentes idiomas, en nuestro caso particular en el idioma español, otro punto importante ya que la mayoría de extensiones se las desarrolla en inglés por lo que valoramos si los CMS 's poseen la función o el soporte para traducción de los mismos.

3.5.3.4. Estructura de las extensiones

La estructura de las extensiones es muy importante porque estas son la parte fundamental de cada CMS, al conocer su distribución y organización se puede extender su funcionalidad, interactividad, operatividad y optimizar los servicios web además de hacer crecer los sitios desarrollados con cada CMS hacia una escalabilidad sin límites.

3.5.3.4.1. Uso del esquema MVC (modelo, vista, controlador)

El modelo de desarrollo más avanzado y actual de Software es el MVC (Modelo Vista Controlador). Su estructura/arquitectura está separada en 3 capas:

- **Modelo:** Almacenamiento y relación de datos (Mysql),
- **Controlador:** lógica funcional y estructuras de control del sistema (PHP), y
- **Vista:** Presentación del aspecto estético (XHTML, CSS, imágenes y multimedia).

3.5.3.4.2. Distribución de los archivos de la extensión

Cada uno de los CMS's presenta una organización diferente de los archivos que se van creando para el desarrollo de un componente, la ordenada y correcta ubicación de estos es lo que nos permitirá realizar correcciones o mejoras de una forma más rápida, los archivos de cada extensión deben estar organizados jerárquicamente.

3.5.3.4.3. Localización específica de una parte de la extensión

Otro aspecto importante en el desarrollo de extensiones es saber en qué parte del archivo de la misma se encuentra tal o cual función, con la utilización del esquema MVC esta tarea se nos hace mucho más fácil.

3.6. Descripción de los módulos de prueba

3.6.1. Módulo 1

El módulo 1 será elaborado para probar, dentro del parámetro de facilidad de uso, el indicador del desarrollo de interfaces de usuario.

El módulo consta de un formulario para el ingreso de artículos de librería, el cual consta de varios campos simples, campos de listas, campos de

selección, y otros elementos de interfaz de usuario, como estilos de visualización, imágenes. A través de la construcción de este módulo se podrá evaluar la facilidad que brinda cada uno de los componentes en su respectivo CMS, analizados para el diseño de interfaces de usuario.

3.6.2. Módulo 2

El módulo 2 será elaborado para probar, dentro del parámetro de facilidad de uso, el indicador de acceso a datos.

Este módulo constará de un formulario en el que se listan todos los artículos de librería, previamente ingresados, el mismo que nos permitirá realizar una evaluación de la forma en que los distintos componentes en su respectivo CMS trabajan con el acceso a los datos almacenados en una base de datos MySQL, y cómo son desplegados los mismos.

3.6.3. Módulo 3

El módulo 3 será elaborado para probar, dentro del parámetro de facilidad de uso, el indicador de flexibilidad.

Este módulo hará uso de los formularios del módulo 1 y del módulo 2, adicionalmente constará nuevas tablas para especificar las categorías y sus categorías de un artículo de librería, así como sus respectivos formularios de ingreso en el back end y visualización de los datos en el frontend.

Además se desarrollara un formulario para el ingreso de los datos de los proveedores de los artículos, un formulario para listarlos a todos ellos con sus respectivos artículos abastecidos, este modulo nos permitirá probar que tan flexible puede ser cada componente con su CMS, al re utilizarlo y expandirlo para cumplir con nuevos requerimientos del sitio web.

3.6.4. Módulo 4

El módulo 4 será elaborado para probar el parámetro de gestión de componentes dentro del CMS.

Este módulo constara de un componente básico y común existente en los tres CMS, el cual lo vamos a añadir a nuestro sitio web, el mismo nos servirá para encontrar las facilidades que nos brinda cada CMS en la administración de los componentes que se están haciendo uso en el sitio web.

3.7. Desarrollo de los Módulos de Prueba

3.7.1. Desarrollo de los Módulos en la Herramienta Joomla! 1.5

3.7.1.1. Módulo 1

Para la construcción del módulo 1 vamos a crear un componente en el Back-End llamado `com_libreria`, dentro del directorio `..\administrator\components`, con sus respectivos directorios posibilitando el uso del patrón MVC (modelo, vista, controlador), ver figura III. 18.

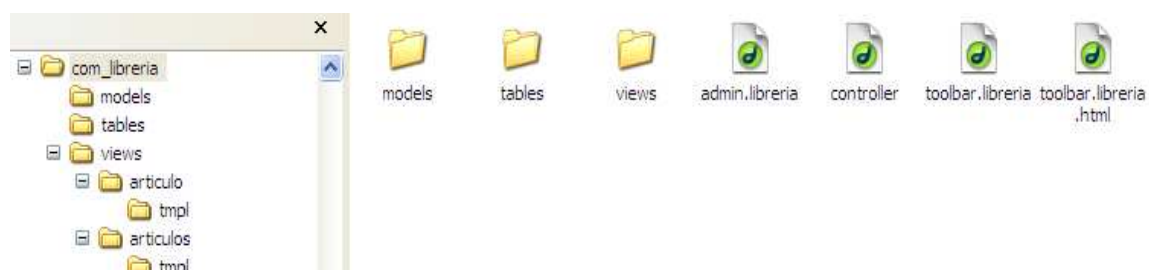
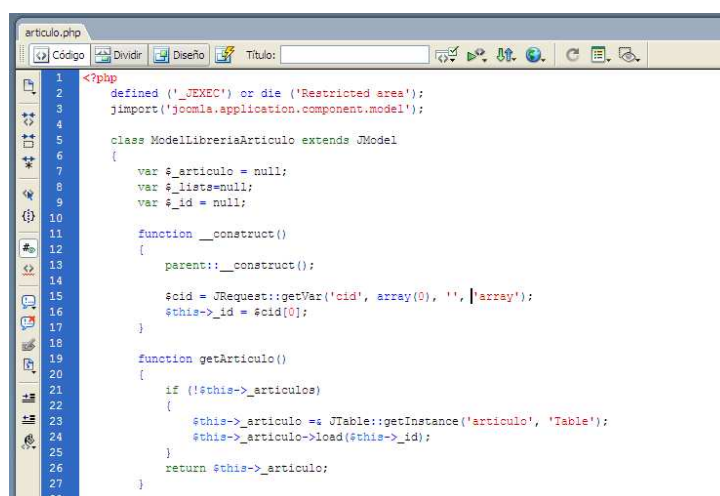


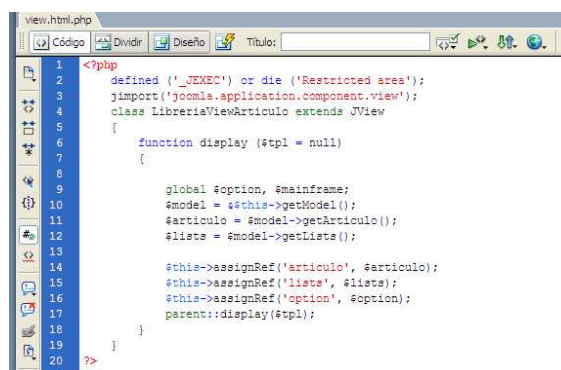
Figura III. 18: Creación de los directorios y archivos .php necesarios, para el desarrollo del componente `com_libreria` – Módulo 1 Herramienta Joomla! 1.5

Creación de la interfaz de usuario, donde se usa 3 archivos, primero dentro de la carpeta models, se crea el archivo articulo.php, con el código necesario para gestionar los datos de ingreso de artículo, segundo dentro de la carpeta views crea el directorio articulo y luego crea el archivo view.html.php con las asignaciones necesarias para enviar los datos a pantalla, tercero dentro de la carpeta views\articulo\tmpl creamos el archivo default.php con el código html necesario para mostrar el formulario de datos en pantalla, ver Figuras III.19, III.20 y III.21.



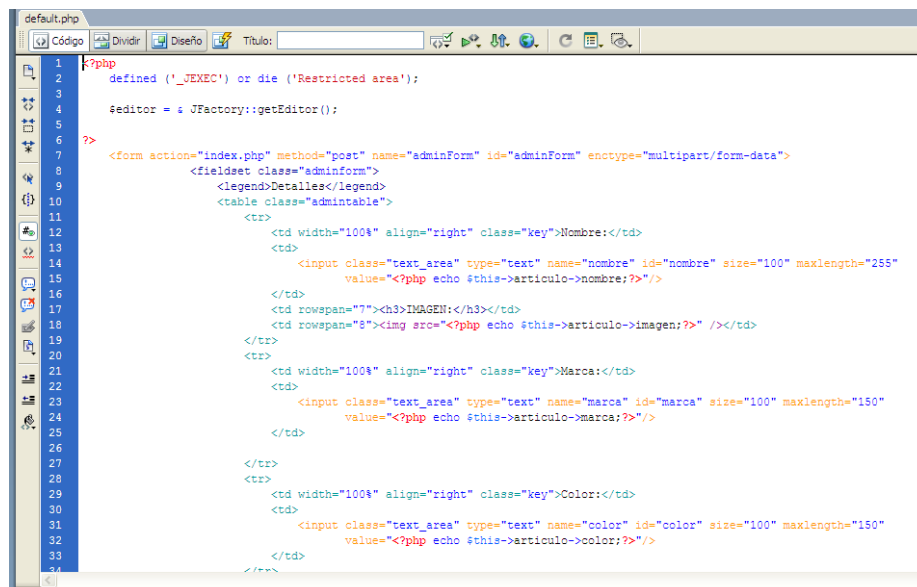
```
1 <?php
2 defined ('_JEXEC') or die ('Restricted area');
3 jimport('joomla.application.component.model');
4
5 class ModelLibreriaArticulo extends JModel
6 {
7     var $_articulo = null;
8     var $_lists=null;
9     var $_id = null;
10
11     function __construct()
12     {
13         parent::__construct();
14
15         $cid = JRequest::getVar('cid', array(0), '', 'array');
16         $this->_id = $cid[0];
17     }
18
19     function getArticulo()
20     {
21         if (!$this->_articulos)
22         {
23             $this->_articulo = JTable::getInstance('articulo', 'Table');
24             $this->_articulo->load($this->_id);
25         }
26         return $this->_articulo;
27     }
28 }
```

Figura III. 19: Creación dentro de la carpeta models, el archivo articulo.php– Módulo 1 Herramienta Joomla! 1.5



```
1 <?php
2 defined ('_JEXEC') or die ('Restricted area');
3 jimport('joomla.application.component.view');
4 class LibreriaViewArticulo extends JView
5 {
6     function display ($tpl = null)
7     {
8
9         global $option, $mainframe;
10        $model = $this->getModel();
11        $articulo = $model->getArticulo();
12        $lists = $model->getLists();
13
14        $this->assignRef('articulo', $articulo);
15        $this->assignRef('lists', $lists);
16        $this->assignRef('option', $option);
17        parent::display($tpl);
18    }
19 }
20 ?>
```

Figura III. 20: Creación dentro de la carpeta views\articulo, el archivo view.html.php – Módulo 1 Herramienta Joomla! 1.5



```
default.php
1 <?php
2 defined ('_JEXEC') or die ('Restricted area');
3
4 $editor = JFactory::getEditor();
5
6 >?
7
8 <form action="index.php" method="post" name="adminForm" id="adminForm" enctype="multipart/form-data">
9   <fieldset class="adminform">
10     <legend>Detalles</legend>
11     <table class="admintable">
12       <tr>
13         <td width="100%" align="right" class="key">Nombre:</td>
14         <td>
15           <input class="text_area" type="text" name="nombre" id="nombre" size="100" maxlength="255"
16             value="<?php echo $this->articulo->nombre;?>"/>
17         </td>
18         <td rowspan="7"><h3>IMAGEN:</h3></td>
19         <td rowspan="8"></td>
20       </tr>
21       <tr>
22         <td width="100%" align="right" class="key">Marca:</td>
23         <td>
24           <input class="text_area" type="text" name="marca" id="marca" size="100" maxlength="150"
25             value="<?php echo $this->articulo->marca;?>"/>
26         </td>
27       </tr>
28       <tr>
29         <td width="100%" align="right" class="key">Color:</td>
30         <td>
31           <input class="text_area" type="text" name="color" id="color" size="100" maxlength="150"
32             value="<?php echo $this->articulo->color;?>"/>
33         </td>
34       </tr>
35     </table>
36   </fieldset>
37 </form>
```

Figura III. 21: Creación dentro de la carpeta views\articulo\tmpl, el archivo default.php – Módulo 1 Herramienta Joomla! 1.5

A continuación se ejecuta el proyecto, ingresar al back'end de la aplicación mediante un navegador de internet, en este caso Mozilla Firefox, y escoger el componente com_libreria, y la opción Nuevo. La pantalla final es la que se muestra en la Figura III.22:

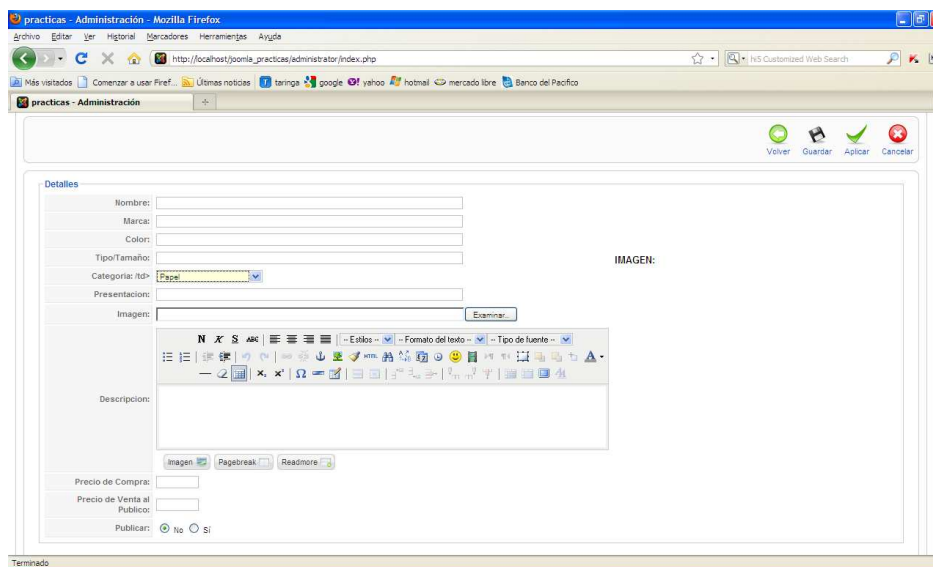


Figura III. 22: Ejecución de la aplicación – Módulo 1 Herramienta Joomla! 1.5

3.7.1.2. Módulo 2:

En este módulo en primera instancia, se crea una tabla llamada jos_libreria_articulo en la Base de Datos de MySQL, en la que se van a almacenar los valores (ver Figura III.23).



Figura III. 23: Creación de la tabla jos_libreria_articulo en la base de datos deMySQL – Módulo 2 Herramienta Joomla! 1.5

Ya establecida la tabla, se crea dentro del directorio tables, el archivo articulo.php, donde se crea la clase TableArticulo, donde se declara todos los campos de la tabla como atributos, ver Figura III.23.

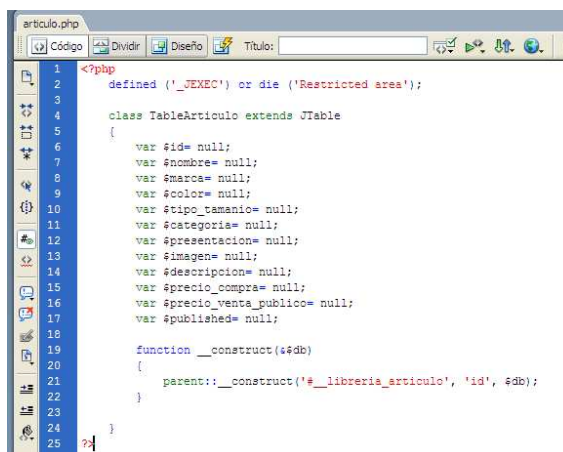
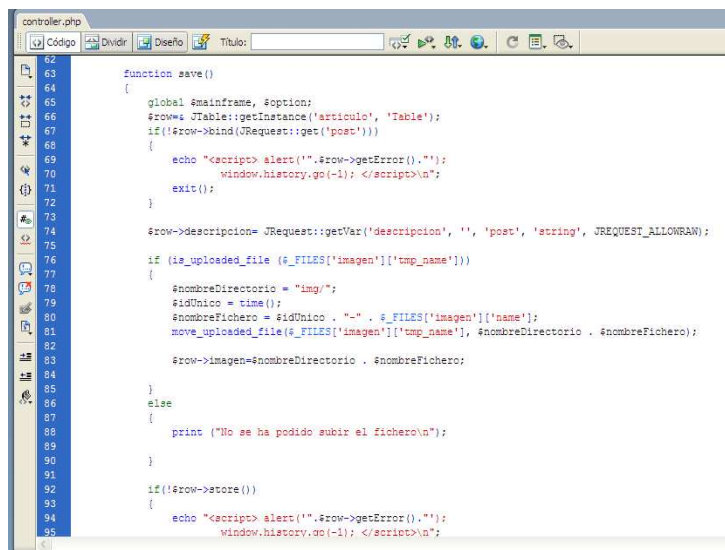


Figura III. 24: Creación del archivo articulo.php, dentro del directorio tables – Módulo 2 Herramienta Joomla! 1.5

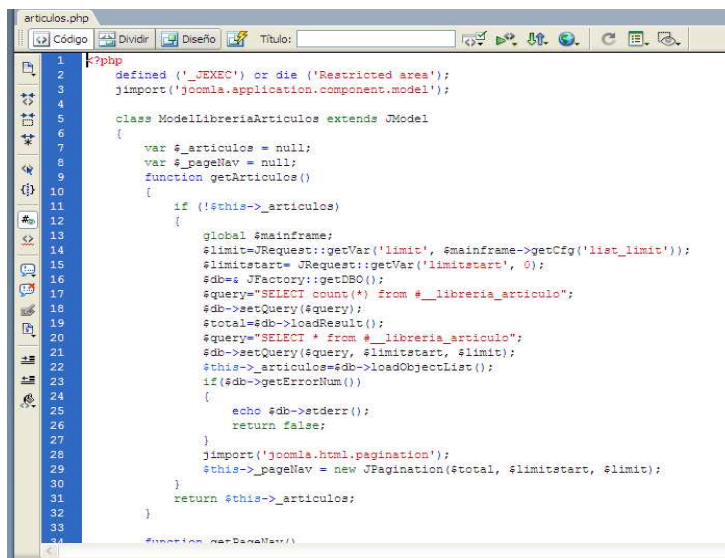
En el controlador del archivo controller.php, se crea la función save, la cual contiene el código necesario para realizar la inserción de los datos a la tabla, en la base de datos, ver Figura III.25.



```
62
63 function save()
64 {
65     global $mainframe, $option;
66     $row= JTable::getInstance('articulo', 'Table');
67     if(!$row->bind(JRequest::get('post')))
68     {
69         echo "<script> alert('".$row->getError()."');
70             window.history.go(-1); </script>\n";
71         exit();
72     }
73
74     $row->descripcion= JRequest::getVar('descripcion', '', 'post', 'string', JREQUEST_ALLOWRAW);
75
76     if (is_uploaded_file ($_FILES['imagen']['tmp_name']))
77     {
78         $nombreDirectorio = "img/";
79         $idUnico = time();
80         $nombreFichero = $idUnico . "-" . $_FILES['imagen']['name'];
81         move_uploaded_file($_FILES['imagen']['tmp_name'], $nombreDirectorio . $nombreFichero);
82
83         $row->imagen=$nombreDirectorio . $nombreFichero;
84     }
85     else
86     {
87         print ("No se ha podido subir el fichero\n");
88     }
89
90     if(!$row->store())
91     {
92         echo "<script> alert('".$row->getError()."');
93             window.history.go(-1); </script>\n";
94     }
95 }
```

Figura III. 25: Creación de la función save en el controlador controller.php – Módulo 2 Herramienta Joomla! 1.5

Luego se crea el modelo, y la vista respectiva en el back-end de la aplicación, para el listado de los artículos almacenados en la base de datos. Primero dentro de la carpeta models se crea el archivo artículos.php con el código preciso para extraer los datos de la tabla, ver Figura III.26.



```
1
2 defined ('_JEXEC') or die ('Restricted area');
3 jimport('joomla.application.component.model');
4
5 class ModelLibreriaArticulos extends JModel
6 {
7     var $_articulos = null;
8     var $_pageNav = null;
9     function getArticulos()
10    {
11        if (!$this->_articulos)
12        {
13            global $mainframe;
14            $limit=JRequest::getVar('limit', $mainframe->getConfig('list_limit'));
15            $limitstart= JRequest::getVar('limitstart', 0);
16            $db= JFactory::getDBO();
17            $query="SELECT count(*) from #__libreria_articulo";
18            $db->setQuery($query);
19            $total=$db->loadResult();
20            $query="SELECT * from #__libreria_articulo";
21            $db->setQuery($query, $limitstart, $limit);
22            $this->_articulos=$db->loadObjectList();
23            if ($db->getErrorNum())
24            {
25                echo $db->stderr();
26                return false;
27            }
28            jimport('joomla.html.pagination');
29            $this->_pageNav = new JPagination($total, $limitstart, $limit);
30        }
31        return $this->_articulos;
32    }
33
34    function getPageNav()
```

Figura III. 26: Creación dentro de la carpeta models, el archivo articulos.php– Módulo 2 Herramienta Joomla! 1.5

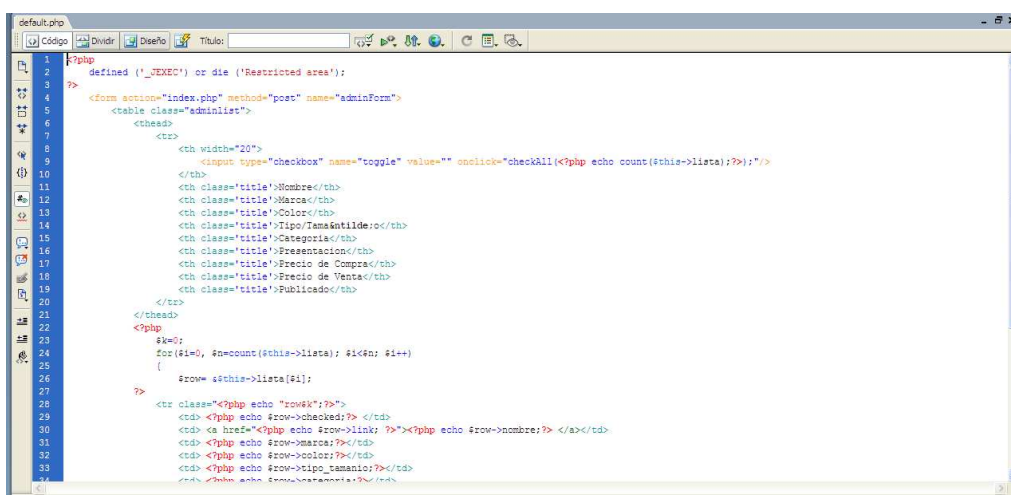
Segundo en la carpeta view se crea el directorio artículos y dentro de él el archivo view.html.php donde se asigna a variables los datos extraídos del modelo, ver Figura III.27.



```
1 <?php
2 defined ('_JEXEC') or die ('Restricted area');
3 jimport ('joomla.application.component.view');
4 class LibreriaViewArticulos extends JView
5 {
6     function display ($tpl = null)
7     {
8
9
10
11         global $option;
12         $model = $this->getModel();
13         $list = $model->getArticulos();
14         $pageNav = $model->getPageNav();
15         jimport ('joomla.filter.output');
16         for ($i=0; $i<count($list); $i++)
17         {
18             $row = $list[$i];
19             $row->checked = JHTML::_('grid.id', $i, $row->id);
20             $row->published = JHTML::_('grid.published', $row, $i);
21             $row->link = JFilterOutput::ampReplace ('index.php?option='.$option.'&task=edit&id='.$row->id.'&view=articulo');
22         }
23         $this->assignRef ('list', $list);
24         $this->assignRef ('pageNav', $pageNav);
25         parent::display ($tpl);
26     }
27 }
```

Figura III. 27: Creación dentro de la carpeta views\articulos, el archivo view.html.php - Módulo 2 Herramienta Joomla! 1.5

Tercero dentro de la carpeta view\articulos\tmpl se crea el archivo default.php, ver Figura III.28, donde se muestra en pantalla los datos que se encuentra en las variables.



```
1 <?php
2 defined ('_JEXEC') or die ('Restricted area');
3
4 <form action="index.php" method="post" name="adminForm">
5     <table class="adminlist">
6         <thead>
7             <tr>
8                 <th width="20%">
9                     <input type="checkbox" name="toggle" value="" onclick="checkAll(<?php echo count($this->list);?>);" />
10                 </th>
11                 <th class="title">Nombre</th>
12                 <th class="title">Marca</th>
13                 <th class="title">Color</th>
14                 <th class="title">Tipo/Tamaño</th>
15                 <th class="title">Categoría</th>
16                 <th class="title">Presentación</th>
17                 <th class="title">Precio de Compra</th>
18                 <th class="title">Precio de Venta</th>
19                 <th class="title">Publicado</th>
20             </tr>
21         </thead>
22         <tbody>
23             <?php
24             $k=0;
25             for ($i=0, $n=count($this->list); $i<$n; $i++)
26             {
27                 $row = $this->list[$i];
28
29                 <tr class="rowk">
30                     <td class="text" ><?php echo $row->checked?> </td>
31                     <td ><?php echo $row->nombre?> </td>
32                     <td ><a href="<?php echo $row->link?>"><?php echo $row->nombre?> </a></td>
33                     <td ><?php echo $row->marca?></td>
34                     <td ><?php echo $row->tipo_tamano?></td>
35                     <td ><?php echo $row->categoria?></td>
36                     <td ><?php echo $row->precio_compra?></td>
37                     <td ><?php echo $row->precio_venta?></td>
38                     <td ><?php echo $row->publicado?></td>
39                 </tr>
40             }
41         </tbody>
42     </table>
43 </form>
```

Figura III. 28: Creación dentro de la carpeta views\articulos\tmpl, el archivo default.php - Módulo 2 Herramienta Joomla! 1.5

A continuación se realiza la ejecución del proyecto, se ingresa al back'end de la aplicación mediante el navegador de internet, y se escoge al componente com_libreria, también se escoge la opción Nuevo, se ingresan los datos requeridos y se pulsa en la opción Guardar, ver Figura III.29.

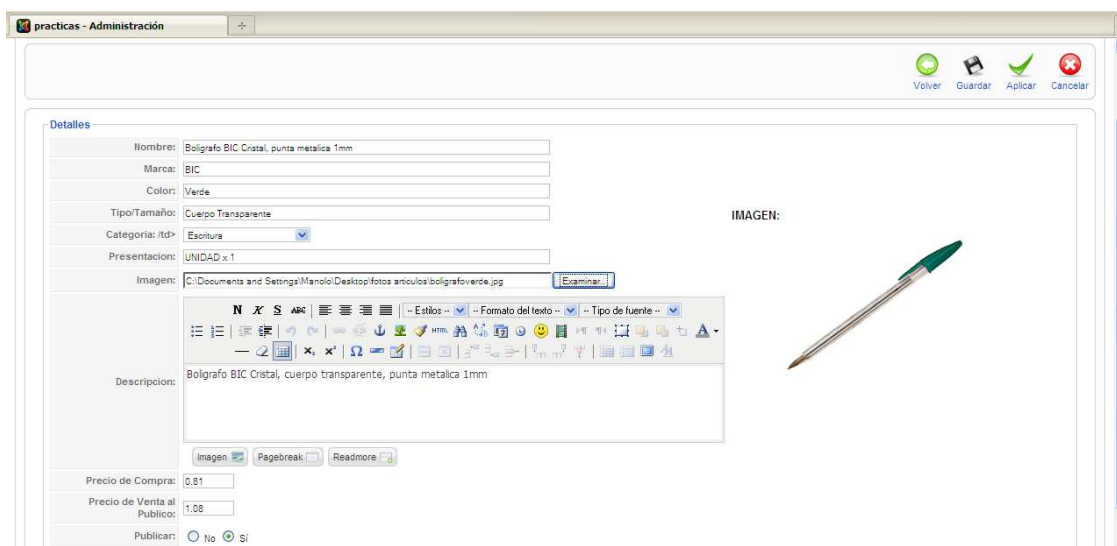


Figura III. 29: Ejecución de la aplicación en el Back-End, ingreso de un artículo - Módulo 2 Herramienta Joomla! 1.5

Luego como resultado se mostrará un listado de los artículos con la información que se ha ingresado en el paso anterior, en el cual se verifica los resultados, ver Figura III.30.

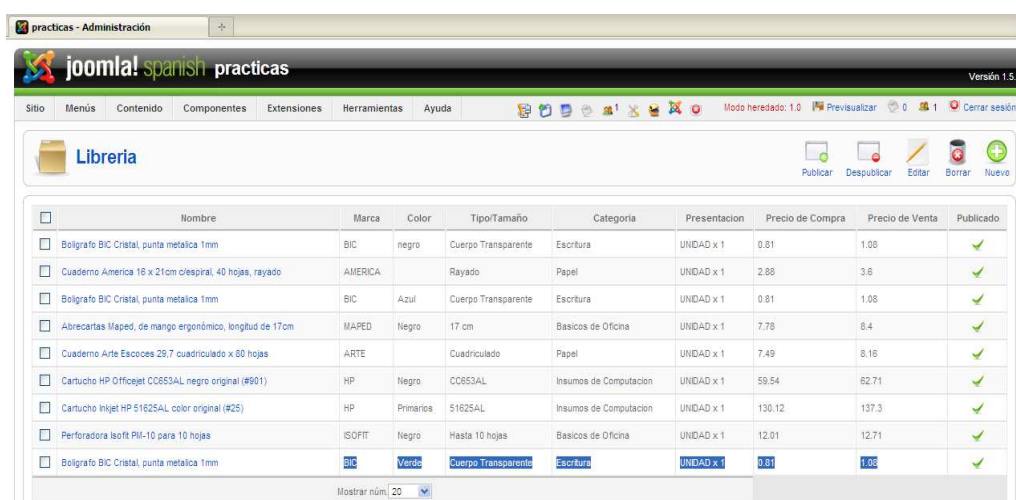


Figura III. 30: Ejecución de la aplicación en el Back-End, listado de los artículos ingresados - Módulo 2 Herramienta Joomla! 1.5

Así mismo en el Front-End, se crea el componente com_libreria con sus respectivos directorios y archivos, en el directorio ..\components, ver Figura III.31, para la visualización del listado de productos, con sus detalles, a los usuarios de la página. De igual manera que en el Back-End se usa el patrón MVC (modelo, vista, controlador).

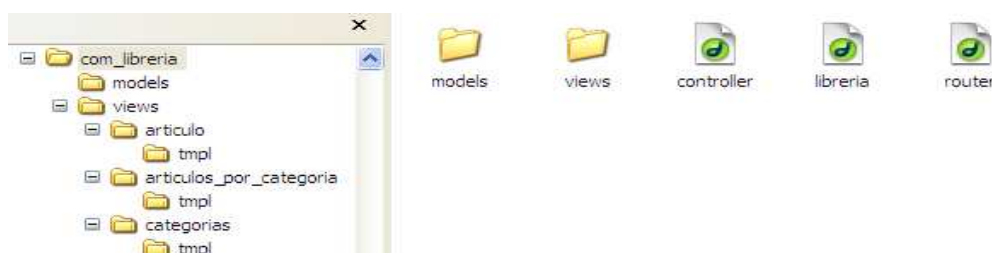


Figura III. 31: Creación del componente com_libreria en el Front-End - Módulo 2 Herramienta Joomla! 1.5

A continuación en la carpeta models, se crean 3 archivos que servirán de modelos, el primer archivo categorías.php donde se realiza un select para obtener los nombres de las categorías existentes en la tabla artículo de la base de datos, el segundo archivo artículos_por_categoria.php en el cual se obtiene el listado de los artículos dada una categoría, el tercer archivo articulo.php ver Figura III.32, donde se consigue los datos de un artículo previamente seleccionado.

```
articulo.php
1 <?php
2 defined('_JEXEC') or die ('Restricted access');
3 jimport('joomla.application.component.model');
4
5 class ModelLibreriaArticulo extends JModel
6 {
7     var $_articulo=null;
8     var $_id=null;
9
10    function __construct()
11    {
12        parent::__construct();
13        $id=JRequest::getVar('id',0);
14        $this->_id=$id;
15    }
16
17    function getArticulo()
18    {
19        if(!$this->_articulo)
20        {
21            $query="SELECT * FROM #__libreria_articulo WHERE id='".$this->_id."'";
22            $this->_db->setQuery($query);
23            $this->_articulo=$this->_db->loadObject();
24            if (!$this->_articulo->published)
25            {
26                JError::raiseError(404, "ID invalido");
27            }
28        }
29        return $this->_articulo;
30    }
31 }
32 ?>
```

Figura III. 32: Modelo 3, creación del archivo articulo.php dentro de la carpeta models en el Front-End - Módulo 2 Herramienta Joomla! 1.5

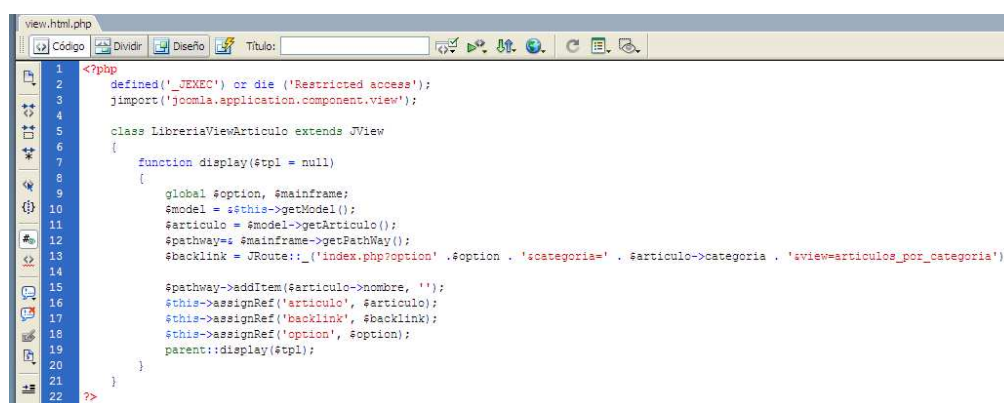
Seguidamente por cada modelo se crean la vista necesaria para mostrar los datos en pantalla, estas se crean dentro del directorio views con los directorios con los archivos necesarios para cada modelo ver Figura III.33.


Dentro del directorio views, se crea la carpeta artículo, y dentro de esta creamos el archivo view.html.php ver Figura III.34, en donde se asigna las variables correspondientes a las datos obtenidos del modelo, dentro de esta carpeta también se crea la carpeta tmpl y dentro de este directorio se crea el archivo default.php ver Figura III.35, en donde se recoge las variables asignadas en el paso anterior y se muestra en pantalla.

Este mismo procedimiento se realiza para la vista categorias, y para la vista artículos_por_categoria, asignando las variables correspondientes de cada modelo y mostrando en pantalla los datos adecuados.



Figura III. 33: Creación del directorio y archivos necesarios para cada una de las vistas en el Front-End - Módulo 2 Herramienta Joomla! 1.5





```
1 k?php defined('_JEXEC') or die ('Restricted access'); ?>
2
3 <p class="contentheading">
4   <?php echo $this->articulo->nombre . ' -- ' . $this->articulo->categoria;?>
5 </p>
6 <table>
7   <tr>
8     <td>
9       <p><strong>Marca: </strong><?php echo $this->articulo->marca;?></p>
10    </td>
11    <td rowspan="3">
12      
13    </td>
14  </tr>
15  <tr>
16    <td>
17      <p><strong>Color: </strong><?php echo $this->articulo->color;?></p>
18    </td>
19  </tr>
20  <tr>
21    <td>
22      <p><strong>Tipo/Tamano: </strong><?php echo $this->articulo->tipo_tamano;?></p>
23    </td>
24  </tr>
25  <tr>
26    <td>
27      <p><strong>Presentacion: </strong><?php echo $this->articulo->presentacion;?></p>
28    </td>
29  </tr>
30  <tr>
31    <td>
32      <p><strong>Precio de Venta (PVP): </strong><?php echo $this->articulo->precio_venta_publico;?></p>
33    </td>
34  </tr>
</table>
```

Figura III. 35: Creación del archivo default.php en el directorio ../views/articulo/tmpl en el Front-End - Módulo 2 Herramienta Joomla! 1.5

Acabado de crear todas las vistas pertinentes, se accede a la página web, al fron-end de la aplicación, usando un navegador de internet. Ver Figura III.36.

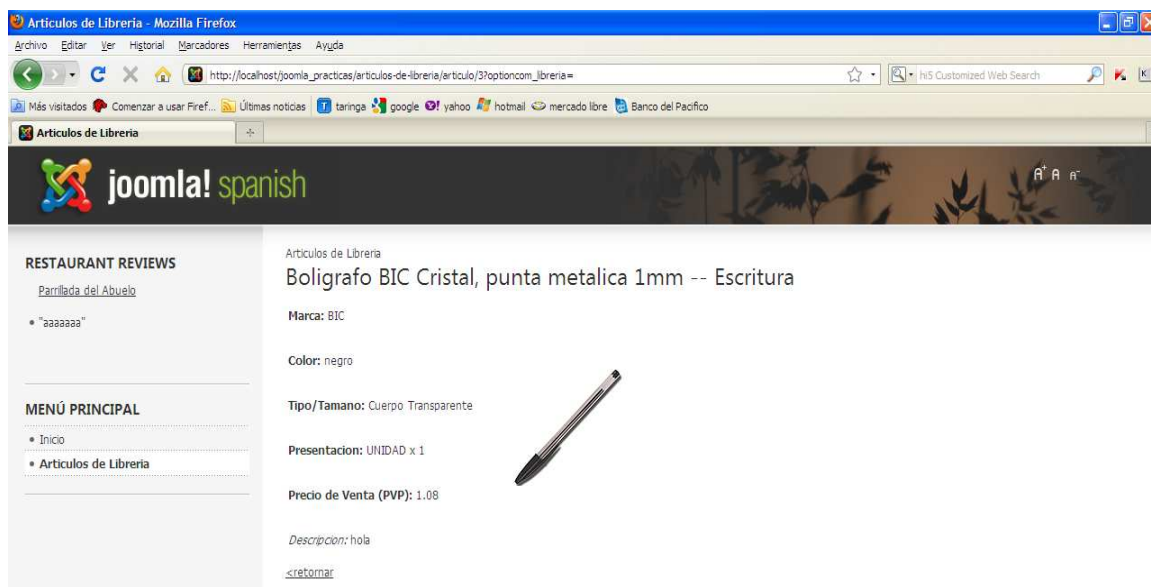


Figura III. 36: Ejecución de la aplicación en el Front-End, datos de un artículo escogido - Módulo 2 Herramienta Joomla! 1.5

3.7.1.3. Módulo 3:

En este módulo, al componente del módulo 2, se lo va a expandir, por lo cual se añadirán 4 tablas más a la base de datos, la tabla jos_libreria_categoria, jos_libreria_subcategoria, jos_libreria_proveedor y la tabla jos_libreria_articuloproveedor. Y a la vez se modifica la estructura de la tabla jos_libreria_articulo ver Figura III.37, eliminando el campo categoría para tenerla en una tabla aparte, con su respectiva sub categoría Ver Figura III.38.

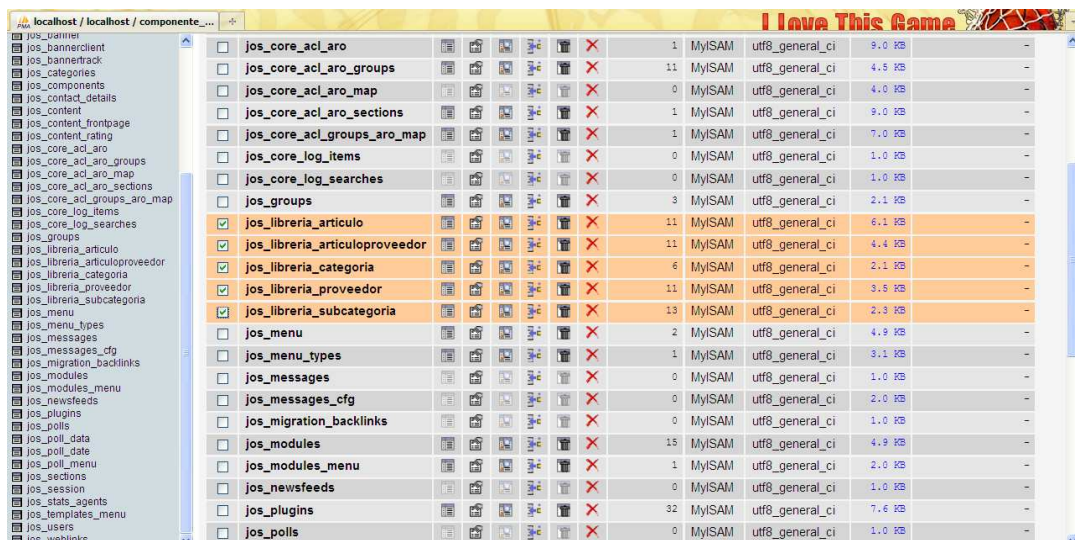


Figura III. 37: Creación de las nuevas tablas en la base de datos de MySQL – Módulo 3 Herramienta Joomla! 1.5



Figura III. 38: Nueva estructura de la tabla jos_libreria_articulo en la base de datos de MySQL – Módulo 3 Herramienta Joomla! 1.5

Consecuentemente en el componente dentro de la carpeta tables Ver Figura III.39, se crea las clases respectivas para cada tabla, y como atributos de la clase los campos pertenecientes a cada tabla. El archivo categoría.php contiene la clase respectiva a su tabla y los campos de la misma como atributos inicializados en null, el constructor de la clase donde se especifica a que tabla corresponde la clase, y su llave primaria. Ver Figura III.40.

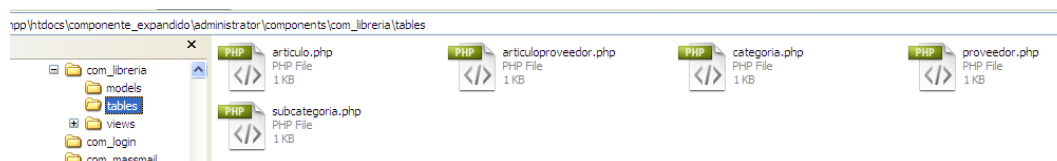


Figura III. 39: Creación de los archivos correspondientes a las tablas del componente en la base de datos, dentro del directorio tables – Módulo 3 Herramienta Joomla! 1.5

```
1 <?php
2 defined ('_JEXEC') or die ('Restricted area');
3
4 class TableCategoria extends JTable
5 {
6     var $id_categoria= null;
7     var $nombre_categoria= null;
8     var $published= null;
9
10    function __construct($db)
11    {
12        parent::__construct('#__libreria_categoria', 'id_categoria', $db);
13    }
14
15 }
16
```

Figura III. 40: Creación del archivo categoria.php, dentro del directorio tables – Módulo 3 Herramienta Joomla! 1.5

Ahora se crean los archivos y directorios pertinentes, para el desarrollo del componente expandido, usando el patrón MVC (modelo, vista, controlador) Ver Figura III.41.

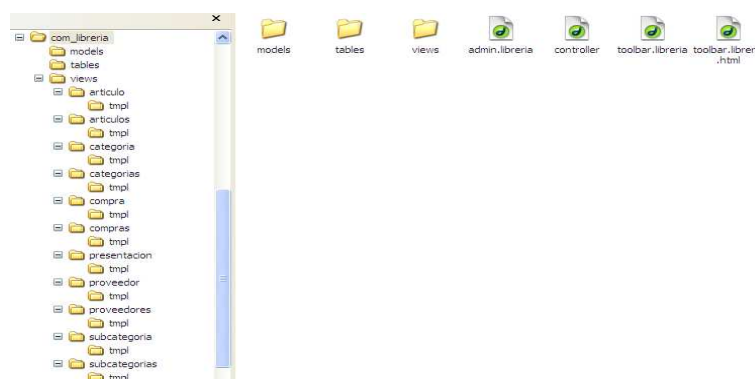


Figura III. 41: Creación del patrón MVC, para el desarrollo del componente expandido com_libreria en el back-end – Módulo 3 Herramienta Joomla! 1.5

Ahora, dentro de la carpeta models, se crean los archivos necesarios para extraer los datos concretos tanto para una vista específica, como para una vista de un listado, de todas las tablas en la base de datos. Ver Figura III.42.



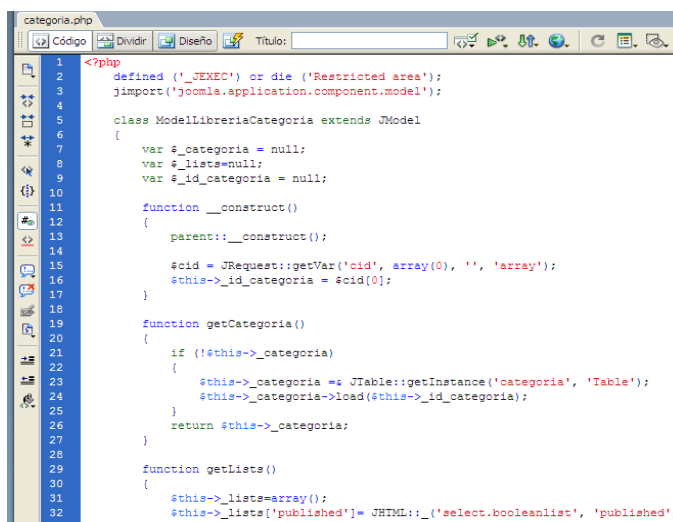
Figura. 42: Creación los archivos dentro de la carpeta models, modelos correspondientes para la obtención de datos específicos – Módulo 3 Herramienta Joomla! 1.5

Por ejemplo para obtener el listado de categorías existentes en el back-end, se crea el archivo categorías.php. Ver Figura III.43.

```
1  <?php
2  defined('_JEXEC') or die('Restricted area');
3  jimport('joomla.application.component.model');
4
5  class ModelLibreriaCategorias extends JModel
6  {
7      var $_categorias = null;
8      var $_pageNav = null;
9      function getCategorias()
10     {
11         if (!$this->_categorias)
12         {
13             global $mainframe;
14             $limit=JRequest::getVar('limit', $mainframe->getConfig('list_limit'));
15             $limitstart= JRequest::getVar('limitstart', 0);
16             $db= JFactory::getDBO();
17             $query="SELECT count(*) from #__libreria_categoria";
18             $db->setQuery($query);
19             $total=$db->loadResult();
20             $query="SELECT * from #__libreria_categoria";
21             $db->setQuery($query, $limitstart, $limit);
22             $this->_categorias=$db->loadObjectList();
23             if($db->getErrorNum())
24             {
25                 echo $db->getError();
26                 return false;
27             }
28             jimport('joomla.html.pagination');
29             $this->_pageNav = new JPagination($total, $limitstart, $limit);
30         }
31         return $this->_categorias;
32     }
33 }
```

Figura III. 43: Creación del archivo categorías.php, modelo para la obtención de los datos para el listado de las categorías existentes en el back-end – Módulo 3 Herramienta Joomla! 1.5

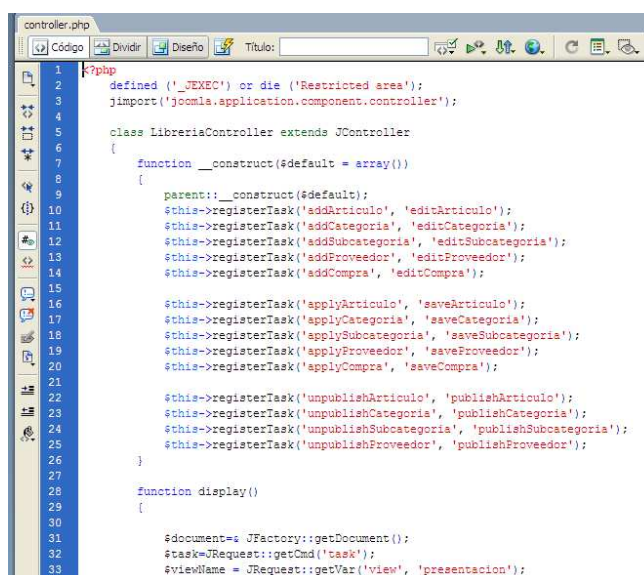
Para obtener los datos de una categoría en específica se crea el archivo categoría.php. Ver Figura III.44.



```
1 <?php
2 defined ('_JEXEC') or die ('Restricted area');
3 jimport ('joomla.application.component.model');
4
5 class ModelLibreriaCategoria extends JModel
6 {
7     var $_categoria = null;
8     var $_lists=null;
9     var $_id_categoria = null;
10
11     function __construct()
12     {
13         parent::__construct();
14
15         #cid = JRequest::getVar('cid', array(0), '', 'array');
16         $this->_id_categoria = $cid[0];
17     }
18
19     function getCategory()
20     {
21         if (!$this->_categoria)
22         {
23             $this->_categoria = JTable::getInstance('categoria', 'Table');
24             $this->_categoria->load($this->_id_categoria);
25         }
26         return $this->_categoria;
27     }
28
29     function getLists()
30     {
31         $this->_lists=array();
32         $this->_lists['published']= JHTML::_('select.booleanlist', 'published','
```

Figura III. 44: Creación del archivo categoría.php, modelo para la obtención de los datos para una categoría específica existente en el back-end – Módulo 3 Herramienta Joomla! 1.5

Consecuentemente en el controlador controller.php en la carpeta del componente, se crean las funciones necesarias para la gestión de los datos de las tablas, como son: guardar, modificar, eliminar Ver Figura III.45.



```
1 <?php
2 defined ('_JEXEC') or die ('Restricted area');
3 jimport ('joomla.application.component.controller');
4
5 class LibreriaController extends JController
6 {
7     function __construct($default = array())
8     {
9         parent::__construct($default);
10         $this->registerTask('addArticulo', 'editArticulo');
11         $this->registerTask('addCategoria', 'editCategoria');
12         $this->registerTask('addSubcategoria', 'editSubcategoria');
13         $this->registerTask('addProveedor', 'editProveedor');
14         $this->registerTask('addCompra', 'editCompra');
15
16         $this->registerTask('applyArticulo', 'saveArticulo');
17         $this->registerTask('applyCategoria', 'saveCategoria');
18         $this->registerTask('applySubcategoria', 'saveSubcategoria');
19         $this->registerTask('applyProveedor', 'saveProveedor');
20         $this->registerTask('applyCompra', 'saveCompra');
21
22         $this->registerTask('unpublishArticulo', 'publishArticulo');
23         $this->registerTask('unpublishCategoria', 'publishCategoria');
24         $this->registerTask('unpublishSubcategoria', 'publishSubcategoria');
25         $this->registerTask('unpublishProveedor', 'publishProveedor');
26     }
27
28     function display()
29     {
30
31         $document= JFactory::getDocument();
32         $task=JRequest::getCmd('task');
33         $viewName = JRequest::getVar('view', 'presentacion');
```

Figura III. 45: Modificación del archivo controller.php, controlador para la gestión de datos de las tablas existentes en el back-end – Módulo 3 Herramienta Joomla! 1.5

En seguida, dentro de la carpeta views del componente se realiza la vista por cada modelo que se tiene, se crea un directorio por cada vista, ver Figura III.46.

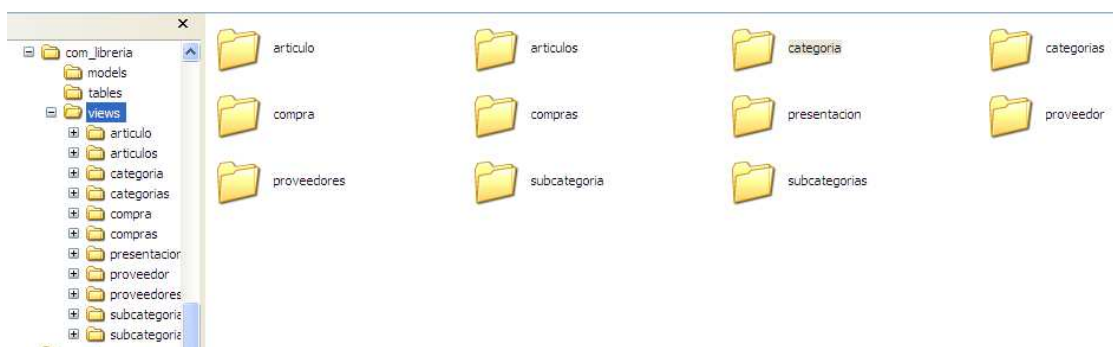


Figura III. 46: Creación dentro de la carpeta views\categorias, el archivo view.html.php en el back-end - Módulo 3 Herramienta Joomla! 1.5

Creación de la vista para listar las categorías existentes, en el directorio categorías, ver Figura III.47.

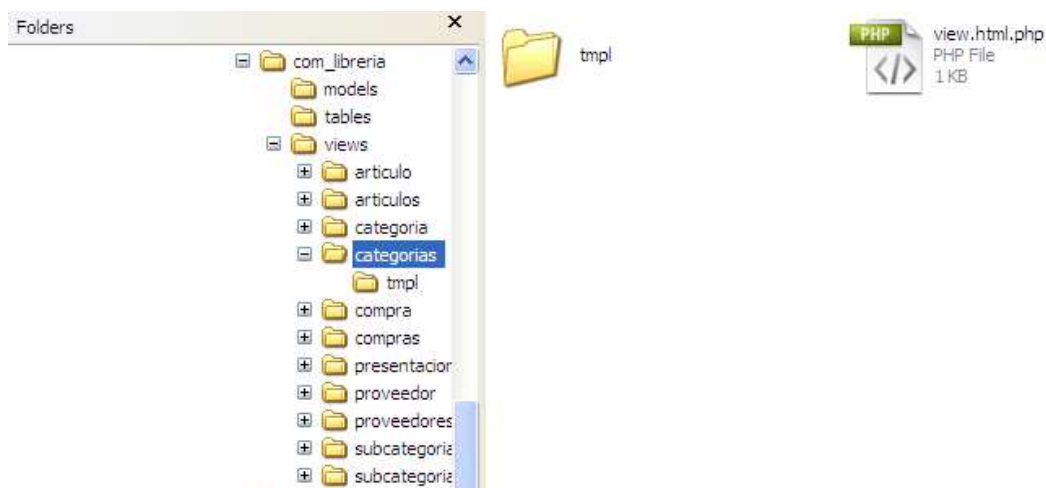
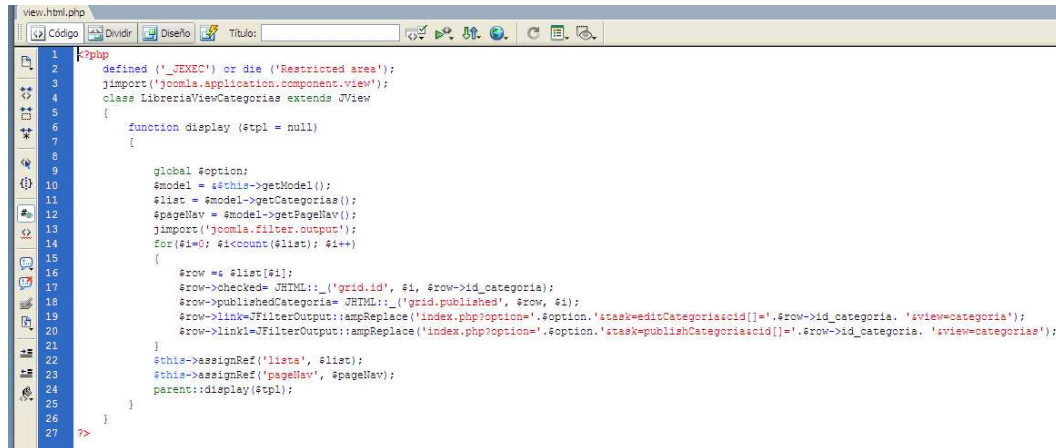


Figura III. 47: Creación de la vista para el listado de categorías en el back-end - Módulo 3 Herramienta Joomla! 1.5

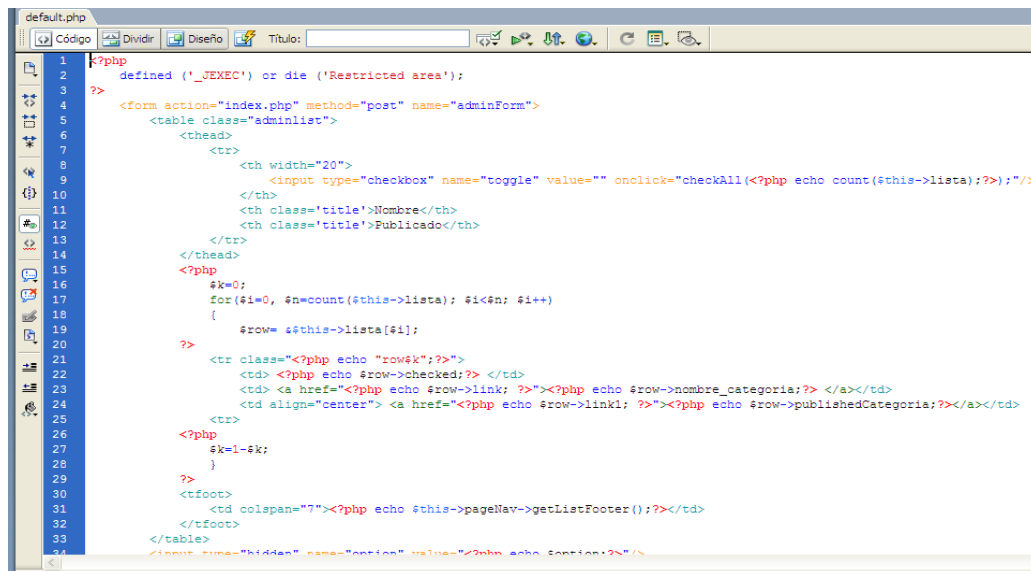
En esta carpeta se crea el archivo view.html.php en el cual se asigna variables con los datos obtenidos en el modelo correspondiente, ver Figura III.48.



```
1 <?php
2 defined ('_JEXEC') or die ('Restricted area');
3
4 jimport('joomla.application.component.view');
5 class LibreriaViewCategorias extends JView
6 {
7     function display ($tpl = null)
8     {
9
10         global $option;
11         $model = $this->getModel();
12         $list = $model->getCategorias();
13         $pageNav = $model->getPageNav();
14         jimport('joomla.filter.output');
15         for($i=0; $i<count($list); $i++)
16         {
17             $row = $list[$i];
18             $row->checked= JHTML::_('grid.id', $i, $row->id_categoria);
19             $row->publishedCategoria= JHTML::_('grid.published', $row, $i);
20             $row->link=JFilterOutput::ampReplace('index.php?option='.$option.'task=editCategorias&id='.$row->id_categoria.'&view=categorias');
21             $row->link1=JFilterOutput::ampReplace('index.php?option='.$option.'task=publishCategorias&id='.$row->id_categoria.'&view=categorias');
22         }
23         $this->assignRef('lista', $list);
24         $this->assignRef('pageNav', $pageNav);
25         parent::display($tpl);
26     }
27 }
```

Figura III. 48: Creación dentro de la carpeta views/categorias, el archivo view.html.php en el back-end - Módulo 3 Herramienta Joomla! 1.5

Dentro de esta misma carpeta de la vista se crea el directorio tmpl, y dentro de la carpeta tmpl se crea del archivo default.php ver Figura III.49, en el cual se recoge las variables previamente asignadas y se las muestra en pantalla.



```
1 <?php
2 defined ('_JEXEC') or die ('Restricted area');
3
4 <?php
5 <form action="index.php" method="post" name="adminForm">
6 <table class="adminlist">
7 <thead>
8 <tr>
9 <th width="20">
10 <input type="checkbox" name="toggle" value="" onclick="checkAll(<?php echo count($this->lista);?>);"/>
11 </th>
12 <th class="title">Nombre</th>
13 <th class="title">Publicado</th>
14 </thead>
15 <tbody>
16 <?php
17 $k=0;
18 for($i=0, $n=count($this->lista); $i<$n; $i++)
19 {
20     $row= $this->lista[$i];
21
22     <tr class="<?php echo "row$k";?>">
23 <td><?php echo $row->checked;?> </td>
24 <td><a href="<?php echo $row->link;?>"><?php echo $row->nombre_categoria;?> </a></td>
25 <td align="center"><a href="<?php echo $row->link1;?>"><?php echo $row->publishedCategoria;?></a></td>
26 </tr>
27 <?php
28 $k=1-$k;
29 }
30 <tbody>
31 <td colspan="7"><?php echo $this->pageNav->getListFooter();?></td>
32 </tbody>
33 </table>
34 <input type="hidden" name="option" value="<?php echo $option;?>" />
```

Figura III. 49: Creación dentro de la carpeta views/categorias/tmpl, el archivo default.php en el back-end - Módulo 3 Herramienta Joomla! 1.5

Este mismo procedimiento se repite en cada vista que se cree.

Ahora se ejecuta la aplicación en la parte del back-end, como demostración se escogió el listado de artículos existentes, ver Figura III.50, y posteriormente se escoge un artículo del listado para visualizarlo de forma individual, ver Figura III.51.

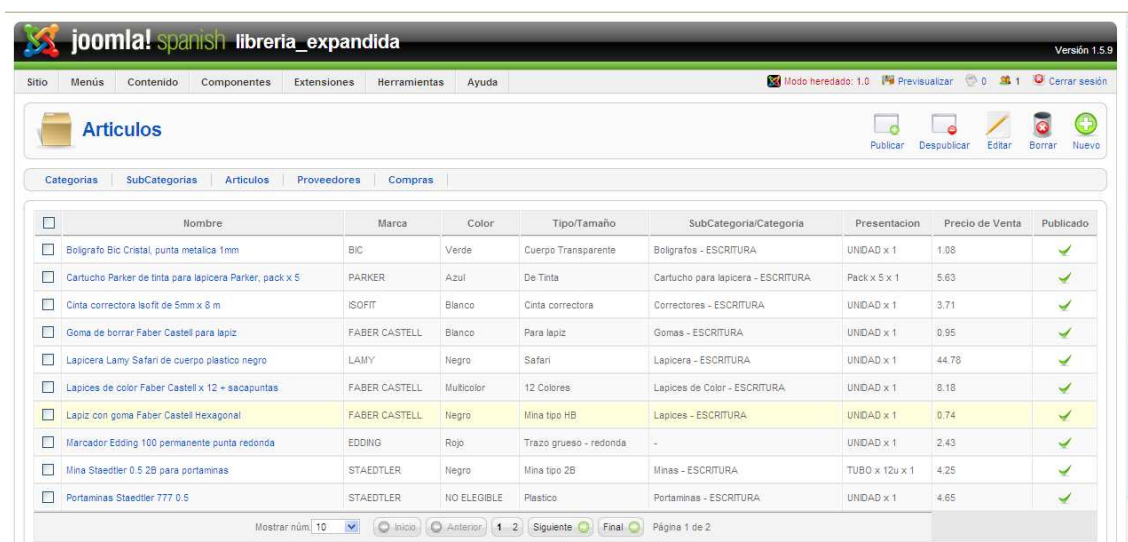


Figura III. 50: Pantalla de visualización del listado de artículos existentes, en el back-end - Módulo 3 Herramienta Joomla! 1.5

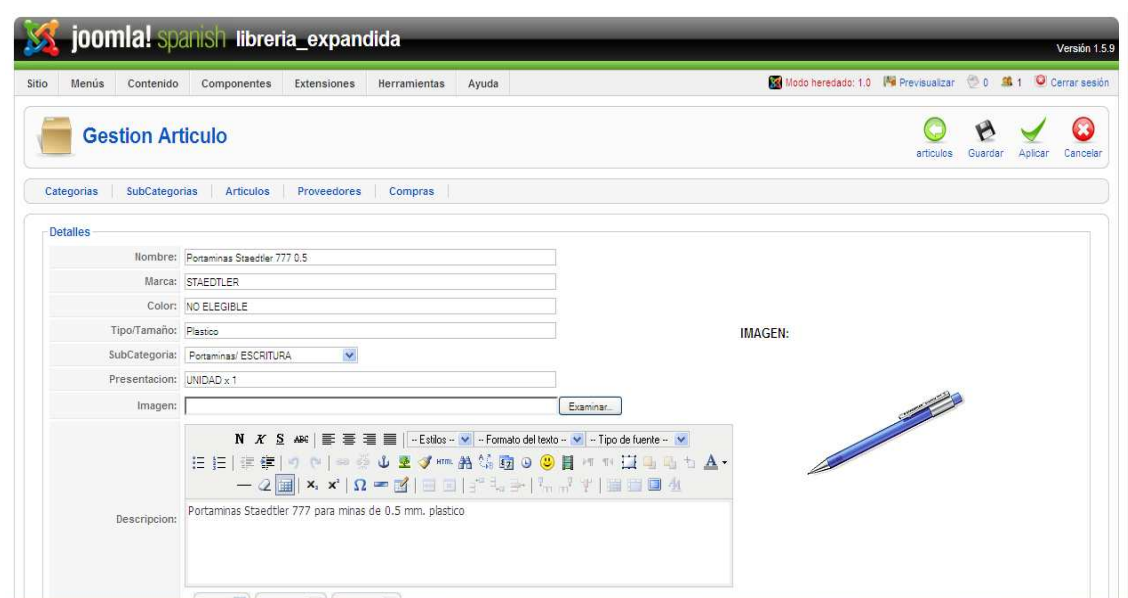


Figura III. 51: Pantalla de visualización de un artículo específico, en el back-end - Módulo 3 Herramienta Joomla! 1.5

Ahora que ya está implementado el componente en el back-end, como paso final se instaura el componente en el front-end, para lo cual se crean los directorios y archivos necesarios, para poder usar el patrón MVC ver Figura III.52.

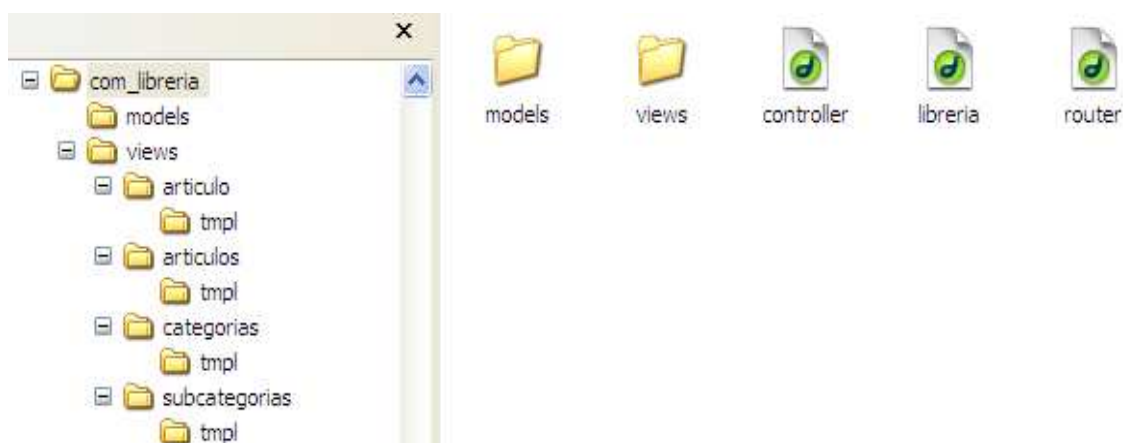


Figura III. 52: Creación del patrón MVC, para el desarrollo del componente expandido com_libreria en el front-end - Módulo 3 Herramienta Joomla! 1.5

En el controlador, el archivo controller.php, solo existirá la función display, la cual direcciona a la vista que se mostrara inicialmente, Ver Figura III.53.



Figura III. 53: Creación del archivo controller.php, controlador del componente expandido en el front-end - Módulo 3 Herramienta Joomla! 1.5

Dentro de la carpeta models, se crean los archivos necesarios para extraer los datos de la base de datos, que van a ser mostrados por las vistas respectivas a los usuarios finales en el front-end, ver Figura III.54. Los modelos a crear son los del listado de categorías, listado de sub categorías dada una categoría, listado de artículos dada una sub categoría, y los datos de un artículo específico previamente escogido. Para demostración se incluye el modelo categorias.php, ver Figura III.55.

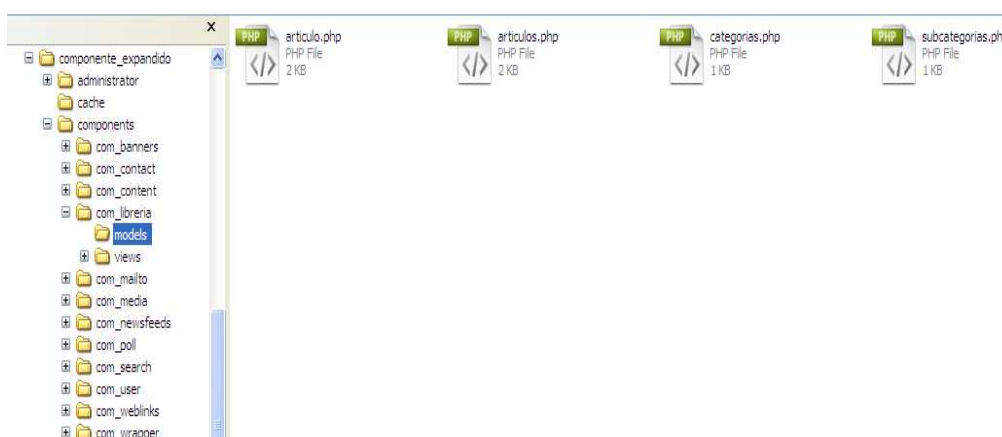


Figura III. 54: Creación de los archivos para cada modelo necesario en el componente en el front-end – Módulo 3 Herramienta Joomla! 1.5

```
1 <?php
2 defined('_JEXEC') or die ('Restricted access');
3 jimport('joomla.application.component.model');
4
5 class ModelLibreriaCategorias extends JModel
6 {
7     var $_categorias=null;
8
9     function getCategorias()
10    {
11        if(!$this->_categorias)
12        {
13            $query="SELECT id_categoria, nombre_categoria FROM #__libreria_categoria WHERE published = '1'";
14            $this->_categorias=$this->_getList($query, 0, 0);
15        }
16        return $this->_categorias;
17    }
18 }
19 ?>
```

Figura III. 55: Creación del archivo categorias.php, modelo para la obtención de los datos para el listado de las categorías existentes en el front-end – Módulo 3 Herramienta Joomla! 1.5

En seguida, dentro de la carpeta views del componente se realiza la vista por cada modelo que se tiene, se crea un directorio por cada vista, ver Figura III.56, y dentro de esta carpeta se crea el archivo view.html.php en el cual se asignan las variables con los datos obtenidos en el modelo correspondiente, dentro de esta misma carpeta de la vista se crea el directorio tmpl, y dentro de la carpeta tmpl se crea el archivo default.php, en el cual se recogen las variables previamente asignadas y se muestran en pantalla.

Modificación de la vista para listar las categorías existentes, en el directorio views\categorias el archivo view.html.php Ver Figura III.57 y el archivo default.php ver Figura III.58.



Figura III. 56: Creación de los directorios para cada vista correspondientes, el archivo en el front-end - Módulo 3 Herramienta Joomla! 1.5

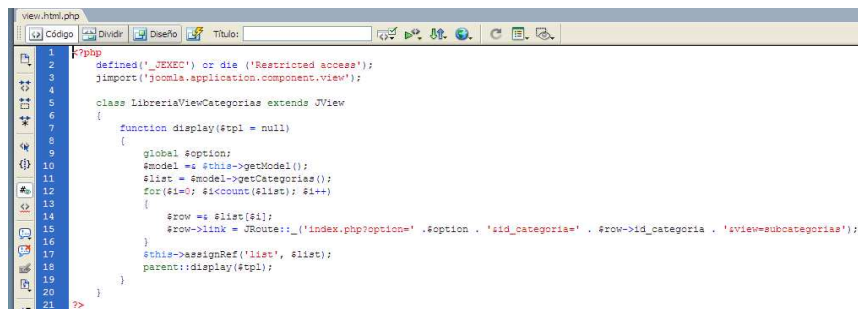


Figura III. 57: Modificación dentro de la carpeta views\categorias, el archivo view.html.php en el front-end - Módulo 3 Herramienta Joomla! 1.5

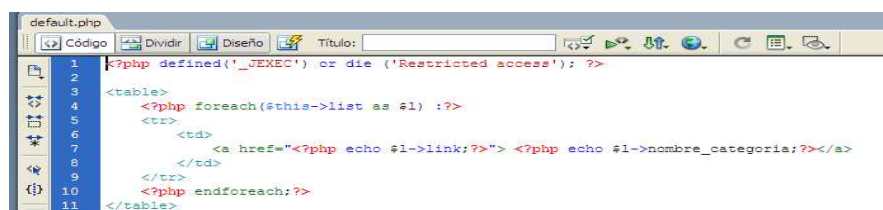


Figura III. 58: Modificación dentro de la carpeta views\categorias\tmpl, el archivo default.php en el front-end - Módulo 3 Herramienta Joomla! 1.5

Ahora se ejecuta la aplicación mediante el front-end, y se observa los resultados. Para demostración es escoge la pantalla del listado de categorías, ver Figura III.59.



Figura III. 59: Visualización del listado de Categorías en el front-end - Módulo 3 Herramienta Joomla! 1.5

Dentro de Artículos de Librería en Escritura se muestra la pantalla de la visualización de un artículo en específico, en este caso de Lápices de Color. Ver Figura 3.60.



Figura III. 60: Visualización de un artículo específico en el front-end - Módulo 3 Herramienta Joomla! 1.5

3.7.1.4. Módulo 4

En este módulo, se estudia las facilidades que Joomla! ofrece para la administración de los componentes, su instalación, desinstalación, habilitación, des habilitación, entre otros ver Figura III.61.

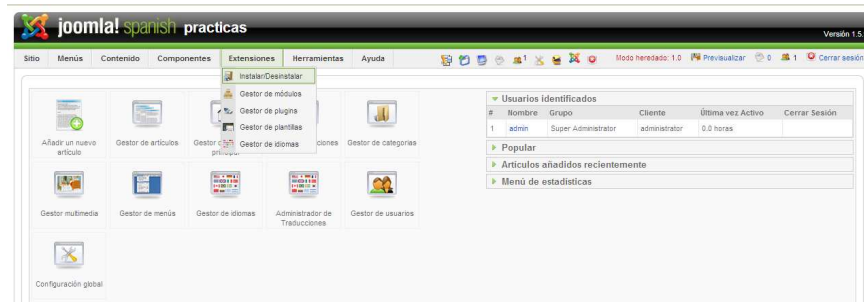


Figura III. 61: Menú para la administración de extensiones - Módulo 4 Herramienta Joomla! 1.5

Para la instalación de un componente, Joomla! brinda amigables prestaciones, ver Figura III.62 y permite buscar el componente en de forma local o en una dirección web, ver Figura III.63.

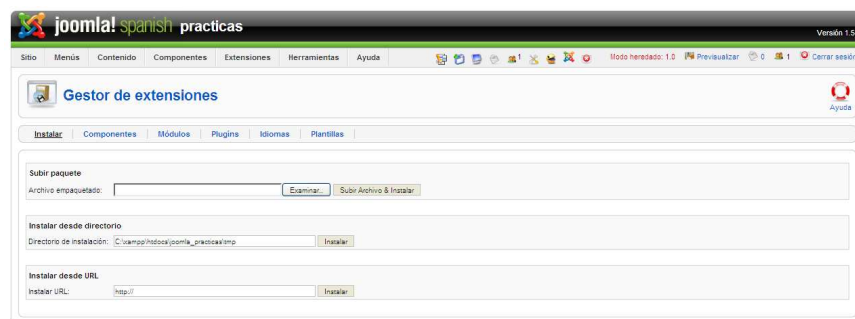


Figura III. 62: Instalación de componentes - Módulo 4 Herramienta Joomla! 1.5



Figura III. 63: Búsqueda del componente a instalar - Módulo 4 Herramienta Joomla! 1.5

Si el procedimiento se realizó correctamente, y el componente a instalar contiene todos los archivos necesarios de instalación, se mostrará en pantalla un mensaje de éxito en la instalación. Ver Figura III.64.



Figura III. 64: Componente instalado correctamente - Módulo 4 Herramienta Joomla! 1.5

Para la desinstalación de un componente, Joomla! muestra un listado de los mismos, en el cual se puede solamente habilitarlo o des habilitarlo, solamente con dar clic en la figura con forma de visto de color verde, y si se desea desinstalarlo completamente se lo selecciona y se da clic en desinstalar, ver Figura III.65.



Figura III. 65: Gestión de componentes, desinstalación, habilitación, des habilitación - Módulo 4 Herramienta Joomla! 1.5

Si el proceso de desinstalación se realizó completamente se mostrará un mensaje de éxito de la desinstalación. Ver Figura III.66.

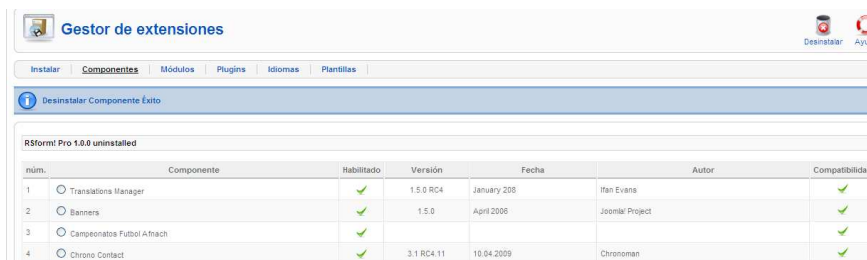


Figura III. 66: Desinstalación exitosa de un componente - Módulo 4 Herramienta Joomla! 1.5

3.7.2. Desarrollo de los Módulos en la Herramienta TYPO3

3.7.2.1. Módulo 1

Para la creación de este módulo se usa la aplicación kickstarterwizard, la cual ayuda en la creación de los archivos necesarios para el funcionamiento de la extensión que se llamará LibreriaArticulos, a la cual se define como FrontendPlugin. Ver Figura III.67.

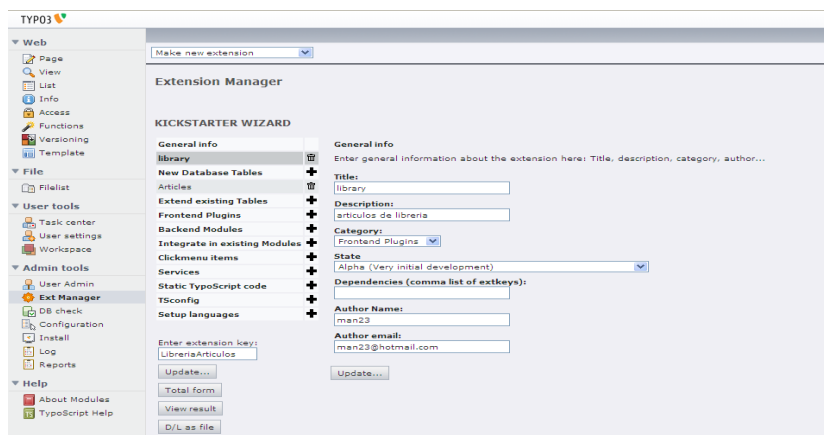


Figura III. 67: Creación de la extensión LibreriaArticulos, con la aplicación kickstarterwizard - Módulo 1 Herramienta Typo3 v. 4

Los archivos a crearse de la extensión LibreriaArticulos con la aplicación kickstarterwizard, son los siguientes. Ver Figura III.68

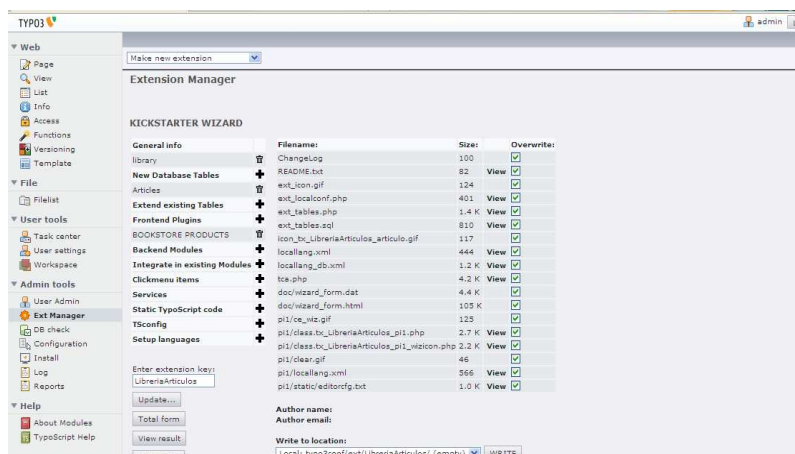
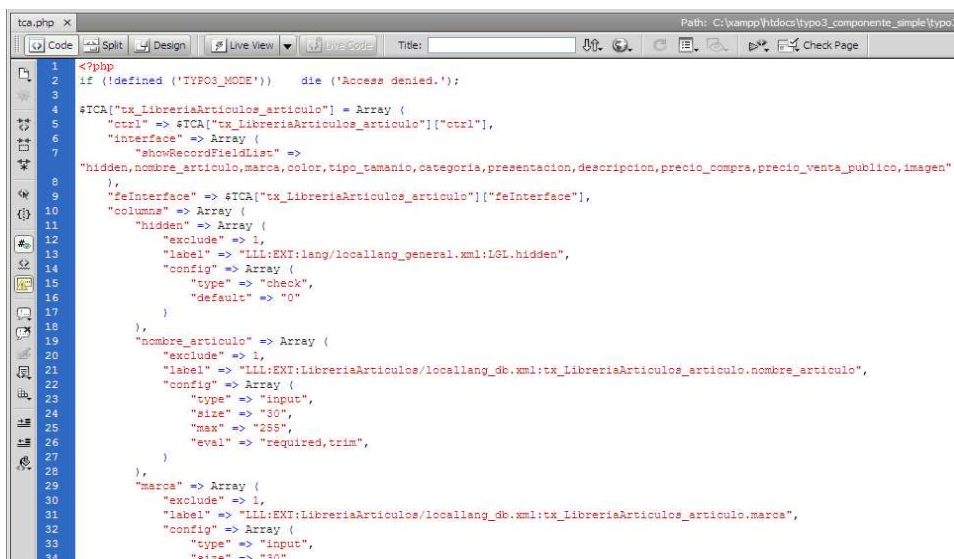


Figura III. 68: Estructura de archivos de la extensión - Módulo 1 Herramienta Typo3 v. 4

Seguidamente ir al archivo ubicado en el directorio typo3conf\ext\LibreriaArticulos\tca.php el cual contiene los campos de nuestra interfaz con su respectivo tipo y dimensiones. Ver Figura III.69.



```
1 <?php
2 if (!defined('TYPO3_MODE')) die ('Access denied.');
```

```
4 $TCA['tx_LibreriaArticulos_articulo'] = Array (
5   "ctrl" => $TCA['tx_LibreriaArticulos_articulo']['ctrl'],
6   "interface" => Array (
7     "showRecordFieldList" =>
8     "hidden,nombre_articulo,marca,color,tipo,tamano,categoria,presentacion,descripcion,precio_compra,precio_venta_publico,imagen"
9   ),
10  "feInterface" => $TCA['tx_LibreriaArticulos_articulo']['feInterface'],
11  "columns" => Array (
12    "hidden" => Array (
13      "exclude" => 1,
14      "label" => "LLL:EXT:lang/locallang_general.xml:LGL:hidden",
15      "config" => Array (
16        "type" => "checkbox",
17        "default" => "0"
18      )
19    ),
20    "nombre_articulo" => Array (
21      "exclude" => 1,
22      "label" => "LLL:EXT:LibreriaArticulos/locallang_db.xml:tx_LibreriaArticulos_articulo.nombre_articulo",
23      "config" => Array (
24        "type" => "input",
25        "size" => "30",
26        "max" => "255",
27        "eval" => "required,trim",
28      )
29    ),
30    "marca" => Array (
31      "exclude" => 1,
32      "label" => "LLL:EXT:LibreriaArticulos/locallang_db.xml:tx_LibreriaArticulos_articulo.marca",
33      "config" => Array (
34        "type" => "input",
35        "size" => "30",
```

Figura III. 69: Archivo tca.php con los datos para nuestra interfaz de usuario - Módulo 1 Herramienta Typo3

Ahora hay que ubicarse en el back-end de la aplicación, se va a la página y se añade un nuevo elemento, en este caso un nuevo artículo de librería. Ver Figura III.70.

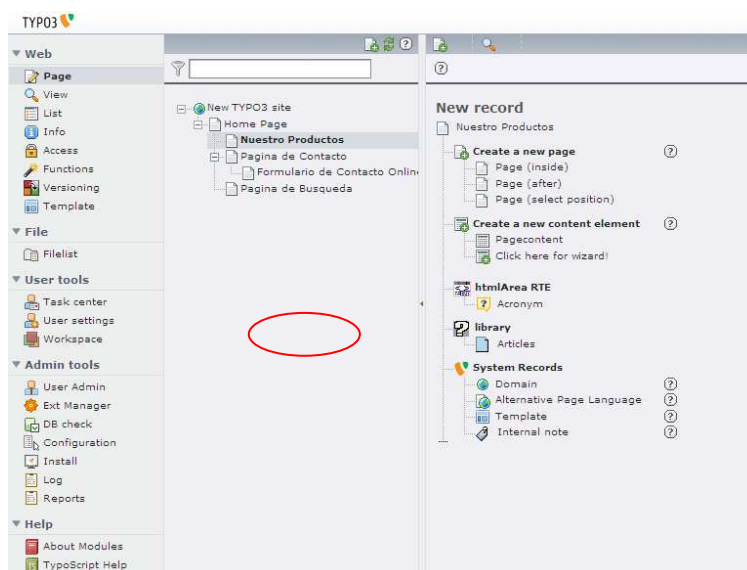


Figura III. 70: Añadir nuevo elemento de artículo de librería - Módulo 1 Herramienta Typo3

Una vez escogido el nuevo elemento de artículo de librería se navega en la interfaz del elemento. Ver Figura III.71.

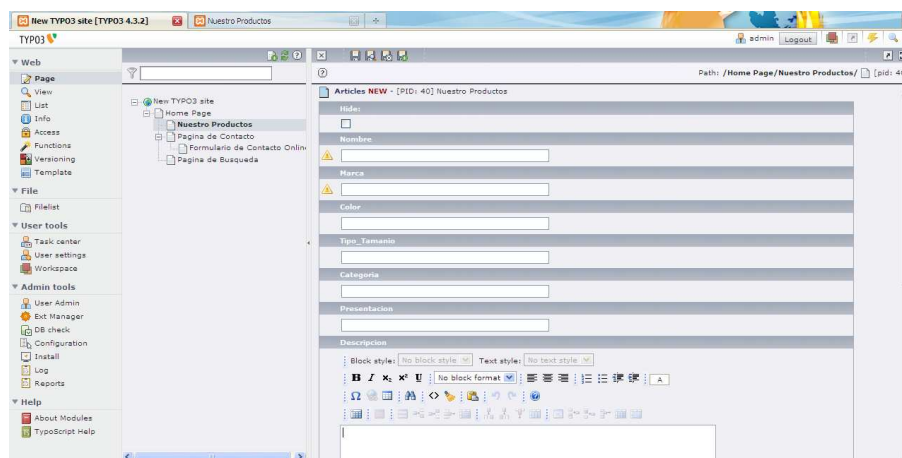


Figura III. 71: Formulario de ingreso de datos de un Artículo de librería - Módulo 1 Herramienta Typo3

3.7.2.2. Módulo 2

Para la creación del presente módulo, de igual manera al anterior se hace uso de kickstarterwizard en el establecimiento de la tabla tx_libreriaarticulos_articulo, con sus respectivos campos. Ver Figura III.72.

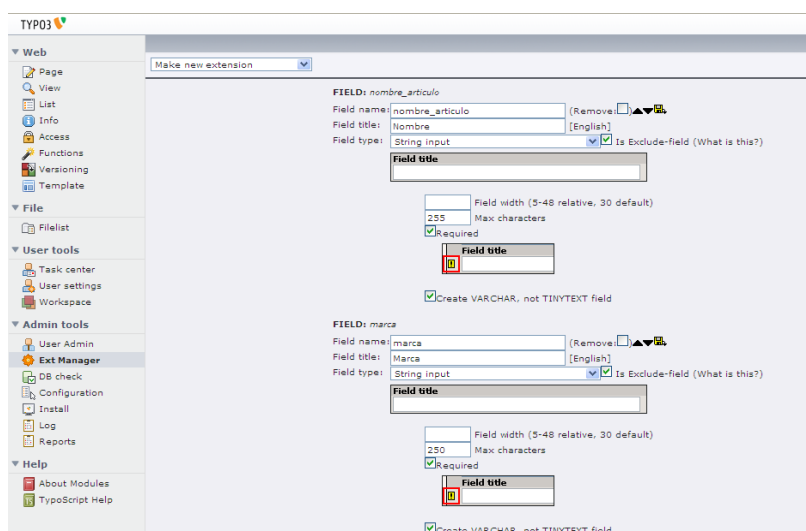


Figura III. 72: Creación de los campos de la tabla tx_libreriaarticulos_articulo con la aplicación kickstarterwizard - Módulo 2 Herramienta Typo3

Para comprobación se observa que la tabla es creada en la base de datos. Ver Figura III.73.

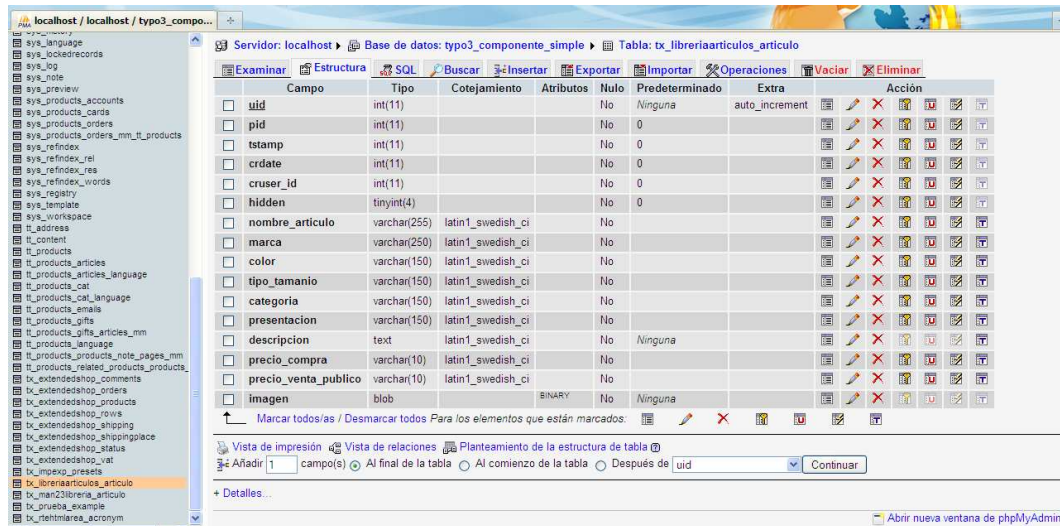


Figura III. 73: Tabla creada en la base de datos, con sus respectivos campos - Módulo 2 Herramienta Typo3

En el archivo typo3conf\ext\LibreriaArticulos\ext_tables.php se encuentra el código necesario para guardar los datos en la tabla tx_libreriaarticulos_articulo. Ver Figura III.74.

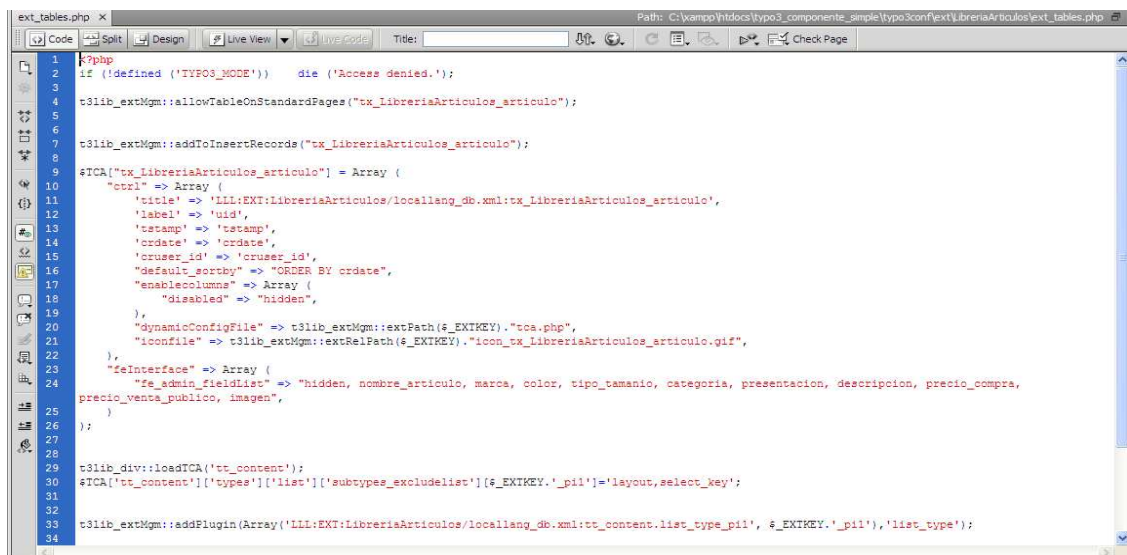


Figura. 74: Código necesario para el ingreso de datos de un artículo en la base de datos en el archivo ext_tables.php - Módulo 2 Herramienta Typo3

La interfaz para que el usuario ingrese un nuevo artículo es simple y amigable, Ver Figura III.75.

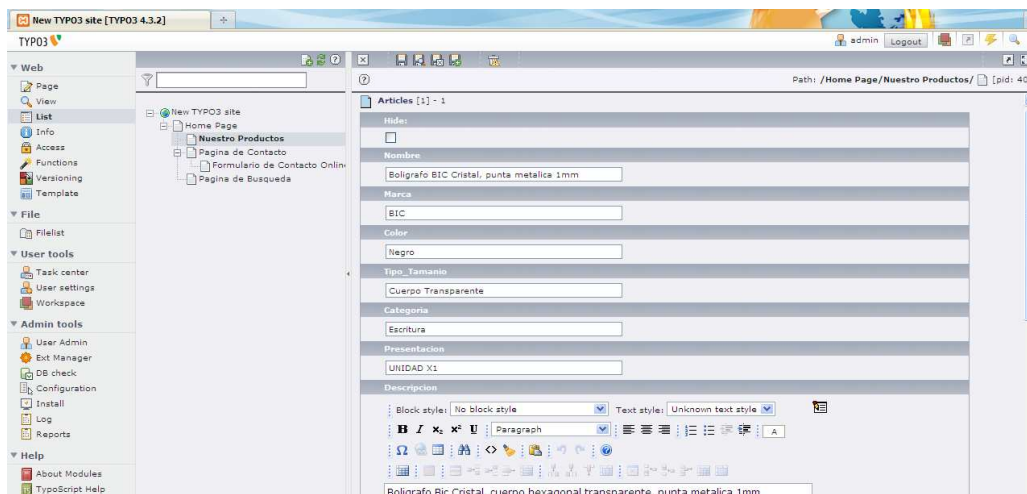


Figura III. 75: Ingreso de datos de un nuevo artículo de librería en el back-end - Módulo 2 Herramienta Typo3

Así como el listado de los artículos que se encuentran ya ingresados. Ver Figura III.76.

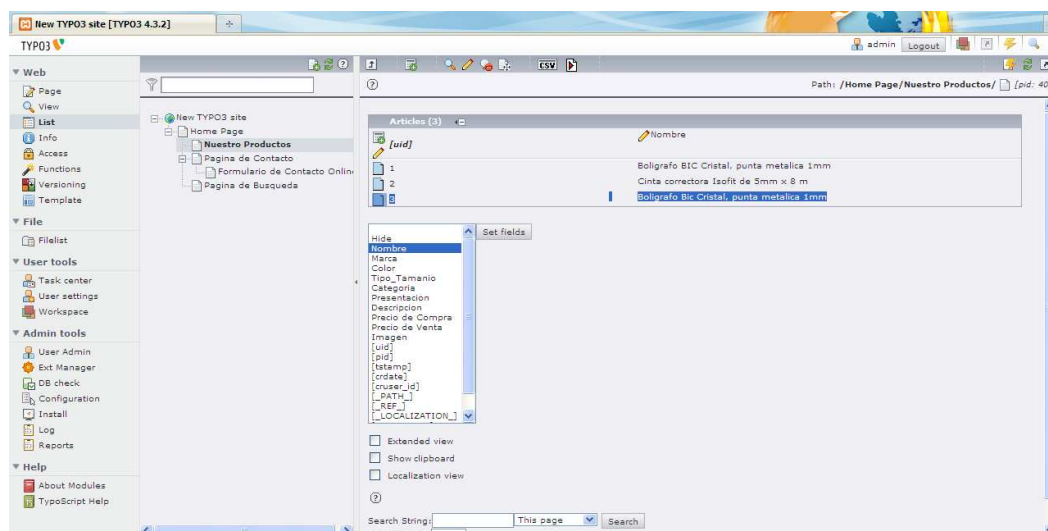


Figura. 76: Visualización del artículo ingresado en el back-end de nuestra aplicación - Módulo 2 Herramienta Typo3

En el archivo ubicado en:

typo3conf\ext\LibreriaArticulos\pi1\class.tx_LibreriaArticulos_pi1.php se encuentra el código necesario para la visualización en el front-end del listado de artículos ingresados en la base de datos. Ver Figura III.77.

```

195 function pi_list_row($c) {
196     $editPanel = $this->pi_getEditPanel();
197     if ($editPanel) $editPanel=<TD>$.editPanel.</TD>;
198
199     return '<tr', ($c%2 ? $this->pi_classParam('listrow-odd') : ''), '>'
200     <td><p>$.this->getFieldContent('uid').</p></td>
201     <td valign="top"><p><h5>$.this->getFieldContent('nombre_articulo').</h5></p></td>
202     <td valign="top"><p><h5>$.this->getFieldContent('marca').</h5></p></td>
203     <td valign="top"><p><h5>$.this->getFieldContent('color').</h5></p></td>
204
205     <td valign="top"><p>$.this->getFieldContent('categoria').</p></td>
206     <!--<td valign="top"><p>$.this->getFieldContent('tipo_tamano').</p></td>
207     <td valign="top"><p>$.this->getFieldContent('presentacion').</p></td>
208     <td valign="top"><p>$.this->getFieldContent('precio_compra').</p></td>
209     <td valign="top"><p>$.this->getFieldContent('precio_venta_publico').</p></td>-->
210     <td valign="center"><p>getFieldContent('imagen')."/"></p></td>
211     </tr>;
212 }
213
214 function pi_list_header() {
215     return '<tr', $this->pi_classParam('listrow-header'), '>'
216     <td><p>$.this->getFieldHeader_sortLink('uid').</p></td>
217     <td><p>$.this->getFieldHeader_sortLink('nombre_articulo').</p></td>
218     <td><p>$.this->getFieldHeader_sortLink('marca').</p></td>
219     <td><p>$.this->getFieldHeader_sortLink('color').</p></td>
220
221     <td><p>$.this->getFieldHeader_sortLink('categoria').</p></td>
222     <!--<td><p>$.this->getFieldHeader_sortLink('tipo_tamano').</p></td><!-- -->
223     <td><p>$.this->getFieldHeader_sortLink('presentacion').</p></td>
224     <td><p>$.this->getFieldHeader_sortLink('precio_compra').</p></td>
225     <td><p>$.this->getFieldHeader_sortLink('precio_venta_publico').</p></td>
226     <td nowrap><p>$.this->getFieldHeader('imagen').</p></td>-->
227     </tr>;
228 }
229

```

Figura. 77: Archivo class.tx_LibreriaArticulos_pi1.php, con el código para visualizar el listado de artículos en el front-end - Módulo 2 Herramienta Typo3

La visualización del listado de los artículos se da de forma prolija y amigable. Ver Figura III.78.

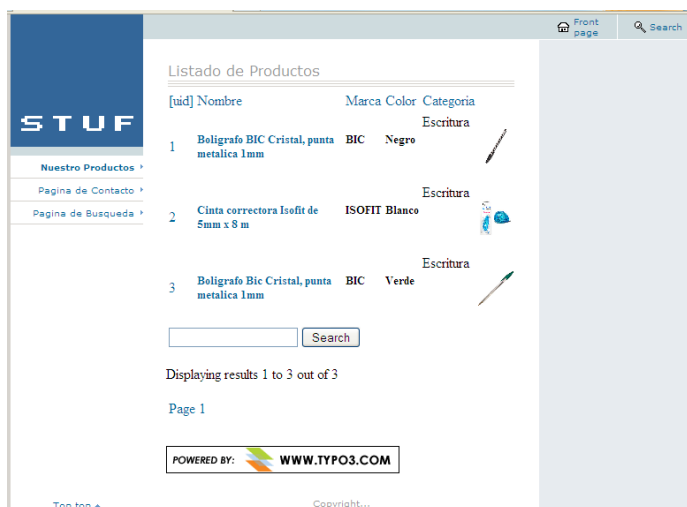
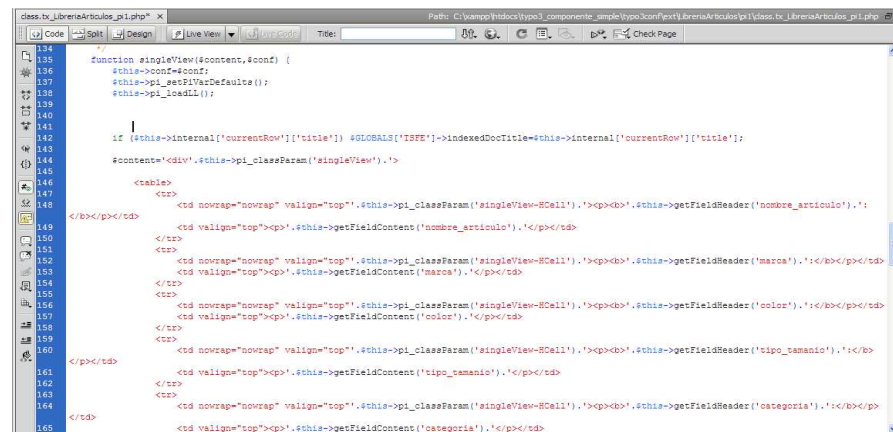


Figura III. 78: Visualización del listado de artículos en el Front-End - Módulo 2 Herramienta Typo3

En el archivo ubicado en:

typo3conf\ext\LibreriaArticulos\pi1\class.tx_LibreriaArticulos_pi1.php se encuentra el código necesario para la visualización en el front-end de un artículo específico. Ver Figura III.79.



```
134 function singleView($content,$conf) {
135     $this->pi_classParam($conf);
136     $this->pi_getPIVarDefaults();
137     $this->pi_loadLL();
138
139     if ($this->internal['currentRow']['title'] $GLOBALS['TSFE']->indexedDocTitle=$this->internal['currentRow']['title'];
140     $content=<div'.'.$this->pi_classParam('singleView').>
141     <table>
142     <tr>
143     <td nowrap="nowrap" valign="top".'.$this->pi_classParam('singleView-Header').><p><b>.$this->getFieldHeader('nombre_articulo').</p></td>
144     <td valign="top"><p>.$this->getFieldContent('nombre_articulo').</p></td>
145     </tr>
146     <tr>
147     <td nowrap="nowrap" valign="top".'.$this->pi_classParam('singleView-Header').><p><b>.$this->getFieldHeader('marca').</p></td>
148     <td valign="top"><p>.$this->getFieldContent('marca').</p></td>
149     </tr>
150     <tr>
151     <td nowrap="nowrap" valign="top".'.$this->pi_classParam('singleView-Header').><p><b>.$this->getFieldHeader('color').</p></td>
152     <td valign="top"><p>.$this->getFieldContent('color').</p></td>
153     </tr>
154     <tr>
155     <td nowrap="nowrap" valign="top".'.$this->pi_classParam('singleView-Header').><p><b>.$this->getFieldHeader('tipo_tamano').</p></td>
156     <td valign="top"><p>.$this->getFieldContent('tipo_tamano').</p></td>
157     </tr>
158     <tr>
159     <td nowrap="nowrap" valign="top".'.$this->pi_classParam('singleView-Header').><p><b>.$this->getFieldHeader('categoria').</p></td>
160     <td valign="top"><p>.$this->getFieldContent('categoria').</p></td>
161     </tr>
162     </table>
163     </div>
164     </td>
165 }
```

Figura III. 79: Archivo class.tx_LibreriaArticulos_pi1.php, con el código para visualizar un artículo específico en el front-end - Módulo 2 Herramienta Typo3

La visualización del listado de los artículos se da de forma prolija y amigable. Ver Figura III.80.

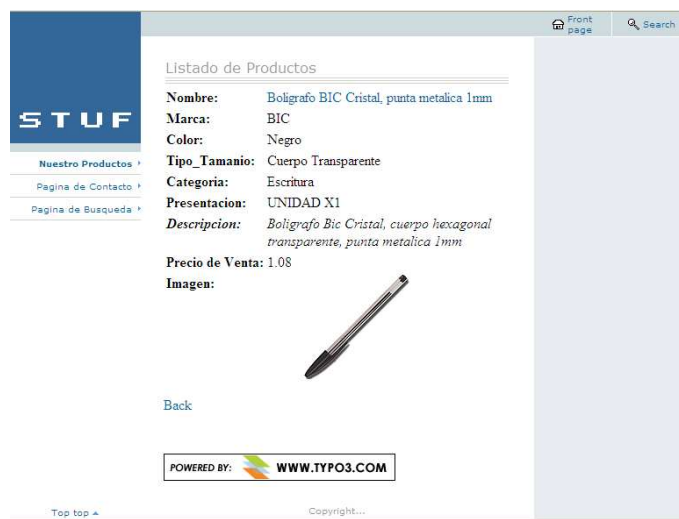


Figura III. 80: Visualización de un artículo específico en el Front-End - Módulo 2 Herramienta Typo3

3.7.2.3. Módulo 3

Para este módulo, se expandirá el componente haciendo uso de kickstarterwizard, se añaden 4 tablas más a la base de datos, la tabla tx_libreriaarticulos_categoria, tx_libreriaarticulos_subcategoria, tx_libreriaarticulos_proveedor y la tabla tx_libreria_articuloproveedor. Y a la vez se modifica la estructura de la tabla tx_libreriaarticulos_articulo, eliminando el campo categoría para tenerla en una tabla aparte, con su respectiva sub categoría. Para la demostración se muestra la creación de la tabla tx_libreriaarticulos_categoria. Ver Figura 3.76.

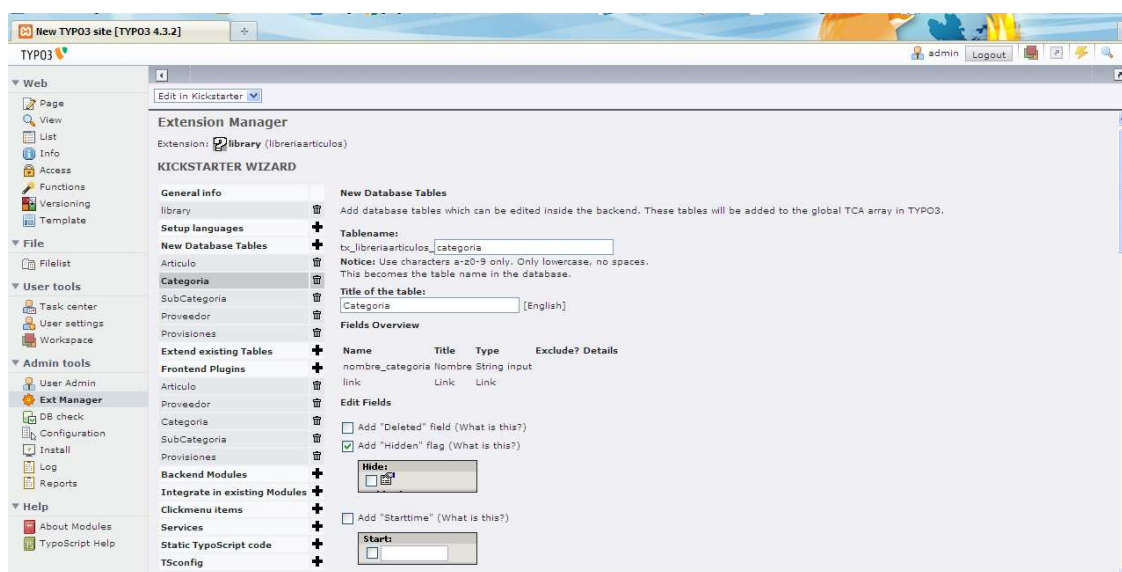


Figura III. 81: Creación de los campos de la tabla tx_libreriaarticulos_categoria con la aplicación kickstarterwizard - Módulo 3 Herramienta Typo3

Continuando con la demostración del componente expandido, aquí se presenta la tabla creada en la base de datos. Ver Figura III.82.

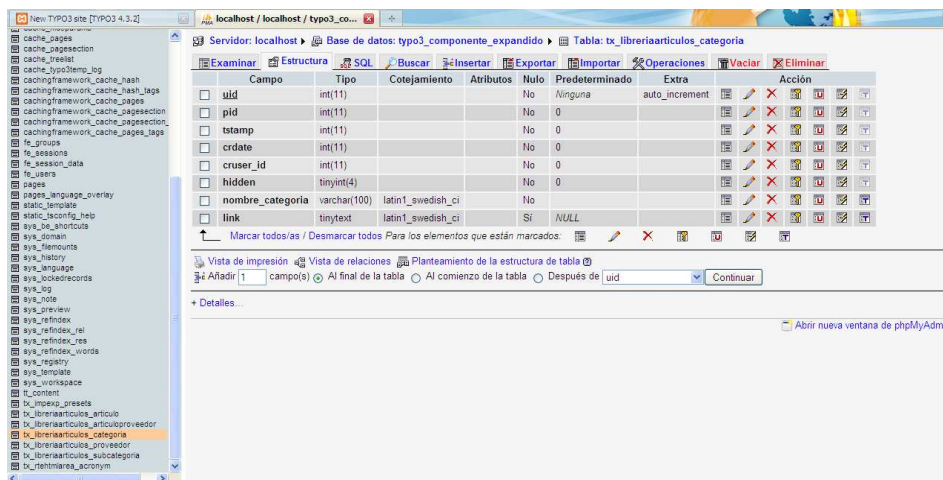


Figura III. 82: Tabla creada en la base de datos, con sus respectivos campos - Módulo 3 Herramienta Typo3

En el archivo typo3conf\ext\LibreriaArticulos\ext_tables.php se encuentra el código necesario para guardar los datos de todas las tablas creadas anteriormente, tx_libreriaarticulos_categoria, tx_libreriaarticulos_subcategoria, tx_libreriaarticulos_articulo, tx_libreriaarticulos_proveedor, tx_libreriaarticulos_articulosproveedor. En el archivo ext_tables.php se encuentra el código para el ingreso de los datos de todas las tablas creadas. Ver Figura III.83.



Figura III. 83: Código necesario para el ingreso de datos de un artículo en la base de datos en el archivo ext_tables.php - Módulo 3 Herramienta Typo3

Ahora se va a ingresar datos a cada una de las tablas mediante el formulario de ingreso de cada una de las tablas en el back-end. Aquí se presenta la demostración el ingreso de los datos respectivos de una nueva categoría. Ver Figura III.84.

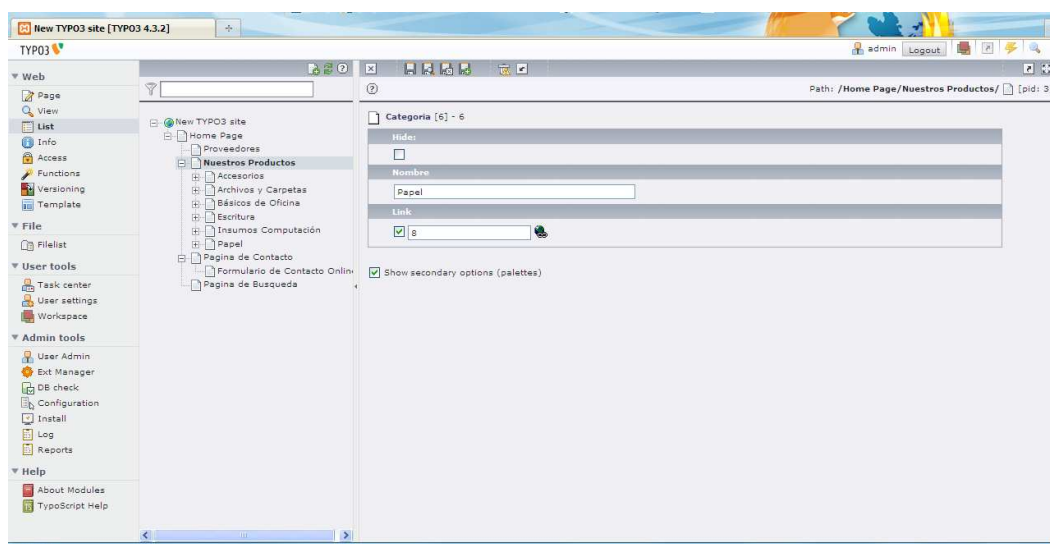


Figura III. 84: Ingreso de datos de una nueva categoría en el back-end - Módulo 3 Herramienta Typo3

Una vez ingresado la categoría deseada, se mostrara el listado de las categorías existentes. Ver Figura III.85.

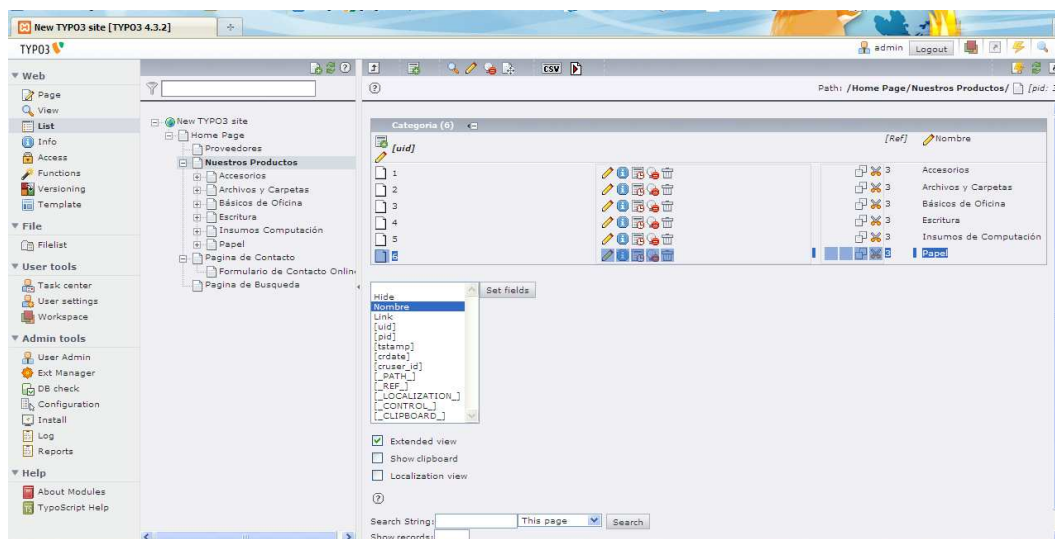


Figura III. 85: Visualización de la categoría ingresada en el back-end de nuestra aplicación - Módulo 3 Herramienta Typo3

En el archivo ubicado en `typo3conf\ext\LibreriaArticulos\pi3\class.tx_LibreriaArticulos_pi3.php` se encuentra el código necesario para la visualización en el front-end del listado de categorías disponibles ingresadas en la base de datos. Ver Figura III.86.



```
181 function pi_list_row($c) {
182     $editPanel = $this->pi_getEditPanel();
183     if ($editPanel) $editPanel='<TD>'. $editPanel.'</TD>';
184
185     return '<tr>.{#42 ? $this->pi_classParam('listrow-odd') : ''}.>
186     <!--<td><p>'. $this->getFieldContent('uid'). '</p></td-->
187     <td valign="top"><p>'. $this->getFieldContent('nombre_categoria'). '</p></td>
188     </td>';
189 }
190
191 function pi_list_header() {
192     return '<tr>.'. $this->pi_classParam('listrow-header'). '>
193     <!--<td><p>'. $this->getFieldHeader_sortLink('uid'). '</p></td>
194     <td><p>'. $this->getFieldHeader_sortLink('nombre_categoria'). '</p></td-->
195     </tr>';
196 }
197
198 function getFieldContent($FN) {
199     switch($FN) {
200         case 'uid':
201             return $this->pi_list_linkSingle($this->internal['currentRow'][$FN], $this->internal['currentRow']['uid'], 1);
202             break;
203         case 'nombre_categoria':
204             return '<a href=index.php?id='.$this->internal['currentRow']['link'].'>'. $this->internal['currentRow']['nombre_categoria'].'</a>';
205             break;
206         default:
207             return $this->internal['currentRow'][$FN];
208             break;
209     }
210 }
211 }
212 }
213 }
```

Figura. 86: Archivo `class.tx_LibreriaArticulos_pi3.php`, con el código para visualizar el listado de categorías en el front-end - Módulo 3 Herramienta Typo3

El cual presenta los datos requeridos en pantalla en el front-end. Ver Figura III.87.

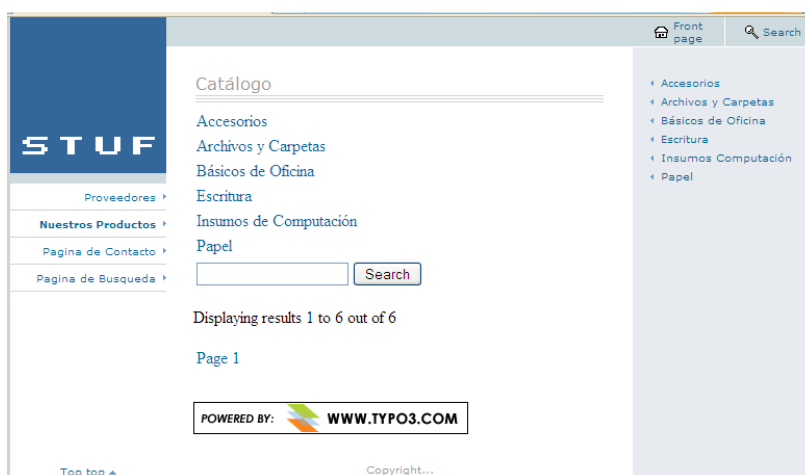


Figura III. 87: Visualización del listado de categorías en el Front-End - Módulo 3 Herramienta Typo3

Cada visualización de la información requerida tiene su propio código que supervisa el diseño de presentación de los datos en pantalla en el front-end, como se muestra en las dos figuras anteriores.

3.7.2.4. Módulo 4

En este módulo, se va a estudiar las facilidades que TYPO3 ofrece para la administración de las extensiones o componentes, su instalación, desinstalación, habilitación, des habilitación, entre otros. Ver Figura III.88.

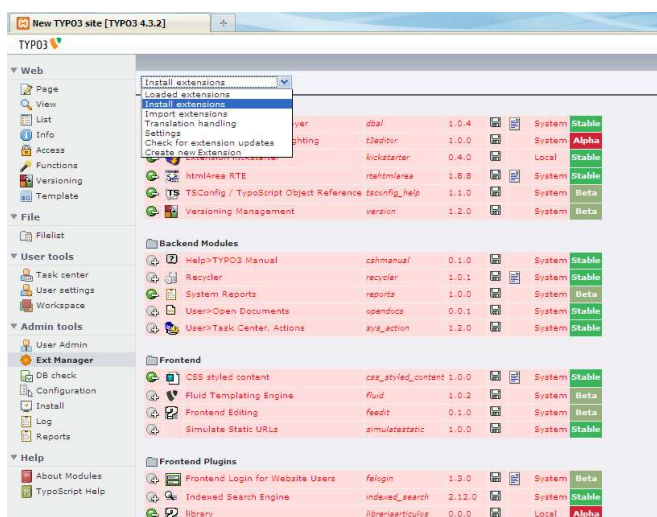


Figura III. 88: Menú para la administración de extensiones - Módulo 4 Herramienta Typo3

Para la instalación de un componente Typo3 brinda interfaces sin complicaciones, ver Figura III.89.

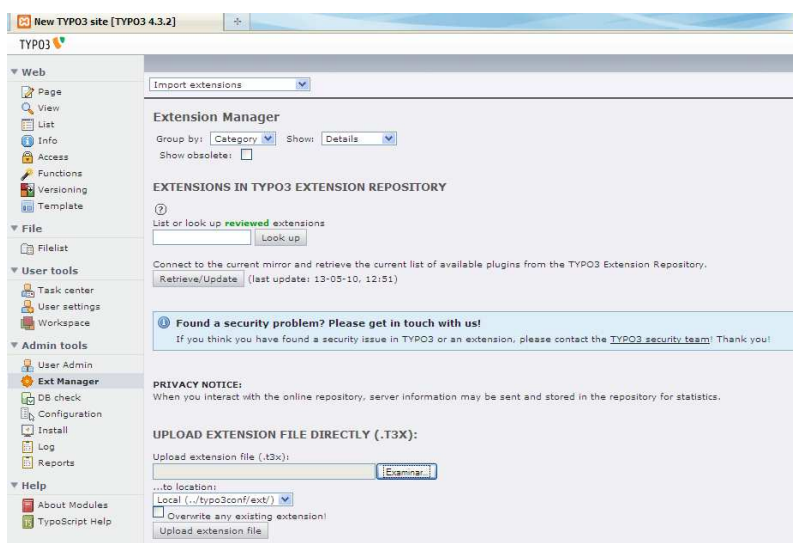


Figura III. 89: Instalación de extensiones - Módulo 4 Herramienta Typo3

Typo 3 permite buscar el componente en forma local o en una dirección web, ver Figura III.90.

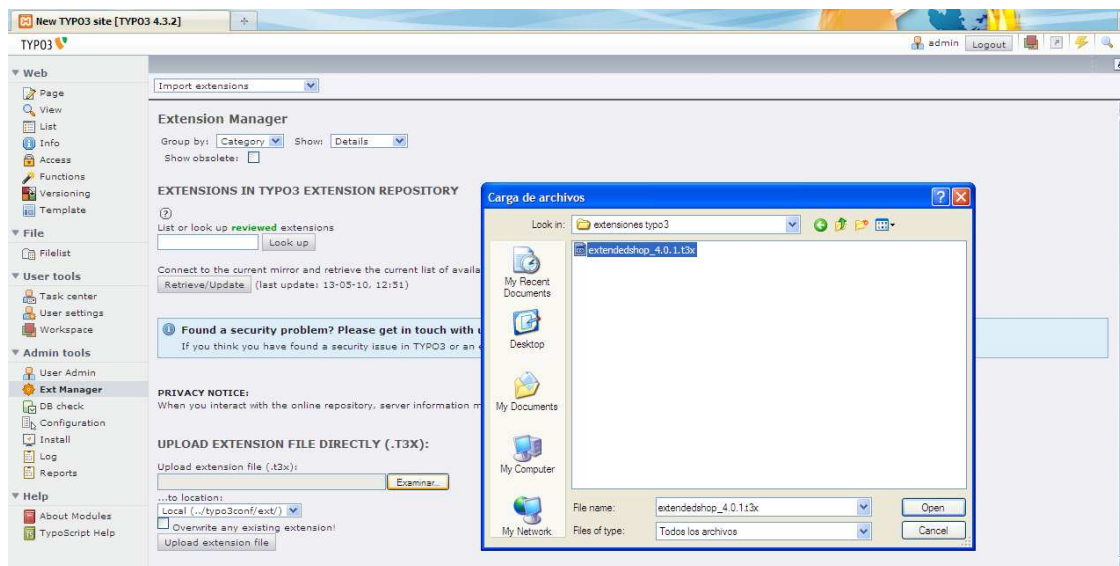


Figura. 90: Búsqueda del componente a instalar - Módulo 4 Herramienta Typo3

Si el procedimiento se realizó correctamente, y el componente a instalar contiene todos los archivos necesarios de instalación, se mostrará en pantalla un mensaje de éxito en la instalación. Ver Figura III.91.

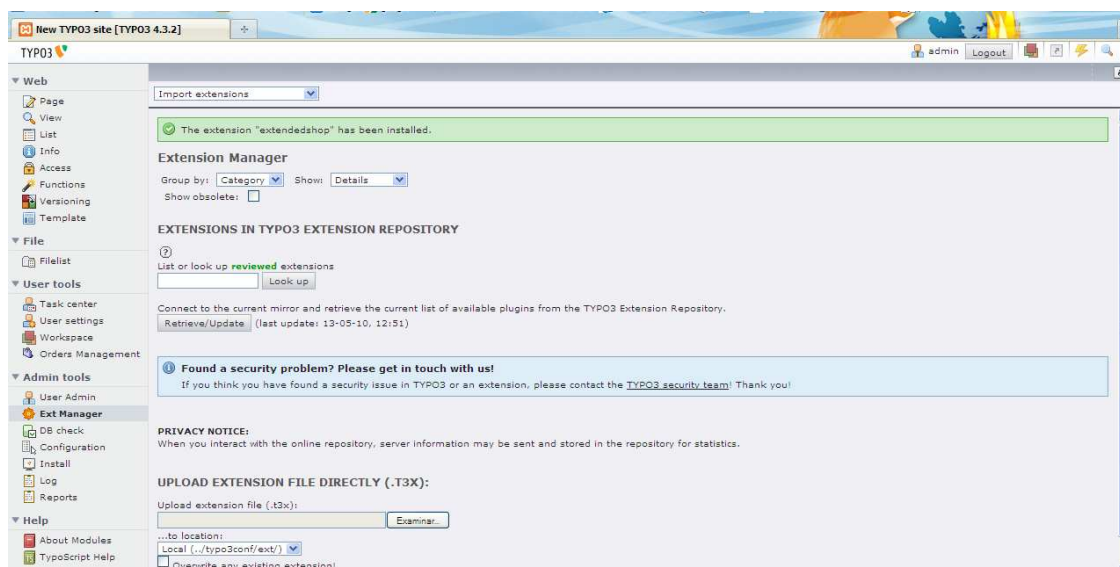


Figura III. 91: Extensión instalada correctamente- Módulo 4 Herramienta Typo3

Para la desinstalación de un componente, Typo3 muestra un listado de los mismos, si se desea desinstalarlo completamente se lo ubica y se da clic en la figura de una esfera verde con un signo "-" sobre esta. Ver Figura III.92.

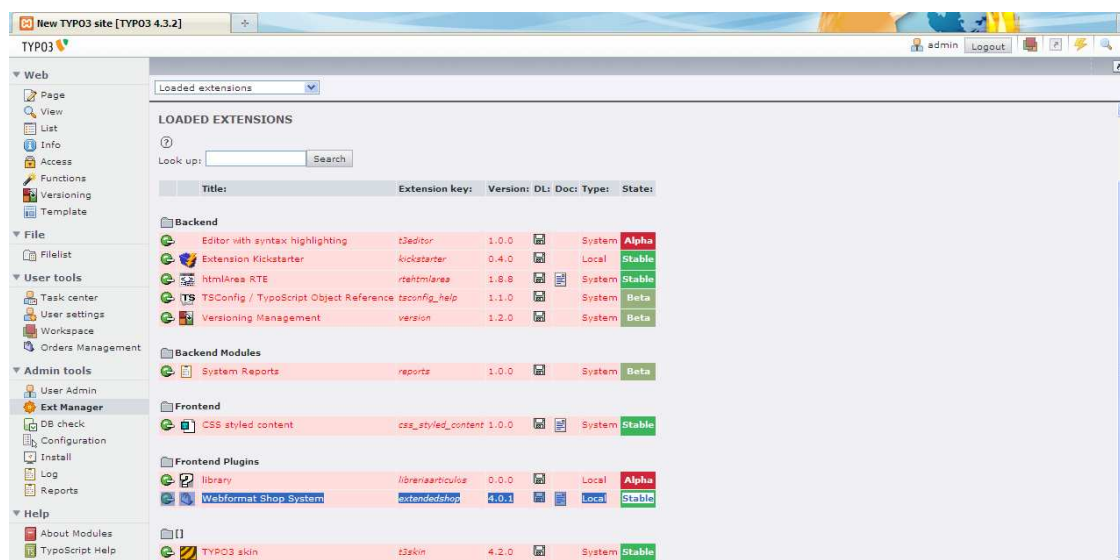


Figura. 92: Gestión de componentes, desinstalación - Módulo 4 Herramienta Typo3

Si el proceso de desinstalación se realizó completamente se mostrará un mensaje de éxito de la desinstalación. Ver Figura III.93.

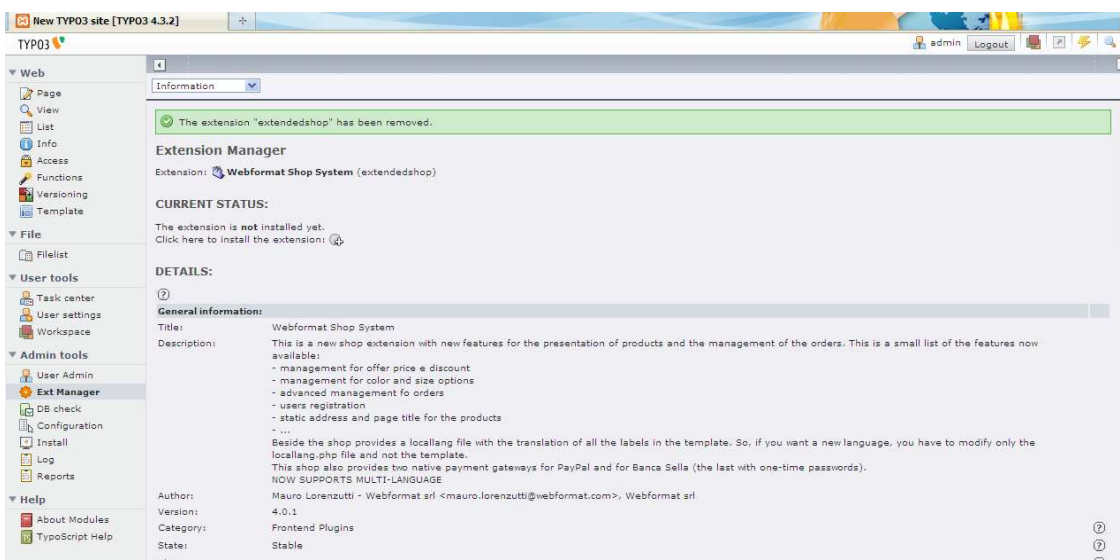


Figura III. 93: Desinstalación exitosa de una extensión - Módulo 4 Herramienta Typo3

3.7.3. Desarrollo de los Módulos en la Herramienta Drupal 6.19

3.7.3.1. Módulo 1

Para la construcción del módulo 1 se crea un directorio llamado lib_mod1 en drupal\sites\all\modules, Figura III.94, en este directorio se va a almacenar los archivos necesarios para el funcionamiento del módulo con el que se va a trabajar.

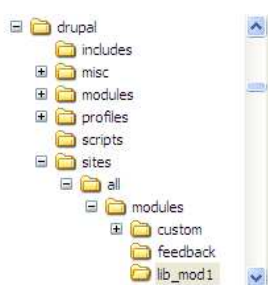


Figura III. 94: Creación del directorio lib_mod1 – Módulo 1 Herramienta Drupal

En el modelo de desarrollo de módulos en el CMS Drupal se requiere de tres archivos, como se ve la Figura III.95, dentro de los cuales encontramos:

.info: Contiene la descripción del modulo, incluyendo nombre, versión, paquetes y core requerido.

.install: Contiene el schema de las tablas que el modulo va a hacer uso durante su funcionamiento.

.module: Contiene la lógica y el diseño que se mostrara al usuario dentro del CMS Drupal.

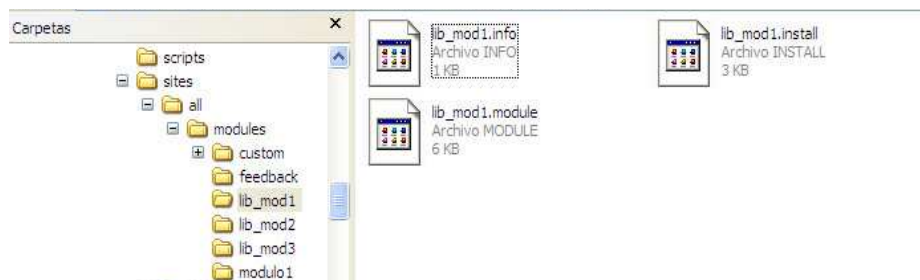
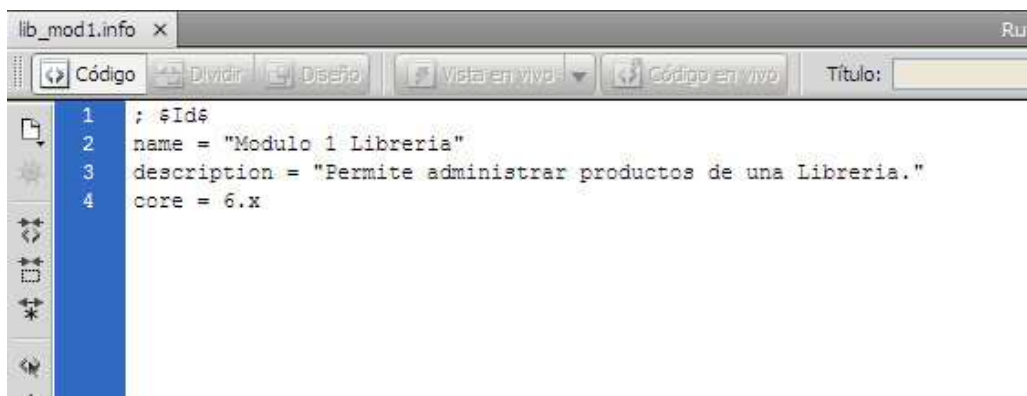


Figura III. 95: Archivos necesarios para la creación del Módulo 1 Herramienta Drupal

Se inicia con el desarrollo del archivo donde se describe al módulo en desarrollo para este caso **lib_mod1.info**, ver Figura III.96.

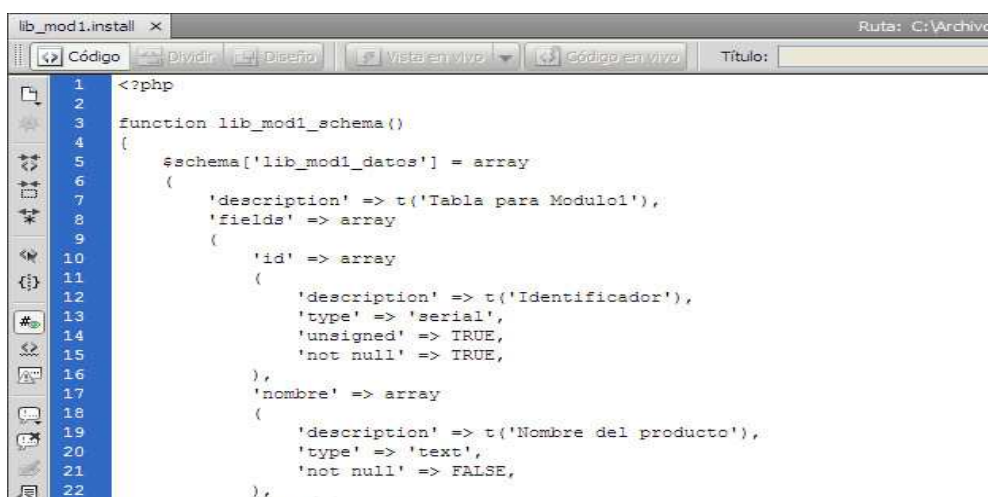


```
lib_mod1.info x Ruta
Código Dividir Diseño Vista en vivo Código en vivo Título:
1 ; $Id$
2 name = "Modulo 1 Libreria"
3 description = "Permite administrar productos de una Libreria."
4 core = 6.x
```

Figura III. 96: Contenido del archivo lib_mod1.info – Módulo 1 Herramienta Drupal

Aquí se describe un identificador (\$Id\$) en el caso de que el CMS esté utilizando un sistema de versiones para el control de módulos, además el nombre, la descripción y el core requerido para la ejecución de este módulo, en esta ocasión el core que se indica es la versión 6 o superior.

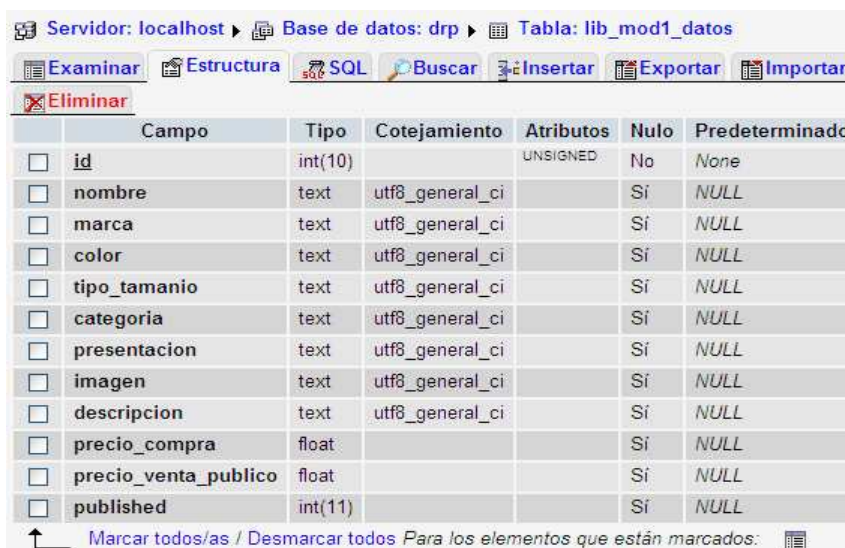
En el archivo **lib_mod1.install**, como se ve en la Figura III.97, se define el schema de la tabla que se creará en la base de datos, al momento de activar el módulo, Drupal instalará las tablas necesarias las cuales están definidas en el siguiente archivo.



```
lib_mod1.install x Ruta: C:\Archivos
Código Dividir Diseño Vista en vivo Código en vivo Título:
1 <?php
2
3 function lib_mod1_schema()
4 {
5     $schema['lib_mod1_datos'] = array
6     (
7         'description' => t('Tabla para Modulo1'),
8         'fields' => array
9         (
10            'id' => array
11            (
12                'description' => t('Identificador'),
13                'type' => 'serial',
14                'unsigned' => TRUE,
15                'not null' => TRUE,
16            ),
17            'nombre' => array
18            (
19                'description' => t('Nombre del producto'),
20                'type' => 'text',
21                'not null' => FALSE,
22            ),
23        )
24     )
25 }
```

Figura III. 97: Contenido del archivo lib_mod1.install – Módulo 1 Herramienta Drupal

Este schema creará una nueva tabla llamada **lib_mod1_datos**, como se ve en la Figura III.98, en la base de datos en MySQL.

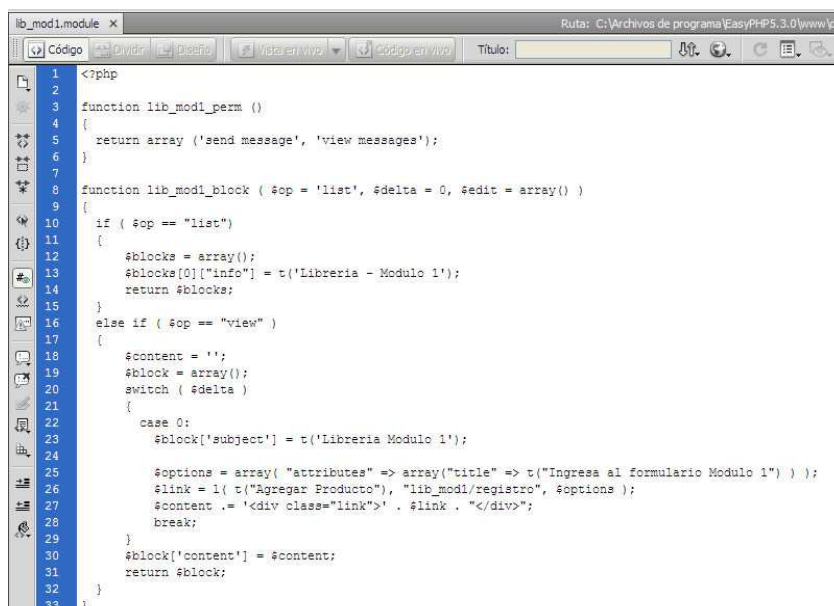


The screenshot shows a MySQL database interface with the following table schema:

	Campo	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado
<input type="checkbox"/>	id	int(10)		UNSIGNED	No	None
<input type="checkbox"/>	nombre	text	utf8_general_ci		Sí	NULL
<input type="checkbox"/>	marca	text	utf8_general_ci		Sí	NULL
<input type="checkbox"/>	color	text	utf8_general_ci		Sí	NULL
<input type="checkbox"/>	tipo_tamano	text	utf8_general_ci		Sí	NULL
<input type="checkbox"/>	categoria	text	utf8_general_ci		Sí	NULL
<input type="checkbox"/>	presentacion	text	utf8_general_ci		Sí	NULL
<input type="checkbox"/>	imagen	text	utf8_general_ci		Sí	NULL
<input type="checkbox"/>	descripcion	text	utf8_general_ci		Sí	NULL
<input type="checkbox"/>	precio_compra	float			Sí	NULL
<input type="checkbox"/>	precio_venta_publico	float			Sí	NULL
<input type="checkbox"/>	published	int(11)			Sí	NULL

Figura III. 98: Contenido de la tabla lib_mod1_datos – Módulo 1 Herramienta drupal

En el archivo **lib_mod1.module**, como se ve en la Figura III.99, se describe la lógica y la vista del módulo, aquí por medio de hooks (funciones predefinidas en el core de Drupal) se indica como el CMS debe procesar cada uno de los requerimientos del módulo.



```
1 <?php
2
3 function lib_mod1_perm ()
4 {
5     return array ('send message', 'view messages');
6 }
7
8 function lib_mod1_block ( $op = 'list', $delta = 0, $edit = array() )
9 {
10     if ( $op == "list" )
11     {
12         $blocks = array();
13         $blocks[0]['info'] = t('Libreria - Modulo 1');
14         return $blocks;
15     }
16     else if ( $op == "view" )
17     {
18         $content = '';
19         $block = array();
20         switch ( $delta )
21         {
22             case 0:
23                 $block['subject'] = t('Libreria Modulo 1');
24
25                 $options = array( "attributes" => array("title" => t("Ingresa al formulario Modulo 1") ) );
26                 $link = l( t("Agregar Producto"), "lib_mod1/registro", $options );
27                 $content .= '<div class="link">' . $link . "</div>";
28                 break;
29             }
30         $block['content'] = $content;
31         return $block;
32     }
33 }
```

Figura III. 99: Contenido del archivo lib_mod1.module – Módulo 1 Herramienta drupal

A continuación realizar la ejecución del módulo, ingresar al area administrativa de Drupal, *Administer – SiteBuilding – Modules*, buscar el módulo en el grupo *Other*, activar y guardar la configuración, ver Figura III.100.



Figura III. 100: Activación del Módulo 1 Herramienta Drupal

En este momento está activado el módulo, pero aún no se indica en que parte del sitio aparecerá, para esto ir al área administrativa, *Administer – SiteBuilding – Blocks*, buscar el módulo en el área *Disable* y arrastrarlo al bloque *Rightsidebar*, y guardar la configuración, ver Figura III.101.

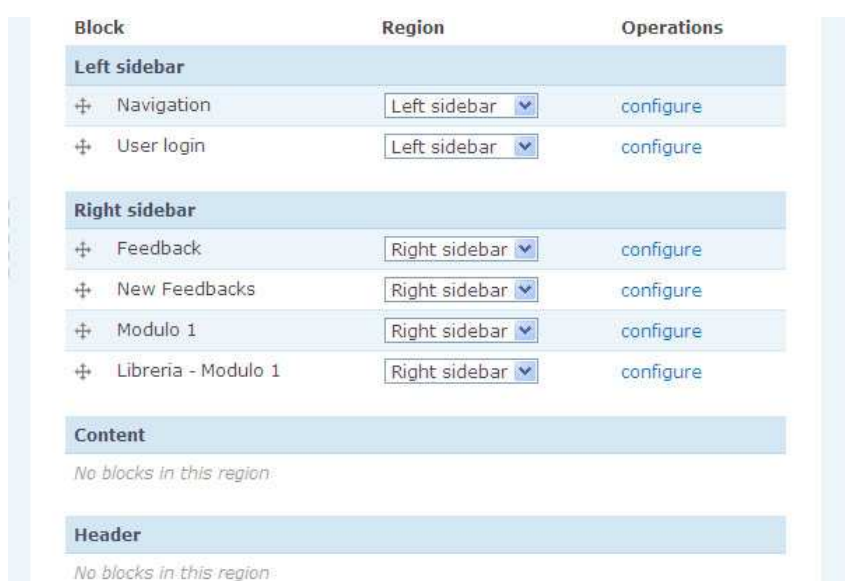


Figura III. 101: Lugar de la visualización del Módulo 1 Herramienta Drupal

En la interfaz de Drupal ya aparece el módulo Librería Modulo 1 en el bloque derecho, ver Figura III.102.



Figura III. 102: Librería Modulo 1 - Módulo 1 Heramienta Drupal

En la Figura III.103 está la ejecución del primer módulo en el cual se observa el formulario de ingreso de un nuevo artículo.

The image displays the 'Agregar Producto' form in a Drupal interface. On the left side, there is a search bar and an 'admin' menu with links for Contact, My account, Create content, Administer, and Log out. The main form area contains the following fields: 'Nombre:' (text input), 'Marca:' (text input), 'Color:' (text input), 'Tipo-Tamaño:' (text input), 'Categoría:' (dropdown menu with 'Papel' selected), 'Presentación:' (text input), 'Imagen:' (file selection button labeled 'Seleccionar archivo' and the text 'No se ha ...n archivo'), 'Descripción:' (text area), 'Precio Compra:' (text input), 'Precio de Venta al Público:' (text input), and 'Publicado:' (radio buttons for 'Si' and 'No'). A 'Submit' button is located at the bottom of the form. On the right side, there is a sidebar with 'Feedback' (New Feedback), 'Feedback Messages' (There are 1 new feeds), 'Modulo 1' (Entrar), and 'Librería Modulo 1' (Agregar Producto).

Figura. 103: Formulario de ingreso de un nuevo artículo - Módulo 1 Herramienta Drupal

3.7.3.2. Módulo 2:

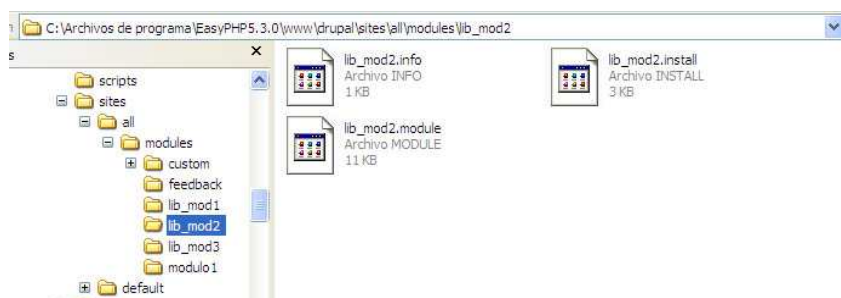


Figura III. 104: Archivos necesarios para la creación del Módulo 2 Herramienta Drupal

En este módulo se hace uso de la lógica creada en el módulo 1 pero se agregará otra opción al menú lateral para poder listar todos los artículos ingresados, ver Figura III.104 y III.105.



Figura III. 105: Librería Modulo 2 - Módulo 2 Herramienta Drupal

Se inicia con el desarrollo del archivo donde se describe al módulo en desarrollo para este caso **lib_mod2.info**, ver Figura III.106.

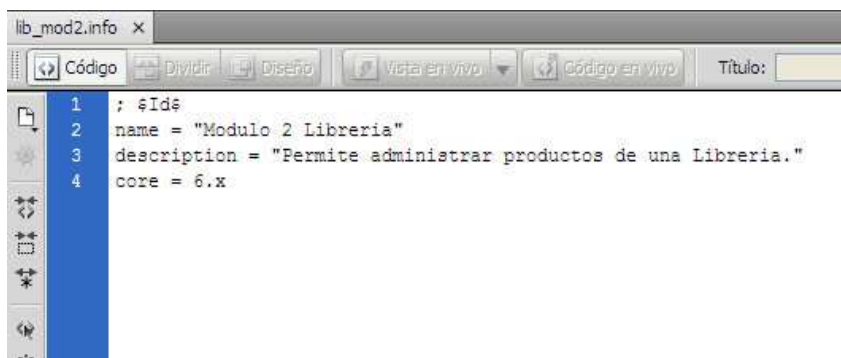
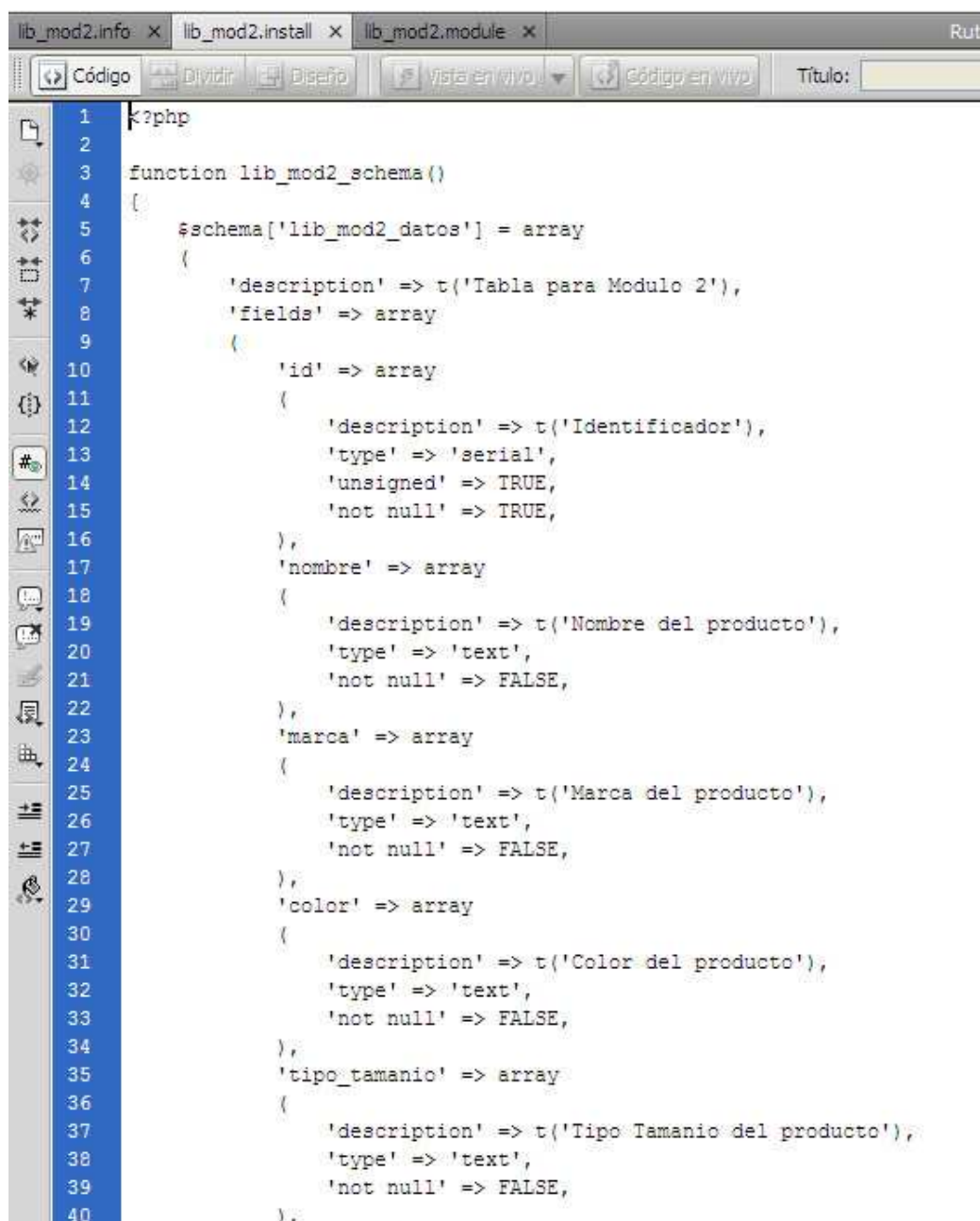


Figura. 106: Contenido del archivo lib_mod2.info - Módulo 2 Herramienta Drupal

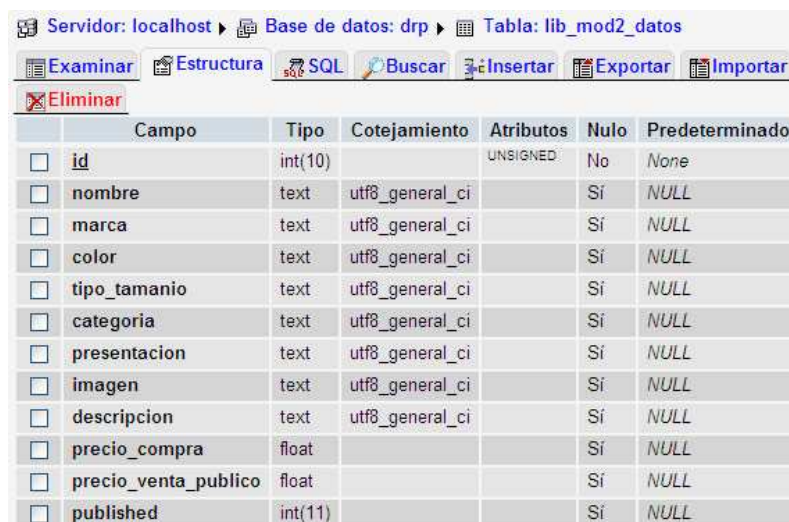
En el archivo **lib_mod2.install**, ver Figura III.108, se define el schema de la tabla que se crea en la base de datos, al momento de activar el módulo, en este schema se utiliza el mismo modelo descrito para el módulo 1, con esto se mantiene la tabla.



```
1 <?php
2
3 function lib_mod2_schema()
4 {
5     $schema['lib_mod2_datos'] = array
6     (
7         'description' => t('Tabla para Modulo 2'),
8         'fields' => array
9         (
10            'id' => array
11            (
12                'description' => t('Identificador'),
13                'type' => 'serial',
14                'unsigned' => TRUE,
15                'not null' => TRUE,
16            ),
17            'nombre' => array
18            (
19                'description' => t('Nombre del producto'),
20                'type' => 'text',
21                'not null' => FALSE,
22            ),
23            'marca' => array
24            (
25                'description' => t('Marca del producto'),
26                'type' => 'text',
27                'not null' => FALSE,
28            ),
29            'color' => array
30            (
31                'description' => t('Color del producto'),
32                'type' => 'text',
33                'not null' => FALSE,
34            ),
35            'tipo_tamano' => array
36            (
37                'description' => t('Tipo Tamano del producto'),
38                'type' => 'text',
39                'not null' => FALSE,
40            ),
41        ),
42    );
43 }
```

Figura III. 107: Contenido del archivo lib_mod2.install - Módulo 2 Herramienta Drupal

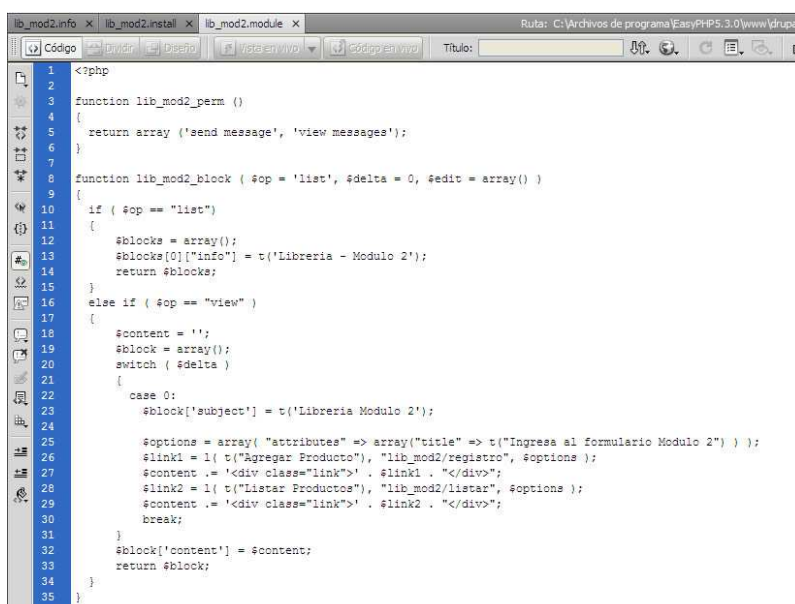
Este schema creará una nueva tabla llamada lib_mod2_datos en la base de datosdrupal en MySQL, ver Figura III.108.



	Campo	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado
<input type="checkbox"/>	id	int(10)		UNSIGNED	No	None
<input type="checkbox"/>	nombre	text	utf8_general_ci		Si	NULL
<input type="checkbox"/>	marca	text	utf8_general_ci		Si	NULL
<input type="checkbox"/>	color	text	utf8_general_ci		Si	NULL
<input type="checkbox"/>	tipo_tamano	text	utf8_general_ci		Si	NULL
<input type="checkbox"/>	categoria	text	utf8_general_ci		Si	NULL
<input type="checkbox"/>	presentacion	text	utf8_general_ci		Si	NULL
<input type="checkbox"/>	imagen	text	utf8_general_ci		Si	NULL
<input type="checkbox"/>	descripcion	text	utf8_general_ci		Si	NULL
<input type="checkbox"/>	precio_compra	float			Si	NULL
<input type="checkbox"/>	precio_venta_publico	float			Si	NULL
<input type="checkbox"/>	published	int(11)			Si	NULL

Figura III. 108: Contenido de la tabla lib_mod2_datos - Módulo 2 Herramienta Drupal

En el archivo **lib_mod2.module**, ver Figura III.109, se describe la lógica y la vista del módulo 2, se toma como base la estructura del módulo 1, en este caso se aumenta en la descripción del hook del menú y las funciones necesarias para procesar los nuevos formularios, este archivo cuenta con nuevos métodos para mostrar en forma de tabla la lista de artículos.



```
<?php
function lib_mod2_perm ()
{
    return array ('send message', 'view messages');
}

function lib_mod2_block ( $op = 'list', $delta = 0, $edit = array() )
{
    if ( $op == "list" )
    {
        $blocks = array();
        $blocks[0]['info'] = t('Libreria - Modulo 2');
        return $blocks;
    }
    else if ( $op == "view" )
    {
        $content = '';
        $block = array();
        switch ( $delta )
        {
            case 0:
                $block['subject'] = t('Libreria Modulo 2');

                $options = array( "attributes" => array("title" => t("Ingresa al formulario Modulo 2") ) );
                $link1 = l( t("Agregar Producto"), "lib_mod2/registro", $options );
                $content .= '<div class="link">' . $link1 . "</div>";
                $link2 = l( t("Listar Productos"), "lib_mod2/listar", $options );
                $content .= '<div class="link">' . $link2 . "</div>";
                break;
        }
        $block['content'] = $content;
        return $block;
    }
}
```

Figura III. 109: Contenido del archivo lib_mod2.module - Módulo 2 Herramienta Drupal

A continuación se realiza la ejecución del módulo, ingresar al área administrativa de Drupal, *Administer - SiteBuilding - Modules*, buscar el módulo en el grupo *Other*, activar y guardar la configuración, ver Figura III.110.

Block	Region	Operations
Left sidebar		
+ Navigation	Left sidebar	configure
+ User login	Left sidebar	configure
Right sidebar		
+ Feedback	Right sidebar	configure
+ New Feedbacks <small>Drag to re-order</small>	Right sidebar	configure
+ Modulo 1	Right sidebar	configure
+ Libreria - Modulo 1	Right sidebar	configure
+ Libreria - Modulo 2 [™]	Right sidebar	configure

Figura III. 110: Lugar de la visualización del módulo 2 Herramienta Drupal

Luego se observa en el bloque derecho que ya aparece el módulo 2 activado y listo para ser utilizado, ver Figura III.111.

The image shows a sidebar menu with the following items:

- Libreria Modulo 1
 - Agregar Producto
- Libreria Modulo 2
 - Agregar Producto
 - Listar Productos

Figura III. 111: Librería Modulo 2 - Módulo 2 Herramienta Drupal

Al realizar la ejecución de Librería Modulo 2 en Agregar producto se observa el formulario de ingreso de un nuevo artículo, ver Figura III.112.

Search this site:

admin

- Contact
- My account
- Create content
- Administer
- Log out

Agregar Producto

Nombre:

Marca:

Color:

Tipo-Tamano:

Categoría:

Presentación:

Imagen: No se ha ...n archivo

Descripción:

Precio Compra:

Precio de Venta al Publico:

Publicado:

Si

No

Feedback

[New Feedback](#)

Feedback Messages

There are 1 new feeds

Modulo 1

[Entrar](#)

Libreria Modulo 1

[Agregar Producto](#)

Libreria Modulo 2

[Agregar Producto](#)

[Listar Productos](#)

Figura III. 112: Formulario de ingreso de un nuevo artículo - Módulo 2 Herramienta Drupal

Además la segunda opción, listar todos los artículos, ver Figura III.113.

Listar Productos

Nombre	Marca	Color	Tipo - Tamano	Categoría	Presentación	Imagen	Descripción	Precio Compra	Precio Venta Publico	Publicada
Nombre			Tipo	1	Presentación	ff.jpg	Descripción	10	15	No
Esfero Negro	Bic	Negro	Medio	1	Caja 12	esf.jpg	Esfero de color negro, marca bic	0	0	Si

Feedback

[New Feedback](#)

Feedback Messages

There are 1 new feeds

Modulo 1

[Entrar](#)

Libreria Modulo 1

[Agregar Producto](#)

Libreria Modulo 2

[Agregar Producto](#)

[Listar Productos](#)

Figura. 113: Listado de productos - Módulo 2 Herramienta Drupal

Cada artículo posee un enlace en el nombre mediante el cual podemos acceder a información detallada de cada uno, incluyendo la imagen que se adjunto, ver Figura III.114.

Product details for 'Esfero Negro':

- Nombre: Esfero Negro
- Marca: Bic
- Color: Negro
- Tipo/Tamaño: Medio
- Categoría: 1
- Presentación: Caja 12
- Descripción: Esfero de color negro, marca bic
- Precio Compra: 0
- Precio Venta Público: 0
- Publicado: Si

Nombre	Marca	Color	Tipo - Tamaño	Categoría	Presentación	Imagen	Descripción	Precio Compra	Precio Venta Público	Publicada
Nombre	Marca	Color	Tipo	1	Presentación	ff.jpg	Descripción	10	15	No
Esfero Negro	Bic	Negro	Medio	1	Caja 12	esf.jpg	Esfero de color negro, marca bic	0	0	Si

Figura. 114: Enlace del listado de productos - Módulo 2 Herramienta Drupal

3.7.3.3. Módulo 3:

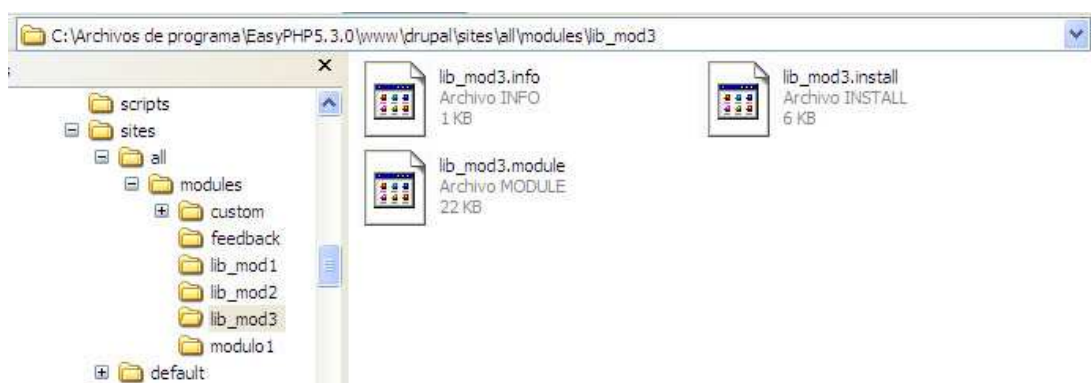
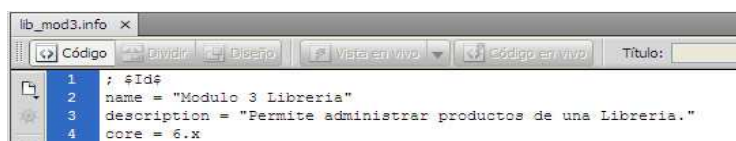


Figura III. 115: Archivos necesarios para la creación del módulo 3 - Módulo 3 Herramienta Drupal

Al componente del módulo 2, se lo va a expandir, por lo cual se añade 4 tablas mas a la base de datos drupal, la tabla lib_mo3_categoria, lib_mo3_subcategoria, lib_mo3_proveedor y la tabla lib_mo3_articulo_proveedor. Y a la vez se modifica la estructura de la tabla lib_mo3_articulo, se elimina el campo categoría para tenerla en una tabla aparte, con su respectiva sub categoría.

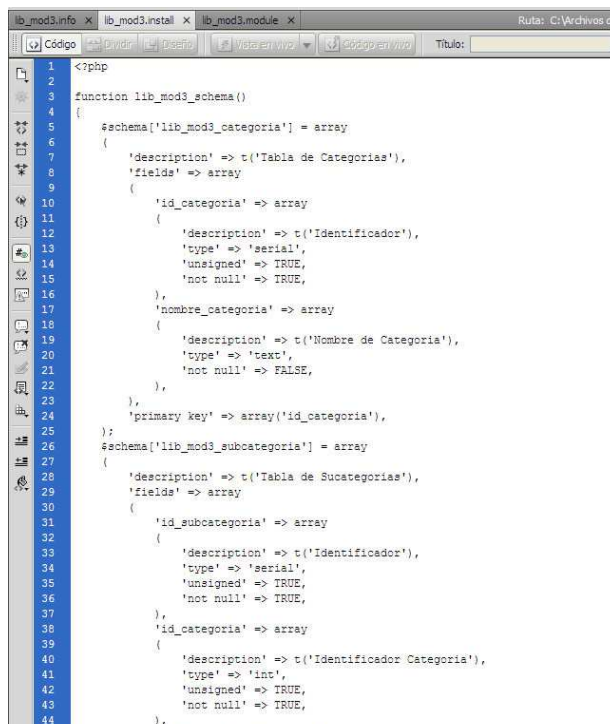
Se inicia con el desarrollo del archivo donde se describe al modulo en desarrollo para este caso **lib_mod3.info**, ver Figura III.116.



```
lib_mod3.info x
Código
; $Ids
1 name = "Modulo 3 Libreria"
2 description = "Permite administrar productos de una Libreria."
3 core = 6.x
4
```

Figura III. 116: Contenido del archivo lib_mod3.info - Módulo 3 Herramienta Drupal

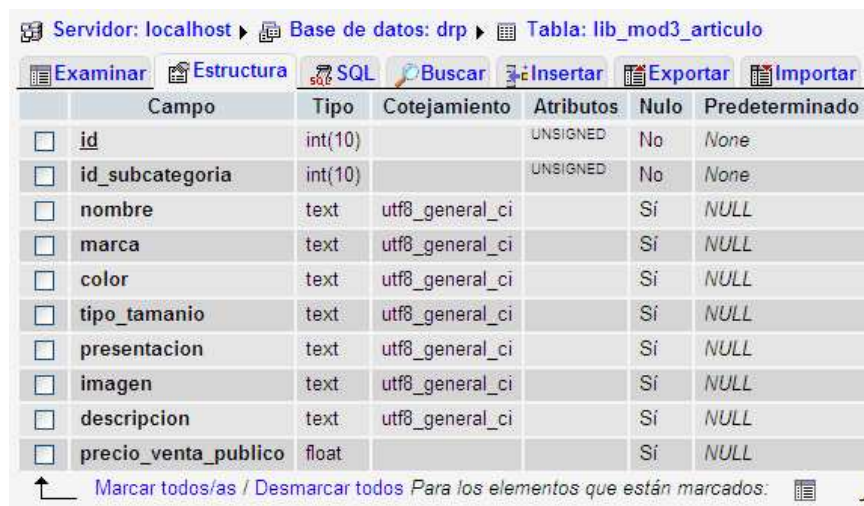
En el archivo **lib_mod3.install**, se define el schema de las tablas que se crearán en la base de datos drupal, al momento de activar el módulo, en este schema se describen los 5 modelos con los que se va a trabajar, cada uno se define dentro de un elemento en un array general, ver Figura III.117.



```
lib_mod3.info x lib_mod3.install x lib_mod3.module x Ruta: C:\Wchivos de
Código
<?php
1 function lib_mod3_schema()
2 {
3     $schema['lib_mod3_categoria'] = array
4     (
5         'description' => t('Tabla de Categorias'),
6         'fields' => array
7         (
8             'id_categoria' => array
9             (
10                'description' => t('Identificador'),
11                'type' => 'serial',
12                'unsigned' => TRUE,
13                'not null' => TRUE,
14            ),
15            'nombre_categoria' => array
16            (
17                'description' => t('Nombre de Categoria'),
18                'type' => 'text',
19                'not null' => FALSE,
20            ),
21        ),
22        'primary key' => array('id_categoria'),
23    );
24     $schema['lib_mod3_subcategoria'] = array
25     (
26         'description' => t('Tabla de Sucategorias'),
27         'fields' => array
28         (
29             'id_subcategoria' => array
30             (
31                'description' => t('Identificador'),
32                'type' => 'serial',
33                'unsigned' => TRUE,
34                'not null' => TRUE,
35            ),
36            'id_categoria' => array
37            (
38                'description' => t('Identificador Categoria'),
39                'type' => 'int',
40                'unsigned' => TRUE,
41                'not null' => TRUE,
42            ),
43        ),
44    );
45 }
```

Figura III. 117: Contenido del archivo lib_mod3.install - Módulo 3 Herramienta Drupal

Este schema creará una nueva tabla llamada lib_mod3_articulo en la base de datos drupal en MySQL, ver Figura III.118.



	Campo	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado
<input type="checkbox"/>	id	int(10)		UNSIGNED	No	None
<input type="checkbox"/>	id_subcategoria	int(10)		UNSIGNED	No	None
<input type="checkbox"/>	nombre	text	utf8_general_ci		Sí	NULL
<input type="checkbox"/>	marca	text	utf8_general_ci		Sí	NULL
<input type="checkbox"/>	color	text	utf8_general_ci		Sí	NULL
<input type="checkbox"/>	tipo_tamano	text	utf8_general_ci		Sí	NULL
<input type="checkbox"/>	presentacion	text	utf8_general_ci		Sí	NULL
<input type="checkbox"/>	imagen	text	utf8_general_ci		Sí	NULL
<input type="checkbox"/>	descripcion	text	utf8_general_ci		Sí	NULL
<input type="checkbox"/>	precio_venta_publico	float			Sí	NULL

Figura III. 118: Contenido de la tabla lib_mod3_articulo - Módulo 3 Herramienta Drupal

También creará una nueva tabla llamada lib_mod3_articulo_proveedor en la base de datos drupal en MySQL, ver Figura III.119.



	Campo	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado
<input type="checkbox"/>	id_articulo_proveedor	int(10)		UNSIGNED	No	None
<input type="checkbox"/>	id_proveedor	int(10)		UNSIGNED	No	None
<input type="checkbox"/>	id_articulo	int(10)		UNSIGNED	No	None
<input type="checkbox"/>	factura	int(10)		UNSIGNED	Sí	NULL
<input type="checkbox"/>	precio_compra	float			Sí	NULL
<input type="checkbox"/>	cantidad	int(10)		UNSIGNED	Sí	NULL

Figura III. 119: Contenido de la tabla lib_mod3_articulo_proveedor - Módulo 3 Herramienta Drupal

Además se creará una nueva tabla llamada lib_mod3_categoria en la base de datos drupal en MySQL, ver Figura III.120.



The screenshot shows the MySQL table structure for 'lib_mod3_categoria'. The table has two columns: 'id_categoria' (int(10), UNSIGNED, No NULL) and 'nombre_categoria' (text, utf8_general_ci, Sí NULL).

	Campo	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado
<input type="checkbox"/>	<u>id_categoria</u>	int(10)		UNSIGNED	No	None
<input type="checkbox"/>	nombre_categoria	text	utf8_general_ci		Sí	NULL

Figura III. 120: Contenido de la tabla lib_mod3_categoria - Módulo 3 Herramienta Drupal

Asimismo se creará una nueva tabla llamada lib_mod3_proveedor en la base de datos drupal en MySQL, ver Figura III.121.

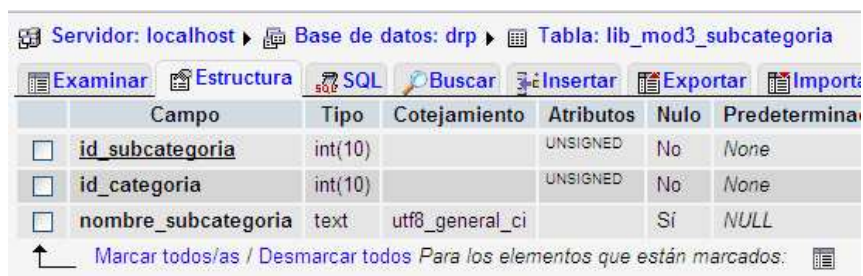


The screenshot shows the MySQL table structure for 'lib_mod3_proveedor'. The table has eight columns: 'id_proveedor' (int(10), UNSIGNED, No NULL), 'nombre_proveedor' (text, utf8_general_ci, Sí NULL), 'ci_ruc' (text, utf8_general_ci, Sí NULL), 'direccion' (text, utf8_general_ci, Sí NULL), 'telefono' (text, utf8_general_ci, Sí NULL), 'movil' (text, utf8_general_ci, Sí NULL), 'pais_ciudad' (text, utf8_general_ci, Sí NULL), and 'email' (text, utf8_general_ci, Sí NULL).

	Campo	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado
<input type="checkbox"/>	<u>id_proveedor</u>	int(10)		UNSIGNED	No	None
<input type="checkbox"/>	nombre_proveedor	text	utf8_general_ci		Sí	NULL
<input type="checkbox"/>	ci_ruc	text	utf8_general_ci		Sí	NULL
<input type="checkbox"/>	direccion	text	utf8_general_ci		Sí	NULL
<input type="checkbox"/>	telefono	text	utf8_general_ci		Sí	NULL
<input type="checkbox"/>	movil	text	utf8_general_ci		Sí	NULL
<input type="checkbox"/>	pais_ciudad	text	utf8_general_ci		Sí	NULL
<input type="checkbox"/>	email	text	utf8_general_ci		Sí	NULL

Figura III. 121: Contenido de la tabla lib_mod3_proveedor - Módulo 3 Herramienta Drupal

Igualmente se creará una nueva tabla llamada lib_mod3_subcategoria en la base de datos drupal en MySQL, ver Figura III.121.

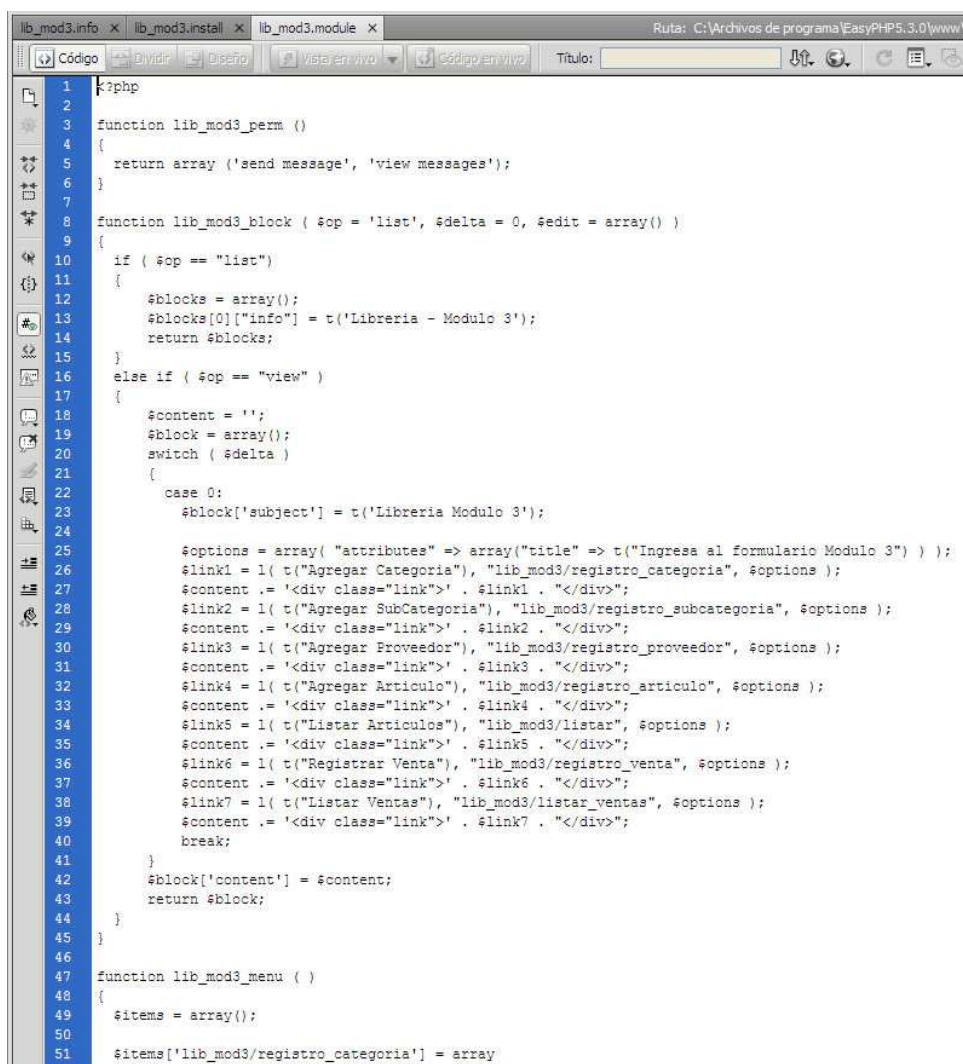


The screenshot shows the MySQL table structure for 'lib_mod3_subcategoria'. The table has three columns: 'id_subcategoria' (int(10), UNSIGNED, No NULL), 'id_categoria' (int(10), UNSIGNED, No NULL), and 'nombre_subcategoria' (text, utf8_general_ci, Sí NULL).

	Campo	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado
<input type="checkbox"/>	<u>id_subcategoria</u>	int(10)		UNSIGNED	No	None
<input type="checkbox"/>	id_categoria	int(10)		UNSIGNED	No	None
<input type="checkbox"/>	nombre_subcategoria	text	utf8_general_ci		Sí	NULL

Figura III. 122: Contenido de la tabla lib_mod3_subcategoria - Módulo 3 Herramienta Drupal

En el archivo **lib_mod3.module**, se describe la lógica y la vista de nuestro modulo, se toma como base la estructura del módulo 2, en este caso se ha aumentado en la descripción del hook del menú y las funciones necesarias para procesar los nuevos formularios, este archivo cuenta con nuevos métodos para mostrar en forma de tabla la lista de artículos, agregar una categoría, agregar subcategoría, agregar venta, agregar artículos, listar ventas, ver Figura III.123.



```
1 <?php
2
3 function lib_mod3_perm ()
4 {
5     return array ('send message', 'view messages');
6 }
7
8 function lib_mod3_block ( $op = 'list', $delta = 0, $edit = array() )
9 {
10     if ( $op == "list" )
11     {
12         $blocks = array();
13         $blocks[0]["info"] = t('Librería - Modulo 3');
14         return $blocks;
15     }
16     else if ( $op == "view" )
17     {
18         $content = '';
19         $block = array();
20         switch ( $delta )
21         {
22             case 0:
23                 $block['subject'] = t('Librería Modulo 3');
24
25                 $options = array( "attributes" => array("title" => t("Ingresar al formulario Modulo 3") ) );
26                 $link1 = l( t("Agregar Categoría"), "lib_mod3/registro_categoria", $options );
27                 $content .= '<div class="link">' . $link1 . "</div>";
28                 $link2 = l( t("Agregar SubCategoría"), "lib_mod3/registro_subcategoria", $options );
29                 $content .= '<div class="link">' . $link2 . "</div>";
30                 $link3 = l( t("Agregar Proveedor"), "lib_mod3/registro_proveedor", $options );
31                 $content .= '<div class="link">' . $link3 . "</div>";
32                 $link4 = l( t("Agregar Artículo"), "lib_mod3/registro_articulo", $options );
33                 $content .= '<div class="link">' . $link4 . "</div>";
34                 $link5 = l( t("Listar Artículos"), "lib_mod3/listar", $options );
35                 $content .= '<div class="link">' . $link5 . "</div>";
36                 $link6 = l( t("Registrar Venta"), "lib_mod3/registro_venta", $options );
37                 $content .= '<div class="link">' . $link6 . "</div>";
38                 $link7 = l( t("Listar Ventas"), "lib_mod3/listar_ventas", $options );
39                 $content .= '<div class="link">' . $link7 . "</div>";
40                 break;
41             }
42         $block['content'] = $content;
43         return $block;
44     }
45 }
46
47 function lib_mod3_menu ( )
48 {
49     $items = array();
50
51     $items['lib_mod3/registro_categoria'] = array
```

Figura. 123: Contenido del archivo lib_mod3.module - Módulo 3 Herramienta Drupal

Para activar el módulo 3, ingresar al área administrativa de Drupal, *Administer – SiteBuilding – Modules*, buscar el módulo en el grupo *Other*, activar y guardar la configuración, ver Figura III.124.



Enabled	Name	Version	Description
<input checked="" type="checkbox"/>	MagMax Feedback		Proporciona un sistema de feedbacks fácil para drupal
<input checked="" type="checkbox"/>	Modulo 1		Prueba para el primer modulo.
<input checked="" type="checkbox"/>	Modulo 1 Libreria		Permite administrar productos de una Libreria.
<input checked="" type="checkbox"/>	Modulo 2 Libreria		Permite administrar productos de una Libreria.
<input checked="" type="checkbox"/>	Modulo 3 Libreria		Permite administrar productos de una Libreria.

Figura III. 124: Activación del módulo 3 - Módulo 3 Herramienta Drupal

Ya está activado el módulo, pero aun no se indica en que parte del sitio aparecerá, para esto ir al área administrativa, *Administer – SiteBuilding – Blocks*, buscar el módulo 3 en el área *Disable* y arrastrarlo al bloque *Rightsidebar*, y guardar la configuración, ver Figura III.125.



Block	Region	Operations
Left sidebar		
+	Navigation	Left sidebar configure
+	User login	Left sidebar configure
Right sidebar		
+	Feedback	Right sidebar configure
+	New Feedbacks	Right sidebar configure
+	Modulo 1	Right sidebar configure
+	Libreria - Modulo 1	Right sidebar configure
+	Libreria - Modulo 2	Right sidebar configure
+	Libreria - Modulo 3	Right sidebar configure

Figura. 125: Lugar de la visualización del módulo 3 - Módulo 3 Herramienta Drupal

En el bloque derecho ya aparece el módulo 3, Librería Modulo 3 activado y listo para poder utilizarlo, ver Figura III.126.



Figura III. 126: Librería Modulo 3 - Módulo 3 Herramienta Drupal

Al ejecutar Agregar Categoría se muestra el formulario de ingreso de una nueva categoría, ver figura III.127.




Figura III. 127: Formulario Agregar Categoría - Módulo 3 Herramienta Drupal

Al ejecutar Agregar SubCategoría se muestra el formulario de ingreso de una nueva subcategoría, ver Figura III.128.



Figura III. 128: Formulario Agregar Subcategoría - Módulo3 Herramienta Drupal

Al ejecutar Agregar Proveedor se muestra el formulario de ingreso de un nuevo proveedor, ver Figura III.129.



The screenshot shows a web form titled 'Agregar Proveedor' with a 'Home' link at the top. The form contains the following fields: 'Nombre Proveedor:' (with a red asterisk indicating it is required), 'Numero CI RUC:', 'Direccion:', 'Telefono:', 'Movil:', 'Pais-Ciudad:', and 'eMail:'. Each field is represented by a text input box. At the bottom of the form is a 'Submit' button.

Figura III. 129: Formulario Agregar Proveedor - Módulo 3 Herramienta Drupal

Al ejecutar Agregar Artículo se muestra el formulario de ingreso de un nuevo artículo, ver Figura III.130.



The screenshot shows a web form titled 'Agregar Artículo' with a 'Home' link at the top. The form contains the following fields: 'Nombre:', 'Marca:', 'Color:', 'Tipo-Tamano:', 'SubCategoria:' (a dropdown menu currently showing 'SubCat para 0'), 'Presentacion:', 'Imagen:' (with a 'Seleccionar archivo' button and the text 'No se ha ...n archivo'), 'Descripcion:' (a large text area), and 'Precio de Venta al Publico:'. At the bottom of the form is a 'Submit' button.

Figura III. 130: Formulario Agregar Artículo - Módulo 3 Herramienta Drupal

Al ejecutar Listar Productos se muestra los productos registrados en la base de datos, ver Figura III.131.

Nombre	Marca	Color	Tipo - Tamano	Categoria	Presentacion	Imagen	Descripción	Precio Compra	Precio Venta Publico	Publicada
Esfero Negro	Bic	Negro	Medio	1	Caja 12	esf.jpg	Esfero de color negro, marca bic	0	0	Si

Figura III. 131: Listado de Productos - Módulo 3 Herramienta Drupal

Al ejecutar Registrar Venta se muestra el formulario de registro de venta, ver Figura III.132.

Factura: *

Proveedor: Proveedor 1

Artículo: Nombre

Precio de Compra:

Cantidad:

Submit

Figura III. 132: Formulario Registrar Venta - Módulo 3 Herramienta Drupal

Al ejecutar Listar Productos se muestra el listado de todos los productos registrados en la base de datos, ver Figura III.133.

Factura	Proveedor	Artículo	Precio Compra	Cantidad
1	Proveedor 1	Nombre	15	10
4	Proveedor 1	Nombre	55	55

Figura III. 133: Listado de productos - Módulo 3 Herramienta Drupal

3.7.3.4. Módulo 4:

En este módulo, se estudia las facilidades que Drupal ofrece para la administración de los componentes, su instalación, desinstalación, habilitación, des habilitación, entre otros.

La instalación de un módulo en Drupal viene determinada por varios pasos.

Descarga del módulo del sitio oficial, ver Figura III.134.

Administration menu
By [sun](#) on the 12th of January, 2007

Administration menu module provides a theme-independent administration interface (aka. navigation, back-end) for Drupal. It's a helper for novice Drupal users coming from other CMS, a real time-saver for Drupal site administrators, and definitely a must for Drupal developers and site builders (keyword: [Devel](#) integration).

The module renders all administrative menu items below 'administer' in a clean, attractive and purely CSS-based menu at the top of your website. It contains not only regular menu items - local tasks are also included, giving you extremely fast access to any administrative resource and function your Drupal installation provides.

Drupal exposed! Administering, developing, and working with Drupal has never been that fast, easy and concise.

Version	Date	Links	Status
7.x-3.x-dev	2010-Sep-15	Download · Release notes	Development snapshot
6.x-1.6	2010-Sep-03	Download · Release notes	Recommended for 6.x
5.x-2.8	2009-Jan-24	Download · Release notes	Recommended for 5.x
4.7.x-1.4	2007-Sep-06	Download · Release notes	Recommended for 4.7.x

[Find out more](#) · [Bugs and feature requests](#)

Figura III. 134: Descarga de módulos - Módulo 4 Herramienta Drupal

Descomprimir y subir al hosting la carpeta o el archivo *.module*, a la carpeta modules, ver Figura III.135.

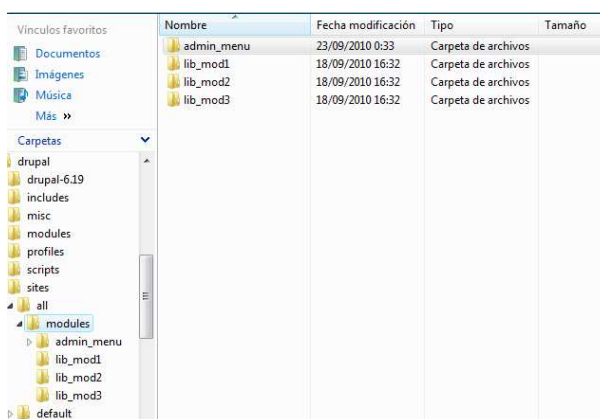


Figura III. 135: Ubicación del módulo descargado - Módulo 4 Herramienta Drupal

En este momento ya está instalado el módulo, ahora hay que activarlo. Para **activar el módulo** ingresar a Administer -> SiteBuildings -> Modules y escoger el módulo descargado Administrationmenu, ver Figura III.136.

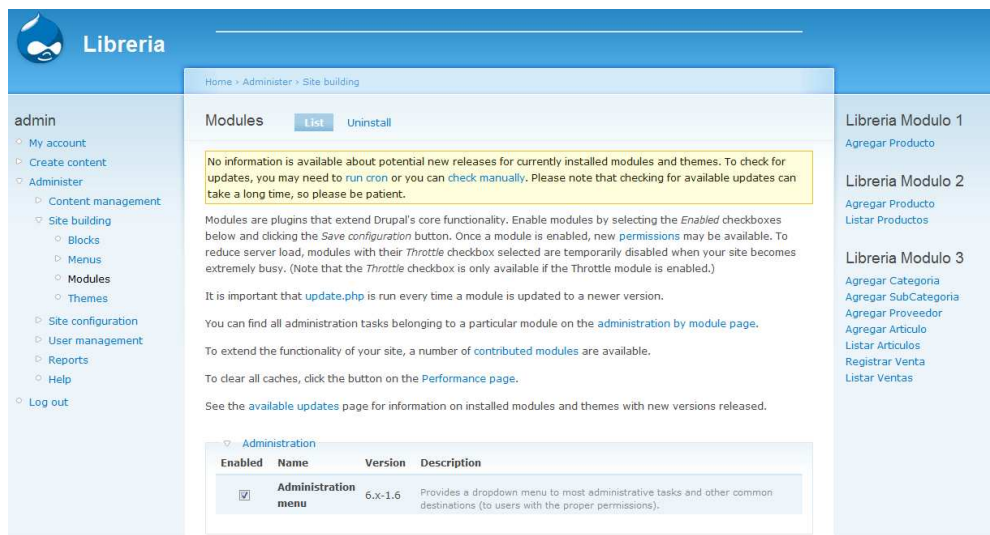


Figura III. 136: Activación del módulo descargado - Módulo 3 Herramienta Drupal

Hacer clic en guardar configuración y ya se ve en la parte superior el módulo instalado y el mensaje de instalación satisfactoria, ver Figura III.137.

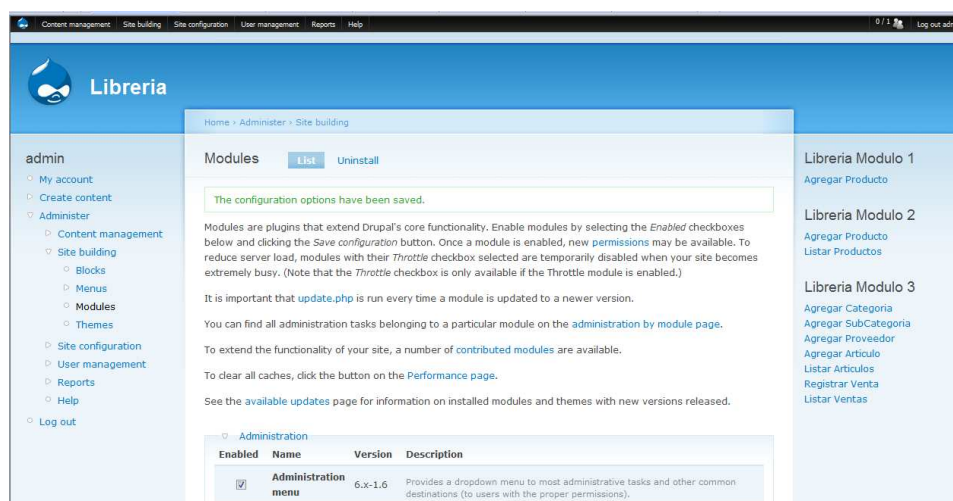


Figura. 137: Mensaje de configuración satisfactoria - Módulo 4 Herramienta Drupal

Configuración del módulo.

En este punto seleccionar las opciones intrínsecas del modulo activado, para esto buscar en el menú del sidebar en **administrar/nombre del módulo** y si no apareciera aquí en **administrar/opciones**

Para **Asignación de permisos** ingresar a Administer ->Usermanagement ->Permissions y seleccionar los usuarios que pueden utilizar este módulo, ver Figura III.138.

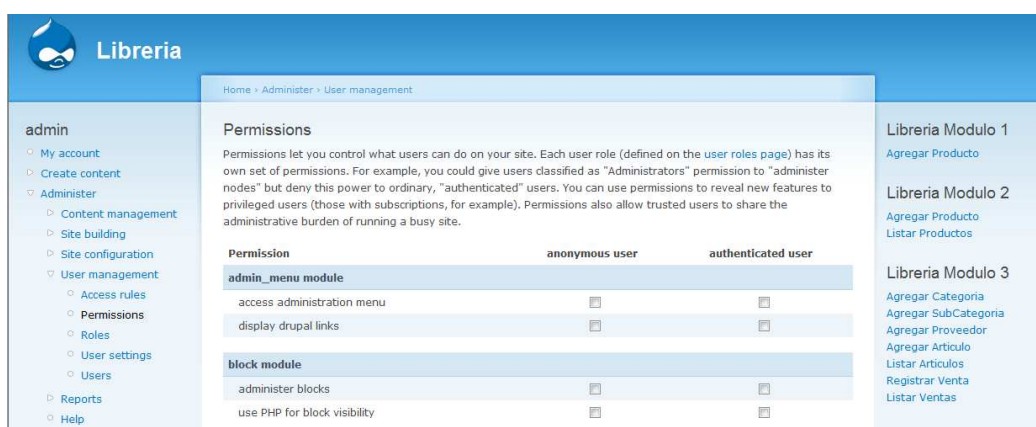


Figura III. 138: Asignación de permisos sobre el módulo descargado - Módulo 4 Herramienta Drupal

Para desinstalar un módulo dentro de Administer ->SiteBuildings -> Modules escoger uninstall, ver Figura III.139.

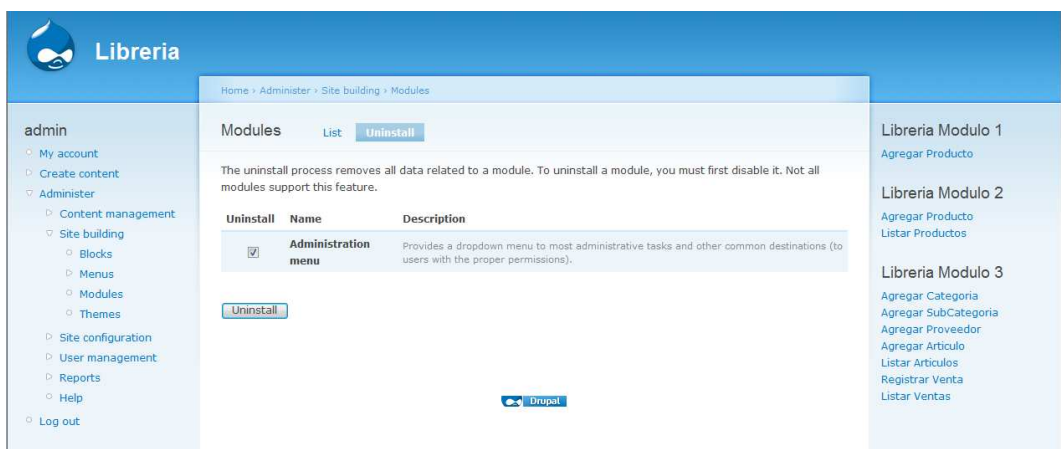


Figura III. 139: Desinstalación del módulo descargado - Módulo 4 Herramienta Drupal

Aparece el mensaje de confirmación del módulo, hacer clic en uninstall, ver Figura III.140.

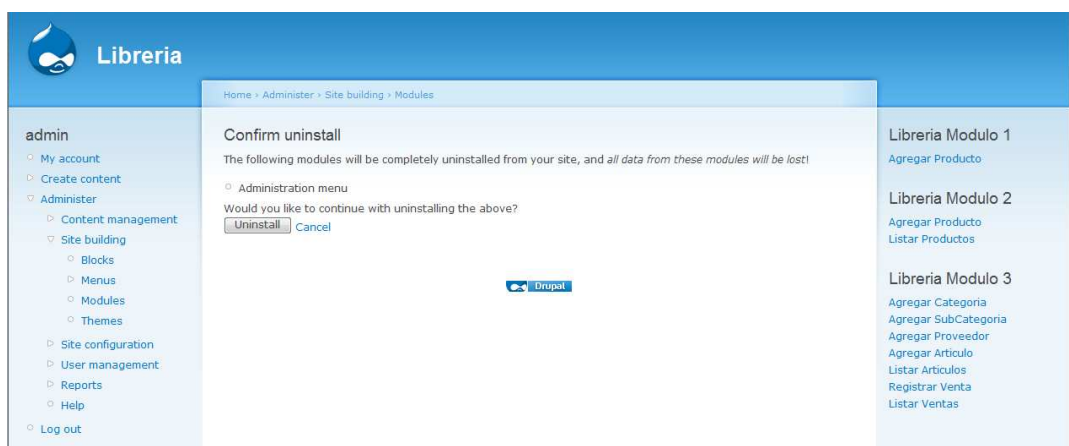


Figura III. 140: Confirmación de desinstalación del módulo descargado - Módulo 4 Herramienta Drupal

Se muestra el mensaje de desinstalación exitosa, ver Figura III.141.

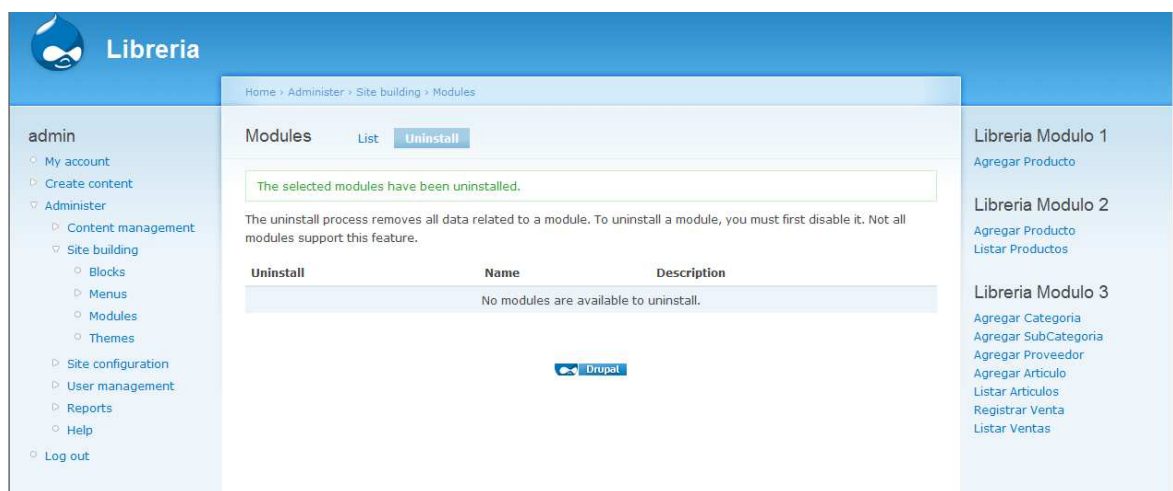


Figura III. 141: Mensaje de desinstalación exitosa del módulo descargado - Módulo 4 Herramienta Drupal

3.8. Análisis Comparativo

En esta sección se va a mostrar el estudio de los Sistemas de administración de Contenidos CMS Joomla!, Drupal y Typo3 a manera de cuadros comparativos, seguidos estos de una interpretación y calificación del criterio evaluado por parte del autor, estos cuadros comparativos se encuentran clasificados de acuerdo a los parámetros de comparación definidos anteriormente.

Para obtener resultados cuantitativos y cualitativos que permitan una selección sustentada de una de las herramientas analizadas, la calificación de cada uno de los parámetros de comparación se basa en las escalas de la Tabla III.I y la Tabla III.II:

Tabla III. I: Escala de Puntuación para calificación de Parámetros

Regular	Bueno	Muy Bueno	Excelente
< 70 %	>= 70 % y < 80 %	>= 80 % y < 90 %	>= 90 %

Tabla III. II: Escalas de valoración cualitativa

1	2	3	4
Mono plataforma	Multiplataforma		
Insuficiente	Parcial	Suficiente	Excelente
Mala	Regular	Buena	Muy Buena
Nada amigable	Parcialmente amigable	En su mayor parte amigable	Totalmente amigable
Deficiente	Poco eficiente	Eficiente	Muy eficiente
No se cumple	Se cumple satisfactoriamente	Se cumple aceptablemente	Se cumple plenamente
Si			No

Cada uno de los ítems de la interpretación incluye la siguiente nomenclatura $(x,y,z)/w$ en donde cada letra significa lo siguiente:

x: Representa el puntaje que obtiene la herramienta Joomla!.

y: Representa el puntaje que obtiene la herramienta Typo3.

z: Representa el puntaje que obtiene la herramienta Drupal.

w: Representa la base del puntaje sobre la cual se está calificando el parámetro.

La calificación definitiva de la herramienta en base a cada parámetro de comparación se obtiene sumando los puntajes obtenidos del análisis, utilizando las siguientes fórmulas:

$$\mathbf{P_{joom} = \Sigma(x), P_{typo} = \Sigma(y), P_{drup} = \Sigma(z), P_c = \Sigma(w)}$$

$$\mathbf{Calificación\ de\ Joomla!\ (Cc-Joom) = (P_{joom} / P_c) * 100\%}$$

$$\mathbf{Calificación\ de\ Typo3\ (Cc-Typo) = (P_{typo} / P_c) * 100\%}$$

$$\mathbf{Calificación\ de\ Drupal\ (Cc-Drup) = (P_{drup} / P_c) * 100\%}$$

En donde:

P_{joom}: Puntaje acumulado por Joomla! en el parámetro.

P_{typo}: Puntaje acumulado por Typo3 en el parámetro.

P_{drup}: Puntaje acumulado por Drupal en el parámetro.

P_c: Puntaje sobre el que se califica el parámetro.

Cc-Joom: Porcentaje de la calificación total que obtuvo Joomla! en el parámetro.

Cc-Typo: Porcentaje de la calificación total que obtuvo Typo3 en el parámetro.

Cc-Drup: Porcentaje de la calificación total que obtuvo Drupal en el parámetro.

3.8.1. Requerimientos para el desarrollo

Es conveniente que la definición de requerimientos para el desarrollo por parte de los diferentes CMS, sea analizado por el programador, ya que facilitará el desarrollo y mantenimiento de los sistemas web a implementar, tomando en cuenta todas las prestaciones de cada uno de ellos además de que hablamos de código libre.

3.8.1.1. Determinación de Variables

- a. Plataforma
- b. Soporte
- c. Seguridad

3.8.1.2. Valoraciones

a. Variable Plataforma

Igual que cualquier programa, todo CMS necesita de un sistema operativo instalado previamente en el ordenador y no sirve la mayoría de veces para otros sistemas operativos ya sea para Windows o Linux, teniendo en cuenta que en cada sitio web oficial existe el instalador para cada plataforma, **un CMS necesita** que el servidor tenga **instalados previamente unos programas que necesita para funcionar.**

Estos requerimientos previos son a grandes rasgos el estándar de un servidor de páginas web. Software de servidor web, lenguajes y bibliotecas de programación típica de web y un sistema de bases de datos. De forma muy general y mayoritariamente este software necesario para que funcione un CMS se abrevia con las siglas LAMP, WAMP para nuestra publicación utilizamos XAMPP, valoración 2.

b. Variable Soporte

El soporte para desarrollo de componentes se puede encontrar en varios lugares. El mejor lugar es empezar por el sitio oficial de cada sistema de administración de contenidos. Aquí se puede hallar la información que regularmente se va publicando y actualizando, valoración 4.

c. Variable Seguridad

Cada vez se valora más la seguridad de un producto a la hora de tomar una elección para seleccionar en qué software se implementará una aplicación. En el mundo de los CMS, por su vocación de producto accesible desde internet, resulta aún más importante valorar cuál ofrece más garantías de no sufrir un desfase, valoración 4.

Tabla III. III: Requerimientos para el desarrollo

Variable	Joomla!	Typo3	Drupal
Plataforma	Multiplataforma	Multiplataforma	Multiplataforma
Soporte	Excelente	Parcial	Suficiente
Seguridad	Buena	Buena	Buena

3.8.1.3. Interpretación de la Tabla III.III

- Tanto Joomla!, Typo3 como Drupal, son sistemas multiplataforma basadas en PHP. Los requisitos son una base de datos MySQL es la principal, aunque soporta otros sistemas como PostgreSQL y Oracle y un servidor web que soporte PHP. La configuración más típica es un entorno LAMP (Linux + Apache + MySQL + PHP), aunque puede funcionar bajo otros sistemas operativos como Windows, WAMP (Windows + Apache + MySQL + PHP). A nivel de cliente, pueden ser accedidos desde cualquier plataforma o sistema operativo, tan solo hace falta conexión a la red apropiada y un navegador de internet. (2, 2, 2)/2.

- Como Soporte, los miembros de las comunidades de cada CMS, publican regularmente en sus blogs sobre temas varios, como técnicas de programación, seguridad, eventos, etc. Existe la opción de suscripción, la cual permitirá conocer las nuevas publicaciones por medio de correo electrónico. (4, 2, 3)/4.
- Gracias a cada una de las comunidades que mueven Joomla!, Typo3 y Drupal, los sistemas están bajo constante vigilancia, desde los propios creadores hasta el usuario final que lo usa. Esto permite decir que, cuando aparece una vulnerabilidad, se soluciona de inmediato minimizando al máximo los posibles efectos de la misma. (3, 3, 3)/4.

3.8.1.4. Calificación

$$P_c = \sum(w) = 2+4+4 = 10$$

$$P_{\text{joom}} = \sum(x) = 2+4+3 = 9 \quad C_c\text{-Joom: } P_{\text{joom}}/P_c = (9/10) * 100\% = 90\%$$

$$P_{\text{typo}} = \sum(y) = 2+2+3 = 7 \quad C_c\text{-Typo: } P_{\text{typo}}/P_c = (7/10) * 100\% = 70\%$$

$$P_{\text{drup}} = \sum(z) = 2+3+3 = 8 \quad C_c\text{-Drup: } P_{\text{drup}}/P_c = (8/10) * 100\% = 80\%$$

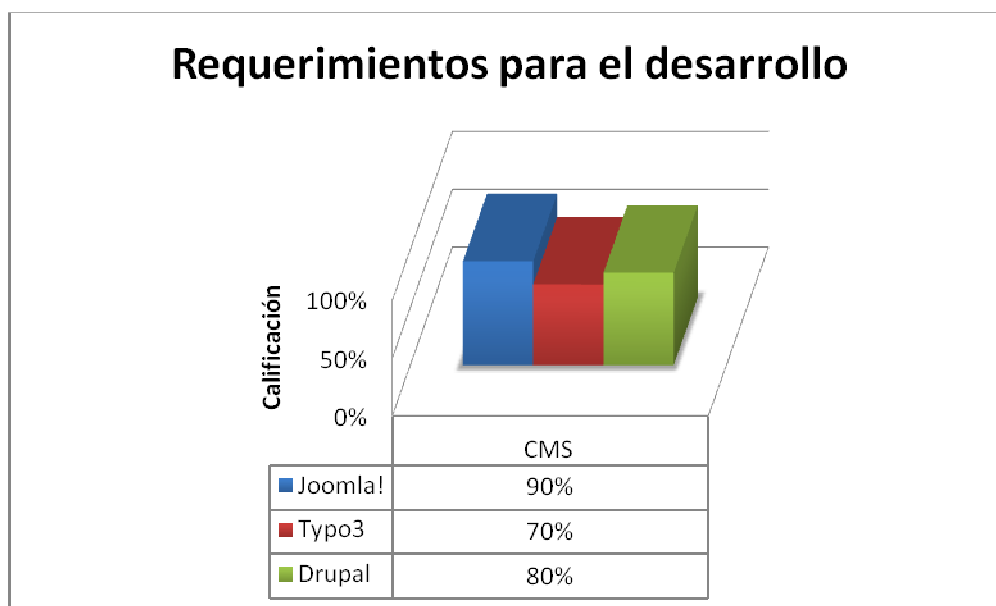


Figura III. 142: Comparación de porcentajes parámetro Requerimientos para el desarrollo

3.8.2. Facilidad de uso

En la presente evaluación se va considerar también el factor facilidad de uso, para conocer el nivel de satisfacción de los usuarios finales del componente desarrollado, el control y personalización de componentes requiere mucha dedicación, organización, planificación, sin querer decir con esto que su desarrollo sea una tarea imposible.

Se recomienda también el uso de la "Lógica" para entender cómo funcionan y sobre todo como reaccionan ante problemas o cambios.

3.8.2.1. Determinación de Variables

- a. Interfaz de usuario
- b. Acceso a datos
- c. Flexibilidad

3.8.2.2. Valoraciones

a. Variable Interfaz de usuario

La elaboración de interfaces de usuario es un aspecto primordial al momento del desarrollo de componentes, ya que es el medio de comunicación entre el usuario final y la aplicación, esta interacción debe ser creada de la forma más sencilla, llamativa y rápida, aquí hacemos referencia al desarrollo del módulo 1, valoración 4.

b. Variable Acceso a datos

El acceso a datos es un ámbito esencial dentro de los componentes de un CMS, porque permite gestionar información almacenada en bases de datos,

a través del uso de ingresos, consultas, modificaciones y eliminaciones, los mismos que deben ejecutarse de una forma que sea lo más rápida y segura posible, para el análisis de esta variable hacemos referencia el desarrollo del módulo 2, valoración 4.

c. Variable Flexibilidad

Esta característica permite la reutilización total del componente, o parte del mismo con la necesidad o no de ampliar su funcionamiento para características propias del sitio web a aplicarse, se debe considerar el lenguaje de programación y la base de datos a utilizar en el CMS escogido, tomamos en cuenta el desarrollo del módulo 3, valoración 4.

Tabla III. IV: Facilidad de uso

Variable	Joomla!	Typo3	Drupal
Interfaz de usuario	Totalmente amigable	Parcialmente Amigable	En su mayor parte amigable
Acceso a datos	Muy eficiente	Eficiente	Eficiente
Flexibilidad	Se cumple plenamente	Se cumple aceptablemente	Se cumple plenamente

3.8.2.3. Interpretación de la Tabla III.IV

- Las Interfaces o "Plantillas" definen la disposición de su sitio web. Usted puede asignar diferentes plantillas a varias páginas de su sitio web. Las plantillas pueden crearse para el Sitio o personalizar la gran variedad ya existentes. (4, 2, 3)/4.
- Excelente Desempeño MySql como base de datos, en el caso de Joomla! en el archivo configuration.php se encuentran las variables respectivas para la conexión a la base de datos y en nuestro componente se invoca a las mismas para realizar las tareas requeridas, en Drupal no es

necesario manipulación directa de la base de datos ya que Drupal gestiona automáticamente a partir de un archivo donde se define el esquema a utilizar. (4,3,3)/4.

- La flexibilidad hace referencia a la amplitud con que pueden ser utilizados los componentes en diferentes aplicaciones web para distintas necesidades (4, 3, 4)/4.

3.8.2.4. Calificación

$$P_c = \sum(w) = 4+4+4 = 12$$

$$P_{joom} = \sum(x) = 4+4+4 = 12 \quad Cc\text{-Joom}: P_{joom}/P_c = (12/12) * 100\% = 100\%$$

$$P_{typo} = \sum(y) = 2+3+3 = 7 \quad Cc\text{-Typo}: P_{typo}/P_c = (8/12) * 100\% = 66.67\%$$

$$P_{drup} = \sum(z) = 3+3+4 = 6 \quad Cc\text{-Drup}: P_{drup}/P_c = (10/12) * 100\% = 83.33\%$$

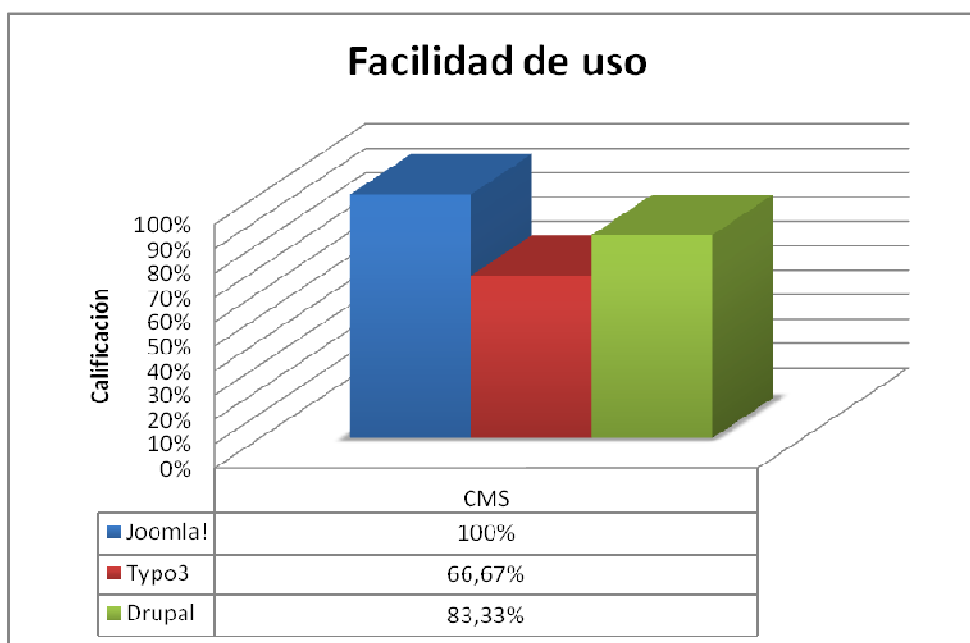


Figura III. 143: Comparación de porcentajes parámetro Facilidad de uso

3.8.3. Gestión de componentes dentro del CMS

Como desarrolladores de sitios web al utilizar sistemas de administración de contenidos, estamos en la responsabilidad del mantenimiento diario del mismos, específicamente refiriéndonos a la administración de componentes las actividades podrían incluir la instalación y desinstalación de los mismos, búsquedas de componentes específicos, habilitación o des habilitación de acuerdo a los requerimientos del sitio web, entre otros.

3.8.3.1. Determinación de Variables

- a. Interfaz para la gestión
- b. Instalación / desinstalación de extensiones
- c. Soporte multilinguaje

3.8.3.2. Valoraciones

a. Variable Interfaz para la gestión

Un administrador puede personalizar los elementos de contenidos que se pueden ver y utilizar dentro de la interfaz, dejándole un control total o bien limitando dentro de las opciones disponibles para simplificar el trabajo de los diferentes tipos de usuarios, lo importante es todas las prestaciones que brinda la interfaz de cada CMS el momento de gestionar los componentes, valoración 4.

b. Variable Instalación / desinstalación de extensiones

La instalación y desinstalación de extensiones es una de las principales fortalezas de los CMS's. Debido a la rapidez con que se desarrollan nuevas extensiones es muy importante la facilidad de esta variable ya que con esto se dota al proyecto de una flexibilidad que le permite incorporar nuevas

funcionalidades día a día, aquí se hace referencia al desarrollo del módulo 4, valoración 4.

c. Variable Soporte multilinguaje

Una aplicación informática, se dice que esta es multilinguaje cuando su Interfaz de usuario puede ser mostrada a elección del usuario en cualquiera de diferentes idiomas, en este caso particular en el idioma español, otro punto importante ya que la mayoría de extensiones se las desarrolla en inglés por lo que se lo valora si los CMS´s poseen la función o el soporte para traducción de los mismos, valoración 4.

Tabla III. V: Gestión de componentes dentro del CMS

Variable	Joomla!	Typo3	Drupal
Interfaz para la gestión	Muy eficiente	Muy eficiente	Eficiente
Instalación / desinstalación de extensiones	Muy eficiente	Muy eficiente	Muy eficiente
Soporte multilinguaje	Se cumple plenamente	Se cumple aceptablemente	Se cumple aceptablemente

3.8.3.3. Interpretación de la Tabla III.V

- La interfaz para la gestión de contenidos refleja la estructura real del sitio web desde el punto de vista del desarrollador, se encuentra que en los tres CMS´s analizados es de tipo amigable, cuando se habla de Joomla! se refiere al back end, en el caso de Typo3 al administrador al igual que en Drupal para la gestión. (4,4,3)/4.
- Una vez que ya se ha familiarizado con la interfaz de cada uno de los CMS´s estudiados en el presente trabajo la instalación y desinstalación

de las extensiones se hace muy fácil simplemente ubicándonos en la pestaña o link de instalación (4,4,4)/4.

- Dentro de los tres CMS´s de manera automática o manual se puede establecer el sitio web en múltiples idiomas, en Joomla! puede ser de forma manual (Joom!Fish) o de forma automática (Gtranslate), Typo3 cuenta con una plataforma común de traducciones y Drupal posee características de traducción automática (método propio del Drupal t) (4,3,3)/4.

3.8.3.4. Calificación

$$P_c = \sum(w) = 4+4+4 = 12$$

$$P_{\text{joom}} = \sum(x) = 4+4+4 = 12 \quad C_c\text{-Joom}: P_{\text{joom}}/P_c = (12/12) * 100\% = 100\%$$

$$P_{\text{typo}} = \sum(y) = 4+4+3 = 11 \quad C_c\text{-Typo}: P_{\text{typo}}/P_c = (11/12) * 100\% = 91.67\%$$

$$P_{\text{drup}} = \sum(z) = 3+4+3 = 10 \quad C_c\text{-Drup}: P_{\text{drup}}/P_c = (10/12) * 100\% = 83.33\%$$

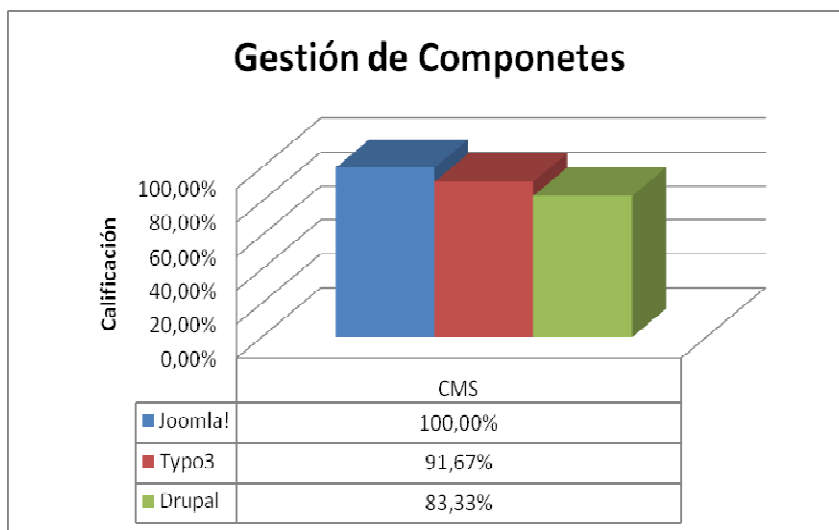


Figura III. 144: Comparación de porcentajes parámetro Facilidad de uso

3.8.4. Estructura de las extensiones

La estructura de las extensiones es muy importante porque estas son la parte fundamental de cada CMS, al conocer su distribución y organización se puede extender su funcionalidad, interactividad, operatividad y optimizar los servicios web además de hacer crecer los sitios desarrollados con cada CMS hacia una escalabilidad sin límites.

3.8.4.1. Determinación de Variables

- a. Uso del esquema MVC (modelo, vista, controlador)
- b. Distribución de los archivos de la extensión
- c. Localización específica de una parte de la extensión

3.8.4.2. Valoraciones

a. Variable Uso del esquema MVC (modelo, vista, controlador)

El modelo de desarrollo más avanzado y actual de Software es el MVC (Modelo Vista Controlador). Su estructura/arquitectura está separada en 3 capas:

- **Modelo:** Almacenamiento y relación de datos (Mysql),
- **Controlador:** lógica funcional y estructuras de control del sistema (PHP), y
- **Vista:** Presentación del aspecto estético (XHTML, CSS, imágenes y multimedia)

Al momento de valorar un CMS es muy importante el uso de este esquema es por esto se lo ha tomado en cuenta, valoración 4.

b. Variable Organización de los archivos de la extensión

Cada uno de los CMS's presenta una organización diferente de los archivos que se van creando para el desarrollo de un componente, la ordenada y

correcta ubicación de estos es lo que nos permitirá realizar correcciones o mejoras de una forma más rápida, los archivos de cada extensión deben estar organizados jerárquicamente, valoración 4.

c. Variable Localización específica de una parte de la extensión

Otro aspecto importante en el desarrollo de extensiones es saber en qué parte del archivo de la misma se encuentra tal o cual función con la utilización del esquema MVC esta tarea se nos hace mucho más fácil, valoración 4.

Tabla III. VI: Estructura de las extensiones

Variable	Joomla!	Typo3	Drupal
Uso del esquema MVC	Si	No	No
Organización de los archivos de la extensión	Muy eficiente	Poco eficiente	Eficiente
Localización específica de una parte de la extensión	Muy eficiente	Eficiente	Poco eficiente

3.8.4.3. Interpretación de la Tabla III.VI

- En el desarrollo de aplicaciones y más aún en el de tipo web se ve cada vez más el uso del esquema MVC por su óptimo ordenamiento y obtención de resultados, mediante el presente trabajo se pudo determinar que tanto Joomla! Como Typo3 entran en el uso de este esquema en tanto que Drupal no lo hace (4,0,0)/4.
- Debido al uso del esquema MVC se puede determinar una mejor organización de los archivos de la extensión en una manera jerárquica pero cabe destacar que aunque Drupal no usa este esquema la

organización de las extensiones es sumamente simple ya que se divide en tres archivos simplemente (4,2,3)/4.

- Tanto la utilización del esquema MVC como la organización de los archivos de la extensión y la localización específica de una parte de la extensión son variables que se encuentran íntimamente relacionadas debido a que en el caso de Joomla! y Typo3 al si utilizar MVC, presentan un mejor ordenamiento de los archivos de la extensión y al entender esta organización la localización de una parte específica de la misma se hace una tarea más sencilla, en el caso de Drupal no usa MVC pero sin embargo su organización es simple, y la localización es muy compleja debido a que la mayoría de la programación se hace en un solo archivo. (4,3,2)/4.

3.8.4.4. Calificación

$$P_c = \sum(w) = 4+4+4 = 4$$

$$P_{joom} = \sum(x) = 4+4+4 = 4 \quad Cc\text{-Joom: } P_{joom}/P_c = (12/12) * 100\% = 100\%$$

$$P_{typo} = \sum(y) = 0+2+3 = 5 \quad Cc\text{-Typo: } P_{typo} / P_c = (5/12) * 100\% = 41.67\%$$

$$P_{drup} = \sum(z) = 0+3+2 = 5 \quad Cc\text{-Drup: } P_{drup} / P_c = (5/12) * 100\% = 41.67\%$$

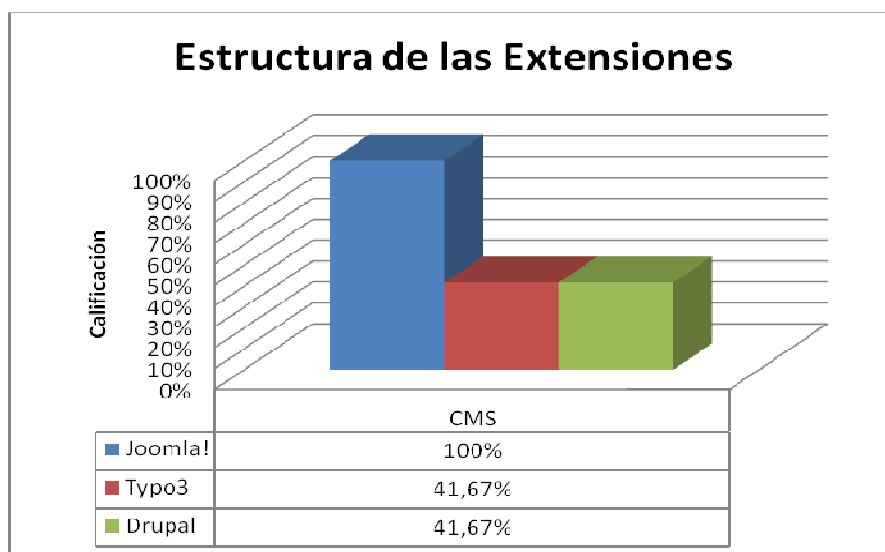


Figura III. 145: Estructura de las Extensiones

3.9. Puntajes Alcanzados

El puntaje y el porcentaje que ha obtenido cada Sistema de Gestión de Contenido se obtiene de la siguiente manera:

Puntaje Total del Análisis : (PT) = Σ (Pc)

Puntaje Total de Joomla! : (PTJoom) = Σ (Pjoom)

Puntaje Total de Typo3: (PTTypo) = Σ (Ptypo)

Puntaje Total de Drupal: (PTDrupal) = Σ (Pdrup)

Porcentaje Total de Joomla: (% Joom) = (PTJoom/PT) * 100%

Porcentaje Total de Typo3: (% Typo) = (PTTypo/PT) * 100%

Porcentaje Total de Drupal: (% Drup) = (PTDrupal/PT) * 100%

Tabla III. VII: Tabla General de Resultados

Parámetro	Variable	Joomla!	Typo3	Drupal
Requerimientos para el desarrollo	• Plataforma	2	2	2
	• Soporte	4	2	3
	• Seguridad	3	3	3
Facilidad de uso	• Interfaz de usuario	4	2	3
	• Acceso a datos	4	3	3
	• Flexibilidad	4	3	4
Gestión de componentes dentro del CMS	• Interfaz para la gestión	4	4	3
	• Instalación / desinstalación de extensiones	4	4	4
	• Soporte multilinguaje	4	3	3
Estructura de las extensiones	• Uso del esquema MVC	4	0	0
	• Distribución de los archivos de la extensión	4	2	3
	• Localización específica de una parte de la extensión	4	3	2
Totales		45	31	33

PT = 10 + 12 + 12 + 12 = 46

PTJoom = 9 + 12 + 12 + 12 = 45

PTTypo = 7 + 8 + 11 + 5 = 31

PTDrupal = 8 + 10 + 10 + 5 = 33

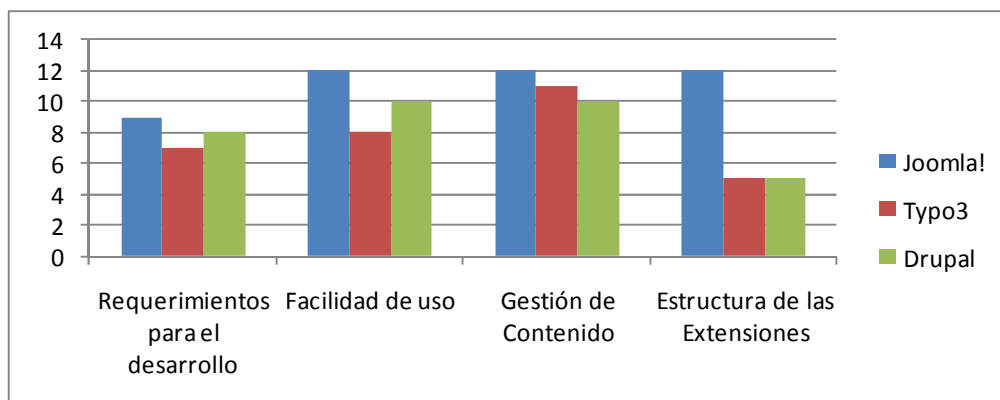


Figura III. 146: Diagrama General de Resultados

(% Joom) = (45/46) * 100 = 97,82% Equivalente a Excelente

(% Typo) = (31/46) * 100 = 67,39% Equivalente a Regular

(% Drup) = (33/46) * 100 = 71,74% Equivalente a Bueno

3.10. Interpretación de la Tabla III.VII y la Figura III.146

Como resultado del análisis y de acuerdo al puntaje alcanzado por cada una de las variables se ha obtenido como resultado que el CMS Joomla! tiene el puntaje más alto con un porcentaje del 97,82% que es equivalente a Excelente.

3.11. Resultado del Análisis de la Tabla III.VII y la Figura III.146

- Joomla!, Typo3 y Drupal son sistemas multiplataforma basadas en PHP que utilizan MySQL como base de datos y el servidor Apache tanto en Linux como en Windows.
- La información de cada uno de los CMS's va aumentando cada vez más, al ser sistemas basados en código libre, la comunidad de cada una de ellas publica regularmente técnicas de programación, seguridades y

temas varios, la información para el desarrollo de componentes en Drupal es menor.

- Los 3 CMS trabajan con templates o plantillas las que se puede crear desde cero o descargar desde la web, facilitando la construcción del sitio web.
- Tienen un desempeño similar con el acceso a datos, al trabajar con MySQL como base de datos.
- En la creación del componente expandido para determinar la flexibilidad no hubo mayor complicación en Joomla! y Typo3 por que ya se conocía la distribución de los archivos necesarios para el desarrollo de componentes, al trabajar con Drupal hubo más inconvenientes por que en un solo archivo se maneja la programación del mismo.
- Joomla! posee una excelente diferenciación del Front-End (usuario) y del Back-End (desarrollador), Drupal y Typo3 presentan interfaces similares para estos, resultando en general interfaces totalmente amigables.
- La instalación y desinstalación de componentes resulta intuitiva cuando ya nos familiarizamos con el entorno de cada CMS ya que los tres poseen un gestor de instalación.
- Drupal posee una función multilinguaje interna que trabaja de forma automática al momento de instalarlo, Joomla! y Typo3 también poseen soporte multilinguaje.
- En el desarrollo de aplicaciones en nuestro caso de tipo web es muy importante el usos del esquema MVC que nos permite un óptimo ordenamiento y obtención de resultados, además es de mucha ayuda al momento de reutilizar y adaptar un componente ya que presenta una

organización de los archivos del mismo y resulta más sencillo encontrar una parte específica del código.

- Typo3 y Drupal no utilizan el esquema MVC, sin embargo presenta una organización propia de archivos del componente, en el caso de Drupal es muy complicada al momento de tratar de encontrar una parte específica de la extensión.

3.12. Conclusión

Por todo lo expuesto anteriormente y de acuerdo a los puntajes alcanzados para cada uno de los parámetros de evaluación se puede concluir que el CMS Joomla! es el que brinda las mejores prestaciones para el desarrollo de componentes en sistemas de administración de contenido.

3.13. Comprobación de la Hipótesis

Mediante la comparación del desarrollo de componentes en sistemas de administración de contenido se desea encontrar el mejor CMS para la implementación del portal web de la librería Majo's, se delimitó tres CMS's para la comparación, siendo los escogidos Joomla!, Drupal y Typo3, para cada uno de ellos se estableció cuatro parámetros y sus tres variables de medición pertenecientes a cada uno de los parámetros mencionados.

Después de la investigación, el desarrollo de los cuatro módulos y la valoración de los tres CMS's expuestos en el presente capítulo se obtienen valores denominados **sin comparación**.

Para llegar a la demostración de la hipótesis con valores porcentuales se compara los valores de Joomla! con Typo3 en la figura, Joomla! con Drupal

en la figura y Typo3 con Drupal, con referencia al valor máximo de cada parámetro para obtener los valores denominados **con comparación**.

Tabla III. VIII: Comparación Joomla! – Typo3

Parámetro	Valor Máximo del Parámetro (100%)	Joomla!	Cc-Joom	Typo3	Cc-typo	Mejor CMS
1	10 - 100%	9	90%	7	70%	Joomla! - 20%
2	12 - 100%	12	100%	8	66,67%	Joomla! - 33,33%
3	12 - 100%	12	100%	11	91,67%	Joomla! - 8,33%
4	12 - 100%	12	100%	5	41,67%	Joomla! - 58,33%
Promedio	11,5 - 100%	11,25	97,5%	7,75	67,50%	Joomla - 30%

Tabla III. IX: Comparación Joomla! - Drupal

Parámetro	Valor Máximo del Parámetro (100%)	Joomla!	Cc-Joom	Drupal	Cc-Drup	Mejor CMS
1	10 - 100%	9	90%	8	80%	Joomla! - 10%
2	12 - 100%	12	100%	10	83,33%	Joomla! - 16,67%
3	12 - 100%	12	100%	10	83,33%	Joomla! - 16,67%
4	12 - 100%	12	100%	5	41,67%	Joomla! - 58,33%
Promedio	11,5 - 100%	11,25	97,5%	8,25	72,08%	Joomla - 25,4%

Tabla III. X: Comparación Typo3 - Drupal

Parámetro	Valor Máximo del Parámetro (100%)	Typo3	Cc-Typo	Drupal	Cc-Drup	Mejor CMS
1	10 - 100%	7	70%	8	80%	Drupal - 10%
2	12 - 100%	8	66.67%	10	83,33%	Drupal - 16,66%
3	12 - 100%	11	91,67%	10	83,33%	Drupal - (-8,34%)
4	12 - 100%	5	41,67%	5	41,67%	Igual
Promedio	11,5 - 100%	7,75	67,5%	8,25	72,08%	Drupal 4,58%

En la comparación expuesta en las tablas: Tabla III.VIII, Tabla III.IX y la Tabla III.X, se determina que entre Joomla! y Typo3, Joomla! es 30% mejor, entre Joomla! y Drupal, Joomla! es 25,4% mejor y entre Typo3 y Drupal, Drupal es 4,58% mejor, por lo que Joomla! es el mejor de los tres CMS, para llegar a esta conclusión en un valor porcentual confrontamos los valores **sin comparación** y **con comparación** en la siguiente tabla:

Tabla III. XI: Mejor CMS (Sin comparación – Con Comparación)

Parámetros	Sin Comparación			Con Comparación
	Cc- Joom	Cc- Drup	Cc- Typo3	Mejor CMS
1	90%	80%	70%	Joomla! - 15%
2	100%	83,33%	66,67%	Joomla! - 25%
3	100%	83,33%	91,67%	Joomla! - 12,5%
4	100%	41,67%	41,67%	Joomla! - 58,33%
Promedio	97,5%	72,08%	67,5%	Joomla - 27,71%

De esta manera queda demostrado que en el desarrollo de componentes en Sistemas de Administración de Contenido en un 27,71%, Joomla! es mejor en comparación con Drupal y Typo3 para la implementación del portal web de la librería Majo ´s como se ve en la Tabla III.XI.

CAPÍTULO IV

DESARROLLO DEL PORTAL WEB

4.1. Parte Teórica

4.1.1. Introducción

En este apartado se tratara acerca de los conceptos básicos, las herramientas que se usarán para el desarrollo del portal web. Las herramientas usadas serán: PHP para el desarrollo del componente web en Joomla!, y como base de datos MySQL, las mismas que son de licencia libre. Además se dará las fases de la metodología que he utilizado para la creación del mismo.

4.1.2. Desarrollo

Para el Modelo escogido se debe seguir el método de desarrollo orientado a objetos que propone Craig Larman. Este proceso no fija una metodología estricta, sino que define una serie de actividades que pueden realizarse en cada fase, las cuales deben adaptarse según las condiciones del proyecto que se esté llevando a cabo como se muestra en la figura 4.1. Se ha escogido seguir este proceso debido a que aplica los últimos avances en Ingeniería del Software, y a que adopta un enfoque eminentemente práctico, aportando soluciones a las principales dudas y/o problemas con los que se enfrenta el desarrollador. Su mayor aportación consiste en atar los cabos sueltos que anteriores métodos dejan.

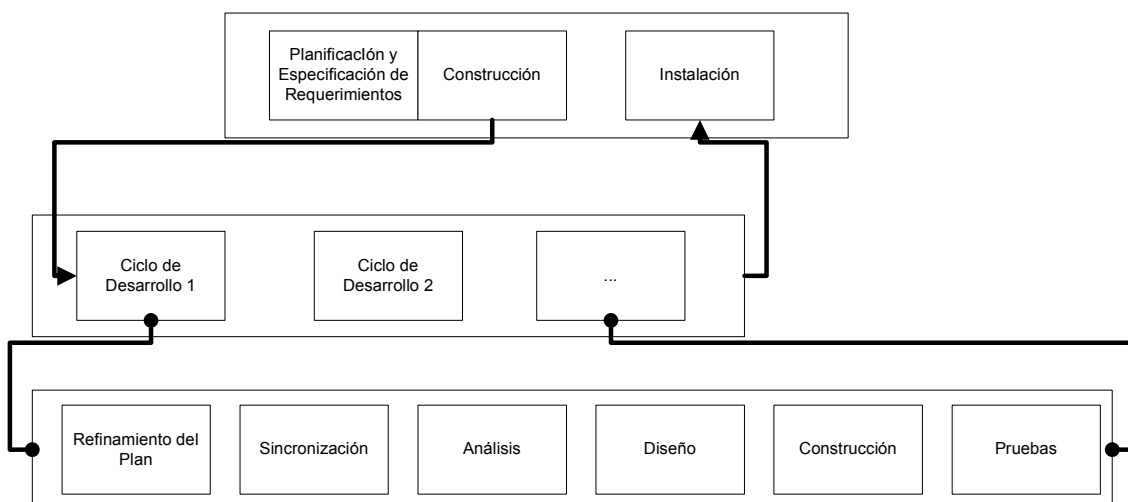


Figura IV. 147: Desarrollo Iterativo de la Aplicación

La notación que se usa para los distintos modelos, es la proporcionada por UML, que se ha convertido en el estándar de facto en cuanto a notación orientada a objetos. El uso de UML permite integrar con mayor facilidad en el equipo de desarrollo a nuevos miembros y compartir con otros equipos la documentación, pues es de esperar que cualquier desarrollador versado en orientación a objetos conozca y use UML (o se esté planteando su uso).

Se va a abarcar todo el ciclo de vida, empezando por los requisitos y acabando en el sistema funcionando, proporcionando así una visión completa y coherente de la producción de sistemas software. El enfoque que toma es el de un ciclo de vida iterativo incremental, el cual permite una gran flexibilidad a la hora de adaptarlo a un proyecto y a un equipo de desarrollo específicos. El ciclo de vida está dirigido por casos de uso, es decir, por la funcionalidad que ofrece el sistema a los futuros usuarios del mismo. Así no se pierde de vista la motivación principal que debería estar en cualquier proceso de construcción de software: el resolver una necesidad del usuario/cliente.

I. Ingeniería de la Información

- I.1 Definición del Ámbito del Software
- I.2. Antecedentes Tecnológicos
 - I.2.1. Recursos Hardware
 - I.2.2. Recursos Software
 - I.2.3. Recursos Humanos
- I.3 Definición del Problema
 - I.3.1 Definir Alternativas de Solución

II. Análisis Orientado a Objetos

- II.1. Caso de Uso General del Sistema Propuesto
- II.2. Diagrama de Caso de Uso General
- II.3. Definir el Modelo Conceptual
- II.4. Definir los Diagramas de Secuencia

III. Diseño Orientado a Objetos

- III.1. Diagrama de base de datos
- III.2. Diagrama de Componentes

IV. Implementación

V. Pruebas

I. Ingeniería de la Información

Se refiere a la fase en donde se definen los requerimientos de los usuarios y del sistema mismo, de manera que se muestra aquí todo lo funcional y no funcional para el desarrollo de la aplicación.

II. Análisis Orientado a Objetos

Se definen los objetos de la aplicación es decir las clases para la implementación del sistema, y se los analiza en busca de la solución

III. Diseño Orientado a Objetos

Luego de analizar los objetos del sistema se diseña el prototipo de la solución, la misma que será orientada a objetos netamente.

IV. Implementación

Esta fase trata de abarcar todo lo que se refiere a la programación de la aplicación web en las herramientas software que se utilizaron para el desarrollo de la solución.

V. Pruebas

Luego de haber desarrollado la aplicación, a partir de los requerimientos, el diseño de la solución, la programación misma, se requiere realizar pruebas sobre esta, para enviar a producción con la aprobación del usuario final.

Herramientas utilizadas:

- **PHP**

PHP proporciona una variedad de beneficios tanto para desarrolladores individuales como para equipos de desarrollo:

- Es un lenguaje multiplataforma.

- Completamente orientado al desarrollo de aplicaciones web dinámicas con acceso a información almacenada en una Base de Datos.
- El código fuente escrito en PHP es invisible al navegador y al cliente ya que es el servidor el que se encarga de ejecutar el código y enviar su resultado HTML al navegador. Esto hace que la programación en PHP sea segura y confiable.
- Capacidad de conexión con la mayoría de los motores de base de datos que se utilizan en la actualidad, destaca su conectividad con MySQL y PostgreSQL.
- Capacidad de expandir su potencial utilizando la enorme cantidad de módulos (llamados ext's o extensiones).
- Posee una amplia documentación en su página oficial, entre la cual se destaca que todas las funciones del sistema están explicadas y ejemplificadas en un único archivo de ayuda.
- Es libre, por lo que se presenta como una alternativa de fácil acceso para todos.
- Permite aplicar técnicas de programación orientada a objetos.
- Biblioteca nativa de funciones sumamente amplia e incluida.
- No requiere definición de tipos de variables aunque sus variables se pueden evaluar también por el tipo que estén manejando en tiempo de ejecución.
- Tiene manejo de excepciones (desde PHP5).
- Si bien PHP no obliga a quien lo usa a seguir una determinada metodología a la hora de programar (muchos otros lenguajes tampoco lo hacen), aun estando dirigido a alguna en particular, el programador puede aplicar en su trabajo cualquier técnica de programación y/o desarrollo que le permita escribir código ordenado, estructurado y manejable. Un ejemplo de esto son los desarrollos que en PHP se han hecho del patrón de diseño Modelo Vista Controlador (o MVC), que

permiten separar el tratamiento y acceso a los datos, la lógica de control y la interfaz de usuario en tres componentes independientes.

- **MySql.**

Para la realización de la base de datos en la Asociación de Fútbol no Aficionado de Chimborazo lo hemos hecho con el motor de base de datos o DBMS denominado MySql, el mismo que nos ayuda básicamente a desarrollar bases de datos consistentes, versátiles y además con la posibilidad de trabajar con nuestra aplicación hecha en PHP ya que es totalmente compatible con ese programa. Este Motor de bases de datos se encuentra en lenguaje Transact-SQL, y nos ofrece un sin número de servicios como el de administrar nuestra base de datos, diagramar el mismo, realizar consultas de nuestros datos, y lo más importante tenerlos seguros y sin duplicidad de información.

4.2. Parte práctica

4.2.1. Introducción

En esta sección se dará a conocer el desarrollo técnico de los Componentes web para la gestión de los productos de la Librería Majos, puesto que en dicho negocio se requiere una forma de almacenar y gestionar la información de todos los artículos que se ofertan en la Librería y consecuentemente exponerlos en el portal web, dando la opción a los usuarios de revisarlos con sus respectivas especificaciones, y realizar un pedido de los artículos deseados.

Para solucionar este requerimiento se ha coincidido con los usuarios finales el desarrollo de una extensión en Joomla! que permita almacenar, gestionar y reportear la información, y que además presente los resultados de forma

ordenada, y como se ha explicado en la parte teórica escogiendo herramientas de desarrollo y de bases de datos a MySQL como DBMS y PHP para desarrollar la extensión (componente). Para la Creación de los Componentes es importante seguir con la metodología planteada anteriormente.

4.2.2. Desarrollo

4.2.2.1. Ingeniería de la Información

4.2.2.1.1. Definición del Ámbito del Software

Actualmente la Librería Majos, no cuenta con un sistema que permita tener en un almacén de datos electrónico de los productos que dispone, así como de un sistema que permita ver al usuario los artículos que se ofertan, todo este proceso lo realiza manualmente ingresando datos en un inventario en documentos físicos, ocasionando pérdidas y confusiones. Para obtener la información necesaria, realizamos Entrevistas, Encuestas con el fin de recopilar los requerimientos necesarios.

Entrevistas

Para facilitar la definición del ámbito del problema, así como también de los requerimientos que se desea solucionar, se han realizado entrevistas a todas las personas involucradas en la gestión de los artículos de la librería Majos.

4.2.2.1.2. Antecedentes Tecnológicos

4.2.2.1.2.1. Recursos Hardware

Tabla IV. XII: Características Hardware para el Sistema

Cantidad: 1 PC	
Mainboard	INTEL 945 MBR
Procesador	INTEL PENTIUM IV
Velocidad	3.0 GHz
Memoria	1GB DDRAM
Disco Duro	250 GB
CD-ROM	DVDRW
Teclado	WIN XP PS/2
Monitor	LG Digital 17"

4.2.2.1.2.2. Recursos Software

Tabla IV. XIII: Características Software para el Sistema

Características	
Instalador de S.O	Windows XP
Lenguajes de Programación	Adobe Dreamweaver CS4, PHP
Base de Datos	MySql
Dominio	www.majoslibreria.com
Antivirus	Nod 32 Actualizaciones
Otros	Mozilla Firefox Office 2007 XAMPP FileZilla Client

4.2.2.1.2.3. Recursos Humanos

Tabla IV. XIV: Recursos Humanos

Cargo	Nº de Personas
Programadores	2
Administrador del Sitio Web	1
Editor de Contenidos	1

4.2.2.1.3. Definición del Problema

La Librería Majos cuenta con varias funciones para la gestión de los artículos de la librería, realizar las órdenes de los productos a sus proveedores, recibir los productos con sus especificaciones, ofertar los productos al público, tomar las órdenes de venta por parte de los clientes.

Uno de los problemas que se han detectado, es que la Librería Majos requiere tener un registro en forma electrónica de todos los datos de la gestión de los productos ofertados.

Todo este proceso se lo realiza manualmente por el personal encargado consumiendo tiempo, recursos y esfuerzos. Además pueden existir pérdidas y confusiones de los mismos.

4.2.2.1.3.1. Definir Alternativas de Solución

Es por eso que se ha visto la necesidad de realizar un Sistema que automatice estos controles y que de alguna manera agilite los procesos, logrando de esta manera optimizar tiempos, recursos, costos.

Para la creación del Sistema se ha analizado todos los puntos de vista para poder realizarla teniendo en cuenta que en la Librería Majos no existe

ninguna aplicación relacionada con el tema. Se ha recopilado información necesaria con el apoyo de las personas encargadas.

Es por ello que se decidió, realizar un sistema que específicamente lo manejen y administren en la Librería Majos, y que toda la comunidad pueda acceder a la misma mediante el sitio web www.majoslibreria.com.

4.2.2.2. Análisis Orientado a Objetos

Una técnica excelente que permite mejorar la comprensión de los requerimientos es la creación de los casos de uso, es decir, descripciones narrativas de los procesos que se van a realizar para la creación del sistema que automatice la gestión de los artículos de la Librería Majos.

Los casos de uso son historias o casos de utilización de un sistema; no son exactamente los requerimientos ni las especificaciones funcionales, sino que ejemplifica e incluyen tácitamente los requerimientos en las historias que narran.

4.2.2.2.1. Caso de Uso: Sistema Actual

A continuación se muestra el caso de uso para el sistema actual que se utiliza en la Librería Majos ver Tabla IV.4, para registrar los artículos de librería, para su posterior oferta dentro de la Librería al público en general:

Tabla IV. XV: Caso de Uso 1: Sistema Actual

Caso de Uso:		Ingreso al Sistema	
Actores:	Propietario		
Tipo:	Esencial Primario		
Descripción:	El propietario realiza los pedidos de los artículos a los proveedores, recibe el pedido, y los organiza según la categoría que correspondan. Consecuentemente fija el precio de venta al público y los oferta a la comunidad, si un cliente opta por realizar un pedido de artículos específicos, el propietario recibe el pedido y lo procesa.		
Propósito	Describir como se realiza el proceso del sistema actual.		
Curso típico de eventos			
Actores	Sistema		
<ol style="list-style-type: none"> 1. El propietario realiza los pedidos de los artículos al o los proveedores. 2. El proveedor realiza la entrega de los artículos al propietario de la librería majos. 3. El propietario recibe los artículos y los organiza según su categoría. 4. El propietario fija el precio de venta al público de cada artículo. 5. El propietario promociona los artículos al público, ofreciendo descuentos, ofertas en artículos seleccionados. 6. El propietario recepta los pedidos de los clientes de los artículos a comprar. 7. El propietario entrega los artículos requeridos, con su precio final del pedido al cliente. 			

4.2.2.2.2. Caso de Uso 2

Tabla IV. XVI: Caso de Uso 2: Ingreso al Sistema

Caso de Uso:		Ingreso al Sistema	
Actores:		Propietario	
Tipo:		Esencial Primario	
Descripción:	El propietario ingresa a la Base de Datos para ingresar o modificar los datos de los proveedores.		
Propósito	Describir como se realiza el proceso de ingreso		
Curso típico de eventos			
Actores		Sistema	
1. El propietario se logea al sistema mediante interfaz gráfica e ingresa los datos de sus proveedores.		2. Lo admite y lo autoriza para poder ingresar y modificar los datos.	
		3. Emite mensaje "OK"	
1. Ingresa al DBMS MySQL, mediante interfaz gráfica e ingresa datos de proveedores			

4.2.2.2.2.1. Diagrama de Caso de Uso 2

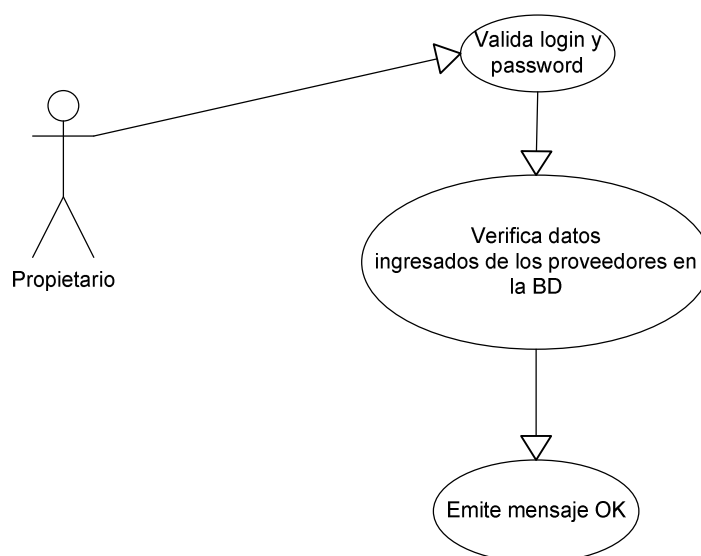


Figura IV. 148: Diagrama de Caso de Uso 2

4.2.2.2.3. Caso de Uso 3

Tabla IV. XVII: Caso de Uso 3: Crear y Gestionar Campeonatos

Caso de Uso:		Ingreso al Sistema
Actores:	Propietario	
Tipo:	Esencial Primario	
Descripción:	El propietario ingresa al sistema, crea los artículos en su respectiva categoría y sub categoría, establece el precio de venta, y gestiona los mismos.	
Propósito	Describir como se realiza el proceso de ingreso de los artículos.	
Curso típico de eventos		
Actores	Sistema	
1. El propietario ingresa al sistema mediante interfaz, crea los artículos en su respectiva categoría, sub categoría, y establece el precio de venta.	2. Lo admite y lo autoriza para poder crear y gestionar los artículos.	
3. El propietario asigna los artículos a su respectivo proveedor, con su precio de compra.	4. Lo admite y guarda los datos de un artículo con su respectivo proveedor, y emite el mensaje "OK".	
1. Ingresa a la aplicación a crear y gestionar los artículos.		

4.2.2.2.3.1. Diagrama de Caso de Uso 3

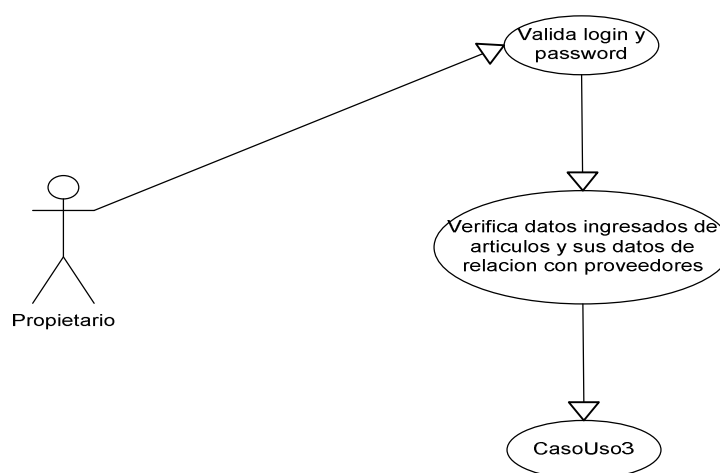


Figura IV. 149: Diagrama de Caso de Uso 3

4.2.2.2.4. Caso de Uso 4

Tabla IV. XVIII: Caso de Uso 4: Emitir Reportes

Caso de Uso:		Ingreso al Sistema
Actores:	Administrador	
Tipo:	Esencial Primario	
Descripción:	La Secretaria ingresa a la interfaz de usuario y escoge la opción de emitir reportes y el sistema acoge esa instrucción la secretaria escoge los diferentes criterios de selección y el sistema devuelve en pantalla la información pertinente.	
Propósito	Describir como se realiza el proceso para la emisión de reportes.	
Curso típico de eventos		
Actores	Sistema	
1. La Secretaria selecciona la opción de emitir reportes	2.- Acoge la instrucción.	
3.- La secretaria escoge el criterio de selección que prefiere.	4.- Busca la información de acuerdo al criterio seleccionado y la lista en tablas con el listado de los datos.	
4.- El sistema emite la información de los pedidos de artículos por parte de los clientes,		

4.2.2.2.4.1. Diagrama de Caso de Uso 4

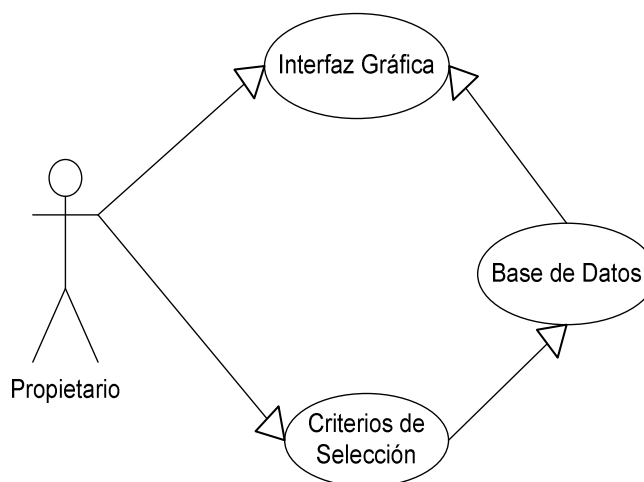


Figura IV. 150: Diagrama de Caso de Uso 4

4.2.2.2.5. Definir el Modelo Conceptual

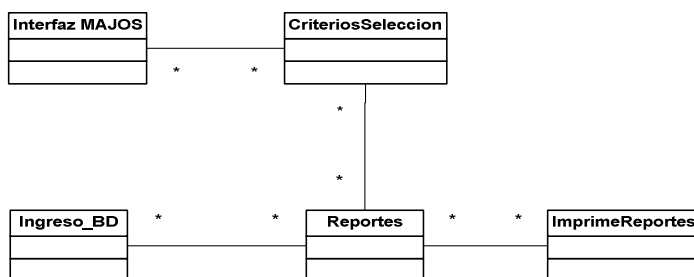


Figura. 151: Modelo Conceptual

4.2.2.2.6. Definir Diagrama de Secuencia

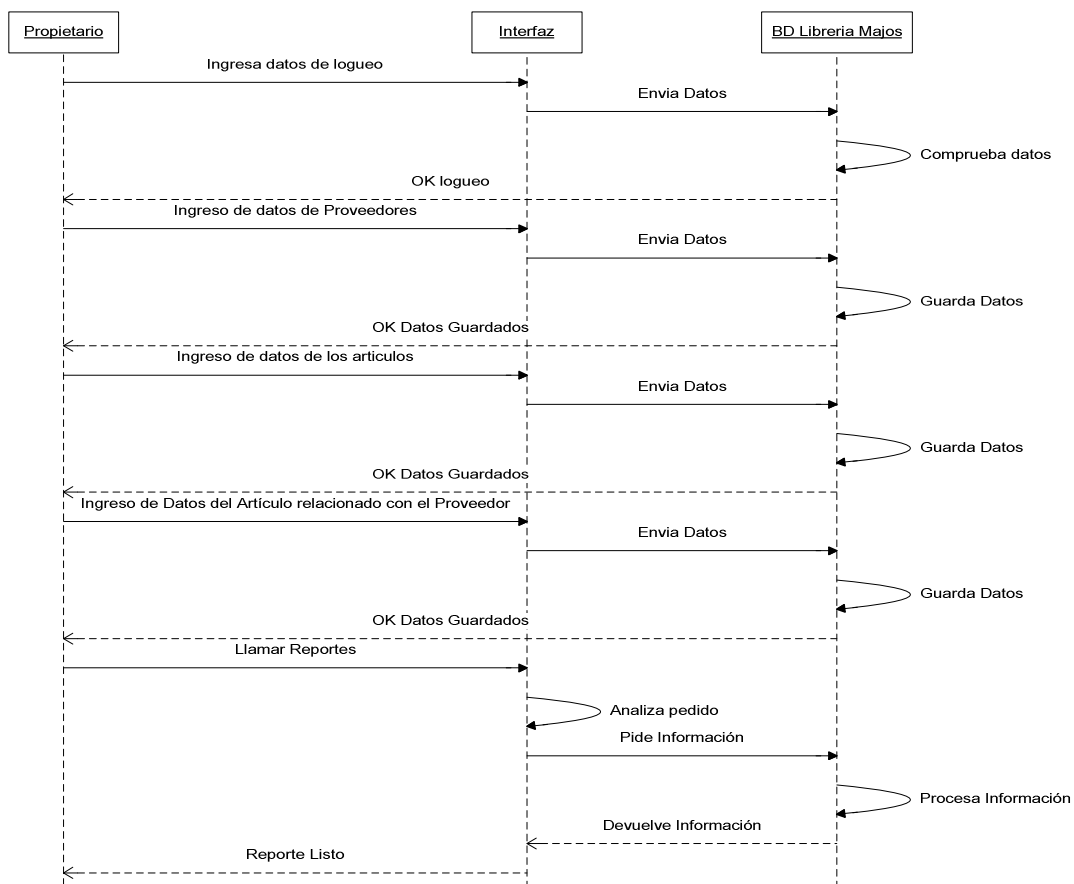


Figura IV. 152: Diagrama de Secuencia

4.2.2.3. Diseño Orientado a Objetos

Los métodos de diseño del software se obtienen del estudio de cada uno de los tres dominios del modelo de análisis, los cuales sirven de guía para la creación del diseño. En esta parte se planteará los primeros pasos para nuestro desarrollo del Software para lo cual plantearemos técnicas y principios para la definición y realización física de un proceso.

4.2.2.3.1. Diagrama de base de datos

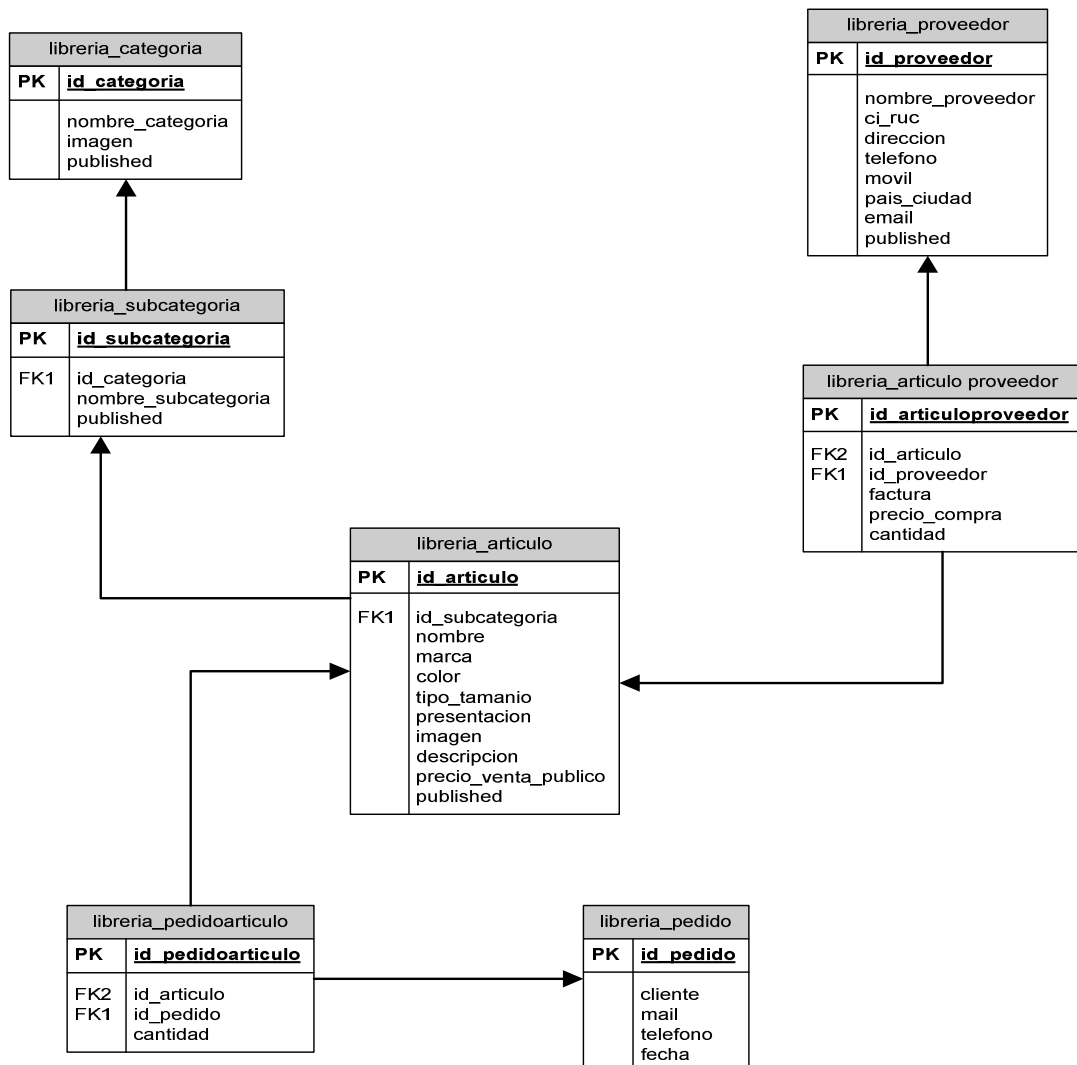


Figura IV. 153: Diagrama de Base de Datos

4.2.2.3.2. Diagrama de Componentes

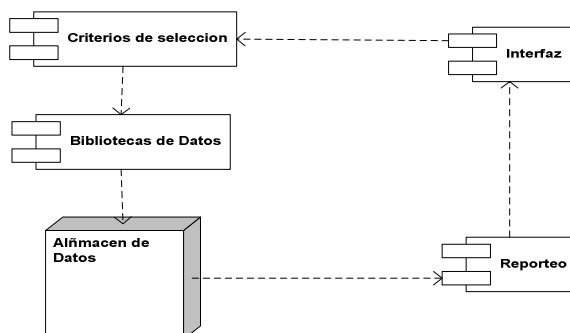


Figura IV. 154: Diagrama de Componentes

4.2.2.4. Implementación

Componente 1: Gestión de Artículos de Librería

El presente componente se encarga de la gestión de los artículos de librería existentes en inventarios, para su fomentación en el sitio web. Consta de dos partes el componente en el back-end para usuarios administrativos, y el componente en el front-end para el público en general.

Back-end

Descripción: El componente gestionará la información de los artículos, clasificándolos en sus respectivas categorías y subcategorías, así como también gestionará los proveedores y las compras de los artículos realizadas a su respectivo proveedor. Este componente se implementará en Joomla 1.5 siguiendo el modelo MVC (modelo - vista - controlador).

Nombre: com_libreria

Ubicación: ..\administrator\components\com_libreria

Estructura de los archivos:siguiendo con el modelo MVC, el componente contendrá:

Controlador controller.php,

La carpeta models: con todos los archivos .php correspondientes a los modelos necesarios para cada vista.

La carpeta views: con todas las vistas necesarias para la visualización en pantalla de los datos solicitados.

Además de otros archivos necesarios para el correcto funcionamiento del componente, como se muestra en la Figura IV.155.

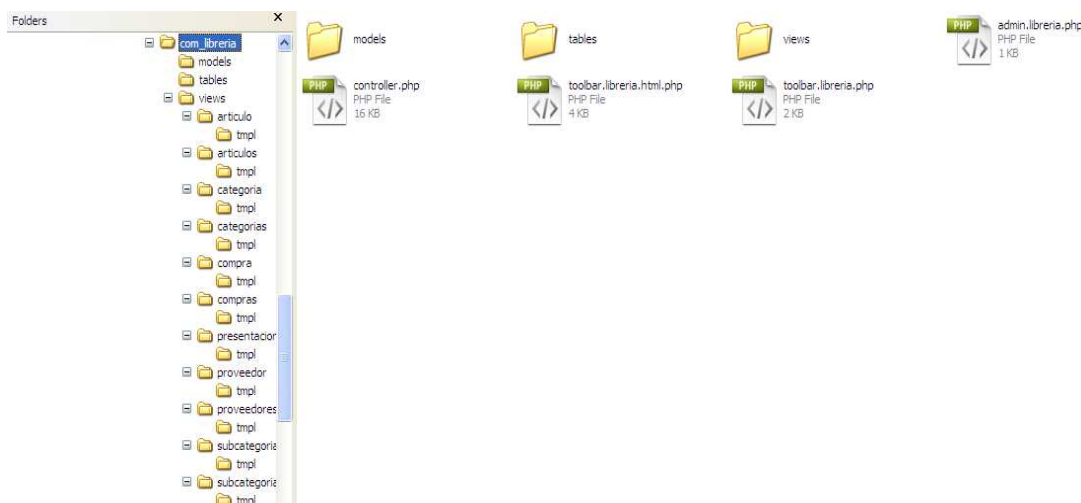


Figura IV. 155: Estructura del componente com_libreria, back-end.

Pantallas de visualización:

La pantalla de visualización inicial del componente. Ver Figura IV.156



Figura IV. 156: Pantalla inicial del componente com_libreria, back-end.

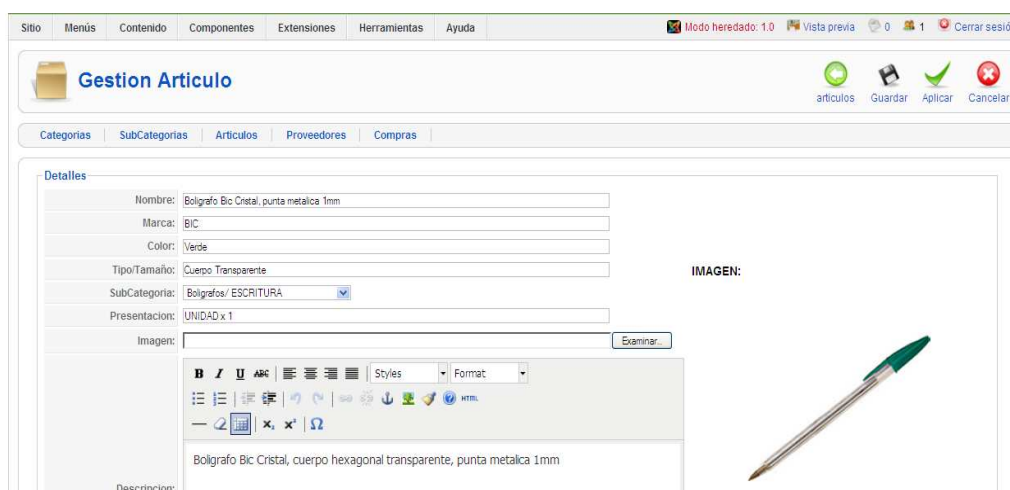
Los listados de los datos se visualizan como se muestra en la Figura IV.157.



<input type="checkbox"/>	Nombre	Marca	Color	Tipo/Tamaño	SubCategoría/Categoría	Presentación	Precio de Venta	Publicado
<input type="checkbox"/>	Boligrafo Bic Cristal, punta metálica 1mm	BIC	Verde	Cuerpo Transparente	Boligrafos - ESCRITURA	UNIDAD x 1	1.08	<input checked="" type="checkbox"/>
<input type="checkbox"/>	Borratintas Pelikan Super Pirat	PELIKAN	Transparente	Borra Tinta	Correctores - ESCRITURA	UNIDAD x 1	3.71	<input checked="" type="checkbox"/>
<input type="checkbox"/>	Cartucho Parker de tinta para lapicera Parker, pack x 5	PARKER	Azul	De Tinta	Cartucho para lapicera - ESCRITURA	Pack x 5 x 1	5.63	<input checked="" type="checkbox"/>
<input type="checkbox"/>	Cinta correctora Isofit de 5mm x 8 m	ISOFIT	Blanco	Cinta correctora	Correctores - ESCRITURA	UNIDAD x 1	3.71	<input checked="" type="checkbox"/>
<input type="checkbox"/>	Corrector Liquid Paper 2 en 1	PAPER MATE	BLANCO	2 en 1	Correctores - ESCRITURA	UNIDAD x 1	9.6	<input checked="" type="checkbox"/>
<input type="checkbox"/>	Goma de borrar Faber Castell para lapiz	FABER CASTELL	Blanco	Para lapiz	Gomas - ESCRITURA	UNIDAD x 1	0.95	<input checked="" type="checkbox"/>
<input type="checkbox"/>	Lapicera Lamy Safari de cuerpo plastico negro	LAMY	Negro	Safari	Lapicera - ESCRITURA	UNIDAD x 1	44.78	<input checked="" type="checkbox"/>
<input type="checkbox"/>	Lapices de color Faber Castell x 12 + sacapuntas	FABER CASTELL	Multicolor	12 Colores	Lapices de Color - ESCRITURA	UNIDAD x 1	8.18	<input checked="" type="checkbox"/>
<input type="checkbox"/>	Lapiz con goma Faber Castell Hexagonal	FABER CASTELL	Negro	Mina tipo HB	Lapices - ESCRITURA	UNIDAD x 1	0.74	<input checked="" type="checkbox"/>

Figura IV. 157: Forma de visualización de los listados de datos, back-end

La visualización de formularios, ver Figura IV.158.



Formulario de gestión de artículos con los siguientes campos:

- Nombre: Boligrafo Bic Cristal, punta metálica 1mm
- Marca: BIC
- Color: Verde
- Tipo/Tamaño: Cuerpo Transparente
- SubCategoría: Boligrafos/ ESCRITURA
- Presentación: UNIDAD x 1
- Imagen: Examinar...
- Descripción: Boligrafo Bic Cristal, cuerpo hexagonal transparente, punta metálica 1mm

El formulario incluye una barra de herramientas de texto y un espacio reservado para la imagen del artículo.

Figura IV. 158: Forma de visualización de formularios, back-end

Front-end

Descripción: El componente visualizará en pantalla la información de los artículos: su nombre, marca, tipo, presentación, descripción, imagen y precio de venta al público. Filtrados con su respectivas categoría y subcategoría. Este componente se implementará en Joomla 1.5 siguiendo el modelo MVC (modelo - vista - controlador).

Nombre: com_libreria

Ubicación: ..\components\com_libreria

Estructura de los archivos: siguiendo con el modelo MVC, el componente contendrá:

Controlador controller.php,

La carpeta models: con todos los archivos .php correspondientes a los modelos necesarios para cada vista.

La carpeta views: con todas las vistas necesarias para la visualización en pantalla de los datos solicitados.

Además de otros archivos necesarios para el correcto funcionamiento del componente, como se muestra en la Figura IV.159.

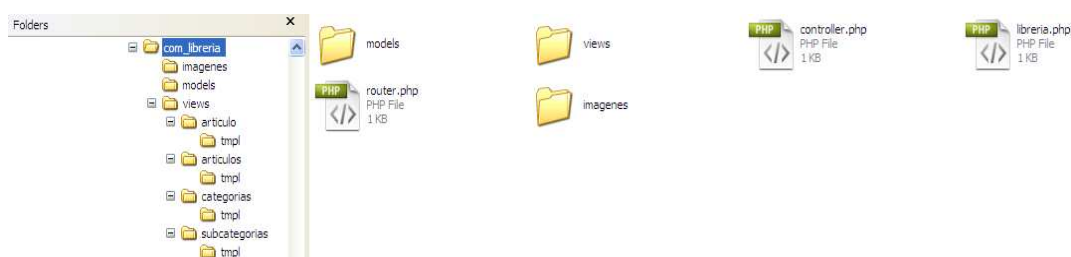


Figura IV. 159: Estructura del componente com_libreria,front-end

Pantallas de visualización:

Forma de visualización de los listados, de los criterios de selección a las que pertenecen los artículos existentes, como se muestra en la Figura IV.160.



Figura IV. 160: Forma de visualización de los listados, de los filtros a los que pertenecen los artículos de librería, front-end

Visualización de un artículo específico escogido. Ver Figura IV.161.



Figura IV. 161: Forma de visualización de un artículo de librería, front-end

Componente 2: Reserva de Artículos mediante el Carrito de compras

Este componente dará la opción a los usuarios clientes, de reservar artículos seleccionados en un carrito de compras emitiendo un recibo que se podrá imprimir, el cual servirá como documento para su posterior pedido. Para el usuario administrativo mostrara las órdenes de pedidos en detalle, para su comprobación con el documento que se recibirá por parte del cliente, para su posterior entrega de los artículos de la orden respectiva.

Back-end

Descripción: El componente mostrara al usuario administrativo, el listado de las ordenes de pedidos realizadas por los clientes, también mostrará en detalle cada orden de compra, visualizando el número de orden, datos del cliente, y los artículos con su cantidad que fueron reservados. Este componente se implementará en Joomla 1.5.

Nombre: com_libreria_cart

Ubicación: ..\administrator\components\com_libreria_cart

Estructura de los archivos: siguiendo con el modelo MVC, el componente contendrá:

Controlador controller.php.

La carpeta models: con todos los archivos .php correspondientes a los modelos necesarios para cada vista.

La carpeta views: con todas las vistas necesarias para la visualización en pantalla de los datos solicitados.

Además de otros archivos necesarios para el correcto funcionamiento del componente, como se muestra en la Figura IV.162.

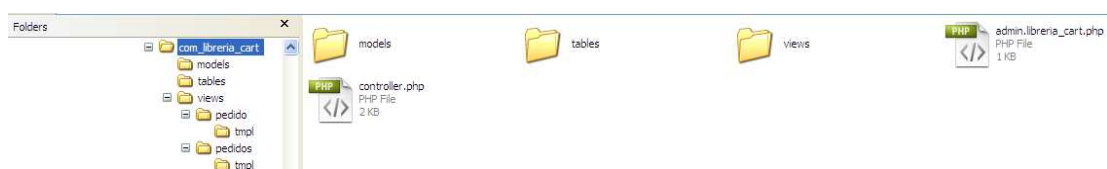


Figura IV. 162: Estructura del componente com_libreria_cart, back-end

Pantallas de visualización:

Forma de visualización del listado de ordenes realizadas por los clientes, ver Figura IV.163.

# Pedido	Fecha	Cliente	e-mail
1	2010-08-24 17:31:56	Victor Oquendo	vmoc_man123@yahoo.com
2	2010-08-24 18:04:52	Victor Oquendo C	vmoc_man23@yahoo.com
3	2010-08-24 18:17:23	Victor Manuel Oquendo C.	vmoc_man1234@yahoo.com
4	2010-08-24 18:18:50	Manolo	vmoc_man1231@yahoo.com
5	2010-08-24 18:20:45	Manolo O	vmoc_man1123@yahoo.com
6	2010-08-27 19:48:33	Daniela	Vizuete
7	2010-08-27 19:50:11	Daniela	Vizuete
8	2010-08-27 19:50:15	Daniela	Vizuete
9	2010-08-27 19:50:24	Daniela	Vizuete
10	2010-08-30 16:42:38	Phil Evey	phil@hotmail.com
11	2010-08-30 17:27:17	Chris Ferguson	chrisferguson@hotmail.com
12	2010-08-30 17:30:31	usuario1	usuario1@hotmail.com
13	2010-08-30 18:01:40	usuario2	usuario2@hotmail.com
14	2010-08-30 18:05:09	usuario3	usuario3@hotmail.com
15	2010-08-30 18:06:15	usuario4	usuario4@hotmail.com

Figura IV. 163: Listado de los pedidos de artículos realizados, back-end

La forma de visualizar un pedido en detalle se muestra en la Figura IV.164.

Artículo	P. Unitario	Cantidad	Total
Cartucho Parker de tinta para lapicera Parker, pack x 5	\$ 5.63	7	\$ 39.41
Mina Staedler 0.5 2B para portaminas	\$ 4.25	6	\$ 25.5
Total Pedido:			\$ 64.91

Joomla! es software libre liberado bajo la Licencia GNU/GPL.

Figura IV. 164: Detalles de un pedido escogido, back-end

Front-end

Descripción: El componente dará la opción al cliente de gestionar los artículos del carrito de compras, dando clic en el botón “Agregar al Carrito” que se encuentra en la vista de detalles de un artículo escogido. Así como dará la posibilidad de realizar el pedido de los artículos que se encuentran en el carrito de compras, emitiendo un recibo con el detalle del pedido. Este componente se implementará en Joomla 1.5.

Nombre: com_libreria_cart

Ubicación: ../components/com_libreria_cart

Estructura de los archivos: La estructura para este componente consta de dos archivos, como se muestra en la Figura IV.165.

libreria_cart.php: con la lógica del componente.

libreria_cart.html.php: con la visualización respectiva de los datos.

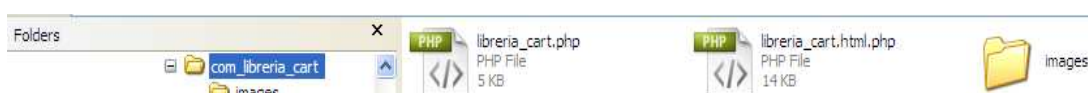


Figura IV. 165: Estructura del componente com_libreria_cart, front-end

Pantallas de visualización:

Botón de agregar a carrito de compras, en la vista individual de un artículo escogido. Ver Figura IV.166.



Figura IV. 166: Botón para agregar al carrito de compras, front-end

Visualización del Carrito de compras con los artículos agregados. Ver Figura IV.166.

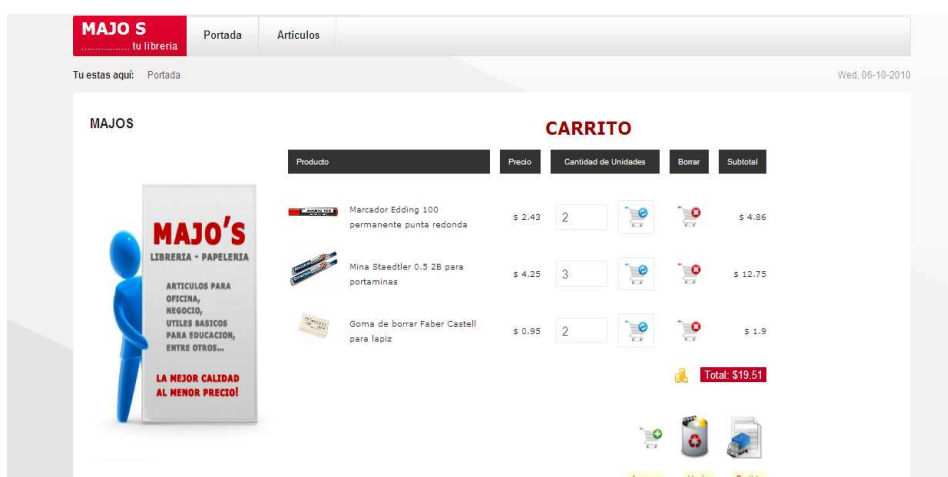


Figura IV. 167: Visualización del Carrito de Compras, front-end

Formulario de ingreso de datos del cliente para realizar el pedido de los artículos del carrito de compras. Ver Figura IV.168.

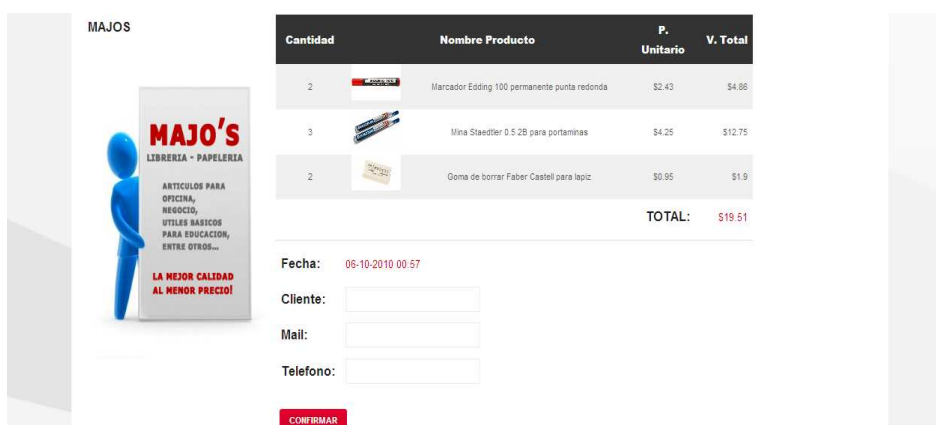


Figura IV. 168: Formulario datos cliente para el pedido, front-end

Comprobante imprimible de la orden del pedido de los artículos que se encuentran en el carrito de compras. Ver Figura IV.169.

COMPROBANTE



MAJO'S
LIBRERIA - PAPELERIA

ARTICULOS PARA OFICINA, NEGOCIO, UTILES BASICOS PARA EDUCACION, ENTRE OTROS...

LA MEJOR CALIDAD AL MENOR PRECIO!

LIBRERIA MAJO'S

... lo mejor en articulos para oficina, negocio, estudio LA MEJOR CALIDAD AL MEJOR PRECIO!

Telefono: 032 956411 Cel: 098 239551

Direccion: Guayaquil y 5 de Junio

CLIENTE: Chris Ferguson

e-mail: chrisferguson@hotmail.com

GRACIAS POR PREFERIRNOS



MAJO'S LIBRERIA

e-mail: majoslibreria@majos.com

Pedido #: 25

Telefono: 098123551

Cantidad	Nombre Producto	P. Unitario	V. Total
2	 Marcador Edding 100 permanente punta redonda	\$2.43	\$4.86
3	 Mina Staedter 0.5 2B para portaminas	\$4.25	\$12.75
2	 Goma de borrar Faber Castell para lapiz	\$0.95	\$1.9
TOTAL:			\$19.51

Figura IV. 169: Comprobante de la orden de pedido, front-end

4.2.2.5. Pruebas

Una vez que se ha generado el código, comienzan las pruebas del programa. Este proceso se centra en los procesos lógicos internos del software, asegurando que todas las sentencias se han comprobado, y en los procesos externos funcionales, es decir realizar las pruebas para la detección de errores y asegurar que la entrada produzca resultados reales de acuerdo con los resultados requeridos.

En cuanto al uso del portal por parte de los clientes, encontramos los siguientes resultados. En todas las pruebas se las realizó con una muestra de 15 usuarios en el front-end, y para el back-end con una muestra de dos usuarios administrativos.

FRONT-END

- **Tiempo de búsqueda de un producto en específico.** Ver Tabla IV.XIX

Tabla IV. XIX: Tiempo de búsqueda de un artículo

Usuario	Tiempo [segundos]
Usuario 1	27
Usuario 2	24
Usuario 3	33
Usuario 4	41
Usuario 5	25
Usuario 6	31
Usuario 7	46
Usuario 8	34
Usuario 9	38
Usuario 10	49
Usuario 11	27
Usuario 12	38
Usuario 13	25
Usuario 14	44
Usuario 15	37

Total de segundos de uso: 495 segundos

Promedio de tiempo en búsqueda: (total segundos de uso)/(muestra)
= 495/15 = 33 segundos.

Intervalos de Tiempo

De 21 a 30 segundos: 5 usuarios.

De 31 a 40 segundos: 6 usuarios.

Mayor a 40 segundos: 4 usuarios. (Ver Figura IV.170).

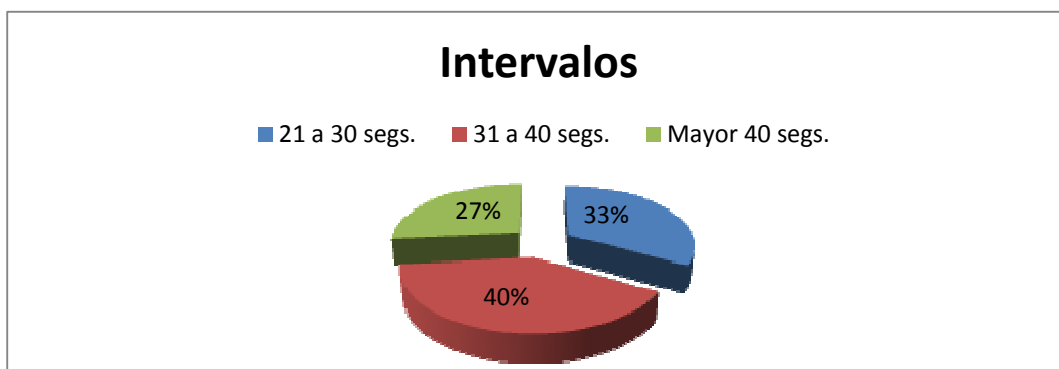


Figura IV. 170: Porcentajes resultantes de los Intervalos de tiempo

Interpretación

Según los resultados del tiempo de búsqueda de un artículo, que nos da la muestra, concluimos que el usuario cliente, que usa el portal se demora en promedio 33 segundos en la búsqueda.

- **Tiempo de demora para realizar el pedido de los artículos del carrito de compras.** Ver Tabla IV.XX

Tabla IV. XX: Tiempo de demora para el pedido de artículos

Usuario	Tiempo [segundos]
Usuario 1	25
Usuario 2	22
Usuario 3	31
Usuario 4	31
Usuario 5	22
Usuario 6	27
Usuario 7	33
Usuario 8	36
Usuario 9	50
Usuario 10	38
Usuario 11	44
Usuario 12	46
Usuario 13	28
Usuario 14	27
Usuario 15	26

Total de segundos de uso: 486 segundos

Promedio de tiempo en demora: (total segundos de uso)/(muestra)

$$= 486/15$$

$$= 32.4 \text{ segundos.}$$

Intervalos de Tiempo

De 21 a 30 segundos: 7 usuarios.

De 31 a 40 segundos: 5 usuarios.

Mayor a 40 segundos: 3 usuarios. (Ver Figura IV.171).

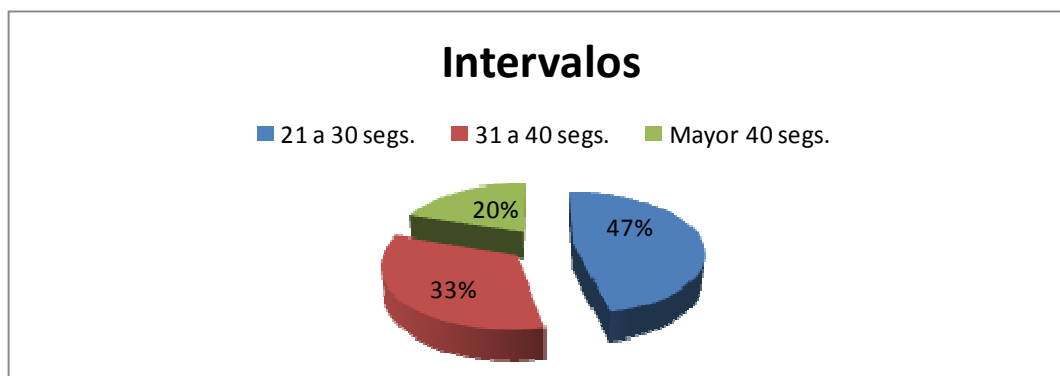


Figura IV. 171: Porcentajes resultantes de los Intervalos de tiempo

Interpretación

Según los resultados del tiempo de demora en realizar el pedido de los artículos incluidos en el carrito de compras, que nos proporciona la muestra, concluimos que el usuario cliente, que usa el portal se demora en promedio 32.4 segundos en la búsqueda.

BACK-END

- **Tiempo de gestión de artículos (ingreso de artículos nuevos).** Ver Tabla IV.XXI

Tabla IV. XXI: Tiempo de gestión de Artículos

Usuario Administrador	Ingreso de Artículo nuevo [segundos]	Modificación de un Artículo existente [segundos]	Eliminación de un Artículo [segundos]
Administrador 1	140	45	11
Administrador 2	163	32	15
Totales	303	77	26

Total de segundos para el ingreso: $t_i = 303$ segundos

Total de segundos para la modificación: $t_m = 77$ segundos

Total de segundos para la eliminación: $t_e = 26$ segundos

Promedio de tiempo en gestión: $(t_i + t_m + t_e)/(muestra)$

$$= 406/2 = 203 \text{ segundos.}$$

Porcentajes de tiempo por tarea

Ingreso de Artículo nuevo: $t_i/muestra = 151.5$ segundos

Modificación Artículo existente: $t_m/muestra = 38.5$ segundos

Eliminación de un Artículo: $t_e/2 = 13$ segundos. (Ver Figura IV.172).

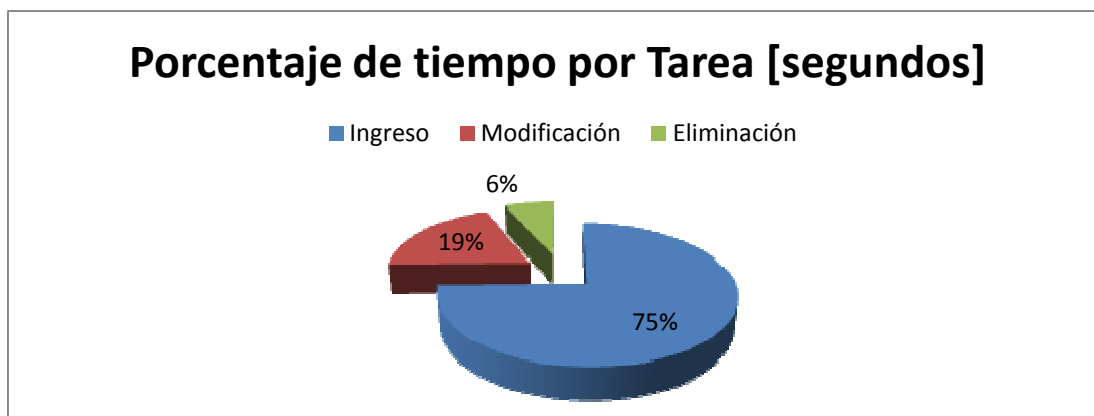


Figura IV. 172: Porcentajes resultantes de los tiempos que se usan en las tareas de gestión de los artículos

Interpretación

Según los resultados de los tiempos que se emplean en las tareas de gestión de los artículos de librería: ingreso, modificación y eliminación de los mismos, que nos suministra la muestra, concluimos que el usuario administrador, que usa el portal se demora en promedio 203 segundos en realizar dichas tareas.

- **Tiempo de demora para comprobar datos de las órdenes de pedido de artículos ofertados en el portal, realizadas por los clientes.** Ver Tabla IV.XXII

Tabla IV. XXII: Tiempo de demora para la comprobación de la orden

Usuario Administrador	Tiempo [segundos]
Administrador 1	19
Administrador 2	22

Total de segundos de uso: 41 segundos

Promedio de tiempo en demora: (total segundos de uso)/(muestra)

$$= 41/2$$

$$= 20.5 \text{ segundos.}$$

Media del tiempo de demora en la comprobación de una orden de pedido

Ver Figura IV.173

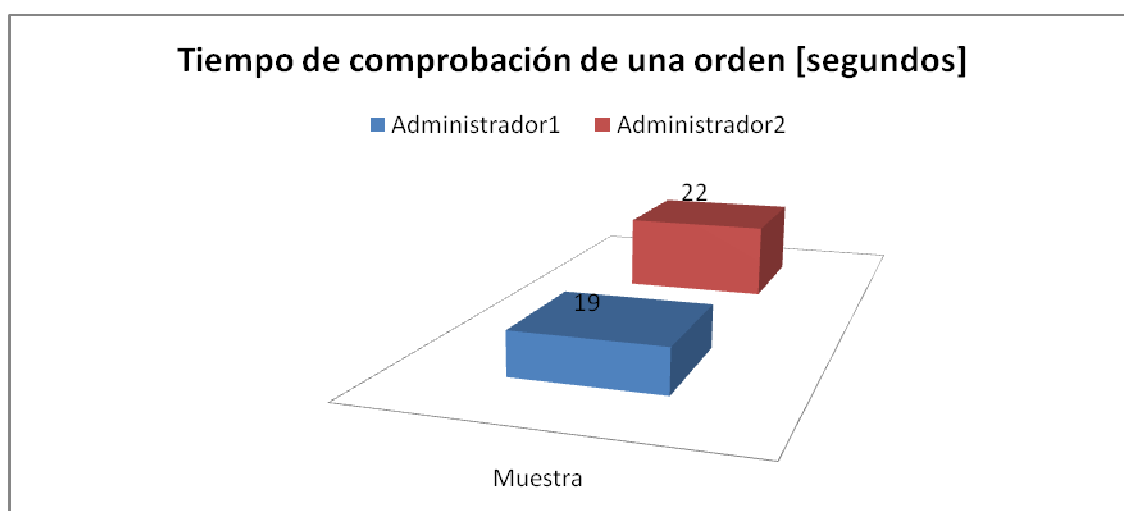


Figura IV. 173: Tiempos resultantes del tiempo de demora

Interpretación

Según los resultados del tiempo de demora en comprobar los datos de una orden de pedido de artículos, que nos provee la muestra, concluimos que el usuario administrador, que usa el portal se demora en promedio 20,5 segundos en la comprobación.

4.2.2.6. Implantación del Portal Web

Ver anexo 1.

CONCLUSIONES

- Dentro del manejo de cualquier CMS es necesario de una base de datos y un servidor web ya sea Linux o Windows la plataforma utilizada.
- Joomla!, Typo3 y Drupal, poseen comunidades en crecimiento las que se preocupan por publicar cada vez más información, dar soporte y brindar seguridades para cada inconveniente que se vaya presentando.
- Los parámetros para la medición del mejor CMS para el desarrollo de componentes son: Requerimientos del Sistema, Facilidad de Uso, Gestión de Extensiones dentro del CMS y la Estructura de las Extensiones.
- La utilización del esquema MVC dentro del desarrollo web brinda un gran número de funcionalidades en especial para la reutilización de código, así como una mejor distribución, diseño y modularidad.
- Debido al análisis y de acuerdo al puntaje alcanzado por cada una de las variables se ha obtenido como resultado que el CMS Joomla! en el desarrollo de componentes tiene el puntaje más alto con un porcentaje del 97,82% que es equivalente a Excelente.
- En el desarrollo de componentes en Sistemas de Administración de Contenido en un 27,71% mejor es Joomla! en comparación con Drupal y Typo3 para la implementación del portal web de la librería Majo´s

RECOMENDACIONES

- Para el manejo local de Joomla!, Drupal y Typo3 se recomienda el uso del paquete XAMPP en la versión 1.7 o superior, que agrupa MySQL como base de datos, el servidor Apache entre otras prestaciones, php como lenguaje, por presentar una mejor administración.
- Para mayor facilidad al momento de utilizar cualquier CMS se recomienda visitar el sitio oficial del mismo, www.nombredelcms.org, foros y blogs, además existe la opción de suscripción, la cual permitirá conocer las nuevas publicaciones por medio de correo electrónico.
- Para establecer parámetros de medición se recomienda delimitar los elementos a comparar mediante la profunda recopilación de información.
- Para el desarrollo de un sitio web se recomienda el uso de CMS (sistema de administración de contenidos), que permitan utilizar el esquema MVC porque mejora notablemente la organización de los archivos de los componentes y la localización específica de una parte de los mismos.
- Se recomienda el uso de Joomla! como el mejor sistema de administración de contenidos en el desarrollo de componentes en sitios web por su soporte, seguridad, facilidad de uso, gestión, estructura y además porque se rige al esquema MVC.

RESUMEN

Se realizó el estudio comparativo del desarrollo de componentes en los CMS Joomla!, Typo3 y Drupal como sistemas de administración de contenido para encontrar el mejor para la implementación del portal web de la Librería Majo's.

Se estableció cuatro parámetros de medición: Requerimientos para el desarrollo, Facilidad de Uso, Gestión de Extensiones, Estructura de las Extensiones, cada uno de los parámetros contiene tres variables, para los resultados cuantitativos y cualitativos, la calificación de los parámetros de comparación se basó en las siguientes escalas: regular cuando el valor sea menor al 70%, Bueno cuando el valor está entre el 70% y 80%, Muy Bueno cuando el valor está entre el 80% y 90% y Excelente cuando sea mayor al 90%, se desarrolló cuatro módulos de prueba para cada uno de los CMS's, en el primer módulo se realizó la creación de un formulario para el ingreso de artículos, el segundo módulo el ingreso y listado de los artículos, el tercer módulo reutilizó los dos primeros módulos y funciones adicionales como agregar categoría, agregar subcategoría, agregar proveedor, agregar artículo, listar artículo, registrar ventas y listar ventas y el cuarto módulo la administración de un componente existente además del estudio de toda la información consultada.

Joomla!, Typo3 y Drupal son sistemas multiplataforma basados en PHP que utilizan MySQL como base de datos y el servidor Apache en Linux y en Windows. Los 3 CMS's trabajan con templates o plantillas las que se pueden crear desde cero o descargar desde la web, facilitando la construcción del sitio, además presentan interfaces amigables pero Joomla! posee una interfaz más completa para el desarrollo de componentes y Drupal la más amigable para la administración, un punto muy importante es que Joomla! utiliza el esquema MVC ideal para la reutilización y adaptación de componentes en el desarrollo de portales web, al término del análisis comparativo y la interpretación de los valores cualitativos y cuantitativos, se obtuvo los siguientes puntajes Joomla! con un 97,82 %, Drupal con un 71,74% y Typo3 67,39%, siendo la mayor la del CMS Joomla! con una equivalencia de Excelente.

Se concluye que en el desarrollo de componentes en Sistemas de Administración de Contenido con un 26.66% mejor es Joomla! sobre sus competidores ya que tiene mejores prestaciones para la gestión de componentes y es el único de los tres que usa el esquema MVC por lo que se recomienda el uso de Joomla! para el desarrollo de mejores aplicaciones web.

SUMMARY

A comparative study was carried out about components development in CMS Joomla!, Typo3 and Drupal as a content management system in order to find the best implementation in the web portal from Librería Majo´s.

Four measuring parameters were established: development requirements, usability, extensions management, extension framework, each parameters contains three variables, for qualitative and quantitative results, the Comparison parameter score is based in the following scales: regular when the value is mayor to 70%, Good when the value is between 80% and 90% and Excellent when is mayor to 90%, four test modules were developed for each CMS´s, the first module was carried out a form creation for articles input, the second module the articles input and list, the third reuse the first two modules and additional applications as add a category, add subcategory, add provider, add article, list an article, sales register, sales list and the fourth module the component management already existing as well as the study of the entire information sources consulted.

Joomla!, Typo3 and Drupal are multi-platform systems based in PHP using MySql as a database and Linux and Windows in the Apache server. The 3 CMS´s work with templates and could be created from zero or could be downloaded from the web, providing the web site construction, besides user-friendly interface are presented but Joomla!, has an interface more complete for the components development and Drupal the most user-friendly interface for management, an important issue is Joomla! use a MVC diagram ideal for reuse and adaptation of components in the web portal development, at the end of the comparative analysis and interpretation of qualitative and quantitative data, the following scores were obtained Joomla! With 97,82%, Drupal with 71,74% and Typo3 with 67,39%, being CMS Joomla! the best score with an Excellent scale degree.

Is concluded the development components in Content Management System with 26.66% the best is Joomla! Because of benefits regarding the components management and it is the unique of the three components using an MVC diagram recommending the use of Joomla! to develop better web applications .

GLOSARIO

Back end: contiene la parte de administración del sitio web, localizado en una diferente URL al sitio web; y es donde se realizan las tareas de configuración, mantenimiento, limpieza, creación de estadísticas, etc.

Host.- El término Host es usado en informática para referirse a los computadores conectados a la red, que proveen o utilizan servicios a/de ella. Los usuarios deben utilizar hosts para tener acceso a la red. En general, los hosts son computadores mono o multiusuario que ofrecen servicios de transferencia de archivos, conexión remota, servidores de base de datos, servidores WWW, etc. Los usuarios que hacen uso de los hosts pueden a su vez pedir los mismos servicios a otras máquinas conectadas a la red.

Hyperlink.- hiperenlace (también llamado enlace, vínculo, o hipervínculo) es un elemento de un documento electrónico que hace referencia a otro recurso, por ejemplo, otro documento o un punto específico del mismo o de otro documento.

Framework: un framework es una estructura conceptual y tecnológica de soporte definida, normalmente con artefactos o módulos de software concretos, con base en la cual otro proyecto de software puede ser organizado y desarrollado. Típicamente, puede incluir soporte de programas, bibliotecas y un lenguaje interpretado entre otros programas para ayudar a desarrollar y unir los diferentes componentes de un proyecto.

Front end: es el sitio web (website), que los visitantes y los usuarios registrados pueden ver.

Metadatos: El término «metadatos» no tiene una definición única. Según la definición más difundida de metadatos es que son «datos sobre datos». También hay muchas declaraciones como «informaciones sobre datos», «datos sobre informaciones» e «informaciones sobre informaciones».

Webmaster: Un webmaster (contracción de las palabras inglesas web y master) o en femenino webmistress es la persona responsable de mantenimiento o programación de un sitio web.

ANEXOS

ANEXO 1

CERTIFICADO

La Librería Majo's certifica que el portal web para la gestión y difusión de los artículos en stock y la reservación de los mismos mediante el uso del carrito de compras integrado a la página desarrollado por los señores: CARLOS ARTURO JARA SANTILLÁN y VÍCTOR MANUEL OQUENDO CORONADO, se encuentra ya en producción.

Se emite el presente documento, a solicitud de los interesados, para los fines que estimen convenientes.

Riobamba, 07 de Octubre del 2010

Lic. Miriam Moreno

GERENTE-PROPIETARIA LIBRERÍA MAJO'S

ANEXO 2

ANEXO 3

ANEXO 4

ANEXO 5

BIBLIOGRAFÍA

BUTCHER, Matt. Learning Drupal 6 Module Development. Birmingham – Reino Unido, Packt Publishing. 2009. Total 150p.

CMS

<http://www.aplicacionesempresariales.com/eligiendo-el-mejor-cms-open-source-para-tu-empresa.html>; (17-08-2008)

(2009-05-10)

<http://www.planetacms.com/interes-general/actualidad/el-mejor-cms-del-2008>

(2009-05-10)

<http://www.scribd.com/doc/9784583/Que-es-un-CMS>

(2009-09-11)

<http://www.martinalia.com/10-caracteristicas-que-todo-CMS>

(2009-09-11)

http://es.wikipedia.org/wiki/Anexo:Sistemas_de_gesti%C3%B3n_de_contenidos

(2009-09-11)

<http://www.multidimensionstudio.com/multi/es/disenio-web/cms>

(2009-09-11)

<http://www.scribd.com/doc/9784583/Que-es-un-CMS>

(2009-09-12)

Drupal

<http://drupal.com>

(2009-07-03)

<http://drupal.org.es> (11-04-2005)

(2009-07-03)

<http://es.wikipedia.org/wiki/Drupal>

(2009-10-11)

<http://www.drupal-centroamerica.org/tipo-de-pagina/logistica>

(2009-10-11)

<http://drupal.org.es/drupal>

(2009-10-11)

<http://www.linuxvenezuela.com/?q=drupal>

(2009-10-11)

<http://kimberly-programacion.blogspot.com/>

(2009-10-11)

<http://drupal.org.es/caracteristicas>

(2009-10-11)

<http://www.slideshare.net/develCuy/el-incompredido-drupal>

(2009-10-11)

<http://www.inmensia.com/articulos/drupal/modulos.html>

(2009-10-11)

GRAF, Hagen. Building Websites with Joomla! A step by step tutorial to getting your Joomla! CMS website up fast. Birmingham – Reino Unido, Packt Publishing. 2008. Total 150p.

Joomla!

<http://www.joomlaspanish.org>; (12-06-2005)

(2009-07-03)

<http://www.joomlaos.net>

(2009-07-03)

<http://www.astrolabio.com.co/disenio-web/2-disenio-web/13-joomla-razones-para-usarlo-en-su-sitio-web.html>

(2009-05-10)

<http://www.joomla.org>

(2009-07-03)

http://es.wikipedia.org/wiki/Joomla!#Historia_del_proyecto

(2009-10-2)

<http://www.nosolocodigo.com/tutorial-componentes-para-joomla-15-ii>

(2009-10-2)

KOCH, Daniel. Mastering Typoscript Typo3 Websites, Template and Extension development. Birmingham – Reino Unido, Packt Publishing. 2009. Total 150p.

LEBLANC, Joseph. Learning Joomla! 1.5 Extension Development Creating Modules, Components, and Plug-Ins with PHP. Birmingham – Reino Unido, Packt Publishing. 2008. Total 150p.

Patrón Modelo – Vista - Controlador

<http://www.proactiva-calidad.com/java/patrones/mvc.html>

(2009-09-12)

PEACOCK, Michael. Building Websites with TYPO3. Birmingham – Reino Unido, Packt Publishing. 2008. Total 150p.

PHP

http://es.wikipedia.org/wiki/PHP#Visi.C3.B3n_general

(2009-09-11)

<http://www.sip.gob.mx/lenguajes-aplicaciones-y-herramientas/397-10-razones-para-el-desarrollo-de-aplicaciones-web-con-php>

(2009-09-11)

RAHMEL, Dan. Professional Joomla!. Indiana – Estados Unidos, Wiley Publishing. 2008. Total 150p.

Typo3

<http://www.typo3performance.com/typo3-cms>

(2009-05-10)

<http://typo3.com> (15-02-2006)

(2009-07-03)

<http://typo3.org> (10-03-2005)

(2009-07-03)

<http://es.wikipedia.org/wiki/TYPO3>

(2009-10-11)

<http://www.proyectoparque.net/index.php?id=87&L=2>

(2009-10-11)

<http://typo3.org/teams>

(2009-10-11)

http://typo3.org/extensions/repository/view/doc_extbase/current/

(2009-10-11)

VANDYK, John K. WESTGATE, Matt. Pro Drupal Development. 2.ed., New York – Estados Unidos, Apress. 2008. Total 150p.

World Wide Web

<http://www.masadelante.com/faqs/www>

(2009-09-11)

http://es.wikipedia.org/wiki/World_Wide_Web#Tecnolog.C3.ADas_web

(2009-09-11)