Universidad Técnica Federico Santa María.
Departamento de Electrónica.

ELO330 Programación de Sistemas.
II Semestre 2002.
Profesor: Agustín González.

Informe Proyecto.

Implementación de un foro en páginas ASP.

Nombre alumno: Felipe Carrillo Oliva.
Rol: 9621013-2
Fecha: 30 de octubre de 2002.

INTRODUCCIÓN

Se ha querido desarrollar en el proyecto personal, un foro electrónico implementado en páginas web activas, utilizando Active Server Pages de Microsoft. El objetivo no es aprender ASP a profundidad, sino tomar las herramientas básicas que permita implementar esta aplicación. Además, se utilizan conceptos de bases de datos, de modo de almacenar la información generada en el foro de manera estructurada. Esta implementación se efectúa en Microsoft Access.

En este informe, se detallan: una breve introducción a ASP; las herramientas ASP que se utilizan en el proyecto; descripción del proyecto; la implementación de la base de datos y finalmente algunas conclusiones.

En el apéndice se muestra la implementación del proyecto, en páginas ASP.

I - Introducción a Active Server Pages.

La investigación y desarrollo en el campo de la creación y mantenimiento de páginas web es uno de los más dinámicos en el ámbito de internet, debido a sus consecuencias comerciales y de utilización de la red. Inicialmente, las páginas eran estáticas, es decir, los usuarios sólo podían acceder a su contenido, escrito en lenguaje HTML. Sin embargo, cuando se requiere mayor interacción entre el usuario y el sitio web, surge la necesidad de reunir y procesar las peticiones del cliente de modo de ofrecerle la información que necesita en forma concisa.
Surge entonces la conveniencia de tener una etapa de procesamiento en la interacción de información entre los clientes y el servidor web.
El proceso en páginas web dinámicas, se puede implementar siguiendo alguno de los siguientes modelos.

1- **Procesamiento en el equipo del usuario**: páginas dinámicas en el cliente.

Se interpreta o ejecuta código relacionado con la página en el equipo del usuario. Entre sus ventajas se tiene:

- Se libera al servidor de una gran carga de cómputo.
- Se puede disminuir el ancho de banda en comunicaciones.
- Se puede utilizar recursos locales

Desventajas:

- A menudo se depende del explorador.
- Los clientes pueden no poseer suficiente capacidad de cómputo.
- En ocasiones pueden es poco seguro dado que se puede acceder a los recursos del cliente.
- No resultan si el problema es de naturaleza centralizada, es decir sin recurrir a ejecuciones en el servidor.
- Cuando el programa es grande y el tiempo de cómputo corto, es más conveniente traspasar los resultados del servidor al cliente en lugar del programa.

Ejemplos de este modelo representan HTML dinámico, Active X de Microsoft y Applets de Java. El último está orientado a ejecución segura de programas en el cliente.

2- **Procesamiento en el equipo servidor:** páginas activas en el servidor.

En este caso, el servidor envía al cliente solamente instrucciones en HTML. Este modelo representa ventajas como:

- Se puede acceder a información centralizada.
- Mayor seguridad al cliente.
- Liberan a los clientes de cómputos que a veces no pueden procesar.
- Se tiene la seguridad que todos los clientes podrán ver la página web.

Ejemplos de implementación de este modelo, CGI (Common Gateway Interface), PHP, y ASP (Active Server Pages) de Microsoft.

Características de ASP

En ASP se utilizan lenguajes de guiones (scripts) para la funcionalidad de las páginas activas. Entre estos lenguajes se encuentran Visual Basic Script (VBScript) y Java Script. Las páginas web pueden ser diseñadas con editores HTML ya que las instrucciones ejecutables y el código HTML están bien delimitados. Los delimitadores <% ….%> indican al soporte de ASP las secciones de página web que deben ser ejecutadas en el servidor.

ASP permite compatibilizar la creación de páginas web activas en el cliente y en el servidor, pudiéndose así balancear la carga de proceso y comunicaciones según lo desee el diseñador. También se permite utilizar varios lenguajes de programación script en la misma página. Las páginas web que devuelve el servidor tras la ejecución de las instrucciones, están formadas por sentencias HTML visualizables por cualquier explorador.

En el programa proyecto, se utiliza Visual Basic Script.

II- Descripción de las herramientas ASP VBScript usadas en el programa proyecto.

Antes de comenzar, es necesario señalar que cada una de las características aquí descritas sobre ASP y VBScript poseen muchos más recursos. Para una referencia más completa, es necesario revisar las referencias al final del informe.

1- Utilización de formularios.

El siguiente código HTML está en la página login.asp, y tiene como objetivo recoger la información que el usuario desea enviar al foro.

```
<FORM METHOD="Post" ACTION="login.asp" id=form1 name=form1>

        E-mail: <INPUT NAME="Email"><BR>
        Clave: <INPUT TYPE="password" NAME="Clave"><BR>
        <INPUT TYPE="Submit" VALUE="Enviar" id=Submit1 name=Submit1>
</FORM>
```

El atributo ACTION indica que los datos recogidos, esto es, las entradas (INPUT) del cliente “Email” y “Clave”, serán pasados a la página activa login.asp (en este caso la misma, solamente que la página está condicionada a actuar de otra forma si se le han enviado datos por formulario). Esta es una implementación simple de un formulario, sin embargo, se puede crear formularios que recogen mucho más cantidad de información, aprovechando además las herramientas de entrada de HTML.
Otra utilización de formularios, se puede ver en la página opinar.asp, donde se recoge la opinión que quiere enviar el usuario a algún foro.

2- Variables

Las variables en VBScript se declaran explícitamente como por ejemplo:

```
Dim Clave1, Clave2, Email
```

La palabra reservada Dim (abreviatura de Dimensión) indica que se han reservado espacios “vacíos” para las variables, espacios que no tienen ningún significado lógico.

Los tipos de datos se especifican en el momento de asignar valores a la variable:

```
Clave1=Request.Form("Clave")
Email=Request.Form("Email")
```

Entre los tipos, se encuentran los numéricos (byte, integer, long, double) y los no numéricos (string, bolean, array).

3- Estructuras de control

En VBScript se encuentran las típicas instrucciones de bifurcación y bucles, que se emplean en la mayoría de los lenguajes de programación.

Por ejemplo, tenemos la instrucción de bifurcación If…Then…Else…End If:

```
IF (strComp(Clave1,Clave2)<>0) OR Obj_RSusuario.EOF  THEN

%>
        <H1>Error en el ingreso</H1>
        <A HREF="login.asp">REINTENTAR</A>
<%
ELSE
        Response.Cookies("idusuario")("email")=Email
        Response.Cookies("idusuario")("clave")=Clave1
        Response.Redirect("http://filipo:8080/elo330/inicses.asp")
END IF
```

Este ejemplo es usado en login.asp, y determina cuando se ha autenticado correctamente el usuario. En *IF* se compara el valor de las claves obtenidas por formulario desde el usuario y desde la base de datos, además se contempla si existe o no un registro de usuario con la información entregada. Si no es así, *THEN* envía un mensaje de error. Pero si la información era correcta *ELSE* permite al usuario iniciar sesión.

Se tiene el bucle Do While…Loop:

```
DO WHILE NOT(Obj_RSforo.EOF)
        'Crear link al foro
        %><STRONG><A HREF="detforo.asp?clas=<%=Obj_RSforo("class_foro")%>&nro=<%=Obj_RSforo("nforo")%>">
        <%=Obj_RSforo ("tema_foro") %></A></STRONG><BR><%

        %>Opiniones:<%Response.Write Obj_RSforo ("nopiniones")%><BR><%
        Obj_RSforo.MoveNext
LOOP
```

Este bucle está en la página foros.asp, y se encarga de llenarla con los foros con sucesivas consultas a la base de datos.

Los ejemplos anteriores, permiten ver además que se dispone de operadores lógicos, como “=”(igual), “<>” distinto de, “AND”, “OR”, “NOT”.

4- Objetos integrados.

a) El objeto Request permite recuperar la información enviada por un usuario al servidor.

El siguiente ejemplo hace uso del atributo Form, que permite recuperar información enviada desde un formulario.

```
Clave1=Request.Form("Clave")
Email=Request.Form("Email")
```

Este ejemplo se encuentra en login.asp, y recupera la información enviada por el usuario en esta misma página y las almacena en las respectivas variables.

Otro ejemplo, muy utilizado en el programa proyecto, es el siguiente, y determina que hacer cuando se ha recibido o no información de formulario por parte del usuario:

```
IF Request.Form="" THEN
…
ELSE
….
END IF
```

Es decir, si no se ha recibido información del usuario, se ejecutará el código que sigue a *THEN*, pero cuando efectivamente se recibe información, se ejecuta el código que corresponde a *ELSE.*

i) La colección *QueryString* recupera los valores de las variables de cadena de consulta enviada en una petición HTTP.

Ejemplo:

```
<FORM METHOD="Post" ACTION="opinar.asp?clas=<%=clasforo%>&nro=<%=nroforo%>" id=form1 name=form1>
        <TEXTAREA rows=10 cols=60 id=textarea1 name="optext">
        </TEXTAREA><BR>
        <INPUT TYPE="Submit" VALUE="Enviar" id=Submit1 name=Submit1>
</FORM>
```

```
nroforo=Request.QueryString("nro")
clasforo=Request.QueryString("clas")
```

En este ejemplo, que recoge una opinión del usuario y se encuenta en la página *opinar.asp*, el atributo ACTION, especifica una cadena de consulta en un link, con los valores *clas* y *nro*, los que se recuperan en la misma página. Nuevamente tenemos el caso en que una página actúa de distinta forma si se le han enviado o no datos por formulario.

En el caso de la información de formulario, ésta se encuentra en el cuerpo de la petición HTTP.

ii) La colección *Cookies* permite recuperar el valor de una cookie:

Ejemplo: `Email=Request.Cookies ("idusuario")("email")`

Aqui se asigna el valor de la clave *email* de la cookie *idusuario* a la variable *Email.* Este ejemplo está en casi todas las páginas del programa proyecto, y permite recuperar la información de autentificación del usuario para determinar si se permite o deniega el acceso a los contenidos de la página.

b) El objeto Response se usa para enviar al cliente cualquier clase de información, y para manejar la manera de su envío y su retorno.

i) El atributo *Write* escribe la variable en el resultado HTML actual en forma de cadena.

Ejemplo: Response.Write(clasforo)

La siguiente sentencia tiene el mismo efecto:
=clasforo

ii) El método *Redirect* indica al navegador que conecte con otra dirección URL.

Ejemplo: Response.Redirect("inicses.asp")

Este ejemplo esta en *login.asp* y desvía al usuario a la página *inicses.asp* cuando se autentifica correctamente.

iii) La colección *Cookies* permite crear y modificar cookies en el navegador cliente.

Ejemplo: Response.Cookies("idusuario")("email")=Email

Aquí, se envía al usuario una cookie de nombre **idusuario**, con una clave llamada **email** a la cuál se le asigna el valor de la variable **Email**. Esto puede crear una cookie o bien modificarla. Se usa en *login.asp* para enviar una cookie con la información de autentificación del usuario, para que las demás páginas identifiquen al usuario.

c) **El objeto Server**, permite consultar aspectos del servidor y trabajar directamente con sus características.

i) El método *CreateObject* crea una instancia de un componente instalado en el servidor.

Ejemplo: SET Obj_Conn = Server.CreateObject ("ADODB.Connection")

Se crea la instancia *Conn* del objeto *Connection* de la librería *ADODB*. Este objeto permite enlazar la aplicación web con una base de datos. A su vez, este objeto creado, posee varios atributos y métodos.

- El método *Open* realiza la conexión real a la base de datos.

  Ejemplo: Obj_Conn.Open "Foro"

  Este ejemplo, muestra como se abre una conexión usando el DSN *Foro*, el cuál posee la información sobre la ubicación de la base de datos.

- El método *Close* cierra la conexión a la base de datos:

  Ejemplo: Obj_Conn.Close

d) El objeto RecordSet permite obtener acceso a las tablas de la base de datos:

Ejemplo: SET Obj_RSusuario = Server.CreateObject ("ADODB.RecordSet")

Se crea la instancia *Obj_RSusuario*.

i) El método *Open* abre un conjunto de registros (RecordSet) dentro de la base de datos:

Ejemplo: Obj_RSusuario.Open "Usuario", Obj_Conn, 3, 3

Se abre la tabla *Usuario*, en la conexión *Obj_Conn*. Las otras dos propiedades, la primera (adOpenStatic) especifica que se puede mover en cualquier dirección del *RecordSet*, pero éste no detectará ningún cambio ajeno al usuario que lo utilice. La última propiedad (adLockOptimistic) especifica que se modificarán registros y que la base de datos se bloqueará sólo en el momento de realizar cambios.

- La propiedad *Filter* se utiliza para seleccionar sólo un conjunto de registros:

  Ejemplo: Obj_RSusuario.Filter="email=' "&Email&"' "

  Se seleccionan aquellas tablas cuyo campo *email* coincida con el valor de la variable *Email*.

- La propiedad *EOF* devuelve Trae cuando se ha sobrepasado el último registro del RecordSet.

  Ejemplo:
  ```
  IF NOT(Obj_RSusuario.EOF) THEN
  Clave2=Obj_RSusuario ("clave")
  END IF
  ```

- El método *AddNew* añade un nuevo registro al RecordSet.

  Ejemplo: Obj_RSopinion.AddNew

- El método *Update* actualiza los cambios en el RecordSet, ya sea por añadir un registro o modificar uno existente.

  Ejemplo: Obj_RSopinion.Update

- El método *Close* cierra el RecordSet.

  Ejemplo: Obj_RSusuario.Close

III- Programa Proyecto

Se trata de utilizar las herramientas de ASP mediante Visual Basic Script, para crear una aplicación web que implementa un foro electrónico. Los datos se guardan en una base de datos Microsoft Access.

1- Modelo de base de datos.

La figura siguiente muestra el modelo relacional de la base de datos para el foro.

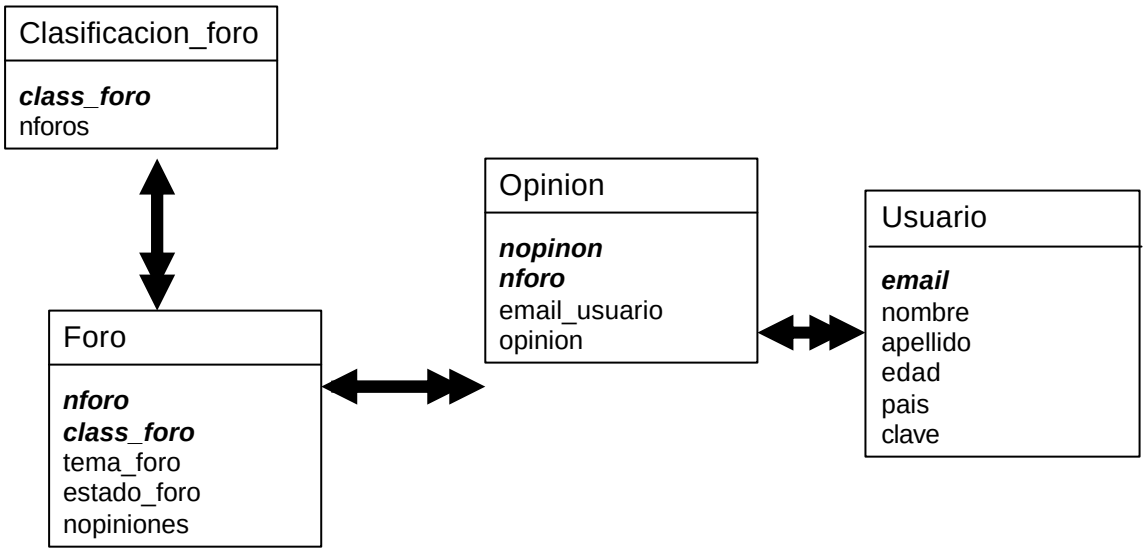

Figura 1. Modelo de base de datos para el foro.

Para comprender este modelo, se explican cada uno de sus registros.

a) Usuario: guarda los datos de usuarios del foro. La clave primaria es *email*, que determina la unicidad de un registro de usuario, es decir, dos o más usuarios no pueden compartir una misma clave *email*. Los demás campos son autoexplicativos.

b) Clasificación_foro: los foros son clasificados dentro de distintas clasificaciones. En este registro esas clasificaciones se especifican en el campo *class_foro*, el cuál es la clave primaria, mientras que *nforos* guarda el número de foros asociado a la clasificación.

c) Foro: se especifican los datos de un foro. Un foro queda determinado por el número de foro dentro de su clasificación (campo *nforo*) y por la clasificación a la que pertenece (campo *class_foro*). Este último campo es una clave foránea (que pertenece al registro clasificación) y que determina integridad referencial, es decir, no se puede crear una instancia del registro foro dentro de una clasificación que no exista. Además, esta relación implica que un foro pueda tener sólo una clasificación, pero una clasificación pueda tener varios foros

(relación uno a muchos). Los demás campos, *tema_foro* guarda el enunciado del tema del foro, *nopiniones* guarda el número de opiniones que tiene el foro, mientras que *estado_foro* señala si el foro está abierto o cerrado.

d) Opinión: registra la opinión de un usuario en un foro. Las claves primarias de este registro son: *nopinion* que es el número de la opinión dentro del foro al que pertenece, *nforo* que es el número de foro al que pertenece, y *class_foro* que indica la clasificación del foro al que pertenece la opinión. Con esto, la opinión queda completamente determinada y que un foro pueda tener varias opiniones y una opinión sólo se relacione con un foro. Además está la clave foránea *email_usuario*, que apunta al campo *email* del registro *Usuario*. Esto determina que un usuario pueda tener varias opiniones en un mismo foro, mientras que una opinión pueda tener un solo usuario.

2- Aplicación ASP

La figura siguiente, muestra esquemáticamente la organización de la aplicación.

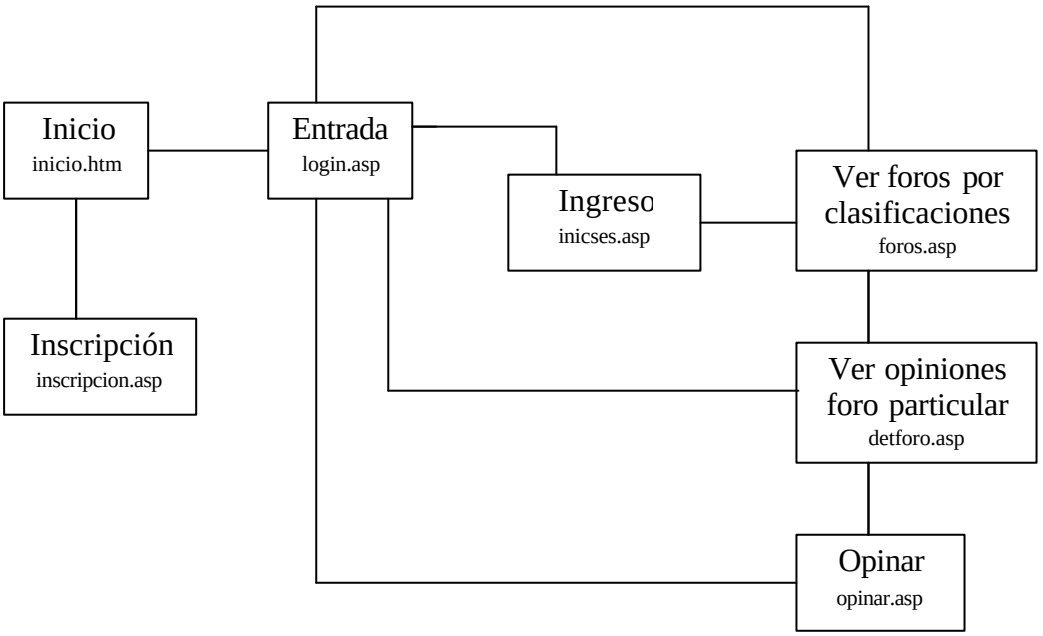

Figura 2. Esquema de aplicación ASP para el foro.

Las flechas indican hacia dónde pueden desviarse las distintas páginas.

El usuario, al ingresar al sitio, tiene las opciones de autentificarse e ingresar. Si no está registrado puede hacer su inscripción. Cuando el usuario es autentificado, puede comenzar a revisar los foros existentes en la base de datos.

La idea es que sólo un usuario registrado en la base de datos pueda participar en los foros, de modo que en cada etapa, o bien, en cada página se autentifique al usuario. Para ello, cuando el usuario se autentifica en la entrada, se le envía una cookie con la información de autentificación (*email* y *clave*), de modo que el servidor recupere esos

datos en cada página y autentifique al usuario. Dicha cookie sólo es válida durante una sesión. Con esto, cuando se intente ingresar a alguna página de la aplicación sin autentificar, se negará el paso con un procedimiento implementado en la misma página. Esto puede verse en que para las páginas que se encuentran después del ingreso, todas pueden regresar a la entrada como lo indican las flechas. Esto se hace en el caso que algún usuario intente abrir alguna de ellas sin haberse autentificado.

Para registrarse, el usuario debe llenar un formulario, que incluye una clave.

Para efectuar las operaciones de autentificación, consultar foros y opinar, necesariamente se debe hacer una consulta a la base de datos del foro, por lo que una aplicación de este tipo es imposible de hacerla en páginas web estáticas.

La implementación se llevó a cabo exitosamente usando las herramientas ASP con VBScript descritas anteriormente. En el apéndice de este informe, están detalladas las páginas ASP.

3- Implementación.

Para el desarrollo del programa, se utilizó la aplicación Microsoft Visual Interdev 6.0, y la aplicación se ejecuta en el servidor web de Internet Information Server 5 en Windows 2000 Professional , el cuál soporta páginas ASP. Como se dijo anteriormente, la base de datos se implementa en Microsoft Access, y la aplicación web se conecta a la base de datos, especificando un DSN de sistema en la Administración de Orígenes de Datos ODBC de Windows 2000 Professional.

CONCLUSIONES

En este trabajo se adquirió conocimiento y práctica de las potencialidades que brindan las páginas web dinámicas para desarrollar aplicaciones web. De hecho, en el programa proyecto se utilizaron algunas características de Visual Basic Script en ASP. Además, se aprendió a utilizar las herramientas de servidor web de Windows 2000 Professional.

Si bien, el diseño e implementación del foro tiene poca elaboración, lo que se quería tener como aplicación se logró.

REFERENCIAS

- **Active Server Pages (ASP 3.0) Iniciación y Referencia.**
  - Jesús Bobadilla Sancho. UPM.
  - Alejandro Alcocer Jarabe. NalCOM.
  - Luis Rodríguez-Manzaneque Sánchez. NalCOM.
  - Editorial McGraw Hill Interamericana. 2001.
- **Aprenda Visual Basic como si estuviera en primero.**
  - Javier García de Jalón.
  - José Ignacio Rodríguez.
  - Alonso Brazalez.
  - Escuela Superior de Ingenieros Industriales. Universidad de Navarra.1999.