



Desarrollo de Software

Programación web I Semestre 7

Unidad 1 Fundamentos web

**Clave
15144738**

Ciudad de México, febrero del 2025

Universidad Abierta y a Distancia de México





Unidad 1. Fundamentos web

Índice

Unidad 1. Fundamentos web	4
Presentación de la unidad.....	4
Logros.....	5
Competencia específica.....	5
1.1. Tecnologías web.....	5
Foro de dudas para el desarrollo de páginas web.....	10
1.1.1. Arquitecturas de la tecnología cliente-servidor	11
1.1.2. Web 1.0, 2.0, 3.0 y 4.0	18
1.1.3. Tipos y características de servidores web	35
1.2. Lenguajes de programación.....	41
1.2.1. Lenguaje HTML y CSS	46
1.2.2. Lenguaje PHP	61
1.2.3. Lenguaje Ruby On Rails	66
1.2.4. Lenguaje ASP	75
1.2.5. Código JavaScript.....	78
Cierre de la unidad.....	82
Para saber más	83
Fuentes de consulta.....	84



Unidad 1. Fundamentos web

Índice de Imágenes

Figura 1. Ejemplo de ventana de un navegador indicando sus componentes	8
Figura 2. Elementos principales de arquitectura tecnológica cliente-servidor. Fuente: Ocio y tecnología (2013). http://ociotec.com/tag/ssh/	12
Figura 3. Modelo cliente-servidor. Fuente: El tamiz (2014). http://eltamiz.com/elcedazo/wp-content/uploads/2010/06/3capas.gif	16
Figura 4. Ejemplo de página web estática. Fuente: Ejemplos de (2014). http://ejemplosde.info/ejemplos-de-paginas-web/	19
Figura 5. . Computadora NeXTcube usada por Tim Berners-Lee en el CERN primer servidor web. Fuente: Wikipedia, 2014.	22
Figura 6. Hipertexto. Fuente: Computadoras CMC (2012). http://computadorascmc.blogspot.mx/2012/06/Internet.html	23
Figura 7. Imagen que representa los anclajes. Fuente: Lamarca, 2013	23
Figura 8. Primera página web creada por Tim Berners-Lee en el CERN Ginebra, Suiza. Fuente: FayerWayer, 2012.	24
Figura 9. La Web 2.0. Fuente: Más que comunicación, 2014. http://www.masquecomunicacion.com/wp-content/uploads/2010/10/web2.jpg	25
Figura 10. Uso del hipertexto en la Web 2.0.	26
Figura 11. La nube de tags, Web 2.0. Fuente: http://cache.metaspacportal.com/49724.jpg	26
Figura 12. Web 3.0 (web semántica). Fuente: http://negraisa.files.wordpress.com/2012/07/nubedepalabraswed3-0.png	28
Figura 13. Ejemplo de resultados de búsqueda en un navegador de la Web 2.0. Fuente: W3C, 2014. http://www.w3c.es/Divulgacion/GuiasBreves/WebSemantica	29
Figura 14. Ejemplo de resultados de búsqueda en un navegador de la Web 3.0. Fuente: W3C, 2014. http://www.w3c.es/Divulgacion/GuiasBreves/WebSemantica	30
Figura 15. Evolución de la Web en relación con el uso del hipertexto. Basada en: Lamarca, 2013.....	31
Figura 16. Elementos que forman una página web	32
Figura 17. La web 4.0 (La web ubicua)	34
Figura 18. Evolución de la tecnología web	35
Figura 19. Servidor web Apache. Fuente: Asolif, 2012. http://www.asolif.es/sites/default/files/foto2_ApacheServer.jpg	38
Figura 20. Servidor Web IIS. Fuente: http://sergiot2.com/blogimages/2008/05May/18_ServerManager.jpg	39
Figura 21. Estadística de servidores web más utilizados año 2013.Basado en: Netcraft. 41	
Figura 22. Pasos para ver el código fuente de una página web	43
Figura 23. Resultado de ver el código fuente de una página web	44
Figura 24. Ejemplo del código lenguaje de programación HTML	47
Figura 25. Resultado del uso del código del lenguaje de programación web HTML	48
Figura 26. Ejemplo.....	50



Unidad 1. Fundamentos web

Figura 27. Ejemplo.....	51
Figura 28. Ejemplo.....	53
Figura 29. Ejemplo.....	55
Figura 30. Ejemplo.....	57
Figura 31. Ejemplo.....	59
Figura 32. Ejemplo.....	60
Figura 33. Ejemplo de aplicación del lenguaje de programación web PHP	62
Figura 34. Logotipo del lenguaje de programación PHP. Fuente: http://www.nuevoiris.com/img/a0/naTexto5_PHP.png	63
Figura 35. Script PHP.Fuente: http://www.sendblaster.com/sendblaster/release2/sites/default/files/image/en/ support/screenshot6.gif	63
Figura 36. Código PHP conexión de base de datos	65
Figura 37. Ejemplo.....	68
Figura 38. Ejemplo.....	69
Figura 39. Ejemplo.....	69
Figura 40. Ejemplo.....	70
Figura 41. Ejemplo.....	71
Figura 42. Ejemplo.....	72
Figura 43. Ejemplo.....	73
Figura 44. Ejemplo.....	73
Figura 45. Ejemplo.....	74
Figura 46. Ejemplo.....	79
Figura 47. Ejemplo.....	79
Figura 48. Ejemplo.....	75
Figura 49. Ejemplo.....	75
Figura 50. Ejemplo de código lenguaje ASP	77
Figura 51. ASP. Fuente: http://www.genbeta.com/mobile.php/web/crea-plantillas-para- wordpress-desde-cero-con-cool-template	77
Figura 52. Ejemplo de código JavaScript. Fuente: Minera, 2011.....	79
Figura 53. Resultado de la utilización del código JavaScript	80
Figura 54. Logotipo del lenguaje de programación JavaScript. Fuente: http://www.planet- source-code.com/vb/2010Redesign/images/LangugeHomePages/JavaScript.png	80
Figura 55. Calendario con JavaScript	81



Unidad 1. Fundamentos web

Unidad 1. Fundamentos web

Presentación de la unidad

En la actualidad, muchas actividades que realizan las organizaciones tales como publicidad, educación, difusión de información, etcétera, están soportadas en la Web, debido a que los sitios son cada vez más amigables con el usuario al proporcionarle la posibilidad de generar contenidos de forma sencilla, proveer de información en diferentes dispositivos electrónicos, y permitir las intercomunicaciones a través de diferentes soportes tecnológicos.

En relación con lo anterior, y como ingeniero en desarrollo de software, necesitas saber cuál es el impacto y lugar que tienen las aplicaciones web en el contexto social actual. Para ello, en el primer tema de esta unidad estudiarás los conceptos básicos relacionados con la programación web. Iniciarás con un recorrido por la evolución que han tenido estas tecnologías y las características generales de la Web 1.0, 2.0, 3.0 y 4.0, así como de la arquitectura de la tecnología cliente-servidor y sus usos en Internet. Por último, revisarás el panorama general acerca del uso a nivel mundial de los diversos servidores web.

En el segundo tema, *1.2. Lenguajes de programación*, revisarás los fundamentos acerca de la programación web, incluyendo los lenguajes de programación que hicieron posible el uso de las tecnologías web. La importancia de estos lenguajes radica en que son la base para llevar a cabo la construcción de sitios web y, como sabes, su uso es generalizado a nivel mundial.

Cabe mencionar que cada lenguaje de programación web tiene funcionamientos diferentes; por ese motivo, en el desarrollo de los contenidos se diferenciará qué lenguajes de programación permiten llevar una administración de información, particularmente las conexiones a bases de datos. Aprendizaje que te permitirá, al concluir la unidad didáctica, implementar un sitio web.



Unidad 1. Fundamentos web

Logros

Al término de esta unidad lograrás:

- Identificar los elementos básicos de las tecnologías web.
- Identificar los tipos de servidores web de acuerdo al tipo de sistema operativo.
- Identificar los tipos de lenguajes de programación.
- Relacionar los tipos de página web que integran un sitio web con la aplicación de los lenguajes de programación.

Competencia específica

- Analizar sitios web para determinar la pertinencia de su construcción mediante la identificación de las tecnologías web y los lenguajes de programación.

1.1. Tecnologías web

En el mundo actual, las tecnologías web se han vuelto una necesidad para la competitividad en las organizaciones pues permiten ahorrar tiempo, agilizan la comunicación, acceder a sitios recreativos, etcétera; por ejemplo, mediante esta tecnología es posible realizar pagos sin necesidad de ir al banco o compras de todo tipo, desde un juguete hasta una casa, todo en forma remota. Por otro lado, las organizaciones, por medio de estas tecnologías web, cuentan con la facilidad del manejo de información en relación con sus actividades productivas, lo que les permite recibir, enviar y monitorear información, tener control de acceso remoto, entre otras posibilidades. Esto tanto a nivel mundial como internacional.



Unidad 1. Fundamentos web

Es de suma importancia ubicar el contexto en el que te desenvolverás en esta unidad didáctica y el cómo han evolucionado las tecnologías web, para así comprender su importancia respecto de los beneficios que representan en diversos ámbitos de la sociedad, pues "la utilización de tecnologías web permite agilizar los procesos, mejorar la productividad y aumentar la eficacia, además de abrir las puertas a nuevas formas de negocio en el mercado global que facilita Internet(*e-business*)"(Cobo, Gómez, Pérez y Rocha, 2005, p. 1).

Las tecnologías web básicas son, según Lamarca (2013):

- **El modelo de la World Wide Web**, basado en el de cliente-servidor, el cual se conforma de los siguientes elementos:
- **Cliente**: en una red, es el nodo que utiliza los recursos que proporciona un servidor. Este nodo cliente es un equipo de cómputo, es decir, puede ser una PC, una laptop, una agenda electrónica o un teléfono móvil (Maciá, et ál., 2008). Debe contar con un sistema operativo, así como con la capacidad para establecer conexión a una red. Este nodo debe contar, como mínimo, con la llamada *aplicación cliente*, por medio de la cual sea posible solicitar el recurso o servicio deseado y, en caso de ser interactiva, realizar estas acciones de interactividad mediante una interfaz con el usuario.
- **Servidor**: es un nodo de red que proporciona servicios a los nodos que funcionan como clientes; por ejemplo, el acceso a la información contenida en archivos, formación de trabajos de impresión o ejecución remota. Los servidores se conforman también por equipos de cómputo, y cuentan con un programa llamado programa servidor
- **Nodo**: cada uno de los equipos de cómputo u otros dispositivos conectados a la red, que funcionan como cliente o servidor.



Unidad 1. Fundamentos web

- **Paquete:** grupo de bits de datos de información asociada, incluida la dirección de origen y de destino, formateado para transmitirse de un nodo a otro.
- **Ruteador:** dispositivo por medio del cual se hace posible la conexión entre dos redes. Opera como un puente y es posible, por medio de este dispositivo, seleccionar rutas en la red.

Otra tecnología la constituyen los **protocolos web**. Por medio de ellos, es posible recibir y enviar información. Se conforma de una serie de estándares de comunicación entre los dispositivos de una red que proveen los mecanismos de acceso e intercambio de información de los sitios web.

Los navegadores web son otra de las tecnologías. Son el software cliente que permite la visualización de los sitios web, así como seguir los enlaces. En la actualidad, los navegadores web más utilizados tienen una interfaz basada en una ventana con barras de desplazamiento, en la que se visualiza la información de las páginas web.

Unidad 1. Fundamentos web

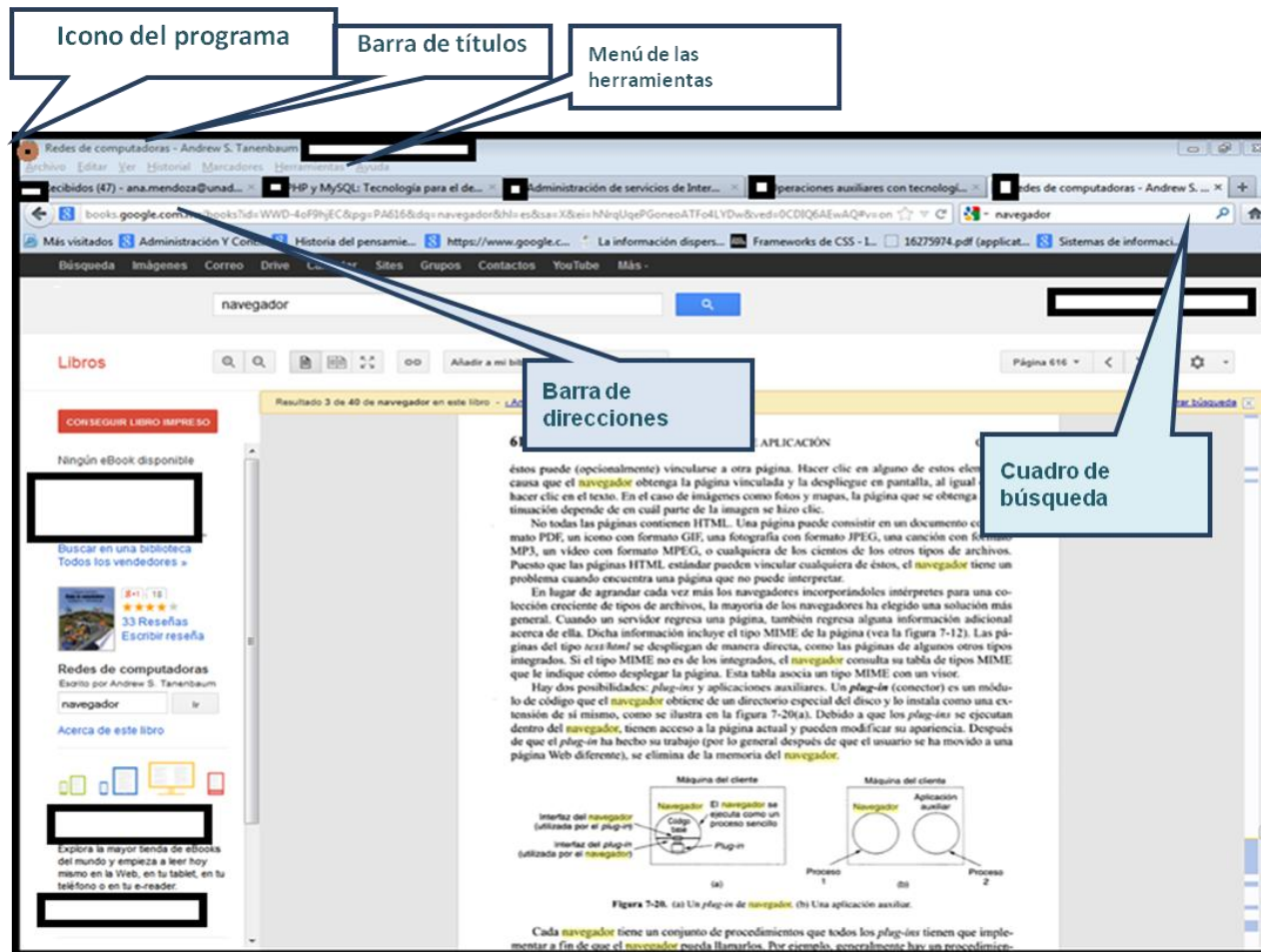


Figura 1. Ejemplo de ventana de un navegador indicando sus componentes



Unidad 1. Fundamentos web

En la imagen se muestran las partes de un navegador web:

1. Barra de títulos. Ahí se identifica el nombre del sitio web en que se está navegando y el icono del navegador web utilizado.
2. Menú de herramientas. Cuenta con tres zonas: barra de favorito, barra de comando y control de pestañas.
3. Barra de favoritos. Permite guardar el sitio web que más gusta y no estar tecleando la dirección URL.
4. Barra de comando. Permite que se pueda imprimir el sitio web, y que se puedan bloquear sitios no seguros.
5. Barra de direcciones. Permite teclear la URL del sitio web que se desea visualizar.
6. Cuadro de búsqueda. Permite encontrar el sitio web de una organización con tan sólo teclear el nombre.
7. Control de pestañas. Permite ver todos los sitios web que están abiertos (Rodil, 2010).

En el primer subtema, 1.1.1. Arquitecturas de la tecnología cliente-servidores, se revisarán aspectos históricos respecto a los inicios de Internet, ya que es una red mundial que permite que se reciba y envíe información, en la que se basan las tecnologías web y cuya estructura y funcionamiento se sustenta en la arquitectura cliente-servidor. La diferencia de un servidor y un cliente es que el primero administra y permite que haya una comunicación, mientras el segundo visualiza lo que el servidor solicita, es decir, envió de bases de datos. Por esto se realiza una arquitectura cliente-servidor.

En el segundo subtema, 1.1.2. Web 1.0, 2.0, 3.0 y 4.0, se revisarán las etapas de la Web que se fueron generando y adecuando a las necesidades de los usuarios en



Unidad 1. Fundamentos web

general. La primera versión nombrada Web 1.0 permitió clasificar diferentes tipos de información. Es conocida como WWW, que significa mundo de telaraña web.

Después apareció la Web 2.0 (Codina, 2009), que realiza aplicaciones interactivas.

La Web 3.0 es un salto o evolución, ésta se puede ejecutar en cualquier dispositivo, tal como un celular (Codina, 2009) en la actualidad se cuenta con la web 4.0 también llamada la web ubicua y propone una nueva manera de interacción con el usuario más completa y personalizada, no limitándose simplemente a mostrar información, sino comportándose como un espejo mágico que dé soluciones concretas a las necesidades del usuario.

En el tercer subtema, 1.1.3. Tipos y características de servidores web, se revisará lo que es un servidor web pues, por medio de él, se administran los sitios y se realiza la conexión con bases de datos. Existen diferentes servidores web, pero los más conocidos son IIS y Apache, la diferencia entre ellos es que en el primero sólo se puede trabajar con el sistema operativo Windows, mientras que en el segundo con todos los existentes (Vara, *et ál.*, 2012).

A partir del estudio de los subtemas podrás darte cuenta de cómo se ha extendido el uso de las tecnologías web en muy diversos aspectos. Analizarás particularmente el ámbito organizacional, pues en éste se inserta tu práctica profesional que consiste en crear sitios web que resuelvan necesidades específicas de la administración de la información para las organizaciones.

Foro de dudas para el desarrollo de páginas web

Esta actividad no ponderable se realizará en un *Foro general* que estará disponible en todas las unidades de la unidad didáctica, en él podrás:



Unidad 1. Fundamentos web

- Compartir tus expectativas y metas acerca de la unidad didáctica.
- Exponer tus dudas sobre la creación de tu sitio web, y resolverlas de manera colaborativa.

Para comenzar, **ingresa** al foro y **sigue** las indicaciones de la Figura académica.

Recomendaciones:

- **Dirígete** con respeto a todos tus compañeros(as) y Figura académica.
- **Sigue** las indicaciones de la actividad.
- **Realiza** críticas constructivas a las aportaciones de tus compañeros(as).
- **Participa** activamente en las discusiones.

1.1.1. Arquitecturas de la tecnología cliente-servidor

En el ámbito de la programación web, los elementos que conforman las tecnologías web tienen una estructura lógica basada en la arquitectura cliente-servidor, la cual desempeña una función muy importante ya que si ésta no existe la administración de un sitio web no se podrá operar.

Las arquitecturas de las tecnologías cliente-servidor se refieren a la creación de aplicaciones para la operación de un sitio web, la principal ventaja de estas arquitecturas radica en que facilita la separación de las funciones de cada sitio web en la plataforma más adecuada para su ejecución (Luján, 2002).

Para que un sitio pueda ser visualizado en Internet debe tener una arquitectura tecnológica cliente-servidor. Según Marcía (2008), en la siguiente imagen se muestran los principales elementos que intervienen en dicha arquitectura tecnológica:



Unidad 1. Fundamentos web

- **Cliente:** está formado por el equipo computacional, ya sea una PC, portátil, agendas electrónicas o teléfonos móviles. Son sus respectivos sistemas operativos.
- **Servidor:** se trata con el equipo de la combinación del hardware de los sistemas operativos. El servidor deberá estar conectado en la red junto con el cliente.

De tal forma, el cliente y el servidor deberán estar comunicados en una red y emplear los protocolos HTTP.

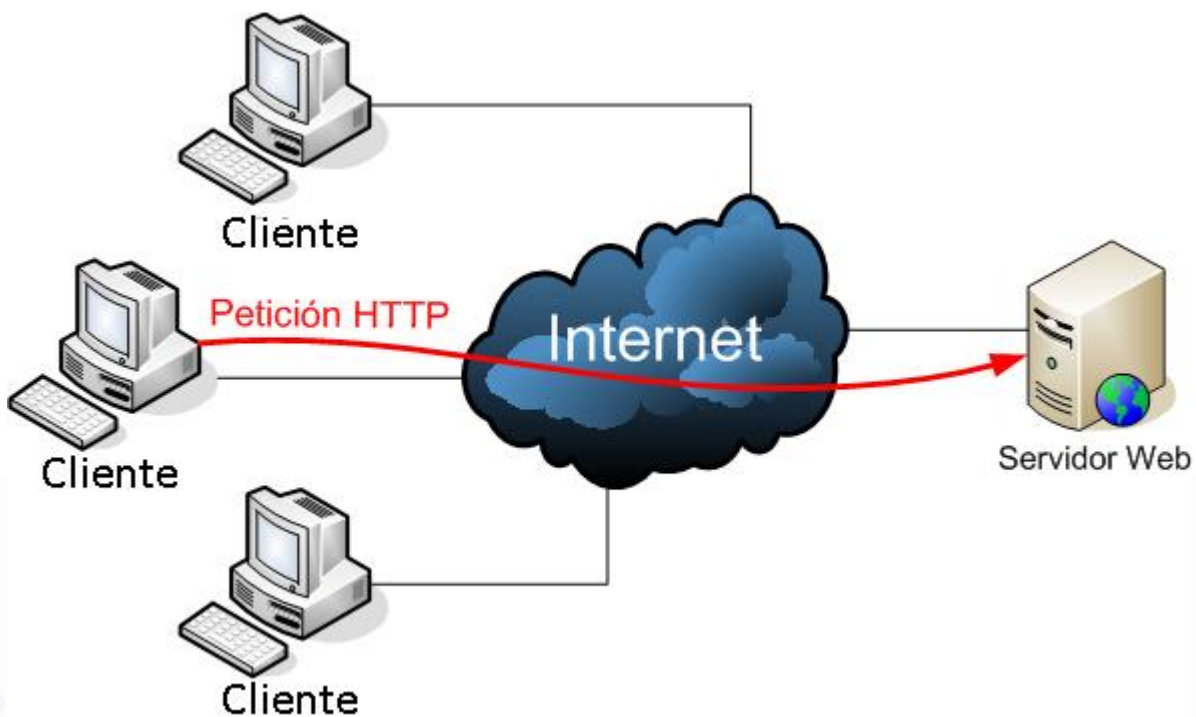


Figura 2. Elementos principales de arquitectura tecnológica cliente-servidor. Fuente: Ocio y tecnología (2013). <http://ociotec.com/tag/ssh/>

Las tecnologías web han evolucionado desde su aparición en los años 90, cuando revolucionaron el campo de la informática y de las comunicaciones. Con este



Unidad 1. Fundamentos web

servicio fue posible abrir el uso de Internet en diversos ámbitos, no únicamente el militar, como originalmente fue pensado.

Para navegar en sitios web es necesario contar con infinidad de arquitecturas de cliente-servidor en diferentes tipos de organizaciones. Por tal motivo, se hace necesario conocer qué es Internet y cómo ha evolucionado.

Internet es una red mundial que permite la comunicación y acceso a la información de todo tipo. Al respecto, Vázquez (2006) menciona que en el año 1969 se creó la primera conexión de varias computadoras en Estados Unidos. Tenía el propósito de mantener comunicación secreta, ya que en esa época el país estaba en guerra. Después se llevó a la universidad de California para que se ampliara el proyecto con el nombre de ARPANET, donde se llegó a la conclusión de que se podía tener comunicación a nivel mundial. Es en este momento cuando cambia el nombre a INTERNET, ya que permite que haya comunicación con infinidad de computadoras por medio satelital. Para llegar a esto, entre sus características, primero tuvo un protocolo de transmisión TCP/IP.

Para que fuera posible la visualización de los sitios web en Internet, se hicieron necesarios los protocolos; es decir, un procedimiento destinado a estandarizaciones (Pastor, 2011).

¿Cuáles son esos protocolos? TCP/IP y HTTP. Ya los revisaste en la unidad didáctica Fundamentos de redes. Fueron creados en 1978 y se hizo posible realizar transmisión de información y comunicarse mediante la red de redes. TCP/IP son protocolos que permiten el control de la transmisión de Internet. Funcionan juntos. Son fundamentales para la comunicación del servidor y el cliente. El protocolo de transferencia de hipertexto HTTP permite el procesamiento de información realizada en cada operación de la Word Wide Web (WWW), que significa mundo de telaraña, según Vara (2012).



Unidad 1. Fundamentos web

Se puede definir al servicio web como un “amplio sistema multimedia de acceso a información heterogénea distribuida por toda la red en forma de documentos hipertextuales (hipertextos) [surge en] 1990 en la Organización Europea para la Investigación Nuclear CERN (por sus siglas en *inglés* *Centre Européen de Recherche Nucléaire*)” (Cobo, Gómez, Pérez y Rocha, 2005, p.4). Éste es un organismo que surge en el año de 1954 en la frontera franco-suiza, cerca de Ginebra (CERN, 2014). En dicho centro se desarrolló un sistema de información que facilitara la distribución de información entre equipos de investigadores que se encontraban en distintos lugares geográficos, para que los recursos en formato digital estuvieran accesibles para cada investigador; es decir, un sistema donde fuera posible “integrar todos los recursos existentes en una red hipertextual” (Cobo, Gómez, Pérez y Rocha, 2005, p.4). En 1991, el sistema desarrollado en el CERN se integra a Internet. En 1992 aparece el primer navegador web llamado Mosaic.

El concepto de hipertexto es característico de las tecnologías web. Se puede definir como “un documento multimedia [que integra] bajo una plataforma informática todas las tecnologías de la información y que incorpora relaciones estructurales que enlazan el documento con otros documentos o recursos” (Cobo, Gómez, Pérez y Rocha, 2005, p.4). Este concepto se considera de primordial importancia para comprender por qué la Web se convirtió en una parte fundamental en la forma de comunicación, a nivel mundial, en todos los ámbitos de acción de la sociedad.

Con la integración del hipertexto es posible:

- Almacenar grandes volúmenes de información. Esto se refiere a las bases de datos, ya que en un sitio web existe una gran cantidad de ellas, las cuales almacenan gran cantidad de información.
- Facilitar el acceso y consulta de textos en formato electrónico. Esto ayuda a que los usuarios puedan mandar a tiempo información, tales como estados de cuenta, facturas, recibos de compra y venta, etcétera.



Unidad 1. Fundamentos web

- Presentar los recursos en forma más accesible.
- Utilizar todas las tecnologías de información. Se refiere a que no nada más se puede emplear una computadora si no también celulares y tabletas. Esto permite, por ejemplo, identificar dónde te encuentras ubicado.
- Permitir una navegación individualizada. Pueden estar varios usuarios en un mismo sitio web y navegar sin ningún problema.
- Una estructuración multidimensional
- La multiplataforma
- El dinamismo y la interactividad.

Respecto a la evolución del Internet y los protocolos, es importante señalar que para posibilitar la navegación de un sitio web se necesita comprender la tecnología denominada **modelo cliente-servidor**.

Tal como lo expresa López 2011):

El **modelo cliente-servidor** proporciona una forma eficiente de utilizar la forma en que se construirá un sitio web, que la seguridad y fiabilidad que proporcionan los entornos mainframe, ¿Qué es una mainframe? Es una computadora grande, potente y tiene la capacidad de almacenar datos en una gran cantidad sin límites, el mainframe en el entorno de la programación web es muy importante ya que permite almacenar varios sitios web esto se traspasa a todas las redes en general. A esto hay que añadir la ventaja de la potencia y simplicidad de los ordenadores personales (p. 3).

En el mundo de TCP/IP, las comunicaciones entre computadoras se rigen básicamente por este modelo, ya que provee usabilidad, flexibilidad, interoperabilidad y escalabilidad.

El término modelo cliente-servidor se utiliza para la formación y levantamiento de los sitios web. Esto permite que el sitio esté en un ordenador local, lo cual hace que se ponga en comunicación con el ordenador remoto para solicitar el servicio deseado, y



Unidad 1. Fundamentos web

así ser administrado por el servidor. En conclusión, el modelo cliente-servidor se encarga de cumplir las siguientes funciones:

- **Visualización del sitio web:** se refiere al acceso a todas las páginas donde es posible observar los elementos que las integran, tales como imágenes, videos, música, texto, etcétera.
- **Almacenamiento de bases de datos:** en el modelo de cliente-servidor, quien almacena la base de datos es el servidor, mientras el cliente solo envía las bases de datos.
- **Administración del sitio web:** conjunto de acciones que se realizan para el buen funcionamiento, tales como integrar actividades de renovación del sitio. El servidor se encarga de administrar el sitio web pero el cliente no tiene permiso para ello, solo envía datos y procesa el sitio con la información que el servidor administra (Vara, 2012).

A continuación, se muestra un ejemplo del modelo cliente-servidor:

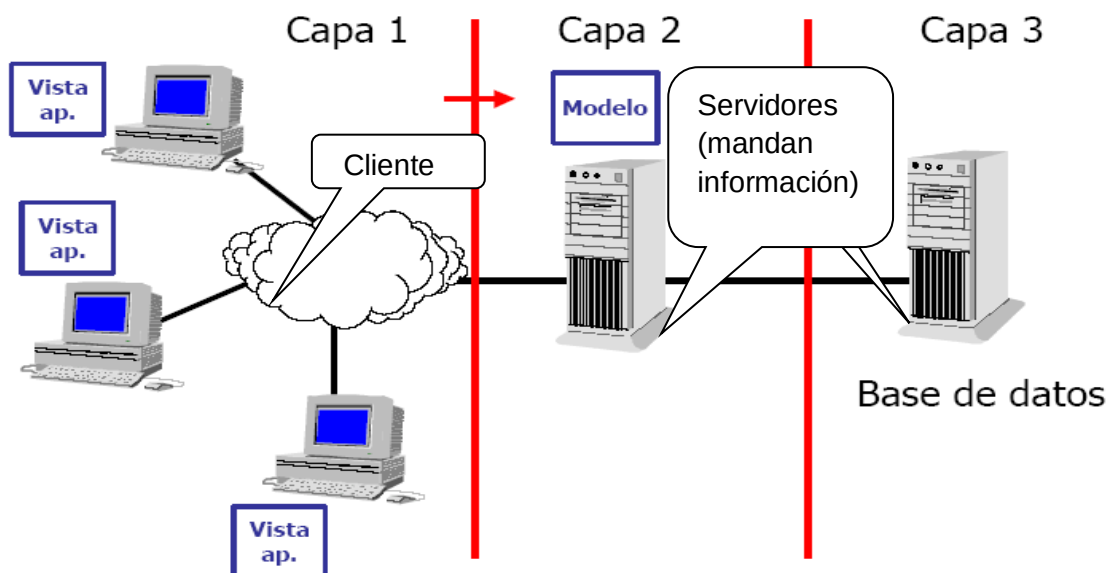


Figura 3. Modelo cliente-servidor. Fuente: El tamiz (2014). <http://eltamiz.com/elcedazo/wp-content/uploads/2010/06/3capas.gif>



Unidad 1. Fundamentos web

En el esquema anterior se observa que hay dos servidores conectados. Uno se encarga del proceso y almacenamiento de bases de datos, el otro de que se visualicen en los equipos que funcionan como clientes (consulta). Los elementos que conforman este modelo son:

Servidores: ordenadores que ofrecen sus servicios al resto de equipos conectados. Suelen tener una presencia estable en la red, lo que se concreta en tener asignadas direcciones IP permanentes. En ellos es donde están alojadas, por ejemplo, las páginas web.

Clientes: equipos que los usuarios individuales utilizan para conectarse a la red y solicitar servicios. Durante el tiempo de conexión tienen presencia física en la red. Normalmente, los proveedores de acceso a Internet asignan a estos equipos una dirección IP durante su conexión, pero es variable; es decir, cambia de unas conexiones a otras (IP dinámica).

Los conceptos de cliente y servidor se suelen utilizar con dos significados diferentes. Con referencia al hardware, el sentido es el indicado anteriormente. El servidor hace referencia al equipo remoto con el que se realiza la conexión, y el cliente sería el equipo local utilizado para efectuarla.

Funciones de un servidor:

- **Mantener la petición de ejecución** que le harán los usuarios de Internet o Intranet.
- **Contestar lo solicitado** de acuerdo con las peticiones de los usuarios de Internet o Intranet, de forma adecuada, entregando como resultado páginas web.
- **Cliente:** se encarga de la **interacción** con el usuario para solicitar a un servidor el envío de los recursos que se desea obtener mediante HTTP (Luján, 2001).



Unidad 1. Fundamentos web

Función de un cliente:

- **Establecer** conexión con el servidor.
- **Interpretar** las páginas web y los diferentes recursos que contiene (imágenes, sonidos, video, etcétera).

Es muy importante comprender en qué consiste la arquitectura cliente-servidor para saber utilizar las tecnologías de programación web, ya que te permitirá identificar los requerimientos. Un ejemplo de éstos puede ser cuántas bases de datos se van a procesar, ya que todo sitio web tiene funciones y aplicaciones diferentes, ya sea en Internet o en Intranet (en ésta sólo se visualiza la información en una organización, nadie más podrá hacerlo fuera de ella, también se basa en una arquitectura cliente-servidor).

1.1.2. Web 1.0, 2.0, 3.0 y 4.0

Las tecnologías web han ido evolucionando conforme las necesidades de los usuarios se han integrado en las tecnologías web. Empezaron a tener suma importancia porque facilitan y ayudan en la realización de infinidad de actividades en forma remota, tales como realizar compras en línea, tener comunicación y visualizar a otra persona en forma sincrónica, mediante las charlas virtuales en una herramienta mejor conocida como chat. Por medio de la Web, las organizaciones pueden proyectarse a nivel mundial expandiendo sus mercados, aumentando su competitividad y crecimiento económico. En un inicio, cuando surgieron estas tecnologías web, no se contaban con algunos recursos tales como el chat o las video llamadas; es decir, había una interactividad muy limitada; no se podían realizar compras vía electrónica como en la actualidad. Ante ello, se integraron nuevas tecnologías que hicieron posible una evolución en la programación web. Es necesario identificar que los lenguajes de programación web son un punto principal



Unidad 1. Fundamentos web

en esta evolución, pues por medio de ellos se logra esta interactividad propia de la Web que se observa en los albores del siglo XXI.

El servicio Web se concibió como un “sistema flexible de compartir información multimedia entre equipos heterogéneos a través de las redes informáticas” (Cobo, Gómez, Pérez y Rocha, 2005, p. 2), desarrollándose un sistema de generación de documentos mediante un lenguaje estándar, el HTML. De esta forma se pudo incorporar texto y elementos gráficos en los documentos; sin embargo, estos documentos eran estáticos, es decir, que “siempre muestran la misma información y no ofrecen ningún grado de interactividad con el usuario” (*Ibídem*). A continuación, se mostrará un ejemplo de una página web estática:



Figura 4. Ejemplo de página web estática. Fuente: Ejemplos de (2014).
<http://ejemplosde.info/ejemplos-de-paginas-web/>



Unidad 1. Fundamentos web

“Los únicos elementos de HTML que podrían de alguna forma considerarse interactivos son los formularios a través de los cuales se solicita información al usuario” (Cobo, Gómez, Pérez y Rocha, 2005, p. 8).

Posteriormente, se desarrollaron tecnologías que enriquecieron el lenguaje HTML “Netscape incorporó en la versión 2.0 de su navegador un **intérprete** de un nuevo lenguaje que podía ser intercalado entre el código HTML y que permitía realizar operaciones no disponibles en un lenguaje puramente descriptivo como es HTML; dicho lenguaje, llamado en sus orígenes **LiveScript**, adquirió posteriormente el nombre de JavaScript” (Cobo, Gómez, Pérez y Rocha, 2005, p. 2). Años después, la empresa Microsystems desarrolló el lenguaje de programación Java, esto hizo posible incrustar programas en las páginas web con las prestaciones propias de cualquier lenguaje de programación.

Un navegador (*browser*) es “un programa de ordenador utilizado para ver páginas web e interactuar con ellas” (*Diccionario web*). Por medio de los navegadores es posible localizar algún sitio en Internet.

Para poder entrar a la Web se necesita contar con conexión a **Internet** y un **navegador**. Estos elementos son las principales tecnologías web. Internet es una red de comunicación mundial y su relación con la Web es esta red Internet mediante tecnologías web, lo que permite la comunicación a nivel mundial y la interacción con varios sitios web (López, 2011).

Un navegador es un “software informático que permite la comunicación con un servidor para acceder a los sitios web” (Alonso, 2007). Internet y navegadores son dos tecnologías web que permiten que se pueda localizar información en la Web.

La Web ha pasado por tres etapas que se consideran como evoluciones, porque se han integrado elementos y recursos que enriquecen sus servicios. Es necesario revisar esta evolución porque el desarrollo de software ha jugado un papel primordial, los conceptos de cliente y servidor que forman parte del modelo en el que



Unidad 1. Fundamentos web

se basa la Web. Estos se suelen utilizar con dos significados, en referencia al hardware el servidor se refiere al equipo remoto al que se realiza la conexión. El cliente es el equipo local utilizado para efectuar dicha conexión. Pero también se utilizan esos conceptos en referencia al software:

Programa servidor: debe estar ejecutándose en el equipo servidor para que pueda ofrecer su servicio. Un documento HTML se encuentra almacenado en un equipo remoto, no basta para que sea accesible como página web por el resto de usuarios de Internet. En ese equipo debe estar ejecutándose una aplicación servidor web.

Uno de los programas servidores web más conocido y utilizados es Apache, (Cobo, Gómez, Pérez y Rocha, 2005, p. 2). En el caso de servicios como el correo electrónico o la transferencia de ficheros, se necesitarían igualmente los correspondientes programas en el servidor; ante ello, se puede observar que el desarrollo de software ha jugado un papel principal en esta evolución de la Web.

A continuación, se exponen las etapas de dicha evolución.

Web 1.0. Se publicó el 6 de agosto de 1991(Nafría, 2008, p.23). Se caracterizó por ser una Web estática, conformada por los siguientes elementos: información e interconexión. Surgió en el año 1989 en Ginebra, Suiza. Era de tipo estático, es decir, no permitía ser actualizada y los contenidos se desarrollaron con el lenguaje de programación HTML (Pastor, 2011).



Unidad 1. Fundamentos web



Figura 5. . Computadora NeXTcube usada por Tim Berners-Lee en el CERN primer servidor web.

Fuente: Wikipedia, 2014.

Creada en 1990 por Tim Berners-Lee, físico británico que colaboraba en el CERN (Nafría, 2008, p.24) en Ginebra, Suiza, la primera página web se componía de información explicando qué era la World Wide Web, cómo se podía disponer de un navegador y cómo se debía preparar un servidor web para que funcionara. Se convirtió en el primer directorio de sitios web del mundo.

Crea Tim Berners-Lee el primer servidor web y el primer cliente web. El objetivo era compartir información entre los investigadores de forma sencilla y “permitir la revisión de referencias que existiesen en el documento original mientras este se leía y siempre y cuando el usuario quisiera hacerlo”, (Márquez, 2007, p. 10). Es esta la primera referencia de aplicación real del concepto de hipertexto.

El hipertexto consta de los siguientes elementos:

- **Nodos o secciones:** son las partes del texto que contiene información accesible para un usuario.
- **Enlaces o hipervínculos:** representan la comunicación que se establece entre los nodos (computadora), y permiten que el usuario realice una lectura secuencial, o no, de los mismos.
- **Anclajes:** son los puntos de activación de los enlaces.



EL HIPERTEXTO

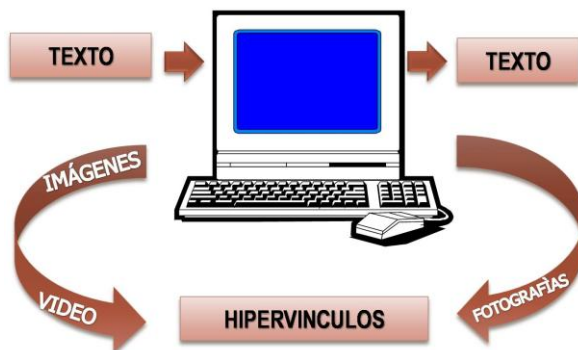


Figura 6. Hipertexto. Fuente: Computadoras CMC (2012).
<http://computadorascmc.blogspot.mx/2012/06/Internet.html>

El hipertexto recibe el texto que se observa en la pantalla de un dispositivo electrónico (computadoras, tabletas, teléfonos celulares, etcétera). También recibe imágenes, videos y sonido. Sin embargo, en la primera versión Web, esto aún no se hacía posible. El software que permite leer los documentos de hipertexto es el navegador o cliente, y cuando se ingresa a un enlace se puede decir que se está navegando por la Web.

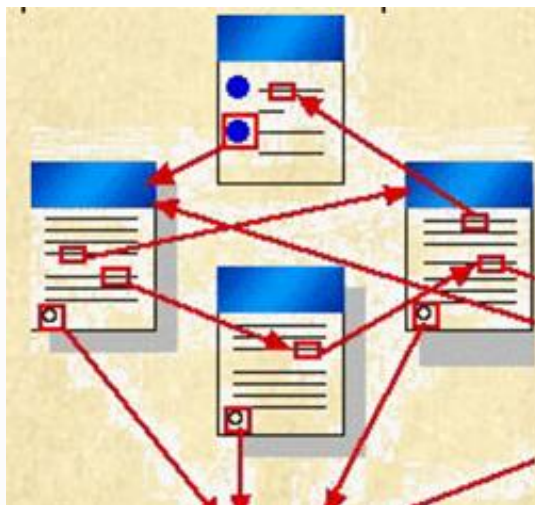


Figura 7. Imagen que representa los anclajes. Fuente: Lamarca, 2013



Unidad 1. Fundamentos web

HTML es un lenguaje puramente descriptivo que permite definir las páginas web, pero por medio de él no es posible generar estructuras *iterativas* o *condicionales*. Por medio de HTML, no es posible definir funciones que sean utilizadas en distintos puntos del documento, no se pueden crear variables, no se pueden realizar cálculos matemáticos. Esta es una de las características principales de la Web 1.0.

En las primeras páginas se explicaba qué era la World Wide Web, cómo se podía disponer de un navegador, y cómo se debía preparar un servidor web para que funcionara, lo cual se convirtió en el primer directorio de sitios web del mundo.

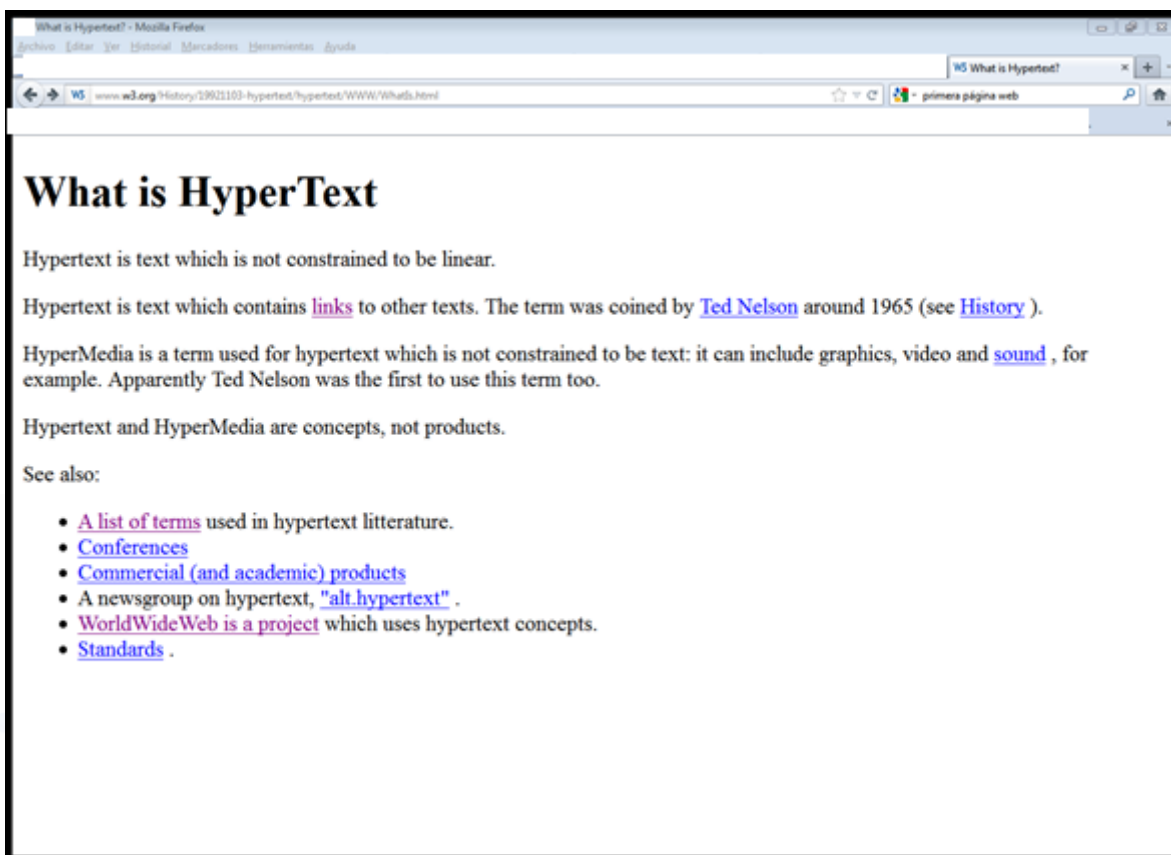


Figura 8. Primera página web creada por Tim Berners-Lee en el CERN Ginebra, Suiza. Fuente: FayerWayer, 2012.

Web 2.0. Debido al gran éxito que tuvo la Web 1.0, en la empresa editorial norteamericana, surgió una tecnología web renovada. La Web 2.0, o también



Unidad 1. Fundamentos web

conocida como la Web **social**, se creó con la llegada de las nuevas tecnologías y aplicaciones interactivas que facilitaban la publicación y la edición del contenido de cada sitio web, así como la creación de redes personales. “La Web 2.0, es una serie de aplicaciones y páginas de Internet que utilizan la inteligencia colectiva para proporcionar servicios interactivos en red dando al usuario el control de sus datos. Algunas aplicaciones más representativas son los blogs y las wikis” (Márquez, 2007, p.62).



Figura 9. La Web 2.0. Fuente: Más que comunicación, 2014.<http://www.masquecomunicacion.com/wp-content/uploads/2010/10/web2.jpg>

Marín (2010) señala que:

La Web 2.0 es la Web que se utiliza en la actualidad, la cual integra recursos de video, permite el acceso e integración de estos recursos, existen grupos de discusión, las llamadas comunidades virtuales, foros de discusión, espacios para integrar puntos de vista y para integrar en forma colaborativa, información de temas de interés. La Web 2.0 está compuesta por las herramientas y servicios susceptibles de utilizar en los más diversos ámbitos de la vida cotidiana, laboral, profesional, etcétera (p. 62).



Figura 10. Uso del hipertexto en la Web 2.0.

Esta web se volvió un fenómeno social ya que permite trabajar con diferentes plataformas, desde las que facilitan la gestión de las actividades de enseñanza-aprendizaje hasta la conformación de verdaderas comunidades virtuales. En conclusión, es posible decir que la Web 2.0 tiene más actitud y modelo de negocio. Se centra en la comunicación. Esta Web se visualiza en diferentes dispositivos, tales como celulares, tabletas, etcétera (Pastor, 2011).



Figura 11. La nube de tags, Web 2.0. Fuente: <http://cache.metaspacesportal.com/49724.jpg>



Unidad 1. Fundamentos web

“La Web 2.0 se involucra en una gran cantidad de conceptos tecnológicos, comportamientos sociales, e incluso tendencias económicas, como se representa en la imagen la nube de tags es uno de los iconos más populares para representar lo que significa la Web 2.0” (Marín, 2010, p. 60). La característica de la Web 2.0 es que permite una comunicación recíproca entre emisor y receptor. Los papeles pueden intercambiarse, es decir, el receptor puede convertirse en emisor y viceversa (Asensio, 2011, p. 99).

Web 3.0. Mediante ésta se realiza comunicación e interactividad, pero se basa más en el mundo de la mercadotecnia. Permite que el usuario pueda hacer compras y realizar pagos de todo tipo; aspecto que facilita mucho al usuario y las organizaciones realizar actividades desde distintas partes del mundo y en cualquier horario (Pastor, 2011).

Como menciona Criado, “la Web 3.0 (denominación utilizada por primera vez en el 2006 por Jeffrey Zeldman) es el siguiente paso evolutivo y añade, a las capacidades anteriores, entre otras cosas, contenido semántico para los sistemas informáticos. Por lo tanto, el concepto de la Web 3.0 contiene el concepto de la Web semántica que es parte de la Web 3.0.” (2013, p. 4). Como semántica se entiende lo que es “perteneciente o relativo a la significación de las palabras. Estudio del significado de los signos lingüísticos y de sus combinaciones, desde un punto de vista sincrónico o diacrónico” (RAE, 2013). Con la Web semántica se crea un entorno en donde es posible acceder a la información que se necesita de forma exacta y completa, facilitando el procesamiento de la misma (Márquez, 2007).

Tim Berners-Lee (citado en Márquez, 2007) define la Web semántica de la siguiente forma:

El primer paso es colocar los datos en la Web de un modo en que las máquinas puedan entenderlos naturalmente o convertirlos a esa forma. Esto crea lo que yo llamo la Web semántica: una red de datos que pueden ser procesados directa o indirectamente por máquinas (p.23).

Unidad 1. Fundamentos web

La Web semántica promueve la comunicación recíproca y permite la generación de contenidos propios por parte de las herramientas tipo wiki (Asensio, 2011, p. 99).



Figura 12. Web 3.0 (web semántica).

Fuente:<http://negraisa.files.wordpress.com/2012/07/nubedepalabraswed3-0.png>

La Web 3.0 es la actual, dotada de significado o Web semántica. Es un espacio donde la información tiene un significado bien definido, de manera que pudiera ser interpretado tanto por las personas como por los sistemas informáticos (Criado, 2013).

La Web semántica, según Márquez (2007), se fundamenta en el hecho de que:

Las máquinas comprendan el significado de la información disponible, pero desde un punto de vista diferente al humano. [Se refiere a] que la Web semántica es pura inteligencia artificial, aunque a las máquinas aún [no puedan] llegar a cierto grado de elaborar conclusiones, deducciones o inferencias mediante procesos de lógica matemática [...] las conclusiones a las que se lleguen dependerán de la validez, o de lo buenas que sean, las reglas que se utilicen para llegar a éstas (p. 25).



Unidad 1. Fundamentos web

Se ilustra a continuación la diferencia en los resultados de búsqueda mediante la Web 2.0 y la Web 3.0 o semántica. Por ejemplo, si se requiere consultar sobre un servicio en un determinado momento:



Figura 13. Ejemplo de resultados de búsqueda en un navegador de la Web 2.0. Fuente: W3C, 2014. <http://www.w3c.es/Divulgacion/GuiasBreves/WebSemantica>

Utilizando la Web 3.0, la búsqueda obtendría los siguientes resultados:



Unidad 1. Fundamentos web

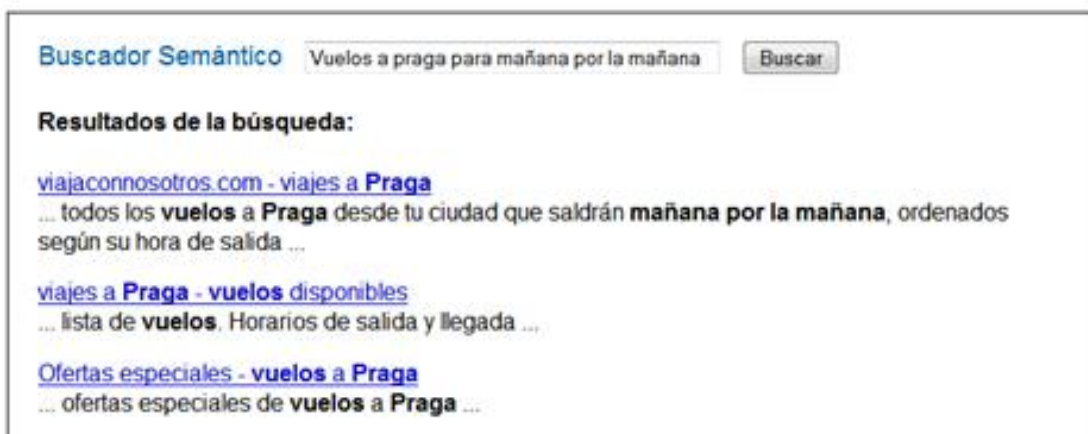


Figura 14. Ejemplo de resultados de búsqueda en un navegador de la Web 3.0. Fuente: W3C, 2014.
<http://www.w3c.es/Divulgacion/GuiasBreves/WebSemantica>

Como puedes observar, el resultado de la búsqueda mediante la Web 3.0 o semántica es más preciso.



Unidad 1. Fundamentos web

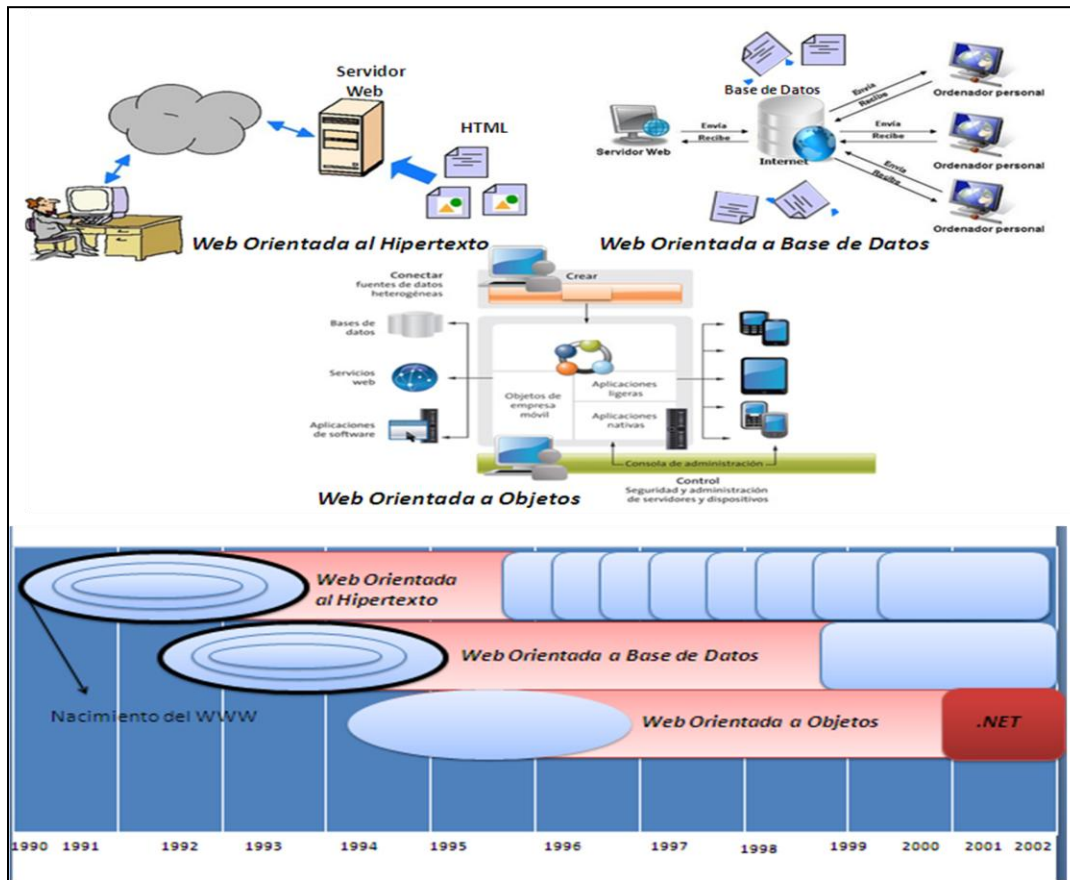


Figura 15. Evolución de la Web en relación con el uso del hipertexto. Basada en: Lamarca, 2013.

continuación se muestra los elementos que forman una página web:

- Texto
- Imagen
- Música
- Videos
- Formularios



Unidad 1. Fundamentos web



Figura 16. Elementos que forman una página web

De la Web 3.0 hacia la Web 4.0 denominada Web ubicua por su capacidad para estar en varios lugares a la vez, logra a través de la interconexión de las personas y las máquinas, una nueva civilización en la que se optimizan las bondades del ser humano a través de los valores inherentes y únicos, con la eficiencia y de la inteligencia artificial.

La Web 4.0 son las nuevas tecnologías, las plataformas sociales, las herramientas que hacen más eficiente la promoción, las aplicaciones pensadas para optimizar las experiencias, en definitiva, los instrumentos a través de los cuales podemos desarrollarnos, los que nos permite estar en varios lugares al mismo tiempo.

La conexión entre lo real y lo virtual a cualquier hora, desde cualquier dispositivo uniendo experiencias, personas, sucesos, construyendo tendencias y formando nuevas ideas innovadoras, es lo que ha modificado la forma en la que vemos el mundo, el orden social establecido y cómo nos desenvolvemos en él.

La convergencia de nuevas tecnologías, la proliferación de servicios basados en la conectividad permanente, el auge del vídeo, las redes sociales, hacen de la



Unidad 1. Fundamentos web

ubicuidad una condición innata para la generación de nativos digitales e híbridos en proceso de aprendizaje, que día a día interactúan incorporando Internet como guía en el proceso de construir un mundo más eficiente.

Adaptarse a un modelo ubicuo en el que la condición espacio–tiempo sea solventada con las nuevas tecnologías, es un camino que ya hemos iniciado.

La web 4.0 es una capa de integración necesaria para la explotación de la web semántica y sus enormes posibilidades.

Se fundamenta en cuatro pilares fundamentales:

- Comprensión del lenguaje natural (NLU) y técnicas de speech-to-text.
- Nuevos modelos de comunicación máquina- máquina (M2M). La red estará formada por agentes inteligentes en la nube, que serán capaces de comunicarse entre sí y delegar la respuesta al agente adecuado.
- Uso de información de contexto del usuario (sentiment analysis, geolocalización, sensores).
- Nuevo modelo de interacción con el usuario. Para que la web no se convierta en un mero almacén de información son necesarios nuevos modelos de interacción, o incluso ejecutar acciones concretas que den respuesta a las necesidades de los usuarios, haciendo hincapié en su uso sobre dispositivos móviles.



Unidad 1. Fundamentos web

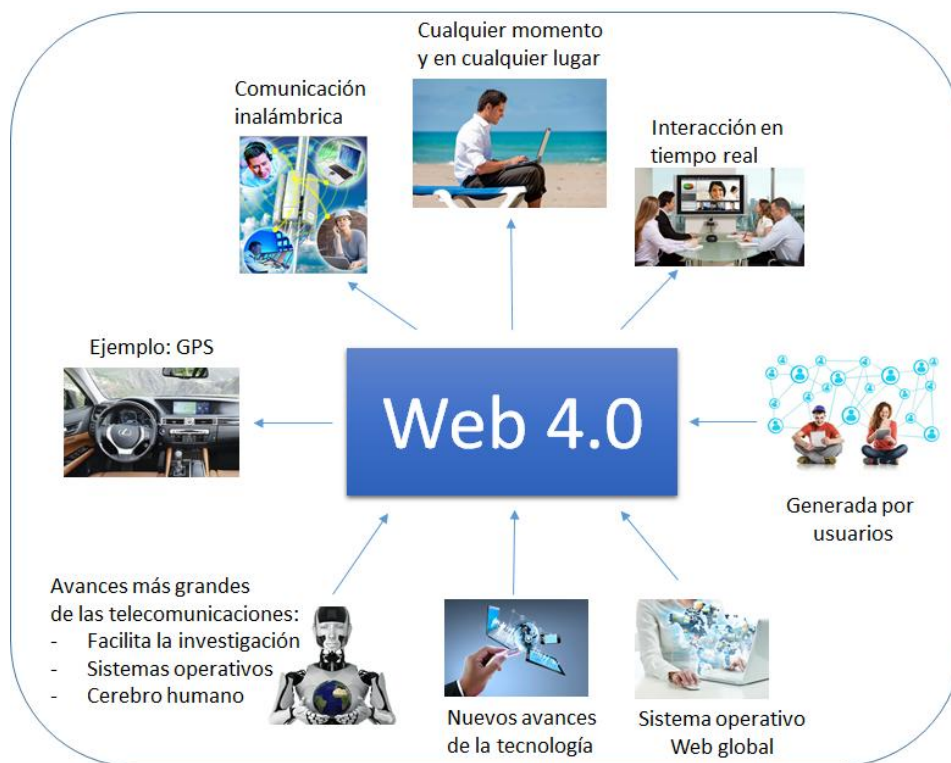


Figura 17. La web 4.0 (La web ubicua)

Como se pudo revisar en este subtema, las Web fueron evolucionando de acuerdo con las necesidades de los usuarios que se fueron integrando. En la construcción de un sitio web no habrá ningún problema al realizar o proponer sitios a las organizaciones emprendedoras, ya que se podrán visualizar los lenguajes de programación, así como identificar las generaciones que permitieron interactividad y mayor funcionalidad.



Unidad 1. Fundamentos web

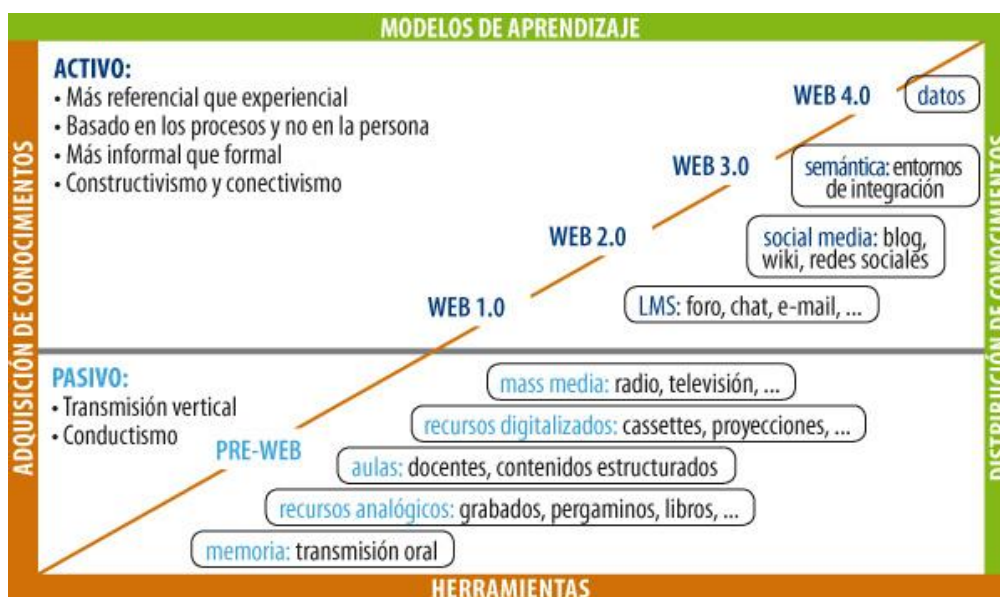


Figura 18. Evolución de la tecnología web

Respecto de la Web 4.0 es importante considerar que, con el avance de las tecnologías de información y comunicación, integra dichos avances utilizando herramientas que permiten visualizar los sitios web desde teléfonos celulares, tabletas, y diversos dispositivos de comunicación.

1.1.3. Tipos y características de servidores web

Un servidor web es necesario para la visualización de los sitios. Si no hay un servidor, el sitio web no se podrá visualizar en Internet o Intranet.

El servidor hace referencia al equipo remoto al que se realiza la conexión. El cliente sería el equipo local utilizado para efectuar dicha conexión. Pero es un software el que se conoce como programa servidor.

Programa servidor: programa que debe estar ejecutándose en el equipo servidor para que pueda ofrecer su servicio. Un documento HTML almacenado en un equipo no podría ser accesible, mediante la Web, para el resto de usuarios de Internet sin un programa servidor. En ese equipo debe estar ejecutándose un servidor web. Uno



Unidad 1. Fundamentos web

de los programas servidores web más conocido y utilizado es Apache (Cobo, Gómez, Pérez y Rocha, 2005, p.6).

A continuación, se muestra una dirección URL en la cual se describe que significa cada parte:

[http:// www.tecnologysarmonweb.com.mx/](http://www.tecnologysarmonweb.com.mx/)

Protocolo de hipertexto: http://

Nombre de la organización: technologysarmonweb

Tipo de organización: com (comercial)

País: mx (México)

Las funciones de un servidor son: visualización en Internet o Intranet, crear y contar con conexión a bases de datos.

Los servidores web más utilizados son:

- Apache
- Microsoft IIS
- NGIX
- GWS

Es muy importante conocer los requisitos que se necesitan para instalar un servidor web, así como cuándo se va a subir en Internet o Intranet.

Los requisitos necesarios son:

1. Un procesador
2. Memoria RAM
3. Disco duro: según el modelo dependerá su velocidad y rendimiento
4. Lenguaje de programación soportado



Unidad 1. Fundamentos web

5. Bases de datos que pueden ser alojadas

Apache: es uno de los servidores web más utilizados, ya que permiten crear varias conexiones de bases de datos y probar las páginas antes de ser publicadas. Apache trabaja con el lenguaje de programación PHP y ASP, los cuales permiten que se creen, haya conexión y se modifiquen las bases de datos. Apache tiene un buen funcionamiento con PHP, ya que permite que haya una buena administración en las bases de datos y facilita la conexión. En cambio, en ASP hay más reglas para la configuración en la conexión de las bases de datos que en los demás lenguajes de programación web, como por ejemplo HTML y JavaScript. Este servidor permite que se visualicen las páginas web elaboradas con dichos lenguajes de programación web en Internet o Intranet. La ventaja de este servidor web es que trabaja con diferentes sistemas operativos (Windows, Linux [Ubuntu], Mac, etcétera. Apache administra los sitios web (Vara, 2012).

Según Macías (2008):

Este servidor es un servidor HTTP de código abierto el cual ha sido desarrollado por el grupo Apache. Este servidor ha sido creado sobre los principales sistemas operativos Unix y Windows haciendo frente a los servidores web propietario de mayor uso en el mercado el servidor web tiene los requerimientos de: seguridad, eficiencia, extensibilidad (p.40).

Los requisitos para instalar el servidor web Apache deberán instalarse en un disco duro mínimo de 80Gb, memoria RAM de 2G mínimo y con sistema operativo Linux, ya que permite que el sitio web tenga seguridad y no pueda ser *hackeada*. Por ello, la mayoría de las organizaciones cuentan con Linux para la administración de su sitio web.



Unidad 1. Fundamentos web

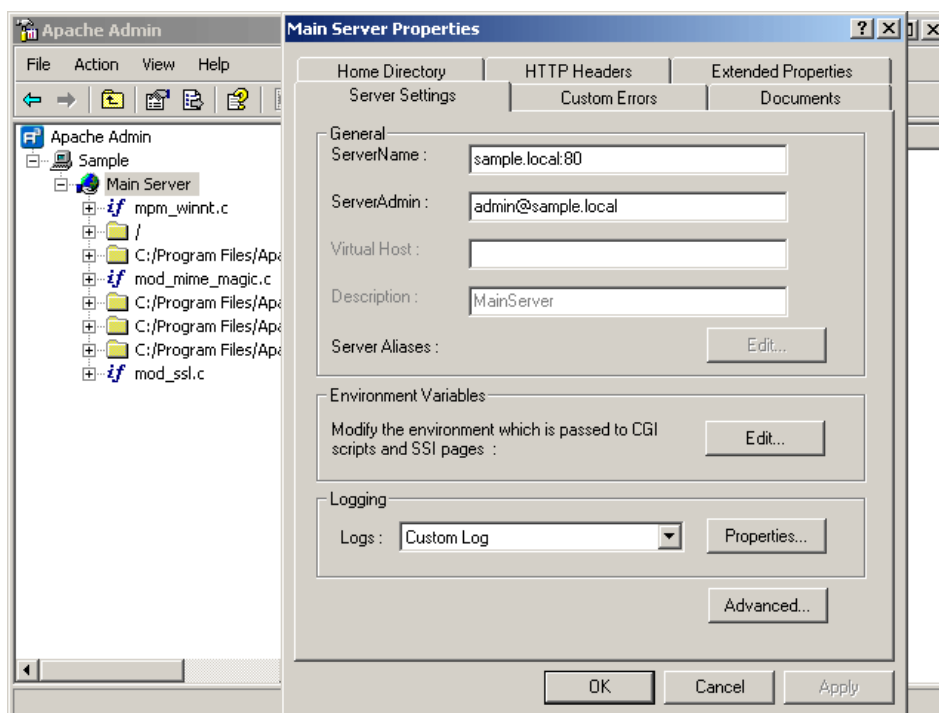


Figura 19. Servidor web Apache. Fuente: Asolif, 2012. http://www.asolif.es/sites/default/files/foto2_ApacheServer.jpg

Microsoft IIS (Internet Information Server): servidor web creado por Microsoft. Por tal motivo, sus PC se operan en el sistema operativo Windows. Se basa en reglas que establece Microsoft que no permiten realizar cambios de su sitio web a otro sistema operativo, y la conexión de base de datos se realiza por ciertas conexiones específicas de Microsoft, que fácilmente permite que haya una actualización de bases de datos (Vara, 2012). “Internet Information Services (IIS) 8.5 en Windows Server 2012 R2 proporciona una plataforma segura y fácil de administrar, modular y extensible, para hospedar de forma fiable sitios web, servicios y aplicaciones” (Microsoft, 2014b).

Según Microsoft (2014b):

Con IIS, es usted quien decide y controla, sin renunciar a la fiabilidad o la seguridad. Personalice y añada nuevas características, como el enrutamiento de solicitudes de aplicaciones y el equilibrio de carga, a través de las extensiones de IIS gratuitas. Maximice la seguridad web gracias a una



Unidad 1. Fundamentos web

exposición reducida del servidor y al aislamiento automático de la aplicación. Implemente y ejecute fácilmente aplicaciones web ASP y PHP en el mismo servidor.

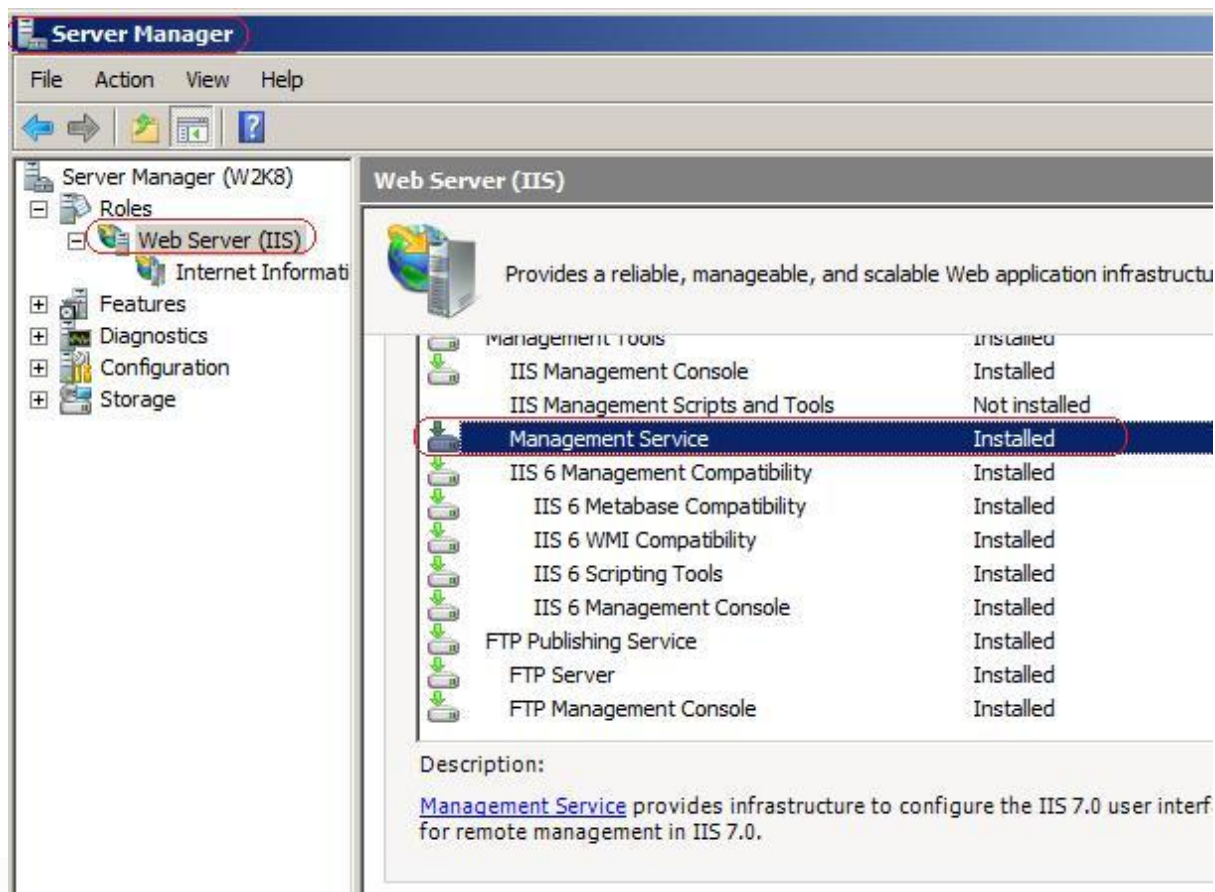


Figura 20. Servidor Web IIS. Fuente:
http://sergiot2.com/blogimages/2008/05May/18_ServerManager.jpg

IIS instala componentes opcionales como archivos comunes, documentación el complemento de servicios de Internet de información del servidor. En IIS se decide y controla sin renunciar a la fiabilidad o la seguridad.

Según Microsoft (2014b):

IIS maximiza la seguridad del servidor web de forma predeterminada gracias a la exposición mínima del servidor web y al aislamiento automático de la aplicación. IIS ofrece un espacio aislado y seguro a través de la característica de velocidad moderada de CPU, que reduce la probabilidad de las aplicaciones descontroladas. Por otra parte, IIS ofrece un mayor aislamiento de la aplicación, ya que proporciona a los procesos de trabajo



Unidad 1. Fundamentos web

una identidad única y una configuración de espacio aislado de forma predeterminada, lo que reduce todavía más los riesgos para la seguridad.

IIS es un servidor web potente y flexible que admite las aplicaciones web más conocidas, de ASP.NET a PHP. “El rol de servidor web (IIS) incluye Internet Information Services (IIS), que es una plataforma web unificada que integra IIS, ASP.NET, Windows Comunicación Fundación y Windows SharePoint Services. IIS permite compartir información con usuarios en Internet, en una Intranet o en una extranet” (Microsoft, 2014a).

Los requisitos para instalar el servidor web Apache deberán instalarse en un disco duro mínimo de 80Gb, memoria RAM de 4G mínimo y con sistema operativo WINDOWS. IIS fue creado por Windows, por lo cual, no es compatible con otros sistemas operativos. Por otro lado, el lenguaje de programación que se recomienda trabajar para la conexión de bases de datos es ASP, facilita la elaboración y la conexión. Por otro lado, es compatible con los lenguajes de programación HTML, JavaScript PHP.

Cabe mencionar que los otros servidores web no son muy usados, pero es importante tener conocimientos de ellos ya que también son utilizados en organizaciones:

NGINX: software libre y de código abierto, ejecutado bajo los sistemas operativos Windows, Linux, Mac y OS.

GWS: sólo es utilizado por Google. Se rige por alta seguridad y sólo lo administra Google.

Con el fin de que puedas observar cuáles servidores son los más utilizados, se expone a continuación una estadística anual sobre su uso.



Unidad 1. Fundamentos web

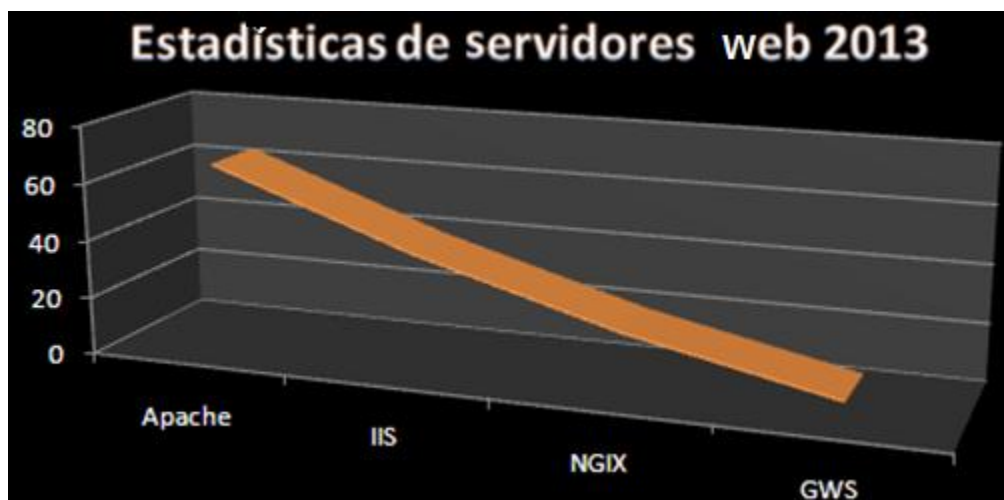


Figura 21. Estadística de servidores web más utilizados año 2013. Basado en: Netcraft.

Básicamente, la diferencia de Apache con IIS es que el primero es un servidor de código abierto y más accesible para poder interactuar con las páginas web. En cambio, IIS genera más restricciones y reglas para la administración de un sitio web y conexiones de base de datos. El servidor GWS es utilizado únicamente por Google. NGIX permite consumir bajos recursos y asegura un funcionamiento óptimo bajo mucha carga en las bases de datos. Conocer los tipos de servidores web existente ayuda a tener más claro cuál es más conveniente para la administración de un sitio web, de acuerdo a la situación y necesidad que tenga cada organización.

Actividad 2. Servidores Web

Realiza la actividad **siguiendo** las instrucciones de la misma y las indicaciones de la Figura académica.

1.2. Lenguajes de programación

Cuando ingresas a Internet, lo que observas son sitios web compuesto por varias páginas. ¿Te has preguntado cómo es que se realizan y tienen diferentes funciones



Unidad 1. Fundamentos web

como por ejemplo enviar, recibir y acceder a información? Esto es posible mediante los lenguajes de programación que “se define como un lenguaje artificial que se utiliza para expresar programas de computadoras, cada computadora, según su diseño *entiende* un cierto conjunto de instrucciones elementales (lenguaje de computadoras)” (Rodríguez, 2003, p. 4).

Una página web se construye por medio de diferentes lenguajes de programación. Para ello empezarás con algunos conceptos básicos, que seguramente en algún momento habrás escuchado. Además, observarás las diferencias de una página estática y una dinámica, distinguiendo a éstas entre páginas dinámicas de cliente y páginas dinámicas de servidor. Por último, analizarás algunos de los lenguajes para construir páginas web y reconocerás sus diferencias.

Dentro de la navegación que todo usuario hace por Internet, se visualizan diferentes tipos de sitios web. Éstos están compuestos por diferentes elementos, como por ejemplo bases de datos, videos, imágenes, descarga de música, visualización de películas, juegos, chat, etcétera. Esto es realizado en el código fuente.

El **código fuente** es un conjunto de instrucciones que se realizan por textos, números, caracteres que se utiliza en diferentes tipos de lenguajes de programación y así elaborar programas y sitios web (Luján, 2002). Cuando uno navega un sitio web y visualiza las páginas. El código fuente puede ser visualizado dando clic derecho sobre la página web que se está navegando; aparecerá un cuadro con opciones y se seleccionará la opción ver código fuente. Este proceso se muestra en la siguiente imagen:



Unidad 1. Fundamentos web

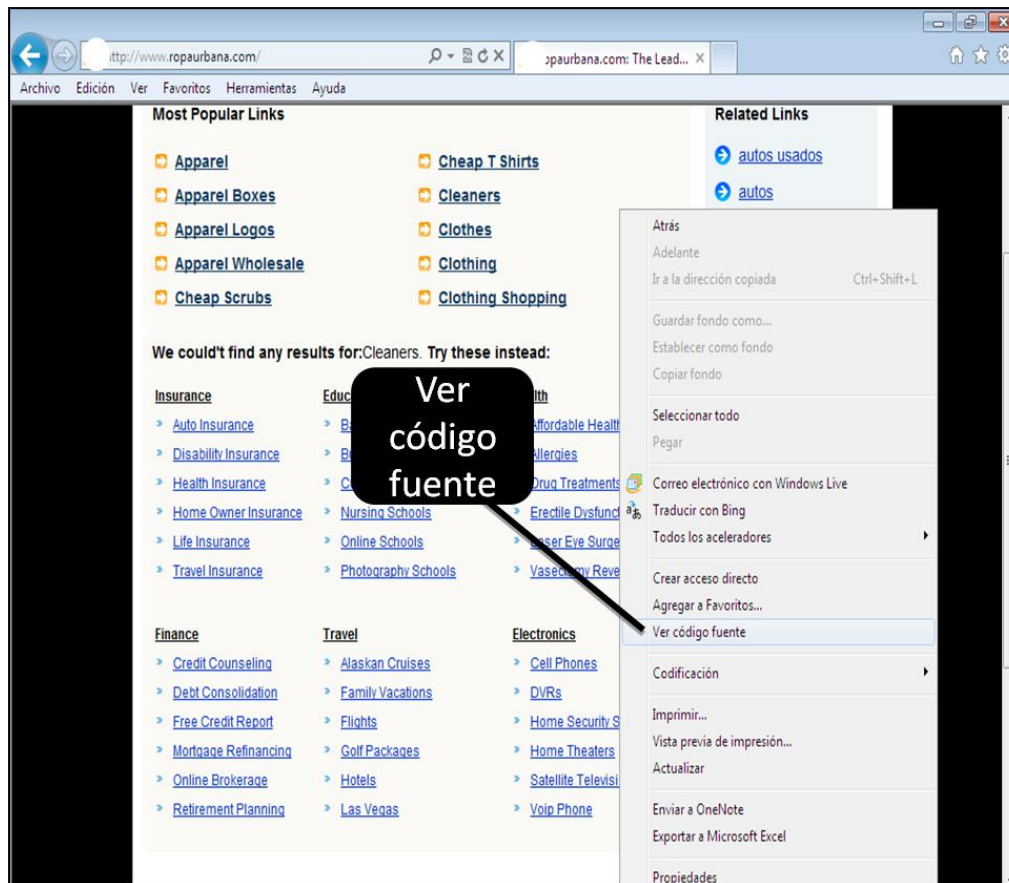


Figura 22. Pasos para ver el código fuente de una página web



Unidad 1. Fundamentos web

La siguiente imagen muestra el resultado cuando se da clic en la opción ver código fuente:

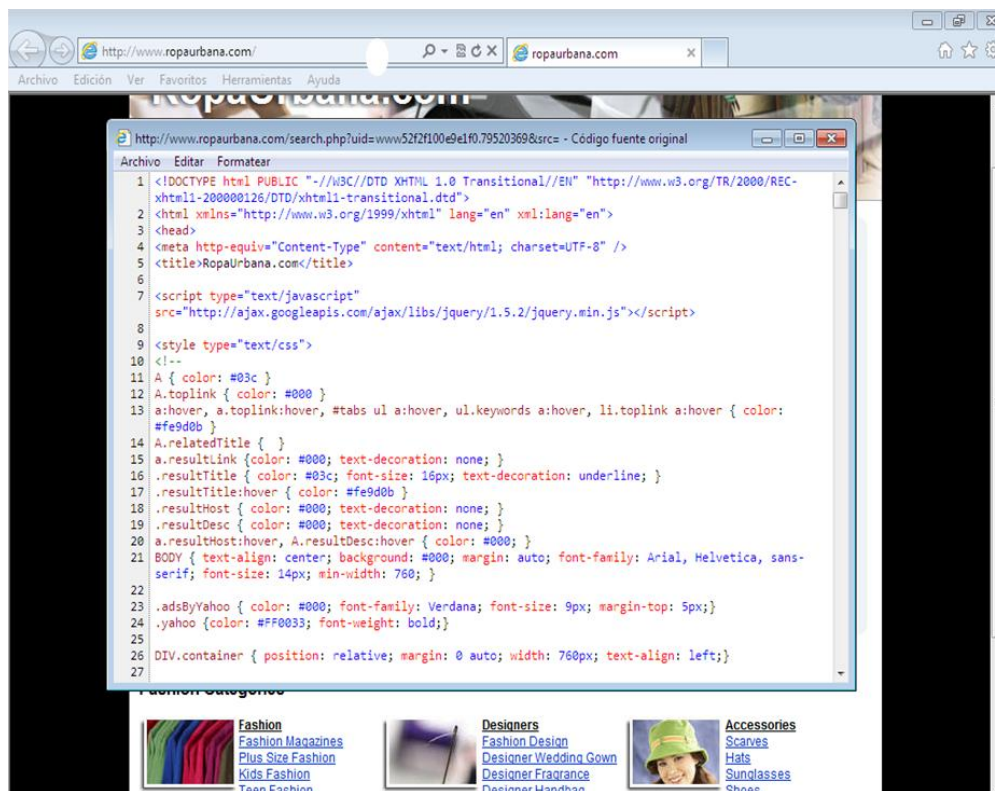


Figura 23. Resultado de ver el código fuente de una página web



Unidad 1. Fundamentos web

Es importante tener muy claro qué es un sitio web. Los lenguajes de programación permiten la construcción de sitios web, los cuales se componen por varias páginas, ya sean estáticas o dinámicas.

Las páginas estáticas son aquellas que sólo llevan texto, imágenes, música, objetos animados. Una página dinámica permite interactuar con varios usuarios por medio de base de datos (Vértice, 2009).

PHP es un lenguaje de programación web. Se basa en la realización de páginas dinámicas, por ejemplo, creación de *login* de usuarios o formularios (Welling y Thomson, 2009).

ASP es un lenguaje de programación que permite realizar páginas dinámicas y estáticas. Se encarga de realizar plantillas de sitios web donde ya están incorporada la conexión de base de datos (Rayo, 2009).

Ruby On Rails es un marco de trabajo (framework) para el desarrollo de aplicaciones web de código abierto, escrito en el lenguaje de programación Ruby siguiendo el paradigma de la arquitectura Modelo Vista Controlador (MVC). Ruby es un lenguaje dinámico orientado a objetos, con sintaxis inspirada en Python, Perl y Smalltalk.

JavaScript, como expresa Orós (2011), “es un lenguaje de programación creado por Netscape con el objetivo de integrarse en HTML y facilitar la creación de páginas interactivas” (p. 57). Por tal motivo, JavaScript funciona con la extensión de HTML y permite que una página web realice varias acciones, desde un calendario hasta un botón, con un solo clic puede almacenar datos, imprimir un formato en PDF y abrir una ventana diciendo gracias.

Para la visualización de la interface gráfica, en el contexto de la interacción persona-computadora como un artefacto interactivo, que por su diseño de la forma en que está realizada, se visualiza signos gráficos como iconos, botones, menús y verbales como tipografía en un navegador web.



Unidad 1. Fundamentos web

Un navegador web es un software por medio del cual es posible visualizar una interfaz gráfica de las páginas web que se están construyendo, con los diferentes lenguajes de programación web.

Saber en general el funcionamiento de cada lenguaje te posibilitará crear y realizar tus propias ideas con combinaciones con estos lenguajes de programación web, donde los diseños sólo tienen como límite tu propia creatividad.

1.2.1. Lenguaje HTML y CSS

HTML se creó en año de 1980 en Suiza por un científico llamado Tim Berners Lee. (Moschovitis, *et al*, 2005). HTML son las iniciales de Lenguaje de etiquetas de hipertexto (por sus siglas en inglés de *Hiper TextMarkupLenguaje*). Es el lenguaje con el que se escriben las páginas web, ya sean estáticas o dinámicas.

El HTML no es más que una serie de etiquetas para definir la forma o estilo que se desea para nuestros documentos (Orós, 2011).

Como expresa Orós (2011) “existe normalmente una etiqueta de inicio<ETIQUETA> y otra de final </ETIQUETA>, si bien algunos comandos solo disponen de la de inicio en cualquier caso todos los elementos situados entre etiquetas o a partir de la de inicio, serán afectados por estas, ya se trate de texto o de más etiquetas” (p. 1). Esta estructura de HTML es la forma básica para construir páginas web.

Un documento HTML siempre debe estar compuesto por seis partes principales que se mencionan a continuación (Orós, 2011):

1. Se empieza a construir con la etiqueta de inicio. Se define <HTML> y </HTML>, en un editor de texto plano. Al igual, se puede realizar la construcción con una herramienta web mejor conocida como editor web, tal como.
2. El encabezado, delimitado por <head> y </head>, donde se colocarán etiquetas de índole informativo como por ejemplo el título de la página o la información.



Unidad 1. Fundamentos web

3. El cuerpo de la página dentro de las etiquetas **<body>** y **</body>**, que es donde se captura el código para la visualización del texto y de las imágenes, delimitados a su vez por otras etiquetas.
4. La etiqueta **<title>** es la encargada de especificar el título del documento (página web).
5. **<P>**.Esta etiqueta se utiliza para separar párrafos.
6. La etiqueta **<Hn>**, donde n haría de 1 al 6.El valor 1 corresponde con un tamaño de letra muy grande y así hasta que llega el 6 y se hace la letra más pequeña (Orós, 2011).

Ejemplo de ejercicio HTML:

Código fuente de lenguaje de programación web HTML.

```
<html>
<head>
<title>Bienvenidos Alumnos</title>
</head>
<body text="white" bgcolor="black">
<h1 align="center">Bienvenido a la asignatura Programación web I
</h1>
<h2 align="center">Ingeniería en Desarrollo de Software</h2>
<p>&nbsp;</p>
<h1 align="center">Lenguaje HTML</h1>
<p align="center">&nbsp;</p>
<p align="center"></p>
</body>
</html>
```

Figura 24. Ejemplo del código lenguaje de programación HTML



Unidad 1. Fundamentos web

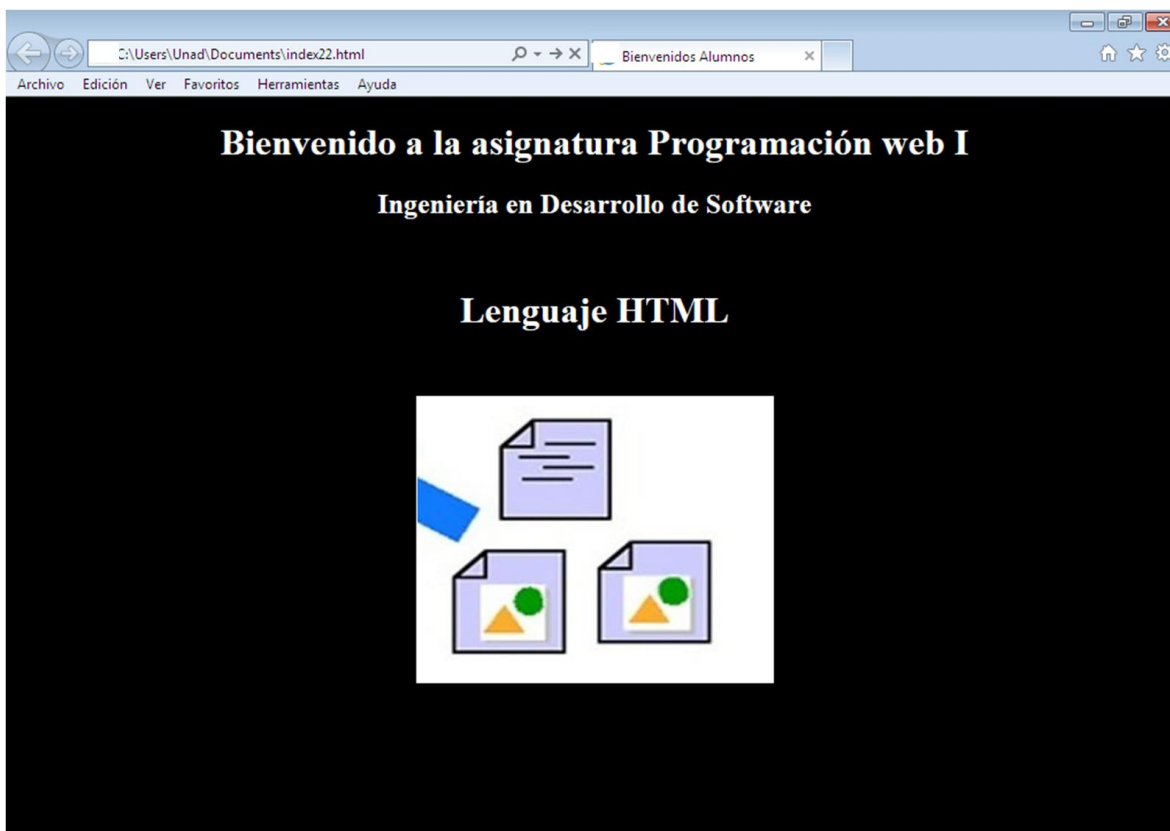


Figura 25. Resultado del uso del código del lenguaje de programación web HTML

Actualmente, el lenguaje HTML se encuentra en su versión 5 y continúa en desarrollo. En esta última versión es posible encontrar algunas etiquetas usadas para la presentación de materiales multimedia en la Web, tales como videos o audios en mp3. El HTML5 es el futuro de la Web, tiene funcionalidades nuevas como por ejemplo mejoras en el video y audio, nuevas etiquetas para la realización de formularios. Esto con el fin de conseguir que los sitios web sean más interesantes, atractivos y útiles (Vértice, 2009).

HTML es un lenguaje de programación web que puede trabajar con diferentes sistemas operativos, por ejemplo, Windows, Linux, Mac, UNIX. HTML. Como tal, no hay un requerimiento para la construcción de una página web en HTML, sólo se necesita un editor de texto como un blog de notas. Una vez elaborada una página web en HTML, es muy importante que se guarde con la extensión **.html** para que se pueda visualizar el contenido de la página web. HTML tiene compatibilidad con diferentes servidores web, tales como Apache y IIS.



Unidad 1. Fundamentos web

HTML tiene como ventaja que es un lenguaje no complicado y accesible para la construcción de las páginas web, y permite estar haciendo modificaciones sin problema alguno.

HTML es un lenguaje de programación web que se inició para la realización de sitios web, tal como expresa Orós (2011), “el lenguaje HTML que no permite crear conexiones a bases de datos se basa en el uso de etiquetas también llamadas marcas, directivas o comandos. Estas Etiquetas son fragmentos de textos delimitados por los símbolos menor que < y mayor que >. Básicamente, estas etiquetas indican al navegador, la forma de representar los elementos (texto, grafico, etcétera.)” (p. 1). De tal forma, HTML permite que se realicen páginas estáticas y en diseño de la página web; por ejemplo, ponerle fondo ya sea color o imagen, realizar tablas, música y videos. Tendrá forzosamente que trabajar con los demás lenguajes de programación web.

Después de haber terminado la primera parte, exclusivamente en HTML, ahora nos encontramos con CSS. CSS no es más complicado que HTML, es el complemento para ayudarte a dar formato a tu página web.

CSS son las siglas de Cascading Style Sheets - Hojas de Estilo en Cascada - que es un lenguaje que describe la presentación de los documentos estructurados en hojas de estilo para diferentes métodos de interpretación, es decir, describe cómo se va a mostrar un documento en pantalla, por impresora, por voz (cuando la información es pronunciada a través de un dispositivo de lectura) o en dispositivos táctiles basados en Braille.

¿Para qué sirve? CSS es una especificación desarrollada por el W3C (World Wide Web Consortium) para permitir la separación de los contenidos de los documentos escritos en HTML, XML, XHTML, SVG, o XUL de la presentación del documento con las hojas de estilo, incluyendo elementos tales como los colores, fondos, márgenes, bordes, tipos de letra, etcétera, modificando la apariencia de una página web de una forma más sencilla, permitiendo a los desarrolladores controlar el estilo y formato de sus documentos.



Unidad 1. Fundamentos web

¿Cómo funciona? El lenguaje CSS se basa en una serie de reglas que rigen el estilo de los elementos en los documentos estructurados, y que forman la sintaxis de las hojas de estilo. Cada regla consiste en un selector y una declaración, esta última va entre llaves y consiste en una propiedad o atributo, y un valor separados por dos puntos.

Ejemplo:

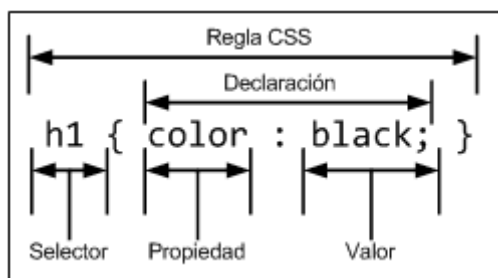


Figura 26. Estructura de una regla CSS

El **Selector** especifica que elementos HTML van a estar afectados por esa declaración, de manera que hace de enlace entre la estructura del documento y la regla estilística en la hoja de estilo.

La **Declaración** que va entre corchetes es la información de estilo que indica cómo se va a ver el selector. En caso de que haya más de una declaración se usa punto y coma para separarlas.

Propiedad o Atributo y Valor dentro de la declaración, la propiedad o atributo define la interpretación del elemento asignándosele un cierto valor, que puede ser color, alineación, tipo de fuente, tamaño, entre otros, es decir, especifican qué aspecto del selector se va a cambiar.

La información CSS se puede proporcionar por varias fuentes, ya sea adjunto como un documento por separado o incorporado en el documento HTML, y dentro de estas posibilidades destacan tres formas de dar estilo a un documento web:

La **Hoja de Estilo Externa** se almacena en un archivo diferente al del archivo con el código HTML al cual está vinculado a través del elemento link, que debe ir



Unidad 1. Fundamentos web

situado en la sección head. Es la manera de programar más eficiente, ya que separa completamente las reglas de formato para la página HTML de la estructura básica de la página.

La **Hoja de Estilo Interna** está incorporada a un documento HTML, a través del elemento style dentro de la sección head, consiguiendo de esta manera separar la información del estilo del código HTML.

El **Estilo en Línea** sirve para insertar el lenguaje de estilo directamente dentro de la sección body con el elemento style. Sin embargo, este tipo de estilo no se recomienda pues se debe intentar siempre separar el contenido de la presentación.

Funcionamiento básico de CSS

Antes de que se generalizara el uso de CSS, los diseñadores de páginas web utilizaban etiquetas HTML especiales para modificar el aspecto de los elementos de la página.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
  <title>Ejemplo de estilos sin CSS</title>
</head>
<body>
  <h1><font color="red" face="Arial" size="5">Titulo de la página</font></h1>
  <p><font color="gray" face="Verdana" size="2">Un párrafo de texto, solo como ejemplo.</font></p>
</body>
</html>
```

Figura 27. Ejemplo de una página HTML con estilos definidos sin utilizar CSS

El ejemplo anterior utiliza la etiqueta con sus atributos color, face y size para definir el color, el tipo y el tamaño de letra de cada elemento de la página.



Unidad 1. Fundamentos web

El problema de utilizar este método para definir el aspecto de los elementos se puede ver claramente con el siguiente ejemplo: si la página tuviera 50 elementos diferentes, habría que insertar 50 etiquetas `<font>`. Si el sitio web entero se compone de 1,000 páginas diferentes, habría que definir 50,000 etiquetas `<font>`. Como cada etiqueta `<font>` tiene tres atributos, habría que definir 150,000 atributos.

Como el diseño de los sitios web está en constante evolución, es habitual modificar cada cierto tiempo el aspecto de las páginas del sitio. Siguiendo con el ejemplo anterior, cambiar el aspecto del sitio requeriría modificar 50,000 etiquetas y 150,000 atributos.

La solución que propone CSS es mucho mejor, como se puede ver en el siguiente ejemplo:



Unidad 1. Fundamentos web

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>Ejemplo de estilos con CSS</title>
<style type="text/css">
    h1 { color: red; font-family: Arial; font-size: large; }
    p { color: gray; font-family: Verdana; font-size: medium; }
</style>
</head>

<body>
    <h1>Titular de la página</h1>
    <p>Un párrafo de texto, como ejemplo.</p>
</body>
</html>
```

Figura 28. Un método menos desgastante que propone CSS.

CSS permite separar los contenidos de la página y la información sobre su aspecto. En el ejemplo anterior, dentro de la propia página HTML se crea una zona especial en la que se incluye toda la información relacionada con los estilos de la página.

Utilizando CSS, se pueden establecer los mismos estilos con menos esfuerzo y sin ensuciar el código HTML de los contenidos con etiquetas . La etiqueta <style> crea una zona especial donde se incluyen todas las reglas CSS que se aplican en la página.



Unidad 1. Fundamentos web

En el ejemplo anterior, dentro de la zona de CSS se indica que todas las etiquetas `<h1>` de la página se deben ver de color rojo, con un tipo de letra Arial y con un tamaño de letra grande. Además, las etiquetas `<p>` de la página se deben ver de color gris, con un tipo de letra Verdana y con un tamaño de letra medio.

Definir los estilos de esta forma ahorra miles de etiquetas y millones de atributos respecto a la solución anterior, pero sigue sin ser una solución ideal. Como los estilos CSS sólo se aplican en la página que los incluye, si queremos que las 1,000 páginas diferentes del sitio tengan el mismo aspecto, se deberían copiar 1,000 veces esas mismas reglas CSS. Para evitar esto CSS propone algunas soluciones las cuales se explican a continuación.

Incluir CSS en el mismo documento HTML

Los estilos se definen en una zona específica del propio documento HTML. Se emplea la etiqueta `<style>` de HTML y solamente se pueden incluir en la cabecera del documento (sólo dentro de la sección `<head>`).



Unidad 1. Fundamentos web

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1"/>
<title>Ejemplo de estilos CSS en el mismo documento</title>
<style type="text/css">
    p { color: black; font-family: Verdana; }
</style>
</head>
<body>
<p>Un párrafo de texto.</p>
</body>
</html>
```

Figura 29. Como definir estilos con CSS en un mismo documento.

Este método se emplea cuando se define un número pequeño de estilos o cuando se quieren incluir estilos específicos en una determinada página HTML que completen los estilos que se incluyen por defecto en todas las páginas del sitio web.

El principal inconveniente es que si se quiere hacer una modificación en los estilos definidos, es necesario modificar todas las páginas que incluyen el estilo que se va a modificar.

Los ejemplos mostrados en este libro utilizan este método para aplicar CSS al contenido HTML de las páginas. De esta forma el código de los ejemplos es más conciso y se aprovecha mejor el espacio.

Definir CSS en un archivo externo



Unidad 1. Fundamentos web

En este caso, todos los estilos CSS se incluyen en un archivo de tipo CSS que las páginas HTML enlazan mediante la etiqueta `<link>`. Un archivo de tipo CSS no es más que un archivo simple de texto cuya extensión es `.css`. Se pueden crear todos los archivos CSS que sean necesarios y cada página HTML puede enlazar tantos archivos CSS como necesite.

Si se quieren incluir los estilos del ejemplo anterior en un archivo CSS externo, se deben seguir los siguientes pasos:

1) Se crea un archivo de texto y se le añade solamente el siguiente contenido:

```
p { color: black; font-family: Verdana; }
```

2) Se guarda el archivo de texto con el nombre `estilos.css`. Se debe poner especial atención a que el archivo tenga extensión `.css` y no `.txt`.

3) En la página HTML se enlaza el archivo CSS externo mediante la etiqueta `<link>`:



Unidad 1. Fundamentos web

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1"/>
<title>Ejemplo de estilos CSS en un archivo externo</title>
<link rel="stylesheet" type="text/css" href="/css/estilos.css" media="screen"/>
</head>
<body>
<p>Un párrafo de texto.</p>
</body>
</html>
```

Figura 30. Como incluir estilos en un archivo CSS externo.

Cuando el navegador carga la página HTML anterior, antes de mostrar sus contenidos también descarga los archivos CSS externos enlazados mediante la etiqueta `<link>` y aplica los estilos a los contenidos de la página.

Normalmente, la etiqueta `<link>` incluye cuatro atributos cuando enlaza un archivo CSS:

- `rel`: indica el tipo de relación que existe entre el recurso enlazado (en este caso, el archivo CSS) y la página HTML. Para los archivos CSS, siempre se utiliza el valor `stylesheet`
- `type`: indica el tipo de recurso enlazado. Sus valores están estandarizados y para los archivos CSS su valor siempre es `text/css`



Unidad 1. Fundamentos web

- href: indica la URL del archivo CSS que contiene los estilos. La URL indicada puede ser relativa o absoluta y puede apuntar a un recurso interno o externo al sitio web.
- media: indica el medio en el que se van a aplicar los estilos del archivo CSS. Más adelante se explican en detalle los medios CSS y su funcionamiento.

De todas las formas de incluir CSS en las páginas HTML, esta es la más utilizada con mucha diferencia. La principal ventaja es que se puede incluir un mismo archivo CSS en multitud de páginas HTML, por lo que se garantiza la aplicación homogénea de los mismos estilos a todas las páginas que forman un sitio web.

Con este método, el mantenimiento del sitio web se simplifica al máximo, ya que un solo cambio en un solo archivo CSS permite variar de forma instantánea los estilos de todas las páginas HTML que enlazan ese archivo.

Aunque generalmente se emplea la etiqueta `<link>` para enlazar los archivos CSS externos, también se puede utilizar la etiqueta `<style>`. La forma alternativa de incluir un archivo CSS externo se muestra a continuación:



Unidad 1. Fundamentos web

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1"/>
<title>Ejemplo de estilos CSS en un archivo externo</title>
<style type="text/css" media="screen">
    @import '/css/estilos.css';
</style>
</head>
<body>
<p>Un párrafo de texto.</p>
</body>
</html>
```

Figura 31. Otra forma de incluir estilos en un archivo CSS externo.

En este caso, para incluir en la página HTML los estilos definidos en archivos CSS externos se utiliza una regla especial de tipo `@import`. Las reglas de tipo `@import` siempre preceden a cualquier otra regla CSS (con la única excepción de la regla `@charset`).

La URL del archivo CSS externo se indica mediante una cadena de texto encerrada con comillas simples o dobles o mediante la palabra reservada `url()`. De esta forma, las siguientes reglas `@import` son equivalentes:

```
@import '/css/estilos.css';
```

```
@import "/css/estilos.css";
```

```
@import url('/css/estilos.css');
```



Unidad 1. Fundamentos web

@import url("/css/estilos.css");

Incluir CSS en los elementos HTML

El último método para incluir estilos CSS en documentos HTML es el peor y el menos utilizado, ya que tiene los mismos problemas que la utilización de las etiquetas .

Ejemplo:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1"/>
<title>Ejemplo de estilos CSS en el propio documento</title>
</head>
<body>
<p style="color: black; font-family: Verdana;">Un párrafo de texto.</p>
</body>
</html>
```

Figura 32. Una tercera forma de incluir estilos en un archivo CSS externo (No es recomendable)

Esta forma de incluir CSS directamente en los elementos HTML solamente se utiliza en determinadas situaciones en las que se debe incluir un estilo muy específico para un solo elemento concreto.



Unidad 1. Fundamentos web

1.2.2. Lenguaje PHP

El lenguaje PHP se utiliza con la finalidad de hacer conexión de base de datos ya que este lenguaje se encarga de realizar páginas dinámicas. Este lenguaje de programación web trabaja con otros lenguajes de programación web por ejemplo HTML, JavaScript, ASP. En PHP tiene compatibilidad con diferentes servidores web para la conexión de base de datos. PHP tiene potencia, alto rendimiento, facilidad de aprendizaje y su escasez de consumo de recursos.

PHP (*HypertextPreprocessor*) fue desarrollado en el año de 1994 por RasmusLerdof. Es un lenguaje dinámico ya que permite interactuar con conexiones de base de datos. Es un lenguaje de código abierto que interactúa con los lenguajes HTML, y JavaScript (Welling y Thomson, 2009).

La ventaja en el uso de PHP principalmente se refiere a que es extremadamente simple para el principiante, pero a su vez ofrece muchas características avanzadas para los programadores profesionales.



```

<form>
  <p>    Clase de Producto:
          <select name="id" id="id" onchange="submit()">
            <option value=""> </option>
          <?php
do {
?>
          <option value="<?php echo $row_Recordset1['id']
?>"<?php if (!(strcmp($row_Recordset1['id'], $_POST['id']))) {echo
"selected=\"selected\"";} ?>"<?php echo $row_Recordset1['cod_depa
']?>"></option>

          <?php
} while ($row_Recordset1 = mysql_fetch_assoc($Recordset1));
$rows = mysql_num_rows($Recordset1);
if($rows > 0) {
    mysql_data_seek($Recordset1, 0);
    $row_Recordset1 = mysql_fetch_assoc($Recordset1);
}
?>

          </select></p>

  Producto:
  <select name="id_producto" id="id_producto"
onchange="submit()">
    <option value="">Seleccione una opcion:</option>
  <?php

```

LINEA DE COMPRAS

Clase de Producto:

Producto:

<form action="mailto:escaner@teleton-neza.org.mx?subject="
php con fecha de php -php " method="post" onsubmi
 Proveedor:

Otro Producto : **Cantidad:**

Solicitante:

Departamento: **Fecha**
de Orden:

Figura 33. Ejemplo de aplicación del lenguaje de programación web PHP



Unidad 1. Fundamentos web

Esta imagen muestra un ejemplo de construcción en PHP. Recuerda que PHP se utiliza para la realización de páginas dinámicas, esto es que permite realizar base de datos y la conexión con un servidor web.

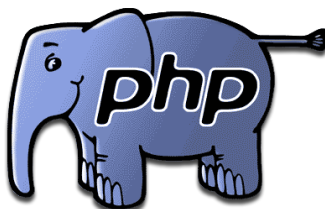


Figura 34. Logotipo del lenguaje de programación PHP. Fuente: http://www.nuevoiris.com/img/a0/naTexto5_PHP.png

PHP se encarga de procesar la información de formularios, generar páginas con contenidos dinámicos y cookies (Welling, 2009). Cookies es una característica de PHP. Permite almacenar archivos en la PC. Los Scripts son conjuntos de instrucciones que se pueden realizar en una página web.

Este lenguaje se utiliza cuando se requiere el uso de scripts que se ejecuten regularmente desde (en Linux) o mediante el planificador de tareas (en Windows).

Ejemplo de un script PHP:

Figura 35. Script

PHP. Fuente: <http://www.sendblaster.com/sendblaster/release2/sites/default/files/image/en/suport/screenshot6.gif>

Se muestra el código fuente de PHP de la creación de base de datos. Se identifica si está elaborado en PHP porque la página que muestra el código presenta la etiqueta:



Unidad 1. Fundamentos web

<?PHP y finaliza ?>

PHP trabaja al 100% con el servidor web Apache.



Unidad 1. Fundamentos web

The image shows a PHP code editor window titled "Sin título: Bloc de notas" with a menu bar (Archivo, Edición, Formato, Ver, Ayuda). The code is a PHP script for connecting to a database and querying user information. A large blue arrow points from the code to a form on the right.

```
<?php require_once('Connections/sist_compras.php'); ?>
<?php
if (!function_exists("GetSQLValueString")) {
function GetSQLValueString($theValue, $theType, $theDefinedvalue = "", $theNotDefinedvalue = "")
{
    if (PHP_VERSION < 6) {
        $theValue = get_magic_quotes_gpc() ? stripslashes($theValue) : $theValue;
    }

    $theValue = function_exists("mysql_real_escape_string") ? mysql_real_escape_string($theValue) : mysql_escape_string($theval

switch ($theType) {
case "text":
    $theValue = ($theValue != "") ? "'" . $theValue . "'" : "NULL";
    break;
case "long":
case "int":
    $theValue = ($theValue != "") ? intval($theValue) : "NULL";
    break;
case "double":
    $theValue = ($theValue != "") ? doubleval($theValue) : "NULL";
    break;
case "date":
    $theValue = ($theValue != "") ? "'" . $theValue . "'" : "NULL";
    break;
case "defined":
    $theValue = ($theValue != "") ? $theDefinedvalue : $theNotDefinedvalue;
    break;
}
return $theValue;
}

$colname_Recordset1 = "-1";
if (isset($_GET['id_usuario'])) {
    $colname_Recordset1 = $_GET['id_usuario'];
}

mysql_select_db($database_sist_compras, $sist_compras);
$query_Recordset1 = sprintf("SELECT * FROM usuario WHERE id_usuario = %s", GetSQLValueString($colname_Recordset1, "int"));
$Recordset1 = mysql_query($query_Recordset1, $sist_compras) or die(mysql_error());
$row_Recordset1 = mysql_fetch_assoc($Recordset1);
$totalRows_Recordset1 = mysql_num_rows($Recordset1);
?>
```

The form on the right is a light blue box with a dotted border. It contains the following fields:

- Id_usuario:** {Recordset1.id_usuario}
- Usuario:** {Recordset1.usuario}
- Pass:** {Recordset1.pass}
- Enviar** (button)

Figura 36. Código PHP conexión de base de datos



Unidad 1. Fundamentos web

PHP puede ser utilizado en los sistemas operativos Linux, Windows y Mac. Permite tener un control administrativo ordenado. Por otro lado, es recomendable utilizar el sistema operativo LINUX (Ubuntu), ya que hay mucha más seguridad en el sitio web en la administración.

PHP es un lenguaje muy factible y muy necesitado por las organizaciones, ya que permite realizar conexiones de bases de datos y tiene una factibilidad para trabajar con los servidores web.

PHP y ASP, que se explicarán a continuación, tienen como relación que por medio de ellos realizan páginas dinámicas.

1.2.3. Lenguaje Ruby On Rails

Ruby on Rails (RoR) es un framework de desarrollo web de código abierto, escrito en Ruby, siguiendo el paradigma de la arquitectura Modelo Vista Controlador (MVC), el cual permite un desarrollo más eficiente, desacoplado y mantenible de las aplicaciones.

Ruby es un lenguaje con un balance cuidado. Su creador, Yukihiro “Matz” Matsumoto, mezcló partes de sus lenguajes favoritos (Perl, Smalltalk, Eiffel, Ada y Lisp) para formar un nuevo lenguaje que incorporara tanto la programación funcional como la imperativa.

A menudo ha manifestado que está “tratando de hacer que Ruby sea natural, no simple”, de una forma que se asemeje a la vida real.

Rails es un almacén para construir aplicaciones web que acceden a bases de datos, es decir es un conjunto de librerías, automatismos y convenciones destinados a resolver los problemas más comunes a la hora de desarrollar una aplicación web, para que el programador pueda concentrarse en los aspectos únicos y diferenciales de su proyecto en lugar de los problemas recurrentes.



Unidad 1. Fundamentos web

Rails fue creado en 2003 por David Heinemeier Hansson y desde entonces ha sido extendido por el Rails core team, más de 2,100 colaboradores y soportado por una extensa y activa comunidad.

Entre sus características principales están:

RoR es un *framework* basado en las metodologías ágiles. Es un ejemplo perfecto de cómo un marco mejora la productividad. De hecho, Ruby on Rails podría definirse como el *framework* ágil por excelencia.

Es un marco de desarrollo sencillo de aprender. Para manejar Ruby on Rails es necesario tener algunos conocimientos de PHP. Si se domina ese lenguaje, muy usado para proyectos digitales de todo tipo, el aprendizaje de esta herramienta se convierte en un juego de niños.

Ruby on Rails funciona bajo un paradigma MVC (Modelo-Vista-Controlador). Este paradigma permite separar la lógica del diseño de la lógica del negocio y facilita la escalabilidad de los desarrollos. Además, esto posibilita al equipo de desarrollo el buen mantenimiento del código sin destinar excesivos recursos.

Permite el desarrollo rápido de aplicaciones web. Uso económico del código para la programación de los proyectos. Menos es más.

Creación automática de base de datos y formularios. Las relaciones entre los distintos modelos de datos se configuran a partir de los macros que vienen por defecto en Ruby on Rails, lo que facilita mucho el trabajo de desarrollo con bases de datos relacionales como MySQL, por ejemplo. De todas formas, RoR proporciona una herramienta llamada ActiveRecord, que posibilita acceder a la base de datos sin ejecutar código SQL. ActiveRecord ofrece una metodología de búsqueda en bases de datos mediante el uso de tablas. Es una forma de economizar las consultas tradicionales SQL mediante un protocolo llamado joins.



Unidad 1. Fundamentos web

Dispone de numerosas librerías que facilitan el trabajo de configuración del propio marco de desarrollo. Esto permite que el desarrollador se concentre más en las necesidades de la aplicación.

Como marco de desarrollo MVC, Ruby on Rails es una herramienta que controla bien los recursos del servidor. Esto evita errores que terminen por afectar negativamente al rendimiento del proyecto.

Mi primera App en RoR

Ahora vamos a crear nuestra primera aplicación, para ello primero nos posicionaremos en la carpeta donde queremos que se creen todos los archivos necesarios para nuestra aplicación. Ejemplo:

```
C:\CursoRuby>
```

```
C:\CursoRuby> cd Proyecto01
```

```
C:\CursoRuby\Proyecto01>
```

Y ahora colocamos lo siguiente:

```
rails new MiPrimerApp
```

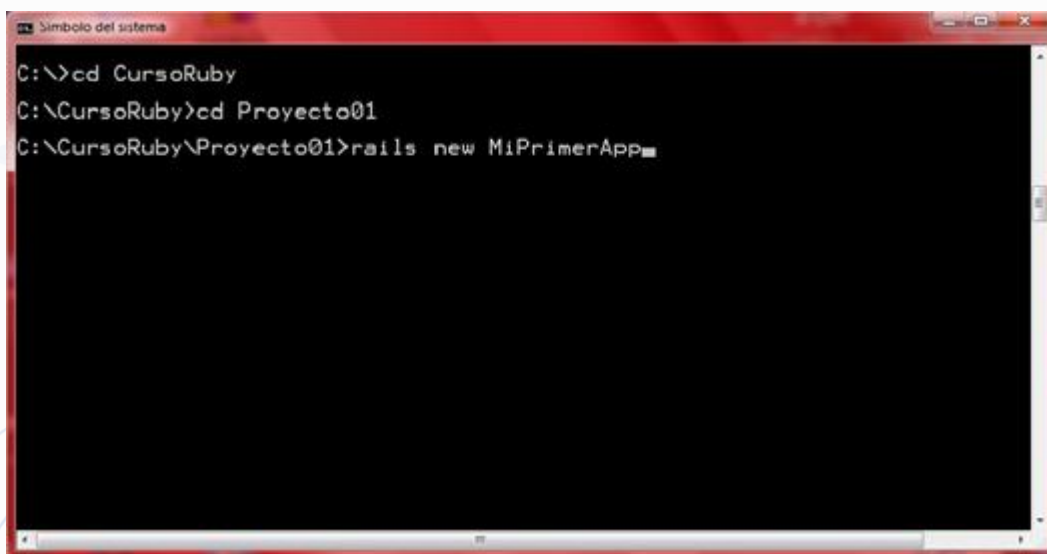
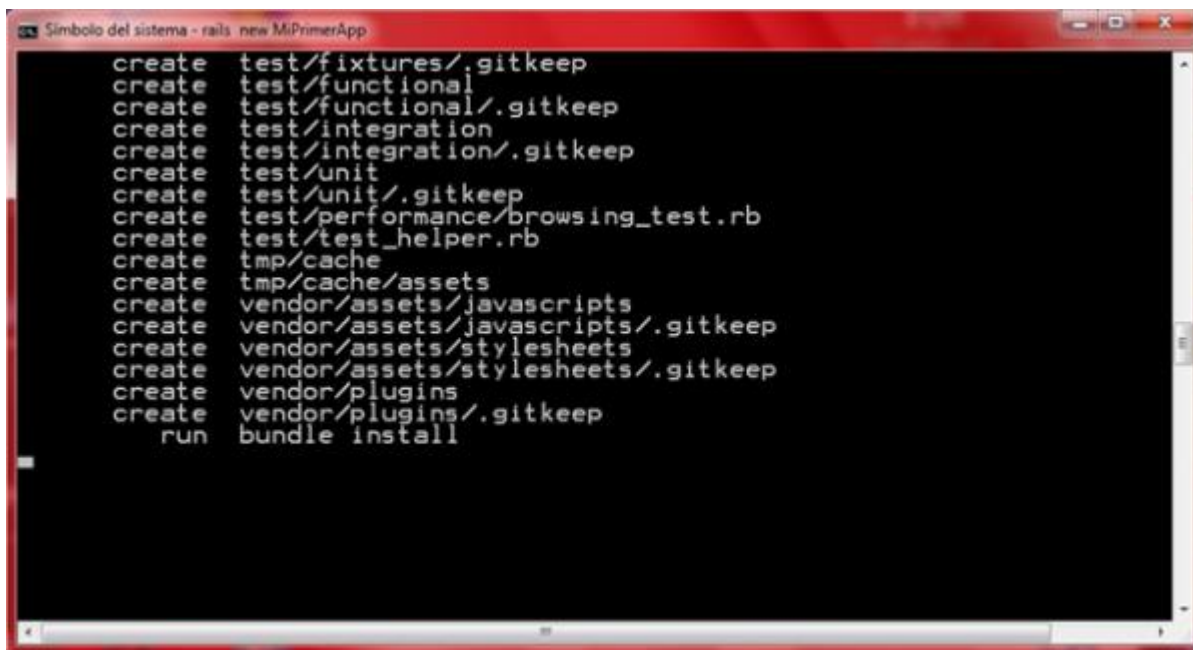


Figura 37. Creando mi primera App



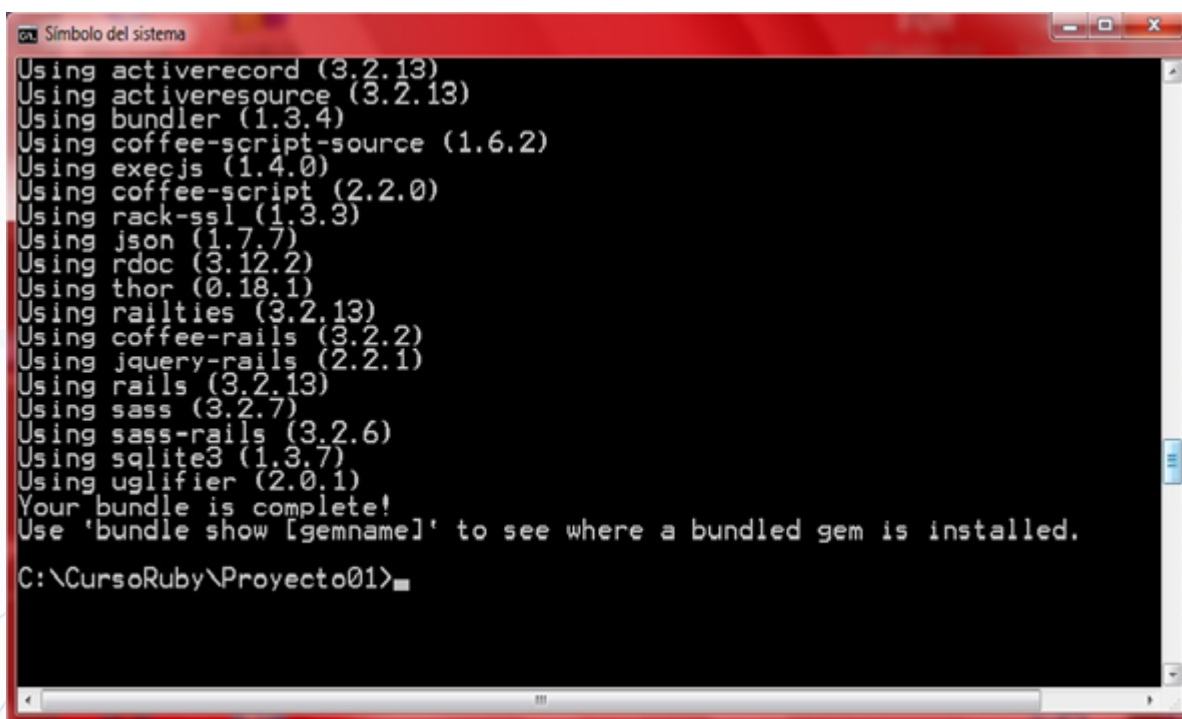
Unidad 1. Fundamentos web

Esto nos puede tomar unos segundos para que genere todos los archivos necesarios para nuestra aplicación.



```
Símbolo del sistema - rails new MiPrimerApp
create  test/fixtures/.gitkeep
create  test/functional
create  test/functional/.gitkeep
create  test/integration
create  test/integration/.gitkeep
create  test/unit
create  test/unit/.gitkeep
create  test/performance/browsing_test.rb
create  test/test_helper.rb
create  tmp/cache
create  tmp/cache/assets
create  vendor/assets/javascripts
create  vendor/assets/javascripts/.gitkeep
create  vendor/assets/stylesheets
create  vendor/assets/stylesheets/.gitkeep
create  vendor/plugins
create  vendor/plugins/.gitkeep
run     bundle install
```

Figura 38. Generación de archivos para una aplicación (1)



```
Símbolo del sistema
Using activerecord (3.2.13)
Using activerecord (3.2.13)
Using bundler (1.3.4)
Using coffee-script-source (1.6.2)
Using execjs (1.4.0)
Using coffee-script (2.2.0)
Using rack-ssl (1.3.3)
Using json (1.7.7)
Using rdoc (3.12.2)
Using thor (0.18.1)
Using railties (3.2.13)
Using coffee-rails (3.2.2)
Using jquery-rails (2.2.1)
Using rails (3.2.13)
Using sass (3.2.7)
Using sass-rails (3.2.6)
Using sqlite3 (1.3.7)
Using uglifier (2.0.1)
Your bundle is complete!
Use 'bundle show [gemname]' to see where a bundled gem is installed.
C:\CursoRuby\Proyecto01>
```

Figura 39. Generación de archivos para una aplicación (2)



Unidad 1. Fundamentos web

Una vez listo abrimos nuestro editor de texto para examinar todas las carpetas y archivos que nos ha creado Rails, esto con el de darnos una idea de cómo es la jerarquía de los ficheros, archivos y/o carpetas que maneja RoR en nuestras aplicaciones.

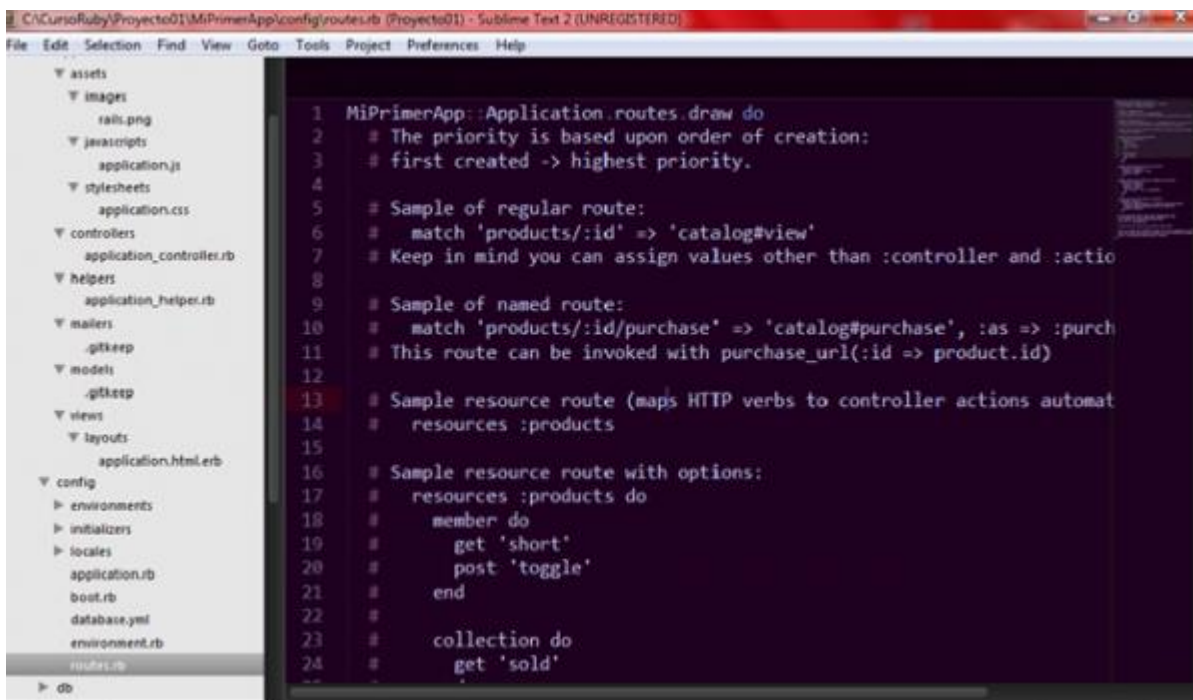


Figura 40. Jerarquía de carpetas y archivos que maneja RoR

Volvemos a la Consola y tecleamos lo siguiente:

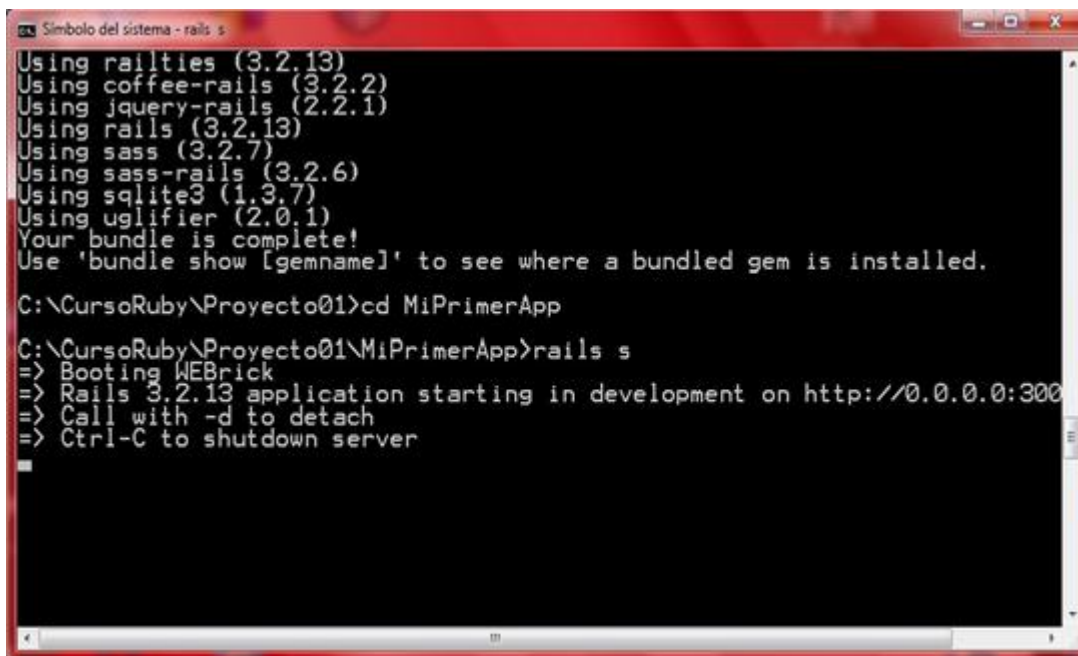
rails s

O también podemos escribir el comando así:

rails server



Unidad 1. Fundamentos web



```
Símbolo del sistema - rails s
Using railties (3.2.13)
Using coffee-rails (3.2.2)
Using jquery-rails (2.2.1)
Using rails (3.2.13)
Using sass (3.2.7)
Using sass-rails (3.2.6)
Using sqlite3 (1.3.7)
Using uglifier (2.0.1)
Your bundle is complete!
Use 'bundle show [gemname]' to see where a bundled gem is installed.

C:\CursoRuby\Proyecto01>cd MiPrimerApp

C:\CursoRuby\Proyecto01\MiPrimerApp>rails s
=> Booting WEBrick
=> Rails 3.2.13 application starting in development on http://0.0.0.0:3000
=> Call with -d to detach
=> Ctrl-C to shutdown server
```

Figura 41. Muestra como cargar la App para visualizarla en un navegador

Lo que hará este comando así de sencillo será cargar nuestra aplicación para poder visualizarla en el navegador, generalmente la carga en el puerto 3000 de nuestro localhost, Ejemplo:

localhost:3000



Unidad 1. Fundamentos web

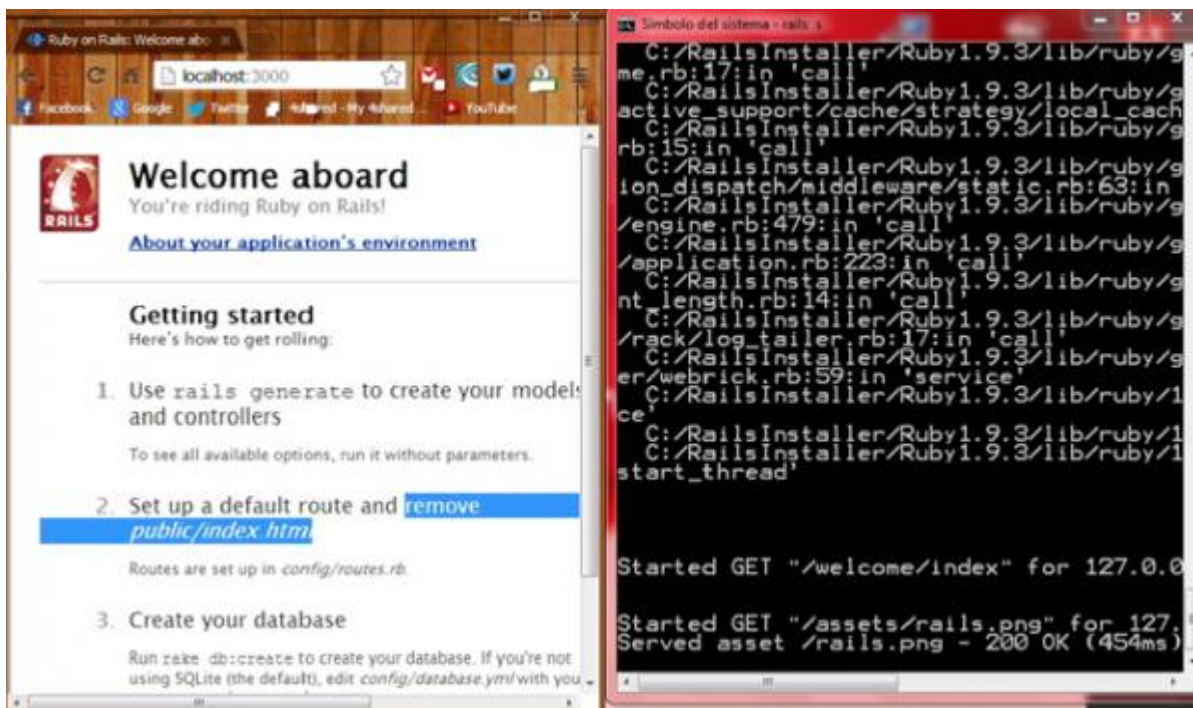


Figura 42. Vista con la portada de Rails al visualizar la aplicación.

Una vez visualizada la aplicación en el navegador nos podremos dar cuenta que por defecto Rails genero una vista con la portada de RoR, en donde nos indica que esta vista o archivo debe ser borrada y reemplazada por la vista índice del controlador welcome.

Muy bien ahora regresamos de nuevo a la consola y nos salimos del servidor mediante la combinación de teclas Ctrl+C.

Después tecleamos lo siguiente:

rails g controller welcome index



Unidad 1. Fundamentos web

```
Simbolo del sistema
C:\CursoRuby\Proyecto01>cd MiPrimerApp
C:\CursoRuby\Proyecto01\MiPrimerApp>rails g controller welcome index
create    app/controllers/welcome_controller.rb
route     get "welcome/index"
invoke    erb
create    app/views/welcome
create    app/views/welcome/index.html.erb
invoke    test_unit
create    test/functional/welcome_controller_test.rb
invoke    helper
create    app/helpers/welcome_helper.rb
invoke    test_unit
create    test/unit/helpers/welcome_helper_test.rb
invoke    assets
invoke    coffee
create    app/assets/javascripts/welcome.js.coffee
invoke    scss
create    app/assets/stylesheets/welcome.css.scss
C:\CursoRuby\Proyecto01\MiPrimerApp>
```

Figura 43. Crear el controlador welcome.

Lo que hará este comando es crearnos el Controlador “Welcome” con la Vista “Index”.

Una vez creados los ficheros, procedemos a borrar el archivo index.html en la Carpeta /Public de nuestra aplicación.

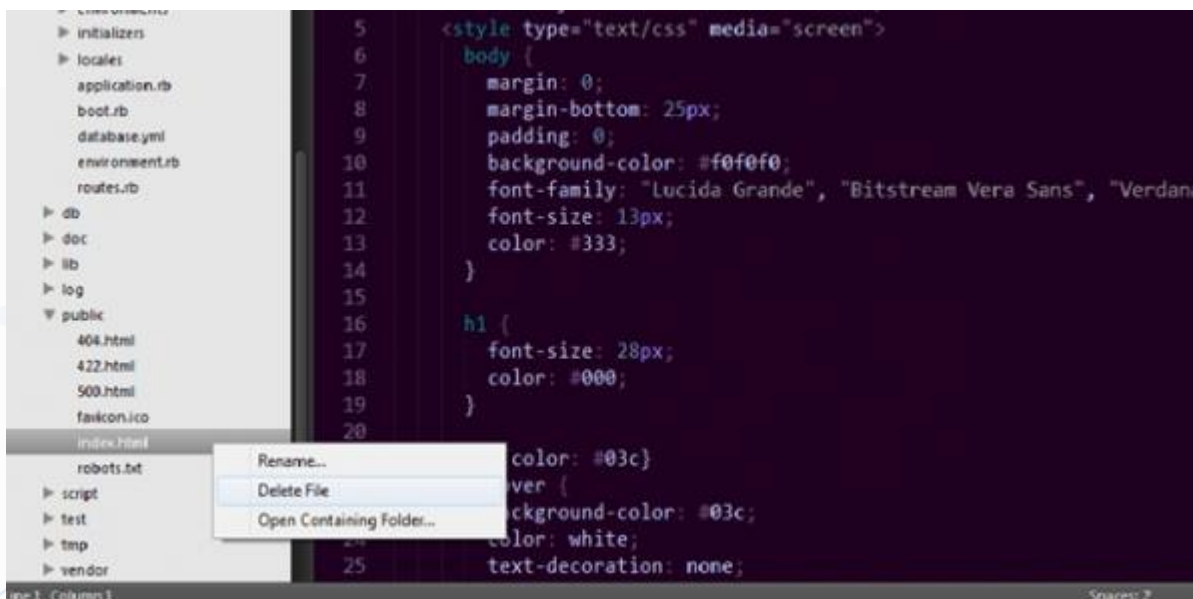


Figura 44. Ejemplo

Luego nos vamos a la carpeta /config y abrimos en nuestro editor de texto el archivo routers.rb en el cual borraremos el “#” de la línea 51 (que nos sirve para



Unidad 1. Fundamentos web

comentar algún texto o código en Ruby), y nos quedará una línea de comandos de Ruby así:

root :to => 'welcome#index'

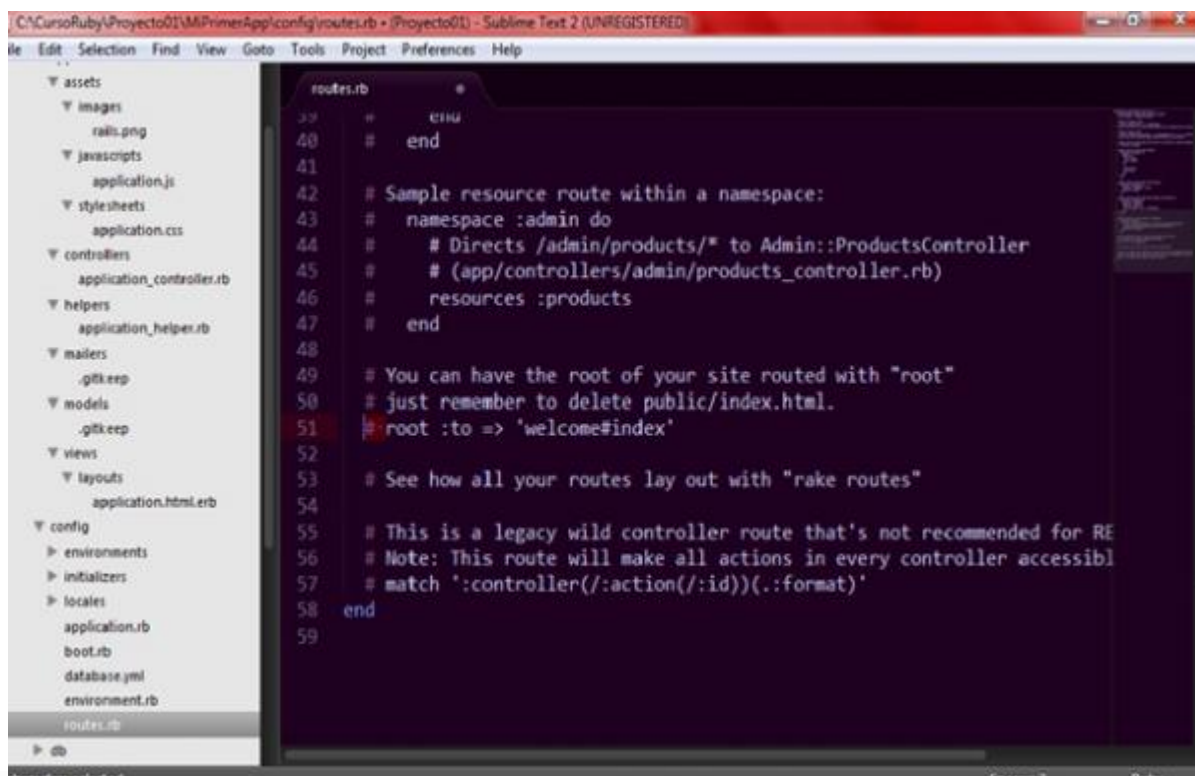


Figura 45. Como comentar líneas de código.

Y ahora para Finalizar nuestra pequeña aplicación de RoR, abrimos el archivo “index.html.erb” que es la Vista “index” de nuestro controlador “Welcome” y colocamos lo siguiente:

<h1> Esta es mi Primer Aplicación en Ruby on Rails </h1>

<p> Hola Mundo!!! </p>

Por último, vamos a cargar nuestra aplicación:

rails s

Qué abrirá un servidor con nuestra app en el puerto 3000 (que ya viene así por defecto).



Unidad 1. Fundamentos web

Abrimos nuestro navegador y visualizamos nuestra aplicación, en:

http://localhost:3000/welcome/index



Figura 46. Ejemplo

Les debe aparecer la Vista en la que creamos la página de HTML5:

<h1> Esta es mi Primer Aplicación en Ruby on Rails </h1>

<p> Hola Mundo!!! </p>



Figura 47. Resultado de una página creada con Ruby on Rails

1.2.4. Lenguaje ASP



Unidad 1. Fundamentos web

El lenguaje ASP cumple una importante función en Internet. Por medio de éste es posible obtener, de forma simple y variada, la información específica que soliciten las organizaciones para la construcción de su sitio web.

Como expresa Rayo (2009) “el desarrollo Web con ASP supuso un gran cambio para los programadores permitiéndoles utilizar un gran número de lenguajes de programación para crear sus aplicaciones. La inclusión de código y controles de servidor permitió ampliar la funcionalidad de las aplicaciones sin requerir un gran conocimiento sobre el lenguaje HTML” (p. 2).

Es posible identificar en el código fuente si está elaborado con ASP por la siguiente etiqueta:

Inicia con: <%y termina con: %>

A continuación, se integra un ejemplo de aplicación de ASP:

En el siguiente ejemplo observarás que ASP se integró con JavaScript y PHP. ASP que trabaja con el servidor web IIS también lo hace con el servidor web Apache, pero son muy pocos los casos donde se utiliza.



Unidad 1. Fundamentos web

```
Sin título: Bloc de notas
Archivo Edición Formato Ver Ayuda

<html>
<head>
<title>visualización de datos enviados por formulario </title>
</head>
<body>
<h4 align="center">Variables recibidas del formulario</h4>
<% for each v_entrada in request.form
for indice=1 to request.form(v_entrada).count
response.write v_entrada&" "&request.form(v_entrada)(indice)&"<br>"
next
next %>
<hr>
<div align="center"><center>
<table border="1" width="44%">
<tr>
<td width="100%"><p align="center"><a href="javascript:window.history.back()">volver a la
pagina anterior</a></td>
</tr>
<tr>
<?php
date_default_timezone_set("America/mexico_city" );
$tiempo = getdate(time());
$día = $tiempo['yday'];
$día_mes=$tiempo['mday'];
$mes = $tiempo['mon'];
$year = $tiempo['year'];
$hora= $tiempo['hours'];
$minutos = $tiempo['minutes'];
$segundos = $tiempo['seconds'];

switch ($día){
case "1": $día_nombre="Lunes"; break;
case "2": $día_nombre="Martes"; break;
case "3": $día_nombre="Miércoles"; break;
case "4": $día_nombre="Jueves"; break;
case "5": $día_nombre="Viernes"; break;
case "6": $día_nombre="Sábado"; break;
case "0": $día_nombre="Domingo"; break;
}
switch($mes){
case "1": $mes_nombre="01"; break;
case "2": $mes_nombre="02"; break;
case "3": $mes_nombre="03"; break;
case "4": $mes_nombre="04"; break;
```

Figura 48. Ejemplo de código lenguaje ASP

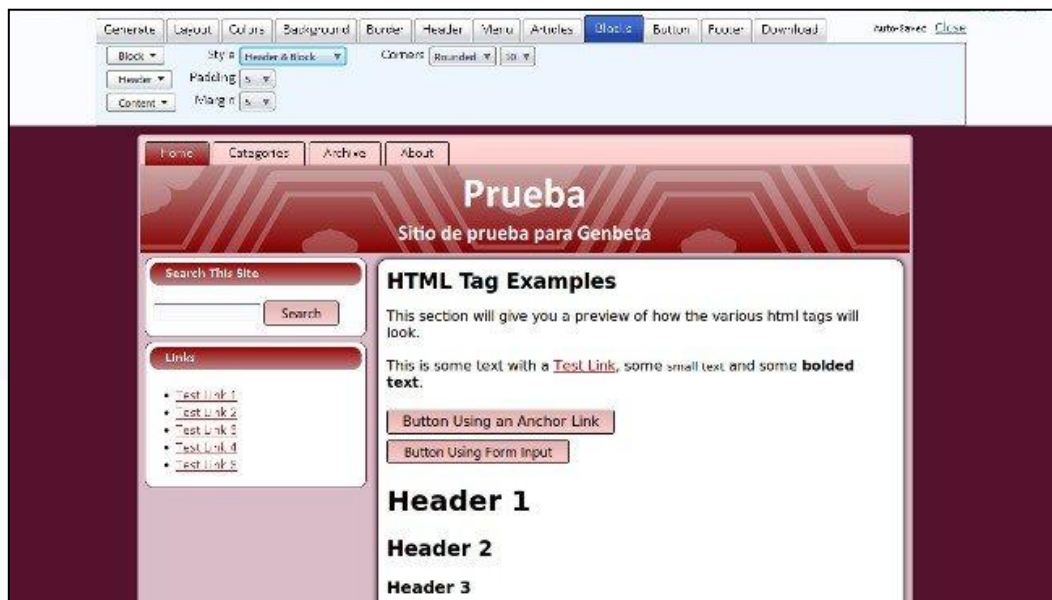


Figura 49. ASP. Fuente: <http://www.genbeta.com/mobile.php/web/crea-plantillas-para-wordpress-desde-cero-con-cool-template>



Unidad 1. Fundamentos web

Por medio del lenguaje ASP podrás elaborar platillas de sitios web y realizar conexiones a bases de datos. Este lenguaje de programación trabaja con el servidor web IIS.

1.2.5. Código JavaScript

En programación web, JavaScript es un lenguaje muy factible ya que permite trabajar con otros lenguajes de programación como por ejemplo PHP, al igual permite realizar animaciones dentro de las páginas web.

JavaScript, según (2011, p. 57) “fue creado por Netscape con el objetivo de integrarse en HTML y facilitar la creación de páginas interactivas. JavaScript funciona como extensión HTML éste lenguaje ha sido diseñado también para el desarrollo de aplicaciones cliente servidor a través de Internet” (p. 57).

Según Minera (2011), “fue creado por Breand Eich estuvo en varios proyectos antes de desarrollar el lenguaje de programación JavaScript, él llegó a Netscape y ahí fue donde desarrollo el lenguaje JavaScript” (p. 283).

Es la tecnología subyacente que permite el uso de programas punteros, como herramientas, juegos y aplicaciones de negocios. Java se ejecuta en más de 850 millones de ordenadores personales de todo el mundo y en miles de millones de dispositivos, como móviles y aparatos de televisión.

Gracias a JavaScript se pueden desarrollar páginas web que se ejecuten en automático en el navegador (cliente), de tal forma que se ejecute determinadas operaciones o se tomen decisiones sin ser necesario acceder al servidor. Por ejemplo, al desarrollar un sitio web y para acceder se solicita una cuenta de usuario con una clave de acceso, JavaScript debe comprobar la información dada por el usuario, verificar que sea correcta y actuar en consecuencia (Orós, 2011).

Un punto importante es que no todos los navegadores pueden soportar JavaScript, por tal motivo, se debe englobar en el código JavaScript con las etiquetas HTML encargadas de introducir comentarios. De esta manera, los



Unidad 1. Fundamentos web

navegadores que no soportan JavaScript, interpretarán el código como un comentario (Orós, 2011).

```
1 <script type="text/javascript">
2 <!--
3 function doubleSubmit(f)
4 {
5     // submit to action in form
6     f.submit();
7     // set second action and submit
8     f.target="_parent";
9     f.action="conex.php";
10    f.submit();
11    return false;
12 }
13 <!-->
14 </script>
```

Mediante este código de JavaScript se está indicando que se deberá generar un formato PDF; a la vez, permite abrir el correo electrónico.

Figura 50. Ejemplo de código JavaScript. Fuente: Minera, 2011.

JavaScript permite realizar diferentes acciones. Se observa que hay un archivo que se llama conex.php. Esa es otra página que está elaborada en PHP, la cual representa que se realizará una tercera acción, que es el almacenamiento en la base de datos, la conectividad con el servidor web Apache donde se almacenará. JavaScript es factible para trabajar con otros lenguajes de programación web.

La siguiente imagen muestra los resultados del código, además de dos acciones que se permitieron realizar con JavaScript. La primera fue generar un formato PDF, y la segunda enviar correos electrónicos.



Unidad 1. Fundamentos web

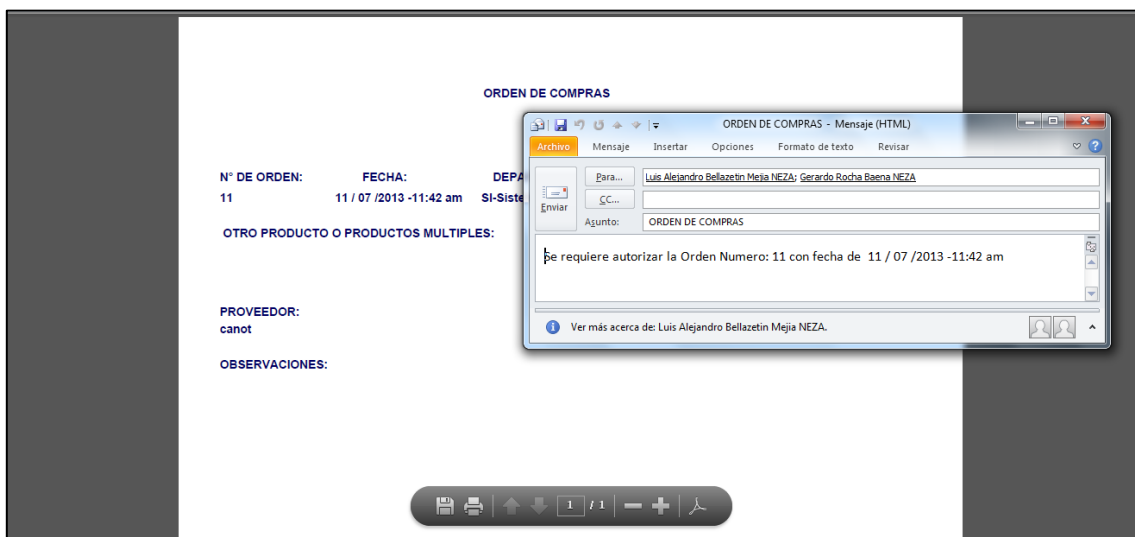


Figura 51. Resultado de la utilización del código JavaScript



Figura 52. Logotipo del lenguaje de programación JavaScript. Fuente: <http://www.planet-source-code.com/vb/2010Redesign/images/LanguageHomePages/JavaScript.png>

JavaScript permite hacer más accesible e interactivo un sitio web. Hace más intuitiva la navegación de los usuarios. Llama más la atención navegar el sitio web con un alto nivel de calidad. Este lenguaje permite realizar animaciones como calendarios interactivos.

En la siguiente imagen se muestra el código fuente de elaboración de un calendario en JavaScript. Éste se utiliza en las páginas web como se identifica que se está elaborado en JavaScript. En el inicio de un programa o código se debe utilizar la etiqueta `<SCRIPT LANGUAGE= "JAVASCRIPT">` y determina el final con `</SCRIPT>`.

Los códigos de JavaScript se colocan en la cabecera o en el cuerpo del documento



Unidad 1. Fundamentos web

(Orós, 2011).

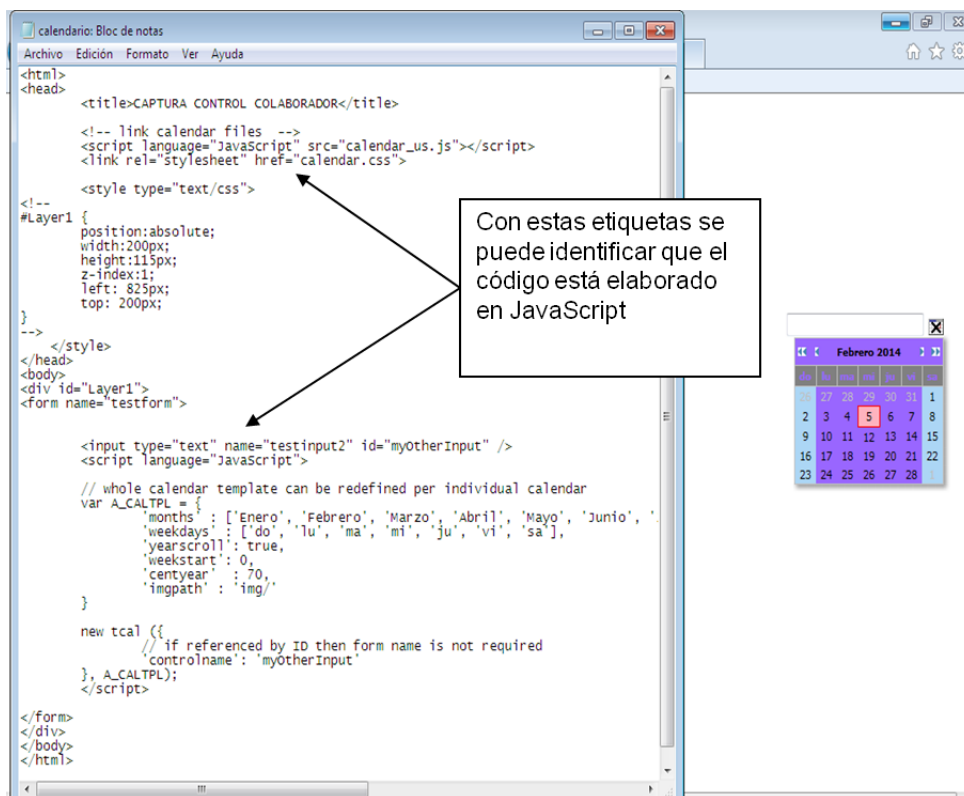


Figura 53. Calendario con JavaScript

A continuación, se expondrá un caso donde se utiliza este lenguaje de programación:

En una empresa que tiene su sitio web, pero quieren agregarle una página donde se permita la entrada a sólo usuarios que cuenten con un *login*. A los que no cuente con él, al intentar ingresar a la página, aparecerá un mensaje diciendo *usuario no autorizado*. También se requiere que tenga un calendario. El programador web propone que la página y el calendario se realicen con JavaScript.

Es muy importante tener presente la definición y funciones de los principales lenguajes de programación web, así como la forma en que se visualiza lo que se está programando; es decir, mediante los navegadores web. Por medio de esta herramienta, a pesar de que aún no se encuentre subida en Internet o Intranet,



Unidad 1. Fundamentos web

conforme se vayan elaborando las páginas se irán visualizando. El lenguaje de HTML es un lenguaje básico que permite escribir páginas web estáticas.

PHP es un lenguaje dinámico que permite interactuar con base de datos sin ningún problema, fue desarrollado en el año de 1994 por Rasmus Lerdorf (PHP, 2014).

La diferencia que hay entre PHP y HTML es que PHP es un lenguaje más dinámico y HTML sólo permite escribir, pero no tiene conexión con base de datos.

EL lenguaje de JavaScript interactúa con HTML y PHP. Se puede hacer infinidad de scripts. Por tal motivo, permite que el sitio web sea más interactivo. Es muy sencillo, permite que una página web realice más acciones y es el hermano menor de Java.

Cierre de la unidad

En esta unidad aprendiste el origen del Internet y en qué año se creó. Por otro lado, Internet es una red mundial que permite que el usuario pueda enviar y recibir información de todo tipo de información. Se compone de sitios web, para su realización se necesita tener conocimiento de qué es un sitio y una página web, los diferentes tipos de páginas que hay, y la importancia de los lenguajes de programación en la dinamicidad de las páginas.

Los lenguajes de programación HTML y CSS, PHP; ASP, Ruby on Rails y JavaScript permiten realizar las diversas acciones de gestión y acceso a la información que proporciona una página web. PHP es un lenguaje dinámico y muy recomendado, ya que permite interactuar con bases de datos. El código JavaScript permite que las páginas web puedan realizar varias acciones a la vez. Es posible, mediante este código, integrar infinidad de scripts así como aplicarse con HTML y PHP y otros lenguajes de programación. La diferencia que tiene PHP y HTML es que el primero interactúa con bases de datos y es necesario



Unidad 1. Fundamentos web

abrirlo desde un navegador web. HTML no trabaja con páginas dinámicas. PHP ha evolucionado y cuenta con varias versiones, la más reciente es la PHP 5.5.

Con estos temas y subtemas podrás analizar qué es lo que se necesita para la construcción de un sitio web, identificar las tecnologías web así como los lenguajes de programación. Mediante éstos podrás realizar lo que solicite una organización y proponer más elementos para un sitio web. También te ayudará a que vayas visualizando tu proyecto que realizarás en las unidades siguientes.

Para saber más

Para saber más sobre los estándares el lenguaje HTML, puedes acceder al sitio web del W3C que es un organismo regulador de la Web.

<https://www.w3c.es/estandares/>

Si deseas saber más sobre CSS puedes consultar el siguiente manual en línea.

<https://www.pdf-manual.es/programacion-web/css/91-css-desde-cero.html>

Si ya has comenzado a programar en PHP y quieres saber más acerca de este lenguaje de programación puedes visitar la página principal de PHP:

- PHP: manual e instructiva del lenguaje de programación. Recuperado de <http://www.php.net/manual/es/index.php>

Si deseas consultar cómo hacer tus propios scripts se recomienda el siguiente libro:

- Eguiluz, J. (2014). *Introducción a JavaScript*. Recuperado de <http://librosweb.es/javascript/>

Para conocer más sobre Ruby on Rails, visita: <https://rubyonrails.org/>

Si te interesa conocer más de ASP. Ingresa al siguiente sitio:

- ASP. Recuperado de <http://www.w3schools.com/asp/>



Unidad 1. Fundamentos web

Navega en el sitio web <https://www.apachefriends.org/es/index.html> para que conozcas la forma de descargar e instalar el servidor web Apache en un equipo de cómputo personal sobre la plataforma Windows.

Para conocer el uso y aplicación del servidor web Apache consulta la página donde se explica este lenguaje.

- Apache. Recuperado de: <http://httpd.apache.org/>

Para saber más sobre la evolución conceptual y tecnológica del hipertexto, así como los tipos de enlaces, se recomienda la siguiente obra:

- Landow, G.P. (2006). *Hipertexto 3.0: teoría crítica y nuevos medios en la era de la globalización*. Barcelona: Paidós.

Para saber más sobre la evolución de la Web y el funcionamiento de la Web semántica, consulta el siguiente texto:

- Márquez, S. (2007). *La Web semántica*. Carolina del Norte: LuluPress.

Fuentes de consulta

Bibliografía básica

- Alonso, J. (2007). *La tercera revolución: comunicación, tecnología y su nomenclatura en inglés*. Coruña: Netbiblo. ISBN: 9788497452144
- Asensio, M. (2011). *Lazos de luz azul: museos y tecnologías 1, 2 y 3.0*. Barcelona: UOC.
- CERN European Organization for Nuclear Research (2014). About CERN. Disponible en <http://home.web.cern.ch/about>
- Cobo, A., Gómez, P., Pérez, D., Rocha, R. (2005). *PHP y MySQL. Tecnologías para el desarrollo de aplicaciones web*. Madrid, España: Díaz de Santos ISBN: 8479787066



Unidad 1. Fundamentos web

- Codina, L. (2009). ¿Web 2.0, Web 3.0 o Web semántica. En: El impacto en los sistemas de información de la Web, I Congreso Internacional de Ciberperiodismo y Web 2.0, Bilbao: Universidad del País Vasco. [En línea] https://www.lluiscodina.com/wp-content/uploads/Web20_WebSemantica2009_Nov2009.pdf
- Criado, L. (2013). Nosotros, los constructores de la Web semántica. Madrid: Amazon.
- Grupo Vértice. (2009). Diseño básico de páginas web en HTML. Málaga: Vértice.
- Javier, P. C. (s.f.). Las tecnologías web. Disponible en <https://tecnoweb2.com/tecnologias-web>
- Lamarca, M. (2013). Aspectos tecnológicos de la Web. Disponible en http://www.hipertexto.info/documentos/web_tecnolog.htm
- LibrosWeb, Introducción a Ruby on Rails. Disponible en <https://makeitrealcamp.gitbook.io/ruby-book/tipos-y-operadores>
- López, J. (2011). Arquitectura cliente-servidor.
- Luján, S (2001). Programación en Internet: clientes web. San Vicente, Alicante: Club Universitario.
- Luján, S. (2002). Programación de aplicaciones web. Historia, principios básicos y clientes web. San Vicente, Alicante: Club Universitario.
- Maciá, F., Mora, F. J., Gil, J.A., et ál. (2008). Administración de los servicios de Internet de la teoría a la práctica. Alicante, España: Universidad de Alicante
- Marín, J. (2010). Web 2.0. Una descripción muy sencilla de los cambios que estamos viviendo. Coruña, España: Netbiblo.
- Mario A. Chávez, Aprendiendo Ruby on Rails.



Unidad 1. Fundamentos web

- Márquez, S. (2007). La Web semántica. Barcelona: UOC.
- Microsoft (2022a). Servidor web IIS. Disponible en [http://technet.microsoft.com/es-es/library/cc753433\(v=ws.10\).aspx](http://technet.microsoft.com/es-es/library/cc753433(v=ws.10).aspx)
- Microsoft (2022b). Flexible Web Server. Disponible en <https://docs.microsoft.com/es-es/webappsserver/deploy-office-web-apps-server>
- Minera, F. (2011). Desarrollador Web. Manuales USERS. Buenos Aires: Fox andina-Gradi S.A.
- Moschovitis, C., Poole, H. W., Pearson, E., Schuyler, T., Senft, T.M., Sisson, M., (2005). The Internet: Biographies. California: ABC CLIO.
- Nafría, I. (2008). Web 2.0: el usuario, el nuevo rey de Internet. Barcelona: Gestión 2000.
- Orós, J. C. (2011). Diseño de páginas web con XHTML, JavaScript y CSS. México: Alfaomega.
- Pastor, J.A. (2011). Tecnologías de la Web semántica. Editorial: UOC.
- PHP (2014). Historia de PHP. Disponible en <https://www.php.net/manual/es/history.php.php>
- RAE. (2013). [Entrada: Semántica]. Disponible en <https://dle.rae.es/sem%C3%A1ntica>
- Rayo, Á. (2009). Aplicación de técnicas Ajax a Asp.net. Ebook: Luarna.
- Rodil, I. (2010). Operaciones auxiliares con tecnologías de la información y la comunicación. Madrid: Paraninfo.
- Rodríguez Sala, J.J. (2003). Introducción a la programación: teoría y práctica. San Vicente, Alicante: Club Universitario.
- Vara, J. M., et ál. (2012). Desarrollo web en entorno servidor. España: Rama.



Unidad 1. Fundamentos web

- Vázquez, P. (2006). Creación de sitios web. Manuales Users. Buenos Aires: Fox Andina-Dalaga
- W3C World Wide Web Consortium (2013). Web Design and Applications. [En línea] <https://www.w3.org/standards/webdesign/>
- Welling, L, y Thomson, L. (2009). Desarrollo web con PHP y MySQL. Madrid: AnayaMultimedia.

Bibliografía complementaria

- Bakken, S., et. al. (2002). Manual PHP. Disponible en http://perun.pmf.uns.ac.rs/budimac/wd/php_manual_en.pdf
- Barry, M. et ál. (1998). Una breve historia de Internet. Revista Novática, (130-131).
- Diccionario web. (s. f.). [Entrada: Navegador].
- Ferrer, J., García, V. y García, R. (s. f.). Curso Completo de HTML. [En línea] <http://es.tldp.org/Manuales-LuCAS/doc-curso-html/doc-curso-html.pdf>
- Juárez, R. (2012). Hábitos de los usuarios de Internet en México.
- WordReference. (2014a). [Entrada: Web]. Disponible en <https://www.wordreference.com/es/translation.asp?tranword=web>
- WordReference. (2014b). [Entrada: Wide]
- WordReference. (2014c). [Entrada: World Wide Web]. Disponible en <https://www.wordreference.com/es/translation.asp?tranword=world%20wide%20web>