Creación de un Portal Web Docente

Jesús Vegas
Departamento de Informática
Universidad de Valladolid
jvegas@infor.uva.es

Capítulo 1

Internet y la Universidad

Antes de entrar en detalles técnicos sobre la creación de portales Web docentes resulta interesante analizar las consecuencias del desarrollo de las tecnologías de la información en la Universidad. Analizando los desafíos y las posibilidades se podrá centrar mejor la utilidad y los objetivos buscados al utilizar las tecnologías de la información, el Web, en la docencia universitaria.

1.1 La Educación Basada en la Tecnología

La educación basada en la tecnología (*Technology-Enabled Education*) es la educación que es posible gracias principalmente a la tecnología. Hasta los tiempos recientes, la mayor contribución a la educación por parte de la tecnología ha sido el invento de la imprenta por Gutenberg en 1450. La impresión de libros hizo posible la difusión del conocimiento y de las ideas a lo largo del mundo.

Ya en nuestro tiempo, la aparición de una nueva tecnología basada en las computadoras, la multimedia e Internet ha hecho que esté disponible multitud de información y que el conocimiento se pueda difundir rápidamente. Muchas de las cosas que realizamos habitualmente dentro de la educación han sido transportadas al Web. Pero la duda es si eso es realmente lo mejor que se puede hacer con estas nuevas tecnologías. El modo en que se enseña actualmente no ha variado mucho en los últimos siglos. Mientras un médico del siglo anterior no sería capaz de reconocer la tecnología utilizada hoy en día en los hospitales, un profesor de ese mismo siglo no observaría grandes diferencias en las herramientas que utilizamos hoy en día para enseñar.

El verdadero desafío está en encontrar en modo en que las nuevas tecnologías pueden crear nuevos entornos educativos.

La tecnología de la información, mediante la interacción, hace que los propios estudiantes sean conscientes de su propio aprendizaje. Además, la tecnología puede hacer que los estudiantes aprendan por sí mismos los aspectos básicos de una materia, permitiendo a los profesores centrarse en los aspectos más complejos e interesantes de las materias. Además, la tecnología puede abaratar los costes de la educación. Por último, la tecnología hace posible la cooperación entre los profesores y las universidades, y esta cooperación puede atraer más recursos.

1.2 El Negocio de la Educación

El negocio de la educación está sufriendo una evolución importante en los últimos años. Han aparecido inversores que han visto en la enseñanza a distancia una posibilidad de negocio al amparo de las nuevas tecnologías. En USA en 1999, el 63% de las 3.600 instituciones de educación superior acreditadas ofrecían cursos a distancia, y se sugería que en el año siguiente esa porción crecería hasta el 84%.

Y no sólo son las instituciones tradicionales las que están implicadas en este negocio, sino que inversores externos a la educación están promoviendo empresas de educación a distancia realizando inversiones muy importantes en las mismas. Estas empresas suelen partir de tres fuentes principales para construir sus cursos:

- Materiales realizados por personal propio.
- Materiales realizados ya en las Universidades que se licencian para su explotación por parte de las empresas con mención explícita del origen.
- Acuerdos directos en las empresas y los profesores universitarios o expertos para la producción de materiales nuevos, al modo de la publicación de libros de texto bajo encargo.

Este entorno produce que la batalla por el mercado de la educación sea ahora cada vez más evidente. Ya no se compite por los alumnos del entorno físico de la universidad, sino que se pretende llegar a todos los alumnos posibles. Además, existe la necesidad de la formación continua, de la hiperformación. Antes bastaba con tener una carrera superior para asegurarse un buen puesto de trabajo. Ahora, la competencia ha hecho que la posesión de un título no sea suficiente, haciendo crecer la demanda de formación de postgrado e incluso la necesidad de obtener más de un título superior.

1.3 Estrategias de las Universidades Tradicionales

Las estrategias que las universidades tradicionales siguen para adecuar el entorno educativo a las nuevas tecnologías pueden ser clasificadas en cuatro categorías, según su grado de innovación:

Incremental La tecnología se aplica como complemento a lo existente, como por ejemplo, colocando los apuntes en la página Web de la asignatura.

Educación a Distancia Haciendo la educación más apropiada a los estudiantes individuales. Realmente, esta estrategia pretende implicar a los estudiantes tradicionales de la universidad.

Alianzas La universidad permite bajo un acuerdo, que una empresa comercialice sus cursos a distancia de forma más agresiva que la que la universidad está dispuesta o sabe hacer. Esto permite acceder a mercados nuevos.

Entrar en el mercado La universidad se lanza a comercializar sus propios cursos en el mercado global sin la participación de un socio.

1.4 Una Experiencia

Para ver si el uso intensivo de las tecnologías mejoraban los resultados obtenidos por el método tradicional de enseñanza, el profesor Stephen Ruth realizó un experimento en 1997 con una asignatura sobre la utilización de las técnicas informáticas en una escuela de empresariales.

El resultado fue un mayor éxito en el aprendizaje, pero no sólo por la utilización intensiva de la tecnologías.

La metodología utilizada se resumen en los siguientes aspectos:

- Utilización de todas las tecnologías disponibles: WWW, Internet, CDROM, audiocasetes, vídeos, educación a distancia, sistemas de autoaprendizaje, programas, etc.

- Reducción de las horas de clase a un tercio (de 45 a 12). Para mantener el contacto se trabajó con correos electrónicos y tutorías (de 5 a 15 horas semanales) mejor conocimiento entre alumnos-profesor.

- Eliminación de los exámenes tradicionales a cambio de la realización de un cuaderno de la asignatura cuya revisión por el profesor se realizaba frecuentemente.

- Alumnos como descubridores, no como receptores de información. Proyectos de autodescubrimiento y ejercicios de laboratorio con sistemas de autoaprendizaje.

Capítulo 2

Portales Web Docentes

Normalmente los portales son sitios web en los que se ofrecen un conjunto de servicios añadidos como buscadores, e-mail, forums y compras en línea. Nuestro objetivo es el diseño y la construcción de un portal docente, así que los contenidos y servicios de nuestro portal se deben centrar en complementar la tarea docente.

Los contenidos de un portal docente pueden ser, entre otros, los siguientes:

- Información sobre el profesor: categoría, centro, despacho, teléfono, correo electrónico, horarios de tutorías.
- Enlaces al departamento y al centro del profesor.
- Actividades investigación: campos de investigación, currículum, etc.
- Por cada asignatura: datos de la misma (créditos, horario, etc.), objetivos, metodología, programa, apuntes, bibliografía, prácticas, lecturas recomendadas, etc.
- Otras informaciones: asociaciones, hobbies, fotos, etc.

Los servicios de un portal docente pueden ser, entre otros, los siguientes:

- Consultas off-line y FAQ (*Frequently Asked Questions*).
- Fórum de alumnos.
- Entrega de trabajos, prácticas, etc.
- Consulta de notas.
- Análisis del diario de visitas.

2.1 Cuestiones Básicas de Diseño Aplicadas

Existen multitud de documentos que describen las características que deben tener los portales Web para que resulten atractivos, útiles y consigan una alta tasa de conversión, que es el número de visitantes que se convierten en clientes. En nuestro caso, el objetivo no es vender nada, sino regalarlo. Así, casi todas las recomendaciones de diseño de sitios Web van a sernos útiles a nosotros también.

2.1.1 Lo que NO hay que hacer

Existen multitud de tentaciones en las que no hay que caer a la hora de diseñar nuestras páginas Web, y la lista que va a continuación es sólo una pequeña muestra de lo que no hay que hacer.

Fondos recargados Es mejor que el fondo de las páginas sea blanco o de un color claro y neutro que facilite la lectura de la misma. No es recomendable utilizar una imagen de fondo, pues solo va a distraer a los visitantes y quizás imposibilitar la lectura del texto.

Gráficos por doquier Sólo hay que utilizar elementos gráficos cuando aporten algo a la página, y en ese caso es mejor que sean sencillos y ocupen poco espacio. Esto hará que la página se cargue más rápidamente.

Parcelar con marcos Los marcos (*frames*) pueden ser útiles para mantener información siempre visible en la página, pero inducen a errores y dificultan la navegación. Es mejor utilizar tablas en su lugar, y si no queda más remedio, que no sean más de 3 marcos.

Modelo papiro Crear páginas lineales produce un derroche de espacio. Es mejor utilizar un modelo de página en dos columnas. Esto se puede conseguir con la utilización de tablas, que aunque no es la mejor solución, es la única sencilla.

Contadores de visitas El mero contador de visitas no aporta demasiada información sobre los visitantes. Es mejor que se utilicen herramientas más sofisticadas y menos agresivas para analizar el diario de visitas de las páginas.

En Construcción Es mejor que un enlace que conduce a una página en blanco con el letrero "en construcción" no existiera, o es que se imagina que el visitante va a volver a la semana a ver si por fin se ha construido la página. Además, la mayoría de las páginas en construcción siguen inacabadas por siempre.

2.1.2 Principios Básicos de Navegación

No importa lo bonito que haya quedado un sitio Web, o lo útil de la información que ofrece, si no tiene un esquema de navegación razonable, confundirá a sus visitantes y los echará del mismo.

La buena navegación es casi siempre un problema de sentido común, y existen unos principios básicos que hay que aplicar siempre. Excepciones son las páginas artísticas, que buscan un efecto en el visitante y por ello organizan la navegación de otro modo. Para la mayoría de los sitios las reglas básicas son las siguientes:

1. Decir a la gente exactamente lo que está disponible en el sitio.
2. Ayudar a ir a las partes deseadas de la forma más rápida posible.
3. Facilitar la solicitud de más información.

Estos principios se pueden extender un poco más en la siguiente relación de características:

La entrada principal La primera página es la primera página que el visitante ve, y debería ser suficientemente atractiva. Una pagina principal debería tener las siguientes características:

- Un resumen de lo que está disponible en el sitio, cada sección a la que se puede acceder desde la página principal ya sea directamente o a 2 o 3 clicks de distancia.
- Debe resultar atractiva y proyectar una imagen correcta de la compañía.
- Debe reforzar la imagen de marca o del producto, de modo que ésta sea fácilmente identificable.
- Debe compartir ciertos elementos con el resto de las páginas del sitio, de forma que se de una imagen corporativa del sitio.
- Una página principal se debe parecer mucho a la tabla de contenidos de un libro o de una revista.

Las páginas principales deben tener enlaces a:

- Sobre la empresa (departamento/centro/universidad)
- productos y servicios (asignaturas, consulta de notas)
- Cómo contactar con nosotros

Esquema Piramidal La mayoría de los sitios siguen una organización jerárquica con forma de triángulo, en el que el vértice está ocupado por la página principal, y las páginas en la base son las que tienen el nivel de detalle menor. Las jerarquías deben ser tales que los lados del triángulo sean similares. Esto significa que si una página apunta a otras 20 del siguiente nivel, quizás se necesita unas páginas en un nivel intermedio. De igual forma, si en cada nivel están sólo dos páginas, quizás convendría agruparlas en menos niveles. Normalmente, los sitios pequeños suelen tener de dos a tres niveles.

Longitud de las páginas No existe un criterio claro a la hora de establecer la longitud idónea de las páginas. Las que son demasiado largas para caber en la ventana del navegador obligan a utilizar el scroll para acceder a toda su información. En cambio, la división de un documento en varias páginas facilita su lectura pero puede ralentizar el acceso a la información pues hay que esperar a que se carguen varias páginas en lugar de una sola. Muchas veces la división de un documento largo en varias páginas atiende más a criterios crematísticos, ya que si las páginas contienen anuncios publicitarios, se van a enseñar más anuncios más veces que con una única página larga.

En cualquier caso, si estamos construyendo una página larga, hay que colocar enlaces a las distintas secciones al principio, y quizás también al final de la misma, para facilitar la navegación dentro de ella.

Barras de Navegación Una barra de navegación es un elemento que indica claramente la situación de la página actual dentro del sitio. Debe tener la misma apariencia en todas las páginas y debe estar siempre disponible.

`el sitio > docencia > asignaturas > la asignatura`

Cada uno de sus elementos será picable, de forma que además de indicar claramente dónde se está en cada momento se facilita la navegación.

Mapa del Sitio Especialmente si se trata de un sitio complejo, un mapa del mismo puede resultar una excelente herramienta para localizar una página concreta. Un mapa del sitio es una enumeración estrictamente jerárquica de todas las páginas del mismo. Todas los elementos del mapa deben ser picables para facilitar la navegación.

A estas recomendaciones de diseño básicas les podrían seguir muchas otras que tienen que ver, una vez más, con la moderación en el uso de elementos distorsionantes como los colores y los gráficos, el equilibrio entre funcionalidad y vistosidad como la coherencia y moderación en el uso de los tipos de letras, etc.

Capítulo 3

Herramientas de Creación Web

Existen en el mercado multitud de aplicaciones que facilitan la creación y el mantenimiento de los sitios Web. Las primeras páginas Web se realizaron a mano, pero la complejidad de los lenguajes y sistemas implicados en el desarrollo de los portales Web hacen que se haya abandonado la creación artesanal en favor de la más industrial. Lo malo que es al final se corre el riesgo de que los portales se parezcan demasiado unos a otros.

Antes de poder analizar y presentar las posibilidades de las herramientas de creación Web conviene repasar algunos conceptos técnicos sobre el Web y las tecnologías relacionadas.

3.1 Introducción a las Aplicaciones Web

La idea fundamental es que los navegadores, *browsers*, presentan documentos escritos en HTML que han obtenido de un servidor Web. Estos documentos HTML habitualmente presentan información de forma estática, sin más posibilidad de interacción con ellos.

El modo de crear los documentos HTML ha variado a lo largo de la corta vida de las tecnologías Web pasando desde las primeras páginas escritas en HTML almacenadas en un fichero en el servidor Web hasta aquellas que se generan al vuelo como respuesta a una acción del cliente y cuyo contenido varía según las circunstancias.

Además, el modo de generar páginas dinámicas ha evolucionado, desde la utilización del CGI ,*Common Gateway Interface*, hasta los *servlets* pasando por tecnologías tipo *JavaServer Pages*. Todas estas tecnologías se encuadran dentro de aquellas conocidas como *Server Side*, ya que se ejecutan en el servidor web.

Otro aspecto que completa (¿complica?) el panorama son las inclusiones del lado del cliente, *Client Side*, que se refieren a las posibilidades de que las páginas lleven incrustado código que se ejecuta en el cliente, como por ejemplo JavaScript y programas Java.

El esquema general de la situación se puede ver en la Figura 3.1, donde se muestran cada tipo de tecnología involucrada en la generación e interacción de documentos Web.

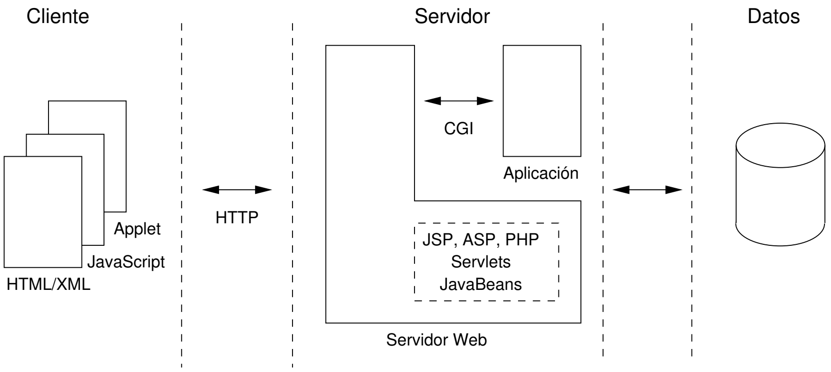

Figura 3.1: Esquema general de las tecnologías Web.

A continuación se va a intentar describir las principales características y funcionalidades de las diferentes tecnologías involucradas en el Web, aunque hay que ser consciente de lo vano del intento, ya que las posibilidades son grandes, la evolución en este tema es constante y el riesgo de quedarse obsoleto muy grande.

Antes de entrar en detalle, presentaremos las características del World Wide Web y del protocolo que lo sustenta, el HTTP.

3.2 World Wide Web, WWW

WWW es un sistema hipermedia interactivo desarrollado sobre Internet. Definir Internet no está dentro de los objetivos de este curso y lo único que simplificando se puede decir sobre ella es que funciona.

La idea de hipermedia es la de juntar texto, imágenes, audio y vídeo dentro de un mismo envoltorio llamado documento. WWW se asienta sobre el protocolo HTTP (Hyper Text Transfer Protocol) y sobre el lenguaje de definición de documentos hipermedia HTML (HyperText Markup Language).

El primer precedente del WWW se puede encontrar en un tratado escrito por *Vannevar Bush* titulado "*As We May Think*", de 1954, en el que básicamente animaba a los científicos a hacer más accesible a todo el mundo sus conocimientos y experiencia. En este tratado surgieron las nociones de hipertexto e hipermedia.

Desde entonces, muchos han sido los sistemas hipermedia que se han creado. Como ejemplo cabe destacar el *Augment* de Doug Englebart en los

70, y en los 90 el *WWW* de Tim Berners-Lee, TBL, creado en el Laboratorio Europeo de Partículas Físicas (CERN) en Ginebra, Suiza. TBL se basó en muchas fuentes para crear el WWW:

- TCP/IP,
- tipos MIME,
- SGML,
- arquitecturas Cliente-Servidor y Sistemas Abiertos.

Así, WWW es el sistema de difusión del conocimiento más importante que implementa estas nociones.

3.3 HyperText Transfer Protocol, HTTP

Aunque entender el modo en que funciona HTTP no es estrictamente necesario para desarrollar aplicaciones Web, algunas nociones sobre lo que esconden esas siglas puede ayudar a desarrollarlas con más facilidad y confianza.

HTTP es un protocolo del nivel de aplicación para sistemas de información multimedia distribuidos. Es un protocolo no orientado a estado que puede ser utilizado para más propósitos que para manejar ficheros HTML.

Entre las propiedades de HTTP se pueden destacar las siguientes:

- Un esquema de direccionamiento comprensible.
  Utiliza el *Universal Resource Identifier* (URI) para localizar sitios (URL) o nombres (URN) sobre los que hay que aplicar un método. La forma general de un URL es `servicio://host/fichero.ext` .

- Arquitectura Cliente–Servidor.
  HTTP se asienta en el paradigma solicitud/respuesta. La comunicación se asienta sobre TCP/IP. El puerto por defecto es el 80, pero se pueden utilizar otros.

- Es un protocolo sin conexión y sin estado.
  Después de que el servidor ha respondido la petición del cliente, se rompe la conexión entre ambos. Además no se guarda memoria del contexto de la conexión para siguientes conexiones.

- Está abierto a nuevos tipos de datos.
  HTTP utiliza tipos MIME (*Multipart Internet Mail Extension*) para la determinación del tipo de los datos que transporta. Cuando un servidor HTTP transmite información de vuelta a un cliente, incluye una cabecera que le indica al cliente sobre los tipos de datos que componen el documento. De la gestión de esos datos se encargan las utilidades que tenga el cliente (visor de imágenes, de vídeo, etc.)

Una transacción HTTP está compuesta por una cabecera, y opcionalmente, por una línea en blanco seguida de los datos. En la cabecera se especifica tanto la acción solicitada en el servidor, como los tipos de datos devueltos o un código de estado.

3.3.1 La Interacción entre el *Browser* y el Servidor

Durante una sesión normal de trabajo en WWW un cliente (navegador) solicita un documento de un servidor Web y una vez obtenido lo muestra al usuario que hizo la solicitud. Si este documento contiene un enlace a otro documento (en el mismo o en distinto servidor), y el usuario activa el enlace el cliente Web efectuará otra petición y mostrará el nuevo documento.

Durante la comunicación entre el cliente y el servidor HTTP en el que el cliente solicita el documento `doc1.html` al servidor se intercambian la siguiente transacción HTTP:

```
GET /doc1.html HTTP/1.0
Accept: www/source
Accept: text/html
Accept: image/gif
User-Agent:  Lynx/2.2  libwww/2.14
From: jvegas@infor.uva.es
/* esto es una linea en blanco */
```

El método `GET` indica el fichero que el cliente solicita y la versión de HTTP. El cliente también muestra una lista de los tipos MIME que puede aceptar como retorno, además de identificar el *browser* que utiliza (para que el servidor pueda optimizar los ficheros para el tipo particular de navegador) y su dirección de correo electrónico. Al final existe una línea en blanco que determina el final de la cabecera HTTP.

El servidor responde mandando la siguiente transacción HTTP:

```
HTTP/1.0  200 OK
Date: Friday, 23-Feb-01 16:30:00 GMT
Server: Apache/1.1.1
Content-type: text/html
Content-length: 230
/* esto es una linea en blanco */
<HTML><HEAD><TITLE> ........ </HTML>
```

En este mensaje el servidor utiliza la versión 1.0 de HTTP, y manda el código de estado 200 para indicar que la petición del cliente ha sido procesada satisfactoriamente. También se identifica como un servidor Apache. Indica al cliente que el contenido del documento es texto en formato HTML y que tiene una longitud de 230 bytes.

3.3.2 Métodos de Petición

La primera línea de una petición contiene los comandos HTTP, conocidos como métodos. Existen varios, pero los más conocidos y utilizados son tres: `GET`, `HEAD` y `POST`.

El método `GET` se utiliza para recuperar información identificada por un URI por parte de los navegadores. Si el URI se refiere a un proceso generador de datos como un programa CGI, en lugar de él, se devuelven los datos generados por el programa. El método `GET` también se puede utilizar para pasar una pequeña cantidad de información al servidor en forma de pares atributo–valor añadidos al final del URI detrás de un símbolo de interrogación, `?`.

```
GET /cgi/saludar.pl?nombre=pepe&email=pepe@infor.uva.es HTTP/1.0
```

La longitud de la petición GET está limitada por el espacio libre en los *buffers* de entrada. Por lo que para mandar una gran cantidad de información al servidor ha de utilizarse el método `POST`.

El método `HEAD` es idéntico al `GET` excepto que el servidor no devolverá el cuerpo del mensaje en la respuesta a un método `HEAD`. Esto es útil para obtener información sobre las entidades implicadas en la petición sin que tengan que transferirse. Sirve para comprobar si los enlaces son válidos o para saber cuando fue la última modificación de la entidad solicitada.

El método `POST` se refiere normalmente a la invocación de procesos que generan datos que serán devueltos como respuesta a la petición. Además se utiliza para aportar datos de entrada a esos programas. En este caso los pares atributo–valor son incluidos en el cuerpo de la petición separados por *ampersand*.

```
POST /cgi/saludar.pl HTTP/1.0
Accept: */*

nombre=pepe&email=pepe@infor.uva.es
```

De este modo el método `POST` no sufre de las limitaciones de espacio y puede enviar mucha más información al servidor.

3.3.3 Códigos de Respuesta del Servidor

El servidor HTTP responde con un código que informa sobre el estado de la transacción. Los códigos se agrupan según las siguientes categorías:

Rango	**Significado**
100-199	Informativo
200-299	Éxito en la resolución de la petición
300-399	Petición redirigida, necesarias más acciones
400-499	Petición incompleta
500-599	Errores en el servidor

3.4 Desarrollo de Aplicaciones Web

Con la introducción de Internet y del Web en concreto, se han abierto infinidad de posibilidades en cuanto al acceso a la información desde casi cualquier sitio. Esto representa un desafío a los desarrolladores de aplicaciones, ya que los avances en tecnología demandan cada vez aplicaciones más rápidas, ligeras y robustas que permitan utilizar el Web.

Afortunadamente, tenemos herramientas potentes para realizar esto, ya que han surgido nuevas tecnologías que permiten que el acceso a una base de datos desde el Web, por ejemplo, sea un mero trámite. El único problema es decidir entre el conjunto de posibilidades la correcta para cada situación.

El viejo CGI ha cumplido con el propósito de añadir interactividad a las páginas Web pero sus deficiencias en el desarrollo de aplicaciones y en la escalabilidad de las mismas ha conducido al desarrollo de APIs específicos de servidor como *Active Server Pages*, ASP, y PHP, que son más eficientes que su predecesor CGI.

Para aprovechar el potencial de estas tecnologías y ofertar una solución de servidor más extensible y portable, Sun ha desarrollado la tecnología llamada *servlet*. Los servlets Java son muy eficientes, debido al esquema de threads en el que se basan y al uso de una arquitectura estándar como la JVM, *Java Virtual Machine*.

Otra nueva tecnología viene a sumarse a las que extienden la funcionalidad de los servidores Web, llamada *JavaServer Pages*, JSP. Los JSP permiten juntar HTML, aplicaciones Java, y componentes como las *JavaBeans* creando una página Web especial que el servidor Web compila dinámicamente en un servlet la primera vez que es llamada.

3.4.1 Arquitectura Web

Ante tal aluvión de posibilidades, conviene repasar algunos aspectos básicos de la arquitectura Web.

Para abrir una página Web en un navegador, normalmente se teclea el correspondiente URL o se pica en el hiperenlace oportuno. Una vez que se solicita esta petición mediante el protocolo HTTP y la recibe el servidor Web, éste localiza la página Web en su sistema de ficheros y la envía de vuelta al navegador que la solicitó, según se muestra en la Figura 3.2.

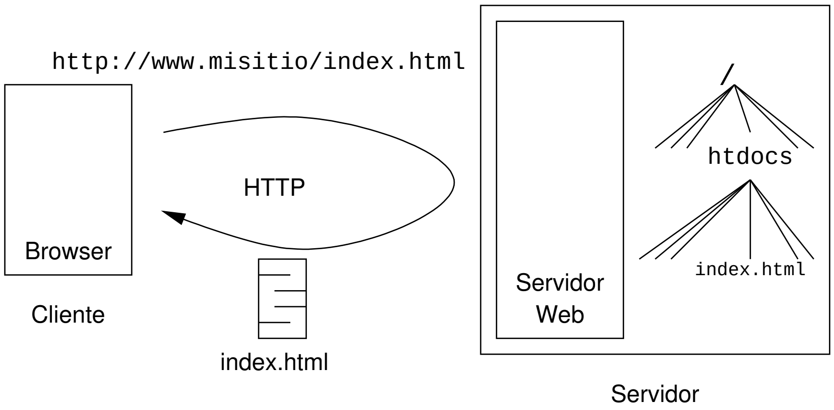

Figura 3.2: Arquitectura Web básica.

3.4.2 El Navegador Web, *Browser*

El navegador puede considerarse como una interfaz de usuario universal. Dentro de sus funciones están la petición de las páginas Web, la representación adecuada de sus contenidos y la gestión de los posibles errores que se puedan producir.

Para todo esto, los fabricantes de navegadores les han dotado de posibilidades de ejecución de programas de tipo *script*, con modelos de objetos que permiten manipular los contenidos de los documentos. Estos lenguajes de programación son VBScript, JScript (ambas de Microsoft) y JavaScript (de Netscape), y proporcionan las soluciones llamadas del lado del cliente, *client side* y permiten realizar validaciones de datos recogidos en las páginas antes de enviarlos al servidor y proporcionan un alto grado de interacción con el usuario dentro del documento.

Otras de las posibilidades de los navegadores es la gestión del llamado HTML dinámico (*Dinamic HTML*, DHTML). Éste está compuesto de HTML, hojas de estilo en cascada, (*Cascade Style Sheets*, CSS), modelo de objetos y *scripts* de programación que permiten formatear y posicionar correctamente los distintos elementos HTML de las páginas Web, permitiendo un mayor control sobre la visualización de las páginas.

En esta línea, los navegadores han ido un poco más allá y permiten la visualizaciones de documentos XML (*eXtensible Markup Language*) después de haber sido transformado adecuadamente a HTML por las hojas de estilo extensibles (*eXtensible Style Sheets*, XSL). De esta manera se puede elegir visualizar ciertos elementos y otros no, dependiendo de las circunstancias.

Además, los navegadores permiten la ejecución de aplicaciones dentro de los documentos mostrados. Las dos posibilidades más populares son la tecnología ActiveX y los *applets* Java. Los applets Java son pequeños programas que se descargan del servidor Web y se ejecutan en la JVM del navegador.

3.4.3 El Servidor Web

El servidor Web es un programa que corre sobre el servidor que escucha las peticiones HTTP que le llegan y las satisface. Dependiendo del tipo de la petición, el servidor Web buscará una página Web o bien ejecutará un programa en el servidor. De cualquier modo, siempre devolverá algún tipo de resultado HTML al cliente o navegador que realizó la petición.

El servidor Web va a ser fundamental en el desarrollo de las aplicaciones del lado del servidor, *server side applications*, que vayamos a construir, ya que se ejecutarán en él.

3.4.4 Aplicaciones Multinivel

Al hablar del desarrollo de aplicaciones Web resulta adecuado presentarlas dentro de las aplicaciones multinivel. Los sistemas típicos cliente/servidor pertenecen a la categoría de las aplicaciones de dos niveles. La aplicación reside en el cliente mientras que la base de datos se encuentra en el servidor. En este tipo de aplicaciones el peso del cálculo recae en el cliente, mientras que el servidor hace la parte menos pesada, y eso que los clientes suelen ser máquinas menos potentes que los servidores. Además, está el problema de la actualización y el mantenimiento de las aplicaciones, ya que las modificaciones a la misma han de ser trasladada a todos los clientes.

Para solucionar estos problemas se ha desarrollado el concepto de arquitecturas de tres niveles: interfaz de presentación, lógica de la aplicación y los datos.

La capa intermedia es el código que el usuario invoca para recuperar los datos deseados. La capa de presentación recibe los datos y los formatea para mostrarlos adecuadamente. Esta división entre la capa de presentación y la de la lógica permite una gran flexibilidad a la hora de construir aplicaciones, ya que se pueden tener múltiples interfaces sin cambiar la lógica de la aplicación.

La tercera capa consiste en los datos que gestiona la aplicación. Estos datos pueden ser cualquier fuente de información como una base de datos o documentos XML.

Convertir un sistema de tres niveles a otro multinivel es fácil ya que consiste en extender la capa intermedia permitiendo que convivan múltiples aplicaciones en lugar de una sola (véase la Figura 3.3).

La arquitectura de las aplicaciones Web suelen presentar un esquema de tres niveles (véase la Figura 3.4). El primer nivel consiste en la capa de presentación que incluye no sólo el navegador, sino también el servidor web que es el responsable de dar a los datos un formato adecuado. El segundo nivel está referido habitualmente a algún tipo de programa o *script*. Finalmente, el tercer nivel proporciona al segundo los datos necesarios para su ejecución.

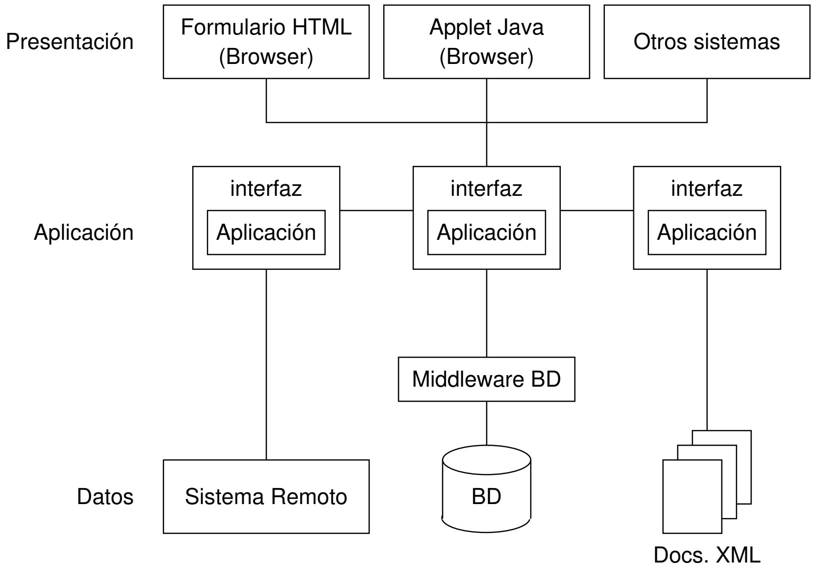

Figura 3.3: Arquitectura Multinivel.

Una aplicación Web típica recogerá datos del usuario (primer nivel), los enviará al servidor, que ejecutará un programa (segundo y tercer nivel) y cuyo resultado será formateado y presentado al usuario en el navegador (primer nivel otra vez).

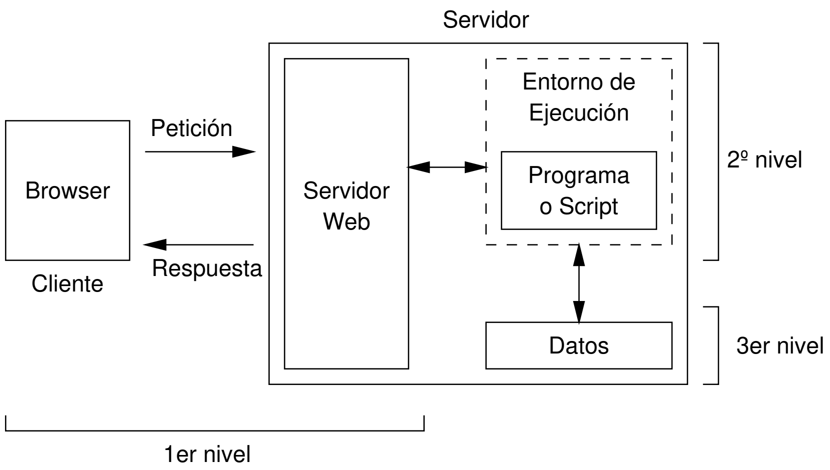

Figura 3.4: Arquitectura Web de tres niveles.

Lamentablemente, el uso de toda esta tecnología pasa por el dominio de técnicas de programación y de acceso a bases de datos, condición esta que no se puede presuponer en un curso de divulgación como éste. Así, nos vamos a restringir al uso de herramientas básicas a la hora de la construcción de un portal docente.

3.5 Ejemplos de Herramientas de Edición Web

Existen multitud de herramientas que permiten desarrollar portales Web con aspecto y prestaciones profesionales, aunque la mayoría de ellos requiere de una fase inicial de entrenamiento de duración variable según los casos.

Entre los más utilizados están:

- Macromedia DreamWeaver
- Microsoft FrontPage
- Netscape Composer

De los dos primeros hay que decir que integran en un entorno amigable casi todos los elementos relacionados con el desarrollo de páginas Web, además de aportar soluciones a la gestión y el mantenimiento de sitios Web grandes y complejos, y la interacción con bases de datos. El problema en su uso es que no resultan fáciles de manejar y requieren ciertos conocimientos y son de pago.

El tercer caso es el editor de páginas Web que proporciona el navegador Netscape. No es más que un pequeño editor de páginas sin más posibilidades, pero lo fácil de su uso hace que sea una buena herramienta para iniciarse en la creación y en el mantenimiento de portales Web. El precio a pagar es que algunas cosas se han de seguir realizando a mano, como la conexión con bases de datos y la gestión de formularios.

Capítulo 4

Un Ejemplo de Portal Web

Llegados a este punto se va a proponer el desarrollo de un portal Web docente con un mínimo de funcionalidades y realizado con la tecnología más sencilla posible. En esta línea de mínimos, se ha optado por utilizar el editor de páginas de Netscape como herramienta de apoyo.

En todo desarrollo de un sitio Web se han de seguir cuatro pasos fundamentales:

1. Establecer la información a mostrar en el Web, estructurando la estructura lógica del sitio.

2. Diseñar la apariencia física de las páginas.

3. Construir el sitio a partir de la información, la estructura lógica y de la apariencia física del mismo.

4. Mantener el sitio actualizado.

Vamos a seguir cada una de estas fases teniendo en cuenta todo lo comentado hasta ahora.

4.1 Información y Estructura Lógica

La información relevante en un sitio Web docente universitario está centrada en cuatro grandes temas:

- la presentación del profesor, con el departamento y los centros donde trabaja, además de los detalles necesarios para localizarle tanto física (despacho, dirección de correo postal, teléfono), como virtualmente (dirección de correo electrónico, página web, etc.);

- la faceta docente del mismo con los estudios donde imparte docencia, los horarios de tutorías, las asignaturas, los apuntes, los libros, etc.;

- la faceta investigadora con el currículo (las publicaciones, conferencias, comités, etc).
- otras informaciones, con aquellos aspectos más lúdicos o de menor importancia relacionados con el profesor (fotos, enlaces a sitios interesantes, ideas, proyectos por realizar, etc.).

Una vez que tenemos toda la información que queremos colocar en el sitio, conviene organizarla de forma lógica disponiéndola ya sobre los directorios y ficheros oportunos. Esto nos dará la oportunidad de disponer de un mapa del sitio.

$$\text{public\_html}\left\{\begin{array}{l} \text{index.html} \\ \text{docencia}\left\{\begin{array}{l}\text{index.html}\\ \text{x.html}\\ \text{asignaturaX}\left\{\begin{array}{l}\text{apuntes.html}\\ \text{practica.pdf}\end{array}\right.\end{array}\right. \\ \text{investigacion}\left\{\begin{array}{l}\text{index.html}\\ \text{cv.doc}\\ \text{libro.pdf}\end{array}\right. \\ \text{otros}\left\{\text{index.html}\right.\end{array}\right. \tag{4.1}$$

4.2 Diseño Físico de las Páginas

A la hora de diseñar la apariencia de las páginas resulta conveniente considerar las indicaciones respecto a colores, simplicidad de elementos y indicación de la situación dentro del sitio. Teniendo todo esto en cuenta se ha llegado a la plantilla mostrada en la Figura 4.1. En ella se pueden ver los distintos elementos considerados.

El código fuente de dicha plantilla es la siguiente:

```
<!doctype html public "-//w3c//dtd html 4.0 transitional//en">
<html>
<head>
   <title>página del profesor</title>
</head>

<body text="#000000" bgcolor="#FFFFFF" link="#0000EE"
      vlink="#551A8B" alink="#FF0000">

<!-- linea con el nombre del profesor y la foto -->
<table BORDER CELLSPACING=0 CELLPADDING=0 WIDTH="700"
```

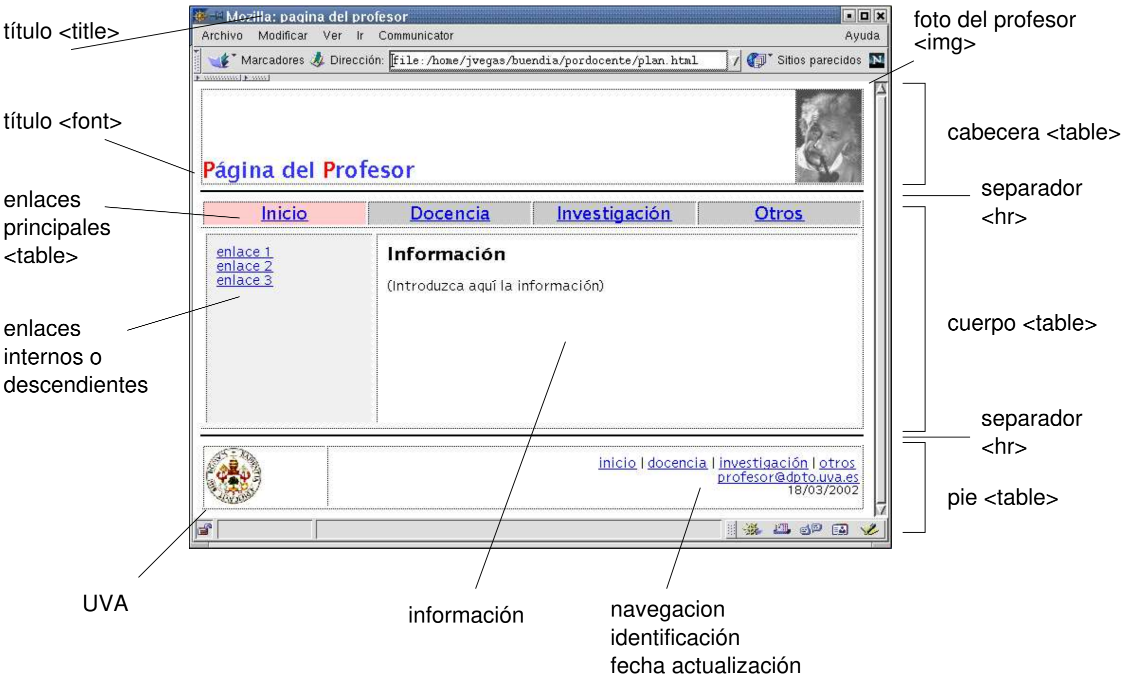

Figura 4.1: Plantilla de página Web.

```
       HEIGHT="20" NOSAVE >
<tr VALIGN=BOTTOM NOSAVE>
<td WIDTH="100%" NOSAVE>
   <font size=+4>
   <font color="#FF0000">P</font><font color="#3333FF">&aacute;gina del
   <font color="#FF0000">P</font><font color="#3333FF">rofesor</font>
   </font></td>
<td ALIGN=CENTER VALIGN=CENTER WIDTH="50%" NOSAVE>
   <img SRC="foto.gif" ALT="foto del profesor" height=98 width=69
        align=ABSCENTER></td>
</tr>
</table>

<!-- linea  separadora -->
<hr ALIGN=LEFT NOSHADE WIDTH="700">

<!-- tabla con el menu principal -->
<table BORDER COLS=4 WIDTH="700" NOSAVE >
<tr ALIGN=CENTER VALIGN=TOP BGCOLOR="#CCCCCC" NOSAVE>
<td BGCOLOR="#FFCCCC" NOSAVE>
   <center><font size=+2>
   <a href="index.html">Inicio</a></font></center></td>
```

```
<td>
   <center><font size=+2>
   <a href="docencia/index.html">Docencia</a></font></center></td>
<td>
   <center><font size=+2>
   <a href="investigacion/index.html">Investigaci&oacute;n</a>
   </font></center></td>
<td>
   <center><font size=+2>
   <a href="otros/index.html">Otros</a></font></center></td>
</tr>
</table>

<!-- tabla con el contenido de la página -->
<table BORDER CELLSPACING=5 CELLPADDING=10 WIDTH="700" NOSAVE >
<tr ALIGN=LEFT VALIGN=TOP NOSAVE>
<td WIDTH="25%" HEIGHT="200" BGCOLOR="#F0F0F0" NOSAVE>
   <!-- enlaces internos o descendentes -->
   <a href="#enlace1">enlace 1</a>
   <br><a href="enlace2.html">enlace 2</a>
   <br><a href="enlace3.html">enlace 3</a>
</td>
<td WIDTH="75%" NOSAVE>
   <!-- información -->
   <h2>Informaci&oacute;n</h2>
   (Introduzca aqu&iacute; la informaci&oacute;n)</td>
</tr>
</table>

<!-- linea  separadora -->
<hr ALIGN=LEFT NOSHADE WIDTH="700">

<!-- pie de página -->
<table BORDER WIDTH="700" NOSAVE >
<tr NOSAVE>
<td ALIGN=LEFT VALIGN=CENTER NOSAVE>
   <a href="http://www.uva.es">
   <img SRC="escudouva.gif" ALT="Universidad de Valladolid"
        BORDER=0 height=60 width=60></a></td>
<td ALIGN=RIGHT VALIGN=CENTER NOSAVE>
   <!-- enlaces principales -->
   <a href="index.html">inicio</a>
   | <a href="docencia/index.html">docencia</a>
   | <a href="investigacion/index.html">investigaci&oacute;n</a>
```

```
    | <a href="otros/index.html">otros</a>
    <br>
    <!-- enlaces secundarios -->
    <!-- firma -->
    <a href="mailto:profesor@dpto.uva.es">profesor@dpto.uva.es</a>
    <!-- fecha última modificación -->
    <br><font size=-1>18/03/2002</font></td>
</tr>
</table>

</body>
</html>
```

4.3 Construcción del Sitio Web

Para construir el sitio, una vez que tenemos estructurada la información y la plantilla de página, sólo hay que crear los directorios y los ficheros adecuados. Hay que poner mucho cuidado en que los enlaces estén correctamente construidos. También debemos modificar los colores y los títulos que nos indican la situación exacta dentro del sitio web.

4.4 Actualización

Quizás una de las fases más duras de la construcción de sitios web es el mantenimiento de los mismos. Resulta de vital importancia presentar información actualizada, pues de otro modo el sitio perderá interés, además de dar una imagen negativa de nosotros mismos.

Hay que recordar modificar la fecha de última actualización de las páginas actualizadas.

Capítulo 5

Referencias Biblográficas

Dada la naturaleza del tema central de este tutorial, parece lógico que este apartado de las referencias bibliográficas se alimente de referencias a documentos Web. No quiere decir esto que no haya buenos libros editados sobre el tema, sino que es más fácil y dinámico acceder a las referencias Web, además de que están actualizadas con más frecuencia y responden mejor a la evolución de la tecnología.

No hemos pretendido impresionar con un listado interminable de referencias sino mostrar las más interesantes y aquellas que se refieren a los sitios oficiales. Además, siempre se pueden conseguir miles de documentos interesantes utilizando alguno de los buscadores Web.

1. *The World Wide Web Consortium.*
   `http://www.w3c.org/`
   Es la página del consorcio WWW y en ella se puede encontrar información técnica sobre cualquier tema relacionado con WWW, ya sea arquitectura de sistemas, protocolos, herramientas, etc. Es básico y el lugar de partida obligatorio.

2. *CGI: The Common Gateway Interface.*
   `http://hoohoo.ncsa.uiuc.edu/cgi/`

3. *The W3Schools.*
   `http://www.w3schools.com`
   Esta página contiene cursos muy bien confeccionados sobre cualquier tema relacionado con el WWW. Imprescindible su visita ya que muchos de ellos son verdaderamente interactivos.

4. *Apache Proyect.*
   `http://www.apache.org`
   Es la página del servidor Web *Apache*. Es fácil de cargar, intalar, configurar y gratis.

5. *Browser Netscape.*
`http://home.netscape.com`
El navegador *Netscape* (gratis).

6. *Java Technology Home Page.*
`http://java.sun.com`
La página central del lenguaje Java.

7. *The Perl Institute.*
`http://www.perl.org`
La página del lenguaje Perl.

8. *Web Developers Journal.*
`http://www.webdevelopersjournal.com`
Una revista de desarrolladores de portales Web con gran cantidad de información técnica y sobre herramientas.

9. *Educause.*
`http://www.educase.edu`
Una iniciativa para el desarrollo de la educación superior a través del uso inteligente de las tecnologías de la información.

10. *The Internet and the University.*
`http://www.educause.edu/asp/doclib/abstract.asp?id=pub5005`
Este libro es el resultado de un fórum en el que se aportaron ideas sobre el modo en que Internet y las tecnologías relacionadas pueden mejorar el proceso de aprendizaje, evaluando los riesgos y las oportunidades de este cambio tecnológico.