



UNIVERSIDAD DE JAÉN
Escuela Politécnica Superior (Jaén)

Proyecto Fin de Carrera

PROCESO DE MIGRACIÓN DE SISTEMAS WEB. APLICACIÓN AL SISTEMA DE RECOMENDACIÓN REJA

Alumno: Juan Francisco Galiano Fernández

Tutores: Dr. D. Luis Martínez López
D^a. Rosa M^a Rodríguez Domínguez

Dpto: Informática
Área: Lenguajes y Sistemas Informáticos

Septiembre, 2012

D. Luis Martínez López y D^a. Rosa M^a Rodríguez Domínguez pertenecientes al Departamento de Informática de la Universidad de Jaén.

INFORMAN

Que la memoria titulada “Proceso de Migración de Sistemas Web. Aplicación al Sistema de Recomendación REJA” ha sido realizada por **D. Juan Francisco Galiano Fernández** con DNI 75020960-C bajo nuestra dirección y se presenta como memoria del Proyecto Fin de Carrera para optar al grado de Ingeniería en Informática.

Jaén, 17 de septiembre de 2012

Vº Bº

Fdo: **D. Luis Martínez López**

Fdo: **D^a. Rosa M^a Rodríguez Domínguez**

PROCESO DE MIGRACIÓN DE SISTEMAS WEB. APLICACIÓN AL SISTEMA DE RECOMENDACIÓN REJA

Alumno: Juan Francisco Galiano Fernández

Tutores: Dr. D. Luis Martínez López
D^a. Rosa M^a Rodríguez Domínguez

Departamento: Informática

Área: Lenguajes y Sistemas Informáticos



Universidad de Jaén
Escuela Politécnica Superior de Jaén
Departamento de Informática

Quiero expresar mi más sincero agradecimiento a todas las personas que de un modo u otro han participado en la realización de este proyecto, con especial dedicación a...

Mis padres, por todos los esfuerzos que han hecho por mí y los valores que me han inculcado.

A Macarena, por estar siempre a mi lado y darme ánimos cuando más los necesitaba.

A mis hermanos, M^a José y Antonio, por ser un gran ejemplo a seguir.

A mis tutores, Rosa y Luis, sin los cuales no podría haber realizado este proyecto, gracias por vuestra ayuda y disponibilidad.

A mis amigos, porque son una parte más de mi familia.

Índice General

| | |
|---|-----------|
| Capítulo 1. Introducción | 1 |
| 1.1. Motivación y Propósito del Proyecto | 3 |
| 1.2. Objetivos | 6 |
| 1.3. Estructura de la Memoria | 6 |
| | |
| Capítulo 2. Procesos de Migración Software | 9 |
| 2.1. Marco Conceptual sobre la Migración de Software | 11 |
| 2.1.1. Definición de Migración Tecnológica | 11 |
| 2.1.2. El Proceso de Migración dentro del Ciclo de Vida Software | 12 |
| 2.1.3. Esquemas de Procesos de Migración | 20 |
| 2.2. Metodología de Migración Clásica..... | 22 |
| 2.3. Terminología Básica sobre Web..... | 29 |
| 2.4. Metodología de Migración basada en Tecnologías Web..... | 33 |
| 2.4.1. Análisis de Requerimientos | 36 |
| 2.4.2. Diseño..... | 40 |
| 2.4.3. Proceso de Migración..... | 50 |
| 2.4.3.1. Migración Parcial..... | 51 |
| 2.4.3.2. Migración Completa..... | 55 |
| 2.4.5. Evaluación | 57 |
| 2.4.6. Mantenimiento..... | 58 |
| | |
| Capítulo 3. Proceso de Migración para un CMS | 61 |
| 3.1. Presentación del Problema..... | 63 |
| 3.1.1. ¿Qué es un CMS? | 63 |
| 3.1.2. Ventajas | 71 |
| 3.1.3. Historia..... | 73 |
| 3.2. Propuesta Metodológica para Migración de CMS..... | 74 |
| 3.2.1. Análisis de Requerimientos | 75 |
| 3.2.2. Diseño..... | 77 |
| | |
| Capítulo 4. Migración del Sistema de Recomendación de Restaurantes REJA en el CMS Joomla | 83 |
| 4.1. Descripción del Sistema de Recomendación de Restaurantes Georreferenciados REJA | 85 |

| | |
|--|------------|
| 4.2. Introducción al CMS utilizado | 87 |
| 4.2.1. Definición de Joomla | 87 |
| 4.2.2. Estructura | 89 |
| 4.2.3. Funcionamiento..... | 91 |
| 4.2.4. Ventajas y Características | 95 |
| 4.3. Proceso de Migración de REJA paso a paso | 96 |
| 4.3.1. Análisis de Requerimientos..... | 96 |
| 4.3.2. Diseño | 106 |
| 4.3.3. Proceso de Migración | 124 |
| 4.3.3.1. Migración Parcial | 125 |
| 4.3.3.2. Migración Completa | 126 |
| 4.3.5. Evaluación..... | 134 |
| 4.3.6. Mantenimiento | 138 |
| 4.4. Procesos Adicionales a la Metodología Propuesta para Problemas Específicos | 140 |
| | |
| Capítulo 5. Conclusiones | 143 |
| | |
| Bibliografía | 147 |
| | |
| Anexo I. Procedimientos de Copias de Seguridad en Joomla..... | 151 |
| a) Procedimiento Manual | 153 |
| b) Procedimiento Automático..... | 155 |
| | |
| Anexo II. Estrategias de Migración en Joomla..... | 161 |
| a) Migración Automatizada. Método 1 | 164 |
| b) Migración Automatizada. Método 2 | 165 |
| | |
| Anexo III. Inventario Hardware y Software..... | 169 |

Capítulo 1

Introducción

En este capítulo se realiza la presentación del Proyecto Fin de Carrera. Se justifican los motivos y propósitos de la realización del mismo, junto con sus objetivos y la estructura de la composición de la memoria en los diferentes capítulos.

1.1. Motivación y Propósito del Proyecto

La migración de un sistema a otro es una necesidad que se ha tenido desde los inicios de la informática, debido a la constante evolución que caracteriza el desarrollo del software. Inicialmente los expertos informáticos desarrollaban software que era usado por ellos mismos, de manera que la migración de este software no representaba excesiva dificultad para ellos. Todavía el software no era un producto de uso masivo.

En la actualidad, el uso de diferente software está muy extendido y las alternativas son muy amplias, de modo que se puede encontrar gran variedad de software para una misma finalidad. Al ser las exigencias del mercado y las demandas de los usuarios sobre la calidad del software y personalización cada vez mayores, el proceso de migración de un software a otro se produce con relativa frecuencia.

Estas migraciones de software pueden deberse a diferentes razones, como que el sistema se haya quedado obsoleto o que surjan necesidades que no cubran los sistemas actuales. En muchas ocasiones, el hecho de cambiar de un software a otro, también se debe al deseo de probar cosas nuevas y mantenerse actualizado con las últimas versiones que existan en el mercado.

Al igual que las migraciones se pueden realizar entre distintos software con funcionalidades similares, las migraciones también pueden darse en un mismo software para cambiar hacia una versión mejorada. Este cambio se puede producir en cualquier momento y por cualquier razón. La Primera Ley de la Ingeniería de Sistemas [1] establece que: *sin*

importar en qué momento del ciclo de vida del sistema nos encontremos, el sistema cambiará, y el deseo de cambiarlo persistirá a lo largo de todo el ciclo de vida.

El origen de estos cambios puede ser muy variado, pero se distinguen cuatro fuentes fundamentales de cambio [2]:

- Nuevos negocios o condiciones comerciales que dictan los cambios en los requisitos del producto o en las normas comerciales.
- Nuevas necesidades del cliente que demandan la modificación de los datos producidos por sistemas de información, funcionalidades entregadas por productos o servicios entregados por un sistema informático.
- Reorganización, crecimiento o reducción del negocio que provoca cambios en las prioridades del proyecto o en la estructura del equipo de ingeniería del software.
- Restricciones presupuestarias o de planificación que provocan una redefinición del sistema o producto.

La implantación y crecimiento de Internet y sus tecnologías asociadas, fundamentalmente la web, han hecho que aparezcan nuevos retos en los procesos de migración de sistemas. La web fue diseñada inicialmente de manera simple y a bajo costo para satisfacer la necesidad de acceso a la información mediante páginas estáticas. Luego se adaptó a la necesidad de interacción entre la web y el usuario, mediante páginas dinámicas. Posteriormente se incluyeron bases de datos en su diseño, de manera que se puede solicitar o almacenar información durante dicha interacción. Esto dio lugar a la aparición de sistemas de gestores de contenido, que permiten crear una estructura de soporte, para la creación y administración de contenidos, principalmente en sitios web, por parte de los administradores, editores, participantes y demás roles que se definan.

El mantenimiento es uno de los factores más importantes para el éxito de cualquier aplicación web. La actualización periódica de los datos y el mantenimiento del diseño son aspectos obligatorios a tener en cuenta como parte de su implementación. De nada sirve tener una página perfectamente diseñada con información desactualizada, o información al día con un diseño poco atractivo por el que es imposible navegar. Ambos aspectos, si bien son totalmente diferentes, están estrechamente relacionados en el éxito de un sitio web.

Hay dos factores que a menudo conducen a la remodelación de un sitio web:

- Cambiar a una nueva plataforma tecnológica, tales como un sistema de gestión de contenidos nuevo, o una nueva versión del usado actualmente, como es el proyecto que nos ocupa.
- El rediseño del sitio web, ya sea para hacer frente a las debilidades en el sitio actual o añadir una funcionalidad significativa.

A menudo, estos dos factores se juntan, con un proceso de selección de la tecnología combinada con un nuevo diseño. Un cambio en la plataforma de la tecnología hace que la migración sea un desafío, al igual que una importante reestructuración o rediseño del sitio web.

En este PFC, realizaremos una revisión de diferentes metodologías de migración, que nos servirá de ayuda para hacer una propuesta de metodología de migración para sistemas gestores de contenidos CMS. A continuación, aplicaremos dicha propuesta al caso real de un Sistema de Recomendación de Restaurantes Georreferenciados de la provincia de Jaén (REJA) [3].

1.2. Objetivos

El principal propósito del proyecto es el estudio y propuesta de una metodología de migración basada en plataformas CMS, y la aplicación de dicha metodología a la migración de un sistema de recomendación de restaurantes, REJA. Por tanto, los objetivos del proyecto serán:

- Recopilación y revisión bibliográfica de metodologías de migración.
- Propuesta de estrategias y herramientas de migración específicas para sistemas CMS.
- Aplicación de la metodología anterior al sistema de recomendación REJA.
- Evaluación de la migración y conclusiones del proceso de migración.
- Realización de una Memoria

1.3. Estructura de la Memoria

En esta sección vamos a presentar los distintos capítulos en los que se estructura este proyecto y los contenidos de los mismos.

El primer capítulo es una introducción mediante la cual se presenta la motivación de este proyecto, los propósitos que se perseguirán y los objetivos a conseguir.

En el capítulo 2 se realiza una contextualización del concepto de migración software, definiendo la migración tecnológica, localizando el proceso de migración en el ciclo de vida software y enumerando diferentes esquemas de procesos de migración. A continuación, se describe una metodología de migración clásica y una metodología de migración basada en tecnologías web.

En el capítulo 3 se presenta el problema concreto referente a la migración de sistemas basados en gestores de contenidos proponiendo una metodología específica de migración adaptada a estos sistemas.

En el capítulo 4 se lleva a la práctica el proceso teórico descrito en el capítulo anterior, utilizando para ello la migración de una aplicación web de un Sistema de Recomendación de Restaurantes de Jaén, denominado REJA, desarrollada sobre el CMS Joomla.

El capítulo 5 recoge las conclusiones generales obtenidas del desarrollo del proyecto.

Capítulo 2

Procesos de Migración Software

En este capítulo se realiza una contextualización del concepto de migración software, para lo cual se define el término migración tecnológica, se ubica el proceso de migración dentro de la definición del ciclo de vida software y se explican diferentes esquemas de procesos de migración. En segundo lugar se describe una metodología de migración clásica. En tercer lugar se definen varios conceptos que son muy utilizados a lo largo del documento. Finalmente se describe una metodología de migración sobre tecnologías web.

2.1. Marco Conceptual sobre la Migración de Software

2.1.1. Definición de Migración Tecnológica

Una migración tecnológica es un proceso de cambio en los elementos del software y/o hardware. Toda migración incluye una serie de pasos a seguir:

- Determinación de la causa de la migración.
- Fijar el momento de la migración.
- Fijar el procedimiento de migración.
- Evaluar la migración.

Un proceso de migración no puede darse sólo con la sustitución del software, pues están involucrados factores de preparación y previsión que deben ser tenidos en cuenta.

Un ejemplo de migración de software puede ser la transición de una plataforma a otra completamente distinta. Otro ejemplo, es el cambio de versión de un sistema operativo a una versión más actualizada. Incluso en este cambio dentro de un mismo sistema operativo, no se debe suponer que los formatos de archivos, por ejemplo, serán transportables, por lo que habrá que hacer pruebas de funcionamiento antes de realizar cualquier modificación.

Todas las migraciones deben basarse en una cuidadosa planificación para evitar posibles pérdidas de información o funcionalidad.

Antes de tomar cualquier decisión, hay que tener en cuenta cuáles son las funcionalidades del nuevo software [4]. Cuando se tienen varias opciones, los responsables del proceso de migración deben conocer las ventajas e inconvenientes de cada producto. Se aconseja consultar a los usuarios y explicarles las razones por las que se va a llevar a cabo la migración y cómo les afectará, para ayudar al éxito del proceso.

2.1.2. El Proceso de Migración dentro del Ciclo de Vida Software

Dado que nuestro PFC se centra en el proceso de migración, vamos a comentar en qué fase está enmarcado este proceso dentro del ciclo de vida software. Para ello primero definimos qué es el ciclo de vida software:

- Según el estándar IEEE 1074 [5]: “Una aproximación lógica a la adquisición, suministro, desarrollo, explotación y mantenimiento del software”.
- Según la norma ISO 12207 [6]: “Es un marco de referencia que contiene los procesos, las actividades y las tareas involucradas en el desarrollo, explotación y mantenimiento de un producto software, abarcando la vida del sistema desde la definición de los requisitos hasta la finalización de su uso”.

La norma UNE 71044 [7] agrupa las actividades que pueden llevarse a cabo durante el ciclo de vida software en cinco procesos principales, ocho procesos de apoyo y cuatro procesos organizativos. Cada proceso del ciclo de vida está dividido en un conjunto de actividades, y cada actividad dividida a su vez, en un conjunto de tareas. Los cinco procesos principales son:

- a) *Proceso de adquisición*: Define las actividades del adquirente, organización que adquiere un sistema, producto software o servicio software.
- b) *Proceso de suministro*: Define las actividades del suministrador, organización que proporciona el sistema, producto software o servicio software al comprador.
- c) *Proceso de desarrollo*: Define las actividades del desarrollador, organización que define y desarrolla el producto software.
- d) *Proceso de operación*: Define las actividades del operador, organización que proporciona el servicio de operar un sistema informático en su entorno real, para sus usuarios.
- e) *Proceso de mantenimiento*: Define las actividades del equipo de mantenimiento, organización que proporciona el servicio de mantenimiento del producto software; esto es, la gestión de las modificaciones al producto software para mantenerlo actualizado y operativo. Este proceso incluye la migración y retirada del producto software.

En la Figura 2.1 se muestran las fases de cada uno de estos cinco procesos.



Figura 2.1. Actividades del Ciclo de Vida Software según UNE 71044

Uno de los procesos principales descritos en las definiciones del ciclo de vida software y mostrado en la Figura 2.1 es el “*Proceso de Mantenimiento*”. Este proceso ha sido definido de diferentes formas:

- El estándar IEEE 1219 [8] lo define como: “la modificación de un producto software después de haber sido entregado a los usuarios o clientes, con el fin de corregir defectos, mejorar el rendimiento u otros atributos, o adaptarlo a un cambio en el entorno”.
- La norma ISO 12207 [6] establece que: “el Proceso de Mantenimiento contiene las actividades y tareas realizadas por el mantenedor. Este proceso se activa cuando el producto software sufre modificaciones en el código y la documentación asociada, debido a un problema o a la necesidad de mejora o adaptación. El objetivo es modificar el producto software existente preservando su integridad. Este proceso incluye la migración y retirada del producto software”.
- Pressman [2] define que: “la fase de mantenimiento se centra en el cambio que va asociado a la corrección de errores, a las adaptaciones requeridas a medida que evoluciona el entorno del software, y a cambios debidos a las mejoras producidas por los requisitos cambiantes del cliente”.

En las anteriores definiciones de mantenimiento aparecen indicados, directa o indirectamente, cuatro tipos de mantenimiento: correctivo, adaptativo, de mejora y preventivo:

- *Correctivo*: Incluso llevando a cabo las mejores actividades de garantía de calidad, es muy probable que el cliente descubra algunos defectos en el software. El mantenimiento correctivo cambia el software para corregir los defectos.

- *Adaptativo*: Con el paso del tiempo, es probable que cambie el entorno original (por ejemplo: CPU, el sistema operativo, las reglas de empresa, las características externas de productos) para el que se desarrolló el software original. El mantenimiento adaptativo produce modificación en el software para acomodarlo a los cambios de su entorno externo.
- *Preventivo*: El software se deteriora debido al cambio, y por esto el mantenimiento preventivo, también llamado reingeniería del software, debe permitir que el software sirva para las necesidades de los usuarios finales. En esencia, el mantenimiento preventivo hace cambios en programas a fin de que se puedan corregir, adaptar y mejorar más fácilmente.
- *Perfectivo*: A medida que se utiliza el software, el cliente/usuario puede descubrir funciones adicionales que van a producir beneficios. El mantenimiento perfectivo lleva al software más allá de sus requisitos funcionales originales.

En la Figura 2.2 se muestra la clasificación de las peticiones de modificación y tipos de mantenimiento que se han definido anteriormente.

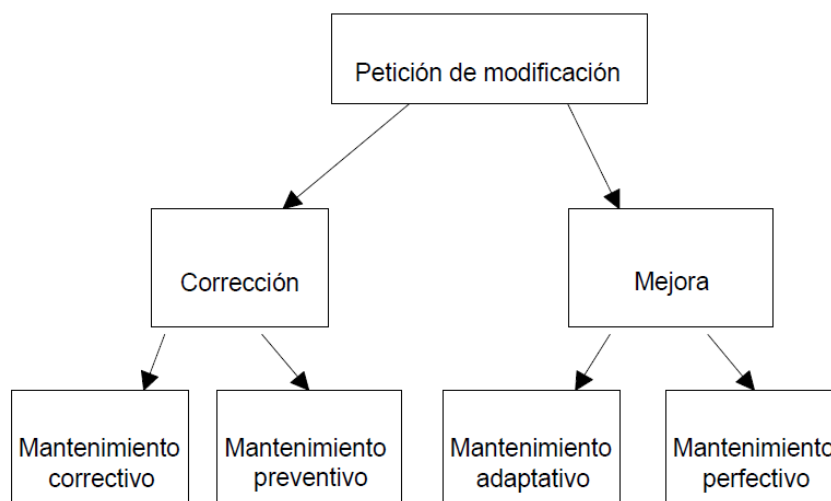


Figura 2.2. Clasificación de Peticiones de Modificación

Sin embargo, en la definición de la norma ISO 14764 [6], el Proceso de Mantenimiento Software consta de seis actividades (ver Figura 2.3):

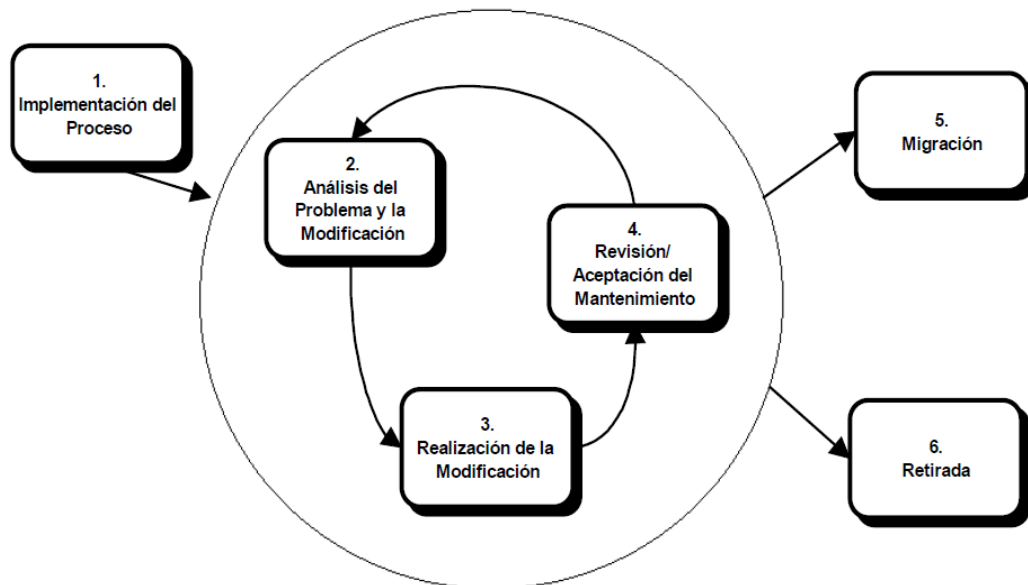


Figura 2.3. Actividades del Proceso de Mantenimiento

1. *Implementación del proceso*: Esta actividad desarrolla los planes correspondientes para llevar a cabo las tareas del mantenimiento. También define los procedimientos necesarios para la gestión de problemas y petición de modificaciones (empleando el proceso de resolución de problemas), e implementa el proceso de gestión de configuración para migrar las modificaciones del sistema existente.
2. *Análisis de problemas y modificaciones*: Esta actividad consiste en analizar los problemas o peticiones de modificación con el fin de evaluar su impacto en el sistema y la organización existentes, determinando el tipo de modificación (preventiva, correctiva, etc.), su alcance (tamaño, coste, tiempo, etc.) y su nivel crítico (rendimiento, seguridad, etc.). La organización encargada del mantenimiento debe también verificar el problema, elaborar distintas opciones

para implementar las modificaciones, y documentar el problema o la petición de modificación, así como los resultados del análisis y las opciones de implementación. Por último, debe obtener la aprobación para la opción seleccionada.

3. *Implementación de las modificaciones*: En esta actividad se incluyen todas las tareas relativas a determinar qué documentación, unidades de software y versiones deben modificarse, y se utiliza el proceso de desarrollo para implementar las modificaciones. Los requisitos del proceso de desarrollo deberán complementarse, según el estándar, de la siguiente manera:
 - a. Se deberá definir y documentar los criterios para probar y evaluar las partes del sistema (unidades, componentes y elementos de la configuración) modificadas y no modificadas.
 - b. Se deberá asegurar la completa y correcta implementación de los nuevo requisitos y/o modificados que no se vean afectados por los requisitos originales no modificados. También se deberá documentar los resultados de las pruebas.
4. *Revisión y aceptación del mantenimiento*: Esta actividad consiste en la revisión de la integridad del sistema modificado, que llevará a cabo la organización encargada del mantenimiento junto con la organización que autorizó la modificación. La organización encargada del mantenimiento deberá obtener también la aprobación de terminación satisfactoria de la modificación.
5. *Migración*: El estándar aconseja el desarrollo de un plan de migración en el que se especifiquen al menos las siguientes cuestiones:
 - a. Análisis de requisitos y definición de la migración

- b. Desarrollo de herramientas de migración
- c. Conversión del software y de los datos
- d. Ejecución de la migración
- e. Verificación de la migración
- f. Soporte del entorno antiguo en el futuro

Hay que notificar a los usuarios la intención de llevar a cabo la migración (describir el nuevo entorno, la fecha en que estará operativo, etc.), así como de ejecutar de forma paralela los dos entornos y de informar a los usuarios cuando se realice la migración prevista.

En esta actividad también se incluye una tarea de revisión post operación, con el fin de evaluar el impacto que suponga el cambio al nuevo entorno.

Por último, se deberá poder acceder a los datos utilizados o asociados al antiguo entorno de acuerdo con los requisitos organizacionales para la protección y auditoría aplicables a los datos.

6. *Retirada del software*: Es necesario desarrollar y documentar un “plan de retirada” que aborde cuestiones como las siguientes:

- a. Cese de soporte total o parcial después de un cierto tiempo
- b. Archivo del producto software y su documentación asociada
- c. Responsabilidad sobre cuestiones de soporte residual futuro
- d. Transición al nuevo producto
- e. Accesibilidad de copias de datos

Es importante que se tenga en cuenta a los usuarios a la hora de planificar la retirada del software y que se les notifique el plan. Las notificaciones deberán incluir lo siguiente:

- i. Descripción de la sustitución o actualización con su fecha de disponibilidad.
- ii. Informe de por qué no se soportará más el software.
- iii. Descripción de otras opciones de soporte disponibles una vez que se haya eliminado el soporte.

También se recomienda llevar a cabo operaciones paralelas entre el software nuevo y el retirado, además de proporcionar formación a los usuarios. Cuando tenga lugar la retirada planeada, se deberá notificar a todos los involucrados. Se debe archivar, según las necesidades, toda la documentación, ficheros y código.

Por último, al igual que en el caso de la migración, se deberá poder acceder a los datos utilizados por, o asociados con, el software retirado de acuerdo a los requisitos organizacionales de protección y auditoría aplicables a los datos.

2.1.3. Esquemas de Procesos de Migración

En cualquier proceso de migración de software se deben tener en cuenta ciertas consideraciones que determinan el desarrollo del proceso. A continuación se presentan las siguientes opciones en cuanto a esquemas de migración [4]:

- a) *En un único paso*: Todos los usuarios cambian del sistema antiguo al nuevo a la vez. En la práctica, esto supone programar el cambio cuando se prevea un bajo uso del sistema para causar los menores inconvenientes posibles.

La ventaja es que no se necesita tener doble acceso y el usuario no se va a encontrar con dos sistemas a la vez. Entre las desventajas está el alto riesgo y la gran exigencia de recursos durante el cambio.

- b) *Transición piloto e implementación*: consiste en migrar en una primera fase una pequeña parte del sistema, para poder comprobar diferentes parámetros como esfuerzo requerido, recursos utilizados, o tiempo empleado para poder obtener conclusiones para llevarlas a cabo en la transición completa.

Una de las ventajas de este esquema es que se pueden corregir errores inesperados o incompatibilidades no contemplados sin perder la funcionalidad o la productividad del sistema actual. Otras ventajas pueden ser, mostrar parte del sistema ya migrado a los usuarios para obtener opiniones, comprobar la forma de llevar a cabo la migración y crear un proceso de migración para aplicarlo posteriormente. Entre las desventajas está el mantener un sistema piloto simultáneamente al sistema actual y se requieren más recursos para la migración.

- c) *Transición por fases en grupos*: Se pasa a los usuarios del antiguo sistema al nuevo en grupos. Puede que los grupos funcionales completos se trasladen juntos para minimizar la compartición de datos y los problemas de trabajo en grupo. Se pueden limitar los riesgos y gestionar los recursos eligiendo grupos del tamaño adecuado.

Una ventaja es que si se identifica un error, éste puede ser subsanado para que no se produzcan en los siguientes grupos, ya que la migración no afecta a todo el sistema. La principal desventaja de esta estrategia es que es costosa tanto en dinero como en tiempo.

d) *Transición de usuario a usuario*: Básicamente se trata de la misma transición en grupos, pero con un grupo compuesto por una sola persona. Ese método tiene escasos requisitos en cuanto a los recursos, pero no resulta eficaz ni apropiado para grandes sistemas. Puede ser una buena manera de ejecutar proyectos piloto.

Al ser similar a la transición por fases en grupos, la ventaja es que se puede identificar un problema y subsanarlo para que no se le presente al resto de usuarios. Como desventaja es que es aún más costosa que la anterior.

2.2. Metodología de Migración Clásica

Una vez ubicada la fase de migración dentro del proceso completo del ciclo de vida del software, vamos a centrarnos en esta fase, ya que es la que atañe a nuestro proyecto.

Según las directrices de Intercambio de Datos entre Administraciones (IDA) de migración a software de fuentes abiertas [4], cualquier ejercicio de migración debe incluir, en general, las siguientes fases:

1. Una fase de definición del proyecto y de recopilación de datos, en la que se contemplen:
 - a. La descripción del conjunto de condiciones iniciales relevantes consistentes, por ejemplo, en:
 - i. arquitectura o arquitecturas de los sistemas,
 - ii. aplicaciones y sus datos asociados,
 - iii. protocolos y normas empleados,
 - iv. hardware,
 - v. el entorno físico, como el ancho de banda de la red, la ubicación,

- vi. los requisitos sociales como el idioma o idiomas y la capacitación del personal.
 - b. Un conjunto de condiciones finales con el mismo detalle.
 - c. Una descripción de cómo llegar de las condiciones iniciales a las condiciones finales.
2. Una justificación de la migración, incluido el coste asociado a la misma.
 3. Una o más fases piloto preparadas para probar si el plan y la justificación funcionan. Los datos de estas fases piloto pueden luego alimentar el modelo de costes usado en el plan.
 4. Despliegue del plan.
 5. Seguimiento de la experiencia real en relación con el plan.

Basándonos en la norma ISO/IEC 14764 [9], estándar específico sobre mantenimiento de software, vamos a revisar cada una de las acciones y tareas que conlleva el proceso de migración, al igual que el desarrollo y documentación necesaria para efectuar la migración.

Esta migración se debe efectuar de forma que cumpla las normas ISO/IEC 12207 [6], estándar para los procesos de ciclo de vida del software, desarrollando un plan de migración, notificando a los usuarios la migración, proporcionando entrenamiento, avisando de la finalización de la migración, calculando el impacto del nuevo entorno y archivando los datos

A continuación pasamos a describir de forma detallada cada uno de estos pasos (ver Figura 2.4):

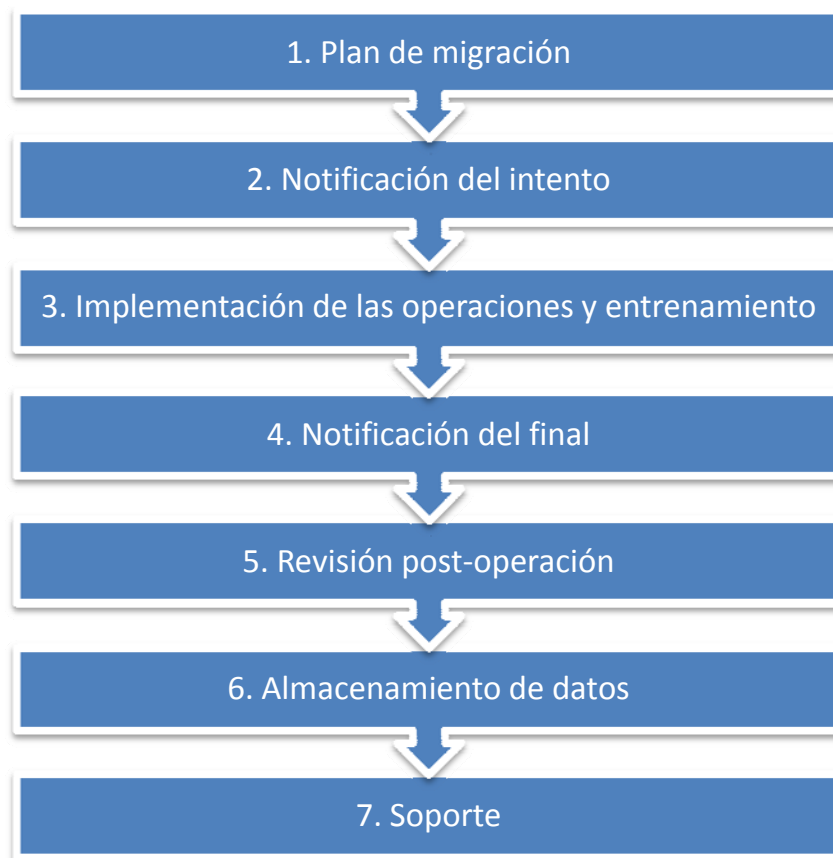


Figura 2.4. Proceso de migración basado en las normas ISO/IEC 14764

1. Plan de migración:

Para que se pueda controlar de forma adecuada la migración de un sistema, se debe crear un plan de migración, además de documentarlo y ejecutarlo.

Las actividades de planificación deberían incluir:

- Análisis de requerimientos y definición de la migración
- Desarrollo de herramientas de ayuda a la migración
- Conversión de datos y productos software

- Ejecución de la migración
- Verificación de la migración
- Soporte para el antiguo entorno

El desarrollo del Plan de Migración debe incluir las entradas que puedan proporcionarnos los usuarios. Como parte de esta tarea, se deberían ejecutar los siguientes pasos:

- Analizar los requerimientos de la migración
- Determinar el impacto de la migración del producto software
- Establecer una planificación para efectuar la migración
- Identificar los requerimientos de los conjuntos de datos para su revisión posterior
- Definir y documentar el esfuerzo de migración
- Determinar y mitigar los riesgos
- Identificar las herramientas de migración necesarias
- Desarrollar y/o adquirir las herramientas de migración
- Descomponer los productos y datos software de forma incremental para su conversión
- Convertir los productos y datos software
- Migrar los productos y datos software al nuevo entorno
- Ejecutar las operaciones paralelas
- Verificar la migración por medio de pruebas
- Proporcionar soporte para el antiguo entorno

2. Notificación del intento

Una vez que se haya completado la planificación de la migración, los usuarios deben recibir la notificación de los planes y actividades de migración. Dentro de estas notificaciones debería estar:

- Explicación de por qué ya no da soporte al antiguo entorno
- Descripción del nuevo entorno junto a la fecha de disponibilidad
- Descripción de otras opciones de soporte disponibles, si las hay, una vez que hayamos abandonado el antiguo entorno

En este momento se debe proporcionar a los usuarios la planificación y procedimientos necesarios para llevar a cabo la migración:

- Identificar los puestos afectados por la migración
- Obtener la realimentación del puesto afectado
- Identificar aspectos específicos del puesto
- Divulgar la planificación

3. Implementación de las operaciones y entrenamiento

Las operaciones paralelas del antiguo y nuevo entorno deberían dirigirse de forma que la transición del antiguo al nuevo entorno sea suave. Como parte de ésta tarea, se deben realizar los siguientes pasos:

- Conservar un puesto en su configuración original
- Instalar el software

- Ejecutar algunas pruebas preliminares para asegurarnos una correcta instalación del hardware y del software
- Ejecutar el software con una carga operativa en el entorno antiguo y en el nuevo
- Recoger datos de los productos nuevos y antiguos
- Reducir y analizar los datos

Si se quiere dar un correcto entrenamiento, se debe:

- Identificar los requisitos de entrenamiento
- Planificar los requisitos de entrenamiento
- Dirigir la revisión del entrenamiento
- Actualizar los planes de entrenamiento

4. Notificación del final

Una vez que se llegue al final de la migración planificada, se envía la notificación a todos los interesados. Toda la documentación asociada al antiguo entorno, así como los registros y código se archivan. Como parte de esta tarea, hay que:

- Divulgar los cambios de la planificación de la migración
- Documentar los aspectos específicos del puesto y como se resolverán
- Archivar los datos y el software antiguo
- Retirar el antiguo equipo

5. *Revisión post-operación*

Este proceso se realiza para calcular el impacto de los cambios en un nuevo entorno. Los resultados de la revisión se envían a las autoridades apropiadas para su información, guía y actuación. Como parte de este riesgo es aconsejable:

- Revisar los resultados al operar con los dos entornos a la vez
- Identificar las áreas con un riesgo potencial
- Identificar aspectos específicos del puesto
- Documentar las lecciones aprendidas
- Generar y anticipar un informe sobre el análisis del impacto

6. *Almacenamiento de datos*

Los datos usados o asociados con el antiguo entorno deben quedar accesibles de acuerdo con los requisitos del contrato para su protección o auditoría. Esta tarea se divide en las siguientes subtareas:

- Hacer copias de los datos y del software antiguo
- Almacenar las copias en un lugar seguro

7. *Soporte*

La actividad de Migración usa los siguientes procesos del ciclo de vida de ISO/IEC 12207 [6] relativos a la organización y al soporte:

- *Proceso de documentación*: Define las actividades para el registro de la información producida por un proceso del ciclo de vida.

- *Proceso de gestión de la configuración*: Define las actividades de la gestión de la configuración.
- *Proceso de aseguramiento de la calidad*: Define las actividades para asegurar, de una manera objetiva, que los productos software y los procesos son conformes a sus requerimientos especificados y se ajustan a sus planes establecidos.
- *Proceso de verificación*: Define las actividades para verificar hasta un nivel de detalle dependiente del proyecto software, los productos software.
- *Proceso de validación*: Define las actividades para validar los productos software del proyecto software.
- *Proceso de revisión conjunta*: Define las actividades para evaluar el estado y productos de una actividad. Este proceso puede ser empleado por cualquiera de las dos partes, donde una de las partes revisa a la otra parte, de una manera conjunta.
- *Proceso de auditoría*: Define las actividades para determinar la conformidad con los requerimientos, planes y contrato. Este proceso puede ser empleado por dos partes cualesquiera, donde una parte audita los productos software o actividades de otra parte.
- *Proceso de solución de problemas*: Define las actividades para analizar y eliminar los problemas (incluyendo las no conformidades) que sean descubiertos durante la ejecución del proceso de desarrollo, operación, mantenimiento u otros procesos, cualesquiera que sea su naturaleza o causa.

2.3. Terminología Básica sobre Web

Antes de avanzar en el desarrollo del proyecto, vemos necesario definir varios términos relacionados con las tecnologías web y que serán muy utilizados a lo largo del

documento y podrían crear confusión. Dependiendo de la bibliografía consultada o el contexto que aparezcan, estos términos pueden tener diferente significado. Para evitar confusión alguna, definimos los siguientes términos y concretamos su significado a lo largo de este documento.

- *Página web*

Una página web es un documento o información electrónica adaptada para Internet y que puede ser accedida mediante un navegador. Esta información se encuentra generalmente en formato HTML o XHTML, y puede proporcionar navegación a otras páginas web mediante enlaces de hipertexto. Las páginas web frecuentemente incluyen otros recursos como hojas de estilo en cascada, imágenes digitales, sonidos, etc.

Las páginas web pueden estar almacenadas en un equipo local o un servidor web remoto. El servidor web puede restringir el acceso únicamente para redes privadas. El acceso a las páginas web es realizado mediante su transferencia desde servidores utilizando el protocolo de transferencia de hipertexto (HTTP).

- *Sitio web*

Un sitio web es un conjunto de páginas web que están relacionadas entre sí, por lo general se acceden desde un mismo dominio o mantienen constante la raíz de la dirección URL. Un sitio web puede estar constituido de una o más páginas web. La página web principal de un sitio web suele llamarse index y que puede tener la extensión .htm, .php, .asp, entre otras.

Los Sitios Web pueden ser de diversos géneros, destacando los sitios de negocios, servicio, comercio electrónico, imagen corporativa, entretenimiento y sitios informativos.

- **Plataforma web**

En informática, una plataforma es un software y/o hardware determinado con el cual una aplicación es compatible y permite su ejecución. Una plataforma es, por ejemplo, un sistema operativo, un gran software que sirve como base para ejecutar determinadas aplicaciones compatibles con éste. También son plataformas la arquitectura de hardware, los lenguajes de programación y sus librerías en tiempo de ejecución, las consolas de videojuegos, etc. Existen programas multiplataforma, que permiten ejecutarse en diversas plataformas, al igual que emuladores, que son programas que permiten ejecutar desde una plataforma programas de otra emulando su funcionamiento.

Para definir una plataforma web, podría hacerse referencia a un sistema que permite crear y administrar un sitio web. Es el caso de Blogspot, MySpace o cualquier otra plataforma web que nos permita crear un blog o página web, empleando un sistema web donde podemos subir artículos, crear categorías, administrar los usuarios que se suscriban, etc.

En otro contexto, una definición correcta de plataforma web podría hacer referencia al conjunto de programas y sistemas que hacen que un sitio web esté en funcionamiento: el servidor web, la base de datos, el intérprete de programación, etc.

A continuación se enumeran dos ejemplos para cada tipo de plataforma dependiendo del contexto del término:

- Ejemplos de plataformas como sistemas operativos: Windows y Linux
- Ejemplos de plataformas de software: Java - JDK y JRE y .NET Framework
- Ejemplos de plataformas de hardware: Macintosh y cualquier variedad de consola de videojuego. Por ejemplo, Xbox, PlayStation, etc.
- Ejemplos de plataformas web: Joomla y Drupal.

Para simplificar y clarificar el concepto de plataforma en nuestro caso, siempre que nos refiramos a plataforma o plataforma web a lo largo del documento, estaremos haciendo referencia a un sistema de gestión de contenidos. En el caso práctico de la migración del capítulo cuarto, esta plataforma será el CMS Joomla, en sus diferentes versiones.

- *Aplicación web*

Una aplicación es un programa informático que permite a un usuario utilizar un ordenador con un fin específico. Las aplicaciones son parte del software de un ordenador y suelen ejecutarse sobre el sistema operativo. Los objetivos de una aplicación pueden ser revisar correo, explorar el disco duro, editar textos, navegar en la web, jugar, etc. Son ejemplos de aplicaciones Mozilla Firefox, Outlook, DropBox, Gimp, etc.

Una aplicación web es aquella aplicación que los usuarios pueden utilizar accediendo a un servidor web a través de Internet o de una intranet mediante un navegador. En otras palabras, es una aplicación software que se codifica en un lenguaje soportado por los navegadores web, y en la que se confía la ejecución de la aplicación al navegador.

En nuestro caso, consideraremos una aplicación web como el programa que se ejecuta sobre la plataforma web utilizada, mediante la cual el usuario solicita información al servidor de información mediante acciones disponibles en la aplicación, logrando que éste obtenga la información solicitada. En el caso práctico del capítulo cuarto, la aplicación web a la que se hará referencia será el Sistema de Recomendación de Restaurantes de Jaén, denominado REJA.

2.4. Metodología de Migración basada en Tecnologías Web

Una vez definido el procedimiento de migración software genérico, en esta sección vamos a describir ese procedimiento concretando las distintas fases en un proceso de migración basado en tecnologías web. Debido a la escasa bibliografía genérica existente referente a este tipo de migración, nos hemos apoyado en la abundante documentación referente a metodologías y procesos de migración de software libre [4], [10], [11], [12], [13], y a la metodología de migración clásica revisada anteriormente, para definir y explicar cada una de las etapas de la metodología de migración basada en tecnologías web.

Una migración de un sitio web [14], es la transferencia de contenido, secciones, funcionalidad, equipo, plantillas, arquitectura de la información y las relaciones entre una plataforma y otra. Al igual que cualquier migración, debe de haber una razón para acometer esta tarea. Una fase importante en un proyecto de migración es la justificación de ésta, la explicación de los motivos por los que se va a realizar la migración.

La migración de un sitio web no consiste sólo en la migración del contenido de éste, si no en otras muchas tareas. En términos generales, antes de realizar la migración hay que tener en cuenta la disposición del contenido, las secciones, la funcionalidad, el equipo, las plantillas, la arquitectura de la información y las relaciones entre diferentes plataformas.

- a) *Contenido*: A menudo nos centramos en el contenido cuando nos referimos a la migración de un sitio web. Como hemos comentado anteriormente, no solo es la migración de contenido, pero si es cierto que es una gran parte del proceso y muy importante. Cuando se habla de contenido tendremos que tener en cuenta la estructura de los datos, su clasificación, la utilidad de toda la información, los medios automatizados para migrarlo, el mantenimiento de éste, etc.

- b) *Secciones*: Mucha información en Internet en la actualidad está compuesta por sitios que son resultado de la unión de varios sitios. En estos casos habrá que diferenciar cada sección, ya que la administración en cada uno puede ser distinta.
- c) *Funcionalidad*: Cuando se realiza la migración de un sitio web a otro nuevo, la funcionalidad del antiguo se debe mantener en el nuevo. Puede darse el caso que algunas funcionalidades anteriores no precisen ser migradas por falta de utilidad o por haber caducado, al igual que aprovechando la migración, se pueden añadir nuevas funcionalidades.
- d) *Equipo*: También hay que tener en cuenta el equipo relacionado con el sitio web. Este equipo está diferenciado en dos grupos. Por un lado las personas que se van a encargar de realizar la migración y por otro los usuarios que seguirán utilizando el sitio web.
- e) *Plantillas*: En muchos casos, cuando se realiza una migración, la plantilla que se venía utilizando puede ser totalmente válida para el nuevo sitio. Pero en otros casos esto no sucede, debido a incompatibilidad de versiones, diferentes lenguajes de programación, tecnologías, etc. En estos casos habría que diseñar una plantilla nueva o actualizar la anterior para que siga siendo válida.
- f) *Arquitectura de la Información*: Si la migración es debida a un cambio de plataforma, es probable que la arquitectura de la información también tenga que ser cambiada. En este caso habría que revisar cómo afecta este cambio de arquitectura a las diferentes páginas.
- g) *Relaciones*: Son los enlaces que existen entre los diferentes elementos, el contenido con la presentación, el sistema, usuarios, etc. Una migración no es simplemente

migrar cada elemento por separado, si no comprobar los enlaces que se producen entre los diferentes elementos.

Algunas tareas que se realizan sobre un sitio web se pueden confundir con un proceso de migración cuando en realidad no lo son. Algunas de estas tareas son, por ejemplo, la creación de un sitio web con nuevos contenidos, el rediseño gráfico del sitio actual, el desarrollo de un sitio que sólo agrega contenido de otros sitios o un nuevo sitio que elimina todo el sitio anterior conservando únicamente la marca.

En la Figura 2.5 se muestra el esquema detallado de la metodología de migración basada en tecnologías web. Esta metodología consta de una serie de tareas a desarrollar para considerar completado el proceso de migración. Estas tareas se agrupan en las fases de análisis de requerimientos, diseño, proceso de migración, evaluación y mantenimiento. A continuación, vamos a ir enumerando las distintas fases de las que se compone el proceso de migración y desarrollando cada una de ellas.

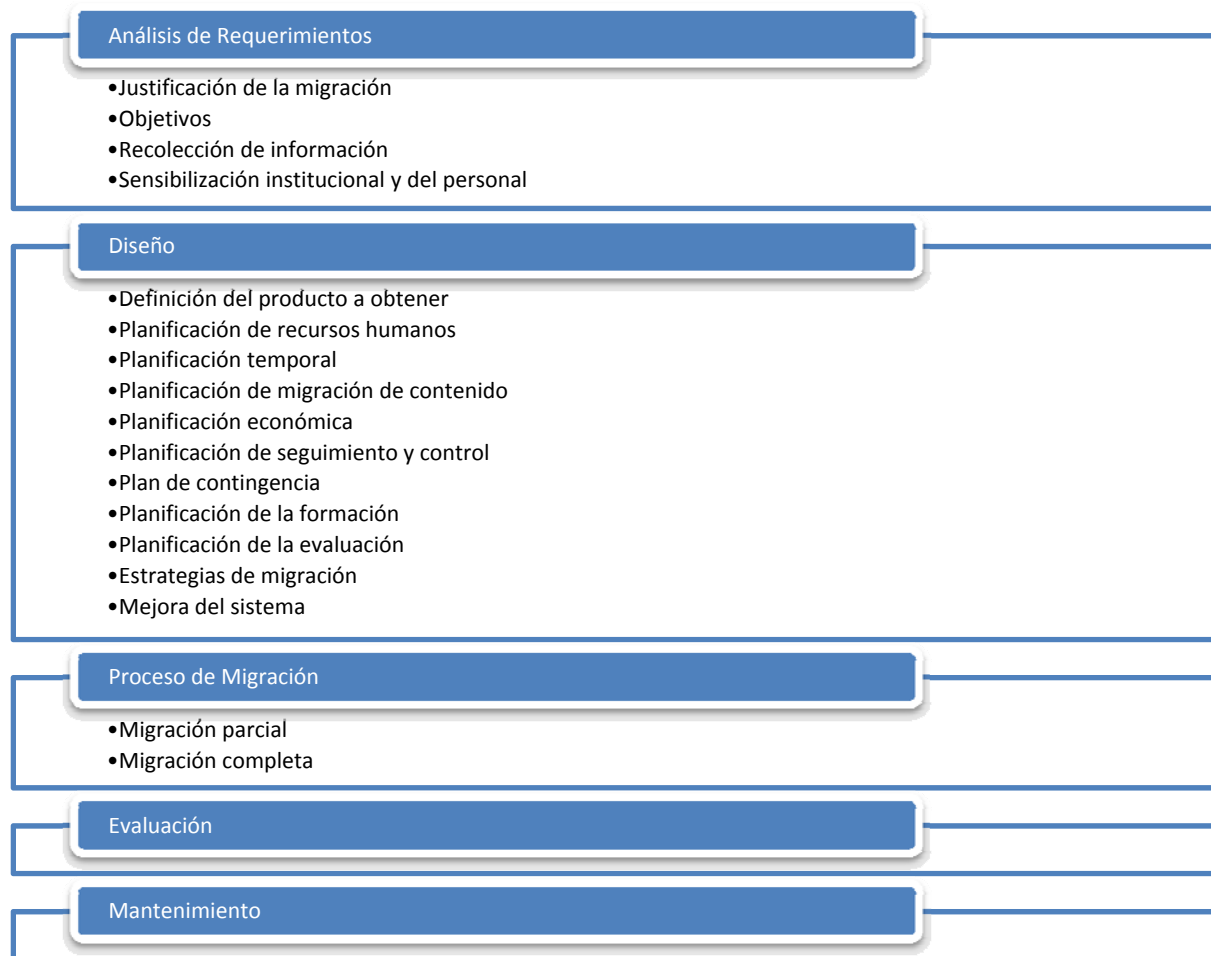


Figura 2.5. Esquema detallado de la metodología de migración basada en tecnologías web.

2.4.1. Análisis de Requerimientos

Es la fase inicial de todo proceso de migración (ver Figura 2.6). En esta fase hay que justificar los motivos por los cuales se va a llevar a cabo este proceso y los objetivos que se desean alcanzar. También hay que realizar una recolección de información necesaria del sistema actual para poder utilizarla en posteriores etapas del proceso. Por último, es necesario sensibilizar de la necesidad de llevar a cabo la migración para la mejora del sistema, tanto a los usuarios finales del mismo como a los propietarios. A continuación, vemos de forma más detallada cada una de estas actividades del análisis de requerimientos



Figura 2.6. El análisis de requerimientos es la primera fase de la metodología de migración basada en tecnologías web.

a) *Justificación de la migración*

En este apartado se justifican los motivos para la realización de la migración basándonos en la situación actual del sistema. Los motivos están relacionados con los objetivos que se desean conseguir, por ejemplo, si uno de los motivos es por problemas de seguridad en el sitio web, uno de los objetivos a conseguir con la migración debería ser mejorar la seguridad. Las razones más usuales para realizar una migración son la reducción de costes, ampliación de funcionalidades, mejora de productividad y mejora de seguridad. Esta justificación de la migración deberá ser facilitada a los usuarios finales del sitio para explicar los motivos de la migración.

b) *Objetivos de la migración*

Una tarea a realizar antes de la migración es fijar los objetivos de ésta. Estos objetivos nos servirán más adelante para evaluar el éxito de la migración. Durante el desarrollo del proceso, estos objetivos pueden verse alterados si nos encontramos con problemas técnicos u obstáculos, o incluso pueden modificarse al estudiar la tecnología más a fondo.

Algunos de los objetivos más comunes pueden ser los siguientes:

- *Mejorar la seguridad:* La mejora de la seguridad de un sistema lo hace más robusto y productivo. Uno de los objetivos principales de toda migración de un sitio web debe ser la seguridad.

- *Mejorar la productividad:* Un sitio web debe funcionar correctamente, pero además debe hacerlo de manera eficiente. Un sitio lento puede producir pérdidas de tiempo para sus usuarios, lo que conlleva pérdida de visitas y por tanto pérdida de clientes o usuarios.
- *Ampliar la funcionalidad:* Un sitio web nuevo desarrollado sobre tecnologías más actuales, puede disponer de nuevas funcionalidades que el sitio antiguo no cumplía o no podían ser implementadas, con lo que se conseguirá un sitio web más competitivo.
- *Reducción de costes:* Uno de los objetivos de una migración también puede ser la reducción de costes, debido al mantenimiento, licencias, hardware, etc.

c) *Recolección de información*

En esta fase se estudian todos los datos necesarios para empezar el proceso de migración, agrupándolos en los siguientes tres bloques:

- *Inventario del capital humano:* Se realiza un inventario sobre el personal que podrá dar soporte al sitio web tanto de asistencia a los usuarios como de administración del mismo. También sobre el personal disponible y cualificado para llevar a cabo la migración y la descripción de los perfiles de los usuarios finales.
- *Inventario hardware:* Se realiza un inventario del hardware disponible actualmente sobre el que está alojado el sitio web actual.
- *Inventario software:* Se realiza un inventario del software actualmente utilizado para la administración y gestión del sitio web actual, para compararlo con los

requerimientos necesarios para realizar la migración y dar soporte al sitio web tras la migración.

d) Sensibilización institucional y del personal

Para que un proyecto de migración en general y de un sitio web en particular tenga éxito, es muy importante que se tenga una visión convincente respecto de la migración por parte de los propietarios del sitio, de los encargados de realizar la migración y de los usuarios finales [14]. Una visión convincente se consigue explicando a los interesados las mejoras sustanciales de la migración. Si no se consigue que la mayoría lo vean como algo útil, la migración podría ser un fracaso. Por este motivo, este debe ser uno de los primeros pasos del proyecto de migración, incluso antes que el diseño y la implementación.

Para poder convencer de su bondad, realmente se tiene que entender así y dominar la materia. Para conseguir esto se puede proceder a realizar los siguientes pasos:

- Difundir ampliamente toda la información posible del proyecto.
- Seguir comunicando durante el desarrollo del mismo.
- Documentar las ventajas y desventajas de la migración.
- Definir indicadores para evaluar la visión durante la migración.

En algunos casos, para justificar la visión positiva o convincente al personal, se argumenta que es ordenado por la dirección y esto puede ser contraproducente. Una visión positiva de los usuarios finales sobre el proceso de migración ayudará al equipo de desarrollo y al proyecto para que garantice el éxito.

2.4.2. Diseño

Una vez realizado el análisis de requerimientos en el proceso de migración, se lleva a cabo la fase de diseño (ver Figura 2.7). En esta etapa se analiza y planifica todo el proceso de migración, tanto la temporalidad, el producto a obtener, la estrategia de migración, los recursos necesarios, tanto humanos como económicos, etc. Dichas actividades se describen a continuación.

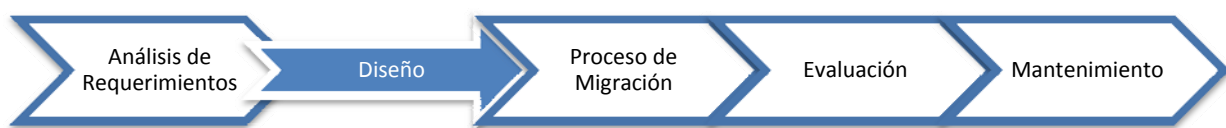


Figura 2.7. La fase de diseño es la segunda fase de la metodología de migración basada en tecnologías web.

a) *Definición del producto a obtener*

En este punto se define la funcionalidad y características que se desean conseguir del nuevo sistema. Antes de comenzar una migración de un sistema, se define cuál será el producto final a obtener. Este producto final contempla todos los objetivos descritos en la sección de análisis de requerimientos.

b) *Planificación de recursos humanos*

A la hora de hacer una migración, hay que tener en cuenta que no siempre se necesita el mismo número de personas encargadas de realizar la migración y su nivel de especialización con el proyecto. Depende de la complejidad del sistema web sobre el cual queremos hacer la migración, relacionado con la complejidad y el tamaño del sitio.

Como explicamos en el apartado 2.1.2 sobre el ciclo de vida software, la migración es sólo uno de los pasos de la larga vida del ciclo de vida software y en este caso de un sitio

web. Por ello, las personas encargadas de la migración puede que sólo participen en este paso a lo largo de la vida del sitio. Es necesario mantener con posterioridad a la migración, una estructura de personal encargada de la administración y mantenimiento del mismo.

A continuación, se definen los diferentes tipos de reglas u organizaciones que participan en la migración de un sistema web. Cada regla no tiene por qué ser realizada por una persona distinta, dependerá del tamaño y complejidad del sitio:

- *Director interno del proyecto*: encargado de la organización, gestión de la configuración y ejecución. Debe trabajar con los distintos grupos de trabajo y debe conocer exactamente qué se espera con la migración, definir exactamente el producto final.
- *Coordinadores*: están más relacionados con grandes proyectos. Coordinan entre el núcleo del equipo de migración y sus diferentes delegaciones, suponiendo que existe un núcleo desde donde se realiza gran parte de la migración y diferentes delegaciones desde donde se realizan partes concretas del proceso.
- *Especialistas en contenido*: una migración de un sistema web puede ser una oportunidad para mejorar el contenido. Dentro de los especialistas de contenido podemos diferenciar varios perfiles:
 - *Publicadores de contenido*: son los encargados de hacer la migración inicial de contenido desde el sistema anterior al nuevo, siempre que este proceso no sea automático.
 - *Editores y escritores*: son los encargados de hacer la parte mecánica de la migración de contenido, como copiar los contenidos del sitio antiguo y pegarlos en el nuevo.

- *Expertos en la materia:* durante la migración, dependiendo de las características del sitio que se esté intentando migrar, se puede necesitar de la ayuda o asesoramiento de expertos en la materia sobre la que el sitio esté basada. Estos expertos resolverán dudas o problemas concretos específicos de una materia, la cual el equipo encargado de la migración puede que no domine.
- *Otros:* Hay una serie de perfiles que puede ser que haya que asignarlos a personas concretas en algún proyecto de migración pero que no tiene por qué tener que asignarse siempre, dependiendo de la complejidad del sitio, el tamaño, etc. Estos perfiles pueden ser los siguientes:
 - *Gerente del proyecto:* aunque el director interno del proyecto puede desempeñar este puesto, se podría necesitar en algún caso alguna persona externa que busque el cumplimiento específico con la agenda prevista y el presupuesto.
 - *Desarrolladores:* siempre se va a necesitar al menos un desarrollador en el proceso de migración. Lo que si variará será qué tipo de desarrolladores en concreto se necesitarán dependiendo de la tecnología o lenguaje con el que está desarrollado tanto el sitio antiguo como el nuevo. Los desarrolladores pueden ser personal externo o personal propio capacitado dentro de la organización.
 - *Formadores:* son los encargados de mostrar el nuevo sitio a los usuarios y formarlos en su uso y funcionalidades.

c) *Planificación temporal*

La planificación temporal de la migración es útil para organizar las tareas englobadas en el proceso. Para cada una, tendremos que especificar su duración y su fecha de comienzo. También se establecen plazos de entrega y una fecha para la finalización de la migración. Para poder hacer una estimación sobre la temporalidad en la migración total, podemos realizar la migración parcial y con ella estimar la temporalidad de la migración total. Una estrategia de migración parcial se podría dividir en la tarea de copias de seguridad parciales, migración parcial del sitio web y evaluación de la migración parcial. Con estos datos reales se podría estimar la temporalidad de la migración completa.

Para hacer la migración del sitio web, hay que definir cuáles son las fechas más interesantes para realizarla, de modo que cause el menor impacto posible para los usuarios ante una posible caída de la web, o un periodo sin ofrecer el servicio óptimo por encontrarse en pruebas. Esta puede ser realizada durante un cierre vacacional si se tratase de una empresa o bien durante las fechas que las estadísticas del sitio web actual nos muestren que tiene un menor número de visitas respecto al total.

La fecha prevista de disponibilidad del nuevo sitio deberá ser facilitada a los usuarios finales una vez que esté planificada, al igual que las fechas en las que el sistema pueda no ofrecer el servicio óptimo a causas de la migración.

d) *Planificación de migración de contenido*

En esta fase se estudia el contenido que actualmente hay en el sitio web para posteriormente realizar su migración. La migración de contenido se divide en las siguientes fases:

- *Diferenciar el contenido*: El primer paso será analizar el contenido para poder diferenciarlo en diferentes tipos. Se puede tener un sitio en el que el contenido disponible esté de forma ordenada, por lo que será fácil de identificar, o por el contrario se tiene que averiguar los tipos de contenidos.
- *Dividir el contenido*: Una vez que el contenido está separado por tipos, es más fácil realizar la tarea de suprimir información mediante el método de análisis ROT (Redundant, Outdated and Trivial Information) [14]. Mediante este análisis, prescindimos de una parte del contenido que actualmente aparece en el sitio a migrar basándonos en una serie de criterios:
 - *Redundancia*: la información que aparece redundante en el sitio habría que suprimirla para no tener la misma información por duplicado.
 - *Contenidos caducados*: hay cierta información que puede estar caducada. Significa que cuando fue subida al sitio tenía utilidad pero que en el momento de realizar la migración no tiene ningún interés mantenerla.
 - *Cualquier necesidad de calidad*: este factor se aplicará para eliminar cualquier información que no supere un cierto criterio de calidad previamente definido.
- La migración de contenido se puede realizar de manera automática o manual. Para realizarlo de manera automática existen diferentes aplicaciones software que ayudan en esta tarea, o bien se puede desarrollar una específica para llevar a cabo la tarea. En el tercer capítulo se describen varias herramientas para automatizar el proceso de migración de contenido en un gestor de contenido. En el caso de no poder disponer de ninguna herramienta para la migración de contenido, habrá que realizarla de manera manual.

e) Planificación económica

La estimación de costes debe llevarse a cabo para establecer qué inversiones y recursos van a ser necesarios. Qué inversiones dependiendo del trabajo a realizar, y qué tiempo dependiendo de la intensidad de trabajo, deben ser diferenciadas en este contexto. Los costes de la migración se pueden clasificar en dos grupos:

- *Costes directos*: son el conjunto de costes que se pueden tener en cuenta en la fase de planificación. Entre ellos se pueden incluir las licencias del software, los costes hardware, los de soporte, formación y de personal.
- *Costes indirectos*: son aquellos que aunque desde un principio podemos preverlos, no podemos tasarlos, ya que dependen de la realización del proceso de migración y de los errores que se produzcan en dicho proceso. Entre ellos podemos incluir los costes de mantenimiento, de seguridad, de disponibilidad, de inoperatividad del sistema y de escalabilidad.

f) Planificación del seguimiento y control

Una vez realizada la planificación temporal, donde dividimos la migración en pequeñas tareas y las programamos en el tiempo, es conveniente establecer un seguimiento de la planificación para comprobar que se cumplen los plazos y detectar posibles desviaciones que ponen en riesgo la migración. La manera más simple es establecer una serie de hitos puntuales a lo largo de la migración que nos sirvan para monitorizar el progreso de la misma.

El objetivo primordial del seguimiento y control es tener controlado en todo momento el proyecto para poder detectar cualquier desviación en la planificación, como podría ser un retraso en alguna de las tareas y poder tomar las acciones correctivas

necesarias. En estas situaciones puede ayudar un plan de contingencia que muestre las acciones a llevar a cabo para acelerar la migración o mitigar sus consecuencias.

Otro de los objetivos del plan de seguimiento es verificar que los cambios realizados en el sistema sean correctos. En lo referente a los tipos de pruebas que podemos realizar durante la migración nos encontramos con:

- *Pruebas de corrección*: Sirven para testear el correcto funcionamiento del sistema.
- *Pruebas de seguridad*: Realizar pruebas para comprobar que el sistema sigue siendo seguro.
- *Pruebas de rendimiento o carga*: El correcto funcionamiento del sistema no es el único requisito necesario para conseguir un sistema de calidad. Por lo tanto, tendremos que asegurarnos de que el sistema funciona de manera fluida.
- *Pruebas de usabilidad*: Es importante que se siga manteniendo la usabilidad del sitio.

g) Plan de contingencia

En el caso de ser un proyecto de envergadura, en la que su alta complejidad provoca que tengamos que tratar con gran cantidad de variables, es importante diseñar un plan de contingencia ante problemas y fallos potenciales. Este plan deberá establecer una serie de acciones ordenadas y prefijadas que ayuden a corregir una posible incidencia.

Aunque realmente la experiencia de la persona encargada de la migración es una de las principales fuentes de diagnóstico de futuros posibles problemas, es fundamental realizar un esfuerzo para estimar los peligros potenciales de la migración y diseñar un conjunto de procedimientos para atenuarlo o solventarlo.

Como mínimo hemos de diseñar una estrategia para poder revertir el proceso de migración y volver a restaurar el antiguo sistema. Para ello se aconseja la implantación del nuevo sitio en un servidor o carpeta distinta para poder volver al sistema anterior en caso de necesidad y la realización de copias de seguridad.

Hacer copias de seguridad es una tarea vital en el proceso de migración de un sitio web. Es muy importante realizar copias de seguridad periódicamente de los sitios web, pero más aún cuando se va a realizar una migración. Estas copias nos servirán para restaurar nuestro sistema antiguo si el proceso de migración fallase en algún momento, al igual que también nos serán útiles a la hora de la migración. Hay que tener en cuenta que estas copias han de estar actualizadas al estado justo anterior a la migración para no perder información en el proceso sobre posibles cambios. También se debe comprobar que las copias son válidas realizando alguna simulación con ellas para ver que cumplen con su finalidad y que no existen problemas que nos puedan surgir en un futuro.

h) Planificación de la formación

Se debe planificar la realización de un plan de formación tanto para los administradores del sitio web como para los usuarios finales. Por muy completo o potente que sea el nuevo sitio web, si los usuarios finales no reciben la formación necesaria o no se han habituado a trabajar con el nuevo sitio, éste producirá rechazo. Por tanto, es muy importante valorar el plan de formación y llevarlo a la práctica. Este plan puede llevar incluido cursos de formación para los usuarios y los administradores, manuales, jornadas, etc.

i) Planificación de la evaluación

Antes de realizar cualquier migración hay que definir una serie de pruebas que se realizarán al sistema una vez migrado. Para realizar una evaluación adecuada del sistema,

estas pruebas no pueden ser improvisadas. Para evitar esta improvisación, en la fase de diseño en la que nos encontramos, se definen todas las pruebas a realizar en la fase de evaluación, una vez llevada a cabo la migración.

j) Estrategias de migración

Al inicio de este capítulo, en la sección del marco conceptual sobre la migración software, se definían cuatro esquemas de migración utilizados de manera general en los procesos de migración. En el caso concreto de migraciones de sitios web, se pueden definir principalmente las siguientes estrategias de migración:

- Migración automatizada
- Migración manual
- Migración parcialmente automatizada

De estas tres estrategias de migración la opción más deseada es la primera, la migración totalmente automática, pero en la práctica existen muy pocos casos en la que ésta se pueda realizar por completo. Por lo general, la opción más utilizada será la migración manual y en algunas partes o contenidos la automatización de ésta [15].

A continuación describimos cada una de estas estrategias:

- *Migración automatizada*

Esta estrategia es preferible a la hora de migrar del sitio antiguo al nuevo. Necesita poco esfuerzo, por lo que se reducirá el tiempo y los recursos necesarios para llevarla a cabo.

Algunos enfoques con los que esta estrategia se podría llevar a cabo serían los siguientes:

- Exportar el contenido del antiguo sistema e importarlo directamente en el nuevo. Este proceso requeriría el desarrollo personalizado para conectar entre sí la exportación con la importación.
- Utilizar las interfaces de programación de aplicaciones (API) que proporciona el sistema antiguo y el nuevo, y escribir un programa para transferir el contenido.
- Utilizar una herramienta de migración de terceros, que proporcione herramientas y normas para el proceso de migración.

Independientemente del enfoque adoptado, hay una serie de requisitos previos antes de la migración automatizada para que pueda ser válida:

- El sitio actual tiene que tener contenido de calidad, de modo que si éste no lo es, en la migración automatizada se pasaría directamente el contenido sin calidad al nuevo sitio.
- El sitio nuevo tiene que estar estructurado de manera similar al sitio anterior.
- El código HTML del sitio antiguo debe ser lo suficientemente limpio y consistente para permitir la migración automatizada.

Esta estrategia no es una opción válida para llevarla a cabo en la mayoría de los casos, ya que es muy frecuente que el sitio a migrar no tenga la suficiente calidad, sea inconsistente o no esté estructurado.

- *Migración manual*

La migración manual es la más simple pero a su vez la que más tiempo y recursos necesita. En la práctica es copiar el contenido del sitio antiguo y pegarlo en

el nuevo. Aunque es una estrategia muy laboriosa, la migración manual ofrece la oportunidad de revisar y limpiar el contenido del sitio, ya que muchas veces éste es uno de los objetivos principales a la hora de reconstruir un sitio.

- *Migración parcialmente automatizada*

Si bien, como se ha comentado anteriormente, la migración automatizada no es posible aplicarla en la mayoría de los casos, existe la posibilidad de realizarla en ciertas secciones del sitio para reducir la cantidad de trabajo manual requerido. Si un sitio tiene un contenido no estructurado o de baja calidad y no permite la migración automatizada, siempre se podrá buscar las secciones de contenido que si cumplan los requisitos y se puedan automatizar.

k) Mejora del sistema

Aprovechando el proceso de migración, es una buena oportunidad el realizar mejoras sobre el sistema. Como el proceso de migración requiere de un conocimiento en profundidad del sistema a migrar, se puede aprovechar este proceso para añadir nuevas funcionalidades o para corregir pequeños problemas que hayan sido detectados en el sistema antes de la migración.

2.4.3. Proceso de Migración

Tras realizar y documentar las fases de análisis y diseño, hay que llevar a cabo el proceso de migración (ver Figura 2.8). Esta fase puede dividirse en varios procesos, realizando migraciones parciales inicialmente y a continuación realizar la migración completa.



Figura 2.8. El proceso de migración es la tercera fase de la metodología de migración basada en tecnologías web.

2.4.3.1. Migración Parcial

Una migración parcial hace referencia a la migración de solo una parte del sitio web a migrar. Las tareas a realizar en una migración parcial se pueden agrupar en las fases de planificación de la migración parcial, ejecución y pruebas (ver Figura 2.9). Los motivos por los cuales se lleva a cabo este tipo de migración, están explicados en los objetivos de la fase de planificación de la migración parcial. No es necesario documentar de nuevo en esta fase todas las tareas de análisis de requerimientos y diseño definidas en las etapas anteriores del proceso general de migración, ya que al ser una migración parcial, únicamente se va a migrar una parte del sistema.

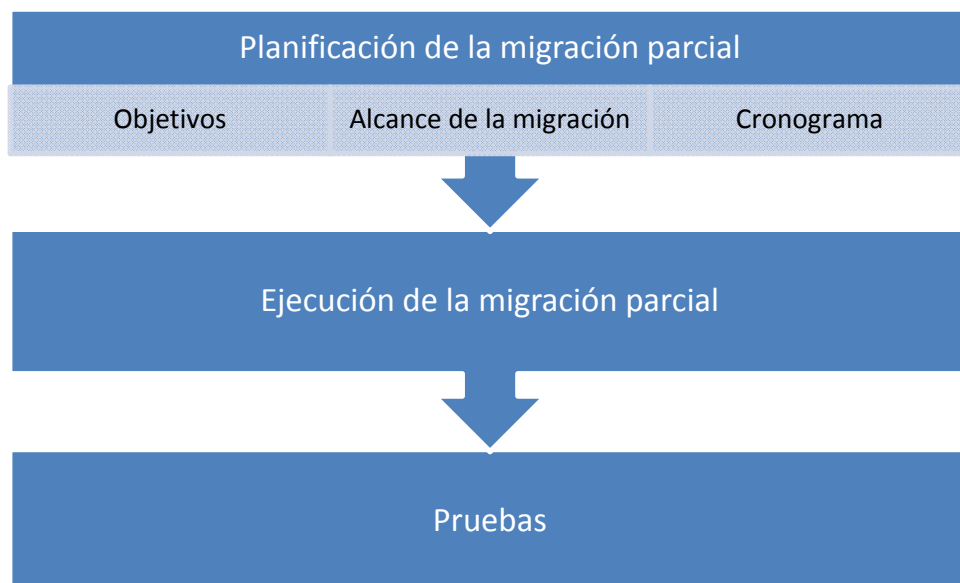


Figura 2.9. Fases de la migración parcial dentro del Proceso de Migración

a) *Planificación de la migración parcial*

La fase de planificación en la migración parcial contiene la definición de los objetivos a conseguir con esta migración, el alcance de la aplicación de las pruebas y su cronograma.

- **Objetivos:** El motivo de realizar una migración parcial en vez de ejecutar directamente la migración completa al nuevo sitio web se puede argumentar mediante los siguientes objetivos:
 - Prueba hacia la opinión convincente del equipo: Si directamente se realiza la migración sobre el nuevo sitio web, puede ser muy probable que los errores se sucedan de manera maximizada. Primero se debe confirmar que la estrategia general a seguir, incluida la tecnología, la configuración, los procesos y el personal, están de acuerdo con la visión inicial del proyecto. Durante esta fase previa a la migración total hacia el sitio definitivo, se tiene aún la posibilidad de reconfigurar algunas especificaciones iniciales que se detecten que en la práctica no son correctas o incluso añadir o descartar algunos requisitos preestablecidos. La creación del sitio en pruebas también puede servir para mostrar a algunas personas que no hayan alcanzado a tener una visión convincente del proyecto. De este modo, con un ejemplo práctico del proceso o resultado similar al final, puede ser más fácil alcanzar esta visión convincente.
 - Prueba de prácticamente todos los aspectos o funcionalidades de la migración: Una prueba piloto es un proceso de migración de un sitio web, no es sólo un producto de usar y tirar. Sobre este sitio piloto se pueden probar prácticamente todas las funciones o aspectos que se llevarán a cabo después, en el proceso de migración completo. Sobre el sitio piloto podemos probar la automatización de la migración, el proceso manual de la migración, la

integración con otros sistemas, la reescritura de los enlaces web y los problemas que surjan con la migración del contenido.

- Comentario y opiniones: La migración parcial generará un sitio piloto que pueda ser consultado por diferentes personas, para poder obtener una retroalimentación con comentarios sobre posibles mejoras u otros aspectos a tener en cuenta antes de realizar la migración definitiva. En particular, este proyecto piloto debe poder ser consultado por los actuales usuarios del sitio web, por los publicadores de contenido, tanto editores, traductores y otros perfiles clave con acceso al sitio, y el equipo técnico y personal involucrado en el proceso de migración. Durante este periodo se recibirán comentarios sobre el sitio piloto, por lo que en la fase de planificación se debe dejar tiempo suficiente en el cronograma del proyecto de migración para subsanar o modificar todos los cambios que se hayan visto necesarios en esta fase.
- Estimación de los niveles de esfuerzo: Otro aspecto muy importante a tener en cuenta en el proceso de migración, es la estimación del esfuerzo necesario para poder realizarla. En esta estimación se pretende saber cuáles son los tipos de contenidos y las funcionalidades necesarias para la implementación del sitio y con esta información, concretar cuánto esfuerzo de personal técnico y no técnico será necesario en cada caso.
- ***Alcance de la migración parcial:*** En la planificación de la migración parcial se debe especificar claramente qué partes del proyecto global van a ser integradas o probadas y cuáles no. En concreto, hay que poner atención en las áreas de contenido, funcionalidad, integración e interfaces. Junto con esta especificación, se debe hacer un seguimiento concreto sobre el esfuerzo que en la práctica conlleva realizar cada fase del proyecto piloto, ya que podrá ser extrapolado para obtener el esfuerzo total requerido en la migración completa del sitio.

- **Cronograma:** A la hora de definir el cronograma para llevar a cabo el proceso de realización de la migración parcial, hay que tener en cuenta dos aspectos básicos para su correcta definición:
 - Como uno de los objetivos es encontrar puntos de mejora y rediseño a la hora de realizar la migración total, es muy importante en la planificación del cronograma tener en cuenta el tiempo necesario para realizar dichos cambios. A priori no es posible saber exactamente de qué temporalidad se trata, ya que esto lo definirá la cantidad de peticiones o comentarios realizados, pero si no se cuenta con el tiempo necesario para realizar los cambios porque no se ha previsto de tiempo suficiente en el cronograma, uno de los objetivos principales de esta fase no habrá sido superado.
 - Hay que tener claro la consideración del alcance de la migración parcial. No se podrá considerar la velocidad de desarrollo del proyecto piloto para extrapolarlo a la migración del sitio total, considerándolo como una pequeña parte de éste, pero sí es válido como prueba real de las tecnologías necesarias a utilizar, los diferentes procesos y las conclusiones que sean obtenidas.

b) Ejecución de la migración parcial

Es en este momento cuando se lleva a la práctica la planificación realizada en el apartado anterior. Se debe tener en cuenta tanto los objetivos, el alcance de la migración y el cronograma propuesto.

c) Pruebas

Una vez que la migración parcial del sitio web se ha llevado a cabo, hay que evaluar el sitio creado con la migración parcial para detectar posibles cambios que haya que realizar sobre el plan de migración. Si se ha detectado algún problema o error en el plan de migración, es el momento de redefinir este plan para evitarlos en la migración completa y de este modo afrontar con garantías el proceso completo de migración.

Además de evaluar los aspectos o funcionalidades de la migración, hay que evaluar igualmente la estimación de los niveles de esfuerzo y los comentarios y opiniones sobre el proceso.

2.4.3.2. Migración Completa

En la migración parcial, al realizar la migración sobre una parte del sitio web, se han definido una serie de fases y tareas concretas sólo para esa migración parcial. La migración completa sí realiza el proceso de migración en su totalidad. Por tanto, las diferentes acciones a llevar a cabo en la migración completa, son las descritas anteriormente en las fases de análisis de requerimientos y diseño (ver Figura 2.10). En estas fases ya han sido descritos los objetivos y tareas a realizar.

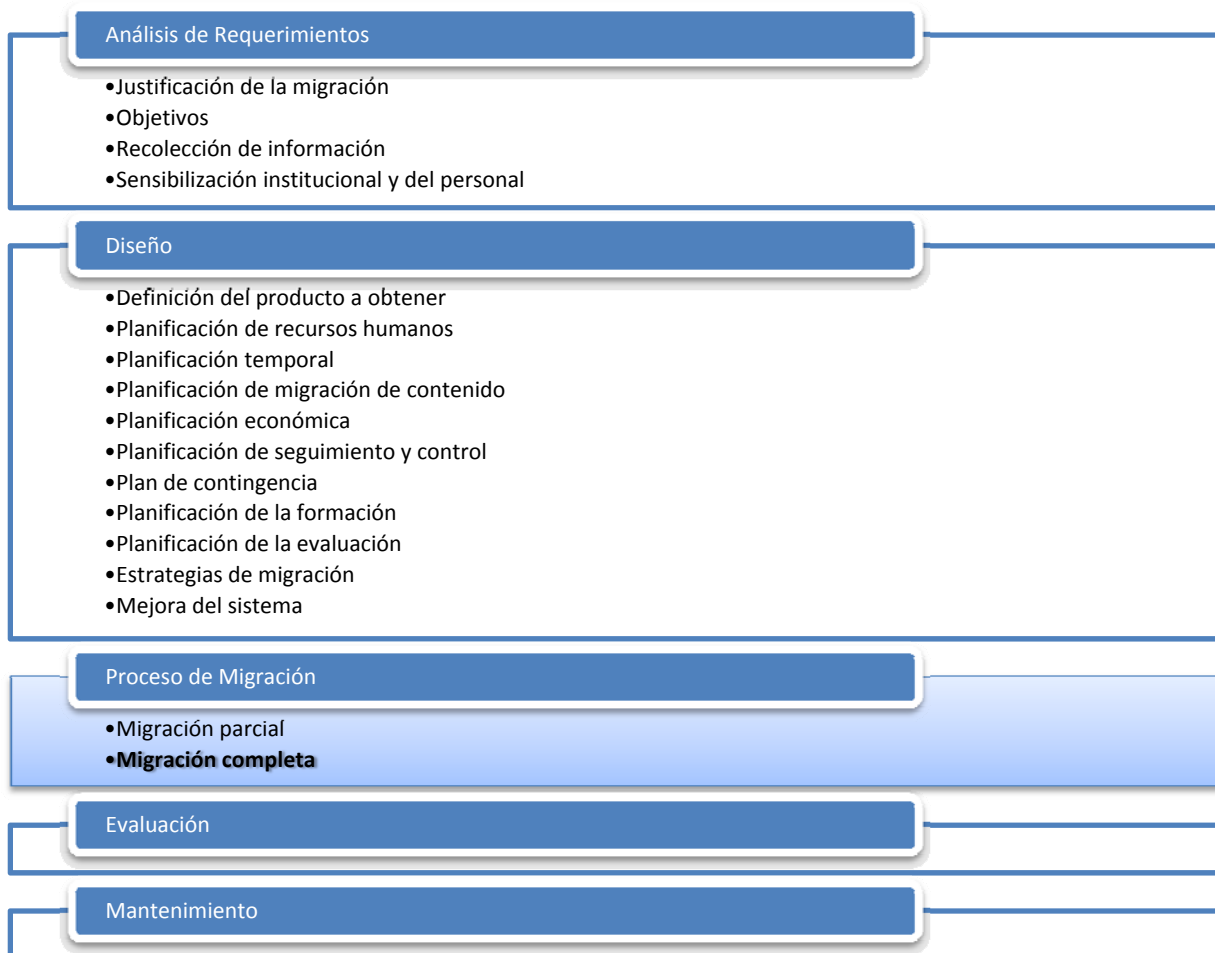


Figura 2.10. Esquema de la metodología de migración

Cuanto más detallada y profunda haya sido la fase de análisis de requerimientos y de diseño, mayor probabilidad de éxito y menor número de incidencias se encontrarán en esta fase. A continuación se describen las tareas que se ejecutan en esta fase, indicando la secuencia que debe llevar:

- Antes de realizar cualquier tipo de tarea, habrá que llevar a cabo una copia de seguridad del sistema para poder revertir el proceso de migración y volver al estado inicial en caso de ser necesario por muy diversos motivos. Esta copia de seguridad está definida en el *plan de contingencia en la fase de diseño*.

- Una vez realizada la copia de seguridad, se realiza la limpieza del contenido del sitio web que se desea migrar, siguiendo la planificación realizada en la *planificación de contenido en la fase de diseño*. Se comentaba la necesidad de dividir el contenido del sitio web para prescindir del que no fuese necesario migrar debido a su redundancia, a su caducidad o a cualquier necesidad de calidad.
- El siguiente paso es la ejecución de la estrategia de migración descrita en la fase de diseño, con la cual se obtendrá como resultado el sistema ya migrado. Todos los pasos y tareas realizadas en esta etapa dependen de la estrategia seleccionada y el tipo de sistema migrado. Todos estos pasos deben ser documentados, indicando las diferentes incidencias presentadas o anotaciones que se crean convenientes.
- Una vez realizada la migración se puede llevar a cabo una serie de mejoras del sistema si han sido definidas en su correspondiente etapa en la fase de diseño.

2.4.5. Evaluación

Al llegar a esta fase del proceso, la migración ya ha sido ejecutada, pero el proceso de migración no se puede dar por finalizado. Es en este momento cuando se lleva a la práctica diferentes planificaciones realizadas. Una de las tareas en la fase de planificación, era la *planificación del seguimiento y control*. Esta tarea se va realizando sobre el sistema durante la ejecución de la migración, pero es en la fase que nos encontramos cuando se evalúan los resultados obtenidos. Las diferentes acciones a realizar para medir el nivel de cumplimiento de los objetivos iniciales propuestos y el correcto funcionamiento, han sido definidas en la planificación de la evaluación en la fase de diseño. En este proceso se llevan a la práctica las pruebas indicadas para corroborar el éxito de la migración o subsanar los errores encontrados (ver Figura 2.11).



Figura 2.11. La cuarta fase de la metodología de migración basada en tecnologías web es la evaluación.

En esta fase también se debe comprobar que se ha cumplido la definición del producto a obtener, la planificación económica, temporal, etc.

2.4.6. Mantenimiento

Una vez finalizada la migración y comprobado que se ha realizado con éxito mediante la evaluación, es el momento de definir y ejecutar una serie de tareas para que el sitio web siga siendo funcional y actualizado, y no caiga en desuso por falta de mantenimiento. Es una tarea que puede conllevar mucho trabajo, dependiendo del sitio, pero es muy necesaria (ver Figura 2.12).

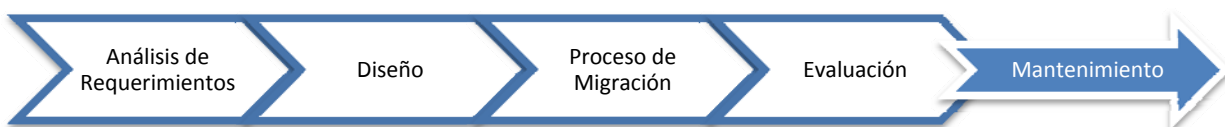


Figura 2.12. El mantenimiento es la última fase de la metodología de migración

Una de las tareas es definir la persona o personas que asumen a partir de ese momento la administración del sitio, con todas las tareas que ello conlleva, como pueden ser la actualización de contenidos, la realización de copias de seguridad, la seguridad del sitio, etc.

Otra tarea a realizar es ejecutar el plan de formación definido en la fase de diseño. Por muy completo y potente que sea el nuevo sitio, si los usuarios finales no reciben la formación necesaria sobre nuevos usos o nuevas funcionalidades del nuevo sitio, éste podría producir rechazo con la consecuente pérdida de interés por el sitio que llevaría a bajar el número de usuarios que lo utilicen. Por tanto, es muy importante llevar a la práctica el plan de formación realizado.

Capítulo 3

Proceso de Migración para un CMS

En este capítulo, se realiza una propuesta de metodología de migración aplicada a un sistema de gestión de contenidos (CMS). Para entender qué es un CMS, en la primera parte se define el concepto de sistema gestor de contenidos, se realiza una breve introducción a su historia y se describen sus ventajas respecto a otros sistemas de desarrollo web. En la segunda parte, se realiza la propuesta de metodología, adaptando el proceso de migración al caso concreto de CMS.

3.1. Presentación del Problema

3.1.1. ¿Qué es un CMS?

Un sistema de gestión de contenido (Content Management System, en inglés, abreviado CMS) es un software que se utiliza para facilitar la gestión de sitios Web, ya sea en Internet o en una Intranet. Consiste en una interfaz que controla una o varias bases de datos donde se aloja el contenido del sitio, y que además permite independizar el contenido del diseño. De esta forma, es posible administrar el contenido y darle al sitio un diseño distinto en cualquier momento y sin tener que darle formato al contenido nuevamente [16].

Este sistema, mediante una interfaz gráfica normalmente intuitiva, permite generar código en lenguaje de programación, sin necesidad de programar directamente ninguna línea. Los CMS utilizan una base de datos para almacenar el contenido (los menús, el texto y los elementos dinámicos) y una plantilla que servirá para definir los aspectos de diseño del sitio. El resultado final es un sitio web totalmente funcional y sobre el que se tendrá el control tanto en los contenidos como en el proceso de diseño.

En un CMS a nivel de estructura se distinguen 3 capas.

- *Capa de la base de datos*: La base de datos tiene almacenado todo el contenido que se ha escrito en la web, así como muchos de los parámetros de configuración,

categorías, organización, usuarios y contraseñas. Los sistemas de bases de datos más habituales de los CMS suelen ser MySQL (www.mysql.com) o PostgreSQL (www.postgresql.org).

- *Capa de programación:* Está contenida en los ficheros de la web. Estos ficheros, al ejecutarse, solicitan la información que el usuario ha pedido desde el navegador de Internet y la extraen para mostrarla al usuario ordenada y estructurada, insertándola en los lugares que le corresponde dentro del diseño de la página web. El lenguaje de programación más habitual suele ser PHP (www.php.net).
- *Capa de diseño:* Reside también en algunos ficheros. Define el diseño de la web sobre el que se insertará el contenido que la programación se encarga de extraer de la base de datos. El lenguaje de programación y maquetación de la web suele ser HTML y CSS, complementado en ocasiones con javascript y AJAX.

En la Figura 3.1 se muestran los logotipos de los CMS más conocidos y utilizados en la actualidad.



Figura 3.1. Logotipos de los CMS más utilizados en la actualidad.

En un CMS se diferencian dos partes fundamentales, la parte pública y la parte privada:

- *Parte web pública:* Es la página web como tal que ven las personas que acceden a ella (ver Figura 3.2).



Figura 3.2. Parte pública o frontend de un sitio web en Drupal

- *Parte web privada:* Desde cualquier lugar con conexión a Internet y un navegador con un usuario y contraseña podemos acceder a la parte interna del gestor de contenidos (ver Figura 3.3). Desde esta parte, según el perfil que tengamos, se podrá actualizar el contenido de la página web pública, modificarlo o crear contenido nuevo. Si se tiene el nivel de permisos adecuado para ello, se podrá también, desde un panel de control, cambiar configuraciones del sitio web, el diseño, gestionar a los usuarios del sistema, etc.



Figura 3.3. Parte privada o backend de un sitio web en WordPress

A continuación exponemos algunas de las características más importantes que hacen útil y necesaria la utilización de un CMS:

- *Inclusión de nuevas funcionalidades en la web:* Esta operación puede implicar la revisión de multitud de páginas y la generación del código que aporta las funcionalidades. Con un CMS eso puede ser tan simple como incluir un módulo realizado por terceros, sin que eso suponga muchos cambios en la web. El sistema puede crecer y adaptarse a las necesidades futuras.
- *Mantenimiento de gran cantidad de páginas:* En una web con muchas páginas hace falta un sistema para distribuir los trabajos de creación, edición y mantenimiento con permisos de acceso a las diferentes áreas. También se tienen que gestionar los metadatos de cada documento, las versiones, la publicación y caducidad de páginas y los enlaces rotos, entre otros aspectos.

- *Reutilización de objetos o componentes:* Un CMS permite la recuperación y reutilización de páginas, documentos, y en general de cualquier objeto publicado o almacenado.
- *Páginas interactivas:* Las páginas dinámicas o interactivas no existen en el servidor tal como se reciben en los navegadores, sino que se generan según las peticiones de los usuarios, ya que el contenido de éstas se encuentra en una base de datos. De esta manera cuando por ejemplo, se utiliza un buscador, el sistema genera una página con los resultados que no existían antes de la petición.
- *Cambios del aspecto de la web:* Si no hay una buena separación entre contenido y presentación, un cambio de diseño puede implicar la revisión de muchas páginas para su adaptación. Los CMS facilitan los cambios con la utilización, por ejemplo, del estándar CSS (Cascading Style Sheets u Hojas de Estilo en Cascada) con lo que se consigue la independencia de presentación y contenido.
- *Consistencia de la web:* La consistencia en una web no quiere decir que todas las páginas sean iguales, sino que hay un orden (visual) en vez de caos. Un usuario nota enseguida cuándo una página no es igual que el resto de páginas de la misma web por su aspecto, la disposición de los objetos o por los cambios en la forma de navegar. Estas diferencias provocan sensación de desorden y dan a entender que la web no la han diseñado profesionales. Los CMS pueden aplicar un mismo estilo en todas las páginas con el mencionado CSS, y aplicar una misma estructura mediante patrones de páginas.
- *Control de acceso:* Controlar el acceso a un sitio web no consiste simplemente en permitir la entrada a la web, sino que permite gestionar los diferentes permisos a cada área del sitio aplicados a grupos o individuos.

Además de las características de los CMS, es importante conocer la clasificación de CMS de acuerdo al tipo de información que publican:

- *Gestión de portales*: Sirven para gestionar el contenido de un sitio web de diversa funcionalidad. Uno de los más populares y conocidos es Joomla (www.joomla.org), aunque existen muchos más como Drupal (www.drupal.org), Xoops (www.xoops.org), etc. En la web <http://php.opensourcecms.com> hay disponible información sobre los diferentes tipos de CMS de código libre (ver Figura 3.4).



Figura 3.4. Ejemplo de gestor de portales

- *Blogs*: Publican artículos en orden cronológico, son periódicamente actualizados por uno o más autores y permiten ser comentados por usuarios registrados o anónimos. Dos de los gestores de Blogs más conocidos son WordPress (www.wordpress.com) y Blogger (www.blogger.com) (ver Figura 3.5).



Figura 3.5. Ejemplo de blog

- *Gestión de foros*: Permiten discusión en línea de usuarios en torno a un tema de interés común. Entre los más populares están phpBB (www.phpbb.com) y SMF (www.simplemachines.org) (ver Figura 3.6).

| Categorías: Temas, Respuestas, Regios, Ayuda, Buscar | | | |
|---|--|------------|------------------|
| Bienvenido, Invitado - Crear cuenta | | | |
| <input type="text" value="Nombre de Usuario"/> <input type="text" value="Nombre de Usuario olvidado?"/> <input type="text" value="Contraseña olvidada?"/> | | | |
| <input type="button" value="Entrar"/> <input type="button" value="Registrarse"/> | | | |
| Plantillas Kunena | | | |
| Categorías del Foro | | | |
| Plantillas Kunena | | | |
| | Plantillas version Kunena Sugerencias y errores para las plantillas Kunena | 4 Temas | 32 Respuestas |
| | Plantillas version Kunena JNM Edition Sugerencias y errores para las plantillas Kunena JNM Edition | 1 Temas | 0 Respuestas |
| | Plantillas Kunena 1.6 | 0 Temas | 0 Respuestas |
| Aportes y Peticiones | | | |
| | Aportes Entra tus aportes y comentarios con los demás. | 0 Temas | 0 Respuestas |
| | Peticiones Si no encuentras un diseño que te guste pide uno personalizado. Resueltas con es gratuito así que es barato y no nada malo. | 0 Temas | 0 Respuestas |
| Tu Web | | | |
| | Muestranos tu Web Enlaza tu Web y tu foro y pídanos críticas y consejos para mejorar su Diseño. | 2 Temas | 2 Respuestas |
| | SEO Cuestiones y debates de SEO, SEO de la página web, pide ayuda y consejos sobre como mejorar. | 2 Temas | 0 Respuestas |

Figura 3.6. Ejemplo de gestión de foros

- *Wikis*: Permiten crear contenidos de manera colaborativa. Sus páginas pueden ser editadas por múltiples usuarios y, a diferencia de los blog, la información no está

necesariamente ordenada cronológicamente. El ejemplo más destacado es la Wikipedia. Otros gestores populares son MediaWiki (www.mediawiki.org) y TikiWiki (www.tiki.org) (ver Figura 3.7).

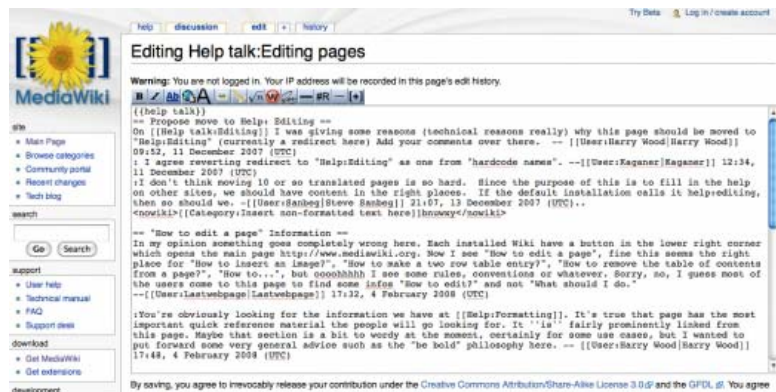


Figura 3.7. Ejemplo de wiki

- *Gestión de Comercio Electrónico*: Permiten generar sitios web específicos para el comercio electrónico, como por ejemplo osCommerce (www.oscommerce.com) (ver Figura 3.8).



Figura 3.8. Ejemplo de gestión de comercio electrónico

- *Gestión de E-Learning*: A estos gestores de contenidos se les denomina Learning Management System, en inglés, abreviado LMS. Son sistemas de gestión del aprendizaje que se utilizan para administrar, distribuir y controlar las actividades de formación no presencial. Actualmente, el más popular es Moodle (www.moodle.org) (ver Figura 3.9).



Figura 3.9. Ejemplo de gestión de e-learning

3.1.2. Ventajas

Un gestor de contenidos Web permite a un usuario actualizar el contenido de una página web, de manera fácil e intuitiva, sin necesidad de tener conocimientos en informática y desde cualquier parte del mundo donde se tenga acceso a Internet; evitando así la dependencia de terceros para su mantenimiento.

Algunas de las principales ventajas de la utilización de un gestor de contenidos son las siguientes [16]:

- *Bajo costo de mantenimiento*: Con un gestor de contenidos, se tiene total control sobre el contenido de su sitio actual. No requiere de gastos extras de

subcontratación o capacitación, ya que todos los detalles técnicos son manejados por la herramienta de manera sencilla y transparente para el usuario.

- *Ahorro de tiempo:* Un gestor de contenidos permite controlar la publicación de páginas de manera inmediata: agregar y modificar páginas Web utilizando esta herramienta toma sólo minutos en comparación con complejos programas de diseño y editores de código HTML. Además un gestor de contenidos estará disponible las 24 horas del día de manera que estos cambios puedan realizarse a tiempo y sin demora.
- *Mejor control de imagen del sitio web:* Un gestor de contenidos separa el contenido de su presentación, usa plantillas y controladores de estilo para el diseño. De esta manera se puede manipular el diseño, para mantener una consistente y profesional imagen corporativa, sin afectar el contenido que existe actualmente.
- *Mayor información para una mejor publicación:* Un gestor de contenidos brinda una manera ágil de controlar y aprobar la publicación de temas e información en un sitio Web. Así, se podrán publicar las páginas cuando se verifique que la información es correcta. Se podrá seguir trabajando y manipulando la información aún cuando ésta no esté publicada.
- *Mayor interacción con el usuario:* Los gestores de contenidos ofrecen módulos y componentes que brindan al sitio Web mucha interacción con sus visitantes, tales como: Foros de Discusión, Registro de Usuarios, Catálogos de Productos en Línea y Carritos de Compra, Libro de Visitas, Gestores de Formularios de Contacto, Galerías de Imágenes, Listas de Correo, Boletines de Suscripción, etc.

3.1.3. Historia

A principios de los años noventa, el concepto de sistemas de gestión de contenidos era desconocido. Algunas de sus funciones se realizaban con aplicaciones independientes: editores de texto y de imágenes, bases de datos y programación a medida.

Los primeros sistemas de administración de contenidos fueron desarrollados por organizaciones que publicaban una gran cantidad de contenido en Internet, y necesitaban continuas actualizaciones; como revistas en línea, periódicos y publicaciones corporativas.

Ya en el año 1994 Illustra Information Technology (compañía dedicada a las bases de datos) utilizaba una base de datos de objetos como repositorio de los contenidos de una web, con el objetivo de poder reutilizar los objetos y ofrecer a los autores un entorno para la creación basado en patrones. La idea no fue muy aceptada por los usuarios, por lo que la parte de la empresa enfocada a la Web fue comprada por AOL (America Online), mientras que Informix adquirió la parte de bases de datos. RedDot es una de las empresas pioneras que empezó el desarrollo de un gestor de contenidos en el año 1994. Y a finales del año siguiente presentaron su CMS basado en una base de datos.

Entre los CMS de código abierto uno de los primeros fue Typo 3 (www.typo3.org), que empezó su desarrollo en el año 1997, en palabras de su autor, Kasper Skårhøj, "antes de que el término gestión de contenidos fuera conocido sobradamente".

PHPNuke (www.phpnuke.org), la herramienta que popularizó el uso de estos sistemas para las comunidades de usuarios en Internet, se empezó a desarrollar en el año 2000. La primera versión supuso tres semanas de trabajo al creador, reescribiendo el código de otra herramienta llamada Thatware.

Hoy en día existen sistemas desarrollados en software libre y propietario. En ambos casos es necesaria una implementación para adaptar el gestor de contenidos al esquema

gráfico y funcionalidades deseadas. Para ciertos gestores existen muchas plantillas disponibles que permiten una sencilla implementación por parte de un usuario sin conocimientos de programación.

La evolución de Internet hacia portales con más contenido y la alta participación de los usuarios, directamente o a través de blogs y redes sociales, han convertido a los gestores de contenidos en una herramienta esencial en Internet, tanto para empresas e instituciones como para los usuarios individuales [17].

3.2. Propuesta Metodológica para Migración de CMS

En el capítulo 2, se realizó un estudio sobre una metodología de migración clásica para un proceso de migración genérico. Basándonos en la revisión de la metodología clásica citada junto con diferentes procesos y metodologías utilizados para la migración de sistemas hacia software libre, se ha descrito una metodología de migración basada en tecnologías web. En el capítulo 4, se pretende realizar la migración de una aplicación web consistente en un sistema de recomendación de restaurantes georreferenciados desarrollada sobre un sistema gestor de contenidos, concretamente, sobre Joomla (*www.joomla.org*). Al estar desarrollada sobre un gestor de contenidos, en esta sección realizamos una propuesta de metodología de migración para CMS, la cuál será la utilizada para la migración del caso propuesto. Esta propuesta estará basada en la metodología de migración sobre tecnologías web, adaptándola al caso concreto de CMS.

A continuación, se describen las tareas localizadas en las fases de análisis de requerimientos y diseño de la metodología, adaptándolas de manera concreta a los sistemas de gestión de contenidos, debido a la peculiaridad de estos.

El resto de las fases de la metodología de migración basada en tecnologías web, como son el proceso de migración, la evaluación y el mantenimiento, son válidas para la migración de un CMS. En la Figura 3.10 se muestran las fases que van a ser descritas a continuación:

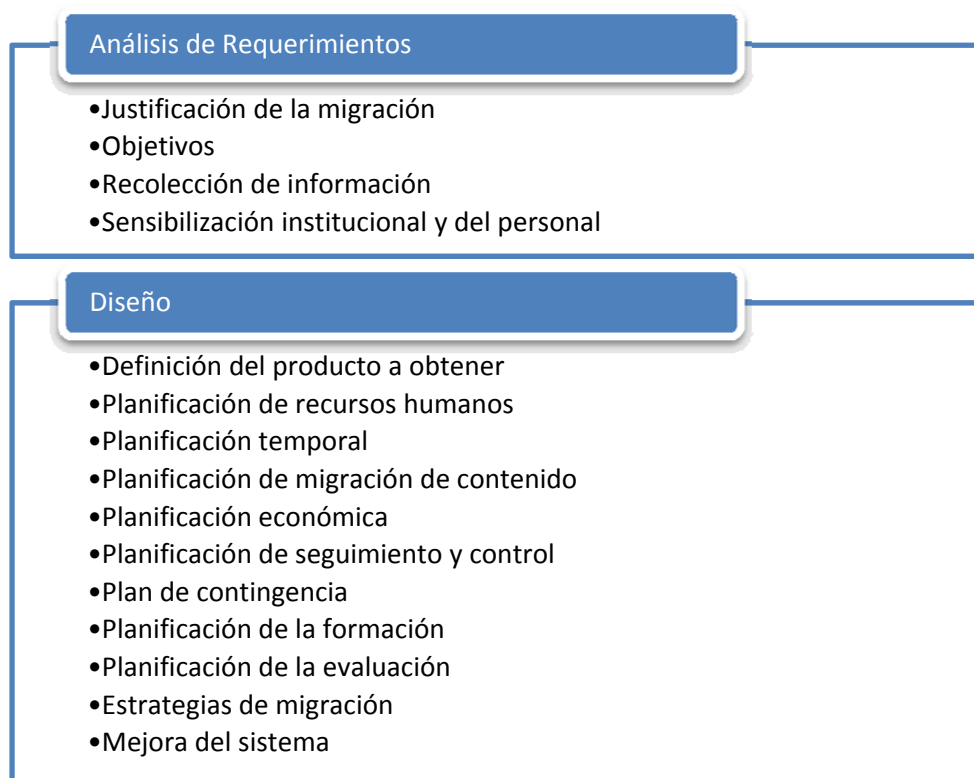


Figura 3.10. Localización de las fases adaptadas a la metodología de migración para CMS

3.2.1. Análisis de Requerimientos

a) Justificación de la migración

A la hora de justificar una migración de un CMS, los motivos suelen ser muy similares. Con frecuencia se realiza la migración para incorporar las novedades que las versiones más actuales nos ofrecen, a la vez que para prevenir de eventuales incidencias o

problemas de seguridad que pueden surgir si no se actualiza. También se realiza el proceso de migración para pasar del sistema actual a otro sistema de gestión de contenidos distinto, por adaptarse éste mejor a nuestras necesidades, mejorar la productividad, la seguridad, etc.

b) Objetivos

Los objetivos a conseguir con la migración de un CMS son similares a los descritos en el proceso de migración de tecnologías web. Entre estos objetivos están el incremento de seguridad de la plataforma web, la mejora de su productividad, el aumento de las funcionalidades y la reducción de costes.

c) Recolección de información

En la recolección de información tenemos que obtener los datos necesarios para hacer un inventario sobre el capital humano del cual se dispone para llevarla a cabo, al igual que los recursos hardware y software, tanto los actuales como sobre los que se realizará la migración en el caso de ser distintos.

El inventario de recursos humanos se realiza para conocer de qué personal se dispone para dar soporte al sistema tanto de administración como de asistencia al mismo, y también para llevar a cabo la migración. La planificación de recursos humanos para la migración se realizará en la fase de diseño.

El inventario tanto hardware como software se realiza para tener la certeza de que el sistema, una vez migrado, sea compatible totalmente con el hardware disponible. Cuando migramos sobre versiones más actuales de un CMS, los requisitos mínimos para el correcto funcionamiento de éste pueden cambiar. Por lo tanto, en el inventario hay que describir

cuáles son los requisitos mínimos de funcionamiento para el nuevo sistema, y cuáles son las características de nuestro hardware.

d) Sensibilización institucional y del personal

Este apartado de la metodología de migración sobre CMS es igual a la migración sobre tecnologías web descrito anteriormente.

3.2.2. Diseño

a) Definición del producto a obtener

Tratándose de migración sobre CMS, al definir el producto a obtener, hay que indicar hacia qué CMS se migrará y su versión. Como consecuencia de esta migración hacia un nuevo sistema, se obtendrán una serie de nuevas características o ventajas respecto al anterior que habrá que especificar.

b) Planificación de recursos humanos

Las personas encargadas de llevar a cabo la migración de un CMS, tienen que tener gran conocimiento sobre su funcionamiento, la estructura de archivos de la que se compone, el lenguaje de programación con el que está desarrollado y el sistema de base de datos que utiliza. En el caso de proyectos de gran tamaño, estos conocimientos están distribuidos en diferentes personas que se encargan conjuntamente de la migración. Estos diferentes perfiles necesarios de recursos humanos, habrá que especificarlos en esta etapa para asegurarnos que se tiene la formación y conocimientos necesarios para llevar a cabo la migración. El resto de perfiles necesarios serán los mismos que en el proceso de migración basado en tecnologías web.

c) Planificación temporal

Dentro de la planificación temporal, además de definir las etapas de las que se compone cualquier proceso de migración sobre tecnologías web, hay que definir las etapas propias del proceso de migración del CMS. Estas etapas son la realización de una copia de seguridad del sistema, la limpieza del estado actual del sistema y la ejecución de la migración.

d) Planificación de migración de contenido

Antes de realizar una migración de un CMS hacia otro CMS distinto, o el mismo pero en una versión diferente, hay que tener en cuenta una serie de puntos para ver la viabilidad de la migración y la planificación de ésta en su caso. Una de las tareas que hay que realizar en la fase de planificación de migración de contenido en un CMS, consiste en ver la disponibilidad de toda la funcionalidad del CMS actual en la versión nueva. Esto hace referencia a que en los CMS es usual añadir extensiones o módulos externos al sistema desarrollados por terceros para ampliar la funcionalidad de estos. Existen extensiones o módulos con diferentes funcionalidades como la gestión de tiendas virtuales, mensajería interna, añadir comentarios a los artículos, organizadores de imágenes, de sonido, de archivos, plantillas para cambiar la apariencia del sitio, para enlazar el CMS con diferentes redes sociales, etc. Una de las características que diferencia a un CMS de otro es el número y la calidad de estas extensiones disponibles. Por tanto, antes de realizar la migración se deberá realizar un inventario de las extensiones que actualmente utiliza el sitio, y ver si existen esas extensiones para la nueva versión del CMS. En caso de no existir, se realizará una búsqueda sobre posibles extensiones que realicen esas mismas funciones. El hecho de no poder disponer en la versión nueva de algunas determinadas extensiones que sean indispensables para el sitio, podrá declinar la intención de realizar la migración en este momento, condicionándola a la disponibilidad de esa extensión. Otra opción disponible es el

desarrollo propio de una extensión que realice la función deseada, e instalarla en la nueva versión del CMS.

Por tanto, antes de comenzar una migración de un CMS, habrá que hacer un inventario de todas las extensiones o módulos que se estén utilizando actualmente y comprobar su disponibilidad en la nueva versión. Para ello, en Internet hay diferentes repositorios de extensiones desde los cuales se pueden consultar la disponibilidad y realizar la descarga en su caso.

Al igual que se realiza la comprobación de disponibilidad de las extensiones o módulos actuales para la nueva versión, habrá que realizar similar proceso para las distintas plantillas o temas que estén configuradas actualmente en el sitio. La plantilla o tema de un CMS es la encargada de la presentación de los contenidos del sitio. De manera similar a las extensiones, en un CMS podemos instalar o desinstalar diferentes plantillas o temas para modificar la presentación del sitio, sin tener para ello que realizar ninguna modificación sobre el contenido. Por tanto, cuando se vaya a realizar una migración, debemos asegurarnos que la plantilla actual sea compatible con la nueva versión. Si no lo fuese, habría que obtener la misma plantilla compatible con la versión deseada o instalar una plantilla diferente.

e) Planificación económica

En el caso de migraciones sobre CMS, hay que añadir a los costes usuales de la migración, el coste de la licencia del sistema de gestión de contenidos y el sistema de gestión de la base de datos. Se puede considerar la gran cantidad de CMS disponibles que son software libre y con características muy avanzadas, de modo que no habría que sumar los costes de estas licencias a la planificación.

f) Planificación de seguimiento y control

Este apartado de la metodología de migración sobre CMS es igual a la migración sobre tecnologías web descrito anteriormente.

g) Plan de contingencia

El plan de contingencia se definió como un plan que debía de establecer una serie de acciones ordenadas y prefijadas que ayudasen a corregir una posible incidencia en el proceso de migración. En este caso, vamos a describir el proceso de realización de copias de seguridad en un CMS. Las copias de seguridad son unos de los procesos más importantes en todo proceso de migración. Sirven para asegurarnos que si por algún motivo, una vez comenzado el proceso de migración, éste no se puede finalizar, siempre estará disponible la opción de volver al estado inicial. Existen diversos métodos y herramientas para realizar copias de seguridad en los CMS. También se pueden realizar de manera mecánica o automática.

Como se ha comentado anteriormente, en todo CMS están diferenciadas las capas de base de datos, de programación y de diseño. Por lo tanto, para hacer la copia de seguridad hay que realizar una copia de cada una de estas capas.

En muchos CMS existen extensiones o módulos que realizan esta tarea de manera automática, bien sea realizando el respaldo de una capa en concreto y las tres a la vez. Para ello basta con instalar esta extensión en el CMS y seguir los pasos indicados por dicha extensión para su correcta ejecución. Una vez realizada, generará uno o varios ficheros de respaldo que se podrán utilizar, en caso de ser necesario, para volver el CMS al estado inicial.

Otra forma de realizar las copias de seguridad del CMS es de manera manual, sin basarnos en ninguna extensión ni módulo. Para ello habrá que seguir unos pasos para su

correcto funcionamiento. Las capas de programación y diseño están formadas por los diferentes ficheros que componen la estructura del CMS. Para realizar la copia de estos ficheros, directamente se realiza una copia en local mediante un cliente ftp de todas las carpetas y archivos que están dentro de la carpeta que agrupa todo el CMS. Para obtener la copia de la capa de base de datos, habrá que realizar una exportación de toda la información contenida en la base de datos desde un gestor de base de datos. Un gestor muy utilizado es phpMyAdmin (www.phpmyadmin.net).

Tras la realización de estos procedimientos de copias de seguridad, tanto realizados de forma automática como manual, hay que confirmar la correcta ejecución de los mismos para asegurarse que la copia se ha realizado correctamente y no tener posibles problemas durante el proceso de migración.

h) Planificación de la formación

Para conseguir que el sistema una vez migrado continúe siendo válido funcionalmente, hay que definir un plan de formación para los administradores de CMS y los usuarios finales. Al realizar una migración hacia una nueva versión de un CMS u otro distinto, son bastantes las características diferentes que se presenten dependiendo de la versión. Estas nuevas funcionalidades tienen que estar en conocimiento de los administradores y los usuarios que lo requieran para obtener el máximo provecho del nuevo sistema. Para ello, se pueden realizar cursos de formación sobre las nuevas materias, tanto para administradores como usuario, al igual que con manuales desarrollados para tal fin.

i) Planificación de la evaluación

En esta apartado definimos las diferentes pruebas que se van a realizar sobre el sistema una vez migrado para comprobar que se ha llevado a cabo la migración

correctamente. En el caso de los CMS, estas pruebas estarán dirigidas al correcto funcionamiento del sistema, comprobando que se devuelven los datos correctos en una consulta que se hará sobre la base de datos, que se muestra correctamente el aspecto de la página para comprobar que la plantilla está bien configurada, comprobar el funcionamiento de las extensiones, etc. Todas estas pruebas se realizarán en la fase de evaluación.

j) Estrategias de migración

En el proceso de migración basado en tecnología web, se definían tres diferentes tipos de estrategias de migración: migración automatizada, manual o parcialmente automatizada.

En el caso de los sistemas gestores de contenidos, realizar una migración manual llevaría mucho tiempo y esfuerzo, ya que habría que modificar manualmente tanto la base de datos como el sistema de ficheros, para adecuarse a la nueva versión. Una migración automatizada será la que realice de manera automática toda la migración de la base de datos y los ficheros a la nueva versión por completo, sin tener que intervenir en nada. Aún siendo más viable que la manual, en la práctica habrá muchos datos que no migren automáticamente, como datos de extensiones externas, por lo que éstos habrá que hacerlos de forma manual. Como estrategia más común para la migración de CMS, se utiliza la migración parcialmente automatizada. Se basa en la migración automatizada de cierta parte del sistema gracias a extensiones programadas para ello, y el resto se realiza de manera manual, siguiendo las especificaciones facilitadas por el desarrollador del sistema en cada caso.

k) Mejora del sistema

Al igual que en la migración basadas en sistemas web, se puede aprovechar el proceso de migración de un CMS para añadir nuevas funcionalidades o mejoras en el sistema. En esta etapa se describen cada una de las mejoras que se desean llevar a cabo.

Capítulo 4

Migración del Sistema de Recomendación de Restaurantes REJA en el CMS Joomla

En este capítulo, vamos a aplicar la metodología de migración propuesta en el capítulo anterior, sobre el Sistema de Recomendación de Restaurantes Georreferenciados REJA. En primer lugar describimos brevemente el sistema que vamos a migrar. En segundo lugar, realizamos una introducción al CMS Joomla, ya que es el sistema gestor de contenidos sobre el que está desarrollado la aplicación web a migrar. Posteriormente, aplicamos la propuesta de metodología de migración de CMS sobre la aplicación REJA, describiendo el proceso paso a paso. Para finalizar, exponemos una serie de procesos adicionales a la metodología propuesta para resolver problemas específicos encontrados durante el proceso.

4.1. Descripción del Sistema de Recomendación de Restaurantes Georreferenciados REJA

El sistema sobre el que se va a poner en práctica la metodología de migración descrita, es el Sistema de Recomendación de Restaurantes Georreferenciados REJA. Este sistema se obtuvo como producto software resultado de los proyectos de investigación “Sistema de Recomendación de Restaurantes basado en lógica difusa REJA” (JA031/06) y “Sistema de Recomendación Georreferenciado de Restaurantes” (2007/BA020), ambos concedidos por la Consejería de Turismo, Comercio y Deporte de la Junta de Andalucía.

Este Sistema de Recomendación de Restaurantes Georreferenciados de Jaén consta principalmente de dos partes:

- Una aplicación web para la recomendación de restaurantes (ver Figura 4.1).
- Una aplicación web para la georreferenciación y geolocalización de restaurantes (ver Figura 4.2).

Permite a sus usuarios recibir recomendaciones sobre a qué restaurantes acudir cuando visiten Jaén o cualquier pueblo de la provincia atendiendo a sus propios gustos, así

como su posición en el plano. De esta forma se pretende mejorar la experiencia del turista, de manera que se sienta tan cómodo en el territorio que visita como en el suyo propio.

El sistema permite el registro de usuarios, la obtención de los datos almacenados en el perfil del usuario, la modificación de estos datos, el acceso a la información de los restaurantes, realizar puntuaciones sobre éstos, modificar dichas puntuaciones, obtener recomendaciones de restaurantes, etc. Por otro lado, también permite geolocalizar los restaurantes, obtener un camino mínimo necesario para llegar a un restaurante desde la dirección del usuario, calcular la ruta entre dos puntos, realizar consultas sobre el mapa, etc.



Figura 4.1. Aplicación web para la recomendación de restaurantes



Figura 4.2. Aplicación web para la georreferenciación y geocodificación de restaurantes

4.2. Introducción al CMS utilizado

4.2.1. Definición de Joomla

Joomla [18] es un sistema de gestión de contenidos que permite construir sitios y aplicaciones web fácilmente. Está basado en tecnologías web (como XHTML, CSS, JavaScript y PHP) y válido tanto en Internet como para intranets. En la Figura 4.3 se muestra su logotipo.



Figura 4.3. Logotipo del CMS Joomla.

Está programado en el lenguaje PHP bajo la licencia GNU GPL, y utiliza una base de datos MySQL para almacenar el contenido y los parámetros de configuración del sitio (ver Figura 4.4).

El CMS Joomla se puede conseguir gratuitamente en Internet. Su gestión se lleva a cabo a través de una interfaz web, por tanto, es fundamental contar con un navegador web instalado.

Al ser Joomla una aplicación web, funciona en servidores http. Esta característica permite que se pueda utilizar en cualquier equipo con cualquier sistema operativo instalado.

La administración está basada en una gestión online de contenidos, esto quiere decir que para actualizar los contenidos de un sitio web basta con utilizar un navegador web conectado a Internet.

Joomla se puede utilizar tanto en un servidor remoto como en un equipo local. Un equipo doméstico podrá ser configurado para que actúe como un servidor local y se pueda probar así todas las funcionalidades de la herramienta antes de colocar sus archivos en el servidor que aloja el sitio. Para poder utilizarlo como servidor local, se ha de tener instalado alguna aplicación que convierta el ordenador en un servidor web, como Apache. El servidor, tanto el local como en remoto, debe disponer de MySQL y PHP. Si es remoto además es necesario un cliente FTP para la subida de los ficheros necesarios. Existen aplicaciones que integran estos servicios para configurar un servidor local, como Wamp (<http://www.wampserver.com>), Xampp (<http://www.apachefriends.org/es/xampp.html>) o EasyPHP (<http://www.easyphp.org>) para ambientes Windows, o Lamp (<http://www.lamp.es>) para ambientes Linux.

Los archivos que constituyen la estructura de Joomla y que se descargan de Internet están comprimidos, por tanto, será necesario también un compresor/descompresor de archivos ZIP.

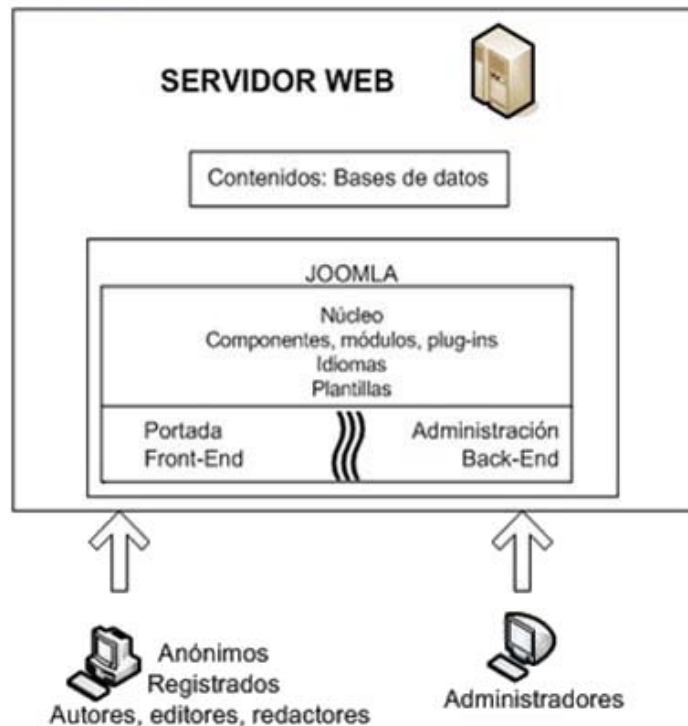


Figura 4.4. Gráfico del funcionamiento de Joomla

4.2.2. Estructura

Como se ha comentado, Joomla está compuesto por una o varias bases de datos y un conjunto de archivos y carpetas. Entender la estructura de Joomla y conocer cuáles son sus principales ficheros y directorios ayuda a una mejor comprensión de su funcionamiento. En la Figura 4.5 se muestra la estructura de archivos y carpetas una vez descargado y descomprimido.

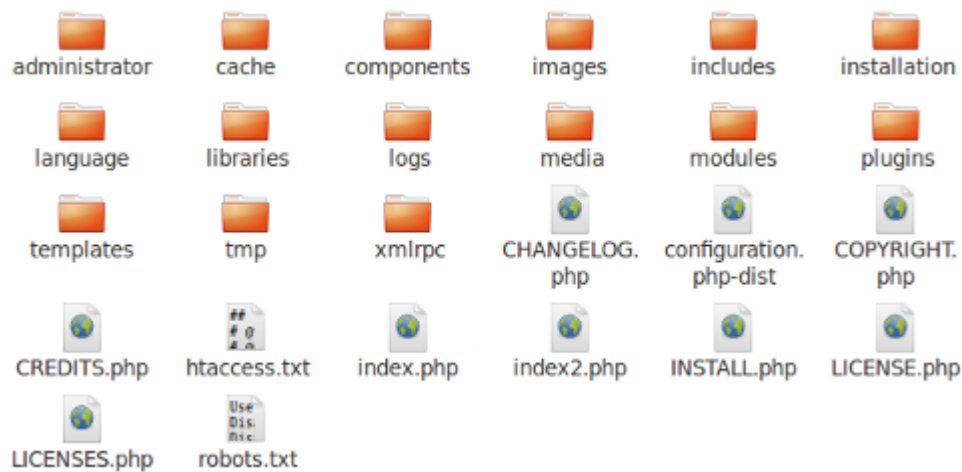


Figura 4.5. Estructura de archivos y carpetas de Joomla 1.5

La funcionalidad de cada una de estas carpetas se explica brevemente a continuación:

- *Administrator*: Contiene todos los archivos necesarios que conciernen a la administración desde el backend de Joomla.
- *Cache*: Almacena los archivos necesario para utilizar el método caché, si éste está habilitado en el sitio web.
- *Components*: Contiene las carpetas con cada uno de los componentes que están instalados en el sitio, el nombre de estas carpetas es generalmente com_nombre-de-componente.
- *Images*: Contiene los iconos base que utiliza la plantilla de Joomla, para algunas vistas.
- *Includes*: Es una de las carpetas que contienen los archivo esenciales que utiliza Joomla como parte de su framework.
- *Installation*: Contienen los archivos necesarios para la instalación del sitio por primera vez; una vez se instala la plantilla de Joomla, esta carpeta se debe borrar.

- *Languages*: Almacena los archivos de los diferentes lenguajes que soporta Joomla.
- *Libraries*: Almacena las librerías básicas del framework de Joomla.
- *Logs*: Si está configurado almacena los eventos que se generan en el administrador.
- *Media*: Almacena las imágenes que se suben por medio del administrador de medios.
- *Modules*: Almacena los módulos que se encuentran instalados en el sitio, estos se encuentran en subcarpetas y llevan el nombre mod_nombre-del-modulo.
- *Plugins*: Almacena los plugins instalados en el sitio.
- *Templates*: Almacena las subcarpetas con los templates instalados y sus respectivos recursos.
- *Tmp*: Carpeta utilizada durante la instalación de los módulos, componentes, plugins o templates.

4.2.3. Funcionamiento

Su funcionamiento se lleva a cabo gracias a sus dos principales componentes:

- *La base de datos MySQL*: En ella se almacena toda la información en una estructura de tablas dentro de la misma.
- *Las páginas PHP*: Son las páginas implementadas en dicho lenguaje de programación que realizan conversión de datos, que residen en la base de datos en páginas web completas, interpretables por los navegadores y totalmente legibles por los usuarios.

A parte de estos dos elementos fundamentales para el funcionamiento de Joomla, también existen otros como archivos XML, CSS, scripts en Javascript, etc., que completan la arquitectura de este sistema.

Una vez instalado el paquete básico de Joomla, comienza la personalización del sitio Web que se está desarrollando. Esto se consigue de manera flexible y escalable, desde las siguientes opciones que proporciona:

- *Zona privada de Administración:* Se dispone de un panel de administración desde donde se gestiona completamente el sitio web (ver Figura 4.6). Dicha zona se denomina “Backend” de administración al que se accede con usuario y contraseña.



Figura 4.6. Zona privada de administración de Joomla.

- *Gestión de los menús del sitio web:* Desde el panel de administración se pueden crear y organizar todos los elementos de los menús del sitio web, creándolos, editándolos y habilitándolos o deshabilitándolos para un determinado tipo de usuario (ver Figura 4.7).

| núm. | Título | Tipo | Item(s) del menú | # Publicado | # Despublicado | # Papelera | Número de Módulos | ID |
|------|--------------------|--------------|------------------|-------------|----------------|------------|-------------------|----|
| 1 | Menú principal | mainmenu | | 9 | - | - | 1 | 1 |
| 2 | Menú de usuario | usermenu | | 4 | - | - | 1 | 2 |
| 3 | Menú principal | topmenu | | 4 | - | - | 1 | 3 |
| 4 | Recursos | othermenu | | 7 | - | - | 2 | 4 |
| 5 | Páginas de ejemplo | ExamplePages | | 4 | - | - | 1 | 5 |
| 6 | Conceptos clave | keyconcepts | | 3 | - | - | 1 | 6 |

Mostrar núm. 20

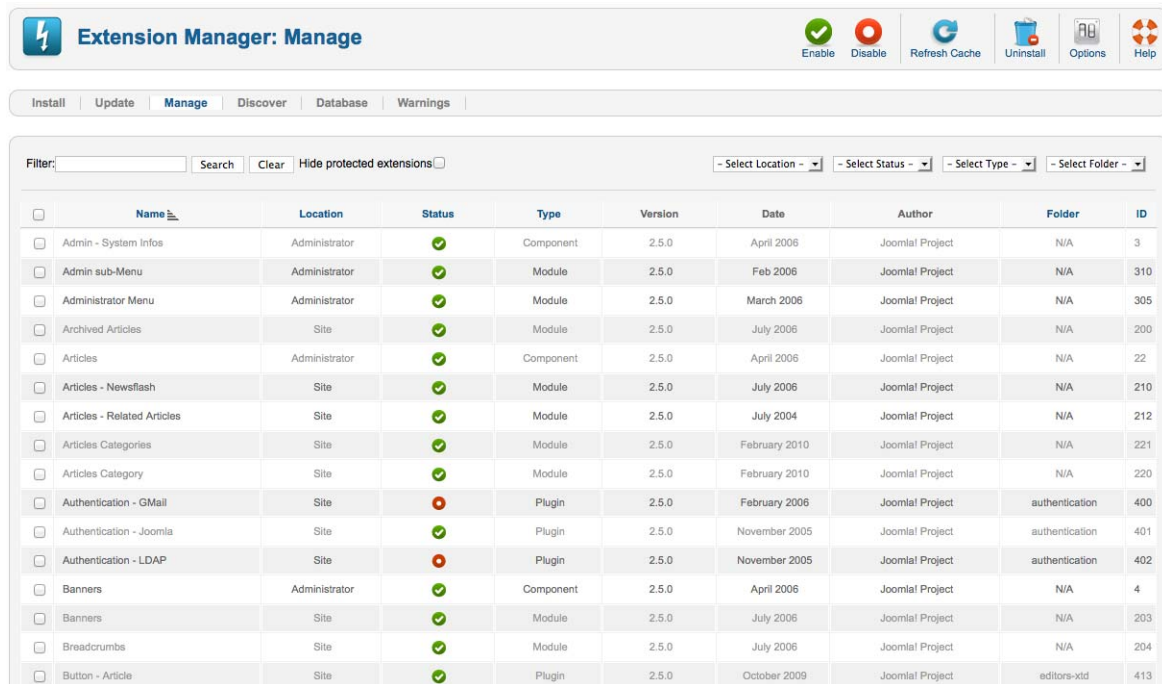
Figura 4.7. Gestión de los menús del sitio web

- *Gestión de contenidos de la web:* También desde el panel de administración se pueden crear y organizar los artículos de contenido de la web en categorías, para su fácil creación, modificación o eliminación (ver Figura 4.8).

| núm. | Título | Publicado | Página principal | Ordenar | Acceso | Sección | Categoría | Autor | Fecha | Impresiones | ID |
|------|---|-----------|------------------|---------|---------|---------|-----------|---------------|----------|-------------|----|
| 1 | Páginas de ejemplo y menú de enlaces | | | 1 | Público | | | Administrador | 12.10.06 | 43 | 43 |
| 2 | ¿Es posible cambiar el tipo de un ítem del menú? | | | 1 | Público | FAQs | General | Administrador | 10.10.06 | 17 | 35 |
| 3 | ¿Qué es un artículo sin clasificar? | | | 2 | Público | FAQs | General | Administrador | 11.10.06 | 5 | 33 |
| 4 | ¿Cómo instalar Joomla! 1.5? | | | 3 | Público | FAQs | General | Administrador | 11.10.06 | 4 | 28 |
| 5 | ¿Puede Joomla! 1.5 operar con el Safe Mode On? | | | 4 | Público | FAQs | General | Administrador | 06.10.06 | 7 | 15 |
| 6 | ¿Requisitos necesarios para que Joomla! 1.5 funcione? | | | 5 | Público | FAQs | General | Administrador | 11.10.06 | 27 | 25 |
| 7 | ¿Para qué es la capa FTP? | | | 6 | Público | FAQs | General | Administrador | 06.10.06 | 22 | 14 |
| 8 | Mi base de datos MySQL no soporta UTF-8, ¿hay algún problema? | | | 7 | Público | FAQs | General | Administrador | 07.10.06 | 8 | 17 |
| 9 | ¿Por qué usa Joomla! 1.5 la codificación utf-8? | | | 8 | Público | FAQs | General | Administrador | 05.10.06 | 28 | 12 |

Figura 4.8. Gestión de contenidos de la web

- *Gestión de componentes y extensiones:* Joomla es una herramienta modular, escalable y configurable según las necesidades. Para ello, posee gran variedad de plantillas, idiomas, componentes, módulos, plugins, etc. (ver Figura 4.9).



The screenshot shows the Joomla! Extension Manager interface. At the top, there is a header with the title "Extension Manager: Manage" and several action buttons: Enable, Disable, Refresh Cache, Uninstall, Options, and Help. Below the header, there are tabs for "Install", "Update", "Manage", "Discover", "Database", and "Warnings". The main area contains a filter bar with a search input, a "Clear" button, and a "Hide protected extensions" checkbox. Below the filter bar, there are four dropdown menus for "Select Location", "Select Status", "Select Type", and "Select Folder". The main content is a table listing various extensions with columns for Name, Location, Status, Type, Version, Date, Author, Folder, and ID.

| <input type="checkbox"/> | Name | Location | Status | Type | Version | Date | Author | Folder | ID |
|--------------------------|-----------------------------|---------------|--------|-----------|---------|---------------|-----------------|----------------|-----|
| <input type="checkbox"/> | Admin - System Infos | Administrator | ✓ | Component | 2.5.0 | April 2006 | Joomla! Project | N/A | 3 |
| <input type="checkbox"/> | Admin sub-Menu | Administrator | ✓ | Module | 2.5.0 | Feb 2006 | Joomla! Project | N/A | 310 |
| <input type="checkbox"/> | Administrator Menu | Administrator | ✓ | Module | 2.5.0 | March 2006 | Joomla! Project | N/A | 305 |
| <input type="checkbox"/> | Archived Articles | Site | ✓ | Module | 2.5.0 | July 2006 | Joomla! Project | N/A | 200 |
| <input type="checkbox"/> | Articles | Administrator | ✓ | Component | 2.5.0 | April 2006 | Joomla! Project | N/A | 22 |
| <input type="checkbox"/> | Articles - Newsflash | Site | ✓ | Module | 2.5.0 | July 2006 | Joomla! Project | N/A | 210 |
| <input type="checkbox"/> | Articles - Related Articles | Site | ✓ | Module | 2.5.0 | July 2004 | Joomla! Project | N/A | 212 |
| <input type="checkbox"/> | Articles Categories | Site | ✓ | Module | 2.5.0 | February 2010 | Joomla! Project | N/A | 221 |
| <input type="checkbox"/> | Articles Category | Site | ✓ | Module | 2.5.0 | February 2010 | Joomla! Project | N/A | 220 |
| <input type="checkbox"/> | Authentication - Gmail | Site | ✗ | Plugin | 2.5.0 | February 2006 | Joomla! Project | authentication | 400 |
| <input type="checkbox"/> | Authentication - Joomla | Site | ✓ | Plugin | 2.5.0 | November 2005 | Joomla! Project | authentication | 401 |
| <input type="checkbox"/> | Authentication - LDAP | Site | ✗ | Plugin | 2.5.0 | November 2005 | Joomla! Project | authentication | 402 |
| <input type="checkbox"/> | Banners | Administrator | ✓ | Component | 2.5.0 | April 2006 | Joomla! Project | N/A | 4 |
| <input type="checkbox"/> | Banners | Site | ✓ | Module | 2.5.0 | July 2006 | Joomla! Project | N/A | 203 |
| <input type="checkbox"/> | Breadcrumbs | Site | ✓ | Module | 2.5.0 | July 2006 | Joomla! Project | N/A | 204 |
| <input type="checkbox"/> | Button - Article | Site | ✓ | Plugin | 2.5.0 | October 2009 | Joomla! Project | editors-xtd | 413 |

Figura 4.9. Gestión de componentes y extensiones

Joomla permite la total personalización y adecuación de su funcionamiento a un tipo de sitio web particular, una vez instalado el paquete básico se pueden ir agregando más funcionalidades, de acuerdo al grado de complejidad del sitio web que se está desarrollando.

Para ampliarlo existen extensiones gratuitas y comerciales que pueden descargarse y probarse, aunque a veces puedan ser incompatibles entre sí o con la versión de Joomla que se haya instalado. Estas extensiones pueden ser componentes, módulos, plugins y plantillas. Permiten instalarse, configurarse y personalizarse desde el propio panel de administración de Joomla

Algunos ejemplos de las funcionalidades de las extensiones disponibles son:

- Realizar y restaurar copias de seguridad
- Crear y gestionar galerías de imágenes

- Mostrar mapas de Google en el sitio
- Crear un mapa del sitio utilizando la estructura de menús.
- Creación de boletines de noticias
- Creación de formularios
- Etc.

4.2.4. Ventajas y Características

Dentro de las principales ventajas de utilizar Joomla se encuentran:

- La publicación de contenidos se organiza en categorías permitiendo crear tantas páginas como se necesiten y organizarlas por temás.
- Se pueden definir tantos menús y submenús como se necesiten.
- La edición de los contenidos se realiza de manera muy fácil e intuitiva, gracias a los editores WYSIWYG que incorpora (What You See Is What You Get en ingles, "lo que ves es lo que obtienes").
- Existen muchas extensiones adicionales, las que permiten ampliar las posibilidades y características de Joomla.
- Joomla puede instalarse en servidores Windows, Linux y Mac.
- Gracias a su sistema de plantillas es posible separar la presentación del contenido. Estas plantillas se encargan de definir el aspecto estético de la web y el orden de los elementos de los sitios web desarrollados con Joomla.

- Joomla es un software en constante evolución. El grupo de desarrolladores y la comunidad trabajan para ofrecer software mejor y más seguro y corregir las posibles vulnerabilidades del código.
- SEO (Optimización para motores de búsqueda): Joomla permite posicionar los sitios web en los motores de búsqueda. Para ello genera etiquetas “meta” distintas y títulos distintos para cada página, URLs amigables, etc.

4.3. Proceso de Migración de REJA paso a paso

Una vez presentado en el capítulo anterior la metodología a utilizar en un proceso de migración de un sitio web centrándolo en un CMS e introducido el CMS en el que está basado nuestro sitio web, pasamos a describir el proceso completo de migración de éste, basándonos en la metodología descrita.

4.3.1. Análisis de Requerimientos

En la fase inicial del proceso de migración, se procede a realizar un análisis de los requerimientos necesarios (ver Figura 4.10). Para ello desarrollaremos los apartados de justificación de la migración, objetivos a cumplir con la migración, recolección de información y sensibilización institucional y del personal.

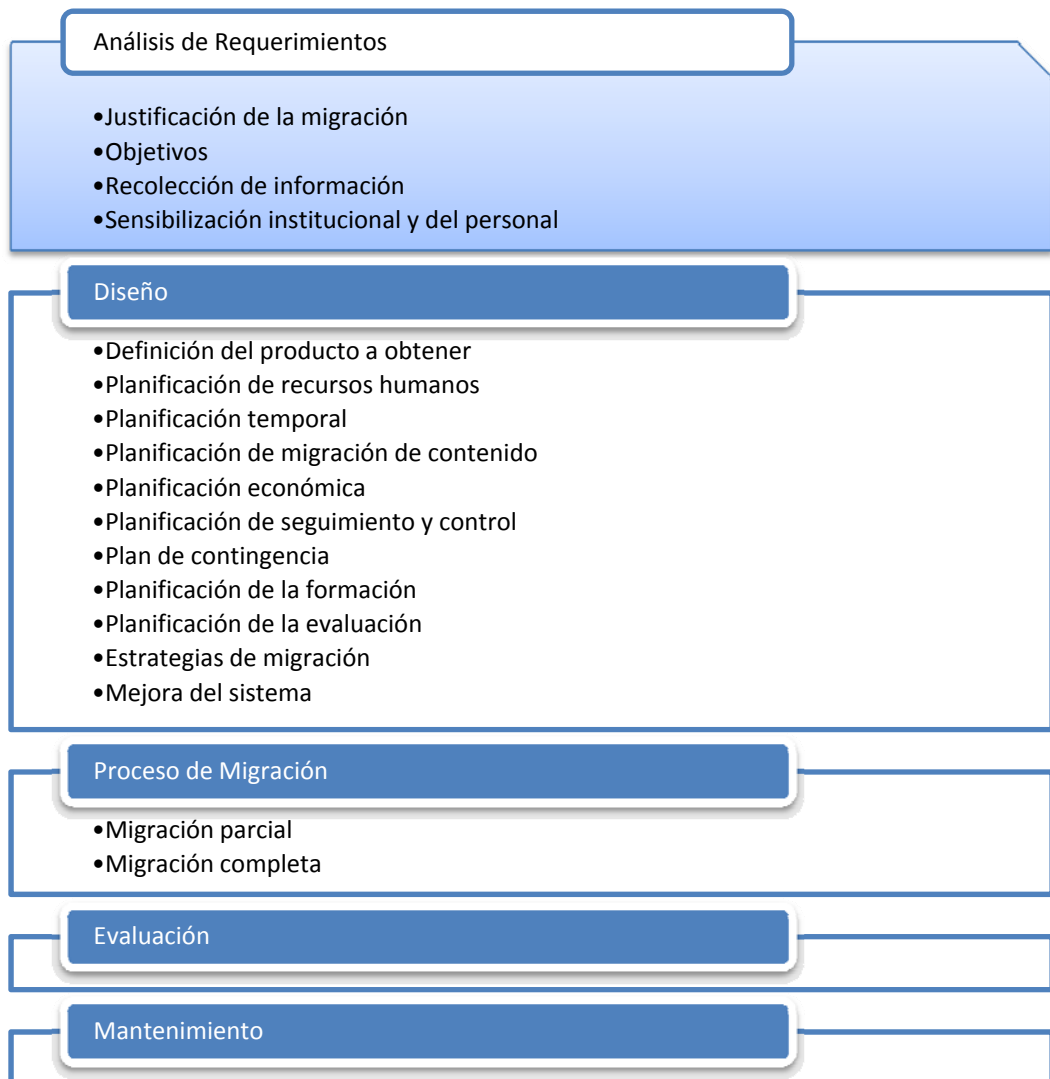
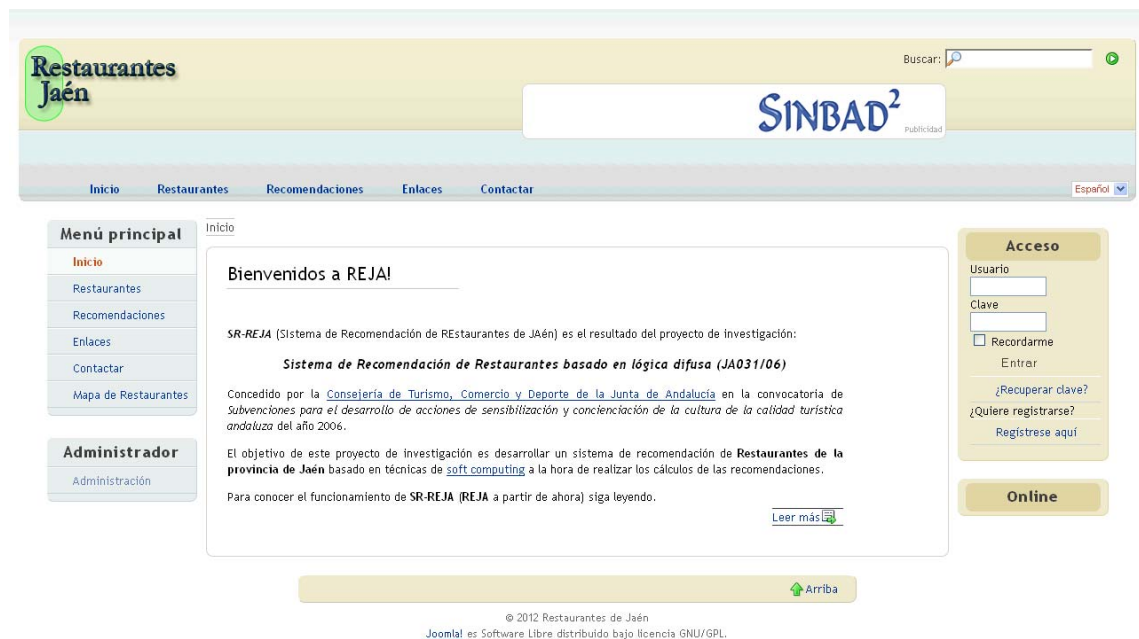


Figura 4.10. El análisis de requerimientos es la primera fase de la metodología

a) *Justificación de la migración*

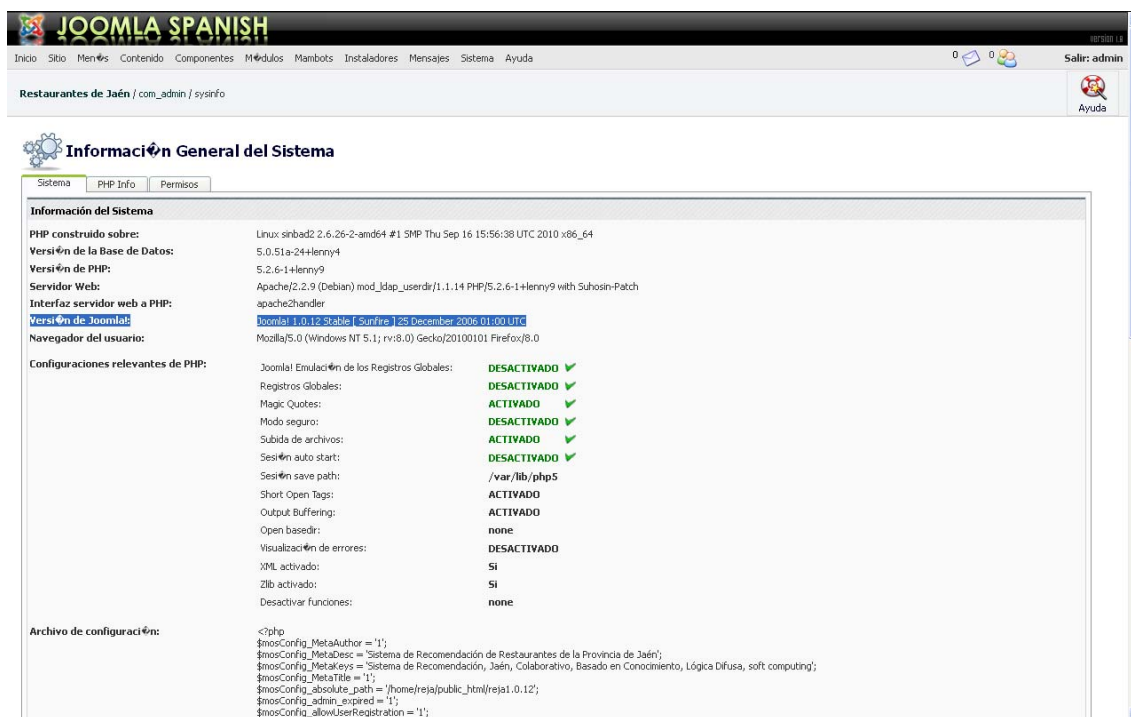
El primer paso que tendríamos que hacer para realizar la migración hacia una versión más actual de Joomla, sería ver el estado actual del sistema. Para ello accedemos a su alojamiento actual que es <http://sinbad2.ujaen.es/reja/>. El aspecto actual de la página de inicio de Reja se muestra en la Figura 4.11.



4.11. Pantalla de inicio del sitio antes de la migración

Lo primero que necesitamos es saber la versión actual del CMS sobre el que está funcionando Reja, para así determinar la necesidad de migración y hacia qué nuevo sistema migrar. Para ver la versión actual del CMS, en este caso de Joomla, accedemos al Backend con el usuario y contraseña de administrador mediante la url siguiente: <http://sinbad2.ujaen.es/reja/administrator>.

Una vez dentro del Panel de Administración, para obtener la versión actual de Joomla, accedemos al menú Sistema y seleccionamos la opción Información del Sistema. En la pestaña Sistema podemos obtener información general del sistema entre la que se encuentra la versión de Joomla, que en este caso es Joomla 1.0.12 Stable (ver Figura 4.12).



4.12. Pantalla del Backend sobre Información General del Sistema

La versión instalada se trata de una versión desactualizada. Los motivos para la realización de la migración hacia una versión actual del mismo CMS son el conseguir mayor seguridad para nuestro sitio web, ampliación de funcionalidades y limpieza del sitio, prescindiendo tanto de los contenidos como extensiones que actualmente estén instalados en el sitio pero que carecen de utilidad.

Para conseguir migrar el sitio a la versión deseada, en nuestro caso la versión 1.5.26, deberemos realizar previamente una actualización hacia la versión 1.0.15 para poder desde ahí, realizar la migración a la versión deseada.

Existen diferencias entre los conceptos de migración y actualización en el entorno de Joomla al igual que el término clonación. Las versiones de Joomla se identifican con tres dígitos, con el formato a.b.c, por ejemplo: Joomla 1.5.23 o Joomla 1.0.15. Un cambio en el

primer dígito indica un cambio importante en el núcleo de Joomla. Un cambio en el segundo dígito indica que se han añadido nuevas funcionalidades en el sistema pero intentando mantener un grado de compatibilidad. El cambio hacia ese sistema es lo que se llama migración. Es adaptar los contenidos de una versión antigua de Joomla para que funcione en una más actual. Un cambio en el tercer dígito indica que solo se han introducido pequeñas correcciones, generalmente de seguridad. La adaptación a ese sistema es el cambio más básico, y es la actualización. La clonación es la duplicación de un sitio web, bien en su propio servidor (para duplicarlo en otra carpeta) o en otro diferente (para trasladar el alojamiento).

b) *Objetivos de la migración*

El principal objetivo a la hora de realizar la migración de este sitio web es mejorar su seguridad. El sitio web está desarrollado bajo la versión 1.0.12 del CMS Joomla. Esta versión actualmente no tiene soporte, con los problemas de seguridad y funcionalidad que ello conlleva.

Al estar basado el sitio web en un CMS cuya versión es obsoleta y desactualizada, perdemos la oportunidad de poder añadir funcionalidades nuevas para poder conseguir un sitio web más competitivo y funcional. En este caso, no podremos añadir a nuestro sitio las nuevas extensiones que se desarrollen para este CMS ya que puede que no sean compatibles con las versiones desactualizadas.

Aprovechando la migración del sitio a una versión actual para conseguir los objetivos anteriores, se propone como objetivo la revisión tanto del contenido como del diseño del sitio.

c) *Recolección de información*

En esta fase se estudian todos los datos necesarios para empezar el proceso de migración, agrupándolos en tres bloques como son el inventario del capital humano, el inventario del hardware y el inventario del software. Al ser distinto el servidor en el que está alojado el sitio web y en el que lo estará una vez finalizada la migración, se detalla el inventario hardware y software tanto del servidor inicial como del final.

- *Inventario del capital humano*: al estar el proceso de migración del sitio web integrado en este Proyecto Fin de Carrera, todo el proceso de migración será llevado a cabo por el alumno que realiza tal proyecto. La fase de mantenimiento será llevada a cabo por personal de la Universidad, ya que no forma parte del proyecto al ser una tarea prolongada en el tiempo durante la vida útil de éste.
- *Inventario hardware*: en este apartado realizamos un inventario del hardware, tanto del servidor actual como del servidor nuevo tras la migración. Para ello utilizamos el documento para tal fin del Anexo III. La Figura 4.13 muestra la información hardware del servidor donde estaba alojada la aplicación web antes de la migración.

| Información del inventario de hardware | |
|--|---------------------|
| Nombre: | ID |
| Empresa: | Universidad de Jaén |
| Sector: | |
| Responsable del inventario: | Juan Fco Galiano |
| | Equipo |
| Nombre del equipo: | suleiman |
| Comentarios: | |
| | Sistema Operativo |
| Nombre: | GNU/Linux Debian |
| Versión: | Versión 5.07 |

| | | | |
|-------------------------|---|--|--|
| Service Pack: | | | |
| | Procesador | | |
| Tipo: | Procesador Intel(R) Xeon(R) CPU X3320 | | |
| Velocidad: | 2.50GHz | | |
| Número de procesadores: | 1 | | |
| | Memoria RAM | | |
| Descripción | DDR2 | | |
| Capacidad: | 4 GB | | |
| Velocidad (MHz): | 233 MHz | | |
| Numero de Ranuras: | | | |
| | Almacenamiento (Lista) | | |
| Fabricante: | | | |
| Modelo: | | | |
| Descripción (IDE,...): | | | |
| Tipo: | HD | | |
| Tamaño | 150 GB | | |
| | Sonido | | |
| Fabricante: | | | |
| Nombre: | | | |
| Descripción: | Integrada en la placa base | | |
| | Tarjeta vídeo | | |
| Nombre: | Matrox Graphics, Inc. MGA G200e | | |
| Chipset: | | | |
| Memoria (MB): | | | |
| Resolución: | | | |
| | Comentarios | | |
| | <ul style="list-style-type: none"> • MySQL versión 5.0 • PhpMyAdmin versión 4.2.11.8.1. | | |

Figura 4.13. Documento con la información hardware del servidor anterior a la migración.

La Figura 4.14 muestra la información hardware del servidor nuevo en el cual va a ser alojada la aplicación web.

| Información del inventario de hardware | | | |
|--|---|--------|--|
| Nombre: | | ID | |
| Empresa: | Universidad de Jaén | | |
| Sector: | | | |
| Responsable del inventario: | Juan Fco Galiano | | |
| Equipo | | | |
| Nombre del equipo: | serezade | | |
| Comentarios: | | | |
| Sistema Operativo | | | |
| Nombre: | Debian GNU/Linux | | |
| Versión: | Wheezy | | |
| Service Pack: | | | |
| Procesador | | | |
| Tipo: | Procesador AMD Opteron(TM) Processor 6274 | | |
| Velocidad: | 2.2GHz | | |
| Número de procesadores: | 1 | | |
| Memoria RAM | | | |
| Descripción | DDR2 | | |
| Capacidad: | 16 GB | | |
| Velocidad (MHz): | 233 MHz | | |
| Numero de Ranuras: | | | |
| Almacenamiento (Lista) | | | |
| Fabricante: | | | |
| Modelo: | | | |
| Descripción (IDE,...): | | | |
| Tipo: | HD | HD | |
| Tamaño : | 1 TB | 500 GB | |
| Sonido | | | |
| Fabricante: | | | |
| Nombre: | | | |
| Descripción: | Integrada en la placa base | | |
| Tarjeta vídeo | | | |

| | |
|---------------|--|
| Nombre: | Matrox Electronics Systems Ltd. MGA G200e |
| Chipset: | |
| Memoria (MB): | |
| Resolución: | |
| | Comentarios |
| | <ul style="list-style-type: none"> • Mysql versión 5.1.61 • PhpMyAdmin versión 3.4.10.2 • Servidor web Apache 2.2.22 Debian |

Figura 4.14. Documento con la información hardware del servidor nuevo tras la migración.

- *Inventario software:* Al igual que en el inventario hardware describimos el hardware antiguo y el nuevo, pasamos a describir la plataforma web sobre la que está desarrollada la aplicación web antes y después de la migración. Para ello utilizamos el documento para tal fin del Anexo III. La Figura 4.15 muestra las características de la plataforma web actual.

| Información del inventario de software | |
|--|-----------------------|
| Nombre: | ID: |
| Empresa: | Joomla! |
| Sector: | |
| Responsable del inventario: | Juan Fco Galiano |
| | Software |
| Nombre del programa: | Joomla! 1.0.12 Stable |
| Descripción: | |
| Comentarios: | |
| | Plataforma |
| Sistema Operativo: | Multiplataforma |
| Maquina virtual o intérprete: | |
| | Aspectos Legales |
| Licencia SW: | GNU/GLP |

| | | | |
|-----------------------------|--|--|--|
| Contratos de mantenimiento: | | | |
| | Requisitos Mínimos | | |
| PHP: | v. 4.2.x | | |
| MySQL: | v. 3.23.x | | |
| Apache: | v. 1.13.19 | | |
| | Dispositivos hardware necesarios (Lista) | | |
| Descripción | | | |
| | Descripción de funcionalidades | | |
| | | | |

Figura 4.15. Documento con la información software de la plataforma web actual.

La Figura 4.16 muestra la información software de la plataforma web sobre la que estará la aplicación web tras la migración.

| | | | |
|--|-----------------------|-----|--|
| Información del inventario de software | | | |
| Nombre: | | ID: | |
| Empresa: | Joomla! | | |
| Sector: | | | |
| Responsable del inventario: | Juan Fco Galiano | | |
| | Software | | |
| Nombre del programa: | Joomla! 1.5.26 Stable | | |
| Descripción: | | | |
| Comentarios: | | | |
| | Plataforma | | |
| Sistema Operativo: | Multiplataforma | | |
| Maquina virtual o intérprete: | | | |
| | Aspectos Legales | | |
| Licencia SW: | GNU/GLP | | |
| Contratos de mantenimiento: | | | |
| | Requisitos Mínimos | | |

| | | | |
|--------------------------------|--|--|--|
| PHP: | v. 4.3.x | | |
| MySQL: | v. 3.23.x | | |
| Apache: | v. 1.3 | | |
| | Dispositivos hardware necesarios (Lista) | | |
| Descripción | | | |
| Descripción de funcionalidades | | | |
| | | | |

Figura 4.16. Documento con la información software de la plataforma web nueva.

d) Sensibilización institucional y del personal

Este proyecto de migración ha sido propuesto por la necesidad de realizar la migración del sitio web actual a una nueva versión por los motivos expuestos en el apartado de Justificación de la Migración. En este caso no es necesaria la sensibilización institucional y de personal ya que la migración viene propuesta por los responsables del proyecto y con ella se van a conseguir una serie de ventajas y funcionalidades nuevas que anteriormente no eran posibles. Como se ha comentado, unos de los motivos es la seguridad del sitio web. Esto es transparente al usuario pero de vital importancia para la continuidad del sitio.

4.3.2. Diseño

La segunda fase del proceso de migración es la fase de diseño (ver Figura 4.17). En esta fase se analiza y planifica todo el proceso de migración. Entre las tareas a desarrollar se encuentran la definición del producto a obtener, la planificación de recursos humanos, temporal, de migración de contenido, económica, de seguimiento y control, el plan de contingencia, la planificación de formación y evaluación, la estrategia de migración a seguir y la mejora el sistema. A continuación detallamos dichas tareas.

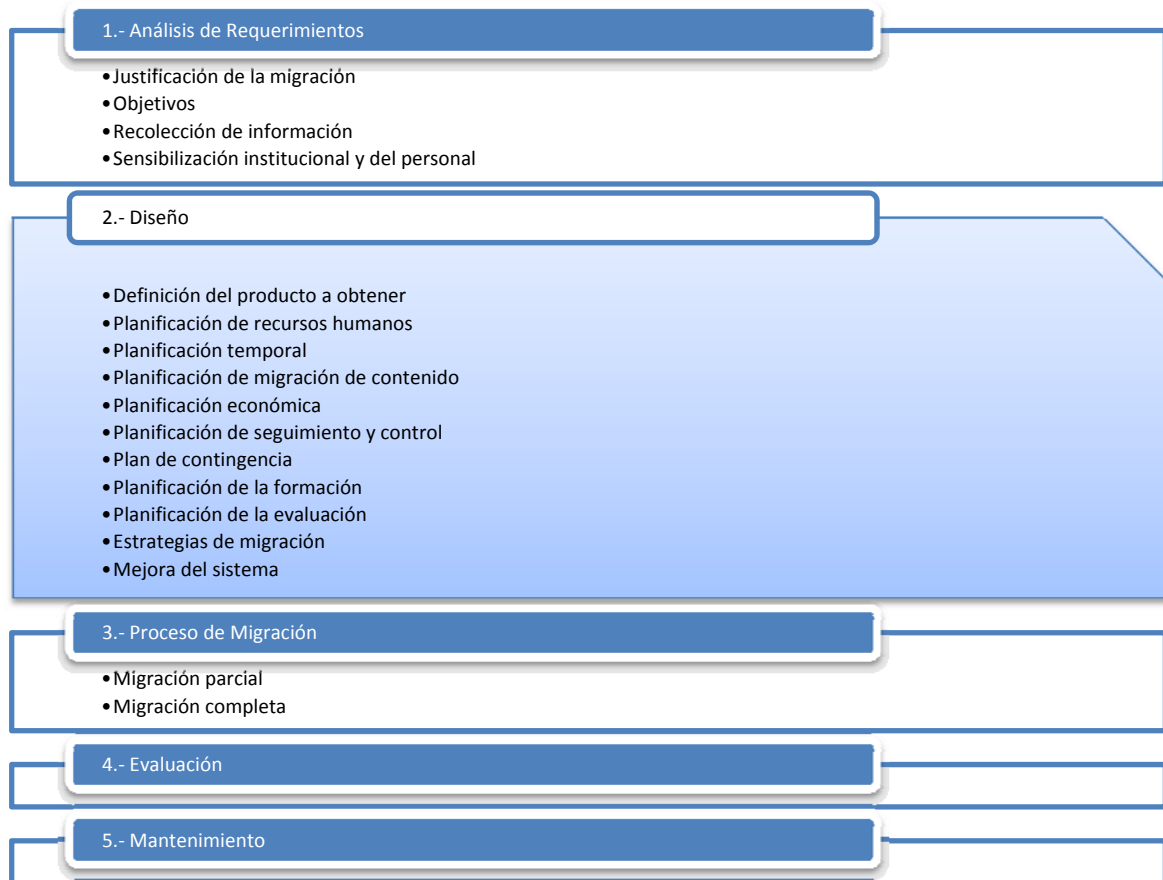


Figura 4.17. La fase de diseño es la segunda fase de la metodología

a) *Definición del producto a obtener*

El producto sobre el cual se inicia el proceso de migración es el Sistema de Recomendación de Restaurantes Georreferenciados REJA descrito en la sección 4.1 de este capítulo. El producto a obtener, por tanto, será este mismo sistema con las mismas funcionalidades y características, pero basado en una versión más actual del sistema de gestión de contenidos sobre el que está desarrollado. La versión de este sistema será la 1.5.26. Si no se modificase la plantilla del sitio web, la apariencia debería ser igual tanto en el sistema actual como en el nuevo, pero todo el núcleo interno del CMS quedará actualizado, con todas las ventajas que ello conlleva.

Este cambio de sistema a una versión más actual, conlleva una serie de nuevas características que afectan al usuario. Pasamos a describir las más importantes:

- Mejora del manejo de contraseñas perdidas. Joomla 1.5 facilita al usuario el obtener sus contraseñas perdidas en base a su nombre de usuario o dirección de correo electrónico.
- Instalador unificado para componentes, módulos, plugins, plantillas e idiomas.
- Una única área de edición para escribir el contenido de introducción y principal.
- Nuevo administrador de imágenes que permite insertar imágenes *al vuelo*.
- Subida de múltiples archivos a la vez.
- Nuevos ajustes de configuración.

b) *Planificación de recursos humanos*

En este proyecto, los diferentes perfiles que se definen a la hora de la planificación de recursos humanos que van a realizar la migración, recaen únicamente sobre dos personas. La primera persona, que en este caso sería el tutor del proyecto, actúa como director interno del proyecto, realizando las tareas de definir exactamente el producto a obtener, organizar el trabajo, conocer qué se espera de la migración y buscar el cumplimiento del calendario y el presupuesto previsto. Una segunda persona, el alumno que realiza este proyecto, actúa como especialista de contenido, tanto como publicador y editor, al igual que desarrollador. La tarea de formación la realizarán en un principio el alumno y en adelante el personal de la organización encargado de su mantenimiento.

c) *Planificación temporal*

Esta planificación se ha considerado que no era necesaria. Esto es debido a que este proceso de migración está localizado en el desarrollo de un proyecto más extenso, en el cual se va definiendo una metodología de migración de sitios web a la vez que se va llevando a la práctica el mismo, lo cual retrasa considerablemente el proceso de migración no pudiéndose definir con antelación la temporalidad del mismo.

d) *Planificación de migración de contenido*

Como se comentó en el capítulo 2, en la fase de “*Planificación de migración de contenido*”, una tarea previa a la migración de contenido es revisar el contenido del sitio web para prescindir de aquel que sea redundante, que contenga información caducada o por alguna necesidad de calidad. Al tratarse de un CMS, también hay que comprobar que todas las plantillas y extensiones instaladas se están utilizando, para prescindir de las que no lo estén.

Vamos a empezar por revisar las plantillas instaladas. Las plantillas instaladas actualmente en el sitio web se pueden ver desde el Backend en Sitio → Administrador de Plantillas → Plantillas del Sitio. Como se muestra en la Figura 4.18, de todas las plantillas instaladas solamente se está utilizando una, por lo que el resto habría que borrarlas para evitar arrastrarlas a lo largo del proceso de migración. Esto puede suceder porque hayan sido instaladas y probadas diferentes plantillas para finalmente seleccionar solamente una y no se ha tenido en cuenta la desinstalación de las que no van a ser usadas.

| # | Nombre | Predeterminado | Asignado | Autor | Versión | Fecha | Web del autor |
|----|-------------------------|----------------|----------|--------------------------|---------|---------------------|---------------------------------------|
| 1 | Apple | | | V.Dehyadgari | 1.0 | 28.09.2006 | www.Studentsdesign.de |
| 2 | balanced | | | DesignForJoomla.com | 1.0 | 28th August 2006 | designforjoomla.com |
| 3 | church_2006 | | | Ben Brown | 1.0 | 11/11/2005 | www.churchministriesonline.com |
| 4 | citron | | | Dirk Winkelbach | 1.1 | 12/11/06 | http://www.winkelbach.de |
| 5 | crossover | | | Robert Rajda | 1.0 | 2006-11-05 | http://www.pc-erlernen.de |
| 6 | dropshadow+10 | | | duvien | 1.0 | 14/03/2007 | http://www.duvien.com |
| 7 | fantasia | | | DesignForJoomla.com | 1.0 | 16th September 2006 | designforjoomla.com |
| 8 | Feel The World 3 | | | Joomlation.eu | 1.0 | 05 dec 2006 | http://www.joomlation.eu |
| 9 | flexi_orange_simple_1 | | | syahzul | 1.0 | 8/16/06 | http://www.flexisaga.com |
| 10 | flowers | | | Dirk Winkelbach | 1.1 | 12/11/06 | http://www.winkelbach.de |
| 11 | frozen_green | | | Anch - Gorsk.net Studio | 1.0 | december 2006 | http://gorsk.net |
| 12 | frozen_ry | | | Anch - Gorsk.net Studio | 1.0 | december 2006 | http://gorsk.net |
| 13 | gossip_bloop | | | Rene Wandel | 1.0 | 08.03.2007 | http://www.gossip.de |
| 14 | je_redenvy | | | joomlaenvy.com | 1.0 | 17 Feb 07 | joomlaenvy.com |
| 15 | 11 Beautiful Day | | | JoomlaJunkie | 1.0 | 07 Mar 07 | http://www.joomlajunkie.com |
| 16 | 11 Pink Sunset | | | JoomlaJunkie | 1.0 | 08 Mar 07 | http://www.joomlajunkie.com |
| 17 | joomladots | | | Be Professional Networks | 1.0 | 08/11/06 | http://www.beprofessional.net |
| 18 | js_industry | | | Joomlashack.com | 1.0 | | www.joomlashack.com |
| 19 | it_millennium_earth | | | Peek | 0.1 | 9/maj/2004 | www.joomla-templates.com |
| 20 | it_millennium_expresso | | | Peek | 0.1 | 9/maj/2004 | www.joomla-templates.com |
| 21 | it_millennium_flower | | | Peek | 0.1 | 9/maj/2004 | www.joomla-templates.com |
| 22 | it_millennium_internet | | | Peek | 0.1 | 9/maj/2004 | www.joomla-templates.com |
| 23 | it_millennium_keyboard | | | Peek | 0.1 | 9/maj/2004 | www.joomla-templates.com |
| 24 | it_millennium_lightning | | | Peek | 0.1 | 9/maj/2004 | www.joomla-templates.com |
| 25 | it_millennium_train | | | Peek | 0.1 | 9/maj/2004 | www.joomla-templates.com |
| 26 | it_simplicity_red | | | peek | 0.1 | 26/02/05 | http://www.joomla-templates.com |
| 27 | lady_beetle | | | R.Balvardi | 1.0 | 31.03.2006 | www.mitra.ir |
| 28 | lavinya3 | | | Murat Esgin | 3.0 | 18/11/06 | www.lavinya.net |
| 29 | lichtenart_simple | | | Sascha Lichtenstein | 1.0 | 11/18/2006 | http://www.lichtenart.com |
| 30 | lovely_1ion | | | DesignForJoomla.com | 1.0 | 06th October 2006 | designforjoomla.com |
| 31 | madeyourweb | | | Marc Hinse | 1.3 | 15 09 2005 | http://www.madeyourweb.com |
| 32 | Metropolitan | | | Intuitive Designs | 1.0 | 18/10/05 | http://www.intuitivedesigns.net |
| 33 | moz_dev | | ✓ | Daniel Siekiera | 1.0 | 30 08 06 | http://www.siekiera-online.de |
| 34 | nautica liquid | | | Oribium AB | 0.1 | 10/20/2006 | http://www.oribium.net/ |
| 35 | pWDNr5 | | | pWD-privat WebDesign | 1.0 | 10.09.2006 | http://www.pwd-webdesign.at |
| 36 | pWDNr9 | | | pWD-privat WebDesign | 1.0 | 24.02.2007 | http://www.pwd-webdesign.at |
| 37 | rhuk_solarflare_ii | | | rhuk | 2.2 | 10/20/05 | http://www.rockettheme.com |
| 38 | rideahorse01 | | | A.Döhling | 0.2 | 27.02.2007 | http://www.mobiler-reitunterricht.com |
| 39 | Rocky Mountain Sunrise | | | Randy Carver | 0.92 | August 20, 2005 | www.carver-family.com |
| 40 | Silent_Power | | | level09 studios | 2.0 | june 2006 | http://www.level09.com |
| 41 | spanish_red | | | joomlaspanish | 1.0.9 | 3/06/06 | http://www.joomlaspanish.org |
| 42 | travelblog_1 | | | Jonny Wüthrich | 1.0 | Sommer 06 | www.wuehtrichs.ch |
| 43 | viridias | | | made by volkann | 1.0 | Dezember 2006 | http://www.arkadasca.de.tc |
| 44 | vista | | | R.Balvardi | 1.0 | 20.11.2006 | www.joomlanarket.de |
| 45 | webnikablueice | | | Monika Broeg | 1.0 | 30. August 2006 | www.webnika.ch |
| 46 | Webverwalter-22 | | | Thomas Kirschnick | 1.0 | 08.09.2006 | http://www.webverwalter.com |
| 47 | WhiteBerry Blue | | | WhiteBerry | 1.3 | 22. Februar 2007 | http://www.whiteberry.de |

Figura 4.18. Listado de todas las plantillas instaladas en el sistema

A continuación se realiza un inventario de las extensiones instaladas en el CMS para desinstalar aquellas que en algún momento han sido instaladas pero que no tienen ninguna funcionalidad en la actualidad. Para detectarlas se requiere de un gran conocimiento del sitio

para poderlas identificar y desinstalar en su caso. Este es un proceso manual y minucioso, pero imprescindible en el proceso de limpieza del sitio.

En la Figura 4.19 se muestran los componentes instalados, accesibles desde el menú Componentes.



Figura 4.19. Listado de componentes instalados

En la Figura 4.20 se muestran los módulos instalados, accesibles desde el menú Módulos.

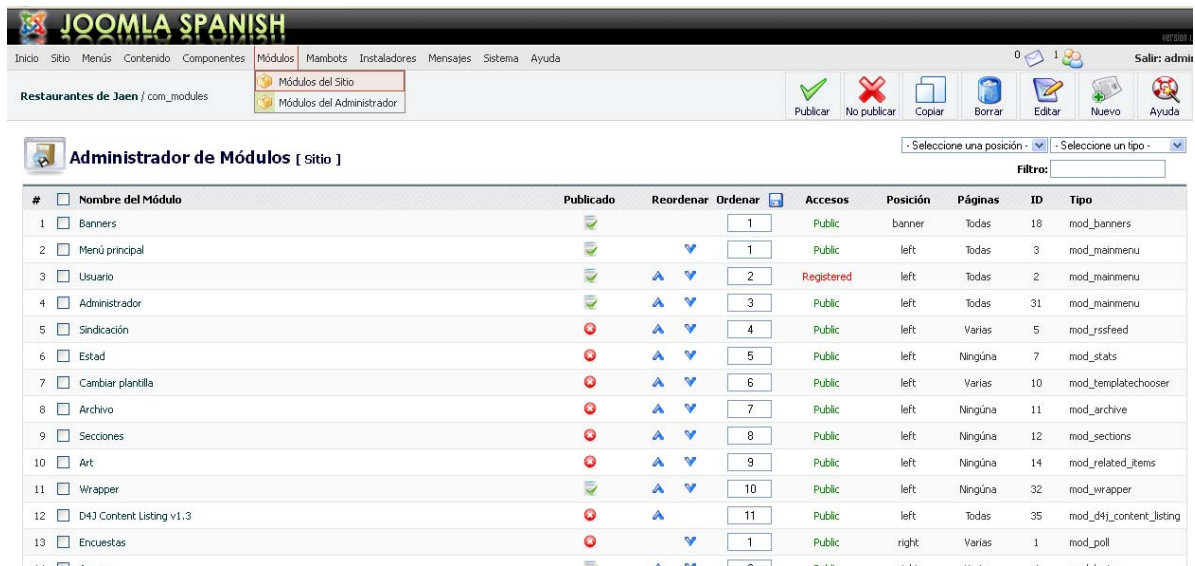


Figura 4.20. Listado de módulos instalados en el sistema

En la Figura 4.21 se muestran las Mambots instalados, accesibles desde el menú Mambots. Los Mambots eran un tipo de extensión utilizadas en versiones de Joomla anteriores a la 1.5.

The screenshot shows the Joomla! Mambots Administrator interface. At the top, there is a navigation menu with options like 'Inicio', 'Sitio', 'Menús', 'Contenido', 'Componentes', 'Módulos', 'Mambots', 'Instaladores', 'Mensajes', 'Sistema', and 'Ayuda'. Below the menu, there are icons for 'Publicar', 'No publicar', 'Borrar', 'Editar', 'Nuevo', and 'Ayuda'. The main content area is titled 'Administrador de Mambots [Sitio]' and contains a table with the following columns: 'Nombre del mambot', 'Publicar', 'Reordenar', 'Orden', 'Accesos', 'Tipos', and 'Archivos'. The table lists 16 mambots with their respective status, order, and file names.

| <input type="checkbox"/> | Nombre del mambot | Publicar | Reordenar | Orden | Accesos | Tipos | Archivos |
|--------------------------|-----------------------------------|----------|-----------|--------|---------|-------------|---------------------|
| <input type="checkbox"/> | MOS Image | | | -10000 | Public | content | mosimage |
| <input type="checkbox"/> | Soporte código | | | 1 | Public | content | moscode |
| <input type="checkbox"/> | MjazTools Personalize | | | 2 | Public | content | personalize |
| <input type="checkbox"/> | Simple Image Gallery Plugin | | | 3 | Public | content | plugin_jw_sig |
| <input type="checkbox"/> | AllVideos Plugin | | | 4 | Public | content | plugin_jw_allvideos |
| <input type="checkbox"/> | Legacy Mambot Include | | | 5 | Public | content | legacybots |
| <input type="checkbox"/> | SEF | | | 6 | Public | content | mossef |
| <input type="checkbox"/> | MOS Calificación | | | 7 | Public | content | mosvote |
| <input type="checkbox"/> | Email Cloaking | | | 8 | Public | content | mosemailcloak |
| <input type="checkbox"/> | GeSHI | | | 9 | Public | content | geshi |
| <input type="checkbox"/> | Cargar la posición de los módulos | | | 10 | Public | content | mosloadposition |
| <input type="checkbox"/> | MOS Paginación | | | 11 | Public | content | mospaging |
| <input type="checkbox"/> | Sin editor WYSIWYG | | | 1 | Public | editors | none |
| <input type="checkbox"/> | Editor WYSIWYG TinyMCE | | | 2 | Public | editors | tinymce |
| <input type="checkbox"/> | JCE Editor Mambot | | | 3 | Public | editors | jce |
| <input type="checkbox"/> | MOS Image Editor Button | | | 0 | Public | editors-xtd | mosimage.btn |

Figura 4.21. Listado de mambots instalados

- Revisamos ahora los enlaces de cada menú, para prescindir de los que no estén publicados y no tengan utilidad, bien sea porque se creó para una prueba y se obvió la eliminación o porque en un inicio sí tenía utilidad pero ya no la tiene (ver Figura 4.22).

| # | Artículo del menú | Publicado | Reordenar | Orden | Accesos | Itemid | Tipos | CID |
|----|----------------------|-----------|-----------|-------|---------|--------|---------------------------------|-----|
| 1 | Inicio | | | 1 | Public | 1 | Componente | 10 |
| 2 | Restaurantes | | | 2 | Public | 29 | Enlace - Url | 0 |
| 3 | Restaurantes (1) | | | 3 | Public | 28 | Enlace - Url | 0 |
| 4 | Licencia Joomla! | | | 4 | Public | 6 | Enlace - Contenido estático | 11 |
| 5 | Noticias | | | 5 | Public | 2 | Tabla - Sección de contenido | 1 |
| 6 | Bloc | | | 6 | Public | 9 | Bloque - Sección de contenido | 0 |
| 7 | Recomendaciones | | | 7 | Public | 43 | Enlace - Url | 0 |
| 8 | Enlaces | | | 8 | Public | 23 | Componente | 4 |
| 9 | Contactar | | | 9 | Public | 3 | Componente | 7 |
| 10 | Buscar | | | 10 | Public | 5 | Componente | 16 |
| 11 | Noticias externas | | | 11 | Public | 7 | Componente | 12 |
| 12 | FAQ | | | 12 | Public | 25 | Tabla - Categoría de contenidos | 7 |
| 13 | Wrapper | | | 13 | Public | 8 | Wrapper | 0 |
| 14 | Buscar Restaurante | | | 14 | Public | 35 | Enlace - Url | 0 |
| 15 | Mapa de Restaurantes | | | 15 | Public | 36 | Enlace - Url | 0 |

Figura 4.22. Listado de menús creados

Una vez que se ha realizado la planificación de la limpieza de nuestro sitio respecto a la información que no queremos migrar, pasaríamos a planificar la manera de llevar a cabo la migración. En nuestro caso, la realizaremos de manera automática utilizando el componente mtwmigrator, el cual es descrito en el Anexo II.

e) Planificación económica

Al trabajar con software libre, estamos prescindiendo de algunos costes directos, como pueden ser el sistema gestor de contenidos CMS Joomla y el sistema de gestión de bases de datos MySQL. El servidor de alojamiento del sitio web, aún siendo distinto al actual, no conllevaría un incremento del coste de la migración al no tener que ser financiado por este proyecto. El coste del personal que realiza la migración en este proyecto no sería aplicable.

Existen unos costes indirectos que en un principio no podemos preverlos ni tasarlos que provienen de errores o problemas durante el proceso de migración, como pueden ser los costes de mantenimiento, de seguridad, de disponibilidad, etc.

f) Planificación del seguimiento y control

El objetivo primordial es tener controlado en todo momento el proyecto, para poder detectar cualquier desviación en la planificación, como podría ser un retraso en alguna de las tareas y poder tomar las acciones correctivas necesarias. En nuestro caso, al no haber realizado una planificación temporal por los motivos expuestos anteriormente, no es posible realizar un seguimiento de la misma.

g) Plan de contingencia

En el Anexo I, se han descrito diversos métodos para realizar el proceso de las copias de seguridad, por lo que no será necesario describirlo aquí. En nuestro caso, vamos a aplicar el método manual, el cual ejecutaremos más adelante en la fase de proceso de migración.

h) Planificación de la formación

En este proyecto sólo se va a realizar formación a los administradores futuros del sitio web para que puedan llevar a cabo el mantenimiento del mismo. Como guía para dicha formación, será muy útil la memoria de este proyecto, en la cual se explica en detalle cada paso realizado, siguiendo la metodología definida. A los usuarios finales no será necesario ya que el sitio conservará la misma funcionalidad que el sitio antes de migrar, pero bajo una versión nueva del CMS.

i) Planificación de la evaluación

Para poder evaluar el proceso realizado de migración, debemos de llevar a cabo una serie de pruebas que nos garanticen la calidad del proceso. Para ello realizaremos los siguientes test al finalizar la migración:

Test 1: Registro válido de un usuario en el sistema

| | |
|--------------------|--|
| Requisito testeado | Registro de usuarios en el sistema |
| Acción | Un usuario introduce, con el formato correcto, los datos requeridos por el sistema y pulsa sobre el botón Aceptar. |
| Checkpoint1 | El sistema debe mostrar un mensaje de aceptación de registro y envía un correo electrónico al usuario con un enlace que permita activar el registro. |

Test 2: Registro incorrecto de un usuario en el sistema

| | |
|--------------------|--|
| Requisito testeado | Registro de usuarios en el sistema |
| Acción | Un usuario introduce todos los datos de forma correcta pero las contraseñas no coinciden y pulsa sobre el botón Aceptar. |
| Checkpoint1 | El sistema debe mostrar un mensaje de error informando que los campos de contraseña no coinciden. |

Test 3: Identificación de usuario

| | |
|--------------------|--|
| Requisito testeado | Obtención de los datos almacenados en el perfil del usuario |
| Acción | Un usuario introduce su nombre de usuario y contraseña y pulsa el botón Entrar. |
| Checkpoint1 | El sistema debe mostrar el menú de dicho usuario y una sección con sus datos personales. |

Test 4: Identificación errónea del usuario

| | |
|--------------------|---|
| Requisito testeado | Obtención de los datos almacenados en el perfil del usuario |
| Acción | Un usuario introduce su nombre de usuario y contraseña de forma incorrecta y pulsa el botón Entrar. |
| Checkpoint1 | El sistema debe mostrar un mensaje de error informando de que la contraseña es incorrecta. |

Test 5: Modificación de los datos de usuario

| | |
|--------------------|--|
| Requisito testeado | Modificación de los datos de los usuarios |
| Acción | Un usuario modifica sus datos pulsa el botón Aceptar. |
| Checkpoint1 | El sistema debe guardar los cambios en la base de datos y mostrar un mensaje advirtiendo al usuario de que sus datos han sido modificados con éxito. |

Test 6: Obtención de información básica de un restaurante

| | |
|--------------------|--|
| Requisito testeado | Acceso a la información los restaurantes |
| Acción | Un usuario pulsa sobre la opción Restaurantes del menú principal. |
| Checkpoint1 | El sistema debe mostrar una pantalla que permita ver la lista de restaurantes ordenada por orden alfabético o por tipo de establecimiento. |
| Acción | El usuario pulsa sobre la opción Ordenados Alfabéticamente |
| Checkpoint2 | El sistema debe mostrar una lista con todos los restaurantes almacenados en la base de datos, ordenados de forma alfabética. |
| Acción | El usuario selecciona un restaurante de la lista. |
| Checkpoint3 | El sistema debe mostrar una pantalla con la información básica del restaurante seleccionado. |

Test 7: Puntuación de un restaurante

| Requisito testeado | Puntuación de los restaurantes |
|--------------------|--|
| Acción | Un usuario identificado pulsa sobre la opción Añadir puntuación del menú de usuario. |
| Checkpoint1 | El sistema debe mostrar una pantalla donde aparecerá una lista desplegable con todos los restaurantes almacenados en la base de datos, una serie de radio buttons que se corresponden con la puntuación que se puede dar al restaurante (MM: Muy Malo, M: Malo, R: Regular, B: Bueno, MB: Muy Bueno) y el botón puntuar. |
| Acción | El usuario selecciona un restaurante de la lista, pulsa sobre el botón que se corresponde con la puntuación que quiere dar al restaurante y hace clic sobre el botón puntuar |
| Checkpoint2 | El sistema debe añadir una nueva puntuación al perfil del usuario. |

Test 8: Modificación de la puntuación de un restaurante

| Requisito testeado | Modificación de las puntuaciones de los restaurantes |
|--------------------|---|
| Acción | Un usuario identificado pulsa sobre la opción Mis Puntuaciones del menú de usuario. |
| Checkpoint1 | El sistema debe mostrar una pantalla donde aparecerá el nombre de los restaurantes ya puntuados y la puntuación que el usuario ha hecho sobre ellos. |
| Acción | El usuario pulsa sobre un restaurante. |
| Checkpoint2 | El sistema debe mostrar una pantalla con el nombre del restaurante, una serie de radio buttons que se corresponden con la puntuación que se puede dar al restaurante (MM: Muy Malo, M: Malo, R: Regular, B: Bueno, MB: Muy Bueno) y el botón puntuar. |
| Acción | El usuario pulsa sobre el radio button que se corresponde con la nueva percepción que tiene del restaurante y posteriormente sobre el botón puntuar. |

| | |
|-------------|--|
| Checkpoint3 | El sistema debe modificar la puntuación del restaurante. |
|-------------|--|

Test 9: Recomendación de restaurantes para usuarios no registrados en el sistema utilizando la opción de restaurante conocido.

| | |
|--------------------|---|
| Requisito testeado | Obtención de recomendaciones de restaurantes |
| Acción | Un usuario pulsa sobre la opción Recomendaciones del menú principal. |
| Checkpoint1 | El sistema debe mostrar una pantalla donde se mostrarán dos opciones de recomendación: Recomendaciones basadas en un restaurante conocido y Recomendaciones por filtro. |
| Acción | El usuario pulsa la opción Recomendaciones basadas en un restaurante conocido. |
| Checkpoint2 | El sistema debe mostrar una pantalla con la lista de restaurantes almacenados en la base de datos. |
| Acción | El usuario elige un restaurante y pulsa sobre el botón Obtener recomendaciones. |
| Checkpoint3 | El sistema debe mostrar una pantalla con los restaurantes que más se asemejen al restaurante seleccionado. |
| Acción | El usuario pulsa sobre un restaurante. |
| Checkpoint4 | El sistema debe mostrar una pantalla donde se mostrará la información básica del restaurante seleccionado. |

Test 10: Recomendación de restaurantes para usuarios no registrados en el sistema utilizando la opción de recomendación por filtro.

| | |
|--------------------|--|
| Requisito testeado | Obtención de recomendaciones de restaurantes |
| Acción | Un usuario pulsa sobre la opción Recomendaciones del menú principal. |

| | |
|-------------|--|
| Checkpoint1 | El sistema debe mostrar una pantalla donde se mostrarán dos opciones de recomendación: Recomendaciones basadas en un restaurante conocido y Recomendaciones por filtro. |
| Acción | El usuario pulsa la opción Recomendaciones por filtro. |
| Checkpoint2 | El sistema debe mostrar una pantalla con una serie de listas desplegadas que permitirán seleccionar el intervalo de precios del menú, el tipo de cocina y tipo de establecimiento. |
| Acción | El usuario elige el precio, el tipo de cocina y el tipo de establecimiento que mejor se adapta a sus preferencias y pulsa sobre el botón Obtener recomendaciones. |
| Checkpoint3 | El sistema debe mostrar una pantalla con los restaurantes que más se asemejen al restaurante seleccionado. |
| Acción | El usuario pulsa sobre un restaurante. |
| Checkpoint4 | El sistema debe mostrar una pantalla donde se mostrará la información básica del restaurante seleccionado. |

Test 11: Recomendación rápida de restaurantes para usuarios identificados

| | |
|--------------------|--|
| Requisito testeado | Obtención de recomendaciones de restaurantes |
| Acción | Un usuario identificado pulsa sobre la opción Recomendaciones del menú principal. |
| Checkpoint1 | El sistema debe mostrar una pantalla con una lista de todos los restaurantes que hay almacenados en la base de datos y un botón "Seleccionar Restaurante". |
| Acción | El usuario selecciona un restaurante y pulsa sobre el botón "Seleccionar Restaurante". |
| Checkpoint2 | El sistema debe mostrar una pantalla con tres restaurantes que están relacionados con el seleccionado y sus correspondientes listas desplegadas que permitirán especificar la preferencia del usuario de estos con respecto al restaurante seleccionado. |

| | |
|-------------|--|
| Acción | El usuario selecciona un valor de cada lista y pulsa sobre el botón obtener recomendaciones. |
| Checkpoint3 | El sistema debe mostrar una pantalla con los restaurantes que más se asemejen al restaurante seleccionado. |
| Acción | El usuario pulsa sobre un restaurante. |
| Checkpoint4 | El sistema debe mostrar una pantalla donde se mostrará la información del restaurante seleccionado. |

Test 12: Recomendación de restaurantes para usuarios identificados usando el módulo colaborativo

| | |
|--------------------|--|
| Requisito testeado | Obtención de recomendaciones de restaurantes |
| Acción | Un usuario identificado pulsa sobre la opción Mis Recomendaciones del Menú de Usuario. |
| Checkpoint1 | El sistema debe mostrar una pantalla con una lista de 10 restaurantes recomendados de acuerdo al perfil del usuario. |
| Acción | El usuario hace clic sobre un restaurante. |
| Checkpoint2 | El sistema debe mostrar una pantalla donde se mostrará la información del restaurante seleccionado. |

Test 13: Geolocalización de un restaurante recomendado.

| | |
|--------------------|--|
| Requisito testeado | Geolocalización de los restaurantes |
| Acción | Un usuario identificado accede a la información de un restaurante recomendado y pulsa sobre el botón Georreferenciación. |
| Checkpoint1 | El sistema debe mostrar una nueva pantalla (el módulo SIG), con la localización del restaurante sobre un mapa virtual. |

Test 14: Geolocalización de un restaurante recomendado, con una dirección almacenada en la base de datos que es errónea.

| | |
|--------------------|---|
| Requisito testeado | Geolocalización de los restaurantes |
| Acción | Un usuario identificado accede a la información de un restaurante recomendado que tiene una dirección que no es correcta y pulsa sobre el botón Georreferenciación. |
| Checkpoint1 | El sistema debe mostrar un aviso advirtiendo de que la dirección del restaurante no se puede geocodificar. |

Test 15: Obtención del camino más corto desde la dirección del usuario hasta el restaurante recomendado

| | |
|--------------------|---|
| Requisito testeado | Obtención del camino mínimo necesario para llegar a un restaurante desde la dirección del usuario |
| Acción | Un usuario identificado accede a la información de un restaurante recomendado y pulsa sobre el botón Georreferenciación. |
| Checkpoint1 | El sistema debe mostrar una nueva pantalla (el módulo SIG), con la localización del restaurante sobre un mapa virtual. |
| Acción | El usuario pulsa sobre el botón "Ruta desde tu dirección". |
| Checkpoint2 | El sistema debe mostrar sobre el mapa virtual, una línea que representa la ruta más corta desde la dirección del usuario hasta el restaurante, y una serie de instrucciones para completar el trayecto. |

Test 16: Cálculo de rutas

| | |
|--------------------|--|
| Requisito testeado | Cálculo de rutas entre dos puntos |
| Acción | Un usuario identificado accede a la información de un restaurante recomendado y pulsa sobre el botón Georreferenciación. |
| Checkpoint1 | El sistema debe mostrar una nueva pantalla (el módulo SIG), con la localización del restaurante sobre un mapa virtual. |

| | |
|-------------|--|
| Acción | El usuario pulsa sobre el botón “Cálculo de rutas” y hace clic sobre el punto de origen y el de destino que desee. |
| Checkpoint2 | El sistema debe mostrar sobre el mapa virtual, una línea que representa la ruta más corta desde la dirección del punto de origen hasta el punto de destino, y una serie de instrucciones para completar el trayecto. |

Test 17: Consultas de los lugares para visitar que están a menos de 200 metros del restaurante recomendado

| | |
|--------------------|---|
| Requisito testeado | Consultas sobre el mapa |
| Acción | Un usuario identificado accede a la información de un restaurante recomendado y pulsa sobre el botón Georreferenciación. |
| Checkpoint1 | El sistema debe mostrar una nueva pantalla (el módulo SIG), con la localización del restaurante sobre un mapa virtual. |
| Acción | El usuario pulsa sobre el botón “Consultas” y posteriormente sobre “Consultas sobre el mapa” |
| Checkpoint2 | El sistema debe mostrar una sección que permita especificar los criterios de búsqueda deseados por el usuario. |
| Acción | El usuario pulsa sobre el checkbox que se corresponde con “lugares para visitar” y pulsa con el botón derecho sobre el punto donde se encuentra situado el restaurante recomendado. |
| Checkpoint3 | El sistema debe mostrar un menú contextual con una serie de opciones de consultas por distancia. |
| Acción | El usuario hace clic sobre la opción Consultas 100 metros. |
| Checkpoint4 | El sistema debe identificar y mostrar los lugares de interés que cumplen las restricciones impuestas por el usuario. |

j) Estrategia de migración

En nuestro caso la estrategia de migración seguida será la migración parcialmente automatizada. Una parte del contenido del sitio web será migrado ayudándonos del componente *mtwmigrator*, descrito en el Anexo II. Existe una importante parte de nuestro sitio web que no podrá ser migrado con esta extensión, de modo que habrá que realizarla de manera manual. Este proceso lo describimos en detalle en la siguiente fase de migración completa.

Una migración parcialmente automatizada realiza una serie de procesos de manera automática ayudándose de una extensión, pero quedarán otra serie de tareas que se deberán llevar a cabo de manera manual. A continuación, se enumeran las tareas que se realizan de manera automatizada y las de forma manual:

De forma automatizada:

- Migrar todos los registros del contenido del núcleo en la base de datos (ejemplo: contenido, enlaces web, banners, contactos, noticias externas, encuestas y usuarios).
- Reconstruir los menús en la nueva estructura de menús con los artículos de menú que enlazan a contenido del núcleo (se excluyen los componentes desarrollados por terceros).
- Trasladar todos los registros de módulos del núcleo desde el sitio antiguo a las tablas de módulos del nuevo sitio. Los registros originales de los módulos se establecen como 'No publicados'. Se conservan todos los ajustes de configuración. El usuario necesitará reconfigurar los ajustes de posición, orden y publicación.

De forma manual

- Migrar los registros de la base de datos para extensiones de terceros.
- Migrar los contenidos que no están incluidos en la base de datos, como las imágenes de banners, archivos, fotos...

k) Mejora del sistema

Como se comentó en el apartado de análisis de requerimientos, el proceso llevado a cabo al cambiar de un CMS sin soporte hacia este mismo pero en una versión actual, conlleva una serie de mejoras del sitio con mayor seguridad, limpieza de éste, mayor funcionalidad, etc. En este proyecto no se ha llevado a cabo ninguna mejora añadida del sistema.

4.3.3. Proceso de Migración

El proceso de migración es la tercera fase de la metodología de migración de CMS (ver Figura 4.23). Esta fase realiza las tareas de migración parcial y migración completa.

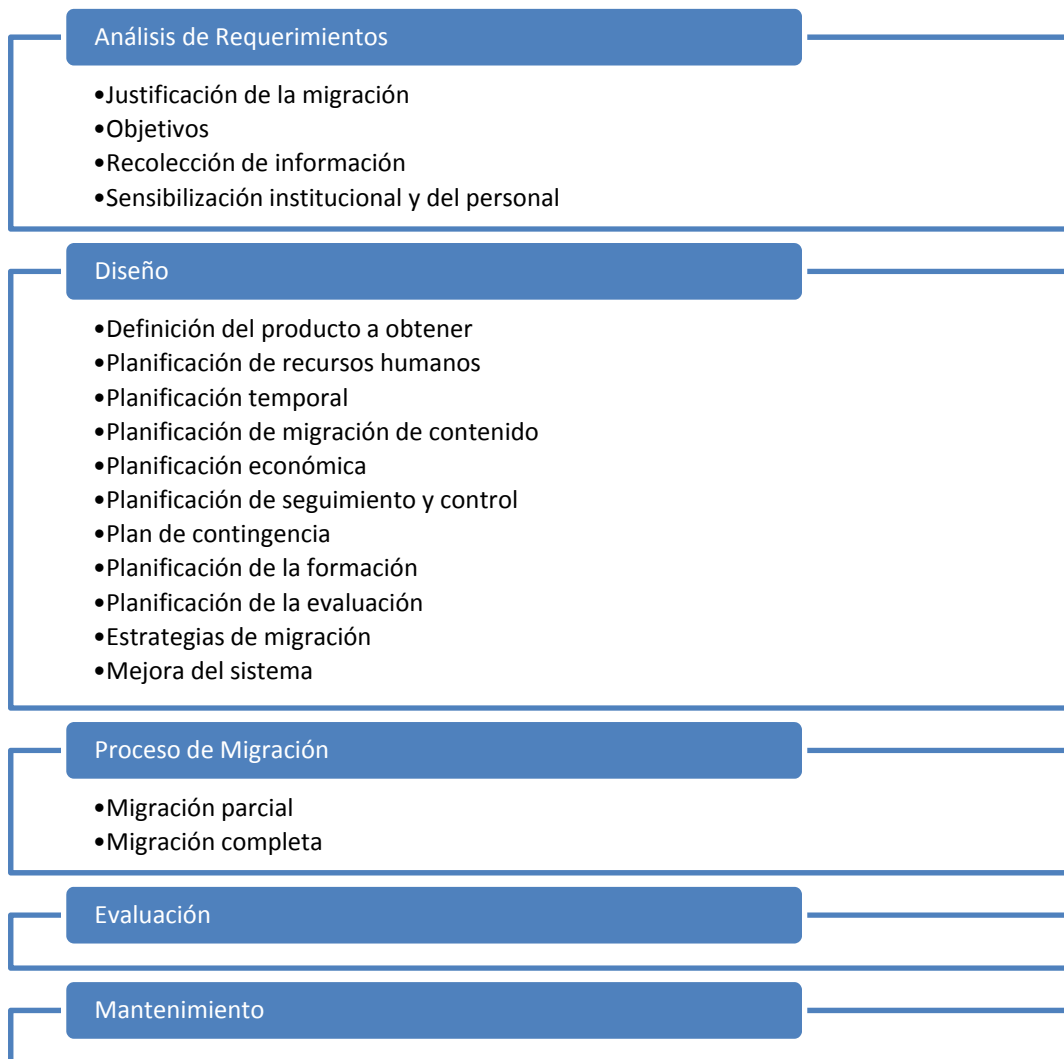


Figura 4.23. La fase de proceso de migración es la tercera fase de la metodología

4.3.3.1. Migración Parcial

En este proyecto se va a utilizar el esquema de migración en un único paso. En este esquema se va a migrar de una plataforma web a otra, sin realizar antes una migración parcial, debido a que el sistema sobre el que estamos trabajando no tiene excesivo tamaño como para realizar este paso intermedio.

Por tanto, pasamos a describir la ejecución del proceso de migración completa anteriormente analizada y planificada.

4.3.3.2. Migración Completa

A continuación se ejecutan algunas de las tareas documentadas en la *fase de diseño* siguiendo la secuencia descrita en la *fase de proceso de migración* de la metodología seguida.

a) Copias de seguridad

Comenzando con la ejecución del proceso de migración, el primer paso es la realización de la copia de seguridad. Para llevarla a cabo seguimos la planificación de contingencia realizada en la fase de *diseño*. Como el proceso elegido es el manual, se obtiene como resultado por una parte el archivo de respaldo de la base de datos, y por otro, todos los ficheros que componen el sitio.

Para comprobar que la copia de seguridad se ha realizado de manera correcta, se realiza la instalación en local del sitio web respaldado y se verifica su funcionamiento. Para la instalación en local, se presupone que se tiene instalado y configurado un servidor local tanto de Apache, PHP y MySQL, necesarios para el funcionamiento del CMS Joomla.

Una vez dispongamos de los dos elementos, archivos y base de datos, necesarios para realizar la nueva instalación procederemos a ello siguiendo los siguientes pasos:

1. Se copian los archivos descargados en la carpeta del servidor local, en nuestro caso en `c:\AppServ\www\reja`.

2. Crearemos la base de datos en el servidor. Al ser un servidor local, accedemos a la dirección `http://localhost/phpmyadmin` y se crea una base de datos nueva llamada `reja`.

3. Importar desde la nueva base de datos el contenido de la antigua. Dado que estamos haciendo una migración y no una instalación desde cero, no dispondremos de un asistente de instalación para configurar la base de datos ni introducir su contenido. En su lugar será necesario acceder al administrador de bases de datos phpMyAdmin para importar desde ahí los datos de nuestra copia de seguridad. Antes de realizar la importación deberemos asegurarnos de que la base de datos de destino está vacía o, al menos, que los nombres de las tablas existentes no son coincidentes con los nombres de las que vamos a importar. Al ser una base de datos recién creada, no existirá este problema, pero es recomendable comprobarlo para confirmarlo.

Para iniciar la importación pulsaremos en el botón importar del menú horizontal superior y buscaremos el archivo con la copia de la BD original descargada anteriormente en nuestro ordenador y pulsaremos en continuar (ver Figura 4.24).

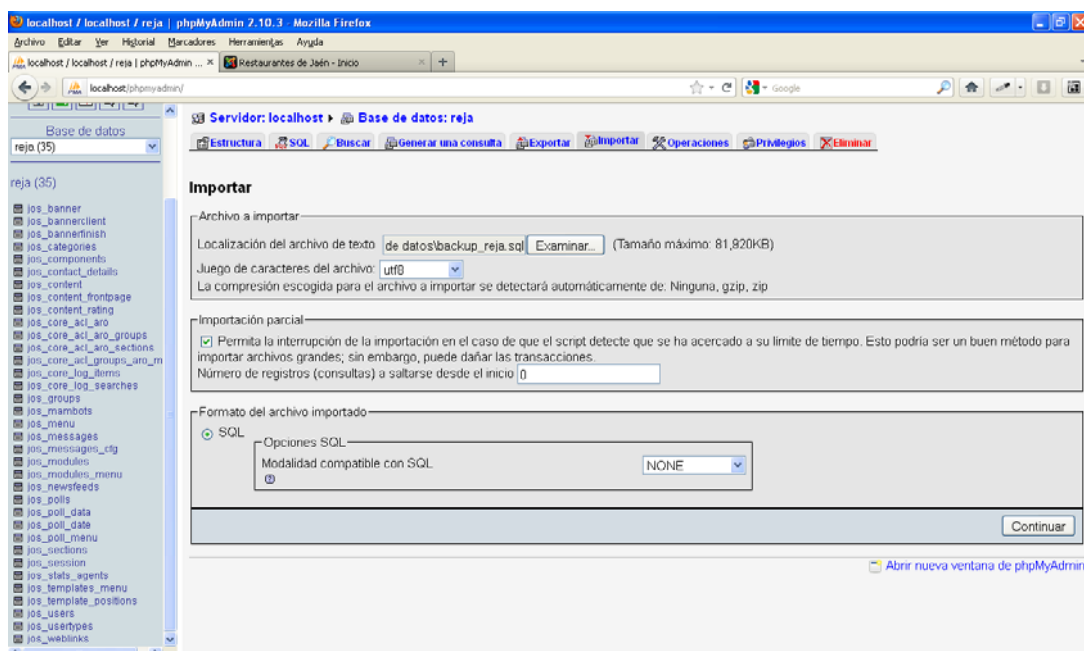


Figura 4.24. Opción de importación en phpMyAdmin

4. Conectar la base de datos con la aplicación modificando los datos del archivo configuration.php. El último paso del proceso será indicar en el archivo configuration.php

los datos de conexión con la nueva base de datos. Se edita el archivo y se buscan los parámetros a modificar con los datos del nuevo servidor y de la nueva base de datos.

Estos parámetros serán el tipo de base de datos (dbtype), alojamiento (host), nombre de la base de datos a utilizar (db) junto con su usuario de acceso (user) y contraseña (password). Existen otros campos que también habrá que tener en cuenta dependiendo de las características del sitio como son la configuración de la capa ftp, el prefijo de las tablas de la base de datos, la localización de los archivos temporales, etc.

Una vez realizados estos cuatro pasos, queda configurado en local la copia de seguridad de nuestro sitio web. Las pruebas a realizar para comprobar su correcto funcionamiento son las descritas en el *plan de evaluación*. El aspecto final del sitio web una vez instalado en local es el mostrado en la Figura 4.25.



Figura 4.25. Aspecto final del sitio web en local

En este punto, queda comprobado que la copia de seguridad está correctamente realizada, por lo que se puede continuar con la migración del sitio web. Esto da la garantía

de que si por algún motivo no se pudiera continuar con la migración, siempre se podrá volver al estado inicial del sistema.

b) Limpieza del estado actual

En esta fase se realiza la limpieza del contenido del sitio web que se desea migrar siguiendo la planificación de migración de contenido descrita en la fase de diseño. Se comentaba la necesidad de dividir el contenido del sitio web para prescindir del que no fuese necesario migrar debido a su redundancia, a su caducidad, o a cualquier necesidad de calidad.

Como se ha comentado en la fase de Planificación de Migración de Contenido, antes de realizar la migración de éste, hay que hacer una limpieza del sitio web actual. En este momento llevamos a la práctica la planificación realizada previamente, eliminando las plantillas, extensiones, enlaces, contenidos, etc. que ya no son necesarios en nuestro sitio web, por lo que no se desea trasladarlos en nuestro proceso de migración al nuevo sitio.

c) Ejecución de la estrategia de migración

Una vez que tenemos realizada la copia de seguridad de nuestro sitio web y la limpieza de contenido, procedemos a continuar con la migración. La migración inicialmente se va a realizar en el servidor local, y una vez que esté finalizada, se configura sobre el servidor final. Como paso inicial, la migración se realiza en local para solventar de forma más rápida y segura cualquier problema que pueda surgir. Al tener ya instalada en local una versión del sitio realizada anteriormente durante el proceso de Copia de Seguridad, utilizamos ésta durante el resto del proceso, para finalmente subirla al servidor definitivo.

Los pasos genéricos que se van a llevar a cabo en nuestro proceso de migración consisten en pasar del CMS Joomla versión 1.0.12, que es el sitio actual, a la versión 1.5.26. Para poder llevar a cabo dicha migración, primero hay que hacer un paso intermedio hacia la versión 1.0.15 y desde ésta a la 1.5.26 (ver Figura 4.26).

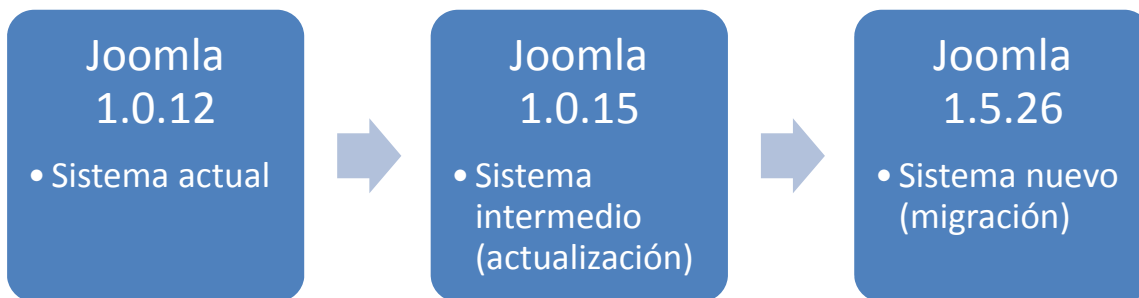


Figura 4.26. Esquema del proceso de migración realizado.

El proceso de pasar de la versión 1.0.12 hacia la 1.0.15 no es un proceso de migración, sino una actualización. Este proceso se detalla a continuación:

1) *Actualización de Joomla 1.0.12 a 1.0.15*

La actualización consiste en la sustitución de una serie de archivos del núcleo de Joomla por los actualizados, por tanto, este proceso no conlleva ninguna modificación en la base de datos. En principio esta debería ser una tarea inmediata, pero primero hay que descartar que se encuentre algún tipo de código propio incrustado en el núcleo de Joomla que pueda ser eliminado con la actualización. En el caso de darse tal situación, habría que estudiar cada caso en concreto y ofrecer una solución óptima para cada uno, manteniendo la funcionalidad del mismo.

En nuestra actualización se ha encontrado un problema en el componente `com_registration` de código propio incrustado, por lo que se ha tenido que subsanar el

mismo, explicado en el apartado 4.4 de este capítulo denominado *Procesos adicionales a la metodología propuesta para problemas específicos*.

Los archivos a sustituir son facilitados en la página web <http://joomla.org>. Desde esta página se busca el archivo de actualización indicando la versión inicial y hacia qué versión se desea actualizar, y se procede a la descarga de un archivo comprimido, en este caso el archivo Joomla_1.0.12_to_1.0.15-Stable-Patch_Package.zip. Posteriormente, hay que descomprimir este archivo y sustituir los archivos por los nuevos, quedando actualizado a la versión deseada.

En la Figura 4.27 se muestra una captura de pantalla con los archivos del sitio web, en la cual están seleccionados de azul los archivos que han sido modificados en esta actualización.

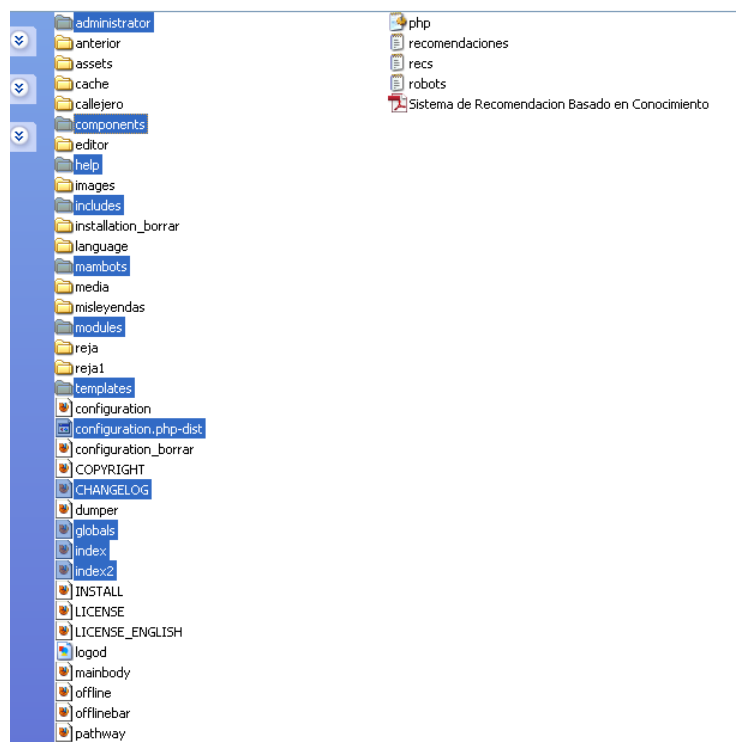


Figura 4.27. Archivos modificados durante la actualización

2) Migración de Joomla 1.0.15 a 1.5.26

Para realizar la migración entre estas dos versiones, existen diferentes métodos para llevarla a cabo. En la fase de diseño, en el apartado de Planificación de Migración de Contenido, se ha especificado llevarla a la práctica mediante el componente mtwmigrator, concretamente la versión 0.2.1. El procedimiento de configuración y ejecución genérico del mismo se ha detallado en el anexo II.

Aplicando este procedimiento sobre nuestro sitio, se obtiene la versión definitiva de Joomla 1.5.26. En la Figura 4.28 se muestra el momento de la configuración del componente en el cual se introducen los datos de alojamiento y base de datos de nuestro sitio, además de diferentes opciones de migración. Este componente ofrece la posibilidad de migrar también los datos de algunas extensiones muy populares externas al desarrollo de Joomla

The screenshot displays the configuration interface for the mtwmigrator component, divided into two main sections: 'Server Configuration' and '3rd Party Extensions'.

Server Configuration:

- Nombre del servidor: localhost
- Nombre de la base de datos: reja
- Nombre de usuario: root
- Contraseña: [Redacted]
- Prefix: jos_

Migration Configuration:

- Make Backup? No Sí
- Migrate Groups? No Sí
- Migrate Users? No Sí
- Migrate Sections? No Sí
- Migrate Categories? No Sí
- Migrate Content? No Sí
- Migrate Frontpage? No Sí
- Migrate Menus? No Sí
- Migrate Modules? No Sí
- Migrate Polls? No Sí
- Migrate Weblinks? No Sí
- Migrate Contacts? No Sí

3rd Party Extensions:

- Artio JoomSEF No Sí Artio JoomSEF Uninstalled
- Community Builder No Sí CB Uninstalled
- DocMan No Sí DocMan Uninstalled
- FacileForms No Sí FacileForms Uninstalled
- FireBoard No Sí FireBoard Uninstalled
- FireBoard to Kunena No Sí Kunena Uninstalled
- JomComment No Sí JomComment Uninstalled
- Letterman No Sí Letterman Uninstalled
- Virtuemart No Sí VM Uninstalled

Figura 4.28. Opciones de configuración del componente mtwmigrator

Una vez llevado a cabo este proceso, se muestra una pantalla (ver Figura 4.29) indicando los resultados obtenidos de la migración. Todos los apartados mostrados deben indicar que se han realizado correctamente para el correcto funcionamiento de la migración.

En caso contrario, habrá que revisar individualmente cada uno y detectar el posible problema para subsanarlo.



| Información | Status |
|--------------------------------|--------|
| Guardar base de datos anterior | OK! |
| Migrate Groups | OK! |
| Migrar Usuarios | OK! |
| Migrar Secciones | OK! |
| Migrar Categorías | OK! |
| Migrar Contenido | OK! |
| Migrar Contenidos de Portada | OK! |
| Migrar Menus | OK! |
| Migrar Módulos de Usuarios | OK! |
| Migrate Polls | OK! |
| Migrate Weblinks | OK! |
| Migrate Contacts | OK! |

| 3rd Party Extensions | Status |
|---------------------------|---------|
| Migrate Community Builder | Disable |
| Migrate Virtuemart | Disable |
| Migrate JomComment | Disable |
| Migrate Docman | Disable |
| Migrate FacileForms | Disable |
| Migrate Arto JoomSEF | Disable |
| Migrate FireBoard | Disable |
| Migrate Kunena | Disable |

Figura 4.29. Resultados obtenidos tras la migración

Finalizado el proceso de migración de contenido automático mediante el componente descrito, el aspecto de la página de inicio del sitio se muestra como en la Figura 4.30. El contenido ha sido migrado, pero se necesita revisarlo para mantener el sistema con las mismas funcionalidades y aspecto que el estado inicial. Para ello, desde el panel de administración del sitio, hay que revisar el estado de publicación de los artículos, componentes, módulos, plugins, plantillas, etc., hasta conseguir que tanto el sitio de inicio como el resultado de la migración sean iguales.



Figura 4.30. Aspecto actual de la página de inicio tras la migración

4.3.5. Evaluación

Una vez realizado el proceso de migración completo, es el momento de comprobar que se ha llevado a cabo correctamente (ver Figura 4.31). Para ello, se ejecuta el proceso de evaluación descrito en la fase de diseño.

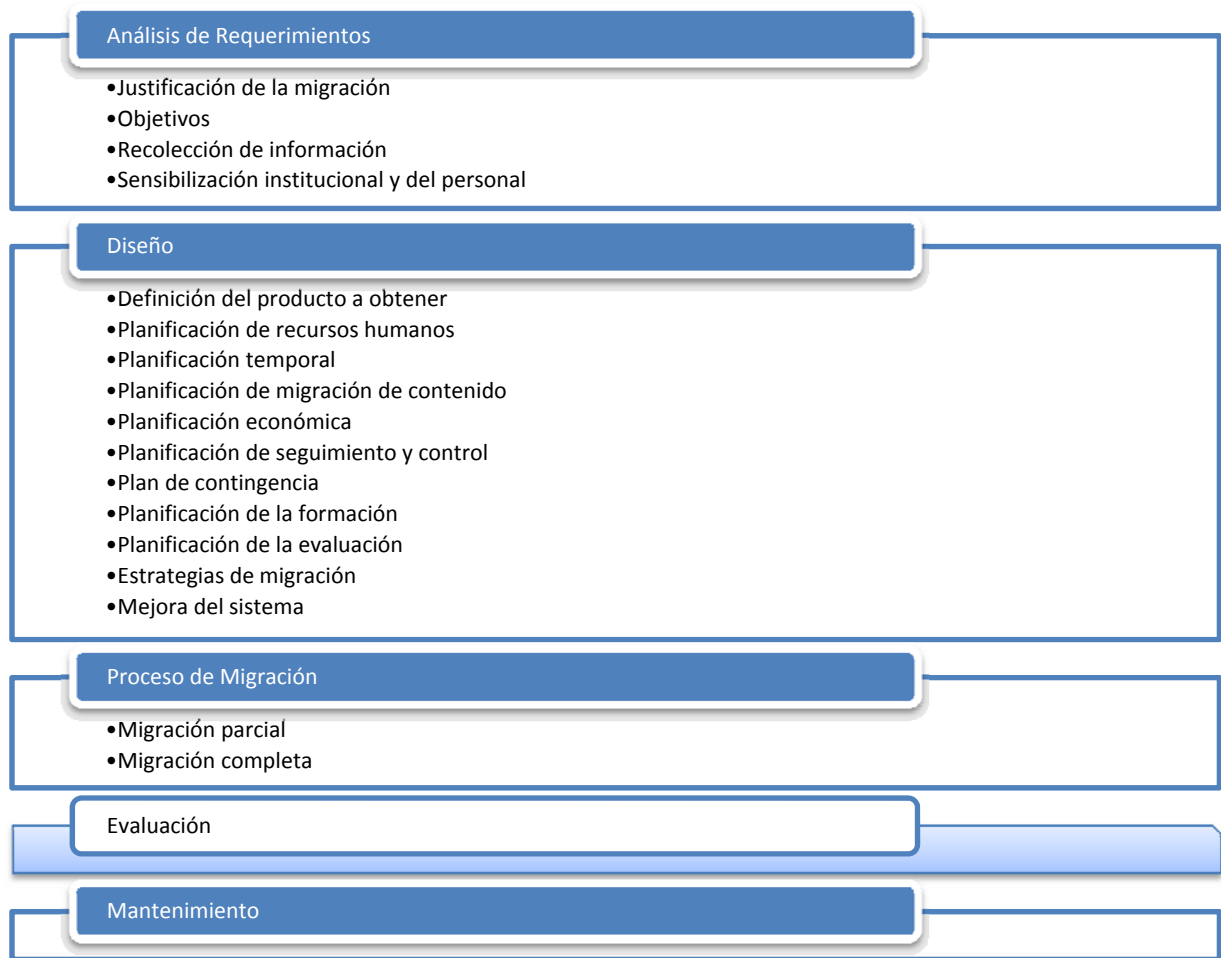


Figura 4.31. La fase de evaluación es la cuarta fase de la metodología

En la siguiente tabla se muestra cada prueba realizada sobre el sistema con sus diferentes puntos de control y el resultado de cada uno.

| Prueba | Resultado |
|-------------|-----------|
| Test 1 | |
| Checkpoint1 | Ok |
| Test 2 | |

| | |
|-------------|----|
| Checkpoint1 | Ok |
| Test 3 | |
| Checkpoint1 | Ok |
| Test 4 | |
| Checkpoint1 | Ok |
| Test 5 | |
| Checkpoint1 | Ok |
| Test 6 | |
| Checkpoint1 | Ok |
| Checkpoint2 | Ok |
| Checkpoint3 | Ok |
| Test 7 | |
| Checkpoint1 | Ok |
| Checkpoint2 | Ok |
| Test 8 | |
| Checkpoint1 | Ok |
| Checkpoint2 | Ok |
| Checkpoint3 | Ok |
| Test 9 | |
| Checkpoint1 | Ok |

| | |
|-------------|----|
| Checkpoint2 | Ok |
| Checkpoint3 | Ok |
| Checkpoint4 | Ok |
| Test 10 | |
| Checkpoint1 | Ok |
| Checkpoint2 | Ok |
| Checkpoint3 | Ok |
| Checkpoint4 | Ok |
| Test 11 | |
| Checkpoint1 | Ok |
| Checkpoint2 | Ok |
| Checkpoint3 | Ok |
| Checkpoint4 | Ok |
| Test 12 | |
| Checkpoint1 | Ok |
| Checkpoint2 | Ok |
| Test 13 | |
| Checkpoint1 | Ok |
| Test 14 | |
| Checkpoint1 | Ok |
| Test 15 | |
| Checkpoint1 | Ok |

| | |
|-------------|----|
| Checkpoint2 | Ok |
| Test 16 | |
| Checkpoint1 | Ok |
| Checkpoint1 | Ok |
| Test 17 | |
| Checkpoint1 | Ok |
| Checkpoint2 | Ok |
| Checkpoint3 | Ok |
| Checkpoint4 | Ok |

4.3.6. Mantenimiento

La fase de mantenimiento es la última fase de la metodología de migración de CMS (ver Figura 4.32). Hasta este momento ya se ha realizado la migración y se ha evaluado para comprobar su perfecto funcionamiento. Pero el proceso de migración no acaba aquí, hay que realizar la fase de mantenimiento para finalizarlo.

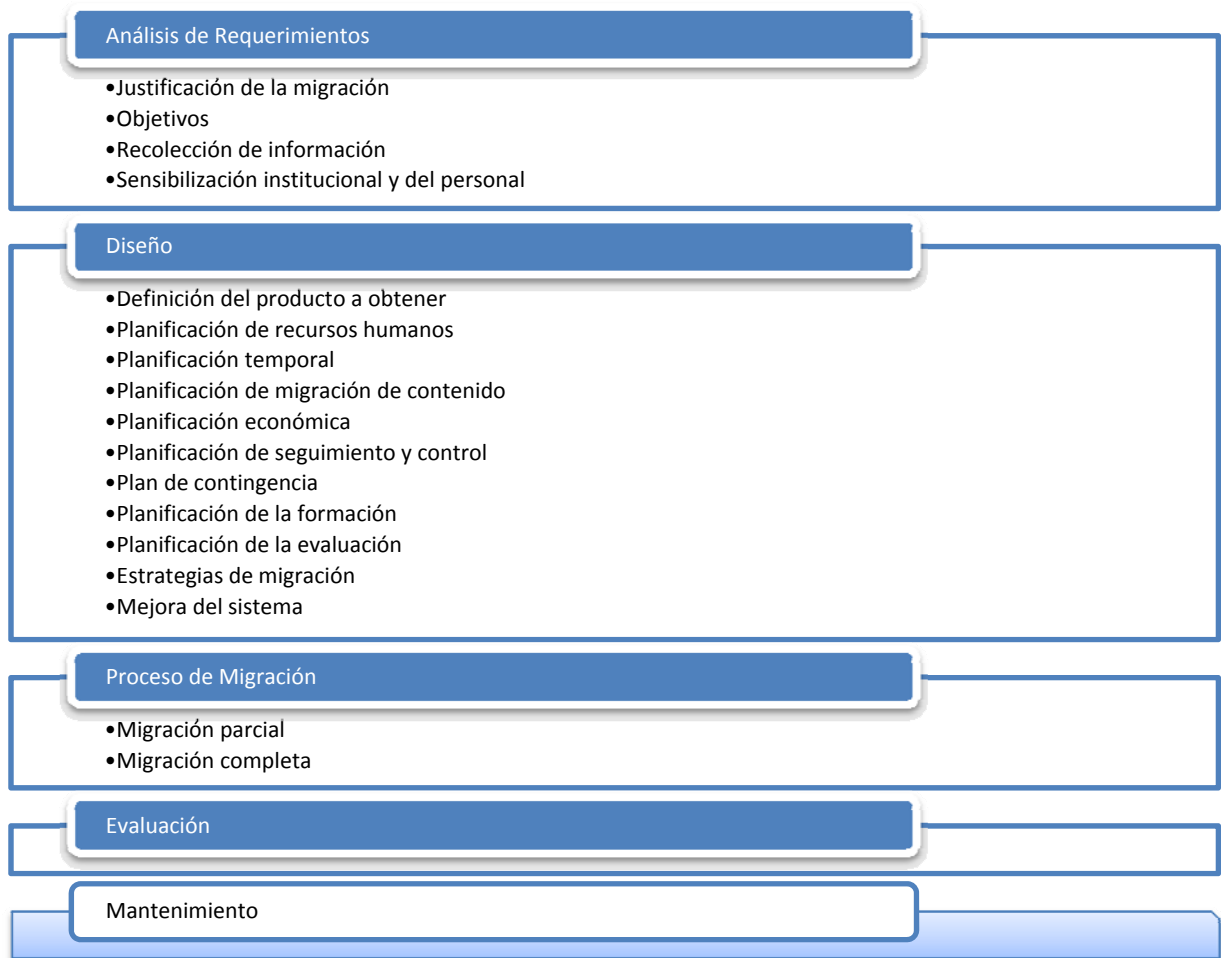


Figura 4.32. La última fase de la metodología es la fase de mantenimiento

Una de las tareas a realizar en la fase de mantenimiento es la formación de los usuarios en el nuevo sistema. Para ello, ejecutaremos el plan de formación definido previamente. Por muy completo y potente que sea el nuevo sitio, si los usuarios finales no reciben la formación necesaria sobre nuevos usos o nuevas funcionalidades del nuevo sitio, este podría producir rechazo con la consecuente pérdida de interés por el sitio, que llevaría un descenso en el número de usuarios que lo utilizan. Por tanto, es muy importante llevar a la práctica el plan de formación realizado.

En esta fase se tiene que definir quién o quiénes van a ser las personas responsables de la administración del sistema. Estas personas son las encargadas de actualizar el

contenido de las páginas existentes, publicar nuevas páginas o secciones, hacer backups del sitio, administrar los usuarios registrados, solventar posibles incidencias, mantener actualizada toda la información de la que se compone tanto el sistema de recomendación como el de georreferenciación, realizar las actualizaciones oportunas, etc.

4.4. Procesos Adicionales a la Metodología Propuesta para Problemas Específicos

Estos procesos adicionales son tareas o decisiones realizadas durante la migración debido a diferentes problemas o circunstancias no previstas en la fase de planificación, pero que se han tenido que llevar a cabo para el correcto funcionamiento de la plataforma y la aplicación. Estas tareas son las siguientes:

- *Incompatibilidad de la aplicación web con nuevas versiones de la plataforma:*

Uno de los problemas encontrados durante el proceso de migración y no detectados en la planificación de ésta, ha sido la incompatibilidad de la aplicación REJA con versiones superiores a la 1.5.26 de la plataforma CMS Joomla sobre la que funciona. Este problema es debido a que el CMS Joomla, a partir de su versión 1.5, utiliza el patrón de desarrollo Modelo-Vista-Controlador (MVC) para todo el sistema, incluidas las extensiones y plantillas.

La aplicación REJA no había sido diseñada bajo este patrón, ya que para la versión del CMS que fue desarrollada (la versión 1.0.12) no era obligatorio. En la versión de Joomla hasta la que se ha podido migrar la plataforma (versión 1.5.26), aún estando ya desarrollada bajo el patrón CMS, existe un plugin denominado "*Legacy mode*" en la instalación, que habilitándolo, permite la ejecución de aplicaciones aún no siguiendo este patrón. En versiones posteriores a esta, no se dispone de este plugin, obligando a que todas las

extensiones y plantillas que se ejecuten sobre la plataforma, sigan el patrón. Para poder seguir utilizando esta aplicación, sería necesario la programación completa desde el inicio de la aplicación REJA siguiendo el patrón CMS (ver Figura 4.33).

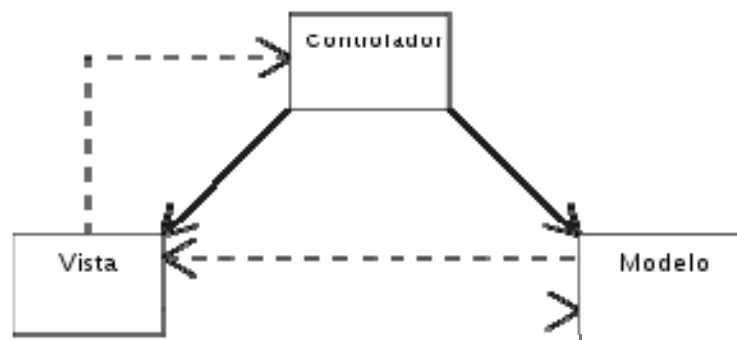


Figura 4.33. Diagrama que muestra la relación entre el modelo, la vista y el controlador.

- *Nueva versión para el sistema de georreferenciación*

Una de las funcionalidades de la aplicación REJA es la georreferenciación de los restaurantes. Para el desarrollo de esta funcionalidad se utilizó la API de Google Maps (<https://developers.google.com/maps>) en su versión 2. Esta versión ha quedado obsoleta al ser publicada la versión 3 de la misma. La versión 3 mejora muchas de las funcionalidades de la versión anterior, como pueden ser mayor compatibilidad con sistemas móviles, mejora en la presentación de los mapas, mejor rendimiento en determinados navegadores web, nuevas funcionalidades, etc.

En nuestro proceso de migración de la aplicación sobre el CMS Joomla, no se ha llevado a cabo la migración de esta funcionalidad, ya que no era un objetivo de este proyecto.

Al producirse un cambio en el servidor de alojamiento de la aplicación, se ha tenido que solicitar una clave nueva para su correcto funcionamiento. Esta clave, denominada Google Maps API key, ha sido solicitada en la página web de desarrollo de Google (<https://developers.google.com/maps>). Para ello es necesario indicar una cuenta de correo

electrónico y el dominio sobre el cual estará activa la aplicación. En caso de cambiar de dominio, esta clave tiene que ser solicitada de nuevo.

- *Código modificado en el núcleo de CMS*

Durante el proceso de migración se ha detectado una serie de problemas de funcionamiento de la aplicación. Estos problemas eran mostrados en la fase de registro de un nuevo usuario. Tras revisar el código en detalle, se ha detectado una modificación en el código del componente encargado de realizar dicho registro, el cual realiza una funcionalidad necesaria dentro del proceso de registro en la aplicación. Al formar este componente parte del núcleo del CMS, siempre que se realice una actualización o migración del núcleo, se corre el riesgo de perder dicha funcionalidad y por tanto, causar problemas en la ejecución de la aplicación.

Para subsanar el problema, primero se ha localizado el código añadido al componente `com_registration` realizando una copia de éste. A continuación, se ha migrado el componente en su totalidad a la versión nueva. Una vez disponible la nueva versión del componente, se ha vuelto a insertar el código copiado anteriormente en su correcta localización, dentro de este componente, de manera manual y se ha realizado un comentario en el código para futuras migraciones. Estos pasos son necesarios realizarlos para dejar el sistema con completa funcionalidad tras la migración.

Otros problemas referentes al código encontrados durante el proceso de migración y subsanados han sido errores en algunas variables por problemas de distinción entre mayúsculas y minúsculas y prefijos de los nombres de las tablas escritos manualmente.

Capítulo 5

Conclusiones

Este proyecto nació por la dificultad detectada a la hora de encontrar una metodología de migración que pudiera ser válida para la migración de sistemas de gestión de contenidos. Tras realizar un estudio, se encontró numerosa documentación referente a los pasos a seguir para realizar una determinada migración en productos concretos, pero no una metodología genérica para llevar a cabo la migración de estos sistemas. Como objetivo principal de este proyecto, se propone realizar una propuesta de metodología de migración para CMS y aplicarla a un caso concreto.

Para poder realizar la propuesta, se ha optado por realizar un estudio de la migración hacia software libre, junto con el proceso de migración clásico, adaptándolos para obtener una metodología de migración para CMS. De este modo, se ha propuesto una metodología definida en diferentes fases y tareas que van desde el análisis de requerimientos y diseño, hasta el proceso propio de migración, junto con la evaluación y mantenimiento del sistema.

Esta metodología ha sido puesta en práctica sobre la aplicación web REJA implementada sobre la plataforma CMS de Joomla. Al haber estado este sistema sin actualización durante largo periodo de tiempo, se ha necesitado realizar varias actualizaciones para obtener el resultado final.

Una vez realizado el proceso de migración, la aplicación web continua su ciclo de vida, por lo que ésta debe continuar siendo mantenida y actualizada sobre futuras mejoras o correcciones que vayan siendo publicadas. De lo contrario podría recaer en un sistema poco seguro y obsoleto.

Como conclusión, este proyecto me ha servido para valorar la gran importancia y dificultad que conlleva cualquier proceso de migración por simple que este sea. Hay que tener en cuenta muchos factores y realizar una planificación detallada antes y después de realizar un proceso de migración. Para realizar una migración e intentar que ésta tenga éxito, hay que realizar un proceso de análisis y planificación muy detallado y comprobar

meticulosamente antes de realizar cualquier cambio. Aun habiendo realizado todo esto, hay que estar preparado para cualquier circunstancia inesperada que pueda surgir, ya que hay muchos factores implicados en el proceso, como software, hardware, compatibilidades, datos, protocolos, etc.

Bibliografía

1. E.H. Bersoff, V.D.H., S.G. Siegel, Software Configuration Management. 1980: Prentice-Hall.
2. Pressman, R.S., Ingeniería del Software: Un Enfoque Práctico. Sexta ed. 2005: Mc Graw Hill. 958 páginas.
3. Ibáñez, A.A., Sistema de Recomendación de Restaurantes Georreferenciados. Escuela Politécnica Superior de Jaén. Universidad de Jaén, 2008.
4. Consejo-Superior-de-Administración-Electrónica, Directrices IDA de migración a software de fuentes abiertas. 2003.
5. IEEE, IEEE 1074-2006. 2006. Estándar para el Desarrollo de Procesos del Ciclo de Vida del Software
6. ISO, ISO 12207. Estándar para los procesos de Ciclo de Vida del Software. 2008.
7. AENOR, UNE 71044 Tecnología de la información - Procesos del ciclo de vida del software. 1999.
8. IEEE, IEEE 1219. 1998. Estándar de Mantenimiento Software
9. ISO, ISO/IEC 14764, in Estándar específico sobre Mantenimiento de Software. 1998.
10. Centro-de-Excelencia-de-Software-Libre-de-Castilla-La-Mancha, Taller de Migración a Software Libre 2009.
11. Daniel Sáez, M.P., Ricard Roca y David Anes, Migración al Software Libre. Guía de Buenas Prácticas, I.T.d. Informatica, Editor. 2007.
12. Instituto-Nacional-de-Estadística-e-Informática, Guía para la Migración de Software Libre en las Entidades Públicas. 2002.
13. Luis-D.-Aguilar-Lemarroy, Migración a Software Libre a Nivel Corporativo y Gubernamental. 2005.
14. Hobbs, D., Website Migration Handbook. Vol. 2. 2011.
15. Robertson, J. Content migration: options and strategies. 2008; <http://www.steptwo.com.au>.

16. Robertson, J., So, what is a content management system?, in KM Column. 2003.
17. Sistemas de Gestión de Contenidos.
http://es.wikipedia.org/wiki/Sistema_de_gesti%C3%B3n_de_contenidos.
18. CMS Joomla!; <http://www.joomla.org>
19. Joomla!-Official-Documentation. Migrating from 1.0.x to 1.5 Stable. 2010;
http://docs.joomla.org/Migrating_from_1.0.x_to_1.5_Stable.
20. com_migrator.
http://joomlancode.org/gf/download/frsrelease/4989/12702/migrator_beta2.zip.
21. Matware. <http://www.matware.com.ar/joomla-projects/mtwmigrator.html>.

Anexo I

Procedimientos de Copias de Seguridad en Joomla

En este anexo se explica cómo llevar a cabo una copia de seguridad del sistema gestor de contenidos Joomla. Para ello vamos a describir dos procedimientos distintos, uno realizado de forma manual y otro realizado de forma automática apoyándonos en una extensión del CMS.

a) Procedimiento Manual

Este primer procedimiento se denomina manual porque no interviene ningún componente ni extensión externa que sea necesaria instalar en el CMS para llevarlo a la práctica. Para ello, necesitamos hacer una copia de seguridad de la estructura completa de carpetas y archivos, y una copia de seguridad de la base de datos.

Utilizando un programa cliente de FTP, por ejemplo Filezilla, disponible en la web <http://filezilla-project.org/> (ver Figura I.1), se copian todas las carpetas y archivos de nuestro sitio web. Simplemente elegir el sitio en el cual queremos guardarlos y realizar el proceso. Una vez finalizada la descarga se debe comprobar que todo el proceso se ha realizado correctamente y sin errores.

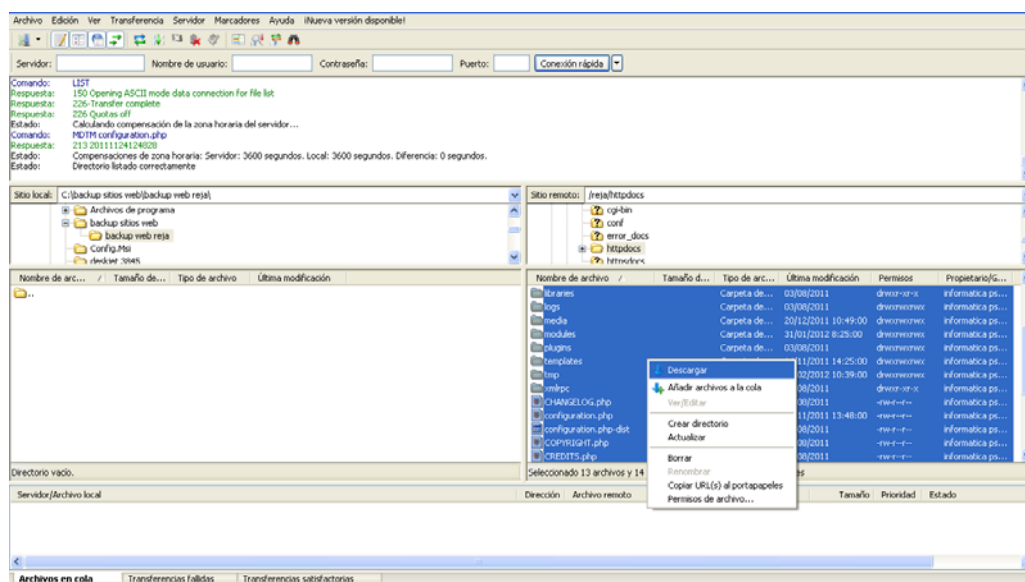


Figura I.1. Software Cliente FTP Filezilla

Para realizar la copia de seguridad de la base de datos, se puede utilizar la aplicación de gestión de base de datos phpMyAdmin (www.phpmyadmin.net). En este caso, para llevar a cabo su copia, se accede a esta aplicación y se selecciona la base de datos deseada (ver Figura I.2). A continuación, se muestra una página con todas las tablas de la base de datos con información de los parámetros que la definen. Se selecciona la pestaña “Exportar” para acceder a la página que define la exportación de la base de datos completa.

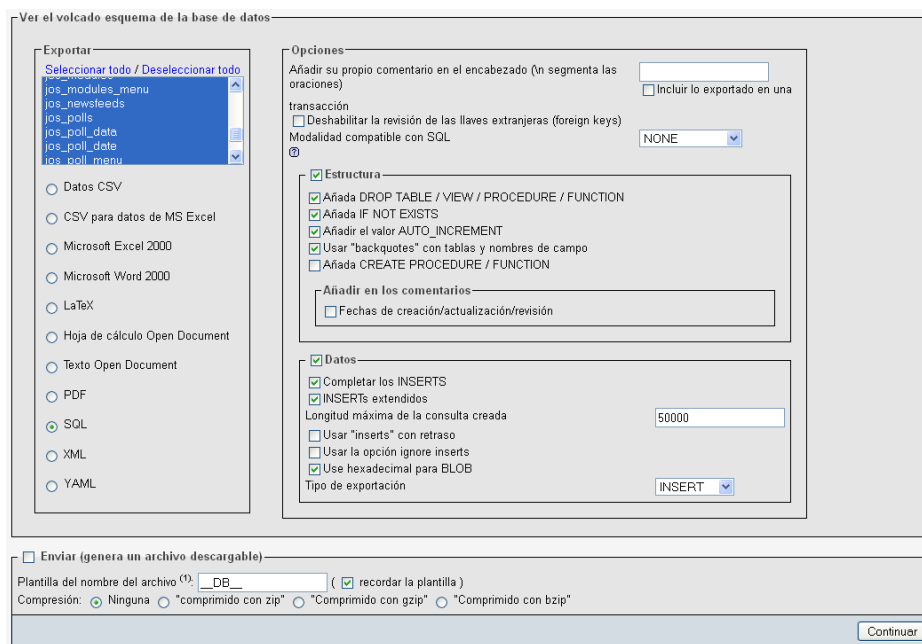


Figura I.2. Software phpMyAdmin

Para la exportación, mantenemos las opciones marcadas que se muestran por defecto, activando las opciones “Añada DROP TABLE” y “Añada IF NOT EXISTS” como se muestra en la Figura I.2, indicando que se genere un archivo descargable y eligiendo si lo deseamos con algún tipo de compresión. A continuación, se pulsa sobre el botón “Continuar” y se elige la carpeta en la cual queremos guardar el archivo. Este archivo será de extensión sql.

Con estos dos procesos, se ha realizado una copia completa del sitio web.

b) Procedimiento Automático

Para realizar el procedimiento de manera automática, hay que hacer uso de alguna extensión para tal fin de las muchas que podemos encontrar en el repositorio de extensiones del proyecto Joomla en la web <http://extensions.joomla.org>. En nuestro caso, a modo de ejemplo, vamos a utilizar la extensión Akeeba backup (<https://www.akeebabackup.com>). Esta extensión es de las más utilizadas actualmente y es la sucesora de la extensión JoomlaPack, que era utilizada para versiones de Joomla 1.0.x, y por lo tanto, ya no tiene mantenimiento ni actualizaciones. La extensión Akeeba Backup es un componente que permite hacer backups completos del sitio web, incluyendo todos los archivos y todo el contenido de la base de datos, de una forma directa y con diversas opciones de configuración.

Una vez descargada e instalada la extensión en nuestro CMS, accedemos a la parte de Backend del sitio. Desde Componentes → Akeeba Backup accedemos al centro de control del componente. Desde aquí se pueden configurar diversas opciones, crear perfiles para nuestras copias de seguridad, iniciar las tareas de respaldo, etc. (ver Figura I.3).



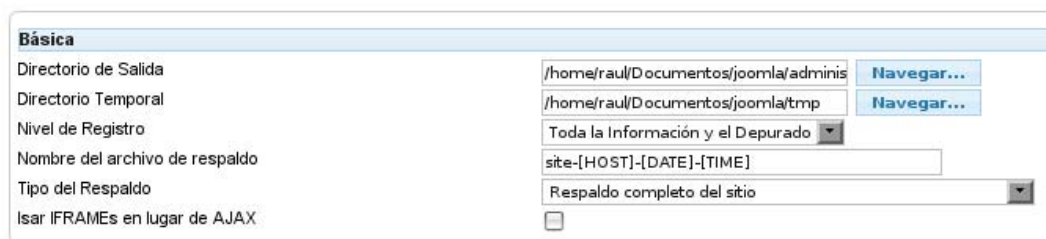
Figura I.3. Panel de control del componente Akkeba Backup

En la sección “Administración de perfiles”, se ofrece la posibilidad de crear distintos perfiles para las operaciones de backup. Para ello, se accede a la opción “Nuevo” y se escribe

el nombre del perfil que se desee crear. Esto permitirá mantener distintas configuraciones y emplear la que más se adecue a cada situación, con sólo seleccionar uno u otro perfil en la pantalla principal de la extensión Akeeba Backup.

Una vez seleccionado el perfil adecuado, se puede acceder a “Configuración” y modificar los parámetros que se deseen. Esta sección consta de varias partes que permitirán configurar un número de opciones bastante elevado:

- *Configuración básica:* Desde aquí se permite modificar, entre otras opciones, el nombre del archivo de respaldo, el directorio de salida de los archivos de backup o el tipo de copia de seguridad. Esta última opción, además, ofrece la posibilidad de realizar una copia completa del sitio, una copia incremental de los archivos, respaldar solo la base de datos, etc. (verFigura I.4).



The screenshot shows the 'Básica' (Basic) configuration section of the Akeeba Backup interface. It contains several settings:

- Directorio de Salida:** /home/raul/Documentos/joomla/adminis (with a 'Navegar...' button)
- Directorio Temporal:** /home/raul/Documentos/joomla/tmp (with a 'Navegar...' button)
- Nivel de Registro:** Toda la Información y el Depurado (dropdown menu)
- Nombre del archivo de respaldo:** site-[HOST]-[DATE]-[TIME] (text input)
- Tipo del Respaldo:** Respaldo completo del sitio (dropdown menu)
- Isar IFRAMEs en lugar de AJAX:**

Figura I.4. Opciones de configuración básicas

- *Configuración avanzada:* En esta sección se pueden ajustar parámetros relacionados con el motor de respaldo de la base de datos, el tipo motor de archivado (zip, jpa, directo a ftp), separar en partes el archivo de salida del backup, e incluso enviar estos archivos por correo electrónico (ver Figura I.5).

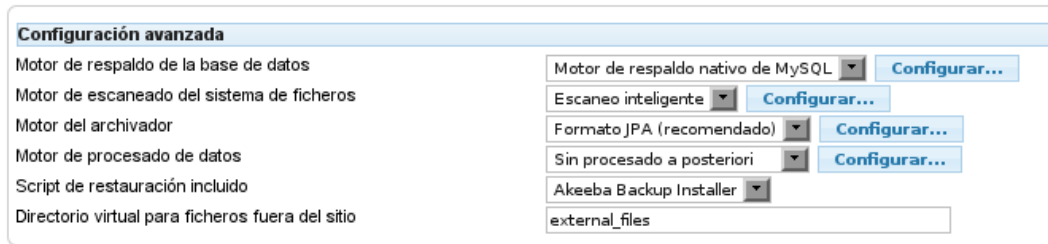


Figura I.5. Opciones de configuración avanzadas

- *Administración de cuota:* La cuota por tamaño permite definir el tamaño máximo del conjunto de archivos de backup de un determinado perfil. Cuando el tamaño del conjunto sobrepase el límite establecido, las copias antiguas serán eliminadas. La cuota por conteo establece el número de archivos de respaldo que puede contener un perfil. Cuando se sobrepasa dicho número, las copias antiguas son borradas (ver Figura I.6).

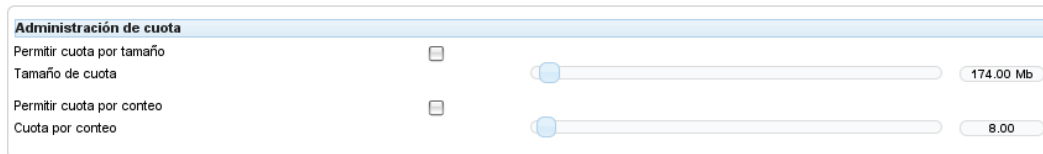


Figura I.6. Opciones de administración de la cuota

- *Ajuste fino:* Durante el proceso de respaldo, la extensión Akeeba Backup tiene que realizar múltiples conexiones con su URL de backup, por lo que puede haber problemas con el servidor, tanto por hacer demasiadas conexiones en muy poco tiempo, como por hacer conexiones demasiado largas. Para evitar esto, se puede ajustar el tiempo mínimo entre cada conexión y el tiempo máximo de cada una de las conexiones. En la última opción se permite establecer un margen para el tiempo máximo de conexión. Cuanto menor sea el porcentaje, más se ajustarán las conexiones al tiempo máximo establecido (ver Figura I.7).



Figura I.7. Opciones de ajuste fino

Además de todas estas opciones, Akeeba Backup ofrece la posibilidad de excluir de nuestra copia de respaldo directorios, ficheros y tablas de la base de datos. Para ello basta con acceder a las secciones “Exclusión de ficheros y directorios” y “Exclusión de tablas de la base de datos” y seleccionar los elementos de los que no interesa hacer copia de seguridad. Una vez que se tengan todas las opciones anteriores bien definidas, se puede comenzar con el backup del sitio seleccionando mediante la opción “Respaldar ahora” (ver Figura I.8).

Comenzar un nuevo respaldo

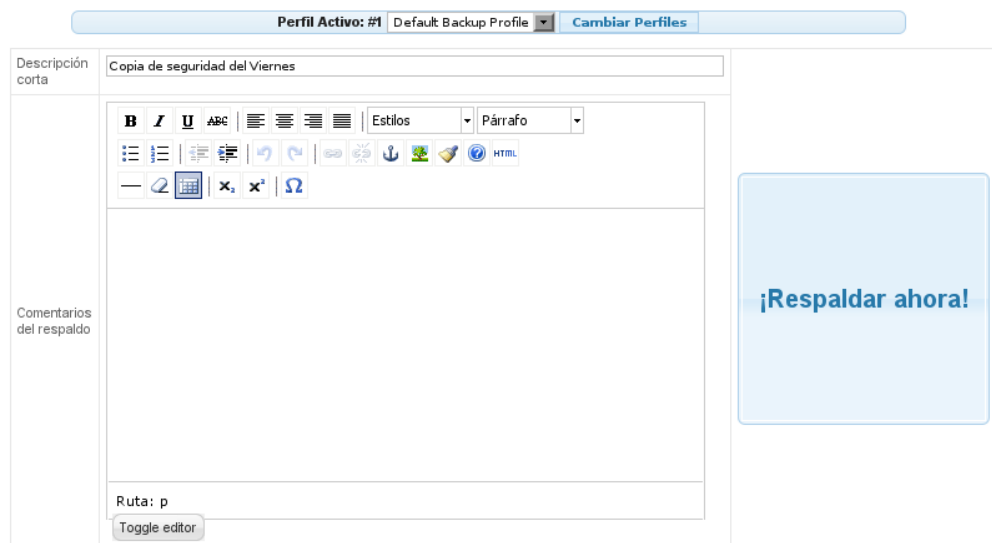


Figura I.8. Inicio del proceso de respaldo

Una vez finalizado el proceso, se mostrará una pantalla informando del éxito de la operación. Desde aquí, se puede ver el archivo de registro o acceder a la pantalla de administración de los archivos de respaldo (ver Figura I.9).



Figura I.9. Resultados obtenidos tras el respaldo

Finalmente, si se desea ver una lista de las copias de seguridad de las que se dispone, eliminar las que no sirvan o descargar aquellas que se necesiten, se podrá hacer desde “Administrar ficheros de respaldo”.

Este procedimiento, una vez que está correctamente configurado, es más rápido que el procedimiento manual explicado anteriormente, pero como contraposición, al utilizarse una extensión externa a nuestro CMS, perdemos transparencia en el proceso de la copia de seguridad, por lo que hay que confiar en el correcto funcionamiento de ésta.

Anexo II

Estrategias de Migración en Joomla

Muchos CMS no proporcionan un sistema para actualizar las versiones antiguas hacia otras más actuales. Este es el caso de Joomla 1.5. La conversión de un sitio en Joomla de una versión anterior a Joomla 1.5 requiere la creación de un nuevo sitio desde cero, usando una instalación limpia de Joomla en la versión 1.5 que deseemos, y luego trasladar el contenido del sitio antiguo al sitio nuevo creado. Esta migración de contenido no es un proceso directo, e implica conversiones y modificaciones del contenido de origen [19].

Existen varios métodos y extensiones para realizar la migración. A continuación pasamos a describir dos de estos métodos:

- Un método de migración automatizado, que utiliza el componente de migración `com_migrator` [20] para crear el volcado de contenido del sitio antiguo al nuevo, y una función de importación inteligente incluida en la instalación de Joomla 1.5, que ejecuta las conversiones y modificaciones necesarias durante el proceso de instalación.
- Otro método de migración automatizado, utilizando el componente `mtwmigrator` [21]. En este caso utilizamos la misma base de datos para el nuevo sitio, desde la cual se extraerá la información de las tablas de Joomla y se convertirá ésta para que sea válida para la nueva versión del CMS.

Sea cual sea el método a emplear, una vez llegado a este punto, se presupone que se han realizado las tareas previas necesarias, como son, la descarga de las diferentes extensiones compatibles con la versión nueva del CMS que están siendo utilizadas en el sitio actual, al igual que la plantilla, la descarga de la versión del CMS a instalar y la copia de seguridad completa del sitio, tanto archivos como base de datos. Una vez realizado todos estos pasos, podemos continuar con el proceso de migración.

A continuación realizamos una descripción detallada de los diferentes métodos:

a) Migración Automatizada. Método 1

Este es un proceso en dos fases y utiliza dos herramientas. La primera herramienta es un componente de migración llamado 'com_migrator' [20]. Este componente de migración debe instalarse y ejecutarse en el sitio antiguo para crear el volcado de exportación. La segunda herramienta forma parte del proceso de instalación de Joomla 1.5. Esta herramienta carga el volcado de contenido exportado en el nuevo sitio y ejecuta todas las conversiones y modificaciones necesarias:

El primer paso es la descarga e instalación del componente en el sitio antiguo. Posteriormente, se accede a él desde el menú componentes del administrador. Una vez dentro, pulsar sobre el icono "Dump it" para crear tres scripts de exportación comprimidos. El primero es una copia de respaldo del sitio antiguo, el segundo es el contenido de migración de todos los elementos del núcleo que serán importados al nuevo sitio y el tercero es una copia de respaldo de las tablas de componentes de terceros. A continuación pulsar sobre el icono de descarga de los archivos de exportación necesarios y se almacenan localmente.

Una vez exportados los archivos necesarios, se realiza una instalación limpia de Joomla en la versión nueva. Se realizan los pasos generales de la instalación hasta el paso número 6 "Configuración". En este paso se elige la opción de "Cargar un script de migración" en la sección 'Cargar Datos de Ejemplo, Restaurar o Migrar el contenido de una Copia de Respaldo'. En el campo de "Prefijo de las tablas existentes" se introduce el prefijo utilizado en las tablas del contenido de origen, normalmente "jos_". Seleccionar la "Codificación del sitio anterior" y en "Script de migración" elegir el archivo generado anteriormente por el componente com_migrator. A continuación, pulsar el botón "Subir y Ejecutar". Se muestra un mensaje informando del éxito de la operación o en su caso una lista de errores. Continuar la instalación normalmente, teniendo en cuenta que los datos del administrador válidos serán los mismos que en el sitio antiguo.

b) Migración Automatizada. Método 2

Este método de migración utiliza el componente mtwmigrator [21]. Este componente es utilizado para la migración de la versión 1.0.x de Joomla a la versión 1.5.x. Para posteriores versiones de Joomla se utiliza la nueva versión de este componente denominada jUpgrade.

En este caso realizamos una instalación de Joomla en la nueva versión directamente, no se instala el componente en el sitio antiguo como en el método anterior. En los parámetros de conexión (ver Figura II.1) introducimos los mismos datos que los del sitio antiguo, ya que vamos a utilizar el mismo servidor y la misma base de datos. En este paso es muy importante elegir bien el "Prefijo de las tablas" en la configuración avanzada. Habrá que introducir el prefijo que se desee que tengan las nuevas tablas, no las que ya existen. Se continúa con la instalación y en el paso 6 de "Configuración", no habrá que seleccionar la opción de "Instalar contenido de ejemplo predeterminado" ni tampoco la opción de "Cargar un script de migración" como se hacía en el método anterior. Una vez finalizada la instalación, obtendremos un sitio web en Joomla con la nueva versión en blanco.

Parámetros de la conexión:

Configurar Joomla para que funcione en su servidor requiere de cuatro pasos sencillos...

Configuración básica

1. Seleccionar el tipo de base de datos que utilizará en la lista desplegable (generalmente **mysql**)
2. Escribir el nombre del servidor en el que se instalará Joomla.
3. Escribir el nombre de usuario de MySQL, la contraseña y el nombre de la base de datos que utilizará para Joomla!. Éstos deben estar previamente configurados en su servidor
4. Configuración avanzada

Si la base de datos contiene tablas de instalaciones anteriores de Joomla!, deberá indicar cómo proceder. Eliminarlas o hacer una copia de seguridad.

Escribir un prefijo con el que comenzarán los nombres de todas las tablas de esta instalación de Joomla.

Configuración básica

Tipo de base de datos
 Normalmente será **MySQL**.

Nombre del servidor
 Normalmente **localhost** o un nombre de host provisto por su proveedor.
Puede ser algo como **root** o un nombre de usuario, para la base de datos, asignado por su proveedor.

Nombre de usuario
 Por razones de seguridad el uso de una contraseña para la cuenta de la base de datos MySQL es altamente recomendado.
Esta es la contraseña para acceder a su base de datos. Es posible que esta información sea predeterminada por su proveedor.

Contraseña
 Algunos hosts permiten solo una base de datos por cuenta. Si debe instalar más de un sitio Joomla! con una misma base de datos, puede modificar la opción de prefijo de tabla en la sección **Parámetros avanzados** para cada sitio instalado.

Nombre de la base de datos

Configuración avanzada

Eliminar las tablas existentes

Hacer una copia de las tablas existentes.

Prefijo de las tablas

Las tablas existentes en la base de datos, de instalaciones anteriores de Joomla!, serán reemplazadas.

No puede utilizarse "bak_" ya que se usa para la copia de seguridad de las tablas.

Figura II.1. Parámetros de configuración para la conexión

El siguiente paso es migrar el contenido del sitio anterior. Para ello accedemos al panel de administración e instalamos el componente mtwmigrator. A continuación accedemos a este componente y pulsamos sobre el botón configuración (ver Figura II.2). En este apartado se insertan los datos de configuración del servidor y de la base de datos y en el apartado prefijo indicamos el prefijo de las tablas del sitio web antiguo. Desde esta pantalla también se permite elegir qué contenido deseamos migrar y ofrece la opción de migrar contenido de algunas extensiones de terceros. Una vez finalizada la configuración, se hace

clik sobre el botón “Empezar migración” para realizar la misma y finalizar la migración de contenido.



Figura II.2. Panel de control del componente mtwmigrator

Anexo III

Inventario Hardware y Software

Documento para inventario de hardware en el proceso de migración:

| Información del inventario de hardware | | | |
|--|------------------------|----|--|
| Nombre: | | ID | |
| Empresa: | | | |
| Sector: | | | |
| Responsable del inventario: | | | |
| | Equipo | | |
| Nombre del equipo: | | | |
| Comentarios: | | | |
| | Sistema Operativo | | |
| Nombre: | | | |
| Versión: | | | |
| Service Pack: | | | |
| | Procesador | | |
| Tipo: | | | |
| Velocidad: | | | |
| Número de procesadores: | | | |
| | Memoria RAM | | |
| Descripción | | | |
| Capacidad: | | | |
| Velocidad (MHz): | | | |
| Numero de Ranuras: | | | |
| | Almacenamiento (Lista) | | |
| Fabricante: | | | |
| Modelo: | | | |
| Descripción (IDE,...): | | | |
| Tipo: | | | |
| Tamaño | | | |
| | BIOS | | |
| Numero serie: | | | |
| Fabricante: | | | |
| Modelo: | | | |
| Versión: | | | |
| Fecha: | | | |

| | Sonido | | |
|------------------------------------|-----------------|--|--|
| Fabricante: | | | |
| Nombre: | | | |
| Descripción: | | | |
| | Tarjeta vídeo | | |
| Nombre: | | | |
| Chipset: | | | |
| Memoria (MB): | | | |
| Resolución: | | | |
| | Red (Lista) | | |
| Descripción: | | | |
| Tipo (Ethernet, Fast Ethernet...): | | | |
| Velocidad: | | | |
| Dirección MAC: | | | |
| Dirección IP: | | | |
| Máscara: | | | |
| Puerta de enlace: | | | |
| Numero de red: | | | |
| | Puertos (Lista) | | |
| Tipo (USB, FireWire...): | | | |
| Nombre: | | | |
| Libre (Si, No): | | | |
| Descripción: | | | |

Documento para inventario de software en el proceso de migración:

| Información del inventario de software | | | |
|--|--|-----|--|
| Nombre: | | ID: | |
| Empresa: | | | |
| Sector: | | | |
| Responsable del inventario: | | | |
| Software | | | |
| Nombre del programa: | | | |
| Descripción: | | | |
| Comentarios: | | | |
| Plataforma | | | |
| Sistema Operativo: | | | |
| Maquina virtual o intérprete: | | | |
| Aspectos Legales | | | |
| Licencia SW: | | | |
| Contratos de mantenimiento: | | | |
| Requisitos Mínimos | | | |
| Microprocesador: | | | |
| Memoria: | | | |
| Espacio en disco: | | | |
| Acceso a base de datos | | | |
| Tipo de BBDD: | | | |
| Marca: | | | |
| Versión: | | | |
| Tamaño de la BBDD: | | | |
| Librerías, software o servicio relacionado (Lista) | | | |
| Librería o SW: | | | |
| Versión: | | | |
| Licencia: | | | |
| ID: | | | |
| Dispositivos hardware necesarios (Lista) | | | |
| Descripción | | | |
| Descripción de funcionalidades | | | |
| | | | |