



INSTITUTO TECNOLÓGICO SUPERIOR DE POZA RICA

INGENIERÍA SISTEMAS COMPUTACIONALES

Materia

Desarrollo en Internet

Nombre

Javier Alexis Mendoza Gracia

Docente

Simón García Ortiz

Introducción

Como introducción acerca de este tema tenemos que, la tecnología es un conjunto de soluciones que nos permite observar crear Web está basada en el protocolo HTTP para la transmisión de hipertexto sobre protocolos de comunicaciones TCP/IP, sirve para poder proporcionar una nueva visión de internet.

Si antes, con servicios como FTP, la Red se concebía como un conjunto de nodos servidores cuya topología y estructura física había que conocer, ahora se logra un nivel de abstracción mayor: lo que importa es la información, los contenidos en términos del WWW, en lugar de su ubicación, de las características de los servidores en los que se encuentra ubicada o de los dispositivos empleados para acceder a la misma: computadora personal, PDA , WebTV, teléfono móvil, etc.

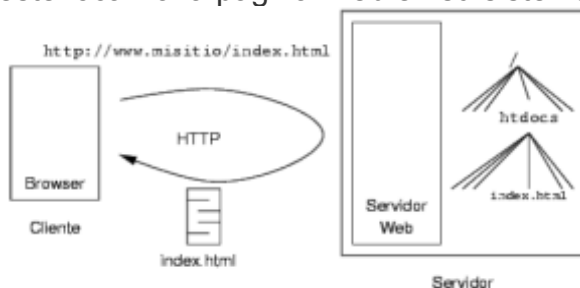
Desarrollo de aplicaciones web

2.1 Arquitectura de las aplicaciones Web.

Una aplicación Web es proporcionada por un servidor Web y utilizada por usuarios que se Conectan desde cualquier punto vía clientes Web (navegadores). La arquitectura de un Sitio Web tiene tres componentes principales:

- Un servidor Web
- Una conexión de red
- Uno o más clientes

El servidor Web distribuye páginas de información formateada a los clientes que las solicitan. Los requerimientos son hechos a través de una conexión de red, y para ello se usa el protocolo HTTP. Una vez que se solicita esta petición mediante el protocolo HTTP y la recibe el servidor Web, éste localiza la página Web en su sistema de archivos y la envía de vuelta al navegador que la solicitó.



Las aplicaciones Web están basadas en el modelo Cliente/Servidor que gestionan servidores web, y que utilizan como interfaz páginas web.

Las páginas Web son el componente principal de una aplicación o sitio Web. Los browsers piden páginas (almacenadas o creadas dinámicamente) con información a los servidores Web. En algunos ambientes de desarrollo de aplicaciones Web, las páginas contienen código HTML y scripts dinámicos, que son ejecutados por el servidor antes de entregar la página.

Una vez que se entrega una página, la conexión entre el browser y el servidor Web se rompe, es decir que la lógica del negocio en el servidor solamente se activa por la ejecución de los scripts de las páginas solicitadas por el browser (en el servidor, no en el cliente). Cuando el browser ejecuta un script en el cliente, éste no tiene acceso directo a los recursos del servidor. Hay otros componentes que no son scripts, como los applets (una aplicación especial que se ejecuta dentro de un navegador) o los componentes ActiveX. Los scripts del cliente son por lo general código JavaScript o VBScript, mezclados con código HTML.

La colección de páginas son en una buena parte dinámicas (ASP, PHP, etc.), y están agrupadas lógicamente para dar un servicio al usuario. El acceso a las páginas está agrupado también en el tiempo (sesión). Los componentes de una aplicación Web son:

1. Lógica de negocio.
 - Parte más importante de la aplicación.
 - Define los procesos que involucran a la aplicación.
 - Conjunto de operaciones requeridas para proveer el servicio.
2. Administración de los datos.
 - Manipulación de BD y archivos.

3. Interfaz

- Los usuarios acceden a través de navegadores, móviles, PDAs, etc.
- Funcionalidad accesible a través del navegador.
- Limitada y dirigida por la aplicación.

Las aplicaciones web se modelan mediante lo que se conoce como modelo de capas, Una capa representa un elemento que procesa o trata información. Los tipos son:

- Modelo de dos capas: La información atraviesa dos capas entre la interfaz y la administración de los datos.
- Modelo de n-capas: La información atraviesa varias capas, el más habitual es el modelo de tres capas.

Modelo de dos Capas.

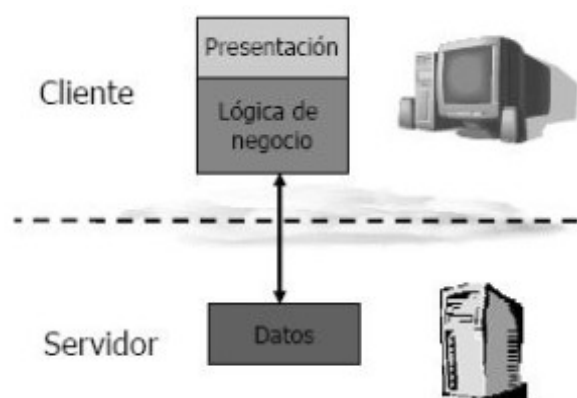
Gran parte de la aplicación corre en el lado del cliente (fat client).

Las capas son:

- Cliente (fat client): La lógica de negocio está inmersa dentro de la aplicación que realiza el interfaz de usuario, en el lado del cliente.
- Servidor: Administra los datos.

Las limitaciones de este modelo son.

- Es difícilmente escalable
- Número de conexiones reducida
- Alta carga de la red.
- La flexibilidad es restringida
- La funcionalidad es limitada.



Modelo de tres Capas.

Esta diseñada para superar las limitaciones de las arquitecturas ajustadas al modelo de dos capas, introduce una capa intermedia (la capa de proceso) Entre presentación y los datos, los procesos pueden ser manejados de forma separada a la interfaz de usuario y a los datos, esta capa intermedia centraliza la lógica de negocio, haciendo la administración más sencilla, los datos se pueden integrar de múltiples fuentes, las aplicaciones web actuales se ajustan a este modelo. Las capas de este modelo son:

1. Capa de presentación (parte en el cliente y parte en el servidor)

- Recoge la información del usuario y la envía al servidor (cliente)

- Manda información a la capa de proceso para su procesamiento
- Recibe los resultados de la capa de proceso
- Generan la presentación
- Visualizan la presentación al usuario (cliente)

2. Capa de proceso (servidor web)

- Recibe la entrada de datos de la capa de presentación
- Interactúa con la capa de datos para realizar operaciones
- Manda los resultados procesados a la capa de presentación

3. Capa de datos (servidor de datos)

- Almacena los datos
- Recupera datos
- Mantiene los datos
- asegura la integridad de los datos

2.1 Los lenguajes de programación del lado del cliente.

Los lenguajes de programación del lado cliente se usan para su integración en páginas web. Un código escrito en un lenguaje de script se incorpora directamente dentro de un código HTML y se ejecuta interpretado, no compilado.

Con la programación del lado del cliente se pueden validar algunos de los datos en la máquina cliente antes de enviarlos al servidor. Esto proporciona a los usuarios informes de error inmediatos, mientras siguen en esa página de formulario y sin necesidad de volver atrás tras recibir un mensaje de error. Puede resultar necesario acceder a una base de datos para validar determinados valores, mientras que no suele disponer de un acceso directo a la base de datos en la máquina del cliente, aunque ese acceso a la base de datos es factible.

Para incorporar un fragmento de código script en una página HTML se introduce el script entre los tags `<SCRIPT>` y `</SCRIPT>`. Dos son los lenguajes de script que hay en la actualidad: el VBScript (derivado de Visual Basic) y el JavaScript (derivado de Java).

Vbscript (Visual Basic Script): Es un lenguaje de script, directamente derivado de Visual Basic. Para insertar código VBScript en una página HTML se añade al tag `<SCRIPT>` el parámetro `LANGUAGE="VBScript"`, que determina cual de los lenguajes de script se utiliza. Decimos que los lenguajes de script se ejecutan interpretados, no compilados. Esto significa que un código escrito en un lenguaje de script no sufre ninguna transformación previa a su ejecución. Cada línea de código es traducida a lenguaje máquina justo antes de su ejecución. Después es ejecutada y la traducción no se conserva en ningún sistema de almacenamiento (como discos, cintas, etc.). Si es necesaria otra ejecución, el intérprete se verá abocado a realizar una nueva traducción de cada línea de código. Sin embargo el lenguaje Visual Basic, del cual deriva el VBScript, es un lenguaje compilado. Esto significa que un código en Visual Basic sufre un proceso global de traducción a lenguaje máquina. Todo el código es traducido de una sola vez y el resultado de esa traducción se almacena en el disco con la extensión .EXE. Cuando llega el momento de la ejecución, se ejecuta el código compilado, no el código original del programa (llamado código nativo o código fuente). Cada sistema tiene sus ventajas e inconvenientes.

Las ventajas e inconvenientes de la interpretación son:

- El código es cómodo para depurar, ya que no es necesario volver a compilar tras un cambio.
- No es necesario disponer de un compilador, ya que el intérprete (que forma parte del navegador) ejecuta el script.
- El mantenimiento es fácil y rápido, por parte del autor o de otro programador.
- La ejecución se ralentiza, al ser necesaria la interpretación línea a línea cada vez.
- El código es visible y puede ser objeto de plagio por parte de otras personas.
- El usuario tiene acceso al código y puede modificarlo, estropeando alguna operación.

Las ventajas e inconvenientes de la compilación son:

- El código compilado se ejecuta muy rápido, al no ser necesaria una traducción cada vez.
- El código compilado no puede ser “abierto” por otras personas. No es necesario transmitir el código fuente.
- El código compilado puede estar, íntegramente, incluido en un solo fichero.
- Es necesario disponer de un compilador-linkador para el proceso de la compilación.
- El código compilado suele ocupar bastante en disco, ya que incorpora en el propio código algunas librerías del sistema.

- Depurar un programa implica volver a compilar tras los cambios.

El código en VBScript puede, además, estar diseñado para su ejecución en el lado del cliente o en el del servidor. La diferencia es que un código que se ejecuta en el lado del servidor no es visible en el lado del cliente. Este recibe los resultados, pero no el código. El código que se debe de ejecutar en el lado del servidor estará incluido en la página web correspondiente entre los tags <% y %>. Además habrá que renombrar la página para aplicarle la extensión .asp (Active Server Page -página activa en servidor-). El funcionamiento intrínseco de la ASP se especificará con mayor detalle en el capítulo 3. Lenguajes del servidor

Un ejemplo de una sencilla página web que incluye código VBScript:

```
<HTML>
<HEAD>
<TITLE>Cuadro de mensaje</TITLE>
</HEAD>
<BODY>
<SCRIPT LANGUAGE = "VBScript">
MSGBOX ("Ejemplo de mensaje")
</SCRIPT>
</BODY>
</HTML>
```

Este código incluye la instrucción MSGBOX ("Cuadro de mensaje"), que genera en pantalla lo siguiente:

JavaScript:

JavaScript es un lenguaje de scripts compacto basado en objetos (y no orientado a objetos). Originariamente era denominado LiveScript, y fue desarrollado por Netscape para su navegador Netscape Navigator 2.0. Fue éste el primer cliente en incorporarlo. Se ejecuta sobre 16 plataformas diferentes, incluyendo los entornos de Microsoft.

Las diferentes versiones de JavaScript han sido finalmente integradas en un estándar denominado ECMAScript-262. Dicho estándar ha sido realizado por la organización ECMA dedicada a la estandarización de información y sistemas de comunicación. Las versiones actuales de los navegadores soportan este estándar.

JavaScript permite la realización de aplicaciones de propósito general a través de la WWW y aunque no está diseñado para el desarrollo de grandes aplicaciones es suficiente para la implementación de aplicaciones WWW completas o interfaces WWW hacia otras más complejas.

Por ejemplo, una aplicación escrita en JavaScript puede ser incrustada en un documento HTML proporcionando un mecanismo para la detección y tratamiento de eventos, como clicks del ratón o validación de entradas realizadas en formularios. Sin existir comunicación a través de la red una página HTML con JavaScript incrustado puede interpretar, y alertar al usuario con una ventana de diálogo, de que las entradas de los formularios no son válidas. O bien realizar algún tipo de acción como ejecutar un fichero de sonido, un applet de Java, etc.

```
<html>
</head>
<body>
<br>
Este es un documento HTML normal
<br>
<script language="JavaScript">
document.write("Esto es JavaScript!")
</script>
<br>
En HTML otra vez.
</body>
```

</html>

Este primer programa se limita a escribir en pantalla un determinado texto para lo que se emplea el código `document.write`. En este código, `document` es un objeto creado por el sistema que hace referencia al propio documento y `write` es uno de los métodos que proporciona para interactuar con él. El resultado de cargar este documento en un visualizador que interprete JavaScript será la aparición de los dos textos, el escrito en JavaScript y el escrito en HTML, sin que el usuario sea consciente del proceso.

Podemos hablar también de páginas dinámicas del servidor, que son reconocidas, interpretadas y ejecutadas por el propio servidor.

2.2 Los lenguajes de programación del lado del servidor.

Los Lenguajes de programación del lado del servidor son especialmente útiles en trabajos que se tiene que acceder a información centralizada, situada en una base de datos en el servidor, y cuando por razones de seguridad los cálculos no se pueden realizar en la computadora del usuario.

Es importante destacar que los lenguajes de programación del lado del servidor son necesarios porque para hacer la mayoría de las aplicaciones web se debe tener acceso a muchos recursos externos a la computadora del cliente, principalmente bases de datos alojadas en servidores de Internet. Un caso claro es un banco: no tiene ningún sentido que el cliente tenga acceso a toda la base de datos, sólo a la información que le concierne.

Las páginas dinámicas del servidor se suelen escribir en el mismo archivo HTML, mezclado con el código HTML, al igual que ocurría en las páginas del cliente. Cuando una página es solicitada por parte de un cliente, el servidor ejecuta los scripts y se genera una página resultado, que solamente contiene código HTML. Este resultado final es el que se envía al cliente y puede ser interpretado sin lugar a errores ni incompatibilidades, puesto que sólo contiene HTML.

Luego es el servidor el que maneja toda la información de las bases de datos y cualquier otro recurso, como imágenes o servidores de correo y luego envía al cliente una página web con los resultados de todas las operaciones. Las ventajas de este tipo de programación son que el cliente no puede ver los scripts, ya que se ejecutan y transforman en HTML antes de enviarlos. Además son independientes del navegador del usuario, ya que el código que reciben es HTML fácilmente interpretable.

Como desventajas se puede señalar que será necesario un servidor más potente y con más capacidades que el necesario para las páginas de cliente. Además, estos servidores podrán soportar menos usuarios concurrentes, porque se requerirá más tiempo de procesamiento para cada uno.

Para escribir páginas dinámicas de servidor existen varios lenguajes.

- **Practical Extraction and Report Language (PERL):** es un lenguaje de programación desarrollado por Larry Wal (lwal at netlabs.com) inspirado en otras herramientas de UNIX. PERL es el lenguaje más utilizado para la creación de programas CGI en los servidores web. Es más rápido que los programas shell script de UNIX, puede leer y escribir ficheros binarios, y puede procesar ficheros grandes. La ventaja más importante de PERL sobre C es que PERL no necesita ser recompilado, es un lenguaje interpretado.
- **Active Server Pages (ASP):** es una tecnología del lado servidor de Microsoft para páginas web generadas dinámicamente, que ha sido comercializada como un anexo a Internet Information Server (IIS). EL estándar ASP permite poder utilizar cualquier lenguaje para la programación, si bien Microsoft favorece que se utilice el VBScript, y en segundo término el JScript (versión del estándar Javascript hecha por Microsoft). En teoría, y mediante la programación, es posible adaptar versiones de lenguajes como Pascal, C y otros, para utilizarlos dentro de páginas ASP.
- **Java Server Pages (JSP):** es la tecnología para generar páginas web de forma dinámica en el servidor, desarrollado por Sun Microsystems, basado en scripts que utilizan una variante del lenguaje Java. La tecnología JSP, o de JavaServer Pages, es una tecnología Java que permite a los programadores generar dinámicamente HTML, XML o algún otro tipo de página web. Esta tecnología permite al código Java y a algunas acciones predefinidas ser embebidas en el contenido estático. La principal ventaja de JSP frente a otros lenguajes es que permite integrarse con clases Java (.class) lo que permite separar en niveles las aplicaciones web, almacenando en clases java las partes que consumen más recursos así como las que requieren más seguridad, y dejando la parte encargada de formatear el documento 'html' en

el archivo jsp. Además Java se caracteriza por ser un lenguaje que puede ejecutarse en cualquier sistema, lo que sumado a jsp le da mucha versatilidad.

- Hipertext Preprocesor (PHP): PHP (acrónimo recursivo de “PHP: Hypertext Preprocessor”, originado inicialmente del nombre PHP Tools, o Personal Home Page Tools) es un lenguaje de programación interpretado. Aunque fue concebido en el tercer trimestre de 1994 por Rasmus Lerdorf no fue hasta el día 8 de Junio de 1995 que fue lanzada la versión 1.0. Se utiliza entre otras cosas para la programación de páginas web activas, y se destaca por su capacidad de mezclarse con el código HTML. Aunque su creación y desarrollo se da en el ámbito de los sistemas libres, bajo la licencia GNU, existe además un compilador comercial denominado (Zend Optimizer). El desarrollo del código abierto por un lado y el firme respaldo comercial por el otro, han elevado a PHP a alcanzar una popularidad más que notable.

Conclusión

En la actualidad nos damos cuenta de que hablar de tecnologías web engloba muchos campos y que sus áreas son diversas y complejas, desde solo consultar información con un navegador ya sea Firefox, safari, opera, chrome, buscar una imagen, consultar sitios de empresas, hacer compras electrónicas, jugar, conocer lugares en 3D, así como consultar datos estadísticos, bases de datos, sitios de gobierno, etc., etc... Las tecnologías juegan un papel muy importante en la web, debido a estas herramientas podemos acceder al internet, que no es solo sentarte frente a un ordenador y abrir el internet sino que hay un gran trabajo detrás para hacer posible este tipo de tecnología, donde miles de personas trabajan actualizando datos, creando los métodos para mantener todos los enlaces y las comunicaciones.

Referencias

- Manuel Montenegro.
- (2018).
- Introducción a las tecnologías Web.
- 26/06/2018,
- Instituto Tecnológico de Matehuala Sitio web: <https://programacionwebisc.wordpress.com/1-2-2-urls>