

CAPITULO 3.
TECNOLOGÍAS DE DESARROLLO PARA CONTENIDO DINÁMICO E
INTERACTIVO EN WEB

Actualmente estamos presenciando páginas en Web con contenido dinámico, ya no es común ver en los sitios de Internet desde la fecha y hora actual, el tipo de cambio, noticias, el estado del tiempo y hasta podemos verificar el marcador de una partido de fútbol, de basketball o algún otro deporte al mismo tiempo que se esta jugando. Ésta información en algunas ocasiones ya se ha podido automatizar a través de pequeños programas que realizan la sustitución de los datos al código HTML (HyperText Markup Language) la causa es que sería demasiado tedioso estar cambiando manualmente cada página, ya siendo directamente al código HTML o en un editor de páginas Web.

Éste capítulo presenta una síntesis de las tecnologías de desarrollo para contenido dinámico e interactivo en Web a través del tiempo hasta nuestros días, sus características, la tecnología escogida para este proyecto, así como su justificación, mostrando sus ventajas, desventajas, portabilidad y viabilidad para el desarrollo de sitios interactivos.

3.1 Aplicación Web

La Web se diseñó originalmente como un medio para suministrar páginas estáticas a los usuarios de Internet, utilizando el Protocolo de Transferencia de Hipertexto (HTTP) como su principal protocolo de transporte. Cuando un navegador Web envía una consulta HTTP a un servidor Web, este último extrae un archivo de consulta de su sistema de archivos y lo devuelve al navegador a través de la conexión HTTP. Sin embargo, lo que devuelve el servidor Web no tiene por qué ser siempre un archivo estático almacenado en el servidor. Puede tratarse de la salida de un programa. En otras palabras, el servidor Web puede actuar, junto con un programa, por ejemplo, de recuperación de base de datos, como un programa de aplicación que responde a consultas HTTP. Por lo tanto, puede decirse que se trata de una aplicación Web.

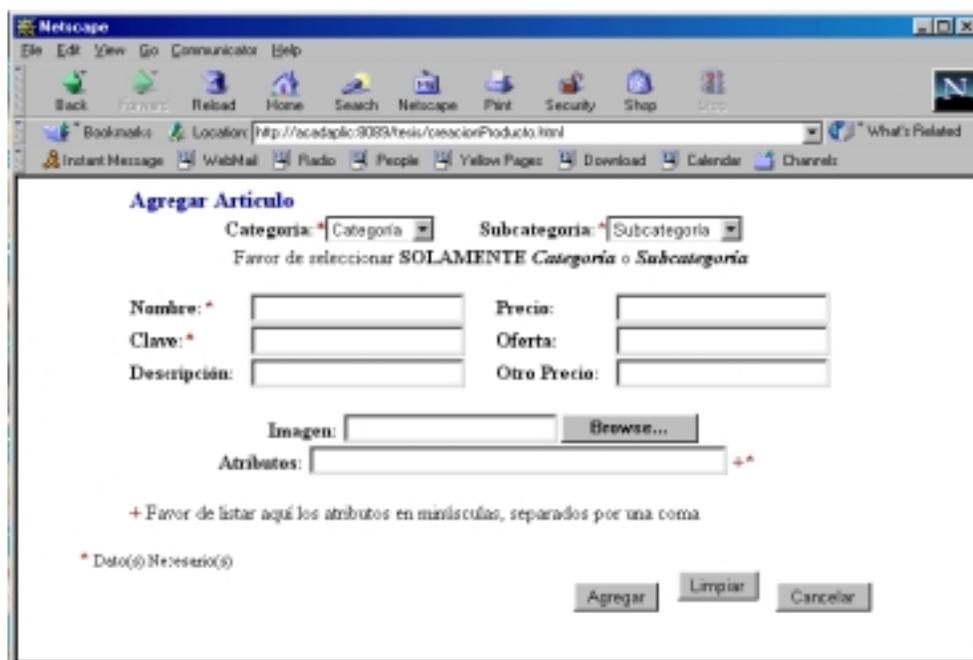
Capítulo 7: Resultados y Trabajo a Futuro

Una forma o formulario HTML es capaz de *establecer contacto y comunicación* a través de la Web para el procesamiento de información y generar interactividad con el usuario.

Es importante mencionar que estas formas no son capaces de procesar los datos, cuentan con métodos que realizan el manejo de sus entradas (input en inglés) estas presentadas en un documento HTML.

Las formas generan una interfaz de usuario que le permite recoger datos de la página Web, luego estos datos son enviados al servidor de Web y este se encarga de pasarle los datos a la aplicación utilizada. Usando formas HTML uno puede crear entradas de:

- Cuadros de texto de una línea (text field)
- Áreas de texto (text area)
- Casillas de verificación (check boxes)
- Botones de opción (radio buttons)
- Menus desplegables (flow options & list {multiple} boxes)
- Botones de Comando (submit & reset buttons)
- Selector de Archivos (input file)



The image shows a Netscape browser window displaying an HTML form titled "Agregar Artículo". The form is contained within a frame and includes the following elements:

- Two dropdown menus for "Categoría" and "Subcategoría".
- A note: "Favor de seleccionar SOLAMENTE Categoría o Subcategoría".
- Text input fields for "Nombre:", "Clave:", "Descripción:", "Precio:", "Oferta:", and "Otro Precio:".
- An "Imagen:" field with a "Browse..." button.
- An "Atributos:" text input field with a "+" sign.
- A note: "+ Favor de listar aquí los atributos en minúsculas, separados por una coma".
- A note: "* Datos Necesarios(*)".
- Buttons for "Agregar", "Limpiar", and "Cancelar" at the bottom.

Figura 3.1. Ejemplo de una forma HTML

Cuando el usuario procesa una forma, se envía el nombre(s) y contenido(s) de cada campo de ésta al controlador del formulario como conjunto de pares valor- nombre. La manera en que el controlador procesa una forma depende del diseño del controlador.

Entre los atributos de una forma se encuentran *method* y *action* que establecen el comportamiento que tendrá la forma. Con *method* se define si la forma va obtener datos va a agregarlos, *get* y *post* respectivamente, y *action* es la acción asociada a tal método. A continuación se muestran las tecnologías de controladores de formas HTML que existen, sus características así como una pequeña descripción de su funcionamiento.

3.2 Common Gateway Interface (CGI)

Las aplicaciones CGI fueron una de las primeras maneras prácticas de crear contenido dinámico para las páginas Web. En el marco de la interfaz CGI el servidor Web pasa las solicitudes del cliente a un programa externo. La salida de dicho programa es enviada al cliente en lugar del archivo estático tradicional. El advenimiento de CGI ha hecho posible la implementación de nuevas y variadas funcionalidades a las páginas Web, de tal manera que esta interfaz rápidamente se volvió un estándar, siendo implementada en todo tipo de servidores Web.

El CGI es un mecanismo (estándar) para extender la funcionalidad del servidor Web mediante la ejecución de programas o secuencias de comandos en el servidor Web como respuesta a peticiones del Browser [Till, 1996]. Una manera común de utilizar CGI es en el proceso de enviar formas, donde el explorador envía los datos a la aplicación CGI que reside en el servidor y éste se encarga de procesar los datos para generar una respuesta, que debe ser una página HTML. La programación de aplicaciones CGI involucra diseñar y escribir programas que reciben sus argumentos -generalmente- de una página Web. Estos programas son escritos en C, C++, Perl entre otros lenguajes.

3.2.1 Desventajas

La creación de contenido dinámico a través de CGI implica que el diseño de la página Web no sea modificado o alterado, es decir ya cuenta con una estructura definida, organizada de tal manera que solamente los datos pueden ser cambiados. El problema con los programas en CGI radica cuando las solicitudes por parte de los clientes son demasiadas para el servidor, por consiguiente el tiempo de respuesta aumenta causando tardía la creación de la página de respuesta y perjudica el *performance* del servidor.

Otro problema es que un CGI es *stateless*, ya que no conoce ni maneja ni mucho menos implementa el concepto de sesión, esto es, que el CGI sólo ejecuta lo que se le ordena y una vez que el programa o acción se han llevado acabo el CGI desaparece y debe ser llamado otra vez en caso de requerir el dato o acción una vez más; esto hace el control de sesiones demasiado pesado para un programador. Por último el mayor problema es que la lógica de desarrollo esta incluida en el programa que está encargado de creación de la página, de está manera es casi imposible cambiar el diseño de la página, debido a que la información necesita concordar con la presentación.

3.2.2 FastCGI

Existe otra opción que es el FastCGI es una alternativa al CGI estándar, cuya diferencia radica principalmente en el hecho que el servidor crea un único proceso persistente por cada programa FastCGI en lugar de por cada solicitud del cliente.

Aunque FastCGI es un paso en la dirección correcta, sigue teniendo problemas con la proliferación de procesos, ya que en todo momento existe al menos un proceso activo por cada programa FastCGI. Para manejar solicitudes concurrentes, habría que mantener un estanque de procesos, uno por cada solicitud. Considerando que cada uno de estos procesos puede estar ejecutando el intérprete de Perl, este modelo no parece tan distinto del CGI estándar. Una solución que FastCGI ofrece para este problema es su habilidad para distribuir todos estos procesos entre múltiples servidores.

3.3 JavaScript

Es un lenguaje de script, interpretado, orientado a objeto, que permite introducir interactividad en documentos HTML y tiene la ventaja de que no necesita ser transmitido hacia el servidor, verificado y devuelto. Éste es ejecutado en el navegador del usuario.

Los scripts de Javascript pueden ser introducidos dentro de sus páginas de HTML. Con Javascript se puede dar respuesta a eventos iniciados por el usuario, eventos tales como la entrada de una forma o algún enlace. Las entradas son verificadas por la aplicación cliente y pueden ser transmitidas después de esto. Permite efectuar cálculos, efectos especiales, verificar formas, crear juegos, personalizar la gráfica, crear password de seguridad, y mucho más. Realiza el manejo de muchos recursos Web.

3.3.1 Características

A continuación se muestra una tabla comparativa de JavaScript contra Applets y CGI

Tabla 3.1 Comparación entre Applet y JavaScript

<i>JavaScript</i>	<i>Applet</i>
Interpretado (no compilado) por el cliente	Cargado en el servidor antes de ejecutarse en el cliente
Basado en Objetos	El código consiste en clases con herencia
Código integrado e incrustado en HTML	Los applets son llamados desde una página HTML.
Tipos de variables no declarados	Los argumentos deben ser de un tipo específico

Tabla 3.2 Comparación entre CGI y JavaScript

<i>JavaScript</i>	<i>CGI</i>
Es un lenguaje de Script	Es una interfaz. Da libertad de elección del lenguaje
No requiere un servidor de Web	Exige la presencia de un servidor Web
La aplicación reside en el cliente	La aplicación reside en el servidor
Requiere un cliente especial (navegador)	Sirve cualquier cliente, por simple que sea

3.3.2 Desventajas

Las desventajas que tiene JavaScript, entre otras, son:

- no permite escribir aplicaciones autónomas sin la presencia de un intérprete,
- tiene una capacidad muy limitada para leer y escribir ficheros,
- depende de que el navegador tenga el intérprete y cuente con la versión adecuada, y
- como no tiene contacto con el servidor no puede tener acceso a base de datos o a una aplicación

3.4 Applet

Un applet es un programa que puede ser ejecutado en la Internet. En términos generales, un applet es una aplicación programada en Java, que emplea la tecnología Cliente/Servidor para realizar animaciones interactivas, cálculos inmediatos, juegos, u otras tareas sin la necesidad de una comunicación continua entre el servidor (donde reside el programa) y los clientes (aquellos que lo ejecutan). [Whatis, 2000]

En adición a ser ejecutado en el cliente en vez del servidor, este es generado dinámicamente en el cliente, claro que si este requiere comunicación con el servidor lo puede tener, transfiere los datos, pero no es necesario cargar el applet de nuevo al cliente, este ya se encuentra residente en memoria.

3.4.1 Ventajas

La gran ventaja del applet sobre el CGI es que la creación de contenido dinámico puede ser independiente del diseño de la página Web, y sobre todo si es contenido si no puede ser representado a través de hipertextos, tales como animaciones, gráficas, también cuenta con cuadros de diálogo, frames, paneles, etc. También es muy práctico para juegos, chats en donde la interacción con el usuario es en tiempo real, pero sin la necesidad de que se actualice o vuelva a cargar la página.

3.4.2 Desventajas

La desventaja con el applet, es que depende de que:

- el navegador del cliente tenga el Plug-in o la versión correcta de VM de Java,
- que el navegador lo pueda visualizar correctamente, y
- a veces toma demasiados recursos de la máquina, reduciendo el performance

Y al igual que el CGI contiene la lógica del programa encapsulada, no separa la lógica de la interfaz o la presentación.

3.5 Java Servlet

Los Servlets son la respuesta de la tecnología Java a la programación CGI. Son programas que se ejecutan en un servidor Web y construyen páginas Web dinámicamente. [Sun, 2000]

3.5.1 Ventajas

Portabilidad

- Debido a su escritura en Java, los servlets pueden ser usados en todos los sistemas operativos y todos los servidores Web, habilitados con el JDK, debido a que el código en Java bytecode es independiente de plataforma.
- A diferencia de los Applets, los servlets corren en el servidor local y no en la máquina que ejecuta el servicio.

Extensión

El uso del API de Java le permite a la tecnología servlet el acceso al JavaMail para leer y enviar e-mails, al Java DataBase Connectivity (JDBC) para acceso de

bases de datos, al File class y otras clases relacionadas con el acceso al file system, interacción con RMI, CORBA y los Enterprise Java Beans (EJB) para acceder objetos remotos.

Eficiencia

Una vez cargado el servlet éste reside en la memoria del servidor como una instancia del objeto y cada vez que se genera una requisición se invoca solamente al método correspondiente.

Seguridad

Entre otras cosas podemos mencionar la propiedad de asignar al servlet autoridad en cierta área del servidor, en donde a criterio del administrador y sólo de esa manera el servlet puede tener autorización para acceder o no áreas del sistema (sandboxing)

Extensibilidad

Mediante el manejo de HTTPServlets se pueden generar completos en Internet mediante la interacción con tecnología de creación de páginas como son las Java Server Pages (JSP), JavaScript, HTML, DHTML, XML, etc.

Un servlet es una clase en Java que necesita ser ejecutada como las demás clases en la Java VM en un servicio que llamaremos el motor del servlet. El motor del servlet carga la clase la primera vez que el servlet es solicitado y este se mantiene recibiendo múltiples peticiones hasta que es explícitamente descargado o el motor del servlet es dado de baja [Casares, 1999]. Algunos servidores Web como el Sun's Java Web Server (JWS), W3C's Jigsaw, el Gefion Software's LiteWebServer y Netscape's i-Planet y el Netscape's Enterprise Server están implementados en Java y tienen montado el motor del servlet. Otros como el Microsoft's Internet Information Server (IIS) y The Apache Group's Apache requieren un módulo extra para el manejo eficiente de servlets (Tomcat).

En Macintosh la implementación MacOS X Server también los soporta, pero solo en MacOS X

3.5.2 Características

Existen 3 métodos que definen el ciclo de vida del servlet: *init*, *service* y *destroy*

Init

```
public void init(ServletConfig config) throws ServletException
```

Este método es llamado una vez que el servlet es cargado en el motor del servlet, antes de que el servlet sea requerido para su primera requisición.

Service

```
public void service(ServletRequest request, ServletResponse response) throws ServletException, IOException
```

Este método es llamado para procesar la requisición, puede ser llamado ninguna o muchas veces hasta que el servlet sea descargado. Múltiples threads (uno por solicitud) pueden ejecutar este método en paralelo.

Destroy

```
public void destroy()
```

Llamado una vez que el servlet ha sido descargado y puesto fuera de servicio.

Los servlets resuelven dos de los problemas asociados con el CGI, el primero es el performance. Un servlet es cargado una sola vez, cuando se levanta el servidor o cuando el primer *request* de éste es solicitado o en algunos casos, cuando una nueva versión del servlet es presentado, es decir, cuando se modifica o se actualiza. El segundo es el manejo de sesiones. El motor del servlet automáticamente genera un identificador único para cada usuario. Por último la creación de la página y el desarrollo del programa depende del programador, en este caso la generación dinámica de información esta contenida dentro del servlet, al igual que el CGI.

3.5.3 Desventajas

La desventaja principal de los Servlets es que la lógica de programación está contenida en él, aunque contiene asegurado el código, no es independiente de la interfaz.

3.6 Active Server Pages (ASP)

Las ASP son un ambiente de aplicación abierto en el que se puede combinar código HTML, scripts y componentes ActiveX del servidor para crear soluciones dinámicas y poderosas para el Web [MS, 2000]. Microsoft introdujo esta tecnología llamada Active Server Pages en diciembre de 1996 y es parte del Internet Information Server (IIS).

El principio de la tecnología ASP es el VBScript, pero existe otra diversidad de lenguajes de programación que pueden ser utilizados como lo es Perl, JScript, etc. El ASP es una tecnología dinámica funcionando del lado del servidor, lo que significa que cuando el usuario solicita un documento ASP, las instrucciones de programación dentro del script son ejecutadas para enviar al navegador únicamente el código HTML resultante. La ventaja principal de las tecnologías dependientes del servidor radica en la seguridad que tiene el programador sobre su código, ya que éste se encuentra únicamente en los archivos del servidor que al ser solicitado a través del Web, es ejecutado, por lo que los usuarios no tienen acceso más que a la página resultante en su navegador. La siguiente figura presenta mejor el funcionamiento del ASP



Figura 3.2 Funcionamiento de un ASP

Sus funciones principales están el acceso a base de datos, envío de correo electrónico, creación dinámica de gráficos y otros, básicamente, muchas cosas que se pueden realizar por medio de CGI pueden ser realizadas con esta tecnología.

Esto debido a que el ASP es tan eficiente con escribir código directamente a la interfase de aplicación del servidor, con la ventaja de que es más eficiente que el CGI que depende de un compilador ya que el ASP corre como un servicio en el servidor, tomando ventaja de la arquitectura de multitareas.

El único inconveniente con las ASP es que solamente trabajan sobre plataformas Windows 9x, NT y Linux con las aplicaciones correspondientes y sus lenguajes de programación son limitados: VBScript, Perl, JScript y ActiveX.

3.7 JavaServer Pages (JSP)

JavaServer Pages (JSP) es una tecnología que nos permite mezclar HTML estático con HTML generado dinámicamente. Muchas páginas Web que están construidas con programas CGI son casi estáticas, con la parte dinámica limitada a muy pocas localizaciones. Pero muchas variaciones CGI, incluyendo los servlets, hacen que generemos la página completa mediante nuestro programa, incluso aunque la mayoría de

Capítulo 7: Resultados y Trabajo a Futuro

ella sea siempre lo mismo. JSP nos permite crear dos partes de forma separada.

La introducción de tags especiales al igual que ASP para la inserción de código especial procesado por el servidor Web y la creación dinámica de páginas Web así como la separación lógica del programa y de la presentación es lo que hacen a JSP el candidato elegido para esta tesis, pues no importa cual sería el diseño de la página Web haciéndolo independiente de este completamente, sin importar que se pueda cambiar el diseño desde un editor HTML, pues los tags quedarán incrustados en ella y hasta pueden ser cambiados de lugar, color, tamaño, etc. El control de sesiones, la seguridad de que el código no será presentado y la portabilidad de que puede ser soportado por cualquier plataforma es lo que hace a JSP el mejor candidato, así como la oportunidad de utilizar componentes reutilizables y fácilmente sustituibles y claro la superioridad sobre las demás tecnologías de desarrollo de contenido dinámico

A continuación se muestra una tabla comparativa entre JSP y ASP

Tabla 3.3 ASP vs. JSP

Tecnología	JSP	ASP
Plataformas	Cualquier plataforma que sea soportada por Java (Solaris, Linux, Windows NT, MacOS, AIX, HP-UX y variantes de UNIX)	Microsoft Windows 9x o NT
Servidor Web	Cualquiera. Los más populares como Apache, Netscape, IIS.	Microsoft IIS o Personal Web Server
Language Scripting	Java	VBScript, JavaScript, Perl
Componentes rehusales y de cross plataforma	JavaBeans, Enterprise JavaBeans	No
Protección de memoria de escape	Si	No

Capítulo 7: Resultados y Trabajo a Futuro

Etiquetas especiales	Si	No
Integración de bases de datos	Cualquier Base de Datos que soporte tecnología JDBC u ODBC	Cualquier Base de Datos que soporte tecnología ODBC
Interpretación de código	Una sola vez	En cada caso
Componentes	JavaBeans, Enterprise JavaBeans Extensiones JSP	COM / DCOM

[Tremblett, 2000]

Además otra gran ventaja de JSP es que es una tecnología que es gratuita, es decir no tiene costo alguno, puede ser obtenido completamente gratis a través de la página de JSP

<http://java.sun.com/products/jsp/>

El funcionamiento de los JSP es mejor que los ASP por su arquitectura, pues esta no solo es un procesamiento continuo del código JSP dentro de las páginas HTML también tiene otras funcionalidades que los hacen más práctico y poderoso. La primera vez que un JSP es cargado por el contenedor JSP (**JPS container** o también llamado **JSP engine**) el código del servlet que ejecuta el JSP es generado, compilado y agregado automáticamente al contenedor del servlet (**servlet container**). Mientras el JSP no sea modificado, el servlet genera presenta la respuesta de cualquier petición del navegador del JSP requerido. Si el JSP es modificado el servlet correspondiente es recompilado y agregado nuevamente en el contenedor (véase fig. 3.3).

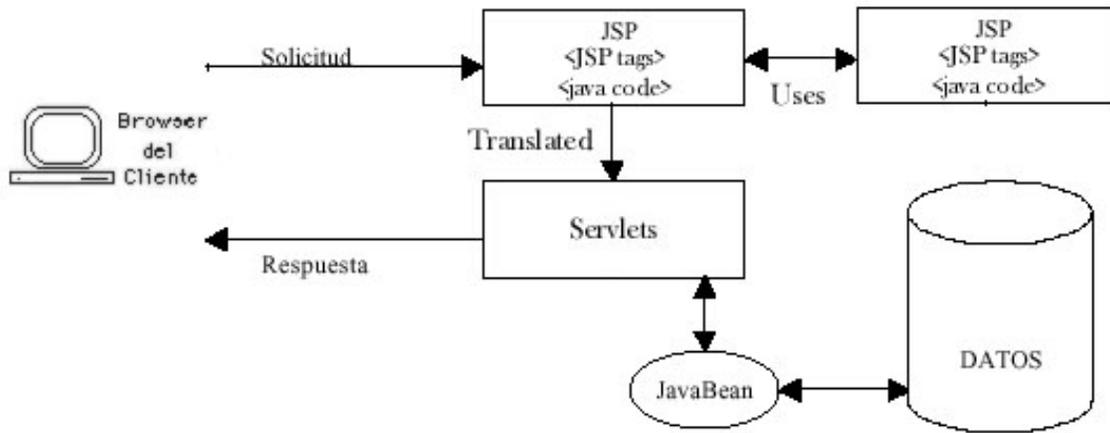


Figura 3.3 Arquitectura de los JSP

La figura arriba muestra como también los JSP pueden invocar dinámicamente otros JSP internamente, pero la funcionalidad más significativamente se muestra en la capacidad de la utilización de los componentes JavaBeans, los cuales como ya fueron mencionados, son componentes reutilizables, capaces de ser modificados, ya sea corregidos o aumentando su actividad.

3.8 Conclusiones

Después de esta investigación sobre las tecnologías de desarrollo de contenido dinámico e interactivo, podemos decir justificadamente que la tecnología JavaServer Pages es lo suficientemente competitiva, viable y segura para realizar el presente proyecto, así como plataforma que asegura el mantenimiento y actualización de los sistemas.

Los JSP's resuelve cada uno de los problemas que presenta el CGI y los ASP's:

- el manejo de Sesiones,
- inserción de código a páginas HTML directamente,
- separación de la lógica de programación y la interfaz,

Capítulo 7: Resultados y Trabajo a Futuro

- ventaja de ser soportado por diferentes plataformas, y
- la utilización de componentes reutilizables de software (JavaBeans),

esta última característica se torna la razón más significativa por lo que se seleccionó la tecnología JSP sin mayor discusión; sin mencionar que el lenguaje en cuestión es Java que le da la oportunidad de expansión y sencillez.

En el presente y los anteriores capítulos se habló sobre el marco teórico de éste trabajo de tesis, en los siguientes capítulos se mostrarán las cuestiones técnicas y de implementación y desempeño del sistema aquí planteado.