UNIVERSIDAD AUTONOMA DE MADRID

ESCUELA POLITECNICA SUPERIOR

PROYECTO FIN DE CARRERA

HERRAMIENTA PARA EL DISEÑO ALTAMENTE INTERACTIVO DE APLICACIONES WEB

David Jiménez Burgos

Noviembre 2012

Herramienta para el diseño altamente interactivo de aplicaciones web

AUTOR: David Jiménez Burgos
TUTOR: Roberto Moriyon Salomón

Grupo GHIA
Dpto. de Ingeniería Informática
Escuela Politécnica Superior
Universidad Autónoma de Madrid
Noviembre 2012

Resumen

Un **sistema de gestión de contenidos** (CMS) es una aplicación que permite la creación y administración de aplicaciones web, normalmente, a través de un navegador. Consiste en una interfaz que hace de traductor con una base de datos que él mismo controla, permitiendo a través de ésta manejar de forma independiente el contenido y diseño del sitio web.

Un gestor de contenidos genera páginas web dinámicas interactuando con un servidor web bajo petición del gestor (la persona encargada de administrar el sitio web), generándose la información sobre el contenido, el formato y la estructura del sitio web que es almacenada en la base de datos del servidor.

Esta manera de trabajar con webs dinámicas permite gestionarlas de una manera más sencilla y rápida respecto a un sitio web estático, en el que cada cambio del diseño debe ser realizado en todas las páginas web, y cada vez que se agrega un nuevo contenido hay que perder tiempo en la maquetación de dicha página HTML.

Una desventaja de los sistemas de gestión de contenidos actuales, es la manera de gestionar toda la información pues no se realiza desde el propio contexto de la aplicación, sino desde un panel en el que no se visualiza el resultado tal y como va a aparecer en el sitio web final.
Esta desventaja se ve reflejada, por ejemplo, a la hora de insertar un contenido donde el gestor de la Web debe entrar a un panel de administración y rellenar un formulario. La manera adecuada para insertar la noticia sería en la propia página Web visualizando el resto de contenidos del sitio. De una manera similar ocurre a la hora de editar un contenido, donde los sistemas de contenidos actuales recurren a un formulario fuera del contexto Web para editar el contenido.

El objetivo de este proyecto final de carrera es realizar un estudio para abordar la solución al problema citado, estudiando los diferentes sistemas de gestión de contenidos y encontrando una solución práctica que no modifique las ventajas que ya ofrecen estos sistemas.

Palabras clave

Página Web, Base de datos, Contexto Web, Sistema de gestión de contenidos, Interactividad, Reutilización, WYSIWYG, Interfaz, contexto web

Agradecimientos

En primer lugar, quiero dar las gracias a mi tutor Roberto Moriyon Salomón, por brindarme la oportunidad de realizar este Proyecto Fin de Carrera y por transmitirme su conocimiento y pasión por las tecnologías relacionadas con la web. De verdad que sin su dedicación, consejos y debates no habría sido posible completar este trabajo de una manera tan satisfactoria.

También quiero dar las gracias a todos los profesores de la Escuela Politécnica Superior que nos han aportado sus conocimientos durante todos estos años, realizando una magnifica labor profesional y teniendo tanta paciencia con nosotros, haciéndome sentir orgulloso de haber estudiado en esta Escuela.

A mi familia y amigos, por todo el amor que me transmiten cada día, sin el cual todo se haría infinitamente más difícil. Gracias por vuestro apoyo incondicional, y por levantarme el ánimo siempre que lo necesitaba.

Y a todos vosotros, compañeros, amigos de la universidad, por todos los buenos y malos momentos que hemos pasado juntos tanto en la escuela como fuera de ella, por hacer que me sienta parte de una gran familia que espero y deseo resista el paso del tiempo, un fuerte abrazo.

David Jiménez Burgos
Noviembre 2012.

INDICE DE CONTENIDOS

Capítulo 1

Introducción

1.1. Motivación

Mi principal motivación viene de los 17 años, la edad con la que empecé a desarrollar distintas páginas web. En su momento y debido a mi desconocimiento en lenguajes de programación de páginas web tuve que acudir a programadores y a herramientas de gestión de contenidos para conseguir desarrollarlas.

Las herramientas de gestión de contenidos son una gran oportunidad para las personas cuyo dominio de HTML, PHP o MySQL es limitado, ya que permiten de una manera intuitiva, fácil y rápida la gestión de contenidos de las páginas web.

Actualmente existen varios sistemas de gestión de contenidos tales como Joomla, Wordpress o Drupal; cada uno, tiene características propias que le diferencian del resto (las cuales se mostrarán en el estudio del arte del proyecto), pero todos tienen como objetivo que el gestor del sitio web consiga la funcionalidad deseada de la página web de una manera sencilla, rápida y sin tener que programar.

Manteniendo una conversación con mi actual tutor, Roberto Moriyon, coincidimos en un aspecto negativo que comparten todos estos sistemas: el contexto donde se gestiona el contenido no es el adecuado (ya que en todos los sistemas de gestión de contenidos, para crear/editar un contenido, el usuario debe entrar a un panel fuera del contexto del sitio web), y es que al igual que cuando usamos un editor de texto como *Microsoft Word*, no nos planteamos que para rellenar una tabla que hemos insertado debamos abandonar parcialmente el documento de texto, lo mismo debería ocurrir a la hora de crear un contenido de un sitio web con un gestor de contenidos.

1.2. Objetivos y estructuración de la memoria

El objetivo de este proyecto es la **propuesta de mecanismos y técnicas** que permitan mejorar algunas características de los sistemas de gestión de contenidos online actuales como son la *interactividad* y la *reutilización de contenidos*.

Siendo más concreto, los objetivos principales que se quieren abordar en este proyecto son los siguientes:

(1) Permitir que la gestión del sitio web se realice en el propio contexto de la aplicación, pudiéndose realizar las distintas tareas de gestión en la propia página web visualizándose así el sitio web en todo momento tal y como le va a aparecer al usuario final.

(2) Permitir almacenar estructuras de contenidos más usados para conseguir una mayor reutilización de contenidos y dar mayor agilidad a la creación de contenidos de la misma manera que lo realizan algunos gestores de contenidos ya existentes (por ejemplo *Drupal*). Con este objetivo se pretende almacenar diferentes contenidos como imágenes, conjuntos de imágenes, formularios… y reutilizarse en otro momento, pero con el valor añadido de estar en armonía con el objetivo (1) que se detalló anteriormente.

Esta memoria está estructurada como sigue:

En el segundo capítulo de la memoria de este proyecto se describirá el **estado del arte**, donde se detallará la situación actual de las distintas tecnologías relacionadas con el proyecto y se entrará en un estudio más profundo de los sistemas de gestión de contenidos más utilizados en la actualidad.

En el tercer capítulo, al que se le ha denominado **diseño**, se explicará en primer lugar el concepto de *WYSIWYG* destinado a los gestores de contenidos con el que se conseguirá paliar el problema de contexto en cuanto a interactividad en estos sistemas (Objetivo 1). En segundo lugar, se detallará un *método de inserción de contenidos* que ayuda a la interactividad del usuario sin perder el contexto del sitio Web (Objetivo 2).
Y para finalizar este tercer capítulo, se estudiará un modelo de *reutilización de contenido* que permitirá al gestor guardar diferentes estructuras de contenido para reutilizarlas cuando lo desee (Objetivo 3).

En el cuarto capítulo de este proyecto, **implementación**, se explica detalladamente la manera de llevar a cabo las técnicas planteadas en el tercer capítulo.

En el quinto capítulo, se mostrarán las **conclusiones finales** y se propondrá un **futuro trabajo** relacionado con la continuación de este proyecto.

Capítulo 2

Estado del arte

2.1. Tecnologías

Lenguaje HTML

Es el lenguaje con el que se definen las páginas web, usado para describir la estructura y el contenido en forma de hipertexto, así como para complementar el texto con objetos tales como imágenes y programas. Consta de etiquetas (concretamente, etiquetas de apertura y de cierre), a las cuales se les atribuye propiedades en función del nombre de las mismas.

Así por ejemplo, si escribimos algo como: <*B*>***Esto está en negrita***</*B*>, conseguiremos que el texto escrito entre las etiquetas <*B*> y </*B*> quede en negrita.

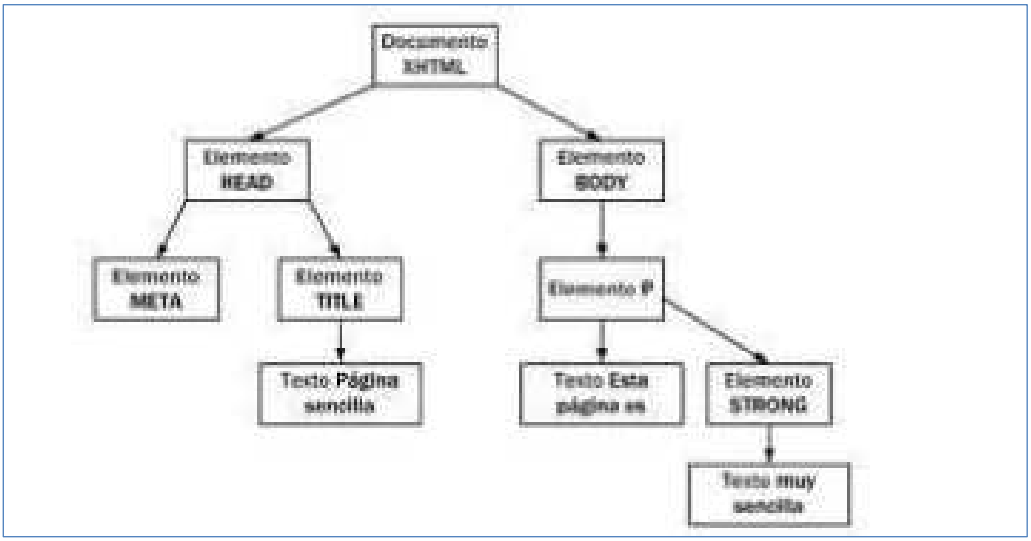


Fig.2.1.1 Visión del código HTML de una página web en modo de árbol

Más adelante surgió el **CSS** (*Cascading Style Sheets*), un lenguaje de hojas de estilos creado para controlar el aspecto y presentación de las páginas Web.

Con la creación del CSS, se consigue separar el contenido del sitio Web y la presentación, pudiéndose modificar el estilo de la Web sin alterar su contenido modificando, únicamente, algunos parámetros del CSS.

```
#wrapper {
    width:800px;
    margin:0 auto;
}

#header {
    height:90px;
    position:relative;
}

#home_content {
    height:228px;
    position:relative;
}

#feature {
    margin-bottom:20px;
    clear:both;
}

#feature .features {
    width:550px;
    height:250px;
    float:right;
}
```

Fig.2.1.2 Ejemplo de lenguaje CSS

Lenguaje JavaScript

JavaScript es un lenguaje de programación que permite a los desarrolladores crear acciones en sus páginas Web. Aunque JavaScript no es el único lenguaje que se puede utilizar para añadir interacción a los documentos HTML, es el lenguaje estándar que se utiliza en este ámbito.

Con la ayuda de JavaScript se consiguen realizar páginas Web con una mayor interactividad para el usuario final, ya que permite ir realizando acciones en la página como respuesta a actos del mismo.

Dentro de JavaScript es importante conocer los conceptos de **AJAX** y **DOM**:

Acrónimo de Java script asíncrono y XML (**AJAX**), es una técnica muy útil para la creación de páginas interactivas. Este tipo de aplicaciones se ejecutan en el cliente, teniendo una comunicación asíncrona con el servidor en un segundo plano.
Con esta forma de trabajo es posible hacer que el servidor realice cambios sobre una página web sin necesidad de recargar la misma, consiguiendo un aumento en interactividad, velocidad y usabilidad.

DOM (Modelo de objetos del documento), es un modelo de objetos que genera el navegador cuando se carga un documento y se puede alterar mediante JavaScript, u otro lenguaje que lo soporte, para cambiar dinámicamente el contenido y aspecto de la página.
El modo de representación de los objetos se realiza como una estructura de árbol, visualizándose el documento como conjuntos de hojas con distintos hijos y hermanos, pudiéndose navegar y operar sobre estos de una manera muy fácil con la API de DOM.

La manera que tiene de trabajar DOM es identificar cada etiqueta HTML como un nodo dentro del árbol, identificando como hijo a una etiqueta dentro de otra etiqueta padre o, en el caso de las hojas, a una parte del contenido del documento.
Esta forma de trabajar permite recorrer todas las etiquetas del árbol de manera sencilla, pudiendo editar atributos tanto de nodos como de ramas, así como añadir o borrar nodos.

El sigiente código es un ejemplo básico de código HTML donde se muestra también su gráfico en modo de árbol análogo:

```
<body>
<p>Esto es un párrafo que contiene <a href="#">un enlace</a>
<ul>
<li>Primer punto en la lista</li>
<li>Otro punto en la lista</li>
</ul>
</body>
```

Fig.2.1.3 Ejemplo de lenguaje HTML con distintos elementos.

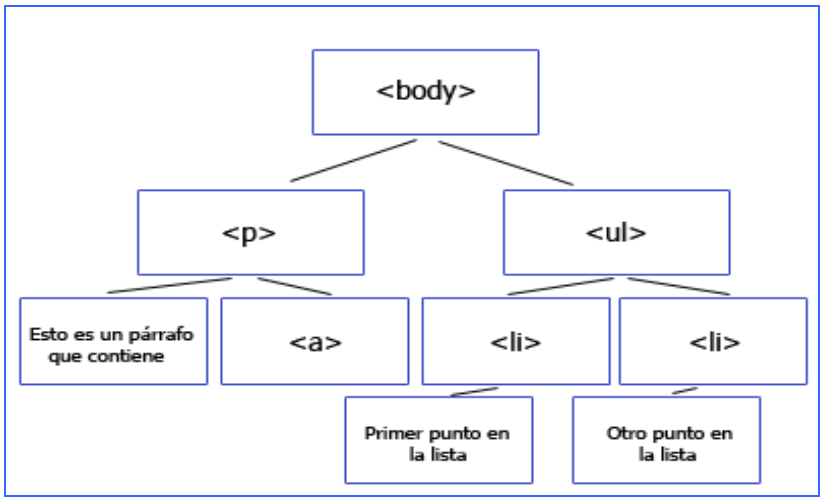
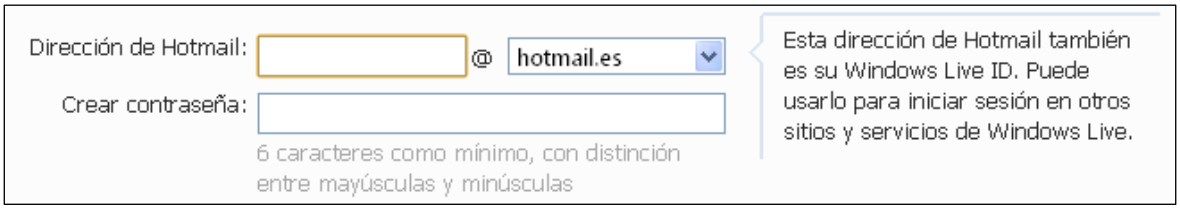


Fig.2.1.4 Visión en árbol (DOM) del código HTML de la figura 2.3.

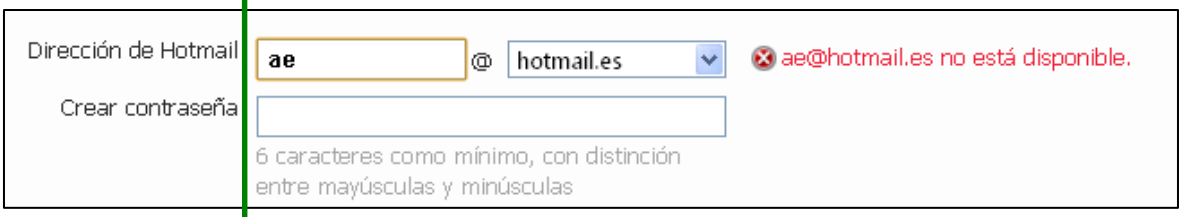
La utilización conjunta de AJAX y DOM, permite que el servidor Web envíe órdenes de modificación del contenido de la página Web dependiendo de las acciones del usuario.

Un ejemplo práctico de esta utilización conjunta de conceptos, se puede ver es algunos formularios de sitios Web donde hay una gran afluencia de usuarios:

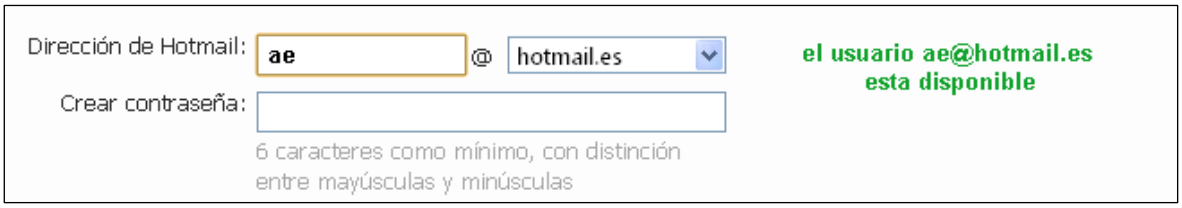
Estos formularios 'inteligentes', lo que pretenden es que mientras el usuario esta completando el formulario, por detrás se estén realizando consultas a la base de datos del servidor (mediante AJAX) y mostrarle los resultados de esas consultas al usuario mediante notificaciones de texto (utilizando DOM), y que así el usuario no tenga que perder mucho tiempo en completar satisfactoriamente el formulario.



El servidor tras realizar una consulta a la base de datos mientras el usuario esta completando el resto del formulario, establece que ese usuario ya existe en la base de datos y muestra un mensaje al usuario notificándoselo.



El servidor tras realizar una consulta a la base de datos mientras el usuario esta completando el resto del formulario, establece que ese usuario no existe en la base de datos y muestra un mensaje al usuario notificándoselo.



Bases de datos relacionales

Una base de datos relacional es una base de datos que cumple con el modelo relacional, es decir, establece interconexiones (relaciones) entre los datos (que están guardados en tablas), y a través de dichas conexiones relaciona los datos de ambas tablas, y así evita la duplicidad de los registros. Las relaciones se guardan a su vez en tablas, lo que permite realizar la gestión de la base de datos mediante la implementación de un conjunto reducido de operaciones sobre tablas (Algebra Relacional).

El software específico dedicado a servir como interfaz entre la base de datos, el usuario y las aplicaciones es el llamado *sistema de gestión de base de datos*, cuyas principales capacidades son:

- Definir y crear datos dentro de las diferentes tablas.
- Manipular los distintos datos.
- Darle seguridad a los datos.
- Recuperar los datos, asegurando un alto rendimiento en cuanto a velocidad de recuperación.

El lenguaje de consulta estructurada o SQL, es un lenguaje declarativo de acceso, creación y gestión de bases de datos relacionales que permite especificar diversos tipos de operaciones en éstas. Una de sus características es el manejo del álgebra y el cálculo relacional permitiendo efectuar consultas con el fin de recuperar, de una forma sencilla, información de interés de una base de datos, así como también hacer cambios sobre ella.

2.2. Editores Web

Editores de código HTML

Un editor HTML es una herramienta que acelera determinadas tareas comunes en la edición de código HTML y a la vez realza la presentación del código fuente, por ejemplo, diferenciando las etiquetas del texto. Gracias a los editores HTML la creación de documentos HTML resulta más rápida, cómoda y legible.

Aunque también se pueden utilizar editores de texto como el *Notepad* para editar documentos HTML, a estos no se les consideran editores de HTML per sé, ya que el código generado no es demasiado legible y el proceso de edición es muy lento.

Los editores más útiles para el usuario son los **editores gráficos o "WYSISWYG" (What You See Is What You Get),** que son los editores de HTML más intuitivos, ya que nos permiten diseñar una página web sin tener que escribir el código con las etiquetas. Presentan una facilidad adicional para las personas que no conocen el lenguaje HTML ya que el propio editor HTML es el encargado de programar código HTML internamente a partir de lo que nosotros estamos diseñando visualmente.

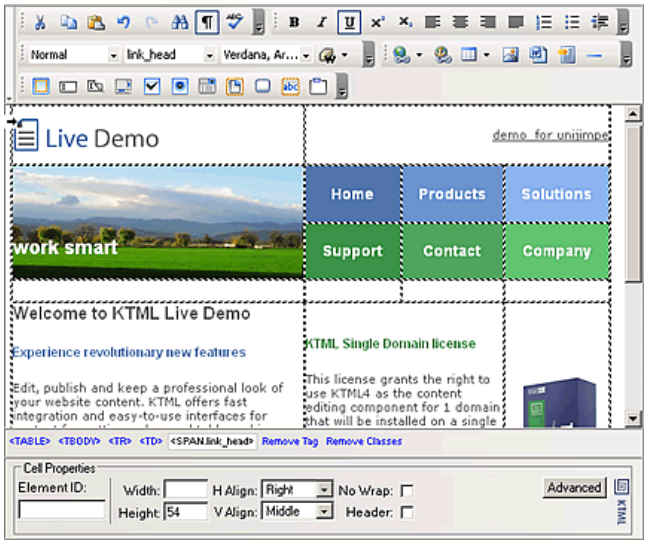
Algunos de los muchos editores gráficos que existen actualmente son: *Dreamweaver, Frontpage, Kompozer...*

Fig. 2.2.1 Ejemplo gráfico del Dreamweaver

Dentro de los editores WYSISWYG, los especialmente interesantes son los **editores de código HTML online,** que son editores HTML WYSIWYG que permiten escribir todo tipo de código HTML en una página web mostrada en un navegador sin necesidad de instalar ningún plugin o ActiveX adicional.

Estos editores WYSIWYG online, generalmente están escritos en JavaScript ya que gracias a este lenguaje se consigue obtener una alta interacción con el usuario. Normalmente dan la posibilidad de guardar el documento en el propio ordenador del usuario o en un servidor mediante AJAX.

Un ejemplo de estos tipos de editores son el *KTML, TinyMCE* ó *CKeditor*.



2.2.2 Ejemplo gráfico del KTML

De cara al objetivo principal de este proyecto de paliar el problema de interactividad en los sistemas de gestión de contenidos es conveniente detallar algunos aspectos del editor *CKeditor*, ya que la filosofía de edición que sigue se ajusta perfectamente a la que se pretende llegar para los CMS con este proyecto final de carrera.

El **CKeditor** es un editor de texto enriquecido online de código abierto, que permite la edición de textos desde el navegador sin la necesidad de instalar ningún componente en la computadora del cliente.

CKeditor no es una aplicación de escritorio como Microsoft Word, sino un componente para ser utilizado por los desarrolladores para mejorar sus aplicaciones web (un editor que se utiliza dentro de las páginas web).

2.2.3 Ejemplo gráfico del CKeditor

Como se puede apreciar en la imagen 2.8, el menú de edición ofrece características muy similares a editores de textos para escritorio del estilo de Microsoft Word. Con la ayuda de esta herramienta, los usuarios pueden editar textos de una manera sencilla y muy interactiva, consiguiendo que se vea en todo momento el resultado final del texto.

Desde el punto de vista del diseñador de un sitio web, CKeditor ofrece una amplia API de JavaScript que permite personalizar cada aspecto de la herramienta y su comportamiento. Esta personalización engloba desde los colores, plantillas y botones del menú hasta la manipulación del contenido y plugins instalables, acorde con las necesidades de cada sitio web.

Además, CKeditor esta diseñado para realizar comunicaciones con el servidor, convirtiéndose en una herramienta perfecta para aplicaciones AJAX.

Como se verá en el apartado de diseño de este proyecto, el uso de esta herramienta y de su filosofía de trabajo hará posible abordar el propósito de este proyecto.

2.3. Aplicaciones web

Una aplicación Web es cualquier aplicación que es accedida vía Web por una red como Internet o una intranet. Ejemplos muy conocidos de aplicaciones Web son los webmails, wikis, weblogs, etc…

Las aplicaciones Web permiten una gran compatibilidad entre plataformas (dispositivo móvil, ordenador Linux, Ordenador Windows…), ya que operan en un navegador Web.

Pueden existir miles de usuarios pero una única aplicación instalada en un servidor, por lo que se puede actualizar y mantener una única aplicación y todos sus usuarios verán los resultados inmediatamente.

Algunas de las tecnologías que se emplean para desarrollar las aplicaciones son por ejemplo: PHP, Java EE y .NET, entre otras.

PHP

PHP es el acrónimo de Hipertext Preprocesor. Es un lenguaje de programación del lado del servidor gratuito e independiente de plataforma, rápido, con una gran librería de funciones y mucha documentación.

El código fuente escrito en PHP es invisible para el navegador y el cliente, ya que es ejecutado en el servidor, generando HTML, que es enviado al cliente.
Las páginas que se ejecutan en el servidor pueden realizar accesos a base de datos, conexiones en red, y otras tareas para crear la página final que verá el usuario.

Debido a su amplia distribución, PHP está perfectamente soportado por una gran comunidad de desarrolladores que crean las llamadas *extensiones* que expanden el potencial de PHP.

```php
<?php
    echo "Hola, ¡soy un script PHP!";
?>
```

2.3.1 Ejemplo de código PHP

Java EE

Java Platform Enterprise Edition o Java EE, es una plataforma de programación para desarrollar y ejecutar software de aplicaciones Web en lenguaje de programación Java.

Con esta plataforma se permite la creación de contenidos web a través de un servidor web, el cual puede manejar transacciones, la seguridad, escalabilidad, concurrencia y gestión de los componentes. Esto hace que los desarrolladores puedan concentrarse más en la lógica de negocio de los componentes en lugar de en tareas de bajo nivel.

Java EE incluye varias especificaciones de API, tales como JDBC, RMI, e-mail, JMS, Servicios Web, XML… y define cómo coordinarlos, a la vez que también configura algunas especificaciones únicas para sus componentes. Estas incluyen:

- **Enterprise JavaBeans (EJB):** Los EJB proporcionan un modelo de componentes distribuido estándar del lado del servidor.

  El objetivo de los EJB es dotar al programador de un modelo que le permita abstraerse de los problemas generales de una aplicación empresarial (concurrencia, transacciones, persistencia, seguridad, etc.) para centrarse en el desarrollo de la lógica de negocio en sí. El hecho de estar basado en componentes permite que éstos sean flexibles y sobre todo reutilizables.

- **Servlets:** Son objetos que se ejecutan en un contenedor, especialmente diseñado para ofrecer contenido dinámico desde un servidor web, generalmente HTML.

  El ciclo de vida de un Servlet se divide en los siguientes puntos:

  1. El cliente solicita una petición a un servidor vía URL.
  2. El servidor recibe la petición y la asocia con un servlet, el contenedor de Servlets gestiona su utilización cargándolo si es necesario y asociándole un hilo.
  3. Se procesa la petición devolviendo el resultado al cliente.
  4. Cuando se desactiva el contenedor de Servlets, estos se destruyen y se liberan los recursos abiertos.

  El uso más común de los Servlets es generar páginas web de forma dinámica a partir de los parámetros de la petición que envíe el navegador web.

- **JavaServers Pages (JSPs):** Es una tecnología Java que permite generar servlets a partir de documentos que contienen código HTML con etiquetas específicas y código Java embebido.

La utilización de todos estos componentes permite al desarrollador crear una Aplicación de Empresa portable entre plataformas y escalable, a la vez que integrable con tecnologías anteriores.

.NET

Microsoft creó .Net como respuesta a las dos tecnologías mencionadas anteriormente (PHP y JEE) proponiendo este Framework para desarrollar aplicaciones Web de una manera sencilla, económica y segura, con la intención de dominar el negocio de las aplicaciones Web e integrar el desarrollo de todos sus sistemas.

La principal ventaja de la utilización de .Net es que este utiliza un mismo entorno de ejecución (CLR) donde se cargan las aplicaciones desarrolladas en los distintos lenguajes que este soporta (C++, Visual Basic, Pascal, Python…), permitiendo así integrar proyectos en distintos lenguajes.

La herramienta de desarrollo compila el código fuente de cualquiera de los lenguajes soportados por .NET en un código intermedio, y luego se genera un código máquina real que se ejecuta en la plataforma del cliente.

Sistemas de gestión de contenidos (CMS)

Los CMS son sistemas de publicación estructurada de contenidos, que permiten separar el trabajo de redacción y elaboración, de la presentación y organización.

Desde el punto de vista del usuario interno, son sistemas para gestionar, de forma uniforme, accesible, y cómoda, sitios Web dinámicos sobre los que pueden trabajar una o más personas, cada una de las cuales tiene una función determinada. Desde el punto de vista del usuario de la web, son sitios Web dinámicos, con apariencia e interfaz uniforme, con un diseño centrado en el usuario, y que permiten llevar a cabo fácilmente las tareas para las que han sido diseñados.

Los CMS disponen de dos partes: **la parte pública**, que es la aplicación web tal y como la ven las personas que acceden a ella y **la parte privada** donde únicamente se puede acceder si se tienen los permisos adecuados y que consiste en un panel de administración desde el cual se puede crear y modificar de una manera rápida y sencilla el contenido de la página web pública (artículos, imágenes, archivos…), y sobre el que pueden trabajar una o más personas.

El acceso al gestor (parte privada) se realiza generalmente a través del navegador web, y se puede requerir el uso de FTP para subir contenido. Las tareas que se realizan en el panel de gestión se complementan con una base de datos donde se almacena toda la información sobre el contenido y estructuración del mismo.

Los CMS con la utilización de las bases de datos se convierten en **sistemas dinámicos en la creación de contenidos**, ofreciendo así una gran flexibilidad y adaptabilidad para la creación de webs con estructuras y funcionalidades muy distintas.

En palabras coloquiales, la función que desempeñan los sistemas de gestión de contenidos es de **traductor de órdenes/datos**, ya sea del panel de administración hacia la base de datos (cuando el gestor de la web crea/modifica el contenido o estructura de la web y se guarda en la base de datos), o bien de la base de datos a la aplicación web que visualiza el usuario final (cuando un usuario entra en el sitio web, visualiza todos los contenidos con el diseño, estructura y contenido que el gestor había elegido y que ha sido recuperado de la base de datos).

2.3.1 Esquema del modo de funcionamiento de un gestor de contenidos

La base de datos tiene **almacenado todo el contenido de la aplicación web**, como los parámetros de configuración (título del sitio web, idioma, permisos…), los parámetros de diseño (logotipo, plantilla del sitio web, color, tipo de letra…), los parámetros de organización (número de post por página, categorías de las noticias…), el almacenamiento de los usuarios, el contenido y demás información de interés (como pueden ser los comentarios de los usuarios sobre los contenidos, encuestas, valoraciones…).

Los sistemas más famosos en la gestión de contenidos son **Drupal**, **Wordpress** y **Joomla**.

Antes de concretar las características propias de cada uno de los sistemas de gestión de contenidos citados anteriormente, es importante destacar las funcionalidades comunes que comparten estos tres CMS.

La primera de ellas es la **gestión del contenido,** donde el gestor puede ver, editar y borrar los artículos, páginas y archivos de la página Web. Normalmente, también se dispone de un buscador para encontrar los artículos más fácilmente a través de un filtro (por categorías, palabras, fecha, etc.).

El editor de contenido que tienen suele ser muy similar a los editores de texto tipo Word, en el que se pueden insertar imágenes, hipervínculos, vídeos, música, etc. Una vez terminado el artículo, se pueden seleccionar diferentes configuraciones como la categoría a la que pertenece, el autor, la fecha, habilitación de comentarios…

La segunda funcionalidad común es la **configuración del sitio Web,** donde se le permite al gestor cambiar la mayor parte de las parametrizaciones de carácter general que no son elementos estructurales, de contenido o usuario como el título, slogan, misión, correo de contacto, fecha, hora, etc.

La tercera característica que comparten estos sistemas es el bloque de **presentación,** donde el gestor puede modificar el estilo de la página Web, cambiando la plantilla, los colores y elegir los complementos/plugins que aparecerán en las columnas laterales.

El cuarto bloque que tienen en común es el de **informes**, donde se pueden visualizar las estadísticas de la página Web (últimas visitas, páginas más vistas, etc.).

Y la quinta característica que comparten es la **gestión de usuarios** donde el gestor puede dar de alta usuarios, y asignarles diferentes roles, los cuales se traducen en privilegios para cada una de las funcionalidades y módulos de la página Web.
Por ejemplo, el rol de “gestor” tiene todos los privilegios de edición y configuración de la página Web, otro rol puede ser el de “redactor”, el cual solo puede escribir artículos y publicarlos, el rol de “moderador” le asigna los privilegios de validación y edición de comentarios, etc.

Para entender un poco más como es el funcionamiento de un CMS, se ha decidido explicar más detalladamente el sistema de gestión de contenidos de ***Drupal***, por ser el gestor con más características propias que lo diferencian de los demás CMS, los otros dos sistemas de gestión de contenidos Wordpress y Joomla se enunciarán algunas características propias pero no se profundizará en ninguno de ellos para no repetir el contenido.

Drupal

Drupal es un sistema de gestión de contenido modular y muy configurable. Ha sido desarrollado en PHP en código abierto por una activa comunidad de usuarios. Destaca por la calidad de su código y por el énfasis en la usabilidad y consistencia de todo el sistema.

Con Drupal se pueden construir casi cualquier tipo de webs definiendo tipos de contenidos, estructuras, permisos, sistema de registro de usuarios, sistemas de categorización y módulos complementarios.

La manera con la que esta diseñado Drupal permite definir **tipos de contenidos** diferentes, sobre los que se podrán aplicar permisos, flujos de publicación, categorías, etc.

Los contenidos podrán trabajar con texto enriquecido, tener comentarios, aceptar anexos, estructurarse formando “libros” (capítulos y subcapítulos) e incluso generar salidas RSS.

La organización de la información se realiza fundamentalmente a través de un **sistema de categorías**. Estas categorías permiten una navegación por diferentes tipos de contenidos que pertenecen a un mismo tema.

Los contenidos pueden adoptar una estructura jerárquica a través del sistema de **menús**; y la página Web se estructura en **bloques** que se sitúan en distintas **zonas** (barra lateral izquierda, cabecera, pie, etc.) según el tema/plantilla usado.

2.3.2 Ejemplo de estructura de un sistema Drupal

Drupal ofrece una **gestión de usuarios** que permite darles de alta, y asignarles diferentes roles que permiten manejar los privilegios para cada una de las funcionalidades y módulos de la página. Cada módulo de Drupal ofrece sus propias opciones de permisos que se asignan a los distintos roles.

En cuanto a la personalización de la presentación, se usan las plantillas o temas. Cada tema define la visualización específica de los menús, bloques y página general, aplicando hojas de estilo CSS.

Desde el panel de administración, se puede manejar **la gestión del contenido, la construcción y configuración de la página, la gestión de usuarios y los informes.**

Gestión del contenido

En Drupal, la célula básica de información es el llamado **nodo**, en donde son guardados los contenidos. Sea el contenido del tipo que sea, su nodo tiene asociado a él una URL para acceder al contenido, esto hace que todo se pueda enlazar y acceder muy fácilmente.

Los nodos tienen asociadas algunas opciones que pueden ser usadas o no (publicar, activar comentarios, publicar en la página de inicio, etc.), aunque la activación de módulos complementarios dan la opción de añadir más campos a un tipo de nodo.

En Drupal hay dos tipos de nodos nativos: **story** (Artículo) y **page** (Página) aunque se pueden activar, crear e instalar más.

Desde el punto de vista del administrador, en la opción de "Administrar contenidos" se pueden realizar entre muchas otras acciones:

- Crear y editar nodos.
- Definir y editar los contenidos.
- Administrar y moderar comentarios.
- Buscar internamente contenidos según características (tipo, temas…).
- Configurar opciones de publicación.

Construcción de la página Web

El secreto de Drupal para conseguir su flexibilidad y facilidad en la creación de sitios Web es la organización en capas que aplica en el tratamiento de los contenidos.
Drupal estructura los contenidos en una serie de elementos básicos: bloques, menús, módulos y temas:

Bloques. Son cajas que se activan en diferentes zonas: cabecera, pie, central, columna izquierda… que vienen definidas en el tema.

Menús. Permiten presentar elementos de navegación, que serán presentados visualmente en bloques, o según el tema, en barras horizontales con diferentes efectos.

Módulos. Son elementos que permiten añadir nuevas funcionalidades a Drupal para adaptarlo a las necesidades de cada sitio.

Temas. Son los principales responsables de la apariencia gráfica o estilo con que se mostrará la información al usuario. El uso de temas posibilita la separación entre información y aspecto gráfico permitiendo cambiar el diseño o apariencia sin necesidad de modificar el contenido. Los temas juegan un papel análogo en Drupal al de CSS en los documentos HTML; de hecho, es habitual que cada tema incluya un documento CSS, pero incluyen también programas PHP que consiguen efectos más sofisticados y otros tipos de información. Un sitio Web puede tener un solo tema o dar la opción al usuario de elegir entre varios.

Configuración de la página Web

Permite cambiar la mayor parte de las parametrizaciones de carácter general que no son elementos estructurales, de contenido o usuario como el título, slogan, misión, correo de contacto, fecha, hora, etc.

Gestión de usuarios

Los miembros pueden registrarse o ser registrados por el administrador, para luego con un nombre de usuario y clave acceder a más servicios.

Se puede crear perfiles o roles de usuarios basados en permisos asignados a cada uno de ellos, por ejemplo, pueden existir grupos de usuarios que pueden publicar contenidos directamente, otros pueden necesitar aprobación de un tercer usuario que desempeña el rol de administrador, etc.

Cada vez que se active un módulo, éste ofrece nuevos permisos que pueden ser configurados y asignados a distintos roles de usuarios.

Informes

El administrador puede visualizar los informes y estadísticas de todo lo relacionado con el sitio Web (últimas visitas, páginas más vistas, etc.).

2.3.2 Panel de administración de un sistema Drupal

2.3.3 Ejemplo de sitio Web que usa Drupal

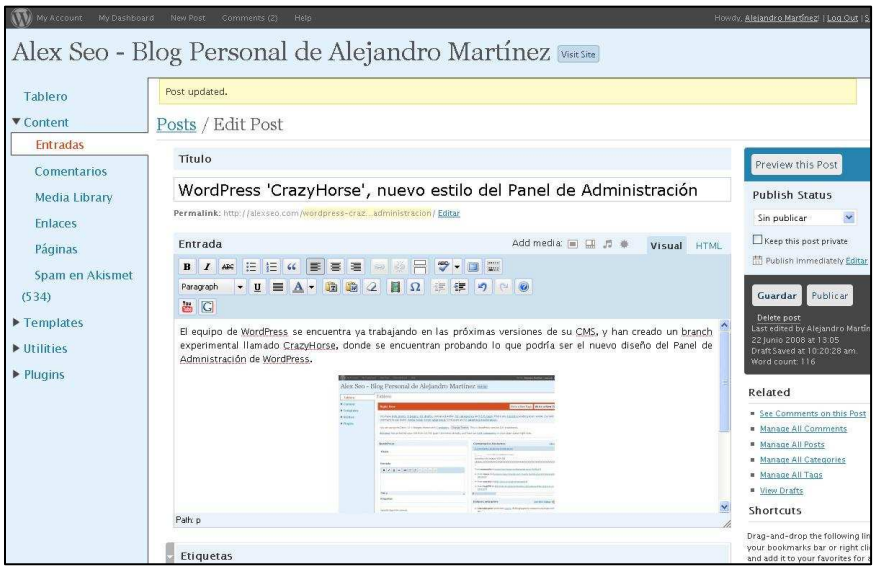
WordPress

Se trata de un gestor de contenidos libre bajo licencia GPL. Es uno de los sistemas más utilizados, y eso se traduce en una gran comunidad de usuarios, con soporte, ayuda y creación de nuevos plugins constantemente. Gracias a la gran cantidad de plugins creados hace que WordPress permita un nivel de personalización y configuración muy elevado.

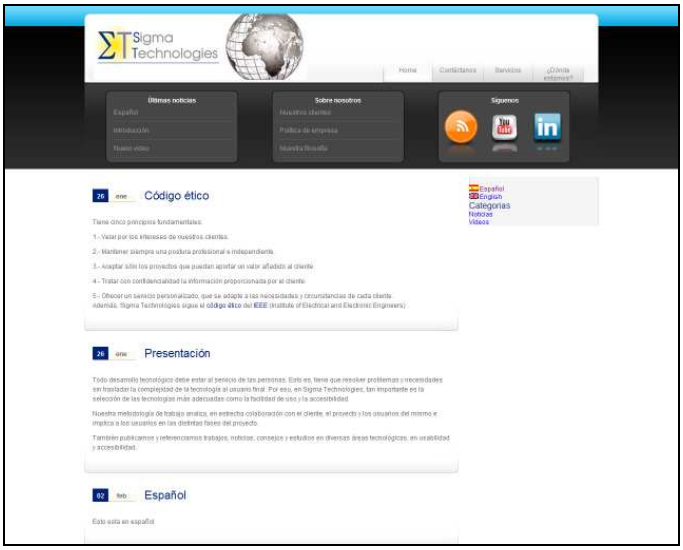
Este gestor de contenidos esta enfocado a la creación de blogs (Tipo de Web que recopila cronológicamente artículos de uno o varios autores), aunque con su gran cantidad de plugins se puede crear casi cualquier tipo de Web.

La gran ventaja por la que se ha destacado WordPress frente al resto de los CMS es por la rapidez de su instalación, ya que en apenas 5 minutos se puede tener en funcionamiento el sitio Web, y así poder centrarse en trabajar en el contenido de la página.

El panel de administración de Wordpress ofrece unas opciones muy básicas como **la gestión de usuarios, configuración de la Web, configuración de los contenidos, redacción de contenidos, gestión de enlaces, control de comentarios, presentación de la Web e informes de estadísticas.** Estas opciones de administración se pueden ampliar a través de los plugins instalables comentados con anterioridad, pudiéndose llegar a controlar casi cualquier aspecto de la página Web.



2.3.4 Panel de administración de un sistema Wordpress



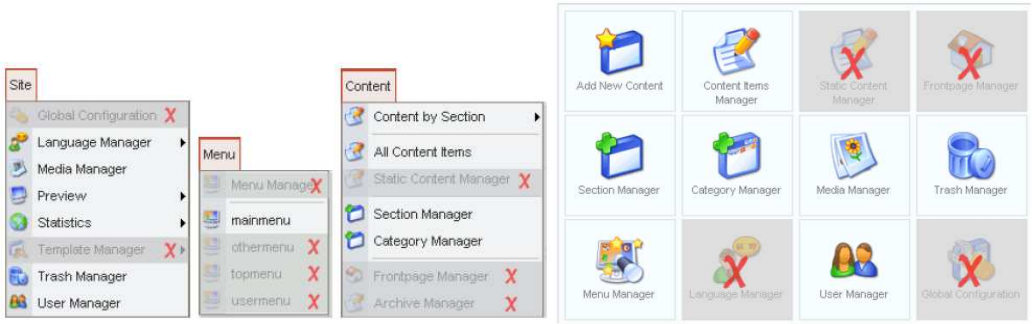
2.3.5 Ejemplo de una página Web que usa Wordpress

Joomla

Las características por las que destaca Joomla son la gran cantidad de plantillas ya creadas, ofreciendo al gestor del sitio gran variedad de posibilidades de diseño para su web, y la facilidad que ofrece para organizar información, aunque no es tan potente como otros sistemas de gestión de contenidos.

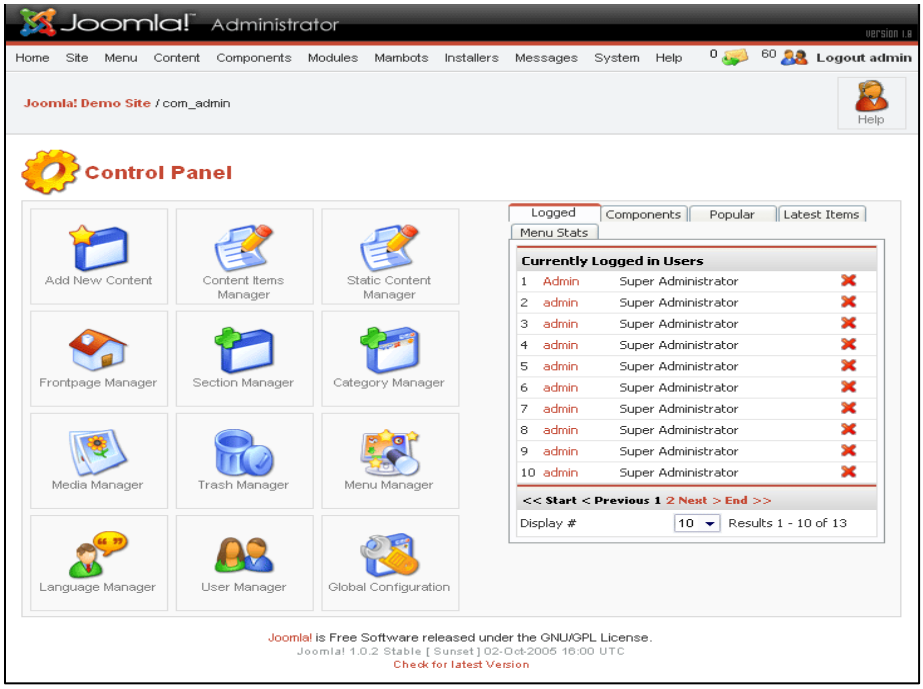
Su panel de administración se divide en dos partes: un menú desplegable en la parte superior, con todas las opciones que dispone el CMS, y de unos botones grandes de acceso directo a las tareas más comunes. Esto hace que su aprendizaje sea sencillo y muy útil para personas con poca experiencia en la utilización de sistemas de gestión de contenidos.

Esta facilidad de manejo va asociada a algo negativo para Joomla, y es que se convierte así en un sistema con una categorización de contenido rígida y unas opciones de configuración y de diseño limitadas.

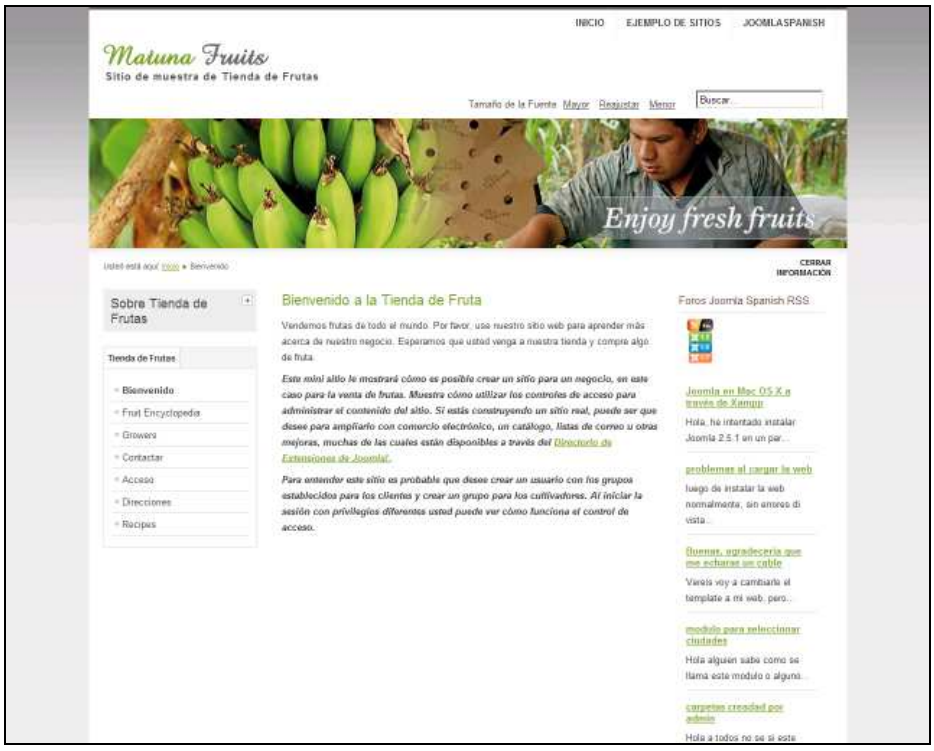


2.3.6 Menú superior y botones de ataco del panel de administración de Joomla

Para no ser muy repetitivo, se ha decidido no explicar el funcionamiento del panel de administración de Joomla, ya que es el sistema menos usado de los tres. Esto es debido a que aunque ofrece unas opciones generales muy parecidas (control de usuarios, gestión del contenido, configuración del sitio Web, plantillas de estilo…) a la de Drupal o Wordpress, sólo que Joomla ofrece una menor flexibilidad dentro de cada una de las opciones frente al resto de CMS



2.3.7 Panel de administración de un sistema Joomla



2.3.8 Ejemplo de una página Web que usa Joomla

2.4. Programación por demostración (Programming by example)

La programación por demostración (PBD), es una técnica que permite definir un programa que generaliza una secuencia de operaciones sin tener que aprender un lenguaje de programación. De esta manera, una persona sin conocimientos de un lenguaje determinado, puede mediante una interfaz dedicada u otras técnicas, transferir la información necesaria para que el ordenador programe lo requerido. Un primer ejemplo simple de programación por demostración son los macros que incluyen algunas aplicaciones ofimáticas, que permiten repetir operaciones complejas definidas interactivamente por el usuario en contextos diferentes. A continuación se muestran algunos otros ejemplos en los que se aplica con éxito la programación mediante demostración.

Programación de Robots por demostración

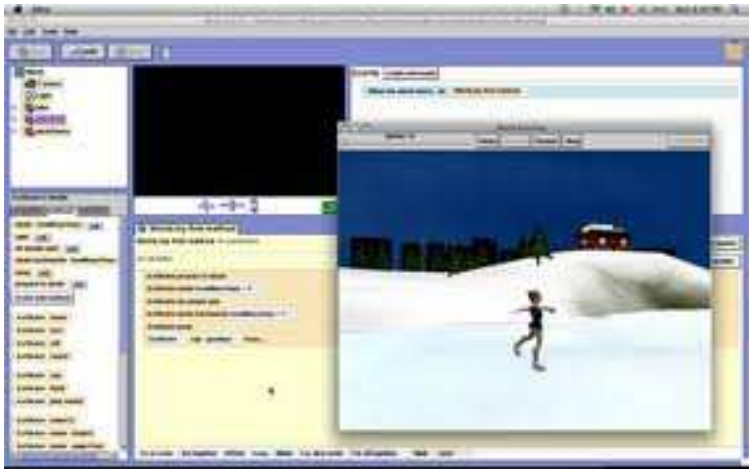
Esta técnica es un ejemplo claro de Programación por demostración, donde se consigue programar los movimientos de un robot sin necesidad de programarlo. Está basado en el teach-in, esto es, enseñar al robot la secuencia que debía reproducir mecánicamente. Esto se hace moviendo las partes del robot (a través de una interfaz o manualmente) en la forma que posteriormente se desea que se reproduzca. Una vez terminada esta fase de teach-in, el robot replica una o varías veces la secuencia que el técnico ha movido en la fase anterior.

2.4.1 Ejemplo de PBD sobre un robot

Alice, Software educacional

Alice es una aplicación desarrollada por la Universidad Carnegie Mellon, distribuido de forma gratuita para todas las plataformas, creada para enseñar los primeros pasos de la programación en un ambiente OOP (Object Oriented Programming), a través de una experiencia de juego.

Toda la plataforma funciona de manera “drag and drop”, o arrojar y soltar, con lo que programar en Alice es tan sencillo como seleccionar una palabra y arrastrarla a un listado de acciones que el personaje o el escenario llevará a cabo.



2.4.2 Screenshot de la herramienta Alice

Simulink, Programación visual

Simulink es un entorno de programación visual, que funciona sobre el entorno de programación Matlab.

La aplicación Simulink permite construir y simular modelos de sistemas físicos y sistemas de control mediante diagramas de bloques. El comportamiento de dichos sistemas se define mediante funciones de transferencia, operaciones matemáticas, elementos de Matlab y señales predefinidas de todo tipo.

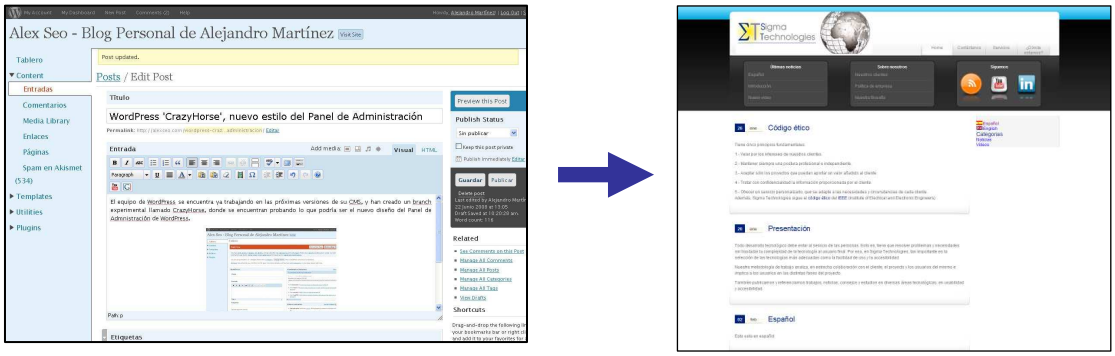
Mediante esta herramienta, a través de conexiones entre elementos y definiendo algunos parámetros de estos elementos, se pueden realizar simulaciones de sistemas físicos sin necesidad de conocer, ni de programar sobre Matlab.

Programación mediante demostración, los CMS

En el contexto del PFC, la programación por demostración se puede aplicar en la utilización de módulos de los sistemas de gestión de contenidos (CMS) para el diseño y configuración de aplicaciones web.

La interfaz de un CMS proporciona las herramientas suficientes para agregar y actualizar el contenido Web, sin necesidad de conocer los lenguajes de programación. Antes de la aparición de los sistemas de gestión de contenidos el conocimiento de los lenguajes de programación como HTML o JavaScript era esencial para actualizar un sitio Web. Esto significaba un costoso y lento proceso por parte de una persona para cada cambio y cada actualización.

Resulta deseable que la utilización de módulos de un CMS para diseñar y configurar una aplicación web se pueda realizar de forma interactiva más simple que la actualización paso a paso de aspectos dispersos de la aplicación. Por ejemplo, un diseñador de aplicaciones web puede basar su trabajo con un módulo determinado en la adaptación del mismo modificando únicamente algunos parámetros específicos, como el título y la posición y el color del fondo de un calendario. En ese caso el CMS debe permitir que el diseñador pueda realizar por una vez todas las actividades de diseño y configuración completas especificando cuáles de las acciones serán fijas en repeticiones posteriores y cuáles no y que el CMS cree un programa que le permita en otros casos especificar el diseño mediante la especificación interactiva únicamente de las acciones deseadas.



2.4.3 Ejemplo de programación por demostración sobre Wordpress

Capítulo 3

Diseño

La interactividad es un concepto utilizado en informática para señalar la acción recíproca a modo de diálogo entre el ordenador y el usuario. Si llevamos este concepto a los sistemas de gestión de contenidos, estos tienen un gran déficit en este campo.

En este capítulo, se diseñarán las soluciones a los principales problemas que tienen los sistemas de gestión de contenidos actuales con respecto a la **modificación, inserción y borrado de contenido y el uso de componentes** de una manera interactiva.
En primer lugar se describirá la situación actual de los CMS de cada uno de los aspectos anteriores citados (modificación, inserción, borrado y uso de componentes), y a continuación se detallará la solución diseñada al problema.

Para unificar conceptos, es conveniente definir los siguientes términos:

Contexto de ejecución. Se llamará así a la propia aplicación Web como la ve el usuario que visita el sitio Web.

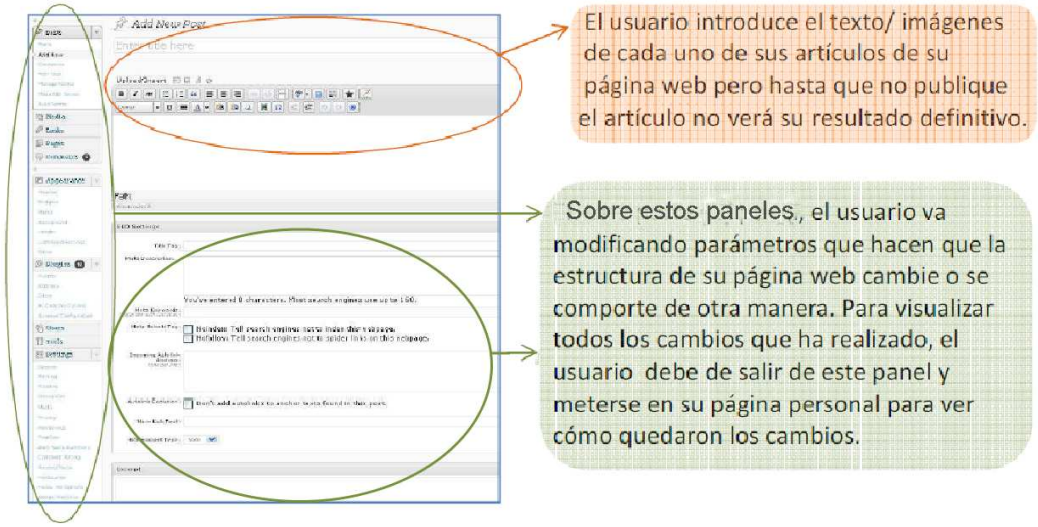
Contexto de edición. Se llamará así a la parte de la aplicación Web desde la cual el gestor puede modificar el contenido, es decir, el contenido del contexto de ejecución.

3.1 Situación actual en la modificación de contenido de un CMS

Los sistemas actuales de gestión de contenidos mantienen una estructura en la que la edición del sitio Web se realiza desde unos paneles de administración donde se muestran todas las opciones que tiene el gestor del sitio (tanto para añadir/editar contenido como para las configuraciones de diseño del sitio).

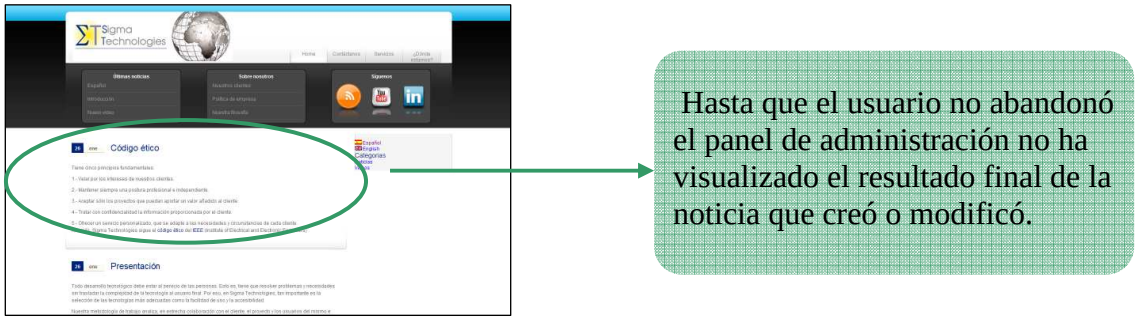
Esta manera de trabajar hace que el gestor no pueda visualizar el aspecto final de su página Web mientras esta editando/creando el contenido. Para entenderlo mejor, veámoslo mediante un ejemplo:

La siguiente imagen es de un típico panel de administración de un sistema de gestión de contenidos donde el gestor puede editar una noticia, eso si, sin ver su resultado hasta que no salga de su panel y se dirija a su página Web.



3.1.1 Ejemplo de panel de administración en Wordpress

Una vez que el gestor termine de completar el formulario anterior y pulse la opción de publicar, su aplicación Web cambiará de aspecto incluyéndose/editándose la noticia. Para visualizarla, debe salir del panel de administración y dirigirse a la URL de su sitio Web.



3.1.2 Publicación de una noticia en Wordpress

Se muestran a continuación dos situaciones en las que una filosofía semejante a la de la propuesta de este proyecto resuelve problemas análogos:

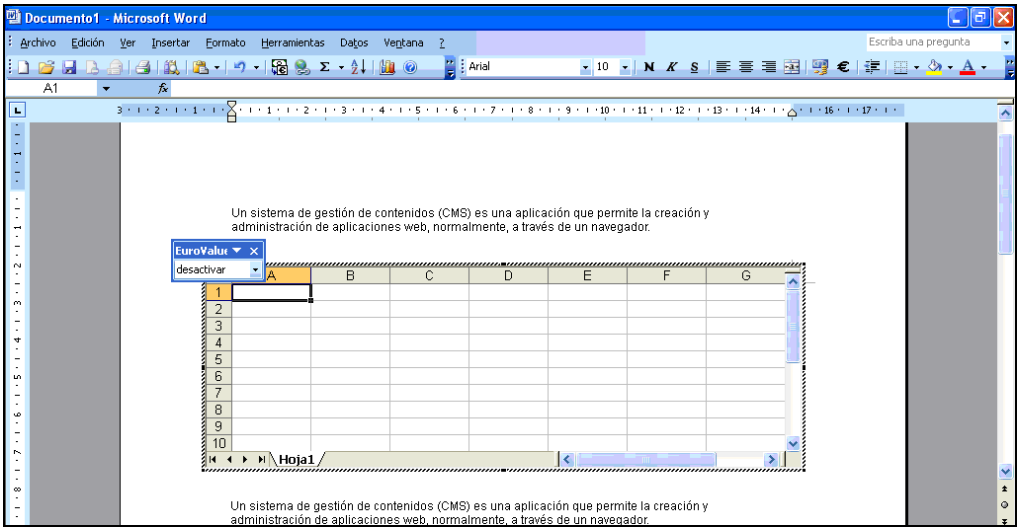
Caso 1, COM

En 1993 Microsoft introdujo **COM**, **Component Object Model**, una tecnología que permite la comunicación entre procesos y la creación dinámica de objetos en cualquier lenguaje de programación que la soporte.

Basándose en esta tecnología, a partir del Word 97, Microsoft introdujo a su editor una herramienta que permite insertar documentos y objetos de otras aplicaciones (***Menú Insertar*** >> ***Insertar Objeto***).

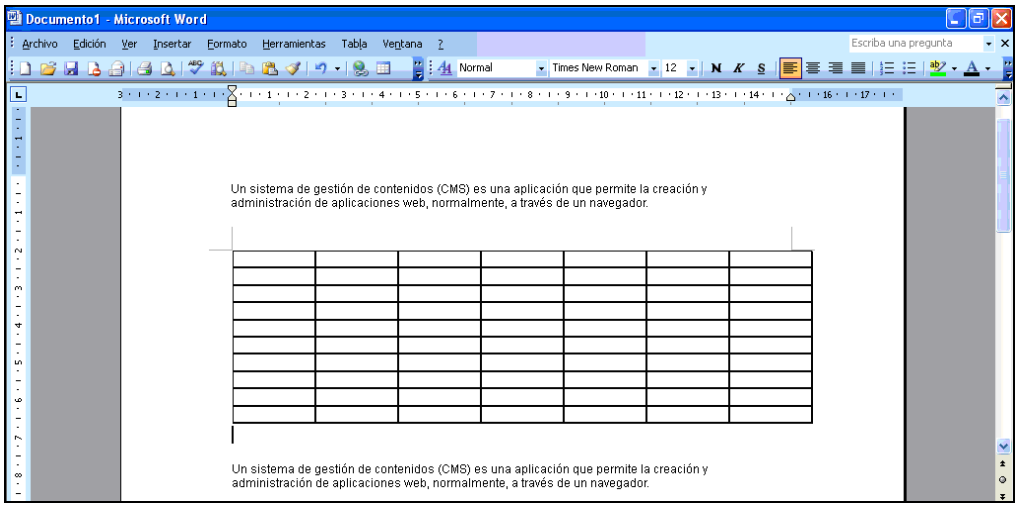
De esta manera el usuario puede incluir información a partir de archivos creados en otros programas de Microsoft Office o cualquier otro programa que admita los objetos vinculados e incrustarlos sobre su documento Word, como un editor de ecuaciones matemáticas.

La siguiente imagen muestra una inserción de un documento Excel sobre el documento Word:



3.1.3 Inserción de documento Excel sobre Word – modo de edición

Como se puede apreciar en la figura anterior, al insertar el objeto, el programa Word pasa a un segundo plano, dándole al objeto de Excel incrustado la posibilidad de interactuar con el usuario como si estuviera en el propio programa Excel y no en el programa Word.



3.1.4 Inserción de documento Excel sobre Word – modo de visualización

Cuando el usuario deja de editar sobre el objeto incrustado (en este caso el Excel), el programa Word vuelve a quedarse en un primer plano pudiéndose seguir editando el resto del documento.

Esta manera de trabajar es la que se pretende diseñar para los CMS actuales, solo que en nuestro caso en vez de trabajar sobre procesos diferentes se trabaja sobre una única aplicación Web.

Con este ejemplo he querido mostrar la filosofía a diseñar: el poder editar sobre la propia aplicación Web, elementos que están insertados en ella (diferentes contenidos).

Caso 2, Aplicaciones Web específicas

Existen hoy día aplicaciones Web que incorporan una filosofía muy parecida a la que buscamos. Un caso de esto es el portal de ***Linkedin***, cuya red social permite crear y editar de manera online parámetros de curriculums. Otro caso más cercano a los profesores de la Universidad Autónoma podría ser la aplicación Web ***Moodle*** que permite la edición de la página del profesor con la misma filosofía de trabajo con la que lo realiza Linkedin.

En la siguiente imagen vemos el estado normal de un perfil en el contexto de ejecución, donde un usuario cualquiera puede visualizar el curriculum con normalidad:

3.1.5 Perfil público de la red social Linkedin en "modo visualización"

Además, al administrador del curriculum (y sólo al administrador) le aparece un botón de edición de curriculum que pulsándolo, el curriculum que visualiza pasa a ser el siguiente:

3.1.6 Perfil público de la red social Linkedin en "contexto de edición"

Como se puede apreciar, la vista del curriculum se ha transformado a un **contexto de edición**, donde aparecen unos botones/enlaces que en el modo de visualización no estaban, y que sirven para ir añadiendo/editando el curriculum sin perder de vista en ningún momento el resto del sitio.

De esta manera, el usuario es capaz de ir editando su curriculum hasta conseguir la apariencia que desea que el resto de usuarios visualicen, de una manera rápida y precisa.

Llevando esta filosofía a los CMS, todo tiene una mayor dificultad ya que en la aplicación de Linkedin/Moodle la edición e inserción de elementos es muy limitada ya que se editan componentes atómicas preestablecidas y se mantiene una estructura sobre la cual trabaja el usuario/profesor.
En los CMS hay que permitir una mayor libertad (casi total) ya que cada aplicación Web tiene su propia estructura y contenido diferente y se pueden editar componentes con una complejidad arbitraria. La dificultad radica en que para llevar esta filosofía a un CMS es necesario definir un contexto de edición como parte de una herramienta, activable por lo tanto durante el diseño de una aplicación web arbitraria, en lugar de hacerlo para un contexto específico como el diseño de un currículum vitae.

3.2 Solución teórica en la modificación de contenido de un CMS

Si se combinan las dos ideas enunciadas anteriormente para llevarlas al ámbito de los sistemas de gestión de contenidos, la situación sería la siguiente:

Se partiría de una aplicación Web en un **"modo de visualización"** en el que aparece la aplicación Web en la forma en que la vería cualquier usuario al visitar el sitio.

3.2.1 Ejemplo de aplicación Web – modo de visualización

En este modo, el usuario podrá navegar por la aplicación y actuar con ella con normalidad sin tener ninguna percepción del sistema inteligente que hay por detrás.

Este sistema inteligente, es el "**contexto de edición"**, el cuál se activará al pulsar sobre el botón de edición, que únicamente estará disponible para al gestor del sitio.

Al cambiar al contexto de edición, la aplicación cambiará de aspecto tal y como se aprecia en la siguiente figura, insertándose sobre los diferentes contenidos unos botones/enlaces. Estos botones permitirán al gestor editar, insertar, mover… contenidos y elementos del sitio Web y lo que es muy importante, **sin cambiar en ningún momento de contexto**.

3.2.2 Ejemplo de aplicación Web – modo de edición con botones de edición

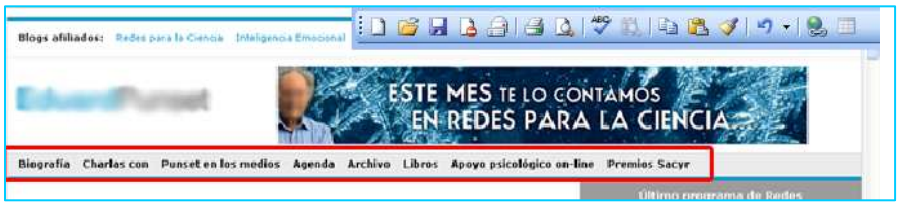
Como se puede apreciar en la figura, la apariencia de la aplicación es la misma, exceptuando los botones/enlaces que muestran al gestor de los elementos disponibles que tiene este para modificar.
Si el gestor pulsa alguno de los enlaces/botones, el sistema reconocerá de qué tipo de elemento se trata y desplegará una serie de opciones con las que el usuario podrá jugar hasta conseguir llegar al aspecto deseado.

3.2.3 Edición de texto sobre aplicación Web – modo de edición

Esto permitirá al gestor que en lugar de tener que intercalar durante el proceso de edición y diseño de la aplicación Web formularios fuera de contexto que modifican la base de datos subyacente, la propia interfaz de la aplicación muestre campos o controles adicionales que permitan su modificación progresiva.

Otra alternativa posible en vez de añadirse sobre los diferentes contenidos unos botones/enlaces, consiste en añadir un marco sobre un componente (de forma que remarque al gestor qué componente esta seleccionada).
Con la ayuda de un menú auxiliar y las flechas del teclado, el gestor puede desplazar el marco entre las componentes del árbol HTML y jugar con las distintas opciones que permite el menú.

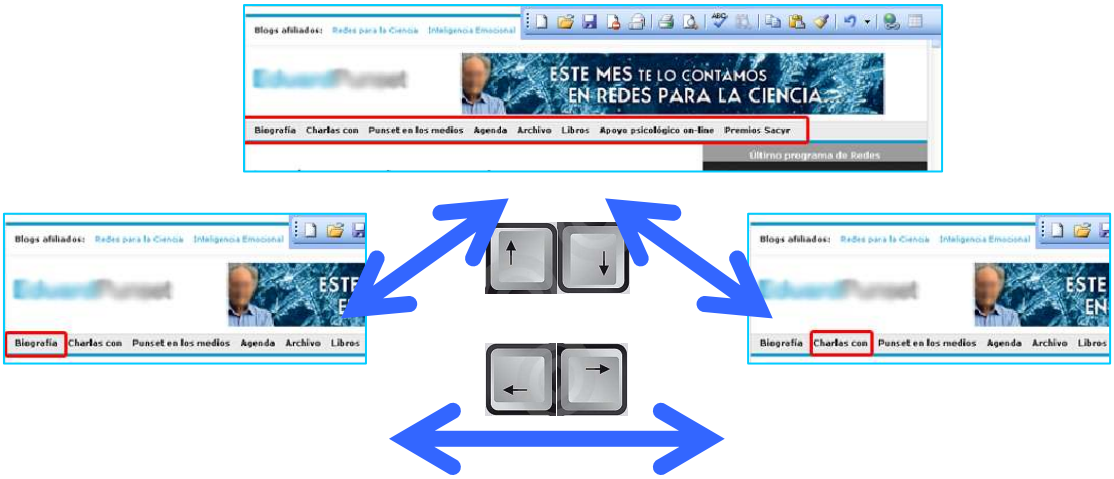
Analizando con detalle el ejemplo, el gestor empieza pulsando sobre el modo de edición de un aspecto similar al siguiente:



3.2.4 Ejemplo de aplicación Web – modo de edición con marco sobre componentes

Donde el gestor visualiza dos diferencias frente al modo de visualización: la aparición de un menú auxiliar en la parte superior derecha el cual le permitirá escoger entre las distintas opciones de edición (editar, eliminar, añadir componentes…) y también observará la aparición de un marco que engloba a un componente sobre el que actúan esas opciones del menú citado.

El gestor tiene la posibilidad de moverse sobre el árbol HTML usando las flechas de su teclado siendo las flechas laterales (→ ←) para moverse entre los hermanos del mismo nivel y las flechas superior e inferior (↓ ↑) para desplazarse sobre los padres o hijos.



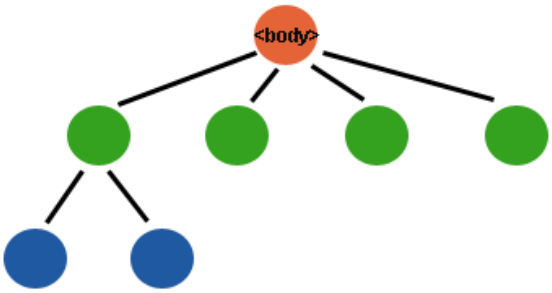
3.2.5 Ejemplo de selección de componentes en modo de edición

3.3 Diseño de la solución en la modificación de contenido de un CMS

En este apartado existen dos problemas que se deben diseñar: el insertar los botones/enlaces o marcos correspondientes al pasar del contexto de aplicación al contexto de edición y que al pulsar uno de estos botones de edición, se cambie la interfaz permitiendo al gestor dicha edición.

3.3.1 Inserción de los botones/enlaces

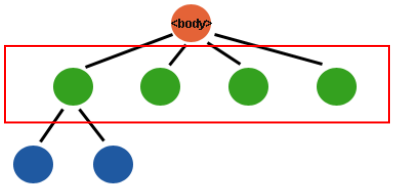
Para poder insertar dichos botones tenemos que fijarnos en el árbol HTML del sitio Web que tendrá una estructura similar a la siguiente:



3.3.1.1 Ejemplo de árbol HTML

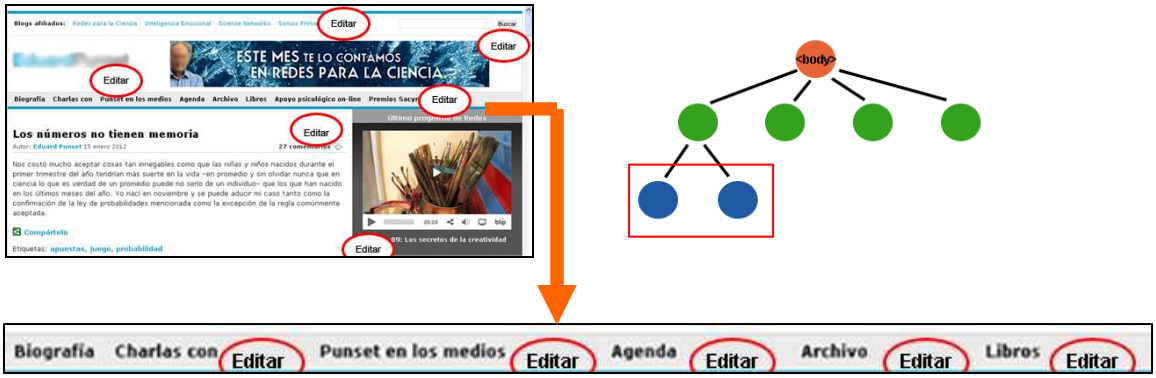
El padre “body” tendrá hijos (los círculos en verde) y estos a su vez pueden tener otros hijos (los círculos en azul).

Cuando el gestor pulse el botón editar, a los ***hijos editables*** (ver el punto 3.4 donde se habla del campo editable en la base de datos) que desciendan de body se le insertará el botón editar. La manera de hacerlo es mediante la API de DOM (concretamente la función **getChildNodes()**) con la que podemos recorrer los hijos de body e insertar en el árbol DOM de cada uno el botón/enlace de editar.



3.3.1.2 Edición de hijos de un árbol HTML y su resultado sobre Web

De la misma manera, desde un nodo (distinto al nodo body), podemos recorrer sus hijos. Así podremos recorrer y editar todos los nodos editables de la aplicación Web.



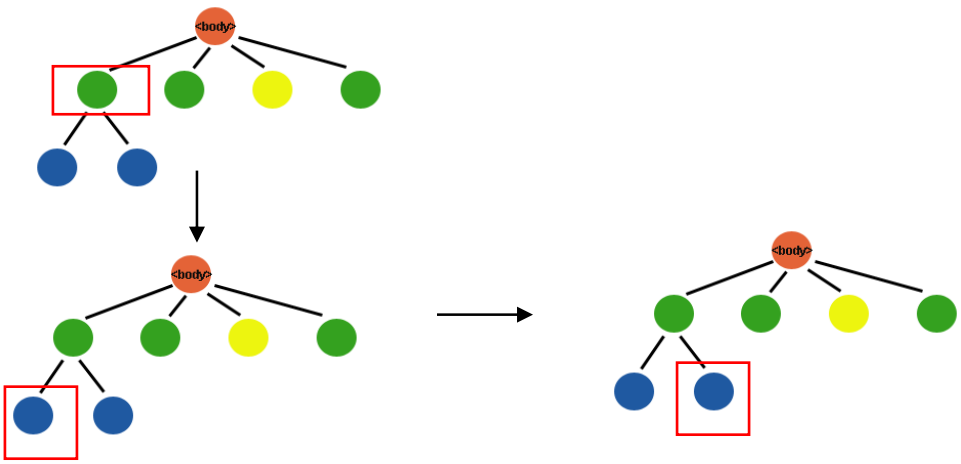
3.3.1.3 Recorrido sobre los hijos de un árbol HTML y su resultado sobre Web

3.3.2 Creación del marco y desplazamiento por los nodos del árbol

Para insertar sobre el árbol HTML el marco que seleccionará nuestro componente, se seguirá la misma metodología que en el apartado anterior.
Con la ayuda de la api de DOM podremos modificar el árbol del componente a seleccionar metiéndolo dentro de una tabla con un borde con lo que crearemos el efecto deseado.

A medida que el gestor se va desplazando por el padre, hijo o hermano de ese componente/nodo, ese marco de selección desaparecerá de la componente anteriormente seleccionada y aparecerá remarcando al nuevo componente.

Cuando el gestor desee editar el componente seleccionado y pulse sobre el menú auxiliar el botón de “edición”, el nodo será reemplazado para editarse (ver apartado siguiente).



3.3.2.1 Creación de marco sobre elementos de un árbol HTML

3.3.3 Edición del contenido

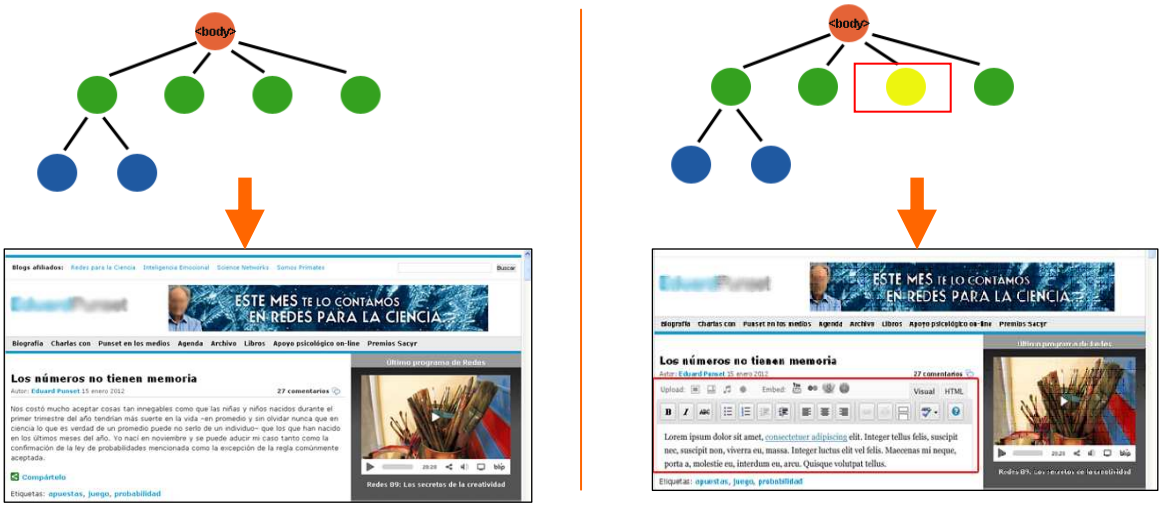
Al pulsar sobre el botón editar de alguno de estos hijos, mediante una consulta a la base de datos podemos averiguar el valor de su etiqueta (por ejemplo, una tabla tiene la etiqueta <table> y un párrafo tiene la etiqueta <p>).

Para cada etiqueta se le proporcionará un menú de edición u otro (por ejemplo, si fuera un *texto*, se le acoplaría un editor de texto, pero en cambio si fuera un *bloque de enlaces* (elemento artificial, el cual consiste en una lista de enlaces) se le dará la posibilidad de añadir/editar/eliminar enlaces).

3.3.3.1 Ejemplo de inserción de editor de texto

3.3.3.2 Ejemplo de inserción de editor de bloque de enlaces

La manera de colocar el menú de edición o editor de texto es una vez más reemplazando el nodo anterior por un nuevo nodo con el editor dentro del árbol con la API DOM.



3.3.3.2 Ejemplo de inserción de editor de bloque de enlaces

3.3.4 Edición de una componente

Aunque se profundizará mucho más de lo que es una componente en el apartado 3.3.6, es importante puntualizar que una componente a diferencia de un contenido, además de contener un contenido, contiene una lista de pares (parámetros - valor de parámetro) asociados a la componente.

Con lo que, de la misma manera que en el apartado anterior de edición de contenido, el gestor, al pulsar sobre el botón editar de alguno de los hijos del árbol, se producía una consulta a la base de datos para averiguar el valor de su etiqueta. Si se diese el caso en el que el valor de esa etiqueta correspondiese con el de una componente almacenada en el sistema el modo de edición sería algo distinto al enunciado en el apartado anterior:

Cada componente, además de tener su menú de edición de contenido (enunciado en el apartado anterior), tendrá además un menú de edición de parámetros donde el gestor podrá asignar un valor a cada parámetro que necesite la componente.

Menú de edición de contenido

Menú de edición de parámetros

3.3.4.1 Edición de una componente sobre la aplicación Web

Tal y como aparece en la imagen superior como ejemplo, la componente “noticia” podría tener un menú de edición de contenido del cuerpo de la noticia y un menú de edición de parámetros con el título de la noticia, el autor, etiquetas…

Este menú de edición de parámetros tiene una gran utilidad, ya que como valor de parámetros se podrán utilizar valores globales de la página Web como son el título de la página, el nombre del gestor que está editando la componente… Esto hace que la herramienta propuesta en este PFC sea más rápida y potente.

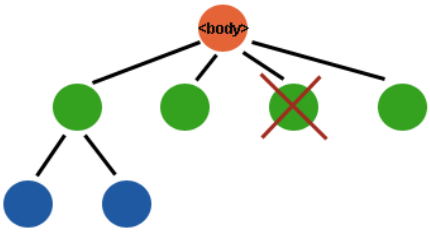
Incluso, aunque esto se detallará en el apartado de propuesta futura de este documento, para hacer aún más potente esta herramienta, se podrán utilizar valores de parámetros que podrán ser compartidos entre componentes y que si es el valor de dicho parámetro es editado, todas las componentes que contengan dicho parámetro verán actualizados automáticamente su valor.

3.3.5 Inserción y borrado de contenido

En el contexto de edición, el gestor tendrá además de la oportunidad de editar el contenido, la de inserción y borrado de contenido.

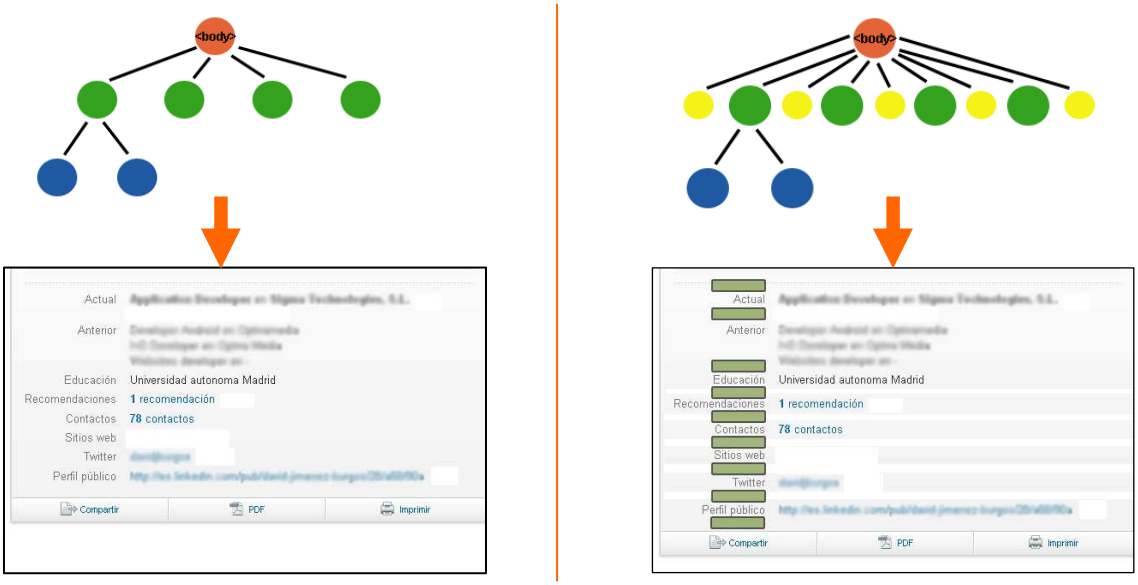
Si el gestor deseara eliminar un contenido, mediante el API DOM se puede eliminar ramas completas o un nodo en concreto.

Si el gestor deseara eliminar el nodo hijo de otro nodo padre (en nuestras figuras, los círculos azules), de la misma manera que hemos descendido por el árbol HTML para la opción de editar, para la opción de eliminar podremos ir descendiendo igualmente y así eliminar exactamente el nodo deseado.



3.3.5.1 Eliminación de un nodo en el árbol HTML

Para la opción de insertar, el gestor debe seleccionar primeramente dónde desea insertar el contenido. Para ello, al darle a añadir nuevo contenido o añadir componente de la biblioteca (ver en el siguiente apartado), la interfaz cambiará y le aparecerán unos botones ( ) para que el gestor seleccione la posición donde desea situar en el árbol el nuevo contenido.



3.3.5.2 Inserción de un nodo en el árbol HTML

Si una vez insertado el contenido, el gestor deseara moverlo a una posición concreta, el menú de edición proporcionará dicha opción.

3.3.5.1 Recolocación de un contenido en una aplicación Web

Esto es posible gracias a **HTML5**, ya que existen funciones que permiten mover contenidos a unas coordenadas exactas de la aplicación Web. Con esto, el gestor puede ver y seleccionar las coordenadas donde quiere que se sitúe cada contenido de su aplicación Web.

3.3.6 Uso de componentes

Este objetivo surgió debido a que muchos de los gestores de contenido actuales (como por ejemplo, Drupal) permiten almacenar estructuras de los contenidos más usados para que no se tengan que estar creándose continuamente.

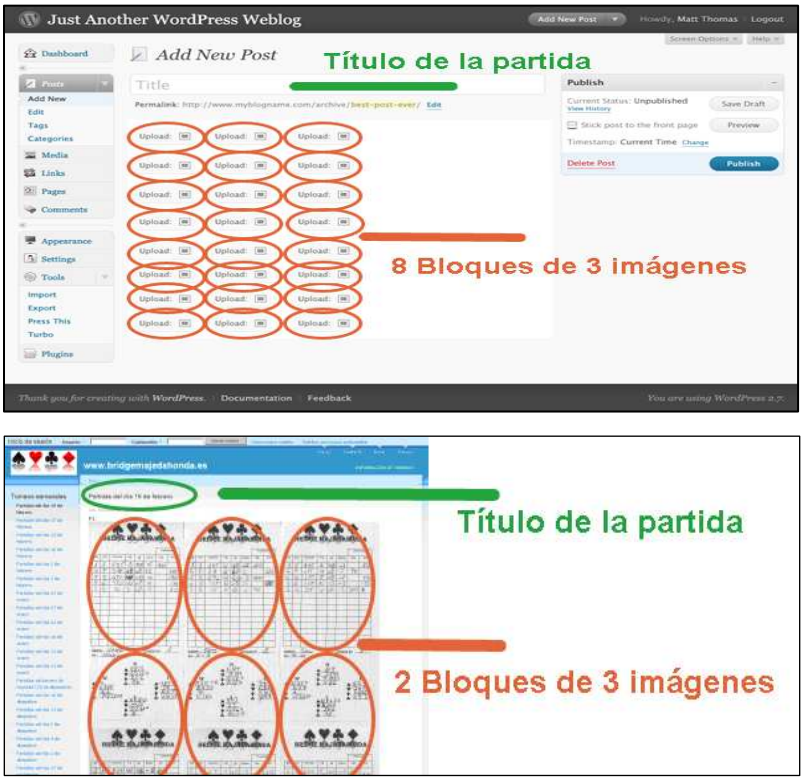
Cuando en un gestor de contenidos se esta creando/editando un contenido, el gestor puede seleccionar una opción que le permite almacenar con un nombre asociado la estructura del contenido en la base de datos; para que así, la siguiente vez que el gestor desee crear un contenido similar, pueda seleccionarlo y le aparecerá la estructura del contenido vacío para que únicamente se preocupe de rellenar el contenido.

Por ejemplo:

Bridgemajadahonda.es, es una aplicación Web que muestra los resultados de las partidas que se juegan en el club semanalmente. La mayoría de las noticias que aparecen en esta aplicación Web suelen tener la misma estructura: un título señalando la fecha de las partidas y 8 bloques de 3 imágenes cada bloque con las imágenes de las partidas de ese día.

Como la estructura de los contenidos se repite, el gestor tiene una estructura almacenada que al seleccionarla le aparece la plantilla de partidas vacía.

Al tener guardada la estructura, el gestor únicamente tiene que escribir el título de la noticia y seleccionar las 24 imágenes de las partidas y se le creará la estructura de partidas.



3.3.6.1 Ejemplo de componentes – Bridgemajadahonda.es

Diseño de la solución

Para nuestra aplicación Web, la situación de la que partimos es que el usuario va a querer almacenar un nodo con o sin hijos del árbol HTML, lo cuál se corresponde con la noticia o contenido que el gestor desea almacenar:

3.3.6.2 Almacenamiento de un componente – Árbol HTML

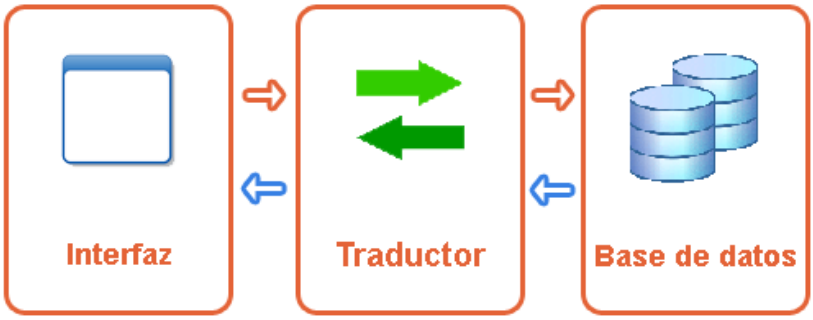
Para realizar esto, nos tenemos que apoyar en la API de DOM, que nos permite almacenar el código HTML de un nodo.
Guardando en la base de datos este código HTML, se podrá insertar dicho nodo en cualquier otro lugar y momento, facilitando así al gestor la creación de sus propias componentes sin necesidad de saber programar.

Una vez almacenado el código HTML, el gestor podrá reutilizarlo, insertándolo de la misma manera que se ha insertado un contenido en el apartado anterior (3.2 inserción de contenidos). Si el HTML almacenado incluye componentes, no hay ninguna diferencia en cuanto a la manera de inserción, exceptuando que el contenido y sus parámetros aparecerán con el valor por defecto que el programador de la componente haya establecido.

3.3.7 La base de datos

Tras plantear las distintas soluciones de los problemas para la interfaz, es importante resolverlas igualmente para el funcionamiento que hay por detrás de toda esa interfaz, es decir, para la **base de datos**.

El proceso que tiene como resultado final la visualización de la página en modo de visualización o modo de edición se realiza a través de tres componentes principales (interfaz, traductor, base de datos):



3.3.7.1 Almacenamiento de un componente – Árbol HTML

Los componentes son los siguientes:

- **Interfaz.** Este componente a su vez se separa en dos vistas, una la que se le ofrece al usuario (modo de visualización) y la vista que se le muestra al gestor cuando edita (modo de edición).
- **Traductor.** Este componente es el encargado de traducir los datos que se guardan en la base de datos a código HTML, consiguiéndose así que aparezca el sitio Web tal y como el gestor diseñó en el modo de edición.
- **Base de datos.** Este componente se encarga de guardar todos los datos, tanto de contenido como de estilo para su posterior uso.

Como se puede apreciar en la imagen, la comunicación es bilateral. El camino marcado en azul, refleja el proceso de cuando un usuario corriente entra en el sitio Web y visualiza la página en el modo de visualización, donde realmente por detrás lo que esta pasando es que se ha realizado una consulta a la base de datos para recoger todo los contenidos y estilos que componen la página y el traductor se encarga de convertirlo a HTML y conseguir así el resultado que visualiza el usuario sobre la interfaz.

El otro camino marcado en naranja, refleja cuando el gestor de contenidos realiza una acción de edición (crea/edita/borra algún contenido o crea un nuevo componente), la cual se debe ver reflejada en la base de datos. Es el traductor el encargado de traducir la información que el gestor del sitio introdujo en la interfaz del CMS y realizar esos cambios en la base de datos.

La base de datos se descompone en distintas tablas que el CMS utilizará y editará en distintas situaciones como cuando el gestor se logea, cuando se añade un nuevo componente, cuando se cambia el template o color del sitio Web, etc.

Para resolver los problemas enunciados en los apartados anteriores, se requerirá crear las siguientes tres tablas:

Tabla de contenidos

Campo	Descripción
Id	Número de identificación del elemento.
Father_id	ID del identificador del padre.
Position	Posición que ocupa como hijo de entre los hermanos.
Estyle	Parámetros de estilo (por ejemplo color de fondo o tipo de letra).
Tag	Nombre de la etiqueta (por ejemplo table, p, etc.).
Content	Contenido del elemento (puede haber HTML en él)
Editable	Booleano que determina si un elemento es editable o no.

Tabla de componentes

Campo	Descripción
Id	Número de identificación del elemento.
Tag	Nombre de la etiqueta (por ejemplo menú, header, etc.).
Content	Contenido del elemento (puede haber HTML en él)

Tabla de parámetros

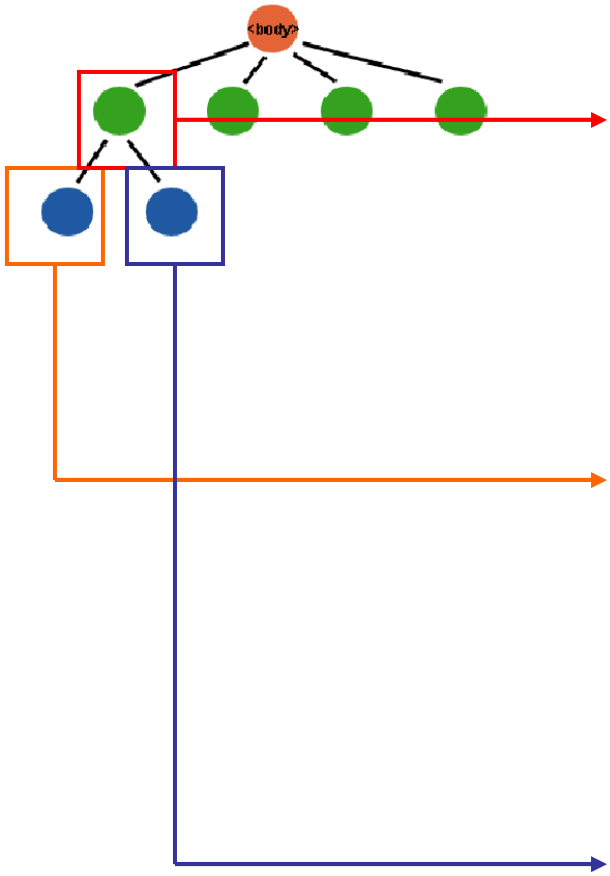
Campo	Descripción
Id	Identificador de la componente.
Parámetros	JSON con la lista de los parámetros1: nombre:valor1;nombre2:valor2

Con estas tres tablas, el traductor será capaz de resolver las dificultades para mostrar y guardar el contenido de la página Web:

Camino azul - Modo de visualización

La primera dificultad que se nos presenta es cuando un usuario corriente entra a la página Web. En ese momento el sistema debe realizar una consulta a la base de datos para obtener toda la información tanto de los elementos que componen el sitio como de los estilos de cada uno de ellos como de la página Web en general.
Con toda esta información el traductor es capaz de reconstruir una y otra vez el sitio Web tal y como el gestor lo diseñó con anterioridad desde el modo de edición.

Para entender mejor lo anterior, la siguiente figura muestra como el traductor reconstruye parte del árbol HTML, tras haber hecho, previamente, la consulta a la base de datos y obtener todos los campos de la tabla de contenidos.



Id: 34 (elegido por la base de datos)
Father id: 0 (body)
Position: 1 (ya que esta situado como primer hijo en el árbol HTML)
Estyle: marginTop=2, marginLeft=3
Tag: Bloque de enlaces
Content: (vacio)
Editable: true

Id: 42 (elegido por la base de datos)
Father id: 34 (ID del padre)
Position: 2 (ya que esta situado como segundo hijo en el árbol HTML)
Estyle: (vacio)
Type: P (párrafo)
Content: Google (enlace)
Editable: false

Id: 57 (elegido por la base de datos)
Father id: 42 (ID del padre)
Position: 1 (ya que esta situado como primer hijo en el árbol HTML)
Estyle: (vacio)
Type: P (párrafo)
Content: Marca (enlace)
Editable: false

La reconstrucción de esta parte del árbol HTML se traduce para el usuario del sitio en la aparición de un bloque de enlaces de la siguiente forma:

Google (con un enlace hacia el sitio de Google)
Marca (con un enlace hacia el sitio de Marca)

Se puede ver tras el ejemplo que con esta manera de trabajar, se consigue rellenar de una manera eficaz el árbol HTML alcanzándose la visualización esperada que el gestor de contenidos diseñó previamente.

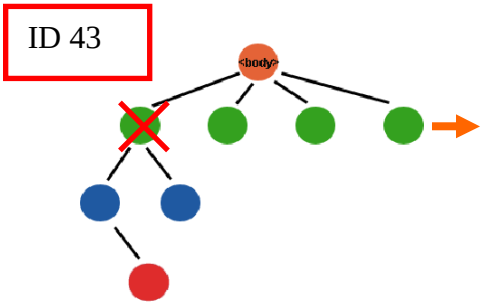
Camino naranja - Modo de edición

La segunda dificultad que se nos presenta es cuando un gestor de contenidos realiza una edición, borrado o inserción de un elemento. En ese momento el sistema debe transformar esa acción realizada por el gestor en una consulta adecuada para que se quede reflejado en la base de datos para que luego cuando un usuario entre en el sitio Web, el traductor sea capaz de reconstruir de nuevo esas acciones para que el usuario vea la página Web tal y como la diseñó el gestor.

Para entender mejor lo anterior, pondré un ejemplo de las 4 situaciones que se pueden dar:

Eliminación de contenido

Cuando el gestor de contenidos realiza una acción de eliminación de algún contenido, el traductor realiza un borrado en la base de datos contra la ID del contenido y de todos sus hijos, consiguiéndose así la eliminación de una rama completa del árbol.

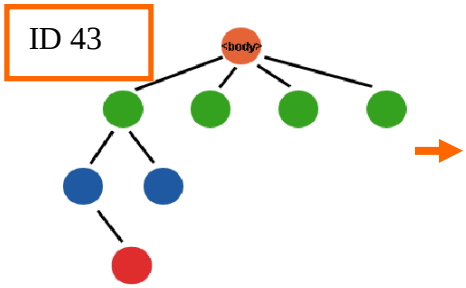


El traductor elimina en la base de datos el contenido con la ID=43 y de todos los contenidos que a su vez tengan como padre la ID 43 (y así sucesivamente con toda la rama del árbol HTML si hubiera más hijos hacía abajo).

Edición de contenido

Como se explicó en el punto 3.2, cuando el gestor pulse sobre el botón de edición sobre algún componente editable, ser realizará una consulta a la base de datos contra la ID del elemento para así determinar el TAG y mostrar al gestor el menú correspondiente de edición para dicho componente o el editor de texto si no hubiera ningún menú de edición establecido para ese TAG. A su vez, si ese TAG está asociado a una componente, se mostrará al gestor el menú de edición de parámetros de dicha componente.

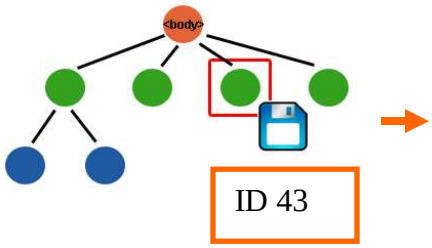
Cuando el gestor termine de realizar la edición de algún contenido, el traductor realizará una edición en la base de datos contra la ID del contenido realizando un volcado con la nueva información de dicho contenido y de los valores de los parámetros en el caso de ser una componente.



El traductor edita en la base de datos el contenido con la ID=43 cambiando así su estilo y/o contenido según el gestor de contenidos diseño en el modo de edición.

Creación de contenido

Cuando el gestor de contenidos realiza una acción de creación de algún contenido, el traductor realiza una adición en la base de datos insertando toda la información del nuevo contenido poniendo como ID del padre a la ID correspondiente al nodo padre del nuevo elemento y como posición, la posición que ocupa como hijo dentro de sus hermanos.



El traductor inserta en la base de datos el nuevo contenido con la ID=43 insertando así su estilo y/o contenido según el gestor de contenidos diseño en el modo de edición, además incorporará la información del padre (FatherID=0 porque su padre es el nodo body) y la información que ocupa entre sus hermanos (posición=3).

Almacenamiento de un nuevo componente

Cuando el gestor de contenidos realiza una acción de almacenamiento de algún elemento, el traductor realiza una adición en la base de datos en la tabla de componentes insertando toda la información del nuevo elemento.

Cuando se de la situación de que el gestor de contenidos quiera reutilizar ese componente, el traductor recogerá el texto HTML de ese elemento y lo insertará donde el gestor determine dentro del árbol HTML tal y como se explicó en el apartado 3.3.6. Posteriormente cuando el gestor guarde esa estructura de la página Web, el traductor reflejará esa acción como si el usuario hubiera generado un nuevo

elemento y realizará las mismas acciones que se han explicado en este mismo apartado en la parte de creación de contenidos.

En el caso en el que el texto HTML tenga componentes internas dentro del código, y el gestor decida reutilizar la componente almacenada, las componentes y sus parámetros quedarán inicializados con su valor por defecto (que el programador de la componente haya decidido). También se crearán los registros oportunos en las tablas de la base de datos según quede la nueva estructura de árbol de la página Web, y se creará un registro nuevo en la base de datos de parámetros con el identificador único de la componente insertada y la lista de par de valores (nombre parámetro – valor de parámetro) para que el traductor pueda reconstruir la componente con sus parámetros y contenido cuando un usuario visite el sitio Web.

3.3.8 Componentes almacenados por defecto

Gracias al modelo diseñado en los puntos anteriores, los programadores experimentados podrán crear y añadir *componentes por defecto*.

Un componente por defecto es un elemento que el gestor puede usar como si fuera uno de sus componentes almacenados pero la diferencia es que estos elementos tienen un menú editable creado especialmente para este elemento. Esto permitirá que el gestor pueda editar dichos componentes de una manera más sencilla y rápida.

Por ejemplo, un posible componente por defecto podría ser el elemento *bloque de enlaces* mencionado anteriormente que consiste en una lista de enlaces a otras páginas.

El programador podrá diseñar y acoplar a nuestro CMS un menú de edición con unas opciones más precisas que harán la edición de dicho componente más sencilla (por ejemplo con las opciones añadir, editar o eliminar enlace).

Esta adición al sistema provocará que cuando el gestor elija en el modo de edición, el editar un componente con el TAG “bloque de enlaces”, el sistema determinará que existe un menú de edición y lo mostrará en lugar de mostrar un editor de texto (que es el editor por defecto).

Además del menú de edición, el programador deberá diseñar un menú de parámetros para que el gestor a la hora de editar la componente, pueda asignar a un parámetro el valor deseado.
El valor de los parámetros pueden ser o bien valores que introduzca el gestor o valores globales de la Web que el gestor seleccionará cuando edite la componente en el modo de edición, estos valores a seleccionar pueden ser como el nombre del sitio, el autor, el idioma…que son comunes para todas las componentes de la Web.

La creación de una nueva componente, también requerirá por parte del diseñador, la creación de una función que sea la encargada de a través del contenido y parámetros que requiera la aplicación, devolver el código HTML que deberá ser acoplado en el árbol de la aplicación Web para su visualización.

El diseñador de componentes deberá encapsular todos estos archivos en un .zip con un nombre único entre todas las componentes existentes en el sistema para que la herramienta pueda descomprimirlo y posteriormente utilizarlo.
La utilización de un nombre único entre todas las componentes del sistema, es un requisito fundamental para que la parte de la herramienta encargada en la construcción del HTML para mostrarle al usuario, no confunda unas componentes con otras.

3.3.9 Estructura de los documentos y componentes

Dada la tabla de la base de datos explicada en apartados anteriores, existe un parámetro llamado “Content”, que consiste en el contenido que contiene la componente.

La estructura de dicho contenido es muy importante para que “el traductor” de nuestra propuesta del CMS pueda traducir toda la información exactamente como el gestor lo dejó cuando lo editó en el modo de edición.

Dicha estructura del contenido de la componente debe ser texto según una extensión de XHTML que incluye etiquetas especiales de tres tipos diferentes:

- Etiquetas para indicar valores que se extraen de una base de datos. Este tipo de etiquetas indican que para recoger el valor, se debe hacer una consulta a una base de datos.

Así, una etiqueta **<SQL>** *CONSULTA* **</SQL>** contendrá en su interior directamente la consulta SQL que debe realizarse para así obtener como respuesta el valor requerido.

- Etiquetas que indican la inclusión de valores de parámetros. Este tipo de etiquetas indican que para recoger el valor, se debe recoger el valor de de un parámetro situado en la tabla ‘parámetros’ de la base de datos.

Así, una etiqueta **<PARAM>** ID_PARÁMETRO**</PARAM>** contendrá en su interior el identificador del parámetro. Con esta información se realizará una consulta a la tabla parámetros de la base de datos y así obtenerse el valor de un parámetro.

- Etiquetas para indicar la inserción de otras componentes o contenidos. Este tipo de etiquetas indican que el contenido, es un HTML correspondiente a una componente.

Así, una etiqueta **<COMPONENT>** ID DE COMPONENTE **</COMPONENT>** contendrá en su interior el identificador de la componente. Con esta información se realizará una consulta a la tabla componentes de la base de datos y así obtenerse toda la información necesaria para llamar a la función 'traductor' de dicha componente y obtener así el HTML deseado.

Del mismo modo, un documento, es una componente que cuelga directamente del padre body en el árbol HTML.

Capítulo 4

Conclusiones y trabajo futuro

Conclusiones

A lo largo de este proyecto final de carrera se ha presentado la situación actual de los sistemas de gestión de contenidos, viendo su gran déficit en cuanto a interactividad.

Los CMS actuales comparten un aspecto negativo: el contexto donde se gestiona el contenido no es el adecuado. La manera de trabajar de los CMS actuales es que la edición del sitio Web se realice desde unos paneles de administración a modo de formularios que se encuentran fuera del contexto de la aplicación, con lo que el gestor no puede visualizar cómo va a quedar esa edición de su aplicación Web hasta que no salve los cambios y vuelva a entrar a la aplicación Web como un visitante más del sitio.

Gracias a la propuesta de diseño de CMS diseñada en este PFC, el gestor mediante la propia interfaz de la aplicación podrá realizar una modificación de todo el contenido de su Web mientras ve el resultado tal y como va a quedar cuando la Web sea visitada por un usuario.

Aunque la profundidad de este documento es limitada, la idea de este PFC es hacer una propuesta de diseño que sea capaz de realizar todas las funciones que realiza un CMS actual como la edición de contenido, adicción de contenido, creación de componentes… pero de una manera interactiva, viendo en todo momento cómo quedará el sitio Web después de dichos cambios.

Esta forma de trabajar permitirá trabajar sobre los CMS a una mayor velocidad y de una forma más intuitiva, consiguiéndose casi eliminar la curva de aprendizaje, ya que la gestión Web se asemejará a una edición de texto desde el programa Microsoft Word que tanto estamos acostumbrados.

Trabajo a futuro

Como trabajo a futuro de la herramienta enunciada en este documento, se ve conveniente trabajar en dos aspectos distintos:

Parámetros entre componentes

Dado que en el estado del arte de este proyecto se enunció la programación por demostración, esta se podría utilizar para definir editores interactivos de componentes del CMS sin necesidad de programar:

Tal y como se ha explicado en el apartado de edición de una componente de este documento, una componente además de contener un contenido, contiene una lista de pares (parámetros - valor de parámetro) asociados a la componente.
Estos valores de parámetros, tal y como esta enunciado en este PFC son valores o bien insertados por el propio gestor o bien valores globales de la aplicación Web como pueden ser el título del sitio Web, el nombre del administrador, el idioma predeterminado…

Una línea de investigación que podría hacer aún más potente esta propuesta, es que el gestor de la aplicación Web pueda crear parámetros que puedan ser utilizados por diversas componentes. Si se decidiese editar el valor de dicho parámetro, todas las componentes que lo utilicen verán actualizados automáticamente su valor (sería como dar la posibilidad al gestor de crear un parámetro global de la aplicación Web predefinido por él).
A su vez, se podría dar la oportunidad de que el gestor pudiera elegir como valor de parámetro de una componente el valor de parámetro de otra, siguiendo un funcionamiento semejante a las fórmulas en las hojas de cálculo.
De esta manera, si el valor del parámetro es editado, todas las componentes que tengan una dependencia con dicho parámetro se verán modificadas automáticamente

Aplicación móvil

Se ha visto que la fusión de conceptos tecnológicos tales como Internet y dispositivos móviles ha hecho que surja una auténtica revolución informática, con lo que queda evidente que si consiguiéramos llevar lo explicado en este proyecto final de carrera a un entorno móvil se abriría una importante línea de investigación.

Pienso que la creación de una **aplicación móvil** que fuera un gestor de contenidos sería una idea revolucionaria en este mercado, ofreciendo así a las personas encargadas de gestionar su sitio web la posibilidad de poder hacerlo desde cualquier sitio del mundo y únicamente con un dispositivo móvil con Internet. A su vez, se podría aprovechar las pantallas táctiles de los dispositivos móviles para ganar en interactividad con el sitio Web.

Capítulo 5

Anexos y Referencias

Referencias

[1] Sun's Official Java EE Tutorial , http://java.sun.com/javaee/5/docs/tutorial/doc

[2] Nicholas C. Zakas, Jeremy McPeak, Joe Fawcett, “Professional Ajax”, Wrox.

[3] Allen Cypher, “Watch What I Do”, MIT Press.

[4] Henry Lieberman, “Your Wish is my command, Programming by example”, Morgan Kaufmann Publishers.

[5] Brian Travis, “Pro Drupal 7 for Windows Developers”, Apress.

[6] Eric Tiggeler,“Joomla! 1.6: Managing Site Users with Access Control” Packt.

[7] April Hodge Silver, “WordPress 3 Complete” Packt Publishing.

[8] Especificaciones de HTML: http://www.w3.org/TR

[9] Especificación de JavaScript: http://www.ecma-international.org/publications/standards/Ecma-262.htm

[10] Especificación de Java EE 6: http://jcp.org/en/jsr/detail?id=316

Glosario

Base de datos. Conjunto de datos pertenecientes a un mismo contexto, almacenados sistemáticamente para su posterior uso**.**

CKEditor. Editor WYSIWYG de texto para ser utilizado dentro de las páginas web.

DOM. API que proporciona un conjunto funciones para representar documentos HTML y XML, pudiendo acceder a ellos para manipularlos.

JSON: Acrónimo de JavaScript Object Notation, es un formato ligero para el intercambio de datos.

Sistema de gestión de contenidos (CMS). Programa que permite crear una estructura para la creación y administración de contenidos, principalmente en páginas web.

Wordpress / Drupal / Joomla. Sistemas de gestión de contenidos.

WYSIWYG. Es el acrónimo de "What You See Is What You Get", se aplica a los programas que permiten escribir un documento viendo directamente el resultado final.

Programación por demostración. Técnica que permite definir una secuencia de operaciones sin tener que aprender un lenguaje de programación.

Linkedin. Es una red social orientada a los negocios. Un perfil en Linkedin es un resumen de tu experiencia y logros profesionales. (www.linkedin.com)

Anexos

A. Comparativa entre Drupal, Joomla y Wordpress

Instalación

WordPress está basado en instalaciones estándar de PHP + MySQL, disponibles en todos los servidores. Dispone de una famosa instalación *en 3 pasos* y las actualizaciones se realizan desde el mismo panel de administración con un clic.

La interfaz de WordPress ofrece una interfaz ordenada con menús y submenús, cuyo orden ha sido sometido mediante encuesta a los usuarios y actualmente ofrece un equilibrio inmejorable entre funcionalidades y sencillez.

Número de temas/plantillas y complementos

Una de las ventajas de usar un sistema de gestión de contenidos es que se pueden usar plantillas para que nuestra web tenga una apariencia profesional sin que sepamos nada de diseño web y módulos para ofrecer una gran cantidad de funciones sin necesidad de saber programar.

En el caso de los complementos, en Joomla, es mucho más frecuente que los módulos de más calidad sean de pago, en cambio Drupal y WordPress publican en sus páginas solo los complementos que son gratuitos y con licencias libres, aunque también pueden adquirirse en otras páginas otros complementos de pago.

El sistema de Wordpress tiene alrededor de 7.000 módulos y 1.000 plantillas diferentes, Drupal ofrece en torno a 5.000 módulos y 600 plantillas, mientras que en el caso de Joomla es de algo más de 3.500 módulos y 3.000 plantillas.

Usabilidad

Es importante remarcar que con cualquiera de los tres sistemas, un usuario sin conocimientos de programación ni diseño puede ser capaz de crear un sitio web, pero no son igual de sencillos de usar:

WordPress es más inmediato para una web sencilla, pero esta principalmente diseñado para crear blogs con lo que si deseamos crear una web más compleja acabaremos teniendo algunos problemas. Hay plugins que ayudan a usar WordPress como sistema de gestión de contenidos más universal, pero tienen sus limitaciones, aunque poco a poco Wordpress esta evolucionando a una solución para la creación de cualquier tipo web genérica y no tanto como gestor de blogs.

Joomla tiene el problema de ser muy rígido en su organización del contenido. Esto obliga a planificar muy bien qué contenido tendrá nuestro sitio web y su estructuración, porque no es sencillo cambiarlo una vez tengamos haya mucho contenido publicado. La presentación del contenido es algo más rígida que el resto de sistemas, haciendo que muchas veces las webs desarrolladas con Joomla tengan un aspecto y una organización espacial muy semejante.

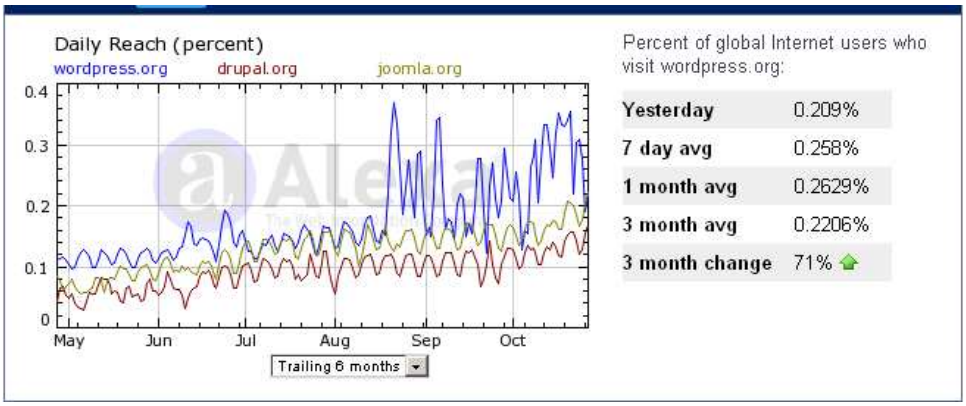
Drupal es más difícil de instalar y usar, y sus módulos son más difíciles de gestionar, pero existen complementos que permiten crear auténticas aplicaciones web sin necesidad de programar, con lo que si necesitamos hacer una web compleja, el esfuerzo dedicado a aprender a usar Drupal sin duda se verá recompensado.

Tendencias y cuota de mercado

La siguiente figura muestra la relevancia de los tres sistemas desde el 2005 hasta el 2009, viendo cómo los sistemas de Wordpress (rojo) y Drupal (azul) se han visto superados por Joomla (amarillo).

2.3.5 Gráfica de evolución de los principales sistemas de gestión de contenidos

Sin embargo como se puede observar en la siguiente figura, en cuanto a número de visitas de los sitios webs se muestra como Wordpress esta por encima de los otros dos sistemas debido al alto crecimiento de blogs durante estos años:



2.3.6 Gráfica de visitas de los principales sistemas de gestión de contenidos

Para tener una comparativa entre los 3 sistemas de gestión de contenidos más general se ha construido la siguiente tabla, donde se enfrentan a los CMS estudiados con anterioridad con las distintas características importantes que un usuario debe evaluar para elegir correctamente cuál de ellos escoger según sus necesidades:

----------	**Drupal**	**WordPress**	**Joomla**
Dificultad de instalación			
Curva de aprendizaje			
Número de plantillas			
Número de plugins			
Tendencias de uso			
Flexibilidad			

 Bueno/Fácil/Muchos

 Regular/Normal

 Malo/Difícil/Pocos

Capítulo 6

Presupuesto y Condiciones

PRESUPUESTO

1) Ejecución Material

- Compra de ordenador personal (Software incluido)..... ..... 2.000 €
- Material de oficina ..... 150 €
- Total de ejecución material ..... **2.150 €**

2) Gastos generales

- 16 % sobre Ejecución Material ..... 344 €

3) Beneficio Industrial

- 6 % sobre Ejecución Material ..... 129 €

4) Honorarios Proyecto

- 880 horas a 15 € / hora ..... 13200 €

5) Material fungible

- Gastos de impresión ..... 60 €
- Encuadernación ..... 50 €

6) Subtotal del presupuesto

- Subtotal Presupuesto ..... **15933 €**

7) I.V.A. aplicable

- 21% Subtotal Presupuesto ..... 3346 €

8) Total presupuesto

- Total Presupuesto ..... **19279 €**

Madrid, Noviembre de 2012

El Ingeniero Jefe de Proyecto

Fdo.: David Jiménez Burgos
Ingeniero Superior de Telecomunicación

PLIEGO DE CONDICIONES

Este documento contiene las condiciones legales que guiarán la realización, en este proyecto, de una herramienta altamente interactiva de páginas web. En lo que sigue, se supondrá que el proyecto ha sido encargado por una empresa cliente a una empresa consultora con la finalidad de realizar dicho sistema. Dicha empresa ha debido desarrollar una línea de investigación con objeto de elaborar el proyecto. Esta línea de investigación, junto con el posterior desarrollo de los programas está amparada por las condiciones particulares del siguiente pliego.

Supuesto que la utilización industrial de los métodos recogidos en el presente proyecto ha sido decidida por parte de la empresa cliente o de otras, la obra a realizar se regulará por las siguientes:

Condiciones generales

1. La modalidad de contratación será el concurso. La adjudicación se hará, por tanto, a la proposición más favorable sin atender exclusivamente al valor económico, dependiendo de las mayores garantías ofrecidas. La empresa que somete el proyecto a concurso se reserva el derecho a declararlo desierto.

2. El montaje y mecanización completa de los equipos que intervengan será realizado totalmente por la empresa licitadora.

3. En la oferta, se hará constar el precio total por el que se compromete a realizar la obra y el tanto por ciento de baja que supone este precio en relación con un importe límite si este se hubiera fijado.

4. La obra se realizará bajo la dirección técnica de un Ingeniero Superior de Telecomunicación, auxiliado por el número de Ingenieros Técnicos y Programadores que se estime preciso para el desarrollo de la misma.

5. Aparte del Ingeniero Director, el contratista tendrá derecho a contratar al resto del personal, pudiendo ceder esta prerrogativa a favor del Ingeniero Director, quien no estará obligado a aceptarla.

6. El contratista tiene derecho a sacar copias a su costa de los planos, pliego de condiciones y presupuestos. El Ingeniero autor del proyecto autorizará con su firma las copias solicitadas por el contratista después de confrontarlas.

7. Se abonará al contratista la obra que realmente ejecute con sujeción al proyecto que sirvió de base para la contratación, a las modificaciones autorizadas por la superioridad o a las órdenes que con arreglo a sus facultades le hayan comunicado por escrito al Ingeniero Director de obras siempre que dicha obra se haya ajustado a los preceptos de los pliegos de condiciones, con arreglo a los cuales, se harán las modificaciones y la valoración de las diversas unidades sin que el importe total pueda exceder de los presupuestos aprobados. Por consiguiente, el número de unidades que se consignan en el proyecto o en el presupuesto, no podrá servirle de fundamento para entablar reclamaciones de ninguna clase, salvo en los casos de rescisión.

8. Tanto en las certificaciones de obras como en la liquidación final, se abonarán los trabajos realizados por el contratista a los precios de ejecución material que figuran en el presupuesto para cada unidad de la obra.

9. Si excepcionalmente se hubiera ejecutado algún trabajo que no se ajustase a las condiciones de la contrata pero que sin embargo es admisible a juicio del Ingeniero Director de obras, se dará conocimiento a la Dirección, proponiendo a la vez la rebaja de precios que el Ingeniero estime justa y si la Dirección resolviera aceptar la obra, quedará el contratista obligado a conformarse con la rebaja acordada.

10. Cuando se juzgue necesario emplear materiales o ejecutar obras que no figuren en el presupuesto de la contrata, se evaluará su importe a los precios asignados a otras obras o materiales análogos si los hubiere y cuando no, se discutirán entre el Ingeniero Director y el contratista, sometiéndolos a la aprobación de la Dirección. Los nuevos precios convenidos por uno u otro procedimiento, se sujetarán siempre al establecido en el punto anterior.

11. Cuando el contratista, con autorización del Ingeniero Director de obras, emplee materiales de calidad más elevada o de mayores dimensiones de lo estipulado en el proyecto, o sustituya una clase de fabricación por otra que tenga asignado mayor precio o ejecute con mayores dimensiones cualquier otra parte de las obras, o en general, introduzca en ellas cualquier modificación que sea beneficiosa a juicio del Ingeniero Director de obras, no tendrá derecho sin embargo, sino a lo que le correspondería si hubiera realizado la obra con estricta sujeción a lo proyectado y contratado.

12. Las cantidades calculadas para obras accesorias, aunque figuren por partida alzada en el presupuesto final (general), no serán abonadas sino a los precios de la contrata, según las condiciones de la misma y los proyectos particulares que para ellas se formen, o en su defecto, por lo que resulte de su medición final.

13. El contratista queda obligado a abonar al Ingeniero autor del proyecto y director de obras así como a los Ingenieros Técnicos, el importe de sus respectivos honorarios facultativos por formación del proyecto, dirección técnica y administración en su caso, con arreglo a las tarifas y honorarios vigentes.

14. Concluida la ejecución de la obra, será reconocida por el Ingeniero Director que a tal efecto designe la empresa.

15. La garantía definitiva será del 4% del presupuesto y la provisional del 2%.

16. La forma de pago será por certificaciones mensuales de la obra ejecutada, de acuerdo con los precios del presupuesto, deducida la baja si la hubiera.

17. La fecha de comienzo de las obras será a partir de los 15 días naturales del replanteo oficial de las mismas y la definitiva, al año de haber ejecutado la provisional, procediéndose si no existe reclamación alguna, a la reclamación de la fianza.

18. Si el contratista al efectuar el replanteo, observase algún error en el proyecto, deberá comunicarlo en el plazo de quince días al Ingeniero Director de obras, pues transcurrido ese plazo será responsable de la exactitud del proyecto.

19. El contratista está obligado a designar una persona responsable que se entenderá con el Ingeniero Director de obras, o con el delegado que éste designe, para todo relacionado con ella. Al ser el Ingeniero Director de obras el que interpreta el proyecto, el contratista deberá consultarle cualquier duda que surja en su realización.

20. Durante la realización de la obra, se girarán visitas de inspección por personal facultativo de la empresa cliente, para hacer las comprobaciones que se crean oportunas. Es obligación del contratista, la conservación de la obra ya ejecutada hasta la recepción de la misma,

por lo que el deterioro parcial o total de ella, aunque sea por agentes atmosféricos u otras causas, deberá ser reparado o reconstruido por su cuenta.

21. El contratista, deberá realizar la obra en el plazo mencionado a partir de la fecha del contrato, incurriendo en multa, por retraso de la ejecución siempre que éste no sea debido a causas de fuerza mayor. A la terminación de la obra, se hará una recepción provisional previo reconocimiento y examen por la dirección técnica, el depositario de efectos, el interventor y el jefe de servicio o un representante, estampando su conformidad el contratista.

22. Hecha la recepción provisional, se certificará al contratista el resto de la obra, reservándose la administración el importe de los gastos de conservación de la misma hasta su recepción definitiva y la fianza durante el tiempo señalado como plazo de garantía. La recepción definitiva se hará en las mismas condiciones que la provisional, extendiéndose el acta correspondiente. El Director Técnico propondrá a la Junta Económica la devolución de la fianza al contratista de acuerdo con las condiciones económicas legales establecidas.

23. Las tarifas para la determinación de honorarios, reguladas por orden de la Presidencia del Gobierno el 19 de Octubre de 1961, se aplicarán sobre el denominado en la actualidad "Presupuesto de Ejecución de Contrata" y anteriormente llamado "Presupuesto de Ejecución Material" que hoy designa otro concepto.

Condiciones particulares

La empresa consultora, que ha desarrollado el presente proyecto, lo entregará a la empresa cliente bajo las condiciones generales ya formuladas, debiendo añadirse las siguientes condiciones particulares:

1. La propiedad intelectual de los procesos descritos y analizados en el presente trabajo, pertenece por entero a la empresa consultora representada por el Ingeniero Director del Proyecto.

2. La empresa consultora se reserva el derecho a la utilización total o parcial de los resultados de la investigación realizada para desarrollar el siguiente proyecto, bien para su publicación o bien para su uso en trabajos o proyectos posteriores, para la misma empresa cliente o para otra.

3. Cualquier tipo de reproducción aparte de las reseñadas en las condiciones generales, bien sea para uso particular de la empresa cliente, o para cualquier otra aplicación, contará con autorización expresa y por escrito del Ingeniero Director del Proyecto, que actuará en representación de la empresa consultora.

4. En la autorización se ha de hacer constar la aplicación a que se destinan sus reproducciones así como su cantidad.

5. En todas las reproducciones se indicará su procedencia, explicitando el nombre del proyecto, nombre del Ingeniero Director y de la empresa consultora.

6. Si el proyecto pasa la etapa de desarrollo, cualquier modificación que se realice sobre él, deberá ser notificada al Ingeniero Director del Proyecto y a criterio de éste, la empresa consultora decidirá aceptar o no la modificación propuesta.

7. Si la modificación se acepta, la empresa consultora se hará responsable al mismo nivel que el proyecto inicial del que resulta el añadirla.

8. Si la modificación no es aceptada, por el contrario, la empresa consultora declinará toda responsabilidad que se derive de la aplicación o influencia de la misma.

9. Si la empresa cliente decide desarrollar industrialmente uno o varios productos en los que resulte parcial o totalmente aplicable el estudio de este proyecto, deberá comunicarlo a la empresa consultora.

10. La empresa consultora no se responsabiliza de los efectos laterales que se puedan producir en el momento en que se utilice la herramienta objeto del presente proyecto para la realización de otras aplicaciones.

11. La empresa consultora tendrá prioridad respecto a otras en la elaboración de los proyectos auxiliares que fuese necesario desarrollar para dicha aplicación industrial, siempre que no haga explícita renuncia a este hecho. En este caso, deberá autorizar expresamente los proyectos presentados por otros.

12. El Ingeniero Director del presente proyecto, será el responsable de la dirección de la aplicación industrial siempre que la empresa consultora lo estime oportuno. En caso contrario, la persona designada deberá contar con la autorización del mismo, quien delegará en él las responsabilidades que ostente.