

Temas Generales para la preparación de la Oposición al Cuerpo Superior de Sistemas y Tecnologías de la Información de la Administración del Estado.

**Cuerpo Superior de Estadísticos del Estado  
Especialidad de Estadística-Ciencia de Datos.**

**Almacenamiento y modelos de datos**

**Tema 7. Tecnologías web. I: la web.**

**AUTOR:** Alexandra Vlad

**Asociación Profesional de Cuerpos Superiores de Sistemas y  
Tecnologías de la Información de las Administraciones Públicas**

Creación: Junio 2021

---

## ÍNDICE

1	INTRODUCCIÓN .....	3
2	EL PROTOCOLO DE TRANSFERENCIA DE HIPERTEXTO (HTTP) .....	4
3	EL LENGUAJE DE MARCADO DE HIPERTEXTO (HTML) ...	7
4	LOCALIZADORES DE RECURSOS UNIFORMES (URL UNIFORM RESOURCE LOCATORS) .....	9
5	PÁGINAS WEB ESTÁTICAS Y DINÁMICAS .....	10
6	SERVICIOS WEB .....	11
7	REQUISITOS PARA LA INTEGRACIÓN WEB-DBMS.....	12
8	VENTAJAS Y DESVENTAJAS DEL ENFOQUE WEB-DBMS.	13
9	ENFOQUES PARA LA INTEGRACIÓN DE LA WEB Y LOS DBMSS .....	17
10	RESUMEN ESQUEMÁTICO .....	18
11	GLOSARIO .....	19
12	BIBLIOGRAFÍA BÁSICA.....	20

---

# 1 Introducción

## La Web

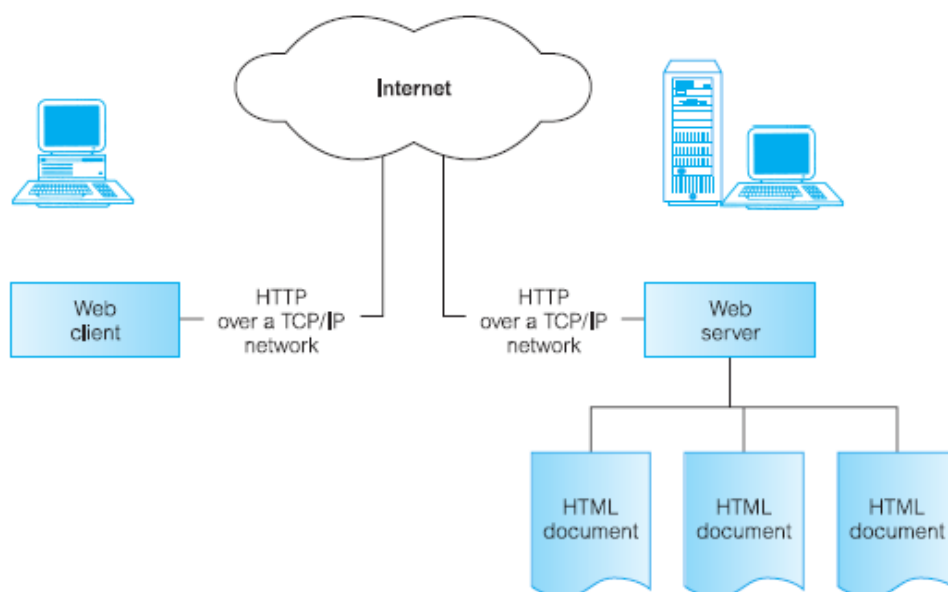
La web es un sistema hipermedia global o inter redes, que proporciona los medios para buscar información en internet de una manera no secuencial a través de hiperenlaces.

La World Wide Web (Web para abreviar) proporciona un simple “apuntar y pulsar” (“point and click”) que permite explorar un volumen enorme de páginas de información que residen en Internet (Berners-Lee, 1992). La información en la Web se presenta a través de las páginas web, que son una colección de texto, gráficos, imágenes, sonidos y videos. Además, una página web puede contener hiperenlaces a otras páginas web, que permite a los usuarios navegar de una manera no secuencial a través de la información.

Gran parte del éxito de la web se debe a la simplicidad con la que permite a los usuarios proporcionar, utilizar y hacer referencia a información distribuida geográficamente en todo el mundo. Además, proporciona a los usuarios la capacidad de buscar documentos multimedia independientemente del dispositivo que usan. También es compatible con otros protocolos de comunicación de datos, tales como Gopher, FTP (File Transfer Protocol), NNTP (Network News Transfer Protocol) y Telnet.

La web consiste de una red de ordenadores que pueden tener dos roles: como *servidor*, proporcionando información; y como *cliente*, normalmente referenciados como navegadores, solicitando información. Ejemplos de Servidores Web son Apache HTTP Server, Microsoft Internet Information Server o iPlanet de Oracle. Y como navegadores tenemos el ejemplo de Microsoft Internet Explorer, Edge, Google Chrome, Mozilla Firefox o Safari, entre otros.

Gran parte de la información en la web se almacena en documentos que usan el lenguaje llamado HTML (HyperText Markup Language) y los navegadores tienen que entender e interpretar el lenguaje HTML para poder mostrar dichos documentos. El protocolo que gobierna el intercambio de información entre servidores web y el navegador se denomina HTTP (HyperText Transfer Protocol)



---

## 2 El protocolo de transferencia de hipertexto (HTTP)

HTTP HyperText Transfer Protocol es un protocolo que se utiliza para transferir páginas web a través de internet. El protocolo http define como se comunican los clientes y servidores. HTTP es un protocolo genérico, sin estado, orientado a objetos que permite transmitir información entre servidores y clientes (Berners-Lee, 1992).

HTTP/0.9 se utilizó durante el desarrollo inicial de la web. HTTP/1.0, que se publicó en 1995 como RFC 1945 informativo, reflejaba el uso común del protocolo. Esta versión no llegó a ser un estándar oficial. La versión HTTP/1.1, ofrece más funcionalidades que su versión anterior entre las cuales destaca que se permiten transacciones múltiples entre cliente y servidor en la misma petición. Se considera el protocolo estándar. Éste fue publicado inicialmente como [RFC 2068](#) en Enero de 1997. HTTP/2, también conocida como HTTP 2.0, se publicó en mayo de 2015. Tiene notables diferencias fundamentales respecto a la versión anterior HTTP/1.1

- Es un protocolo binario, en contraposición a estar formado por cadenas de texto, tal y como estaban basados sus protocolos anteriores. Así pues, no se puede leer directamente, ni crear manualmente. A pesar de este inconveniente, gracias a este cambio es posible utilizar en él técnicas de optimización.
- Es un protocolo multiplexado. Peticiones paralelas pueden hacerse sobre la misma conexión, no está sujeto pues a mantener el orden de los mensajes, ni otras restricciones que tenían los protocolos anteriores HTTP/1.x
- Comprime las cabeceras, ya que estas, normalmente son similares en un grupo de peticiones. Esto elimina la duplicación y retardo en los datos a transmitir.
- Esto permite al servidor almacenar datos en la caché del cliente, previamente a que estos sean pedidos, mediante un mecanismo denominado 'server push'.

HTTP/3 es la próxima y tercera versión principal del Protocolo de Transferencia de Hipertexto utilizado para intercambiar información en la World Wide Web, que sucederá a HTTP/2.1. Aún está en borrador.

El protocolo HTTP se basa en un paradigma petición-respuesta (o paradigma cliente-servidor). Una transacción HTTP consiste en lo siguiente:

- Conexión: El cliente establece una conexión con el servidor web.
- Petición: El cliente envía un mensaje de petición al servidor web.
- Respuesta: El servidor web envía una respuesta (por ejemplo, un documento HTML) al cliente.
- Cierre: El servidor web cierra la conexión.

Actualmente, HTTP es un protocolo sin estado, es decir (independiente del estado), el servidor no guarda ninguna información sobre conexiones anteriores. Esto significa que la información que un usuario introduce en una página web (a través de un formulario, por ejemplo) no está disponible de forma automática en la siguiente página solicitada, al menos que el servidor web implemente medidas para ello. En este caso, el servidor debe identificar de alguna manera que solicitudes, entre las miles de peticiones que recibe, provienen del mismo usuario. Para la mayoría de las aplicaciones, esta propiedad sin estado de HTTP es un beneficio que permite una comunicación basada en una lógica simple y ejecutarse "lean", es decir sin necesidad de memoria adicional o sin ocupar espacio en disco con información de anteriores peticiones. Se han propuesto varios esquemas para compensar la naturaleza sin estado del protocolo HTTP, tales como proporcionar páginas web con campos ocultos que contienen identificadores de transacción, y el uso de formularios web donde toda la información se introduce de forma local y luego se envía en una sola transacción. Todos estos esquemas tienen sus limitaciones en función de las aplicaciones web y requieren extensiones especiales en los servidores web.

---

## MIME Multipurpose Internet Mail Extensions

Las especificaciones MIME (en español “extensiones multipropósito de correo de internet”) definen un estándar para codificar datos binarios en ASCII, así como un estándar para indicar el tipo de datos que contiene un mensaje. Aunque originalmente se usaba por el software cliente de correo electrónico, la web también utiliza el estándar MIME para determinar cómo gestionar los diferentes tipos de datos. Los tipos MIME se identifican por el formato tipo/subtipo, donde el **tipo** define el tipo de general de datos a enviar y el **subtipo** identifica el tipo específico del formato utilizado. Por ejemplo, una imagen GIF se formateará como imagen/gif. Algunos otros tipos se enumeran en la siguiente tabla.

TIPO MYPE	SUBTIPO MIME	DESCRIPCIÓN
texto	html plain	Ficheros HTML (*.htm, *.html) Ficheros ASCII (*.txt)
imagen	jpeg gif x-bitmap	Ficheros Joint Photographic Experts Groups (*.jpeg) Ficheros Gpaohics Interchange Format (*.gif) Ficheros Microsoft bitmap (*.bmp)
vídeo	x-msvideo quicktime mpeg	Ficheros Microsoft Audio Video Interleave (*.avi) Ficheros Apple QuickTime Movie (*.mov) Ficheros Moving Picture Experts Group (*.mpeg)
aplicación	postscript pdf java	Ficheros PostScript (*.ps) Ficheros Adobe Acrobat (*.pdf) Ficheros Java class (*.class)

## PETICIÓN HTTP

Una petición HTTP consiste en una cabecera que indica el tipo de petición, el nombre del recurso, y la versión HTTP, seguido por un cuerpo opcional. La cabecera está separada del cuerpo por una línea en blanco. Los tipos principales de petición HTTP son:

- GET. Es uno de los tipos de petición más comunes, que recupera (gets) el recurso que ha solicitado el usuario.
- POST. Otro tipo de petición común, transfiere (posts) datos a un recurso específico. Generalmente, los datos enviados provienen de un formulario HTML que el usuario ha rellenado, y el servidor puede usar estos datos para buscar en internet o hacer una petición a la base de datos.
- HEAD. Similar a GET pero obliga al servidor a devolver solamente una cabecera HTML en vez de unas respuesta.
- PUT (HTTP/1.1). Carga el recurso en el servidor.
- DELETE (HTTP/1.1). Elimina el recurso del servidor.
- OPTIONS (HTTP/1.1). Solicita las opciones de configuración del servidor.

## RESPUESTA HTTP

Una respuesta HTTP tiene un encabezado que contiene la versión HTTP, el estado de la respuesta e información de encabezado para controlar el comportamiento de la respuesta, así como cualquier dato

---

solicitado en un cuerpo de respuesta. Nuevamente, el encabezado está separado del cuerpo por una línea en blanco.

---

### 3 El lenguaje de marcado de hipertexto (HTML)

El lenguaje de marcado de hipertexto (HTML HypeText Markup Language) es un sistema para marcar o etiquetar un documento para que pueda ser publicado en la Web. HTML define lo que generalmente se transmite entre los nodos de la red. Es un lenguaje sencillo, pero potente, independiente de la plataforma.

El lenguaje HTML fue creado por el padre de la web, Berner Tim Berners-Lee en 1989, mientras estaba trabajando como investigador en el CERN (Organización Europea para la Investigación Nuclear), aunque HTML no se llegó a estandarizar hasta noviembre de 1995 por parte del organismo IETF (Grupo de trabajo de ingeniería de Internet) como RFC 1866, comúnmente conocido como HTML versión 2. A pesar de su nombre, HTML 2.0 es el primer estándar oficial de HTML.

En diciembre de 1999, se publicó HTML 4.10 como recomendación del W3C, el cual incluye mecanismos para marcos, hojas de estilo, secuencias de comandos y objetos incrustados (W3C, 1999a). A principios del año 2000, W3C publicó XHTML 1.0 (eXtensible HyperText Markup Language) como una reformulación de HTML 4 en XML (eXtensible Markup Language) (W3C, 2000a).

En 2004 la W3C reabrió el debate de la evolución del HTML, y se dieron a conocer las bases para la versión HTML5. No obstante, este trabajo fue rechazado por los miembros del W3C y se daría preferencia al desarrollo del XML. Apple, Mozilla y Opera anunciaron su interés en seguir trabajando en el proyecto bajo el nombre de WHATWG, que se basa en la compatibilidad con tecnologías anteriores. En 2006, el W3C se interesó en el desarrollo de HTML5, y en 2007 se unió al grupo de trabajo del WHATWG para unificar el proyecto. Finalmente se publicó a finales de 2014.

HTML 5 añade nuevas características sintácticas como elementos de video, audio y canvas, así como la integración de contenido SVG (scalable vector graphics) y MathML para fórmulas matemáticas. Estas características se diseñaron para facilitar la inclusión y el manejo de contenido multimedia y contenido gráfico en la web sin tener que

HTMLS se desarrolló con la intención de que varios tipos de dispositivos pudiesen usar la información de la web: PCs con pantallas de diferentes resoluciones y colores, móviles, dispositivos PDAs, dispositivos de entrada y salida de voz, etc.

HTML es una implementación del lenguaje de marcado generalizado estándar (SGML Standardized Generalized Markup Language), un sistema que define tipos de documentos estructurados y lenguajes de marcado para representar instancias de esos tipos de documentos (ISO, 1986).

Ejemplo:

```
<HTML>
<HEAD>
<TITLE>Database Systems: A Practical Apporach to Design, Implementation and Mamagement
</TITLE>
</HEAD>
<BODY bgcolor=#FFFFCC>
<H2> Database Systems: A Practical Approach to Design, Implementation and Mamagement </H2>
<P>Thank you for visiting the Home Page of our database text book, From this page you can view online
a selection of chapters from the book. Academics can also Access the Instructos's Guide, but this
requires the specification of a user name and password, wich must first be obtained from Addison
Wesley Longman. <BR>
```

---

<BR>

<A HREF = ""> Table of Contents

</BODY>

</HTML>

---

## 4 Localizadores de recursos uniformes (URL Uniform Resource Locators)

Una URL es una cadena de caracteres alfanuméricos que representa la localización o dirección de un recurso en Internet y la forma de acceder dicho recurso.

Las URLs definen de forma unívoca donde se pueden encontrar los documentos (recursos) en internet. URIs y URNs son otros términos relacionados que podemos encontrar. Uniform Resource Identifier (URI) o Identificador de Recursos Uniformes, es el conjunto genérico de todos los nombres/direcciones que hacen referencia a los recursos de Internet. Uniform Resource Name (URN) o Nombre de Recurso Uniforme también designa un recurso en Internet pero lo hace usando un nombre persistente, independiente de la ubicación. URNs son muy genéricos y dependen de los servicios de búsqueda de nombres, y, por lo tanto, dependen de servicios adicionales que generalmente no están siempre disponibles. Por otro lado, las URLs, identifican un recurso en Internet usando un esquema basado en localización de recursos. Las URLs son los esquemas de identificación de más usados y son la base para HTTP y la web.

La sintaxis de una URL es bastante sencilla y consiste en tres partes básicas: el protocolo utilizado para la conexión, el nombre del host, y el nombre de host y el nombre de la ruta en ese host donde puede encontrarse el recurso. Además, la URL puede especificar opcionalmente el puerto a través del cual se debe realizar la conexión al host (por defecto puerto 80 para HTTP), y una cadena de consulta (query string), que es uno de los métodos principales para pasar datos del cliente al servidor. La sintaxis de una URL es la siguiente:

`<protocolo>:// <host> [:<puerto>] / ruta_absoluta [? argumentos]`

El `<protocolo>` especifica el mecanismo que utilizará el navegador para comunicarse con el recurso. Los métodos de acceso comunes son HTTP, S-HTTP (secure HTTP), file (carga un archivo desde un disco local), FTP, mailto (envía un correo a la dirección de correo especificada), Gopher, NNTP y Telnet. Por ejemplo,

`http://www.w3.org/MarkUp/MarkUp.html`

es una URL que identifica la página de inicio general de información HTML en W3C. El protocolo es HTTP, el host es `www.w3.org` y la ruta virtual del archivo HTML es `/MarkUp/MarkUp.html`.

---

## 5 Páginas web estáticas y dinámicas

Un documento HTML almacenado en un fichero es un ejemplo de una página web estática, es decir el contenido del documento no cambia al menos que el fichero en si cambie. Por otro lado, el contenido de una página web dinámica se genera cada vez que se accede a ella. Como resultado, una página web dinámica tiene características que no se encuentran en una página web estática, por ejemplo:

- Puede responder a una entrada del usuario desde el navegador. Por ejemplo, devolviendo datos solicitados a través de un formulario o resultados de una consulta a base de datos.
- Se puede customizar por y cada uno de los usuarios. Por ejemplo, una vez el usuario haya especificado algunas preferencias cuando accede a un sitio o página en concreto (como un área de interés o nivel de expertise), esta información se puede almacenar y devolver resultados en base a dichas preferencias.

Cuando los documentos a publicar son dinámicos, como aquellos que son consecuencias de consultas a base de datos, el hipertexto se tiene que generar desde el servidor. Para ello, se pueden escribir scripts que implementan la conversión desde diferentes formatos de datos a HTML al vuelo. Estos scripts también necesitan entender las consultas implementadas por los clientes a través de formularios HTML y los resultados generados por las aplicaciones que contienen los datos (por ejemplo, un sistema de administración de base de datos, DBMS). Como una base de datos es dinámica, cambia a medida que los usuarios crean, insertan, actualizan y eliminan datos, generar páginas webs dinámicas es un enfoque mucho más apropiada que crear páginas web estáticas.

---

## 6 Servicios Web

En los últimos años, los servicios web se han establecido como un paradigma importante en el desarrollo de aplicaciones y procesos de negocio para la integración de aplicaciones heterogéneas en un futuro. Los servicios web se basan en estándares abiertos y se centran en la comunicación y la colaboración entre personas y aplicaciones. A diferencia de otras aplicaciones web, los servicios web no tienen interfaz de usuario y no están soportados por los navegadores web. En cambio, consisten en componentes de software reutilizables diseñados para ser consumidos por otras aplicaciones, tales como aplicaciones cliente tradicionales, aplicaciones web u otros servicios web.

Existen varias definiciones de servicios web, por ejemplo, “una colección de funciones que se empaquetan en una única entidad y se publica en la red para que las utilicen otros programas”, o la definición de W3C, “un sistema software diseñado para soportar interacción interoperable máquina a máquina a través de la internet”. Un ejemplo típico de servicios web es la cotización de acciones de una empresa, que recibe una petición para conocer el precio actual de una acción y responde con el precio solicitado. Como un segundo ejemplo, Microsoft ha desarrollado un servicio de mapa web que permite mapas de alta calidad, direcciones y otra información de localización para que se pueda integrar en aplicaciones de usuarios, procesos de negocio o páginas web.

En el enfoque de servicios web es fundamental el uso de tecnologías ampliamente reconocidas y estándares de uso común, tales como:

- eXtensible Markup Language (XML).
- SOAP (Simple Object Access Protocol), basado en XML y usado para la comunicación a través de Internet.
- WSDL (Web Services Description Language), de nuevo basado en XML, y utilizado para describir los servicios web. WSDL añade una capa de abstracción entre la interfaz y la implementación, proporcionando un servicio desacoplado para mayor flexibilidad.
- UDDI (Universal Discovery, Description and Integration), que se utiliza para registrar los servicios web para posibles usuarios.

---

## 7 Requisitos para la integración Web-DBMS

La web basada en sistemas de gestión de bases de datos es una de las partes esenciales de un DBMS (DataBase Management System o sistema de gestión de base de datos) y se utiliza para almacenar los datos de las aplicaciones web. Una web basada en un sistema de gestión de base de datos se utiliza para manejar esas bases de datos que contienen datos relativos a comercio electrónico, negocio electrónico, blogs, correo electrónico y otras aplicaciones online.

Si bien muchos proveedores de DBMS están trabajando para proporcionar soluciones propietarias de conectividad con bases de datos para la Web, la mayoría de las organizaciones requieren una solución más general para evitar estar atados a una sola tecnología. A continuación, se enumeran algunos de los requisitos más importantes para la integración de las aplicaciones de bases de datos con la web:

- La capacidad de acceder a datos corporativos valiosos de manera segura.
- Conectividad independiente entre datos y proveedores para permitir la libertad de elección en la selección del DBMS ahora y en el futuro.
- La capacidad de interactuar con la base de datos independientemente de cualquier navegador web propietario o servidor web.
- Una solución de conectividad que aprovecha todas las características del DBMS de una organización.
- Un enfoque de arquitectura abierta que permita la interoperabilidad con una variedad de sistemas y tecnologías; por ejemplo, soporte para:
  - diferentes servidores web;
  - el framework .NET de Microsoft;
  - CORBA / IIOP (Internet Inter-ORB protocol);
  - Java / RMI (Remote Method Invocation);
  - XML;
  - Servicios web (SOAP, WSDL y UDDI; RESTful).
- Una solución rentable que permite escalabilidad, crecimiento y cambios en las decisiones estratégicas y ayuda a reducir los costes de desarrollo y mantenimiento de aplicaciones.
- Soporte para transacciones que abarcan múltiples peticiones HTTP.
- Soporte para autenticación basada en sesión y autenticación basada en aplicación.
- Rendimiento aceptable.
- Gastos de administración mínimos.
- Un conjunto de herramientas de productividad de alto nivel que permitan desarrollar, mantener, y desplegar aplicaciones con relativa facilidad y velocidad.

---

## 8 Ventajas y desventajas del enfoque Web-DBMS.

La web, como plataforma para sistemas de base de datos, puede proporcionar soluciones innovadoras para operaciones de negocio tanto inter- como intra- compañías. Desafortunadamente, dentro de este enfoque también existen desventajas.

Las ventajas se enumeran a continuación:

- Simplicidad
  - En su formato original, HTML como lenguaje de marcado fue fácil tanto para desarrolladores como para usuarios finales que quería aprender. Hasta cierto punto, esto sigue siendo cierto siempre que la página HTML no tenga funcionalidad demasiado compleja. Sin embargo, HTML se está ampliando continuamente con nuevas o mejoradas características, y los lenguajes de scripts se pueden incrustar en el HTML, por lo que podría decirse que la simplicidad original ha desaparecido.
- Independencia tecnológica
  - Una razón de peso para crear una versión basada en web de una aplicación de base de datos es que los clientes web (los navegadores) son en su mayoría independientes de la plataforma. Como existen navegadores para las principales plataformas informáticas, siempre que se utilice HTML / Java estándar, no es necesario modificar las aplicaciones para que se ejecuten en diferentes sistemas operativos o entornos basados en Windows. Por otro lado, los clientes de bases de datos tradicionales requieren una gran modificación, si no una reingeniería total, para exportarlos a múltiples plataformas. Desafortunadamente, algunos proveedores de navegadores web ofrecen características propietarias y los beneficios de esta ventaja podría decirse que han desaparecido.
- Interfaz gráfica de usuario
  - Un problema importante en el uso de una base de datos es el acceso a los datos. A las bases de datos se puede acceder a través de una interfaz de menú basada en texto o mediante una interfaz de programación, como la especificada en el estándar SQL. Sin embargo, estas interfaces pueden ser engorrosas y difíciles de usar. Por otro lado, una buena interfaz gráfica de usuario (GUI) puede simplificar y mejorar el acceso a la base de datos. Desafortunadamente, las GUI requieren una programación extensa y tienden a depender de la plataforma y, en muchos casos, son específicas del proveedor. Por otro lado, los navegadores web proporcionan una GUI común y fácil de usar que se puede utilizar para acceder a muchas cosas, incluida una base de datos. Tener una interfaz común también reduce los costes de capacitación para los usuarios finales.
- Estandarización
  - HTML es un estándar de facto al que se adhieren todos los navegadores web, lo que permite que los usuarios lean un documento HTML en una máquina o en cualquier máquina del mundo con una conexión a Internet y un navegador web. Con HTML, los desarrolladores aprenden un solo lenguaje y los usuarios finales usan una única GUI. Sin embargo, como se señaló anteriormente, el estándar se está fragmentando a medida que los proveedores ahora ofrecen características patentadas que no están disponibles universalmente. La introducción más reciente de XML ha agregado una mayor estandarización y XML se ha convertido muy rápido en el estándar de facto para el intercambio de datos.
- Soporte multi plataforma
  - Los navegadores web están disponibles para prácticamente todos los tipos de plataformas informáticas. Este soporte multiplataforma permite a los usuarios de la mayoría de los tipos de ordenadores acceder a una base de datos desde cualquier parte del mundo. De esta forma, la información se puede difundir con un mínimo de

---

tiempo y esfuerzo, sin tener que resolver los problemas de incompatibilidad de diferentes hardware, sistemas operativos y software.

- Acceso transparente a red
  - Un beneficio importante de la web es que el acceso a la red es prácticamente transparente para el usuario, excepto por la especificación de una URL, que se gestiona íntegramente por el navegador y el servidor web. Este soporte integrado para redes simplifica enormemente el acceso a base de datos, eliminando la necesidad de un software de red costoso y la complejidad de lograr que diferentes plataformas se comuniquen entre sí.
- Despliegue escalable
  - La arquitectura más tradicional de dos capas cliente-servidor da lugar a clientes “pesados” que procesan de manera ineficiente tanto la interfaz de usuario como la lógica de la aplicación. Por el contrario, una solución basada en web tiende a crear una arquitectura de tres capas más natural que proporciona una base para escalabilidad. Al alojar la aplicación en un servidor separado en lugar de en el cliente, la Web elimina el tiempo y coste ligados al despliegue de la aplicación. Simplifica el manejo de actualizaciones y la administración de la gestión de múltiples plataformas en múltiples oficinas. Ahora, desde el servidor de aplicaciones, se puede acceder a la aplicación desde cualquier sitio web del mundo. Desde una perspectiva empresarial, el acceso global a las aplicaciones del lado del servidor brinda la posibilidad de crear nuevos servicios y abrir nuevas bases de clientes.
- Innovación
  - Como plataforma de Internet, la Web permite a las organizaciones proveer nuevos servicios y llegar a nuevos clientes a través de aplicaciones accesibles a nivel mundial. Estos beneficios no estaban disponibles anteriormente a través de las aplicaciones de software colaborativo y cliente-servidor tradicionales o basadas en host. Durante la última década, hemos visto el auge de las empresas "puntocom" y hemos sido testigos de la expansión significativa de las transacciones de empresa a empresa (B2B) y de empresa a consumidor (B2C) a través de la Web. Hemos sido testigos de nuevas estrategias de marketing, así como de nuevos modelos comerciales que antes del desarrollo de la Web y sus tecnologías asociadas no eran posibles.

Ventajas que se obtienen mediante el uso de un sistema gestor de base de datos (DBMS)

Simplicidad

Independencia tecnológica

Interfaz gráfica de usuario

Estandarización

Soporte multi plataforma

Acceso transparente a red

Despliegue escalable

Innovación

Desventajas:

- Fiabilidad
  - Internet es actualmente un medio de comunicación lento y no fiable, cuando una petición se lleva a cabo a través de Internet, no hay una garantía real de entrega (por ejemplo, el servidor puede estar caído). Las dificultades surgen cuando los usuarios intentan acceder la información alojada en un servidor en un momento pico cuando está significativamente sobrecargado o cuando usan una red que es particularmente

---

lenta. La fiabilidad de Internet es un problema que llevará tiempo abordar. Junto con la seguridad, la fiabilidad es una de las principales razones por las que las organizaciones continúan dependiendo de sus propias intranets en lugar de la Internet pública para aplicaciones críticas. La intranet privada está bajo control organizacional, para ser mantenida y mejorada cuando la organización lo considere necesario.

- Seguridad

- La seguridad es una gran preocupación para una organización que hace que sus bases de datos sean accesibles en la Web. La autenticación de usuarios y las transmisiones seguras de datos son fundamentales debido a la gran cantidad de usuarios potencialmente anónimos.

- Coste

- Contrario a la creencia popular, mantener una presencia en internet no trivial puede resultar costoso, especialmente con la creciente demanda y expectativas de los usuarios. Por ejemplo, un informe de Forrester Research indicó que el coste de un sitio web comercial varía desde 300000 \$ a 3,4 millones \$, dependiendo de los objetivos de la organización para la web, y predijo que los costes aumentarán entre un 50% y un 200 % en el próximo par de años. En el extremo superior de la escala se encontraban los sitios que vendían productos o realizaban transacciones, con el 20% de los costes en hardware y software, el 28% en marketing de la propia web y el 56% restante en el desarrollo del contenido del sitio. Claramente, poco se puede hacer para reducir el coste del desarrollo creativo del material web, sin embargo, con herramientas mejoradas y middleware de conectividad, debería ser posible reducir significativamente los costes de desarrollo técnico.

- Escalabilidad

- Las aplicaciones web pueden enfrentarse a picos de carga impredecibles y potencialmente enormes. Esto requiere el desarrollo de una arquitectura de servidor de alto rendimiento que sea altamente escalable.

Para mejorar la escalabilidad, se han introducido granjas web con dos o más servidores que alojan el mismo sitio. Las solicitudes HTTP generalmente se enrutan a cada servidor de la granja en forma de turnos, para distribuir la carga y permitir que el sitio maneje más solicitudes. Sin embargo, esto puede hacer que el mantenimiento de la información del estado sea más complejo.

- Funcionalidad limitada de HTML

- Aunque HTML proporciona una interfaz común y fácil de usar, su simplicidad significa que algunas aplicaciones de bases de datos altamente interactivas pueden no convertirse fácilmente en aplicaciones basadas en la Web y, al mismo tiempo, brindar la misma facilidad de uso. Es posible agregar funcionalidad adicional a una página web usando un lenguaje de scripting como JavaScript o VBScript, o utilizar componentes Java o ActiveX, pero la mayoría de estos enfoques son demasiado complejos para usuarios finales aficionados. Además, hay una sobrecarga de rendimiento al descargar y ejecutar este código.

- Sin estado

- El enfoque sin estado del entorno web dificulta la gestión de las conexiones a base de datos y las transacciones de usuarios, lo que requiere que las aplicaciones mantengan información adicional.

- Ancho de banda

- Actualmente, un paquete se mueve a través de una LAN a un máximo de 10 millones de bits por segundo (bps) para Ethernet y 2500 millones de bps para ATM. Por el contrario, en una de las partes más rápidas de Internet, un paquete solo se mueve a una velocidad de 1.544 millones de bps. En consecuencia, el recurso restrictivo de Internet es el ancho de banda, y depender de las llamadas a través de la red al servidor para realizar incluso la tarea más simple (incluido el procesamiento de un formulario) agrava el problema.

- Rendimiento
  - Muchas partes de los clientes web complejos de base de datos se centran alrededor de lenguajes interpretados, haciendo que vayan más lentos que los clientes tradicionales de base de datos, que se compilan de forma nativa. Por ejemplo, HTML debe ser interpretado y renderizado por un navegador web; JavaScript y VBScript son lenguajes de secuencias de comandos interpretados (scripts) que amplían HTML con construcciones de programación; un applet Java se compila en un código de bytes, y es este código de bytes el que descarga e interpreta el navegador. Para aplicaciones de tiempo crítico, la sobrecarga de los lenguajes interpretados puede ser demasiado prohibitivos. Sin embargo, hay muchas más aplicaciones para las que el tiempo no es tan importante.
- Inmadurez de las herramientas de desarrollo
  - Los desarrolladores que crean aplicaciones de bases de datos para la web identificaron rápidamente la inmadurez de las herramientas de desarrollo que estaban disponibles inicialmente. Hasta hace poco, la mayor parte del desarrollo de Internet utilizaba lenguajes de programación de primera generación con el entorno de desarrollo que consistía en poco más que un editor de texto. Este fue un inconveniente significativo para el desarrollo de Internet, particularmente porque los desarrolladores de aplicaciones ahora esperan entornos de desarrollo gráfico maduros. Se ha trabajado mucho en los últimos años para abordar este problema y los entornos de desarrollo se están volviendo mucho más maduros. Al mismo tiempo, hay muchas tecnologías en competencia y aún no está claro si estas tecnologías alcanzarán su potencial. Tampoco existen pautas reales sobre qué tecnología será mejor para una aplicación en particular. Aun no tenemos el nivel de experiencia con las aplicaciones de base de datos para la Web que tenemos con las aplicaciones no basadas en Web más tradicionales, aunque con el tiempo esta desventaja debería desaparecer.

#### Desventajas del enfoque Web-DBMS

Fiabilidad

Seguridad

Coste

Escalabilidad

Funcionalidad limitada de HTML

Sin estado

Ancho de banda

Rendimiento

Inmadurez de las herramientas de desarrollo

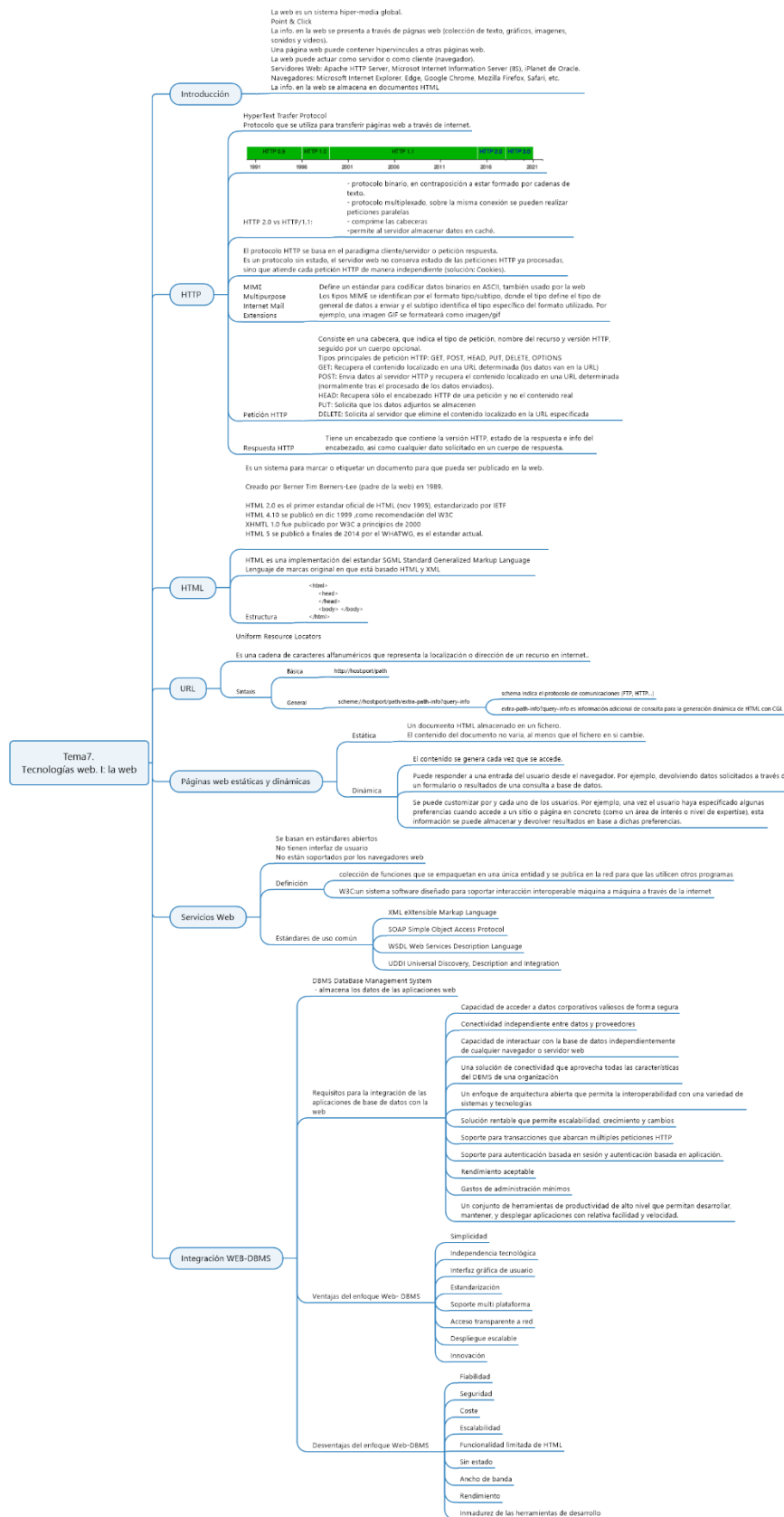
---

## 9 Enfoques para la integración de la web y los DBMSs

Los actuales enfoques para la integración de la web y los DBMSs son:

- Lenguajes de script como JavaScript y VBScript;
- Common Gateway Interface (CGI);
- Cookies HTTP;
- Extensiones a los servidores web, como Netscape API (NSAPI) y Microsoft Internet Information Server API (ISAPI);
- Java, J2EE, JDBC, SQLJ, JDO, Servlets y Java Server Pages (JSP);
- Plataforma de soluciones web de Microsoft: .NET, Active Server Pages (ASP) y ActiveX Data Objects (ADO);
- Plataforma de Internet de Oracle.

## 10 RESUMEN ESQUEMÁTICO



## 11 GLOSARIO

ASCII American Standard Code for Information Interchange

CERN Conseil Européen pour la Recherche Nucléaire, Organización Europea para la Investigación Nuclear (nombre oficial en español)

DBMS DataBase Management System

FTP File Transfer Protocol

HTML HyperText Markup Language

HTTP HyperText Transfer Protocol

IETF Internet Engineering Task Force

MIME Multipurpose Internet Mail Extensions

NNTP Network News Transfer Protocol

RFC Request for Comment. Es un tipo de documento que define estándares o proporciona información sobre varios temas.

SGML Standardized Generalized Markup Language

SOAP Simple Object Access Protocol

SVG scalable vector graphics

XHTML eXtensible HyperText Markup Language

XML eXtensible Markup Language

UDDI Universal Discovery, Description and Integration

URL Uniform Resource Locators

WSDL Web Services Description Language

WWW World Wide Web

W3C World Wide Web Consortium

## 12 BIBLIOGRAFÍA BÁSICA

T. Connolly and C. Begg. Database systems: a practical approach to design, implementation, and management (6th ed.). Capítulo 29. Sección 29.2.