

**UNIVERSIDAD DE ALCALÁ**

**Escuela Técnica Superior de Ingeniería Informática**

**INGENIERÍA INFORMÁTICA**



**Trabajo Fin de Carrera**

**Implantación de un Museo Virtual en el Real  
Observatorio de Madrid**

Héctor Navarro Martín  
2010



**UNIVERSIDAD DE ALCALÁ**

**Escuela Técnica Superior de Ingeniería Informática**

**INGENIERÍA INFORMÁTICA**

**Trabajo Fin de Carrera**

**Implantación de un Museo Virtual en el Real Observatorio  
de Madrid**

Alumno: Héctor Navarro Martín

Director: José Javier Martínez Herráiz

**Tribunal:**

**Presidente:**

**Vocal 1º:**

**Vocal 2º:**

Calificación: .....

Fecha: .....



# **Agradecimientos**

*A HBO, AMC y ABC, por hacerme más ameno el trabajo.*



# Índice general

<b>1. Resumen</b>	<b>15</b>
1.1. Resumen . . . . .	17
<b>2. Memoria</b>	<b>19</b>
2.1. Introducción . . . . .	21
2.2. Base teórica . . . . .	21
2.2.1. Estado del Arte . . . . .	21
2.2.1.1. ¿Qué puede ser un museo virtual? . . . . .	21
2.2.1.2. Museos . . . . .	22
2.2.1.2.1. Tate Modern (Londres, Reino Unido) . . . . .	22
2.2.1.2.2. MoMA (New York, EEUU) . . . . .	24
2.2.1.2.3. Actual museo del OAN . . . . .	24
2.2.1.3. Enciclopedias . . . . .	25
2.2.1.3.1. Wikipedia . . . . .	25
2.2.1.3.2. Indianopedia, Frikipedia, *pedia . . . . .	26
2.2.1.4. Sistemas gestores de contenidos vs Wikis . . . . .	26
2.3. Descripción del diseño . . . . .	27
2.3.1. Bareación de alternativas . . . . .	27
2.3.1.1. Comentarios . . . . .	27
2.3.2. Comparativa . . . . .	28
2.3.2.1. Premisas iniciales . . . . .	28
2.3.2.2. Candidatos pre-seleccionados . . . . .	28
2.3.2.3. Comparativa con CMSMatrix . . . . .	29
2.3.2.3.1. System Requirements (Requisitos de sistema) . .	29
2.3.2.3.2. Security (Seguridad) . . . . .	30
2.3.2.3.3. Support (Asistencia) . . . . .	30
2.3.2.3.4. Ease of Use (Facilidad de uso) . . . . .	30
2.3.2.3.5. Performance (Rendimiento) . . . . .	32
2.3.2.3.6. Management (Administración) . . . . .	33
2.3.2.3.7. Interoperability (Interoperatividad) . . . . .	33
2.3.2.3.8. Flexibility (Flexibilidad) . . . . .	34
2.3.2.3.9. Built-in Applications (Aplicaciones embebidas) .	34
2.3.2.3.10Commerce (Comercio electrónico) . . . . .	35
2.3.2.4. Propuesta preliminar . . . . .	35
2.3.2.4.1. Liferay Portal (Community Edition) . . . . .	35
2.3.2.4.2. Drupal . . . . .	37
2.3.2.4.3. Moodle . . . . .	38
2.3.2.4.4. Plone . . . . .	38
2.3.3. Anexo a comparativa . . . . .	38

2.4. Drupal . . . . .	38
2.4.1. Entorno de (pre)desarrollo . . . . .	39
2.4.2. <i>Understanding Drupal</i> (Entender Drupal) . . . . .	39
2.4.2.1. General . . . . .	40
2.4.2.2. Módulos . . . . .	40
2.4.2.3. Usuarios . . . . .	41
2.4.2.4. Nodos . . . . .	41
2.4.2.5. Formas de organizar el contenido . . . . .	42
2.4.2.5.1. Bloques . . . . .	42
2.4.2.5.2. Menús . . . . .	42
2.4.2.5.3. Taxonomías . . . . .	42
2.4.2.5.4. CCK (Content Construction Kit) . . . . .	43
2.4.2.5.5. Views . . . . .	43
2.4.2.6. Temas . . . . .	43
2.4.2.6.1. Archivos .info . . . . .	44
2.4.2.6.2. Regiones . . . . .	44
2.4.2.6.3. Características . . . . .	45
2.4.2.6.4. CSS y JavaScript . . . . .	45
2.4.2.6.5. Archivos de plantilla . . . . .	45
2.4.2.6.6. template.php . . . . .	45
2.5. Conclusiones y trabajo futuro . . . . .	46
2.5.1. Logros alcanzados . . . . .	46
2.5.2. Conocimientos adquiridos . . . . .	46
2.5.2.1. Generalidades . . . . .	46
2.5.2.2. Particularidades . . . . .	47
2.5.2.3. Modificaciones . . . . .	48
2.5.2.3.1. Web Semántica . . . . .	48
2.5.2.3.2. Uso de estándares . . . . .	49
2.5.2.3.3. Wikis . . . . .	49
2.5.3. Tiempo empleado . . . . .	49
2.5.4. Líneas futuras de trabajo . . . . .	49
<b>3. Planos</b> . . . . .	<b>51</b>
3.1. Esquemáticos . . . . .	53
3.1.1. Modelo de datos . . . . .	53
3.1.1.1. Catálogo . . . . .	53
3.1.1.1.1. OBJETO . . . . .	53
3.1.1.1.2. ÁREA . . . . .	54
3.1.1.1.3. TIPO_OBJETO . . . . .	54
3.1.1.1.4. ARTÍCULOS . . . . .	54
3.1.1.1.5. AREA_ARTÍCULO . . . . .	54
3.1.1.1.6. INVENTOR . . . . .	55
3.1.1.2. Revisión del catálogo . . . . .	55
3.1.1.2.1. Ficha de objeto . . . . .	55
3.1.2. Esquema general de servicio . . . . .	56
3.1.2.1. Servidor del museo . . . . .	56
3.1.2.2. Clientes a través de Internet . . . . .	56
3.1.2.3. Puesto en sala . . . . .	56
3.1.2.4. Terminales móviles . . . . .	57
3.1.3. Esquema general de creación de contenidos . . . . .	59

3.1.3.1. Servidor . . . . .	60
3.1.3.2. Administrador . . . . .	60
3.1.3.3. Editor . . . . .	60
3.1.4. Redactor . . . . .	60
<b>4. Pliego de condiciones</b>	<b>61</b>
4.1. Consideraciones previas . . . . .	63
4.1.1. Rendimiento necesario . . . . .	63
4.1.1.1. Afluencia . . . . .	63
4.1.1.2. Caudal . . . . .	63
4.1.1.3. Criticidad . . . . .	64
4.1.2. Particularidades del software elegido . . . . .	64
4.1.2.1. Rendimiento de la pila LAMP . . . . .	64
4.1.2.2. Cuellos de botella . . . . .	65
4.1.2.3. Invitados indeseables . . . . .	65
4.1.2.4. Rendimiento de Apache . . . . .	65
4.1.2.5. Ajustes de PHP . . . . .	65
4.1.2.6. Ajustes de rendimiento de MySQL . . . . .	65
4.1.2.7. Ancho de banda . . . . .	66
4.1.2.8. Consumo de recursos de Drupal . . . . .	66
4.1.2.9. Configuración de Drupal hacia el rendimiento . . . . .	66
4.1.3. Umbral de costes . . . . .	66
4.2. Requisitos de Hardware . . . . .	67
4.2.1. Desarrollo . . . . .	67
4.2.2. Servidor . . . . .	67
4.2.3. Infraestructura de red . . . . .	68
4.2.4. Cliente en sala . . . . .	68
4.2.5. Dispositivo móvil . . . . .	68
4.3. Requisitos de Software . . . . .	69
4.3.1. Desarrollo . . . . .	69
4.3.2. Servidor . . . . .	69
4.3.3. Cliente en sala . . . . .	70
4.3.4. Dispositivo móvil . . . . .	70
<b>5. Manual de Usuario</b>	<b>71</b>
5.1. Manual . . . . .	73
<b>6. Apéndices</b>	<b>75</b>
6.1. Apéndice I: Alfresco . . . . .	77
6.1.1. Análisis . . . . .	77
6.1.1.1. Sabores . . . . .	77
6.1.1.2. Edición Web . . . . .	78
6.1.1.2.1. Sitios colaborativos . . . . .	78
<b>7. Bibliografía</b>	<b>79</b>



# Índice de figuras

2.1. Imagen de dispositivo de mano del Tate Modern [1] . . . . .	23
2.2. Captura de pantalla del sitio del MoMA . . . . .	24
2.3. Captura de pantalla del museo virtual del OAN . . . . .	25
2.4. "System Requirements" desde CMSMatrix.org . . . . .	30
2.5. "Security" desde CMSMatrix.org . . . . .	31
2.6. "Support" desde CMSMatrix.org . . . . .	31
2.7. "Ease of Use" desde CMSMatrix.org . . . . .	32
2.8. "Performance" desde CMSMatrix.org . . . . .	32
2.9. "Management" desde CMSMatrix.org . . . . .	33
2.10."Interoperability" desde CMSMatrix.org" . . . . .	34
2.11."Flexibility" desde CMSMatrix.org . . . . .	35
2.12."Built-in Applications" (I) desde CMSMatrix.org . . . . .	36
2.13."Built-in Applications" (II) desde CMSMatrix.org . . . . .	37
2.14."Commerce" desde CMSMatrix.org . . . . .	37
2.15Estructura básica de Temas en Drupal [2] . . . . .	44
3.1. Vista preliminar de una página del catálogo en papel . . . . .	57
3.2. Esquema general de servicio . . . . .	58
3.3. Esquema general de creación de contenidos . . . . .	59



# **Índice de cuadros**

2.1. Cuadro resumen de la ponderación de características . . . . .	27
4.1. Comandos útiles para buscar cuellos de botella . . . . .	65
4.2. Características hardware de la plataforma de desarrollo . . . . .	67
4.3. Requisitos hardware del servidor . . . . .	67
4.4. Requisitos hardware del cliente en sala . . . . .	68
4.5. Requisitos hardware del dispositivo móvil . . . . .	69



# **Capítulo 1**

## **Resumen**



## 1.1. Resumen

El objetivo principal de este proyecto es la implantación de una solución software que desempeñe funciones de museo virtual para el Real Observatorio de Madrid. Además, comprenderá una plataforma de gestión de contenidos operable por personal no informático. Para ello, la solución propuesta es la búsqueda comparada, elección e integración de una plataforma de gestión de contenidos (CMS), añadiendo, si se da el caso, las funcionalidades no incluidas en el paquete escogido pero sí en los requisitos recogidos de la empresa. Se hace especial hincapié en la reducción del mantenimiento posterior, la accesibilidad móvil, la facilidad de uso para personal no técnico y la posibilidad de modificar y/o añadir funcionalidades en un futuro.

Palabras clave: cms, museo virtual, móvil, wiki



## **Capítulo 2**

# **Memoria**



## 2.1. Introducción

El trabajo expuesto en este documento tiene como fin la implantación de un museo virtual en el Observatorio de Madrid. La parte más importante en este tipo de proyectos (y, por extensión, de todos) es la recogida de requisitos y el análisis (en este caso, elección principalmente). Para poder entender mejor lo que se solicita por parte del cliente es necesario conocer el concepto de “museo virtual” al que tratamos de acercarnos. Se trataría de una plataforma accesible físicamente desde puestos instalados en una sala de exposiciones dentro del propio observatorio, accesible en línea desde Internet y que comprenda una base de datos posiblemente reutilizable en el futuro. Con estas pinceladas he deducido que la solución existente (no hay que olvidar que esto es ingeniería) más cercana a lo que se nos pide pasaría, en mi opinión, por un gestor de contenidos o CMS (Content Management System). Dada la amplísima oferta que hay de estos productos en el mercado, creo conveniente hacer una selección a partir de criterios, unos surgidos y otros sugeridos, analizados mano a mano con el cliente. A partir de ahí, se escogerá el CMS candidato con mejores cualidades en cuanto a facilidad de uso, bajo mantenimiento y características incluidas por defecto. Las características no comprendidas en esta selección (pero necesarias igualmente) serán incorporadas junto con las diferentes modificaciones previstas para darle forma final a la plataforma. Posteriormente, se integrará y se pondrá en servicio con los plazos que el cliente disponga para ello.

## 2.2. Base teórica

Al tratarse de un proyecto tan práctico, la base teórica del mismo no tiene un gran misterio. Se trata de realizar el clásico enfoque de presentar distintas aplicaciones al cliente cumpliendo unos requerimientos mínimos y adaptando e integrando la solución escogida para que encaje con el sistema solicitado.

En esta ocasión, presentaremos varios subapartados según las distintas fases del proyecto.

### 2.2.1. Estado del Arte

Para elaborar este proyecto pensé que lo más sensato era, una vez acotado el problema, tratar de echar un vistazo a cómo estaban las cosas en la actualidad en otros museos. Considero importante saber en qué estado se encuentran tanto el círculo de los CMS, wikis, museos y catálogos informatizados. En este periplo, he topado con varias fuentes de información desconocidas para mí, algunas ideas sacadas de museos visitados en persona y una exhaustiva (aunque difícil de discernir) cantidad de datos sobre gestores de contenido.

#### 2.2.1.1. ¿Qué puede ser un museo virtual?

Al hablar de museos virtuales, a uno se le pueden venir a la mente distintos conceptos sobre lo que representan esas palabras:

- Visita virtual al museo (modo “Second Life”).

- Galería de imágenes.
- Sitio web del museo (meramente informativo).
- Catálogo de objetos.
- Ampliación de lo expuesto en el museo.

En el primer caso, nos referimos a esa idea de recrear a través de gráficos en 3D la experiencia de una visita real al museo. Esta solución es muy poco usada debido a lo costoso de la misma. Además de modelar infográficamente la arquitectura del museo y construir una aplicación que utilice un potente motor gráfico para sostenerla, hay que tener en cuenta el gran tráfico generado para que esto funcione adecuadamente. En mi opinión, dado que difícilmente se puede reemplazar la experiencia de visitar el museo en persona, creo que no compensa tanto despliegue de medios ya que no contribuye (al menos en el caso expuesto) sensiblemente al objetivo marcado.

El segundo caso, una mera galería de imágenes, aunque pobre, podría vale como museo virtual para un museo de arte en según qué circunstancias. Obviamente, y aunque la presentación de la información en este formato sea algo a tener en cuenta, si limitáramos nuestro museo virtual a este tipo de sitio, estaríamos, creo yo, limitando muchísimo la información ofrecida, cosa que no cuadra con el objetivo de complementar el museo.

El tercer caso lo he incluido como posibilidad de que alguien lo identificase al hablar de "museos virtuales", dado que crear una página con información corporativa, de horarios y accesos es sólo una parte de lo que pretendería este proyecto.

El cuarto caso será aquel en el que básicamente tenemos una somera base de datos con una sencilla interfaz de búsqueda y acceso a la colección, sin tratar de explicar o poner en contexto los objetos que tenemos delante.

El quinto caso es el que nos ocupa, siendo el cometido del mismo complementar la exposición física, permitiendo a los visitantes (y no visitantes) conseguir un trasfondo y una explicación más completa de los objetos y su contexto para contribuir a la divulgación de un conocimiento relacionado con lo expuesto. Obviamente incluirá parte de galería de imágenes en cuanto a interfaz, información de típico sitio web como caso anecdótico de contenido, componentes que mejoren la interactividad sin sacrificar el desempeño y el coste y datos concretos sobre los objetos como manera de acceder a la base de datos.

### **2.2.1.2. Museos**

Aprendiendo de museos conocidos, podemos poner ejemplos prácticos de los que sacar ideas o matices que el cliente pueda identificar como deseables. Está claro que sería imposible abarcar todos o siquiera la mayoría de los museos del mundo, ni siquiera los más importantes (¿quién decide esa lista?), pero sin embargo sí se pueden incluir ideas que hayamos rescatado de otras implementaciones satisfactorias. Algunos ejemplos son:

**2.2.1.2.1. Tate Modern (Londres, Reino Unido)** Pude comprobar en directo algunas de las ideas puestas en práctica por esta entidad gracias a un viaje reciente.

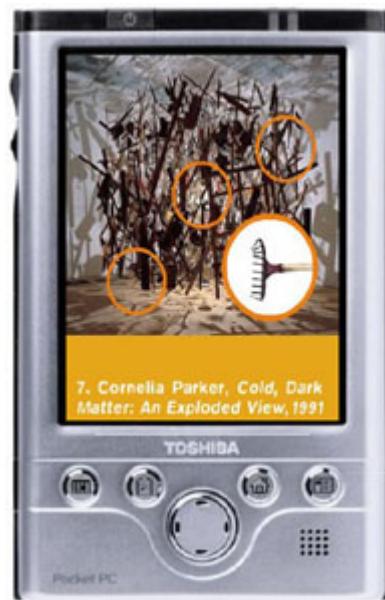


Figura 2.1: Imagen de dispositivo de mano del Tate Modern [1]

Allí pude ver buenas ideas como el acercamiento de los contenidos de las salas al gran público (en especial, a niños) a base de actividades interactivas que introducían con naturalidad y entretenimiento conceptos difíciles de asimilar de otra manera. Sin duda es una buena idea, aunque no se encuadra en este proyecto, podría ser considerada para futuros trabajos.

Más concretamente, en materia de museos virtuales, vi la adopción de terminales de mano que complementan la experiencia del visitante físico. Estamos acostumbrados a ver las típicas guías locutadas en diversos idiomas que nos amplían el contenido de las exposiciones introduciendo un número del índice de grabaciones. Suelen ser aparatos con usos muy restringidos (especiales para museos) con lo que eso supone para el coste final para el museo. En cambio, en este caso, se trataba de dispositivos móviles que están en el mercado, en nuestro caso, una especie de PDA de marca Toshiba o Dell (según renovaciones) similares a los que se muestran en la figura 2.1. Son dispositivos básicos que permiten mostrar contenidos web y multimedia sin muchas complicaciones. Esto es: pantalla en color, interfaz intuitiva, salida de auriculares y conectividad inalámbrica. Este último punto permite renovar sin coste temporal importante (y de manera transparente al visitante) los contenidos presentados. El hecho de que sea un dispositivo común abarataría los costes, permitiendo, incluso, trabajar con tecnología no necesariamente de última generación, dada la baja exigencia de los contenidos a presentar. Parece evidente que será más fácil conseguir un buen precio por una buena cantidad de dispositivos a punto de ser retirados del mercado o simplemente de stock sobrante. Debajo de todo esto subyace la idea de utilizar unos estándares mínimos, aunque sean estándares de facto. Esto nos permitiría (como en el caso del Tate Modern) reemplazar los dispositivos por otros nuevos sin hacer cambios significativos en la plataforma y haciendo que coexistiesen ambos modelos. Y lo que es, quizás, más interesante, dar la opción a usuarios que posean dispositivos con capacidades similares a los ofrecidos de conectarse al museo virtual por su cuenta y en directo, ahorrando bastante en costes.

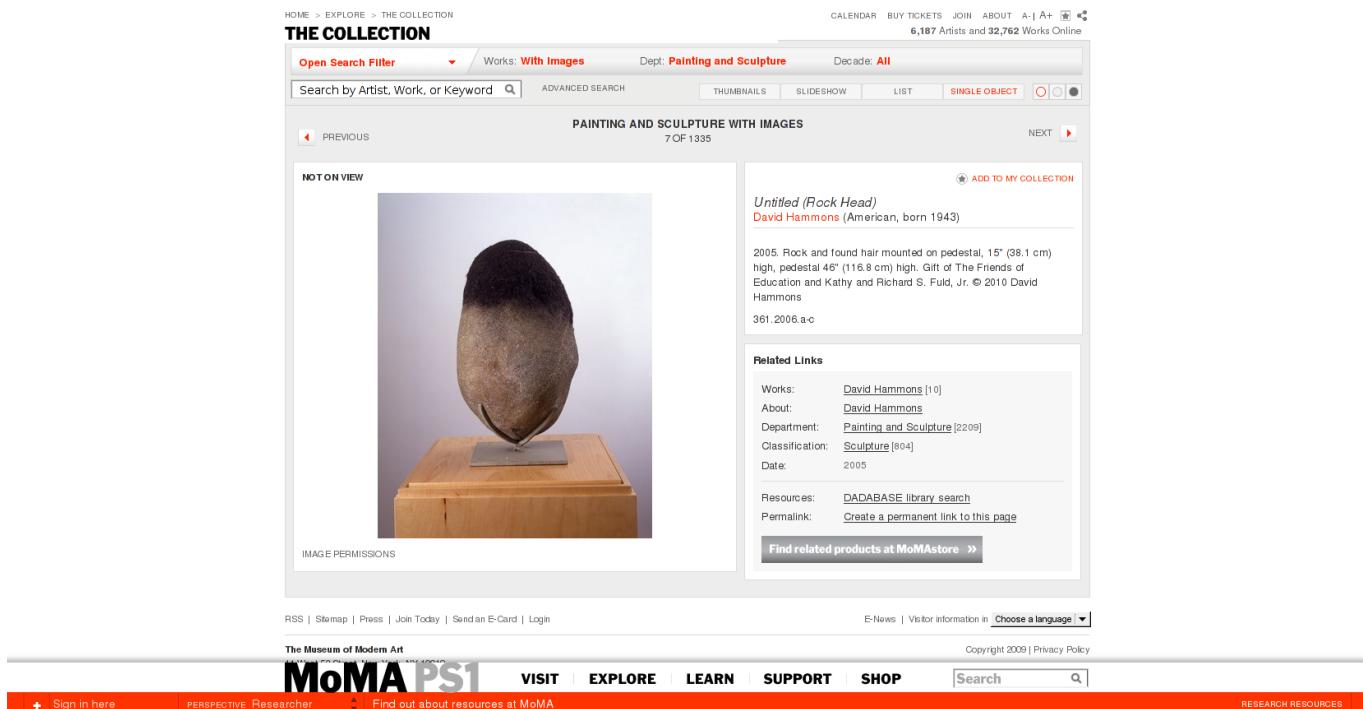


Figura 2.2: Captura de pantalla del sitio del MoMA

**2.2.1.2.2. MoMA (New York, EEUU)** Este apartado, desgraciadamente, se refiere a las ideas y modelos observados directamente en su página web. Nos encontramos con ideas, sobre todo en cuanto a presentación visual de contenidos. Digamos que la idea de presentar los contenidos en una matriz de imágenes con pies identificativos que, al pulsar sobre ellas, nos lleven a la información específica de ese objeto podría ser cercana a lo buscado. También sería una idea a tener en cuenta la categorización de las obras por área temática, época o tipo de objeto, aunque no lo firmamos como algo seguro en nuestra plataforma. Por último, reseñar que ellos parece que hacen (o al menos tienen la posibilidad de hacer) uso de la vista "por perspectivas". Esto sería como tener una lista desplegable en la pantalla que nos permitiera fácilmente cambiar entre los distintos puntos de vista (como si de idiomas se tratara) pudiendo adaptar la plataforma a una presentación para, por ejemplo, niños, estudiantes o expertos en el tema. El sistema finalmente escogido para implementar esa "granularidad" de contenidos se tomó de otro sitio web que veremos más adelante. Se puede observar una captura que muestra una idea de lo que sería una ficha de objeto en la figura 2.2

**2.2.1.2.3. Actual museo del OAN** Existe una versión antigua del museo virtual del Observatorio Astronómico Nacional, cuya información será contenida presumiblemente cuando se complete el contenido de la plataforma objeto de este proyecto. Si entramos al sitio web que lo hospeda dentro del sitio del Instituto Geográfico Nacional (Ministerio de fomento) descubrimos una estructura de navegación con algunas deficiencias para movernos entre objetos, para dar un vistazo rápido y desde luego una descontextualización importante. Podríamos catalogarlo como uno de esos museos virtuales del caso cuarto, un catálogo de objetos, como podemos apreciar en la figura 2.3. No se puede negar su utilidad pero son las funciones de las que carece las que hacen necesaria la creación del proyecto actual.

**Instituto Geográfico Nacional**

Inicio > Museo Virtual

**Instituto Geográfico Nacional**

**Museo Virtual**

## ALTIMETRÍA Y NIVELACIÓN

Encontrados 21: listado del 1 al 7

foto	nombre	firmado
	Mareógrafo	Constructor: La Maquinista Valenciana, Vda. De Enrique Climent, Valencia. España. Hacia 1874.
	Nivel de agua	Anónimo. Hacia 1850.
	Nivel de precisión	Brunner à París. Hacia 1860
	Nivel de precisión	Anónimo. Hacia 1890.
	Nivel de precisión	L. Casella & Co., 23 Hatton Garden, London. Hacia 1890.
	Nivel de precisión	Salmoiraghi, núm. 15. Italia. Hacia 1890
	Nivel de precisión	Anónimo. Hacia 1890.

**Siguiente >**

**Volver**

Figura 2.3: Captura de pantalla del museo virtual del OAN

### 2.2.1.3. Enciclopedias

Como parte de la gestión del conocimiento, además de las webs de los propios museos, he indagado en otras webs que se dedican a funciones parecidas (no olvidemos que el museo al que se refiere este trabajo dista de ser un museo de arte) a lo que queremos conseguir en cuanto a organización del conocimiento.

**2.2.1.3.1. Wikipedia** Digamos que este sería el paradigma de enciclopedia dinámica y colaborativa hoy en día. Es más o menos familiar para cualquier persona que use con asiduidad internet (sobre todo para buscar información) y aunque sus fuentes, consistencia y fiabilidad puedan ser cuestionables, su manera de disponer la información no puede tomarse en absoluto como mala. De aquí podríamos tomar

en consideración el aspecto colaborativo del medio, pero en reuniones con el cliente se convino que al ser una institución de referencia, los contenidos (aunque aportados por diferentes personas) debían ser acotados a unos poco usuarios (generalmente trabajadores) y moderados en última instancia por la figura de un editor que revisaría y daría el visto bueno a los mismos. La disposición de portal inicial ofrecida por este sitio es una idea que podría funcionar en nuestro caso, incluyendo esos pequeños “ganchos” que cambian diaria o aleatoriamente para que determinados artículos puedan ser descubiertos por un navegante no asiduo. Además, mantiene una división temática necesaria para el sitio y tiene un sistema de control de versiones válido para documentos escritos, así como la relación semántica de entradas y la inclusión de contenido multimedia fácilmente. Referido al punto anterior, hay que señalar que, en un principio, se ha optado por la manera de Wikipedia de ampliar la profundidad de un artículo: en vez de diseñar distintas capas para distintos individuos, enlazar los artículos que profundizan sobre un tema para que sea el usuario el que decida cuándo parar.

**2.2.1.3.2. Indianopedia, Frikipedia, \*pedia** Esta entrada engloba varios sitios por una razón: existe un fenómeno creciente que consiste en que cada “ecosistema” de conocimiento puede generar su propio repositorio de información. Esto contribuye a diversificar las fuentes evitando fraudes y manipulaciones, a la vez que descentraliza y permite pluralidad de visiones. Igual que existen el diccionario de la RAE, María Moliner, Espasa Calpe, etc., el concepto de la propia enciclopedia, en nuestro caso, puede servir para contener los artículos a los que no llegan otras encyclopedias más generales o de los que tenemos un conocimiento certero y fuentes propias.

#### **2.2.1.4. Sistemas gestores de contenidos vs Wikis**

Llegados a este punto parece evidente el camino que estamos sugiriendo. Para crear nuestra plataforma y que tenga algunos factores básicos como diferentes presentaciones al usuario (al menos internet y móviles), incluya una base de datos, interfaz amigable para creadores de contenido no informáticos y demás funciones de administrador facilitadas al máximo.

En un principio se tuvo la tentación de basar toda la plataforma en un mero wiki (La base de datos más simple del mundo) pero se vio que las características más destacables de un wiki nos sobraban salvo la facilidad de edición y de uso, que se puede apreciar también en los CMS más comunes. Además, claro está, si nos ciñéramos a un simple wiki estaríamos renunciando a una serie de herramientas útiles que ya se integran en los CMS con naturalidad.

De la toma de decisiones sobre la elección de plataforma base hablaré en la sección de diseño.

## 2.3. Descripción del diseño

### 2.3.1. Baremación de alternativas

Llegados a este punto, nuestro proyecto tiene que empezar a definirse y descartar unas alternativas de otras. Hasta la fecha, recapitulando, hemos puesto encima de la mesa la necesidad de un gestor de contenidos con o sin componente wiki. A este esqueleto inicial habría que añadirle una serie de características y su grado de importancia (donde 0 es prescindible y 5 es imprescindible). Este grado de importancia ha sido recogido en entrevista directa con responsables de la supervisión del proyecto y se muestra en el cuadro 2.1.

Característica	Nota
Interfaz de creador en español	4
Interfaz de usuario en múltiples idiomas	5
Varias perspectivas diferenciadas	1
Web semántica	0
Uso de estándares	0
Editor WYSIWYG	5
Soporte comercial	1
Comunidad de desarrolladores / soporte	4
Actualizaciones recientes	1
Entorno colaborativo	3
Buzón de sugerencia / Preguntas frecuentes	0
Página de control de cambios / CVS	5
Licencia libre	5
Base de datos	3
Sistema de ficheros (en contraposición a BD)	0
Posibilidad de exportar a pdf/LaTeX	1
Versión para plataformas móviles	4
Blog	0

Cuadro 2.1: Cuadro resumen de la ponderación de características

#### 2.3.1.1. Comentarios

La cuestión sobre las perspectivas diferenciadas se ha decidido plantear a la manera de Wikipedia, es decir, a base de enlaces con contenido más exhaustivo en vez de un selector de perspectiva que nos dejase “ponernos las gafas” de “estudiante”, “visitante” o “investigador” con un botón, al menos, en esta fase del proyecto.

Aun no habiendo resuelto temas importantes como si la figura del “responsable de mantenimiento” de la aplicación existirá y quién se encargaría de ella, en la misma reunión se esbozó la idea de que existiría una fase inicial de carga con un principio y un final y una fase de mantenimiento en la que se harán cambios pero a menor ritmo. Todas ellas irán recubiertas de la figura del supervisor, que se encargará de aprobar o rechazar los cambios u añadidos propuestos por los generadores de contenido antes de que éstos se reflejen en la producción final.

En contrapartida a las ideas de blog y buzón de sugerencias, se propondrá (con mayor o menor grado de insistencia) al usuario realizar una encuesta evaluación con su propio espacio para sugerencias.

El tema de uso de estándares y web semántica, si bien es un objetivo deseable, no debe obstaculizar el resto de ellos, se asume de la última reunión.

Por último, y para que la base de datos creada pueda ser fácilmente usada por otros proyectos, se ha optado por buscar una solución orientada a bases de datos y no a sistemas de ficheros como modo de almacenamiento. Esto descartará unas cuantas opciones.

### **2.3.2. Comparativa**

Habiendo descartado o relativizado la importancia de algunos de los requerimientos iniciales para nuestro gestor de contenidos, vamos a tratar de hacer una propuesta inicial. Para ello, vamos a utilizar herramientas gratuitas en línea que presentan los datos como una matriz de características (CMSMatrix.org y CMSMatch.com), aunque sólo se presentan en este documento las relativas a la primera, puesto que la segunda carecía de un montón de datos relevantes.

#### **2.3.2.1. Premisas iniciales**

Las premisas iniciales para los CMS (y las que más descartan) son:

- Más populares y mejor puntuados. De esta manera potencialmente hallaremos estabilidad y una comunidad de soporte activa. Para ello, echaremos un vistazo a las estadísticas disponibles en la red sobre CMS más utilizados, mejor puntuados y más premiados, tratando de sopesar adecuadamente cada fuente.
- Editor visual (WYSIWYG Editor / Drag'n'Drop interface).
- Con capacidad para gestionar la plataforma en varios idiomas (Internationalization = Multilingual Content / Multilingual Content Integration / Web-based Translation Management).
- Página de cambios (Versioning / Asset Management=Media Library)
- Coste cero.
- Licencia OpenSource.
- Comunidad de desarrolladores (Developer Community).
- Interfaz en español (Interface Localization).
- Versión para móviles (Mobile Internet).

#### **2.3.2.2. Candidatos pre-seleccionados**

Primeros elegidos por su demostrada utilización, premios, etc:

1. dotCMS
2. Drupal
3. Joomla!
4. LifeRay Portal
5. MODx
6. mojoPortal
7. Moodle
8. Plone
9. SilverStripe
10. WordPress

El primer problema que encontramos es que estos sistemas que hemos elegido para comparar los diferentes CMS no nos dejan saber qué idiomas tiene la interfaz de usuario, así que ha que hacer una pequeña investigación entrando en las páginas web de cada uno de los sitios. Tras esto, sabremos que, al menos en cuanto a la interfaz de administración, todas las plataformas preseleccionadas pueden ejecutarse en español.

A continuación, pasamos a desgranar las secciones más interesantes de la comparativa reseñada:

### **2.3.2.3. Comparativa con CMSMatrix**

En esta ocasión, vamos a ir comparando las características más reseñables de cada producto, por bloques, según está organizado en el servicio consultado.

**2.3.2.3.1. System Requirements (Requisitos de sistema)** Vamos a ir analizando lo que se ve en la figura 2.4:

En este caso, debemos fijarnos en la parte “License” principalmente, comprobando que todos menos Liferay Portal ostentan una licencia libre. Posteriormente he podido comprobar que la licencia de Liferay Portal cambió de licencia MIT a LGPL a partir de la versión 6 [3], así que, a falta de argumentación en contra sobre los pormenores de cada licencia, todos entrarían dentro del requisito principal.

Hay que comentar que una de las guías sugeridas por la empresa era que, de ser posible, la plataforma fuese familiar con lo que ya se conoce en la casa por parte de los trabajadores fijos. En este caso, las cosas irían si no ya por la plataforma en sí, por el lenguaje de programación el que están programados, siendo estos Java, PHP, Phyton y C Sharp los que ocupan nuestra comparativa.

System Requirements	dotCMS 1.9	Drupal 6.10	Joomla! 1.5.10	Liferay Portal 5.2	MODx 1.0.2	mojoPortal 2.3.1.3	Moodle 1.9	Plone 3.0	SilverStripe CMS 2.x Open Source 2.3.1	WordPress 2.2.1
Application Server	J2EE	Apache	CGI	J2EE	Apache	IIS/Net	Apache	Zope	Other	Apache
Approximate Cost	Depends on license model	Free	Free	Also available in an Enterprise Edition with professional support.	Free	Free	0	Free	No Cost	Free
Database	Other	MySQL	MySQL	DB2	MySQL	MSSQL	MySQL	Other	MySQL	MySQL
License	Open Source	Open Source	Open Source	Closed Source	Open Source	Open Source	Open Source	Open Source	Open Source	Open Source
Operating System	Platform Independent	Platform Independent	Platform Independent	Platform Independent	Platform Independent	Platform Independent	Platform Independent	Platform Independent	Platform Independent	Platform Independent
Programming Language	Java	PHP	PHP	Java	PHP	C#	PHP	Python	PHP	PHP
Root Access	No	No	No	Yes	Yes	No	No	No	No	No
Shell Access	Yes	No	No	Yes	No	No	No	Yes	No	No
Web Server	Other	Apache	Apache	Other	Apache	Apache	Any	Apache	Apache	Apache

Figura 2.4: “System Requirements” desde CMSMatrix.org

**2.3.2.3.2. Security (Seguridad)** Según el gráfico de la figura 2.5, los datos que más nos interesan son “Content Approval” (aprobar contenidos antes de publicación), “Granular Privileges” (posibilidad de tener distintos privilegios para cada usuario o grupo de usuarios), “Versioning” (control de versiones).

La única opción con inconvenientes en este campo sería Joomla!, que según cmsmatrix no permite la asignación de privilegios personalizada, señalando que MODx en supervisión de contenidos y control de versiones necesitaría un añadido gratuito, al igual que Joomla! para este último caso.

**2.3.2.3.3. Support (Asistencia)** Esta es una de las partes menos importantes de la comparativa, pues es incierto el futuro en cuestión de mantenimiento de la plataforma. En este caso nos estaríamos fijando más en “Commercial Support” (asistencia de pago), “Developer Community” (comunidad de desarrolladores) y “Public Forum” (foros públicos). Destacar únicamente que WordPress no dispone de ese servicio, aunque el servicio gratuito sí que exista. Para analizarla más en detalle, observar la figura 2.6

**2.3.2.3.4. Ease of Use (Facilidad de uso)** Este es uno de los puntos más importantes de la comparativa. Si observamos en la figura 2.7:

Encontraremos que, a destacar, podemos seleccionar como las más relevantes las opciones de “Drag-N-Drop Content” (arrastrar y soltar contenido), “Image Resizing” (redimensionado de imágenes), “Mass Upload” (subida de archivos masiva) y “WYSIWYG Editor” (editor visual).

De la primera característica, tenemos Joomla! que no la soporta, MODx con una implementación limitada y “Drupal” con un añadido para esa función. De la segunda, tendríamos que tanto “mojoPortal” como “WordPress” presentan esta función limitada y “Drupal” necesita, otra vez, añadidos. En cuanto a la subida de ficheros masiva, mojoPortal sería incapaz mientras que Drupal y WordPress necesitarían un add-on. El editor visual está presente en todas, aunque se nos avisa que Drupal necesita de un añadido también. Lo que nos hace pensar que Drupal está construido de forma modular ya que muchas de sus características se consiguen a través de módulos supplementarios. Esto puede tener consecuencias positivas en cuanto a agilidad, tamaño

Security	dotCMS 1.9	Drupal 6.10	Joomla! 1.5.10	Liferay Portal 5.2	MODx 1.0.2	mojoPortal 2.3.1.3	Moodle 1.9	Plone 3.0	SilverStripe CMS 2.x Open Source 2.3.1	WordPress 2.2.1
<input type="checkbox"/> Audit Trail	Yes	Yes	No	Yes	Yes	No	Yes	Yes	Yes	Limited
<input type="checkbox"/> Captcha	Yes	Free Add On	Free Add On	Yes	Yes	Yes	No	Free Add On	Yes	No
<input type="checkbox"/> Content Approval	Yes	Yes	Yes	Yes	Free Add On	Yes	Yes	Yes	Yes	Yes
<input type="checkbox"/> Email Verification	Yes	Yes	Yes	Free Add On	Yes	Yes	Yes	Yes	Yes	Free Add On
<input type="checkbox"/> Granular Privileges	Yes	Yes	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes
<input type="checkbox"/> Kerberos Authentication	Costs Extra	No	No	Yes	No	Yes	Yes	Free Add On	No	No
<input type="checkbox"/> LDAP Authentication	Yes	Free Add On	Yes	Yes	Free Add On	Yes	Yes	Yes	Free Add On	No
<input type="checkbox"/> Login History	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Free Add On	No	Free Add On
<input type="checkbox"/> NIS Authentication	Costs Extra	No	No	No	No	Yes	No	Free Add On	No	No
<input type="checkbox"/> NTLM Authentication	Costs Extra	Free Add On	No	Yes	No	Yes	Yes	Free Add On	No	No
<input type="checkbox"/> Pluggable Authentication	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
<input type="checkbox"/> Problem Notification	Yes	No	No	No	No	Yes	Yes	Free Add On	Yes	Free Add On
<input type="checkbox"/> Sandbox	Yes	No	No	Yes	Yes	Yes	Yes	Yes	Yes	Limited
<input type="checkbox"/> Session Management	Yes	Yes	Yes	Yes	Yes	Limited	Yes	Free Add On	Limited	Free Add On
<input type="checkbox"/> SMB Authentication	Yes	No	No	No	No	No	No	Free Add On	No	No
<input type="checkbox"/> SSL Compatible	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
<input type="checkbox"/> SSL Logins	Yes	No	Yes	Yes	Yes	Yes	Yes	Free Add On	Yes	Free Add On
<input type="checkbox"/> SSL Pages	Yes	No	Yes	Yes	Yes	Yes	Yes	No	Limited	Limited
<input type="checkbox"/> Versioning	Yes	Yes	Free Add On	Yes	Free Add On	Yes	Yes	Yes	Yes	Free Add On

Figura 2.5: “Security” desde CMSMatrix.org

Support	dotCMS 1.9	Drupal 6.10	Joomla! 1.5.10	Liferay Portal 5.2	MODx 1.0.2	mojoPortal 2.3.1.3	Moodle 1.9	Plone 3.0	SilverStripe CMS 2.x Open Source 2.3.1	WordPress 2.2.1
<input type="checkbox"/> Certification Program	Yes	No	No	Yes	No	No	Yes	No	No	No
<input type="checkbox"/> Code Skeletons	Yes	Yes	No	Yes	Yes	Yes	Yes	Yes	Yes	No
<input type="checkbox"/> Commercial Manuals	Yes	Yes	Yes	Yes	Yes	No	Yes	Yes	Yes	No
<input type="checkbox"/> Commercial Support	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No
<input type="checkbox"/> Commercial Training	Yes	Yes	Yes	Yes	No	Yes	Yes	Yes	Yes	No
<input type="checkbox"/> Developer Community	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
<input type="checkbox"/> Online Help	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
<input type="checkbox"/> Pluggable API	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
<input type="checkbox"/> Professional Hosting	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
<input type="checkbox"/> Professional Services	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No
<input type="checkbox"/> Public Forum	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
<input type="checkbox"/> Public Mailing List	Yes	Yes	No	Yes	No	Yes	Yes	Yes	Yes	Yes
<input type="checkbox"/> Test Framework	Yes	Free Add On	Yes	Limited	No	No	Yes	Yes	Yes	Yes
<input type="checkbox"/> Third-Party Developers	Yes	Yes	Yes	Yes	Yes	No	Yes	Yes	Yes	Yes
<input type="checkbox"/> Users Conference	Yes	Yes	Yes	Yes	Yes	No	Yes	Yes	No	Yes

Figura 2.6: “Support” desde CMSMatrix.org

Ease of Use	dotCMS 1.9	Drupal 6.10	Joomla! 1.5.10	Liferay Portal 5.2	MODx 1.0.2	mojoPortal 2.3.1.3	Moodle 1.9	Plone 3.0	SilverStripe CMS 2.x Open Source 2.3.1	WordPress 2.2.1
<input type="checkbox"/> Drag-&Drop Content	Yes	Free Add On	No	Yes	Limited	Yes	Yes	Yes	Yes	Yes
<input type="checkbox"/> Email To Discussion	Costs Extra	Free Add On	Free Add On	Yes	No	No	Yes	Free Add On	No	Limited
<input type="checkbox"/> Friendly URLs	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
<input type="checkbox"/> Image Resizing	Yes	Free Add On	Yes	Yes	Yes	Limited	Yes	Yes	Yes	Limited
<input type="checkbox"/> Macro Language	Yes	Free Add On	Yes	Yes	Yes	No	No	Yes	No	Free Add On
<input type="checkbox"/> Mass Upload	Yes	Free Add On	Yes	Yes	Yes	No	Yes	Yes	Yes	Free Add On
<input type="checkbox"/> Prototyping	Yes	Limited	Yes	Yes	Yes	No	No	Yes	Yes	No
<input type="checkbox"/> Server Page Language	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
<input type="checkbox"/> Spell Checker	Yes	Free Add On	No	Yes	Limited	Yes	Free Add On	Free Add On	Limited	Free Add On
<input type="checkbox"/> Style Wizard	Costs Extra	Limited	No	Limited	No	No	Yes	Free Add On	No	No
<input type="checkbox"/> Subscriptions	Yes	Free Add On	Costs Extra	Yes	Yes	Yes	Yes	Yes	Yes	Yes
<input type="checkbox"/> Template Language	Yes	Limited	Yes	Yes	Yes	Yes	No	Yes	Yes	No
<input type="checkbox"/> UI Levels	Yes	No	Yes	Yes	Yes	No	Yes	Yes	Yes	Yes
<input type="checkbox"/> Undo	Yes	Limited	No	Limited	Limited	No	Yes	Yes	Yes	Free Add On
<input type="checkbox"/> WYSIWYG Editor	Yes	Free Add On	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
<input type="checkbox"/> Zip Archives	Yes	No	No	Yes	Limited	No	Yes	Free Add On	No	Free Add On

Figura 2.7: “Ease of Use” desde CMSMatrix.org

Performance	dotCMS 1.9	Drupal 6.10	Joomla! 1.5.10	Liferay Portal 5.2	MODx 1.0.2	mojoPortal 2.3.1.3	Moodle 1.9	Plone 3.0	SilverStripe CMS 2.x Open Source 2.3.1	WordPress 2.2.1
<input type="checkbox"/> Advanced Caching	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Limited	Free Add On
<input type="checkbox"/> Database Replication	Yes	Limited	No	Yes	No	Yes	Yes	Yes	No	No
<input type="checkbox"/> Load Balancing	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Limited	No
<input type="checkbox"/> Page Caching	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Free Add On
<input type="checkbox"/> Static Content Export	Yes	No	No	Yes	Yes	No	No	Free Add On	Yes	Limited

Figura 2.8: “Performance” desde CMSMatrix.org

y complejidad y negativas en cuanto que el proceso inicial de puesta en marcha puede (aún no lo sabemos) ser más complicado que otros.

**2.3.2.3.5. Performance (Rendimiento)** Este apartado (fig. 2.8 no contiene aspectos demasiado relevantes para el proyecto que tenemos entre manos, pues no se espera que la carga de trabajo del servidor sea elevada como para preocuparse por estos detalles, aunque llegado el momento del desempate, podría declinar la balanza entre una opción y otra. Siendo tanto “WordPress” como “Joomla!” los más perjudicados.

Management	dotCMS 1.9	Drupal 6.10	Joomla! 1.5.10	Liferay Portal 5.2	MODx 1.0.2	mojoPortal 2.3.1.3	Moodle 1.9	Plone 3.0	SilverStripe CMS 2.x Open Source 2.3.1	WordPress 2.2.1
<input type="checkbox"/> Advertising Management	Yes	Free Add On	Yes	Yes	Free Add On	Limited	No	Free Add On	Limited	No
<input type="checkbox"/> Asset Management	Yes	Yes	Yes	Yes	Yes	Limited	Yes	Yes	Yes	Limited
<input type="checkbox"/> Clipboard	Yes	No	No	No	Yes	No	Yes	Yes	Yes	No
<input type="checkbox"/> Content Scheduling	Yes	Free Add On	Yes	Yes	Yes	Yes	No	Yes	Yes	Free Add On
<input type="checkbox"/> Content Staging	Yes	Free Add On	No	Yes	Yes	No	Yes	Free Add On	Yes	No
<input type="checkbox"/> Inline Administration	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Limited	No
<input type="checkbox"/> Online Administration	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Limited	Yes
<input type="checkbox"/> Package Deployment	Yes	No	No	Yes	Free Add On	Yes	Yes	Yes	No	No
<input type="checkbox"/> Sub-sites / Roots	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No
<input type="checkbox"/> Themes / Skins	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
<input type="checkbox"/> Trash	Yes	No	Yes	No	Yes	No	No	Free Add On	No	No
<input type="checkbox"/> Web Statistics	Yes	Yes	Yes	Limited	Yes	Limited	Yes	Free Add On	Limited	Free Add On
<input type="checkbox"/> Web-based Style/Template Management	Yes	Yes	Yes	Yes	Yes	No	Yes	Yes	No	Yes
<input type="checkbox"/> Web-based Translation Management	Yes	Yes	Free Add On	Yes	No	No	Yes	Yes	Yes	Limited
<input type="checkbox"/> Workflow Engine	Yes	Limited	No	Limited	No	No	Yes	Yes	Yes	No

Figura 2.9: “Management” desde CMSMatrix.org

**2.3.2.3.6. Management (Administración)** La figura 2.9 presenta aspectos importantes para el administrador de la plataforma como: “Online Administration” (gestión en línea), “Themes/Skins” (temas gráficos), “Web-based Style/Template Management” (gestión de estilos y plantillas vía web) y “Web-based Translation Management” (gestión de traducciones vía web).

Las características que he citado primera y cuarta son, a mi juicio, fundamentales, ya que permiten la gestión de la plataforma virtualmente desde cualquier dispositivo que pueda mostrar páginas web. Las fallas más significativas son las de “MODx” y “mojoPortal”, que gracias a la falta de gestor de traducciones vía web quedarían, por ahora, descalificadas por completo. Hay que destacar que “WordPress” tiene limitaciones en esto y acumularía ya unas cuantas pegas que no lo hacen muy aconsejable.

**2.3.2.3.7. Interoperability (Interoperatividad)** Toda las características mostradas en la figura 2.10 tienen una importancia técnica alta, aunque su repercusión en el resultado visual final no sea grande. Quiero, en este apartado, analizarlas una por una para que se sepa de qué trata cada una.

Content Syndication (RSS) (sindicación de contenidos): Se trata de un sistema que permite a los usuarios estar al tanto de las nuevas publicaciones de una plataforma. Como en nuestro caso el contenido será mayoritariamente estático no es algo relevante.

FTP Support (soporte para transferencia de ficheros por FTP): FTP es un protocolo especialmente diseñado para la transferencia de ficheros, siendo, aunque ahora menos utilizado, un valor más. Su importancia es media.

UTF-8 Support (soporte para codificación UTF-8): Esta codificación es la que pre-

Interoperability	dotCMS 1.9	Drupal 6.10	Joomla! 1.5.10	Liferay Portal 5.2	MODx 1.0.2	mojoPortal 2.3.1.3	Moodle 1.9	Plone 3.0	SilverStripe CMS 2.x Open Source 2.3.1	WordPress 2.2.1
<input type="checkbox"/> Content Syndication (RSS)	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
<input type="checkbox"/> FTP Support	Costs Extra	Limited	Yes	Limited	Yes	No	Yes	Yes	Limited	Free Add On
<input type="checkbox"/> UTF-8 Support	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
<input type="checkbox"/> WAI Compliant	Costs Extra	Limited	No	Limited	Yes	Yes	Yes	Yes	Yes	Limited
<input type="checkbox"/> WebDAV Support	Yes	No	No	Yes	Free Add On	No	Yes	Yes	No	No
<input type="checkbox"/> XHTML Compliant	Yes	Yes	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes

Figura 2.10: “Interoperability” desde CMSMatrix.org”

tende ser universal para la inmensa mayoría de idiomas. Utilizarla es un paso adelante para la interoperatividad de aplicaciones, pero no es altamente relevante para el proyecto.

WAI Compliant (cumple con WAI): Web Accessibility Initiative (Iniciativa para la Accesibilidad Web) es una serie de reglas que buscan la accesibilidad de los contenidos para usuarios impedidos. Propone unos mínimos y puede ser interesante implementarlo, aunque no vital.

WebDAV Support (soporte para WebDAV): “Edición y versionado distribuidos sobre la web” es, en esencia, un protocolo que nos permite cambiar a distancia los contenidos de un servidor con cualquier aplicación que lo implemente (sin necesidad de una herramienta propia). No es imprescindible ni es el único modo.

XHTML Compliant (cumple con XHTML): XHTML es una especificación que trata de dar coherencia y acercar el lenguaje HTML a los estándares XML. Busca la ausencia de errores (se puede validar con facilidad) y la fiel reproducción de un contenido en distintos navegadores sin demasiadas diferencias. Es decir, busca establecer un estándar. Yo lo consideraría un mínimo exigible para cualquier plataforma web aunque en el futuro se de la posibilidad de cambiar a otros estándares como HTML5 (que aún no están implantados masivamente).

**2.3.2.3.8. Flexibility (Flexibilidad)** De esta sección, mostrada en la figura 2.11, lo que más nos va a interesar es la “Interface Localization” (interfaz en el idioma deseado), la capacidad de presentar e integrar contenido en distintas lenguas, representado por “Multilingual Content” y “Multilingual Content Integration”. Tanto esta como las dos siguientes secciones, como se comprueba en el estilo austero, no son tan relevantes como las ya comentadas.

**2.3.2.3.9. Built-in Applications (Aplicaciones embebidas)** En esta sección se presentan todas esas funcionalidades extras que podrían (o no) interesarnos a la hora de diseñar nuestra plataforma (figuras 2.12 y 2.13).

Entre las presentes, me parecen reseñables las que tienen que ver con calendarios y eventos, manejo de enlaces, encuestas, manejo de documentos y galería de imágenes, entre otras, aunque, en general, no son demasiado importantes porque no sería difícil recrearlas.

Flexibility	dotCMS 1.9	Drupal 6.10	Joomla! 1.5.10	Liferay Portal 5.2	MODx 1.0.2	mojoPortal 2.3.1.3	Moodle 1.9	Plone 3.0	SilverStripe CMS 2.x Open Source 2.3.1	WordPress 2.2.1
<input type="checkbox"/> CGI-mode Support	No	Yes	Yes	No	Yes	No	Yes	Yes	Yes	No
<input type="checkbox"/> Content Reuse	Yes	Limited	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No
<input type="checkbox"/> Extensible User Profiles	Yes	Yes	Yes	Limited	Free Add On	Yes	Yes	Yes	No	No
<input type="checkbox"/> Interface Localization	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
<input type="checkbox"/> Metadata	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
<input type="checkbox"/> Multi-lingual Content	Yes	Yes	Free Add On	Yes	Yes	Yes	Yes	Yes	Free Add On	
<input type="checkbox"/> Multi-lingual Content Integration	Yes	Free Add On	Free Add On	Yes	Free Add On	No	Yes	Yes	Yes	Free Add On
<input type="checkbox"/> Multi-Site Deployment	Yes	Yes	Free Add On	Yes	Free Add On	Yes	Yes	Yes	Free Add On	No
<input type="checkbox"/> URL Rewriting	Yes	Yes	Yes	Yes	Yes	Yes	No	Yes	Yes	Yes

Figura 2.11: “Flexibility” desde CMSMatrix.org

**2.3.2.3.10. Commerce (Comercio electrónico)** Esta última figura (2.14) se incluye casi a modo testimonial, dado que la parte comercial tanto del museo como, más generalmente, de todo el IGN, se concentra en otro departamento, fuera del alcance de este proyecto. En ella podemos ver distintas funcionalidades orientadas a la parte más comercial de la actividad: tienda, carrito, pasarela, etc.

Dicho todo esto, considero que Joomla! también debería ser apartado de los candidatos dada la falla en cuanto a XHTML. Los costes extra de dotCMS para algunas características me hacen pensar que la filosofía de ese proyecto no es tan cercana a nuestros objetivos como podría parecer, así que también los desaconsejaría.

#### 2.3.2.4. Propuesta preliminar



Los candidatos serán cuatro. Todos ellos son multiplataforma, corren en cualquier servidor y se componen de código abierto. Todos, además, disponen de interfaz en idioma español e inglés (como mínimo), galería de fotos y sistema de control de versiones (más o menos completo). Todos funcionan alimentándose de una base de datos con API estándar.

Como particularidades reseñables, podemos decir que, en cuanto a servidor web, Plone y Drupal están muy bien integrados con Apache, mientras que Liferay Portal se integra con otros servidores más orientados a Java (Tomcat, Glassfish, etc.) y Moodle que se integra con cualquier plataforma que tenga soporte para PHP.

##### 2.3.2.4.1. Liferay Portal (Community Edition)



Esta herramienta no presenta inconvenientes importantes ahora que su código es libre [3], pues antes no era así. Más a su favor, tiene una empresa como IBM detrás lo

Built-in Applications	dotCMS 1.9.1.3	Drupal 6.10	Joomla! 1.5.10	Liferay Portal 5.2	MODx 1.0.2	mojoPortal 2.3.1.3	Moodle 1.9	Plone 3.0	SilverStripe 2.4.2	WordPress 2.2.1
<input type="checkbox"/> Blog	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
<input type="checkbox"/> Chat	Costs Extra	Free Add On	Free Add On	Yes	Free Add On	Yes	Yes	Free Add On	No	Free Add On
<input type="checkbox"/> Classifieds	Yes	Free Add On	Free Add On	No	No	No	Yes	Free Add On	No	Free Add On
<input type="checkbox"/> Contact Management	Yes	Free Add On	Yes	Yes	Free Add On	No	Yes	Free Add On	No	Free Add On
<input type="checkbox"/> Data Entry	Yes	Free Add On	Free Add On	No	Free Add On	No	Yes	Free Add On	Yes	No
<input type="checkbox"/> Database Reports	Yes	No	Free Add On	No	Yes	No	Yes	Yes	Yes	No
<input type="checkbox"/> Discussion / Forum	Costs Extra	Yes	Free Add On	Yes	Free Add On	Yes	Yes	Free Add On	Yes	Free Add On
<input type="checkbox"/> Document Management	Yes	Limited	Free Add On	Yes	Free Add On	Yes	Yes	Yes	Limited	No
<input type="checkbox"/> Events Calendar	Yes	Free Add On	Free Add On	Yes	Free Add On	Yes	Yes	Yes	Limited	Free Add On
<input type="checkbox"/> Events Management	Yes	Free Add On	Free Add On	Limited	Free Add On	No	Yes	Yes	Limited	No
<input type="checkbox"/> Expense Reports	Costs Extra	No	Free Add On	Limited	No	No	Yes	Free Add On	No	No
<input type="checkbox"/> FAQ Management	Yes	Yes	Yes	Limited	Free Add On	No	Yes	Free Add On	Yes	Free Add On
<input type="checkbox"/> File Distribution	Yes	Free Add On	Free Add On	Yes	Free Add On	Yes	Yes	Yes	Limited	Free Add On
<input type="checkbox"/> Graphics and Charts	Yes	No	Free Add On	Free Add On	Free Add On	No	Yes	Free Add On	Limited	No
<input type="checkbox"/> Groupware	Yes	Free Add On	Free Add On	Yes	No	No	Yes	Free Add On	No	No
<input type="checkbox"/> Guest Book	Yes	Free Add On	Free Add On	Yes	Free Add On	No	Yes	Free Add On	Yes	Free Add On
<input type="checkbox"/> Help Desk / Bug Reporting	Costs Extra	Free Add On	Free Add On	Costs Extra	No	No	Yes	Free Add On	Yes	No
<input type="checkbox"/> HTTP Proxy	Yes	No	No	Yes	No	No	No	Free Add On	Yes	No
<input type="checkbox"/> In/Out Board	Costs Extra	No	No	No	No	No	Yes	Free Add On	No	No
<input type="checkbox"/> Job Postings	Yes	Free Add On	Free Add On	No	No	No	Yes	Free Add On	Yes	Free Add On
<input type="checkbox"/> Link Management	Yes	Free Add On	Yes	Yes	Free Add On	Yes	Yes	Yes	Yes	Yes
<input type="checkbox"/> Mail Form	Yes	Free Add On	Yes	Yes	Free Add On	Yes	Yes	Free Add On	Yes	Free Add On
<input type="checkbox"/> Matrix	Yes	No	No	No	No	No	No	No	No	No
<input type="checkbox"/> My Page / Dashboard	Yes	Free Add On	No	Yes	Free Add On	Yes	Yes	Yes	No	Yes

Figura 2.12: “Built-in Applications” (I) desde CMSMatrix.org

que supone que su código viene siendo mantenido y desarrollado por profesionales. Es una solución seria, empresarial y con una curva de aprendizaje media. Se usa en sitios como los de CISCO, Honda, T-Mobile, Harvard Business School, China Mobile, etc. La rumorología parece indicar que sería la solución que el Instituto Geográfico Nacional estaría adoptando para su gestión de contenidos hipertextuales, con lo que, de ser cierto, ganaría puntos como solución definitiva, dado que habría personal cualificado para mantener la aplicación si fuese necesario. Además de todo lo anterior, posee soporte comercial consolidado y está bastante probado.

Todo lo anterior la convierte en una opción sólida a mi modo de ver.

<input type="checkbox"/> <b>Newsletter</b>	Yes	Free Add On	Free Add On	Limited	Free Add On	Yes	Yes	Free Add On	Yes	Free Add On
<input type="checkbox"/> <b>Photo Gallery</b>	Yes	Free Add On	Free Add On	Yes	Free Add On	Yes	Yes	Yes	Yes	Free Add On
<input type="checkbox"/> <b>Polls</b>	Yes	Yes	Yes	Yes	Free Add On	Yes	Yes	Free Add On	Yes	Free Add On
<input type="checkbox"/> <b>Product Management</b>	Yes	Free Add On	Yes	Limited	Free Add On	Yes	Yes	Yes	Limited	No
<input type="checkbox"/> <b>Project Tracking</b>	Costs Extra	Free Add On	Free Add On	Limited	No	No	Yes	Free Add On	Yes	No
<input type="checkbox"/> <b>Search Engine</b>	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
<input type="checkbox"/> <b>Site Map</b>	Yes	Free Add On	Free Add On	Yes	Yes	Yes	No	Yes	Yes	Free Add On
<input type="checkbox"/> <b>Stock Quotes</b>	Costs Extra	Free Add On	Free Add On	Yes	Free Add On	No	Free Add On	Free Add On	No	No
<input type="checkbox"/> <b>Surveys</b>	Free Add On	Free Add On	Free Add On	Yes	No	Yes	Yes	Free Add On	Yes	Free Add On
<input type="checkbox"/> <b>Syndicated Content (RSS)</b>	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
<input type="checkbox"/> <b>Tests / Quizzes</b>	Costs Extra	Free Add On	Free Add On	No	No	No	Yes	Free Add On	Yes	Free Add On
<input type="checkbox"/> <b>Time Tracking</b>	Costs Extra	Free Add On	No	Costs Extra	No	No	Yes	Free Add On	No	Free Add On
<input type="checkbox"/> <b>User Contributions</b>	Yes	Yes	Yes	Yes	Free Add On	No	Yes	Yes	Yes	Yes
<input type="checkbox"/> <b>Weather</b>	Yes	Free Add On	Free Add On	Yes	Free Add On	No	No	Free Add On	No	Free Add On
<input type="checkbox"/> <b>Web Services Front End</b>	Yes	Limited	Yes	Yes	Yes	No	No	No	Yes	Free Add On
<input type="checkbox"/> <b>Wiki</b>	Limited	Free Add On	Free Add On	Yes	No	No	Yes	Free Add On	No	Free Add On

Figura 2.13: “Built-in Applications” (II) desde CMSMatrix.org

<b>Commerce</b>	dotCMS 1.9.1.3	Drupal 6.10	Joomla! 1.5.10	Liferay Portal 5.2	MODx 1.0.2	mojoPortal 2.3.1.3	Moodle 1.9	Plone 3.0	SilverStripe 2.4.2	WordPress 2.2.1
<input type="checkbox"/> <b>Affiliate Tracking</b>	Costs Extra	Free Add On	Free Add On	No	No	No	No	No	No	No
<input type="checkbox"/> <b>Inventory Management</b>	Yes	Free Add On	Free Add On	Yes	Free Add On	No	Yes	Free Add On	No	No
<input type="checkbox"/> <b>Pluggable Payments</b>	Yes	Free Add On	Free Add On	Limited	Free Add On	Yes	Yes	Free Add On	Yes	No
<input type="checkbox"/> <b>Pluggable Shipping</b>	Yes	Free Add On	Free Add On	Limited	Free Add On	No	No	Free Add On	Yes	No
<input type="checkbox"/> <b>Pluggable Tax</b>	Yes	Free Add On	Free Add On	Limited	Free Add On	No	No	Free Add On	Yes	No
<input type="checkbox"/> <b>Point of Sale</b>	Costs Extra	No	Free Add On	Yes	No	No	No	Free Add On	No	No
<input type="checkbox"/> <b>Shopping Cart</b>	Yes	Free Add On	Free Add On	Yes	Free Add On	Yes	No	Free Add On	Yes	No
<input type="checkbox"/> <b>Subscriptions</b>	Costs Extra	Free Add On	Free Add On	No	Free Add On	No	No	Free Add On	No	No
<input type="checkbox"/> <b>Wish Lists</b>	Costs Extra	Free Add On	Free Add On	No	No	No	Yes	No	Yes	No

Figura 2.14: “Commerce” desde CMSMatrix.org

### 2.3.2.4.2. Drupal

Se trata de una de las plataformas más populares y amigables para la gestión de contenidos. Es usada para todo tipo de proyectos desde poco importantes hasta de

envergadura media-alta. Sin embargo, no parece (por lo leído en algunos artículos) que sea la opción más potente. No tiene, tampoco, carencias que señalar, destacando, eso sí, su carácter modular, necesitando así para algunas de las funcionalidades deseables de la instalación de módulos adicionales (gratuitos). Tiene una gran base de usuarios y está mantenido por la comunidad principalmente. Destaca, además, por su buena integración con plataformas móviles. Es rápido, intuitivo y versátil. También una opción muy destacable

#### **2.3.2.4.3. Moodle**

Este CMS está orientado más al e-learning aunque es cierto que cumple con prácticamente todas las funciones deseables. Se trata de un software maduro y completo. Está programado en PHP, al igual que el anterior.

#### **2.3.2.4.4. Plone**

Probablemente este CMS sea el más innovador de los presentados. Muy potente, está programado en Python y construido encima del framework Zope, independientemente del servidor que vaya a correr por debajo (que en general será Apache). Al igual que el resto, no presenta deficiencias importantes, si bien, como Drupal, está pensado de manera modular, con añadidos para según qué funciones. Si hay que poner una pega (y no es trivial), es su curva de aprendizaje, dado que llena los huecos dejados por Drupal en cuanto a potencia pero también en cuanto a dificultad de uso haciéndolo una opción no recomendable para no-informáticos, quizás.

### **2.3.3. Anexo a comparativa**

Como resultado de la comunicación con el cliente, se hace una rápida referencia a una plataforma como candidata a ser la destinataria de la migración del CMS de todo el Instituto Geográfico Nacional. Normalmente, todo lo relacionado con la gestión de contenidos y, en general, todo el software generado suele ser subcontratado a otras empresas, dejando los desarrollos propios o artesanales casi exclusiva y paradójicamente al personal especializado (astrónomos, físicos, etc.).

Debido a la variabilidad de plazos y la diferencia apreciable de alcance en los proyectos del IGN y de este proyecto respectivamente, al final se decidió abandonar la línea de Alfresco sobre Liferay Portal y se optó por Drupal, que reunía los requisitos adecuados y tiene una gran comunidad de usuarios y desarrolladores detrás.

Sin embargo, he creído conveniente adjuntar un anexo donde se indague un poco más sobre Alfresco y sus posibilidades (6.1).

## **2.4. Drupal**

Como dije más arriba, Drupal será inicialmente la plataforma seleccionada para comenzar con la integración y desarrollo del sistema. Se trata de un sistema complejo, del que prefiero describir algunos fundamentos básicos que ayuden a entenderlo todo.

Existen multitud de recursos en Internet sobre Drupal, empezando por su sitio web principal [4], pero el recurso más útil que he encontrado es el libro editado por O'Reilly [5] y creado por el staff del sitio web Lullabot [6]. Aunque la versión traducida al español editada por Anaya [7] facilita el seguimiento general del libro, considero que en algunos aspectos es una traducción, cuanto menos, apresurada.

#### 2.4.1. Entorno de (pre)desarrollo

Para la prueba inicial he utilizado un "software appliance", es decir, una imagen de disco dispuesta a ser arrancada por una máquina virtual (en este caso VirtualBox) y empezar a funcionar. esto facilita mucho las cosas a la hora de realizar configuraciones rápidas y dinámicas dejando para más adelante (cuando tengamos una arquitectura definida) la integración y configuración de un servidor real.

La máquina elegida fue sacada de la web de TurnKey [8] en la que se pueden encontrar multitud de imágenes listas para hacer la función de servidor web con CMS nada más ser arrancadas. Suele tratarse de un montaje LAMP (Linux + Apache + MySQL + PHP) [9] En concreto se utilizó la que integraba la versión 6 de Drupal [10] con un entorno que se describe con los siguientes componentes básicos:

- TurnKey Core [11], compuesto de:
  - Ubuntu 8.04.3 LTS
  - Auto-actualizable en el arranque
  - Shell en AJAX incorporada (shellinabox [12])
  - Interfaz de administración web (Webmin [13])
  - Apache, MySQL y PHP.
- Soporte para SSH por defecto.
- MTA integrado para el envío de correo local.
- Módulos de Webmin para PHP, Apache2 y MySQL.
- PHPMyAdmin como administrador de la base de datos.
- Módulos más conocidos de Drupal (CCK, Views, Lightbox2, ImageCache, CKEditor, etc.).

En este caso, a la configuración inicial se le hicieron varias modificaciones, como instalar la distribución en español (que era uno de los requisitos) e instalar un módulo FTP en Webmin. Para la primera cosa hubo que borrar la instalación que venía por defecto con el "appliance" y volver a instalar Drupal en español descargándolo desde su página [14]. Para la segunda, sin embargo, salvo modificar ligeramente permisos en el servidor vía SSH, el propio Webmin es bastante intuitivo.

#### 2.4.2. *Understanding Drupal (Entender Drupal)*

Como en el libro original, una buena manera de comenzar nuestro periplo por este CMS es conocer sus fundamentos.

Se vio más arriba que Drupal es un CMS gratuito y de código abierto con una amplia comunidad de usuarios y colaboradores, lo que lo convierte en una plataforma bastante actualizada, potente y en contacto con las necesidades demandadas por la mayoría de usuarios.

#### **2.4.2.1. General**

En su concepción, Drupal no necesita un sistema operativo, servidor web o gestor de bases de datos concreto. Puede valer cualquiera capaz de correr una implementación de PHP y una base de datos compatible con SQL. Digamos que el “alma” de Drupal estaría contenida en la base de datos y se articularía gracias al “core” que hace uso del intérprete PHP. Además de los subsistemas que permiten el interpretado de temas gráficos, la gestión de sesiones de usuario y el filtrado para mantener la seguridad del sistema. Alrededor de este núcleo tendríamos los módulos (tanto incluidos por defecto como los que han sido asimilados gracias a la comunidad de desarrolladores) que extenderían la funcionalidad del sistema a lugares más específicos que van desde la creación de contenidos hasta la asistencia a tareas de administración pasando por todo tipo de funciones imaginables. Los contenidos de Drupal se basan en un concepto simple pero efectivo: los nodos. Todos los contenidos son organizados en nodos que pueden ser accedidos independientemente. Serían, por así decirlo, las unidades mínimas de contenido en Drupal. Todos ellos, además de los temas, configuraciones y usuarios se almacenan en la base de datos. A la hora de servir las páginas, el motor de Drupal en PHP trasladará esa información de la base de datos, produciendo una salida en XHTML con CSS (al menos, predeterminadamente) gracias a la interpretación de las plantillas que acompañan a los temas gráficos.

#### **2.4.2.2. Módulos**

Los módulos, tanto los que se incluyen oficialmente junto al “core”, como los que son aportados y mantenidos por la comunidad desarrolladores, son los que aportan la funcionalidad más cercana al usuario. Articulan lo que de otra manera sería una simple plataforma de edición de páginas web con una base de datos como soporte fundamental. En esencia, son archivos que contienen código php que proporciona un conjunto de funcionalidades y se integran en Drupal. Estas funcionalidades, salvo en las excepciones de módulos necesarios para el funcionamiento “nuclear” de Drupal, pueden ser activados o desactivados en cualquier momento, claro está, atendiendo a las interdependencias entre unos módulos y otros.

Prácticamente todas las funcionalidades, desde la gestión de usuarios y credenciales (módulo User) o los formularios de contacto con el responsable de la página (módulo Contact) hasta la creación de Galerías de imágenes o taxonomías para ordenar el contenido semánticamente, son aportadas por estos componentes. La verdadera potencia de los módulos reside en la interconexión de los mismos y el ajuste minucioso de las numerosas opciones de configuración que aporta cada uno de ellos. Nunca podrá conseguirse un sistema a pleno rendimiento si las opciones se dejan por defecto (aunque esta generalidad aplica a todo sistema informático de cierto nivel). Esta potencia que se nos ofrece en un principio, puede, dado la gran cantidad de alternativas disponibles, sin embargo, convertirse en un dolor de cabeza para dar con el módulo adecuado a nuestras necesidades (dado que hay varios que pueden aportar

similares funcionalidades).

#### 2.4.2.3. Usuarios

Una de los cuerpos más importantes de un sistema informático es la gestión de usuarios. Su mayor baza es la de personalizar el entorno según las credenciales de la persona (o máquina) que accede. En el caso de Drupal, por defecto, encontramos un sistema de usuarios con multitud de opciones. Por defecto, se crea el "User 1", que sería algo así como el usuario "root" en sistemas UNIX, es decir, un superusuario con derechos plenos sobre todo el sistema. Si bien la clásica organización en usuarios podría ser una solución, en el caso de Drupal, se añade el concepto de "rol". Un rol, siguiendo con la analogía de UNIX, sería algo similar a un grupo de usuarios. De esta manera, se permite separar el espacio individual posible de cada usuario (que en nuestro caso no tendría demasiada utilidad, salvo la meramente identificativa a la hora de hacer cambios y puede que la capacidad de tener entradas como borradores) de su asignación de permisos. Los permisos son asignados a un rol determinado, y ese rol determinado puede ser asignado a varios usuarios, permitiendo, en todo caso, que todo usuario (incluso los anónimos) estén bajo control a la hora de decidir el acceso, creación, modificación y borrado de datos, así como los contenidos presentados en cada caso.

#### 2.4.2.4. Nodos

Este punto podría decirse que es la piedra angular de Drupal. Si en un CMS lo importante son los contenidos, en Drupal, si queremos hablar de contenidos, tenemos que remitirnos a los nodos. Como dijimos anteriormente, los nodos son la unidad mínima de publicación de contenidos en Drupal. No es raro, al visitar un sitio al azar (dada la extendida implantación de Drupal en los sitios web) que topemos con urls que contengan, entre barras, la palabra "node", aunque esto sea, según algunos gurús, algo a evitar por los desarrolladores, ya sea mediante alias que escondan esas urls o mediante plantillas que engloben los nodos como un componente más. Y ahí radica una de las características más poderosas de los nodos y es que pueden ser tomados como partes de estructuras más grandes, permitiendo cosas como que una foto, su título, y si se quiere, un comentario, puedan ser en sí mismas un nodo, y a su vez, este nodo, formar parte como referencia en otro nodo. Todo esto aporta una fortaleza importante a la hora de indexar contenidos, proyectar búsquedas o presentar las entradas de diferentes maneras y atendiendo a unos y otros criterios, ya que todos los contenidos parten de una estructura común y son, hasta cierto punto, compatibles entre sí.

Los nodos, en su estructura más básica, están compuestos por:

- Autor
- Fecha de creación
- Título del nodo
- Contenido principal

A partir de ahí pueden darse diferenciaciones o mutaciones que permitan distinguir entre distintos tipos de contenido. Como por ejemplo la diferencia existente entre Página (Page) y Artículo (Story), que prepara la página para ser un contenido esencialmente estático e informativo (incluso institucional) y el artículo para ser algo más dinámico, corriente y genérico. “Page” y “Story” son los dos medios básicos que son incluidos por defecto en Drupal, dejando Page para páginas tipo “¿Quiénes somos?” o “Política de privacidad”, que no requieren interactividad, ni cambios y que están preparadas para ocultar por defecto la fecha y el autor del contenido. Sin embargo, “Story” contaría con la intención (al menos en su creación) de formar parte de lo que llamaríamos la portada del sitio, como una “noticia” más o una “novedad” más, que tiene un contexto temporal y un autor más relevantes.

Combinando las dos funcionalidades pasadas podremos asignar las capacidades de los usuarios (a través de sus roles asignados) individualmente para cada tipo de contenido. Dejando, puede ser, un contenido institucional a un editor y un contenido más dinámico a un redactor (o cualquier otra combinación que se nos ocurra).

#### **2.4.2.5. Formas de organizar el contenido**

Dejando de lado la parte más semántica e íntegra del sistema, las cuestiones organizativas son las que afloran. En este caso, hay distintas aproximaciones posibles que nos permiten presentar y complementar los contenidos descritos más arriba.

**2.4.2.5.1. Bloques** Pasando a la parte más visible (la v dentro del modelo MVC) aunque con ramificaciones en las otras dos, nos encontramos con esta estructura. Los bloques constituyen las unidades más pequeñas para mostrar información. Digamos que son pequeños trozos de código que aportan una funcionalidad que se plasmará visiblemente en la pantalla. Cosas como el número de usuarios conectados, los últimos comentarios en entradas o los artículos más populares son los ejemplos más conocidos de este tipo de objetos, que seguro que son conocidas a la mayoría de usuarios medios por haber visto cosas similares en otros sitios de la red.

Se enmarcan en regiones distribuidas por la estructura gráfica de la página (como encabezado, pie, laterales o cuerpo principal). Serían, pues, los diferentes componentes que podríamos organizar (o quitar) de nuestra página para que mostrase la información que consideramos necesaria. Juegan un papel fundamental a la hora de trabajar con plantillas, que veremos en el apartado de Temas.

**2.4.2.5.2. Menús** No hay mucho que explicar sobre los menús en Drupal salvo que son, como casi todo, altamente personalizables. Serían colecciones de enlaces que normalmente están situados en regiones reconocibles de nuestro sitio web y nos permiten movernos por el sitio sin perdernos y sin dar muchos rodeos. Dependiendo de cómo están construidos pueden incluso darnos una idea de dónde estamos. Por defecto se crea un menú principal que se mostraría en todo momento y un secundario que se mostraría a partir de la opción seleccionada en el menú principal. Su uso, como desde hace mucho tiempo, es muy intuitivo.

**2.4.2.5.3. Taxonomías** Aunque hablaremos más adelante de esto, hay que resaltar que el módulo “Taxonomy” nos proporciona una funcionalidad que en nuestro

caso vendrá a sustituir por ahora lo que en principio se iba a construir a base de ontologías y redes semánticas. Es una implementación muy simple pero en nuestro caso resulta que cumple con nuestras necesidades. Permitiría clasificar los contenidos (incluso haciendo distinciones entre distintos tipos de contenido) según clasificaciones que nosotros previamente hayamos creado, a las que el programa llamará “vocabularios”. Es decir, que en función de la clasificación (o clasificaciones, dado que se pueden aplicar varias a la vez) que le otorguemos a un contenido en concreto, podrá ser posteriormente relacionado en una jerarquía que ordenará los contenidos por su área semántica. Por inmediatez del ejemplo, podremos catalogar los instrumentos de nuestro museo por área de conocimiento a la que pertenecen (astronomía, geofísica, geodesia, ...) y a la vez enmarcarlos en una época temporal, por los materiales de su construcción (pudiendo estar a la vez en varios grupos) o por cualquier otra escala que se nos ocurra.

Esta funcionalidad, combinada con la anterior de los menús, puede ser muy interesante a la hora de ayudar al usuario a navegar con sentido por el sistema.

**2.4.2.5.4. CCK (Content Construction Kit)** Si uno de los puntos fuertes de Drupal es la definición de tipos de contenido a través del concepto de “nodo”, el Kit de construcción de contenido (CCK) nos permite estructurar semántica y lógicamente cada uno de los apartados de nuestro tipo de contenido. En definitiva, proporciona una herramienta para especificar campos concretos (con su definición de tipo de datos) que contengan datos definidos, como en el caso de un registro, que contiene campos. En este caso, podremos elegir el nombre de la variable o campo, el tipo de datos que contendrá y, en su caso, qué interfaz (o widget) se le presentará al usuario a la hora de tener que llenar ese campo. Es un módulo tan esencial que a partir de la versión 7 de Drupal forma parte del núcleo de la plataforma. Digamos que es un paso más hacia la ingeniería de software más clásica.

**2.4.2.5.5. Views** El módulo anterior se complementa a la perfección con otro módulo más: se trata de Views, y como su propio nombre indica, su cometido es asistir a la hora de la creación de vistas personalizadas. Aludiendo estas a muy diversos parámetros que regulan desde quién verá cada vista, pasando por los campos que se mostrarán de cada nodo, parámetros disponibles para realizar búsquedas o la manera que tendremos de mostrar el contenido al usuario.

Se trata de un módulo muy potente y, para hacerse una idea, recoge cierto paralelismo con las “vistas” de una base de datos SQL. De hecho, se puede, incluso, proporcionar código SQL que defina (con la potencia que eso implica) en detalle la consulta que queremos hacer a la hora de ejecutar una vista.

#### 2.4.2.6. Temas

Los temas son la cara visible de Drupal hacia el usuario, son un conjunto de ficheros CSS, HTML/PHP e imágenes que permiten dar forma a la salida por defecto del contenido. Sería lo correspondiente a la capa de visualización o a la parte de vista en el modelo MVC. La noción de “tema” se supone conocida en este ámbito. Brevemente, el paso de una url a html supondría dar los pasos de:

1. El módulo Node recibe la petición (<http://www.dominio.com/node/2>)

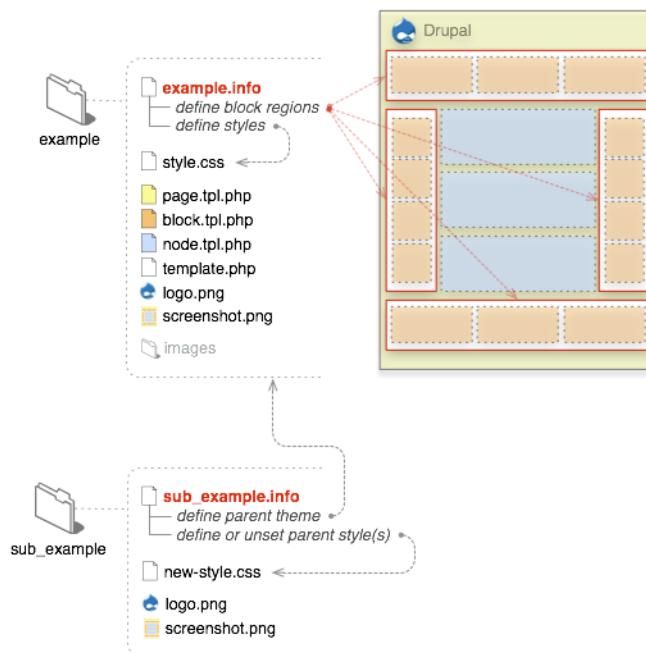


Figura 2.15: Estructura básica de Temas en Drupal [2]

2. Node extrae el contenido del nodo y llama al módulo Theme
3. Theme, a través del motor del tema, estructura el contenido de la salida
4. Se le aplica el aspecto gráfico del tema utilizado en ese momento.

Sin embargo, se podría ilustrar la estructura usada por Drupal a través de un pequeño esquema con el que vemos en la figura 2.15

**2.4.2.6.1. Archivos .info** Este tipo de archivos, ubicados normalmente en la carpeta “themes”, define, obligatoriamente, metadatos del tema en concreto como su nombre, descripción, versiones de Drupal con las que es compatible y el motor de temas que usa. Igualmente, puede incluir la definición de regiones en las que dividiremos nuestro tema, los ficheros CSS y/o JavaScript que lo acompañarán o las funcionalidades que el administrador podrá utilizar en el panel de administración de temas.

**2.4.2.6.2. Regiones** Se trata de las divisiones del espacio gráfico que definirá nuestro tema y en las que se pueden ubicar los bloques. Las definidas por defecto en Drupal son:

- Header (encabezamiento)
- Footer (pie de página)
- Left Sidebar (barra lateral izquierda)
- Right Sidebar (barra lateral derecha)

- Content (sección central dedicada al contenido)

Por supuesto, como todo en Drupal, podemos definir las nuestras propias. Eso sí, en el momento en que creamos nuestro propio fichero .info y definamos una sola región, las regiones por defecto no serán tenidas en cuenta si no se refieren explícitamente.

**2.4.2.6.3. Características** Se trata de todas esas opciones que podrán ser administradas desde el panel de administración de Drupal. Entre ellas encontramos, por defecto, las siguientes:

- Logotipo (logo)
- Nombre del sitio (name)
- Lema (o eslogan) del sitio (slogan)
- Declaración de intenciones (mission)
- Avatar de usuario en entradas (node\_user\_picture)
- Avatar de usuario en comentarios (comment\_user\_picture)
- Cuadro de búsqueda (search)
- Icono de marcadores (favicon)
- Enlaces primarios del menú (primary\_links)
- Enlaces secundarios del menú (secondary\_links)

De la misma manera que ocurría con las regiones, las características también son personalizables aquí.

**2.4.2.6.4. CSS y JavaScript** Como en las otras ocasiones, existen algunas rutas por defecto para incluir nuestro código CSS y JavaScript en nuestro proyecto, pero además podemos redefinir y añadir las referencias de nuestros ficheros adicionales.

**2.4.2.6.5. Archivos de plantilla** Aquí es donde reside la mayoría de la maquetación real de las páginas. Estos archivos, con abundancia de código HTML y algunas sentencias PHP desvelan la estructura que presentarán a la salida cada uno de los elementos de contenido. La extensión de los archivos es del tipo <elemento>.tpl.php, como por ejemplo page.tpl.php o node.tpl.php. Dentro de ellos, las sentencias de php interrogan a la base de datos para saber si se han completado las regiones correspondientes para llenarlas de contenido y darles una forma.

**2.4.2.6.6. template.php** Esta sería la piedra angular de todo programador, ya que se utiliza para añadir lógica y variables a la aplicación y para redefinir funciones que son llamadas por los archivos de plantilla. Se trata, casi exclusivamente, de código php por lo que deberían ser editados por programadores o diseñadores con conocimientos de este lenguaje.

## 2.5. Conclusiones y trabajo futuro

Las conclusiones en este trabajo son muy variadas y pueden verse desde distintas perspectivas.

### 2.5.1. Logros alcanzados

Básicamente, la consecución de este proyecto ha conseguido preparar la creación de una plataforma de gestión de contenidos con el objetivo de ampliar y enriquecer la experiencia de la visita a un museo. Dado que no se trata de un museo al uso si no de una exposición de instrumentos de una rama muy específica del conocimiento (muy atractiva al público general pero poco conocida entre la gente) que requiere, salvo en contadísimas excepciones, de la explicación pertinente por parte de un guía.

La consecución de este museo virtual soluciona la obligatoriedad de la presencia de ese guía y de ese usuario en el mismo espacio y tiempo permitiendo ese acercamiento de manera más útil para todos.

Con este proyecto, además, se pretende obtener una base de conocimiento que permita disponer del mismo y trabajar con él de manera que en un futuro esté accesible para otros proyectos como la edición de contenido en papel a partir de lo alojado en la base de datos.

Además, permite una cierta interacción con el usuario, incluyéndole, aunque sea de en la esfera más privada y con pequeñas parcelas de participación en la experiencia de formación que quizá no todo el mundo pueda experimentar al visitar el museo por diversas razones.

En el plano más técnico, se ha conseguido realizar una comparativa adecuada a las necesidades del cliente de los distintos CMS (y algunos wikis) disponibles en el mercado, poniendo de manifiesto que se trata de una parcela de la informática con muchísima actividad e interés tanto comercial como de utilidad.

Como último y evidente punto podemos concluir que se han sentado las bases para adaptar con éxito una plataforma de código abierto como es Drupal a unas necesidades concretas que eran demandadas por una nueva perspectiva en el cliente.

### 2.5.2. Conocimientos adquiridos

#### 2.5.2.1. Generalidades

Personalmente, este proyecto me ha dado la oportunidad de comprobar y entender un poco más lo compleja (aunque no imposible, como se podría creer) que resulta la ingeniería del software y, en general, la ingeniería informática. Compleja no sólo en lo imbricado que pude llegar a ser un sistema de información, si no más bien en, como para casi todo, la parte más impredecible y más variable: el cliente. En mi caso el cliente no es tal ya que más que cliente es “empleador” a través de una beca. He comprobado (al igual que cualquiera que lo lea lo hará) lo que cambia un proyecto desde sus fases más tempranas hasta su ejecución. He visto la cantidad de disciplina (yo diría que más que en otros trabajos aunque menos, quizás, que en otras ingenierías) que requiere un proyecto de este tipo. En parte por la impredecibilidad de ciertos

acontecimientos que modifican el rumbo y el cometido del proyecto y que alargan los plazos a través del descarte y posterior recomienzo desde cero si los términos no han sido los esperados.

Son muchas partes a tener en cuenta que, ya no tan sorprendentemente (pues nos lo habían avisado con insistencia), tienen no demasiado que ver con la parte más técnica de un programa informático. Como por ejemplo, la fase de evaluación (en mi caso) de soluciones preliminares. Imagino que sería difícil para un analista no ingeniero tratar de evaluar innumerables plataformas con la intención de elegir alguna, pero, desde luego, no es la más grata de las labores que se me ocurre podría desempeñar un ingeniero.

Una de esas partes, como decía al comienzo, es el cliente. Creo que la parte más importante de un proyecto (hasta donde yo he experimentado) es la comunicación con el cliente. Ya no sólo en el proceso de, más o menos pasivamente, recolectar requisitos y ayudar a traducir esas ideas a un metalenguaje no tan ambiguo como el humano. Incluso, me atrevería a decir, en los proyectos de investigación en los que la figura del cliente está más desdibujada, una buena manera de entenderse a uno mismo (o a un equipo) en sus objetivos es seguir el mismo proceso que tomariamos para “diagnosticar” a un supuesto cliente.

### **2.5.2.2. Particularidades**

Este proyecto, como supongo que lo harán todos, presenta un carácter y un contexto propios. Se trata de un proyecto emarcado dentro de una beca de formación para un objetivo no crítico y en un entorno especial como es la rama de la investigación científica que va desde la ciencia de base (como la astronomía y la geofísica) a la más cercana a las ingenierías, que sería la instrumentación, que es en lo que han confluido buena parte de los esfuerzos de la institución del Real Observatorio de Madrid. Dentro del Instituto Geográfico Nacional, que, como ya he dicho en otras ocasiones, depende directamente del Ministerio de Fomento, se encuentran diversas ramas. El ROM se inscribe dentro de la Subdirección General de Astronomía, Geodesia y Geofísica, que gestiona otros observatorios astronómicos como el OAN con su sede en Yebes (Guadalajara), el de Pico Veleta (Granada) o el del Roque de los Muchachos (Canarias). Sin embargo, lo expuesto en el museo recorre todos los ámbitos más o menos diferenciados que cuelgan del IGN: Astronomía, Geodesia, Geofísica, Geomagnetismo, Sismología, Vulcanología, Cartografía y, además, una breve sección de instrumentos de una rama que ya no forma parte del IGN, que es la Meteorología. Esto lo cuento para dar una idea de la transversal del proyecto y de la cantidad de gente en cuanto a la creación de contenidos, además de poner de manifiesto la ausencia de personal especializado en Informática (salvo, de refilón, el caso de algunos Ingenieros de Telecomunicaciones que hábilmente han trabajado en construir los sistemas de control y telecontrol de instrumentos como radiotelescopios, gravímetros superconductores y demás tecnología punta). Lo que implica esta marcada ausencia es quizá una falta de rumbo claro a la hora de optar por unas u otras soluciones.

Por otro lado, como ya hemos comentado está la ausencia de una criticidad clara del proyecto. Es decir, que no está en la agenda más inmediata de la mayoría de participantes. Todo esto, unido a la ya clásica lentitud de los procesos que dependen de las administraciones públicas (no sólo en materia de adjudicación de fondos o contratación de personal) acentúan mucho más lo que comentaba en las generalidades a

propósito de la variabilidad de un proyecto y de la complejidad del trato con el factor humano en el cliente.

Por otra parte, se trata de un proyecto que me presenta de brúces con el público dado que la consecuencia de mi trabajo estará disponible al público y deberá dar una respuesta fiable (dentro de unos márgenes temporales y de actuación) a trabajadores expertos en su campo. Sin duda es un proyecto ilusionante.

### **2.5.2.3. Modificaciones**

Como en todo proyecto, como ya dije, pocas cosas permanecen invariables cuando pasa el tiempo. En mi caso, lo más invariable del mismo quizás ha sido el espíritu de iniciativa que, sin quizás una definición cristalina, ha conseguido hacerse un hueco y prosperar poco a poco. Dicho de otra manera: un proyecto que comenzó ambicioso en proyección no ha cesado en esa proyección aunque haya tenido que ajustar los resultados más a corto plazo. Esto quiere decir que, si bien no formarán parte de este trabajo de fin de carrera, surgirán unos cuantos proyectos que terminen (o no) de definir y construir las potenciales utilidades del inicial.

En general, creo que la aproximación más como técnico que como ingeniero hace que inicialmente podamos cometer este tipo de errores, que sin derrumbar el pragmatismo que es el objetivo, tratan de hacer “virguerías”. Supongo que ese es un campo destinado a la investigación y no a la aplicación de tecnologías a problemas concretos.

**2.5.2.3.1. Web Semántica** En el plano de las convicciones más íntimas, he tenido que “recular” en propuestas, algunas por iniciativa propia y otras por exigencias del proyecto. Una de ellas pasaba por la inclusión intensiva de tecnologías de Web Semántica que permitieran catalogar los contenidos del museo con “mayor” eficacia y modularidad. Tras continuar mi formación (de varias asignaturas) con otro curso más en estos temas he observado varias cosas:

1. Que se trata de una tecnología que por ahora está, digamos, no muy madura. Ni por la globalidad de Internet, ni por sus bases teóricas (que no acaban de cuajar en todos los agentes) y no sé sabe si seguirá adelante, pues para resolver un problema muy grande (la extracción de conocimiento útil a partir de cantidades ingentes de datos) ha implementado soluciones casi tan complicadas de resolver: principalmente las ontologías (estandarización, compartición, herencia, etc.), la utilidad de los agentes y ¿por qué no? la duda sobre si realmente un agente es un concepto útil para organizar operaciones (aunque es cierto que han corrido mejor suerte que las primeras). Dejo de lado la parte de más éxito como los buscadores que, de forma poco matemática, han utilizado algoritmos de “detección” de conocimiento en vez de esa catalogación ontológica que se proponía, echando un poco más de peso en las espaldas de los grandes defensores del “etiquetado” y catalogación minuciosa de la información.
2. Que supondría una carga de trabajo demasiado exigente para los resultados que obtendría.
3. Que no aportaría gran cosa que otras tecnologías sin más esfuerzos ya obtienen (no hay que reinventar la rueda).

**2.5.2.3.2. Uso de estándares** Esta en realidad no es un renuncia, si no que, en aras de la utilidad de sus componentes, se han relajados ciertos aspectos de la “afición” por XML. Como el lector ya habrá adivinado, esto tiene mucho que ver con la parte de Web Semántica, dado que la mayoría de tecnologías implicadas (RDF, RDFS, OWL) son estándares de la familia XML. Es en esa parte en la que más se ha renunciado, dejando la parte XML más universal (XHTML, XSLT) para la presentación final. Drupal, en general, cumple con estos estándares siendo XHTML utilizado para mostrar el contenido al cliente y espero poder utilizar XSLT para la transformación en PDF de los documentos contenidos en la plataforma.

Sin embargo, y como reflexión, creo que ha cambiado mi manera de percibir la importancia del uso de estándares. Como en otras ocasiones (y de eso trata mucho de este proyecto) se trata de evaluación. Los estándares son útiles en la medida que nos ayudan a proporcionar un servicio o una ventaja a nuestros sistemas. Seguir un estándar como hecho en sí mismo a perseguir no tiene mucho sentido.

**2.5.2.3.3. Wikis** La otra gran modificación que ha sufrido este proyecto (que incluso está entre las palabras clave del resumen) es el descartar, al menos por ahora, la posibilidad de utilizar un wiki en exclusiva para una plataforma tan importante como ésta. Como digo, no se trata de una decisión arbitraria, si no de una decisión necesaria para asegurar la integridad y coherencia de los datos de carácter oficial.

Pese a todo, y si se atiende con atención el proyecto, la apertura a nuevas modificaciones por parte de personal cualificado sigue siendo una verdad, con lo que, mínimamente, se persigue la estela de la filosofía del wiki conservando el tinte colaborativo del asunto.

### 2.5.3. Tiempo empleado

Este proyecto se ha venido realizando desde su primera reunión a principios de mayo hasta principios de diciembre con intermitencias, retrasos y demás contingencias que han hecho que el tiempo no fuese aprovechado con el máximo rendimiento.

Compartiendo el tiempo con otros proyectos englobados dentro de la beca convocada, la jornada es de unas siete horas y media durante 5 días a la semana, contando con las vacaciones de verano entre medias, con lo que el esfuerzo real dedicado al proyecto disminuye respecto de la jornada total.

La parte más extensa fue la de reuniones con diferentes profesionales y responsables, las revisiones de estos requisitos, la elaboración de propuestas y la evaluación de estas propuestas por parte del cliente, siendo parte importante de todo esto el tiempo empleado en dilucidar cuál sería la plataforma más adecuada para este trabajo.

La parte de desarrollo todavía continúa por haber, como ya he explicado a lo largo del documento, discrepancias internas sobre la estructura que deberían seguir algunos aspectos del diseño.

### 2.5.4. Líneas futuras de trabajo

Aunque creo que lo he repetido varias veces a lo largo de este documento, voy a tratar de exponerlo aquí resumidamente.

Este proyecto, como evaluación, elección y análisis está terminado, quedando a falta de detalles menores de implementación que dependerán como ya he explicado de la solución que se tome en cuanto a hardware. Además, el trabajo puramente de diseño, sería la parte más variable y más propensa a cambios y/u optimizaciones. Esa es una de las razones por las que se presentan un número tan bajo de capturas de pantalla en este documento y no se entrega un ejecutable de la misma (la otra sería que el contenido no está todavía incluido).

Siguiendo en la dirección del hardware, las posteriores fases pasan por la adquisición de terminales móviles y terminales fijos en sala, así como los componentes que definan la infraestructura de red necesaria para que cumplan su cometido.

De la parte del software habría que ultimar los detalles de plantillas para móviles y las medidas de seguridad necesarias en la implantación de los sistemas.

Como última idea y a futuro, quedaría la integración de cursos de “eLearning”, la integración con la tienda del museo y la consolidación de la base de datos respecto a los estándares de la administración pública.

En definitiva, quedando fuera del cometido de este proyecto, hay unas cuantas actividades con las que proseguiré de aquí en adelante para completar la aplicación. Las mismas serán incorporadas a medida que la maquinaria de toma de decisiones y financiación actúen en consecuencia.

## **Capítulo 3**

### **Planos**



### 3.1. Esquemáticos

En esta sección, voy a tratar de encajar algunos de los datos más técnicos del proyecto, sobre todo los referentes a la estructura del sistema y los flujos de información representados.

Entre ellos, tengo un prototipo de modelo de datos, esquemas sobre el funcionamiento de Drupal, y una idea de los flujos de información de la visita.

#### 3.1.1. Modelo de datos

Hemos dejado bien claro que el propósito de este trabajo no es desarrollar de cero un CMS si no escoger y adaptar uno ya existente a las necesidades que se nos han presentado. Antes de tomar una decisión (que es independiente del autor y tomará tiempo), es conveniente señalar que sí que debería existir una parte de “diseño original” que correspondería al modelado de algunas informaciones.

##### 3.1.1.1. Catálogo

Respondiendo a los requisitos enunciados por miembros de la empresa, he elaborado un diseño de modelo de datos que podría utilizarse para abstraer los contenidos de un catálogo de objetos expuestos (subconjunto de ítems informativos del total del museo virtual) y no expuestos. Para ello, me he ayudado de un catálogo editado en papel por el propio Ministerio de Fomento titulado “Instrumentos históricos del Instituto Geográfico Nacional”. Debido a su creación artesanal cada instrumento no sigue la debida canonización que nos piden las bases de datos, y es por eso que he tratado de ordenarlo en un modelo de datos como el que se muestra:

###### 3.1.1.1.1. OBJETO

- id/ref: número entero
- nombre: caracteres(30)
- firmado: (inventor, lugar\_origen, fecha\_origen) (se eliminaría del original)
- inventor: relación(n:n)
- lugar\_origen: relación(n:n)
- fecha\_origen: fecha
- dimensiones: caracteres(12)
- características\_técnicas: texto
- materiales: caracteres(20)
- depósito\_habitual: relación(1:n)
- observaciones: texto

- área: relación(1:1)
- tipo de objeto: relación(1:n)
- foto: caracteres(50)
- depósito actual: caracteres(50)

### **3.1.1.1.2. ÁREA**

- id: numérico
- nom\_area: caracteres(20) Astronomía esférica, Geodesia geométrica, Topografía planisférica y Fotogrametría terrestre. Topografía altimétrica, Gabinete, Meteorología, Geofísica

### **3.1.1.1.3. TIPO\_OBJETO**

- id: numérico
- nom\_tipo: caracteres(40)
- id\_area: relación(n:1)

### **3.1.1.1.4. ARTÍCULOS**

- id: numérico
- titulo: caracteres(25)
- contenido: texto

### **3.1.1.1.5. AREA\_ARTÍCULO**

- id\_area: relación(1:n)
- id\_articulo: relación(n:1)

### 3.1.1.1.6. INVENTOR

- id\_inventor: numérico
- nom\_inventor: caracteres(25)

Este modelo será reutilizado (si se aprueba) para crear una interfaz sencilla de entrada de datos para el personal interno.

### 3.1.1.2. Revisión del catálogo

En reuniones posteriores, el cliente indicó que estaban desarrollando un catálogo de instrumentos en papel paralelamente a este proyecto. Este catálogo vendría a dar unas pautas sobre los campos y de algún modo la presentación deseable para, además, el museo virtual.

Cabe destacar que el modelo nuevo es obviamente más simplificado, aunque algunas voces dentro de los mandos decisivos del cliente todavía dudan sobre si debería llegar a un tercer modelo intermedio o no.

La idea es tratar de reducir la complejidad, aumentar la integridad y la unicidad de los datos y conseguir, a la vez, una buena coherencia semántica entre los diferentes objetos, artículos explicativos y clasificaciones.

En el nuevo modelo, el sugerido para el catálogo en papel, tendríamos los siguientes componentes principales.

**3.1.1.2.1. Ficha de objeto** En una página del catálogo, contamos con los siguientes elementos, expuestos, sucintamente en la figura 3.1. He añadido una estimación sobre el tipo de datos que podría asimilarse a la estructura propuesta.

- Título: (Cadena de caracteres).
- Fecha: (Fecha o alguna modificación, dado que algunas son aproximadas).
- Constructor: (Cadena de caracteres, posiblemente relacionado con una tabla de autores).
- Lugar de origen: (Cadena de caracteres).
- Características técnicas: (Campo de texto).
- Observaciones: (Campo de texto).
- Foto: (Cadena de caracteres dirigida a la ruta del archivo de imagen, o del nodo con las fotos a distintos tamaños).
- Dimensiones: (Tripleta de enteros, aunque, en este caso, por su nula utilidad, puede ser almacenado como cadena).
- Materiales: (Cadena de caracteres, relacionado con una tabla de materiales).

- Lugar actual: (Cadena de caracteres, relacionado con una tabla de lugares, dado que son muy pocos posibles).
- Referencia: (Entero, aunque se está dudando si utilizarlo como identificador único).

### **3.1.2. Esquema general de servicio**

En el esquema que se puede ver en la figura 3.2 se describe muy sencillamente la base de funcionamiento del servicio de museo virtual. Específico que se representa el servicio de museo virtual dado que la parte de generación de contenidos no está presente, dado que ocupará la sección siguiente.

#### **3.1.2.1. Servidor del museo**

Como ya se ha expresado anteriormente, la idea es tener un único servidor en un principio. Es casi inmediato ver la necesidad de contar con un servidor de pruebas. Además, en un mundo ideal, sería deseable que hubiese una separación entre el servidor de base de datos (con buena parte del contenido textual), el servidor de contenidos multimedia, el servidor de backup y el servidor web propiamente dicho.

#### **3.1.2.2. Clientes a través de Internet**

La idea es ofrecer a los potenciales o satisfechos visitantes una ampliación a los contenidos vistos en el visita guiada. El porqué de esta decisión es, además de la falta de recursos que requeriría abrir permanentemente al público, que los contenidos de este museo requieren, para ser disfrutados, de una explicación, ya sea textual, audiovisual o hablada, en persona, que es realmente donde más se explota esta posibilidad. Para paliar la falta de tiempo disponible en esta visita, se crea esta alternativa virtual, aunque no se descartan aproximaciones diferentes en un futuro.

Para todo esto es necesario que la página esté accesible (tras routers, firewalls, etc.) al gran público a través de Internet y en horario ininterrumpido. Los flujos de comunicación deben ser, en el caso de haber encuestas, cursos o foros, bidireccionales.

#### **3.1.2.3. Puesto en sala**

Al ser un proyecto ambicioso pero de larga duración, el detalle del número de puestos aún no ha sido delimitado. Aunque pueda parecer una contradicción con las visitas guiadas de público general. Sí que es cierto que pueden proporcionar una herramienta puntual muy útil a la hora de atender a grupos específicos de nivel más avanzado y con más tiempo para detenerse e indagar en la historia de los instrumentos de la colección.

Los puestos en sala deberían exclusivamente limitarse a mostrar los contenidos alojados en el servidor, con la salvedad de posibles encuestas, comentarios o sugerencias que saliesen a la luz mientras se usan.

Geodes

**Teodolito geodésico  
de segundo orden**  
c. 1850

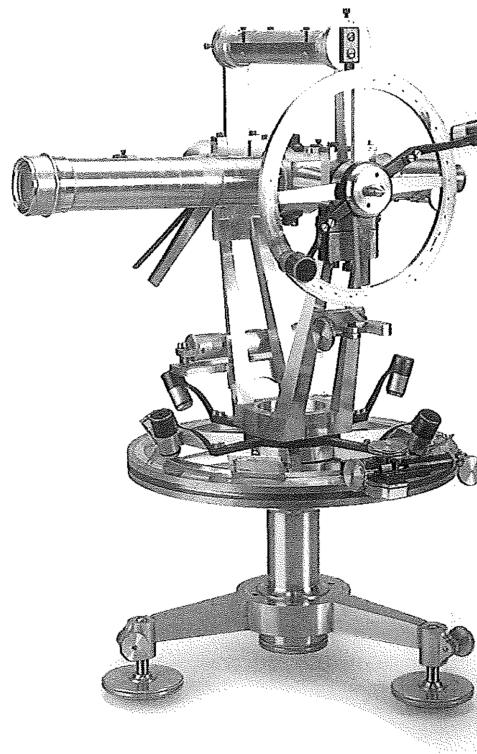
*Ertel & Sohn* in München. Alemania

**CARACTERÍSTICAS TÉCNICAS**

Anteojo astronómico de 35,5 cm de distancia focal y 3 cm de apertura, con cruz filar en forma de «X» complementada con un hilo vertical; limbo azimutal y cenital sexagesimales de 25 y 16 cm de diámetro respectivamente, numerados de  $10^\circ$  en  $10^\circ$  y divididos cada  $10'$ , con cuatro y dos vernier respectivamente divididos en 60 partes, lo que permite apreciar  $10''$  reiterando la medida; dispone nivel caballero de 12,5 cm y otro paralelo al anteojo de 9 cm.

**OBSERVACIONES**

Este instrumento es semejante a los descritos anteriormente en cuanto a la precisión (capaz de apreciar  $10''$ ); no obstante, el presente ejemplar pertenece a una época anterior, debiendo formar parte de un conjunto de piezas adquiridas en 1855 por la Comisión de la Carta Geográfica, con la que ésta comenzó su andadura. En los primeros inventarios de la Comisión figura la adquisición en la casa Ertel de varios teodolitos de este tipo. Con estos aparatos debieron realizarse los primeros trabajos geodésicos, como se ha señalado en la introducción de este catálogo.



<b>DIMENSIONES</b> $48 \times 36 \times 30$ cm.
<b>MATERIALES</b> Latón, acero, vidrio.
Madrid. Instituto Geográfico Nacional. Ref.: 575.

Instituto Geográfico

Figura 3.1: Vista preliminar de una página del catálogo en papel

#### 3.1.2.4. Terminales móviles

Quizá la parte más novedosa y además, en boga. Estos terminales (tanto privados como alquilados) supondrían una pequeña puerta abierta a visitas más libres y,

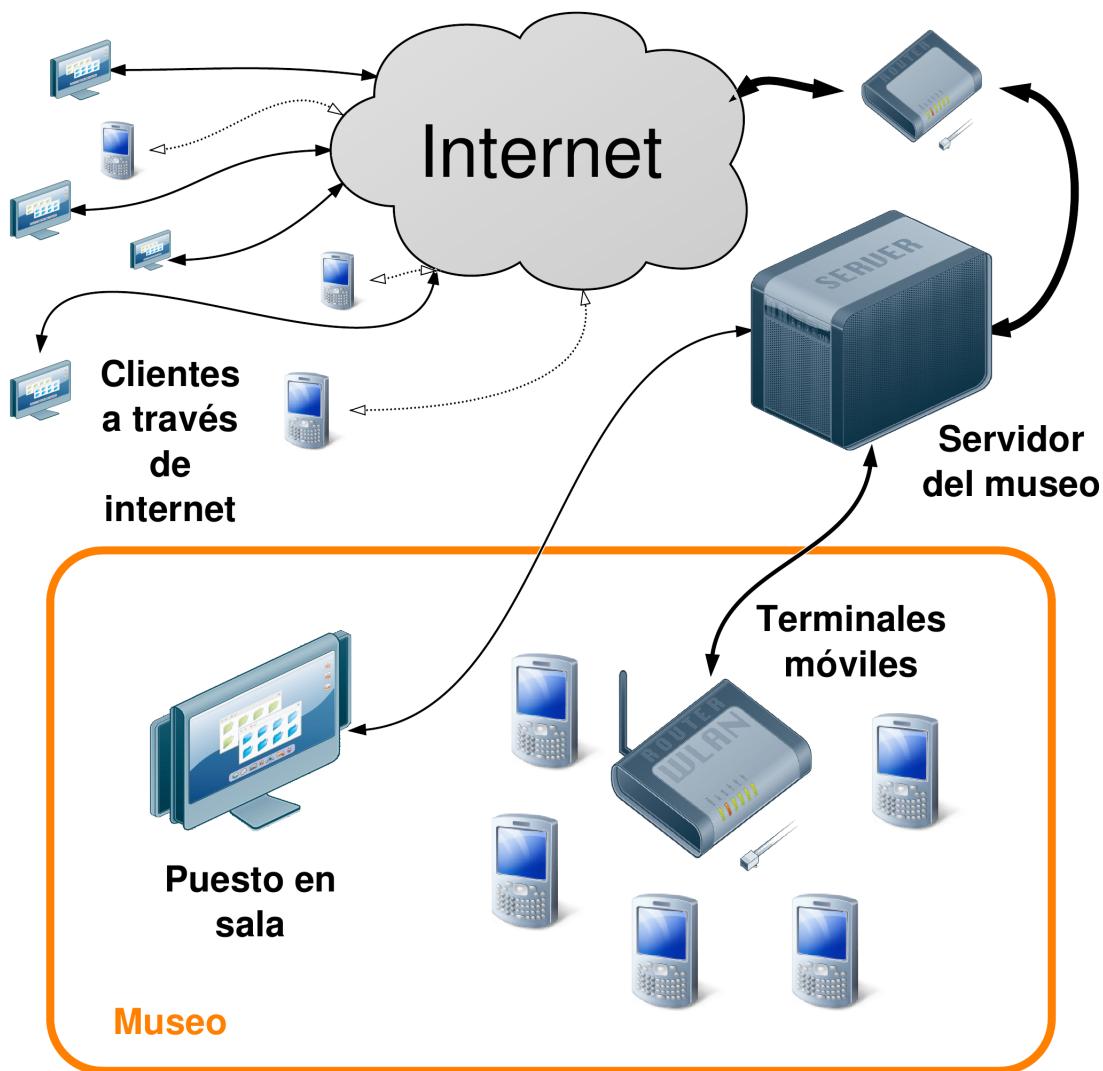


Figura 3.2: Esquema general de servicio

además, a visitas plurilingües. Sus capacidades de portabilidad, versatilidad e inmediatez a la hora de conseguir contenidos las convierten en un componente idóneo. En el caso del museo, que sería el área de cobertura principal de las mismas, también deberían acceder exclusivamente a contenidos del servidor, evitando cualquier otro camino a tomar dentro de la red. Su capacidad interactiva, por dificultades obvias de comodidad y espacio, podría verse limitada tanto en la versión inalámbrica dentro del museo como en la versión en línea de la misma (que también debe estar disponible al gran público).

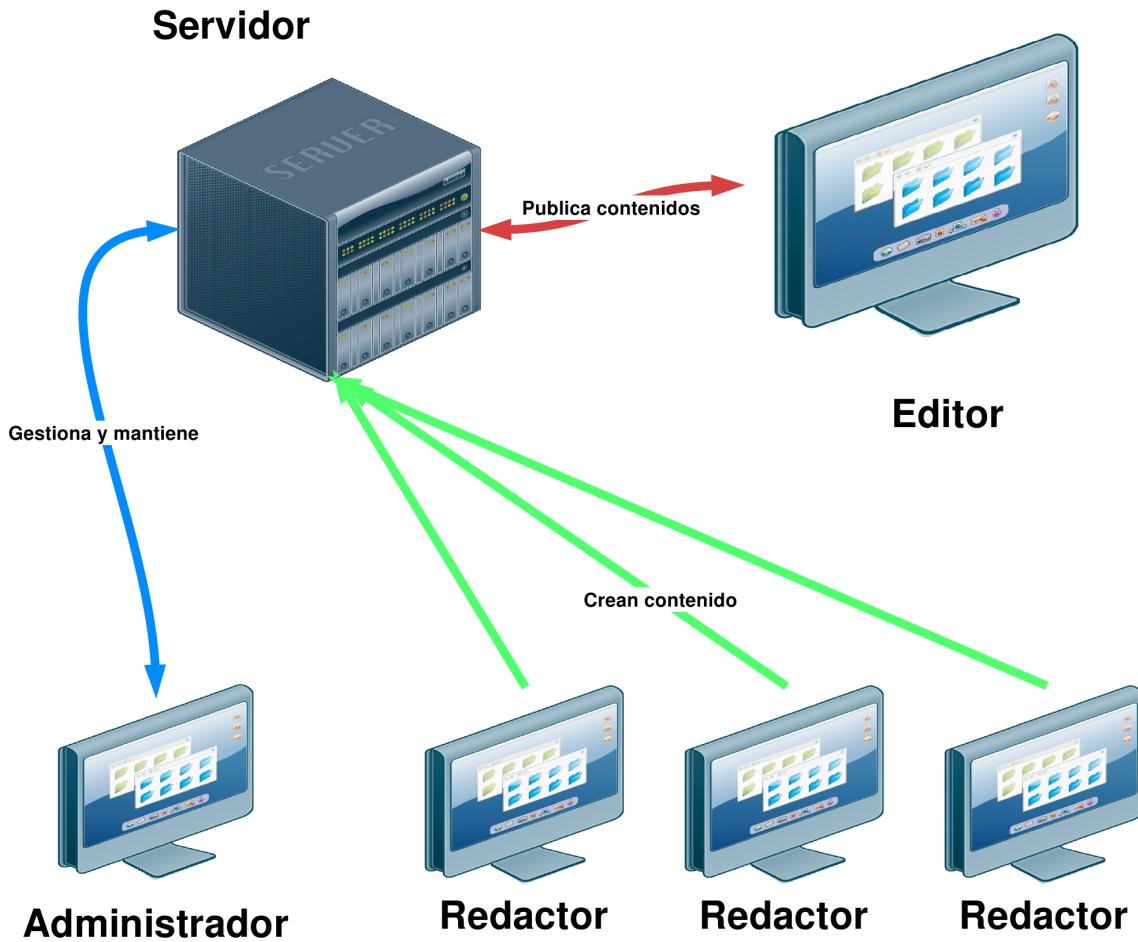


Figura 3.3: Esquema general de creación de contenidos

### 3.1.3. Esquema general de creación de contenidos

Esta parte quiere detallar el proceso de edición y publicación de contenidos que se seguiría, en principio en la fase de carga, en caso de que se definiera un esquema colaborativo (actualmente esto está sin determinar). En cualquier caso, la infraestructura estaría preparada para eso; es una característica contemplada en la baremación y la mayoría de sistemas gestores de contenido son válidos para esta cuestión. De hecho, en la sección 2.4.2.3.

Siguiendo el esquema propuesto en la figura 3.3, pasamos a detallar el rol de cada uno de los participantes.

### **3.1.3.1. Servidor**

Poco que añadir a lo dicho anteriormente salvo reiterar la deseable división de servidores. Una cosa importante sería el decir que la labor de publicación de contenidos recaería directamente sobre el servidor en producción si el esquema propuesto es el colaborativo dado que ese sería un servicio a prestar; o siguiendo el ciclo que describí en la sección 4.3.2 si se trata de un esquema estático y unidireccional con una fase de carga de datos única y no cíclica.

### **3.1.3.2. Administrador**

Se trataría del mantenedor de la aplicación, siguiendo el mismo ciclo al que hemos hecho referencia en el párrafo anterior a la hora de implementar cambios que amplíen o modifiquen las funcionalidades requeridas en cada nueva etapa del proyecto. Además, tendrá que prevenir y reaccionar frente a los fallos que puedan ocurrir, tratando de mantener su sistema lo más estable y seguro posible. Incluso, dado la naturaleza inusual del proyecto, deberá dedicarse a optimizar el rendimiento del servidor con las ideas expuestas en el epígrafe 4.1.

### **3.1.3.3. Editor**

"Editor" se utiliza, en este caso en sustitución de otra designación como podría ser supervisor o coordinador. Normalmente será ocupado por un responsable designado para la tarea que, principalmente y aunque pueda asumir otros roles, como editor, tendrá que revisar los contenidos y dar el visto bueno a cada nueva publicación o modificación. El sistema elegido ya presenta la posibilidad de restringir de esta manera la, llamémosla así, "versión oficial" de los contenidos. Quiero incidir en este tema porque es importante que los contenidos presentados por esta plataforma sean fiables, ya que se considera una referencia a nivel nacional (y también de cara a la comunidad internacional) la calidad y fiabilidad de sus textos.

### **3.1.4. Redactor**

Como ya hemos venido sugiriendo en los párrafos anteriores, lo que estamos explicando aquí es la parte de "roles" que han de asumir los efectivos de los que dispondrá la organización para este proyecto. Esto no supone que deban ser específicamente personas con únicamente ese trabajo. Lo más normal (repito: si se sigue este esquema) es que este rol sea asumido por los profesionales que ya trabajan en el Observatorio, es decir: astrónomos, geofísico, vulcanólogos, cartógrafos, etc.

Su rol les permitirá, ya hablando meramente del sistema informático, crear y proponer modificaciones en los contenidos, reservando, como ya hemos dicho, la última palabra y la responsabilidad sobre su publicación definitiva al editor.

## **Capítulo 4**

# **Pliego de condiciones**



## 4.1. Consideraciones previas

En esta sección trataré de exponer la problemática asociada a la asignación razonada de requisitos. Hemos dejado claro que nuestra apuesta será Drupal. En concreto, la versión de la rama 6.x, que está debidamente probada y se supone estable. He buscado información sobre los requerimientos asociados a ejecutar Drupal y esta investigación creo que se podría dividir en dos partes, aunque están interconectadas y no deben ser aisladas del todo a la hora de tomar en consideración: las necesidades de rendimiento y las particularidades del software elegido.

### 4.1.1. Rendimiento necesario

¿Qué quiero expresar con esto? Como es bien sabido en ingeniería, a menudo (por no decir constantemente), nos encontramos en disyuntivas que conducen al compromiso entre características y costes. Este caso no es distinto. En concreto, necesitamos definir qué necesidades tenemos y ver hasta dónde podemos llegar en materia de costes.

En nuestro caso podríamos decir que tenemos, como referencia previa, la actual página dedicada a museo virtual en la página del IGN [15], pero, como ya hemos hablado anteriormente, existen numerosas diferencias entre los sitios que deberían ser consideradas, a saber: la casi nula visibilidad de la página actual, la inexistente publicidad de la misma, el acceso exclusivo desde internet y la no asociación a ningún fenómeno “vivo”.

#### 4.1.1.1. Afluencia

En el caso del sitio que intentamos crear, tenemos un fenómeno asociado que es la apertura (por primera vez en su historia) del recinto del ROM al público general, hasta ahora, en forma de visitas guiadas que pasan de manera no exhaustiva sobre algunos de los objetos referidos en nuestro museo virtual. Al tener una gestión más cercana y ágil, la idea es que los dos fenómenos (visitas físicas y visitas virtuales) evolucionen a la vez apoyándose y refiriéndose el uno al otro. Dado este hecho, podemos suponer que la afluencia virtual será más alta que la actual (desconocida) aunque desgraciadamente es muy difícil tener una estimación decente sobre este aspecto. Si nos basamos en el cálculo de visitas físicas actual y suponemos un “worst-case scenario” en el que todos los grupos de visitas guiadas están llenos tendríamos 25 personas por grupo y 8 vistas por semana. Lo que nos daría 200 vistas seguras a la semana, suponiendo que todos los visitantes repitan virtualmente (cosa harto difícil dado el gran número de asistentes que pertenecen a grupos poco familiarizados con las nuevas tecnologías) y que lo hagan una vez (cosa que puede variar). Por esta métrica, tendríamos una afluencia baja, que podría variar a lo largo del tiempo pero no se estima que fuesen cambios muy bruscos o muy significativos.

#### 4.1.1.2. Caudal

Para nuestro museo, si bien no tendremos una afluencia muy elevada, sí es verdad que serviremos, además de texto, contenido multimedia como imágenes (fundamen-

talmente), vídeos y audios, lo que podría suponer un tráfico más elevado y una carga más alta al servidor. Yo anotaría como posibles recursos a proteger un ancho de banda razonable y una capacidad de almacenamiento nada restringida (aunque sin llegar a cifras especiales). La potencia de CPU no veo que vaya a estar afectada mucho por estos fenómenos dada la naturaleza de los mismos si está bien implementada.

#### **4.1.1.3. Criticidad**

Añadido casi de manera anecdótica, podríamos decir que la criticidad de este sistema es muy baja, siendo, quizás, más importante la integridad y custodia de los datos que la disponibilidad total de los mismos. Es decir, si nuestro servidor se cae eventualmente o funciona más lento en picos de tráfico (cosa harto difícil) no supondrá un perjuicio apreciable para la empresa.

### **4.1.2. Particularidades del software elegido**

Como toda solución tecnológica, la plataforma elegida (Drupal) tiene asociados unos requisitos, que veremos más adelante, y unos desempeños más o menos perfilados por los componentes que integra y/o necesita para funcionar. En cuestiones de rendimiento hay mucho escrito sobre Drupal. Como no es el propósito de este trabajo analizar el rendimiento de manera escrupulosa (dado que estamos hablando de una máquina no crítica), voy a dar una visión general sobre el tema.

Para ello, tras hacer unas búsquedas, encontré, además de otras fuentes más específicas [16] [17], un artículo [18] entre la documentación de Drupal disponible en su sitio web. En él se expone de manera clara y concisa la particularidad del sistema. A continuación, glosaré algunas de las ideas allí expuestas (original en inglés).

Además de los apartados anteriores, este artículo pone énfasis en la forma en la que está construida la pila de procesos de Drupal. Una buena consideración siempre que nos referimos a configuraciones hardware y software es la prudencia. Prudencia en la toma de decisiones y prudencia en los cambios.

#### **4.1.2.1. Rendimiento de la pila LAMP**

LAMP (Linux + Apache + MySQL + PHP) es la pila de procesos sobre la que correrá Drupal. Si bien puede cambiarse, por ejemplo, a Windows, o a otros servidores dentro de Linux, nos ceñiremos a este sistema primeramente por existir más documentación sobre él y también por la falta de fondos para pagar y mantener licencias.

Se nos cuenta, citando un tercer estudio, que Apache tiene limitaciones en el ancho de banda, PHP está limitado por la CPU y MySQL se ve enmarcado entre la capacidad de la memoria y el rendimiento de los discos duros. Por ello se recomienda que, al menos en la versión final, estos elementos sean compilados en la máquina de destino.

```
top  
ps -aux  
netstat -anp | sort -u
```

Cuadro 4.1: Comandos útiles para buscar cuellos de botella

#### 4.1.2.2. Cuellos de botella

Debemos analizar los cuellos de botella particulares a nuestra instalación (CPU, memoria, ancho de banda y rendimiento de entrada/salida). Para ello, se nos sugiere utilizar una serie de herramientas que podemos ver en el cuadro 4.1 y, en el caso de encontrar problemas significativos, es recomendable, entre otras cosas, pensar en los servicios adicionales que están disponibles en ese servidor y desactivar o trasladar los que sean posibles.

#### 4.1.2.3. Invitados indeseables

Deberemos evitar invitados indeseables que consumen los recursos de nuestra máquina como por ejemplo: “crawlers”, “spiders”, indexadores, agregadores, emisores de spam, “malware”, etc. Podemos paliar esta degradación añadiendo retrasos cuando se detecten esos agentes, vetando algunas partes del sitio web o remitiendo a las versiones almacenadas en caché.

#### 4.1.2.4. Rendimiento de Apache

Como hemos dicho, hay que tener cuidado con las restricciones que se suelen imponer sobre el caudal asignado al servidor. Como en este caso, estará en nuestro poder, habrá que configurar con especial cuidado, entre otras, las opciones: MaxSpa-reServers, ServerLimit y MaxClients. Además, existen, según la versión, en el propio sitio de Apache así como en otros lugares de internet, guías para optimizar el rendimiento del servidor [19].

#### 4.1.2.5. Ajustes de PHP

El punto débil, como ya he dicho, de PHP es su consumo de CPU. Para solucionar esto existen diversas optimizaciones en el mercado que habrá que analizar con atención: Alternate PHP Cache, Zend Optimizer [20] o eAccelerator entre otras. Además, para profundizar en el tema, podríamos referirnos al libro “Optimizing PHP” [21].

#### 4.1.2.6. Ajustes de rendimiento de MySQL

Para este caso, por lo extenso del tema, podemos remitirnos a un artículo de la propia Drupal [22], que sin duda excede el alcance de este apartado.

#### **4.1.2.7. Ancho de banda**

Como ya he comentado, las imágenes y los vídeos pueden suponer una reducción drástica del rendimiento a la hora de ser servidos. Lo ideal es proporcionar acceso directo a ellos, eliminando, en la medida de lo posible, los pasos intermedios y los procesos “al vuelo”.

#### **4.1.2.8. Consumo de recursos de Drupal**

Trataremos, en todo momento, de vigilar cómo se usa la memoria, qué módulos son los más lentos, qué consultas a la base de datos tardan más y cuáles de ellas serían innecesarias por duplicidad o por ser estáticas. Las soluciones pasarán por implantar módulos de caché de consultas y contenidos, desactivar, modificar o reemplazar los módulos más problemáticos y tratar de guardar una integridad y unicidad en las consultas a la base de datos.

#### **4.1.2.9. Configuración de Drupal hacia el rendimiento**

Obviamente, tras configurar todos los actores que influyen en el entorno de trabajo de Drupal (su pila de procesos) deberíamos optimizar la propia plataforma para evitar consumos innecesarios de recursos, como los producidos por tareas pesadas programadas (vía “cron”) con mucha frecuencia, las páginas almacenables por su contenido estático y demás cuestiones que podemos gestionar desde el panel de administración de Drupal. Además, existen en el mercado optimizaciones específicas de Drupal orientadas a la mejor del rendimiento, es decir, distribuciones (recordemos que se trata de código libre), como por ejemplo Pressflow [23].

### **4.1.3. Umbral de costes**

Una vez analizadas las necesidades del cliente en cuanto a rendimiento pasamos a los costes asumibles. Los costes asumibles en cuanto a hardware son relativamente bajos. Existen en el cliente, varios servidores dedicados a distintas tareas (ftp, ssh, http, mail) siendo la más importante, quizás, el correo electrónico. La topología de la red, por razones de seguridad, no me ha sido revelada en su totalidad, así que, digamos que debo pensar simplemente en las necesidades asumiendo que el coste asociado al servidor será relativamente bajo, llegando, en el máximo inicial, a una máquina dedicada. No descarto que en un futuro los costes vayan a aumentar debido a la afluencia de visitas, pero inicialmente, voy a contar, para este supuesto, con una máquina estándar que detallaré en la parte de requisitos de hardware. De la parte de terminales móviles (handhelds) y estaciones físicas dentro del museo, serían modelos de perfil bajo y costes marginales, pudiéndose tratar de modelos no precisamente punteros. Todo esto desde la perspectiva de rendimiento. Estéticamente, intuyo que la empresa se inclinaría (es razonable) por una opción no demasiado agresiva o estridente con lo encontrado en el museo (pantallas lcd, líneas básicas, tecnología “invisible”, etc.), lo que haría aumentar los costes moderadamente. De este particular no se han concretado números, aunque sí hay una disposición previa a proporcionar lo que sea necesario teniendo en cuenta las limitaciones presupuestarias (de fondos y de sometimiento a escrutinio exhaustivo) que hay en la administración pública.

## 4.2. Requisitos de Hardware

Teniendo todo lo anterior en cuenta, podemos listar una estimación de requisitos específicos de hardware.

### 4.2.1. Desarrollo

Actualmente, esta plataforma está siendo desarrollada en una estación de sobremesa relativamente modesta que contaría con unas características que yo considero medias (cuadro 4.2). Las páginas cargan con relativa fluidez (aunque todavía no está cargada toda la base de datos) y eso teniendo en cuenta que corre sobre una máquina virtual como ya he descrito más arriba.

- Modelo: HP Compaq dc7700 Convertible Minitower
- Arquitectura: 32 bit
- Procesador: Intel(R) Core(TM)2 CPU 6600 @ 2.40GHz
- Memoria RAM: 2 x (DIMM 1GB 667MHz)
- Disco duro: SATA 232GiB (250GB)
- Red: Ethernet Gigabit Network Connection (1Gbit/s)

Cuadro 4.2: Características hardware de la plataforma de desarrollo

### 4.2.2. Servidor

Para el servidor necesitaremos, en principio, cualquier máquina de potencia media actual (diciembre de 2010) o incluso, ligeramente inferior, pudiendo darse el caso de compartir, temporalmente hasta tener un conteo de los rendimientos, espacio con algún servidor ya utilizado. Deberá contar, eso sí, con algunas características técnicas que se detallan en el cuadro 4.3. Cabe destacar que, para la instalación del núcleo de Drupal, bastaría con unos 3MiB de espacio de disco duro, pero si a todo eso le añadimos un sistema operativo, un servidor web, un sistema gestor de base de datos, una instalación de PHP y unos cuantos módulos, los requisitos pueden variar notablemente.

- Arquitectura: 32 o 64 bit
- Procesador: Doble núcleo @ >= 2,4GHz
- Memoria RAM: >2GB (>= 667MHz)
- Disco duro: SATA >= 500GB (a ser posible, replicado en RAID)
- Red: Ethernet Gigabit

Cuadro 4.3: Requisitos hardware del servidor

#### 4.2.3. Infraestructura de red

Además, sería necesario disponer de la infraestructura adecuada de red. Si se decide que va a ser este el que tenga la salida a Internet, deberá estar dotado del correspondiente **enlace** con un “upstream” aceptable para no verse asediado por el cuello de botella de la red general. Claro está, las medidas de seguridad pertinentes deberán ser instaladas (**firewall, segmentación, NIDS/NIPS**) y el **enrutamiento** necesario para obtener acceso tanto desde internet como desde la intranet de una supuesta wifi o una estación en sala (vía segmentación ethernet de la red).

Para desplegar la infraestructura de red inalámbrica en el recorrido de la visita, habrá que implementar **puntos de acceso** donde no los haya y cablear los mismos hasta la red principal. La cobertura de uno de los edificios (actualmente ya desplegada) deberá ser desdoblada para permitir una red de trabajo y otra de visitantes que sólo permita el acceso a los contenidos del museo virtual.

#### 4.2.4. Cliente en sala

Para este cometido, valdrá cualquier equipo capaz de ejecutar un entorno gráfico, un navegador relativamente moderno y complementos de audio y vídeo no demasiado exigentes (cuadro 4.4).

- Arquitectura: 32 bit
- Procesador: @ >= 1GHz
- Tarjeta de video: >= 64MB RAM
- Memoria RAM: >= 1GB
- Disco duro: >= 100GB
- Red: Ethernet Gigabit
- Pantalla: >= 20 pulgadas, delgada, si es posible, táctil-capacitiva

Cuadro 4.4: Requisitos hardware del cliente en sala

#### 4.2.5. Dispositivo móvil

Aun tratándose de un dispositivo sin confirmar en esta fase del proyecto, creo conveniente agregarlo aquí. La idea es un dispositivo ergonómico, similar, en tamaño a una PDA y compatible con tecnologías parecidas a las del cliente en sala (cuadro 4.5). Se discute en diferentes foros [1] por dónde, después de haber visto como la mayoría de los dispositivos, a menudo bajo contrato de leasing, utilizados en exposiciones se iban quedando obsoletos rápidamente, deberían ir los tiros a partir de ahora en cuestiones de visitas “auto-guiadas”. Las conclusiones en materia de software están más o menos claras: se ha de crear una plataforma común (que bien podría ser un cms adaptado a plataformas móviles), incluso común a otras instituciones museísticas, encima de las cuales poder correr las “aplicaciones” necesarias para enriquecer y extender la experiencia del usuario en la exposición. En el lado hardware es donde se

presenta la disyuntiva: seguir con el modelo de dispositivos bajo préstamo/alquiler, optar porque los propios visitantes accedan con sus teléfonos inteligentes al contenido que se ofrece, o una mezcla de estas dos. Bajo esa premisa, veo precipitado aventurar unas características, aunque ahí quedan, lo que yo consideraría, un dispositivo apto para esta función.

- Procesador: @ >= 500MHz
- Memoria RAM: >= 64MB
- Disco duro: >= 256MB
- Red: Wi-Fi 802.11g/n
- Pantalla: en color >= 4 pulgadas, a poder ser, táctil-capacitiva
- Batería: larga vida
- Audio: salida de audio
- Construcción: resistente, ligera y lo más simple posible en cuanto a piezas.

Cuadro 4.5: Requisitos hardware del dispositivo móvil

## 4.3. Requisitos de Software

Para el caso del software vamos a glosar lo que en principio tiene que ver con la opción elegida (Drupal).

### 4.3.1. Desarrollo

El software que corre encima de la máquina descrita en la sección anterior está, como ya expliqué contenido en una máquina virtual. Esta máquina virtual ha optado por un sistema “**LAMP**”, aunque es cierto que Drupal puede correr sobre Windows con IIS (más la pertinente implementación de PHP), sobre otros servidores (LightHTTPD) o sobre implementaciones específicas de PHP. Incluso, algunas distribuciones de Drupal permiten interactuar con otros sabores de sistema gestor de base de datos diferente de MySQL.

### 4.3.2. Servidor

El servidor, dado que sería, en lo posible, el trabajo resultado del desarrollo, debería estar bajo otro sistema LAMP. Visto lo visto, puede ser recomendable integrar un servidor con un sistema Debian Linux (por estabilidad) o **Ubuntu Server** (por similitud con la plataforma de desarrollo). Es cierto, eso sí, que, sea en una u otra máquina, debería llegarse a un acuerdo y, preparar, si el presupuesto lo permite, un “servidor” de pruebas, independiente del de desarrollo y del de producción. Se buscará que el software en ellos sea el mismo, dejando las versiones más estables para el servidor

de producción, y siguiendo un ciclo clásico de **Do-Check-Act** desde el entorno de desarrollo, al servidor de pruebas y después, si no hay errores, al de producción.

Así que, en principio, la estructura será la misma que para el entorno de desarrollo, aunque es previsible que la modificación inicial será al revés: una vez configurado un entorno estable para el servidor de producción, se igualarán el resto de servidores. A partir de ahí, el ciclo de actualizaciones y/o cambios seguirá el curso propuesto más arriba.

#### **4.3.3. Cliente en sala**

En este caso, los requisitos software son muy sencillos: un sistema operativo que soporte la pantalla táctil (en su caso) y un navegador moderno que sepa interpretar los estándares del **W3C** y reproducir contenido multimedia. Además, claro está, se deberán tomar las medidas de seguridad necesarias para que no haya brechas importantes de seguridad, ya sea a base de políticas, permisos, antivirus, etc.

#### **4.3.4. Dispositivo móvil**

Para el dispositivo móvil necesitaremos prácticamente lo mismo que hemos requerido para el cliente en sala. Se trata, en realidad de una réplica que trata de ofrecer los mismos contenidos ahorrando en transferencias y procesamiento, pero al fin y al cabo, lo mismo. Un sistema operativo optimizado para plataformas móviles (Android, Windows Phone, iPhoneOS, Symbian) que tenga conectividad wifi, incluya un navegador como los anteriormente descritos y permita, conscientes de las limitaciones de hardware, la reproducción de la mayoría de contenidos multimedia ofrecidos, si no todos.

## **Capítulo 5**

# **Manual de Usuario**



## **5.1. Manual**

En esta sección se resumirá el manual de usuario de la aplicación en cada caso.



## **Capítulo 6**

## **Apéndices**



## 6.1. Apéndice I: Alfresco

La propuesta es analizar brevemente el candidato “Alfresco” para ver si sería viable tenerlo en cuenta como candidato para nuestro desarrollo.

### 6.1.1. Análisis

Para proceder con este análisis, la principal fuente de información será el sitio web de la compañía dedicado a hablar de la entidad. Alfresco es una empresa fundada por John Newton (Documentum<sup>1</sup>) y John Powell (Business Objects<sup>2</sup>) e integrada por ingenieros de Documentum y Oracle en 2005 con capital de diferentes empresas de inversión como SAP Ventures, Accel Partners y Mayfield Fund. Se autodenominan la “alternativa de código abierto para la gestión de contenidos empresariales (ECM) [26].

Aquí aparece un concepto desconocido hasta ahora en este documento: gestión de contenidos empresariales. Según la Asociación para la gestión de imágenes e información (AIIM en inglés), ECM se define como [27]: “las estrategias, métodos y herramientas usadas para capturar, gestionar, almacenar, preservar y entregar contenido y documentos relacionados con procesos organizativos. Las estrategias y herramientas ECM permiten la administración de la información no estructurada de una organización donde quiera que ésta exista.”\*

#### 6.1.1.1. Sabores

Aunque quizá ya era de esperar, podemos ver que este tipo de plataforma abarca mucho más de lo que en principio necesitaríamos y está enfocado de manera diferente a lo que sería nuestro museo virtual. Si nos atenemos a la plataforma, rápidamente vemos que forma parte del modelo de negocio que algunas empresas han encontrado para explotar el código abierto: el soporte. No hay otra manera (salvo una evaluación de 30 días) de utilizar la plataforma Enterprise que la de contratar una suscripción con la empresa atendiendo a diversos paquetes de soporte integrado. Las características incluidas en todas las opciones son:

- Pruebas certificadas de escalabilidad
- Un modelo controlado del lanzamiento
- Mantenimiento y Actualizaciones
- Resolución de Problemas
- Consejos de Configuración, Compatibilidad y Migraciones
- Soporte de Actualización
- Consejos de Puesta a punto

<sup>1</sup> Empresa americana dedicada a la creación de un gestor de contenidos empresariales con el mismo nombre, hoy mantenido por EMC Corporation. [24]

<sup>2</sup> Empresa francesa dedicada a Business Intelligence y desde 2007 parte de SAP AG [25]

\* Traducción propia.

- Tiempos de reacción garantizados

Como todo sitio corporativo, su lenguaje es marcadamente general por lo que conviene entrar un poco más en detalle. Para ello podemos utilizar el mismo recurso antes mencionado de CMSMatrix.org [28]. En este caso, no tenemos muy clara cuál es la plataforma analizada, ya que Alfresco se presenta en distintos sabores [29]: Cloud Trial (prueba online, no incluye gestión de contenidos web, Alfresco Enterprise (probada y certificada para plataformas tanto código abierto como propietarias, con soporte comercial completo) y Alfresco Community (para plataformas de código abierto, soportada por la comunidad).

Desgraciadamente, la página de CMSMatrix.org muestra unos valores de comienzos del 2008 con lo que vamos a evitarnos el trabajo de listarla.

#### 6.1.1.2. Edición Web

Alfresco se presenta como una opción flexible para desarrolladores. Para ello, ofrecen algo llamado Alfresco Quick Start. Se trata de una especie de sitio dinámico prefabricado construido en Spring Surf pero con el repositorio de contenidos marca de la casa (Alfresco Share). De hecho, lo mencionan en varios documentos como algo especial. Es traído a colación a cuenta de que, al parecer, llega un momento en todo desarrollo de sitios dinámicos en el que el usuario acaba por pedir más y más control, lo que carga al desarrollador de un montón de trabajo tratando de implementar funciones sencillas de administración del sitio. Esta herramienta presume de traer todas esas características de fábrica, para permitir a los constructores de sitios web seguir con tareas más importantes como la experiencia de usuario final. En teoría, lo engorroso de la gestión de contenidos queda del lado de Alfresco Share.

**6.1.1.2.1. Sitios colaborativos** Esta plataforma trata de simplificar, como toda aplicación informática presume, las tareas del usuario, dejando el horizonte libre para lo que sería, directamente, crear.

Entre otras características, cuenta con:

b en el navegador , lo que facilita mucho esta tarea al usuario final pudiendo valerse de contenidos previamente publicados.

- Integración de documentos ofimáticos automáticamente en la web (Office-to-Web), lo que permitiría, basándose en Google Docs (el sistema en la nube de edición colaborativa de documentos), con muy poco esfuerzo, pasar de bocetos en MS Office a publicar en la web respetando las normas CSS de la empresa.
- Unidad de red compartida, que no es más que extender el concepto que popularizó Microsoft hace muchos años pero en este caso, como si se tratase de la librería de la web.

# **Capítulo 7**

# **Bibliografía**



# Bibliografía

- [1] Tate modern | symposia | from audiotours to iphones. [Online]. Available: <http://www.tate.org.uk/modern/eventseducation/symposia/15441.htm>
- [2] Overview of theme files | drupal.org. [Online]. Available: <http://drupal.org/node/171194>
- [3] Liferay adopting the lgpl license - blog - liferay.com. [Online]. Available: <http://www.liferay.com/web/bryan.cheung/blog/-/blogs/4656695/maximized>
- [4] Drupal - open source cms | drupal.org. [Online]. Available: <http://drupal.org/>
- [5] A. Byron, A. Berry, N. Haug, J. Eaton, J. Walker, and J. Robbins, *Using Drupal*, 1st ed. O'Reilly Media, 12 2008.
- [6] Lullabot | drupal videos, training, consulting and guidance. [Online]. Available: <http://www.lullabot.com/>
- [7] A. Byron, A. Berry, N. Haug, J. Eaton, J. Walker, and J. Robbins, *Drupal*, 1st ed. Anaya Multimedia / O'Reilly, 09 2009.
- [8] Turnkey linux virtual appliance library | virtual appliances for virtual machines, cloud computing and bare metal. [Online]. Available: <http://www.turnkeylinux.org/drupal6>
- [9] Lamp (software bundle) - wikipedia, the free encyclopedia. [Online]. Available: [http://en.wikipedia.org/wiki/LAMP\\_\(software\\_bundle\)](http://en.wikipedia.org/wiki/LAMP_(software_bundle))
- [10] Drupal 6 appliance - turnkey linux virtual appliance library. [Online]. Available: <http://www.turnkeylinux.org/drupal6>
- [11] Turnkey core - turnkey linux virtual appliance library. [Online]. Available: <http://www.turnkeylinux.org/core>
- [12] shellinabox - project hosting on google code. [Online]. Available: <http://code.google.com/p/shellinabox/>
- [13] Webmin. [Online]. Available: <http://www.webmin.com/>
- [14] Drupal hispano | comunidad de usuarios de drupal. [Online]. Available: <http://drupal.org.es/>
- [15] M. de Fomento. [museo virtual] instituto geográfico nacional - ministerio de fomento. [Online]. Available: [http://www.ign.es/ign/es/IGN/Museo\\_Virtual.jsp](http://www.ign.es/ign/es/IGN/Museo_Virtual.jsp)

- [16] Drupal performance tuning and optimization for large web-sites - 2bits.con.inc. [Online]. Available: <http://2bits.com/articles/drupal-performance-tuning-and-optimization-for-large-web-sites.html>
- [17] E. Webb. (2010, 09) Getting started with drupal performance | erikwebb dot net. Atlanta Drupal User's Group Meeting. [Online]. Available: <http://2bits.com/articles/drupal-performance-tuning-and-optimization-for-large-web-sites.html>
- [18] Server tuning considerations | drupal.org. [Online]. Available: <http://drupal.org/node/2601>
- [19] Documentation: Apache http server - the apache http server project. [Online]. Available: <http://httpd.apache.org/docs/>
- [20] Zend Technologies, Inc. and Acquia, Inc. (2010, 02) Optimizing drupal performance - benchmark results. [Online]. Available: <http://www zend com/topics/Optimizing-Drupal-Performance-Zend-Acquia-Whitepaper-Feb2010v2.pdf>
- [21] A howto on optimizing php with tips and methodologies. [Online]. Available: <http://phplens.com/lens/php-book/optimizing-debugging-php.php>
- [22] Tuning mysql | drupal.org. [Online]. Available: <http://drupal.org/node/51263>
- [23] Pressflow | enhaced performance and scalability for drupal. [Online]. Available: <http://pressflow.org/>
- [24] Documentum - wikipedia, the free encyclopedia. [Online]. Available: <http://en.wikipedia.org/wiki/Documentum>
- [25] Bussiness objects (company) - wikipedia, the free encyclopedia. [Online]. Available: [http://en.wikipedia.org/wiki/Business\\_Objects\\_\(company\)](http://en.wikipedia.org/wiki/Business_Objects_(company))
- [26] Sobre alfresco - la alternativa para la gestión de contenidos empresariales en código libre. [Online]. Available: <http://www.alfresco.com/es/about/>
- [27] Aiim - what is ecm? what is enterprise content management. [Online]. Available: <http://www.aiim.org/What-is-ECM-Enterprise-Content-Management.aspx>
- [28] The cms matrix - cmsmatrix.org - the content management comparison tool. [Online]. Available: <http://cmsmatrix.org/>
- [29] Alfresco - cmsmatrix.org - the content management comparison tool. [Online]. Available: <http://cmsmatrix.org/matrix/cms-matrix/alfresco>