

Apuntes
de
Introducción a Internet

Tema 5:
Tecnologías Web

Uploaded by

Ingteleco

<http://ingteleco.webcindario.com>

ingtelecowed@hotmail.com

La dirección URL puede sufrir modificaciones en el futuro. Si no funciona contacta por email

5. Tecnologías actuales para aplicaciones web

5.1 Introducción

Gracias a HTML es posible diseñar un gran número de páginas web que proporcionen información de todo tipo a los usuarios y, de este modo, conseguir que visiten más una determinada sede web.

Sin embargo HTML tiene una limitación insalvable: **su naturaleza estática**. Una página HTML no varía desde que el diseñador la maquetó y la dejó en la web, a menos que el propio diseñador actualice manualmente su contenido cada cierto tiempo.

Ejemplo: supongamos que una tienda de deportes permite que sus clientes puedan consultar los datos de los productos a través de la web. Si las páginas fuesen HTML estático, deberían modificarse cada vez que los datos de un producto varían (para que una empresa sea competitiva y se adapte al mercado debe ser flexible según la necesidad). Ello implica muchas horas de desarrollo de los encargados del mantenimiento de la información en la web.

Una alternativa más avanzada sería disponer de ciertos procesos capaces de acceder a la base de datos de los productos, extraer de ahí la información, y generar la página web con dicha información, cada vez que el usuario la solicita. De este modo la página web no existía originalmente, sino que un programa la ha generado dinámicamente sobre la marcha (no confundir con el concepto de HTML dinámico), a petición de un usuario.

De este modo, la empresa no debe perder el tiempo actualizando los datos de sus páginas, están siempre al día, puesto que en realidad las páginas en sí no existen: **existen los procesos que las crean dinámicamente** y utilizan los datos almacenados en la base de datos, que siempre están actualizados.

De este modo se pueden crear las denominadas **aplicaciones web**: no un mero conjunto de páginas estáticas enlazadas, sino servicios dinámicos, más útiles y personalizados para cualquier tipo de organización.

Estos procesos residen y se ejecutan siempre en el servidor. Para programarlos tenemos distintas tecnologías parecidas entre sí: por una parte CGI/Servlets y por otra ASP/JSP.

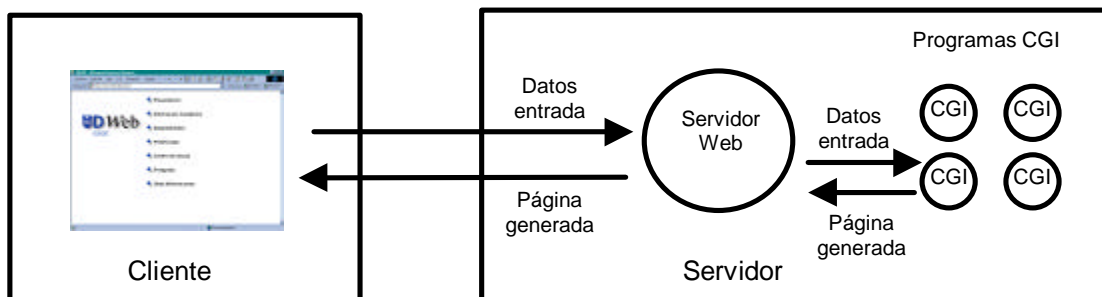
5.2 CGI (Common Gateway Interface)

A partir del estándar HTML 2.0 se incluyeron unas etiquetas que permitían insertar en las páginas HTML unos formularios con la intención de recoger datos del usuario. Estos formularios, una vez completados, se envían al servidor que a su vez se los pasa a los procesos encargados de su interpretación y ejecución, generando una respuesta en forma de página HTML.

Un CGI (Common Gateway Interface) no es más que un programa escrito en cualquier lenguaje de programación que recibe los datos a través del dispositivo de lectura estándar o `stdin`, y que genera un resultado hacia el dispositivo de escritura estándar `stdout`.

Cuando el servidor web recibe una solicitud de ejecución de un CGI junto con los parámetros necesarios, inicia la ejecución del CGI como si de un proceso independiente se tratase pasándole los parámetros al mismo a través del dispositivo estándar de entrada de datos (`stdin`), es decir, el CGI lee de dicho dispositivo los datos que necesita para su realizar su tarea, habiendo sido depositados allí por el servidor web.

A medida que el CGI se va ejecutando puede ir generando los resultados del proceso y los envía al dispositivo de escritura estándar `stdout`, cuya salida estará capturada por el servidor web, que la redirecciona hacia el navegador del usuario que solicitó la ejecución del CGI.



Los CGIs al ejecutarse de forma independiente para cada solicitud no pueden conservar información del estado.

5.3 Servlets

La alternativa Java a los CGIs son los *servlets*, con un funcionamiento similar, pero pudiendo utilizar toda la potencia de la plataforma Java respecto a bibliotecas de clases, de tal modo, que se puede utilizar un servlet para poder recoger los datos de un usuario en un formulario, y enviarlos por correo electrónico gracias a las clases de e-mail que proporciona Java.

```
public class ServletFicheroAnadir extends HttpServlet {  
  
    //Procesar la solicitud POST de HTTP  
  
    public void doPost(HttpServletRequest request, HttpServletResponse  
response) throws ServletException, IOException {  
  
        response.setContentType("text/html");  
        PrintWriter out = new PrintWriter (response.getOutputStream());  
        out.println("<html>");  
        out.println("<head><title>ServletFicheroAnadir</title></head>");  
        out.println("<body>");  
  
        // Recoger el valor del campo nombre  
        String nombre = request.getParameter("txtNombre");  
  
        // Mostrar el mensaje de bienvenida  
        out.println("Bienvenido a nuestra web, " + nombre + "<BR>");  
        out.println("</body></html>");  
        out.close();  
    }  
}
```

Ejemplo del código de un servlet que recibe un nombre y genera una página web con un mensaje de bienvenida

5.4 Active Server Pages

ASP (Active Server Pages) es una tecnología desarrollada por *Microsoft* para diseñar e implementar procesos complejos, haciéndolos accesibles mediante web. La principal innovación de las ASP respecto a otras tecnologías como CGI consiste en que tanto el formato de la información en HTML como el código del proceso a ejecutar residen en la misma entidad: la página ASP. Ya no es necesario disponer por una parte de páginas HTML estáticas y por otra parte pseudo-programas CGI, sino que con sólo un elemento, la página ASP, es suficiente para contener ambas naturalezas y funciones.

Un ejemplo de página ASP sencilla podría ser el siguiente:

```
<HTML>
<HEAD>
<TITLE>Este es un ejemplo</TITLE>
</HEAD>

<BODY>
Estos son los números del 1 al 5:

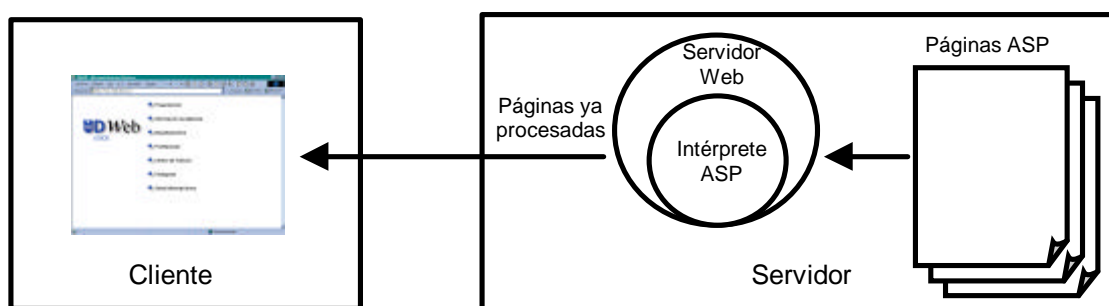
<% For n = 1 To 5 %>
Número <% Response.Write(n) %> <BR>
<% Next n%>

</BODY>
</HTML>
```

ejemplo.asp

Como se puede ver, un documento ASP es básicamente un documento HTML al que se le han añadido unas etiquetas especiales (<% y %>) que indican el principio y el fin del código ASP, código que se procesa en el servidor antes de servir la página al cliente que se la ha solicitado.

Esquemáticamente el proceso sería:



Cuando un cliente HTTP solicita esta página a un servidor con soporte ASP, éste la abre antes de servirla e interpreta cada línea de la página mediante las siguientes reglas:

1. Mientras no se encuentre la etiqueta <% el texto de la página se sirve tal cual.

2. Cuando se encuentra la etiqueta <% de inicio de código ASP, el intérprete de ASP ubicado en el servidor empieza a ejecutar el código ASP, escrito en un lenguaje de programación como VBScript o JScript.
3. Cuando se encuentra la etiqueta %> de fin de código ASP, se vuelve al paso 1, hasta acabar de procesar toda la página.

Como es posible apreciar, el servidor web contiene un intérprete ASP que es el encargado de llevar a cabo la ejecución del código ASP. Actualmente sólo los servidores web de Microsoft (el Personal Web Server y el Internet Information Server) incorporan dicho intérprete, por lo que el uso de las ASP determina la elección del servidor web.

Como resultado del ejemplo, un cliente que hubiese solicitado la página ejemplo.asp al servidor, hubiese recibido como respuesta:

```
<HTML>
<HEAD>
<TITLE>Este es un ejemplo </TITLE>
</HEAD>

<BODY>
Estos son los números del 1 al 5:

Número 1<BR>
Número 2<BR>
Número 3<BR>
Número 4<BR>
Número 5<BR>

</BODY>

</HTML>
```

Resultado de la ejecución de ejemplo.asp

Al cliente siempre le llega el resultado del procesamiento de la página ASP en el servidor, y ese resultado siempre es íntegramente código HTML, de tal modo, que no es posible analizar la página recibida y determinar qué parte de la misma era HTML constante en su origen y qué parte se ha generado de forma dinámica. Por supuesto las páginas ASP también pueden recibir datos que el usuario envía a través de un formulario.

La tecnología ASP es muy flexible y potente, permitiendo que las páginas ASP puedan acceder a bases de datos, recuperar ciertos valores de las mismas y generar dinámicamente documentos HTML que muestran el contenido de la

base de datos. Asimismo las páginas ASP se integran perfectamente con el resto de tecnologías Microsoft, permitiendo vincular el código ASP con el sistema de correo electrónico (para enviar e-mails como resultado del procesamiento de una página), con objetos COM que representan los procesos de negocio de una organización (para permitir que el código ASP pueda acceder al sistema de facturación de la empresa y poder facturar por web, obteniendo el número de tarjeta del cliente).

5.5 Java Server Pages

Básicamente la tecnología y el funcionamiento de las JSP es la misma que las de las ASP, la diferencia radica en que mientras la última ha sido desarrollada por *Microsoft* y se integra perfectamente con el resto de tecnologías de la corporación norteamericana, las JSP se han desarrollado con la vista fija en la plataforma Java, y se integra perfectamente con el resto de productos basados en Java.

La sintaxis de las JSP es similar a ASP, con las etiquetas `<%` y `%>`, pero el lenguaje de programación es de tipo script basado en la sintaxis de Java. La vinculación de JSP con Java le permite interactuar con otros elementos de la plataforma como Enterprise JavaBeans (que son la alternativa Java a los objetos COM de Microsoft) o con bases de datos accesibles mediante JDBC (mecanismo para la conexión de aplicaciones Java con bases de datos).

Actualmente el único servidor web con intérprete JSP es el Java Web Server, lo cual limita la elección del servidor web a uno sólo. Sin embargo, al ser Java una plataforma medianamente libre, se intenta promover que otros servidores web incorporen estas capacidades, y se espera que paulatinamente crezca el número de los que incluyen soporte JSP.

5.6 Tecnologías de cliente: applets y ActiveX

Las tecnologías que se han visto anteriormente conseguían dotar a la web de mayor potencia y versatilidad, al añadir la capacidad de ejecutar procesos.

Para actividades que requieren una interacción continua con el usuario como jugar a las damas o dibujar, las tecnologías de aplicaciones web no son tan adecuadas.

Ejemplo: si un usuario quiere dibujar una línea, debe introducir las coordenadas de origen y las de destino, enviarlas en el formulario y esperar a que se procesen y se genere la página web de respuesta con la imagen resultante.

En estos casos de interactividad máxima, es más adecuado que **los datos no viajen hasta el servidor** para ser procesados, sino que se procesen en la propia máquina cliente, lo cual implica que la página web que el navegador muestra debe ser capaz de realizar la tarea, sin necesidad de comunicarse con el servidor.

Para ello las páginas web pueden traer incrustados mediante las etiquetas HTML apropiadas ciertos mini-programas con los que interactúa el usuario. Actualmente existen dos modelos distintos para realizar estos programas: los applets y los ActiveX.

Applets

Son pequeños programas desarrollados en Java, con una estructura especial que utilizan el entorno proporcionado de un navegador para su ejecución. Fueron de hecho los applets los que aportaron el espaldarazo definitivo a la plataforma Java, por un hecho fundamental: Java es un lenguaje multiplataforma e Internet es una red multiplataforma donde máquinas de diversas arquitecturas y sistemas operativos conviven simultáneamente.

Si se quieren realizar programas que viajen con las páginas web hacia el navegador del usuario, dichos programas debieran ser capaces de ejecutarse bajo cualquier plataforma en la que se encontrase el navegador: *Macintosh* con *MacOS*, *Intel* con *Windows*, *HP* con *Unix*, *Intel* con *Linux*, etc., y para ello era necesario que el lenguaje en el que se desarrollaran los programas generase código ejecutable en cualquier plataforma. El lenguaje sería Java.

Para incluir un applet en una página web se utiliza la etiqueta `<APPLET>` como en el ejemplo:

```
<APPLET CODE="chatterlite.ChatClient" CODEBASE="clases">
  <PARAM NAME="NICK" VALUE="Pepito">
  <PARAM NAME="CANAL" VALUE="#hola">
</APPLET>
```

Mediante las etiquetas `<PARAM>` es posible pasar parámetros al applet.

Los applets están soportados por los principales navegadores del mercado.

ActiveX

ActiveX es una tecnología de *Microsoft* que permite definir componentes (pequeños programas) con capacidades especiales que les permiten incrustarse dentro de contenedores, como documentos o ventanas de aplicación, y comunicarse con ellos. ActiveX se utiliza cada vez más con el propósito de crear componentes con algún tipo de interfaz gráfica para su manipulación, que puedan ser incrustados dentro de aplicaciones, páginas web o documentos de *Word*, entre otros.

Los ActiveX están orientados hacia la plataforma *Intel-Windows* con lo que no se pueden ejecutar en otras plataformas distintas, al ser código máquina ya compilado. Además el único navegador con soporte ActiveX es el *Microsoft Internet Explorer*.