

*Títol: Disseny i implementació web per a una escola d'arts
marcial*

Volum:

Alumne: Raül Borrachero Sánchez

Director/Ponent: Pau Fonseca i Casas

Departament: Estadística i Investigació Operativa

Data: 07/01/2009

GESTIÓ ACADÈMICA-FIB
ENTRADA
12 ENE. 2009

ÍNDICE

1. INTRODUCCIÓN	<i>pag. 8</i>
2. DIMENSIONADO	
2.1. Definición de objetivos	<i>pag. 9</i>
2.2. Tecnologías usadas	<i>pag. 10</i>
2.3. Costes y desarrollo	
2.3.1. <i>Desarrollo</i>	<i>pag. 14</i>
2.3.2. <i>Costes</i>	<i>pag. 18</i>
2.4. Drupal	
2.4.1 <i>¿Porqué Drupal?</i>	<i>pag. 22</i>
2.4.2 <i>Estructura y funcionamiento</i>	<i>pag. 23</i>
3. EL PORTAL WEB	
3.1. INSTALACIÓN	
3.1.1. <i>Archivos de Drupal</i>	<i>pag. 30</i>
3.1.2. <i>La base de datos</i>	<i>pag. 30</i>
3.1.3. <i>ffmpeg y JW Player</i>	<i>pag. 34</i>
3.2. SERVICIOS	
3.2.1. <i>General</i>	<i>pag. 36</i>
3.2.2. <i>Alumno</i>	<i>pag. 37</i>
3.2.3. <i>Entrenador</i>	<i>pag. 40</i>
<i>Vídeo y archivos adjuntos</i>	<i>pag. 41</i>
3.2.4. <i>Administrador</i>	<i>pag. 44</i>
<i>Construcción del sitio</i>	<i>pag. 46</i>
<i>Alias</i>	<i>pag. 46</i>
<i>Bloques</i>	<i>pag. 46</i>
<i>Disparadores</i>	<i>pag. 47</i>
<i>Formulario de contacto</i>	<i>pag. 47</i>
<i>Menús</i>	<i>pag. 48</i>
<i>Módulos</i>	<i>pag. 48</i>
<i>Temas</i>	<i>pag. 51</i>
<i>Traducción del interfaz</i>	<i>pag. 52</i>
<i>Vistas</i>	<i>pag. 53</i>
<i>Gestión de Contenido</i>	<i>pag. 56</i>
<i>Agregador de Feeds</i>	<i>pag. 56</i>
<i>Comentarios</i>	<i>pag. 57</i>
<i>Contenido</i>	<i>pag. 57</i>
<i>Galerías de imágenes</i>	<i>pag. 57</i>
<i>Importar imágenes</i>	<i>pag. 57</i>
<i>Opciones de publicación</i>	<i>pag. 57</i>
<i>Publicación RSS</i>	<i>pag. 58</i>
<i>Taxonomías</i>	<i>pag. 58</i>
<i>Tipos de contenido</i>	<i>pag. 60</i>
<i>Gestión de Usuarios</i>	<i>pag. 62</i>
<i>Configuración de usuarios</i>	<i>pag. 62</i>
<i>Perfiles</i>	<i>pag. 62</i>
<i>Permisos</i>	<i>pag. 63</i>
<i>Reglas de accesos</i>	<i>pag. 63</i>
<i>Roles</i>	<i>pag. 63</i>
<i>Usuarios</i>	<i>pag. 63</i>
<i>Configuración del Sitio</i>	<i>pag. 65</i>
<i>Acciones</i>	<i>pag. 65</i>
<i>Eventos</i>	<i>pag. 65</i>

<i>Fecha y hora</i>	<i>pag. 65</i>
<i>Formatos de entrada</i>	<i>pag. 65</i>
<i>Idiomas</i>	<i>pag. 67</i>
<i>Imágenes</i>	<i>pag. 68</i>
<i>Información del sitio</i>	<i>pag. 68</i>
<i>Informe de errores</i>	<i>pag. 68</i>
<i>Juegos de herramientas de imágenes</i>	<i>pag. 68</i>
<i>Mantenimiento del sitio</i>	<i>pag. 68</i>
<i>Preferencias de Flashvideo</i>	<i>pag. 68</i>
<i>Preferencias de búsqueda</i>	<i>pag. 71</i>
<i>Registros y alertas</i>	<i>pag. 71</i>
<i>Rendimiento</i>	<i>pag. 71</i>
<i>Sistema de ficheros</i>	<i>pag. 72</i>
<i>Subida de ficheros</i>	<i>pag. 72</i>
<i>Tema de administración</i>	<i>pag. 72</i>
<i>URLs limpias</i>	<i>pag. 72</i>
<i>Grupos</i>	<i>pag. 73</i>
<i>Informes</i>	<i>pag. 73</i>
<i>Ayuda</i>	<i>pag. 75</i>
<i>Actualizaciones de drupal 6</i>	<i>pag. 76</i>
3.2.5. Ficheros de Pulsómetro	<i>pag. 77</i>
4. AMPLIACIONES	<i>pag. 90</i>
5. CONCLUSIONES	<i>pag. 92</i>
6. BIBLIOGRAFIA	<i>pag. 94</i>
7. AGRADECIMIENTOS	<i>pag. 96</i>
ANEXO I: NetBeans 6.5 y Drupal 6	<i>pag. 97</i>
ANEXO II: Introducción al desarrollo de aplicaciones en Drupal 6	<i>pag. 100</i>
ANEXO III: Instalación de FFMPEG en una Debian Etch (archivos binarios)	<i>pag. 108</i>
Índice de figuras.	
Figura 2-1. Descripción de tareas correspondiente al informe previo.	Pag. 14
Figura 2-2. Descripción de tareas correspondiente la finalización del proyecto.	Pag. 14
Figura 2-3. Diagrama de gantt correspondiente al informe preliminar.	Pag. 15
Figura 2-2. Descripción de tareas correspondiente la finalización del proyecto.	Pag. 15
Figura 2-5. Dificultad de un proyecto en función de sus líneas de código.	Pag. 21
Figura 2-6. Coeficientes COCOMO	Pag. 21
Figura 2-7. Capas en el diseño de Drupal	Pag. 23
Figura 2-8. Una vista general del núcleo de Drupal (Parcial).	Pag. 24
Figura 2-9. Al activar módulos adicionales, se añaden más funcionalidades.1	Pag. 25
Figura 2-10. La estructura de carpetas en una instalación de Drupal por defecto.	Pag. 27
Figure 2-11. El directorio sites permite guardas todas las modificaciones hechas a Drupal.	Pag. 28
Figura 3-1. Settings.php	Pag. 30
Figura 3-2. Crear base de datos <i>drupal_bd</i>.	Pag. 31
Figura 3-3. seleccionar base de datos <i>drupal_bd</i>.	Pag. 31
Figura 3-4. Importar base de datos. Paso 1.	Pag. 32
Figura 3-4. Importar base de datos. Paso 2.	Pag. 32
Figura 3-6. Menú de Sistema de ficheros.	Pag. 33
Figura 3-7. Opciones de Flashvideo	Pag. 35
Figura 3-8. Path a los archivos de FFMPEG.	Pag. 35
Figura 3-9. Página inicial del portal.	Pag. 37
Figura 3-10. Pagina de inicio de un alumno.	Pag. 38
Figura 3-11. Muestra de eventos	Pag. 38

Figura 3-12. Página de búsquedas sencilla y avanzada.	Pag. 39
Figura 3-13. Página de listados de logs (logs→ listado logs).	Pag. 39
Figura 3-13(bis). Página de subida de ficheros de pulsómetro (logs→subir logs).	Pag. 39
Figura 3-14. Menú de grupos de trabajo.	Pag. 39
Figura 3-15. Dentro de un grupo.	Pag. 40
Figura 3-16. Perfil de usuario.	Pag. 40
Figura 3-17. Listado de páginas publicadas por grupo, no leídas y general.	Pag. 41
Figura 3-18. Tipos de contenido	Pag. 41
Figura 3-19. Archivos adjuntos.	Pag. 42
Figura 3-20. Opciones de archivos adjuntos.	Pag. 42
Figura 3-21. Título y Cuerpo	Pag. 42
Figura 3-22. Opciones de publicación en grupos de trabajo.	Pag. 43
Figura 3-23. Imágenes adjuntas.	Pag. 43
Figura 3-24. Opciones de comentarios.	Pag. 43
Figura 3-25. Opciones de ruta URL.	Pag. 43
Figura 3-26. Opciones de Evento.	Pag. 44
Figura 3-27. Opciones de la creación de un grupo de trabajo.	Pag. 44
Figura 3-28. Formato de entrada y opciones de grupo.	Pag. 45
Figura 2-29. Información de revisión	Pag. 45
Figura 3-30. Opciones de auditoria y de publicación.	Pag. 45
Figura 3-31. Opciones de tema y privacidad.	Pag. 45
Figura 3-32. Página de administración de bloques.	Pag. 45
Figura 3-33. Configuración de Módulos.	Pag. 49
Figura 3-34. Códulo CCK (Content Construction Kit).	Pag. 49
Figura 3-35. Módulos opcionales básicos de Drupal (1)	Pag. 49
Figura 3-36. Módulos opcionales básicos de Drupal (2)	Pag. 49
Figura 3-37. Módulos básicos de Drupal y módulos de tiempo y fecha.	Pag. 49
Figura 3-38. Módulos de video e imagen.	Pag. 49
Figura 3-39. Módulos de grupos.	Pag. 50
Figura 3-40. Módulos de Vistas (Views) y otros módulos.	Pag. 50
Figura 3-41. Menú de configuración de temas.	Pag. 52
Figura 3-42. Personalización de los colores de un tema.	Pag. 52
Figura 3-43. Traducción del interfaz.	Pag. 52
Figura 3-44. Opciones para listar cadenas traducidas.	Pag. 52
Figura 3-45. Lista de cadenas traducidas.	Pag. 53
Figura 3-46. Formulario para la traducción de cadenas.	Pag. 53
Figura 3-47. Página de gestión de vistas.	Pag. 54
Figura 3-48. Edición de una vista.	Pag. 54
Figura 3-49. Resultado de la vista.	Pag. 56
Figura 3-50. Listado de usuario con el rol de alumno (vista de administrador)	Pag. 56
Figura 3-51. Listado de usuario con el rol de alumno (vista del resto de roles)	Pag. 56
Figura 3-52. Taxonomías	Pag. 59
Figura 3-53. Terminos de un vocabulario.	Pag. 59
Figura 3-54. Edición de un vocabulario.	Pag. 59
Figura 3-55. Edición de un término.	Pag. 59
Figura 3-56. Tipos de contenido.	Pag. 59
Figura 3-57. Tipos de contenido	Pag. 60
Figura 3-57. Identificación y opciones sobre la publicación.	Pag. 60
Figura 3-58. Opciones del flujo de trabajo.	Pag. 61
Figura 3-59. Grupos.	Pag. 61
Figura 3-60. Opciones de comentarios.	Pag. 61
Figura 3-61. Eventos de calendario.	Pag. 61
Figura 3-62. Imágenes adjuntas de image.	Pag. 61
Figura 3-63. Administración de perfiles.	Pag. 63
Figura 3-64. Permisos por roles.	Pag. 64
Figura 3-65. Administración de usuarios.	Pag. 65
Figura 3-66. Edición de usuario.	Pag. 65
Figura 3-67. Añadir un nuevo usuario.	Pag. 65
Figura 3-68. Lista de idiomas.	Pag. 67
Figura 3-69. Añadir un nuevo idioma.	Pag. 67
Figura 3-70. Opciones de presentación de contenido por idioma.	Pag. 68
Figura 3-71. Opciones de Imágenes.	Pag. 68
Figura 3-72. Preferencias de Flashvideo.	Pag. 70
Figura 3-73. Opciones de configuración.	Pag. 70
Figura 3-74. update.php	Pag. 76
Figura Anexo I-1. Versioning→CVS→Cehckout.	Pag. 98
Figura Anexo I-2. Tools Plugin Downloaded.	Pag. 98
Figura Anexo I-3. Creación de un nuevo Proyecto.	Pag. 99
Figura Anexo I-4. Crear conexión con la base de datos.	Pag. 99
Figura Anexo I-5. Instalación de Drupal.	Pag. 99
Figura Anexo I-6. Creación de módulo o tema personalizado.	Pag. 99



1. INTRODUCCIÓN

Hoy en día, dentro del mundo del deporte profesional es necesaria una gran preparación física y mental para obtener buenos resultados en competiciones deportivas. Para ello, los profesionales de dicho sector, deportistas y preparadores físicos, deben elaborar juntos planes de entrenamiento para cada disciplina y individuo, o en su defecto, grupo de personas, con el cual poder desarrollar una estrategia que les permita afrontar los retos con los mejores resultados posibles y conseguir los objetivos propuestos desde un inicio.

La estrategia de entrenamiento por su parte, necesita un buen plan de entrenamiento que haga posible su buen término, y es por ello que esencialmente son necesarias un conjunto de medios, procedimientos y conocimientos tanto técnicos como personales para que el plan sea eficiente. Dentro de los conocimientos la utilización de herramientas estadísticas es primordial para la buena planificación, ya que permite realizar un estudio personalizado de cada deportista individualmente y/o en conjunto, para más tarde basarse en los resultados de dichos procedimientos y modelos estadísticos a la hora de tomar decisiones sobre cómo el plan de entrenamiento debe encauzarse y prever la evolución de dicho deportista durante toda su duración.

Las herramientas y modelos estadísticos que se pueden usar para confeccionar un plan de entrenamiento no están limitados a una sola disciplina deportiva, ni siquiera dentro del ámbito del deporte profesional. Un buen ejemplo de ello son la mayoría de estilos de artes marciales no considerados como deporte en sí, que poseen diversas modalidades de competición. Los artistas marciales que participan en dichas competiciones han de seguir unas estrategias y planes de entrenamiento para obtener buenos resultados y para ello es perfectamente posible aplicar los métodos de estudio y confección de planes de entrenamiento de cualquier otra modalidad considerada como deporte de competición, si bien hay que tener más factores en cuenta en la fase de diseño.

Actualmente, cada vez está más diversificada la utilización de este tipo de prácticas, con lo cual hace más difícil la recaptación de los datos necesarios llevar a cabo el diseño de los planes de entrenamiento y la toma de decisiones, dada la dispersión de los alumnos y profesores. Cada día se hace más necesaria una manera de simplificar los trámites y permitir más autonomía a cada entrenador y a cada alumno, y les permita concentrarse en otras tareas más importantes que la manera en cómo captar los datos necesarios.

DISSENY I IMPLEMENTACIÓ WEB PER A UNA ESCOLA D'ARTS MARCIALS pretende poner al alcance de todo el mundo una solución sencilla y potente a la hora de mantener la comunidad de entrenamiento de una escuela de artes marciales. Mediante una portal web que permite la centralización de los datos a la vez que proporciona a la comunidad una manera de coordinar los eventos, las prácticas del arte marcial en cuestión facilitando información y supervisión a cada alumno por parte de los instructores, así como asentar las bases para la posibilidad del uso de herramientas estadísticas en un futuro para el diseño de los planes de entrenamiento, es posible dar a conocer la escuela de artes marciales dentro de la comunidad internauta y gestionar los procesos internos de dicha escuela. Dos puntos esenciales de cara al buen funcionamiento de cualquier disciplina de competición.

2. DIMENSIONADO

2.1. Definición de objetivos

Como en toda práctica de disciplinas en las cuales se enseñe y se aprenda algo, hay una comunidad de la cual forman parte alumnos y profesores y en la cual es necesaria una gran interacción entre sus miembros para que las enseñanzas fructifiquen y el aprendizaje se pueda llevar a cabo de manera satisfactoria. Aparte de una herramienta para facilitar la enseñanza y el aprendizaje de las actividades que una asociación de artes marciales necesita, también es positivo tener un vehículo con el cual poder comunicarse con futuros alumnos y gente interesada en las actividades de la asociación, así como mantener contacto con otros grupos que practiquen estas disciplinas. Para ello, hemos creído conveniente la utilización de un vehículo al alcance de todos: Internet.

La red es un recurso tremendamente útil en el cual podemos disponer de una gran cantidad de herramientas para nuestras necesidades, y ahora más que nunca gracias a las nuevas tecnologías de Web 2.0. Gracias a estas tecnologías se hemos pensado en un portal donde los alumnos, profesores y gente interesada en lo que una asociación de artes marciales pueda aportarles, puedan acceder en cualquier momento y consultar los datos que les hagan falta, ponerse en contacto con otros miembros, acceder a material de la escuela, conocer los eventos de la asociación y muchas otras cosas.

Desafortunadamente, muchos de los procesos a realizar para implantar y/o mantener este servicio a través de internet requieren ciertos conocimientos y recursos que hoy en día no poseen la mayoría del público al cual va dirigido este portal y por ello hemos pensado en una herramienta sencilla de implantar, sencilla de mantener y con un coste muy reducido para el cliente, en este caso, una asociación de artes marciales.

La mayor parte de las asociaciones de artes marciales acostumbran a ser recelosos sobre los alumnos a los cuales aceptan como discípulos, a veces por un secretismo implantado hace mucho tiempo por antiguos maestros los cuales veían sus enseñanzas como un legado a modo de herencia para unos pocos, y la mayor parte de ellas, por motivos de seguridad hacia los alumnos, ya que sin una preparación física y mental adecuada, es posible hacerse daño o hacer daño a terceras personas. También es necesario apuntar que la principal fuente de ingresos económicos de muchos maestros proviene de la enseñanza de estas prácticas. Por estos motivos, hemos creído esencial que el portal ofrezca una parte privada donde maestros y alumnos puedan publicar material y consultarlo siempre que quieran, y sólo ellos puedan hacerlo.

Aparte de los servicios que una página web normal pueda proporcionar, también hemos creído oportuna la integración de un servicio de hosting de vídeo propio en el portal teniendo en cuenta los motivos comentados en el anterior punto y el hecho que todas las artes marciales poseen muchas formas de ejercicio repetitivas que constan de una serie de movimientos encadenados, las cuales son mucho más comprensibles si se muestra su ejecución entera que paso a paso. Con ello, los maestros podrán subir vía navegador los vídeos que crean oportunos en una amplia variedad de formatos, e incrustarlos en la publicación que deseen añadir a la página.

Dado que muchas de las asociaciones de artes marciales compiten en torneos y eventos similares, es necesario un buen entrenamiento y por ello es importante que se añada a su estrategia de entrenamiento el diseño de un plan. Para diseñar un plan de entrenamiento eficiente en competición, como ya hemos comentado anteriormente, es imprescindible el uso de herramientas estadísticas y por ello el portal dispone de la integración de unos servicios para facilitar la recopilación de datos y la toma de decisiones. A través de un pulsómetro se pueden tomar los datos de una sesión de entrenamiento y mediante el portal, subir a un servidor central los datos recabados de esta manera a través de un archivo que sería procesado posteriormente. Este servicio de subida de ficheros es personal para cada miembro, así es más fácil llevar a cabo

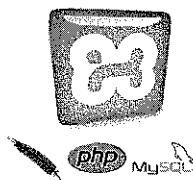
estudios personalizados para cada individuo. También sienta las bases para el desarrollo de una integración de dicho portal con paquete de software estadístico para llevar a cabo dichos estudios estadísticos y obtener la información que deriva de ellos, crucial en la toma de decisiones. Dicha integración se ha llevado a cabo con éxito en otro proyecto paralelo pero dado que dicho portal es una herramienta compleja para elaborar de este tipo de aplicaciones, se ha preferido elaborar una guía de programación y configuración de las herramientas necesarias para el desarrollo dentro del portal en cuestión, ya que no ha sido posible englobar la creación de dicha integración en el presente proyecto.

DISSENY I IMPLEMENTACIÓ WEB PER A UNA ESCOLA D'ARTS MARCIALS es un proyecto dentro de un marco de investigación mucho más amplio en estudios estadísticos aplicados a competición, y tiene como fin la facilitación de la recaptación de datos de deportistas tanto marciales como tradicionales. Incorpora otras herramientas desarrolladas por otros proyectistas las cuales son utilizadas en conjunto, como la integración mencionada en el punto anterior.

2.2. Tecnologías usadas

Para implementar todas las funcionalidades descritas en los objetivos del proyecto, han sido necesarias un conjunto de tecnologías específicas que funcionen conjuntamente. Ya que uno de los puntos clave dentro del diseño de objetivos es que fuera lo más barato y lo menos restrictivo en cuanto a licencias de software se refiere a la hora de hacer ampliaciones en un futuro, se han utilizado una serie de aplicaciones y sistemas dentro del movimiento Open Source y licencias de Creative Commons que hagan posible la libre distribución y posteriores modificaciones sin los problemas legales ni económicos que un software propietario conlleva.

He aquí una breve explicación de cada uno de las tecnologías usadas:



XAMPP: Es un servidor independiente de plataforma, software libre, que consiste principalmente en la base de datos MySQL, el servidor Web Apache y los intérpretes para lenguajes de script: PHP y Perl. El nombre proviene del acrónimo de X (para cualquiera de los diferentes sistemas operativos), Apache, MySQL, PHP, Perl. El programa está liberado bajo la licencia GNU y actúa como un servidor Web libre, fácil de usar y capaz de interpretar páginas dinámicas. Actualmente XAMPP esta disponible para Microsoft Windows, GNU/Linux, Solaris, y MacOS X.

Solamente requiere descargar y ejecutar un archivo zip, tar, o exe, con unas pequeñas configuraciones en alguno de sus componentes que el servidor Web necesitará. XAMPP se actualiza regularmente para incorporar las últimas versiones de Apache/MySQL/PHP y Perl. También incluye otros módulos como OpenSSL y phpMyAdmin. Para instalar XAMPP se requiere solamente una pequeña fracción del tiempo necesario para descargar y configurar los programas por separado.

En este proyecto se ha decantado por XAMPP como herramienta principal para el servicio de servidor web y bases de datos dado que es uno de los motores del proyecto de investigación en el cual está englobado el nuestro, además de ser una herramienta fácil de usar y sencilla de

instalar. Aunque el servidor XAMPP sobre el cual se hayan hecho la mayor parte de pruebas corra sobre un SO Microsoft Windows, también se ha testeado en un servidor LAMP (Linux, Apache, MySQL, Php) bajo una Deviant Etch 3.02.1, usando las mismas aplicaciones auxiliares que se han usado en Windows con los mismo resultados positivos.

Detallemos un poco más en que consisten cada uno de los servicios de XAMPP:

Apache: Es un servidor HTTP de código abierto para varias plataformas (entre ellas Unix, Windows y Macintosh) que implementa el protocolo HTTP/1.1 y la noción de sitio virtual. Entre otras características cabe destacar sus altamente configurables mensajes de error, bases de datos de autenticación y negocio de contenido. Tiene una amplia aceptación en la red: desde 1996, Apache, es el servidor http más usado. Las ventajas de apache respecto a otros sistemas que ofrezcan el mismo tipo de servicios es que es altamente modular y extensible (lo cual implica que es altamente configurable y adaptable a nuestras necesidades), Open Source, multiplataforma y popular, con lo cual podemos encontrar grandes cantidades de tutoriales de configuración y uso y abundante ayuda a los problemas que puedan se presentar.

MySQL: Se trata de un sistema de gestión de bases de datos relacional, multithread y multiusuario basado en el lenguaje de bases de datos SQL. MySQL posee dos tipos de licencias: por un lado se ofrece bajo la GNU GPL para cualquier uso compatible con dicha licencia, y una licencia para la inclusión en productos comerciales privados. MySQL es propiedad y está patrocinado por MySQL AB, subsidiaria de Sun Microsystems, lo cual posibilita es esquema dual anteriormente mencionado. Es muy utilizado en aplicaciones web y está fuertemente ligado a PHP dada su popular aplicación web en conjunto.

MySQL es una base de datos muy rápida en la lectura para entornos de baja concurrencia en la modificación de los datos, lo cual en aplicaciones web, ya que se da el caso mencionado y el entorno es intensivo en la lectura de datos, hace de MySQL una solución ideal para este tipo de aplicaciones.

En nuestro proyecto se usan varias bases de datos para gestionar tanto la información de la escuela como para administrar la información del sistema que se encarga de gestionar la página web.

PHPMyAdmin: Se trata de una herramienta escrita en PHP con la intención de manejar la administración de un servidor y/o bases de datos en MySQL a través de páginas web utilizando Internet. Actualmente, puede crear y eliminar bases de datos; crear, eliminar y alterar tablas; borrar, editar y añadir campos en tablas, ejecutar cualquier sentencia SQL, administrar claves en campos, administrar privilegios, exportar datos en varios formatos. Está disponible en 50 idiomas bajo una licencia GPL.

Es un proyecto tremendamente popular en la red internauta y por ello es fácil encontrar ayuda y documentación y ya que es una aplicación web, es posible ejecutarla a través de una serie de medidas de seguridad, en cualquier lugar del que se disponga de conexión, convirtiéndola en una herramienta extremadamente útil. Su sencillez en el manejo y su eficiencia lo hacen ideal para usuarios de todo tipo y es esencial en este proyecto a la hora de instalar y gestionar las bases de datos necesarias para su correcto funcionamiento.

PHP: (siglas de PHP Hypertext Pre-processo) Es un lenguaje de programación interpretado diseñado originalmente para la creación de páginas web dinámicas. Está pensado principalmente para ser evaluado en la parte del servidor, transmitiendo la información resultante al cliente a través de código html mayormente. También se puede usar como interfaz de línea de comando o en la creación de otros programas.

PHP, Publicado bajo la PHP License y siendo considerada esta licencia como software libre, es un lenguaje interpretado de propósito general ampliamente usado, que está diseñado especialmente

para desarrollo web y puede ser embebido dentro de código HTML. Generalmente se ejecuta en un servidor web, tomando el código en PHP como su entrada y creando páginas web como salida. Puede ser desplegado en la mayoría de los servidores web y en casi todos los sistemas operativos y plataformas sin costo alguno. PHP se encuentra instalado en más de 20 millones de sitios web y en un millón de servidores, aunque el número de sitios en PHP ha compartido algo de su preponderante sitio con otros nuevos lenguajes no tan poderosos desde agosto de 2005.

Dentro del marco del proyecto es quizás la herramienta más usada y extendida, ya que la mayor parte de las aplicaciones usadas utilizan PHP para su desarrollo. El portal web está programado a través de php y varias aplicaciones dentro del mismo desarrolladas íntegramente en este proyecto, también están escritas en PHP.

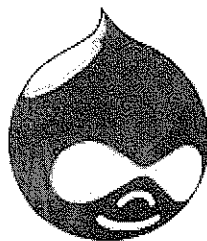
JW Player: Es un reproductor flash de video creado por Jeroen Wijering bajo una licencia Creative Commons, la cual permite modificar y distribuir libremente el software siempre y cuando no sea con fines lucrativos y se mencione la autoría de dicho software. Permite la reproducción de varios formatos de video incluidos cualquier formato que el reproductor flash de Adobe pueda. Es uno de los reproductores multimedia más populares en Internet, motivo por el que es más fácil encontrar soporte técnico. Está diseñado para ser usado principalmente en servicios web, lo cual hace de él una herramienta necesaria para nuestro portal web. Para ello, utilizamos una pequeña parte del reproductor para reproducir los videos de la escuela como lo haría cualquier comunidad o portal de hosting de video (como YouTube), haciendo de esta manera que sea más complicada la obtención del material de la escuela y su distribución.



FFMPEG: Es una colección de software libre que puede grabar, convertir y hacer streaming de audio y vídeo. Incluye una biblioteca de códecs. FFmpeg está desarrollado en Linux, pero puede ser compilado en la mayoría de los sistemas operativos, incluyendo Windows. Es destacable que la mayoría de los desarrolladores de FFmpeg lo sean también del proyecto MPlayer, más un miembro del proyecto Xine y que FFmpeg esté hospedado en el servidor del proyecto MPlayer.

FFMPEG posee una licencia que ha generado cierta controversia internacionalmente ya que en algunos países es ilegal su uso. En dichos países su política de actuación vulnera algunas patentes de software que utilizan los mismos procedimientos, dado que considera la ingeniería inversa como un método fundamental de operación en su comunidad de desarrollo, por ello muchas distribuciones de sistemas operativos de software libre no la incluyen en sus distribuciones iniciales. Se considera lícito su uso siempre y cuando se utilice en proyectos que empleen una licencia GPL.

En nuestro caso es usado porque partes del código del portal web en cuestión, utilizan dicho software, y dado que es un producto muy extendido, es fácil encontrar soporte técnico y ayuda a través de la red, además de no tener coste monetario alguno.



Drupal: Es un CMS (Content Management System), sistema de gestión de contenido para sitios Web. Permite publicar artículos, imágenes, u otros archivos y servicios añadidos como foros, encuestas, votaciones, blogs y administración de usuarios y permisos. Drupal es un sistema dinámico: en lugar de almacenar sus contenidos en archivos estáticos en el sistema de ficheros del servidor de forma fija, el contenido textual de las páginas y otras configuraciones son almacenados en una base de datos y se editan utilizando un entorno Web incluido en el producto.

Drupal fue originalmente escrito por Dries Buytaert y a pesar de que empezó como un pequeño BBS, ha llegado a ser mucho más que sólo un portal de noticias gracias a su arquitectura flexible. Drupal se compone de una infraestructura base y un conjunto de módulos que ofrecen un amplio conjunto de funciones, incluyendo sistemas de comercio electrónico, galerías de fotos, administración de listas de correo electrónico, e integración de CVS. Es posible añadir módulos de terceros para modificar el comportamiento de Drupal u ofrecer nuevas funciones. Drupal se usa, entre otros, en intranets de compañías, enseñanza en línea, comunidades de arte y administración de proyectos. Muchos piensan que la relevancia de Drupal en las comunidades de usuarios es lo que lo hace destacarse de la competencia.

Drupal es un proyecto de software libre bajo una licencia GPL esencial para el desarrollo del portal web que nos ocupa. Gracias a su enorme y nutrida comunidad de desarrollo, es posible encontrar cantidades ingentes de contribuciones, tutoriales, guías de programación y soporte técnico en línea.

2.3. Costes y desarrollo

2.3.1. Desarrollo

El desarrollo de este proyecto consta de varias partes, las cuales se han ido llevando a cabo según un calendario marcado por los objetivos. Cabe destacar que aunque se haya variado el número de horas a realizar, las tareas desempeñadas así como su orden, ha habido poca variación desde la descripción de las tareas en cuanto a horas se refiere.

He aquí la descripción de las tareas así como su programación según un informe de desarrollo de proyecto presentado el 18/10/2008 y lo mismo una vez finalizado. Es necesario apuntar que los tiempos están expresados en días y se ha calculado unas 4 horas por día de trabajo. El total expresa el número de días requeridos desde el inicio del proyecto hasta su final. Nótese que varias tareas forman parte de otras y algunas se han llevado a cabo simultáneamente.

NUM.	TAREA	PRECEDE	INICIO	FIN	PREVISTO	REAL	FINALIZADO
1	Especificación del proyecto	-	02/06/2008	17/06/2008	15	15	SI
2	Diseño de la página web	Especificación del proyecto	23/06/2008	08/07/2008	15	15	SI
3	Documentación y aprendizaje	Especificación del proyecto	23/06/2008	12/12/2008	100	80	NO
4	Implementación página web	Especificación del proyecto	12/07/2008	19/12/2008	160	100	NO
5	Implementación via Drupal	Diseño de la página web	14/07/2008	12/09/2008	60	60	SI
6	Testing Drupal	Implementación via Drupal	15/09/2008	24/10/2008	39	35	NO
7	Diseño comunicación entre Drupal y R-php	Implementación via Drupal	19/10/2008	24/10/2008	5	1	NO
8	Implementació comunicació entre Drupal i R-php	Diseño comunicación entre Drupal y R-php	25/10/2008	14/11/2008	30	0	NO
9	Migración a de Drupal 5 a Drupal 6	Diseño comunicación entre Drupal y R-php	13/11/2008	28/11/2008	15	0	NO
10	Mejoras de seguridad en el portal web	Testing Drupal	29/10/2008	28/11/2008	30	0	NO
11	Testing del proyecto	Implementación via Drupal	28/11/2008	18/12/2008	20	0	NO
12	Elaboración memoria del proyecto	Testing del proyecto	20/11/2008	25/12/2008	25	0	NO
TOTAL					206	139	

Figura 2-1. Descripción de tareas correspondiente al informe previo.

NUM.	TAREA	PRECEDE	INICIO	FIN	T.PREVISTO	T.REAL
1	Especificación del proyecto	-	02/06/2008	17/06/2008	15	15
2	Diseño de la página web	Especificación del proyecto	23/06/2008	08/07/2008	15	15
3	Documentación y aprendizaje	Especificación del proyecto	23/06/2008	12/12/2008	100	100
4	Implementación página web	Especificación del proyecto	12/07/2008	02/1/2009	160	174
5	Implementación via Drupal	Diseño de la página web	14/07/2008	12/09/2008	60	60
6	Testing Drupal	Implementación via Drupal	15/09/2008	24/10/2008	39	39
7	Migración a de Drupal 5 a Drupal 6	Implementación via Drupal	25/10/2008	06/11/2008	15	12
8	Mejoras de seguridad en el portal web	Testing Drupal	06/11/2008	13/11/2008	30	7
9	Solución a problemas en la base de datos	-	13/11/2008	20/12/2008	0	37
10	Elaboración memoria del proyecto	Mejoras Seguridad, Testing proyecto	08/12/2008	02/1/2009	5	25
11	Testing del proyecto	Implementació via Drupal	28/11/2008	05/1/2009	20	38
TOTAL					217	

Figura 2-2. Descripción de tareas correspondiente la finalización del proyecto.

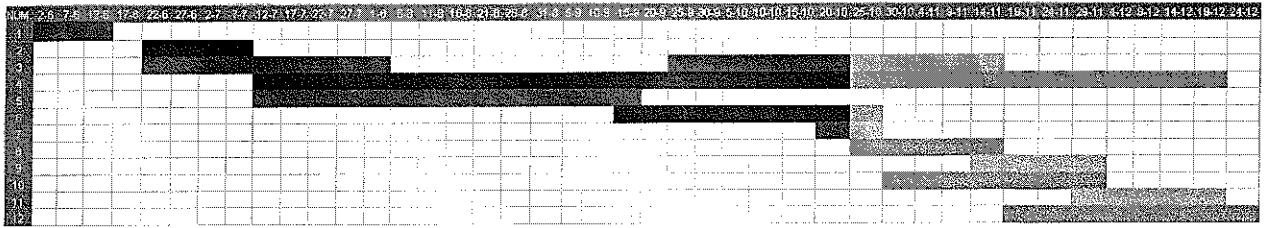


Figura 2-3. Diagrama de gantt correspondiente al informe preliminar.

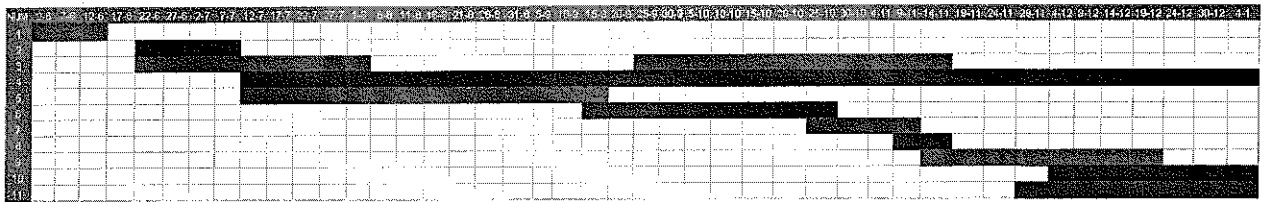


Figura 2-4. Diagrama de gantt correspondiente a la finalización del proyecto.

Como se puede observar, la diferencia entre ambos diagramas en poca y la mayor parte de los cambios se tratan de reestructuraciones debidas a un único problema. Seguidamente se detalla en qué han consistido cada una de las tareas realizadas en la conclusión del proyecto.

Especificación del proyecto: En esta parte se han especificado los requerimientos del proyecto, qué tecnologías se debían usar mencionadas anteriormente en el apartado de tecnologías usadas y las funcionalidades a desarrollar según las exigencias del mismo. Se ha hecho un análisis exhaustivo del tipo de servicios que un portal web de estas características necesita y/o puede llegar a necesitar. Se pensó en integrar un motor 3D en lugar del servicio de video flash en primera instancia, pero después de un análisis en más profundidad de los requisitos y las tecnologías a desarrollar en este apartado, se descartó dado su elevada complejidad. También se decidió entre varios sistemas y lenguajes de programación para llevar a cabo el portal web, entre ellos java, perl y php, descartando los dos primeros a favor de un CMS y se incluyó el tercero tras sopesar los diferentes CMS disponibles que satisficieran los requisitos de la idea original y decantarse por Drupal, del cual se describirá con mucha más amplitud en posteriores capítulos de esta memoria, al estar enteramente escrito en este lenguaje.

En referencia al software estadístico, se tuvo muy claro desde un buen principio que se usaría R dada su robustez y fiabilidad en el caso de integrar servicios de estudios estadísticos. Entonces surgió el problema de integrar R en el portal web en cuestión. En primera opción se sugirió usar R-web, un software de código abierto desarrollado en la por Jeff Banfield del departamento de ciencias matemáticas de la universidad de Washington en Arkansas, Estados Unidos; pero dado que estaba escrito en perl y al encontrar R-php, se decantó por este último al no haber que interpretar ningún tipo de código entre el portal y dicho servidor.

Diseño de la página web: Aquí se ha profundizado en la estructura del portal web en sí. Está estructurada en dos partes diferenciadas: la parte pública, la cual tiene la función de dar a conocer la asociación a través de secciones de información general y específica añadida por cada una de las escuelas que usen el producto, y una parte privada en la cual se desarrollen todas las actividades internas que sean necesarias y solamente accesible a través de un sistema de cuentas de usuario y password que garantice la privacidad de alumnos y profesores. También se ha contemplado el hecho de que los alumnos se puedan organizar en grupos de trabajo o clases dentro de la escuela y ello se ha traducido al portal, siendo posible que los solos los alumnos

pertenecientes a determinados grupos puedan acceder a la información que en ellos se ofrezca (los motivos pueden ser los diferentes niveles de aprendizaje dentro de una escuela de artes marciales, por ejemplo). En el capítulo destinado al portal web se describen todas sus funcionalidades de manera mucho más detallada.

Documentación y aprendizaje: Dentro del funcionamiento de DISSENY I IMPLEMENTACIÓ WEB PER A UNA ESCOLA D'ARTS MARCIALS se engloban una gran cantidad de tecnologías de las cuales se ha consultado abundante material informativo para su correcto uso. Entre ellos podemos englobar la API de Drupal, los manuales de instalación del propio CMS, su estructura y funcionamiento internos, abundantes tutoriales y guías de programación, documentación sobre ffmpeg y los problemas causados en su implantación en las diferentes distribuciones de varios sistemas operativos como son pueden ser Linux o Windows, manuales de programación de php, solución de errores y guías de SQL y un largo etcétera.

Personalmente he encontrado esta parte una de las más árduas del proyecto, dado que partía con prácticamente nulos conocimientos del funcionamiento de muchas de las tecnologías utilizadas como pueden ser php, Drupal, ffmpeg o MySQL y más bien escasos en otras como Apache, html o el paquete XAMPP. Todo ello ha supuesto una parte nada despreciable en los esfuerzos por conseguir el buen funcionamiento del portal.

Implementación página web: Todos los puntos anteriormente descritos en la en diseño del portal se han llevado a cabo en esta parte. Las tareas van desde desarrollar la página en el CMS y sus funcionalidades, hasta el diseño de algunas partes de la base de datos y la comunicación de servicios entre del propio CMS y otras aplicaciones. La mayor parte de estas interrelaciones se describirán con más detalle en la sección destinada al portal web.

Implementació via Drupal: Drupal posee la funcionalidad de ser tremendamente ampliable. Para llevar a cabo ciertas funcionalidades presentes en los objetivos del proyecto han hecho falta la adición de los llamados módulos, que implementan las funcionalidades que no posee la instalación básica del CMS. La se ha efectuado una búsqueda exhaustiva de módulos que mejor satisficieran las necesidades del proyecto y se han usado los módulos CCK, Views, Flashvideo a través del cual se usa ffmpeg y el reproductor de flash JW Player, Image para le creación de galerías de imágenes, Organic Groups que permite el uso de grupos de trabajo dentro del área privada, OG Calendar para la presentación de eventos en un calendario y todo un seguido de módulos menores necesarios para el buen funcionamiento de los módulos principales.

También cabe destacar la funcionalidad de la subida de ficheros de pulsómetro desarrollada especialmente para este proyecto a través del CMS. No se ha usado ningún módulo o funcionalidad externa en su integración con Drupal ya que él mismo posee herramientas adecuadas para este tipo de funcionalidades sencillas.

Testing Drupal: En esta fase se han probado las funcionalidades del portal en los sistemas operativos Linux en las distribuciones Ubuntu 7.10 y 8.05.4, Devian Etch, Windows 2000 y Windows XP, compatibles con Apache, con éxito. También se ha provado en diferentes versiones de los navegadores Internet Explorer 6.0 y 7.0, Mozilla Firefox 2 y 3 y Opera 8, 9.4 y 9.5, también con éxito. Han surgido diversos problemas a la hora del instalar los codificadores de audio y video de ffmpeg ya que es un sistema complicado de instalar. Se han probado varias versiones por compilar y finalmente se ha usado la instalación de paquetes precompilados en las versiones de Linux cuya instalación es mucho más sencilla, y un ejecutable en las versiones de Windows con resultados satisfactorios. La distribución que se presenta en este proyecto incluye una el ejecutable de ffmpeg para Windows sin la necesidad de instalarlo previamente.

Todas las funcionalidades anteriormente mencionadas del portal se han probado satisfactoriamente en las pruebas realizadas en los diversos navegadores y distribuciones de los diversos sistemas operativos.

Migración de Drupal 5 a Drupal 6: En un principio se decidió implementar el portal en la versión 5 de Drupal ya que la versión 6 todavía estaba en una fase de desarrollo poco avanzada y la versión 5 ofrecía en este aspecto una mayor fiabilidad i un soporte técnico muchos más amplio así como un número de contribuciones mucho mayor y más estables. Se contempló la posibilidad de hacer una migración del portal más adelante si las aportaciones y el desarrollo de la nueva versión lo hacían posibles. Fue el caso positivo y el portal finalmente pudo ser migrado de una versión a otra satisfactoriamente y con prácticamente todas las funcionalidades intactas, con lo que podemos alargar el tiempo de soporte y contribuciones oficiales a nuestro portal web, así como nuevos servicios que hacen más fácil . En el apartado *Actualizaciones de Drupal 6* del tercer capítulo se detallarán los pormenores en más profundidad que, básicamente, han sido saber escoger los módulos y las versiones apropiadas para Drupal 6 a partir de la prueba y error.

Mejoras de seguridad en el portal web: Se han llevado a cabo una serie de mejoras en el portal web a través de las actualizaciones de Drupal 6 que solucionan varios problemas de seguridad interna, así como de la mejora del código de la aplicación para subir ficheros de pulsómetro al servidor. Se han llevado a cabo mayoritariamente sobre el código PHP de las aplicaciones para evitar posibles vulnerabilidades contra ataque de Cross Site Scripting (XSS), que consiste en aprovechar la arquitectura del sitio web e insertar una serie de parámetros maliciosos en el código html de la página para acceder a privilegios no autorizados para dichos clientes, y SQL injection que son ataques deliberados contra la base de datos mediante la adición de sentencias en código SQL a variables que contengan dicho código y se usen en la interacción del servidor y la base de datos MySQL. En el apartado dedicado a la especificación del código de la aplicación de ficheros de pulsómetro se explica con más detalle.

Solución de problemas en la base de datos: Como en todo proyecto pueden surgir problemas, en este en concreto debemos resaltar por encima de todo los problemas causados por incompatibilidades en la base de datos del servidor principal donde se hospedan el resto de servicios. Se han detectado anomalías causadas por incompatibilidades del sistema operativo Windows con MySQL, mostrando un error en los archivos temporales generados por la base de datos en el proceso de consulta y modificación de información en las diferentes bases de datos contenidas en el servidor MySQL, anunciando la imposibilidad de escribir en un archivo ya creado. Se han eliminado todos los archivos temporales de dicho directorio y se ha modificado el path del directorio temporal por uno dentro del directorio de instalación del servidor de bases de datos para que el problema no persista. Teniendo en cuenta que también pudiera tratarse de un conflicto entre MySQL y el programa de antivirus, se ha investigado dichos problemas y se ha excluido del escaneo rutinario en directorio temporal para evitar posibles conflictos, dejando dicho escaneo en revisiones en intervalos de tiempo más dilatados. También se ha instalado un servidor alternativo para comprobar que no sugieran más problemas en el funcionamiento del portal web, pero algunos de los servicios del servidor principal no han sido posibles de trasladar, como pueden ser R-php y R o el resto de bases de datos usadas por otros proyectos dada la persistencia del error en la base de datos que impedía su importación a otros servidores.

Elaboración de la memoria i manual de usuario: Dada su extensión y la constante reestructuración del trabajo en el tramo final del proyecto, se decidió empezar la elaboración de la memoria en un plazo anterior al descrito en el informe preliminar a la vez que se solucionaban problemas y se añadían las funcionalidades restantes. En un principio se había planteado como un manual de usuario y una memoria por separados, pero hemos creído oportuno su elaboración en un sólo documento conjunto explicado a varios niveles técnico,s ya que hace que sea más cómoda su consulta y uso. Este es el resultado apreciable de su conclusión.

Diseño entre la comunicación entre Drupal i R-php: Dados los problemas mantenidos con el servidor de bases de datos principal, se ha dejado en segundo término la oferta de este servicio, dado que de ser integrado más exhaustivamente, serían necesarios una cantidad muy superior de recursos y tiempo de implementación e implantación. Se ha optado pues, por un diseño sencillo y funcional de un servicio básico con el cual poder trabajar con las tecnologías ya desarrolladas en el proyecto. En un principio se diseñó la comunicación a través de la especificación e

implementación de un módulo totalmente nuevo para Drupal 6 pero dada su complejidad, hemos utilizado la inserción de código PHP en varias páginas del portal mediante las opciones que Drupal 6 dispone de base para ello y se ha elaborado una guía de programación y desarrollo de módulos de Drupal 6 en el IDE NetBEan 6.5 para facilitar el trabajo en futuras ampliaciones del proyecto y/o desarrollo de terceros en Drupal para el conjunto de proyectos en el cual se engloba el presente. No obstante, la comunicación entre una página web y R mediante R-php ha sido posible su integración con éxito en un proyecto paralelo a éste, englobado en el ámbito del proyecto de investigación sobre la aplicación de modelos estadísticos en deporte de competición y otras actividades físicas.

Testing del proyecto: En este apartado se engloba todas las pruebas realizadas en el portal desarrollado en Drupal, el funcionamiento de la comunicación de la página web con el resto de aplicaciones y en diversos sistemas operativos y navegadores, así como el buen funcionamiento de la base de datos de MySQL y las mejoras de seguridad.

2.3.2. Costes

Uno de los objetivos de este proyecto, como se ha expuesto anteriormente, consiste en proporcionar un servicio lo más barato posible en cuanto a adquisición se refiera, ya que se usará como una herramienta de difusión y aprendizaje gratuita en las comunidades. El uso de software bajo licencias GPL y Creative Commons no es fruto de la casualidad ya que esto nos permite la libre distribución y modificación del proyecto según las necesidades de cada usuario, además de facilitar mucho las cosas en cuanto a ayuda en su desarrollo se refiere.

Pasemos a describir brevemente en qué consisten las licencias de software principalmente usadas en este proyecto:



Las licencias Creative Commons o CC están inspiradas en la licencia GPL (General Public License) de la Free Software Foundation, sin embargo no son un tipo de licenciamiento de software. La idea principal es posibilitar un modelo legal ayudado por herramientas informáticas para así facilitar la distribución y el uso de contenidos.





Existen una serie de licencias Creative Commons, cada una con diferentes configuraciones o principios, como el derecho del autor original a dar libertad para citar su obra, reproducirla, crear obras derivadas, ofrecerla públicamente y con diferentes restricciones como no permitir el uso comercial o respetar la autoría original.

Una de las licencias que ofrecía Creative Commons es la que llevaba por nombre "Developing Nations" (Naciones en Desarrollo), la cual permitía que los derechos de autor y regalías por el uso de las obras se cobraran sólo en los países desarrollados del primer mundo, mientras que se ofrecían de forma abierta en los países en vías de desarrollo. Esta licencia ha sido retirada por problemas comerciales.

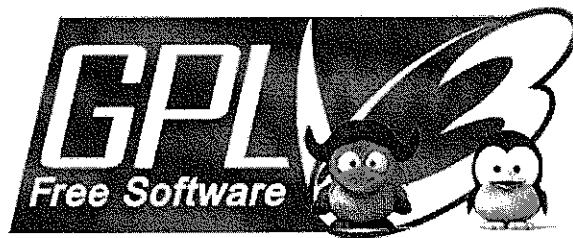
Aunque originalmente fueron redactadas en inglés, las licencias han sido adaptadas a varias legislaciones alrededor del mundo. Entre otros idiomas, han sido traducidas al castellano, al portugués, al euskera y al catalán a través del proyecto Creative Commons International. Existen varios países de habla hispana que están involucrados en este proceso: España, Chile, Argentina, México, Perú, Colombia y Puerto Rico ya tienen las licencias traducidas y en funcionamiento, en

tanto que Guatemala, Ecuador y Venezuela se encuentran en proceso de traducción e implementación de las mismas. Brasil también tiene las licencias traducidas y adaptadas a su legislación.

Las diferentes licencias Creative Commons se basan en combinar distintas propiedades. Estas propiedades son:

-  • **Attribution (by):** Obliga a citar las fuentes de esos contenidos. El autor debe figurar en los créditos.
-  • **Noncommercial o NonCommercial (nc):** Obliga a que el uso de los contenidos no pueda tener bonificación económica alguna para quien haga uso de contenidos bajo esa licencia.
-  • **No Derivative Works or NoDerivs (nd):** Obliga a que esa obra sea distribuida inalterada, sin cambios.
-  • **ShareAlike (sa):** Obliga a que todas las obras derivadas se distribuyan siempre bajo la misma licencia del trabajo original.

Muchas de las licencias Creative Commons se identifican con el acrónimo CC, que hace referencia a su nombre.



La Licencia Pública General de GNU o más conocida por su nombre en inglés, GNU General Public License o simplemente su acrónimo del inglés GNU GPL, es una licencia creada por la Free Software Foundation a mediados de los 80 y está orientada principalmente a proteger la libre distribución, modificación y uso de software. Su propósito es declarar que el software cubierto por esta licencia es software libre y protegerlo de intentos de apropiación que restrinjan esas libertades a los usuarios.

Existen varias licencias "hermanas" de la GPL, como la licencia de documentación libre de GNU (GFDL) que cubre los artículos de la Wikipedia, la Open Audio License para trabajos musicales, etcétera, y otras menos restrictivas, como la MGPL, o la LGPL (Lesser General Public License, antes Library General Public License), que permiten el enlace dinámico de aplicaciones libres a aplicaciones no libres.

La licencia GPL, al ser un documento que cede ciertos derechos al usuario, asume la forma de un contrato, por lo que usualmente se le denomina contrato de licencia o acuerdo de licencia. En los países de tradición anglosajona existe una distinción doctrinal entre licencias y contratos, pero esto no ocurre en los países de tradición civil o continental. Como contrato, la GPL debe cumplir los requisitos legales de formación contractual en cada jurisdicción.

La licencia ha sido reconocida por juzgados en Alemania, particularmente en el caso de una sentencia en un tribunal de Munich, lo que indica positivamente su validez en jurisdicciones de derecho civil.

Después de 14 años de ser usada sin cambios, la licencia GNU GPL vuelve a la mesa de diseño. La Free Software Foundation está actualmente abocada a actualizarla en varios sentidos:

- Resolver formas en que a pesar de todo alguien podía quitar libertades a los usuarios.
- Como un caso especial de lo anterior: Prohibir el uso de software cubierto por la licencia en sistemas diseñados para quitar libertades (DRM).
- Resolver ambigüedades y aumentar su compatibilidad con otras licencias.
- Facilitar su adaptación a otros países.
- Incluir cláusulas que defiendan a la comunidad de software libre del uso indebido de patentes de software.
- Mostrar usuarios registrados

El proceso de revisión de la nueva versión de la licencia se inició el 16 de enero de 2006 en el MIT con la presentación del primer borrador. La versión final de la GPL 3 fue hecha pública el 29 de junio de 2007 y es accesible a través del Portal de GNU.

Costes.

Así pues, bajo estas licencias es posible reducir considerablemente los costes derivados de la utilización de licencias de las distintas aplicaciones usadas para el desarrollo del proyecto que nos ocupa.

En referencia a los costes humanos derivados de la implementación de dichas tecnologías en un producto útil, cabe destacar la separación entre dos tipos de cálculo de costes: el derivado del diseño y desarrollo de la aplicación de subida de archivos de pulsómetro y el resto del proyecto.

Para calcular los costes de la primera parte, se ha utilizado un método llamado COCOMO (CONstructive COSt MOdel) desarrollado por por Barry M. Boehm. Un método de estimación de coste de desarrollo no es otra cosa que establecer una relación matemática entre el esfuerzo y el tiempo requerido para desarrollar un producto / proyecto.

Lo primero que necesitamos saber es el tipo de proyecto del que estamos hablando, hay tres tipos diferentes:

- **Orgánico:** proyectos relativamente sencillos. En este modo, un pequeño grupo de programadores experimentados desarrollan software en un entorno familiar. El tamaño del software varía de unos pocos miles de líneas (pequeño) a unas decenas de miles de líneas (medio).
- **Semi-acoplado:** proyectos intermedios en complejidad y tamaño. En este tipo, el proyecto tiene unas fuertes restricciones, que pueden estar relacionadas con el procesador y el interface hardware. El problema a resolver es único y es difícil basarse en la experiencia, puesto que puede no haberla.
- **Empotrado:** proyectos bastante complejos, en los que apenas se tiene experiencia y se engloban en un entorno de gran innovación técnica. Además se trabaja con unos requisitos muy restrictivos y de gran volatilidad.

Ahora por otro lado tendríamos los diferentes modelos que define COCOMO: Modelo básico, Modelo Intermedio y Modelo Avanzado.

En este punto quiero recordar que este tipo de modelos están orientados a la magnitud del producto final, midiendo el tamaño del proyecto en líneas de código principalmente, por el hecho de realizarse en base al número de líneas de código se mide el producto y su tamaño pero en ningún momento podremos llegar a medir la productividad.

Categoría	Programadores	Duración	Líneas de código	Ejemplo
Trivial	1	0 - 4 semanas	< 1k	Utilidad de ordenación
Pequeño	1	1 - 6 meses	1k - 3k	Biblioteca de funciones
Media	2 - 5	0,5 - 2 años	3k - 50k	Compilador de C
Grande	5 - 20	2 - 3 años	50k - 100k	SO pequeño
Muy grande	100 - 1000	4 - 5 años	100k - 1M	Grandes SO
Gigante	1000-5000	5 - 10 años	> 1M	Sistema de Distribución

Figura 2-5. Dificultad de un proyecto en función de sus líneas de código.

Proyecto de software	a	b	c	d
Orgánico	2,4	1,05	2,5	0,38
Semiacoplado	3	1,12	2,5	0,35
Empotrado	3,6	1,2	2,5	0,32

Figura 2-6. Coeficientes COCOMO

La ecuación del esfuerzo de COCOMO es:

$$E = a(Kl)^b * m(X)$$

Donde:

a y b son constantes con valores definidos en cada submodelo

Kl son las líneas de código (en miles)

el resultado es en salarios/mes y horas-hombre.

$m(X)$ Es un multiplicador que depende de 15 atributos.

En nuestro caso usaremos el modelo básico ya que describe a la perfección la necesidades pequeño código desarrollado. Dicho código posee unas 500 líneas de código por lo tanto, según hemos podido ver se trata de un Proyecto Orgánico, por lo tanto el tamaño es de 0,5 miles de líneas de código. Si aplicamos las fórmulas suponiendo el sueldo medio de un programador senior son 2166,67 €/mes en bruto, estzs son las fórmulas usadas:

Personas necesarias por mes para llevar adelante el proyecto (MM) = $a*(Kl)^b$

Tiempo de desarrollo del proyecto (TDEV) = $c*(MM)^d$

Personas necesarias para realizar el proyecto (CosteH) = MM/TDEV

Coste total del proyecto (CosteM) = CosteH * Salario medio entre los programadores y analistas.

$$MM = 2,4 * 0,5^{1,05} = 1,16 \text{ personas / mes}$$

$$TDEV = 2,5 * 1,16^{0,38} = 2,64 \text{ meses}$$

$$\text{CosteH} = MM/TDEV = 1,16/2,64 = 0,44 \text{ personas (para 1 persona se necesitan 34,85 días aprox.)}$$

$$\text{CosteM} = 0,44 * 2166,67 = 953,33 \text{ €}$$

Si comparamos con el trabajo realizado en la previsión y el diagrama de gannt veremos una mejoría en el precio considerable suponiendo el mismo precio en el salario del personal.

Para el resto de horas podemos realizar un sencillo cálculo entre las horas y el sueldo medio en bruto de un analista de sistemas:

Sueldo en bruto de un analista de sistemas por hora: 33.000 €/año - 17,19 €/mes (suponiendo 20 días laborables por mes y una jornada de 8 horas)

Horas/día de trabajo: 4h/día

Horas totales: $217 * 4 = 868$ h

Horas sin contar desarrollo de software: $858 - 34,85 * 8 = 579,2$ h (ya que el cálculo de desarrollo de software preveía jornadas de 8 horas semanales)

Coste total: $579,2 * 17,19 + 953,33 = 10909,778$ €

2.4. Drupal

2.4.1. ¿Porqué Drupal?

Préviamente ya se han expuesto razones de porqué se escogió un CMS a la hora de implementar el portal pero también se ha mencionado que se barajaron varias posibilidades en cuanto a CMS se refiere. Primordialmente se escogieron como opciones Drupal y Joomla ya que son dos de los CMS más consolidados en Internet y tras investigar un poco, nos decantamos por Drupal por varios motivos expuestos a continuación. He aquí la relación de pros y contras de cada uno de ellos:

Drupal

- Plataforma muy sólida y de gran calidad técnica.
- Permite una mantener un varios sitios reales en un mismo servidor (solo una instalación para varios sitios).
- Cualquier tipo de permisos de usuario, grupos de usuario además de cumplir con el estándar OpenId en la versión 6.
- Permite desarrollar una membresía y una comunidad de sitios aparte del un solo CMS.
- Posee un potente sistema de templates. Cualquier template de XHTML o CSS puede ser fácilmente convertido en Drupal.
- Necesita una inversión de tiempo mayor para tener en cuenta las tremendas posibilidades que posee en el desarrollo de un sitio.
- Código claro y de gran calidad. Posee una API (fácil de integrar con otras soluciones, etc.)
- Flexibilidad y limitaciones aún desconocidas.
- Permite gestionar sitios con gran volumen de visitas. (p.ej.: MTV UK, BBC, the Onion, Nasa, Greenpeace UK, etc.)

Joomla

- Si el nivel técnico del usuario no es muy avanzado, es un buen comienzo.
- Fácil de instalar y poner a punto con un simple clic de ratón.
- Curva de aprendizaje fácil.
- No permite la integración de otros scripts en el sitio.
- Generalmente no se pueden crear sitios de gran tráfico sin invertir una gran cantidad de esfuerzo y recursos.
- No posee un gran buscador interno y las URLs no son fáciles de manejar por los buscadores.
- La utilización de recursos del servidor es mayor en comparación con Drupal.
- Solo permite un sitio por instalación.
- No permite el registro centralizado en varios sitios.
- No permite grupos de usuarios y permisos.

- El interfaz de administración es más intuitivo.
- Posee algunos módulos de mayor calidad para servicios como calendarios, encuestas, etc.
- La ampliación de módulos cuesta dinero.

Requisitos del sistema:

- Drupal permite trabajar con MySQL y Postgre mientras Joomla sólo se tiene constancia de que soporte MySQL.
- Drupal permite trabajar con Apache o IIS mientras que Joomla aparentemente solo lo hace con Apache
- Joomla soporta paginas y registro de SSL. Drupal no se tiene constancia de que lo haga.

Administración del sitio:

- Drupal posee modulo gratuitos para la gestión de Workflow. No se tiene constancia que Joomla not posea alguno.
- Drupal posee un gestor de traducciones interno. Joomla posee un addon gratuito para lo mismo.
- Drupal posee una gestión de privilegios mucho más granular.

Interoperabilidad:

- Drupal posee soporte para iCal (Add on) , Joomla no se tiene constancia que posea alguno.
- Drupal cumple los estándares de XHTML. Joomla no se tiene constancia que lo haga.
- Drupal posee un excelente seguimiento de versiones por parte de la comunidad, cosa de la que Joomla carece.

Después de de este balance es fácil decantarse por una opción sabiendo los objetivos de este proyecto.

2.4.1 Estructura y funcionamiento.

Drupal funciona en a diversos niveles con diversas tecnologías tal y como se muestra en la figura 2-7 .

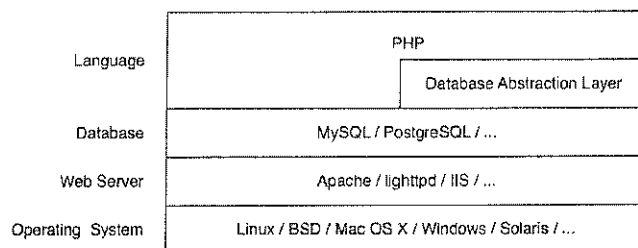


Figura 2-7. Capas en el diseño de Drupal

Descritas de abajo a arriba, se encuentra en primer lugar el sistema operativo. Está en un nivel demasiado bajo como para que Drupal pueda verse afectado y funciona perfectamente en cualquier sistema operativo que soporte PHP.

El servidor que más se usa hoy en día con Drupal es Apache ya que la mayor parte de su desarrollo se ha llevado a cabo con dicho servidor. Drupal utiliza los ficheros *.htaccess* para

asegurar una instalación segura en el servidor. Posee la opción de trabajar con URLs limpias (que significa que no aparecen los parámetros de ejecución de cada página del CMS en la barra de direcciones del navegador con el cual se acceda al contenido) a través de el módulo de Apache *mod_rewrite*. Este concepto es importante ya que al migrar desde otro CMS o desde una página estática, no es necesario cambiar las direcciones para su correcto funcionamiento.

Drupal se comunica con la siguiente capa de la pila (la base de datos) a través de una ligera capa de abstracción de la base de datos. Esta capa permite el saneamiento de las sentencias SQL en cuanto a seguridad se refiere y hace posible la utilización los productos de varios fabricantes sin tener que rehacer el código del sitio en caso de cambiar la el software de la base de datos. Las bases de datos más testeadas en Drupal son MySQL y PostgreSQL. Aunque Drupal acceda a la base de datos mediante la capa de abstracción, también permite usar otras bases de datos mediante php de manera tradicional a través de Snippets, que son fragmentos de PHP insertados en las páginas de Drupal. Hablaremos de ellos con detalle más adelante.

Drupal está escrito en PHP y al ser un lenguaje interpretado es necesaria la instalación en el servidor donde se haga funcionar el CMS. PHP tiene mala reputación dentro del mundo del desarrollo de aplicaciones web ya que es fácil aprender y gran parte del código que se usa en Internet ha sido escrito por principiantes. También, como en muchos otros lenguajes de programación, es a menudo atacado o usado para hackear sistemas. De todas formas es posible escribir código robusto y fiable y la prueba es que Drupal está escrito bajo unas estrictas normas de programación dictadas por los estándares PEAR.

El núcleo.

Un framework ligero compone el núcleo de Drupal, la parte que se obtiene al descargar drupal de *drupal.org*. Es el responsable de proveer las funcionalidades básicas que se usen para dar soporte a otras partes del sistema. Incluye código que permite al sistema de Drupal arrancar (bootstrap) cuando recibe una petición una librería común de funciones usadas frecuentemente en Drupal y módulos que permiten usar funcionalidades básicas como gestión de usuarios, taxonomía y el uso de plantillas (templates).

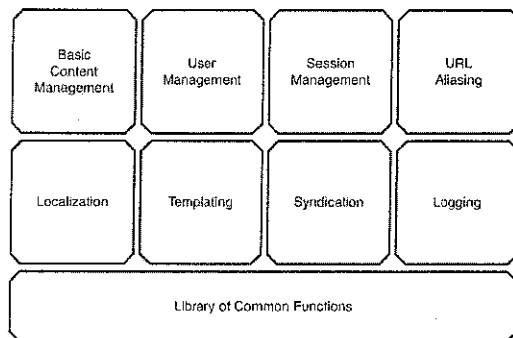


Figura 2-8. Una vista general del núcleo de Drupal (Parcial.)

El interfaz administrativo.

El interfaz administrativo está estrechamente integrado con el resto del sitio y por defecto, utiliza el mismo tema. El primer usuario, el 1, es el superusuario el cual posee acceso completo a todas las funcionalidades del sitio. Después de registrarse como el usuario 1, es posible ver un enlace en la página de administración con el boque de dicho usuario (ver más en la sección de bloques). Cada bloque de usuario puede contener diferentes enlaces dependiendo del nivel de acceso que posean en el sitio web.

Módulos

Drupal es un framework tremendamente modular. Las funcionalidades están incluidas en *módulos*, los cuales pueden ser activados o desactivados (algunos pueden no ser desactivados ya que puede que otros módulos necesiten de su activación para su funcionamiento). Los nuevos servicios se añaden a una página en Drupal mediante la activación de módulos existentes en el sistema, instalando módulos escritos por miembros de la comunidad de Drupal o escribiendo los propios. De esta manera, los sitios web que no necesiten algunas de las funcionalidades que se ofrecen con la instalación básica o módulos previamente usados y actualmente no necesitados, pueden desactivar su uso permitiendo un mejor funcionamiento del sistema, y aquellos que necesiten nuevas funcionalidades, pueden añadirlas sin problemas. Esto se puede observar en la figura 2-9.

La adición de nuevos tipos de contenido tales como recetas, entradas de blog, ficheros adjuntos o nuevos comportamientos como notificaciones por correo electrónico, publicaciones peer-to-peer y la agregación de feeds, están regidas y administradas a través de módulos.

Drupal hace uso de patrones de diseño llamados *control de inversión*, en las cuales las funcionalidades modulares son llamadas por el framework a su tiempo. Estas oportunidades para los módulos de hacer su función son llamadas *hooks*.

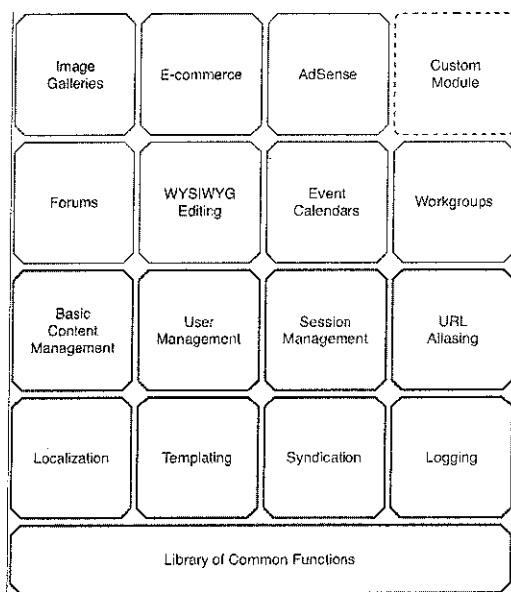


Figura 2-9. Al activar módulos adicionales, se añaden más funcionalidades.

Hooks

Los *hooks* (o ganchos, en inglés) pueden pensarse como eventos internos de Drupal. También se llama *callbacks*, dado que están contruidos a más bien través de una convención de nombramientos de funciones que por una espera activa, ya que no son llamados realmente por un proceso específico del cual parten. Los *hooks* permiten cualquier módulo “engancharse” a lo que sucede en el resto de Drupal.

Supongamos un registro de usuarios en un sitio contruido a través de Drupal. En el mismo momento que el usuario se valida en el sitio, Drupal ejecuta un *hook*, lo que significa que cualquier función nombrada de acuerdo con la convención *nombre_del_módulo* más *nombre_del*

_hook, será llamada en ese instante. Por ejemplo, la función *comment_user()* en el módulo *comment*, *locale_user()* en el módulo *locale*, *node_user()* en el módulo *node* y cualquier otra función nombrada de la misma forma será llamada por el sistema. Si se escribiera un módulo personalizado llamado *spam.module* e incluyera una función llamada *spam_user()* que enviara un e-mail al usuario, la función sería llamada también y dicho usuario recibiría un correo electrónico en cada inicio de sesión. La manera más común de entrar en las funcionalidades del núcleo de Drupal es mediante la implementación de *hooks* en módulos.

Temas

Al crear una página web que enviar a un navegador, hay dos razones a tener en cuenta: construir los datos pertinentes y presentarlos de cara a la Web. En Drupal, la capa de temas (theme layer) es la responsable de crear el código HTML que el navegador recibe. Se pueden usar diferentes aproximaciones populares sobre cómo crear temas tales como Smarty, un Lenguaje de atributos para la creación de temas para PHP (PHPATAL) y PHPTemplate. Es importante señalar que Drupal recomienda la separación de contenido y presentación.

Drupal permite varias maneras de personalizar y reemplazar el *look and feel* de un sitio web. La manera más simple es mediante el uso de CSS para reemplazar la clases preconstruidas y los IDs de Drupal. De todas formas, si lo que se desea es ir más allá de personalizar la salida HTML, es fácil de hacer. Los ficheros de patrones de Drupal consisten en código estándar de HTML y PHP. Además, cada parte dinámica de una página de Drupal (como un recuadro o una lista) pueden ser reemplazadas simplemente declarando una función con el nombre apropiado, usando Drupal entonces dicha función en vez de la original.

Nodos

Los tipos de contenido en Drupal derivan de un tipo básico referenciado como *nodo*. Ya sea una entrada de blog, una receta o incluso una tarea de un proyecto, la estructura subyacente de datos es la misma. El mecanismo bajo esta aproximación es su extensibilidad. Los desarrolladores de módulos pueden añadir nuevas funcionalidades tales como valoraciones, comentarios, ficheros adjuntos, información geolocalizada y demás para nodos en general sin preocuparse si el tipo de nodo es blog, receta, etc. El administrador del sitio puede mezclar y unir funcionalidades mediante el tipo de contenido, como por ejemplo, escogiendo la opción de permitir comentarios en blogs pero no recetas o activando la subida de ficheros para tareas de proyectos solamente.

Los nodos también pueden contener una colección básica de propiedades de comportamiento que otros tipos de contenido pueden heredar. Cualquier nodo puede ser promocionado a la página principal, ser marcado como publicado o no, o incluso permitir su búsqueda. Es gracias a esta estructura uniforme que el interfaz administrativo ofrece una pantalla de edición para trabajar con nodos.

Bloques

Un bloque es información que puede ser activada o desactivada en un lugar específico del sitio web. Es posible tener un bloque que contenga a los usuarios más activos o a una lista de eventos venideros. Los bloques se colocan típicamente en la barra lateral de una plantilla, o en la cabecera o a pie de página. También se pueden configurar para mostrarse en ciertos tipos de nodos, solamente en la página principal o de acuerdo con los criterios de cada uno.

A menudo, los bloques se usan para representar información propia para el usuario que tenga una sesión abierta en un momento dado. Por ejemplo, un bloque de navegación contiene enlaces solamente a las funciones de administración solo para aquellos que tengan este tipo de acceso. La colocación y la visibilidad de los bloques se administra a través de un interfaz web en las opciones de administración.

El diseño del sistema de ficheros.

Es importante entender la estructura de directorios de Drupal si queremos depurar los errores que podamos encontrar en nuestra página web y saber sobre varias de las mejores prácticas a seguir, tales como: Dónde se encuentran los ficheros subidos al servidor, dónde se deberían encontrar los temas o cómo poder disponer de varios perfiles de Drupal. La figura 2-10 muestra los directorios de una instalación por defecto de Drupal.

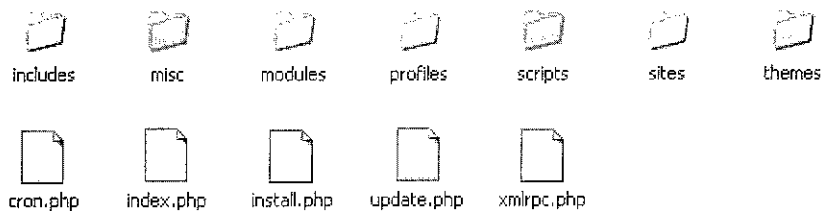


Figura 2-10. La estructura de carpetas en una instalación de Drupal por defecto.

Los detalles sobre cada uno de los elementos en la estructura de directorios son:

- El directorio *includes* contiene la librería de funciones comunes que usa Drupal.
- El directorio *misc* guarda código JavaScript y varios iconos e imágenes disponibles de la instalación inicial de Drupal.
- El directorio *modules* contiene los módulos núcleo, cada uno en su propio directorio. Es preferible no modificar nada en este directorio para evitar problemas.
- El directorio *profiles* contiene diferentes perfiles de instalación para un sitio. Si hubiera otros perfiles además del perfil por defecto en este subdirectorio, Drupal preguntaría que perfil se quiere instalar la primera vez que se instala el sitio en Drupal. La principal razón de un perfil de instalación es activar ciertos módulos núcleo contribuidos automáticamente como podrían ser las instalaciones en un idioma diferente del inglés o un perfil de e-commerce que automáticamente configurara Drupal como una plataforma de este tipo.
- El directorio *scripts* contiene scripts para comprobar sintaxis, sanear código y gestionar casos especiales con *cron*. No se usa dentro del ciclo de vida de las peticiones dentro del sistema; son scripts de shell y Perl para el mantenimiento, tareas puntuales u otras situaciones en las que se requieran.
- El directorio *sites* (ver figura 2-11) contiene todas las modificaciones que necesitemos hacer en Drupal en la forma de ajustes, módulos y temas. Cuando añadimos módulos a Drupal del repositorio de módulos contribuidos o los nuestros propios, se colocan en *sites/all/modules*. Esto permite mantener todas las modificaciones que se hagan a Drupal en un mismo directorio. Dentro de *sites* hay un directorio llamado *default* que contiene el archivo de configuración por defecto del sitio en Drupal: *settings.php*. El directorio *default* normalmente se copia y renombra con el nombre del URL del sitio en cuestión, quedando la configuración de dicho sitio en *sites/www.example.com/settings.php*, donde *www.example.com* es el nombre del sitio. El directorio *themes* contiene el sistema de plantillas y los temas por defecto en Drupal. Para añadir temas, basta ubicar los temas nuevos en *sites/all/themes* para que puedan mostrarse correctamente.
- Uno de los directorios a destacar es *sites/default/files*. Es el lugar donde se ubican todos los archivos para personalizar el logotipo del sitio web, permitir el uso de avatares o subir archivos asociados con el sitio en cuestión, así como los archivos de traducción de contenido. Es importante que este directorio posea permisos de lectura y escritura adecuados a través del servidor web en el que Drupal funcione. Posteriormente se puede cambiar la ubicación de *files* a otro lugar a través del panel de administración.

- El fichero *robots.txt* permite que los programas de búsqueda (*spiders*) usados por buscadores como Google, omitan las URLs, ficheros y directorios listados en este archivo por URL convencionales o limpias (ver capítulo 3.2.4. *Administración*, sección *Configuración de sitio* → *URLs limpias*).

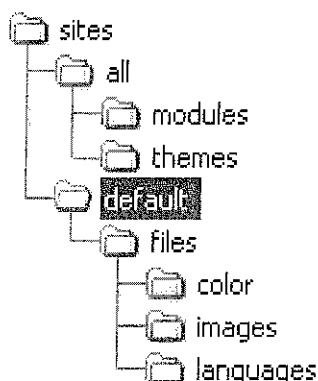


Figure 2-11. El directorio sites permite guardas todas las modificaciones hechas a Drupal.

Sirviendo una petición.

Esta sección describe un esquema de lo que ocurre cuando una petición es recibida por Drupal:

El rol del servidor Web: Drupal se ejecuta bajo un servidor web que acostumbra a ser Apache. Si el servidor web respeta el fichero *.htaccess* de Drupal, algunos ajustes de PHP se inician y las URL limpias se activan (para ello es necesario previamente que *mod_rewrite* se active). Drupal traduce una dirección a *index.php?=foo/bar* en *index.php/foo/bar* a través de una regla de *mod_rewrite* pero continua usando la primera dirección internamente (guardada en el parametro de petición de URL *q*). A esta dirección interna se la llama *Drupal path*.

En otros servidores web alternativos como Microsoft IIS, las URL limpias se pueden guardar mediante un módulo ISAPI (Windows Internet Server Application Programming Interface) como ISAPI_Rewrite.

El proceso de Bootstrap

Drupal se autoarranca (bootstrap) a sí mismo en cada petición a través de una serie de fases de bootstrap. Estas fases se definen en *bootstrap.inc* y proceden como se describe a continuación:

- **Configuración:** Esta fase llena el array interno de configuración de Drupal y establece la URL base (*\$base_url*) del sitio. El archivo *settings.php* se parsea vía la función *include_once()*, y cualquier variable establecida como sobreescrita con anterioridad es aplicada.
- **Caché de Early page:** En situaciones donde se requiere un gran nivel de escalabilidad, un sistema de caché puede necesitar se invocado antes de intentar siquiera establecer una conexión a la base de datos. La fase de caché de early pages permite incluir (con *include()*) un fichero PHP que contenga una función llamada *page_cache_fastpath()*, la cual almacena y devuelve el contenido al navegador. La caché de early pages se activa a través de la variable *page_cache_fastpath* a CIERTO y el fichero a incluir se define asignando a variable *cache_inc* en el path del fichero PHP.

- **Base de Datos:** Durante la fase de la base de datos, se determina el tipo de base de datos y se hace la conexión inicial que se usará para las subsiguientes peticiones a dicha base de datos.
- **Acceso:** Drupal permite restringir hosts por relaciones de entre nombre del host/dirección IP. En la fase de acceso se comprueba rápidamente si la petición proviene de un host no permitido, en cuyo caso el acceso es denegado.
- **Sesión:** Drupal saca partido de la gestión de sesiones de PHP pero sustituye algunas de las reglas de gestión por otras propias para implementar una administración de sesiones con fondo en base de datos. Las sesiones se inician o se reestablecen en la fase de sesión.
- **Caché de late pages:** Aquí es donde el sistema carga suficiente código de soporte para determinar si servir una página de la caché de páginas o no. Esto incluye mezclar configuraciones de la base de datos en un array que ha sido creado durante la fase de configuración, con la carga o el parseo de código de módulos. Si la sesión indica que la petición ha sido hecha por un usuario anónimo y el caching de páginas se ha activado, la página se devuelve de la caché y la ejecución se para.
- **Dirección:** En la fase de dirección, se carga el código que gestiona direcciones y alias de direcciones. Esta fase permite que se resuelvan URLs que puedan ser interpretadas por las personas y gestiona direcciones internas de Drupal.
- **Completa:** Esta fase completa el proceso de bootstrap mediante la carga de la librería de funciones comunes, soporte de temas y soporte para mapeado de callbacks, gestión de ficheros, Unicode, juegos de herramientas de imagen de PHP, la creación y proceso de formularios, tablas autoordenables y los resultados del inicio de páginas. En gestor de errores personalizado de Drupal se inicia, *locale* se inicia y todos los módulos activos se cargan. Finalmente, Drupal ejecuta el hook *init* de manera que los módulos tengan la oportunidad de ser notificados antes del comienzo del proceso oficial de peticiones. Una vez Drupal ha completado el arranque inicial, todos los componentes del framework restan disponibles. Es tiempo de recoger la petición del navegador y pasarla a la función de PHP que la gestione. El mapeado entre URLs y las funciones que las manejan tiene lugar usando un callback de registro que se ocupa del mapeado de la URL y del control de acceso. Los módulos registran sus callbacks usando el hook *menu*. Cuando Drupal ha determinado que existe un callback hacia el lugar en que se encuentra mapeada la URL de la petición del navegador y que el usuario tiene suficientes privilegios para acceder a dicho callback, se ejecuta dicho callback.

Procesando una petición

La función de callback hace el trabajo que se requiera para procesar y acumular los datos necesarios para completar la petición. Por ejemplo, si se recibe una petición de contenido tal como `http://example.com/q=node/3`, la URL se mapea en la función `node_page_view()` en `node.module`. El posterior procesamiento obtendrá los datos para dicho nodo de la base de datos y lo pondrá en una estructura de datos. Entonces es cuando entra la colocación del tema.

Colocando el tema de la página a los datos.

Este proceso involucra la transformación de los datos obtenidos, manipulados, o creados en código HTML. Drupal usará el tema que el administrador haya seleccionado para dar a la página la correcta presentación deseada y enviará en código HTML resultante al navegador.

3. EL PORTAL WEB

3.1. Instalación

Durante este capítulo describiremos detalladamente el proceso de instalación del sitio a través de un tutorial que guiará al usuario en cada momento. La instalación del portal web consta de tres partes o fases bien definidas: la instalación de los archivos de Drupal 6 en el servidor, la instalación de la base de datos de MySQL y la instalación del códer ffmpeg y el reproductor de flash JW Player.

3.1.1. Archivos de Drupal.

Drupal posee un mecanismo de instalación en el caso que hagamos una instalación desde cero, que nos guiará en el proceso. Por nuestra parte, dicho proceso no será necesario ya que poseemos una instalación ya completada. El proceso a seguir es muy sencillo:

- Copiamos el archivo *drupal.zip* en el directorio raíz de nuestra página web y descomprimos su contenido en él.
- Debemos editar el archivo *settings.php* situado en *sites/default/* con un editor de texto y buscar la siguiente línea marcada:

```
* Database URL format:  
* $db_url = 'mysql://username:password@localhost/databasename';  
* $db_url = 'mysqli://username:password@localhost/databasename';  
* $db_url = 'pgsql://username:password@localhost/databasename';  
*/  
$db_url = 'mysqli://usuario:password@localhost/drupal_bd';  
$db_prefix = '';
```

Figura 3-1. Settings.php

Debemos substituir en la línea marcada en azul de la figura 3-1 *usuario* por nuestro usuario en la base de datos en MySQL y *password* por la contraseña de dicho usuario para la base de datos y guardar el archivo. Una vez hecho esto, habremos terminado esta fase de la instalación.

3.1.2 La base de datos.

Para instalar la base de datos necesitamos es necesaria la herramienta *phpmyadmin* que se encarga de gestionar bases de datos en MySQL. Los pasos a seguir son sencillos a la hora de introducirla en nuestro sistema. Debemos importar el archivo *drupal_bd.sql* tal y como muestran las figuras 3-2, 3-3, 3-4, 3-5 y 3-6. Previamente, debemos crear la base de datos llamada *drupal_bd* y una vez creada, acceder a ella mediante un simple clic en su nombre, situado en la barra lateral izquierda, tal y como muestran la figuras 3-7. y 3-8.

localhost

- Versión del servidor: 5.0.51b-community-nt
 - Versión del protocolo: 10
 - Servidor: localhost via TCP/IP
 - Usuario: root@localhost
- Juegos de caracteres de MySQL: UTF-8 Unicode (utf8)
- Cotejamiento de las conexiones MySQL:
utf8_unicode_ci
- Crear nueva base de datos**
Cotejamiento
Crear
- Mostrar información de tiempo de ejecución de MySQL
- Mostrar las variables del sistema MySQL
- Procesos
- Juego de caracteres y sus cotejamientos
- Motores de almacenamiento
- Cargar los privilegios nuevamente
- Privilegios
- Bases de datos
- Exportar
- Importar

phpMyAdmin - 2.11.7

- Versión del cliente: 5.0.51b
- Extensiones PHP utilizadas: mysql
- Idioma - Language: Español - Spanish
- Tema / Estilo: Original
- Tamaño del font: 82%
- Documentación de phpMyAdmin
 - Wiki del phpMyAdmin
- Página oficial de phpMyAdmin
 - [ChangeLog] [Subversion] [Lists]



Figura 3-2. Crear base de datos *drupal_bd*.

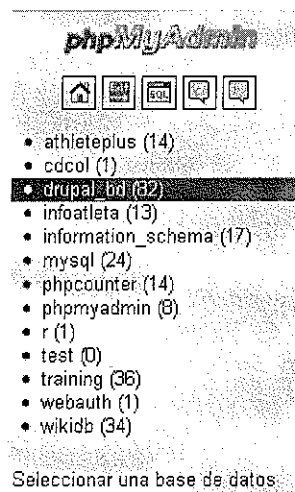


Figura 3-3. seleccionar base de datos *drupal_bd*.

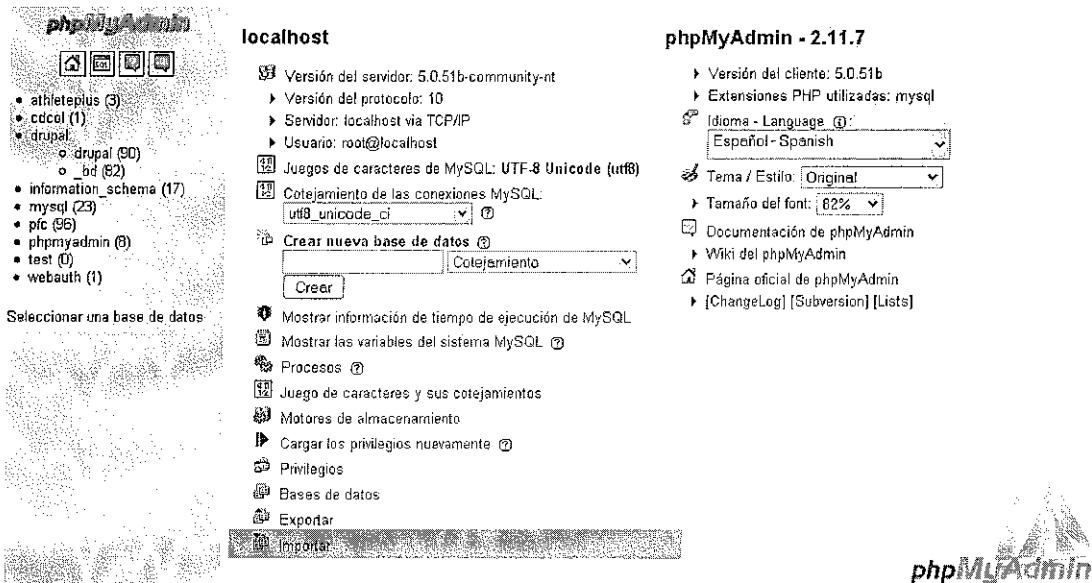


Figura 3-4. Importar base de datos. Paso 1.

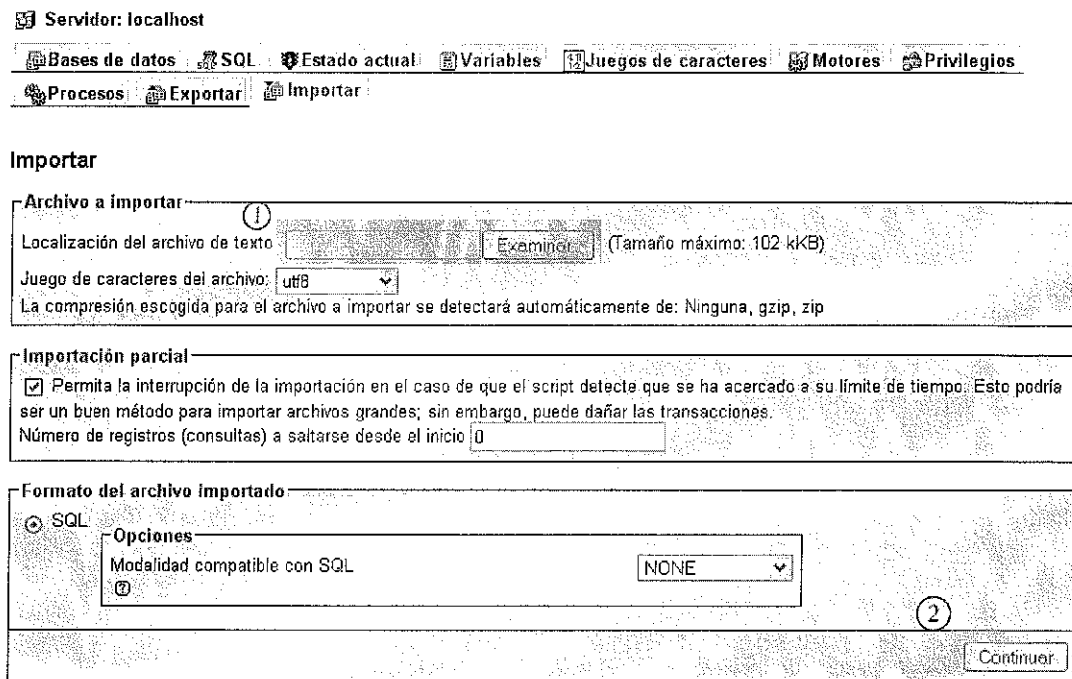


Figura 3-5. Importar base de datos. Paso 2.

Después de haber instalado la base de datos es importante, una vez funcione el portal, proporcionar el path correcto de los archivos del servidor web. Para ello introduciremos dicho path en el campo de *Carpetas temporal* en Administración→Configuración del sitio→Sistema de ficheros tal y como muestra la figura 3-6:

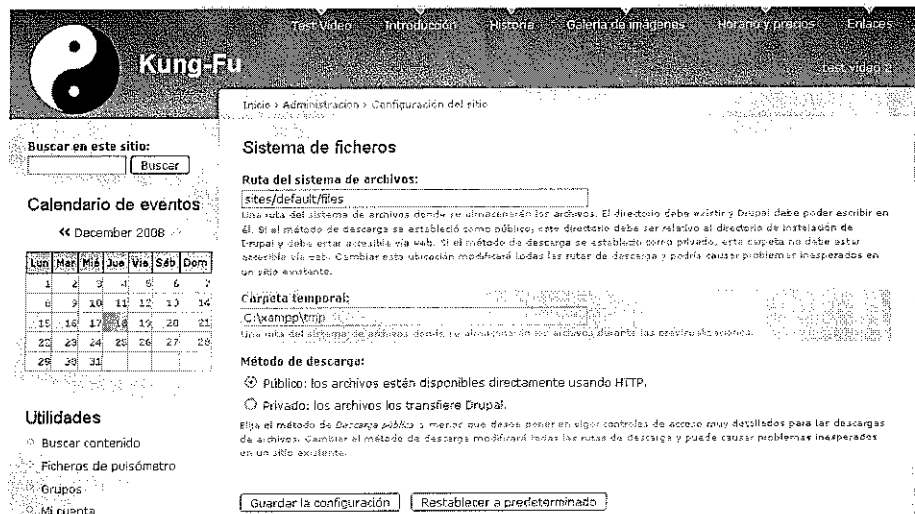


Figura 3-6. Menú de Sistema de ficheros.

En caso de querer instalar un sitio adicional solo es necesario crear una base de datos en el servidor MySQL con nombre diferente e importar los archivos existentes con el procedimiento descrito anteriormente en este punto y sustituir en la línea de código descrita en la figura 3-1 *drupal_bd* por el nombre de la base de datos que hayamos creado. El resto de pasos son exactamente iguales.

Es muy importante que en las bases de datos que hayamos creado con el usuario con el cual nos conectaremos al servidor MySQL tengamos los siguientes privilegios en dicha base de datos: SELECT, INSERT, UPDATE, DELETE, CREATE, DROP, INDEX, ALTER, CREATE TEMPORARY TABLES, LOCK TABLES. Si no poseemos esos privilegios, nuestro portal web no funcionará correctamente y puede ocasionar problemas graves, para ello podemos ejecutar la siguiente instrucción seleccionando la pestaña superior SQL y ejecutar la siguiente instrucción:

```
GRANT SELECT, INSERT, UPDATE, DELETE, CREATE, DROP, INDEX, ALTER, CREATE
TEMPORARY TABLES, LOCK TABLES ON database.* TO 'username'@'localhost'
IDENTIFIED BY 'password';
```

Donde *database* es el nombre de la base de datos (*drupal_bd* u otra en caso de haber instalado más de un sitio) *username* es el nombre del usuario y *password* es la contraseña para dicho usuario. Si no disponemos de privilegios suficientes para ejecutar esta instrucción no devolverá un mensaje de error indicándonoslo, debemos contactar con el administrador del servidor para pedir dichos privilegios o para que nos conceda los privilegios mencionados anteriormente para las bases de datos que lo necesiten.

Multisite

Drupal ofrece la opción de gestionar varios sitios web que posean diferentes funcionalidades y que de cara al usuario final se presenten como portales independientes, pero que en el fondo comparten módulos básicos y contribuidos, la base de datos y algunas, todas o ninguna opción de configuración. El procedimiento a seguir es sencillo.

Añadimos a *setting.php* por cada sitio que deseemos:

```
$db_url = 'mysql://username:password@localhost/drupal_bd'
$base_url = 'http://sitio'
```


Donde *username* es el usuario con el que nos conectamos a la base de datos, *password* es la contraseña para dicho usuario y *sitio* es el nombre del sitio a añadir. Seguidamente deberíamos configurar el archivo *hosts* (en el caso de Windows: *System32\drivers\etc*, en el caso de Unix: */etc*) y añadir:

(Dirección IP sitio1) (nombre del sitio1)
(Dirección IP sitio2) (nombre del sitio2)

...

Ejemplo:

127.0.0.1 kung-fu

Y por último editar el fichero de Apache *httpd-vhosts.conf* y añadir para cada uno de los sitios:

```
<VirtualHost *:80>  
DocumentRoot Directorio_Sitio  
ServerName sitio  
</VirtualHost>
```

Donde *Directorio_Sitio* es el directorio de instalación de cada sitio en el disco duro y *sitio* es el nombre de cada sitio.

Por último, deberemos copiar los archivos personalizables de cada uno de los sitios en el directorio de Drupal a */sites/NombreSitio/* quedando una estructura tal que así:

```
sites/NombreSitio/modules  
sites/NombreSitio/files  
sites/NombreSitio/themes
```

Podemos personalizar la configuración de cada uno de los sitios añadiendo un *settings.php* a su directorio base. Es importante saber que la configuración y archivos comunes se encuentra en *sites/default/all* y *sites/all* respectivamente.

3.1.3. FFMPEG y JW Player

Aunque este códer es una herramienta tremendamente potente, es también complicada de instalar dependiendo de las funciones que queramos agregar. Es posible descargarse los archivos de código fuente para compilarlos en nuestro sistema pero también es muy poco recomendable si no somos expertos en ello. Para los usuarios que usen Windows, la instalación de los archivos de Drupal ya posee una versión compilada y funcional de FFMPEG en el directorio *sites/all/coder/*, con lo cual no hará falta nada más que cambiar el path de ejecución en las opciones del configuración del módulo Flashvideo en el panel de administración (*Administración* → *Configuración del sitio* → *Preferencias de Flashvideo* → *Global FlashVideo Settings (For all node types)*). Es importante introducir el path completo incluyendo la unidad de disco en que está instalada en caso de Windows, y no el relativo al servidor. En caso de contratar un servicio de hosting, será necesario contactar con el administrador para que nos facilite el path a nuestro directorio raíz.

NodeType	Estatus	Operaciones
Global Settings	(Enabled)	Global FlashVideo Settings (For all node types)
Imagen	(Disabled)	FlashVideo Settings for this node type.
Evento	(Enabled)	FlashVideo Settings for this node type.
Grupo	(Disabled)	FlashVideo Settings for this node type.
Página	(Enabled)	FlashVideo Settings for this node type.
Historia	(Enabled)	FlashVideo Settings for this node type.

Figura 3-7. Opciones de Flashvideo

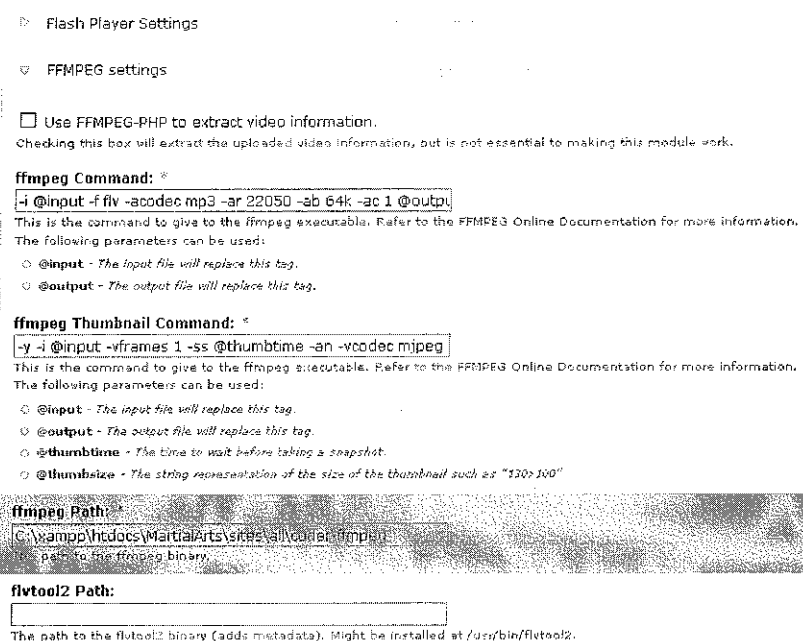


Figura 3-8. Path a los archivos de FFMPEG.

En el caso de poseer una instalación de Linux, aconsejamos instalar una versión precompilada con el gestor de paquetes que c la distribución de Linux que tengamos. En el caso de una Debian Etch podemos hacer lo siguiente:

`pico /etc/apt/sources.list`

Añadimos la siguiente línea:

`deb http://www.debian-multimedia.org stable main`

Esto nos permite instalar una versión precompilada en nuestro servidor con las opciones que necesitamos para la codificación de vídeo

Ejecutamos:

`apt-get update`

*Títol: Disseny i implementació web per a una escola d'arts
marcial*

Volum:

Alumne: Raül Borrachero Sánchez

Director/Ponent: Pau Fonseca i Casas

Departament: Estadística i Investigació Operativa

Data: 07/01/2009

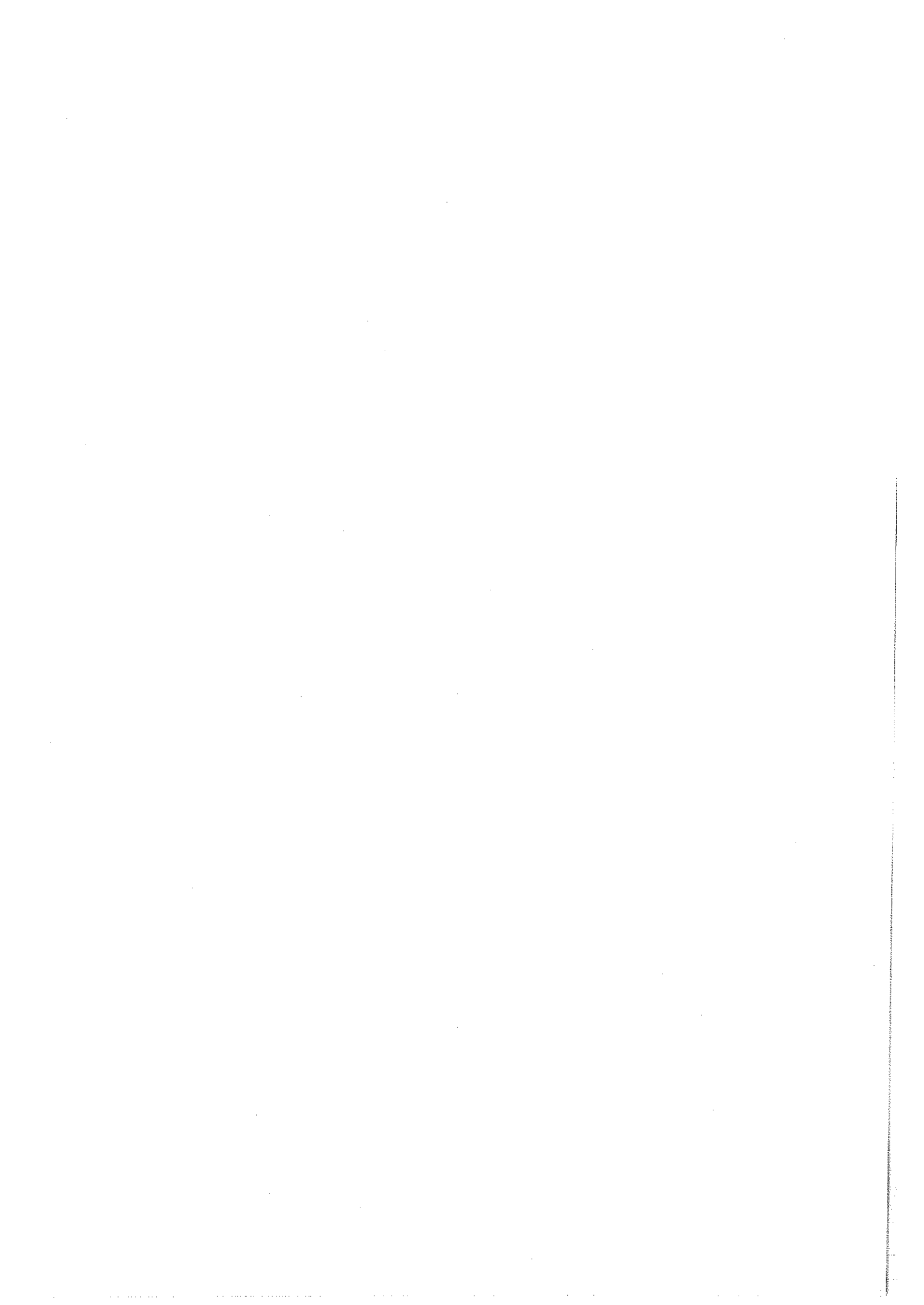
GESTIÓ ACADÈMICA-FIB
ENTRADA
12 ENE. 2009

ÍNDICE

1. INTRODUCCIÓN	<i>pag. 8</i>
2. DIMENSIONADO	
2.1. Definición de objetivos	<i>pag. 9</i>
2.2. Tecnologías usadas	<i>pag. 10</i>
2.3. Costes y desarrollo	
2.3.1. <i>Desarrollo</i>	<i>pag. 14</i>
2.3.2. <i>Costes</i>	<i>pag. 18</i>
2.4. Drupal	
2.4.1 <i>¿Porqué Drupal?</i>	<i>pag. 22</i>
2.4.2 <i>Estructura y funcionamiento</i>	<i>pag. 23</i>
3. EL PORTAL WEB	
3.1. INSTALACIÓN	
3.1.1. <i>Archivos de Drupal</i>	<i>pag. 30</i>
3.1.2. <i>La base de datos</i>	<i>pag. 30</i>
3.1.3. <i>ffmpeg y JW Player</i>	<i>pag. 34</i>
3.2. SERVICIOS	
3.2.1. <i>General</i>	<i>pag. 36</i>
3.2.2. <i>Alumno</i>	<i>pag. 37</i>
3.2.3. <i>Entrenador</i>	<i>pag. 40</i>
<i>Vídeo y archivos adjuntos</i>	<i>pag. 41</i>
3.2.4. <i>Administrador</i>	<i>pag. 44</i>
<i>Construcción del sitio</i>	<i>pag. 46</i>
<i>Alias</i>	<i>pag. 46</i>
<i>Bloques</i>	<i>pag. 46</i>
<i>Disparadores</i>	<i>pag. 47</i>
<i>Formulario de contacto</i>	<i>pag. 47</i>
<i>Menús</i>	<i>pag. 48</i>
<i>Módulos</i>	<i>pag. 48</i>
<i>Temas</i>	<i>pag. 51</i>
<i>Traducción del interfaz</i>	<i>pag. 52</i>
<i>Vistas</i>	<i>pag. 53</i>
<i>Gestión de Contenido</i>	<i>pag. 56</i>
<i>Agregador de Feeds</i>	<i>pag. 56</i>
<i>Comentarios</i>	<i>pag. 57</i>
<i>Contenido</i>	<i>pag. 57</i>
<i>Galerías de imágenes</i>	<i>pag. 57</i>
<i>Importar imágenes</i>	<i>pag. 57</i>
<i>Opciones de publicación</i>	<i>pag. 57</i>
<i>Publicación RSS</i>	<i>pag. 58</i>
<i>Taxonomías</i>	<i>pag. 58</i>
<i>Tipos de contenido</i>	<i>pag. 60</i>
<i>Gestión de Usuarios</i>	<i>pag. 62</i>
<i>Configuración de usuarios</i>	<i>pag. 62</i>
<i>Perfiles</i>	<i>pag. 62</i>
<i>Permisos</i>	<i>pag. 63</i>
<i>Reglas de accesos</i>	<i>pag. 63</i>
<i>Roles</i>	<i>pag. 63</i>
<i>Usuarios</i>	<i>pag. 63</i>
<i>Configuración del Sitio</i>	<i>pag. 65</i>
<i>Acciones</i>	<i>pag. 65</i>
<i>Eventos</i>	<i>pag. 65</i>

<i>Fecha y hora</i>	<i>pag. 65</i>
<i>Formatos de entrada</i>	<i>pag. 65</i>
<i>Idiomas</i>	<i>pag. 67</i>
<i>Imágenes</i>	<i>pag. 68</i>
<i>Información del sitio</i>	<i>pag. 68</i>
<i>Informe de errores</i>	<i>pag. 68</i>
<i>Juegos de herramientas de imágenes</i>	<i>pag. 68</i>
<i>Mantenimiento del sitio</i>	<i>pag. 68</i>
<i>Preferencias de Flashvideo</i>	<i>pag. 68</i>
<i>Preferencias de búsqueda</i>	<i>pag. 71</i>
<i>Registros y alertas</i>	<i>pag. 71</i>
<i>Rendimiento</i>	<i>pag. 71</i>
<i>Sistema de ficheros</i>	<i>pag. 72</i>
<i>Subida de ficheros</i>	<i>pag. 72</i>
<i>Tema de administración</i>	<i>pag. 72</i>
<i>URLs limpias</i>	<i>pag. 72</i>
<i>Grupos</i>	<i>pag. 73</i>
<i>Informes</i>	<i>pag. 73</i>
<i>Ayuda</i>	<i>pag. 75</i>
<i>Actualizaciones de drupal 6</i>	<i>pag. 76</i>
3.2.5. Ficheros de Pulsómetro	<i>pag. 77</i>
4. AMPLIACIONES	<i>pag. 90</i>
5. CONCLUSIONES	<i>pag. 92</i>
6. BIBLIOGRAFIA	<i>pag. 94</i>
7. AGRADECIMIENTOS	<i>pag. 96</i>
ANEXO I: NetBeans 6.5 y Drupal 6	<i>pag. 97</i>
ANEXO II: Introducción al desarrollo de aplicaciones en Drupal 6	<i>pag. 100</i>
ANEXO III: Instalación de FFMPEG en una Debian Etch (archivos binarios)	<i>pag. 108</i>
Índice de figuras.	
Figura 2-1. Descripción de tareas correspondiente al informe previo.	Pag. 14
Figura 2-2. Descripción de tareas correspondiente la finalización del proyecto.	Pag. 14
Figura 2-3. Diagrama de gantt correspondiente al informe preliminar.	Pag. 15
Figura 2-2. Descripción de tareas correspondiente la finalización del proyecto.	Pag. 15
Figura 2-5. Dificultad de un proyecto en función de sus líneas de código.	Pag. 21
Figura 2-6. Coeficientes COCOMO	Pag. 21
Figura 2-7. Capas en el diseño de Drupal	Pag. 23
Figura 2-8. Una vista general del núcleo de Drupal (Parcial).	Pag. 24
Figura 2-9. Al activar módulos adicionales, se añaden más funcionalidades.1	Pag. 25
Figura 2-10. La estructura de carpetas en una instalación de Drupal por defecto.	Pag. 27
Figure 2-11. El directorio sites permite guardas todas las modificaciones hechas a Drupal.	Pag. 28
Figura 3-1. Settings.php	Pag. 30
Figura 3-2. Crear base de datos <i>drupal_bd</i>.	Pag. 31
Figura 3-3. seleccionar base de datos <i>drupal_bd</i>.	Pag. 31
Figura 3-4. Importar base de datos. Paso 1.	Pag. 32
Figura 3-4. Importar base de datos. Paso 2.	Pag. 32
Figura 3-6. Menú de Sistema de ficheros.	Pag. 33
Figura 3-7. Opciones de Flashvideo	Pag. 35
Figura 3-8. Path a los archivos de FFMPEG.	Pag. 35
Figura 3-9. Página inicial del portal.	Pag. 37
Figura 3-10. Pagina de inicio de un alumno.	Pag. 38
Figura 3-11. Muestra de eventos	Pag. 38

Figura 3-12. Página de búsquedas sencilla y avanzada.	Pag. 39
Figura 3-13. Página de listados de logs (logs→ listado logs).	Pag. 39
Figura 3-13(bis). Página de subida de ficheros de pulsómetro (logs→subir logs).	Pag. 39
Figura 3-14. Menú de grupos de trabajo.	Pag. 39
Figura 3-15. Dentro de un grupo.	Pag. 40
Figura 3-16. Perfil de usuario.	Pag. 40
Figura 3-17. Listado de páginas publicadas por grupo, no leídas y general.	Pag. 41
Figura 3-18. Tipos de contenido	Pag. 41
Figura 3-19. Archivos adjuntos.	Pag. 42
Figura 3-20. Opciones de archivos adjuntos.	Pag. 42
Figura 3-21. Título y Cuerpo	Pag. 42
Figura 3-22. Opciones de publicación en grupos de trabajo.	Pag. 43
Figura 3-23. Imágenes adjuntas.	Pag. 43
Figura 3-24. Opciones de comentarios.	Pag. 43
Figura 3-25. Opciones de ruta URL.	Pag. 43
Figura 3-26. Opciones de Evento.	Pag. 44
Figura 3-27. Opciones de la creación de un grupo de trabajo.	Pag. 44
Figura 3-28. Formato de entrada y opciones de grupo.	Pag. 45
Figura 2-29. Información de revisión	Pag. 45
Figura 3-30. Opciones de auditoria y de publicación.	Pag. 45
Figura 3-31. Opciones de tema y privacidad.	Pag. 45
Figura 3-32. Página de administración de bloques.	Pag. 45
Figura 3-33. Configuración de Módulos.	Pag. 49
Figura 3-34. Códulo CCK (Content Construction Kit).	Pag. 49
Figura 3-35. Módulos opcionales básicos de Drupal (1)	Pag. 49
Figura 3-36. Módulos opcionales básicos de Drupal (2)	Pag. 49
Figura 3-37. Módulos básicos de Drupal y módulos de tiempo y fecha.	Pag. 49
Figura 3-38. Módulos de video e imagen.	Pag. 49
Figura 3-39. Módulos de grupos.	Pag. 50
Figura 3-40. Módulos de Vistas (Views) y otros módulos.	Pag. 50
Figura 3-41. Menú de configuración de temas.	Pag. 52
Figura 3-42. Personalización de los colores de un tema.	Pag. 52
Figura 3-43. Traducción del interfaz.	Pag. 52
Figura 3-44. Opciones para listar cadenas traducidas.	Pag. 52
Figura 3-45. Lista de cadenas traducidas.	Pag. 53
Figura 3-46. Formulario para la traducción de cadenas.	Pag. 53
Figura 3-47. Página de gestión de vistas.	Pag. 54
Figura 3-48. Edición de una vista.	Pag. 54
Figura 3-49. Resultado de la vista.	Pag. 56
Figura 3-50. Listado de usuario con el rol de alumno (vista de administrador)	Pag. 56
Figura 3-51. Listado de usuario con el rol de alumno (vista del resto de roles)	Pag. 56
Figura 3-52. Taxonomías	Pag. 59
Figura 3-53. Terminos de un vocabulario.	Pag. 59
Figura 3-54. Edición de un vocabulario.	Pag. 59
Figura 3-55. Edición de un término.	Pag. 59
Figura 3-56. Tipos de contenido.	Pag. 59
Figura 3-57. Tipos de contenido	Pag. 60
Figura 3-57. Identificación y opciones sobre la publicación.	Pag. 60
Figura 3-58. Opciones del flujo de trabajo.	Pag. 61
Figura 3-59. Grupos.	Pag. 61
Figura 3-60. Opciones de comentarios.	Pag. 61
Figura 3-61. Eventos de calendario.	Pag. 61
Figura 3-62. Imágenes adjuntas de image.	Pag. 61
Figura 3-63. Administración de perfiles.	Pag. 63
Figura 3-64. Permisos por roles.	Pag. 64
Figura 3-65. Administración de usuarios.	Pag. 65
Figura 3-66. Edición de usuario.	Pag. 65
Figura 3-67. Añadir un nuevo usuario.	Pag. 65
Figura 3-68. Lista de idiomas.	Pag. 67
Figura 3-69. Añadir un nuevo idioma.	Pag. 67
Figura 3-70. Opciones de presentación de contenido por idioma.	Pag. 68
Figura 3-71. Opciones de Imágenes.	Pag. 68
Figura 3-72. Preferencias de Flashvideo.	Pag. 70
Figura 3-73. Opciones de configuración.	Pag. 70
Figura 3-74. update.php	Pag. 76
Figura Anexo I-1. Versioning→CVS→Cehckout.	Pag. 98
Figura Anexo I-2. Tools Plugin Downloaded.	Pag. 98
Figura Anexo I-3. Creación de un nuevo Proyecto.	Pag. 99
Figura Anexo I-4. Crear conexión con la base de datos.	Pag. 99
Figura Anexo I-5. Instalación de Drupal.	Pag. 99
Figura Anexo I-6. Creación de módulo o tema personalizado.	Pag. 99



1. INTRODUCCIÓN

Hoy en día, dentro del mundo del deporte profesional es necesaria una gran preparación física y mental para obtener buenos resultados en competiciones deportivas. Para ello, los profesionales de dicho sector, deportistas y preparadores físicos, deben elaborar juntos planes de entrenamiento para cada disciplina y individuo, o en su defecto, grupo de personas, con el cual poder desarrollar una estrategia que les permita afrontar los retos con los mejores resultados posibles y conseguir los objetivos propuestos desde un inicio.

La estrategia de entrenamiento por su parte, necesita un buen plan de entrenamiento que haga posible su buen término, y es por ello que esencialmente son necesarias un conjunto de medios, procedimientos y conocimientos tanto técnicos como personales para que el plan sea eficiente. Dentro de los conocimientos la utilización de herramientas estadísticas es primordial para la buena planificación, ya que permite realizar un estudio personalizado de cada deportista individualmente y/o en conjunto, para más tarde basarse en los resultados de dichos procedimientos y modelos estadísticos a la hora de tomar decisiones sobre cómo el plan de entrenamiento debe encauzarse y prever la evolución de dicho deportista durante toda su duración.

Las herramientas y modelos estadísticos que se pueden usar para confeccionar un plan de entrenamiento no están limitados a una sola disciplina deportiva, ni siquiera dentro del ámbito del deporte profesional. Un buen ejemplo de ello son la mayoría de estilos de artes marciales no considerados como deporte en sí, que poseen diversas modalidades de competición. Los artistas marciales que participan en dichas competiciones han de seguir unas estrategias y planes de entrenamiento para obtener buenos resultados y para ello es perfectamente posible aplicar los métodos de estudio y confección de planes de entrenamiento de cualquier otra modalidad considerada como deporte de competición, si bien hay que tener más factores en cuenta en la fase de diseño.

Actualmente, cada vez está más diversificada la utilización de este tipo de prácticas, con lo cual hace más difícil la recaptación de los datos necesarios llevar a cabo el diseño de los planes de entrenamiento y la toma de decisiones, dada la dispersión de los alumnos y profesores. Cada día se hace más necesaria una manera de simplificar los trámites y permitir más autonomía a cada entrenador y a cada alumno, y les permita concentrarse en otras tareas más importantes que la manera en cómo captar los datos necesarios.

DISSENY I IMPLEMENTACIÓ WEB PER A UNA ESCOLA D'ARTS MARCIALS pretende poner al alcance de todo el mundo una solución sencilla y potente a la hora de mantener la comunidad de entrenamiento de una escuela de artes marciales. Mediante una portal web que permite la centralización de los datos a la vez que proporciona a la comunidad una manera de coordinar los eventos, las prácticas del arte marcial en cuestión facilitando información y supervisión a cada alumno por parte de los instructores, así como asentar las bases para la posibilidad del uso de herramientas estadísticas en un futuro para el diseño de los planes de entrenamiento, es posible dar a conocer la escuela de artes marciales dentro de la comunidad internauta y gestionar los procesos internos de dicha escuela. Dos puntos esenciales de cara al buen funcionamiento de cualquier disciplina de competición.

2. DIMENSIONADO

2.1. Definición de objetivos

Como en toda práctica de disciplinas en las cuales se enseñe y se aprenda algo, hay una comunidad de la cual forman parte alumnos y profesores y en la cual es necesaria una gran interacción entre sus miembros para que las enseñanzas fructifiquen y el aprendizaje se pueda llevar a cabo de manera satisfactoria. Aparte de una herramienta para facilitar la enseñanza y el aprendizaje de las actividades que una asociación de artes marciales necesita, también es positivo tener un vehículo con el cual poder comunicarse con futuros alumnos y gente interesada en las actividades de la asociación, así como mantener contacto con otros grupos que practiquen estas disciplinas. Para ello, hemos creído conveniente la utilización de un vehículo al alcance de todos: Internet.

La red es un recurso tremendamente útil en el cual podemos disponer de una gran cantidad de herramientas para nuestras necesidades, y ahora más que nunca gracias a las nuevas tecnologías de Web 2.0. Gracias a estas tecnologías se hemos pensado en un portal donde los alumnos, profesores y gente interesada en lo que una asociación de artes marciales pueda aportarles, puedan acceder en cualquier momento y consultar los datos que les hagan falta, ponerse en contacto con otros miembros, acceder a material de la escuela, conocer los eventos de la asociación y muchas otras cosas.

Desafortunadamente, muchos de los procesos a realizar para implantar y/o mantener este servicio a través de internet requieren ciertos conocimientos y recursos que hoy en día no poseen la mayoría del público al cual va dirigido este portal y por ello hemos pensado en una herramienta sencilla de implantar, sencilla de mantener y con un coste muy reducido para el cliente, en este caso, una asociación de artes marciales.

La mayor parte de las asociaciones de artes marciales acostumbra a ser recelosas sobre los alumnos a los cuales aceptan como discípulos, a veces por un secretismo implantado hace mucho tiempo por antiguos maestros los cuales veían sus enseñanzas como un legado a modo de herencia para unos pocos, y la mayor parte de ellas, por motivos de seguridad hacia los alumnos, ya que sin una preparación física y mental adecuada, es posible hacerse daño o hacer daño a terceras personas. También es necesario apuntar que la principal fuente de ingresos económicos de muchos maestros proviene de la enseñanza de estas prácticas. Por estos motivos, hemos creído esencial que el portal ofrezca una parte privada donde maestros y alumnos puedan publicar material y consultarlo siempre que quieran, y sólo ellos puedan hacerlo.

Aparte de los servicios que una página web normal pueda proporcionar, también hemos creído oportuna la integración de un servicio de hosting de vídeo propio en el portal teniendo en cuenta los motivos comentados en el anterior punto y el hecho que todas las artes marciales poseen muchas formas de ejercicio repetitivas que constan de una serie de movimientos encadenados, las cuales son mucho más comprensibles si se muestra su ejecución entera que paso a paso. Con ello, los maestros podrán subir vía navegador los vídeos que crean oportunos en una amplia variedad de formatos, e incrustarlos en la publicación que deseen añadir a la página.

Dado que muchas de las asociaciones de artes marciales compiten en torneos y eventos similares, es necesario un buen entrenamiento y por ello es importante que se añada a su estrategia de entrenamiento el diseño de un plan. Para diseñar un plan de entrenamiento eficiente en competición, como ya hemos comentado anteriormente, es imprescindible el uso de herramientas estadísticas y por ello el portal dispone de la integración de unos servicios para facilitar la captación de datos y la toma de decisiones. A través de un pulsómetro se pueden tomar los datos de una sesión de entrenamiento y mediante el portal, subir a un servidor central los datos recabados de esta manera a través de un archivo que sería procesado posteriormente. Este servicio de subida de ficheros es personal para cada miembro, así es más fácil llevar a cabo

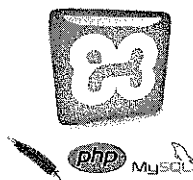
estudios personalizados para cada individuo. También sienta las bases para el desarrollo de una integración de dicho portal con paquete de software estadístico para llevar a cabo dichos estudios estadísticos y obtener la información que deriva de ellos, crucial en la toma de decisiones. Dicha integración se ha llevado a cabo con éxito en otro proyecto paralelo pero dado que dicho portal es una herramienta compleja para elaborar de este tipo de aplicaciones, se ha preferido elaborar una guía de programación y configuración de las herramientas necesarias para el desarrollo dentro del portal en cuestión, ya que no ha sido posible englobar la creación de dicha integración en el presente proyecto.

DISSENY I IMPLEMENTACIÓ WEB PER A UNA ESCOLA D'ARTS MARCIALS es un proyecto dentro de un marco de investigación mucho más amplio en estudios estadísticos aplicados a competición, y tiene como fin la facilitación de la recaptación de datos de deportistas tanto marciales como tradicionales. Incorpora otras herramientas desarrolladas por otros proyectistas las cuales son utilizadas en conjunto, como la integración mencionada en el punto anterior.

2.2. Tecnologías usadas

Para implementar todas las funcionalidades descritas en los objetivos del proyecto, han sido necesarias un conjunto de tecnologías específicas que funcionen conjuntamente. Ya que uno de los puntos clave dentro del diseño de objetivos es que fuera lo más barato y lo menos restrictivo en cuanto a licencias de software se refiere a la hora de hacer ampliaciones en un futuro, se han utilizado una serie de aplicaciones y sistemas dentro del movimiento Open Source y licencias de Creative Commons que hagan posible la libre distribución y posteriores modificaciones sin los problemas legales ni económicos que un software propietario conlleva.

He aquí una breve explicación de cada uno de las tecnologías usadas:



XAMPP: Es un servidor independiente de plataforma, software libre, que consiste principalmente en la base de datos MySQL, el servidor Web Apache y los intérpretes para lenguajes de script: PHP y Perl. El nombre proviene del acrónimo de X (para cualquiera de los diferentes sistemas operativos), Apache, MySQL, PHP, Perl. El programa está liberado bajo la licencia GNU y actúa como un servidor Web libre, fácil de usar y capaz de interpretar páginas dinámicas. Actualmente XAMPP esta disponible para Microsoft Windows, GNU/Linux, Solaris, y MacOS X.

Solamente requiere descargar y ejecutar un archivo zip, tar, o exe, con unas pequeñas configuraciones en alguno de sus componentes que el servidor Web necesitará. XAMPP se actualiza regularmente para incorporar las últimas versiones de Apache/MySQL/PHP y Perl. También incluye otros módulos como OpenSSL y phpMyAdmin. Para instalar XAMPP se requiere solamente una pequeña fracción del tiempo necesario para descargar y configurar los programas por separado.

En este proyecto se ha decantado por XAMPP como herramienta principal para el servicio de servidor web y bases de datos dado que es uno de los motores del proyecto de investigación en el cual está englobado el nuestro, además de ser una herramienta fácil de usar y sencilla de

instalar. Aunque el servidor XAMPP sobre el cual se hayan hecho la mayor parte de pruebas corra sobre un SO Microsoft Windows, también se ha testeado en un servidor LAMP (Linux, Apache, MySQL, Php) bajo una Deviant Etch 3.02.1, usando las mismas aplicaciones auxiliares que se han usado en Windows con los mismo resultados positivos.

Detallemos un poco más en que consisten cada uno de los servicios de XAMPP:

Apache: Es un servidor HTTP de código abierto para varias plataformas (entre ellas Unix, Windows y Macintosh) que implementa el protocolo HTTP/1.1 y la noción de sitio virtual. Entre otras características cabe destacar sus altamente configurables mensajes de error, bases de datos de autenticación y negocio de contenido. Tiene una amplia aceptación en la red: desde 1996, Apache, es el servidor http más usado. Las ventajas de apache respecto a otros sistemas que ofrezcan el mismo tipo de servicios es que es altamente modular y extensible (lo cual implica que es altamente configurable y adaptable a nuestras necesidades), Open Source, multiplataforma y popular, con lo cual podemos encontrar grandes cantidades de tutoriales de configuración y uso y abundante ayuda a los problemas que puedan se presentar.

MySQL: Se trata de un sistema de gestión de bases de datos relacional, multithread y multiusuario basado en el lenguaje de bases de datos SQL. MySQL posee dos tipos de licencias: por un lado se ofrece bajo la GNU GPL para cualquier uso compatible con dicha licencia, y una licencia para la inclusión en productos comerciales privados. MySQL es propiedad y está patrocinado por MySQL AB, subsidiaria de Sun Microsystems, lo cual posibilita es esquema dual anteriormente mencionado. Es muy utilizado en aplicaciones web y está fuertemente ligado a PHP dada su popular aplicación web en conjunto.

MySQL es una base de datos muy rápida en la lectura para entornos de baja concurrencia en la modificación de los datos, lo cual en aplicaciones web, ya que se da el caso mencionado y el entorno es intensivo en la lectura de datos, hace de MySQL una solución ideal para este tipo de aplicaciones.

En nuestro proyecto se usan varias bases de datos para gestionar tanto la información de la escuela como para administrar la información del sistema que se encarga de gestionar la página web.

PHPMyAdmin: Se trata de una herramienta escrita en PHP con la intención de manejar la administración de un servidor y/o bases de datos en MySQL a través de páginas web utilizando Internet. Actualmente, puede crear y eliminar bases de datos; crear, eliminar y alterar tablas; borrar, editar y añadir campos en tablas, ejecutar cualquier sentencia SQL, administrar claves en campos, administrar privilegios, exportar datos en varios formatos. Está disponible en 50 idiomas bajo una licencia GPL.

Es un proyecto tremendamente popular en la red internauta y por ello es fácil encontrar ayuda y documentación y ya que es una aplicación web, es posible ejecutarla a través de una serie de medidas de seguridad, en cualquier lugar del que se disponga de conexión, convirtiéndola en una herramienta extremadamente útil. Su sencillez en el manejo y su eficiencia lo hacen ideal para usuarios de todo tipo y es esencial en este proyecto a la hora de instalar y gestionar las bases de datos necesarias para su correcto funcionamiento.

PHP: (siglas de PHP Hypertext Pre-processo) Es un lenguaje de programación interpretado diseñado originalmente para la creación de páginas web dinámicas. Está pensado principalmente para ser evaluado en la parte del servidor, transmitiendo la información resultante al cliente a través de código html mayormente. También se puede usar como interfaz de línea de comando o en la creación de otros programas.

PHP, Publicado bajo la PHP License y siendo considerada esta licencia como software libre, es un lenguaje interpretado de propósito general ampliamente usado, que está diseñado especialmente

para desarrollo web y puede ser embebido dentro de código HTML. Generalmente se ejecuta en un servidor web, tomando el código en PHP como su entrada y creando páginas web como salida. Puede ser desplegado en la mayoría de los servidores web y en casi todos los sistemas operativos y plataformas sin costo alguno. PHP se encuentra instalado en más de 20 millones de sitios web y en un millón de servidores, aunque el número de sitios en PHP ha compartido algo de su preponderante sitio con otros nuevos lenguajes no tan poderosos desde agosto de 2005.

Dentro del marco del proyecto es quizás la herramienta más usada y extendida, ya que la mayor parte de las aplicaciones usadas utilizan PHP para su desarrollo. El portal web está programado a través de php y varias aplicaciones dentro del mismo desarrolladas íntegramente en este proyecto, también están escritas en PHP.

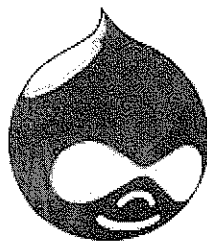
JW Player: Es un reproductor flash de video creado por Jeroen Wijering bajo una licencia Creative Commons, la cual permite modificar y distribuir libremente el software siempre y cuando no sea con fines lucrativos y se mencione la autoría de dicho software. Permite la reproducción de varios formatos de video incluidos cualquier formato que el reproductor flash de Adobe pueda. Es uno de los reproductores multimedia más populares en Internet, motivo por el que es más fácil encontrar soporte técnico. Está diseñado para ser usado principalmente en servicios web, lo cual hace de él una herramienta necesaria para nuestro portal web. Para ello, utilizamos una pequeña parte del reproductor para reproducir los videos de la escuela como lo haría cualquier comunidad o portal de hosting de video (como YouTube), haciendo de esta manera que sea más complicada la obtención del material de la escuela y su distribución.



FFMPEG: Es una colección de software libre que puede grabar, convertir y hacer streaming de audio y vídeo. Incluye una biblioteca de códecs. FFmpeg está desarrollado en Linux, pero puede ser compilado en la mayoría de los sistemas operativos, incluyendo Windows. Es destacable que la mayoría de los desarrolladores de FFmpeg lo sean también del proyecto MPlayer, más un miembro del proyecto Xine y que FFmpeg esté hospedado en el servidor del proyecto MPlayer.

FFMPEG posee una licencia que ha generado cierta controversia internacionalmente ya que en algunos países es ilegal su uso. En dichos países su política de actuación vulnera algunas patentes de software que utilizan los mismos procedimientos, dado que considera la ingeniería inversa como un método fundamental de operación en su comunidad de desarrollo, por ello muchas distribuciones de sistemas operativos de software libre no la incluyen en sus distribuciones iniciales. Se considera lícito su uso siempre y cuando se utilice en proyectos que empleen una licencia GPL.

En nuestro caso es usado porque partes del código del portal web en cuestión, utilizan dicho software, y dado que es un producto muy extendido, es fácil encontrar soporte técnico y ayuda a través de la red, además de no tener coste monetario alguno.



Drupal: Es un CMS (Content Management System), sistema de gestión de contenido para sitios Web. Permite publicar artículos, imágenes, u otros archivos y servicios añadidos como foros, encuestas, votaciones, blogs y administración de usuarios y permisos. Drupal es un sistema dinámico: en lugar de almacenar sus contenidos en archivos estáticos en el sistema de ficheros del servidor de forma fija, el contenido textual de las páginas y otras configuraciones son almacenados en una base de datos y se editan utilizando un entorno Web incluido en el producto.

Drupal fue originalmente escrito por Dries Buytaert y a pesar de que empezó como un pequeño BBS, ha llegado a ser mucho más que sólo un portal de noticias gracias a su arquitectura flexible. Drupal se compone de una infraestructura base y un conjunto de módulos que ofrecen un amplio conjunto de funciones, incluyendo sistemas de comercio electrónico, galerías de fotos, administración de listas de correo electrónico, e integración de CVS. Es posible añadir módulos de terceros para modificar el comportamiento de Drupal u ofrecer nuevas funciones. Drupal se usa, entre otros, en intranets de compañías, enseñanza en línea, comunidades de arte y administración de proyectos. Muchos piensan que la relevancia de Drupal en las comunidades de usuarios es lo que lo hace destacarse de la competencia.

Drupal es un proyecto de software libre bajo una licencia GPL esencial para el desarrollo del portal web que nos ocupa. Gracias a su enorme y nutrida comunidad de desarrollo, es posible encontrar cantidades ingentes de contribuciones, tutoriales, guías de programación y soporte técnico en línea.

2.3. Costes y desarrollo

2.3.1. Desarrollo

El desarrollo de este proyecto consta de varias partes, las cuales se han ido llevando a cabo según un calendario marcado por los objetivos. Cabe destacar que aunque se haya variado el número de horas a realizar, las tareas desempeñadas así como su orden, ha habido poca variación desde la descripción de las tareas en cuanto a horas se refiere.

He aquí la descripción de las tareas así como su programación según un informe de desarrollo de proyecto presentado el 18/10/2008 y lo mismo una vez finalizado. Es necesario apuntar que los tiempos están expresados en días y se ha calculado unas 4 horas por día de trabajo. El total expresa el número de días requeridos desde el inicio del proyecto hasta su final. Nótese que varias tareas forman parte de otras y algunas se han llevado a cabo simultáneamente.

NUM	TAREA	PRECEDE	INICIO	FIN	PREVISTO	REAL	FINALIZADO
1	Especificación del proyecto	-	02/06/2008	17/06/2008	15	15	SI
2	Diseño de la página web	Especificación del proyecto	23/06/2008	08/07/2008	15	15	SI
3	Documentación y aprendizaje	Especificación del proyecto	23/06/2008	12/12/2008	100	80	NO
4	Implementación página web	Especificación del proyecto	12/07/2008	19/12/2008	160	100	NO
5	Implementación via Drupal	Diseño de la página web	14/07/2008	12/09/2008	60	60	SI
6	Testing Drupal	Implementación via Drupal	15/09/2008	24/10/2008	39	35	NO
7	Diseño comunicación entre Drupal y R-php	Implementación via Drupal	19/10/2008	24/10/2008	5	1	NO
8	Implementació comunicació entre Drupal i R-php	Diseño comunicación entre Drupal y R-php	25/10/2008	14/11/2008	30	0	NO
9	Migración a de Drupal 5 a Drupal 6	Diseño comunicación entre Drupal y R-php	13/11/2008	28/11/2008	15	0	NO
10	Mejoras de seguridad en el portal web	Testing Drupal	29/10/2008	28/11/2008	30	0	NO
11	Testing del proyecto	Implementación via Drupal	28/11/2008	18/12/2008	20	0	NO
12	Elaboración memoria del proyecto	Testing del proyecto	20/11/2008	25/12/2008	25	0	NO
TOTAL					206	139	

Figura 2-1. Descripción de tareas correspondiente al informe previo.

NUM	TAREA	PRECEDE	INICIO	FIN	T.PREVISTO	T.REAL
1	Especificación del proyecto	-	02/06/2008	17/06/2008	15	15
2	Diseño de la página web	Especificación del proyecto	23/06/2008	08/07/2008	15	15
3	Documentación y aprendizaje	Especificación del proyecto	23/06/2008	12/12/2008	100	100
4	Implementación página web	Especificación del proyecto	12/07/2008	02/1/2009	160	174
5	Implementación via Drupal	Diseño de la página web	14/07/2008	12/09/2008	60	60
6	Testing Drupal	Implementación via Drupal	15/09/2008	24/10/2008	39	39
7	Migración a de Drupal 5 a Drupal 6	Implementación via Drupal	25/10/2008	06/11/2008	15	12
8	Mejoras de seguridad en el portal web	Testing Drupal	06/11/2008	13/11/2008	30	7
9	Solución a problemas en la base de datos	-	13/11/2008	20/12/2008	0	37
10	Elaboración memoria del proyecto	Mejoras Seguridad, Testing proyecto	08/12/2008	02/1/2009	5	25
11	Testing del proyecto	Implementació via Drupal	28/11/2008	05/1/2009	20	38
TOTAL					206	217

Figura 2-2. Descripción de tareas correspondiente la finalización del proyecto.

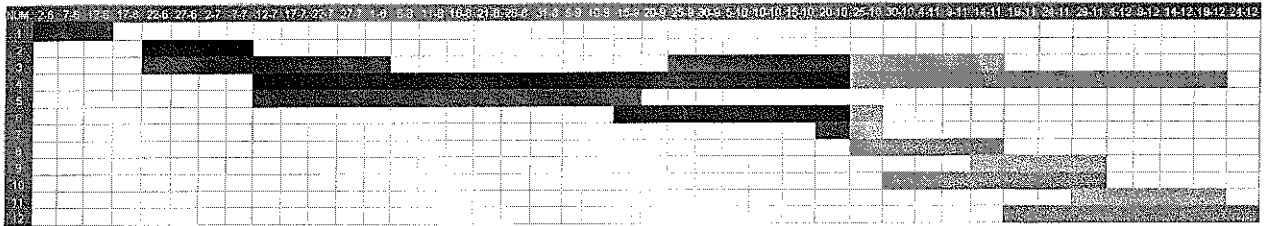


Figura 2-3. Diagrama de gantt correspondiente al informe preliminar.

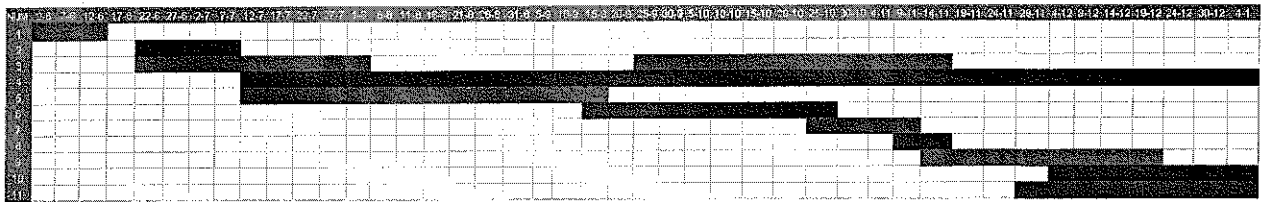


Figura 2-4. Diagrama de gantt correspondiente a la finalización del proyecto.

Como se puede observar, la diferencia entre ambos diagramas en poca y la mayor parte de los cambios se tratan de reestructuraciones debidas a un único problema. Seguidamente se detalla en qué han consistido cada una de las tareas realizadas en la conclusión del proyecto.

Especificación del proyecto: En esta parte se han especificado los requerimientos del proyecto, qué tecnologías se debían usar mencionadas anteriormente en el apartado de tecnologías usadas y las funcionalidades a desarrollar según las exigencias del mismo. Se ha hecho un análisis exhaustivo del tipo de servicios que un portal web de estas características necesita y/o puede llegar a necesitar. Se pensó en integrar un motor 3D en lugar del servicio de video flash en primera instancia, pero después de un análisis en más profundidad de los requisitos y las tecnologías a desarrollar en este apartado, se descartó dado su elevada complejidad. También se decidió entre varios sistemas y lenguajes de programación para llevar a cabo el portal web, entre ellos java, perl y php, descartando los dos primeros a favor de un CMS y se incluyó el tercero tras sopesar los diferentes CMS disponibles que satisficieran los requisitos de la idea original y decantarse por Drupal, del cual se describirá con mucha más amplitud en posteriores capítulos de esta memoria, al estar enteramente escrito en este lenguaje.

En referencia al software estadístico, se tuvo muy claro desde un buen principio que se usaría R dada su robustez y fiabilidad en el caso de integrar servicios de estudios estadísticos. Entonces surgió el problema de integrar R en el portal web en cuestión. En primera opción se sugirió usar R-web, un software de código abierto desarrollado en la por Jeff Banfield del departamento de ciencias matemáticas de la universidad de Washington en Arkansas, Estados Unidos; pero dado que estaba escrito en perl y al encontrar R-php, se decantó por este último al no haber que interpretar ningún tipo de código entre el portal y dicho servidor.

Diseño de la página web: Aquí se ha profundizado en la estructura del portal web en sí. Está estructurada en dos partes diferenciadas: la parte pública, la cual tiene la función de dar a conocer la asociación a través de secciones de información general y específica añadida por cada una de las escuelas que usen el producto, y una parte privada en la cual se desarrollen todas las actividades internas que sean necesarias y solamente accesible a través de un sistema de cuentas de usuario y password que garantice la privacidad de alumnos y profesores. También se ha contemplado el hecho de que los alumnos se puedan organizar en grupos de trabajo o clases dentro de la escuela y ello se ha traducido al portal, siendo posible que los solos los alumnos

pertenecientes a determinados grupos puedan acceder a la información que en ellos se ofrezca (los motivos pueden ser los diferentes niveles de aprendizaje dentro de una escuela de artes marciales, por ejemplo). En el capítulo destinado al portal web se describen todas sus funcionalidades de manera mucho más detallada.

Documentación y aprendizaje: Dentro del funcionamiento de DISSENY I IMPLEMENTACIÓ WEB PER A UNA ESCOLA D'ARTS MARCIALS se engloban una gran cantidad de tecnologías de las cuales se ha consultado abundante material informativo para su correcto uso. Entre ellos podemos englobar la API de Drupal, los manuales de instalación del propio CMS, su estructura y funcionamiento internos, abundantes tutoriales y guías de programación, documentación sobre ffmpeg y los problemas causados en su implantación en las diferentes distribuciones de varios sistemas operativos como son pueden ser Linux o Windows, manuales de programación de php, solución de errores y guías de SQL y un largo etcétera.

Personalmente he encontrado esta parte una de las más áridas del proyecto, dado que partía con prácticamente nulos conocimientos del funcionamiento de muchas de las tecnologías utilizadas como pueden ser php, Drupal, ffmpeg o MySQL y más bien escasos en otras como Apache, html o el paquete XAMPP. Todo ello ha supuesto una parte nada despreciable en los esfuerzos por conseguir el buen funcionamiento del portal.

Implementación página web: Todos los puntos anteriormente descritos en la en diseño del portal se han llevado a cabo en esta parte. Las tareas van desde desarrollar la página en el CMS y sus funcionalidades, hasta el diseño de algunas partes de la base de datos y la comunicación de servicios entre del propio CMS y otras aplicaciones. La mayor parte de estas interrelaciones se describirán con más detalle en la sección destinada al portal web.

Implementació via Drupal: Drupal posee la funcionalidad de ser tremendamente ampliable. Para llevar a cabo ciertas funcionalidades presentes en los objetivos del proyecto han hecho falta la adición de los llamados módulos, que implementan las funcionalidades que no posee la instalación básica del CMS. La se ha efectuado una búsqueda exhaustiva de módulos que mejor satisficieran las necesidades del proyecto y se han usado los módulos CCK, Views, Flashvideo a través del cual se usa ffmpeg y el reproductor de flash JW Player, Image para le creación de galerías de imágenes, Organic Groups que permite el uso de grupos de trabajo dentro del área privada, OG Calendar para la presentación de eventos en un calendario y todo un seguido de módulos menores necesarios para el buen funcionamiento de los módulos principales.

También cabe destacar la funcionalidad de la subida de ficheros de pulsómetro desarrollada especialmente para este proyecto a través del CMS. No se ha usado ningún módulo o funcionalidad externa en su integración con Drupal ya que él mismo posee herramientas adecuadas para este tipo de funcionalidades sencillas.

Testing Drupal: En esta fase se han probado las funcionalidades del portal en los sistemas operativos Linux en las distribuciones Ubuntu 7.10 y 8.05.4, Devian Etch, Windows 2000 y Windows XP, compatibles con Apache, con éxito. También se ha provado en diferentes versiones de los navegadores Internet Explorer 6.0 y 7.0, Mozilla Firefox 2 y 3 y Opera 8, 9.4 y 9.5, también con éxito. Han surgido diversos problemas a la hora del instalar los codificadores de audio y video de ffmpeg ya que es un sistema complicado de instalar. Se han probado varias versiones por compilar y finalmente se ha usado la instalación de paquetes precompilados en las versiones de Linux cuya instalación es mucho más sencilla, y un ejecutable en las versiones de Windows con resultados satisfactorios. La distribución que se presenta en este proyecto incluye una el ejecutable de ffmpeg para Windows sin la necesidad de instalarlo previamente.

Todas las funcionalidades anteriormente mencionadas del portal se han probado satisfactoriamente en las pruebas realizadas en los diversos navegadores y distribuciones de los diversos sistemas operativos.

Migración de Drupal 5 a Drupal 6: En un principio se decidió implementar el portal en la versión 5 de Drupal ya que la versión 6 todavía estaba en una fase de desarrollo poco avanzada y la versión 5 ofrecía en este aspecto una mayor fiabilidad i un soporte técnico muchos más amplio así como un número de contribuciones mucho mayor y más estables. Se contempló la posibilidad de hacer una migración del portal más adelante si las aportaciones y el desarrollo de la nueva versión lo hacían posibles. Fue el caso positivo y el portal finalmente pudo ser migrado de una versión a otra satisfactoriamente y con prácticamente todas las funcionalidades intactas, con lo que podemos alargar el tiempo de soporte y contribuciones oficiales a nuestro portal web, así como nuevos servicios que hacen más fácil . En el apartado *Actualizaciones de Drupal 6* del tercer capítulo se detallarán los pormenores en más profundidad que, básicamente, han sido saber escoger los módulos y las versiones apropiadas para Drupal 6 a partir de la prueba y error.

Mejoras de seguridad en el portal web: Se han llevado a cabo una serie de mejoras en el portal web a través de las actualizaciones de Drupal 6 que solucionan varios problemas de seguridad interna, así como de la mejora del código de la aplicación para subir ficheros de pulsómetro al servidor. Se han llevado a cabo mayoritariamente sobre el código PHP de las aplicaciones para evitar posibles vulnerabilidades contra ataque de Cross Site Scripting (XSS), que consiste en aprovechar la arquitectura del sitio web e insertar una serie de parámetros maliciosos en el código html de la página para acceder a privilegios no autorizados para dichos clientes, y SQL injection que son ataques deliberados contra la base de datos mediante la adición de sentencias en código SQL a variables que contengan dicho código y se usen en la interacción del servidor y la base de datos MySQL. En el apartado dedicado a la especificación del código de la aplicación de ficheros de pulsómetro se explica con más detalle.

Solución de problemas en la base de datos: Como en todo proyecto pueden surgir problemas, en este en concreto debemos resaltar por encima de todo los problemas causados por incompatibilidades en la base de datos del servidor principal donde se hospedan el resto de servicios. Se han detectado anomalías causadas por incompatibilidades del sistema operativo Windows con MySQL, mostrando un error en los archivos temporales generados por la base de datos en el proceso de consulta y modificación de información en las diferentes bases de datos contenidas en el servidor MySQL, anunciando la imposibilidad de escribir en un archivo ya creado. Se han eliminado todos los archivos temporales de dicho directorio y se ha modificado el path del directorio temporal por uno dentro del directorio de instalación del servidor de bases de datos para que el problema no persista. Teniendo en cuenta que también pudiera tratarse de un conflicto entre MySQL y el programa de antivirus, se ha investigado dichos problemas y se ha excluido del escaneo rutinario en directorio temporal para evitar posibles conflictos, dejando dicho escaneo en revisiones en intervalos de tiempo más dilatados. También se ha instalado un servidor alternativo para comprobar que no sugieran más problemas en el funcionamiento del portal web, pero algunos de los servicios del servidor principal no han sido posibles de trasladar, como pueden ser R-php y R o el resto de bases de datos usadas por otros proyectos dada la persistencia del error en la base de datos que impedía su importación a otros servidores.

Elaboración de la memoria i manual de usuario: Dada su extensión y la constante reestructuración del trabajo en el tramo final del proyecto, se decidió empezar la elaboración de la memoria en un plazo anterior al descrito en el informe preliminar a la vez que se solucionaban problemas y se añadían las funcionalidades restantes. En un principio se había planteado como un manual de usuario y una memoria por separados, pero hemos creído oportuno su elaboración en un sólo documento conjunto explicado a varios niveles técnico,s ya que hace que sea más cómoda su consulta y uso. Este es el resultado apreciable de su conclusión.

Diseño entre la comunicación entre Drupal i R-php: Dados los problemas mantenidos con el servidor de bases de datos principal, se ha dejado en segundo término la oferta de este servicio, dado que de ser integrado más exhaustivamente, serían necesarios una cantidad muy superior de recursos y tiempo de implementación e implantación. Se ha optado pues, por un diseño sencillo y funcional de un servicio básico con el cual poder trabajar con las tecnologías ya desarrolladas en el proyecto. En un principio se diseñó la comunicación a través de la especificación e

implementación de un módulo totalmente nuevo para Drupal 6 pero dada su complejidad, hemos utilizado la inserción de código PHP en varias páginas del portal mediante las opciones que Drupal 6 dispone de base para ello y se ha elaborado una guía de programación y desarrollo de módulos de Drupal 6 en el IDE NetBEan 6.5 para facilitar el trabajo en futuras ampliaciones del proyecto y/o desarrollo de terceros en Drupal para el conjunto de proyectos en el cual se engloba el presente. No obstante, la comunicación entre una página web y R mediante R-php ha sido posible su integración con éxito en un proyecto paralelo a éste, englobado en el ámbito del proyecto de investigación sobre la aplicación de modelos estadísticos en deporte de competición y otras actividades físicas.

Testing del proyecto: En este apartado se engloba todas las pruebas realizadas en el portal desarrollado en Drupal, el funcionamiento de la comunicación de la página web con el resto de aplicaciones y en diversos sistemas operativos y navegadores, así como el buen funcionamiento de la base de datos de MySQL y las mejoras de seguridad.

2.3.2. Costes

Uno de los objetivos de este proyecto, como se ha expuesto anteriormente, consiste en proporcionar un servicio lo más barato posible en cuanto a adquisición se refiera, ya que se usará como una herramienta de difusión y aprendizaje gratuita en las comunidades. El uso de software bajo licencias GPL y Creative Commons no es fruto de la casualidad ya que esto nos permite la libre distribución y modificación del proyecto según las necesidades de cada usuario, además de facilitar mucho las cosas en cuanto a ayuda en su desarrollo se refiere.

Pasemos a describir brevemente en qué consisten las licencias de software principalmente usadas en este proyecto:



Las licencias Creative Commons o CC están inspiradas en la licencia GPL (General Public License) de la Free Software Foundation, sin embargo no son un tipo de licenciamiento de software. La idea principal es posibilitar un modelo legal ayudado por herramientas informáticas para así facilitar la distribución y el uso de contenidos.





Existen una serie de licencias Creative Commons, cada una con diferentes configuraciones o principios, como el derecho del autor original a dar libertad para citar su obra, reproducirla, crear obras derivadas, ofrecerla públicamente y con diferentes restricciones como no permitir el uso comercial o respetar la autoría original.

Una de las licencias que ofrecía Creative Commons es la que llevaba por nombre "Developing Nations" (Naciones en Desarrollo), la cual permitía que los derechos de autor y regalías por el uso de las obras se cobraran sólo en los países desarrollados del primer mundo, mientras que se ofrecían de forma abierta en los países en vías de desarrollo. Esta licencia ha sido retirada por problemas comerciales.

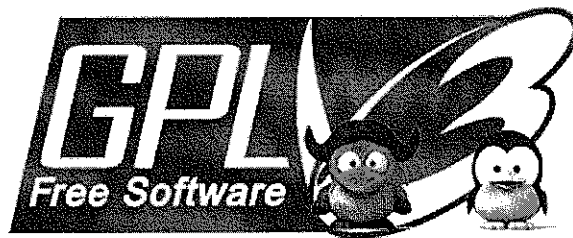
Aunque originalmente fueron redactadas en inglés, las licencias han sido adaptadas a varias legislaciones alrededor del mundo. Entre otros idiomas, han sido traducidas al castellano, al portugués, al euskera y al catalán a través del proyecto Creative Commons International. Existen varios países de habla hispana que están involucrados en este proceso: España, Chile, Argentina, México, Perú, Colombia y Puerto Rico ya tienen las licencias traducidas y en funcionamiento, en

tanto que Guatemala, Ecuador y Venezuela se encuentran en proceso de traducción e implementación de las mismas. Brasil también tiene las licencias traducidas y adaptadas a su legislación.

Las diferentes licencias Creative Commons se basan en combinar distintas propiedades. Estas propiedades son:

-  • **Attribution (by):** Obliga a citar las fuentes de esos contenidos. El autor debe figurar en los créditos.
-  • **Noncommercial o NonCommercial (nc):** Obliga a que el uso de los contenidos no pueda tener bonificación económica alguna para quien haga uso de contenidos bajo esa licencia.
-  • **No Derivative Works or NoDerivs (nd):** Obliga a que esa obra sea distribuida inalterada, sin cambios.
-  • **ShareAlike (sa):** Obliga a que todas las obras derivadas se distribuyan siempre bajo la misma licencia del trabajo original.

Muchas de las licencias Creative Commons se identifican con el acrónimo CC, que hace referencia a su nombre.



La Licencia Pública General de GNU o más conocida por su nombre en inglés, GNU General Public License o simplemente su acrónimo del inglés GNU GPL, es una licencia creada por la Free Software Foundation a mediados de los 80 y está orientada principalmente a proteger la libre distribución, modificación y uso de software. Su propósito es declarar que el software cubierto por esta licencia es software libre y protegerlo de intentos de apropiación que restrinjan esas libertades a los usuarios.

Existen varias licencias "hermanas" de la GPL, como la licencia de documentación libre de GNU (GFDL) que cubre los artículos de la Wikipedia, la Open Audio License para trabajos musicales, etcétera, y otras menos restrictivas, como la MGPL, o la LGPL (Lesser General Public License, antes Library General Public License), que permiten el enlace dinámico de aplicaciones libres a aplicaciones no libres.

La licencia GPL, al ser un documento que cede ciertos derechos al usuario, asume la forma de un contrato, por lo que usualmente se le denomina contrato de licencia o acuerdo de licencia. En los países de tradición anglosajona existe una distinción doctrinal entre licencias y contratos, pero esto no ocurre en los países de tradición civil o continental. Como contrato, la GPL debe cumplir los requisitos legales de formación contractual en cada jurisdicción.

La licencia ha sido reconocida por juzgados en Alemania, particularmente en el caso de una sentencia en un tribunal de Munich, lo que indica positivamente su validez en jurisdicciones de derecho civil.

Después de 14 años de ser usada sin cambios, la licencia GNU GPL vuelve a la mesa de diseño. La Free Software Foundation está actualmente abocada a actualizarla en varios sentidos:

- Resolver formas en que a pesar de todo alguien podía quitar libertades a los usuarios.
- Como un caso especial de lo anterior: Prohibir el uso de software cubierto por la licencia en sistemas diseñados para quitar libertades (DRM).
- Resolver ambigüedades y aumentar su compatibilidad con otras licencias.
- Facilitar su adaptación a otros países.
- Incluir cláusulas que defiendan a la comunidad de software libre del uso indebido de patentes de software.
- Mostrar usuarios registrados

El proceso de revisión de la nueva versión de la licencia se inició el 16 de enero de 2006 en el MIT con la presentación del primer borrador. La versión final de la GPL 3 fue hecha pública el 29 de junio de 2007 y es accesible a través del Portal de GNU.

Costes.

Así pues, bajo estas licencias es posible reducir considerablemente los costes derivados de la utilización de licencias de las distintas aplicaciones usadas para el desarrollo del proyecto que nos ocupa.

En referencia a los costes humanos derivados de la implementación de dichas tecnologías en un producto útil, cabe destacar la separación entre dos tipos de cálculo de costes: el derivado del diseño y desarrollo de la aplicación de subida de archivos de pulsómetro y el resto del proyecto.

Para calcular los costes de la primera parte, se ha utilizado un método llamado COCOMO (CONstructive COSt MOdel) desarrollado por por Barry M. Boehm. Un método de estimación de coste de desarrollo no es otra cosa que establecer una relación matemática entre el esfuerzo y el tiempo requerido para desarrollar un producto / proyecto.

Lo primero que necesitamos saber es el tipo de proyecto del que estamos hablando, hay tres tipos diferentes:

- **Orgánico:** proyectos relativamente sencillos. En este modo, un pequeño grupo de programadores experimentados desarrollan software en un entorno familiar. El tamaño del software varía de unos pocos miles de líneas (pequeño) a unas decenas de miles de líneas (medio).
- **Semi-acoplado:** proyectos intermedios en complejidad y tamaño. En este tipo, el proyecto tiene unas fuertes restricciones, que pueden estar relacionadas con el procesador y el interface hardware. El problema a resolver es único y es difícil basarse en la experiencia, puesto que puede no haberla.
- **Empotrado:** proyectos bastante complejos, en los que apenas se tiene experiencia y se engloban en un entorno de gran innovación técnica. Además se trabaja con unos requisitos muy restrictivos y de gran volatilidad.

Ahora por otro lado tendríamos los diferentes modelos que define COCOMO: Modelo básico, Modelo Intermedio y Modelo Avanzado.

En este punto quiero recordar que este tipo de modelos están orientados a la magnitud del producto final, midiendo el tamaño del proyecto en líneas de código principalmente, por el hecho de realizarse en base al número de líneas de código se mide el producto y su tamaño pero en ningún momento podremos llegar a medir la productividad.

Categoría	Programadores	Duración	Líneas de código	Ejemplo
Trivial	1	0 - 4 semanas	< 1k	Utilidad de ordenación
Pequeño	1	1 - 6 meses	1k - 3k	Biblioteca de funciones
Media	2 - 5	0,5 - 2 años	3k - 50k	Compilador de C
Grande	5 - 20	2 - 3 años	50k - 100k	SO pequeño
Muy grande	100 - 1000	4 - 5 años	100k - 1M	Grandes SO
Gigante	1000-5000	5 - 10 años	> 1M	Sistema de Distribución

Figura 2-5. Dificultad de un proyecto en función de sus líneas de código.

Proyecto de software	a	b	c	d
Orgánico	2,4	1,05	2,5	0,38
Semiacoplado	3	1,12	2,5	0,35
Empotrado	3,6	1,2	2,5	0,32

Figura 2-6. Coeficientes COCOMO

La ecuación del esfuerzo de COCOMO es:

$$E = a(Kl)^b * m(X)$$

Donde:

a y b son constantes con valores definidos en cada submodelo

Kl son las líneas de código (en miles)

el resultado es en salarios/mes y horas-hombre.

$m(X)$ Es un multiplicador que depende de 15 atributos.

En nuestro caso usaremos el modelo básico ya que describe a la perfección la necesidades pequeño código desarrollado. Dicho código posee unas 500 líneas de código por lo tanto, según hemos podido ver se trata de un Proyecto Orgánico, por lo tanto el tamaño es de 0,5 miles de líneas de código. Si aplicamos las fórmulas suponiendo el sueldo medio de un programador senior son 2166,67 €/mes en bruto, estzs son las fórmulas usadas:

Personas necesarias por mes para llevar adelante el proyecto (MM) = $a*(Kl)^b$

Tiempo de desarrollo del proyecto (TDEV) = $c*(MM)^d$

Personas necesarias para realizar el proyecto (CosteH) = MM/TDEV

Coste total del proyecto (CosteM) = CosteH * Salario medio entre los programadores y analistas.

$$MM = 2,4 * 0,5^{1,05} = 1,16 \text{ personas / mes}$$

$$TDEV = 2,5 * 1,16^{0,38} = 2,64 \text{ meses}$$

$$\text{CosteH} = MM/TDEV = 1,16/2,64 = 0,44 \text{ personas (para 1 persona se necesitan 34,85 días aprox.)}$$

$$\text{CosteM} = 0,44 * 2166,67 = 953,33 \text{ €}$$

Si comparamos con el trabajo realizado en la previsión y el diagrama de gannt veremos una mejoría en el precio considerable suponiendo el mismo precio en el salario del personal.

Para el resto de horas podemos realizar un sencillo cálculo entre las horas y el sueldo medio en bruto de un analista de sistemas:

Sueldo en bruto de un analista de sistemas por hora: 33.000 €/año - 17,19 €/mes (suponiendo 20 días laborables por mes y una jornada de 8 horas)

Horas/día de trabajo: 4h/día

Horas totales: $217 * 4 = 868$ h

Horas sin contar desarrollo de software: $858 - 34,85 * 8 = 579,2$ h (ya que el cálculo de desarrollo de software preveía jornadas de 8 horas semanales)

Coste total: $579,2 * 17,19 + 953,33 = 10909,778$ €

2.4. Drupal

2.4.1. ¿Porqué Drupal?

Préviamente ya se han expuesto razones de porqué se escogió un CMS a la hora de implementar el portal pero también se ha mencionado que se barajaron varias posibilidades en cuanto a CMS se refiere. Primordialmente se escogieron como opciones Drupal y Joomla ya que son dos de los CMS más consolidados en Internet y tras investigar un poco, nos decantamos por Drupal por varios motivos expuestos a continuación. He aquí la relación de pros y contras de cada uno de ellos:

Drupal

- Plataforma muy sólida y de gran calidad técnica.
- Permite una mantener un varios sitios reales en un mismo servidor (solo una instalación para varios sitios).
- Cualquier tipo de permisos de usuario, grupos de usuario además de cumplir con el estándar OpenId en la versión 6.
- Permite desarrollar una membresía y una comunidad de sitios aparte del un solo CMS.
- Posee un potente sistema de templates. Cualquier template de XHTML o CSS puede ser fácilmente convertido en Drupal.
- Necesita una inversión de tiempo mayor para tener en cuenta las tremendas posibilidades que posee en el desarrollo de un sitio.
- Código claro y de gran calidad. Posee una API (fácil de integrar con otras soluciones, etc.)
- Flexibilidad y limitaciones aún desconocidas.
- Permite gestionar sitios con gran volumen de visitas. (p.ej.: MTV UK, BBC, the Onion, Nasa, Greenpeace UK, etc.)

Joomla

- Si el nivel técnico del usuario no es muy avanzado, es un buen comienzo.
- Fácil de instalar y poner a punto con un simple clic de ratón.
- Curva de aprendizaje fácil.
- No permite la integración de otros scripts en el sitio.
- Generalmente no se pueden crear sitios de gran tráfico sin invertir una gran cantidad de esfuerzo y recursos.
- No posee un gran buscador interno y las URLs no son fáciles de manejar por los buscadores.
- La utilización de recursos del servidor es mayor en comparación con Drupal.
- Solo permite un sitio por instalación.
- No permite el registro centralizado en varios sitios.
- No permite grupos de usuarios y permisos.

- El interfaz de administración es más intuitivo.
- Posee algunos módulos de mayor calidad para servicios como calendarios, encuestas, etc.
- La ampliación de módulos cuesta dinero.

Requisitos del sistema:

- Drupal permite trabajar con MySQL y Postgre mientras Joomla sólo se tiene constancia de que soporte MySQL.
- Drupal permite trabajar con Apache o IIS mientras que Joomla aparentemente solo lo hace con Apache
- Joomla soporta paginas y registro de SSL. Drupal no se tiene constancia de que lo haga.

Administración del sitio:

- Drupal posee modulo gratuitos para la gestión de Workflow. No se tiene constancia que Joomla not posea alguno.
- Drupal posee un gestor de traducciones interno. Joomla posee un addon gratuito para lo mismo.
- Drupal posee una gestión de privilegios mucho más granular.

Interoperabilidad:

- Drupal posee soporte para iCal (Add on) , Joomla no se tiene constancia que posea alguno.
- Drupal cumple los estándares de XHTML. Joomla no se tiene constancia que lo haga.
- Drupal posee un excelente seguimiento de versiones por parte de la comunidad, cosa de la que Joomla carece.

Después de de este balance es fácil decantarse por una opción sabiendo los objetivos de este proyecto.

2.4.1 Estructura y funcionamiento.

Drupal funciona en a diversos niveles con diversas tecnologías tal y como se muestra en la figura 2-7 .

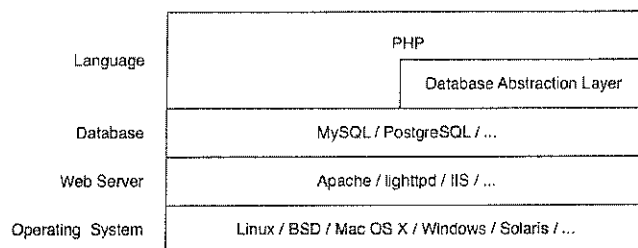


Figura 2-7. Capas en el diseño de Drupal

Descritas de abajo a arriba, se encuentra en primer lugar el sistema operativo. Está en un nivel demasiado bajo como para que Drupal pueda verse afectado y funciona perfectamente en cualquier sistema operativo que soporte PHP.

El servidor que más se usa hoy en día con Drupal es Apache ya que la mayor parte de su desarrollo se ha llevado a cabo con dicho servidor. Drupal utiliza los ficheros *.htacce*s para

asegurar una instalación segura en el servidor. Posee la opción de trabajar con URLs limpias (que significa que no aparecen los parámetros de ejecución de cada página del CMS en la barra de direcciones del navegador con el cual se acceda al contenido) a través de el módulo de Apache *mod_rewrite*. Este concepto es importante ya que al migrar desde otro CMS o desde una página estática, no es necesario cambiar las direcciones para su correcto funcionamiento.

Drupal se comunica con la siguiente capa de la pila (la base de datos) a través de una ligera capa de abstracción de la base de datos. Esta capa permite el saneamiento de las sentencias SQL en cuanto a seguridad se refiere y hace posible la utilización los productos de varios fabricantes sin tener que rehacer el código del sitio en caso de cambiar la el software de la base de datos. Las bases de datos más testeadas en Drupal son MySQL y PostgreSQL. Aunque Drupal acceda a la base de datos mediante la capa de abstracción, también permite usar otras bases de datos mediante php de manera tradicional a través de Snippets, que son fragmentos de PHP insertados en las páginas de Drupal. Hablaremos de ellos con detalle más adelante.

Drupal está escrito en PHP y al ser un lenguaje interpretado es necesaria la instalación en el servidor donde se haga funcionar el CMS. PHP tiene mala reputación dentro del mundo del desarrollo de aplicaciones web ya que es fácil aprender y gran parte del código que se usa en Internet ha sido escrito por principiantes. También, como en muchos otros lenguajes de programación, es a menudo atacado o usado para hackear sistemas. De todas formas es posible escribir código robusto y fiable y la prueba es que Drupal está escrito bajo unas estrictas normas de programación dictadas por los estándares PEAR.

El núcleo.

Un framework ligero compone el núcleo de Drupal, la parte que se obtiene al descargar drupal de *drupal.org*. Es el responsable de proveer las funcionalidades básicas que se usen para dar soporte a otras partes del sistema. Incluye código que permite al sistema de Drupal arrancar (bootstrap) cuando recibe una petición una librería común de funciones usadas frecuentemente en Drupal y módulos que permiten usar funcionalidades básicas como gestión de usuarios, taxonomía y el uso de plantillas (templates).

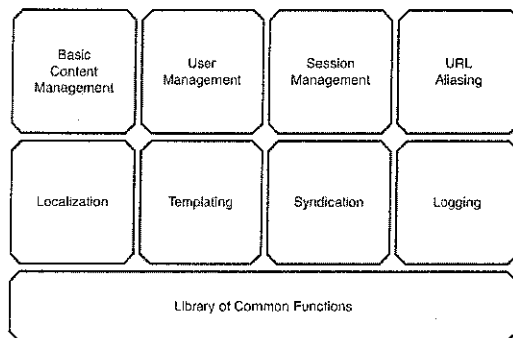


Figura 2-8. Una vista general del núcleo de Drupal (Parcial.)

El interfaz administrativo.

El interfaz administrativo está estrechamente integrado con el resto del sitio y por defecto, utiliza el mismo tema. El primer usuario, el 1, es el superusuario el cual posee acceso completo a todas las funcionalidades del sitio. Después de registrarse como el usuario 1, es posible ver un enlace en la página de administración con el boque de dicho usuario (ver más en la sección de bloques). Cada bloque de usuario puede contener diferentes enlaces dependiendo del nivel de acceso que posean en el sitio web.

Módulos

Drupal es un framework tremendamente modular. Las funcionalidades están incluidas en *módulos*, los cuales pueden ser activados o desactivados (algunos pueden no ser desactivados ya que puede que otros módulos necesiten de su activación para su funcionamiento). Los nuevos servicios se añaden a una página en Drupal mediante la activación de módulos existentes en el sistema, instalando módulos escritos por miembros de la comunidad de Drupal o escribiendo los propios. De esta manera, los sitios web que no necesiten algunas de las funcionalidades que se ofrecen con la instalación básica o módulos previamente usados y actualmente no necesitados, pueden desactivar su uso permitiendo un mejor funcionamiento del sistema, y aquellos que necesiten nuevas funcionalidades, pueden añadirlas sin problemas. Esto se puede observar en la figura 2-9.

La adición de nuevos tipos de contenido tales como recetas, entradas de blog, ficheros adjuntos o nuevos comportamientos como notificaciones por correo electrónico, publicaciones peer-to-peer y la agregación de feeds, están regidas y administradas a través de módulos.

Drupal hace uso de patrones de diseño llamados *control de inversión*, en las cuales las funcionalidades modulares son llamadas por el framework a su tiempo. Estas oportunidades para los módulos de hacer su función son llamadas *hooks*.

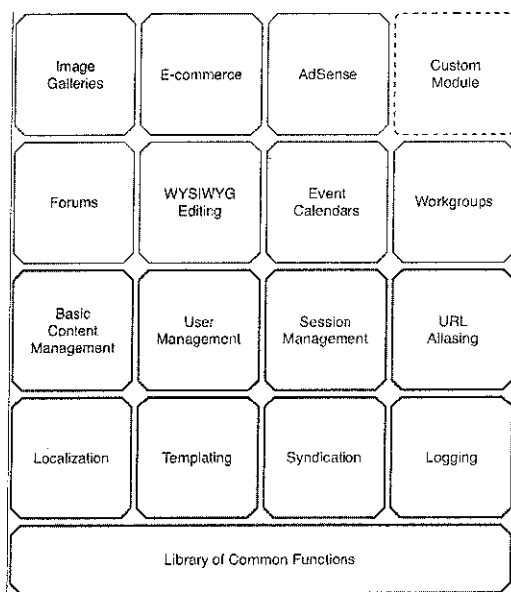


Figura 2-9. Al activar módulos adicionales, se añaden más funcionalidades.

Hooks

Los *hooks* (o ganchos, en inglés) pueden pensarse como eventos internos de Drupal. También se llama *callbacks*, dado que están contruidos a más bien través de una convención de nombramientos de funciones que por una espera activa, ya que no son llamados realmente por un proceso específico del cual parten. Los *hooks* permiten cualquier módulo “engancharse” a lo que sucede en el resto de Drupal.

Supongamos un registro de usuarios en un sitio contruido a través de Drupal. En el mismo momento que el usuario se valida en el sitio, Drupal ejecuta un *hook*, lo que significa que cualquier función nombrada de acuerdo con la convención *nombre_del_módulo* más *nombre_del*

_hook, será llamada en ese instante. Por ejemplo, la función *comment_user()* en el módulo *comment*, *locale_user()* en el módulo *locale*, *node_user()* en el módulo *node* y cualquier otra función nombrada de la misma forma será llamada por el sistema. Si se escribiera un módulo personalizado llamado *spam.module* e incluyera una función llamada *spam_user()* que enviara un e-mail al usuario, la función sería llamada también y dicho usuario recibiría un correo electrónico en cada inicio de sesión. La manera más común de entrar en las funcionalidades del núcleo de Drupal es mediante la implementación de *hooks* en módulos.

Temas

Al crear una página web que enviar a un navegador, hay dos razones a tener en cuenta: construir los datos pertinentes y presentarlos de cara a la Web. En Drupal, la capa de temas (theme layer) es la responsable de crear el código HTML que el navegador recibe. Se pueden usar diferentes aproximaciones populares sobre cómo crear temas tales como Smarty, un Lenguaje de atributos para la creación de temas para PHP (PHPATAL) y PHPTemplate. Es importante señalar que Drupal recomienda la separación de contenido y presentación.

Drupal permite varias maneras de personalizar y reemplazar el *look and feel* de un sitio web. La manera más simple es mediante el uso de CSS para reemplazar la clases preconstruidas y los IDs de Drupal. De todas formas, si lo que se desea es ir más allá de personalizar la salida HTML, es fácil de hacer. Los ficheros de patrones de Drupal consisten en código estándar de HTML y PHP. Además, cada parte dinámica de una página de Drupal (como un recuadro o una lista) pueden ser reemplazadas simplemente declarando una función con el nombre apropiado, usando Drupal entonces dicha función en vez de la original.

Nodos

Los tipos de contenido en Drupal derivan de un tipo básico referenciado como *nodo*. Ya sea una entrada de blog, una receta o incluso una tarea de un proyecto, la estructura subyacente de datos es la misma. El mecanismo bajo esta aproximación es su extensibilidad. Los desarrolladores de módulos pueden añadir nuevas funcionalidades tales como valoraciones, comentarios, ficheros adjuntos, información geolocalizada y demás para nodos en general sin preocuparse si el tipo de nodo es blog, receta, etc. El administrador del sitio puede mezclar y unir funcionalidades mediante el tipo de contenido, como por ejemplo, escogiendo la opción de permitir comentarios en blogs pero no recetas o activando la subida de ficheros para tareas de proyectos solamente.

Los nodos también pueden contener una colección básica de propiedades de comportamiento que otros tipos de contenido pueden heredar. Cualquier nodo puede ser promocionado a la página principal, ser marcado como publicado o no, o incluso permitir su búsqueda. Es gracias a esta estructura uniforme que el interfaz administrativo ofrece una pantalla de edición para trabajar con nodos.

Bloques

Un bloque es información que puede ser activada o desactivada en un lugar específico del sitio web. Es posible tener un bloque que contenga a los usuarios más activos o a una lista de eventos venideros. Los bloques se colocan típicamente en la barra lateral de una plantilla, o en la cabecera o a pie de página. También se pueden configurar para mostrarse en ciertos tipos de nodos, solamente en la página principal o de acuerdo con los criterios de cada uno.

A menudo, los bloques se usan para representar información propia para el usuario que tenga una sesión abierta en un momento dado. Por ejemplo, un bloque de navegación contiene enlaces solamente a las funciones de administración solo para aquellos que tengan este tipo de acceso. La colocación y la visibilidad de los bloques se administra a través de un interfaz web en las opciones de administración.

El diseño del sistema de ficheros.

Es importante entender la estructura de directorios de Drupal si queremos depurar los errores que podamos encontrar en nuestra página web y saber sobre varias de las mejores prácticas a seguir, tales como: Dónde se encuentran los ficheros subidos al servidor, dónde se deberían encontrar los temas o cómo poder disponer de varios perfiles de Drupal. La figura 2-10 muestra los directorios de una instalación por defecto de Drupal.

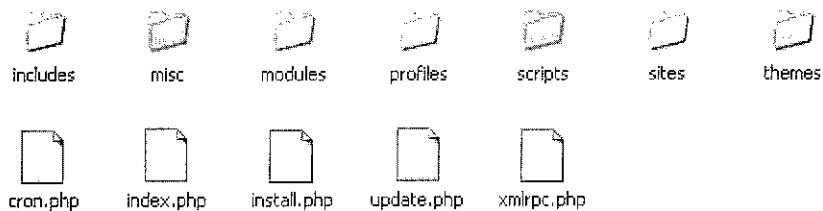


Figura 2-10. La estructura de carpetas en una instalación de Drupal por defecto.

Los detalles sobre cada uno de los elementos en la estructura de directorios son:

- El directorio *includes* contiene la librería de funciones comunes que usa Drupal.
- El directorio *misc* guarda código JavaScript y varios iconos e imágenes disponibles de la instalación inicial de Drupal.
- El directorio *modules* contiene los módulos núcleo, cada uno en su propio directorio. Es preferible no modificar nada en este directorio para evitar problemas.
- El directorio *profiles* contiene diferentes perfiles de instalación para un sitio. Si hubiera otros perfiles además del perfil por defecto en este subdirectorio, Drupal preguntaría que perfil se quiere instalar la primera vez que se instala el sitio en Drupal. La principal razón de un perfil de instalación es activar ciertos módulos núcleo contribuidos automáticamente como podrían ser las instalaciones en un idioma diferente del inglés o un perfil de e-commerce que automáticamente configurara Drupal como una plataforma de este tipo.
- El directorio *scripts* contiene scripts para comprobar sintaxis, sanear código y gestionar casos especiales con *cron*. No se usa dentro del ciclo de vida de las peticiones dentro del sistema; son scripts de shell y Perl para el mantenimiento, tareas puntuales u otras situaciones en las que se requieran.
- El directorio *sites* (ver figura 2-11) contiene todas las modificaciones que necesitemos hacer en Drupal en la forma de ajustes, módulos y temas. Cuando añadimos módulos a Drupal del repositorio de módulos contribuidos o los nuestros propios, se colocan en *sites/all/modules*. Esto permite mantener todas las modificaciones que se hagan a Drupal en un mismo directorio. Dentro de *sites* hay un directorio llamado *default* que contiene el archivo de configuración por defecto del sitio en Drupal: *settings.php*. El directorio *default* normalmente se copia y renombra con el nombre del URL del sitio en cuestión, quedando la configuración de dicho sitio en *sites/www.example.com/settings.php*, donde *www.example.com* es el nombre del sitio. El directorio *themes* contiene el sistema de plantillas y los temas por defecto en Drupal. Para añadir temas, basta ubicar los temas nuevos en *sites/all/themes* para que puedan mostrarse correctamente.
- Uno de los directorios a destacar es *sites/default/files*. Es el lugar donde se ubican todos los archivos para personalizar el logotipo del sitio web, permitir el uso de avatares o subir archivos asociados con el sitio en cuestión, así como los archivos de traducción de contenido. Es importante que este directorio posea permisos de lectura y escritura adecuados a través del servidor web en el que Drupal funcione. Posteriormente se puede cambiar la ubicación de *files* a otro lugar a través del panel de administración.

- El fichero *robots.txt* permite que los programas de búsqueda (*spiders*) usados por buscadores como Google, omitan las URLs, ficheros y directorios listados en este archivo por URL convencionales o limpias (ver capítulo 3.2.4. *Administración*, sección *Configuración de sitio* → *URLs limpias*).

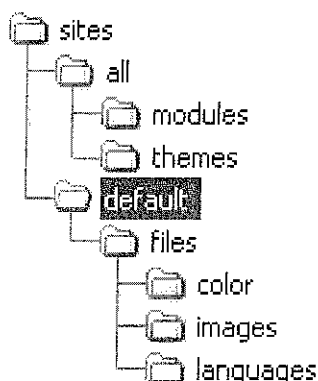


Figure 2-11. El directorio sites permite guardas todas las modificaciones hechas a Drupal.

Sirviendo una petición.

Esta sección describe un esquema de lo que ocurre cuando una petición es recibida por Drupal:

El rol del servidor Web: Drupal se ejecuta bajo un servidor web que acostumbra a ser Apache. Si el servidor web respeta el fichero *.htaccess* de Drupal, algunos ajustes de PHP se inician y las URL limpias se activan (para ello es necesario previamente que *mod_rewrite* se active). Drupal traduce una dirección a *index.php?=foo/bar* en *index.php/foolbar* a través de una regla de *mod_rewrite* pero continua usando la primera dirección internamente (guardada en el parametro de petición de URL *q*). A esta dirección interna se la llama *Drupal path*.

En otros servidores web alternativos como Microsoft IIS, las URL limpias se pueden guardar mediante un módulo ISAPI (Windows Internet Server Application Programming Interface) como ISAPI_Rewrite.

El proceso de Bootstrap

Drupal se autoarranca (bootstrap) a sí mismo en cada petición a través de una serie de fases de bootstrap. Estas fases se definen en *bootstrap.inc* y proceden como se describe a continuación:

- **Configuración:** Esta fase llena el array interno de configuración de Drupal y establece la URL base (*\$base_url*) del sitio. El archivo *settings.php* se parsea vía la función *include_once()*, y cualquier variable establecida como sobreescrita con anterioridad es aplicada.
- **Caché de Early page:** En situaciones donde se requiere un gran nivel de escalabilidad, un sistema de caché puede necesitar se invocado antes de intentar siquiera establecer una conexión a la base de datos. La fase de caché de early pages permite incluir (con *include()*) un fichero PHP que contenga una función llamada *page_cache_fastpath()*, la cual almacena y devuelve el contenido al navegador. La caché de early pages se activa a través de la variable *page_cache_fastpath* a CIERTO y el fichero a incluir se define asignando a variable *cache_inc* en el path del fichero PHP.

- **Base de Datos:** Durante la fase de la base de datos, se determina el tipo de base de datos y se hace la conexión inicial que se usará para las subsiguientes peticiones a dicha base de datos.
- **Acceso:** Drupal permite restringir hosts por relaciones de entre nombre del host/dirección IP. En la fase de acceso se comprueba rápidamente si la petición proviene de un host no permitido, en cuyo caso el acceso es denegado.
- **Sesión:** Drupal saca partido de la gestión de sesiones de PHP pero sustituye algunas de las reglas de gestión por otras propias para implementar una administración de sesiones con fondo en base de datos. Las sesiones se inician o se reestablecen en la fase de sesión.
- **Caché de late pages:** Aquí es donde el sistema carga suficiente código de soporte para determinar si servir una página de la caché de páginas o no. Esto incluye mezclar configuraciones de la base de datos en un array que ha sido creado durante la fase de configuración, con la carga o el parseo de código de módulos. Si la sesión indica que la petición ha sido hecha por un usuario anónimo y el caching de páginas se ha activado, la página se devuelve de la caché y la ejecución se para.
- **Dirección:** En la fase de dirección, se carga el código que gestiona direcciones y alias de direcciones. Esta fase permite que se resuelvan URLs que puedan ser interpretadas por las personas y gestiona direcciones internas de Drupal.
- **Completa:** Esta fase completa el proceso de bootstrap mediante la carga de la librería de funciones comunes, soporte de temas y soporte para mapeado de callbacks, gestión de ficheros, Unicode, juegos de herramientas de imagen de PHP, la creación y proceso de formularios, tablas autoordenables y los resultados del inicio de páginas. En gestor de errores personalizado de Drupal se inicia, *locale* se inicia y todos los módulos activos se cargan. Finalmente, Drupal ejecuta el hook *init* de manera que los módulos tengan la oportunidad de ser notificados antes del comienzo del proceso oficial de peticiones. Una vez Drupal ha completado el arranque inicial, todos los componentes del framework restan disponibles. Es tiempo de recoger la petición del navegador y pasarla a la función de PHP que la gestione. El mapeado entre URLs y las funciones que las manejan tiene lugar usando un callback de registro que se ocupa del mapeado de la URL y del control de acceso. Los módulos registran sus callbacks usando el hook *menu*. Cuando Drupal ha determinado que existe un callback hacia el lugar en que se encuentra mapeada la URL de la petición del navegador y que el usuario tiene suficientes privilegios para acceder a dicho callback, se ejecuta dicho callback.

Procesando una petición

La función de callback hace el trabajo que se requiera para procesar y acumular los datos necesarios para completar la petición. Por ejemplo, si se recibe una petición de contenido tal como `http://example.com/q=node/3`, la URL se mapea en la función `node_page_view()` en `node.module`. El posterior procesamiento obtendrá los datos para dicho nodo de la base de datos y lo pondrá en una estructura de datos. Entonces es cuando entra la colocación del tema.

Colocando el tema de la página a los datos.

Este proceso involucra la transformación de los datos obtenidos, manipulados, o creados en código HTML. Drupal usará el tema que el administrador haya seleccionado para dar a la página la correcta presentación deseada y enviará en código HTML resultante al navegador.

3. EL PORTAL WEB

3.1. Instalación

Durante este capítulo describiremos detalladamente el proceso de instalación del sitio a través de un tutorial que guiará al usuario en cada momento. La instalación del portal web consta de tres partes o fases bien definidas: la instalación de los archivos de Drupal 6 en el servidor, la instalación de la base de datos de MySQL y la instalación del códer ffmpeg y el reproductor de flash JW Player.

3.1.1. Archivos de Drupal.

Drupal posee un mecanismo de instalación en el caso que hagamos una instalación desde cero, que nos guiará en el proceso. Por nuestra parte, dicho proceso no será necesario ya que poseemos una instalación ya completada. El proceso a seguir es muy sencillo:

- Copiamos el archivo *drupal.zip* en el directorio raíz de nuestra página web y descomprimos su contenido en él.
- Debemos editar el archivo *settings.php* situado en *sites/default/* con un editor de texto y buscar la siguiente línea marcada:

```
* Database URL format:  
* $db_url = 'mysql://username:password@localhost/databasename';  
* $db_url = 'mysqli://username:password@localhost/databasename';  
* $db_url = 'pgsql://username:password@localhost/databasename';  
*/  
$db_url = 'mysqli://usuario:password@localhost/drupal_bd';  
$db_prefix = '';
```

Figura 3-1. Settings.php

Debemos substituir en la línea marcada en azul de la figura 3-1 *usuario* por nuestro usuario en la base de datos en MySQL y *password* por la contraseña de dicho usuario para la base de datos y guardar el archivo. Una vez hecho esto, habremos terminado esta fase de la instalación.

3.1.2 La base de datos.

Para instalar la base de datos necesitamos es necesaria la herramienta *phpmyadmin* que se encarga de gestionar bases de datos en MySQL. Los pasos a seguir son sencillos a la hora de introducirla en nuestro sistema. Debemos importar el archivo *drupal_bd.sql* tal y como muestran las figuras 3-2, 3-3, 3-4, 3-5 y 3-6. Previamente, debemos crear la base de datos llamada *drupal_bd* y una vez creada, acceder a ella mediante un simple clic en su nombre, situado en la barra lateral izquierda, tal y como muestran la figuras 3-7. y 3-8.

localhost

- Versión del servidor: 5.0.51b-community-nt
 - Versión del protocolo: 10
 - Servidor: localhost via TCP/IP
 - Usuario: root@localhost
- Juegos de caracteres de MySQL: UTF-8 Unicode (utf8)
- Cotejamiento de las conexiones MySQL:
utf8_unicode_ci
- Crear nueva base de datos**
Cotejamiento
Crear
- Mostrar información de tiempo de ejecución de MySQL
- Mostrar las variables del sistema MySQL
- Procesos
- Juego de caracteres y sus cotejamientos
- Motores de almacenamiento
- Cargar los privilegios nuevamente
- Privilegios
- Bases de datos
- Exportar
- Importar

phpMyAdmin - 2.11.7

- Versión del cliente: 5.0.51b
- Extensiones PHP utilizadas: mysql
- Idioma - Language: Español - Spanish
- Tema / Estilo: Original
- Tamaño del font: 82%
- Documentación de phpMyAdmin
 - Wiki del phpMyAdmin
- Página oficial de phpMyAdmin
 - [ChangeLog] [Subversion] [Lists]



Figura 3-2. Crear base de datos *drupal_bd*.

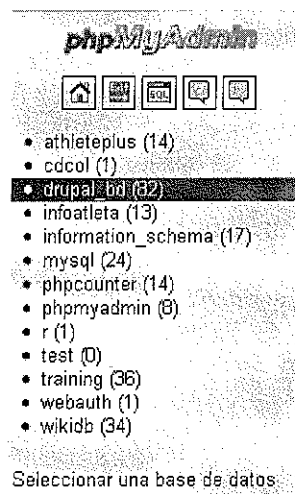


Figura 3-3. seleccionar base de datos *drupal_bd*.

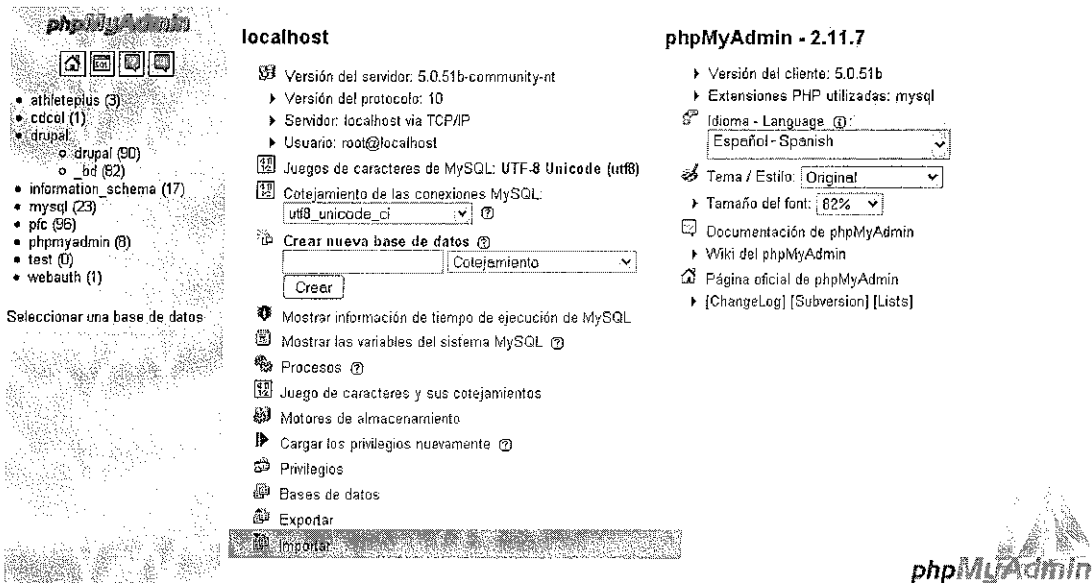


Figura 3-4. Importar base de datos. Paso 1.

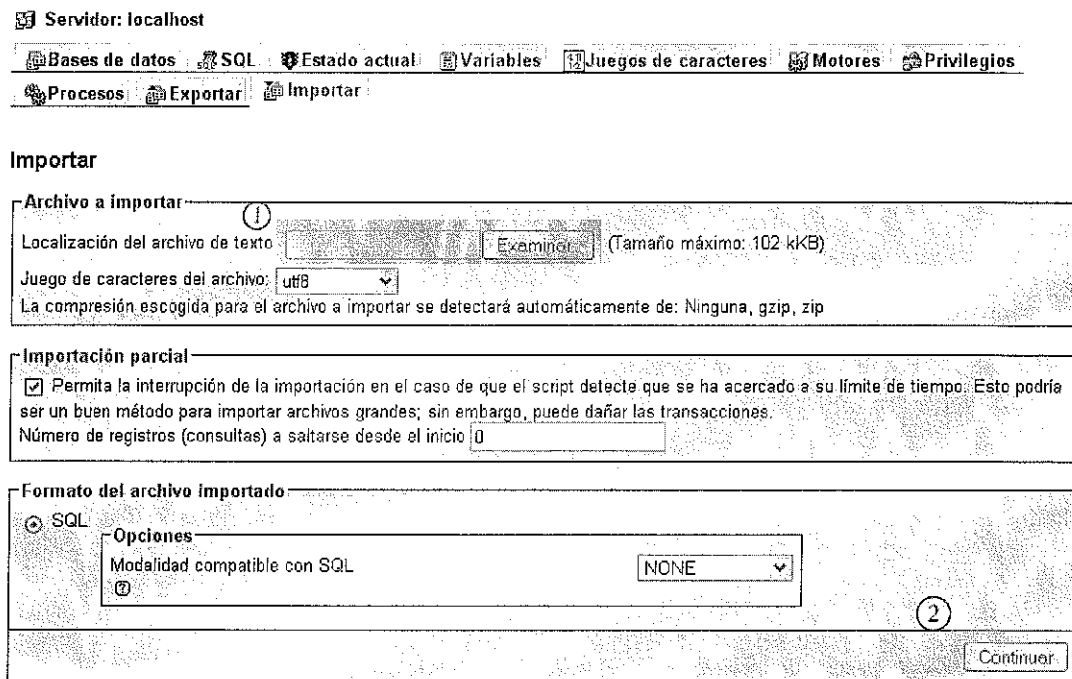


Figura 3-5. Importar base de datos. Paso 2.

Después de haber instalado la base de datos es importante, una vez funcione el portal, proporcionar el path correcto de los archivos del servidor web. Para ello introduciremos dicho path en el campo de *Carpetas temporal* en Administración→Configuración del sitio→Sistema de ficheros tal y como muestra la figura 3-6:

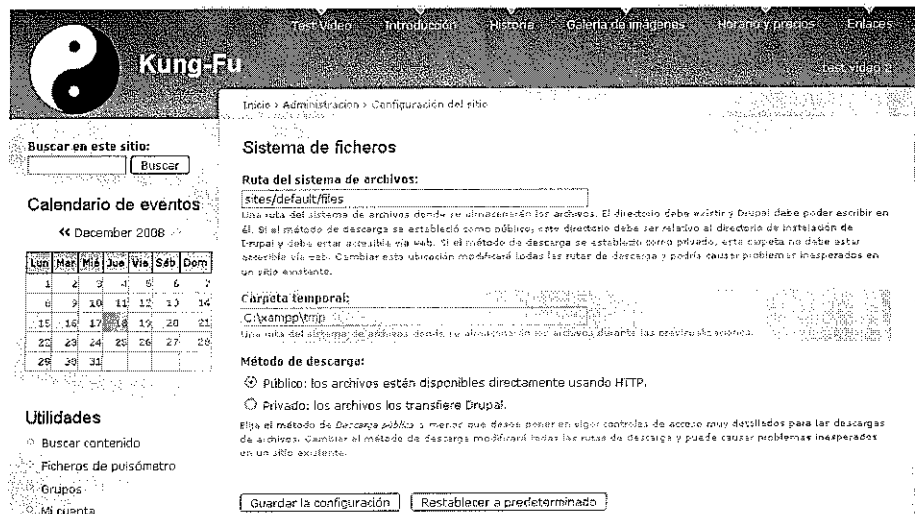


Figura 3-6. Menú de Sistema de ficheros.

En caso de querer instalar un sitio adicional solo es necesario crear una base de datos en el servidor MySQL con nombre diferente e importar los archivos existentes con el procedimiento descrito anteriormente en este punto y sustituir en la línea de código descrita en la figura 3-1 *drupal_bd* por el nombre de la base de datos que hayamos creado. El resto de pasos son exactamente iguales.

Es muy importante que en las bases de datos que hayamos creado con el usuario con el cual nos conectaremos al servidor MySQL tengamos los siguientes privilegios en dicha base de datos: SELECT, INSERT, UPDATE, DELETE, CREATE, DROP, INDEX, ALTER, CREATE TEMPORARY TABLES, LOCK TABLES. Si no poseemos esos privilegios, nuestro portal web no funcionará correctamente y puede ocasionar problemas graves, para ello podemos ejecutar la siguiente instrucción seleccionando la pestaña superior SQL y ejecutar la siguiente instrucción:

```
GRANT SELECT, INSERT, UPDATE, DELETE, CREATE, DROP, INDEX, ALTER, CREATE
TEMPORARY TABLES, LOCK TABLES ON database.* TO 'username'@'localhost'
IDENTIFIED BY 'password';
```

Donde *database* es el nombre de la base de datos (*drupal_bd* u otra en caso de haber instalado más de un sitio) *username* es el nombre del usuario y *password* es la contraseña para dicho usuario. Si no disponemos de privilegios suficientes para ejecutar esta instrucción no devolverá un mensaje de error indicándonoslo, debemos contactar con el administrador del servidor para pedir dichos privilegios o para que nos conceda los privilegios mencionados anteriormente para las bases de datos que lo necesiten.

Multisite

Drupal ofrece la opción de gestionar varios sitios web que posean diferentes funcionalidades y que de cara al usuario final se presenten como portales independientes, pero que en el fondo comparten módulos básicos y contribuidos, la base de datos y algunas, todas o ninguna opción de configuración. El procedimiento a seguir es sencillo.

Añadimos a *setting.php* por cada sitio que deseemos:

```
$db_url = 'mysql://username:password@localhost/drupal_bd'
$base_url = 'http://sitio'
```

Donde *username* es el usuario con el que nos conectamos a la base de datos, *password* es la contraseña para dicho usuario y *sitio* es el nombre del sitio a añadir. Seguidamente deberíamos configurar el archivo *hosts* (en el caso de Windows: *System32\drivers\etc*, en el caso de Unix: */etc*) y añadir:

(Dirección IP sitio1) (nombre del sitio1)
(Dirección IP sitio2) (nombre del sitio2)

...

Ejemplo:

127.0.0.1 kung-fu

Y por último editar el fichero de Apache *httpd-vhosts.conf* y añadir para cada uno de los sitios:

```
<VirtualHost *:80>  
DocumentRoot Directorio_Sitio  
ServerName sitio  
</VirtualHost>
```

Donde *Directorio_Sitio* es el directorio de instalación de cada sitio en el disco duro y *sitio* es el nombre de cada sitio.

Por último, deberemos copiar los archivos personalizables de cada uno de los sitios en el directorio de Drupal a */sites/NombreSitio/* quedando una estructura tal que así:

```
sites/NombreSitio/modules  
sites/NombreSitio/files  
sites/NombreSitio/themes
```

Podemos personalizar la configuración de cada uno de los sitios añadiendo un *settings.php* a su directorio base. Es importante saber que la configuración y archivos comunes se encuentra en *sites/default/all* y *sites/all* respectivamente.

3.1.3. FFMPEG y JW Player

Aunque este códer es una herramienta tremendamente potente, es también complicada de instalar dependiendo de las funciones que queramos agregar. Es posible descargarse los archivos de código fuente para compilarlos en nuestro sistema pero también es muy poco recomendable si no somos expertos en ello. Para los usuarios que usen Windows, la instalación de los archivos de Drupal ya posee una versión compilada y funcional de FFMPEG en el directorio *sites/all/coder/*, con lo cual no hará falta nada más que cambiar el path de ejecución en las opciones del configuración del módulo Flashvideo en el panel de administración (*Administración* → *Configuración del sitio* → *Preferencias de Flashvideo* → *Global FlashVideo Settings (For all node types)*). Es importante introducir el path completo incluyendo la unidad de disco en que está instalada en caso de Windows, y no el relativo al servidor. En caso de contratar un servicio de hosting, será necesario contactar con el administrador para que nos facilite el path a nuestro directorio raíz.

NodeType	Estatus	Operaciones
Global Settings	(Enabled)	Global FlashVideo Settings (For all node types)
Imagen	(Disabled)	FlashVideo Settings for this node type.
Evento	(Enabled)	FlashVideo Settings for this node type.
Grupo	(Disabled)	FlashVideo Settings for this node type.
Página	(Enabled)	FlashVideo Settings for this node type.
Historia	(Enabled)	FlashVideo Settings for this node type.

Figura 3-7. Opciones de Flashvideo

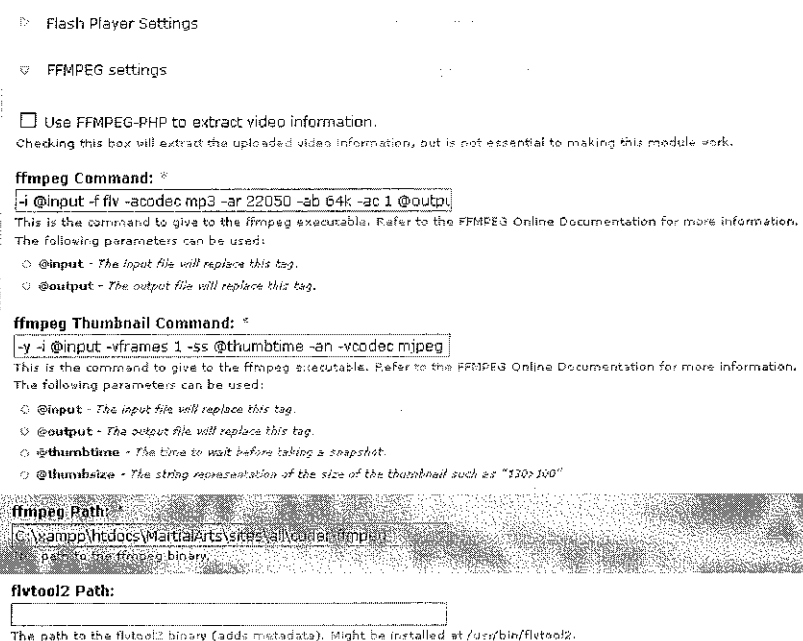


Figura 3-8. Path a los archivos de FFMPEG.

En el caso de poseer una instalación de Linux, aconsejamos instalar una versión precompilada con el gestor de paquetes que c la distribución de Linux que tengamos. En el caso de una Debian Etch podemos hacer lo siguiente:

```
pico /etc/apt/sources.list
```

Añadimos la siguiente línea:

```
deb http://www.debian-multimedia.org stable main
```

Esto nos permite instalar una versión precompilada en nuestro servidor con las opciones que necesitamos para la codificación de vídeo

Ejecutamos:

```
apt-get update
```

Módulos básicos.

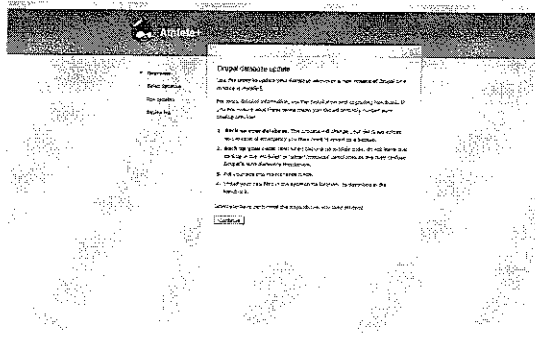


Figura 3-74. *update.php*

Este caso requiere varios pasos:

1. Hacemos una copia de seguridad de la base de datos de Drupal mediante *phpmyadmin* de la manera que hemos descrito en el capítulo de instalación del portal, y de los archivos *robots.txt*, *.htaccess* y todo el contenido del directorio */sites*.
2. Activamos la opción de desactivar las funcionalidades del sitio en *Administración* → *Configuración del sitio* → *Mantenimiento del sitio*.
3. En *Administración* → *Construcción del sitio* → *Temas* cambiamos el tema actual por uno básico como *Garland*. Es importante que lo hagamos ya que los temas contribuidos pueden generar errores. También cambiaremos el idioma predefinido al inglés para evitar posibles pérdidas de traducción en *Administración* → *Configuración del sitio* → *Idiomas*. Y por último desactivaremos todos los módulos contribuidos en *Administración* → *Construcción de sitio* → *Módulos*.
4. Descargamos la nueva versión de los módulos básicos para nuestra versión de Drupal. En nuestro caso, Drupal 6.
5. Eliminamos todos los archivos del directorio de instalación y copiamos los nuevos. También debemos copiar los archivos de drupal de los cuales hemos hecho copia de seguridad anteriormente descrita en el punto 1, sobrescribiendo los existentes en caso de haber.
6. Ejecutamos en un navegador el script *update.php*. Por ejemplo, si nuestro sitio se llama *www.ejemplo.com*, ejecutaremos *http://www.ejemplo.com/update.php*.
7. En este punto, se ejecutarán una serie de actualizaciones en las cuales se nos mostrarán los pasos a seguir. Es posible que, dado que tenemos el módulo *CCK*, debemos activar ciertas funcionalidades tras ejecutar el *update.php* y volver a ejecutarlo nuevamente tras activarlas. Una vez hayamos hecho esto, debemos activar los módulos tal y como los teníamos en un principio y volver a ejecutar *update.php* para actualizar la base de datos. Finalizado este punto, no debería hacer falta volver a ejecutar dicho script.
8. Restauramos el tema antiguo, ejecutamos el script de cron en *Administración* → *Informes* → *Informes de estado* y reconstruimos los permisos en esta misma página.

Una vez hecho esto, ya habremos finalizado la actualización de nuestro portal. Es posible que queramos mover el fichero *update.php* del directorio raíz de drupal para evitar problemas de seguridad a otro lugar. Debemos recordar que *update.php* sólo funciona en dicho directorio raíz en posteriores actualizaciones.

El proceso que hemos llevado a cabo para el paso de la versión de Drupal 5 a Drupal 6 ha sido el mismo pero teniendo en cuenta que las versiones de módulos a descargar son de la versión 6 y no de la 5. No hemos tenido constancia de ningún tipo de error en dicho proceso de actualización.

3.5. Ficheros de pulsómetro.

Ficheros de pulsómetro es una aplicación desarrollada especialmente para este proyecto desde cero. Consiste en un pequeño script incrustado en una página de contenido de Drupal que permite la inserción de datos generales en la base de datos *Athleteplus* correspondientes a una sesión de entrenamiento y su correspondiente archivo de pulsómetro que se alojará en un directorio llamado *logs* en el directorio raíz de apache para su posterior uso por otras aplicaciones del resto de proyectos englobados en una mayor de investigación. Podemos ejecutar dicho script en *Logs* → *Subir log* y listar su información a través de un filtro de búsqueda en *Logs* → *Listado de logs*. Describiremos su código por partes:

Subir ficheros de pulsómetro e información adicional.

```
<?php
global $user;
$uid = $user->uid;

$db_host = "localhost";
$db_user = "user";
$db_pass = "password";

if ($uid < 1) {
    watchdog('access denied', check_plain($_GET['q']), NULL, WATCHDOG_WARNING);
    drupal_set_title(check_plain("Acceso denegado"));
    echo check_plain("No está autorizado a ver a esta página.");
}
else {
    $campo="profile_DNI";
    $result = db_query("SELECT v.value FROM {profile_values} v INNER JOIN
    {profile_fields} f ON f.fid = v.fid WHERE v.uid = %d AND f.name = '%s'", $uid,
    $campo);

    while ($row = db_fetch_object($result)) {
        $dni=$row->value;
    }

    $c_fallida = 0;
    //cargar las actividades de la tabla de actividades.
    //Base de datos athleteplus
    $con = mysql_connect($db_host,$db_user,$db_pass);
    if (!$con){
        drupal_set_message('No se puede conectar a la base de datos: ' .
        mysql_error(),'error');
        $c_fallida = 1;
    }
    if ($c_fallida != 1) {
        mysql_select_db("athleteplus", $con);
        $where = sprintf("SELECT identificador,nomActivitat FROM activitats WHERE
        Descripcio = 'general' OR Descripcio = 'marcial' ORDER BY identificador");
        $where_q = mysql_query($where,$con);
        if (!$where_q){
            drupal_set_message('Error: ' . mysql_error(),'error');
            $c_fallida = 1;
        }
    }
}
?>
```

En esta parte se describe el siguiente flujo de trabajo: comprobamos que se trate de un usuario registrado a través de la variable *\$uid*. En caso que sea menor que 1 se trata de un usuario anónimo y por lo tanto, lanzaremos un mensaje de error denegando el acceso y usando la función *watchdog* para registrar dicho error de acceso. Colocamos el título de la página mediante *drupal_set_title()* y le pasamos un string des descripción del error mediante la función *check_plain()*. Es necesario saber que la función *check_plain()* se usa para filtrar cadenas de caracteres e impedir ataques de Cross site Scripting (XSS), que añaden instrucciones maliciosas en nuestro código HTML y pueden comprometer la seguridad de los datos guardados o dañar

nuestra página. `$_GET['q']` proporciona el path URL de la página, en este caso el cuerpo del mensaje de error.

Si es cualquier otro tipo de usuario, entonces podemos ejecutar el resto del código. Lo primero a hacer es escoger el campo DNI a través del identificador de usuario consultando la base de datos. En esta consulta, usamos la interficie de bases de datos de Drupal. Escogemos sólo una línea ya que solo debería haber un valor. Inicializamos la laviable de control de errores de conexiones `$c_fallida` con un 0 lo que muestra que no se ha producido ningún error de conexión y realizamos la conexión a la base de datos Athleteplus para obtener los datos de las actividades a mostrar en el desplegable correspondiente del formulario HTML para la instrucción de los datos. Primero llevamos a cabo control de errores en la conexión con la base de datos. Es preciso señalar que en estas conexiones usaremos las funciones de PHP para la gestión de bases de datos y no el interfaz de Drupal dado que las tablas de Athleteplus están fuera de la base de datos de Drupal. Para realizar la conexión usamos `mysql_connect()` que devuelve un descriptor de estado de la conexión usado por otras funciones de PHP, y le pasamos el host donde se encuentra la base de datos, el usuario y la contraseña con los cuales nos conectamos a la base de datos y que hemos definido al principio del script una vez como `$db_host`, `$db_user` y `$db_pass` respectivamente. No hemos querido publicar el usuario y contraseña usados en el código por motivos de seguridad del servidor en el cual se ejecuta actualmente la aplicación, por eso usamos `"user"` y `"password"` respectivamente. Con `drupal_set_message()` podemos mostrar mensajes en formato de aviso o error propios de drupal pasándolo por parámetro el mensaje a mostrar (en este caso `mysql_error()` devuelve el error de la conexión en caso de haber) y el tipo de mensaje (`'error'` para mensajes de error y `'status'` para mensajes de informe que se mostrarán en rojo y verde respectivamente. Por defecto, si se omite este último parámetro, serán mensajes de informe). En caso de no haber una conexión fallida, seleccionamos la base de datos en la cual consultar mediante la función `mysql_select_db()` a la cual le pasamos por parámetro el nombre de la base de datos y la variable que hemos usado para la conexión, `$con` (es preciso notar que en PHP las variables las definimos en el momento que deseamos mediante la asignación de un valor, no es necesario predefinirlas antes y siempre han de llevar el símbolo `$` delante del nombre. Pueden contener cualquier tipo de datos). Lanzamos la consulta a la base de datos sobre el tipo de actividades existentes mediante `mysql_query()` especificando en los parámetros la consulta en un string y la conexión, y llevamos a cabo un control de errores.

```
<form action="?q=subida_pulsometro" method="post" enctype="multipart/form-data">
<label for="file">Nombre del fichero:</label>
<input type="file" name="file" id="file" />
<br/>
<br/>
Hora de inicio (hh:mm:ss): <br/><input name="horaini" type="text" size="30"
maxlength="128"><br/>
Actividad:<br/> <select name="actividad" type="text">
  <option value=""></option>
  <?php
    if ($c_fallida != 1){
      while($fila = mysql_fetch_array($where_q)) {
        echo "<option value='" . check_plain($fila['identificador']) .
          "'> " . check_plain($fila['nomActivitat']) . '</option>\n';
      }
      mysql_close($con);
    }
  ?>
</select><br/>
Duración (hh:mm:ss): <br/><input name="duracion" type="text" size="30"
maxlength="128"><br/>
Distancia: <br/><input name="distancia" type="text" size="30" maxlength="128"><br/>
<style>
  .notas_st {height: 80px;}
</style>
Notas: <br/><input name="notas" type="text" size="55" maxlength="500"
class = notas_st><br/>
Sensación de Actividad:<br/> <select name="sensact" type="text">
  <option value=""></option>
```

```

<option value ="1">1</option>
<option value ="2">2</option>
<option value ="3">3</option>
<option value ="4">4</option>
<option value ="5">5</option>
</select><br/>
<br/>
<div align="left"><input type="submit" value="Enviar "/></div>
</form>

```

En esta parte insertamos el código HTML para el formulario de envío de ficheros de pulsómetro y la información adicional para la sesión de entrenamiento. Se puede observar que las partes de HTML y PHP están claramente diferenciadas dado que en la estructura de snippets de Drupal, que son fragmentos de código PHP a ejecutar en el contenido de una página, es la manera de que el código funciona, ya que normalmente los archivos .html y los php se referencian pero van separados. Dentro del contenido de Drupal no funcionarían correctamente de esta forma. Primero construimos los campos para enviar el archivo definiendo el formulario con un método post y especificando la página de la cual extraemos la información. Dado que el código que usamos es un snippet que insertamos en una página de Drupal, especificaremos la dirección de dicha página con una dirección interna de Drupal: `?q=subida_pulsometro`, que previamente hemos definido en la creación del contenido. Posteriormente, definimos los campos de información adicionales *Hora de Inicio* (un recuadro de texto), *Actividad* (un desplegable) donde insertaremos el código PHP correspondiente a la obtención de las actividades en la base de datos (a resaltar el uso de comillas dobles y simples en la generación del tags HTML dentro de código PHP para no confundir al intérprete de este último al declarar el string y los caracteres especiales / y " que haya dentro. También cerramos la conexión con la base de datos mediante `mysql_close($con)`, *Duración*, *Distancia*, *Notas* (recuadros de texto) y *Sensación de Actividad* (un desplegable). En el recuadro de texto *Notas* definimos previamente una cláusula de estilo `<style>` para personalizar el ancho del recuadro ya que se trata de un campo de comentarios y por lo tanto es necesaria una cantidad mayor de caracteres.

```

<?php
//Al cargar la pagina no queremos añadir nada a la BD tontamente a menos que haya
//algo que añadir
if ($FILES["file"]["name"] != "") {
    //control de errores al subir el fichero
    $ok = 1;
    if ( $_POST["horaini"] != "" ) {
        $horaini = $_POST["horaini"];
    }
    else {
        drupal_set_message("No hay Hora de inicio <br />", 'error');
        $ok = 0;
    }
    if ( $_POST["actividad"] != "" ) {
        $actividad = $_POST["actividad"];
    }
    else {
        drupal_set_message("No hay Actividad <br />", 'error');
        $ok = 0;
    }
    if ( $_POST["duracion"] != "" ) {
        $duracion = $_POST["duracion"];
    }
    else {
        drupal_set_message("No hay Duración <br />", 'error');
        $ok = 0;
    }
    if ( $_POST["distancia"] != "" ) {
        $distancia = $_POST["distancia"];
    }
    else {
        $distancia = 0;
    }
}
$notas = $_POST["notas"];

```



```

if ( $_POST["sensact"] != ""){
    $sensact = $_POST["sensact"];
}
else {
    drupal_set_message("No hay Sensación de Actividad <br />", 'error');
    $ok = 0;
}

```

En este fragmento controlamos que se introduzcan valores para los campos *Hora de inicio*, *Actividad*, *Duración*, *Distancia* y *Sensación de Actividad*. En el caso de que uno de ellos no se introdujera, se mostraría un mensaje de error y no se cargaría la información de fichero ni del resto del proceso del formulario mediante el uso de la variable \$ok.

```

if ($FILES["file"]["error"] > 0){
    drupal_set_message("Error: " . check_plain($FILES["file"]["error"])) .
    "<br />", 'error');
    $ok = 0;
}

if ($ok != 0) {
    $mensaje = "Fichero: " . check_plain($FILES["file"]["name"]) . "<br />";
    $mensaje .= "Tipo: " . check_plain($FILES["file"]["type"]) . "<br />";
    $mensaje .= "Tamaño: " . check_plain($FILES["file"]["size"] / 1024) .
    " Kb<br />";
    drupal_set_message($mensaje, 'status');

    //El directorio destino:
    //[NombreDNI]/[Año]/Fecha_Hora_NombreArchivo en /logs
    $fecha = strftime("%d%m%Y_%H%M%S_");
    $fecha_sql = date("Y-m-d");
    $any=strftime("%Y");
    $dir .= $_SERVER['DOCUMENT_ROOT'] . "/logs/" . $user->name . $dni .
    "/" . $any . "/";
    $FILES['file']['name'] = $fecha . $FILES['file']['name'];

    if (file_exists($dir . $FILES["file"]["name"]) &&
    ($FILES["file"]["name"] != "")){
        drupal_set_message($FILES["file"]["name"] . " Ya existe. <br />", 'error');
    }
}

```

En este fragmento se comprueban los errores en la subida de fichero, en caso que se produzca alguno, se pone \$ok a 0 y se lanza un mensaje de error. En caso contrario, Se muestran mensajes de informe sobre el nombre del fichero, el tipo de fichero y el tamaño de este. Una vez llevado a cabo este proceso, creamos la fecha con el formato correspondiente para añadirla como prefijo en el nombre del archivo definitivo ya que de momento el fichero subido es un archivo temporal de PHP, e introducirla en la base de datos respectivamente. Creamos el path donde se ubicará el fichero subido con el directorio completo donde se ubica nuestro servidor web, el subdirectorio *logs/*, el año, la información del campo DNI del perfil de usuario extraída de la base de datos de Drupal y comprobaremos que no exista. En caso afirmativo, lanzaremos un mensaje de error y en caso contrario proseguiremos con la ejecución del script.

```

else{
    if (!file_exists($dir)){
        //creamos subdirectorios si no estan creados
        mkdir_recursive($dir);
    }
    move_uploaded_file($FILES["file"]["tmp_name"], $dir .
    $FILES["file"]["name"]);
    $ruta = $dir . $FILES["file"]["name"];
    //Base de datos athleteplus
    $con = mysql_connect($db_host,$db_user,$db_pass);
    if (!$con){
        drupal_set_message('No se puede conectar a la base de
        datos: ' . mysql_error(),'error');
    }
}

```


Una vez llevado a cabo la petición de inserción, comprobaremos que no se haya producido ningún error en dicha petición (en cuyo caso haremos lo habitual: mostraremos un mensaje y terminaremos el flujo de ejecución mediante *\$c_fallida*. Es importante señalar que en todo el flujo de ejecución del programa, cuando se genera un error, no se ejecuta más código posterior y se termina el script) y se genera un mensaje de estado al usuario. Para finalizar, terminaremos la conexión con la base de datos y saldremos del script.

```
function mkdir_recursive($pathname){
    is_dir(dirname($pathname)) || mkdir_recursive(dirname($pathname));
    return is_dir($pathname) || @mkdir($pathname);
}
?>
```

Al final del código del script definimos la función que nos permite crear directorios no existentes. Para ello usamos un algoritmo recursivo en el cual comprobamos para cada subdirectorío, su existencia y en caso negativo creamos el directorío y pasando al siguiente nivel, y en caso afirmativo únicamente esto último.

Listados de la información de la subida de datos de pulsómetro.

```
<?php
global $user;
$uid = $user->uid;

$db_host = "localhost";
$db_user = "usuario";
$db_pass = "password";

if ($uid < 1) {
    watchdog('access denied', check_plain($_GET['q']), NULL, WATCHDOG_WARNING);
    drupal_set_title(check_plain("Acceso denegado"));
    echo check_plain("No está autorizado a ver a esta página.");
}
else {
    $c_fallida = 0;
    //cargar las actividades de la tabla de actividades.
    //Base de datos athleteplus
    $con = mysql_connect($db_host,$db_user,$db_pass);
    if (!$con){
        drupal_set_message('No se puede conectar a la base de datos: ' .
            mysql_error(),'error');
        $c_fallida = 1;
    }
    if ($c_fallida != 1) {
        mysql_select_db("athleteplus", $con);
        $where = sprintf("SELECT identificador,nomActivitat FROM activitats WHERE
            Descripcio = 'general' OR Descripcio = 'marcial' ORDER BY identificador");
        $where_q = mysql_query($where,$con);
        if (!$where_q){
            drupal_set_message('Error: ' . mysql_error(),'error');
            $c_fallida = 1;
        }
    }
}
?>
```

El script del listado empieza de manera similar al de subida de pulsómetro. Se controla el acceso de los usuario, se conecta a la base de datos mediante *\$db_host*, *\$db_user* y *\$db_pass*, controlando los errores que ésta pueda lanzar en caso de haber y seleccionando los identificadores de las actividades de la tabla *logs* de la base de datos *athleteplus*. Una vez obtenidas las actividades se inicia la construcción del formulario primeramente añadiendo los parámetros de estilo para la tabla en la que se mostrarán los resultados mediante la cláusula *<style>* y fijando los valores del tamaño de la letra y su alineación para la cabecera de la tabla (*th*) y el resto de datos de las filas (*td*).

```

<style>
    td {font-size: 70%; text-align: center;}
    th {font-size: 80%; text-align: center;}
</style>

<form action="?q=listado_logs" method="post"
enctype="multipart/form-data">
Fecha desde (AAAA-MM-DD): <br/><input name="fechadesde" type="text" size="30"
maxlength="20"><br/>
Fecha hasta (AAAA-MM-DD): <br/><input name="fechahasta" type="text" size="30"
maxlength="20"><br/>
Actividad:<br/> <select name="actividad" type="text">
    <option value=""></option>
<?php
    if ($c_fallida != 1){
        while($fila = mysql_fetch_array($where_q)) {
            echo "<option value='" . check_plain($fila['identificador']) .
            "'> ". check_plain($fila['nomActivitat']) . '</option>\n';
            $act_to_array .= $fila['identificador'] . "_" .
            $fila['nomActivitat'] . " ";
        }
        mysql_close($con);
    }
?>
</select><br/>
Sensación de Actividad:<br/> <select name="sensact" type="text">
    <option value=""></option>
    <option value ="1">1</option>
    <option value ="2">2</option>
    <option value ="3">3</option>
    <option value ="4">4</option>
    <option value ="5">5</option>
</select><br/>
<br/>
Usar filtro <input type="checkbox" name="filtro" value="ok"><br/>
<br/>
<br/>
<div align="left"><input type="submit" value="Enviar"/></div>
</form>

```

La manera de crear el formulario es exactamente la misma que para el script de subida del pulsómetro. Definimos el formulario mediante el nombre de la página en Drupal e insertamos los filtros por *fecha desde* y *fecha hasta*, que nos permitirá filtrar los datos por fecha de creación siendo posible añadir únicamente un valor para *fecha desde*, lo cual mostrará los valores a partir de esa fecha en concreto. Es importante entrar la fecha en el formato especificado entre paréntesis para su correcto funcionamiento: cuatro dígitos para el año, dos para el mes y dos para el día todos ellos separados por guiones medios (-). También podemos filtrar los datos mediante el tipo de actividad mostrada en un desplegable que rellenaremos exactamente de la misma forma que el script anterior, guardando los valores y nombres en un string (*\$act_to_array*) que nos será útil más adelante para mostrar el nombre de la actividad ya que la consulta a los datos de subida de pulsómetro se hace mediante el identificador, y por sensación de actividad también creado del mismo modo. La novedad es la casilla de selección que nos indica si queremos filtrar los datos o que se nos muestren todos. Por defecto, se muestra el resultado de esta última opción. Por último definimos el botón *Enviar* para procesar los datos.

Seguidamente, comprobamos que las conexiones y peticiones a la base de datos se hayan ejecutado con éxito para seguir con el resto del script. Construimos la sentencia base en SQL usando el número de usuario en el sistema de Drupal que previamente hemos obtenido al comenzar el script y comprobamos que se haya activado el filtro y que cada uno de los componentes de filtrado tenga alguna valor que no sea vacío. En caso afirmativo para cada uno de ellos, se añadirá la parte correspondiente de código SQL a la variable *\$sql*, se contabilizará y se guardará dicho valor. Solamente contabilizaremos tres de los cuatro valores a escoger dado que fecha hasta solo se usará en caso que fecha desde posea algún valor, en cuyo caso se añadirá el fragmento SQL correspondiente para la petición y se guardará su valor. Es importante el hecho

de añadir los valores como %s o %d ya que posteriormente deberemos blindar la sentencia SQL contra posibles ataques de inyección SQL, mediante la función *anti_SQLinjection()*.

```
<?php
if ($c_fallida != 1){
    //Construimos la consulta
    $sql="SELECT
    data,Path,HoraInici,activitat,durada,distancia,sensacioActivitat,notes
    FROM log WHERE uid = '%d'";
    //construimos el filtro de consulta
    if ($_POST["filtro"] == "ok"){
        $num_param = 0;
        if ($_POST["fechadesde"] != "") {
            $num_param++;
            $fd = $_POST["fechadesde"];
            if ($_POST["fechahasta"] != "") {
                $fh = $_POST["fechahasta"];
                $sql.= " AND (data >= '%s' OR data <= '%s')";
            }
        }
        else{
            $sql.= " AND data >= '%s'";
        }
    }
    if ($_POST["actividad"] != "") {
        $num_param++;
        $act = $_POST["actividad"];
        $sql.= " AND activitat = '%s'";
    }
    if ($_POST["sensact"] != ""){
        $num_param++;
        $sact = $_POST["sensact"];
        $sql.= " AND sensacioActivitat = '%d'";
    }
}
$sql.=" ORDER BY data,HoraInici";

$con = mysql_connect($db_host,$db_user,$db_pass);
if (!$con){
    drupal_set_message('No se puede conectar a la base de datos: ' .
    mysql_error(),'error');
    $c_fallida = 1;
}
if ($c_fallida != 1){
    mysql_select_db("athleteplus", $con);
    $sql = anti_SQLinjection($fd,$fh,$act,$sact,$sql,$_POST['filtro'],
        $num_param,$uid);
    $where_q = mysql_query($sql,$con);
    if (!$where_q){
        if (mysql_errno($con) != 1065){
            drupal_set_message('No se puede conectar a la base de
            datos: ' . mysql_error(),'error');
        }
        else{
            drupal_set_message('No hay logs','status');
        }
    }
    $c_fallida = 1;
}
}
```

Una vez construida la sentencia SQL, conectamos al servidor MySQL mediante las variables *\$db_host*, *\$db_user* y *\$db_pass* previamente definidas al principio, seleccionamos la base de datos mediante *mysql_select_db()*, creamos una sentencia de SQL segura mediante la función *anti_SQLinjection()* pasándole los datos de las fechas de inicio y final del intervalo (*\$fd* y *\$fh* respectivamente), el tipo de actividad (*\$act*), la sensación de actividad (*\$sact*), la sentencia a sanear y si hemos activado el filtro o no mediante la variable *\$_POST['filtro']*. Seguidamente, hacemos el tratamiento de errores en caso de haber en la conexión y en el lanzamiento de la petición a dicha base de datos. En este punto distinguiremos dos casos de error: el primero es en caso que *mysql_errno()* devuelva cualquier error que no sea al 1065, código para peticiones cuyo resultado sea vacío, que trataremos como un error general y mostraremos en pantalla como error

de ejecución mediante *drupal_set_message()*; en caso contrario, mostraremos un mensaje de estado diciendo que no hay datos para dicha petición con la misma función. En ambos casos abortaremos el resto del código y finalizaremos la ejecución. Una vez la petición se haya lanzado y recibamos un resultado con datos, construimos la tabla en la cual mostraremos los resultados. Para ello usaremos los tags HTML `<table>` para definir la tabla, `<th>` para definir los campos de cabeceras y `<td>` para los campos de datos. El tag `<tr>` lo usamos en cabecera y datos indistintamente para definir filas quedando una fila de la siguiente manera: `<tr><th>variable 1</th><th>variable 2</th></tr>` para los campos de cabecera y sustituyendo `<th>` por `<td>` en caso de las filas de datos. Todo ello lo mostramos por pantalla mediante la instrucción `echo` de PHP. En la parte de las filas cabecera insertamos texto y en las de datos insertamos las variables que contienen el resultado de la petición SQL mediante un bucle que recorra los resultados a través del array *\$fila* que genera la función *mysql_fetch_array()*.

```

if ($c_fallida != 1){
    echo "<table border='0'>";
    echo "<tr><th>Fecha</th>";
    echo "<th>Fichero</th>";
    echo "<th>Hora de inicio</th>";
    echo "<th>Actividad</th>";
    echo "<th>Duración</th>";
    echo "<th>Distancia</th>";
    echo "<th>Sensación de actividad</th>";
    echo "<th>Notas</th></tr>";

    while ($fila = mysql_fetch_array($where_q)){
        echo "<tr><td>" . check_plain($fila['data']) . "</td>";
        echo "<td>" .
        check_plain(substr($fila['Path'],strrpos($fila['Path'],"/")+1)
        ) . "</td>";
        echo "<td>" . check_plain($fila['HoraInici']) . "</td>";
        echo "<td>" .
        check_plain(nombre_actividad($fila['activitat'],$act_to_array)
        ) . "</td>";
        echo "<td>" . check_plain($fila['durada']) . "</td>";
        echo "<td>" . check_plain($fila['distancia']) . "</td>";
        echo "<td>" . check_plain($fila['sensacioActivitat']) .
        "</td>";
        echo "<td>" . check_plain($fila['notes']) . "</td></tr>";
    }
    echo "</table>";
}
mysql_close($con);
}
}

```

Es importante notar que para la inserción del nombre del archivo mediante *\$fila['Path']*, usamos las funciones *strrpos()* para buscar la última posición del separador de directorio / (le sumamos 1 ya que los strings en PHP empiezan desde la posición 0 y queremos evitar mostrar este símbolo por pantalla) y *substr()* para escoger el substring desde dicha posición hasta el final (el primer parámetro es el path completo en el servidor, la segunda es la posición inicial y falta una tercera que designa el final del substring que omitimos en este caso ya que por defecto es el final del string origen). También usamos la función *nombre_actividad()* para devolver el nombre de la actividad pasándole el identificador en la base de datos y la lista de actividades disponibles en el string *\$act_to_array*. Una vez construida la tabla, cerramos la conexión con el servidor y el script finaliza.

Seguidamente describiremos las funciones auxiliares que se usan para el tratamientos de la sentencia SQL y los nombres de el tipo de actividad:

```

function anti_SQLinjection($fdes,$fhas,$act,$sact,$sql,$filtro,$num_param,$uid){
    if($filtro == "ok"){
        switch($num_param){
            case 1:
                if ($fdes != ""){

```

```

        if ($fhas != ""){
            $sql_seguro = sprintf($sql,$uid,
                mysql_real_escape_string($fdes),
                mysql_real_escape_string($fhas));
        }
        else{
            $sql_seguro = sprintf($sql,$uid,
                mysql_real_escape_string($fdes));
        }
    }
    if($sact != ""){
        $sql_seguro = sprintf($sql,$uid,
            mysql_real_escape_string($sact));
    }
    if ($sact != ""){
        $sql_seguro = sprintf($sql,$uid,$sact);
    }
    break;
case 2:
    if($fdes != "" && $sact != "" && $sact == ""){
        if ($fhas != ""){
            $sql_seguro = sprintf($sql,$uid,
                mysql_real_escape_string($fdes),
                mysql_real_escape_string($fhas),
                mysql_real_escape_string($sact));
        }
        else{
            $sql_seguro = sprintf($sql,$uid,
                mysql_real_escape_string($fdes),
                mysql_real_escape_string($sact));
        }
    }
    if($fdes != "" && $sact == "" && $sact != ""){
        if ($fhas != ""){
            $sql_seguro = sprintf($sql,$uid,
                mysql_real_escape_string($fdes),
                mysql_real_escape_string($fhas),$sact);
        }
        else{
            $sql_seguro = sprintf($sql,$uid,
                mysql_real_escape_string($fdes),$sact);
        }
    }
    if($fdes == "" && $sact != "" && $sact != ""){
        $sql_seguro = sprintf($sql,$uid,
            mysql_real_escape_string($sact),$sact);
    }
    break;
case 3:
    if ($fhas != ""){
        $sql_seguro = sprintf($sql,$uid,
            mysql_real_escape_string($fdes),
            mysql_real_escape_string($fhas),
            mysql_real_escape_string($sact),$sact);
    }
    else{
        $sql_seguro = sprintf($sql,$uid,
            mysql_real_escape_string($fdes),
            mysql_real_escape_string($sact),$sact);
    }
    break;
}
}
else{
    $sql_seguro = sprintf($sql,$uid);
}
return $sql_seguro;
}

```

La función `anti_SQLinjection()` pretende sanear todo el código SQL de posibles ataques de inyección SQL. Para ello debemos saber primero de todo si el filtro se activa en esta búsqueda o no mediante `$filtro`. En caso negativo, se sana la sentencia mediante el identificador de usuario `$uid` y se devuelve el resultado. Para sanear siempre usaremos `sprintf()` ya que hemos preparado la variable `$sql` que contiene la sentencia SQL mediante los valores `%s` para cadenas de caracteres y `%d` para numerales de tal forma que podamos aplicar `mysql_real_escape_string()` a las variables que sean de tipo cadena de caracteres, para evitar posibles inyecciones de código malicioso. Es importante el orden de valores en la sentencia, con lo cual desgranaremos los casos posibles mediante el recuento del usos de filtros que contiene `$num_param` y dividiremos en tres casos según sean 1, 2 o 3 parámetros. Para cada uno de estos casos distinguiremos todas las combinaciones posibles para la fecha de inicio (`$fdes`), el tipo de actividad (`$act`) o la sensación de actividad (`$sact`) por este orden, ya que las sentencias se han construido previamente siguiendo dicho orden. En los casos que haya valores para `$fdes`, se comprobará inmediatamente que también haya una fecha final (`$fhas != ""`) y si hay, se tratará el valor como el resto quedando el orden como se construyó la sentencia SQL.

```
function nombre_actividad($id,$actividades){
    $aa = explode(";", $actividades);
    $i = 0;
    $found = FALSE;
    while($i < count($aa) && !$found){
        $aux = explode("_", $aa[$i]);
        if ($aux[0] == $id) {
            $nombre = $aux[1];
            $found = TRUE;
        }
        $i++;
    }
    return $nombre;
}

?>
```

Por último, la función `nombre_actividad()` desgrana el string `$actividades` que recoge el identificador de actividad y su nombre en el siguiente formato: `número actividad 1_nombre actividad1;número actividad 2_nombre actividad 2`; etc. De esta forma, mediante la función `explode()` que convierte un string en un array mediante un substring que haga de separador entre posición y posición del array, convertimos `$actividades` en `$aa` donde `$aa[0]` es `número actividad 1_nombre actividad1`, `$aa[1]` es `número actividad 2_nombre actividad 2`, etc. A través del símbolo “;”. Una vez separadas las parejas identificador-actividad correspondientemente, buscamos mediante un bucle el identificador que nos ha sido pasado como parámetro a través de `$id` usando `explode()` pero esta vez con el carácter “_” con tal de separar identificador y nombre de actividad. Al encontrarlo, salimos del bucle y devolvemos el nombre de la actividad mediante la variable `$nombre`.

Añadir nuevos campos en el formulario.

Para añadir nuevos campos e información en el formulario de entrada podemos seguir los siguientes pasos:

Añadir el tag correspondiente en HTML en el fragmento de definición del formulario. Podemos escoger entre:

Recuadro de texto:

```
(Nombre) <br/><input name="(nombre)" type="text" size="(tamaño)" maxlength="(número de caracteres)"/>
```

Desplegable:

```
(Nombre) <br/> <select name="(nombre)" type="text">
  <option value="(Valor opcion 1)">(Nombre opcion 1)</option>
  <option value="(Valor opcion 2)">(Nombre opcion 2)</option>
  <option value="(Valor opcion 3)">(Nombre opcion 3)</option>
  ...
</select>
```

Posteriormente, deberemos introducir el siguiente fragmento en la comprobación de errores de los campos:

```
if { $_POST["(nombre)"] != "" } {
  $(nombre_variable_campo) = $_POST["(nombre)"];
}
else {
  drupal_set_message("No hay (Nombre del campo) <br />", 'error!');
  $ok = 0;
}
```

Y finalmente en la sentencia *INSERT SQL*:

```
$sql=sprintf("INSERT INTO log
(IdentificadorUsuari,data,Path,uid,HoraInici,activitat,durada,distancia,notes,sensacioActivi
tat, nombre_campo_nuevo) VALUES ('%s','%s','%s',%d,'%s','%s',%d,%d,'%s',%d,
%*)",mysql_real_escape_string($dni), mysql_real_escape_string($fecha_sql),
mysql_real_escape_string($ruta), $uid,
mysql_real_escape_string($horaini), mysql_real_escape_string($actividad), $duracion,
$distanca, mysql_real_escape_string($notas), $sensact, **);
```

Donde * es %s o %d dependiendo si es texto o numérico y ** my_real_escape_string(\$nombre_variable_campo) o \$nombre_variable_campo respectivamente.

En el caso del script de listado de logs, para listar más campos de la tabla *log* solamente deberíamos añadir lo siguiente cuando construyamos por primera vez la sentencia SQL:

```
$sql="SELECT data,Path,HoraInici,activitat,durada,distancia,sensacioActivitat,
notes,nuevo_campo FROM log WHERE uid = '%d'";
```

Y posteriormente en la creación de la tabla en la cabecera:

```
echo "<th>Sensación de actividad</th>";
echo "<th>(nuevo campo)</th>";
echo "<th>Notas</th></tr>";
```

y en las líneas de datos:

```
echo "<td>" . check_plain($fila['sensacioActivitat']) . "</td>";  
echo "<td>" . check_plain($fila['nuevo_campo']) . "</td>";  
echo "<td>" . check_plain($fila['notes']) . "</td></tr>";
```

Si quisieramos listar más opciones de filtro se debería reestructurar gran parte del script, sobre todo en la función *anti_SQLinjection()*, con lo cual no es posible aplicar directrices simples para su modificación.

4. AMPLIACIONES

El portal web presentado ofrece muchas y diversas opciones de presentación de cara al usuario y un abanico de posibilidades a la hora de añadir nuevas funcionalidades al portal, extremadamente grande. Al dimensionar los objetivos de este proyecto, se previó la posibilidad en un futuro de añadir ampliaciones de manera que no hiciera falta reestructurar todo el trabajo y el núcleo del proyecto pudiendo aprovechar todas las partes de las cuales dispone dicho portal. Esta razón fue además, una de las más importantes a la hora de escoger el CMS Drupal para su desarrollo dada su tremenda modularidad y su vasta comunidad de desarrollo y soporte.

Lamentablemente, en el transcurso de este proyecto han surgido una serie de problemas que han disminuido el tiempo de desarrollo y a causa de los cuales no ha sido posible la implementación de algunos servicios propuestos tales como la integración del CMS con el paquete de software estadístico R a través de R-php. He aquí una introducción de dichas ampliaciones:



R: Es un lenguaje y entorno de programación para análisis estadístico y gráfico. Se trata de un proyecto de software libre, resultado de la implementación GNU del premiado lenguaje S. R y S-Plus -versión comercial de S- son, probablemente, los dos lenguajes más utilizados en investigación por la comunidad estadística, siendo además muy populares en el campo de la investigación biomédica, la bioinformática y las matemáticas financieras. A esto contribuye la posibilidad de cargar diferentes librerías o paquetes con finalidades específicas de cálculo o gráfico.

R-php: Es una herramienta desarrollada por los profesores Alfredo Pontillo y Angelo Mineo del Departamento de Ciencias Matemáticas y Estadísticas Silvio Vianelli de la Universidad de Palermo, Italia. Consiste en un servidor de R a través de un Apache/MySQL que permite introducir datos por línea de comandos a través del servidor y recibir una salida de datos de R a través del mismo. La salida se puede representar de varias maneras, las cuales incluyen datos en bruto o gráficos y otras utilidades como un archivo en formato pdf con cualquiera de las dos anteriores modalidades de salida. También incorpora un interfaz gráfico que permite gestionar los datos más cómodamente. El software se encuentra registrado bajo una licencia GPL.

La integración dentro de una página escrita en php se llevó a cabo por otro proyectista en un proyecto paralelo a este llamado WEB PER A UN CLUB D'ATLETISME AMB INCORPORACIÓ D'ANÀLISI ESTADÍSTIC R desarrollado por Adrián Pérez Barbosa, también alumno de la FIB. En este proyecto se ha llegado a contruir la integración de R-php i una página escrita en el mencionado lenguaje con éxito. Se pretendía, una vez sabiendo que la integración era posible, hacer lo mismo con el CMS Drupal mediante la modularidad que éste ofrece y ya que Drupal está escrito en PHP, es una garantía. La idea original era usar los famosos Snippets en página de contenido de la misma forma que hemos desarrollado la subida de ficheros de pulsómetro al servidor, mostrando una serie de datos obtenidos a través de una base de datos y la interpretación de los archivos de pulsómetro disponibles en el servidor, para generar unos datos de salida a través de R mediante R-php. Esto no se ha podido llevar a cabo a causa de los problemas surgidos en el servidor con la base de datos central los cuales han tardado tiempo en solucionarse. También han surgido problemas de compatibilidad en los snippets de Drupal y el servicio de R-php, lo cual ha agotado el tiempo de desarrollo. No obstante, previendo la imposibilidad de desarrollar la integración en este proyecto y creyendo más cierta dicha integración a través de un módulo personalizado para Drupal, ya que de esta forma se usan directamente elementos del sistema de nuestro CMS de desarrollo y se

pueden ajustar nuestras necesidades proporcionando una mayor fiabilidad y una integración mucho más robusta y eficiente; se ha creído oportuno desarrollar una guía para el desarrollo de módulos y la configuración de un IDE que integre Drupal 6 para mejor comodidad en dicho desarrollo. Todo ello está documentado en los anexos I y II. Además, el código implementado para la subida y listado de la información de archivos de pulsómetro, se puede usar como base para el desarrollo de nuevos módulos para Drupal que amplíen las funcionalidades estadísticas de la página así como la integración de R con el CMS.

Además, esta guía de desarrollo permitirá añadir de esta forma, nuevas funcionalidades no previstas en este proyecto que cada uno de los usuarios finales estime oportuno aplicar y/o desarrollar dentro del portal web proporcionado así como en desarrollo de temas personalizados en la presentación del contenido.

A parte del desarrollo de aplicaciones para el portal, también es posible desarrollar nuevas formas de presentar el contenido y estructurar la página a través del uso del módulo Vistas (*Views*) que describimos en el capítulo 3. Así el usuario final, podrá acabar de adaptar el portal a sus necesidades y gustos específicos sin necesidad de programar en PHP.

5. CONCLUSIONES

A través de DISSENY I IMPLEMENTACIÓ WEB PER A UNA ESCOLA D'ARTS MARCIALS se ha desarrollado un portal genérico a través del cual un usuario puede gestionar todas las opciones del portal sin la necesidad de poseer conocimientos avanzados en es diseño y desarrollo de aplicaciones web 2.0, es decir, páginas web dinámicas que ofrecen aplicaciones que se ejecutan en un servidor y cuyo resultado, se muestra al usuario en tiempo real. Aunque está diseñado para una escuela de artes marciales, también se posible modificar el contenido para adaptarlo a cualquier tipo de asociación o comunidad.

Tal y como describimos con más amplitud en el capítulo 2 de esta memoria, se barajaron varias opciones en la fase de dimensionado del portal entre las que cabe destacar la integración de un motor 3D a modo de presentación para las diferentes movimientos y formas que un arte marcial posee, mostrando desde diversos ángulos y diversas vistas la consecución de dichas formas y movimientos para su mejor comprensión. Sin embargo, quedó descartada dada su compleja y costosa implementación en el portal, y dado que dicha parte tendría un peso mucho mayor en importancia dentro del proyecto, se prefirió usar un sistema de vídeo alojado en el servidor para mostrar dichas formas. Más tarde, se decidió darle un nuevo giro al diseño de los objetivos del proyecto simplificando las tareas de gestión del portal para que usuario sin conocimientos avanzados pudieran gestionar el portal fácilmente. A partir de este punto, se consolidó la decisión de usar Drupal como vehículo de desarrollo gracias a su robustez, compatibilidad, extensibilidad y facilidad de desarrollo de aplicaciones futuras.

Otra parte importante a destacar en la fase de dimensionado, fue la elección de aplicaciones y servicios que estuvieran exentos de licencias restrictivas para el uso y modificación en la posterior fase de desarrollo. Se pensó en las alternativas ofrecidas por las licencias GNU y Creative Commons en materia de software y se inició una búsqueda de material compatible con nuestros objetivos dentro de dichas licencias. Unos de los motivos adicionales para la elección de Drupal fue que funciona bajo una licencia GPL, lo cual no hacía sino confirmar su uso. Todas las aplicaciones y servicios aquí usados se rigen por dichas licencias y permiten siempre bajo unos parámetros poco restrictivos, su libre uso y adquisición, y en consecuencia, su coste tremendamente reducido estando al alcance de todo aquel que lo desee.

Pese a que la fase de desarrollo del portal no ha tenido un gran componente de desarrollo de software propiamente dicho, sí que ha generado un gran volumen de tareas de administración de sistemas tales como el dimensionado del soporte físico, el conjunto de aplicaciones auxiliares para el portal y sus funcionalidades. Entre las tecnologías usadas, cabe destacar el funcionamiento del webserver Apache, la gestión de bases de datos SQL a través de un servidor MySQL, la instalación y configuración del codificador FFMPEG de video y audio a partir de binarios, la adecuación del portal a sistemas Unix y Windows, y el estudio de las amplias posibilidades que ofrece el CMS Drupal y su posterior descripción en esta memoria en forma de guía de usuario. A través de dicha guía, elaborada en esta memoria, se describe cómo funcionan las diversas partes del CMS, qué se puede encontrar por defecto en la instalación, cómo instalar el portal en varia plataformas y sus servicios auxiliares, como gestionar los usuarios y función que puede desempeñar cada uno de ellos, cómo gestionar el contenido y crear nuevo, etc. Se ha querido buscar un lenguaje sencillo para describir los diferentes apartados y la extensa gama de opciones de autogestión, de modo que es usuario final pueda entender fácilmente su funcionamiento. Cabe reseñar también, la confección de los anexos para la instalación de FFMPEG en sistemas Devian Etch y el manual de desarrollo introductorio de módulos en Drupal 6, con lo cual asentamos una base para futuras ampliaciones mayores del sitio y su uso en otro tipo de proyectos.

Dentro del desarrollo de software, hemos aprendido a usar el lenguaje PHP, célebre en la comunidad de desarrolladores de aplicaciones web de código abierto, para escribir una sencilla aplicación que permita subir ficheros que detallen los datos físicos de una sesión de

entrenamiento de cualquier arte marcial, y reportar información adicional sobre dicha sesión. Posteriormente, se puede consultar la información reportada mediante un filtro de búsquedas por fecha y tipo de actividad, para cada usuario que utilice que se haya registrado en el portal, es decir, cualquier miembro de la asociación. Aunque la aplicación es sencilla, cuenta con la restricción adicional de estar incrustada dentro de un sistema de gestión de contenido que aunque administra dicho contenido mediante rutinas en PHP, es necesario usar dicho programa dentro de unos parámetros predefinidos si queremos que funcione correctamente.

Como en todo proyecto de esta magnitud, los ensayos de prueba y error en las diferentes tecnologías que son compatibles con este portal, han sido esenciales para su buen funcionamiento y también han consumido gran parte del tiempo dedicado a su correcta consecución. Partíamos con unos conocimientos sobre la materia casi nulos y la búsqueda de información sobre dichas tecnologías y tutoriales para el funcionamiento y personalización de los servicios, así como la solventación de errores generados durante dicho proceso ha sido extensa y laboriosa, consultando material principalmente en Internet y en la comunidad de desarrollo de Drupal. Hemos tenido crecientes problemas en el buen funcionamiento de la base de datos a causa de incompatibilidades en sistemas Windows con el servidor MySQL que gestionaba la información necesaria para su uso del portal web, además de la incompatibilidad de algunas opciones del servidor de peticiones al software estadístico, así como la instalación de los códecs y librerías de condificación necesarias para que FFMPEG funcionase en cualquier plataforma de sistemas operativos anteriormente mencionada, ya que no es una instalación fácil ni amigable, ni siquiera para alguien con conocimientos avanzados en materia de administración de sistemas y mucho menos para un usuario estándar del sistema.

También es importante apuntar el aprendizaje adquirido en los pasos de desarrollo del proyecto: cómo distribuir la carga del trabajo durante al hacer una previsión de las tareas a realizar, ajustar las fases de desarrollo dentro del tiempo disponible, la gestión de recursos disponibles durante dicho periodo y el constante *feedback* entre el cliente, en este caso el director de proyecto, y el desarrollador, en este caso el alumno. Se ha conseguido completar la mayoría de los objetivos previstos y se ha proporcionado una alternativa para los que no se han podido llevar a cabo, para facilitar su correcta consecución en posibles ampliaciones.

En definitiva, DISSENY I IMPLEMENTACIÓ WEB PER A UNA ESCOLA D'ARTS MARCIALS proporciona una acercamiento al desarrollo y uso de portales comunitarios en internet a usuarios sin conocimientos avanzados sobre el desarrollo de software, mediante un portal web para asociaciones de artes marciales y una guía de uso y administración de dicho portal, además de asentar bases para el desarrollo de futuros proyectos en Drupal en la investigación estadística relacionada con el deporte y actividades deportivas.

6. BIBLIOGRAFÍA

Drupal

Página principal de Drupal. Dryes Buytaert. <http://drupal.org>
The Drupal cookbook (for beginners). <http://drupal.org/node/120612>
PHP page Snippets. <http://drupal.org/node/23220>
How to handle text in a secure way. <http://drupal.org/node/28984>
Writing secure code. <http://drupal.org/writing-secure-code>
API de Drupal 6. <http://api.drupal.org/api>
Display a table of node entries sorted by a start date taken from the event module. <http://drupal.org/node/231239>
Creating Modules - a tutorial. <http://drupal.org/node/82920>
Localizer: Building multilingual sites. <http://drupal.org/node/103419>
Content Construction Kit (CCK). <http://drupal.org/project/cck>
Views. <http://drupal.org/handbook/modules/views>, <http://drupal.org/project/views>
Organig Groups. <http://drupal.org/project/og>
Event. Aaron Welch, Gerhard Killesreite. <http://drupal.org/project/event>
Image. <http://drupal.org/project/image>
Flashvideo. Travis Tidwell. <http://drupal.org/project/flashvideo>, <http://www.travistidwell.com/flashvideo>
OG Calendar. Paul Booker, http://drupal.org/project/og_calendar
Token. <http://drupal.org/project/token>
Advanced Help. http://drupal.org/project/advanced_help
Artículo sobre Drupal en la Wikipedia. <http://es.wikipedia.org/wiki/Drupal>

Pro Drupal Development. John K. VanDyk, Matt Westgate. Apress, 2007. ISBN: 978-1-59059-755-2

NetBeans

Página oficial de NetBeans. <http://www.netbeans.org/>
NetBeans IDE PHP Quick Start Tutorial. <http://www.netbeans.org/kb/docs/php/quickstart.html>
Configuring PHP Development Environment in Windows. <http://www.netbeans.org/kb/docs/php/configure-php-environment-windows.html>
Página oficial del proyecto Drupal 6 Netbeans 6 IDE. <https://nbdrupalsupport.dev.java.net/>

Drupal 6.x support on NetBeans 6.x IDE. Sujit Nair. 29 de Marzo 2008.

Download, Install and use NetBeans plugins for Drupal support. Sujit Nair, Manisha Birajdar. 25 de Mayo 2008.

PHP y HTML

Página oficial de PHP. <http://es.php.net/>
Artículo sobre PHP en la Wikipedia. <http://es.wikipedia.org/wiki/Php>
Tutoriales de PHP. <http://www.w3schools.com/PHP/default.asp>
Tutoriales de HTML. <http://www.w3schools.com/html/>

FFMPEG

Página oficial de FFMPEG. <http://www.ffmpeg.org/>
Artículo sobre FFMPEG en la Wikipedia. <http://es.wikipedia.org/wiki/FFMPEG>
Manual de uso de FFMPEG. <http://ffmpeg.mplayerhq.hu/ffmpeg-doc.html>
Distribución de FFMPEG en Windows. <http://www.videohelp.com/tools/ffmpeg>
Bleeding edge FFMPEG on Ubuntu Feisty. Paul Battley. <http://po-ru.com/diary/bleeding-edge-ffmpeg-on-ubuntu-feisty>

How to install FFMPEG in Debian Etch. PhillC. <http://slashdot.org/~PhillC/journal/190325>

JW Player

Página oficial de JW Player. Jeroen Wijering. <http://www.longtailvideo.com/players/jw-flv-player/>

XAMPP

Página oficial de XAMPP. <http://www.apachefriends.org/en/xampp.html>

Artículo sobre XAMPP en la Wikipedia. <http://es.wikipedia.org/wiki/XAMPP>

MySQL Error Code 17? <http://cfsilence.com/blog/client/index.cfm/2008/5/21/MySQL-Error-Code-17>

R i R-php

Página oficial de R. <http://www.r-project.org/>

Artículo sobre R en la wikipedia. R-project. http://es.wikipedia.org/wiki/Lenguaje_R

Página oficial de R-php. Alfredo Pontillo, Angelo Mineo. <http://dssm.unipa.it/R-php/>

Web per a un club d'atletisme amb incorporació d'anàlisi estadística R. Pérez Barbosa, Adrià. 2009.

Otros

Costes del desarrollo de software. <http://www.tufuncion.com/desarrollo-software>

Artículo sobre COCOMO en la Wikipedia. <http://es.wikipedia.org/wiki/COCOMO>

Salario medio de un programador. <http://www.tufuncion.com/trabajo-programador>

Infojobs Trends salarios. <http://salarios.infojobs.net/>

BOE numero 281. 21 Noviembre 2008. <http://www.boe.es/boe/dias/2008/11/21/pdfs/A46851-46852.pdf>

7. AGRADECIMIENTOS.

Hay mucha gente a la que agradecer que este proyecto sea posible. Si alguien cree que debo agradecerle algo, que me envíe un correo y le daré las gracias a través de una nota ;)

Manel Martínez Torres, por hacerme un sitio en www.strongestprogrammer.com y por la ayuda prestada en las ideas.

Daniel Ramírez Guerrero, por ponerme en ruta con Drupal y su coche.

Xavier Francisco Gilabert, por darme pistas sobre Linux y dejarme el gran y famoso libro de Drupal y el de cuarta edición.

Pep Ciureneta, por la ayuda prestada sobre Linux, por hacerme un hueco en LinuxUPC y por las patatas bravas.

A mi tutor, Pau Fonseca i Casas, por facilitarme las herramientas para el desarrollo y agilizar los trámites del proyecto.

Albert Muntané y Albert Donaire, por sus consejos sobre PHP y SQL.

A toda la gente que altruístamente publica guías y tutoriales en la red.

A toda la comunidad de desarrollo de Drupal por su increíble cantidad de material útil y el soporte en línea.

Al CRUPC por los buenos ratos.

A la wii que no he podido tocar a penas.

Al gimnasio que no he pisado.

A todo el staff de www.undeadmonkeys.com

Y finalmente, pero no menos importantes:

A María Arrogante Mesas, por sus consejos sobre introducciones y conclusiones, por su perspicacia en los detalles y por hacerle pasar las navidades más aburridas de la historia y querer seguir aguantándome.

A mis padres y mis hermanos por tener que soportar todas mis tonterías y mis cabreos.

“Don’t Panic.”

- Hitchhiker’s Guide to the Galaxy.

ANEXO I: NetBeans 6.5 y Drupal 6.

Para desarrollar aplicaciones en el entorno de programación de Drupal es muy útil usar un IDE en el cual podamos mantener la depuración de los posibles errores controlada y fuera del entorno real. Para ello, están disponibles una serie de plugins de NetBeans para el desarrollo en Drupal 6. En esta sección explicaremos como configurar el IDE y los plugins para empezar a desarrollar módulos y temas personalizados. He aquí los pasos a seguir:

1. Instalar XAMPP: Suponemos que el usuario ya posee una instalación de este paquete en el cual hace correr los archivos de Drupal de este proyecto.
2. Editamos `php.ini` de la instalación de XAMPP (`.\xampp\apache\bin\php.ini`). Debemos desactivar las extensiones Xend y activar el soporte par XDebug que nos permitirá la depuración de errores. Dejamos las líneas siguientes de esta forma:

```
[Zend]
zend_extension_ts = ".\xampp\php\zendOptimizer\lib\ZendExtensionManager.dll"
zend_extension_manager.optimizer_ts =

".\xampp\php\zendOptimizer\lib\Optimizer"
zend_optimizer.enable_loader = 0
zend_optimizer.optimization_level=15
zend_optimizer.license_path =
; Local Variables:
; tab-width: 4
; End:

[XDebug]
;; Only Zend OR (!) XDebug
zend_extension_ts=".\xampp\php\ext\php_xdebug.dll"
xdebug.remote_enable=true
xdebug.remote_host=127.0.0.1
xdebug.remote_port=9000
xdebug.remote_handler=dbgp
;xdebug.profiler_enable=1
;xdebug.profiler_output_dir=".\xampp\tmp"
```

Es importante notar que “.” se refiere al directorio de instalación de XAMPP en nuestro disco duro.

3. Bajamos NetBeans 6.5 para el desarrollo de PHP desde <http://www.netbeans.org> e instalamos el ejecutable.
4. Ejecutamos NetBeans 6.5 y configuramos las extensiones en *Tools* → *Options* → *Miscellaneous* → *Files* añadiendo *module* e *install* como ficheros PHP5. Es importante no añadir un punto delante de las extensiones (Ej: `.module`).
5. Bajar desde <https://nbdrupalsupport.dev.java.net> → *plugin* el archivo `nbdrupalsupportv1.1`
6. Debemos instalar el prugin para NetBeans en *Tools* → *Plugins* → *Downloaded*. Mediante en botón "Browse" seleccionamos el anterior archivo y pulsamos el botón *Install*. Nos aparecerán una serie de advertencias sobre la fiabilidad del plugin. Aceptamos y continuamos con la instalación.

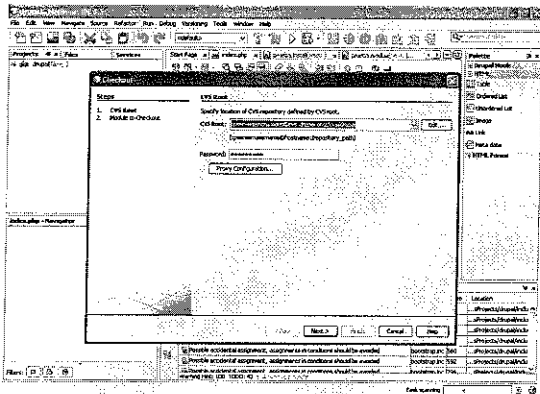


Figura Anexo I-1. Versioning → CVS → Checkout.

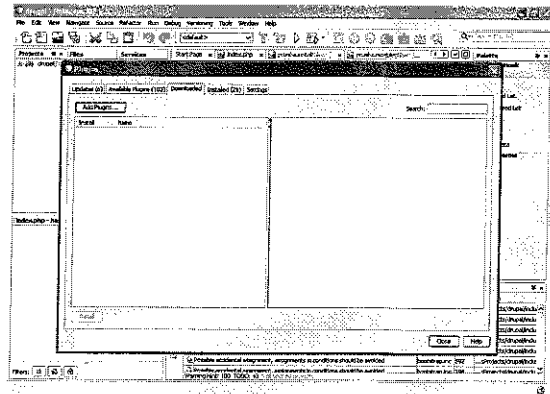


Figura Anexo I-2. Tools Plugin Downloaded.

7. Entramos en las opciones de menú *Versioning* → *CVS* → *Checkout* en Netbeans e insertamos en el recuadro de texto la siguiente línea: `:pserver:anonymous@cvs.drupal.org:/cvs/drupal` y en la casilla de Password escribimos `anonymous` y le damos al botón *Next*. Ahora debemos insertar un nombre para el proyecto. Nosotros hemos preferido insertar `drupal_test` para mantener clara la parte de desarrollo y test.

8. Creamos un nuevo proyecto mediante la opción *PHP Application with existing sources*. En la siguiente pantalla seleccionamos el directorio de drupal previamente creado, es decir, el nombre del proyecto. Normalmente estará en `Mis Documentos\NetBeansProjects\drupal_test`. Escogemos `localhost` como servidor de desarrollo y la opción *Copy files from Sources to another location* e introducimos el path base de las aplicaciones web (`./xampp/htdocs/xampp`). Aceptamos.

9. Creamos la BD de desarrollo en phpmyadmin como se describe en la instalación del portal web de este proyecto. Por ejemplo `drupal_test`.

10. Seleccionamos la pestaña *Services*, la opción *Databases* y pulsamos en *drivers* → *MySQL* con el botón derecho y seleccionamos la opción *Connecting Using* e introducimos los siguiente en los diferentes campos:

Host: localhost
 Port: 3306
 Database: drupal_test
 User Name: root
 Password: (contraseña de root)

En este procedimiento hemos escogido los datos de ejemplo anteriormente mencionados en los pasos previos.

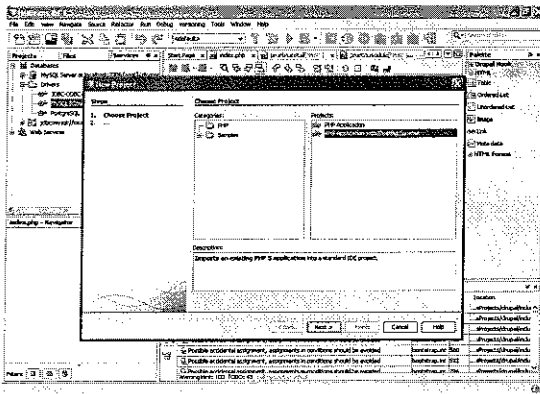


Figura Anexo I-3. Creación de un nuevo Proyecto.

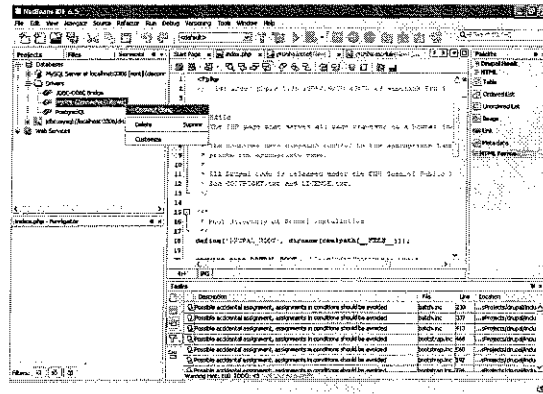


Figura Anexo I-4. Crear conexión con la base de datos.

11. Ejecutamos el proyecto con la opción *Run* e instalamos Drupal a través del proceso que marca la página usando *localhost* y *drupal_test* como base de datos y *root* y la contraseña correspondiente, como usuario de dicha base de datos.

12. Finalmente, de vuelta a NetBeans 6.5 para crear un nuevo módulo o tema de drupal, podemos ir a: *File* → *New File* y seleccionar *Other* en la pantalla de creación. Aquí podemos decidir si queremos crear un módulo o un tema. En caso del módulo podemos añadir *hooks* del sistema para nuestro módulo a través de la ventana *Palette* → *Drupal Forms* la cual nos mostrará los *hooks* de sistema personalizables. Para el tema no es necesario hacer nada más.

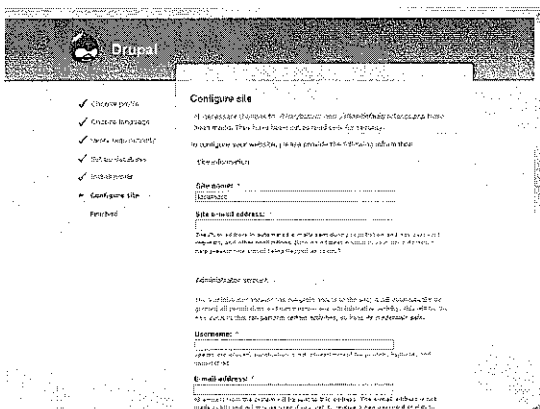


Figura Anexo I-5. Instalación de Drupal.

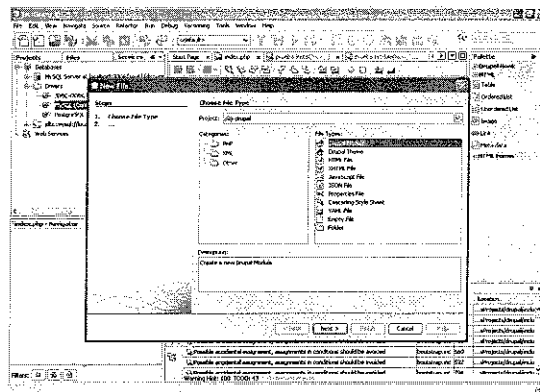


Figura Anexo I-6. Creación de módulo o tema personalizado.

ANEXO II: Introducción al desarrollo de aplicaciones en Drupal 6.

En muchas aplicaciones de código abierto podemos personalizar la aplicación modificando el código fuente. Aunque este método es una manera eficaz de conseguir que la aplicación haga lo que deseamos, es una de las últimas opciones a tener en cuenta dentro de la comunidad de desarrollo de Drupal, ya que con cada actualización, se ha de volver a aplicar los cambios en las partes modificadas y como podemos observar, no es un buen método. Para ello, es posible crear módulos propios que se ajusten a nuestras necesidades y que modifiquen el comportamiento de nuestro sitio sin modificar ningún elemento básico o de otros módulos, ya que Drupal está diseñado para ser tremendamente modular y extensible.

Este capítulo muestra cómo crear un módulo desde cero. Para ello elaboraremos un módulo que permita añadir anotaciones en una página web en adición a los comentarios, las cuales pueden ser útiles para revisiones personales de contenido.

Creando los ficheros.

Lo primero que debemos hacer es escoger el nombre del módulo. El nombre "anotaciones" parece apropiado ya que es corto y descriptivo. Lo siguiente es colocar el módulo en el lugar adecuado (*sites/all/modules*) y para ello crearemos un directorio que contenga los ficheros del módulo ya que es de esta manera que Drupal reconoce los módulos adicionales. Creamos el fichero con el nombre *anotaciones.module*, otro llamado *README.TXT* para explicar a los usuario qué funcionalidades posee nuestro módulo contribuido y *anotaciones.info* para proporcionar información a Drupal sobre nuestro módulo.

Empezaremos añadiendo a *anotaciones.info* lo siguiente:

```
; $Id$
name = Anotaciones
description = Permite a los usuario anotar módulos.
package = Ejemplo
version = "$Name$"
```

El fichero es un tipo *.ini*, un tipo simple estándar de archivos de configuración de PHP (http://php.net/parse_ini_file para más señas). Empezaremos con un tag para versiones concurrentes del sistema (CVS) y proveeremos un nombre y una descripción a Drupal para mostrar en la sección de administración de módulos del sitio. Los módulos se presentan en grupos y la agrupación se determina mediante *package*. Por ello, si poseemos varios módulos que posean *package = Ejemplo*, se mostrarán en un solo grupo. El valor para *version* es otro tag identificador CVS. Si queremos compartir nuestro módulo con otros mediante la comprobación del repositorio de módulos contribuidos de Drupal, este valor automáticamente se rellenará con ello.

Es importante generar este fichero *.info* ya que cuando la página de administración de módulos se carga, el hecho de tener un fichero de este tipo con las directrices anteriores reduce considerablemente el gasto de recursos por parte del sistema de Drupal.

Entrando de lleno en la creación de nuestro módulo. Creamos el fichero *anotaciones.module* en nuestro directorio *anotaciones* y empezamos el fichero abriendo un tag PHP y otro para identificación CVS seguidos de un comentario:

```

<?php
// $Id$
/**
 * @file
 * Permite añadir anotaciones a los usuarios.
 *
 * Añade un campo de texto cuando un nodo no se muestra
 * de tal manera que los usuarios autenticados puedan crear notas
 */

```

Varias cosas. La primera, empezamos comentando con `/**` y en cada subsiguiente línea usamos un asterisco con un espacio (`*`) y `*/` en una línea posterior al fin del comentario. La segunda, el token `@file` implica que lo que sigue en la siguiente línea es la descripción de lo que hace el fichero. Es una manera estándar para describir el fichero según la comunidad de desarrollo de Drupal para que el programa encargado de automatizar la documentación pueda extraer la información y darle formato. Después añadimos una descripción para los programadores que modifiquen o mejoren nuestro código. Es importante saber que no hemos añadido el tag `?>` para cerrar el código ya que esto puede ocasionar problemas en el proceso de interpretación de los ficheros (para más señas, consultar: <http://drupal.org/node/545>). Es importante que usemos este formato para escribir módulos ya que es la manera estándar usada para el desarrollo de aplicaciones en Drupal.

Guardamos los cambios en el fichero y ahora podremos consultar la aparición de nuestro módulo en la página de administración de módulos. Ahora será necesaria la implementación de un *hook* apropiado, necesitamos un path para ajustar las preferencias y crear el formulario de preferencias.

Implementando un Hook

Recordemos que Drupal se ha construido a través de sistema de *hooks*, también llamados *callbacks*. Drupal pregunta a los módulos si quieren hacer algo y estos, en caso de ser necesaria alguna acción por su parte, usarn un *hook* personalizado para llevarla a cabo. Por ejemplo, al determinar qué módulo es responsable de la petición actual, se pide a todos los módulos que proporcionen los paths correspondientes que los módulos necesitan. Esto se lleva a cabo mediante la creación de una lista de todos los modulos y llamando a la función que lleva el nombre del módulo más el sufijo `_menu` para cada módulo. En nuestro caso, sería `anotaciones_menu()`. También puede ser necesario el paso de un parámetro el la función que determina si la respuesta se usará o no. Normalmente, los elementos de un menú se usan siempre.

He aquí lo que escribiremos en nuestro archivo:

```

/**
 * Implementacion de hook_menu().
 */
function anotaciones_menu($may_cache) {
  $items = array();
  if ($may_cache) {
    $items[] = array(
      'path' => 'admin/settings/anotaciones',
      'title' => t('Configuración de Anotaciones'),
      'description' => t('Cambia el comportamiento de las anotaciones.'),
      'callback' => 'drupal_get_form',
      'callback arguments' => array('anotaciones_admin_settings'),
      'access' => user_access('administrar configuracion del sitio')
    );
  }
  return $items;
}

```

El código quiere decir más o menos lo siguiente:

“Cuando el usuario acceda a `http://example.com/?q=admin/settings/ anotaciones`, llama a la función `drupal_get_form()` y pásale el ID del formulario `anotaciones_admin_settings`.”

A la hora de mostrar el formulario, Drupal pedirá una definición de éste (en breve explicaremos esto). Al acabar de preguntar a los módulos por sus elementos de menú, el sistema posee un formulario de menú en el cual se puede seleccionar la función deseada para el path de la petición. Es importante notar que cualquier texto que se muestre al usuario debe ir dentro de la función `t()`, ya que permite la traducción de la cadena de caracteres haciendo que la localización de nuestro texto sea mucho más sencilla.

Añadiendo Ajustes específicos del módulo.

Drupal posee varios tipos de nodos como historias y páginas. Para el uso de anotaciones, queremos restringir el uso a los últimos mencionados. Para hacerlo, necesitamos crear una página donde podamos decirle a nuestro módulo que tipos de nodos queremos anotar, añadiendo lo siguiente a nuestro fichero `anotaciones.module`:

```
/**
 * Definimos el formulario de preferencias.
 */
function anotaciones_admin_settings() {
  $form['anotaciones_nodetypes'] = array(
    '#type' => 'checkboxes',
    '#title' => t('Los usuarios pueden anotar estos tipos de nodos'),
    '#options' => node_get_types('names'),
    '#default_value' => variable_get('anotaciones_nodetypes', array('story')),
    '#description' => t('Un campo de texto estará disponible en estos tipos para crear notas específicas de usuario.'),
  );
  $form['array_filter'] = array('#type' => 'hidden');
  return system_settings_form($form);
}
```

Los formularios en Drupal se representan como una estructura de árbol anidado, o sea, un array de arrays. Esta estructura describe al mecanismo de renderización de formularios de Drupal como debe ser representado dicho formulario. Para una mejor lectura, hemos colocado cada elemento del array en su propia línea. Cada directiva va precedida de un símbolo `#` y actúa como una clave de array. Hemos empezado declarando el tipo de elemento de formulario como casillas de selección (checkboxes), lo que significa que varias casillas de selección se construirán usando un array con clave. Le damos un título al formulario a través de la función `t()` como es costumbre hacer. Agregamos las opciones en la salida de la función `node_get_types('names')`, que devuelve un array en clave de los tipos de nodo que están disponibles en esta instalación de Drupal. La salida de `node_get_types('names')` sería algo parecido a:

```
'page' => 'Page', 'story' => 'Story'
```

Las claves del array son los nombres internos de Drupal para los tipos de nodos, con los nombres más claros (aquellos mostrados al usuario) a la izquierda. En nuestro formulario web, se generarán las casillas de selección para *página* e *historia*.

La siguiente directriz es `#default_value`, que será el valor por defecto para este elemento del formulario. Dado que las casillas de selección son elementos múltiples ya que puede haber más de una casilla, el valor para `#default_value` será un array. Comentemos el valor de esta directiva con más detalle:

`variable_get('anotaciones_notetypes', array('story'))` Drupal permite a los programadores guardar y consultar cualquier dato especial mediante `variable_get()` y `variable_set()`. Los valores se guardan en las variables de la tabla en la base de datos y están disponibles en cualquier momento del proceso de una petición. Ya que estas variables se obtienen de la base de datos de esta forma, no es buena idea guardar una cantidad grande de datos de esta forma. De todas formas, es un sistema ideal para guardar valores tales como los ajustes de configuración de un módulo. Lo que pasamos a `variable_get()` es una clave describiendo nuestro valor (de manera que podamos recuperarlo) y un valor por defecto. En este caso, el valor por defecto es un array de los nodos en los cuales es posible crear anotaciones. Las permitiremos por defecto.

Por último, proporcionamos una descripción para mostrar al administrador del sitio un poco de información de guía sobre lo que debería ir en el campo correspondiente. Al consultar *Administración* → *configuración del sitio* → *Anotaciones* deberíamos poder ver el formulario para `anotaciones.module`.

La línea que defina `$form['array_filter']` es un poco misteriosa. De momento es suficiente con decir que es necesaria al guardar múltiples casillas de selección usando el hook de ajustes. En unas pocas líneas de código poseemos un formulario funcional de configuración para nuestro módulo que guardará y recordará nuestros ajustes.

Añadiendo el formulario de entrada de datos.

Para permitir al usuario introducir anotaciones en una página o historia, hemos de proporcionar un lugar para ello añadiendo un formulario para las anotaciones de la siguiente manera:

```
/**
 * Implementacion de hook_nodeapi().
 */
function anotaciones_nodeapi(&$node, $op, $teaser, $page) {
  switch ($op) {
    case 'view':
      global $user;
      // Si solamente se muestra en nodo resumen o
      // si el usuario es anónimo, abortamos.
      if ($teaser || $user->uid == 0) {
        break;
      }
      $types_to_anotaciones = variable_get('anotaciones_nodetypes', array('story'));
      if (!in_array($node->type, $types_to_anotaciones)) {
        break;
      }
      // Añadimos nuestro formulario como un elemento del contenido.
      $node->content['anotaciones_form'] = array(
        '#value' => drupal_get_form('anotaciones_entry_form', $node),
        '#weight' => 10
      );
    }
  }
}
```

En primer lugar, es preciso notar que estamos implementando otro *hook* de Drupal. Esta vez es el `_nodeapi()`, que es llamado cuando Drupal hace varias actividades con un nodo para que otros módulos como el nuestro, puedan modificar el nodo en cuestión antes que el proceso continúe. El nodo se nos pasa a través del `$node` disponible. El ampersand en el primer parámetro muestra que es una referencia al objeto `$node`, lo cual significa que cualquier modificación que hagamos al objeto `$node` en nuestro módulo se guardará. Ya que nuestro objetivo es añadir al formulario la información que deseamos, nos interesa poder modificar el nodo en cuestión. También se nos proporciona información sobre lo que Drupal estaba haciendo en el momento que nuestro código fue ejecutado. La información reside en el parámetro `$op` y puede ser *insert* (el nodo está siendo creado), *delete* (el nodo está siendo eliminado) u otro tipo de valores. Por el momento solamente estamos interesados en modificar el nodo cuando se está preparando para la presentación (en este caso `$op` será *view*). Estructuramos el código usando una instrucción

switch ya que así podremos ver qué hará nuestro módulo en cada caso. Lo siguiente es comprobar rápidamente los casos que no queremos mostrar en el campo de anotaciones. En caso que el parámetro *\$teaser* sea TRUE dicho nodo no se mostrará solo sino que lo hará en una lista similar a la de los resultados de un programa de búsqueda. No nos interesa añadir nada más en este caso. Otro caso es cuando el uid de *\$user* sea 0, lo cual significa que el usuario será anónimo. Emplearemos la instrucción *break* para salir del flujo de ejecución y así, evitar modificar la página. Antes de añadir el formulario de anotación a la página web, necesitamos comprobar si el nodo procesado para la presentación es uno de los tipos para los cuales hemos permitido las anotaciones en nuestra página de ajustes, por lo tanto, recuperamos el array de tipos de nodo que habíamos guardado previamente al implementar el *hook settings* y guardarlo en una variable con el nombre *\$types_to_anotaciones*. Como segundo parámetro de la llamada a *variable_get()*, debemos especificar el array por defecto a usar en caso de que el administrador del sitio no haya visitado aún la página de ajustes de nuestro módulo y haya entrado la configuración. El siguiente paso es comprobar que el nodo con el que trabajamos es efectivamente, del tipo que contiene *\$types_to_anotaciones*. De nuevo, salimos usando la instrucción *break* si no es un tipo de nodo deseado. Finalmente, debemos crear el formulario y añadirlo al atributo de contenido del objeto *\$node*. Primero hemos de definir el formulario para que tengamos algo que añadir. Lo haremos en una función separada cuyo único cometido es definir el formulario:

```
/**
 * Definimos el formulario para introducir una anotación.
 */
function anotaciones_entry_form($node) {
  $form['anotaciones'] = array(
    '#type' => 'fieldset',
    '#title' => t('Anotaciones')
  );
  $form['anotaciones']['nid'] = array(
    '#type' => 'value',
    '#value' => $node->nid
  );
  $form['anotaciones']['note'] = array(
    '#type' => 'textarea',
    '#title' => t('Notas'),
    '#default_value' => $node->anotaciones,
    '#description' => t('Crea una anotación personal sobre este contenido aquí. Solo tú o el administrador podrán verla.')
  );
  $form['anotaciones']['submit'] = array(
    '#type' => 'submit',
    '#value' => t('Update')
  );
  return $form;
}
```

Creamos el formulario de la misma forma que hicimos en nuestra función *anotaciones_admin_settings()* mediante la creación de un array por claves, sólo que esta vez queremos colocar nuestro recuadro de texto y el botón de *Enviar* dentro de un listado de campos de manera que se agrupen juntos en la página web. Primero creamos el array con *#type 'fieldset'* y le damos un título. Después creamos el array del área de texto. Es preciso notar que la clave del array del área de texto es un miembro del array de listado de campos. En otras palabras, usamos *\$form['anotaciones']['note']* en vez de *\$form['note']*. De esta forma, Drupal puede discernir si el área de texto de ese elemento es miembro de los elementos de la lista de campos. Por último, creamos el botón *Enviar* y devolvemos el array que define nuestro formulario.

De vuelta en *anotaciones_nodeapi()*, añadimos el formulario al contenido de la página añadiendo el valor y el peso del contenido del nodo. El valor contiene qué mostrar y el peso comunica a Drupal dónde mostrarlo. Queremos que nuestro formulario de anotaciones esté en la parte baja de la página y por ello le asignamos un número relativamente pesado como 10. Lo que queremos mostrar es nuestro formulario y para ello llamamos a *drupal_get_form()* para cambiar nuestro formulario desde un array describiendo como debería construirse en el formulario HTML final.

Cabe reseñar la manera en que pasamos el objeto *\$node* a través de nuestra función de formulario, ya que lo necesitaremos al obtener cualquier anotación anterior y rellenar el formulario con ella. Si consultamos el formulario hasta ahora creado, no pasará nada ya que de momento no hemos añadido código para gestionar la información de las anotaciones respecto a los comentarios. Pero antes de escribir el código pertinente, necesitamos pensar dónde vamos a guardar los datos introducidos por el usuario.

Gardando datos en una tabla de la base de datos.

El acercamiento más común para guardar datos usados por un módulo es crear una tabla separada para el módulo en la base de datos. Esto mantiene los datos separados de las tablas básicas de Drupal. Al decidir qué campos crear en nuestro módulo debemos preguntarnos: ¿Qué datos necesitamos guardar? Si hago una petición a esta tabla, ¿qué necesitaré? y por último, ¿Qué planes de futuro tengo para mi módulo?

Los datos que necesitamos guardar en este caso son simplemente el texto de la anotación, el ID numérico del nodo al que se aplica y el ID del usuario (*uid*) que escribió la anotación. También puede ser útil guardar una fecha de tal manera que podamos listar las anotaciones recientes ordenadas por fecha (*timestamp*).

Finalmente, la pregunta principal que debemos hacernos sobre la tabla es “¿Qué es la anotación de este usuario para este nodo?”. Crearemos un índice de compendios de los campos *uid* y *nid* (ID de un nodo) para hacer lo más rápida posible nuestras peticiones más frecuentes. El código SQL para nuestra tabla será algo similar a esto:

```
CREATE TABLE anotaciones (
  uid int NOT NULL default '0',
  nid int NOT NULL default '0',
  note longtext NOT NULL,
  timestamp int NOT NULL default '0',
  PRIMARY KEY (uid, nid),
);
```

Podríamos solamente añadir esta sentencia SQL en nuestro fichero *README.txt* y los que quisieran instalar el módulo deberían añadir manualmente las tablas a sus bases de datos. Por el contrario, haremos que Drupal se encargue de la instalación sacando provecho de las funcionalidades destinadas a estos procesos cada vez que el módulo sea activado. Crearemos un fichero especial con el nombre de nuestro módulo y el sufijo *.install* de esta manera: *anotaciones.install*. En el introduciremos el siguiente código:

```
<?php
// $Id$
function anotaciones_install() {
  drupal_set_message(t('Empezando la instalación del módulo anotaciones.'));
  switch ($GLOBALS['db_type']) {
    case 'mysql':
    case 'mysqli':
      db_query("CREATE TABLE anotaciones (
        uid int NOT NULL default 0,
        nid int NOT NULL default 0,
        note longtext NOT NULL,
        timestamp int NOT NULL default 0,
        PRIMARY KEY (uid, nid)
      ) /*!40100 DEFAULT CHARACTER SET utf8 */;");
    };
    $success = TRUE;
    break;
    case 'pgsql':
      db_query("CREATE TABLE anotaciones (
        uid int NOT NULL DEFAULT 0,
        nid int NOT NULL DEFAULT 0,
        note text NOT NULL,
```

```

timestamp int NOT NULL DEFAULT 0,
PRIMARY KEY (uid, nid)
);"
);
$success = TRUE;
break;
default:
drupal_set_message(t('Base de datos no soportada.'));
}
if ($success) {
drupal_set_message(t('El módulo ha instalado las tablas con éxito.'));
}
else {
drupal_set_message(t('La instalación del módulo anotaciones no ha tenido éxito.'),'error');
}
}
}

```

El fichero es simple. Cuando el módulo anotaciones se activa por primera vez, Drupal busca un fichero *anotaciones.install* ejecuta la función *anotaciones_install()* y si todo va bien, las tabla de la base de datos se habrán creado. En caso de querer desinstalar del todo el módulo, debemos desactivarlo en el panel de administración de módulos y eliminar la entrada de la tabla del sistema en la base de datos.

Después de crear la tabla para guardar los datos, tenemos que hacer ciertas modificaciones a nuestro código por un motivo, hemos de añadir código para gestionar el procesamiento de datos una vez el usuario entre una anotación y pulse el botón *Enviar*. La función de envío de datos de nuestro formulario será:

```

/*
 * Guardamos la anotación en la base de datos.
 */
function anotaciones_entry_form_submit($form_id, $form_values) {
global $user;
$nid = $form_values['nid'];
$note = $form_values['note'];
db_query("DELETE FROM {anotaciones} WHERE uid = %d and nid = %d", $user->uid,
$nid);
db_query("INSERT INTO {anotaciones} (uid, nid, note, timestamp) VALUES (%d, %d,
'%s', %d)", $user->uid, $nid, $note, time());
drupal_set_message(t('Tu anotación se ha guardado.'));
}

```

Desde el momento en que solamente permitimos una anotación por nodo y usuario, podemos borrar perfectamente cualquier anotación previa en caso de haber, e insertar la nuestra en la base de datos. Hay algunas cosas que debemos tener en cuenta sobre las interacciones con la base de datos. La primera es que no necesitamos preocuparnos de conectarnos a la base de datos ya que Drupal ya lo ha hecho por nosotros en la fase de *bootstrap*. Lo segundo es que cada vez que nos referamos a la tabla en la base de datos, debemos poner corchetes {} para que el prefijo de la tabla se use sin problemas (para más información, consultar: <http://drupal.org/node/2622>). Y tercero y último, usamos elementos de referencia (*placeholders*) en nuestras peticiones a la base de datos y posteriormente especificar los datos a través de variables para que el mecanismo de saneamiento de Drupal para sentencias SQL para prevenir ataques de inyección de SQL, funcione correctamente. Usaremos %d para números y %s para cadenas de caracteres. También usaremos *drupal_set_message()* para publicar mensajes en la sesión de usuario y que Drupal los muestra como una noticia en la siguiente página que el usuario vea, permitiendo, así, que el usuario reciba cierto *feedback*.

Por último, necesitamos cambiar el código de nuestro *hook nodeapi* de manera que si existe una anotación previa, se extraiga de la base de datos. Justo antes de asinar nuestro formulario a *\$node->content*, añadimos las siguientes líneas:

```
// Obtener una anotación previamente guardada en caso de haber.  
$result = db_query("SELECT note FROM {anotaciones} WHERE uid = %d AND nid = %d",  
$user->uid, $node->nid);  
$node->anotaciones = db_result($result);
```

Primero hacemos una petición a la table de la base de datos para este usuario y este nodo. Después usamos `db_result()` para obtener la primera línea del resultado. Ya que solo permitimos una sola nota por usuario y nodo, solo debería haber una línea.

Llegados a este punto, ya hemos acabado de elaborar nuestro módulo *anotaciones*. Si queremos compartir este módulo con la comunidad de desarrollo de Drupal deberíamos tener los siguientes ficheros en nuestro directorio *anotaciones*: *anotaciones.info*, *anotaciones.install*, *anotaciones.module* y *README.txt* y enviarlo al repositorio de contribuciones de la comunidad en *drupal.org* creando una página de proyecto para mantener un feedback con terceros.

ANEXO III: Instalación de FFMPEG en Devian Etch (archivos binarios)

FFMPEG es un software de manipulación de video muy potente aunque en cuanto a instalación se refiere no es nada amigable. Mientras que muchos programas de Linux pueden ser instalados mediante una versión precompilada del software o mediante archivos binarios con los cuales únicamente es necesaria su compilación, la instalación de FFMPEG depende en gran parte de los codecs y contenedores de vídeo que se quieran usar en la entrada o salida de datos a codificar. A mayor número de formatos soportados, la dificultad de la instalación crece exponencialmente. Asumiremos que el usuario está familiarizado en el uso de la línea de comandos de Linux, moverse entre directorios, editando ficheros de texto y cierta experiencia compilando programas. Lo primero que es recomendable hacer es añadir a nuestra lista de fuentes de repositorios uno nuevo. Para ello haremos lo siguiente:

```
pico /etc/apt/sources.list
```

Y añadir la siguiente línea:

```
deb http://www.debian-multimedia.org stable main
```

Este repositorio contiene algunas de las bibliotecas necesarias para xvid y x264 (un codec H.264 de código abierto) entre otras cosas. Ahora necesitaremos instalar el software. Este software es posible que se pueda instalar a través de otros repositorios que podamos tener en nuestro fichero *sources.list*, pero para evitar confusiones utilizaremos el ya mencionado. Actualizamos las fuentes de la lista de repositorios:

```
apt-get update
```

Recomendamos la instalación de *checkinstall*, ya que lo podemos usar en vez de *make install* y generará un paquete *.deb* que podremos usar en futuras re-instalaciones o instalaciones en varias máquinas para facilitarnos el trabajo. Si no poseemos *checkinstall* es posible bajarlo desde: <http://www.asic-linux.com.mx/~izto/checkinstall/download.php>

Navegando podemos encontrar el fichero que necesitamos y una vez descargado el último paquete *.deb* podemos ejecutar la siguiente instrucción como root:

```
dpkg -i checkinstall_1.6.1-1_i386.deb
```

Ahora *checkinstall* debería estar instalado en el sistema. Pasemos de lleno a FFMPEG y construyamos las dependencias:

```
apt-get build-dep ffmpeg
```

Lo siguiente a hacer es instalar más software que nos permita añadir más tipos de fichero a parte de los que ya contiene FFMPEG a dicha aplicación.

```
apt-get install liblame-dev libfaad-dev libfaac-dev libxvidcore4-dev liba52-0.7.4 liba52-0.7.4-dev libx264-dev build-essential subversion
```

Es necesario tener instaladas las herramientas necesarias para compilar el código binario y obtener las versiones de los ficheros de los repositorios de Subversion. Vamos a comprobarlo para FFMPEG.

```
svn checkout svn://svn.mplayerhq.hu/ffmpeg/trunk ffmpeg
```

Cuando se escribió esta guía de instalación la última revisión era la 11212 y por ello será la que usaremos. Si se requiere la última versión de todas, podemos ejecutar:

```
svn checkout -r 11212 svn://svn.mplayerhq.hu/ffmpeg/trunk ffmpeg
```

Esto asegurará que también nos descarguemos la revisión 11212. Una vez descargada, nos situaremos en el directorio de ffmpeg (*cd ffmpeg*) y haremos la siguiente configuración:

```
./configure --enable-gpl --enable-pp --enable-libvorbis --enable-liba52 --enable-libdc1394 --enable-libgsm --enable-libmp3lame --enable-libfaad --enable-libfaac --enable-pthreads --enable-libx264 --enable-libxvid
```

Hasta ahora hemos hecho:

- gpl - permitir el uso de código GPL, las resultantes libav* y ffmpeg will estarán bajo GPL
- pp - permitir el soporte de postprocesado bajo GPL.
- libvorbis - activar la codificación Vorbis via libvorbis (<http://www.vorbis.com/>)
- liba52 - permitir el soporte liba52 bajo GPL (<http://liba52.sourceforge.net/>)
- libdc1394 - activar la obtención IIDC-1394 usando libdc1394 y libraw1394 (<http://sourceforge.net/projects/libdc1394/>)
- libgsm - permitir el soporte GSM via libgsm (<http://packages.debian.org/testing/sound/libgsm-tools>)
- libmp3lame - activa la codificación MP3 via libmp3lame (<http://lame.sourceforge.net/download.php>)
- libfaad - activar el soporte FAAD via libfaad (<http://www.audiocoding.com/faad2.html>)
- libfaac - activar el soporte AAC via libfaac (<http://www.audiocoding.com/faac.html>)
- pthreads - usar pthreads (http://en.wikipedia.org/wiki/POSIX_Threads)
- libx264 - activar la codificación H.264 via x264 (<http://www.videolan.org/developers/x264.html>)
- libxvid - activar la codificación Xvid via xvidcore. Existen codificadores nativos MPEG-4/Xvid (<http://www.xvid.org/>)

Esta información en concreto y mucha más, se puede encontrar escribiendo *./configure -help*. Ya estamos listos para hacer la instalación de los ficheros en la línea de comandos:

make

Si algo ha ido mal y es necesario empezar de nuevo, podemos hacer:

make distclean

Una vez hayamos acabado con la instrucción *make*, haremos finalmente:

checkinstall

Se nos preguntarán varias opciones, las cuales deberían ser fáciles de contestar: Crear la documentación (*yes*), escoger un nombre, escoger D para paquetes Debian, seleccionar el número 3 y escribir un nombre de versión que signifique algo para nosotros. Hemos escogido el nombre *svn11212-etch-20081225*. Checkinstall creará un paquete Devian para FFMPEG para nuestro sistema con la configuración anteriormente introducida. Checkintall NO instalará en paquete, con lo cual es necesario hacer lo siguiente:

```
dpkg -i ffmpeg_svn11212-etch-20071213-1_i386.deb
```

Con un poco de suerte, ya dispondremos de una versión funcional de FFMPEG instalada en nuestro servidor Debian Etch. Podremos generar ficheros codificados con H.264 en una importante variedad de contenedores. Para consultar las opciones codificación de FFMPEG, podemos visitar la documentación en línea en: <http://ffmpeg.mplayerhq.hu/ffmpeg-doc.html>

