Número 5, desembre 2000

ASP en biblioteques i centres de documentació: publicació d'una base de dades *Microsoft Access* en el web

[versión castellana]

Josep Manuel Rodríguez i Gairín
Facultat de Biblioteconomia i Documentació
Universitat de Barcelona
rzgairin@fbd.ub.es

Resum

Es descriuen els passos necessaris per publicar a Internet una base de dades creada amb *Microsoft Access* utilitzant la tecnologia ASP. Es parteix d'un exemple de base de dades creada per a controlar el procés d'adquisicions i es van resseguint, en forma de tutorial, els diferents passos que seran necessaris per a la seva consulta des del web. Finalment, s'indiquen algunes aplicacions de la tecnologia ASP que podren ser útils per a biblioteques i centres de documentació.

Introducció

El present article recull, a tall de guia, els passos necessaris per publicar una base de dades de *Microsoft Access* en Internet. Si bé s'introdueixen al llarg d'aquest definicions i sigles com ASP, ODBC o SQL, s'ha pretès fugir de la realització d'un redactat teòric i concentrar-se en una exposició pràctica seguint un exemple concret de l'entorn bibliotecari que permeti valorar la seva utilitat.

La nostra experiència demostra que, en moltes ocasions, la millor manera d'aprendre determinats processos tècnics és executar-los i per això, convidem a llegir aquest article mentre es realitza a la pràctica l'objectiu proposat de publicar una base de dades.

Si bé són aconsellables coneixements elementals tant d'*Access* i HTML com de programació bàsica, l'exposició es basarà en senzills exemples i trucs que poden ser seguits sense excessiva dificultat per qualsevol professional de la biblioteconomia i la documentació.

Aquest article és interactiu. Si el lector està connectat a Internet podrà veure els resultats dels diferents formularis instal·lats en un servidor NT i interaccionar directament amb la base de dades d'exemple. Si detecteu alguna anomalia o disfunció en el servidor, us agrairem que ens ho comuniqueu per correu electrònic.

El nostre exemple

Els avantatges i inconvenients d'usar un sistema de gestió de bases de dades relacional com *Microsoft Access* en la gestió del catàleg d'una biblioteca ja han estat àmpliament tractats per diversos autors (Salse, 1998; Trigueros, 1997; Codina, 1994). Tanmateix, molts dels processos relacionats amb tasques que impliquen gestió econòmica —adquisició, obtenció de documents, préstec interbibliotecari— estan sent tractats amb motors de bases de dades relacionals com *Microsoft Access*, *GTBib* o *Dbase*. Fins i tot grans centres automatitzats amb potents sistemes de gestió de biblioteques com *VTLS* o d'altres, usen aquestes bases de dades relacionals per als processos administratius. Per la seva part, els petits centres, que disposen d'escàs pressupost per a automatització, troben en el paquet *Office* de Microsoft una de les solucions de primera línia per a tots aquests propòsits.

Sense voler entrar més en aquesta polèmica i encara que podríem haver usat un exemple administratiu d'un altre camp, ens ha semblat més oportú utilitzar-ne un que pogués resultar familiar al lector. A aquest efecte, usant *Microsoft Access,* dissenyarem una base de dades per controlar les adquisicions d'una biblioteca que anomenarem "adqui.mdb". La gestió de les adquisicions és un procés complex que implica la gestió de partides pressupostàries, proveïdors, comandes, reclamacions, tancaments d'exercici, etc. Tanmateix i ja que no és el tema principal d'aquest article, hem simplificat l'exemple usant només tres taules relacionades entre si: els usuaris que sol·liciten la compra per mitjà de desiderates o similars, els llibres que es desitgen adquirir i els proveïdors d'aquells documents. En el

nostre exemple, cada llibre és sol·licitat per un únic usuari i se sol·licita a un únic proveïdor; la realitat seria més complexa ja que un llibre podria ser sol·licitat per més d'un usuari i en cas de no ser servit es podria tornar a sol·licitar a un altre proveïdor. Necessitaríem, per tant, altres taules auxiliars per millorar tot aquest procés.

Gràficament l'estructura de la nostra base de dades simplificada tindrà l'aspecte següent:

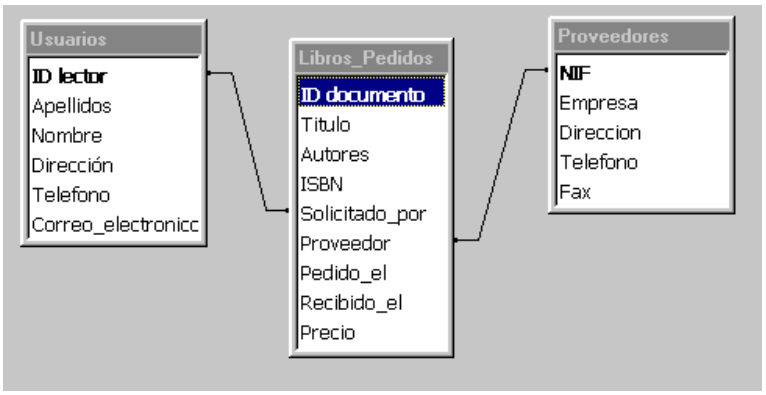

Figura 1. Estructura i relacions de les taules a la base de dades

Per realitzar els processos relacionats amb l'adquisició potser no seria necessari publicar aquesta base en el web. Si els responsables del servei són una o dues persones, poden treballar directament en un entorn Windows amb *Microsoft Access.* El problema es planteja quan decidim emprar aquesta base de dades per oferir subproductes a l'usuari final a través del WWW.

Un exemple clar el constitueixen els típics expositors que trobem a l'entrada de moltes biblioteques on es mostren els últims llibres rebuts. La seva versió electrònica bé podria ser una llista de noves adquisicions ingressades al nostre centre que pengés de la pàgina principal del nostre web. Fins i tot podríem dissenyar un sistema que permetés a l'usuari consultar l'estat de la seva desiderata, si s'ha demanat ja al proveïdor, si s'ha rebut, etc. Evidentment, en centres petits aquesta consulta podria convertir-se en la interfície del mateix catàleg. Insistirem a recordar les limitacions del sistema: camps de longitud fixa, no existeix possibilitat de repetició, indexació per frase, etc.

Aquests processos podrien resoldre's en manera local per mitjà de consultes. Com a exemple, dissenyarem una consulta de selecció sobre la taula "libros_ pedidos" per llistar els documents rebuts fins a una data concreta i una consulta d'unió per llistar els llibres demanats per un usuari o a un proveïdor concret.

Algunes d'aquestes consultes han de tenir l'aspecte següent o similar:

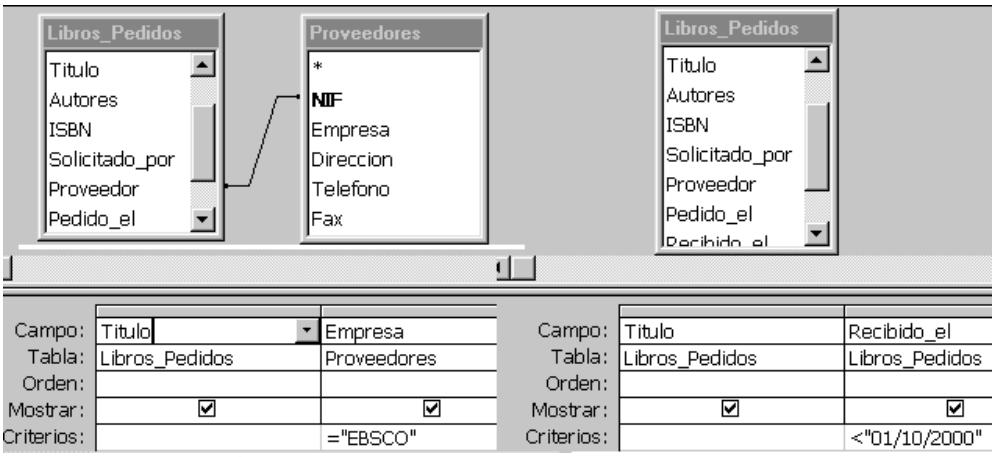

Figura 2. Consultes en *Microsoft Access*

Observem que en aquests casos, si no volem entrar en programació, les consultes han de ser predeterminades, és a dir, necessitarem fer tantes consultes com proveïdors o usuaris. Podríem generar un formulari en *Access* i lligar-lo a la consulta però necessitaríem conèixer més a fons el sistema. En el cas de la consulta via web, com veurem, la solució també requereix crear un formulari previ que ens permeti escollir el proveïdor o fixar una data concreta en cada cas.

Nota: Encara que recomanem que sigui el mateix lector el que generi la seva pròpia base de dades i les consultes, si ho desitja pot descarregar la que hem confeccionat nosaltres, que és la que hem utilitzat com a exemple al llarg d'aquest article.

La tecnologia que descriurem permet no tan sols consultar les dades via web sinó també realitzar totes les funcions d'administració clàssiques (afegir, modificar, esborrar registres). Tanmateix detallar tots aquests processos excedeix el propòsit d'aquesta exposició i requereix nocions més avançades d'informàtica en les quals hi ha implicats aspectes de seguretat i control de registres. De tota manera és important conèixer que podríem publicar igualment informes o formularis per a l'entrada o la modificació de dades.

Els requisits

Internet es basa en una arquitectura client-servidor; els usuaris que consultin la nostra base de dades via web simplement necessitaran disposar d'una connexió a Internet i d'un navegador, sense que calgui tenir instal·lat *Access* ni cap altre complement. Per la nostra part, necessitem un servidor de web connectat permanentment a Internet per poder disposar de l'esmentada base les 24 hores. Els fitxers *Access* són aplicacions Windows de 32 bits i per tant, han d'estar instal·lats en un entorn Windows. Es recomana *Windows NT* amb *Internet Information Server 4.0* o superiors. Tanmateix, per fer proves podem configurar un sistema en local sense necessitat de connectar-nos a Internet, simplement instal·lant en el nostre PC (*Windows 95/98*) el *Servidor Personal de Web* (*PWS)* que es pot descarregar d'Internet (http://www.microsoft.com/download), encara que també s'inclou directament en el CD-ROM de *Windows 98*. Es recomana *Windows 98* amb *PWS 4.0* que ja ofereix un suport complet a la tecnologia ASP (Active Server Pages).

Evidentment al nostre ordinador necessitarem tenir també *Access* i un editor de pàgines web (encara que amb el bloc de notes ja en tindrem prou per al nostre exemple). La versió d'*Access* que utilitzem és *Access 2000*, si bé versions anteriors són igualment vàlides per configurar la base de dades. *Access 2000* disposa d'un assistent per crear les pàgines web però no l'usarem sinó que s'editaran manualment per aprendre el que realment fan.

Els passos a realitzar

Copiar la base de dades en el servidor

Els passos següents són vàlids tant si instal·lem la base de dades en el servidor d'Internet (IIS) com en el cas de la instal·lació local amb *Personal Web Server.*

Si ja tenim creada la nostra base de dades com s'ha descrit anteriorment i hem introduït alguns registres, el primer que hem de fer és copiar-la al servidor. Podem copiar-la en qualsevol directori; no és necessari que sigui visible des del web, és a dir, no és necessari que pengi de \inetpub\wwwroot\. De fet, es recomana copiar-la fora d'aquesta estructura per a una major seguretat.

Establir una connexió ODBC

ODBC (Open Database Connectivity) és el protocol que el sistema operatiu emprarà per comunicar-se amb la base de dades. Serà el mateix servidor el que interrogarà la base i n'obtindrà els resultats. Això és important ja que implica que per la xarxa només circula la pregunta formulada i el conjunt de resultats, disminuint notablement el temps de resposta.

El llenguatge que empra el servidor per comunicar-se amb la base es denomina SQL (Structured Query Languaje). Tant el protocol com el llenguatge són estàndards oberts, no requereixen comprar cap programa addicional i són compatibles amb multitud de formats de bases de dades, entre ells el MDB (Microsoft Database).

ODBC empra uns fitxers controladors (*drivers*) específics per a cada tipus de bases de dades. Microsoft facilita aquests controladors i normalment estaran instal·lats en el nostre servidor o s'instal·laran al costat de l'aplicació d'*Office.*

ODBC s'instal·la per defecte amb el Windows. Simplement hem de definir una connexió o enllaç, indicant al sistema que tenim una base de dades que volem fer accessible a aquest protocol. Això és molt fàcil i no requereix més de 5 minuts. Vegem com fer-ho al nostre ordinador; el procés és idèntic en el servidor NT però en aquest cas, per qüestions de seguretat, possiblement només pot fer-ho l'administrador del sistema.

Anem a "Inici" ---> "Configuració" --> "Panell de control" i escollim la icona "Fonts de dades ODBC". Aquesta icona, que és com el que es mostra a la figura adjunta, s'instal·la directament amb Windows i sempre la trobarem al nostre ordinador.
Ens apareixerà un quadre de diàleg com el que es mostra a continuació:

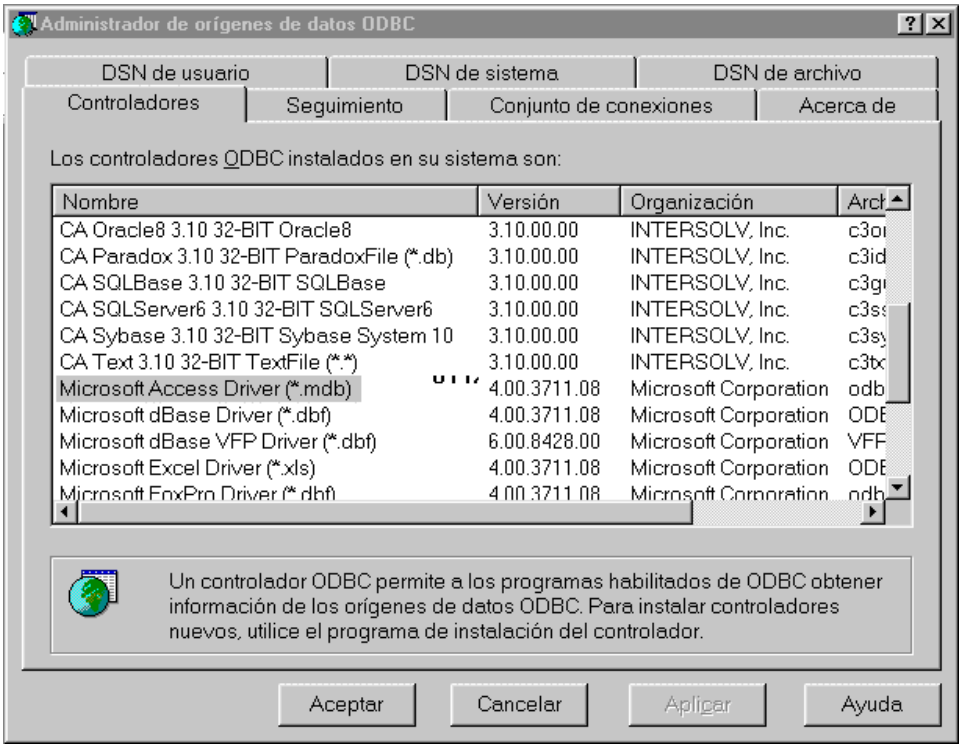

Figura 3. Finestra de l'administrador d'origen de dades ODBC

En aquest exemple es mostren els controladors que el nostre sistema té instal·lats. Quan s'obre aquest diàleg, potser apareix una altra pantalla; polsi en la pestanya "Controladors" per accedir a aquesta pantalla en concret. En el nostre exemple apareixen molts controladors; observeu que ODBC al nostre ordinador pot interrogar molts tipus de bases de dades. Alguns d'aquests controladors sí que són propietaris i tenen un cost però als efectes que ens ocupa simplement necessitem el controlador *Microsoft Access Driver* (*.mdb) que apareixerà si el nostre ordinador té instal·lat Access o que podrà ser instal·lat en el servidor pel nostre administrador de xarxa sense cost descarregant-lo d'Internet.

Per crear la nostra connexió ODBC simplement polsarem la pestanya "DSN de sistema (*data source names*, noms d'origen de dades de sistema)".

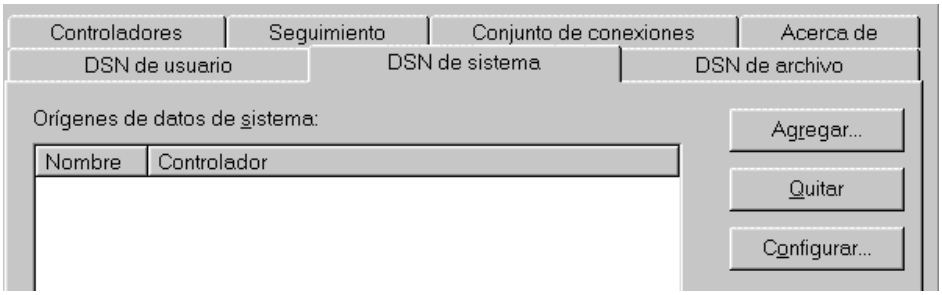

Figura 4. Quadre d'origen de dades de sistema

Al quadre "Origen de dades de sistema" apareixeran les connexions ODBC que ja tinguem establertes. És necessari crear una connexió/associació per a cada base de dades que vulguem publicar. Polsarem "Agregar" per definir-la, ens demanarà que escollim el controlador que utilitzarem (en el nostre cas *Microsoft Access Driver)* i en polsar "Finalitzar" ens sol·licitarà una sèrie de dades:

Nom de l'origen de dades: Escollirem un nom únic per a la nostra base, que no necessàriament ha de coincidir amb el nom del fitxer i que serà el nom que utilitzarem des de les nostres pàgines quan vulguem accedir-hi amb la instrucció "Open" com es veurà més endavant.

Descripció: Una breu descripció, encara que no és imprescindible, pot ser útil sobretot si tenim diverses bases de dades.

Seleccionar: En polsar aquest botó se'ns obrirà un quadre de diàleg perquè escollim la ubicació del fitxer MDB que hem copiat o generat prèviament.

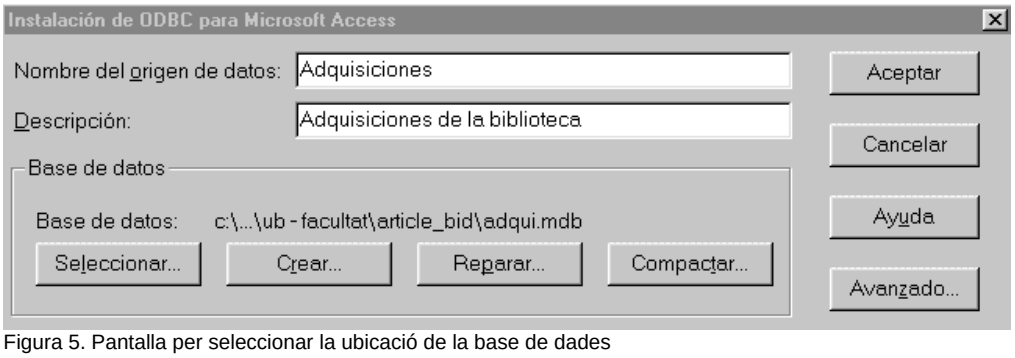

Figura 5. Pantalla per seleccionar la ubicació de la base de dades

Premem el botó "Acceptar" i ja està!, l'associació ja ha estat creada. Aquesta operació només s'ha d'executar una vegada, llevat que canviem d'ubicació la base de dades. No és necessari fer res quan modifiquem o afegim dades a la base sempre que no canviem ni el nom ni la ubicació.

Usuaris més experts poden accedir al botó "Avançat" per fixar altres paràmetres com ara identificacions d'accés, etc.

Crear els fitxers necessaris per a la seva consulta

En el cas de bases de dades petites, *Access* permet la creació de pàgines estàtiques compostes per taules HTML que reprodueixen les mateixes taules de la base de dades. L'usuari descarrega la totalitat de la taula cada vegada que accedeix via web, i el sistema de recerca es basa únicament en les capacitats de recerca que ofereix el mateix navegador. Com que són pàgines estàtiques han de ser regenerades manualment al servidor cada vegada que s'actualitza la base de dades, si es pretén mantenir actualitzada la consulta.

En el cas de bases grans o d'actualització freqüent, aquest sistema no resulta gens apropiat i hem de recórrer a consultes dinàmiques en les quals el servidor interacciona directament amb la base de dades i torna al client només els registres que compleixen la condició sol·licitada; per això aquest sistema resulta molt ràpid i totalment interactiu amb les dades actualitzades.

Existeixen dos mecanismes per realitzar aquest tipus de consulta dinàmica: un basat en fitxers IDC/HTX (Internet Database Connector) i un altre en fitxers ASP (Active Server Pages). En ambdós casos el llenguatge de consulta emprat és SQL (Structured Query Language). Poden trobar-se definicions ampliades d'aquests termes al glossari que s'adjunta al final de l'article.

Microsoft Access disposa d'un assistent capaç de generar les pàgines amb el codi necessari. Les versions d'*Office 97* inclouen una opció de "Guardar com a pàgina web" que llança aquest assistent; en el cas d'*Office 2000* ha d'accedir-se a "Crear una nova pàgina d'accés a dades". En el nostre cas generarem les pàgines manualment amb un editor HTML o fins i tot amb el bloc de notes. Usarem el mètode de pàgines *ASP*. Generarem les tres pàgines bàsiques per consultar qualsevol base de dades en el web:

- "formulario.htm": un formulari de consulta.
- "listado.asp": una pàgina de resultats intermedis de la consulta a manera de llista.
- "ficha.asp": una pàgina amb el resultat final d'un registre.

El formulari

Dissenyem una pàgina web amb el codi següent:

```
<html>
<head>
</head>
<body>
<h1 align="center">Formulario de búsqueda</h1>
 <form method="post" action="listado.asp">
    Buscar : <input name="texto">
    en campo :
    <select name="campo">
       <option value="titulo">Titulo</option>
       <option value="Autores">Autores</option>
    </select>
    <input type="submit" value="consultar">
 </form>
```

```
</body>
</html>
```

Figura 6: Codi font de formulario.htm i aspecte de la pàgina

Veure formulari

Nota: Simplement visualitzi el codi del formulari, més endavant ja podrà provar-lo.

En realitat aquest codi és fàcilment interpretable per qualsevol persona amb coneixements bàsics d'HTML; es tracta d'un formulari amb dos camps, un de tipus text (anomenat *text*) el valor del qual estarà definit per l'usuari en entrar-lo a la casella corresponent i un de tipus selector (anomenat *camp*) el valor del qual està predefinit pel dissenyador de la pàgina i l'usuari es limitarà a escollir entre les opcions. En aquest cas cada opció conté el nom del camp de la base de dades sobre el qual volem buscar el text.

Potser el més interessant és veure que, si bé habitualment els formularis són enviats directament a programes anomenats CGI (Common Gateway Interface) que recullen els camps del formulari per realitzar accions sobre les bases de dades i tornar els resultats al navegador, en aquest cas, el formulari envia les dades directament a un fitxer amb extensió ASP.

La llista

En segon lloc dissenyarem un fitxer que anomenarem "listado.asp" que contingui el codi següent:

```
<%@language="VBScript"%>
<html>
<head>
   <title>Resultados de la búsqueda</title>
</head>
<body>

<%
dim campo, texto
campo = request.form("campo")
texto = request.form("texto")
set cn=Server.createObject("ADODB.Connection")
cn.ConnectionTimeout=20
cn.open "DSN=Adquisiciones;DriverId=25; FIL=MS Access;MaxBufferSize=512;PageTimeout=5;"
set rs=Server.CreateObject("ADODB.Recordset")
rs.Open "SELECT * FROM libros_pedidos WHERE InStr(Ucase(" & campo & "), Ucase('" & texto & "') )>0", cn
%>

<h1>Resultado de la búsqueda</h1>
<ul>

<%
Do until rs.eof
   Response.write "<li><a href=ficha.asp?Id_documento=" & rs.fields("Id_documento") & "> Ver ficha</a> -- "
   Response.write  rs.fields("titulo")
   Response.write chr(10)
rs.MoveNext
loop
%>
</ul>

<%
rs.close
cn.close
set rs = Nothing
set cn = Nothing
%>

</body>
</html>
```

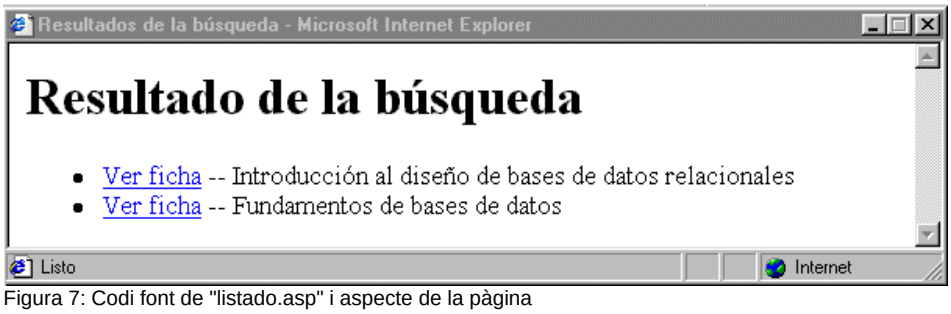

Figura 7: Codi font de "listado.asp" i aspecte de la pàgina

L'ús dels tres colors en el codi és per indicar que aquest fitxer conté tres tipus de llenguatges: el negre representa codi HTML, el blau és Visual Basic Script i el vermell és SQL. La interpretació d'aquest codi és una mica més complicada però intentarem explicar-ho d'una manera senzilla.

En primer lloc hem de remarcar que tots els fitxers ASP s'inicien amb la línia <% @language="VBScript"%> que indica al navegador que el fitxer que s'està llegint conté codi informàtic en llenguatge Visual Basic Script. ¿Com reconeix el navegador què és HTML i com què és codi VBScript? Observem que cada vegada que s'inicia codi informàtic va precedit de <% i finalitza amb %> . El contingut entre ambdues marques no es visualitzarà com HTML sinó que es processarà.

Vegem què fa aquest codi.

Uns petits coneixements de programació seran molt útils en aquest punt. A grans trets direm que la programació combina l'entrada de dades (*input*), el seu emmagatzemament en variables, el seu processament i la seva sortida (*output*). En VBScript hi ha diverses maneres de rebre dades. En aquest cas request.form és la funció que recupera els camps del formulari que ha invocat aquesta pàgina i els emmagatzema en les variables.

VBScript és un llenguatge de programació orientat a objectes que empra objectes del tipus ADO (Activex Data Objects). La programació clàssica es basa en una sèrie d'instruccions que el processador executa seqüencialment. La programació orientada a objectes es basa en la definició d'una sèrie d'objectes amb unes característiques o propietats que el programa modifica dinàmicament. Aquest tipus de programació és més flexible ja que no està condicionada per una seqüencialitat. A efectes del nostre exemple necessitarem dos objectes, un objecte connexió per comunicar-nos amb el servidor i un objecte taula per emmagatzemar i manipular els resultats de la consulta.

La línia set cn=Server.createObject ("ADODB.Connection") crea un objecte connexió i l'emmagatzema amb el nom **cn** per al seu ús posterior. La línia cn.open "DSN=Adquisiciones;DriverId=25; FIL=MS Access;MaxBufferSize=512;PageTimeout=5" obre la connexió amb la base de dades. Observeu que li especifiquem la base de dades que utilitzarem. En realitat, dient-li simplement el nom de l'origen de dades (DSN) definit en la connexió ODBC cn.open "DSN=Adquisiciones" en tindríem prou però normalment s'afegeixen la resta de paràmetres.

La línia set rs=Server.CreateObject ("ADODB.Recordset") crea un objecte taula i la següent línia l'omple amb els registres que compleixen una condició concreta. Encara que la sintaxi SQL pot semblar complicada, és interessant veure que es compon simplement d'algunes ordres com SELECT (especifica els camps de les taules que volem usar per a la visualització o tots si usem un asterisc), FROM (especifica les taules de la base de dades que usarem), WHERE (especifica la condició que han de complir els registres), ORDER BY (especifica el camp pel qual volem veure'ls ordenats). Aquesta instrucció es llegiria: *selecciona tots els camps de la taula "libros_pedidos" que continguin el valor de la variable text al camp que té per nom el valor de la variable camp.* Per complicar-ho una mica més, com que la recerca és sensible a majúscules i minúscules, prèviament es converteixen a majúscula ambdós costats de la igualtat. No hem d'espantar-nos si no entenem totalment aquesta sintaxi, més endavant veurem un petit truc amb *Access* per usar-la sense necessitat de conèixer-la.

Una vegada ja tenim el conjunt de registres, el fragment de codi següent usa el que en programació es coneix com un bucle Do until/rs.movenext/loop per mostrar a la pàgina Response.write els camps rs.fields que vulguem. La instrucció Do until rs.eof es llegeix: *fer mentre no es trobi el final de la base* (eof=end of file).

Observem la línia Response.write "Ver ficha". En aquest cas estem creant un hipervincle que ens permeti enllaçar cada un dels resultats intermedis amb la fitxa completa. Aquesta línia combina frases literals entre cometes amb camps de la base de dades units per l'operador &.

Aquesta línia es repeteix per a cada un dels registres trobats ja que està dins del bucle i en cada cas l'hipervincle generat cridarà la pàgina "ficha.asp" però passant-li el camp "ID_documento" (que és el camp clau) amb un valor diferent corresponent al registre en el qual estigui situat.

Finalment, el fragment final de codi VBScript tanca la taula rs.close i la connexió cn.close i destrueix els objectes creats set rs = Nothing, set cn = Nothing.

Provar formulari

Nota: Busqui el terme "datos" ja que l'exemple conté bibliografia sobre el tema.

La fitxa final

Finalment dissenyarem un fitxer que anomenarem "ficha.asp"que contingui el codi següent:

```
<%@language="VBScript"%>
<html>
<head>
  <title>Ficha final</title>
</head>
<body>

<%
dim registro
registro = Request.QueryString("ID_documento")
set cn=Server.createObject("ADODB.Connection")
cn.ConnectionTimeout=20
cn.open "DSN=Adquisiciones;DriverId=25; FIL=MS Access;MaxBufferSize=512;PageTimeout=5;"
set rs=Server.CreateObject("ADODB.Recordset")
rs.Open "SELECT * FROM libros_pedidos WHERE ID_documento=" & registro , cn
%>

<h1 align="center"> Documento <%=rs.fields("ID_documento") %> </h1>
<p>
<hr>
Título : <%=rs.fields("Titulo") %> <br>
Autores: <%=rs.fields("Autores") %> <br>
ISBN: <%=rs.fields("ISBN") %> <br>
<hr>

<%
rs.close
cn.close
set rs = Nothing
set cn = Nothing
%>

</body>
</html>
```

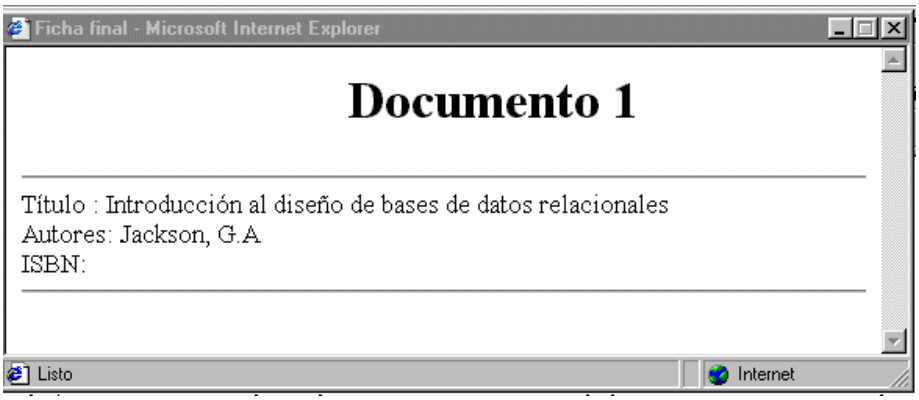

Figura 8. Codi de "ficha.asp" i aspecte de la recerca

Si hem comprès l'explicació del punt anterior observarem que aquesta pàgina és bastant similar, encara que conté algunes diferències que és necessari comentar.

Observem que en aquest cas només definim una variable registre per emmagatzemar l'identificador que rebem de la pàgina anterior. La funció que ens omple aquesta variable és Request.QueryString en comptes de request.form. La raó és que en aquest cas l'identificador ens arriba des de la mateixa cadena dins de l'URL en comptes d'arribar-nos des d'un formulari.

request.form recupera variables des de camps de formulari (INPUT, SELECT, RADIO, etc.)
request.queryString recupera variables des d'URL (pagina.asp?campo=valor1&campo2=valor2)

La selecció SQL en aquest cas és més senzilla, simplement recuperarem el registre que compleix la

condició "ID_documento = valor" rebut de l'URL. Com que es tracta d'un camp clau amb valors únics, simplement recuperarem un registre.

En aquesta pàgina no emprem cap bucle ja que simplement tenim un registre i volem visualitzar els camps. Una diferència respecte a la pàgina anterior és que aquí per visualitzar emprem <%=rs.fields en comptes de Response.write. En realitat ho podríem haver fet de la manera següent:

```
Response.write "Autor :" & rs.fields('autor') & "<br>"
Response.write "Titulo :" & rs.fields('titulo') & "<br>"
Response.write "ISBN :" & rs.fields('ISBN')& "<br>"
```

Ambdues formes són vàlides, potser la primera resulta més senzilla.

Atenció: Si el camp "ID_documento" no fos clau i la base pogués tenir més d'un registre, simplement visualitzaríem les dades del primer registre trobat.

Provar formulari

Nota: Busqui el terme "datos" i polsi sobre "ver ficha" per a un dels resultats intermedis. Visualitzi el codi font de la pàgina intermèdia per veure què ha fet el bucle.

Copiar els fitxers al servidor de web i provar-ho

Una vegada creats els tres fitxers simplement haurem de copiar-los al servidor de web i ja tindrem la nostra base de dades consultable. Evidentment això és una visió molt simplificada que ens servirà únicament per comprendre vagament el mecanisme de consulta. En aquest exemple hem prescindit completament del control d'errors. Suposem que introduïm al formulari un text del qual no existeix cap document. ¿Què succeeix? La instrucció SQL no recupera cap registre i posiciona el punter de la base al final, raó per la qual el bucle no s'executa. A la pràctica no apareix gens per pantalla. Una bona programació ha de preveure aquesta eventualitat incloent un codi com el següent abans o després del bucle:

```
if rs.eof and rs.bof then
Response.write "Lo siento. Ningún registro cumple esa condición"
end if
```

*Nota: Quan una taula és buida, el punter és al principi i al final de la base alhora i, per tant,*bof (begin of file) *i*eof (end of file) *coincideixen.*

Provar formulari ABANS d'afegir la condició

Provar formulari DESPRÉS d'afegir la condició

Nota: Busqui el terme "aspirina" que no figura a la base de dades.

Algunes aplicacions interessants

Llista de noves adquisicions

El desenvolupament anterior es basava en un model en què l'usuari interaccionava amb el sistema fent una pregunta: Quins documents hi ha que tinguin tal contingut en un camp? De vegades volem fer pàgines dinàmiques amb una condició preestablerta. Suposem que volem penjar una pàgina web amb les noves adquisicions, entenent com a tals les rebudes en els últims trenta dies. Podem fer-ho d'una manera estàtica, generant una taula el primer dia de cada mes, o d'una manera dinàmica generant una pàgina ASP que faci la consulta i que no necessitarem modificar amb el pas del temps.

Com seria això en *Access*?

Access ens permet veure el codi SQL de les consultes generades amb l'assistent i, per tant, podem tallar i enganxar aquest codi des d'*Access* a la nostra pàgina ASP. Aquest senzill truc ens estalvia conèixer SQL. Usant el generador d'expressions podem fer una consulta en Access per recollir aquells documents en els quals la data actual menys la data de recepció ("campo recibido_el") no superi els trenta dies. Evidentment hem de conèixer com funciona el generador d'expressions d'*Access* i aquest exemple potser no serà dels senzills. El resultat seria com el següent:

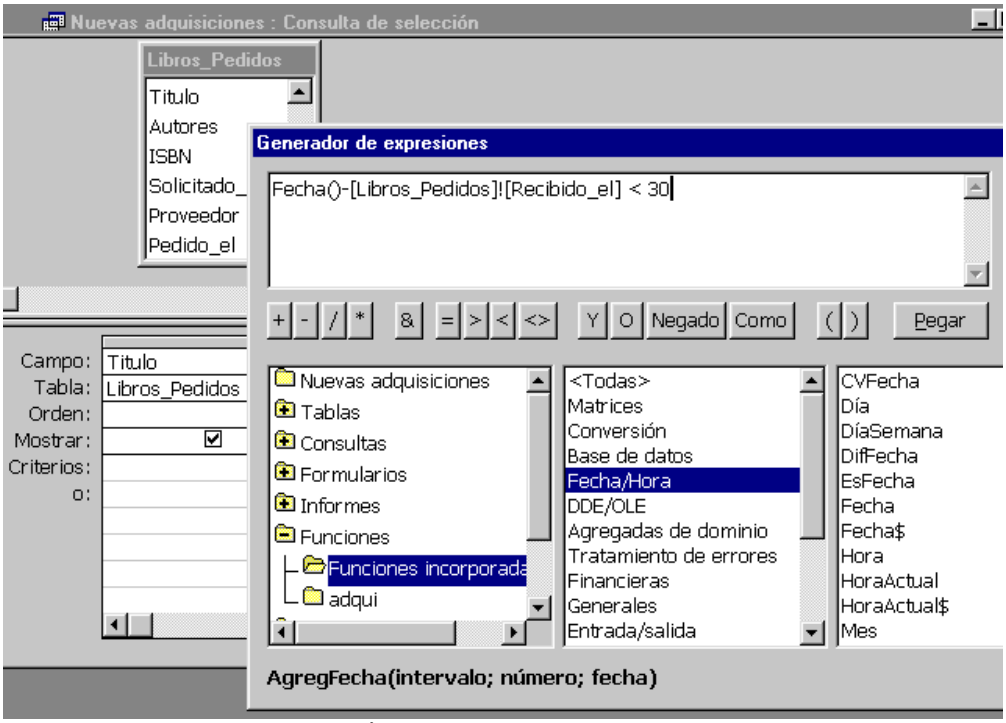

Figura 9. Consulta de llibres rebuts els últims 30 dies.

Si premem sobre la icona "vista" i escollim "vista SQL" veiem el codi SQL.

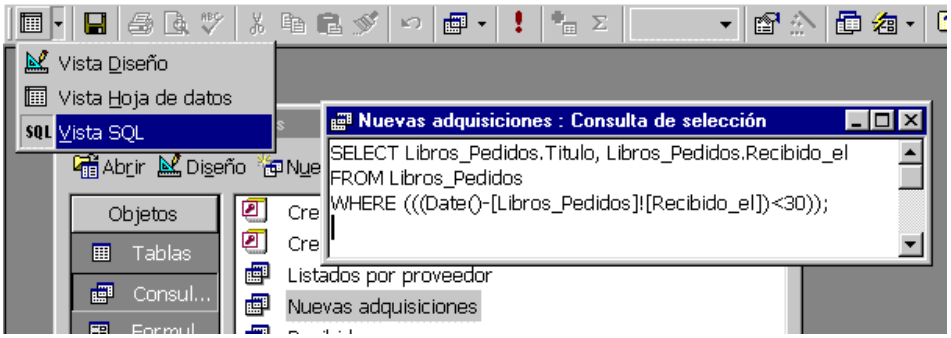

Figura 10. Vista SQL d'una consulta

Amb aquestes dades és fàcil dissenyar una pàgina que anomenarem "nuevas_adq.asp" el codi de la qual seria:

```
<%@language="VBScript"%>
<html>
<head>
   <title>Ultimas Adquisiciones</title>
</head>
<body>

<%
set cn=Server.createObject("ADODB.Connection")
cn.ConnectionTimeout=20
cn.open "DSN=Adquisiciones;DriverId=25;
FIL=MS Access;MaxBufferSize=512;PageTimeout=5;"
set rs=Server.CreateObject("ADODB.Recordset")
rs.Open "SELECT * FROM libros_pedidos WHERE (((Date()-[Libros_Pedidos]![Recibido_el])<30)) ORDER BY Recibido_el DESC;",
cn
%>

<h1 align=center> Ultimas adquisiciones recibidas en la biblioteca</h1>
<table width=100%>
<tr><th></th><th>Título</th><th>Recibido el</th></tr>

<%
Do until rs.eof
   Response.write "<tr><td><a href=ficha.asp?Id_documento=" & rs.fields("Id_documento") & ">Ver ficha</a></td>"
   Response.write "<td>" & rs.fields("titulo") & "</td>"
```

```
    Response.write "<td align='right'>" & rs.fields("recibido_el") & "</td></tr>"
    Response.write chr(10)
rs.MoveNext
loop
%>

</ul>

<%
rs.close
cn.close
set rs = Nothing
set cn = Nothing
%>

</body>
</html>
```

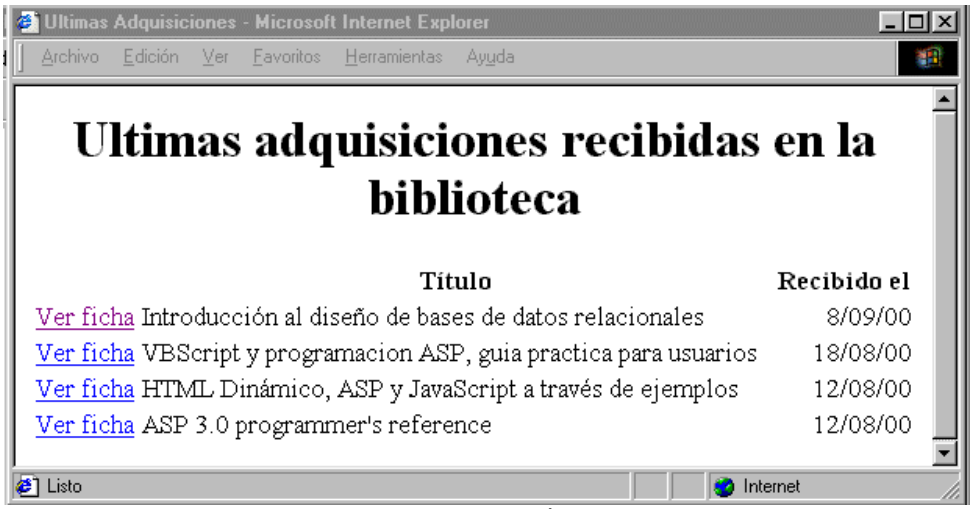

Figura 11. Codi font de "nuevas_adq.asp" i aspecte de la pàgina

Observem que el codi és fins i tot més senzill que els anteriors. En no haver-hi interacció de l'usuari no és necessari definir variables ni rebre dades. Aquesta pàgina pot ser cridada directament com un enllaç des de qualsevol pàgina HTML, i encara que aparentment sembla una pàgina estàtica, les dades canviaran en funció de la data. És interessant observar que, a la instrucció SQL, s'hi ha afegit un estament ORDER BY que permet ordenar els resultats de manera descendent pel camp "recibido_el"; d'aquesta manera l'últim llibre rebut apareix en primer lloc.

veure noves adquisicions a la biblioteca

Consultes per proveïdor

A partir d'aquí no importa com sigui de complexa la instrucció SQL, simplement hem de tallar-la des d'*Access* i enganxar-la a la nostra pàgina. Així, si mirem la instrucció SQL de la consulta d'unió definida per obtenir llistes per proveïdor veurem alguna cosa com ara això:

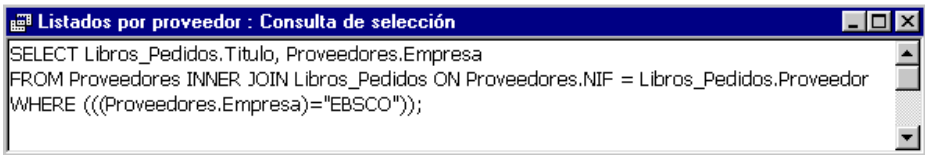

Figura 12. Vista SQL de la consulta *Access*

Substituint "EBSCO" per una variable heretada d'un formulari, podem construir un sistema per obtenir llistes per subministrador.

Veure formulari

Fins i tot podríem dissenyar un formulari dinàmic en el qual l'usuari, en comptes d'haver d'introduir el nom del proveïdor, el seleccionés d'un desplegable generat a partir de la taula de proveïdors. El codi seria:

```
<%@language="VBScript"%>

<%
set cn=Server.createObject("ADODB.Connection")
cn.ConnectionTimeout=20
cn.open "DSN=Adquisiciones;DriverId=25;
FIL=MS Access;MaxBufferSize=512;PageTimeout=5;"
set rs=Server.CreateObject("ADODB.Recordset")
rs.Open "SELECT * FROM proveedores", cn
%>

<html>
<head>
</head>
<body>
<h1 align="center"> Listado por proveedores</h1>
   <center>
   <form method="post" action="proveedor.asp">
      Seleccione proveedor
      <select name="texto">

<%
Do until rs.eof
   Response.write "<option value='" & rs.fields("empresa") & "'>" & rs.fields("empresa") & "</option>"
   Response.write chr(10)
rs.MoveNext
loop
%>

      </select>
      <input type="submit" value="consultar">
      </center>
   </form>
</body>
</html>

<%
rs.close
cn.close
set rs = Nothing
set cn = Nothing
%>
```

Figura 13. Codi font del desplegable dinàmic

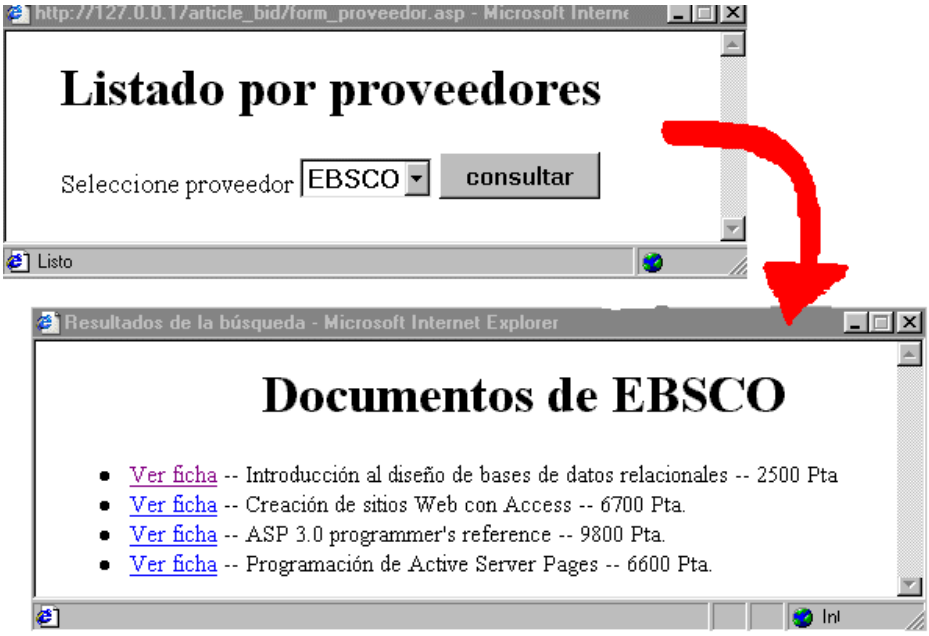

Figura 14. Aspecte de les pantalles de la llista per proveïdors

Veure formulari

Els usuaris més avançats poden usar Javascript per sumar els costos dels documents.

Captura de desiderates / bústia de suggeriments

Ens vam resistir a finalitzar aquest conjunt d'exemples sense mostrar al lector un exemple d'administració de bases de dades des de web, és a dir, permetre a l'usuari afegir registres a la base des del mateix navegador. En una biblioteca l'exemple podria ser el 'llibre de suggeriments' o de reclamacions; en el nostre exemple usarem un formulari per recollir l'opinió o els comentaris del lector sobre aquest article. Les dades del formulari quedaran recollides en una base "opinion.mdb" que conté una taula amb cinc camps (dia, hora, nom, correu, comentari). El sistema simplement consta d'un formulari amb tres camps que és enviat a un fitxer "opinion.asp" similar als descrits. El codi rellevant d'aquest és:

```
<%
set cn=Server.createObject("ADODB.Connection")
cn.ConnectionTimeout=20
cn.open "DSN=opinion;DriverId=25;FIL=MS Access;MaxBufferSize=512;PageTimeout=5;"
set rs=Server.CreateObject("ADODB.Recordset")
rs.Open "comentarios", cn, 1, 2, 2
rs.addnew
rs.fields("dia")= date
rs.fields("hora")= time
rs.fields("nombre")=request.form("nombre")
rs.fields("correo")=request.form("correo")
rs.fields("comentario")=request.form("comentario")
rs.update
%>
```

Figura 15. Fragment del codi font d'"opinion.asp"

Observem que després d'obrir la taula comentaris (sense necessitat de seleccionar registres) simplement tenim un mètode rs.addnew per afegir un registre en blanc seguit d'unes instruccions per omplir els camps amb els valors del formulari i un mètode rs.update per actualitzar realment el registre. Les constants date i time serveixen per omplir els camps respectius amb la data i hora en què es produeix l'actualització.

Observem igualment que el mètode rs.Open inclou uns paràmetres finals especials que permeten usar la base de dades per afegir registres (vegeu http://www.asp-magazine.com/fr/blitz/bdf5.asp).

Registrar la seva opinió

Veure les opinions enregistrades

El formulari per llistar les opinions registrades és similar a "listado.asp" descrit anteriorment.

Conclusions

Arribats a aquest punt podem intuir que les possibilitats enllaçant taules i formularis són pràcticament infinites i que, en moltes ocasions, l'important en el disseny dels processos no és la tècnica sinó la imaginació, les idees i el coneixement de la matèria. En aquest article s'han exposat quatre nocions bàsiques de tècnica i algunes idees o solucions pràctiques. Si el lector pertany al món de la biblioteconomia és possible que trobi la tècnica excessivament complicada, mentre que els processos exposats, lògics i intuïtius. Al contrari, un informàtic acostumat a la programació en C++ o altres sistemes gairebé segur que haurà considerat les nocions tècniques molt elementals, mentre que no necessàriament s'hauria imaginat les possibles aplicacions. Potser aquest és el problema de moltes de les solucions existents en el mercat; d'una banda, programes creats per experts informàtics que coneixen ben poc el món de les biblioteques i, d'una altra, l'absència d'una formació informàtica bàsica en el bibliotecari que li impedeix l'autosuficiència. La solució es basa en la col·laboració. Només si tècnica i coneixement de la matèria s'uneixen podrem obtenir uns resultats pràctics per als nostres fins.

Bibliografia

Anderson R, Denault D, Francis, B. (2000). *ASP 3.0 programmer's reference*. ISBN 1-86100-323-4.

Bobadilla Sancho, J [et al.] (1999) . HTML Dinámico, ASP y JavaScript a través de ejemplos. Madrid: Ra-ma. 511 p.

Codina, L. “Bases de datos relacionales: qué son y qué aportan a la gestión documental”. *Information World en español*, núm. 29 (1994), p. 18-19.

Cornell, G (1998). Aprenda Microsoft Visual Basic Script ya. Madrid: McGraw-Hil. 290 p. ISBN 84-481-

2004-3.

Figuerola C.G, Alonso Berrocal J.L, Zazo Rodríguez A.F. “Disseny d'un motor de recuperació de la informació per a ús experimental i educatiu”. *BiD,* núm 4, (juny 2000). [en línia] <http://www.ub.edu/bid/04figue1.htm> [consulta 14/08/2000]

Gonzalez Moreno, O. (1997). *VBScript y programación ASP, guía practica para usuarios*.

Hillier, S. (1998). Programación de Active Server Pages. Madrid: MacGraw-Hill; Interamericana de España. ISBN 84-481-1466-3.

Hobuss, James J.(1998). Creación de sitios Web con Access. Madrid: Prentice Hall. 497 p. ISBN 84-8322-131-4.

Kilcullen M. “Publishing a newspaper index on the World Wide Web using Microsoft Access 97”. *Indexer,* vol. 20, núm. 4 (1997) p. 195-6.

Pereda García A, Cruz Álvarez D. “Acceso a bases de datos remotas con Microsoft RDO”. *RPP: Revista Profesional para programadores*, vol. 3, núm 7, (1996) p. 41-44.

Salse, M. “Limitacions i possibilitats de Microsoft Access en la gestió documental”. *BiD,* núm. 2 (març 1999) [en línia] <http://www.ub.edu/bid/02salse.htm> [consulta 14/08/2000]

Trigueros Díaz, J. L.; Higuera Matas, R. “Bases de datos relacionales versus bases de datos documentales”. *Boletín de la Asociación Andaluza de Bibliotecarios*, vol. 13, núm. 49, (1997) p. 43-57.

Glossari

A l'apartat següent es presenten les definicions dels principals termes emprats al llarg de l'article així com adreces de manuals o informació accessible en Internet.

ActiveX

Conjunt de tecnologies d'interoperabilitat creades per Microsoft, independents del llenguatge que permeten que diferents components de programaris escrits en diferents llenguatges funcionin junts en entorns de xarxa. Els elements fonamentals de la tecnologia ActiveX són el model d'objectes components (COM) i el model d'objectes components distribuït (DCOM).

ADO

ActiveX Data Objects. Un conjunt d'interfícies d'accés a dades basades en objectes, optimitzades per a les aplicacions basades en Internet i centrades en dades. ADO està basat en una especificació publicada i s'inclou a Microsoft Internet Information Server i a Microsoft Visual InterDev.

ASP

Active Server Pages. Pàgines que combinen seqüències d'instruccions ActiveX que s'executen en el servidor amb seqüències HTML o altres components permetent crear aplicacions basades en Web.

Podem trobar manuals i informació d'ASP a la xarxa en http://webtech.metropoli2000.com/asp/index.html, http://www.montreat.edu/docs/aspdocs/roadmap.asp, http://www.learnasp.com/, http://www.stardeveloper.com/databases.asp i algunes revistes electròniques gratuïtes d'ASP en http://www.asptoday.com/ i http://www.asp-magazine.com.

CGI

Common Gateway Interface. S'anomenen així aquells programes dissenyats per interaccionar directament amb un servidor de web. Un programa convencional rep les dades d'entrada des del teclat o un altre dispositiu d'entrada i envia els resultats a la interfície Windows. Un programa CGI rep les dades d'entrada des del navegador i torna la resposta al mateix navegador.

IDC

Internet Database Connector. IDC és un component bàsic del servidor de web Internet Information Server que proporciona connectivitat de bases de dades entre les aplicacions de IIS i les bases de dades compatibles amb ODBC.

ODBC

Open Database Connectivity. Una interfície de programació que permet a les aplicacions tenir accés a dades des de diverses especificacions estàndard d'orígens de dades per a accés a bases de dades multiplataforma. Els venedors de bases de dades subministren controladors de baix nivell en conformitat amb la interfície per a ODBC.DLL. Els programadors d'aplicacions fan crides estàndard a ODBC.DLL per tenir accés a qualsevol base de dades sense importar el seu propi format.

SQL

Structured Query Languaje. Llenguatge de consulta i programació de bases de dades àmpliament utilitzat per tenir accés, consultar, actualitzar i administrar dades en sistemes de bases de dades relacionals.

Podem trobar alguns manuals de SQL a la xarxa en les adreces http://www.lobocom.es/~claudio/sql.html i http://www.pntic.mec.es/ies2000/iessql.htm.

VBScript

Microsoft Visual Basic Scripting Edition. Subconjunt del llenguatge Microsoft Visual Basic, un intèrpret ràpid, portàtil i lleuger per al seu ús en exploradors de World Wide Web i altres aplicacions que usen controls ActiveX, servidors d'automatització OLE i subprogrames Java.

Nota: Existeixen una gran quantitat de manuals, fins i tot en castellà, tant d'ASP com de SQL accessibles de manera gratuïta a Internet. S'aconsella utilitzar algun multicercador com Copernic (http://www.copernic.com) o un altre i buscar seqüències com SQL, ASP, al costat de termes com curs o manual. En molts casos trobarem informació sobre cursos presencials però també trobarem manuals electrònics.

Data recepció: 15/09/2000. Data acceptació: 2/10/2000.

Articles similars a BiD

- El catàleg d'arquitectura del Centre de Documentació de Projectes d'Arquitectura de Catalunya. Pouplana, Xavier; Poves, Lluís. (2005)
- Més enllà de la usabilitat : característiques mínimes exigibles per a les interfícies de bases de dades web. Rodríguez Yunta, Luis; Giménez Toledo, Elea. (2004)
- RCLIS : cap a una biblioteca digital de biblioteconomia i documentació. Barrueco Cruz, José Manuel; Subirats Coll, Imma. (2003)
- Revistes font en el SCI-E. Badia Segura, Maria; Alonso, José; Juan, Joan. (2003)
- Distribució de bases de dades en el web amb Knosys Internet. Abadal, Ernest; Martínez Rivas, Raúl. (2000)

Articles similars a Temària

- Bases de datos documentales en el web : análisis del software para su publicación. Abadal, Ernest. (2005)
- Fuentes de información para el análisis de la publicidad : las bases de datos publicitarias en Internet. Malalana Ureña, Antonio. (2004)
- Apuntes para una historia de la documentación deportiva. Arquesolo Vegas, José. (2000)
- Una propuesta metodológica para el diseño de bases de datos documentales (parte II). Codina, Lluís. (1997)
- Automatización de la información clínica odontológica : un proyecto de software íntegramente español. Sánchez Hernández, María F.. (1996)

Articles del mateix autor a Temària

Rodríguez Gairín, Josep Manuel

[més informació]

Facultat de Biblioteconomia i Documentació
Universitat de Barcelona
Barcelona, desembre de 2000
http://www.ub.edu/biblio • Comentaris

Citació recomanada • Metadades
UB • Facultat • BiD