

Administración de la web

Jordi Serra Ruiz
Miquel Colobran Huguet
Josep Maria Arqués Soldevila
Eduard Marco Galindo

PID_00190210



Los textos e imágenes publicados en esta obra están sujetos –excepto que se indique lo contrario– a una licencia de Reconocimiento-NoComercial-SinObraDerivada (BY-NC-ND) v.3.0 España de Creative Commons. Podéis copiarlos, distribuirlos y transmitirlos públicamente siempre que citéis el autor y la fuente (FUOC. Fundació para la Universitat Oberta de Catalunya), no hagáis de ellos un uso comercial y ni obra derivada. La licencia completa se puede consultar en <http://creativecommons.org/licenses/by-nc-nd/3.0/es/legalcode.es>

Índice

Introducción	5
Objetivos	6
1. Los servidores web y la organización	7
1.1. Servidor web y aplicaciones	7
1.2. Servidor web de páginas	8
1.3. Servidor web y servidor de aplicaciones	8
2. El administrador y el servidor	10
2.1. Estructura de un servidor web	10
2.1.1. Servidor/cortafuegos	11
2.1.2. Puertos	11
2.2. El administrador	11
2.3. Responsabilidades genéricas del administrador	13
2.4. El administrador y la creación de páginas	14
3. Recursos para crear páginas	16
3.1. HTML	16
3.2. XML/XHTML	17
3.3. CSS	17
3.4. Galletas (<i>cookies</i>)	18
3.5. Páginas dinámicas	18
3.6. DHTML de cliente	19
3.6.1. JavaScript y JScript	19
3.6.2. <i>Applets</i>	20
3.6.3. VBScript	21
3.7. DHTML de servidor	21
3.7.1. SSI	22
3.7.2. CGI	22
3.7.3. Perl	23
3.7.4. Java	23
3.7.5. <i>Servlets</i>	24
3.7.6. PHP	25
3.7.7. ASP	25
3.7.8. JSP	25
3.8. Aplicaciones web	26
3.8.1. CMS	26
3.8.2. Wikis	28
3.8.3. RSS	29
3.9. Aplicaciones de ejemplo	30

4. Seguridad	32
4.1. Seguridad de publicación	32
4.2. Seguridad del software	32
4.3. Protocolos de comunicación cifrados (HTTPS)	32
4.4. Registros de las conexiones	33
4.5. Copias de seguridad	33
4.6. Ataques al servidor	33
4.6.1. Falsificación de datos HTTP	34
4.6.2. Manipulación URL	35
4.6.3. Inyección de código	35
4.6.4. Inyección SQL	36
4.7. Métodos para acceder a bases de datos de la organización	36
4.7.1. Arquitectura a tres niveles	36
4.7.2. Restricción de accesos	37
5. Aspectos legales	39
5.1. Propiedad intelectual y sociedad de la información	39
6. Tareas/responsabilidades	41
Resumen	42
Actividades	43
Ejercicios de autoevaluación	43
Solucionario	45
Glosario	46
Bibliografía	48

Introducción

En este módulo veremos qué es un servidor web, los componentes que lo forman, los protocolos que puede tener y las maneras seguras de instalarlo, ya que puede contener información sensible de la organización.

También veremos su utilidad dentro de la intranet o de Internet y cómo se relacionan todos estos elementos entre sí para formar una parte activa de la organización.

La web puede ser nuestra imagen pública o una herramienta para crear aplicaciones intranet. También tiene el potencial de acceder a bases de datos. Veremos maneras seguras de hacerlo.

Actualmente una web puede ser la base para aplicaciones y usos muy diversos. Hay que tenerlo presente cuando se diseña el entorno web de la organización.

Finalmente, hablaremos de la figura más importante: el *webmaster* o administrador de la web. Posiblemente no se dedique a llenar la web de contenido, sino a controlarla. En este caso son los usuarios los que ponen el contenido, y tienen que conocer los elementos con los que se pueden construir las páginas para poder ser realmente creativos. En este caso se tiene que tener cuidado, puesto que la información publicada en la web de la empresa por los usuarios puede estar en contradicción con alguna política de privacidad de la empresa o no respetar alguna de las leyes de privacidad de datos. Por lo tanto, los contenidos siempre es mejor que tengan algún tipo de control antes de hacerlos públicos.

Objetivos

Los materiales de este módulo contienen las herramientas necesarias para que el estudiante alcance los objetivos siguientes:

- 1.** Conocer los componentes de un servidor web.
- 2.** Saber las posibilidades que puede ofrecer un servidor web, tanto si está en Internet como si forma parte de la intranet.
- 3.** Saber cómo se puede instalar un servidor web.
- 4.** Conocer las tecnologías y aplicaciones más importantes asociadas a los entornos web.
- 5.** Saber cómo se tienen que instruir los creadores de contenidos de un servidor web, a partir de los recursos y los elementos que pueden utilizar.
- 6.** Conocer las cuestiones de seguridad y legales asociadas a los servidores web.

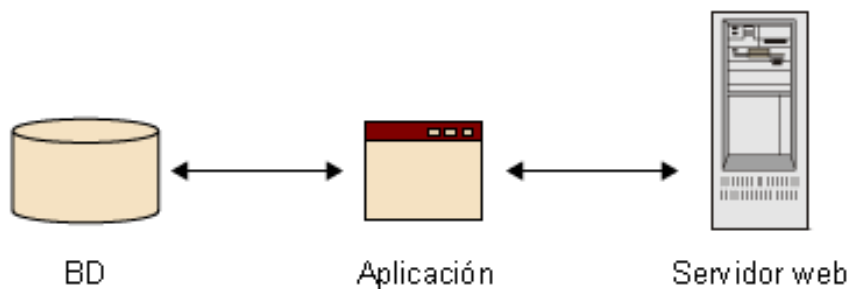
1. Los servidores web y la organización

Cuando se habla de administrar una web, se habla de ser responsable del funcionamiento de un servidor web. Ahora bien, un servidor web se puede ver desde dos vertientes: como interfaz de una aplicación o como servidor de páginas.

1.1. Servidor web y aplicaciones

Normalmente, cuando un servidor web se utiliza formando parte de una aplicación, simplemente hace de interfaz para la aplicación, de manera que es un elemento más. Esta aplicación utiliza un navegador como enlace para presentar las pantallas de salida y como mecanismo para pedir información al usuario. Los datos se guardan en una base de datos. Genéricamente, el esquema es el siguiente:

Servidor web como elemento de una aplicación



La distribución de estos elementos en servidores físicos depende de muchos factores como, por ejemplo, el tamaño y la complejidad de la aplicación, que seguramente estará dividida en módulos. Muy posiblemente, la estructuración estará condicionada por el servidor web que se utilice.

No podemos olvidar que aquí habrá seguridad porque esta aplicación no tiene que llegar necesariamente hasta Internet. Posiblemente, sólo interesará que funcione dentro de la intranet. La seguridad también nos marcará mucho la estructura de los servidores.

En esta situación, pues, el servidor web es un medio, aunque muy flexible, para hacer funcionar la aplicación, que es el centro principal de todo.

1.2. Servidor web de páginas

Si hace de servidor web con entidad como tal, lo puede hacer dentro de la intranet, en Internet o, incluso, una parte en la intranet y la otra en Internet.

La primera cuestión que hay que tener en cuenta es que exportamos la imagen de la organización al exterior y, por lo tanto, se tiene que cuidar la forma en que presentamos la web, cómo la mostramos y qué información introducimos en ella.

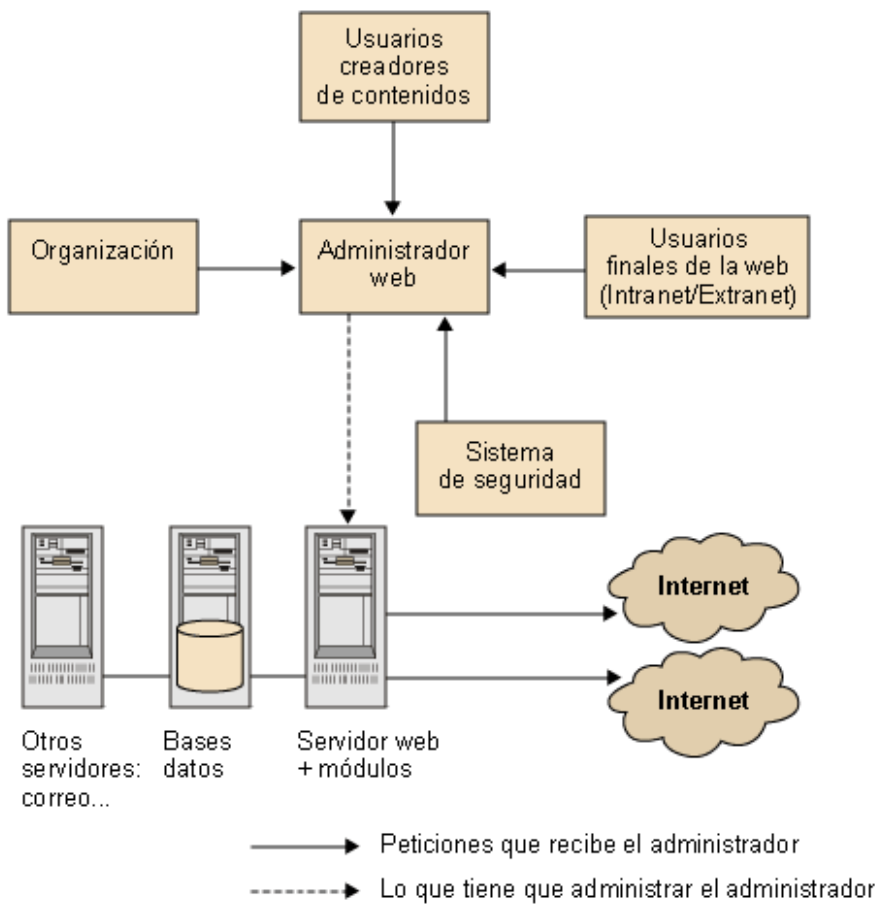
La web muestra la imagen corporativa de la organización a la comunidad Internet.

Terminología

Se usa el término intranet para referirse a una red usada internamente en una organización. El término intraweb, para una web que abarca los nodos de esta intranet. Pero a menudo se usa intranet como sinónimo de intraweb.

Cuando se utiliza una intranet y se publica información en Internet, a veces se utiliza el término *extranet*.

Esquema de la relación entre el administrador, el servidor web, la organización y los usuarios



1.3. Servidor web y servidor de aplicaciones

Inicialmente, el servicio web provenía sólo del servidor web, que proporcionaba todos los recursos necesarios. Actualmente, los sistemas han evolucionado y ya no podemos afirmar que todos los recursos necesarios para un servicio web vengan dados por el servidor web.

Con la llegada de los servidores de aplicaciones se inicia una auténtica revolución en el tratamiento de los servicios web. Aunque el punto de entrada y primer tratamiento de las peticiones se realiza en el servidor web, los sistemas complejos de generación de páginas que utilizan procesamiento y acceso a datos necesitan los recursos que ofrecen los servidores de aplicaciones.

Así pues, actualmente tenemos dos posibles vías de servicio según los niveles de acceso necesarios:

- **En dos niveles (servicios estáticos).** Servidos desde un servidor web. En el primer nivel situaríamos las estaciones de trabajo, ya sean locales o remotas, y en el segundo nivel el servidor web. Estaríamos hablando de una estructura cliente/servidor aplicada a un entorno web.
- **En tres niveles (servicios dinámicos).** Servidos desde un servidor web y un servidor de aplicaciones. En el primer nivel encontraremos las estaciones de trabajo y los usuarios (locales o remotos). En el segundo nivel encontraremos el servidor web (puerta de entrada) y el servidor de aplicaciones (proporciona capacidad de procesamiento y gestión de datos). Finalmente, en el tercer nivel encontraríamos el servidor de base de datos (datos centralizados del servicio web).

Se tiene que tener en cuenta pues que, aunque en general siempre hablaremos de servidor web en relación a los servicios web, puede ser que a causa de las necesidades de la organización el sistema se componga de servidor web y de servidor de aplicaciones (que no tienen por qué ser máquinas físicas independientes).

Ved también

Sobre la arquitectura a tres niveles, consultad el subapartado 4.7.1 de este mismo módulo.

2. El administrador y el servidor

El binomio administrador-servidor tiene que estar bien coordinado para obtener el mejor rendimiento posible.

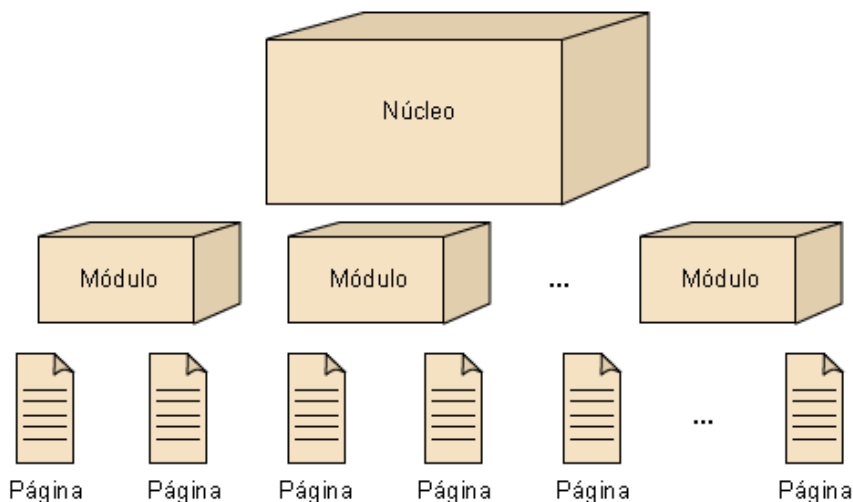
Conocer la arquitectura de software del servidor web que se utiliza es fundamental para el administrador. En general, todos los servidores trabajan de una manera parecida y podemos hacer un esquema genérico de cómo son. También son bastante generales las funciones del administrador, aunque es la cultura y la historia de la organización concreta quien marca las tareas y las necesidades reales que tiene que cubrir.

2.1. Estructura de un servidor web

Los servidores web se componen, básicamente, de los elementos siguientes:

- **Núcleo.** Es el servidor como tal, y es el centro del servidor web. Siempre está cargado y funcionando cuando el servidor está puesto en marcha.
- **Módulos.** Actualmente los servidores web pueden ser bastante complejos. Esto es posible porque se ofrecen módulos para aumentar la funcionalidad del servidor. Estos módulos, conocidos también como extensiones del servidor, se instalan cuando se configuran los servidores web. Entonces informamos de qué módulos se tienen que hacer funcionar. En general, y sin entrar en detalles de los módulos, un servidor puede hacer muchas más funciones que dedicarse exclusivamente a entregar páginas HTML que estén guardadas en el disco duro.
- **Páginas.** Las páginas son el lugar donde está guardada la información que presenta el servidor. Están estructuradas en forma de árbol de directorios, aunque en algunos casos hay módulos que modifican esta información.

Estructura de un servidor web



Un servidor web es, a grandes rasgos, el núcleo, los módulos y las páginas.

2.1.1. Servidor/cortafuegos

La arquitectura de conexión física más común cuando hay un servidor web que proporciona información es tenerlo detrás de un cortafuegos¹. El concepto es que el cortafuegos sólo deja pasar las peticiones de datos que el servidor tiene que proporcionar; simplemente se tiene que instalar el servidor de acuerdo con unas normativas de seguridad que la organización ya tiene que tener. Como el servidor web mantendrá una cierta interacción con el usuario, el hecho de estar protegido por un cortafuegos obligará a tener un control riguroso de qué puertas de acceso tiene que ofrecer y cuáles no.

⁽¹⁾En inglés, *firewall*.

Ved también

Ved el módulo “Administración de la red”, donde se habla más de los cortafuegos.

2.1.2. Puertos

El puerto estándar de petición de documentos u otros recursos web es el 80. En principio, un servidor web puede escuchar peticiones por cualquier puerto. Los servidores CAU², en cambio, pueden estar escuchando por el mismo puerto 80 o por el 8080, y en los dos casos se trata de una configuración estándar.

⁽²⁾En inglés, *proxy server*.

Servidor CAU

Un servidor CAU actúa como intermediario entre una red interna y una conexión externa en Internet. Así se puede compartir la misma conexión para recibir información desde servidores web. Se usa también para filtrar contenidos.

2.2. El administrador

El administrador es quien mantiene el servidor en funcionamiento, y también quien administra la seguridad y los permisos, y por lo tanto, quien tiene que conocer mejor que nadie todas sus posibilidades.

En concreto, la administración ha de:

- **Instalar y administrar el servidor web.** El servidor web es un recurso con el cual la organización da acceso a información que quiere publicar. Este servidor web, por ejemplo, podría dar servicio sólo a usuarios de una red local, o quizás de una intranet, o como en la mayoría de casos, dar una puerta de acceso libre para cualquier usuario conectado a la red Internet. Dependiendo del servicio al que esté destinado este servidor web, las configuraciones y políticas de diseño variarán.
Un servidor web no plantea muchos problemas a la hora de instalarlo, ya que como concepto sólo se trata de indicar dónde está la información y cómo tiene que aceptar las conexiones de los usuarios que quieran acceder a él. Además, el servidor ya proporciona herramientas de control que permiten monitorizar y, posteriormente, analizar el rendimiento. Por ejemplo, es muy común que un servidor web tenga disponible la cantidad de accesos simultáneos que tiene y cuál es su rendimiento.
- **Conocer el software.** Hay dos clases de software que el administrador de la web tiene que utilizar. Por un lado, el que se utiliza para configurar el servidor, y por el otro, el que proporciona las herramientas de preparación, modificación y organización de las diferentes páginas que forman la web que hay que servir. Un tema muy importante, con respecto al software relacionado con un servidor web, es lo que funcionará en la máquina cliente, ya que no todos los navegadores interpretan el código de la misma manera. Así pues, es importante tomar conciencia de los clientes que visitarán la web ofrecida por nuestro servidor. En caso de desarrollar una intranet, quizás podemos ser más libres, dado que es muy probable que haya un navegador estándar para toda la organización.
- **Administrar los directorios.** Cuando se tiene que diseñar una nueva web, también se tiene que tener en cuenta el modo en que se organizan los diferentes recursos. Lo más habitual, y que el mismo software de desarrollo de las páginas ya gestiona, es tenerlos clasificados en carpetas o directorios según el tipo. Es habitual que exista un directorio de imágenes al cual todas las páginas hagan referencia. Eso permite compartir, por ejemplo, ficheros de logos de la organización, ya que no se tienen que incluir en todas las ubicaciones donde haya documentos que los requieran. La estructura en directorios también permite, si es necesario, una migración mucho menos costosa.
- **Administrar las páginas.** Normalmente, las páginas quedan fuera del alcance del administrador de la web, ya que, como veremos más adelante, hay una figura que acostumbra a dedicarse a generarlas. De todos modos, es importante ser consciente de la importancia de llegar con “pocos pasos” a donde sea necesario. Así pues, con la experiencia de organizar información, ficheros, directorios, y este conocimiento abstracto de cómo es debido moverse por dentro de una máquina, un administrador tiene que dar

apoyo al creador de páginas y aconsejarle cómo disponer los enlaces para acceder de una manera más rápida a toda la web desde cualquier sitio.

El administrador instala y cuida de la web, no del contenido.

2.3. Responsabilidades genéricas del administrador

Un servidor web proporciona una puerta de acceso a una gran cantidad de información de una manera muy cómoda para el cliente y, por lo tanto, se trata de un servidor que, aunque durante la instalación no presenta muchos problemas, sí que se tiene que plantear muy bien cómo se administra cada una de las partes que lo componen y todos los temas relacionados con la web, ya que el hecho de proporcionar información es una responsabilidad que recae en manos del administrador.

Podemos destacar las siguientes responsabilidades del administrador:

- **Responsabilidad del servidor.** Un servidor web no es excesivamente complejo de administrar. Lo más habitual es que el mismo servidor registre los datos que se necesita evaluar. Es aconsejable tenerlo (como el resto de servidores de la organización) bajo una alimentación eléctrica ininterrumpida y que disponga de una conexión con unos mínimos de caudal garantizados para que un incremento de tráfico en la red local no afecte a los clientes remotos que consulten la web. Habitualmente, los servidores web se configuran con dos líneas alternativas de conexión a Internet (ADSL y fibra óptica, por ejemplo) y se les proporciona un mecanismo que permite un cambio instantáneo en la conexión a Internet de copia de seguridad cuando se detecte algún problema, y una vez se haya arreglado vuelva a servir la web por el canal habitual.
- **Responsabilidad de los contenidos.** Es importante darse cuenta de los dos aspectos del contenido en un servicio web:
 - Tiene que haber un control de qué datos se proporcionan en cada instante, de manera que no haya contenidos anticuados, repetidos, sensibles o que no sean coherentes entre ellos.
 - En relación con el acceso a estos datos, tiene que haber un control de cómo se gestiona, consulta y modifica la base de datos a la cual puede estar accediendo el servidor web.
- **Responsabilidad de la homogeneidad.** Un aspecto más de diseño que de administración, pero que recae muy a menudo en la administración de la web, y que es muy importante, es el control de la imagen que refleja la web. Como se trata de una puerta abierta a muchos usuarios potenciales, que pueden no ser conocidos, se tiene que tener mucho cuidado al presentar los recursos. Así, es importante tener en cuenta que en todas

las páginas tiene que haber un estilo constante; por ejemplo, no podemos encontrarnos con una página con el fondo azul y la siguiente con el fondo negro, y, si vamos a otro apartado de la web, ver que todas las páginas que lo forman tienen un fondo común con el logo de la organización.

- **Generación de estadísticas.** Según las necesidades de cada organización, esta tarea puede ser más o menos importante. En general hay que tener un registro de los que han visitado la web; tanto de lo que han visto como de cuándo y desde dónde lo han hecho. Aparte, hay otros tipos de estadísticas muy interesantes como, por ejemplo, a qué velocidad se ofrecen los documentos o la cantidad de peticiones que hay sobre las diferentes páginas, y el tiempo total que un cliente se pasa navegando por toda la web.

El administrador no tiene que hacer necesariamente las páginas, pero se preocupa por todas, y también del diseño global de la web.

2.4. El administrador y la creación de páginas

Actualmente, el papel del administrador web³ está dividido. Recordad que la estructura de una web está formada por el núcleo, los módulos y las páginas. El administrador de la web se encarga de que la web funcione bien, de que los aspectos de seguridad sean los que necesita el usuario y la organización, de que se sigan las directrices que marca la organización y de que el contenido global sea homogéneo.

Sin embargo, puede que no se encargue de cuidar de los contenidos. La tendencia es que las páginas las mantenga directamente la misma fuente de información. Eso se hace dando plantillas y formando a las personas que se tengan que encargar de ellas. Por lo tanto, el administrador de la web tiene dos clases de usuarios:

- **Consumidores del material de la web.** Los consumidores del material de la web son los usuarios finales, los que se conectan por medio de un navegador y consultan la web para saber alguna cosa.
- **Creadores del material de la web.** Ya hemos dicho que el administrador de la web supervisa los contenidos, pero no los fabrica. Puede aconsejar sobre cómo hacerlos, dar plantillas, dar herramientas, explicar las posibilidades de la web y facilitar los mecanismos del servidor web para que los creadores del material de la web puedan hacer la página de la mejor manera posible.

⁽³⁾Administrador web en inglés se expresa como *webmaster*.

Ved también

Recordad que en el subapartado 2.1 tenéis un esquema en el que se muestra la estructura de un servidor web.

En sus inicios, el creador de páginas web era una persona con un perfil básicamente artístico. No tenía conocimientos técnicos, pero sí que sabía bien cómo combinar elementos para crear efectos vistosos. Nosotros le teníamos que proporcionar los conocimientos de los elementos que integran las webs: HTML, Java, JavaScript, etc.

Podríamos considerar que el administrador web tendría que seguir pautas como las que se detallan a continuación:

- Como hay unos propietarios que tienen la responsabilidad de la información que el servidor web está sirviendo, se les tiene que involucrar en la manera en que esta información se presenta. Así, es importante que el administrador se encargue de orientarlos e introducirlos en el mundo del HTML, el FTP, que les enseñe a utilizar escáneres, a convertir formatos de ficheros, a buscar por Internet o a utilizar las propiedades del navegador.
- Para ahorrar tiempo, ayuda mucho disponer de plantillas de diseño (habitualmente HTML) que se puedan proporcionar en caso de que algún creador pida información.
- Hay que tener una política de uso del servidor web. Así, como hay un grupo que ha aceptado tal política, no puede haber malentendidos sobre contenidos.
- Verificar los enlaces. Los documentos se mueven constantemente de un lugar a otro y eso representa un problema muy serio, ya que es muy importante que un servidor web no ofrezca enlaces “caducados”.

El administrador es quien promueve la utilización de los recursos de la web entre los creadores de páginas. Les proporciona todos los recursos y conocimientos necesarios para que puedan hacer páginas web y ponerlas en el servidor.

3. Recursos para crear páginas

El mantenimiento de las páginas cada vez migra más hacia la misma fuente de información. Esto permite que se actualice al momento, no haya tantos errores, se simplifique el mantenimiento y se distribuya la carga de trabajo.

Se tiene que formar a los creadores de páginas en el uso de estas tecnologías, ya que posiblemente, a medida que las utilicen, encontrarán más posibilidades e irán siendo más exigentes.

Como muchas veces este creador es el usuario final, se le tiene que formar desde la perspectiva de las posibilidades que le ofrece el servidor. Puede hacer páginas muy simples (una página HTML) o muy complejas (programación JSP⁴). El creador tiene que saber qué herramientas y qué recursos hay para decidir si los necesita o no para su necesidad en concreto. Es el administrador de la web quien, de la manera más simple y didáctica posible, orienta al usuario final en las posibilidades del servidor web, para que las utilice como más le convenga. Hacemos, pues, un repaso de todas las características de creación de las páginas web.

⁽⁴⁾JSP es la abreviatura de *Java Server Pages*.

3.1. HTML

El HTML⁵ es el corazón de todas las páginas web. Es el “lenguaje” con que están hechas todas las páginas y es el lenguaje que interpretan todos los navegadores.

⁽⁵⁾HTML es la sigla de *hypertext markup language*, en castellano, lenguaje de marcaje de hipertexto.

El HTML es un lenguaje basado en marcas. Siempre van entre '<' y '>' y no es importante si están escritas en minúsculas o en mayúsculas. Las marcas van en parejas de la forma: <Marca> </marca> y todo lo que engloban se escribe en medio.

El HTML contiene el texto que se quiere presentar al usuario, junto con información sobre el formato como, por ejemplo, el tamaño de la letra, el color, la información sobre si hay fotografías, las imágenes de las fotografías, la información sobre tablas, márgenes, etc. El HTML es un lenguaje muy completo y actualmente vamos camino de la versión 5.

HTML 5, la nueva versión

Actualmente, estamos usando la versión 4.01 del HTML, que fue aprobada el 24 de diciembre de 1999 por el World Wide Web Consortium (W3C), el consorcio que marca el estándar de HTML, entre muchas otras cosas (la especificación se puede consultar en línea).

La nueva versión del lenguaje de Internet (en desarrollo por el W3C), el HTML 5, incorpora muchos cambios como, por ejemplo, las etiquetas *artículo*, *dialog*, *figure*, *footer*, *header*, *section*. Se espera tener una revisión estable hacia el 2012.

En este formato se ponen los famosos y conocidos enlaces⁶, que permiten ir de un punto a otro de la misma página, a otra página dentro del mismo servidor o a otra página de otro servidor cualquiera. La marca para hacerlo es `Text`.

Ver una página

Es posible ver el código fuente de una página web. Por ejemplo, con el navegador Firefox está la opción "Visualiza>Código fuente de la página" (o la combinación CONTROL + U), que abre una ventana donde se ve toda la página que muestra el navegador con el formato HTML.

Conocer el HTML es básico para todo administrador de web.

3.2. XML/XHTML

En virtud de la existencia del SGML⁷, y en un intento de tener un lenguaje de marcas único, el World Wide Web Consortium (W3C) define el XML⁸. Es un estándar ajustado a las necesidades de Internet, ya que el SGML es demasiado complejo de utilizar. Como el HTML 4.01 no es XML compatible, el W3C ofreció un estándar HTML compatible con XML, el XHTML.

3.3. CSS

Todo lo que hemos hecho hasta ahora nos permite fabricar páginas, pero nos tenemos que preocupar de dar un estilo uniforme a cada una de ellas. ¿Y si queremos, por ejemplo, que todas las páginas tengan la misma fuente de letras y los mismos colores?

Hasta ahora lo teníamos que hacer de una en una. Con la aparición del CSS⁹ ya no es necesario. Es un fichero de estilo en que se incluye toda la información relevante del estilo de la página, y todas las páginas lo seguirán.

La separación de la presentación del contenido en los documentos simplifica la creación de páginas y el mantenimiento de la web.

⁽⁶⁾ *Enlace* es el término castellano. También se utiliza mucho el término original inglés *link* o *hiperlink*.

Ejemplo de enlace

Para ir a la página de la UOC pondríamos ` Conectaros a la UOC `.

⁽⁷⁾ SGML es la sigla de *standard generalized markup language*, en castellano, lenguaje de marcaje generalizado estándar.

⁽⁸⁾ XML es la sigla de *xtensible markup language*, en castellano, lenguaje de etiquetado extensible.

⁽⁹⁾ CSS es la sigla de *cascading style sheets*, estándar propuesto en 1996 por el W3C.

Como era un gran problema para los diseñadores de páginas, la idea se aceptó rápidamente y, en 1998, apareció la versión 2, el CSS2.

3.4. Galletas (*cookies*)

Si queremos hacer una web dinámica, tenemos la posibilidad de utilizar galletas¹⁰, que son pequeños ficheros que algunas webs guardan en el ordenador del usuario. De esta manera, podemos saber las páginas que visita el cliente con el fin de ofrecerle, desde el servidor, páginas a medida según las que haya visitado.

3.5. Páginas dinámicas

Todo lo que hemos hecho hasta ahora nos permite fabricar páginas, pero éstas son “estáticas”. Residen en el servidor y se entregan a través de la red al cliente a petición de éste. El cliente siempre muestra los elementos en el mismo orden y posición. Si queremos cambiar algo (orden, posición y/o elementos), tenemos que reescribir el código HTML. Así pues, si quisiéramos hacer un buscador o un gestor de correo estaríamos muy limitados con HTML.

A raíz de estas limitaciones, apareció lo que se conoce como DHTML¹¹, que es el conjunto de técnicas para crear páginas dinámicas. DHTML es muy amplio e incluye muchas técnicas y muchos lenguajes de programación. Por lo tanto, el DHTML realmente no es una nueva especificación del HTML. De hecho, es una manera de ver y controlar los códigos y mandos del HTML estándar.

A diferencia del HTML estático, una página de DHTML contiene elementos (como por ejemplo *scripts*) en el lado del cliente que cambian el HTML del documento. Eso afecta a la apariencia y funciones de los objetos de la página. De esta manera, obtenemos más control sobre la página. Eso es posible porque los navegadores actuales incluyen estas capacidades.

Los componentes de una página de HTML dinámico son:

- **Modelo en objetos para la representación de documentos (DOM¹²).** Es un modelo orientado a objetos que los programas y *scripts* pueden usar para acceder y modificar dinámicamente el contenido, estructura y estilo de los documentos HTML y el XML.
- **Scripts.** Son pequeños programas (escritos habitualmente en JavaScript) incrustados en el HTML que permiten la interactividad del código y el control de los objetos especificados en el DOM.

CSS versión 3

La versión 3 de CSS parece accidentada. La versión 2, aparece el 12 mayo de 1998. Con los problemas de adopción por parte de los navegadores, ha pasado por diversos estadios de revisión, y el 19 de julio de 2007 aparece la CSS 2.1. La versión 3 se empezó a desarrollar en 1998, y diez años más tarde todavía no está terminada.

⁽¹⁰⁾En inglés, *cookies*.

⁽¹¹⁾DHTML es la sigla de *dynamic HTML*, en castellano, HTML dinámico.

⁽¹²⁾DOM es la sigla de *document object model*.

Dirección recomendada

El modelo en objetos para la representación de documentos ha sido desarrollado por el W3C y se puede encontrar su especificación en <http://www.w3.org/DOM>.

- **Páginas de estilo (CSS).** Se usan en DHTML para controlar la apariencia de los documentos. Las páginas de estilo definen los colores, fuentes de texto, color de fondo, imágenes y el posicionamiento de objetos.
- **HTML 4.x.** Usado para crear la página y los elementos que usarán las páginas de estilo y el modelo DOM.

Hay dos clases diferenciadas de DHTML, el DHTML de cliente y el DHTML de servidor.

3.6. DHTML de cliente

El DHTML de cliente incluye el conjunto de técnicas para la construcción de páginas dinámicas que se desarrollan dentro del ámbito de una página web, cuando la página se está viendo en la pantalla del usuario, es decir, en el navegador. En esta situación, para hacer cualquier efecto o interactividad, el recurso es el navegador. Por eso se habla de *cliente*.

Este tipo de programación se utiliza para dotar de efectos, sonido, vídeo, menús interactivos y respuesta a las acciones del usuario en las páginas. Permite liberar al servidor de tareas que puede resolver localmente el ordenador cliente (como la validación de parte de los datos de un formulario, por ejemplo).

3.6.1. JavaScript y JScript

Hasta ahora, las páginas eran estáticas. El usuario las veía tal como las hacíamos y no se modificaba nada en ningún momento. A partir de este momento, intentamos hacer páginas más dinámicas, con más posibilidades.

Queremos hacer algún cálculo, una página interactiva, o quizás una página que haga cosas dependiendo de la posición del ratón en la pantalla del usuario. Sea el caso que sea, una posible solución es utilizar JavaScript. Programación dentro de la página, que se ejecuta en el cliente.

JavaScript es un lenguaje para escribir *scripts* (guiones) que fue desarrollado por Netscape. Permite crear páginas interactivas de una manera muy sencilla. JavaScript no es Java. Lo que se necesita para ejecutar JavaScript es simplemente un navegador con la opción de ejecutar JavaScript activada.

El JavaScript se introduce directamente en el código HTML, entre las marcas `<script language="JavaScript">` y `</script>`.

Uno de los puntos más importantes en la programación en JavaScript es la captura de acontecimientos. Así, se puede crear una página que tenga en cuenta los movimientos y cada vez que se haga clic con el ratón. Aparte, también se pueden definir funciones dentro de la página y, por ejemplo, crear botones que al presionarse hacen que se ejecute esta función.

Finalmente, tened presente que JScript es la versión de Microsoft para su navegador de Javascript, con algunas incompatibilidades.

Pero se tiene que ir con mucho cuidado a la hora de habilitar y usar el Javascript, ya que, por el hecho de poder ejecutar desde el servidor programas en las máquinas clientes, los hacen muy interesantes para ejecutar código malicioso, puesto que sólo hay que cargar una página Web con un Javascript que el cliente se bajará y ejecutará en su ordenador sin que le pida permiso. Hay técnicas en las que se pueden conseguir las *cookies*.

3.6.2. *Applets*

Un *applet*¹³ es una aplicación que se ejecuta en el contexto de otro programa, típicamente un navegador web. El *applet* se tiene que ejecutar en un entorno llamado contenedor a través de un conector¹⁴. A diferencia de un programa, un *applet* no puede ejecutarse de manera independiente y tiene unos privilegios de seguridad muy restringidos.

⁽¹³⁾ *Applet* en castellano se expresa como "miniaplicación".

⁽¹⁴⁾ En inglés, *plug-in*.

La principal ventaja de utilizar *applets* reside en que son mucho menos dependientes del navegador que los *scripts* y son independientes del sistema operativo del ordenador en el cual se ejecutan. Por contra, sus principales desventajas en relación con los *scripts* es que son más lentos de procesar y no tienen acceso a todos los demás componentes de la página.

Los ***applets* de Java** son una manera de incluir código que se ejecuta en los clientes que visualizan una página web. Se trata de pequeños programas hechos en Java, que se transfieren junto con las páginas web y que el navegador ejecuta automáticamente en el espacio de la página. Los *applets* de Java están programados en Java y precompilados. Por este motivo, funcionan de modo diferente que los lenguajes de *script* como Javascript. Los *applets* son más difíciles de programar y requieren unos conocimientos medios del lenguaje Java.

Las restricciones de seguridad (el entorno de seguridad) donde funcionan son muy fuertes, de manera que el programa Java no puede hacer todo lo que quiera. De hecho, no puede abrir conexiones en otros ordenadores que no sean el *host* del cual proviene ni puede borrar los ficheros del disco duro, por

ejemplo. Todo el Java está orientado y pensado para funcionar por Internet de manera que, aunque se fabriquen programas maliciosos, no puedan operar abiertamente en los ordenadores objetivo.

Al ser Java más potente que Javascript, el número de aplicaciones potenciales de los *applets* puede ser mayor.

Los *applets* Java se copian del servidor al cliente y se ejecutan sobre el ordenador cliente, con fuertes restricciones de seguridad.

Ejemplos de *applets*

Ejemplos de *applets* son los *Java applets*, las animaciones Flash y el Windows Media Player usado para mostrar vídeo en los navegadores.

3.6.3. VBScript

VBScript o Visual Basic Scripting es similar al lenguaje Visual Basic, pero tiene básicamente la misma estructura que JavaScript.

VBScript es el lenguaje de *scripting* desarrollado por Microsoft para páginas web y su navegador. VBScript no tiene relación con Javascript. Son dos lenguajes que compiten pero que pueden hacer las mismas funciones.

3.7. DHTML de servidor

Cuando un cliente solicita una página, el servidor ejecuta los *scripts* y genera una página resultado, que es enviada al cliente. La página resultado sólo contiene código HTML, de manera que puede ser interpretada por cualquier navegador sin errores, independientemente de su versión. Existe una serie de lenguajes basados en servidor para ejecutar sus *scripts*. Genéricamente se llaman *embedded scripting language* (ESL).

Sólo la independencia del navegador ya es una ventaja que justifica la existencia de esta programación, pero ahora contamos con todos los recursos del servidor donde están alojadas las páginas. Estos recursos, como por ejemplo gestores de bases de datos, servidores de correo o el sistema de archivos del servidor, nos permiten construir todo tipo de aplicaciones.

Hay que destacar que el código de las páginas nunca llega al cliente. El navegador sólo recibe HTML, y eso implica que los visitantes de una página nunca pueden acceder al núcleo de las aplicaciones que se hayan desarrollado, es decir, a los *scripts* del lado del servidor.

3.7.1. SSI

El Server Side Includes (SSI¹⁵) no se puede considerar un lenguaje. Es un conjunto de funciones del servidor que permite modificar el documento pedido antes de ser enviado al navegador del cliente. La finalidad básica es insertar contenido como la salida de pedidos del sistema.

⁽¹⁵⁾SSI es la sigla de la expresión inglesa *server side includes*, que se puede traducir por 'inclusiones en el lado del servidor'.

El funcionamiento básico de las SSI es:

- 1) El cliente hace una petición de una página por medio del navegador.
- 2) El servidor recibe la petición y ejecuta las *server side includes* que se indican en el fichero, y como resultado de eso obtiene un fichero HTML que guarda en memoria.
- 3) Envía al navegador del usuario este fichero HTML para que lo muestre.

Las inclusiones al lado del servidor están escritas como comentarios HTML de la forma `<! funcio attribut="argument" >`.

3.7.2. CGI

CGI¹⁶ es de las primeras extensiones que aparecieron. La idea originaria se basaba en hacer funcionar un programa en el servidor que fuera capaz de hacer cualquier cosa y generar unos resultados que fueran una página HTML. Fue el primer "truco" para acceder a bases de datos y para crear páginas que no estuvieran estáticas en el servidor (páginas *on line* o páginas en línea).

⁽¹⁶⁾CGI es la sigla de *common gateway interface*.

Hay que destacar que, con CGI, el servidor ejecuta un programa arbitrario, hecho con cualquier lenguaje de programación. El método, a grandes rasgos, es el siguiente:

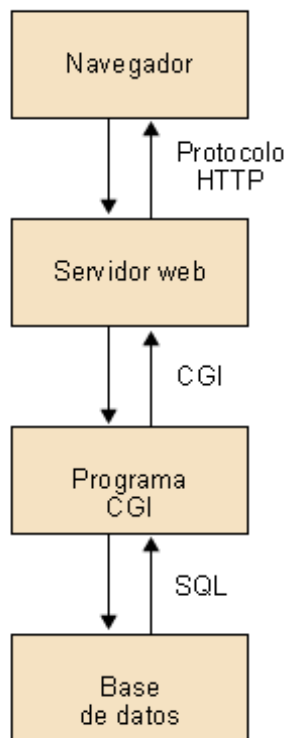
- 1) El cliente informa de que hace falta que se ejecute la aplicación.
- 2) El servidor pone en marcha el programa CGI que le ha pedido el cliente.
- 3) Este programa se ejecuta (puede consultar una base de datos, por ejemplo) y saca el resultado por la salida estándar (*standard output*) ya preparado y formateado en formato HTML. De hecho, es una página HTML.
- 4) El servidor captura esta salida y la envía al cliente.

Los inconvenientes de los CGI son:

- Carga mucho el servidor, ya que se ejecuta siempre.

- En cada petición se tiene que cargar y ejecutar el programa y, por lo tanto, es muy costoso.

Como hay tecnologías más actualizadas, ha caído bastante en desuso.



CGI es un recurso utilizado ampliamente, ya que fue el primero en ejecutar código arbitrario en el servidor y en enviar el resultado al cliente.

3.7.3. Perl

El CGI¹⁷ es un enlace para ejecutar programas arbitrarios en un servidor web. El Perl es uno de los lenguajes usados habitualmente para programar los CGI, aunque se pueden usar otros lenguajes como C, C++ o VisualBasic. Sin embargo, el Perl se utiliza en muchos otros sitios, y no sólo para CGI.

Perl viene de *practical extraction and report language* y tiene unas funciones de manipulación de texto muy potentes. Con este lenguaje se pueden definir todo tipo de algoritmos para tratar ficheros en el sistema y generar las salidas que se quieren.

3.7.4. Java

Java apareció como un lenguaje de propósito general pero muy orientado a Internet. No está pensado para ninguna plataforma concreta, sino para una plataforma inexistente, una plataforma “virtual” y, por lo tanto, funciona en

⁽¹⁷⁾Recordad que CGI es la sigla de *common gateway interface*.

Origen de Perl

Perl fue creado por Larry Wall. Quería simplificar las tareas de administración del sistema operativo Unix. Ahora es un lenguaje de propósito general.

Dirección recomendada

Java es un lenguaje creado por Sun. Encontraréis más información en la siguiente dirección de Sun: <http://java.sun.com>.

cualquier plataforma sobre la máquina virtual correspondiente, sin necesidad de recompilar el código generado por el compilador. Esta independencia de plataforma es lo que lo ha hecho uno de los lenguajes preferidos por Internet.

Existen diferentes ediciones de la plataforma Java con funcionalidades diferentes. Encontramos *Java 2 Standard Edition* (J2SE), que es el entorno de desarrollo de aplicaciones Java orientado a las aplicaciones individuales y los *applets*. *Java 2 Enterprise Edition* (J2EE) es la plataforma Java para crear aplicaciones web. Y la edición *Java 2 Micro Edition* (J2ME), que es una versión reducida de la plataforma Java 2 para usar en pequeños dispositivos como teléfonos móviles o agendas electrónicas.

De hecho, Java se usa casi para cualquier cosa que se pueda hacer en cualquier lenguaje (muchas veces con grandes ventajas). Podemos programar páginas web dinámicas, con accesos a bases de datos. La programación dinámica con Java se ha extendido tanto que no es difícil encontrar *servlets* en el lado del servidor y *applets* de Java en el lado del navegador.

3.7.5. *Servlets*

Los *servlets* son la respuesta de la tecnología Java a la programación CGI. Son programas que se ejecutan en un servidor web y construyen una página web. Así, se puede construir una página con datos que el usuario acaba de introducir, o quizás con el resultado de los motores de búsqueda. Si tenemos que tener páginas con información de bases de datos, se puede utilizar JDBC¹⁸ para acceder a ella y después sacar por la salida estándar el código HTML que el servidor web capturará y enviará al navegador del cliente. Una ventaja importante con respecto a los CGI es que cada petición se arranca como un hilo de ejecución ligero de la máquina virtual, mientras que con CGI se arranca cada vez un proceso, y eso puede comportar una sobrecarga del sistema.

Los *servlets*, comparados con los CGI, tienen muchas ventajas. Pueden arreglar y sincronizar muchas peticiones concurrentes (se podría hacer un sistema de conferencia *on line*, por ejemplo). La primera generación de *servlets* era pesada y requería bastantes conocimientos. La solución pasó por JSP, que permite incluir código Java dentro de las páginas HTML.

Ved también

Recordad que hemos tratado los *applets* en el subapartado 3.6.2 de este mismo módulo.

⁽¹⁸⁾JDBC es la sigla de *Java Database Connectivity*. Permite interactuar con una base de datos mediante un programa Java.

3.7.6. PHP

El PHP¹⁹ es una herramienta que permite crear páginas web dinámicas. Son páginas que se tratan como si fueran HTML. Por defecto, la extensión es “php” y hace falta que el servidor lo soporte. En este caso, el servidor las interpretará antes de enviarlas al cliente. No se tiene que compilar nada ni instalar herramientas adicionales. La marca especial para el PHP es `<?php instrucciones?>`.

Una de las características más potentes del PHP es la manera de tratar los formularios HTML, ya que un elemento en un formulario se convierte directamente en una variable con el mismo nombre, con el valor que el usuario ha puesto en el formulario.

El lenguaje es como una mezcla de lo mejor de C, Java y Perl, con características propias de PHP. El resultado está orientado a crear páginas web dinámicas rápidamente.

3.7.7. ASP

ASP²⁰ es una DLL que extiende el Microsoft Internet Information Server (servidor web de Microsoft). Genéricamente, los pasos son los siguientes:

- 1) El cliente hace una petición de una página ASP por medio del navegador.
- 2) El servidor lee el fichero ASP y lo ejecuta, con lo cual obtiene un fichero HTML que guarda en memoria.
- 3) Envía al navegador del usuario este fichero HTML para que lo muestre.

3.7.8. JSP

JSP²¹ es una tecnología que nos permite mezclar HTML estático con HTML generado dinámicamente, e integrar código Java dentro de la página JSP. Aparte, tenemos la ventaja con respecto a ASP²² que se escribe en un lenguaje que no es propiedad de Microsoft y que se puede llevar a otros sistemas operativos y servidores web.

Acceder a otros programas por medio del servidor

Con todo lo que hemos visto, parece que una de las cosas más importantes es el acceso a otros programas por medio del servidor. ¿Por qué?

Pues, sencillamente, porque la mayoría de veces la petición de información que haga un cliente navegador a nuestro servidor implicará una serie de acciones que hacen falta llevar a cabo con el fin de preparar la información correspondiente. En la práctica, la información no acostumbra a ser estática, sino que se tiene que ir a buscar a bases de datos y, por lo tanto, el resultado tiene que ser diferente según el momento en que se haga la consulta. Por ejemplo, si un cliente navegador hace preguntas por medio de una página web que proporciona información sobre entradas de conciertos, es evidente que se tiene que hacer una consulta para obtener el resultado, y después presentarlo al cliente.

⁽¹⁹⁾PHP es la sigla de *hypertext pre-processor*, en castellano, preprocesador de hipertexto.

Dirección recomendada

La versión actual de PHP es gratuita y está bajo licencia GPL. Se puede encontrar en <http://www.php.net>.

⁽²⁰⁾ASP es la sigla de *active server page*, en castellano, servidor de páginas activo.

ASP.net

La última versión sacada por Microsoft, la llamada *plataforma.net* incorpora muchos cambios que la hacen incompatible. Así, ya no puede usar VBScript, sino VB.net. ASP.net está orientado a objetos.

⁽²¹⁾JSP son las siglas de *Java server pages*; en castellano, servidor de páginas Java.

⁽²²⁾ASP son las siglas de *active server page*.

3.8. Aplicaciones web

El avance en las tecnologías web ha provocado que se utilicen los servidores web y las bases de datos como tecnologías de base para ofrecer servicios y resolver las nuevas necesidades que ha provocado la situación. Así, nos encontramos con nuevas aplicaciones que utilizan tecnología web no como fin, sino como medio para proporcionar un nuevo servicio a la organización y al usuario.

3.8.1. CMS

Un sistema de gestión de contenidos (CMS²³) es un programa que permite crear una estructura de apoyo para la creación y administración de contenidos por parte de los participantes. Es una aplicación de base de datos que automatiza el proceso de administrar y mantener la publicación de contenido. Un buen sistema de administración de contenido separa el contenido de la presentación.

⁽²³⁾CMS es la sigla de *content management system*.

La división entre contenido y presentación lleva diferentes ventajas a lo largo del ciclo de vida de la publicación y más allá de éste. El diseño puede ser concebido, extendido y de hecho relanzado sin la necesidad de extraer o reinsertar el contenido existente.

Un sistema de gestión de contenidos consiste en una interfaz que controla una o diversas bases de datos donde se aloja el contenido (textos, imágenes, etc.) que veremos en el sitio web. El sistema permite manejar de manera independiente el contenido y el diseño. Así, es posible conservar el contenido y cambiar en cualquier momento el diseño sin tener que modificar el formato de todo el contenido. Además, permite la publicación fácil y controlada en el sitio web. Un ejemplo clásico es el de editores que cargan el contenido en el sistema y otro de nivel superior que permite que estos contenidos sean visibles para todo el público.

Un CMS tiene dos funciones principales: facilitar la creación de contenidos y presentar estos contenidos. Con relación a la primera, provee una serie de herramientas para que publicar contenido sea tan fácil como rellenar un formulario, y exista, además, una sola fuente. Con respecto a la segunda, facilita la publicación de contenidos en múltiples formatos a partir de una sola fuente, y añade metadatos a los mismos para facilitar la navegación.

Un CMS provee las herramientas necesarias para gestionar el ciclo de vida de los contenidos: creación, gestión, presentación y mantenimiento y actualización.

Las ventajas de los CMS son las siguientes:

- **Mantenimiento descentralizado.** La actualización del contenido se puede llevar a cabo desde cualquier navegador. Eso permite hacer las tareas desde cualquier estación de trabajo, en cualquier momento y por varios editores simultáneamente.
- **Autores de contenido con pocos conocimientos técnicos.** Los autores sólo necesitan un conocimiento estándar de aplicaciones de procesador de texto para crear contenido. No es necesario ningún conocimiento de HTML o herramientas adicionales.
- **Restricciones de acceso configurables.** Se asignan roles y permisos a los diferentes usuarios. De esta manera, se protege el contenido de ser modificado sólo por los elementos autorizados.
- **Consistencia del diseño.** Al separar el contenido del diseño, este último se consigue preservar mucho mejor. El contenido de todos los autores se presenta en un estilo predefinido, conservando la unidad, identidad y consistencia de la información.
- **Vínculos, menús y navegación generados automáticamente.** Los menús son generados automáticamente, basados en el contenido de la base de datos. Los vínculos también, y se evitan los vínculos erróneos o que no enlazan a ningún lugar.
- **Contenido almacenado en una base de datos relacional.** El almacenaje central e independiente de la presentación se traduce en la posibilidad de usar el mismo contenido de diferentes maneras (diferentes webs, por ejemplo) y en diferentes medios (WAP, PDA).
- **Contenido dinámico.** Las extensiones como, por ejemplo, foros, encuestas, aplicaciones de compra, busca o noticias, pueden añadirse de manera modular. Incluso el CMS permite la creación de extensiones a medida que se pueden añadir de manera modular.
- **Actualizaciones diarias.** No es necesario personal para las modificaciones rutinarias.
- **Coordinación de tareas.** Simplifica las actualizaciones, registra los cambios y actividad hecha por cada autor (*log*) y proporciona herramientas de colaboración y trabajo en grupo entre autores.
- **Programación del contenido según calendario.** El contenido puede tener una vigencia para ser publicado o requerir contraseña. Mientras tanto, puede estar escondido para ser visto sólo por los editores.

3.8.2. Wikis

Las herramientas para la creación y mantenimiento de *wikis* son un tipo de software colaborativo. Una *wiki*²⁴ es una web que permite leer, editar y crear páginas que al final son visualizadas en formato HTML. Las *wikis* son usadas como apoyo para la colaboración, documentación, recopilación de informaciones, etc. Las páginas se almacenan en una base de datos y se incluyen enlaces generados dinámicamente. Para el usuario de la *wiki* representa la creación de páginas web de forma rápida y eficaz sin tener que tener muchos conocimientos de informática ni programación; pueden incluir de forma muy intuitiva texto, enlaces, documentos digitales y otros elementos multimedia.

La tecnología *wiki* permite que las páginas *wiki* sean escritas de forma colaborativa a través de un navegador web, conservando un historial de cambios que permite recuperar fácilmente cualquier estado anterior de la página. Es decir, que están diseñadas con la filosofía de que sea fácil corregir los errores, en vez de que sea difícil cometerlos.

En las *wikis* se puede consultar (desde el registro de cambios) los “Cambios Recientes”, donde encontramos una lista con los cambios hechos durante un periodo de tiempo. El “Historial de Revisión” muestra versiones anteriores de una página, y se pueden ver los cambios entre dos revisiones. A través del “Historial”, un editor puede ver y decidir restaurar una versión anterior del artículo.

Las ventajas de las *wikis* son las siguientes:

- Libertad para el usuario. Provoca mucha más participación en su uso, a diferencia de los sistemas tradicionales.
- Los contenidos se actualizan rápidamente (a tiempo real).
- El documento final acostumbra a ser producto de la comunidad de usuarios.
- Hay un registro y control de cambios.
- Las páginas permiten un mecanismo de comentarios, de manera que se puede llegar a acuerdos sin presencia en tiempo y espacio.
- Las herramientas de edición y actualización son sencillas y no requieren conocimientos muy técnicos.

⁽²⁴⁾ El término *wiki wiki* es de origen hawaiano y significa *rápido*.

Wikipedia

Wikipedia (2001) es una *wiki*. Contiene más de un millón de artículos en inglés. Es una enciclopedia que permite a los usuarios consultar y modificar sus contenidos.

El primer software para *wikis* fue creado por Ward Cunningham hacia 1995. En la actualidad existe una gran variedad de herramientas para *wikis*, muchas de ellas en licencia GPL.

3.8.3. RSS

Las webs que nos interesan pero que cambian muy a menudo (por ejemplo, sitios de noticias o weblogs) las tenemos que visitar regularmente. Puede llegar a ser un problema. De la misma manera, si nuestra web genera contenidos de forma continuada forzará a los usuarios a visitar muy a menudo algunas páginas.

La solución reside en crear unas páginas que contienen la información de una página web con las últimas actualizaciones que se han hecho. Este tipo de archivos son los RSS²⁵ y se han convertido en muy útiles para cualquier web.

El usuario (consumidor) de contenidos RSS sólo tiene que recoger información de diversas fuentes utilizando unos programas, llamados *agregadores* o lectores de RSS. El *agregador*, con una interfaz única, permite el acceso inmediato a las diversas fuentes de información, evitando visitar los diversos sitios web. Los *agregadores* comprueban, bajo demanda o periódicamente, si ha habido cambios en el contenido de las fuentes RSS a que están suscritos, y actúan en consecuencia, mostrando un aviso visual o sonoro al usuario, por ejemplo.

El fichero RSS es una especificación de formato de archivo, basada en XML 1.0, utilizada en Internet para difusión de noticias y titulares. Es muy habitual en sitios web de prensa, blogs y de contenidos técnicos. Al igual que el XML, es independiente de la plataforma y se utiliza en las arquitecturas y sistemas operativos más populares.

Esta forma de distribución de información tiene diversas ventajas sobre la suscripción clásica en boletines de noticias mediante correo electrónico:

- El usuario no tiene que facilitar una dirección electrónica, y no se le llenará ningún buzón de correo en caso de que no lo consulte.
- La disponibilidad de contenidos es inmediata. Está disponible en RSS, en formato web y no hay que esperar a mensajes en ninguna lista de correo.
- No hay problemas de disponibilidad en servidores de correo o filtros antispam que identifiquen erróneamente mensajes de listas de correo.

Puede ser muy útil para una empresa tener un gestor de RSS en su página web, donde va insertando noticias y actas relacionados con los productos que ofrece, puesto que tiene más proyección externa de la web y de sus productos.



Icono para indicar que una web tiene contenido RSS

⁽²⁵⁾RSS es la sigla de *Rich Site Summary* (RSS 0.91), *RDF Site Summary* (RSS 0.9 y 1.0) o *Really Simple Syndication* (RSS 2.0)

Dirección recomendada

La especificación completa de RSS 2.0 se puede encontrar en: <http://cyber.law.harvard.edu/rss>.

Blogs

No entraremos en detalles, pero los blogs se han vuelto tan habituales como el correo electrónico. En esencia, un blog (del inglés *web log*) o bitácora es una web que recopila cronológicamente artículos de uno o de diversos autores, como un diario personal. Habitualmente, los lectores pueden escribir comentarios y el autor, responder. Existen blogs de carácter general y temático.

3.9. Aplicaciones de ejemplo

Veamos algunas aplicaciones de uso de un servidor web.

1) Acceder a información de la organización por medio de la web

La organización puede aprovechar el servidor web para proporcionar una puerta de acceso a toda la información que sus integrantes necesiten. Con una política de usuarios y grupos, se puede montar un servidor de documentación que permita acceder a toda la información de la organización de una manera ordenada y jerarquizada, si es necesario. Por ejemplo, se podría utilizar el servidor web de manera que cuando alguien necesite rellenar un formulario de petición de vacaciones sólo tenga que acceder a la web e imprimirlo.

Se puede ir más allá, e incluso montar un sistema de compartimentación de información y actos de la organización. Si se agrupa por proyectos, o por áreas, se puede tener toda la información centralizada y hacerla accesible para todo el mundo.

2) Creación de aplicaciones

Uno de los problemas con los cuales se encuentran las organizaciones es que cada vez que se tiene que desarrollar una nueva aplicación, hay que formar a los usuarios para que sepan interactuar. La solución a este problema sería simplemente utilizar siempre la misma interfaz entre el usuario y la aplicación, de manera que el usuario ya estuviera acostumbrado a moverse dentro de la aplicación. De una manera natural, surgió la idea de aprovechar el navegador que, es independiente de plataforma, como herramienta entre el usuario y la aplicación, y por lo tanto, se basó siempre en un servidor que acaba sirviendo páginas HTML. Así pues, como los usuarios ya saben cómo escoger entre diversas opciones, saben cómo añadir datos, modificar, consultar, etc. Desarrollar una aplicación con formato web es una opción muy atractiva con vistas a la eficacia de interacción con el usuario final.

Ved también

Ved el módulo “El sistema informático dentro de la organización”, que habla de la creación de aplicaciones desde el punto de vista del responsable de informática.

3) Correo web (*webmail*)

El correo web permite a un usuario consultar el correo electrónico sin tener que estar siempre en la misma ubicación. A la mayoría de usuarios eso les representa un incremento importante de la comodidad, ya que utilizar programas específicos de correo implica, por una parte, tenerlos instalados en el ordenador y, por lo tanto, depender de ellos para mirar el correo, y por otra, todo un proceso de aprendizaje para poder utilizar las funcionalidades que proporciona.

Se hace evidente pensar en el hecho de que un medio de acceso común a todas las aplicaciones facilita mucho las cosas y, por lo tanto, cada vez surgen más aplicaciones que se basan en un cliente (navegador) y un servidor (servidor web).

4. Seguridad

La seguridad es un aspecto muy complejo, y más cuando se trata de una aplicación orientada específicamente hacia el exterior. Aquí, la seguridad se puede ver desde muchas vertientes.

4.1. Seguridad de publicación

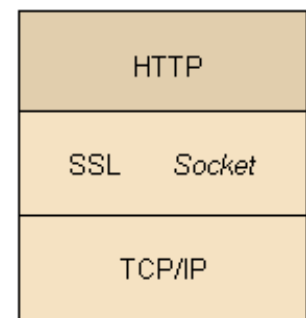
Como la web publica información de la organización, se tiene que vigilar de no publicar por error información que sea confidencial. Esto puede pasar, por ejemplo, cuando el servidor accede a una base de datos: por un problema de permisos relajados, quizás se puede acceder a más información de la necesaria.

4.2. Seguridad del software

Todos los servidores web son aplicaciones que escuchan peticiones por unos puertos determinados y dan la información cogiéndola de los servidores y proporcionándola a los clientes. Eso los hace interesantes y potencialmente vulnerables para ser atacados por piratas informáticos, que constantemente buscan las debilidades para utilizarlas como puerta para poder acceder a los sistemas informáticos.

4.3. Protocolos de comunicación cifrados (HTTPS)

Para comunicarse con el servidor web que contiene información sensible es recomendable utilizar protocolos que utilicen cifrado. El más utilizado es el HTTPS (HTTP seguro), que utiliza cifrado en la comunicación, a diferencia del protocolo HTTP, en que la información circula de modo visible, por lo cual con un *sniffer* (detector) se puede ver toda la sesión. Por lo tanto, en caso de una aplicación de correo electrónico, transacciones electrónicas, información personal o información privada de la organización, es muy recomendable instalar el módulo de HTTPS.



Esquema del estándar HTTPS

HTTPS es una versión cifrada de HTTP, implementada por medio del estándar SSL/TLS⁽²⁶⁾. La conexión cifrada se lleva a cabo simplemente con una conexión HTTP ordinaria sobre una conexión SSL cifrada. Para el resto, HTTPS y HTTP son iguales. El puerto estándar de conexión HTTPS es el 443.

⁽²⁶⁾SSL es la sigla de *secure socket layer*, y TLS, de *transport layer security*.

4.4. Registros de las conexiones

Los servidores hacen un log de las conexiones, pero, además, con poco trabajo añadido, es posible llevar un registro de qué máquinas son las que se han conectado, o incluso verificar en un servidor de nombres de dominio (DNS²⁷) el nombre de la máquina. Si no está en ningún DNS, una opción es no permitir la entrada, y si está en el DNS, guarda la dirección IP, el nombre IP y la dirección MAC de la máquina.

⁽²⁷⁾DNS es la sigla de *domain name server*.

4.5. Copias de seguridad

El servidor web, sin embargo, tiene pocos ficheros de configuración; vista su importancia dentro de la organización, se hace indispensable disponer de un sistema de copias de seguridad que proporcione todo lo necesario en caso de un fallo. Lo primero que se tendría que tener guardado es todo el proceso para instalarlo y cómo está organizado internamente. Aparte, es importante tomar nota de todos los cambios que se van haciendo y todas las políticas de registros que se han establecido. Se acostumbra a tener toda esta documentación por escrito y aparte, y a hacer diaria o semanalmente una copia de seguridad íntegra de todos los componentes que forman el servidor web.

4.6. Ataques al servidor

Un servidor está expuesto a gran variedad de ataques, por lo que tiene que ser un elemento importante dentro del esquema de seguridad del sistema informático. Un ataque (con éxito) a un servidor puede tener consecuencias que hay que tener presentes:

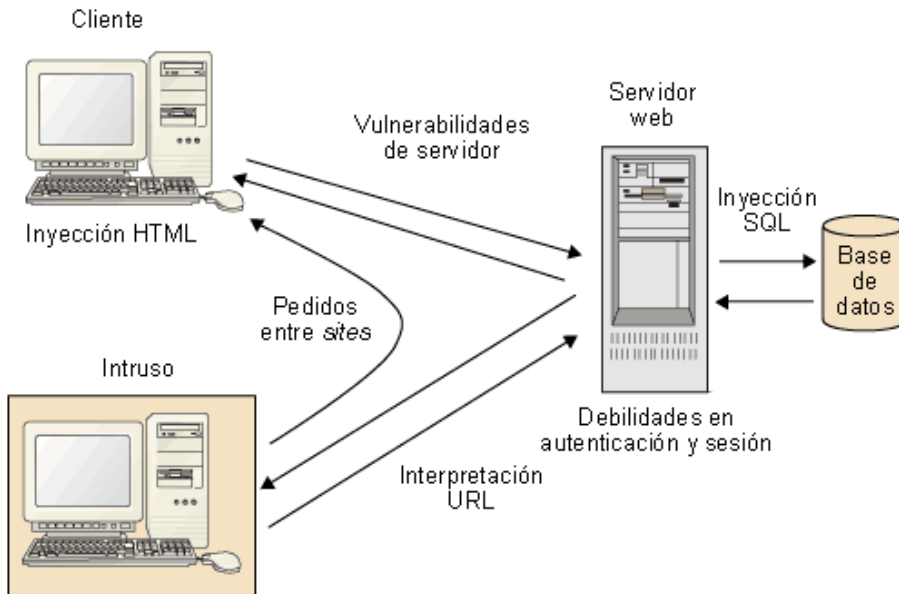
- Degradación de la imagen de la organización.
- Funcionamiento incorrecto o destrucción de parte o de toda la web.
- Robo de información.
- Modificación de datos, y en especial de datos personales y/o sensibles de los usuarios y de la organización.
- Intrusión en el servidor web.

Al haber muchos ataques diferentes, se pueden clasificar de la siguiente manera:

- Vulnerabilidad del servidor web. Ya no es habitual. Los servidores web cada vez son más seguros.
- Manipulación de URL. Modificación manual de parámetros de URL para modificar el comportamiento de espera del servidor web.
- Debilidades de los identificadores de sesión y sistemas de autenticación.
- Inyección de código HTML y secuencias de pedidos entre *sites*.

- Inyección de pedidos SQL²⁸.

Resumen de ataques a un servidor



⁽²⁸⁾SQL es la sigla de *structured query language*; en castellano, lenguaje de consultas estructurado.

4.6.1. Falsificación de datos HTTP

La falsificación de datos HTTP es un ataque que consiste en enviar al servidor web datos introducidos manualmente para crear un contexto anormal.

Se envían datos al servidor a través de los formularios, de las URL y de las cabeceras HTTP. De manera que se pueden hacer ataques diferentes en base a cómo se enfoquen los datos que se modifiquen. Así pues, como toda esta información puede ser manipulada fácilmente por un usuario, no se puede considerar fiable desde la perspectiva de la seguridad. La seguridad no puede estar basada en verificaciones del cliente. Hay que tener presente que establecer una conexión SSL²⁹ no protege contra la manipulación de datos enviados hacia el servidor, sino que certifica la confidencialidad de la información entre la web y el usuario final.

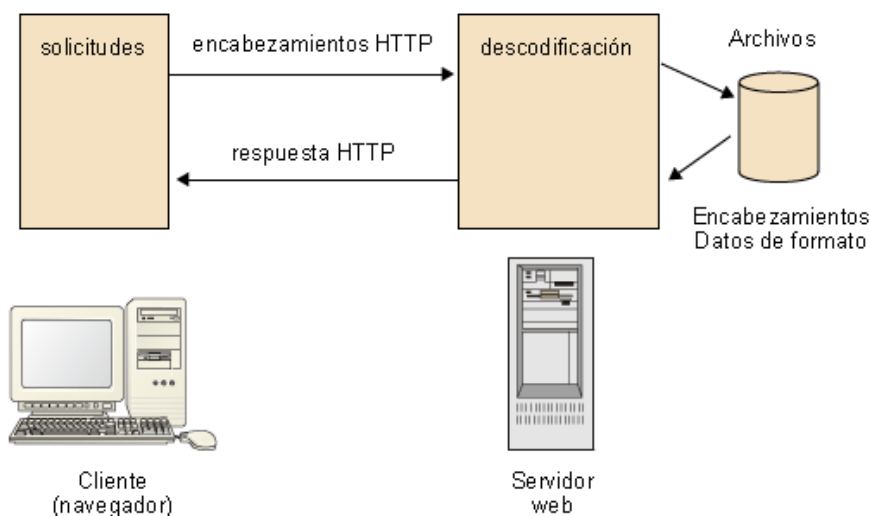
Dentro de este tipo de ataques, podemos encontrar la manipulación de URL, inyección SQL, modificación de cabeceras HTTP o la modificación de formularios. Veamos algunos para entender cómo funcionan.

⁽²⁹⁾SSL es la sigla de *secure socket layer*.

Dirección recomendada

La especificación del protocolo http 1.1 se puede consultar en: <http://www.ietf.org/rfc/rfc2616.txt>.

Esquema de falsificación de datos HTTP



4.6.2. Manipulación URL

Una URL tiene esta estructura:

Estructura de una URL

http://	Usuari:password@	www.uoc.edu	:80	/portal/castellano/inde5.html
Protocolo	Contraseña (opcional)	Nombre del servidor		Ruta
		Puerto (opcional si es 80)		

La URL permite enviar parámetros al servidor añadiendo el signo de interrogación después de la ruta. Por ejemplo, *http://www.uoc.edu/forum/?cat=5&page=1*.

La manipulación de algunas partes de una URL puede producir que un servidor web permita el acceso a páginas a las que supuestamente no tenía acceso. También puede provocar el procesamiento de un caso no previsto (parámetro inexistente o un valor no contemplado) con un comportamiento anómalo de la página web ofreciendo información.

4.6.3. Inyección de código

La inyección de código es conocida habitualmente como *cross site scripting* (XSS). La técnica se basa en inyectar y hacer funcionar *scripts* (habitualmente, *javascript*) que no están en el servidor atacado. Se puede hacer de varias maneras. Una de ellas se basa en enviar el código desde un formulario, aprovechando que el formulario no transforme los caracteres en secuencias HTML.

4.6.4. Inyección SQL

Los ataques de inyección SQL³⁰ se hacen a webs que tienen bases de datos relacionales. En este tipo de ataque, las páginas web reciben parámetros que se pasan a la base de datos como una consulta SQL. Si no se verifican los parámetros, se puede modificar la consulta para acceder a todo el contenido o modificarlo.

⁽³⁰⁾Recordad que SQL son las siglas de *structured query language*, en castellano, lenguaje de consulta estructurado.

4.7. Métodos para acceder a bases de datos de la organización

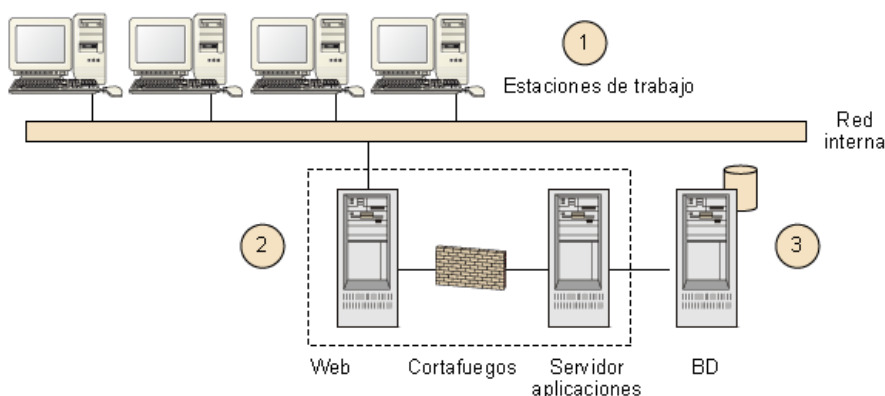
El servidor web es la parte pública de la red de la organización. Por lo tanto, la seguridad en esta parte es especialmente delicada. Veamos posibles maneras de asegurarla.

4.7.1. Arquitectura a tres niveles

La idea que hay detrás de la arquitectura a tres niveles es la de tratar de modo diferente la parte de la web de la parte de los datos. Los niveles son los siguientes:

- 1) Son los clientes que acceden a la web para consultar y/o modificar información.
- 2) Puede ser un solo servidor físico o dos, como es el caso representado en el esquema. Es recomendable que sean dos si la información es especialmente sensible. El servidor web hace de visualizador.

Esquema de la arquitectura a tres niveles



El cortafuegos está configurado para que sólo el servidor web pueda llegar hasta el servidor de aplicaciones. De esta manera, sólo una máquina puede utilizar las aplicaciones que tienen acceso a las bases de datos. También tenemos separadas la parte de la visualización de la parte de aplicación.

Otra de las razones de separar la parte de aplicación de la parte de web es para no dejar en ningún momento la máquina inconsistente por problemas del visualizador.

La comunicación entre el servidor web y el servidor de aplicaciones se puede hacer utilizando, por ejemplo, *servlets*, JSP o IAS.

3) El servidor de bases de datos tiene los datos. Es un servidor diferente, de manera que los datos están seguros y son independientes de la aplicación. Eso permite que incluso los usuarios de la base de datos los puedan mantener independientemente.

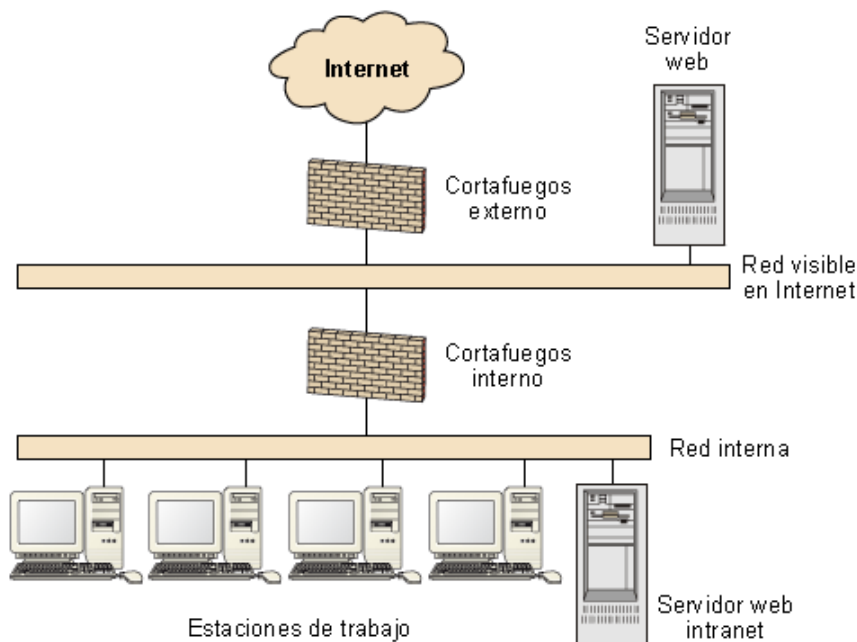
La comunicación entre el servidor de aplicaciones y la base de datos se puede hacer, por ejemplo, utilizando JDBC³¹.

⁽³¹⁾Recordad que JDBC es la sigla de *Java Database Connectivity*.

4.7.2. Restricción de accesos

Otra arquitectura para controlar el acceso es la que se muestra en el siguiente esquema.

Esquema de la arquitectura de restricción de accesos



Ved también

Consultad, en el módulo "Administración de la red", las arquitecturas de seguridad y veréis cómo estamos implementando una zona desmilitarizada (DMZ) para nuestra red.

Este esquema evita, en la medida en que puede, los ataques por Internet, ya que los cortafuegos tienen que tener seguridad al nivel de la máquina para asegurar que no entra nadie no autorizado. El servidor web también tiene que estar configurado para que máquinas (a nivel de dirección) que no sean de la organización no puedan acceder a la información.

La seguridad de los servidores web se puede ver desde muchas ópticas, ya que un servidor se dedica a publicar información de la organización posiblemente en Internet.

5. Aspectos legales

Los aspectos legales de los servidores web, y en este caso de su administrador, se centran en un punto muy discutido: la propiedad de la información que se hace pública.

Una vez más, y a costa de hacernos reiterativos, cabe decir que la legislación es muy cambiante en estos puntos, por lo cual se tiene que consultar a un asesor legal en caso de que se sospeche que puede haber problemas sobre esta materia.

¿Qué contestaríais ante las preguntas siguientes?

- ¿Podemos coger cualquier cosa de Internet y ponerla en la web?
- (Si antes habéis contestado que no) ¿Y si decimos en nuestra página de dónde lo hemos sacado?
- Si queremos poner alguna cosa en nuestra web pero no queremos que nos lo copien, ¿lo tenemos que registrar para evitarlo?
- Si todo lo que hay en Internet es de todos y podemos coger lo que queramos, ¿qué discutimos?

5.1. Propiedad intelectual y sociedad de la información

Bajo un punto de vista legal, se utilizan los términos siguientes: la **información** es un concepto que incluye datos y hechos, pero también imágenes, sonidos, textos, etc. Por ejemplo, datos meteorológicos, una película, una canción, un logotipo, un cuadro o una obra literaria. La información es un **bien valioso** y un **bien inmaterial** que está en un **soporte físico (material)** del cual se puede ser **propietario**.

Se comete un error grave al pensar que la propiedad de un soporte da derechos sobre la información que se fija. Por ejemplo, comprar un libro da derecho a leerlo, pero no a hacer copias de él y a venderlas.

Ejemplo sobre propiedad intelectual

Se ha diseñado la web de la organización y, además, se ha decidido que los usuarios tendrán páginas personales en las que podrán poner lo que quieran. Esta acción pretende motivar al personal a utilizar las nuevas tecnologías. Al cabo de poco tiempo, como no hacéis la web vosotros, como administradores del sistema os dais cuenta de que tiene texto, imágenes, fotografías, animaciones, sonidos e incluso música cogida de Internet. ¿Qué pensáis?

a) Esto se tiene que arreglar enseguida. Estas cosas no pueden estar, ya que es ilegal que aparezcan elementos que no ha hecho el autor de la web.

b) Todo esto ya estaba en Internet, e Internet es libre. Por lo tanto, se puede coger todo lo que se quiera y volver a colgarlo en Internet. ¿Qué problema hay? Si bajamos fotografías de Internet, las podemos poner en nuestra página web, ¿verdad? Si están en Internet, bien que habrá algún tipo de "licencia implícita". También podemos hacer enlaces sin pedir permiso (como podemos acceder a la página, también podemos dirigir hacia ella). ¿Y la música? Por un trocito de canción no tiene que pasar nada. Y las estadísticas de la revista del sector también están en la web de la organización (total, si aparecían en una revista, cualquiera las puede tener, comprándola o haciendo fotocopias).

La respuesta, como pasa siempre en estas situaciones, no es simple, pero sí clara: la información tiene propietario.

Un caso extremo

Pongamos un ejemplo de un caso extremo. Podemos ser propietarios de un soporte y no tener ni siquiera derecho a utilizar la información que contiene. Si compramos un software copiado ilícitamente, nadie pondrá en duda la propiedad del disquete (el soporte), pero si instalamos el programa, cometemos un acto ilícito.

Cualquier creación producto de la creatividad humana se intenta proteger mediante la propiedad intelectual, y el caso del ciberespacio no es ninguna excepción.

Framing o linking

El *framing* o *linking* consiste en mostrar contenidos de una página diferente como propios (según la percepción del usuario). El *framing* se considera una vulneración de la propiedad intelectual.

6. Tareas/responsabilidades

El administrador de la web tiene un conjunto de funciones muy amplio, especialmente si también tiene que cuidar de los contenidos:

- Hacer que la web siempre esté activa.
- Procurar que el tiempo de respuesta esté dentro de los márgenes aceptados.
- Hacer que la web sea homogénea.
- Recoger todas las consultas, quejas y propuestas que se hagan sobre la web.
- Controlar la seguridad del servidor.
- Generar las estadísticas.
- Preparar la formación de las personas que tienen que mantener los contenidos.
- Responsabilizarse (aunque parcialmente) de los contenidos.

Resumen

El administrador de la web es una figura que tiene que formar un binomio con el software del servidor web, que se compone del núcleo, los módulos y las páginas.

El administrador cuida del servidor y de la seguridad que se asocia a él. También cuida de la imagen que transmite, ya que es la imagen de la organización.

Como responsable de la web, tiene que saber comunicar los recursos de los que dispone la web para que los creadores de contenido, generalmente los usuarios finales, puedan diseñar las páginas aprovechando todo el potencial del servidor.

Primero aparecieron las páginas estáticas; posteriormente, las páginas estáticas con código insertado, en un intento de hacerlas más interactivas, y hoy en día hay páginas con código que se interpreta en el servidor, que por medio de algún mecanismo busca información y modifica la página antes de darla al navegador.

Actualmente existen muchas aplicaciones asociadas a servidores web, y por lo tanto su potencial de uso es muy grande. Ello hace que sea necesario tener especial cuidado en los aspectos de seguridad.

Actividades

1. Conectaos a una web y mirad el código fuente de la página para ver el HTML con el que está hecha. ¿Veis las marcas `<...>`?

Ejercicios de autoevaluación

1. ¿Cuáles de estas frases son ciertas y cuáles son falsas?

¿Pondríais el servidor web en el mismo servidor donde está el servidor de bases de datos de la organización?

- a) Sí, porque así es mucho más sencillo acceder a los datos para presentarlos vía web.
- b) No, hay un riesgo de seguridad.
- c) Sí, porque si no se podría cargar mucho la red
- d) No, porque si se cuelga el servidor web hay peligro de que se cuelgue el servidor de bases de datos.
- e) Si es para una intranet sí que lo pondría.
- f) En caso de que haya protección de cortafuegos, sí, porque no podrá entrar ningún ordenador no autorizado.

2. ¿Cuáles de estas acciones considerarías acertadas y cuáles no en la situación siguiente?:

Habéis visto que en una página de la organización hay una imagen que está cogida de Internet. ¿Qué hacéis?

- a) Hablar con la persona que la ha puesto para ver de dónde la ha sacado y pedir permiso para ponerla. Si os dicen que sí, la ponéis; si os dicen que no, la sacáis.
- b) La dejáis: Internet es de todos y por una imagen no hay que perder muchas horas.
- c) Sacarla inmediatamente. Cualquier información que aparece en Internet tiene propietario.
- d) Si os dice que la ha cogido de un sitio que se puede utilizar libremente, no hay que hacer nada.
- e) Aleccionad al usuario sobre los peligros legales de poner información cogida de Internet en una web.

3. Un usuario os comenta lo siguiente: "He entrado en una web y me ha cargado una página `www/.../tauler.php`. Es muy extraño, porque sé que todas las páginas acaban con `htm`. ¿Me han intentado instalar un virus en el ordenador?". ¿Qué le explicaríais?

- a) Es un lenguaje que se mezcla con el HTML y que interpreta un módulo del servidor antes de enviar la página. PHP quiere decir *hypertext preprocessor* (preprocesador de hipertexto).
- b) No, es un lenguaje diferente del HTML; por eso tiene otra extensión.
- c) No, es una cuestión de seguridad. PHP quiere decir *PHirewall protection*.
- d) Es un lenguaje que, igual que el Java, ejecuta código en vuestra máquina. Vigilad con esta web, porque os puede hacer cosas extrañas en el ordenador.

4. Una de estas tareas no es responsabilidad del administrador de web.

- a) Recoger todas las consultas, quejas, propuestas que se hagan de la web.
- b) Que la web esté siempre activa.
- c) Asegurar la disponibilidad de los datos para todos los usuarios.
- d) Controlar la seguridad del servidor.
- e) Generar las estadísticas.
- f) Controlar la seguridad del servidor.

5. Relacionad las tecnologías con la base de funcionamiento.

JavaScript
CGI
Perl
PHP
ASP
JSP
Java
Servlets

Ejecuta en el cliente
Ejecuta en servidor
Código mezclado HTML
Interpreta cliente
Código mezclado HTML
Interpreta servidor

Solucionario

Ejercicios de autoevaluación

1. Argumentemos un poco cada respuesta.

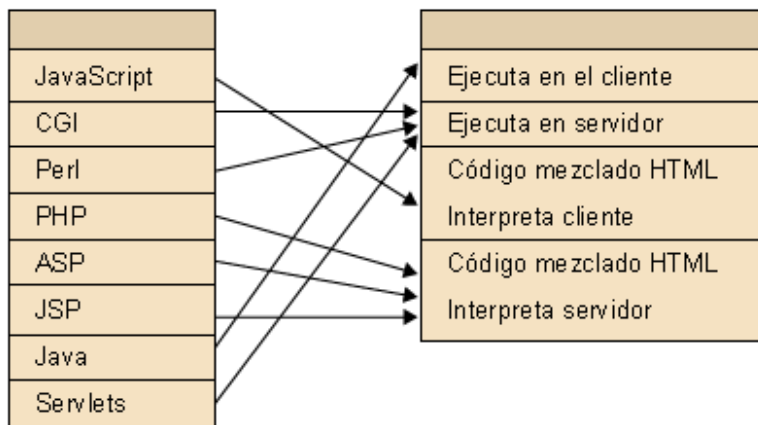
- a) Falso. De entrada, para acceder a una base de datos desde un servidor web, no tiene importancia dónde está la base de datos. Es igual de complejo si está en el mismo servidor o en otro servidor físico.
- b) Cierto. Siempre es un riesgo de seguridad. Cualquier intrusión que deje abierto el servidor web permitirá acceder fácilmente a los datos, dado que estarán en el mismo ordenador.
- c) Falso. Para presentar páginas viaja poca información por la red. Si se prevé mucho tráfico hay diseños de red que lo aíslan entre el servidor web y el servidor de bases de datos. También se pueden utilizar conectores en lugar de concentradores para no cargar la red con este tráfico.
- d) Cierto, aunque depende de los servidores y los operativos. Por ejemplo, los Unix, si el servidor da problemas, es muy extraño que puedan llegar a colgar todo el sistema. Sin embargo, es una posibilidad que no se puede descartar. Según la función que haga el servidor de base de datos es un riesgo que no se puede correr. Lógicamente, para b) no es aconsejable.
- e) Cierto, siempre que además no contenga datos sensibles. También se tendría que tener en cuenta d), el peligro de colgar el ordenador.
- f) Falso. Además, no tiene mucho sentido un servidor web abierto a todo Internet con protección de acceso a sólo unos ordenadores determinados.

2. Excepto b, todas las otras son coherentes.

3. a

4. c Asegurar la disponibilidad de los datos para todos los usuarios no es tarea del administrador web, sino del administrador de los datos.

5.



Glosario

CMS *m* Sigla de *content management system*. Aplicación para crear y administrar contenidos.

CSS *m* Sigla de *cascading style sheets*, estándar propuesto en 1996 por el W3C.

cookie *f* Ved **galleta**.

detector *m* Programa destinado a mirar de una manera ilícita la información que pasa por una red informática.
en sniffer

DHTML *m* Es la sigla de *dynamic HTML*, 'HTML dinámico'.

enlace *m* Mecanismo específico para acceder a un recurso de Internet. Lo utilizan especialmente las páginas web, en un formato definido `<A> <href="URL">text`.
en link.

extranet *f* Nombre para designar Internet. Nace por oposición a *intranet*.

framing Ved **linking**.

galleta Las galletas (*cookies*) son pequeños ficheros que algunas webs guardan en el ordenador del usuario.

HTML *m* Ved **lenguaje de etiquetado de hipertexto**.

hypertext markup language *m* Ved **lenguaje de etiquetado de hipertexto**.

intranet *f* Red interna de una organización que puede funcionar con TCP/IP, y además puede utilizar los recursos de Internet, como pueden ser navegadores, servidores web, etc.

link *m* Ved **enlace**.

linking *m* Consiste en mostrar contenidos de una página diferente como propios. Se considera una vulneración de la propiedad intelectual.

lenguaje de etiquetado de hipertexto *m* Lenguaje estándar para etiquetar documentos en formato de hipertexto con el fin de indicar a un navegador cómo tiene que visualizar un documento en la pantalla del ordenador.
en hypertext markup language.
sigla: **HTML**.

localizador uniforme de recursos *m* Manera única de especificar un recurso en Internet.
en uniform resource locator.
sigla: **URL**.

modelo en objetos para la representación de documentos *m* Modelo orientado a objetos que los programas y *scripts* pueden usar para acceder y modificar dinámicamente el contenido, estructura y estilo de los documentos HTML.
sigla: **DOM**.

rich site summary *m* Formato de datos usado para difundir contenidos a suscriptores de una web.
sigla: **RSS**.

servidor web *m* Aplicación que funciona en un ordenador que escucha por un puerto y sirve páginas web bajo demanda.

sniffer *m* Ved **detector**.

uniform resource locator *m* Ved **localizador uniforme de recursos**.

URL *m* Ved **localizador uniforme de recursos**.

webmaster *m y f* Responsable de la web. Es la persona que cuida de los contenidos (no tiene que ser necesariamente el administrador de la web).

webmail *m* Correo basado en web.

W3C *m* World Wide Web Consortium. El consorcio que marca el estándar de HTML, entre muchas otras cosas. <http://www.w3c.org>.

wiki *f* Web donde colaboran varios autores.

XML *m* *EXtensible Markup Language* o lenguaje de etiquetado extensible. Es un estándar ajustado a las necesidades de Internet.

Bibliografía

Coar, K. (1998). *Apache Server for Dummies*. Estados Unidos: IDG Books Worldwide, Inc.

Dwivedi (2007). *Hacking Exposed Web 2.0: Web 2.0 Security Secrets and Solutions*. Estados Unidos: McGraw-Hill.

Garfinkel, S.; Spafford, G. (1997). *Web Security & Commerce*. Estados Unidos: O'Reilly & Associates, Inc.

Morales, O. (2001). *Informàtica Jurídica*. Barcelona: Universitat Oberta de Catalunya.

Musciano, C.; Kennedy, B. (2000). *HTML & XHTML The Definitive Guide*. Estados Unidos: O'Reilly & Associates, Inc.

Pfleeger, C. (1997). *Security in Computing*. Estados Unidos: Prentice Hall.

Pressman (2008). *Web Engineering: A Practitioner's Approach*. Estados Unidos: McGraw-Hill.

Spainhour, S.; Quercia, V. (1996). *WebMaster in a Nutshell A Desktop Quick Reference*. Estados Unidos: O'Reilly & Associates, Inc.