

UNIVERSIDAD AUTÓNOMA DE QUERÉTARO
BIBLIOTECA
FACULTAD DE INFORMÁTICA



UNIVERSIDAD AUTÓNOMA DE QUERÉTARO
FACULTAD DE INFORMÁTICA



TESINA

Que para obtener el título de:

Licenciado en informática

Presenta

Paginas ASP con base de datos oracle 8i

Nombre del aspirante :

Jesús Eduardo Lamas Rivera

Generación 1998-2002

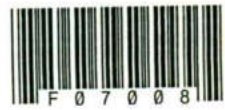
Nombre del asesor de tesina

I.S.C. Jabel Resendiz González

Querétaro, Qro. 20 de Enero de 2003

UNIVERSIDAD AUTÓNOMA DE QUERÉTARO
BIBLIOTECA
FACULTAD DE INFORMÁTICA

No. Adq. F07008
Clasif. TS 005.75
Cutter L2L6p



TS
005.75
L216p

F07008

TS
005.75
L216p

F07008



UNIVERSIDAD AUTÓNOMA DE QUERÉTARO
BIBLIOTECA
FACULTAD DE INFORMÁTICA



CARTA DE ACEPTACIÓN

Por este medio, se otorga constancia de aceptación de tesina para obtener el título de Licenciado en Informática, que presenta el pasante JESUS EDUARDO LAMAS RIVERA, con el tema denominado “PAGINAS ASP CON BASES DE DATOS ORACLE 8i”.

Este trabajo fue desarrollado como una investigación derivada del curso de titulación “ADMINISTRACIÓN DE BASES DE DATOS ORACLE 8i”, dando cumplimiento a uno de los requisitos contemplados en el artículo 34 del reglamento de titulación vigente, en lo referente a la opción de titulación por realización y aprobación de cursos de actualización.

Se extiende la presente para los fines legales a que haya lugar y para su inclusión en todos los ejemplares impresos de la tesina, a los 21 días del mes de enero de 2004.

ATENTAMENTE


I.S.C. JABEL RESENDIZ GONZÁLEZ

Índice

Capítulo I.

1.- Introducción a la programación en ASP.....	1
2.- Creación del archivo de acceso a datos (DSN).....	1
3.- Configurar DSN para una base de datos de Microsoft Access.....	2
4.- Configurar DSN para base de datos SQL server.....	2
5.- Información de conexión y seguridad de SQL.....	3
6.- Configuración DSN para base de datos Oracle.....	4
7.- Conexión con una base de datos.....	4
8.- Que es ADO.....	5
9.- Características ADO.....	6
10.- Abandonar los intentos de conexión.....	8
11.- Aceptar conexiones.....	8

Capítulo II.

12.- Inicio a la programación ASP.....	10
13.- Bucles y condiciones I.....	11
14.- Las condiciones IF.....	12
15.- Bucles y condiciones II.....	14
16.- Bucles y condiciones III Do While/loop.....	15
17.- objetos ASP.....	16
18.- Objeto Request I.....	16
19.- Transferir variables por URL.....	17
20.- Transferir variables por formulario.....	18
21.- Objeto Response.....	20
22.- Las cookies.....	22
23.- Objeto sesión.....	23
24.- Trabajar con bases de datos en ASP.....	26
24.- Selecciones en una tabla.....	25
25.- Creación de un nuevo registro.....	28
26.- Actualización de un registro existente.....	30
27.- Borrado de un registro.....	32
28.- Ejecución de consultas con el objeto connection.....	34
29.- Uso del objeto recordset para manipular resultados.....	36
30.- Recuperar un conjunto de registros.....	36
31.- Manejo de consultas con el objeto Command.....	38
32.- Manejo de conexiones a bd.....	40
33.- Detalles de IP.....	40
34.- Ip estático y dinámico.....	40
35.- Construir un DSN local.....	41

Capítulo III.

36.- IIS servidor web.....	42
37.- Directorio virtual.....	43
38.- Seguridad.....	45
39.- Clientes FTP.....	48
40.- IIS servidor de correo SMTP.....	49
41.- Archivo log.txt.....	52

Capítulo IV.

42.- Personal web server.....	53
43.- Instalación de PWS.....	54
44.- Directorios virtuales.....	58
45.- Utilizar paginas ASP por medio de Dreamweaver.....	64
46.- Crear un juego de registros de BD.....	64
47.- Insertar objeto de aplicación maestro detalle.....	67
48.- Visualizar las paginas ASP.....	69
49.- Crear una pagina para inserción de registro.....	70
50.- Añadir objetos de formulario.....	72
51.-Defina comportamiento de servidor insertar registro.....	75
52.- Comprobar la pagina.....	76
53.- Aspectos básicos de las conexiones BD con asp.....	77
54.- Aspectos básicos de los DSN.....	78
55.- Creación de conexión utilizando un DSN local.....	78
56.- Creación de conexión utilizando un DSN remoto.....	79
57.- Creación de conexión sin DSN.....	80
58.- Conexión de BD con cadena de conexión.....	81
59.- Como escribir una cadena de conexión.....	82
60.- Creación de una conexión OLE BD.....	84
61.- Utilización de ruta virtual para conectar con BD.....	84
62.- crear conexión sin DSN en el método mapPath.....	85
63.- Aspectos básicos de las rutas físicas y virtuales.....	86
64.- Localización de una ruta física de archivo con la ruta virtual.....	86
65.- Creación de la pagina de búsqueda.....	87
66.- Creación rápida de paginas maestro / detalle.....	100
67.- Introducción.....	iii
68.- Conclusión.....	iv
69.- Bibliografía.....	vi

INTRODUCCIÓN

En esta tesina explicaré la manera en que se crean las páginas web con respecto al código Visual Basic script los famosos ASP, antes de empezar en el desarrollo de la misma debo indicar que para crear páginas web tenemos diferentes capas, estas se pueden describir como sigue:

La primer capa la llamaremos capa de usuario: En esta capa cada persona que requiere ver la página ASP en su monitor se conecta por medio de una dirección llamada URL para que el servidor sepa que este usuario requiere una de sus paginas.

Segunda capa la llamaremos capa de interprete: En esta capa, lo que se realiza es que el servidor recibe una petición de parte del usuario que indica que quiere ver la pagina ASP que tiene guardada. El servidor por medio de un interprete verifica el código que la página ASP contiene, realiza las instrucciones que la página tiene programada, después de eso el mismo servidor envía al usuario que hizo la llamada por medio de la dirección URL el código de la página ASP, el usuario ve la página con extensión .ASP y realiza los cambios que necesite y después la cierre. cuando ve la página ASP en su monitor si quiere ver el código de esa página solo verá puro código HTML ya que el mismo interprete del servidor no permite revelar en código solamente en el mismo servidor.(Digamos que es una forma de proteger el código de la página).

La manera que se programa la página ASP es:

- 1.- Instalando el IIS o PWS en la máquina que se correrá la página.
- 2.- crear la página con el código que sea necesario en esta parte es poner el código para hacer inserciones, modificaciones, bajas, consultas, maestro detalle, etc. El desarrollador puede ayudarse de algunos interpretes el mas común es Dreamweaver.
- 3.- hacer la conexión de la página ASP con el servidor Oracle por medio de ODBC, esto cuando ya se tiene terminada la página y se requiere conectar al servidor para que funcione a bien.

Capítulo I

Introducción a la programación en ASP y conexión a bases de datos.

ASP (Active Server Pages) es la tecnología para la creación de páginas dinámicas del lado del servidor desarrollada por Microsoft. El tipo de servidores que emplean este lenguaje son aquellos que funcionan con sistema operativo de la familia de Windows NT. Afortunadamente, también podemos visualizar páginas ASP sobre Windows 95/98, pero esto lo veremos más adelante.

Para escribir páginas ASP se utiliza un lenguaje de scripts, que se colocan en la misma página web junto con el código HTML. Comúnmente este lenguaje de scripts es Visual Basic. Existe una versión de Visual Basic Script en el lado cliente y otra en el lado del servidor. En los dos casos, como su nombre indica, el lenguaje de base es Visual Basic por lo que su aprendizaje puede ser perfectamente coordinado, ya que las sentencias y las sintaxis son prácticamente las mismas. En ASP, al estar programando páginas del lado del servidor, se utiliza Visual Basic Script del lado del servidor.

Creación del Archivo de Acceso a Datos (DSN)

Antes de crear secuencias de comandos para bases de datos tiene que proporcionar una forma para que ADO se comunique con una base de datos.

Los controladores de bases de datos (programas que pasan información desde su aplicación Web a una base de datos) utilizan un Nombre de origen de datos (DSN) para encontrar e identificar una base de datos ODBC en particular. Normalmente, el DSN contiene información de configuración de la base de datos, seguridad de usuarios y ubicación, y puede tener la forma de una entrada en el registro del sistema de Windows NT o de un archivo de texto.

Con ODBC puede elegir el tipo de DSN que va a crear: Usuario, Sistema o Archivo.

Los DSN de Usuario y de Sistema residen en el registro del sistema de Windows NT. Los DSN de Sistema permiten que todos los usuarios que han

iniciado una sesión en un servidor concreto (Oracle 8i) tengan acceso a una base de datos, mientras que los DSN de Usuario limitan la conectividad con la base de datos a los usuarios que dispongan de las credenciales de seguridad apropiadas. Los DSN de Archivo, que tienen la forma de archivos de texto, proporcionan acceso a varios usuarios y son fácilmente transferibles entre un servidor y otro mediante la copia de los archivos DSN.

Pueden crear un DSN de Archivo si abren **Panel de control** desde el menú **Inicio** de Windows. Haga doble clic en el icono ODBC y seleccione la hoja de propiedades **DSN de Archivo**. Haga clic en **Agregar**, elija el controlador de la base de datos y haga clic en **Siguiente**.

Para configurar un DSN de Archivo para una base de datos de Microsoft Access

1. En el cuadro de diálogo **Crear nuevo origen de datos**, seleccione **Controlador Microsoft Access** en el cuadro de lista, después, haga clic en **Siguiente**.
2. Escriba el nombre de su DSN de Archivo y haga clic en **Siguiente**.
3. Haga clic en **Finalizar** para crear el origen de datos.
4. En el cuadro de diálogo **Instalación de ODBC para Microsoft Access**, haga clic en **Seleccionar**. Elija un archivo de base de datos de Microsoft Access (*.mdb) y haga clic en **Aceptar**.

Nota Por razones de rendimiento y confiabilidad, se recomienda encarecidamente que utilice un motor de base de datos cliente-servidor para el despliegue de aplicaciones Web controladas por datos que requieran una elevada demanda de accesos por parte de más de 10 usuarios simultáneos. Si bien Páginas Active Server funciona con cualquier base de datos compatible con ODBC, se ha probado y está diseñado para funcionar con bases de datos cliente-servidor como Microsoft SQL Server u Oracle.

ASP acepta bases de datos de archivos compartidos (Microsoft(r) Access o Microsoft(r) FoxPro, Oracle) como orígenes válidos de datos. Aunque algunos ejemplos de la documentación de ASP utilizan una base de datos de archivos compartidos, se recomienda utilizar estos tipos de motores de bases de datos únicamente con fines de desarrollo o en situaciones de despliegue limitado. Las bases de datos de archivos compartidos pueden no ser tan adecuadas como las bases de datos cliente-servidor para aplicaciones Web muy exigentes.

- Para configurar un DSN de Archivo para una base de datos de **SQL Server**

Nota el siguiente procedimiento utiliza la configuración predeterminada de ODBC para SQL Server, que puede no funcionar con su configuración hardware.

1. En el cuadro de diálogo **Crear nuevo origen de datos**, seleccione **SQL Server** en el cuadro de lista y haga clic en **Siguiente**.
2. Escriba el nombre de su DSN de Archivo y haga clic en **Siguiente**.
3. Haga clic en **Finalizar** para crear el origen de datos.
4. Escriba el nombre del servidor que ejecuta SQL Server, su Id. de inicio de sesión y su contraseña.
5. En el cuadro de diálogo **Crear un nuevo origen de datos SQL Server**, escriba en el cuadro de lista **Servidor** el nombre del servidor que contenga la base de datos de SQL Server y haga clic en **Siguiente**.
6. Seleccione el método de comprobar la autenticidad del Id. de inicio de sesión.
7. Si elige la autenticación SQL Server, escriba el Id. de inicio de sesión y la contraseña; después, haga clic en **Siguiente**.
8. En el cuadro de diálogo **Crear un nuevo origen de datos SQL Server**, establezca la base de datos predeterminada, la configuración de los procedimientos almacenados del controlador y los identificadores ANSI; después, haga clic en **Siguiente**. (Para obtener más información, haga clic en **Ayuda**.)
9. En el cuadro de diálogo (también se llama **Crear un nuevo origen de datos SQL Server**), elija el método de traducción de caracteres y haga clic en **Siguiente**. (Para obtener más información, haga clic en **Ayuda**.)
10. En el siguiente cuadro de diálogo (también se llama **Crear un nuevo origen de datos SQL Server**), seleccione las opciones de registro.
11. En el cuadro de diálogo **Instalación de ODBC para Microsoft SQL Server**, haga clic en **Comprobar origen de datos**. Si el DSN se ha creado correctamente, el cuadro de diálogo **Resultados de la comprobación** indicará que la comprobación es correcta.

Información de conexión y seguridad de SQL Server

Si va a desarrollar una aplicación ASP de base de datos diseñada para conectar con una base de datos remota de SQL Server, debe tener en cuenta también los aspectos siguientes:

Esquema de conexión Puede elegir entre los métodos Sockets TCP/IP y Canalizaciones con nombre para tener acceso a bases de datos remotas de SQL Server. Con Canalizaciones con nombre, los clientes de la base de datos tienen que estar autenticados por Windows NT antes de establecer una conexión, dejando la posibilidad de que un equipo remoto que ejecute canalizaciones con nombre pueda denegar el acceso a un usuario que tenga las credenciales apropiadas de acceso a SQL Server, pero no tenga una cuenta de usuario de Windows NT en dicho equipo.

Como alternativa, las conexiones que utilizan Sockets TCP/IP conectan directamente con el servidor de la base de datos. Como las conexiones establecidas mediante Sockets TCP/IP conectan directamente con el servidor de la base de datos, los usuarios pueden tener acceso mediante la

autenticación de SQL Server, no mediante la autenticación de Windows NT. Nota Para aumentar el rendimiento en la conexión con bases de datos remotas, utilice Sockets TCP/IP.

Seguridad Si utiliza las características de seguridad Integrada o Mixta de SQL Server y la base de datos de SQL Server reside en un servidor remoto, no podrá utilizar la autenticación Desafío/Respuesta de Windows NT. Específicamente, no puede reenviar credenciales Desafío/Respuesta de Windows NT al equipo remoto. Esto significa que quizá tenga que utilizar la autenticación Básica, que se basa en que el usuario proporcione la información del nombre de usuario y la contraseña.

Para configurar un DSN de Archivo para una base de datos de Oracle

Asegúrese de que el software cliente de Oracle esté correctamente instalado en el equipo donde vaya a crear el DSN. Para obtener más información, consulte con el administrador del servidor y la documentación de la base de datos.

1. En el cuadro de diálogo **Crear nuevo origen de datos**, seleccione **Controlador ODBC de Microsoft para Oracle** en el cuadro de lista y haga clic en **Siguiente**.
2. Escriba el nombre del DSN de Archivo y haga clic en **Siguiente**.
3. Haga clic en **Finalizar** para crear el origen de datos.
4. Escriba un nombre de usuario, una contraseña y el nombre del servidor; después, haga clic en **Aceptar**.

Nota Los archivos DSN tienen la extensión .DSN y residen en el directorio \Archivos de programa\Archivos comunes\ODBC\Data Sources.

Conexión con una Base de Datos

El primer paso para tener acceso a la información de una base de datos consiste en establecer una conexión con el origen de datos. ADO proporciona el objeto **Connection**, que puede utilizar para establecer y administrar las conexiones entre sus aplicaciones y las bases de datos de ODBC. El objeto **Connection** incorpora diversas propiedades y métodos que puede utilizar para abrir y cerrar conexiones con bases de datos, y para enviar consultas de actualización de la información.

Para establecer una conexión con una base de datos, cree primero una instancia del objeto **Connection**. Por ejemplo, la siguiente secuencia de comandos crea una instancia del objeto **Connection** y procede a abrir una conexión:

```
<%  
'Crea un objeto Connection
```

```
Set cn = Server.CreateObject("ADODB.Connection")
'Abre una conexión; la cadena hace referencia al DSN
cn.Open "FILEDSN=MibaseDeDatos.dsn"
%>
```

Nota La cadena DSN no contiene espacios en blanco ni antes ni después del signo igual (=).

En este caso, el método **Open** del objeto **Connection** hace referencia al DSN de Archivo, que contiene la información de ubicación y la configuración de la base de datos. Opcionalmente, también puede hacer referencia directamente a un proveedor, origen de datos, Id. de usuario y contraseña, en lugar de al DSN Conexión a la base de datos

Una vez que tenemos creada la base de datos en nuestro servidor (oracle), el siguiente paso es conectarnos a la misma desde una página ASP.

Para acceder a la base de datos usaremos **ADO** (ActiveX Data Objects), ADO son un conjunto de objetos que nos permiten acceder a la base de datos independientemente del motor de base de datos que usemos, así pues estos ejemplos usan MS Access pero funcionarían igual si el motor de Base de Datos fuese oracle. Tan solo habría que cambiar el driver.

```
<html>
<head>
  <title>Ejemplo de ASP</title>
</head>
<body>
<%
Dim oConn
Set oConn = Server.CreateObject("ADODB.Connection")
oConn.Open("DRIVER={Microsoft Access Driver (*.mdb)}; DBQ=" &
Server.MapPath("\webestilo\db\datos.mdb"))
Response.Write("Conexión con la base de datos conseguida.<br>")
oConn.Close
set oConn = nothing
%>
</body>
</html>
```

Para conectarnos a la Base de datos creamos un objeto de ADO de tipo conexión para ello usamos `Server.CreateObject`, una vez que tenemos el objeto, le indicamos el driver que tiene que usar, y en donde se encuentra la base de datos.

Para cerrar la conexión con la base de datos usaremos el método Close y seguidamente como ya no usaremos el objeto lo destruimos asignándole nothing.

Qué es ADO?

ActiveX Data Objects (ADO) es una tecnología ampliable y de fácil uso para agregar acceso a bases de datos a sus páginas Web. Puede utilizar ADO para escribir secuencias de comandos compactas y escalables que conecten con bases de datos compatibles con Open Database Connectivity (ODBC, Conectividad abierta de bases de datos) y orígenes de datos compatibles con OLE DB. Si no tiene mucha experiencia en conectividad con bases de datos, encontrará que las instrucciones de ADO son accesibles y no complicadas. Del mismo modo, si ya tiene experiencia en la programación con bases de datos, apreciará las características avanzadas de conexión y de manipulación de consultas independientes del lenguaje de ADO.

Nota. Si se desean usar las constantes de ADO se deben incluir los siguientes archivos.

```
<!--#includevirtual="/PROGRAM  
FILES/COMMONFILES/SYSTEM/ADO/ADOVBS.INC"--> ó
```

```
<!--#include virtual="/PROGRAM  
FILES/COMMONFILES/SYSTEM/ADO/ADOJAVAS.INC" -->
```

Características de ADO

- Objetos creados independientemente.
- Ayudas de modificación en lote.
- Soporte para procedimientos almacenados.
- Diferentes tipos de cursores.
- Soporte para retornar múltiples recordsets
- Aplicaciones libres de Hilos para aplicaciones de servidor web eficientes.

Objetos ADO

- **Connection:** Mantiene la información de conexión con el proveedor de datos.

```
set VarConexion = Server.CreateObject(ADODB.Connection)
```

Métodos: **Open:** VarConexion.Open ConnectionString [,UserID] [,Password]

Close: VarConexion.Close

Execute: VarConexion.Execute CommandText,RecordsAffected,Options

Propiedades: CommandTimeout, connectionString, ConnectionTimeout, State, Version, DefaultDataBase

Command: Mantiene información acerca de un comando

```
set VarComando = Server.CreateObject(ADODB.Command)
```

Métodos:

CreateParameter:

```
Set varParametro= command.CreateParameter(Name, Type, Direction, Size, Value)
```

Execute: Set recordset = command.Execute(RecordsAffected, Parameters, Options)

Propiedades: ActiveConexionComandtext, CommandTimeout, CommandType Name, Prepared, Satate.

Error: Mantiene información acerca de algún error ocurrido en el proveedor

```
set VarError = Server.CreateObject(ADODB.Error)
```

Field: Mantiene información de un único campo dentro de un recordset

```
set VarConexion = Server.CreateObject(ADODB.Connection)
```

Parameter: Es un parámetro para ser usado en un objeto command.

```
set VarParametro = Server.CreateObject(ADODB.Parameter)
```

- **Property:** Características de un objeto.

```
set VarPropiedad = Server.CreateObject(ADODB.Property)
```

- **Recordset:** Conjunto de registros retornados desde una consulta con un cursor asociado.

```
set VarConexion = Server.CreateObject(ADODB.Recordset)
```

Métodos:

AddNew: recordset.AddNew Fields, Values

CancelUpdate: recordset.CancelUpdate

Close: object.Close

Delete: recordset.Delete AffectRecords

GetRows: array = recordset.GetRows(Rows, Start, Fields)

Move: recordset.Move NumRecords, Start

Movefirst, MoveNext, MoveLast, MovePrevious: recordset.MoveX

Open: recordset.Open Source, ActiveConnection, CursorType, LockType, Options

Requery: recordset.Requery

Update: recordset.Update Fields, Values

Supports: boolean = recordset.Supports(CursorOptions)

Propiedades:

AbsolutePage, AbsolutePosition, ActiveConnection, BOF, Bookmark, CacheSize, CursorLocation, CursorType, EditMode, EOF, Filter, LockType, MarshalOptions, MaxRecords, PageCount, PageSize, RecordCount, Source, State, Status

Abandonar los intentos de conexión

Un servidor de base de datos que experimente un repentino incremento en su actividad puede quedar saturado en sus conexiones, lo que incrementa considerablemente el tiempo necesario para establecer una nueva conexión. Como resultado, los retrasos excesivos en la conexión pueden reducir el rendimiento de su aplicación de base de datos.

Con la propiedad **ConnectionTimeout** del objeto **Connection** puede limitar la cantidad de tiempo que su aplicación espera antes de abandonar un intento de conexión y emitir un mensaje de error.

Por ejemplo, la siguiente secuencia de comandos establece la propiedad **ConnectionTimeout** para esperar veinte segundos antes de cancelar el intento de conexión:

```
Set cn = Server.CreateObject("ADODB.Connection")
cn.ConnectionTimeout = 20
cn.Open "FILEDSN=MiBaseDeDatos.dsn"
```

El valor predeterminado de la propiedad **ConnectionTimeout** es 30 segundos.

Nota Antes de incorporar la propiedad **ConnectionTimeout** a sus aplicaciones de bases de datos, asegúrese de que su proveedor de conexiones y el origen de datos acepten esta propiedad.

Aceptar conexiones

Las aplicaciones Web de bases de datos que establecen y terminan con frecuencia conexiones con bases de datos pueden reducir el rendimiento del servidor de base de datos.

ASP acepta la eficiente administración de las conexiones que se consigue mediante la función de agrupamiento de conexiones de ODBC. El agrupamiento de conexiones mantiene abiertas las conexiones con las bases de datos y administra el uso compartido de las conexiones entre distintas peticiones de usuario para mantener el nivel de rendimiento y reducir el número de conexiones inactivas.

En cada petición de conexión, el conjunto de conexiones determina si hay alguna conexión inactiva en el conjunto. Si es así, el conjunto de conexiones devuelve una de ellas en lugar de establecer una nueva conexión con la base de datos.

Si desea que su controlador ODBC participe en el conjunto de conexiones, debe configurar su controlador específico de base de datos y después establecer la propiedad **CPTimeout** del controlador en el registro del sistema de Windows NT.

Cuando ODBC desconecta una conexión, la conexión se guarda en un conjunto, en lugar de desconectarse.

La propiedad **CPTimeout** determina el periodo de tiempo de inactividad que una conexión puede permanecer en el conjunto de conexiones. Si la conexión permanece en el conjunto durante un periodo de tiempo mayor que el establecido en **CPTimeout**, la conexión se cierra y se quita el conjunto. El valor predeterminado de **CPTimeout** es 60 segundos.

Puede establecer selectivamente la propiedad **CPTimeout** para permitir el agrupamiento de conexiones por cada controlador específico de base de datos ODBC si crea una entrada en el registro del sistema con los siguientes valores:

```
\HKEY_LOCAL_MACHINE\SOFTWARE\ODBC\ODBCINST.INI\controlador
\CPTimeout = timeout
(REG_SZ, en segundos)
```

Por ejemplo, la clave siguiente establece el tiempo de inactividad del conjunto de conexiones a 180 segundos (3 minutos) para el controlador de ORACLE

```
\HKEY_LOCAL_MACHINE\SOFTWARE\ODBC\ODBCINST.INI\ODBC      para
ORACLE\CPTimeout = 180
```

Nota De forma predeterminada, su servidor Web activa el conjunto d

Capítulo II

Programación en ASP

Nos ha quedado claro que el ASP es un lenguaje orientado a las aplicaciones en red creado por Microsoft que funciona del lado servidor. Es en efecto el servidor quien se ocupa de ejecutarlo, interpretarlo y enviarlo al cliente (navegador) en forma de código HTML.

ASP es principalmente utilizado sirviéndose del lenguaje Visual Basic Script que no es más que una versión light del Visual Basic. Sin embargo, es posible programar páginas ASP en Java Script. Lo único que hay que hacer es especificar en la propia página qué tipo de lenguaje estamos utilizando.

Dado que el lenguaje ASP está muy frecuentemente embebido dentro del código HTML, es importante poder marcar al servidor qué partes están escritas en un lenguaje y cuáles en otro. Es por ello que todas las partes del archivo que están escritas en ASP estarán siempre delimitadas por los símbolos: `<% y %>`.

De este modo, cuando realicemos nuestros scripts, lo primero que debemos definir es el tipo de lenguaje utilizado, lo cual se hace del siguiente modo:

```
<% @ LANGUAGE="VBSCRIPT" %> Para el caso en el que programemos en  
Visual Basic Script
```

Los scripts que serán presentados en estarán basados en el VBS, el cual presenta toda una serie de prestaciones que lo hacen sin duda más accesible y apto para ASP.

Con los elementos que he presentado hasta ahora, ya estamos en situación de poder escribir nuestro primer programa en ASP. Vamos a crear un programa que calcule el 20% de impuestos que habría que añadir a una serie de artículos. Para plasmar el concepto de función, vamos a definir una función "impuesto" que emplearemos sucesivas veces. El programa podría resultar algo así:

```
<% @ LANGUAGE="VBSCRIPT" %>  
<HTML>
```

```
<HEAD>
<TITLE>Funcion impuesto</TITLE>
</HEAD>
<BODY>
<%Function impuesto(precio_articulo)
precio_final=precio_articulo+precio_articulo*20/100
Response.Write precio_final
End Function%>
Un libro de 3500 se quedará en un precio de <% impuesto(3500) %>
<br>
Una camisa de 6000 tendrá un precio final de <% impuesto(6000) %>
<br>
Un CD de música de 2000 costaría <% impuesto(2000) %>
</BODY>
</HTML>
```

Como puede verse, el script contiene dos partes fundamentales: Una primera en la que definimos la función que llamamos impuesto que depende únicamente de una variable (precio_artículo). Impuesto permite añadir un 20% al precio del artículo e imprimir el resultado en pantalla (Response.Write). En la segunda parte nos servimos de la función para realizar los cálculos necesarios y mostrarlos en pantalla acompañados de texto.

Resulta muy interesante, una vez ejecutado el script, ver el código fuente. Como puede verse, el código HTML que muestra el browser no coincide con el que nosotros hemos escrito. Algo que no debe sorprendernos ya que, como ya hemos explicado, el servidor se encarga de procesarlo y hacerlo comprensible al navegador.

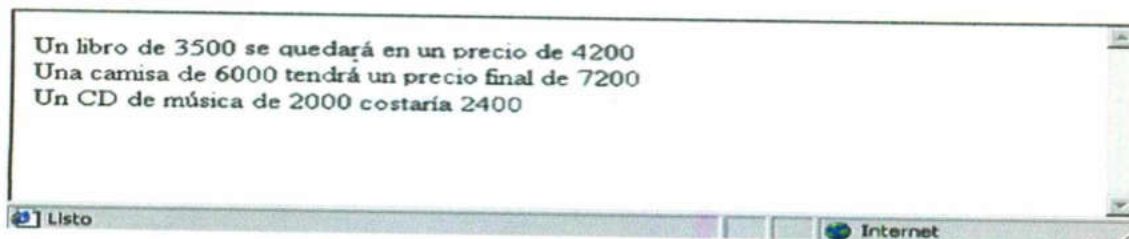


Fig. 2.1 Referencia al código que muestra el resultado de la función anterior

Bucles y condiciones I

La programación exige en muchas ocasiones la repetición de acciones sucesivas o la elección de una determinada secuencia y no de otra dependiendo de las condiciones específicas de la ejecución.

Como ejemplo, podríamos hacer alusión a un script que ejecute una secuencia diferente en función del día de la semana en el que nos encontramos.

Este tipo de acciones pueden ser llevadas a cabo gracias a una paleta de instrucciones presentes en la mayoría de los lenguajes. En este capítulo describiremos someramente algunas de ellas propuestas por el VBS y que resultan de evidente utilidad para el desarrollo de páginas ASP.

Para evitar complicar el texto, limitaremos a introducir las más importantes dejando de lado otras cuantas que podrán ser fácil de comprender

Las condiciones: IF

Cuando queremos que el programa, llegado a un cierto punto, tome un camino determinado en determinados casos y otro diferente si las condiciones de ejecución difieren, nos servimos del conjunto de instrucciones If, Then y Else. La estructura de base de este tipo de instrucciones es la siguiente:

```
IFcondiciónTHEN
  Instrucción1
  Instrucción2
ELSE
  InstrucciónA
  InstrucciónB
ENDIF
```

Llegados a este punto, el programa verificará el cumplimiento o no de la condición. Si la condición es cierta las instrucciones 1 y 2 serán ejecutadas. De lo contrario (Else), las instrucciones A y B serán llevadas a cabo. Una vez finalizada la estructura, deberemos cerrar con un End If.

Esta estructura de base puede complicarse un poco más si tenemos cuenta que no necesariamente todo es blanco o negro y que muchas posibilidades pueden darse. Es por ello que otras condiciones pueden plantearse dentro de la condición principal. Hablamos por lo tanto de condiciones anidadas que tendrían una estructura del siguiente tipo:

```
IFcondiciónTHEN
  Instrucción1
  Instrucción2
ELSE
  IFcondición2THEN
    InstrucciónA
    InstrucciónB
  ELSE
    InstrucciónX
```


ENDIF
ENDIF

De este modo podríamos introducir tantas condiciones como queramos dentro de una condición principal. En este tipo de estructuras es importante cerrar correctamente cada uno de los IF con sus END IF correspondientes. De gran ayuda es la instrucción ELSE IF que permite en una sola línea y sin necesidad de añadir un END IF introducir una condición anidada.

El uso de esta herramienta resultará claro con un poco de práctica. Pongamos un ejemplo sencillo de utilización de condiciones. El siguiente programa permitiría detectar la lengua empleada por el navegador y visualizar un mensaje en dicha lengua

```
<%@LANGUAGE="VBSCRIPT"%>  
<HTML>  
<HEAD>  
<TITLE>DetectordeLengua</TITLE>  
</HEAD>  
<BODY>  
<%
```

```
'Antes de nada introducimos mensajes en forma de variables  
  espanol="Hola"  
  ingles="Hello"  
  frances="Bonjour"
```

```
'Ahora leemos del navegador cuál es su lengua oficial  
  idioma=Left(Request.ServerVariables("HTTP_ACCEPT_LANGUAGE"),2)
```

```
'Formulamos las posibilidades que se pueden dar  
If idioma="es" Then  
  Response.Write espanol  
Elseif idioma="fr" Then  
  Response.Write frances  
Else  
  Response.Write ingles  
End If %>  
</BODY>  
</HTML>
```

Como puede verse, las variables que contienen texto son almacenadas entre comillas.

Para leer la lengua aceptada por el navegador lo que hacemos es definir una variable (idioma) que recoge las dos primeras letras empezando por la izquierda del idioma aceptado por el navegador

("HTTP_ACCEPT_LANGUAGE"). Este idioma aceptado puede ser requerido como una variable del objeto ServerVariables. Por el momento dejaremos esto tal cual.

La tercera parte de script se encarga de ver si el navegador está en español (es), francés (fr) o en cualquier otro idioma que no sea ninguno de estos dos y de imprimir cada uno de los mensajes que proceda en cada caso.

Otro punto a comentar es el hecho de poder comentar los programas. Como puede observarse, dentro del script hemos introducido unos mensajes que nos sirven para leerlo más fácilmente. Estos mensajes no ejercen ninguna influencia en el desarrollo del mismo. Para introducirlos es necesario escribirlos detrás de un apóstrofe: '

Los comentarios son de gran utilidad cuando tratamos con programas muy extensos y complicados los cuales. En estos casos, resultan de gran ayuda a la hora de depurar fallos o introducir modificaciones. Es altamente aconsejable el acostumbrarse a utilizarlos

Bucles y condiciones II

Los bucles FOR

En muchas ocasiones resulta necesario ejecutar un conjunto de instrucciones un número definido de veces. Esto puede ser llevado a cabo a partir de la instrucción FOR / NEXT. La estructura clásica:

```
FOR contador=número inicial to número final STEP incremento
  Instrucción 1
  Instrucción 2
NEXT
```

A partir de este tipo de estructuras ejecutamos las instrucciones contenidas entre el FOR y el NEXT un cierto número de veces definido por el número inicial, final y el incremento. El incremento resulta de 1 por defecto.

Pongamos un ejemplo:

```
<% @ LANGUAGE="VBSCRIPT" %>
<HTML>
<HEAD>
<TITLE>Bucle for/next</TITLE>
</HEAD>
<BODY>
```

```
<%For i=1 to 5%>
```

```

<font size=<%Response.Write i%>>Vuelta número <%Response.Write
i%></font><br>
<%Next

For i=5 to 1 Step -1%>
<font size=<%Response.Write i%>>Contamos atrás: <%Response.Write
i%></font><br>
<%Next%>

</BODY>
</HTML>

```

Este script compuesto de dos bucles cuenta primero de 1 a 5. La variable *i* toma por lo tanto todos los valores enteros comprendidos entre estos dos números y puede ser utilizada dentro del bucle como lo hacemos en este caso para aumentar el tamaño de la letra. El segundo bucle realiza el proceso inverso (el incremento es negativo) produciendo una disminución del tamaño de la letra

Bucles y condiciones III

Los bucles DO WHILE/LOOP

Otra forma de realizar este tipo de secuencias bucle es a partir de la instrucción DO WHILE. En este caso lo que especificamos para fijar la extensión del bucle no es el número de vueltas sino el que se cumpla o no una condición.

```

DO WHILE condición
  Instrucción 1
  Instrucción 2
LOOP

```

El bucle se dará mientras la condición propuesta siga siendo válida.

Este tipo de bucles resulta muy práctico para la lectura de bases de datos. Todo este tipo de controladores de flujo (condiciones y bucles) pueden ser matizados y optimizados a partir del uso de operadores lógicos. Así, podemos exigir que sean dos las condiciones que se den para llevar a cabo un conjunto de instrucciones:

```

IF condición 1 AND condición 2 THEN ...

```


También podemos requerir que sea una de las dos:

IF condición 1 OR condición 2 THEN...

Del mismo modo, es posible exigir que la condición de un bucle DO sea la inversa a la enunciada:

DO WHILE NOT condición

Los objetos ASP

El ASP es un lenguaje diseñado para la creación de aplicaciones en internet. Esto quiere decir que existen toda una serie de tareas bastante corrientes a las cuales debe dar un tratamiento fácil y eficaz. Me refiero por ejemplo al envío de e-mails, acceso a archivos, gestión de variables del cliente o servidor como pueden ser su IP.

El lenguaje VB propiamente dicho no da una solución fácil y directa a estas tareas sino que invoca a los denominados objetos que no son más que unos módulos incorporados al lenguaje que permiten el desarrollo de tareas específicas. Estos objetos realizan de una manera sencilla toda una serie de acciones de una complejidad relevante. A partir de una llamada al objeto este realizará la tarea requerida. En cierta forma, estos objetos nos ahorran el tener que hacer largos programas para operaciones sencillas y habituales.

Algunos de estos objetos están incorporados en el propio ASP, otros deben de ser incorporados como si se tratase de componentes accesorios. Por supuesto, no podríamos ejecutar correctamente un script en el cual tuviésemos que llamar a un objeto que no estuviese integrado en el servidor. Este tipo de "plug-in" son generalmente comprados por el servidor a empresas que los desarrollan.

Como todo objeto del mundo real, los objetos del mundo informático tienen sus propiedades que los definen, realizan un cierto número de funciones o métodos y son capaces de responder de una forma definible ante ciertos eventos.

Objeto Request I

Bucles y condiciones son muy útiles para procesar los datos dentro de un mismo script. Sin embargo, en un sitio internet, las páginas vistas y los scripts utilizados son muchos. Muy a menudo necesitamos que nuestros distintos scripts estén conectados unos con otros y que se sirvan de variables comunes. Por otro lado, el usuario interacciona por medio de formularios cuyos campos han de ser procesados para poder dar una respuesta. Todo este tipo de factores dinámicos han de ser eficazmente regulados por un lenguaje como el

ASP.

Todo este tipo de aspectos interactivos pueden ser gestionados a partir del objeto Request.

El objeto Request nos devuelve informaciones del usuario que han sido enviadas por medio de formularios, por URL o a partir de cookies. También nos informa sobre el estado de ciertas variables del sistema como pueden ser la lengua utilizada por el navegador, el número IP del cliente...

Transferir variables por URL

Para pasar las variables de una página a otra lo podemos hacer introduciendo dicha variable en la dirección URL de la página destino dentro del enlace hipertexto. La sintaxis sería la siguiente:

```
<a href="destino.asp?variable1=valor1&variable2=valor2&..."></a>
```

Para recoger la variable en la página destino lo hacemos por medio del objeto Request con el método Querystring:

```
Request.querystring("variable1")  
Request.querystring("variable2")
```

Las dos páginas serían así

```
<HTML>  
<HEAD>  
<TITLE>Página origen.asp</TITLE>  
</HEAD>  
<BODY>  
<a href="destino.asp?saludo=hola&texto=Esto es una variable texto">Paso  
variables saludo y texto a la página destino.asp</a>  
</BODY>  
</HTML>
```

```
<HTML>  
<HEAD>  
<TITLE>destino.asp</TITLE>  
</HEAD>  
<BODY>  
Variable saludo: <%Response.Write Request.Querystring("saludo")%><br>  
Variable texto: <%Response.Write Request.Querystring("texto")%><br>  
</BODY>  
</HTML>
```

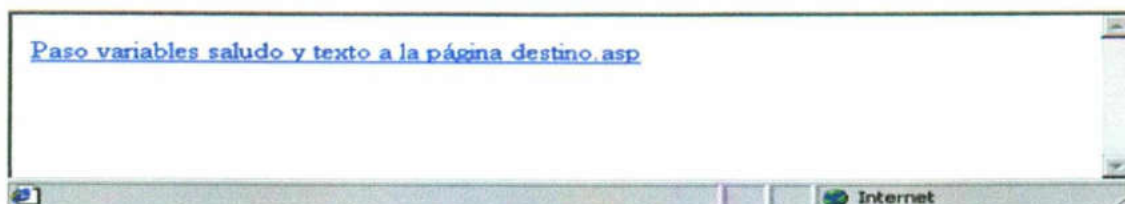



Fig: 2.2 Referencia al código que muestra el método paso de variables

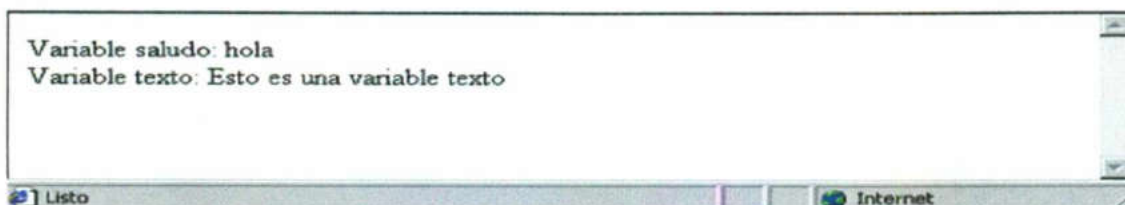


Fig.: 2.3 Referencia al código que muestra el método Response.Write

Objeto Request II

Transferir variables por formulario

El proceso es similar al explicado para las URLs. Primeramente, presentamos una primera página con el formulario a rellenar y las variables son recogidas en una segunda página que las procesa

```

<HTML>
<HEAD>
<TITLE>formulario.asp</TITLE>
</HEAD>
<BODY>
<FORM METHOD="POST" ACTION="destino2.asp">
Nombre<br>
<INPUT TYPE="TEXT" NAME="nombre"><br>
Apellidos<br>
<INPUT TYPE="TEXT" NAME="apellidos"><br>
<INPUT TYPE="SUBMIT">
</FORM>
</BODY>
</HTML>

<HTML>
<HEAD>
<TITLE>destino2.asp</TITLE>
</HEAD>
<BODY>
Variable nombre: <%=Request.Form("nombre")%><br>
Variable apellidos: <%=Request.Form("apellidos")%>

```

```
</BODY>
</HTML>
```

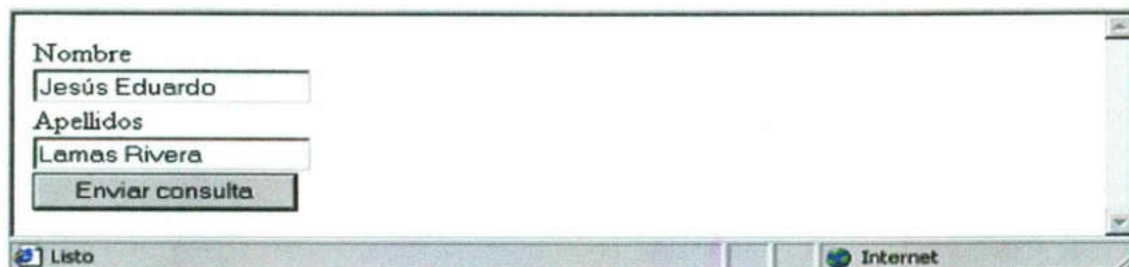


Fig. 2.4 Referencia al código muestra el método de transferir variables por formulario

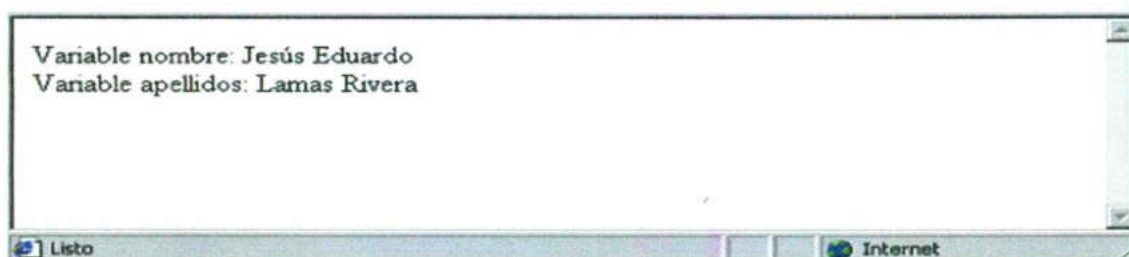


Fig. 2.5 Referencia al código para mostrar resultado método Request.form

Otras utilidades de Request: las ServerVariables

El objeto Request nos da acceso a otras informaciones relativas al cliente y el servidor las cuales pueden resultar de una gran utilidad. Estas informaciones son almacenadas como variables las cuales son agrupadas en una colección llamada ServerVariables.

Dentro de esta colección tenemos variables tan interesantes como:

HTTP_ACCEPT_LANGUAGE	Nos informa de la lengua preferida por el navegador
HTTP_USER_AGENT	Indica cuál es el navegador utilizado.
PATH_TRANSLATED	Nos devuelve el path físico del disco duro del servidor en el que se encuentra nuestro script
SERVER_SOFTWARE	Nos dice qué tipo de software utiliza el servidor

Tabla 2.1 Diferentes tipos de variables y su significado

Para visualizar en pantalla alguna de estas variables, debemos escribir

```
Response.write request.servervariables("nombre de la variable")
```


Una forma rápida de visualizar todas estas variables es a partir de un script con esta secuencia:

```
<%  
For Each elemento in Request.ServerVariables  
  Response.Write elemento & " : "&Request.ServerVariables(elemento)& "<br>"  
Next  
>%
```

Esto nos daría por un lado el nombre de la variable y del otro su valor. Este tipo de bucle For Each/Next se parece a otros ya vistos. En este caso, el bucle se realiza tantas veces como elementos tiene la colección (ServerVariables) que no es más que el conjunto de elementos comprendidos en la extensión del objeto (Request). Este tipo de bucle es aplicable a otras colecciones de éste y otros objetos como por ejemplo los Request.Form o Request.QueryString o las cookies. De esta forma seríamos capaces de visualizar el nombre y contenido de tales colecciones sin necesidad de enunciarlas una por una

Objeto Response

Tal y como se ha visto, el objeto Request gestiona todo lo relativo a entrada de datos al script por parte del usuario (formularios), provenientes de otra URL, del propio servidor o del browser.

Queda pues por explicar cómo el script puede "responder" a estos estímulos, es decir, después de procesar los datos, podemos imprimir estos en pantalla, inscribirlos en las cookies o enviar a una u otra pagina. En definitiva, queda por definir la forma en la que ASP regula el contenido que es enviado al navegador.

Esta función es tomada en cargo por el objeto Response
En efecto, sentencias del tipo:

```
<%Response.Write "Una cadena de texto"%>
```

o bien,

```
<%Response.Write variable%>
```

Tienen como cometido imprimir en el documento HTML generado un mensaje o valor de variable. Este método es tan comúnmente utilizado que existe una abreviación del mismo de manera a facilitar su escritura:

```
<% = variable %> es análogo a <%response.write variable%>
```

Es importante precisar el hecho que imprimir en el documento HTML no significa necesariamente visualizar en pantalla ya que podríamos servirnos de

estas etiquetas para crear determinadas etiquetas HTML. He aquí un ejemplo de lo que se pretende decir:

```
<% path="http://www.misitio.com/graficos/imagen.gif" %>
```

```

```

Este fragmento de script nos generaría un código HTML que sería recibido en el navegador del siguiente modo:

```

```

Otro elemento interesante de este objeto Response es el método Redirect. Este método nos permite el enviar automáticamente a una página que nosotros decidamos.

Esta función, puede ser empleada en scripts en los que enviamos al visitante de nuestro sitio a una u otra página en función del navegador que utiliza o la lengua que prefiere. También es muy útil para realizar páginas "escondidas" que realizan una determinada función sin mostrar ni texto ni imágenes y nos envían a continuación a otra página que es en realidad la que nosotros recibimos en el navegador.

Referencia: La función redirect se puede estudiar en un artículo más extensamente: Redirigir al navegador a una URL con ASP al detalle.

Aquí se puede ver una secuencia de script que nos permitiría usar este método para el caso en el que tengamos páginas diferentes para distintas versiones de navegadores.

```
<%  
tipo_navegador = Request.ServerVariables("HTTP_USER_AGENT")  
If Instr(1, tipo_navegador, "MSIE 3", 1) <> 0 then  
Response.Redirect "MSIE3.asp"  
Elseif Instr(1, tipo_navegador, "MSIE 4", 1) <> 0 then  
Response.Redirect "MSIE4.asp"  
Elseif Instr(1, tipo_navegador, "MSIE 5", 1) <> 0 then  
Response.Redirect "MSIE5.asp"  
Elseif Instr(1, tipo_navegador, "Mozilla/4", 1) <> 0 then  
Response.Redirect "Netscape4.asp"  
Else  
Response.Redirect "cualquiera.asp"  
%>
```

Por supuesto, acompañando este script, debemos tener los correspondientes archivos MSIE3.asp, MSIE4.asp, MSIE5.asp... Cada uno de ellos con las

Jesús Eduardo Lamas Rjvera

particularidades necesarias para la buena visualización de nuestra página en tales navegadores

Las cookies

Las cookies son información almacenada por un sitio web en el disco duro del usuario. Esta información es almacenada en un archivo tipo texto que se guarda cuando el navegador accede al sitio web.

Es posible, por supuesto, ver estos archivos. Para abrirlos hay que ir al directorio C:\Windows\Cookies para los usuarios de Internet Explorer 4 ó a C:\...\Netscape\Users\defaultuser para usuarios de Netscape. Como se puede comprobar, en la mayoría de los casos la información que se puede obtener es indescifrable.

La utilidad principal de las cookies es la de poder identificar al navegador una vez éste visita el sitio por segunda vez y así, en función del perfil del cliente dado en su primera visita, el sitio puede adaptarse dinámicamente a sus preferencias (lengua utilizada, colores de pantalla, formularios rellenos total o parcialmente, redirección a determinadas páginas...).

Para crear un archivo cookies, modificar o generar una nueva cookie lo podemos hacer a partir del objeto Response con una sintaxis como la siguiente:

```
Response.Cookies (" nombre de la cookie ") = valor de la cookie
```

El valor de la cookie puede ser una variable, un número o una cadena delimitada por comillas.

Es importante saber que las cookies tienen una duración igual a la sesión, es decir, a menos que lo especifiquemos, el archivo texto generado se borrará desde el momento en que el usuario haya abandonado el sitio por un tiempo prolongado (generalmente unos 20 minutos) o que el navegador haya sido cerrado. Podemos arreglar este supuesto inconveniente mediante la propiedad Expires:

```
Response.Cookies("nombre de la cookie").expires = #01/01/2002#
```

Esto nos permite decidir cuál es la fecha de caducidad de la cookie. Hay que tener en cuenta que esto no es más que hipotético ya que el usuario es libre de borrar el archivo texto cuando le plazca.

Por otra parte, es interesante señalar que el hecho de que definir una cookie ya existente implica el borrado de la antigua. Del mismo modo, el crear una

primera cookie lleva a la generación automática del archivo texto.

Para leer las cookies, es más fácil usando el objeto Request de la siguiente forma:

```
variable = Request.Cookies("nombre de la cookie")
```

Si por ejemplo quisiéramos recuperar del archivo txt la cookie correspondiente a la lengua del cliente y almacenarlo en nuestro script para futuros usos, podríamos escribir:

```
lengua=Request.Cookies("lengua")
```

Las cookies son una herramienta fantástica para personalizar nuestra página pero hay que ser cuidadoso ya que, por una parte, no todos los navegadores las aceptan y por otra, se puede deliberadamente impedir al navegador la creación de cookies. Es por ello que resultan un complemento y no una fuente de variables infalible para nuestro sitio.

Objeto Session

En los programas que hemos visto hasta ahora, hemos utilizado variables que solo existían en el archivo que era ejecutado. Cuando cargábamos otra página distinta, los valores de estas variables se perdían a menos que nos tomemos la molestia de pasarlos por la URL ó inscribirlos en las cookies ó en un formulario para su posterior explotación. Estos métodos, aunque útiles, no son todo lo prácticos que podrían en determinados casos en los que la variable que queremos conservar ha de ser utilizada en varios scripts diferentes y distantes los unos de los otros.

Podríamos pensar que ese problema puede quedar resuelto con las cookies ya que se trata de variables que pueden ser invocadas en cualquier momento. El problema, ya lo hemos dicho, es que las cookies no son aceptadas ni por la totalidad de los usuarios ni por la totalidad de los navegadores lo cual implica que una aplicación que se sirviera de las cookies para pasar variables de un archivo a otro no sería 100% infalible.

Nos resulta pues necesario el poder declarar ciertas variables que puedan ser reutilizadas tantas veces como queramos dentro de una misma sesión. Imaginemos un sitio multilingüe en el que cada vez que queremos imprimir un mensaje en cualquier página necesitamos saber en qué idioma debe hacerse. Podríamos introducir un script identificador de la lengua del navegador en cada uno de los archivos o bien declarar una variable que fuese valida para toda la sesión y que tuviese como valor el idioma reconocido en un primer momento.

Estas variables que son válidas durante una sesión y que luego son "olvidadas" son definidas con el objeto Session de la siguiente forma:

Session("nombre de la variable") = valor de la variable

Una vez definida, la variable Session, será almacenada en memoria y podrá ser empleada en cualquier script del sitio web.

La duración de una sesión viene definida por defecto en 20 minutos. Esto quiere decir que si en 20 minutos no realizamos ninguna acción, el servidor dará por finalizada la sesión y todas las variables Session serán abandonadas. Esta duración puede ser modificada con la propiedad Timeout:

Session.Timeout = n° de minutos que queremos que dure

Una forma de borrar las variables Session sin necesidad de esperara que pase este plazo es a partir del método Abandon:

Session.Abandon

De este modo todas las variables Session serán borradas y la sesión se dará por finalizada. Este método puede resultar practico cuando estemos haciendo pruebas con el script y necesitemos reinicializar las variables.

Lo que se suele hacer es crear un archivo en el que se borran las cookies y se abandona la sesión. Este archivo será ejecutado cuando queramos hacer borrón y cuenta nueva

```
<% @ LANGUAGE="VBSCRIPT" %>
<HTML>
<HEAD>
<TITLE>Puesta a cero</TITLE>
</HEAD>
<BODY>
<%
For Each galleta in Response.Cookies
  Galleta=""
Next
Session.Abandon
%>
Borrón y cuenta nueva!!<br>
<a href="url de la página de inicio">Volver al principio</a>
</BODY>
</HTML>
```

Trabajar con bases de datos en ASP

Una de las principales ventajas que presenta el trabajar con páginas dinámicas es el poder almacenar los contenidos en bases de datos. De esta forma, podemos organizarlos, actualizarlos y buscarlos de una manera mucho más simple.

ASP nos ofrece una forma muy eficaz de interactuar con estas bases de datos gracias al uso del componente ADO (ActiveX Data Objects) el cual permite acceder a dichas bases de una forma sencilla.

Este ADO no es más que un conjunto de objetos que, utilizados en conjunto, nos permiten explotar de una forma muy versátil las bases de datos de nuestra aplicación. No entraremos por el momento en consideraciones teóricas al respecto.

Por otra parte, los scripts ASP deben establecer un dialogo con la base de datos. Este dialogo se lleva a cabo a partir de un idioma universal: el SQL (Structured Query Language) el cual es común a todas las bases de datos. Este lenguaje resulta, muy potente y fácil de aprender.

Selecciones en una tabla

Dentro de una base de datos, organizada por tablas, la selección de una tabla entera o de un cierto número de registros resulta una operación rutinaria.

A partir de esta selección se puede posteriormente efectuar toda una serie de cambios ó bien realizar una simple lectura.

El siguiente script nos permite realizar la lectura de la tabla clientes contenida en nuestra base de datos.

```
<HTML>
<HEAD>
<TITLE>Lectura de registros de una tabla</TITLE>
</HEAD>
<BODY>
<h1><div align="center">Lectura de la tabla</div></h1>
<br>
<br>
<%
```

```
'Antes de nada hay que instanciar el objeto Connection
Set Conn = Server.CreateObject("ADODB.Connection")
'Una vez instanciado Connection lo podemos abrir y le asignamos la base de
datos donde vamos a efectuar las operaciones
Conn.Open "Mibase"
```


'Ahora creamos la sentencia SQL que nos servira para hablar a la BD
sSQL="Select * From Clientes Order By nombre"

'Ejecutamos la orden
set RS = Conn.Execute(sSQL)

```
'Mostramos los registros%>
<table align="center">
<tr>
<th>Nombre</th>
<th>Teléfono</th>
</tr>
<%
Do While Not RS.EOF
%>
<tr>
<td><%=RS("nombre")%></td>
<td><%=RS("telefono")%></td>
</tr>
<%
RS.MoveNext
Loop
```

```
'Cerramos el sistema de conexion
Conn.Close
%>
</table>
<div align="center">
<a href="insertar.html">Añadir un nuevo registro</a><br>
<a href="actualizar1.asp">Actualizar un registro existente</a><br>
<a href="borrar1.asp">Borrar un registro</a><br>
</div>

</BODY>
</HTML>
```

Antes de nada, si lo que deseamos es interrogar una base de datos, lo primero que hay que hacer es obviamente establecer la conexión con ella. Esto se hace a partir del objeto Connection el cual es "invocado" o más técnicamente dicho instanciado por medio de la primera instrucción en la cual el objeto toma el nombre arbitrario de la variable Conn .

El paso siguiente es abrir el objeto y asignarle la base de datos con la que debe contactar. En este caso, la base la he llamado Mibase. Este debe de ser el mismo nombre con el que la hemos bautizado cuando hemos configurado los conectores ODBC

Una vez creada la conexión a nuestra base de datos, el paso siguiente es hacer nuestra petición. Esta petición puede ser formulada primeramente y almacenada en una variable (sSQL) para, a continuación, ser ejecutada por medio de la instrucción siguiente.

La petición que hemos realizado en este caso es la de seleccionar todos los campos que hay en la tabla clientes (* es un comodín) y ordenar los resultados por orden alfabético con respecto al campo nombre. Podemos ver más detalladamente este tipo de instrucciones SQL en nuestro manual de SQL.

El resultado de nuestra selección es almacenado en la variable RS en forma de tabla. Para ver la tabla lo que hay que hacer ahora es "pasearse" por esta tabla "virtual" RS la cual posee una especie de cursor que, a menos que se especifique otra cosa, apunta al primer registro de la selección. El objetivo ahora es hacer desplazarse al cursor a lo largo de la tabla para poder leerla en su totalidad. La forma de hacerlo es a partir de un bucle Do While lo único que hace es ejecutar las instrucciones comprendidas entre el Do y el Loop siempre que la condición propuesta (Not RS.Eof) sea verdadera. Esto se traduce como "Ejecutar este conjunto de instrucciones mientras que la tabla de resultados (RS) no llegue al final (Eof, End of File).

Las instrucciones incluidas en el bucle son, por un lado, la impresión en el documento de los valores de determinados campos (=RS("nombre del campo")) y por otro, saltar de un registro al otro mediante la instrucción RS.MoveNext.

Todo este conjunto de instrucciones ASP viene en combinación con un código HTML que permite su visualización en forma de tabla. Además, se han incluido unos enlaces que apuntan hacia otra serie de scripts que veremos más adelante y que formaran en conjunto una aplicación.

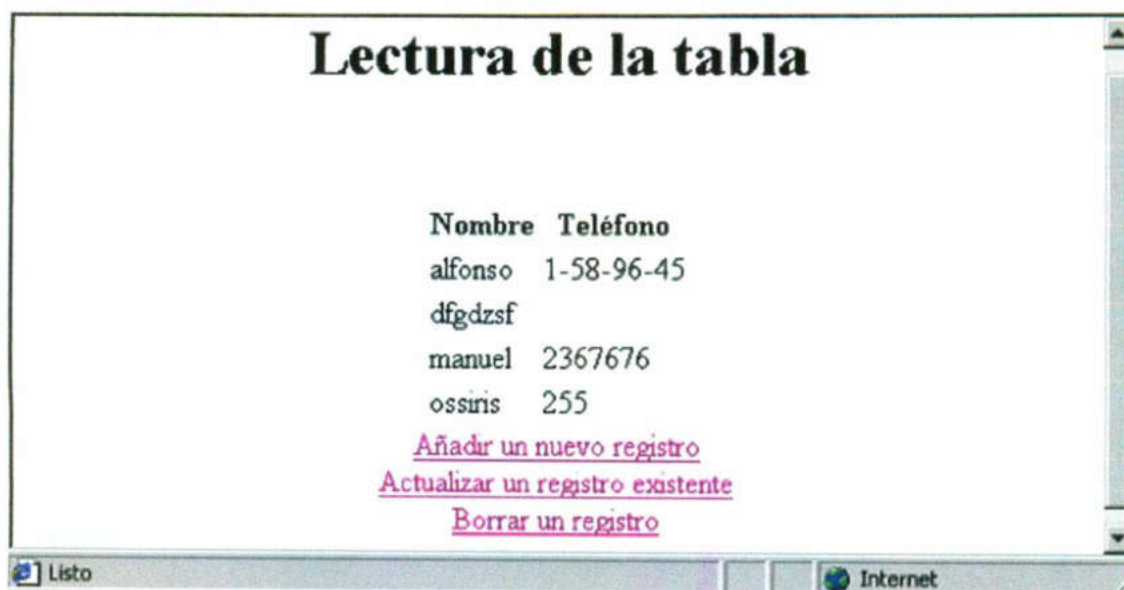


Fig. 2.6 Referencia al código de la página 25 y 26

Creación de un nuevo registro

En este caso lo que buscamos es crear, a partir de los datos recibidos de un formulario, un nuevo registro en nuestra tabla clientes. Tendremos pues dos archivos diferentes, uno que podría ser un HTML puro en el que introducimos el formulario a rellenar y que nos envía al segundo, un script muy parecido al previamente visto para realizar una selección. He aquí los dos scripts:

```
<HTML>
<HEAD>
<TITLE>Insertar.html</TITLE>
</HEAD>
<BODY>
<div align="center">
<h1>Insertar un registro</h1>
<br>
<FORM METHOD="POST" ACTION="insertar.asp">
Nombre<br>
<INPUT TYPE="TEXT" NAME="nombre"><br>
Teléfono<br>
<INPUT TYPE="TEXT" NAME="telefono"><br>
<INPUT TYPE="SUBMIT" value="Insertar">
</FORM>
</div>
</BODY>
</HTML>
```

```
<HTML>
```

```
<HEAD>
<TITLE>Insertar.asp</TITLE>
</HEAD>
<BODY>
<%
'Recogemos los valores del formulario
nombre=Request.Form("nombre")
telefono= Request.Form("telefono")

'Instanciamos y abrimos nuestro objeto conexion
Set Conn = Server.CreateObject("ADODB.Connection")
Conn.Open "Mibase"

'Ahora creamos la sentencia SQL
sSQL="Insert Into Clientes (nombre,telefono) values ('" & nombre & "'," &
telefono & "')"

'Ejecutamos la orden
set RS = Conn.Execute(sSQL)
%>

<h1><div align="center">Registro Insertado</div></h1>
<div align="center"><a href="lectura.asp">Visualizar el contenido de la
base</a></div>

<%
'Cerramos el sistema de conexion
Conn.Close
%>

</BODY>
</HTML>
```

Como puede verse, la forma de operar es idéntica a la vista anteriormente para el display de una tabla. En este caso hemos introducido un enlace a este primer script de lectura para ver cómo los cambios se han hecho efectivos.

La construcción de la sentencia SQL se hace por fusión de los distintos elementos constitutivos. La forma de fusionarlos mediante el símbolo `&`. Todo lo que sea texto tiene que ir entre comillas. Sería interesante introducir una línea suplementaria en vuestro código para imprimir la sSQL formada. La línea sería del siguiente tipo:

```
Response.Write sSQL
```

Esta línea iría situada evidentemente después de haber construido la sentencia

Fig. 2.7 Referencia al código de la página 28 y 29.

Actualización de un registro existente

Para mostrar cómo se actualiza un registro presente en nuestra base de datos, vamos a hacerlo a partir de un caso un poco más complejo para que empecemos a familiarizarnos con estas operaciones. Realizaremos un par de scripts que permitan cambiar el número de teléfono de las distintas personas presentes en nuestra base. El nombre de estas personas, así como el nuevo número de teléfono, serán recogidos por medio de un formulario.

El archivo del formulario va a ser esta vez un script ASP en el que efectuaremos una llamada a nuestra base de datos para construir un menú desplegable donde aparezcan todos los nombres. La cosa quedaría así:

```
<HTML>
<HEAD>
<TITLE>Actualizar1.asp</TITLE>
</HEAD>
<BODY>
<div align="center">
<h1>Actualizar un registro</h1>
<br>

<%
'Instanciamos y abrimos nuestro objeto conexion
Set Conn = Server.CreateObject("ADODB.Connection")
Conn.Open "Mibase"
%>

<FORM METHOD="POST" ACTION="actualizar2.asp">
Nombre<br>
<%
'Creamos la sentencia SQL y la ejecutamos
sSQL="Select nombre From clientes Order By nombre"
set RS = Conn.Execute(sSQL)
```

```

%>
<select name="nombre">
<%
'Generamos el menu desplegable
Do While not RS.eof%>
  <option><%=RS("nombre")%>
  <%RS.movenext
Loop
%>
</select>
<br>
Teléfono<br>
<INPUT TYPE="TEXT" NAME="telefono"><br>
<INPUT TYPE="SUBMIT" value="Actualizar">
</FORM>
</div>

</BODY>
</HTML>

```

La manera de operar para construir el menú desplegable es la misma que para visualizar la tabla. De nuevo empleamos un bucle Do While que nos permite mostrar cada una de las opciones.

El script de actualización será muy parecido al de inserción:

```

<TITLE>Actualizar2.asp</TITLE>
</HEAD>
<BODY>

<%
'Recogemos los valores del formulario
nombre=Request.Form("nombre")
telefono= Request.Form("telefono")

'Instanciamos y abrimos nuestro objeto conexion
Set Conn = Server.CreateObject("ADODB.Connection")
Conn.Open "Mibase"

'Ahora creamos la sentencia SQL
sSQL="Update Clientes Set telefono=" & telefono & " Where nombre=" &
nombre & ""

'Ejecutamos la orden
set RS = Conn.Execute(sSQL)
%>

```



```

<h1><div align="center">Registro Actualizado</div></h1>
<div align="center"><a href="lectura.asp">Visualizar el contenido de la
base</a></div>

<%
'Cerramos el sistema de conexion
Conn.Close
%>

</BODY>
</HTML>

```

Nada que comentar al respecto salvo la estructura de la sentencia SQL que en este caso realiza un Update en lugar de un Insert. Como para el caso precedente imprimir el valor de sSQL de manera a ver cómo queda la sentencia una vez construida



Fig. 2.8 Referencia al código de la página 30,31 y 32.

Borrado de un registro

Otra de las operaciones elementales que se pueden realizar sobre una base de datos es el borrar un registro. Para hacerlo, SQL nos propone sentencias del tipo Delete. Veámoslo con un ejemplo aplicado a nuestra agenda. Primero, crearemos un menú desplegable dinámico como para el caso de las actualizaciones:

```

<HTML>
<HEAD>
<TITLE>Borrar1.asp</TITLE>
</HEAD>
<BODY>
<div align="center">
<h1>Borrar un registro</h1>
<br>
<%

```

```

'Instanciamos y abrimos nuestro objeto conexion
Set Conn = Server.CreateObject("ADODB.Connection")
Conn.Open "Mibase"
%>

<FORM METHOD="POST" ACTION="borrar2.asp">
Nombre<br>
<%
'Creamos la sentencia SQL y la ejecutamos
sSQL="Select nombre From clientes Order By nombre"
set RS = conn.execute(sSQL)
%>
<select name="nombre">
<%
'Generamos el menu desplegable
Do While not RS.eof%>
  <option><%=RS("nombre")%>
  <%RS.movenext
Loop
%>
</select>
<br>
<INPUT TYPE="SUBMIT" value="Borrar">
</FORM>
</div>

</BODY>
</HTML>

```

El siguiente paso es hacer efectiva la operación a partir de la ejecución de la sentencia SQL que construimos a partir de los datos del formulario

```

<HTML>
<HEAD>
<TITLE>Borrar2.asp</TITLE>
</HEAD>
<BODY>
<%
'Recogemos los valores del formulario
nombre=Request.Form("nombre")

'Instanciamos y abrimos nuestro objeto conexion
Set Conn = Server.CreateObject("ADODB.Connection")
Conn.Open "Mibase"

'Ahora creamos la sentencia SQL
sSQL="Delete From Clientes Where nombre=" & nombre & ""

```



```
'Ejecutamos la orden
set RS = Conn.Execute(sSQL)
%>

<h1><div align="center">Registro Borrado</div></h1>
<div align="center"><a href="lectura.asp">Visualizar el contenido de la
base</a></div>

<%
'Cerramos el sistema de conexion
Conn.Close
%>

</BODY>
</HTML>
```

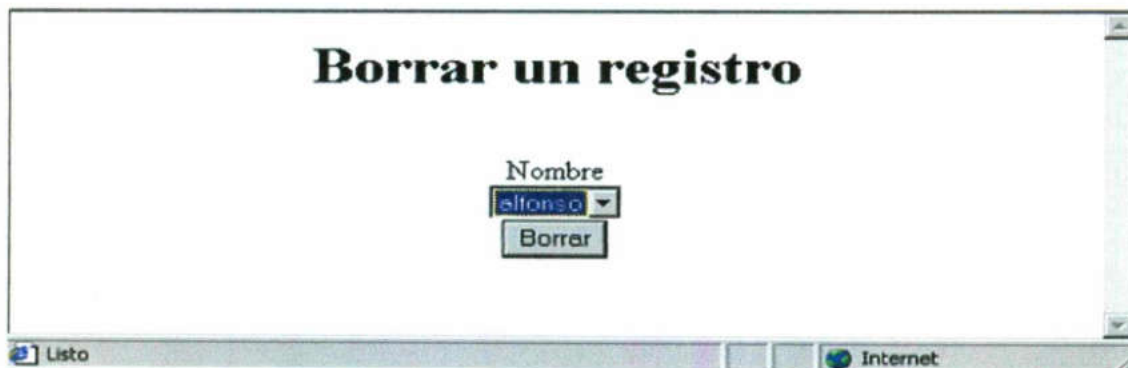


Fig. 2.9 Referencia al código de las páginas 32,33,34.para borrar un registro

Ejecución de Consultas con el Objeto CONNECTION

Con el método **Execute** del objeto **Connection** puede enviar consultas en el Lenguaje de consulta estructurado (SQL) al origen de datos y recuperar los resultados. SQL, un lenguaje estándar para comunicarse con bases de datos, tiene comandos para recuperar y actualizar información.

La siguiente secuencia de comandos utiliza el método **Execute** del objeto **Connection** para enviar una consulta con un comando **INSERT** de SQL, que inserta datos en una tabla concreta de la base de datos. En este caso, el bloque de la secuencia de comandos inserta el nombre José Lugo en una tabla de la base de datos llamada Customers.

```
<%
'Define el DSN de Archivo
strDSN = "FILEDSN=MibaseDeDatos.dsn"
```

'Crea la instancia del objeto Connection y abre una conexión con la base de datos

```
Set cn = Server.CreateObject("ADODB.Connection")
cn.Open strDSN
```

'Define la instrucción SELECT de SQL

```
strSQL = "INSERT INTO Customers (FirstName, LastName) VALUES ('José','Lugo')"
```

'Utiliza el método Execute para enviar una consulta SQL a la base de datos

```
cn.Execute(strSQL)
```

```
%>
```

Nota La ruta del DSN de Archivo no debe incluir espacios en blanco antes ni después del signo igual (=).

Además del comando **INSERT** de SQL, puede utilizar los comandos **UPDATE** y **DELETE** de SQL para modificar y quitar información de la base de datos.

Con el comando **UPDATE** de SQL puede modificar los valores de los elementos de una tabla de la base de datos. La siguiente secuencia de comandos usa el comando **UPDATE** para cambiar todos los campos **FirstName** de la tabla **Customers** a **Eduardo** en todas las filas cuyo campo **LastName** contenga el apellido **Robles**.

```
<%
```

```
Set cn = Server.CreateObject("ADODB.Connection")
cn.Open "FILEDSN=MiBaseDeDatos.dsn"
cn.Execute "UPDATE Customers SET FirstName = 'Eduardo' WHERE LastName = 'Robles' "
```

```
%>
```

Para

quitar determinados registros de una tabla de la base de datos, < BR >

utilice el comando **DELETE** de SQL. La siguiente secuencia de comandos quita todas las filas de la tabla **Customers** cuyo apellido sea **Robles**:

```
<%
```

```
Set cn = Server.CreateObject("ADODB.Connection")
cn.Open "FILEDSN=MiBaseDeDatos.dsn"
cn.Execute "DELETE FROM Customers WHERE LastName = 'Robles'"
```

```
%>
```


Nota Debe tener mucho cuidado al utilizar el comando **DELETE** de SQL. Un comando **DELETE** que no vaya acompañado de una cláusula **WHERE** eliminará todas las filas de la tabla. Asegúrese de incluir una cláusula **WHERE** de SQL, que especifica las filas exactas que se van a eliminar.

Uso del Objeto **RECORDSET** para manipulación de resultados

Aunque el objeto **Connection** simplifica la tarea de conectar con una base de datos y emitir una consulta, el objeto **Connection** tiene sus limitaciones. En concreto, con el objeto **Connection** no se pueden crear secuencias de comandos que recuperen y presenten información de una base de datos; tiene que saber exactamente las modificaciones que desea realizar en la base de datos y después implementar las modificaciones como consultas.

Para recuperar datos, examinar resultados y modificar su base de datos, ADO proporciona el objeto **Recordset**. El objeto **Recordset** tiene las funciones necesarias para, dependiendo de las restricciones de las consultas, recuperar y presentar un conjunto de filas, o registros, de una base de datos. El objeto **Recordset** mantiene la posición de cada registro devuelto por la consulta, lo que permite "recorrer" los resultados de uno en uno.

Dependiendo de cómo configure las propiedades del tipo de cursor del objeto **Recordset**, puede recorrer y actualizar los registros. Los cursores de la base de datos se comportan como punteros que le permiten encontrar un elemento específico de un conjunto de registros. Los cursores son especialmente útil para recuperar y examinar registros, y después realizar operaciones basadas en dichos registros. El objeto **Recordset** tiene propiedades que le permiten controlar con precisión el comportamiento de los cursores, lo que aumenta la capacidad de examinar y actualizar los resultados.

Por ejemplo, puede utilizar las propiedades **CursorType** y **CursorLocation** para configurar un tipo de cursor que devuelva un conjunto de resultados a su aplicación cliente (los resultados permanecen normalmente en el servidor de base de datos) y presente las últimas modificaciones realizadas en la base de datos por otros usuarios.

Recuperar un conjunto de registros

Las buenas aplicaciones de base de datos emplean el objeto **Connection** para establecer un vínculo y el objeto **Recordset** para manipular los datos devueltos. Si utiliza conjuntamente las funciones especializadas de ambos objetos puede desarrollar aplicaciones de bases de datos que realicen casi cualquier tarea de tratamiento de datos. la secuencia de comandos del servidor utiliza el objeto **Recordset** para ejecutar un comando **SELECT** de SQL. El

comando **SELECT** recupera un conjunto específico de información basándose en las restricciones de la consulta.

La consulta también contiene una cláusula **WHERE** de SQL, que se utiliza para establecer el criterio de selección de la consulta. En este ejemplo, la cláusula **WHERE** limita la consulta a todos los registros que contengan el apellido Soto en la tabla Customers de la base de datos.

```
<%  
'Establece una conexión con un origen de datos  
strDSN = "FILEDSN=MiBaseDeDatos.dsn"  
Set cn = Server.CreateObject("ADODB.Connection")  
cn.Open strDSN  
  
'Crea una instancia de un objeto Recordset  
Set rsCustomers = Server.CreateObject("ADODB.Recordset")  
  
'Abre un objeto Recordset con el método Open  
'y utiliza la conexión establecida por el objeto Connection  
    strSQL = "SELECT FirstName, LastName FROM Customers WHERE  
        LastName = 'Soto' "  
rsCustomers.Open strSQL, cn  
  
'Recorre el conjunto de los registros y presenta los resultados  
'e incrementa la posición del registro con el método MoveNext  
Set objFirstName = rsCustomers("FirstName")  
Set objLastName = rsCustomers("LastName")  
Do Until rsCustomers.EOF  
    Response.Write objFirstName & " " & objLastName & "<BR>"  
    rsCustomers.MoveNext  
Loop  
%>
```

Observe que en el ejemplo anterior, el objeto **Connection** estableció la conexión con la base de datos y el objeto **Recordset** utilizó la misma conexión para recuperar resultados de la base de datos. Este método es útil cuando tenga que configurar con precisión la forma en que se establece el vínculo con la base de datos. Por ejemplo, si necesitara especificar el tiempo de espera antes de anular un intento de conexión, tendría que utilizar el objeto **Connection** para establecer dicha propiedad. Sin embargo, si sólo desea establecer una conexión con las propiedades de conexión predeterminadas de ADO, podría utilizar el método **Open** del objeto **Recordset** para establecer el vínculo:

```
<%  
strDSN = "FILEDSN=MiBaseDeDatos.dsn"
```



```

strSQL = "SELECT FirstName, LastName FROM Customers WHERE
LastName = 'Soto' "
Set rsCustomers = Server.CreateObject("ADODB.Recordset")

'Abre una conexión con el método Open
'y utiliza la conexión establecida por el objeto Connection
rsCustomers.Open strSQL, strDSN

'Recorre el conjunto de registros, presenta los resultados
'e incrementa la posición del registro con el método MoveNext
Set objFirstName = rsCustomers("FirstName")
Set objLastName = rsCustomers("LastName")
Do Until rsCustomers.EOF
    Response.Write objFirstName & " " & objLastName & "<BR>"
    rsCustomers.MoveNext
Loop
%>

```

Cuando establece una conexión con el método **Open** del objeto **Recordset**, está utilizando implícitamente el objeto **Connection** para proteger el vínculo

Manejo de consultas con el Objeto **COMMAND**

Con el objeto **Command** de ADO puede ejecutar consultas de la misma forma que con los objetos **Connection** y **Recordset**, excepto en que con el objeto **Command** puede preparar, o compilar, la consulta con el origen de la base de datos y después enviar la misma consulta repetidamente con distintos valores. La ventaja de compilar consultas de esta manera consiste en la enorme reducción del tiempo necesario para volver a enviar las modificaciones de una consulta existente. Además, puede dejar las consultas SQL parcialmente indefinidas, con la posibilidad de alterar partes de las consultas justo antes de su ejecución.

La colección **Parameters** del objeto **Command** le ahorra los problemas de tener que volver a construir la consulta cada vez que tenga que ejecutarla. Por ejemplo, si tiene que actualizar periódicamente la información de suministros y costos en un sistema de inventario Web, puede predefinir su consulta de la siguiente manera:

```

<%
'Abre una conexión mediante el objeto Command del objeto
'Connection no tiene un método Open para establecer
'la conexión
strDSN = "FILEDSN=MibaseDeDatos.dsn"
Set cn = Server.CreateObject("ADODB.Connection")
cn.Open strDSN

```

```
'Crea el objeto Command; utiliza la propiedad
'ActiveConnection para
'adjuntar la conexión al objeto Command
Set cm= Server.CreateObject("ADODB.Command")
Set cm.ActiveConnection = cn
'Define la consulta

SQL cm.CommandText = "INSERT INTO Inventory (Material, Quantity)
VALUES (?, ?)" 'Guarda una versión preparada
(o precompilada) de la consulta <
BR >
'especificada en la propiedad
'CommandText antes de la primera ejecución del objeto Command.
cm.Prepared = True

'Define la información de configuración
'de los parámetros de la consulta
cm.Parameters.Append cm.CreateParameter("material_type",200,,255)
cm.Parameters.Append cm.CreateParameter("quantity",200, ,255 )

'Define y ejecuta la primera inserción
cm("material_type") = "Bombillas"
cm("quantity") = "40"
cm.Execute

'Define y ejecuta la segunda inserción
cm("material_type") = "Fusibles"
cm("quantity") = "600"
cm.Execute
%>
```

Observará que la secuencia de comandos vuelve a construir repetidamente y envía una consulta SQL con distintos valores, sin tener que volver a definir y enviar la consulta al origen de datos. Compilar consultas con el objeto **Command** también le ofrece la ventaja de evitar los problemas que pudieran provenir de concatenar cadenas y variables para formar consultas SQL.

En particular, si utiliza la colección **Parameter** del objeto **Command** puede evitar problemas relacionados con la definición de ciertos tipos de variables de cadenas, fechas y horas. Por ejemplo, los valores de las consultas SQL que contengan apóstrofes (') pueden hacer que la consulta falle:

```
strSQL = "INSERT INTO Customers (FirstName, LastName) " & _
" VALUES ('Roberto','O'Hara)'"
```

Observe que el apellido **O'Hara** contiene un apóstrofo que entra en conflicto con los apóstrofes que se utilizan como delimitadores de datos en la palabra

clave **VALUES** de SQL. Si enlaza el valor de la consulta como un parámetro del objeto **Command**, se evitan este tipo de problemas.

Manejo de conexiones a Bases de Datos

Uno de los mayores retos del diseño de una aplicación Web sofisticada de base de datos, como una aplicación de entrada de pedidos en línea que atienda a miles de clientes, es la correcta administración de las conexiones con la base de datos. Abrir y mantener las conexiones con las bases de datos, incluso cuando no se transmite información, puede afectar severamente a los recursos del servidor de base de datos y provoca problemas de conectividad.

Las aplicaciones Web de bases de datos bien diseñadas reciclan las conexiones con la base de datos y compensan los retrasos debidos al tráfico de la red.

IIS

PWS es una aplicación de servidor de páginas web que se utiliza en Windows 98.

IIS es una aplicación más profesional que se utiliza con Windows XP Professional o con Windows 2000.

IIS es un Servidor de Páginas Web, podemos configurar nuestro servidor para Servir Páginas Web a una red local (Intranet), a una gran red (Internet) o sólo a nuestro servidor para probar nuestras páginas web y estudiar ASP. En realidad IIS es un paquete con el que podemos tener un Servidor de Páginas Web, un Servidor de FTP y un Servidor virtual de correo SMTP

Algunos detalles sobre el IP

Resulta que en una red que trabaje con el protocolo TCP/IP, a cada servidor se le asigna un número de la forma 104.214.23.12 Así que si queremos comunicarnos con otro servidor de nuestra red local o Internet, debemos conocer su IP.

Además de la IP pública cada servidor también tiene otra IP : 127.0.0.1 ésta es para poder comunicarse con sí mismo, se utiliza para hacer prácticas, comprobaciones, estudios,

Así que si cualquier servidor se comunica con el IP 127.0.0.1 se está comunicando consigo mismo.

IP estático e IP dinámico

Si tienes una línea telefónica normal RTB, tu proveedor de Internet te asignará un IP distinto (dinámico) cada vez que te conectes a Internet.

Si tienes una línea con ADSL, tu proveedor de Internet te asignará una IP fija. (Aparte, siempre para tí mismo también serás : 127.0.0.1)

IP dinámico = Tus amigos te localizarán con dificultad, porque cada vez que te conectes te cambian el IP. Si vas a configurar un Servidor de Web, FTP o Correo debes indicarle a tus amigos tu IP actual. Es más difícil que te "entren" en el servidor con troyanos porque pocos conocen tu IP cambiante.

IP estático = Tus amigos saben que tu eres, por ejemplo 104.214.23.12 y será fácil que te encuentren para una video conferencia, chat, escritorio remoto, servidor web, ftp, correo etc.... Lo malo es que al estar identificado algunos "malos amigos" intentarán meterse en tu servidor con troyanos y otros virus, pues conocen tu dirección.

Construir un DNS local

Con el protocolo TCP/IP cada servidor debe tener su número IP, éste número es difícil de recordar, de manera que podemos asignarle a ese número IP un nombre. Ve al archivo `\\Windows\\system32\\drivers\\etc\\hosts` y edítalo con el Bloc de notas, añade la siguiente línea :

```
127.0.0.1 eduardo
```

Ahora podrás localizar a tu servidor con el nombre de eduardo, en vez de poner 127.0.0.1, si tienes una red de servidores, en éste archivo puedes poner el IP de cada servidor y un nombre que lo identifique, de manera que para localizar a ese servidor en la red, puedes poner su nombre en vez de su IP. Hay Servidores de Nombres DNS que traducen un nombre a un número. Con la orden ping en el Intérprete de comandos, puedes deducir si tu servidor está en red y si tiene contactos con otros servidores.

```
ping 127.0.0.1 -----> Para ver si estoy en red
ping 192.168.110.4 -----> Para ver si tomo contacto con ese otro servidor
ping eduardo -----> Si en el archivo hosts he puesto 127.0.0.1
eduardo
ping antonio -----> antonio tendrá un IP que puede estar en el archivo
hosts
ping www.google.com -----> Por ejemplo para ver si tengo contacto con Google.
```


Capítulo III

IIS Servidor Web

IIS lo encontramos en el CD de Windows XP Profesional. Lo instalamos mediante Panel de Control/Agregar o quitar programas/Agregar o quitar componentes de Windows/Servicios de Internet Information Server (IIS) [En Detalles, marcar también FTP]

Nos creará una carpeta en C:\inetpub, en ella localizamos la carpeta C:\inetpub\wwwroot, en esta carpeta podemos poner una página web, por ejemplo : miprueba.htm (Otra carpeta que se crea es C:\WINDOWS\system32\inetsrv)

Tu servidor tendrá un nombre; ve al Intérprete de comandos (Inicio/Ejecutar/cmd.exe) y escribe : SET userdomain

Ese es el nombre de tu servidor por ejemplo USERDOMAIN = MISERVIDOR

También puedes escribir en el Intérprete de comandos: hostname (En el archivo hosts puedes poner tu IP y un nombre) Ahora ve al Internet Explorer y escribe : http://miservidor/miprueba.htm te saldrá la página web. Cualquier servidor de la red Intranet puede poner esa dirección y verá tu página. Tú también puedes poner: http://127.0.0.1/miprueba.htm porque ese número IP te identifica a tí mismo.

También puedes ponerte : http://localhost/miprueba.htm Tu servidor es tu localhost.

Si estás conectado a Internet y conoces tu IP, se la puedes comunicar a un amigo para que vea la página web que tienes en tu servidor. Si tu IP actual es 134.212.34.22 Tu amigo debe escribir : http://134.212.34.22/miprueba.htm Ya ves TU SERVIDOR es un Servidor de Páginas Web. Resulta que el IP cambia cada vez que te conectas a tu Proveedor de Internet, así que cada vez que te conectes, tu amigo debe saber cuál es tu nueva IP.

Si tienes ADSL la IP es siempre (o casi siempre, te la pueden cambiar) la misma.

Por cierto, puedes ver la ayuda del IIS escribiendo en el Internet Explorer :
<http://localhost/iisHelp/iis/misc/default.asp>

Directorio Virtual

La carpeta wwwroot es la carpeta por defecto pero no es preciso tener tu página web en wwwroot, la puedes tener en cualquier directorio de tu disco duro (o CDROM)

Recuerda el concepto de Directorio virtual, consiste en poner un Alias a un directorio, por ejemplo si tu página está en
C:\PERSONAL\ESTUDIO\PRACTICAS\WEBS\inicio.htm.
C:\PERSONAL\ESTUDIO\PRACTICAS\WEBS puede ser el Directorio Virtual de Alias : eduardo, de manera que tu amigo de Internet para entrar en tu página debe poner :
<http://134.212.34.22/eduardo/inicio.htm>

O sea que eduardo = C: \PERSONAL\ESTUDIO\PRACTICAS\WEBS

Vamos a ver como se establece un Directorio Virtual. Créate una subcarpeta de una subcarpeta ... y dentro de ella un inicio.htm. Siguiendo el ejemplo anterior :
C:\PERSONAL\ESTUDIO\PRACTICAS\WEBS\inicio.htm.

Vamos a "Administración de equipos" = Inicio/Ejecutar/compmgmt.msc

(O bien Botón derecho sobre Mi PC/Administrar)

(O bien Inicio/Ejecutar/C:\Windows\system32\inetsrv\iis.msc)

(O Panel de Control/Herramientas Administrativas/Administración de equipos)
Servicios y Aplicaciones/Servicios de Internet Information Server/Sitios
Web/Sitio Web predeterminado



Fig. 3.1 Directorio Virtual

(Desde aquí controlamos la configuración del IIS)

Marca : Sitio Web predeterminado

Botón derecho/Nuevo/Directorio virtual ...

Alias : eduardo

Elige :

C:\PERSONAL\ESTUDIO\PRACTICAS\WEBS\

Nota : En Internet intenta utilizar siempre letras minúsculas en el Alias.

Ya está definido el Directorio Virtual eduardo

Pruébalo :

<http://miservidor/eduardo/inicio.htm>

<http://127.0.0.1/eduardo/inicio.htm>

<http://localhost/eduardo/inicio.htm>

Marcar Sitio Web predeterminado/Botón derecho – Propiedades Ficha = Sitio Web

Nos indica : Que las páginas Web entran por el puerto 80. Que si a los 900 segundos no entra la página, se corta la conexión. Formato de registro activo/Propiedades. Que en C:\WINDOWS\System32\LogFiles, se guarda la actividad del sitio. Visitas. IP de los visitantes. Errores. Hora. Consultar esa carpeta.

Ahora marca tu Directorio Virtual eduardo/Botón derecho – Propiedades

Pestaña:Directorio Virtual

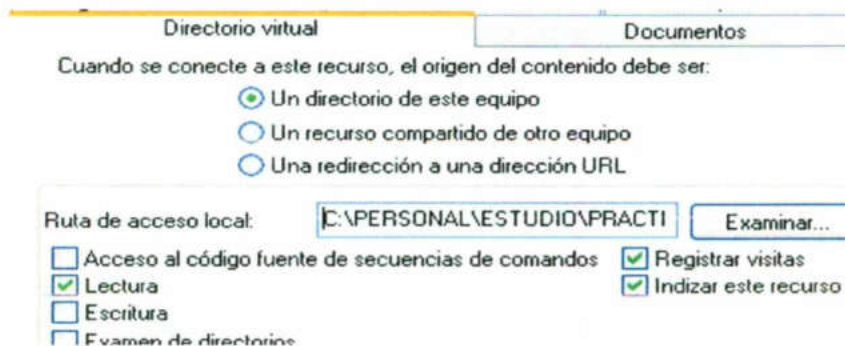


Fig. 3.2 Características del directorio virtual

Nos indica cuál es la Ruta de nuestro directorio real. Podemos hacer que en vez de ir a nuestro directorio vaya a una dirección URL (Una redirección a una dirección URL)(se utiliza por ejemplo cuando estamos actualizando el sitio).

El cuadro para marcar *Habilitar documento* predeterminando permite al PWS visualizar automáticamente un *documento* (cuyo nombre se le proporciona) siempre que el usuario examine un directorio sin especificar un nombre específico de *documento*. Por omisión, se le da el valor Default.htm y Default.asp (para Páginas Activas de Servidor). Por tanto, si ha proporcionado un archivo denominado Default.htm en un subdirectorio / Sub, un usuario que solicite únicamente el directorio /Sub sin especificar un archivo, obtendrá lo que estuviera en Default.htm. Los *documentos* por omisión están activados por omisión. Los *documentos* por omisión son una forma muy potente de controlar la forma en que se suministran contenidos en nuestro sitio Web. El directorio *raíz* virtual, que se describe como <raiz> en la lista Directorios virtuales, contiene un archivo Default.htm que puede ser editado libremente por el usuario. Esto modificará la Información que se proporciona automáticamente a un usuario cuando solicitan el sitio Web sin especificar un directorio o el nombre de un *documento*. Si se deja sin seleccionar el cuadro *Habilitar documento predeterminado*, entonces el PWS no suministrará el documento por omisión de ningún directorio. Este cambio es global y no se puede realizar para directorios individuales. Generalmente, esta debería hacerse tan solo en circunstancias excepcionales.

Permitir exploración de directorios

El cuadro *Permitir exploración de directorios*, cuando está activado, permite a los usuarios examinar el contenido de un directorio del sitio Web con tan solo proporcionar el nombre de la ruta. Si no hay documento por omisión en ese directorio, o si se han desactivado los documentos por omisión, entonces se enumerará todo el contenido del directorio, con un vínculo para cada nombre de archivo. Esto es útil si se desea publicar directorios completos de archivos rápidamente. La opción *Permitir exploración de directorios* está desactivada por omisión.

Permitir la exploración de directorios puede ser peligroso. Porque cualquier entendido puede entrar en ese directorio.

Guardar el registro de la actividad del sitio Web

Al activar el cuadro *Guardar registro de la actividad del sitio Web* se da lugar a que toda la actividad del sitio Web - peticiones del exterior, archivos enviados, direcciones IP de los usuarios del sitio Web, se registre en un archivo. Esta es una forma alternativa muy potente de evaluar la utilización, más allá de las gráficas de utilización incorporadas. Los registros se guardan en el directorio \System\LogFile\W3spc1. Comienza un nuevo registro después de cada mes de actividad. Los registros se escriben con el formato de registro NCSA y se pueden visualizar con cualquier editor de texto. La creación de registros está desactivada por omisión.

Obtención del máximo rendimiento del Personal Web Server

Lo que sigue son varias sugerencias para una utilización inteligente, para optimizar el rendimiento y para obtener lo más posible del PWS en general.

Defragmentar frecuentemente el sistema servidor

Toda computadora que se emplee como servidor Web experimentará una gran

Pestaña : Documentos

Habilitar documentos predeterminados : Podemos configurar un nombre de página de inicio por defecto de tal manera que en la Dirección del Internet Explorer no sea necesaria poner la inicio.htm

Es decir en el Internet Explorer ponemos :

http://miservidor/eduardo (No hay que poner inicio.htm)

En eduardo buscará primero una página llamada Default.htm, si no la encuentra busca Default.asp, si no la encuentra busca index.htm, luego busca iisstart.asp. Nosotros podemos Agregar nuestra página : inicio.htm

La página de entrada o inicio de un sitio web sencillo suele llamarse index.htm

Habilitar pie de página del documento. Esto lo que hace es añadir al final de cada página web del Sitio eduardo, otra pequeña página web de información o propaganda.

Ejemplo : (Ésta página aparecería automáticamente debajo de cada página web de ese sitio

```
<HTML>
<HEAD><TITLE>Abajo</TITLE></HEAD>
<BODY>
<HR>
<FONT SIZE=5 COLOR=#FF0000>Pagina realizada por Eduardo
Antonio</FONT><BR>
<B>http://www.trucosxp.tk</B><BR>
<A HREF="http://www.trucosxp.tk" target="_parent">Iniciate</A>
<HR>
</BODY>
</HTML>
```

Seguridad

Hasta ahora cualquiera (anónimo) puede entrar en nuestras páginas. (Acceso anónimo). Ahora vamos que hacer que tengan que identificarse mediante una cuenta y una contraseña. Para ello, primero tenemos que crear en nuestro servidor a un usuario y su correspondiente contraseña.

En el Directorio virtual que queremos proteger vamos (Botón derecho/Propiedades) a la ficha "Seguridad de directorios" y luego pulsamos el botón "Modificar..." y marcamos solamente:

Autenticación básica.

Métodos de autenticación

Acceso anónimo
No se necesita nombre de usuario ni contraseña para este recurso.
Cuenta usada para acceso anónimo
Nombre de usuario:
Contraseña:
 Permitir que IIS c

Acceso autenticado
Para los siguientes métodos de autenticación de usuario y contraseña cuando:
- el acceso anónimo está deshabilitado
- el acceso está restringido por permisos de acceso NTFS

Autenticación básica (la contraseña)
 Autenticación de texto implícita por el sistema de archivos
Dominio predeterminado:
Territorio:
 Autenticación de Windows integrada

Fig. 3.3 Como establecer autenticación

A partir de ahora cada vez que alguien quiera entrar en esa página, le saldrá una ventanita emergente en donde se le solicitará que introduzca su nombre de cuenta (x) y su contraseña (y)

Con IIS de Windows XP, solo podrás realizar un Sitio Web (aunque dentro de este sitio puedes poner muchos directorios virtuales de manera que podremos tener, por ejemplo:

<http://127.217.19.23/electronica>
<http://127.217.19.23/juegos>

Por supuesto si compras un Dominio y un DNS podrás poner:
<http://www.eduardo.com/electronica>
<http://www.eduardo.com/juegos>

Pero todos crean el mismo archivo log. En Windows 2000 Server y creo que también en Windows .NET puedes tener varios Sitios de manera que puedes hacer un: <http://www.electronica.eduardo.com>, <http://www.juegos.eduardo.com>

Todos con el mismo dominio y el mismo DNS. Cada sitio tendrá un Log distinto

Con las demás fichas puedes controlar las páginas de errores, la caducidad, Puedes incluso configurar Propiedades para cada archivo de la página. Además del Servidor de Páginas Web IIS de Microsoft

IIS Servidor FTP

Consiste en configurar un Directorio en tu disco duro para que cualquiera que sepa tu IP, pueda entrar en ese Directorio de tu servidor y copiar los archivos que tú le des permiso. Esa otra persona debe tener un cliente FTP, que es un programa para trabajar con FTP como Cute_FTP, WS_FTP, etc... En la página FTP de este sitio puedes encontrar información sobre clientes FTP. (El Internet Explorer también funciona como cliente FTP, pero de pocas posibilidades)

Vamos a "Administración de equipos" = Inicio/Ejecutar/compmgmt.msc

(ó bien Botón derecho sobre Mi PC/Administrar)

(ó bien Inicio/Ejecutar/C:\Windows\system32\inetstrv\iis.msc)

(ó Panel de Control/Herramientas Administrativas/Administración de equipos)

Servicios y Aplicaciones/Servicios de Internet Information Server/Sitios FTP

Lo demás es parecido a Sitio Web. Puedes poner una carpeta o un archivo de cualquier tipo (.exe, .mp3, .jpg, .mpg, etc.) en la carpeta C:\inetpub\ftproot y cualquiera puede tomar contacto con ella mediante un cliente de FTP y conociendo tu IP. Crea en C:\inetpub\ftproot una nueva carpeta llamada ANTONIO

¿Cómo se puede entrar en las carpetas?

Necesitamos un Cliente FTP. El Internet Explorer nos puede valer como Cliente FTP. Entramos en el Internet Explorer y en la Dirección ponemos :

```
ftp://mi pc/  
ftp://localhost/  
ftp://127.0.0.1/
```

Entrarás en C:\inetpub\ftproot y verás la carpeta ANTONIO Si estás en Internet y conoces tu IP (por ejemplo 134.212.34.22), tu amigo puede entrar en el Internet Explorer y poner ftp://134.212.34.22/ verá tu carpeta ANTONIO

Si quiere que tu carpeta no esté en C:\inetpub\ftproot, puedes crear un Directorio Virtual. Supongamos que tu carpeta para FTP sea D : \MUSICA \ ESTUDIO Vas a: Servicio y Aplicacion/Servicio de Internet Information Server / Sitio Web/Sitio FTP predeterminado Botón derecho /Nuevo/Directorio virtual

Alias : mimusica

Elige : D:\MUSICA\ESTUDIO

Ya está definido el Directorio Virtual mimusica = D:\MUSICA\ESTUDIO

Pruébalo, con el Internet Explorer :

ftp://mi pc/mimusica
ftp://127.0.0.1/mimusica
ftp://localhost/mimusica

Una vez situado en la carpeta mimusica podrás leer, copiar, borrar, según los permisos que tengas. Para una mejor configuración sitúate alternativamente sobre :

- Sitios FTP
- Sitios FTP predeterminados
- mimusica

Pulsa con el Botón derecho y luego Propiedades. Desde aquí podrás configurar permisos, registrar visitas en C:\WINDOWS\System32\LogFiles, puerto FTP (21), mensajes, etc. Puedes poner que los demás puedan leer, escribir o que tengan ambas posibilidades. Al igual que hacíamos con el Servidor Web, podemos ponerle Seguridad a nuestro Servidor FTP, de manera que para entrar en una carpeta los usuarios deban identificarse con un nombre de cuenta y una contraseña. También debes ir a la carpeta que vas a compartir y estudiar los permisos.

Clientes FTP

Así que para que alguien de Internet o Intranet pueda ver nuestras carpetas, necesitará un Cliente FTP. El Internet Explorer, como he comentado, su puede utilizar como Cliente FTP, escribiendo en la Dirección:
ftp://123.111.23.43/carpeta.

En cambio un Cliente FTP especializado, nos dará mas posibilidades. Hay muchos clientes FTP uno de ellos es el CuteFTP.

Pero nosotros vamos a utilizar otro programa gratuito (freeware) muy conocido llamado: WS_FTP_LE, lo podemos bajar de <http://www.ftpplanet.com> , MEJOR lo puedes bajar directamente de aquí WS_FTP LE (670 K) Deberás poner

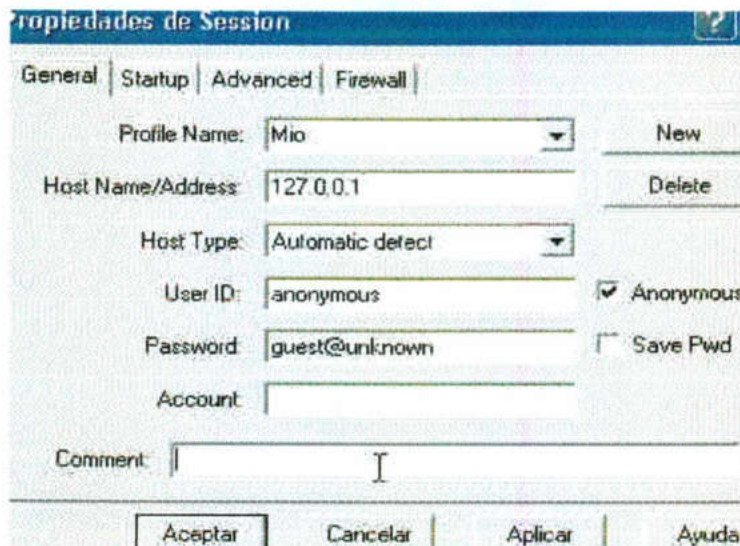


Fig. 3.4 ejemplo de una pantalla para conectarse en FTP

- a) Si te quieres comunicar con el directorio ftproot HostName 127.0.0.1
 - b) si te quieres comunicar con el directorio mimusica HostName 127.0.0.1
- En Startup=mimusica (O bien pulsar el botón ChDir)

Atención al Password :guest@unknown

Si te pide que te identifique como usuario escribe: anonymous

En caso que el Sitio FTP tenga cuentas, deberás poner tu cuenta y tu clave.

Además del Servidor de FTP del IIS de Microsoft, podemos encontrar otros muchos Servidores FTP, por ejemplo GuildFTP, Nitro Downloader, RaidenFTP

IIS Servidor de correo SMTP

Para ver la ayuda ir a C:\Windows\Help\mail.chm Se trata de configurar un Servidor de correo y enviar correos mediante el Outlook. Hay dos tipos de servidores de correos : De salida de mensajes (SMTP, salen por el puerto 25)
De entrada de mensajes (POP3, entran por el puerto 110)

Con este Servidor SMTP vamos a enviar mensajes mediante el Outlook y se van a guardar en la carpeta : C:\inetpub\mailroot\Drop
(Con este servidor no podemos recibir correos pues no tiene el protocolo POP3, más adelante veremos otro que sí lo tiene)

Vamos a "Administración de equipos" = Inicio/Ejecutar/compmgmt.msc

(O bien Botón derecho sobre Mi PC/Administrar)

(O bien Inicio/Ejecutar/C:\Windows\system32\inetrv\iis.msc)

(O Panel de Control/Herramientas Administrativas/Administración de equipos)

Jesús Eduardo Lamas RIVERA

Servicios y Aplicaciones/Servicios de Internet Information Server/Servidor virtual SMTP

Botón derecho/Propiedades - Ficha General - Dirección IP : 127.0.0.1

Dominios/Panel derecho Debe aparecer el nombre del dominio en este caso el nombre del servidor, por ejemplo : miservidor Ahora vas al Outlook y configura una nueva cuenta :

Herramientas/Cuentas.../Agregar/Correo

Nombre para mostrar : Jorge López

Dirección de correo electrónico : jorge@miservidor (Debes poner el nombre de tu dominio)

Servidor de correo entrante POP3 127.0.0.1 (No está configurado, no recibe)

Servidor de correo saliente SMTP 127.0.0.1 (Tú mismo)

Nombre de cuenta : jorge

Ahora desde el Outlook puedes enviar un mensaje a jorge@miservidor, el mensaje se guardará en la carpeta : C:\inetpub\mailroot\Drop Con este Servidor podemos gestionar correo saliente con el protocolo SMTP (puerto 25) y entrante con el protocolo POP3 (puerto 110)

Antes de empezar, y para que no tengamos conflictos vamos a Parar el Servidor SMTP del IIS que hemos configurado en el apartado anterior, para ello vamos a : Inicio/Ejecutar/compmgmt.msc servicios y Aplicaciones / Servicios de Internet Information Server/Servidor virtual SMTP - Botón derecho - Detener

Ahora instala : ArGoSoft Mail Server 1.12 es de <http://www.argosoft.com> una aplicación gratuita que nos instala un Servidor de Correo entrante y saliente, fácil de usar y fácil de desinstalar.

Una vez instalado vamos a configurarlo para hacer pruebas, se trata de enviar y recibir correos de nuestro servidor a nuestro servidor (127.0.0.1)

Herramientas de opciones

En Log File Name Poner la dirección en donde está instalado el programa y un archivo.txt en donde se guardará informaciones de las conexiones. Load server at Windows startup y Automatically start the server, es para que el servidor arranque cuando inicias Windows, si quieres los desmarcas.

De nuevo en la ventana anterior pide el DNS Server, insisto que en Internet los servidores se entienden por números IP (127.217.34.12), de tal manera que en algún lugar debe haber algún servidor que cuando reciba un correo de la forma eduardo@yahoo.com sepa interpretar que yahoo.com es el IP 127.217.34.12 (Éste número no es de yahoo, sólo es un ejemplo). Esos servidores que convierten palabras a números se llaman Servidores DNS. Prueba éstos números como Servidores DNS 195.235.113.3 o 195.235.96.90

Tools/Users Añadir Nuevo Usuario Se creará la carpeta jorge en :
****\ArgoSoftMailServer\jorge

File/Start Para arrancar el servidor, File/Stop Para parar el servidor
File/Suspend Delivery, suspender las entregas. Recibe pero no envía. Los correos recibidos y no enviados se guardan en la carpeta
****\ArgoSoftMailServer_outbox. Ésta opción debe estar como : Delivery resumed (mirad abajo en la barra de estado)

Configuración del Outlook para nuestra cuenta de correo

Herramientas/Cuentas /Agregar/Correo/

Nombre para mostrar : Jorge López

Dirección de correo electrónico : jorge@127.0.0.1

Servidor de correo entrante (POP3) : 127.0.0.1

Servidor de correo saliente (SMTP) : 127.0.0.1

Nombre de cuenta : jorge

Contraseña : jorge

Una vez finalizado, marcarlo y "Establecer como predeterminada"

Ya está todo, ahora vamos a probarlo con el Outlook ---- Crear correo :

De: jorge@127.0.0.1 Para : jorge@127.0.0.1 Escribimos un mensaje. Lo enviamos. Luego consultamos mensajes recibidos.

Una vez que lo hemos "autoprobado", podemos conectar con Internet, buscar nuestra IP actual, y ponerla en ArgoSoftMailServer, Tools/Options, en vez de 127.0.0.1. Crear un Nuevo Usuario, ejemplo : paco Contactémos con un amigo, démosle nuestra IP y un nombre, (ejemplo : paco). Que configure con outlook una Cuenta de correos con nuestra IP. Y luego que se auto envíe un correo a través de nuestro servidor:

De : paco@213.34.124.23 Para : paco@213.34.124.23

Este es el archivo log.txt que muestra información

de una petición (REQUEST) conexión de mensaje (SMTP) y de salida (POP3), de jorge a jorge. - 10:24:52 - Requested SMTP connection from 127.0.0.1 - 10:24:52

- (127) 220 ArGoSoft Mail Server, Version 1.12 (1.1.2.1) - 10:24:52 - (127) HELO miservidor - 10:24:52 - (127) 250 Welcome [127.0.0.1], pleased to meet you - 10:24:52 - (127) MAIL FROM: - 10:24:52 - (127) 250 Sender "jorge@127.0.0.1" OK... - 10:24:52 - (127) RCPT TO: - 10:24:52 - (127) 250 Recipient "jorge@127.0.0.1" OK... - 10:24:52 - (127) DATA - 10:24:52

- (127) 354 Enter mail, end with "." on a line by itself - 10:24:52 - (127) 250 Message accepted for delivery. - 10:24:52 - (127) QUIT - 10:24:52 - (127) 221 Aba he - 10:24:52 - SMTP connection with 127.0.0.1 ended. ID=127 - 10:24:59

- Message from jorge@127.0.0.1 delivered to jorge@127.0.0.1 - 10:27:08 - Requested POP3 connection from 127.0.0.1 - 10:27:08 - (128) +OK ArGoSoft Mail Server, Version 1.12 (1.1.2.1) - 10:27:08 - (128) USER jorge - 10:27:08 - (128) +OK Password required for jorge - 10:27:08 - (128) PASS jorge - 10:27:08 - (128) +OK Mailbox locked and ready - 10:27:08 - (128) STAT - 10:27:08 - (128) +OK 1 1304 - 10:27:08

- (128) LIST - 10:27:08 - (128) +OK - 10:27:08 - (128) 1 1304 - 10:27:08 - (128) . - 10:27:08 - (128) RETR 1 - 10:27:08 - (128) +OK 1304 octets - 10:27:08

- (128) .

Personal Web Server

El Personal Web Server es un programa servidor Web modesto pero sumamente útil, que se ejecuta en Windows 98. Utiliza algunas de las tecnologías principales del servidor Web más profesional de Microsoft, denominado Internet Information Server (o bien IIS). El PWS tiene muchas de las capacidades de su hermano mayor, incluyendo la capacidad de conectarse a bases de datos y de utilizar Active Server Pages para crear contenido dinámico.

PWS es una versión reducida de IIS para tareas de servicio Web menos exigentes, como las de una intranet de nuestra propia oficina. Sin embargo, es posible emplear el PWS para servir páginas a Internet en general, si las demandas de las páginas y las capacidades del hardware son modestas. También se puede emplear el PWS para probar sitios Web antes de activarlos de verdad.

PWS fue creado y activado inicialmente por Windows 98 como aditamento descargable y se encontraba en el sitio Web de Microsoft, en sus juegos de herramientas para desarrollo y en muchos depósitos shareware de todo el mundo. Cuando Windows 98 empezó a tomar forma, se añadió como componente estándar que se podía instalar desde el CD-ROM de Windows 98.

El Personal Web Server permite, simplemente, transformar la computadora en un servidor Web que tiene potencia más que suficiente para satisfacer las necesidades de un servidor Web modesto. Si dispone de una conexión a Internet, o tiene la computadora instalada en una red de área local (LAN), podrá publicar documentos HTML sin tener que invertir en un servidor por separado. Cuando se utiliza en conjunción con una herramienta para publicaciones Web compatible con Microsoft, como Frontpage, el PWS se puede emplear también para publicar bases de datos.

El PWS se puede utilizar de modo completamente independiente del tipo de conexión a Internet o a una red que se utilice, siempre y cuando esa colección soporte el protocolo TCP/IP. Funcionará cualquier cosa, desde una LAN o una conexión T1 hasta una conexión telefónica de 28,8, aunque con diferente velocidad.

El PWS no está diseñado para hacer las veces de un servidor Web de categoría industrial, que sea capaz de manejar, por ejemplo, transacciones seguras u operaciones financieras. Este tipo de capacidades se maneja mejor con Internet Information Server de Microsoft, que funciona en Windows NT y proporciona una plataforma mucho más robusta para servir servicios de tipo comercial basados en la Web.

El Personal Web Server fue diseñado originalmente como un aditamento descargable para Windows 95 Service Release 2. Aparece, con algunas modificaciones actualizaciones, como característica de Windows 98, que se puede instalar directamente desde el CD-ROM de Windows 98.

Requisitos de hardware

Dado que el PWS es reducido y está diseñado para funcionar con poca cosa, la ejecución de PWS no requiere demasiada potencia de hardware. Cualquier computadora que admita Windows 98 podrá servir como punto de partida y debería ajustarse a esta descripción:

- 0 Procesador 486/50 0 superior, Intel o compatible con Intel; es preferible un Pentium o superior.
- 0 16MB de RAM.
- 0 200MB de espacio en disco duro.
- 0 Una conexión de red.

Esta es la parte mas indispensable del PWS, porque es imposible tener acceso al servidor sin un enlace de red de alguna clase. El PWS no considera para nada si se accede a las páginas a través de una conexión telefónica, una LAN o una WAN, con tal de que tenga a su disposición un enlace de red que utilice TCP/IP.

En general, cuanto mejor sea la configuración, mejor funcionará todo, incluyendo el PWS.

Instalación de PWS

Para instalar el Personal Web Server, inserte el disco de Windows 98 en el lector de CD-ROM y pulse Inicio, Ejecutar. Escriba la orden siguiente:

```
letradeunidad] : \add-ons\pws\instalar.exe
```

Sustituya *[letradeunidad]* por la letra de su lector de CD-ROM. Debería ver que aparece el programa de instalación del Personal Web Server (véase la Figura). Para iniciar la instalación, pulse Siguiente y lea el contrato de licencia.

Pulse Siguiente para pasar al menú de opciones de instalación

Opciones de instalación

Con el PWS hay tres tipos básicos de instalación posibles. Se escoge entre los tipos de instalaciones pulsando el botón correspondiente del menú de opciones de instalación:

Mínima. Pulse aquí para instalar solamente el mínimo indispensable de componentes necesario para ejecutar el Personal Web Server. Si está mal de espacio si sólo quiere hacerse a la idea de como es el PWS, esta es una buena opción. Esto contiene los binarios necesarios para ejecutar el programa, pero no contiene documentación ni componentes para transacciones.

Típica. La opción de instalación típica instala todo lo que instala la instalación Mínima, más una documentación básica y el mínimo de herramientas necesarias para crear aplicaciones Web (por ejemplo, bases de datos preparadas para la Web).

Personalizada. Esta opción permite adoptar y descartar los componentes deseados entre la multiplicidad de posibilidades que están disponibles para el PWS. Algunos de estos componentes son indispensables para el PWS y se marcan por omisión; la mayoría son opcionales.

Si selecciona Personalizada, se le presentará una lista de componentes que se pueden seleccionar o deseleccionar (véase la Figura). Véase acto seguido una rápida revisión de los componentes disponibles en el PWS mediante la opción de *menú* Personalizada: *Archivos de programa comunes*. Son los componentes

que constituyen el núcleo del PWS, así que esta opción no debería dejarse sin marcar. *Extensiones de servidor de FrontPage 98*. Esto instala componentes que permiten al servidor funcionar con el programa de creación de páginas Web Front Page 98 de Microsoft. Si tiene intención de emplear FrontPage 98 para construir páginas Web con el PWS, marque este componente; en caso contrario es perfectamente correcto dejarlo sin marcar.

Microsoft Data Access Components 1.5. Los Data Access Components (Componentes de acceso a datos) permiten al PWS interactuar con bases de datos. Si sólo tiene intención de servir páginas estáticas mediante el PWS, puede quitar la marca de esta opción.

Microsoft Message Queue. Este componente es una pieza crítica de la tecnología de aplicaciones Web del PWS. La Message Queue (Cola de mensajes) permite al PWS comunicarse con otras aplicaciones en red y debería cargarse si se tiene la intención de utilizar el PWS para publicar contenido dinámico, como bases de datos

Personal Web Server. Este es otro conjunto crítico de componentes fundamentales del PWS y tampoco debe dejarse sin marcar.

Transaction Server. Esto capacita al PWS para que ejecute aplicaciones basadas en la Web que estén formadas por objetos ActiveX. Una vez más, si no tiene intención de publicar un contenido variable o de crear una base de datos para la Web, puede perfectamente dejar esta opción sin marcar.

Si marca todos los cuadros de la lista, se instalará *todo* lo que se incluye en el paquete del PWS - herramientas de desarrollo, documentación, extensiones del servidor de FrontPage 98 y así sucesivamente -. Tenga en cuenta que una instalación completa del PWS ocupará 52,6 megabytes de espacio en disco, así que sólo debería hacerse en un sistema del que se sepa con seguridad que dispone de espacio suficiente y si se necesitan realmente todos los elementos. La lista de componentes no aparece si no se selecciona Personalizada.

Una vez que haya acabado de seleccionar los componentes que quiera cargar, pulse Siguiente. El programa de instalación de PWS le preguntará a continuación el nombre del directorio en que vaya a almacenar sus archivos de páginas Web (véase la Figura). Por omisión, se toma la ruta [unidadraiz]: \inetpub\wwwroot, pero esto se puede cambiar por lo que se quiera. Los otros dos directorios (FTP Service, Servicio FTP, y Application Installation Point, Punta de Instalación de Aplicaciones) no están disponibles en el PWS y, por tanto, están atenuados por omisión.

Pulse Siguiente para empezar a copiar los archivos de instalación de PWS desde el CD-ROM. Aparecerá una barra de progreso, que indica el estadio en que se encuentra el proceso de copia. Cuando haya finalizado la copia de archivos, se le pedirá que reinicie la máquina. El PWS arrancará automáticamente tras el próximo arranque.

Ejecución y configuración del PWS

Cuando se está ejecutando, el PWS aparece normalmente como un icono en la bandeja del sistema (véase la Figura). El icono cambia dependiendo del estado del servidor (en ejecución, en pausa o detenido). Al pulsar con el botón derecho el icono, aparece una lista de opciones rápidas para PWS.

. *Iniciar servicio.* Esto suele estar atenuado, porque el servicio de PWS esta configurado para que se ejecute por omisión, pero cuando esta desactivado o en pausa, se activa. Seleccione esto para poner en marcha el PWS cuando no esta activado todavía.

. *Detener servicio.* Esto apaga por completo el PWS. Cuando se selecciona Detener servicio, el icono cambia y aparece una «señal de alto» por encima del icono principal de PWS.

. *Hacer una pausa en el servicio.* Esto evita que el PWS se comuniquen, pero no lo descarga de memoria. Cuando esta seleccionado Hacer una pausa en el servicio, el icono cambia y tiene una señal de «ceda el paso» por encima.

. *Continuar servicio.* Esto solamente esta activado si el servicio está en pausa. Al seleccionarlo se dará lugar a que los servicios de red se reanuden como antes y el icono de PWS en la bandeja volverá a adoptar su aspecto original.

. *Propiedades.* Seleccionar esta opción es lo mismo que pulsar dos veces el icono en sí y hará que aparezca la hoja Propiedades de PWS.

La hoja Propiedades de PWS o Administrador personal Web

La hoja Propiedades de PWS, conocida también como Administrador personal Web, aparece cuando el usuario pulsa dos veces el icono de la bandeja, o selecciona Propiedades dentro del menú que aparece al pulsar con el botón derecho el icono. También se puede acceder a ella desde el botón Inicio, seleccionando Programas, Microsoft Personal Web Server, Administrador personal Web. Véase la Figura para observar un ejemplo de Administrador personal Web.

El Manager (Administrador) esta dividido en dos paneles y el panel de la izquierda contiene las cinco opciones principales del programa: Principal, Publicar, Sitio Web, Paseo y Avanzada. Pulsando cada una de ellas aparecerá la información correspondiente en el panel de la derecha. Por omisión, en el panel de la derecha se ve la ventana Main.

Publicación

La subsección Publicación del panel Main, describe los atributos básicos del funcionamiento del PWS: si está o no en marcha, la URL correcta para visualizar el sitio desde su máquina local y el directorio principal del servidor que está registrado en este momento. Las dos URL que se muestran en la subsección Publicación están activadas: se puede pulsar en ellas y aparecerá una copia de Internet Explorer para visualizar el contenido de esa URL. Se trata de una forma buena y rápida de obtener una vista previa de lo que hay en el sitio Web.

El botón Detener tiene la misma funcionalidad que el botón Detener servicio que aparece en el menú que se obtiene al pulsar con el botón derecho en el icono. Cuando se pulsa, pasa a convertirse en un botón Iniciar (una vez más, con idéntica funcionalidad) y los vínculos con el sitio PWS desaparecen. Las estadísticas de la sección de monitorización también desaparecen. Para reactivar el sitio Web, límitese a pulsar el botón Iniciar.

Asistente de publicación

El Asistente de publicación es una forma rápida de crear y publicar documentos sencillos en el PWS. No es un sustituto para un programa completo de

publicación en HTML como pueda ser Microsoft FrontPage, pero es una buena forma de poner una página en el servidor rápidamente y sin programación.

Windows 98 viene con una copia de FrontPage Express, una versión reducida del editor de HTML FrontPage. No lleva a cabo una gestión global de sitios Web, pero es muy válido para editar páginas individuales.

El Asistente de publicación se activa pulsando el icono Publicar que se encuentra en el panel izquierdo del Administrador Web personal. Cuando se hace este, se abre el Asistente de publicación (véase la Figura) en el panel de la derecha.

El Asistente de publicación contiene varios controles (representados en la Figura) que se emplean para publicar la página deseada.

Si nunca había creado anteriormente un documento con el Asistente de publicación, el programa le pedirá que ejecute primero el Asistente de página de inicio.

. *Path (Ruta)*. Rellene la ruta del documento en cuestión, o pulse Examinar, para revisar sus unidades de disco duro y seleccionar un archivo. Si ha abierto el Asistente arrastrando y soltando un archivo en el icono Publicar, este campo ya estará lleno y no será necesario completarlo

. *Description (Descripción)*. Proporciona una descripción del archivo en cuestión, que puede ser cualquier texto. Si tiene activa la exploración de directorios, esta descripción aparecerá situada junto al archivo publicado en el directorio examinado. Este campo es opcional.

. *Files to Publish (Archivos para publicar)*. Contiene una lista de los archivos que van a ser publicados.

. *Add (Agregar)*. Pulse el botón Add (Agregar) para añadir la página descrita en los cuadros de texto Path y Description a la lista «Files to Publish».

. *Delete (Quitar)*. Pulse Quitar para borrar un archivo seleccionado dentro de la lista «Files to Publish». (Pulse primero el nombre de un archivo de la lista para seleccionarlo.

Asistente de página de inicio

El Asistente de página de inicio crea una página predeterminada rápida y atractiva en el servidor, pidiendo al usuario que responda a algunas preguntas básicas.

1.- El Asistente de página de inicio tiene varias *temas* de entre los cuales se puede seleccionar un diseño para la página predeterminada. Se trata de los siguientes:

Looseleaf El diseño de página se parece a una página de un bloc de hojas intercambiables.

Journal. Este diseño pretende parecerse a un diario o a una hoja en blanco.

Gunmetal. Un aspecto gris, de alta tecnología.

Pulse el diseño que quiera utilizar y después pulse la flecha hacia la derecha (el símbolo ») para avanzar.

2. Se puede instalar un *libro de visitantes* en la página predeterminada. Se trata de una sección de la página predeterminada en que los visitantes pueden

firmar y dejar comentarios. Pulse Si o No para admitir o rechazar un libro de visitantes y después pulse la flecha hacia la derecha para continuar.

3. Se puede tener una *bandeja de entrada* en la página predeterminada. Se trata de Una sección para mensajes privados de los visitantes. Pulse Si o No dependiendo de si desea o no tener Una bandeja de entrada y después pulse la flecha hacia la derecha para continuar.

4. La página siguiente se abre en Internet Explorer, o en el navegador de red que pueda estar instalado en ese momento, y es a la vez una recapitulación de las opciones anteriores y la posibilidad de añadir información personal a su página predeterminada (nombre, dirección, correo electrónico, intereses, aficiones, enlaces de páginas externas y otras cosas similares).

Una vez que se ha ejecutado el Asistente de página de inicio, el Asistente de publicación permitirá editar y hacer cambios en las páginas que ya estarán disponibles.

El Paseo del Administrador personal Web

El producto Paseo del Administrador Web personal, que aparece en el panel de la derecha cuando se pulsa el icono Paseo del panel del lado izquierdo, es un resumen no interactivo de las características principales del PWS.

Opciones Avanzadas

El panel Opciones avanzadas (véase la Figura) permite controlar algunos de los ajustes de uso menos común, aunque sean importantes, del PWS. Una persona que vaya a emplear el PWS en un contexto bastante pasivo --esto es, que lo configure y lo deje como está - tiene pocas probabilidades de necesitar usar alguna de las opciones avanzadas. Sin embargo, alguien que tenga intención de sacarle todo el partido al servidor Web necesitará familiarizarse con la pantalla Opciones avanzadas.

Directorio Virtuales

Los directorios virtuales son una forma de estructurar con más libertad el contenido de un sitio Web. Generalmente, toda la estructura de directorios de un sitio Web es virtual --esto es, no existe una correspondencia directa entre la estructura de directorios del sitio Web y la organización de los archivos en el servidor-. Por ejemplo, si se tiene un sitio Web que tenga un directorio virtual denominado /main, el sistema de directorio virtual de su sitio Web le permitirá poner los archivos de /main en cualquier lugar del disco duro. Lo mismo sucede con cualquier otro directorio de toda la jerarquía.

La subsección Directorios virtuales del panel Opciones avanzadas enumera todos los directorios virtuales que haya en ese momento en el servidor Web.

Después de la ventana Directorios virtuales aparece una lista de botones que permiten controlar las lista Directorios virtuales: Agregar, Editar Propiedades y Quitar.

Agregar

El botón Agregar permite insertar un nuevo directorio virtual en la jerarquía contenida en el directorio que esté resaltado en ese momento. Par ejemplo, si

el directorio resaltado es /New, el botón Agregar crea un directorio dentro de /New. Para crear un directorio virtual que esté en el directorio raíz, marque el directorio <raíz>, púselo y después pulse Agregar.

Al pulsar Agregar aparece la ventana Agregar directorio. Considere todas las entradas de la ventana Agregar directorio antes de pulsar Aceptar para confirmar el nuevo directorio.

Directorio. Es el directorio físico de su unidad de disco duro que se va a distribuir. Pulse el botón Examinar para abrir un árbol de directorios del estilo del Explorador y seleccione el directorio que desee gráficamente. Alternativamente, si conoce la ruta de acceso, puede limitarse a escribirla en el cuadro de texto Directorio.

Alias. Es el nombre que se quiere dar al directorio virtual, como lo verán en el sitio Web. Los nombres de alias tienen que construirse siguiendo las convenciones de denominación que se emplean para crear todas las URL: solo pueden contener letras mayúsculas o minúsculas, guiones de subrayado o normales; y no se admiten espacios ni barras.

Editar propiedades

El botón Editar propiedades permite visualizar las propiedades de un directorio virtual, que son las mismas que aparecen en la ventana Agregar directorio, para modificarlas. Para hacer cambios en las propiedades de un directorio virtual, púselo para resaltarlo y después pulse el botón Editar propiedades.

Quitar

El botón Quitar borra el directorio virtual seleccionado. Como sucedía con Agregar y Editar, se borra un directorio pulsándolo primero, para resaltarlo, y pulsando después el botón Quitar.

Acceso

La subsección Acceso de la ventana Agregar directorio controla la forma en que se desea restringir o admitir el acceso a un directorio virtual dado.

. *Lectura.* Permite a los usuarios tomar documentos de ese directorio y leerlos.

. *Ejecución.* Permite a los usuarios del servidor Web ejecutar programas de ese directorio.

Archivo de ordenes. Permite que un motor de guiones --como un programa CGI- se ejecute en este directorio sin tener activado el permiso Execute. Utilice el permiso Script para aquellos directorios que contengan guiones ASP, guiones IDC (Conector de Bases de datos para Internet) u otros guiones. El permiso

Script es más seguro que el permiso Execute porque se pueden limitar las aplicaciones que se pueden ejecutar en el directorio.

En aras de la seguridad no active a las vez los permisos Lectura y Escritura.

Pulse Aceptar para finalizar la configuración y establecer el nuevo directorio virtual, o bien pulse Cancelar para salir del paso.

Activar documento por omisión

cantidad de actividad de disco. Utilice el Programador de tareas para configurar el sistema de modo que se defragmente al menos una vez a la semana. Además, si la máquina en que se ejecuta el PWS se utiliza primariamente como servidor Web y para poco más, utilice la opción Reordenar los archivos de programa para que mis aplicaciones inicien más rápido en el Defragmentador de disco para poner el PWS y los archivos asociados en la parte anterior del disco, para que el acceso a ellos sea más rápido

Utilice los registros de actividad para determinar la mejor política de utilización

Utilice los gráficos de utilización a primera vista y los registros de actividad para obtener una idea mejor del momento en que aparecen máximos de tráfico en el sitio Web. Si la máquina se emplea para algo más que los servicios Web, puede imponer una norma consistente en prohibir a los usuarios locales el acceso a la máquina durante las horas más conflictivas.

Utilice el PWS como «campo de pruebas» para un sitio Web

Una de las mejores aplicaciones del PWS es proporcionar una forma localizada de probar un sitio Web antes de implantarlo realmente. Si se utiliza el PWS junto con una herramienta para publicar en la Web como Microsoft Frontpage, se puede aprovechar buena parte de la potencia del PWS para probarlo todo, desde las páginas estáticas hasta el contenido dinámico, incluyendo las bases de datos y las páginas activas de servidor.

Utilización del Personal Web Server en la Red a tiempo completo

Una de las mejores características del Personal Web Server es su eficiencia en términos de coste. Permite que una computadora de sobremesa normal sirva páginas Web a Internet en cualquier canal principal. Esto hace posible que cualquiera utilice su computadora de sobremesa para crear un sitio Web de alcance mundial sin emplear nada más sofisticado que su cuenta telefónica de acceso a Internet.

Hasta hace poco, esto no resultaba práctico por una razón importante. La mayoría de los proveedores de servicio de Internet, o PSI, hacen uso de «IP dinámico» para sus clientes de conexión telefónica. Esto significa que cada vez que alguien llama por teléfono y se conecta a Internet, se le asigna un número IP de un determinado conjunto que está reservado específicamente para clientes de acceso telefónico. Desafortunadamente, esto significa que es raro que los usuarios obtengan dos veces el mismo IP cuando se conectan por teléfono. Esto hace imposible establecer un sitio Web a tiempo completo mediante una conexión telefónica, porque no se le puede dar un nombre de dominio a una dirección IP dinámica.

Han existido algunas soluciones intermedias para este problema. Por ejemplo, hay muchos programas *shareware* que, en el momento de la conexión, determinan el número IP y lo publican en una página Web, en un sitio FTP o incluso en un canal IRC. Pero esto sigue exigiendo que alguien sepa donde buscar la dirección IP y puede causar más problemas que beneficios. Ahora existe una solución más fiable para este problema. Hay compañías que, con la oportuna tarifa, permiten asignar a la dirección IP dinámica un subdominio de subdominio, como "micomputadora.dinamica.red". Esto no es tan impresionante como un nombre de dominio del más alto nivel, como "www.micomputadora.com", pero hace muchísimo más fácil crear y mantener una

presencia en la red y atraer visitantes regulares a nuestro sitio Web. Para hacer este posible, el usuario descarga e instala un programa que notifica al host de IP dinámico cada vez que la computadora obtiene una nueva dirección IP (esto es, cada vez que llama por teléfono). Los clientes en sí son muy pequeños y apenas se notan.

Hacer esto tiene sus desventajas. Una de ellas es lo reducido del ancho de banda, por cuanto el «tubo» más ancho que puede ofrecer actualmente una conexión telefónica son 56 Kb de entrada (y 33,6K de salida). La otra desventaja es que requiere dedicar continuamente una línea telefónica a la computadora y habría que utilizarla en una zona en la que no haya cargos por tiempos de llamada (sólo por número de llamadas) o en un lugar en el que no se cobren las llamadas locales. Con todo, si no se tiene intención de servir nada más exigente que páginas de texto y algún gráfico ocasional, ésta es una opción que merece la pena investigar. Por ejemplo, las organizaciones sin ánimo de lucro y centros de dispersión de información harían bien en investigar esta opción, porque proporciona buenos resultados con un coste mínimo.

En el momento de escribir estas líneas, hay dos proveedores principales de alojamiento de servidores con IP dinámico.

Capítulo V

Páginas web ASP por medio de un manejador

Ahora que ya sabemos las bases para hacer la programación de páginas web ASP, y conocemos que cosas es necesario instalar para que funcione como servidor IIS ó PWS ya podemos enfocarnos a un manejador de páginas web como lo es dreamweaver de los más completos en el cuál podemos hacer lo siguiente:

Crear un juego de registros de base de datos

A continuación, creará un juego de registros para mostrar datos almacenados en una base de datos. Un **juego de registros** es un subconjunto de datos extraídos de una base de datos mediante una consulta de base de datos. (En ASP.NET, un juego de registros se conoce como Conjunto de datos.) Una consulta de base de datos está formada por criterios de búsqueda que definen lo que debe incluirse en el juego de registros. Luego utilizará la información extraída como origen de datos para sus páginas dinámicas.

ofrece una interfaz fácil de usar para la creación de consultas SQL sencillas (no es necesario tener conocimientos de SQL para crear un juego de registros). Crearán un juego de registros que seleccione todos los valores de la tabla Locations En Dreamweaver, abra el cuadro de diálogo Juego de registros o Conjunto de datos (ASP) realizando una de estas operaciones: En la ficha Aplicación de la barra Insertar, haga clic en el botón Juego de registros o Conjunto de datos (ASP).

Elija Ventana > Vinculaciones para abrir el panel Vinculaciones y luego haga clic en el botón de signo más (+) y seleccione Juego de registros o Conjunto de datos. En el grupo de paneles Aplicación, seleccione el panel Vinculaciones, haga clic en el botón de signo más (+) y seleccione Juego de registros o Conjunto de datos. Aparecerá el cuadro de diálogo Juego de registros o Conjunto de datos sencillo. La siguiente pantalla muestra el cuadro de diálogo de juego de registros de . (En ASP, se trata del cuadro de diálogo Conjunto de datos; la mayoría de las opciones de juego de registros son las mismas para todos los modelos de servidor.)

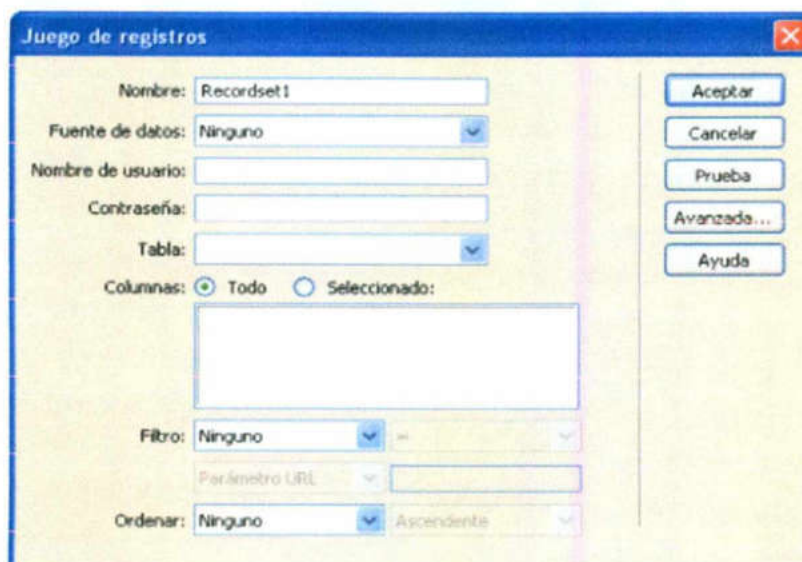


Fig. 5.1 Creación de juegos de registros en dreamweaver

En el cuadro de texto **Nombre**, introduzca **rsLocations**. Este es el nombre del juego de registros que está definiendo. En el menú emergente **Fuente de datos** o **Conexión** (otros tipos de páginas del servidor), seleccione **connGlobal**. El cuadro de diálogo **Juego de registros** o **Conjunto de datos** se actualizará y mostrará información para la primera tabla de la base de datos de **Global**.

Nota: “Si no aparece la conexión **connGlobal** en el menú, haga clic en el botón **Definir** para crearla. “

En el menú emergente **Tabla**, seleccione **LOCATIONS**. El juego de registros se actualizará con los registros de la tabla **LOCATIONS**.

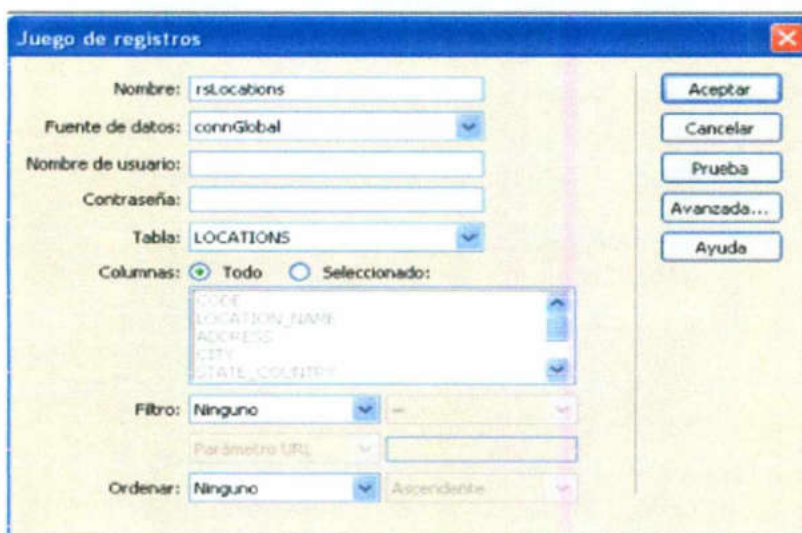


Fig. 5.2 Conexión a las tablas que se tienen hechas

Para Columnas, acepte el valor predeterminado, Todo. **Sugerencia:** Para limitar la información incluida en el juego de registros, puede elegir Seleccionado y luego seleccionar la columna o columnas con las que desea trabajar. Haga clic en Prueba para comprobar el juego de registros. Los registros de la base de datos que coincidan con su petición de juego de registros se mostrarán en la ventana Declaración SQL de prueba.

Registro	CODICE	LOCALI	ADDRESS	CITY	STATE	REGIO	TELEP	FAX
1	BWA	Baltimore	Asport Blvd	Baltimore	MD	1	410-121-	410-121-
2	CAI	Cairo Inte	Cairo Inte	Cairo	Egypt	4	2022652	2022652
3	CBR	Carberna	Carberna	Carberna	Australia	6	02 62496	02 62496
4	CNS	Cairo	Cairo Au	Cairo	Queensla	6	07 40359	07 40359
5	CPT	Cape To	Cape To	Cape Town	South Afs	4	0272 19	0272 19
6	CWB	Alonso P	Av. N. Sr	Curtiba	Brazil	2	55 41269	55 41269
7	CZM	Aeropuert	Cozumel	Cozumel	Mexico	2	5298723	5298723
8	DEN	Denver I.	100 Pena	Denver	CO	1	303-667-	303-667-
9	DFW	Dallas Ft	3838 Nor	Dallas, TX	TX	1	972-455-	972-455-
10	EZE	Eze	Autopista	Buenos A.	Argentina	2	00-54-44	00-54-44
11	FRA	Frankfurt	General	Frankfurt	Germany	3	069 6959	069 6959
12	KDC	Kansai In	Senshuk	Osaka	Japan	5	(0)724 55	(0)724 55
13	LHR	Heathrow	Heathrow	London	United Ki	3	020 8897	020 8897
14	MO	Kansas C.	902 Tel A	Kansas City	MO	1	916-243	916-243
15	MSP	Minneapolis	4300 Glu	Minneapolis	MN	1	651-698	651-698
16	MUC	Franz-Jos	Franz-Jos	Munich	Germany	3	089 9788	089 9788
17	NRT	New Tok	Chiba 28	Nanta	Japan	5	(0)476 32	(0)476 32
18	PMO	Punta Rasi	Punta Ra	Palermo	Sicily	3	091 213	091 213
19	PNA	Noan	Asport	Ponplona	Spain	3	948 3115	948 3115
20	SIN	Singapor	Singapor	Singapore	Singapore	5	(65)542-5	(65)542-5
21	TPS	Vincenzo	Vincenzo	Trapani	Italy	3	0923 842	0923 842
22	YGK	Kingston	675 Princ	Kingston	Ontario	1	613-531-	613-531-
23	YQG	Windsor I	Windsor	Windsor	Ontario	1	519-250	519-250

Fig. 5.3 Muestra el contenido de la tabla que seleccionó

Haga clic en Aceptar para cerrar la ventana Declaración SQL de prueba. Haga clic en Aceptar para cerrar el cuadro de diálogo Juego de registros o Conjunto de datos y añadir el código del juego de registros a la página. Dreamweaver confirmará que se ha añadido un juego de registros a la página y éste aparecerá en el panel Vinculaciones.

Sugerencia: Si no ve todos los campos del juego de registros, haga clic en el botón de signo más para ampliar la ramificación del juego de registros.



Fig. 5.4 Muestra que la conexión existe

Insertar un objeto de aplicación Juego de páginas Maestro/Detalle

A continuación, deberá utilizar un objeto de aplicación para crear un juego de páginas Maestro/Detalle. Los objetos de aplicación crean scripts de disposición y del lado del servidor para numerosas aplicaciones Web comunes. Después de seleccionar varias opciones, el objeto de aplicación Juego de páginas Maestro/Detalle generará los formularios y los scripts tanto para la lista maestra como para la página que contiene los detalles. Además, el objeto de aplicación crea una barra de navegación por los registros y un contador de registros en la página Maestro. Guarde el documento (Archivo > Guardar). El archivo locationMaster debe estar aún abierto en la ventana de documento; sitúe el punto de inserción después del texto "Rental Locations" y presione Intro o Retorno para establecer el lugar en el que debe insertarse el objeto Live. Inserte los datos mediante la realización de una de las siguientes operaciones: En la ficha Aplicación de la barra Insertar, haga clic en el botón Maestro-Detalle o arrástrelo al documento. Elija Insertar > Objetos de aplicación > Juego de páginas Maestro/Detalle. Aparecerá el cuadro de diálogo Juego de páginas Maestro-Detalle.

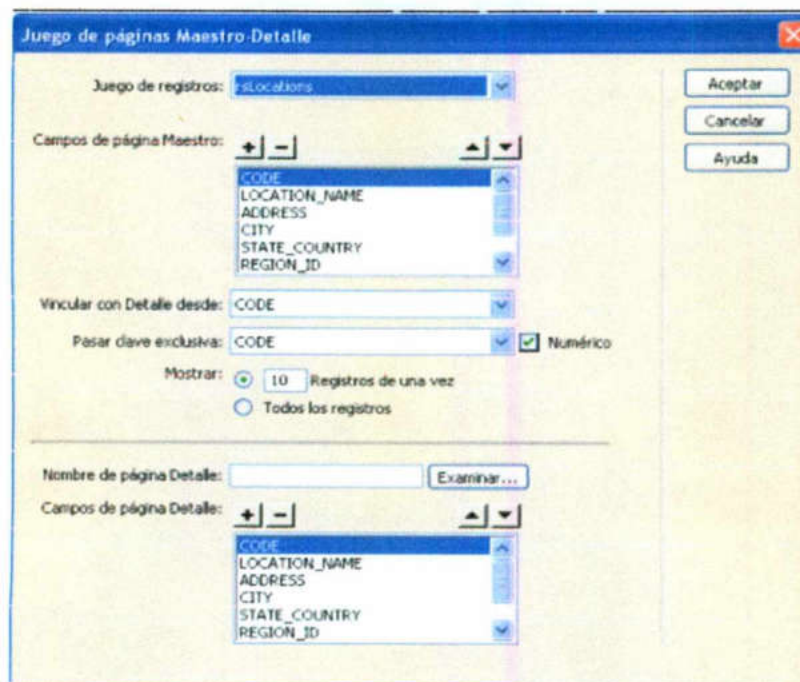


Fig. 5.5 Juego de pagina maestro detalle

En el menú emergente Juego de registros de este cuadro de diálogo, seleccione rsLocations. En la lista Campos de página Maestro, seleccione CODE y luego haga clic en el botón de signo menos (-) para eliminarlo de la lista de datos que aparece en la página Maestro. Elimine todos los campos de la lista Campos de página Maestro excepto LOCATION_NAME, CITY y STATE_COUNTRY.

En el menú emergente Vincular con Detalle desde, seleccione LOCATION_NAME. Para ver la página Detalle correspondiente a una sucursal, es necesario crear un vínculo para cada sucursal que aparezca en la lista Maestro. Cuando un visitante del sitio haga clic en Location Name en la página Maestro, se abrirá la página Detalle correspondiente.

En el menú emergente Pasar clave exclusiva, acepte el valor predeterminado, CODE, y asegúrese de que Numérico no está activado. El campo CODE es la clave exclusiva o clave principal de la tabla Locations. Este campo contiene un código exclusivo de tres letras para cada registro. En Mostrar, acepte el valor predeterminado para mostrar los registros de diez en diez. En Nombre de página Detalle, haga clic en Examinar; seguidamente, en el cuadro de diálogo Seleccionar archivo, vaya hasta el archivo locationDetail situado en la carpeta del sitio. Haga clic en Aceptar.

En Campos de página Detalle, establezca los campos que deben aparecer en la página Detalle realizando una de las siguiente operaciones: En la lista Campos de página Detalle, seleccione CODE y luego haga clic en el botón de

signo menos (-). Seleccione REGION_ID y luego haga clic en el botón de signo menos (-). Haga clic en Aceptar.

Dreamweaver actualizará las páginas Maestro y Detalle y añadirá todos los scripts de servidor y las consultas para la página Maestro y la página Detalle. La página Maestro se actualizará. Ésta incluye una tabla para los datos de la base de datos, una tabla para navegación por la página del juego de registros y un contador de registros del juego.

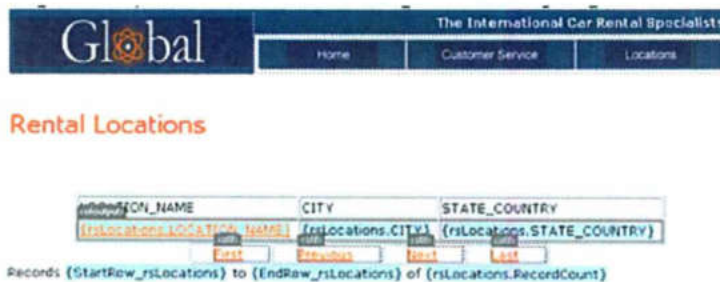


Fig. 5.6 Ejemplo de paginas maestro detalle

La página Detalle también se actualizará. Esta incluye una tabla en la que se enumeran los detalles de cada registro de la página Maestro.



Fig. 5.7 Como utilizar el recordset en dreamweaver

Visualizar las páginas

A continuación, deberá guardar las páginas y visualizarlas para comprobar el funcionamiento de la aplicación que ha desarrollado. Para ver las páginas tal y como se mostrarían en una vez procesadas por el servidor, obtenga una vista previa de éstas en un navegador. En la ventana de documento, seleccione el documento locationDetail y elija Archivo > Guardar para guardar el trabajo realizado. En el panel Sitio, seleccione el documento locationDetail y luego haga clic en el botón Colocar archivos (flecha azul que señala hacia arriba) para copiar el archivo local en el servidor. En la ventana de documento, seleccione el documento locationMaster y elija Archivo > Guardar para guardar el trabajo realizado. En el panel Sitio, seleccione el documento locationMaster y luego haga clic en el botón Colocar archivos (flecha azul que señala hacia arriba) para copiar el archivo local en el servidor. Con el documento locationMaster aún seleccionado, presione F12 para ver la página en un navegador. Compruebe los vínculos de la página Maestro con la página Detalle

LOCATION_NAME	CITY	STATE_COUNTRY
Baltimore-Washington International	Baltimore	MD
Cairo International Airport	Cairo	Egypt
Canberra	Canberra	Australia
Cairns	Cairns	Queensland
Cape Town Airport	Cape Town	South Africa
Curitiba Fozes	Curitiba	Brazil
Aeropuerto Intl. De Cozumel	Cozumel	Mexico
Denver International	Denver	CO
Dallas Ft. Worth International	Dallas/Ft Worth	TX
Eze	Buenos Aires	Argentina

Records 1 to 10 of 23

Next Last

Fig. 5.8 Visualización de una página dinámica

Debajo de la tabla de datos, haga clic en **Siguiente** para ver el siguiente grupo de diez registros. Se actualizarán la tabla de datos, la barra de navegación y el contador de registros. Haga clic en uno de los nombres de sucursales para ver la página **Detalle**. La página **Maestro** pasará la clave exclusiva al servidor, el servidor procesará la consulta y se abrirá la página **Detalle** con los datos correspondientes a la sucursal seleccionada. Presione el botón **Locations** situado en la parte superior de la página para regresar a la lista de sucursales. Cierre la ventana del navegador cuando termine de ver las páginas.

Creación de una página de inserción de registro

Este tutorial le servirá de guía para crear un formulario que inserte información en una base de datos. Al crear la página de inserción, aprenderá a insertar y definir campos de formulario. También aprenderá a crear un campo de formulario que permite a los usuarios insertar información mediante un menú generado dinámicamente.

Puede realizar este tutorial en cualquiera de los modelos de servidor compatibles con : , ASP, ASP.NET, JSP y PHP.

Al realizar este tutorial, llevará a cabo las siguientes tareas:

- Cree una página de inserción
- Añada objetos de formulario
- Defina un comportamiento de servidor Insertar registro
- Compruebe la página

Cree una página de inserción

La página que va a crear permitirá al administrador del sitio insertar información sobre nuevas sucursales de alquiler de vehículos en la base de datos de Global Car Rental.

Comenzará en una página parcialmente finalizada. Durante el desarrollo del tutorial, añadirá a la página objetos de formulario y contenido dinámico.

Comencemos seleccionando una página en la que trabajar.

Realice una de las siguientes operaciones para abrir el panel Sitio: En el grupo de paneles Archivos, haga clic en la flecha de ampliación y seleccione la ficha Sitio si no está ya seleccionada. Elija Ventana > Sitio. Presione F8.

Se abrirá el panel Sitio.

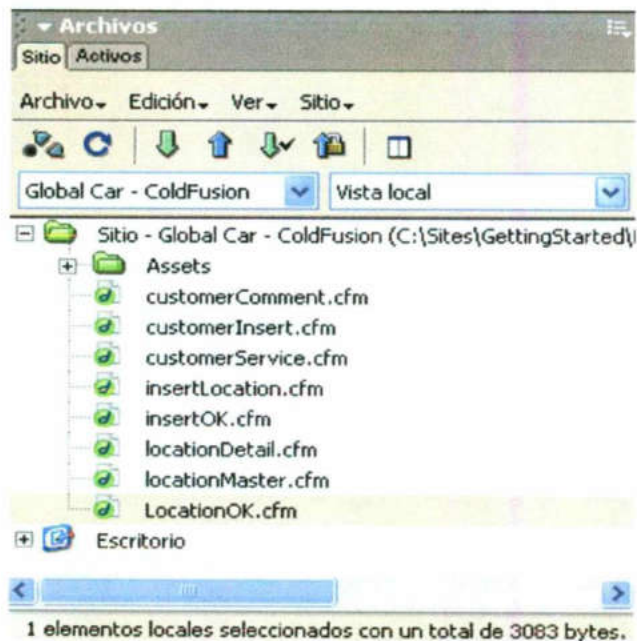


Fig. 5.9 Crea una página de inserción

En el menú emergente Sitio, seleccione el sitio Global Car que definió para su tipo de servidor de aplicaciones. En el panel Sitio, haga doble clic en el documento insert Location para abrirlo. El documento se abrirá en la ventana de documento

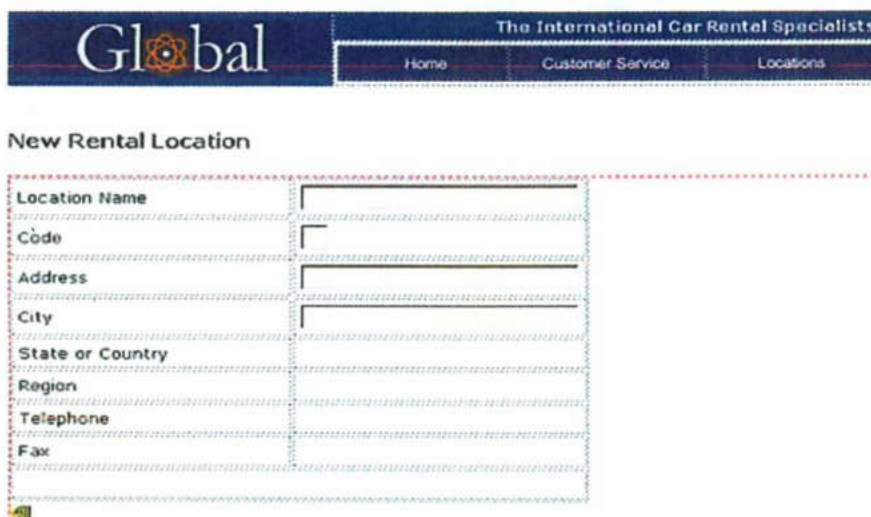


Fig. 5.10 Plantilla de página web para insertar datos

Añada objetos de formulario

Ya se han añadido a esta página una tabla y algunos de los campos del formulario. En el documento aparece un rectángulo rojo con contorno discontinuo que indica los límites del formulario. Dreamweaver inserta automáticamente en un documento los límites de un formulario al insertar un objeto de formulario.

A continuación, añadirá el resto de los campos del formulario para la página de inserción: tres campos de texto, un campo lista/menú, un botón submit (enviar) y un botón restablecer.

También proporcionará etiquetas para los campos del formulario. Al asignar nombres a los campos, le resultará más fácil identificarlos mientras trabaja con ellos al conectar los campos del formulario con los campos de la base de datos.

Comenzaremos observando uno de los campos de formulario ya existentes.

Abra el inspector de propiedades (Ventana > Propiedades), si aún no está abierto. Mueva el puntero a la celda superior derecha y haga clic en el campo de formulario para seleccionarlo. El inspector de propiedades se actualizará para mostrar las propiedades del campo de formulario.



Fig. 5.11 barra de propiedades de dreamweaver

El campo Campo de texto contiene el nombre del objeto de formulario y el campo Ancho car. (ancho en caracteres) establece el ancho del campo en 30

caracteres. Deberá actualizar estos mismos dos campos en los campos de texto que añada al formulario.

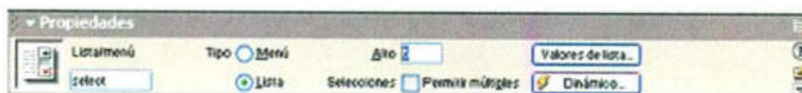
En el documento, sitúe el punto de inserción en la celda de tabla situada a la derecha de la etiqueta State or Country y luego realice una de las siguientes operaciones para insertar un campo de texto: En la barra Insertar, haga clic en la ficha Formulario y haga clic en el botón Campo de texto (el segundo por la izquierda) o arrástrelo desde la barra Insertar hasta la celda de la tabla. Elija Insertar > Objetos de formulario> Campo de texto. El objeto de formulario Campo de texto se insertará en el documento.

Con el campo de texto aún seleccionado, en el inspector de propiedades, introduzca **state_country** en el cuadro de texto Campo de texto y **30** en el cuadro de texto Ancho car. (ancho en caracteres).



Fig. 5.12 Propiedades de texto

En el documento, sitúe el punto de inserción en la celda de tabla situada a la derecha de la etiqueta Region y luego realice una de las siguientes operaciones para insertar una Lista/menú: En la categoría Formularios de la barra Insertar, haga clic en el botón Lista/menú o arrástrelo desde la barra Insertar hasta la celda de tabla. Elija Insertar > Objetos de formulario> Campo de texto. El objeto de formulario Lista/menú se insertará en el documento



En el inspector de propiedades, introduzca **region_id** en el cuadro de texto Nombre. Los objetos Lista/menú generan su ancho en caracteres en función de la longitud del elemento más largo de la lista o menú. Un poco más adelante definiremos los elementos de menú y los valores para este campo.

Inserte un campo de texto para Telephone y luego, en el inspector de propiedades, asígnele el nombre **telephone** y especifique en el campo Ancho car. (ancho en caracteres) el valor **30**. Inserte un campo de texto para Fax y luego, en el inspector de propiedades, asígnele el nombre **fax** y especifique en el campo Ancho car. (ancho en caracteres) el valor **30**. En el documento, sitúe el punto de inserción en la última celda de tabla

The screenshot shows a web page for 'Global The International Car Rental Specialists'. The navigation bar includes 'Home', 'Customer Service', and 'Locations'. The main content area is titled 'New Rental Location' and contains a form with the following fields: Location Name, Code, Address, City, State or Country, Region (a dropdown menu), Telephone, and Fax. The form is enclosed in a dashed red border.

Fig. 5.14 Plantilla de pagina web con campos de selección

Añada al formulario un botón Submit realizando una de las siguientes operaciones: En la categoría Formularios de la barra Insertar, haga clic en el botón Botón o arrástrelo desde la barra Insertar hasta la celda de tabla. Elija Insertar > Objetos de formulario> Botón. Se insertará un botón Submit (enviar) en el documento. En el inspector de propiedades, el botón ya cuenta con una etiqueta y la acción configurada es Enviar formulario



Fig. 5.15 Propiedades del botón submit

Repita el paso 10 para añadir otro botón al formulario. Añadirá un botón restablecer para permitir al usuario restablecer el formulario si necesita hacerlo. En el inspector de propiedades, seleccione la Acción Restablecer formulario. En el cuadro de texto Nombre del botón, introduzca **Reset**. El formulario ya está completo y debe tener un aspecto similar a este:

The screenshot shows a web page header for 'Global The International Car Rental Specialists'. Below the header is a navigation menu with 'Home', 'Customer Service', and 'Locations'. The main content area is titled 'New Rental Location' and contains a form with the following fields: Location Name, Code, Address, City, State or Country, Region (a dropdown menu), Telephone, and Fax. At the bottom of the form are 'Submit' and 'Reset' buttons.

Fig. 5.16 pag web

Defina un comportamiento de servidor Insertar registro

Deberá utilizar un comportamiento de servidor para crear los scripts de servidor necesarios para la aplicación. Deberá crear una conexión entre los datos del campo de formulario y los de la base de datos para que, al completar el formulario y hacer clic en el botón Submit, se inserte la información en la base de datos.

En el panel Comportamientos del servidor (Ventana > Comportamientos del servidor), haga clic en el botón de signo más (+) y seleccione Insertar registro del menú emergente. Aparecerá el cuadro de diálogo Insertar registro

The screenshot shows a dialog box titled 'Formulario de inserción de registros'. It contains several fields: 'Fuente de datos' (set to 'Registro'), 'Nombre de usuario', 'Contraseña', and 'Tabla'. There is a checkbox for 'Pasará cadena de consulta original' and an 'Examinar...' button. Below these is a 'Campos de formulario' section with a table for mapping columns to form fields. At the bottom, there are fields for 'Etiqueta' and 'Mostrar como' (set to 'Campo de texto') and 'Enviar como' (set to 'Texto'). On the right side, there are 'Aceptar', 'Cancelar', and 'Ayuda' buttons.

Columna	Etiqueta	Mostrar como	Enviar como

Fig. 5.17 Formulario de inserción de registros

En el menú emergente Fuente de datos () o Conexión (otros tipos de páginas de servidor), seleccione connGlobal. En el menú emergente Insertar en la tabla, seleccione LOCATIONS.

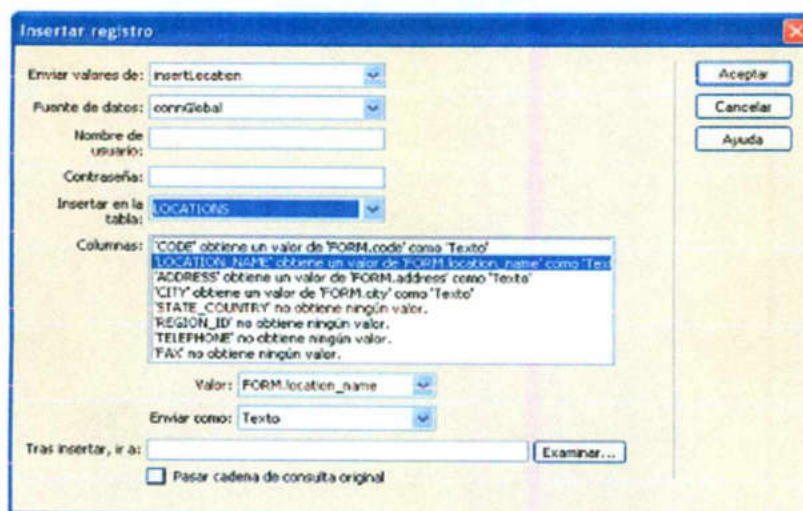


Fig. 5.18 Como insertar registros

La lista Columnas se actualizará con información sobre las relaciones entre los campos del formulario y los de la base de datos. Asegúrese de que el valor de REGION_ID está configurado como Numérico o Entero (ASP.NET), dado que el valor de Region ID es un valor numérico, no un valor de texto.

En Tras insertar, ir a o Si es correcto, ir a (ASP.NET), haga clic en Examinar y, en el cuadro de diálogo que aparece, seleccione el archivo LocationOK; luego haga clic en Aceptar para cerrar el cuadro de diálogo. Haga clic en Aceptar para cerrar el cuadro de diálogo Insertar registro. En la ventana de documento, el formulario se actualizará y el panel Comportamientos del servidor mostrará el comportamiento Insertar registro añadido. Guarde el documento

Compruebe la página

A continuación, deberá copiar la página finalizada en el servidor, abrirla en un navegador e insertar algunos datos.

En el panel Sitio, seleccione insertLocation y luego haga clic en el botón Colocar archivos (flecha azul que señala hacia arriba) para copiar el archivo local en el servidor. Seleccione Sí cuando se le pregunte si desea copiar los archivos dependientes. Con insertLocation seleccionado en la ventana de documento, elija Archivo > Vista previa en el navegador o presione F12 para ver el documento. Introduzca datos de prueba en el formulario y haga clic en el botón Submit. Aparecerá la página locationOKpage indicando que la información se ha insertado correctamente en la base de datos de Global. Para ver la entrada que ha añadido a la base de datos, haga lo siguiente:

En el grupo de paneles Aplicación, seleccione el panel Bases de datos. Localice la base de datos connGlobal y haga clic en el botón de signo más (+) situado delante de ésta para ampliarla. Haga clic en el botón de signo más (+) situado delante de Tables para ver las tablas de la base de datos. Haga clic con el botón derecho del ratón (Windows) o presione Control y haga clic (Macintosh) en la tabla LOCATIONS y luego elija Ver datos. Aparecerán los registros de la base de datos; el nuevo registro es la última entrada de la base de datos

Aspectos básicos de las conexiones de base de datos con ASP

Una aplicación ASP debe conectar con una base de datos a través de un controlador ODBC (Controlador de conectividad de base de datos abierta, Open Database Connectivity) o de un proveedor OLE DB (Base de datos de vinculación e incrustación de objetos, Object Linking and Embedding Database). El controlador o proveedor actúa como un intérprete que permite que la aplicación Web se comuniquen con la base de datos. Para obtener más información sobre la función de los controladores de base de datos, consulte Interfaz con la base de datos. La siguiente tabla muestra algunos de los controladores que se pueden utilizar con bases de datos Microsoft Access, Microsoft SQL Server y Oracle:

Base de datos	Controlador de base de datos
Microsoft Access	Controlador de Microsoft Access (ODBC)
Microsoft SQL Server	Controlador de Microsoft SQL Server (ODBC) Proveedor de Microsoft SQL Server (OLE DB)
Oracle	Controlador de Microsoft para Oracle (ODBC) Proveedor de OLE DB para Oracle

Tabla 5.1 Conexiones de base de datos con ASP

Puede utilizar un DSN (Nombre de fuente de datos, Data Source Name) o una cadena de conexión para conectarse a la base de datos.

Un DSN es un identificador de una palabra, como Acme, que señala a la base de datos y contiene toda la información necesaria para conectar con ella. Un DSN se define en Windows. Puede utilizar un DSN si está conectando a través de un controlador ODBC instalado en un sistema Windows. Si desea obtener instrucciones detalladas, consulte Creación de una conexión con DSN.

Una cadena de conexión es una expresión codificada manualmente que identifica la base de datos y contiene la información necesaria para conectar con ella. Ejemplo:

```
Driver={SQL Server};Server=Socrates;Database=AcmeMktg;  
UID=wiley;PWD=roadrunner
```

Debe utilizar una cadena de conexión si se conecta con la base de datos mediante uno de los siguientes elementos:

- Un proveedor OLE DB
- Un controlador ODBC que no esté instalado en un sistema Windows

Para obtener instrucciones detalladas, consulte las siguientes secciones:

- Creación de una conexión sin DSN
- Conexión con una base de datos en un ISP

Nota: También puede utilizar una cadena de conexión si conecta a través de un controlador ODBC instalado en un sistema Windows, pero en tal caso, resulta más sencillo utilizar un DSN.

Puede utilizar un DSN para crear una conexión ODBC entre la aplicación Web y la base de datos. Un DSN es un nombre que contiene todos los parámetros necesarios para conectar con una base de datos ODBC concreta

Aspectos básicos de los DSN

Un DNS es una palabra exclusiva que identifica un grupo de parámetros de conexión de base de datos. Entre los parámetros pueden figurar el nombre del servidor, la ruta de la base de datos o el nombre de la base de datos, el controlador ODBC que se va a utilizar y el nombre de usuario y la contraseña, en caso de que sean necesarios.

Por ejemplo, supongamos que tiene una base de datos Microsoft SQL Server denominada Precinct ubicada en un servidor llamado Kojak. Para obtener acceso a la base de datos, debe introducir el nombre de usuario "columbo" y la contraseña "savalas7". Tras utilizar estos parámetros para definir un DSN denominado ourcops, puede crear la conexión introduciendo la palabra ourcops en , en lugar de utilizar todos los demás parámetros

Creación de una conexión utilizando un DSN local

Puede utilizar un DSN definido en un equipo local para crear una conexión de base de datos en Dreamweaver. Si desea utilizar un DSN local, el DSN debe estar definido en el equipo Windows que está ejecutando Dreamweaver.

Para crear una conexión de base de datos con un DSN definido en el equipo local:

Defina un DSN en el equipo Windows que está ejecutando Dreamweaver. Para obtener instrucciones, consulte Creación de un DSN. Abra una página ASP en Dreamweaver y, a continuación, abra el panel Bases de datos (Ventana > Bases de datos). Dreamweaver muestra todas las conexiones definidas para el sitio. Haga clic en el botón de signo más (+) en el panel y seleccione Nombre de fuente de datos (DSN) en el menú emergente. Aparecerá el cuadro de diálogo Nombre de fuente de datos (DSN)

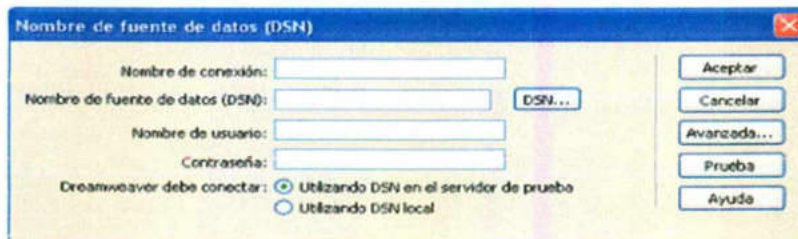


Fig. 5.19 Conexión a BD con DSN local

Introduzca un nombre para la nueva conexión. **Nota:** No utilice espacios ni caracteres especiales en el mismo.

En la parte inferior del cuadro de diálogo, seleccione la opción Utilizando DSN local. En el menú emergente Nombre de fuente de datos (DSN), seleccione el DSN que desee utilizar. Si desea utilizar un DSN local pero aún no lo ha definido, haga clic en DSN para abrir el Administrador de origen de datos ODBC de Windows. Para obtener instrucciones, consulte Creación de una conexión con DSN. Si es preciso, complete los cuadros Nombre de usuario y Contraseña. Si lo desea, restrinja el número de elementos de la base de datos que puede recuperar Dreamweaver en tiempo de diseño haciendo clic en Avanzada e introduciendo un esquema o nombre de catálogo. Para obtener más información, consulte Restricción de la información de base de datos que se muestra en Dreamweaver.

Nota: No puede crear un esquema o catálogo en Microsoft Access.

Haga clic en Prueba. Dreamweaver intentará conectar con la base de datos. Si la conexión falla, compruebe de nuevo el DSN. Si la conexión continúa fallando, compruebe la configuración de la carpeta que utiliza Dreamweaver para procesar las páginas dinámicas (consulte Especificación de dónde pueden procesarse las páginas dinámicas). Haga clic en Aceptar. La nueva conexión se muestra en el panel Bases de datos

Creación de una conexión utilizando un DSN remoto

Puede utilizar un DSN definido en un equipo remoto para crear una conexión de base de datos en Dreamweaver. Si desea utilizar un DSN remoto, el DSN debe estar definido en el equipo Windows que está ejecutando el servidor de aplicaciones (probablemente IIS).

Para crear una conexión de base de datos con un DSN definido en un equipo remoto:

Defina un DSN en el sistema remoto que está ejecutando el servidor de aplicaciones. Para obtener instrucciones, consulte Creación de un DSN. Abra una página ASP en Dreamweaver y, a continuación, abra el panel Bases de datos (Ventana > Bases de datos). Dreamweaver muestra todas las conexiones definidas para el sitio. Haga clic en el botón de signo más (+) en el panel y seleccione Nombre de fuente de datos (DSN) en el menú emergente. Aparecerá el cuadro de diálogo Nombre de fuente de datos (DSN). Introduzca un nombre para la nueva conexión.

Nota: No utilice espacios ni caracteres especiales en el mismo.

En la parte inferior del cuadro de diálogo, seleccione la opción Utilizando DSN en el servidor de prueba. Los usuarios de Macintosh pueden pasar por alto este paso porque todas las conexiones de base de datos utilizan DSN en el servidor de aplicaciones.

Introduzca el DSN. Puede hacer clic en el botón DSN para conectar con el servidor y elegir uno de los DSN definidos en él. Para obtener más información, consulte Configuración de las opciones del cuadro de diálogo Seleccionar ODBC DSN.

Si es preciso, complete los cuadros Nombre de usuario y Contraseña. Si lo desea, restrinja el número de elementos de la base de datos que puede recuperar Dreamweaver en tiempo de diseño haciendo clic en Avanzada e introduciendo un esquema o nombre de catálogo. Para obtener más información, consulte Restricción de la información de base de datos que se muestra en Dreamweaver.

Nota: No puede crear un esquema o catálogo en Microsoft Access.

Haga clic en Prueba. Dreamweaver intentará conectar con la base de datos. Si falla la conexión, compruebe de nuevo el DSN. Si la conexión continúa fallando, compruebe la configuración de la carpeta que utiliza Dreamweaver para procesar las páginas dinámicas (consulte Especificación de dónde pueden procesarse las páginas dinámicas). Haga clic en Aceptar. La nueva conexión se muestra en el panel Bases de datos

Creación de una conexión sin DSN

Puede utilizar una conexión sin DSN para crear una conexión ODBC u OLE DB entre la aplicación Web y la base de datos.

Para crear la conexión, se utiliza una cadena de conexión. Una cadena de conexión combina toda la información que la aplicación Web del servidor

necesita para conectar con la base de datos. Dreamweaver inserta esta cadena en los scripts del lado del servidor de la página que el servidor de aplicaciones procesará posteriormente.

A continuación se ofrece un ejemplo de cadena de conexión:

```
Driver={Microsoft Access Driver (*.mdb)};  
DBQ=C:\inetpub\wwwroot\Academy\curriculum.mdb
```

He aquí un segundo ejemplo:

```
Driver={SQL Server};Server=Socrates;Database=MedCenter;  
UID=mwelby;PWD=realme
```

Creación de una conexión de base de datos con una cadena de conexión

Puede utilizar una cadena de conexión para crear una conexión de base de datos entre la aplicación Web y la base de datos.

Abra una página ASP en Dreamweaver y, a continuación, abra el panel Bases de datos (Ventana > Bases de datos). Dreamweaver muestra todas las conexiones definidas para el sitio, si las hubiera. Haga clic en el botón de signo más (+) del panel y seleccione Cadena de conexión personalizada en el menú emergente. Aparecerá el cuadro de diálogo Cadena de conexión personalizada como se muestra a continuación.

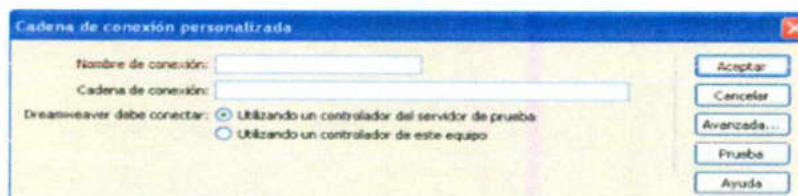


Fig. 5.20 Cadena de conexión

Rellene el cuadro de diálogo y haga clic en Aceptar

Configuración de las opciones del cuadro de diálogo Cadena de conexión personalizada

La finalidad de este cuadro de diálogo es definir una conexión de base de datos sin DSN con una cadena de conexión.

Procedimiento

En el primer cuadro, introduzca un nombre para la nueva conexión.

Nota: No utilice espacios ni caracteres especiales en el mismo.

En el segundo cuadro, introduzca una cadena de conexión con la base de datos. Si no especifica un proveedor OLE DB en la cadena de conexión (es decir, si no incluye un parámetro Provider o de proveedor) ASP utilizará automáticamente el proveedor OLE DB para controladores ODBC. En dicho caso, deberá especificar un controlador ODBC adecuado para la base de datos. Si el servidor en el que se encuentra el sitio es de un ISP y desconoce la ruta completa de la base de datos, utilice el método MapPath del objeto de servidor ASP en la cadena de conexión. Para obtener más información, consulte Conexión con una base de datos en un ISP.

Si el controlador de base de datos especificado en la cadena de conexión no está instalado en el mismo equipo que Dreamweaver, seleccione la opción Utilizando un controlador del servidor de prueba. Los usuarios de Macintosh pueden omitir este paso ya que todas las conexiones de base de datos utilizan el servidor de aplicaciones.

Si lo desea, restrinja el número de elementos de la base de datos que puede recuperar Dreamweaver en tiempo de diseño haciendo clic en Avanzada e introduciendo un esquema o nombre de catálogo. Para obtener más información, consulte Restricción de la información de base de datos que se muestra en Dreamweaver.

Nota: No puede crear un esquema o catálogo en Microsoft Access.

Haga clic en Prueba. Dreamweaver intentará conectar con la base de datos. Si falla la conexión, compruebe de nuevo la cadena de conexión. Si la conexión continúa fallando, compruebe la configuración de la carpeta que utiliza Dreamweaver para procesar las páginas dinámicas (consulte Especificación de dónde pueden procesarse las páginas dinámicas). Haga clic en Aceptar

Cómo escribir una cadena de conexión

Una cadena de conexión combina toda la información que la aplicación Web del servidor necesita para conectar con la base de datos. Dreamweaver inserta esta cadena en los scripts del lado del servidor de la página que el servidor de aplicaciones procesará posteriormente.

Una cadena de conexión para bases de datos Microsoft Access y SQL Server consta de una combinación de los siguientes parámetros separados por punto y coma:

Provider (proveedor) especifica el proveedor OLE DB para la base de datos. Por ejemplo, a continuación se incluyen parámetros para proveedores OLE DB comunes para bases de datos Access, SQL Server y Oracle, respectivamente:

Provider=Microsoft.Jet.OLEDB.4.0;...

Provider=SQLOLEDB;...

Provider=OraOLEDB;...

Para conocer el valor del parámetro de su proveedor OLE DB, consulte la documentación del fabricante o pregunte al administrador del sistema.

Si no incluye el parámetro Provider, se utilizará el proveedor OLE DB predeterminado para ODBC y deberá especificar un controlador ODBC adecuado para la base de datos.

Driver (controlador) especifica el controlador ODBC que se utilizará si no se especifica un proveedor OLE DB para la base de datos.

Server (servidor) especifica el servidor que aloja la base de datos SQL Server si la aplicación Web se ejecuta en un servidor distinto.

Database (base de datos) es el nombre de una base de datos SQL Server.

DBQ es la ruta de una base de datos basada en archivos, como, por ejemplo, una base de datos creada en Microsoft Access. La ruta es la existente en el servidor que aloja el archivo de base de datos. **UID** especifica el nombre del usuario. **PWD** especifica la contraseña del usuario. **DSN** es el nombre de la fuente de datos, en el caso de que se utilice. En función de cómo defina el DSN en el servidor, podrá omitir los demás parámetros de la cadena de conexión. Por ejemplo, DSN=Results puede ser una cadena de conexión válida si define el resto de los parámetros al crear el DSN (consulte Creación de un DSN).

Es posible que las cadenas de conexión para otros tipos de bases de datos no utilicen los parámetros enumerados más arriba o, en caso de hacerlo, llamen a los parámetros o los utilicen de forma distinta. Para obtener más información, consulte la documentación del fabricante de la base de datos o pregunte al administrador del sistema.

A continuación se ofrece un ejemplo de cadena de conexión que crea una conexión ODBC con una base de datos Access denominada trees.mdb:

```
Driver={Microsoft Access Driver (*.mdb)};  
DBQ=C:\inetpub\wwwroot\Research\trees.mdb
```

A continuación se ofrece un ejemplo de cadena de conexión que crea una conexión OLE DB con una base de datos SQL Server denominada Mothra ubicada en un servidor llamado Gojira:

```
Provider=SQLOLEDB;Server=Gojira;Database=Mothra;UID=jsmith;  
PWD=orlando8
```

Creación de una conexión OLE DB

Puede utilizar un proveedor OLE DB para comunicar con la base de datos. La creación de una conexión OLE DB directa puede mejorar la velocidad de la conexión mediante la eliminación de la capa ODBC entre la aplicación Web y la base de datos. Utilizando un proveedor OLE DB específico para base de datos, elimina el intermediario ODBC.

Si no especifica un proveedor OLE DB para la base de datos, ASP utilizará el proveedor OLE DB predeterminado para controladores ODBC para comunicar con un controlador ODBC que, a su vez, comunica con la base de datos.

OLE DB está disponible sólo con Windows NT, 2000 o XP.

Existen distintos proveedores OLE DB para las distintas bases de datos. Puede obtener proveedores OLE DB para Microsoft Access y SQL Server descargando e instalando los paquetes Microsoft Data Access Components (MDAC) 2.5 y 2.6 en el equipo Windows que está ejecutando IIS o PWS (cuando sea pertinente). Puede descargar gratuitamente los paquetes MDAC del sitio Web de Microsoft.

Nota: Instale MDAC 2.5 antes de instalar MDAC 2.6.

Puede descargar proveedores OLE DB para bases de datos Oracle9i y Oracle8i del sitio Web de Oracle (es preciso registrarse).

En , una conexión OLE DB se crea mediante la inclusión de un parámetro Provider (proveedor) en una cadena de conexión. Por ejemplo, a continuación se incluyen parámetros para proveedores OLE DB comunes para bases de datos Access, SQL Server y Oracle, respectivamente:

```
Provider=Microsoft.Jet.OLEDB.4.0;...  
Provider=SQLOLEDB;...  
Provider=OraOLEDB;...
```

Para conocer el valor del parámetro de su proveedor OLE DB, consulte la documentación del fabricante o pregunte al administrador del sistema

Utilización de una ruta virtual para conectar con una base de datos

Para escribir una cadena de conexión sin DSN en un archivo de base de datos situado en un servidor remoto, debe conocer la ruta física del archivo. Por ejemplo, he aquí una cadena de conexión sin DSN típica para una base de datos Microsoft Access:

```
Driver={Microsoft Access Driver (*.mdb)};  
DBQ=c:\inetpub\wwwroot\accounts\users\jsmith\data\statistics.mdb
```


Si no conoce la ruta física de sus archivos en el servidor remoto, puede obtener la ruta utilizando el método `MapPath` de su cadena de conexión.

Para crear una conexión sin DSN con el método `MapPath`:

Cargue el archivo de base de datos en el servidor remoto. Anote su ruta virtual, como por ejemplo `/jsmith/data/statistics.mdb`. Abra una página ASP en Dreamweaver y, a continuación, abra el panel Bases de datos (Ventana > Bases de datos). Dreamweaver muestra todas las conexiones definidas para el sitio. Haga clic en el botón de signo más (+) del panel y seleccione Cadena de conexión personalizada en el menú emergente. Introduzca un nombre para la nueva conexión.

Nota: No utilice espacios ni caracteres especiales en el mismo.

Escriba la cadena de conexión y utilice el método `MapPath` para proporcionar el parámetro `DBQ`. Suponga que la ruta virtual de su base de datos Microsoft Access es `/jsmith/data/statistics.mdb`. La cadena de conexión puede expresarse como se indica a continuación si utiliza el lenguaje de scripts VBScript:

```
"Driver={Microsoft Access Driver (*.mdb)};DBQ=" & Server.MapPath(
"/jsmith/data/statistics.mdb")
```

El carácter ampersand (&) se utiliza para concatenar (combinar) dos cadenas. La primera cadena va entre comillas y la segunda la devuelve la expresión `Server.MapPath`. Al combinar dos cadenas, se crea la siguiente cadena:

```
Driver={Microsoft Access Driver (*.mdb)}; ↵
DBQ=C:\inetpub\wwwroot\accounts\users\jsmith\data\statistics.mdb
```

Si utiliza JavaScript, la expresión es idéntica excepto en que el signo que se utiliza para concatenar dos cadenas es el más (+) en lugar del ampersand (&):

```
"Driver={Microsoft Access Driver (*.mdb)};DBQ=" + Server.MapPath(
"/jsmith/data/statistics.mdb")
```

Seleccione la opción Utilizando un controlador del servidor de prueba. Los usuarios de Macintosh pueden omitir este paso ya que todas las conexiones de base de datos utilizan el servidor de aplicaciones. Haga clic en Prueba.

Dreamweaver intentará conectar con la base de datos. Si falla la conexión, compruebe de nuevo la cadena de conexión. Si la conexión sigue fallando, póngase en contacto con el ISP para comprobar si el controlador de base de datos que ha especificado en la cadena de conexión está instalado en el servidor remoto. Compruebe asimismo que el ISP tiene la versión más reciente del controlador. Por ejemplo, una base de datos creada en Microsoft Access

2000 no funciona con Microsoft Access Driver 3.5. Necesita Microsoft Access Driver 4.0 o una versión posterior. Haga clic en Aceptar.

La nueva conexión se muestra en el panel Bases de datos. Actualice la conexión de base de datos de las páginas dinámicas existentes y utilice la nueva conexión con las páginas que cree. Para actualizar la conexión de una página dinámica, abra la página en Dreamweaver, haga doble clic en el nombre del juego de registros en el panel Vinculaciones o en el panel Comportamientos del servidor y seleccione la conexión que acaba de crear en el menú emergente Conexión

Aspectos básicos de las rutas físicas y virtuales

Después de cargar los archivos en un servidor remoto mediante Dreamweaver, los archivos residen en una carpeta en el árbol local de directorios del servidor. Por ejemplo, en un servidor que ejecute Microsoft IIS, la ruta de la página principal podría ser la siguiente:

```
c:\inetpub\wwwroot\accounts\users\jsmith\index.htm
```

Esta ruta se conoce como la ruta física del archivo.

No obstante, el URL para abrir el archivo no utiliza la ruta física. Utiliza el nombre del servidor o dominio seguido de una ruta virtual, como en el siguiente ejemplo:

```
www.plutoserve.com/jsmith/index.htm
```

La ruta virtual, /jsmith/index.htm, sustituye a la ruta física, c:\inetpub\wwwroot\accounts\users\jsmith\index.htm

Localización de una ruta física de archivo con la ruta virtual

Si trabaja con un ISP, no siempre conocerá la ruta física de los archivos que cargue. Los ISP suelen proporcionarle un servidor FTP, posiblemente un directorio de servidor y un nombre de conexión y la contraseña. Los ISP también especifican un URL para ver las páginas en Internet, como por ejemplo www.plutoserve.com/jsmith/.

Si conoce el URL, puede obtener la ruta virtual del archivo (es la ruta que sigue al nombre de servidor o de dominio en un URL). Una vez conoce la ruta virtual, puede obtener la ruta física del archivo en el servidor mediante el método MapPath.

Entre otras cosas, el método MapPath toma la ruta virtual como un argumento y devuelve la ruta física y el nombre del archivo. He aquí la sintaxis del método:


```
Server.MapPath("/virtualpath")
```

Suponga que `/jsmith/index.htm` es una ruta virtual de archivo. La siguiente expresión devolverá su ruta física:

```
Server.MapPath("/jsmith/index.htm")
```

Puede experimentar con el método `MapPath` como se explica a continuación.

Abra una página ASP en Dreamweaver y cambie a la vista Código (Ver > Código).

Introduzca la siguiente expresión en el código de la página HTML.

```
<%Response.Write(stringvariable)%>
```

Utilice el método `MapPath` para obtener un valor para el argumento `stringvariable`. He aquí un ejemplo:

```
<% Response.Write(Server.MapPath("/jsmith/index.htm")) %>
```

Cambie a la vista Diseño (Ver > Diseño) y active Live Data (Ver > Live Data) para ver la página. La página muestra la ruta física del archivo en el servidor de aplicaciones. Utilizando el ejemplo tratado en esta sección, la página muestra la siguiente ruta física:

```
c:\inetpub\wwwroot\accounts\users\jsmith\index.htm
```

Para obtener más información sobre el método `MapPath`, consulte la documentación en línea que viene con Microsoft IIS o PWS

Creación de la página de búsqueda

Una página de búsqueda en la Web normalmente contiene campos de formulario en los que los usuarios introducen parámetros de búsqueda. Cuando el usuario hace clic en el botón `Buscar` del formulario, se envían los parámetros de búsqueda a la página de resultados residente en el servidor. La página de resultados residente en el servidor, no la página de búsqueda del navegador, es la responsable de recuperar los registros de la base de datos.

Para comenzar con esta parte de la aplicación Web, cree dos páginas: una página de búsqueda que permita a los usuarios introducir parámetros de búsqueda y una página de resultados para mostrar los registros encontrados. Puede incluso combinar las dos páginas en una sola. Como mínimo, la página de búsqueda debe incluir un formulario HTML con un botón `Enviar`.

Para añadir un formulario HTML a una página de búsqueda:

Abra la página de búsqueda y seleccione Formulario del menú Insertar. Se creará un formulario vacío en la página. Quizá deba activar los elementos invisibles (Ver > Ayudas visuales > Elementos invisibles) para ver los límites del formulario, que se representan por medio de líneas rojas finas. Añada objetos de formulario para que los usuarios introduzcan sus parámetros de búsqueda eligiendo Objetos de formulario del menú Insertar. Entre los objetos de formulario figuran los campos de texto, los menús de listas, las casillas de verificación y los botones de opción. Puede añadir tantos objetos de formulario como desee para ayudar a los usuarios a definir búsquedas precisas. No obstante, recuerde que cuanto mayor sea el número de parámetros de búsqueda de la página de búsqueda, más compleja será la declaración SQL. Para obtener más información sobre objetos de formulario, consulte Creación de formularios interactivos.

Añada al formulario un botón Enviar (Insertar > Objetos de formulario > Botón). Si lo desea, cambie la etiqueta del botón Enviar. Para ello, seleccione el botón, abra el inspector de propiedades (Ventana > Propiedades) e introduzca un nuevo valor en el cuadro Etiqueta. Por ejemplo, este es el inspector de propiedades de un botón con la etiqueta "buscar":



Fig 5.21 propiedades de edición

A continuación, deberá indicar al formulario el lugar al que debe enviar los parámetros de búsqueda cuando el usuario haga clic en el botón Enviar. Seleccione el formulario eligiendo la etiqueta <form> en el selector de etiquetas que se encuentra en la parte inferior de la ventana de documento, como se muestra en la imagen

```
<body> <table> <tr> <td> <form> </td> </tr> </table> </tr> </td> </table> </tr>
```

Fig. 5.22 propiedades de las páginas web

En el cuadro Acción del inspector de propiedades del formulario, introduzca el nombre del archivo de la página de resultados que llevará a cabo la búsqueda en la base de datos. En el menú emergente método, elija uno de los siguiente métodos para determinar cómo debe enviar los datos el formulario al servidor. GET envía los datos del formulario añadiéndolos al URL como una cadena de consulta. Dado que los URL están limitados a 8.192 caracteres, no utilice el método GET con formularios largos. POST envía los datos del formulario en el cuerpo de un mensaje. Default usa el método predeterminado del navegador (generalmente, GET). Ya ha terminado la página de búsqueda. A continuación deberá crear la página de resultados

Creación de la página de resultados

Después de que el usuario introduzca los parámetros de búsqueda, la aplicación deberá recuperar los registros de la base de datos. Esta tarea la realiza la página de resultados.

Estas son las tareas que realiza una página de resultados:

- Obtiene los parámetros de búsqueda de la página de búsqueda.
- Conecta con la base de datos y busca registros.
- Crea un juego de registros a partir de los registros encontrados.
- Muestra el contenido del juego de registros.

Si la página de resultados sólo tiene un parámetro de búsqueda (un solo campo de texto, por ejemplo), podrá crear la página de resultados sin consultas ni variables SQL. Lo único que tendrá que hacer es crear un juego de registros y después añadir un filtro para excluir registros que no cumplan las condiciones establecidas por los parámetros de búsqueda enviados por la página de búsqueda. Para obtener instrucciones, consulte *Búsqueda con un solo parámetro de búsqueda*. Si la página de búsqueda tiene varios parámetros de búsqueda, tendrá que escribir una declaración SQL y definir múltiples variables para ella. Para obtener instrucciones

Búsqueda con un solo parámetro de búsqueda

Si la página de búsqueda envía un único parámetro de búsqueda al servidor, puede crear la página de resultados sin consultas ni variables SQL. Un juego de registros básico se crea con un filtro que excluye registros que no cumplan el parámetro de búsqueda enviado por la página de búsqueda.

Nota: Si tiene varias condiciones de búsqueda, deberá utilizar el cuadro de diálogo *Juego de registros avanzado* para definir el juego de registros. El cuadro de diálogo *Juego de registros simple* sólo admite una condición de búsqueda. Para obtener más información, consulte *Búsqueda con múltiples parámetros de búsqueda*.

Para crear el juego de registros que contendrá los resultados de la búsqueda:

Abra la página de resultados en la ventana de documento. Cree un juego de registros abriendo el panel *Vinculaciones* (*Ventana > Vinculaciones*), haciendo clic en el botón de signo más (+) y seleccionando *Juego de registros o Conjunto de datos (ASP.NET)* en el menú emergente. Asegúrese de que aparece el cuadro de diálogo *Juego de registros o Conjunto de datos simple*.

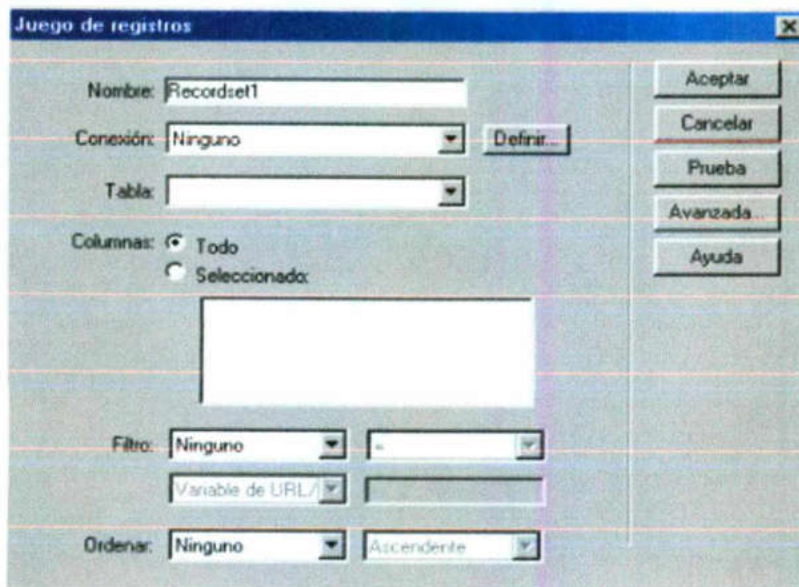


Fig. 5.23 Creando juego de registros

Si aparece la versión avanzada del cuadro de diálogo, cambie a la versión sencilla haciendo clic en el botón Simple. Introduzca un nombre para el juego de registros y elija una conexión.

La conexión debe establecerse con una base de datos que contenga datos en los que desea que el usuario realice búsquedas. En el menú emergente Tabla, seleccione la tabla de la base de datos en la que debe realizarse la búsqueda.

Nota: En una búsqueda con un solo parámetro, puede buscar registros en una única tabla. Para buscar en más de una tabla a la vez, deberá utilizar el cuadro de diálogo Juego de registros avanzado y definir una consulta SQL.

Para incluir algunas de las columnas de la tabla en el juego de registros, haga clic en Seleccionado y elija las columnas deseadas presionando Control (Windows) o Comando (Macintosh) mientras hace clic en ellas en la lista.

Elija las columnas que contienen la información que desea mostrar en la página de resultados.

Deje abierto por ahora el cuadro de diálogo Juego de registros o Conjunto de datos. Tendrá que utilizarlo a continuación para buscar los parámetros enviados por la página de búsqueda y crear un filtro de juego de registros para excluir registros que no cumplan los parámetros.

Para crear un filtro de juego de registros:

En el primer menú emergente del área Filtro, seleccione una columna de la tabla para compararla con el parámetro de búsqueda enviado por la página de búsqueda.

Por ejemplo, si el valor enviado por la página de búsqueda es un nombre de una ciudad, seleccione en la tabla la columna que contiene nombres de ciudades. Desde el menú emergente situado junto al primer menú, seleccione el signo igual (que debe ser el predeterminado). Esta elección indica que el usuario sólo desea obtener los registros para los que la columna de tabla seleccionada coincide exactamente con la especificada en la página de búsqueda. En el tercer menú emergente, seleccione Variable de formulario si el formulario de la página de búsqueda utiliza el método POST, o Parámetro de URL si utiliza el método GET. Este menú emergente especifica el lugar al que la página de búsqueda actualmente almacenada en el servidor debe enviarse el valor. En ASP, si el formulario de búsqueda utiliza el método POST, el valor se almacena en la colección Request.Form. Si el formulario de búsqueda utiliza el método GET, el valor se almacena en la colección Request.QueryString. En el cuarto cuadro, introduzca el nombre del objeto de formulario que acepta el parámetro de búsqueda en la página de búsqueda. Puede obtener el nombre cambiando a la página de búsqueda, haciendo clic en el objeto de formulario dentro del formulario para seleccionarlo y comprobando el nombre del objeto en el inspector de propiedades.

Por ejemplo, supongamos que desea crear un juego de registros que incluya sólo viajes de aventura a un país concreto. Supongamos que la tabla incluye una columna denominada TRIPLOCATION. Supongamos también que el formulario HTML de la página de búsqueda utiliza el método GET y que contiene un objeto Lista/menú denominado Location que muestra una lista de países. Esta es la apariencia que debe presentar la sección Filtro

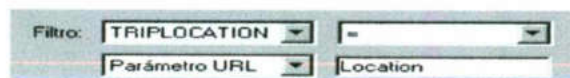


Fig. 5.24 Cómo crear filtros en las páginas web

Si lo desea, haga clic en Prueba, introduzca un valor de prueba y haga clic en Aceptar para conectar con la base de datos y crear una copia del juego de registros. El valor de prueba simula el valor que, de otro modo, habría sido devuelto por la página de búsqueda. Haga clic en Aceptar para cerrar el juego de registros. Si está satisfecho con el juego de registros, haga clic en Aceptar.

Dreamweaver inserta en la página un script del lado del servidor que, cuando se ejecuta en el servidor, comprueba cada registro de la tabla de la base de datos. Si el campo especificado en un registro cumple la condición de filtro, el registro se incluirá en el juego de registros. El script crea un juego de registros que contiene solamente los resultados de la búsqueda.

El siguiente paso consiste en mostrar el juego de registros en la página de resultados

Búsqueda con múltiples parámetros de búsqueda

Si la página de búsqueda envía más de un parámetro de búsqueda al servidor, deberá escribir una consulta SQL para la página de resultados y utilizar los parámetros de búsqueda en las variables SQL.

Después de que Dreamweaver inserte la consulta SQL en la página y que se ejecute la página en el servidor, se comprueba cada registro de la tabla de la base de datos. Si el campo especificado en un registro cumple las condiciones de la consulta SQL, el registro se incluirá en un juego de registros. La consulta SQL crea un juego de registros que contiene solamente los resultados de la búsqueda.

Por ejemplo, el personal de ventas puede estar en condiciones de informar sobre qué clientes de un área determinada tienen ingresos superiores a un nivel concreto. En un formulario de una página de búsqueda, el asociado comercial introduce un área geográfica y un nivel de ingresos mínimo y luego hace clic en el botón Enviar para enviar los dos valores a un servidor. En el servidor, los valores se pasan a la declaración SQL de la página de resultados, que crea un juego de registros sólo con los clientes del área especificada con ingresos superiores al nivel especificado.

Para buscar registros de una base de datos empleando SQL:

Abra la página de resultados en Dreamweaver, cree un nuevo juego de registros abriendo el panel Vinculaciones (Ventana > Vinculaciones), haciendo clic en el botón de signo más (+) y seleccionando Juego de registros o Conjunto de datos (ASP.NET) en el menú emergente.

Asegúrese de que aparece el cuadro de diálogo Juego de registros o Conjunto de datos avanzado

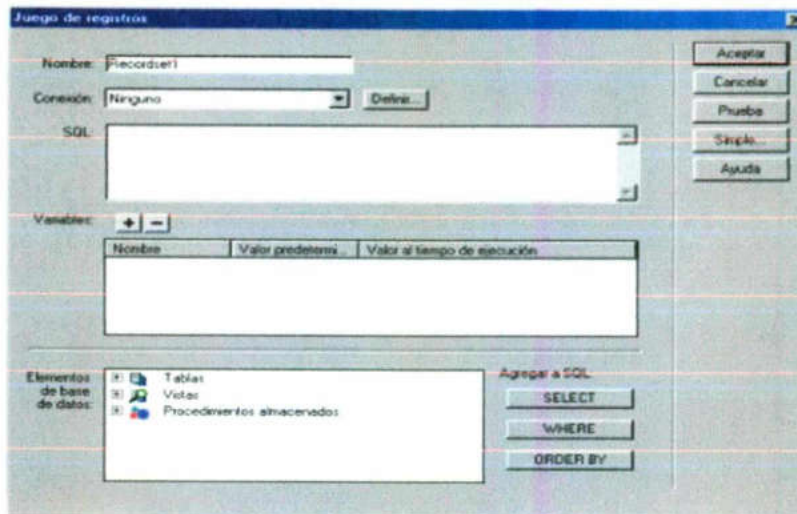


Fig. 5.25 Búsqueda de registros mediante SQL

Si aparece la versión sencilla del cuadro de diálogo, cambie a la versión avanzada haciendo clic en el botón Avanzada.

Introduzca un nombre para el juego de registros y elija una conexión. La conexión debe establecerse con una base de datos que contenga datos en los que desea que el usuario realice búsquedas. Introduzca una declaración de selección (Select) en el área de texto SQL. Asegúrese de que la declaración incluye una cláusula Where (donde) con variables que contengan los parámetros de búsqueda. En el siguiente ejemplo, las variables se denominan varLastName y varDept:

```
SELECT EMPLOYEEID, FIRSTNAME, LASTNAME, DEPARTMENT,
EXTENSION
FROM EMPLOYEE WHERE LASTNAME LIKE 'varLastName' AND
DEPARTMENT
LIKE 'varDept'
```

Para reducir la cantidad de datos que debe teclear, puede utilizar el árbol de elementos de la base de datos situado en la parte inferior del cuadro de diálogo Juego de registros avanzado. Para obtener instrucciones, consulte Creación de un juego de registros avanzado escribiendo SQL.

Para obtener ayuda sobre la sintaxis SQL, consulte Nociones básicas de SQL.

Asigne a las variables SQL los valores de los parámetros de búsqueda haciendo clic en el botón de signo más (+) en el área Variables e introduciendo el nombre de la variable, el valor predeterminado (el valor que toma la variable si no se devuelve ningún valor de tiempo de ejecución) y un valor de tiempo de ejecución (normalmente, un objeto de servidor que contiene un valor enviado por un navegador, como una variable de petición).

En el siguiente ejemplo ASP, el formulario HTML de la página de búsqueda utiliza el método GET y contiene un campo de texto denominado "LastName" y otro llamado "Department".

Nombre	Valor predetermi...	Valor al tiempo de ejecución
varLastName	%	Request.QueryString["LastName"]
varDept	%	Request.QueryString["Department"]

Fig. 5.26 Como agregar variables en una pagina

En , los valores de tiempo de ejecución serían #LastName# y #Department#. En JSP, los valores de tiempo de ejecución serían `request.getParameter("LastName")` y `request.getParameter("Department")`.

Si lo desea, haga clic en Prueba para crear una copia del juego de registros utilizando los valores predeterminados de las variables. Los valores predeterminados simulan el valor que, de otro modo, habría sido devuelto por la página de búsqueda. Haga clic en Aceptar para cerrar el juego de registros de prueba. Si está satisfecho con el juego de registros, haga clic en Aceptar

Creación de una página detalle para una página de resultados

El conjunto de páginas de búsqueda puede incluir una página detalle que muestre información sobre los registros concretos de la página de resultados. En la página de resultados, los registros se muestran normalmente en una región repetida y cada registro cuenta con un vínculo. Cuando un usuario hace clic en uno de los vínculos, se abre la página de resultados y muestra información sobre el registro seleccionado

Modificación de una página de resultados para que funcione con una página detalle

La página de resultados debe contar con una región repetida para mostrar varios registros a la vez, al tiempo que cada registro de la región repetida debe tener un vínculo con la página detalle. El vínculo no sólo debe abrir la página detalle, sino que también debe indicar a la página detalle el registro que ha seleccionado el usuario. El comportamiento de servidor Ir a página Detalle crea este tipo de vínculo (sólo para usuarios de ASP y JSP).

Los usuarios de PHP, ASP.NET y deben crear un vínculo con parámetros de URL en el código. Para obtener más información, consulte Creación de parámetros de URL utilizando vínculos HTML.

Para modificar una página de resultados para que funcione con una página detalle (sólo usuarios de ASP y JSP):

Cree una página detalle vacía (Archivo > Nuevo) y asigne un nombre al archivo.

Deje la página a un lado por el momento. Deberá trabajar en ella posteriormente.

Abra la página de resultados en Dreamweaver. Asegúrese de que aparecen los resultados en una región repetida. Para obtener instrucciones, consulte Visualización de múltiples comportamientos.

En la región repetida, seleccione el texto o la imagen que actuará como vínculo. Si está trabajando en la ventana Live Data, seleccione el texto o la imagen en la primera región mostrada.

Nota: El texto o imagen que seleccione puede ser dinámico.

Cree el vínculo con la página detalle abriendo el panel Comportamientos del servidor (Ventana > Comportamientos del servidor), haciendo clic en el botón de signo más (+) y seleccionando Ir a página Detalle en el menú emergente.

Aparecerá el cuadro de diálogo Ir a página Detalle.

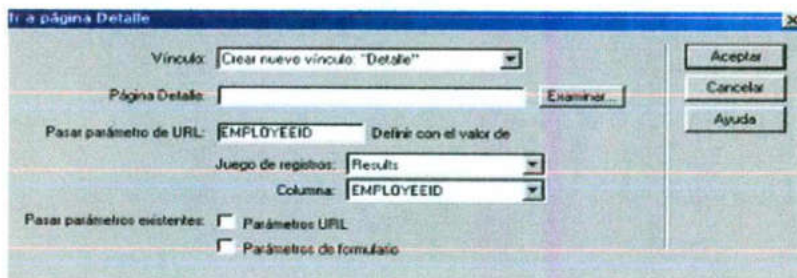


Fig. 5.27 modificar parámetros de una página

Seleccione las opciones deseadas en el cuadro de diálogo. Para obtener más información, consulte Configuración del cuadro de diálogo Ir a página Detalle. Haga clic en Aceptar.

La página de resultados pasa el valor a la página detalle en un parámetro de URL, que consiste simplemente en una variable añadida al final del URL empleado para abrir la página detalle. Por ejemplo, si el parámetro de URL se denomina id y la página detalle se denomina customerdetail.asp, el URL presentará esta apariencia cuando el usuario haga clic en el vínculo:

<http://www.mysite.com/customerdetail.asp?id=43>

La primera parte del URL, <http://www.mysite.com/customerdetail.asp>, abre la página detalle. La segunda parte, [?id=43](http://www.mysite.com/customerdetail.asp?id=43), es el parámetro de URL. Dicho parámetro indica a la página detalle qué registro debe localizar y mostrar. El

término id es el nombre del parámetro URL y 43 es su valor. En este ejemplo, el parámetro URL contiene el número de ID del registro, 43

Creación de la página detalle utilizando un comportamiento de servidor (sólo usuarios de ASP y JSP)

Después de modificar la página de resultados para que funcione con la página detalle, complete la página detalle. Puede crear la página detalle con una combinación de juego de registros normal y un comportamiento de servidor, o bien puede crearla sólo con un juego de registros filtrado. En esta sección se describe cómo crear la primera clase de página detalle (sólo para usuarios de ASP y JSP). Para aprender a crear la segunda clase, consulte Creación de la página detalle utilizando un juego de registros filtrado.

En primer lugar, diseñe la página detalle empleando las herramientas de diseño de Dreamweaver. Para obtener más información, consulte Creación de páginas Maestro/Detalle.

En segundo lugar, defina un juego de registros para la página o copie y pegue el juego de registros de la página de resultados. La página detalle extraerá los datos del registro de dicho juego de registros. Para obtener instrucciones, consulte Definición de un juego de registros y Copiado de un juego de registros de una página a otra.

En tercer lugar, vincule las columnas del juego de registros con la página. En el panel Vinculaciones (Ventana > Vinculaciones), seleccione columnas del juego de registros y arrástrelas hasta la página.

En cuarto lugar, añada un comportamiento de servidor que lea el ID del registro en el parámetro URL pasado por la página de resultados y que recupere el registro. Si omite este paso, el servidor recuperará el primer registro del juego de registros.

Para recuperar un registro específico utilizando un comportamiento de servidor (sólo usuarios de ASP y JSP):

En el panel Comportamientos del servidor (Ventana > Comportamientos del servidor), haga clic en el botón de signo más (+), seleccione Mover a registro del menú emergente y elija Mover a un registro concreto. Aparecerá el cuadro de diálogo Mover a un registro concreto

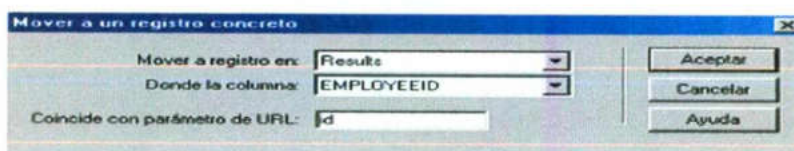


Fig. 5.28 Recuperar un registro especificado

Seleccione las opciones deseadas en el cuadro de diálogo. Para obtener más información, consulte Configuración de las opciones del cuadro de diálogo Mover a un registro concreto. Haga clic en Aceptar.

La próxima vez que un navegador solicite la página, leerá el ID de registro en el parámetro de URL transferido por el navegador y pasará al registro especificado en el juego de registros

Configuración de las opciones del cuadro de diálogo Mover a un registro concreto

La finalidad de este cuadro de diálogo es buscar un registro específico en un juego de registros.

Procedimiento

En el menú emergente Mover a registro en, seleccione el juego de registros definido en la página detalle. En el menú emergente Donde la columna, seleccione la columna que contiene el valor pasado por la página de resultados.

Por ejemplo, si la página de resultados pasa un número de ID de registro, elija la columna que contiene números de ID de registros. En el cuadro de texto Coincide con parámetro de URL, introduzca el nombre del parámetro de URL pasado por la página de resultados. Por ejemplo, si el URL utilizado por la página de resultados para abrir la página detalle es `www.mysite.com/customerdetail.asp?id=43`, introduzca `id` en el cuadro de texto Coincide con parámetro de URL.

También puede obtener el nombre cambiando a la página de resultados, abriendo el panel Comportamientos del servidor (Ventana > Comportamientos del servidor) y haciendo doble clic en el comportamiento de servidor Ir a página Detalle. Compruebe el nombre que figura en Pasar parámetro de URL.

Haga clic en Aceptar

Configuración del cuadro de diálogo Ir a página Detalle

La finalidad de este cuadro de diálogo es crear un vínculo que abra una página detalle y transfiera un ID de registro de modo que el registro seleccionado se pueda encontrar y mostrar en la página detalle.

Procedimiento

En el cuadro Página Detalle, haga clic en Examinar y localice la página detalle. Especifique el valor que desea pasar a la página detalle seleccionando un juego de registros y una columna de los menús emergentes Juego de registros

y Columna. Normalmente, el valor es exclusivo del registro, como, por ejemplo, el ID de clave exclusiva del registro. Si lo desea, transfiera parámetros de una página existente a la página detalle seleccionando las opciones Parámetros de URL o Parámetros de formulario. Haga clic en Aceptar.

Dreamweaver insertará un vínculo especial en la página. Cuando el usuario haga clic en el vínculo, se transferirá a la página detalle un parámetro de URL que contiene el ID de registro

Apertura de la página detalle y transferencia de un ID de registro a dicha página

Después de crear la página maestra y de mostrar registros en ella, deberá crear vínculos que abran la página detalle y transferirle el ID del registro que el usuario ha seleccionado.

Los usuarios de PHP, ASP.NET y deben crear un vínculo con parámetros de URL en el código. Para obtener más información, consulte Creación de parámetros de URL utilizando vínculos HTML. Sitúe las etiquetas anchor alrededor del contenido dinámico que desea utilizar como vínculo.

Para abrir la página detalle y transferirle un parámetro de URL que contenga el ID del registro en el que ha hecho clic el usuario (sólo usuarios de ASP y JSP):

En la región repetida de la página maestra, seleccione el contenido dinámico que va a hacer las veces de vínculo. En el panel Comportamientos del servidor (Ventana > Comportamientos del servidor), haga clic en el botón de signo más (+) y seleccione Ir a página Detalle en el menú emergente.

Aparecerá el cuadro de diálogo Ir a página Detalle.

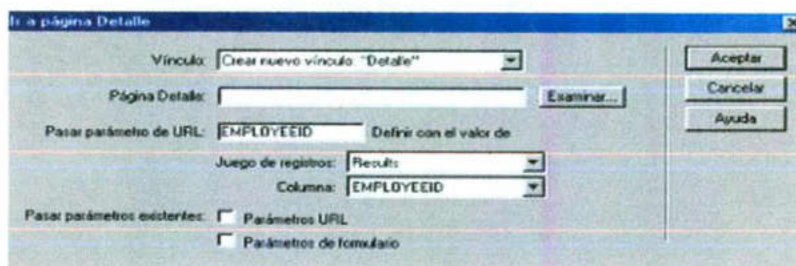


Fig. 5.29 Propiedades de la página detalle

Seleccione las opciones deseadas en el cuadro de diálogo. Para obtener más información, consulte Configuración del cuadro de diálogo Ir a página Detalle. Haga clic en Aceptar.

Dreamweaver situará un vínculo especial alrededor del texto seleccionado. Cuando el usuario hace clic en el vínculo, el comportamiento de servidor Ir a página Detalle transfiere un parámetro de URL que contiene el ID del registro a la página detalle. Por ejemplo, si el parámetro de URL se denomina id y la página detalle se denomina customerdetail.asp, el URL presentará esta apariencia cuando el usuario haga clic en el vínculo:

<http://www.mysite.com/customerdetail.asp?id=43>

La primera parte del URL, <http://www.mysite.com/customerdetail.asp>, abre la página detalle. La segunda parte, [?id=43](http://www.mysite.com/customerdetail.asp?id=43), es el parámetro de URL. Dicho parámetro indica a la página detalle qué registro debe localizar y mostrar. El término id es el nombre del parámetro URL y 43 es su valor. En este ejemplo, el parámetro URL contiene el número de ID del registro, 43.

Búsqueda y visualización del registro solicitado en la página detalle

Después de crear la página maestra, deberá buscar el registro solicitado en la base de datos y mostrarlo en la página detalle. El procedimiento consiste en definir un juego de registros para que contenga un registro —el registro solicitado por la página maestra— y vincular las columnas del juego de registros a la página.

Para buscar y mostrar el registro solicitado en la página detalle:

Cambie a la página detalle.

En el panel Vinculaciones, haga clic en el botón de signo más (+) y seleccione Juego de registros o Conjunto de datos (ASP.NET) en el menú emergente. Aparecerá el cuadro de diálogo sencillo. Si aparece el cuadro de diálogo avanzado, haga clic en Simple.

Asigne un nombre al juego de registros, luego elija una conexión y la tabla de base de datos que proporcionará datos al juego de registros.

En el área Columna, elija las columnas de la tabla que deben incluirse en el juego de registros. El juego de registros puede ser igual o distinto del juego de registros de la página maestra. Generalmente, el juego de registros de una página detalle incluye más columnas para mostrar más información.

Si los conjuntos de registros son diferentes, el juego de registros de la página detalle debe tener al menos una columna en común con la página maestra. La columna común a las dos páginas suele ser la columna de ID del registro, aunque también puede ser el campo de unión o las tablas relacionadas.

Para incluir algunas de las columnas de la tabla en el juego de registros, haga clic en Seleccionado y elija las columnas deseadas presionando Control (Windows) o Comando (Macintosh) mientras hace clic en ellas en la lista.

Complete la sección Filtro de la siguiente forma para localizar y mostrar el registro especificado en el parámetro URL pasado por la página de resultados:

En el primer menú emergente del área Filtro, seleccione la columna de tabla de la base de datos que contiene valores que coinciden con el valor del parámetro de URL pasado por la página maestra.

Por ejemplo, si el parámetro de URL contiene un número de ID de registro, elija la columna que contiene números de ID de registros.

En el menú emergente situado junto al primer menú, seleccione el signo igual (ya debe estar seleccionado).

En el tercer menú emergente, seleccione Parámetro de URL.

La página maestra pasa a la página detalle información que identifica la selección del usuario mediante un parámetro de URL.

En el cuarto cuadro, introduzca el nombre del parámetro de URL pasado por la página maestra. Por ejemplo, si el URL utilizado por la página maestra para abrir la página detalle es `www.mysite.com/customerdetail.asp?id=43`, introduzca id.

También puede obtener el nombre cambiando a la página maestra, abriendo el panel Comportamientos del servidor (Ventana > Comportamientos del servidor) y haciendo doble clic en el comportamiento de servidor Ir a página Detalle. Compruebe el nombre que figura en Pasar parámetro de URL.

Haga clic en Aceptar.

El juego de registros aparecerá en el panel Vinculaciones.

Vincule las columnas del juego de registros a la página detalle seleccionando las columnas del panel Vinculaciones (Ventana > Vinculaciones) y arrastrándolas hasta la página

Creación rápida de páginas Maestro/Detalle

Puede crear rápidamente páginas Maestro/Detalle utilizando el objeto Live Juego de páginas Maestro/Detalle.

Nota: El objeto Live no se admite en PHP ni en ASP.NET.

Para completar el juego de páginas Maestro/Detalle con un objeto Live:

Cree una página maestra vacía y añada a ella un juego de registros.

Asegúrese de que el juego de registros contiene, no sólo todas las columnas necesarias para la página maestra, sino también todas las columnas necesarias para la página detalle.

El juego de registros puede definirlo usted en tiempo de diseño (consulte Definición de un juego de registros) o el usuario en tiempo de ejecución (consulte Creación de páginas que realicen búsquedas en bases de datos).

Normalmente, el juego de registros de la página maestra extrae unas pocas columnas de una tabla de una base de datos, mientras que un juego de registros de la página detalle extrae más columnas de la misma tabla para proporcionar información adicional.

Abra la página maestra en la vista Diseño y elija Insertar > Objetos de aplicación > Juego de páginas Maestro/Detalle.

Aparecerá el cuadro de diálogo Juego de páginas Maestro/Detalle

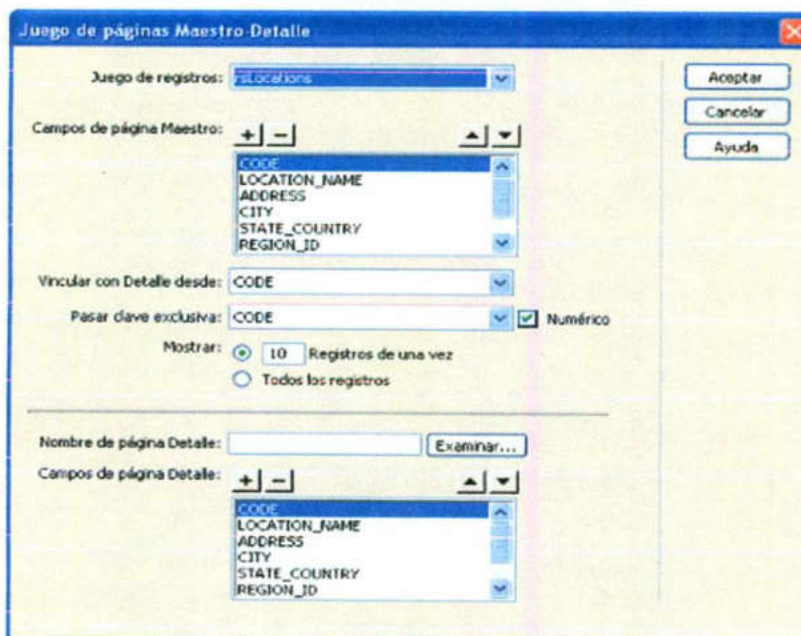


Fig. 5.30 Completa la pagina maestro detalle

Seleccione las opciones deseadas en el cuadro de diálogo. Para obtener más información, consulte Configuración de las opciones del cuadro de diálogo Juego de páginas Maestro/Detalle. Haga clic en Aceptar.

El objeto Live crea una página detalle (si no la ha creado usted antes) y añade contenido dinámico y comportamientos de servidor tanto a la página maestra como a la página detalle. Personalice la disposición de las páginas maestra y detalle de acuerdo con sus necesidades.

Puede personalizar totalmente la disposición de cada página utilizando las herramientas de diseño de páginas de Macromedia . También puede editar los comportamientos de servidor haciendo doble clic en ellos en el panel Comportamientos del servidor

Configuración de las opciones del cuadro de diálogo Juego de páginas Maestro/Detalle

La finalidad de este cuadro de diálogo es crear una página maestra y su correspondiente página detalle.

Procedimiento

En el menú emergente Juego de registros, asegúrese de que está elegido el juego de registros que contiene los registros que desea visualizar en la página maestra.

En el área Campos de página Maestro, seleccione las columnas del juego de registros que deben visualizarse en la página maestra.

De forma predeterminada, Dreamweaver selecciona todas las columnas del juego de registros. Si el juego de registros contiene una columna de clave exclusiva como, por ejemplo, recordID, selecciónela y haga clic en el botón de signo menos (-) para que no aparezca en la página.

Si desea cambiar el orden en el que aparecen las columnas en la página maestra, seleccione una columna de la lista y haga clic en la flecha arriba o abajo. En la página maestra, las columnas del juego de registros están dispuestas horizontalmente en una tabla. Al hacer clic en la flecha arriba, se desplaza la columna hacia la izquierda; al hacer clic en la flecha abajo, se desplaza la columna hacia la derecha.

En el menú emergente Vincular con Detalle desde, elija la columna del juego de registros que mostrará un valor que también servirá como vínculo con la página detalle.

Por ejemplo, si desea que cada nombre de producto de la página maestra tenga un vínculo con la página detalle, elija la columna del juego de registros que contiene los nombres de los productos.

En el menú emergente Pasar clave exclusiva, elija la columna del juego de registros que contiene valores que identifican a los registros.

Normalmente, la columna elegida es la que contiene el número de ID. Este valor se pasa a la página detalle para que pueda identificar el registro elegido por el usuario.

Desactive la casilla de verificación Numérico si la columna de clave exclusiva no es numérica.

Nota: Esta opción aparece activada de manera predeterminada y no aparece para todos los modelos de servidor.

Especifique el número de registros que deben mostrarse en la página maestra.

En el cuadro Nombre de página Detalle, haga clic en Examinar y localice el archivo de página detalle que ha creado o introduzca un nombre para que el objeto Live la cree automáticamente.

En el área Campos de página Detalle, seleccione las columnas que deben visualizarse en la página detalle.

De forma predeterminada, Dreamweaver selecciona todas las columnas del juego de registros de la página maestra. Si el juego de registros contiene una columna de clave exclusiva, como, por ejemplo, recordID, selecciónela y haga clic en el botón de signo menos (-) para que no aparezca en la página detalle.

Si desea cambiar el orden en el que aparecen las columnas en la página detalle, seleccione una columna de la lista y haga clic en la flecha arriba o abajo.

En la página detalle, las columnas del juego de registros están dispuestas verticalmente en una tabla. Al hacer clic en la flecha arriba, se desplaza la columna hacia arriba; al hacer clic en la flecha abajo, se desplaza la columna hacia abajo.

Haga clic en Aceptar.

El objeto Live crea una página detalle (si no la ha creado usted antes) y añade contenido dinámico y comportamientos de servidor tanto a la página maestra como a la página detalle

Creación de una página maestra y definición de un juego de registros para ella

El primer paso consiste en crear una página maestra vacía y añadirle un juego de registros.

El juego de registros puede definirlo usted en tiempo de diseño (consulte Definición de un juego de registros) o el usuario en tiempo de ejecución (consulte Creación de páginas que realicen búsquedas en bases de datos).

Asegúrese de que el juego de registros contiene, no sólo todas las columnas necesarias para la página maestra, sino también todas las columnas necesarias para la página detalle.

Normalmente, el juego de registros de la página maestra extrae unas pocas columnas de una tabla de una base de datos, mientras que un juego de registros de la página detalle extrae más columnas de la misma tabla para proporcionar información adicional

Visualización de los registros en la página maestra

Después de crear una página maestra en blanco y definir un juego de registros para ella, deberá mostrar los registros en la página.

Para ver los registros en la página maestra:

Cree una disposición de página que permita mostrar múltiples registros y vincule las columnas del juego de registros a la página.

Un método sencillo para lograrlo consiste en crear una tabla HTML con dos filas en la página maestra y arrastrar un número limitado de columnas del juego de registros desde el panel Vinculaciones (Ventana > Vinculaciones) hasta la segunda fila de la tabla. (Utilice la primera fila para mostrar los encabezados de las columnas de la tabla.)

Cree una región repetida para mostrar más de un registro a la vez.

La región repetida se aplica normalmente a la fila de la tabla en la que se encuentra el contenido dinámico

Utilización de páginas relacionadas (sólo usuarios de ASP y JSP)

En algunas situaciones, es posible que desee mostrar una página que no sea una página de búsqueda, una página de resultados ni una página detalle, pero sin perder la información que ha recibido la página de un formulario HTML o un parámetro de URL. En lugar de utilizar un vínculo estándar para abrir la página relacionada, cree el vínculo empleando el comportamiento de servidor Ir a página Relacionada. El vínculo resultante abrirá la página relacionada y transferirá parámetros existentes a dicha página. Por ejemplo, puede pasar parámetros de búsqueda de una página a otra para que el usuario no tenga que volver a introducir los parámetros de búsqueda.

Nota: Este comportamiento de servidor sólo está disponibles para páginas ASP y JSP.

Antes de añadir a una página un comportamiento de servidor Ir a página Relacionada, asegúrese de que la página recibe los parámetros de un formulario HTML (es decir, que el atributo ACTION del formulario especifica la página), o de un parámetro de URL, como cuando la página es el destino de un vínculo desde otra página con un comportamiento de servidor Ir a página Relacionada.

Para crear un vínculo que pase parámetros de formulario existentes a una página relacionada:

En la página, seleccione la cadena de texto o las imágenes que actualizarán como vínculo con la página relacionada.

En el panel Comportamientos del servidor (Ventana > Comportamientos del servidor), haga clic en el botón de signo más (+) y seleccione Ir a página Relacionada del menú emergente.

Aparecerá el cuadro de diálogo Ir a página Relacionada

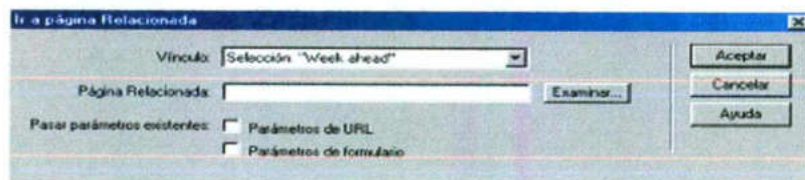


Fig. 5.31 Crea vínculo para pasar parámetros de un formulario a una pagina

Seleccione las opciones deseadas en el cuadro de diálogo.

Para obtener más información, consulte Configuración de las opciones del cuadro de diálogo Ir a página relacionada.

Haga clic en Aceptar.

Cuando el usuario haga clic en el nuevo vínculo, la página transferirá los parámetros a la página relacionada empleando parámetros de URL. Por ejemplo, supongamos que un parámetro de formulario se denomina "lastname" y que la página relacionada se denomina special_offer.asp. El URL tendrá la siguiente apariencia cuando el usuario haga clic en el vínculo:

http://www.mysite.com/special_offer.asp?lastname=Anderson

Creación de una página para insertar registros

Una aplicación puede incluir una página que permita a los usuarios insertar nuevos registros en una base de datos. Por ejemplo, la siguiente página inserta un nuevo registro en una base de datos de empleados

Una página de inserción debe incluir dos elementos:

- Un formulario HTML que permite a los usuarios introducir datos
- Un comportamiento de servidor Insertar registro que actualiza la base de datos

Puede añadir ambos elementos en una sola operación, mediante el objeto Live Formulario de inserción de registro, o por separado, con las herramientas de formulario de Dreamweaver y el panel Comportamientos del servidor.

Nota: La página de inserción sólo puede contener un comportamiento de servidor de edición de registros simultáneamente. Por ejemplo, no puede añadir un comportamiento de servidor Actualizar registro o Borrar registro a la página de inserción

Creación de una página para actualizar registros

Una aplicación puede incluir una página que permita a los usuarios actualizar los registros existentes en una tabla de la base de datos. Generalmente, una página de actualización es una página de detalle que funciona conjuntamente con una página de resultados. La página de resultados permite que el usuario seleccione el registro que desea actualizar y transfiere la información a la página de actualización.

Las páginas de actualización constan de tres elementos:

- Un juego de registros filtrado que permite recuperar el registro de una tabla de la base de datos.
- Un formulario HTML que permite a los usuarios modificar los datos del registro.
- Un comportamiento de servidor Actualizar registro que permite actualizar la tabla de la base de datos.

Puede añadir a la página el formulario HTML y el comportamiento de servidor en una sola operación, mediante el objeto Live Formulario de actualización de registros, o por separado, con las herramientas de formulario de Dreamweaver y el panel Comportamientos del servidor.

Nota: La página de actualización sólo puede contener un comportamiento de servidor de edición de registros. Por ejemplo, no puede añadir un

comportamiento de servidor Insertar registro o Borrar registro a la página de actualización.

Creación de una página para borrar un registro

Una aplicación puede incluir una página para permitir que los usuarios borren los registros existentes en una tabla de la base de datos. Generalmente, una página de borrado es una página de detalle que funciona conjuntamente con una página de resultados. La página de resultados permite que el usuario seleccione el registro que desea borrar y transfiere la información a la página de eliminación.

Las páginas de borrado constan de cuatro elementos:

- Un juego de registros filtrado que permite recuperar el registro de una tabla de la base de datos.
- Una presentación de sólo lectura de los datos que desea borrar.
- Un botón Enviar que permite enviar el comando de borrado al servidor.
- Un comportamiento de servidor Borrar registro que permite actualizar la tabla de la base de datos.

Nota: La página de borrado sólo puede contener un comportamiento de servidor de edición de registros. Por ejemplo, no puede añadir un comportamiento de servidor Insertar registro o Actualizar registro a la página de borrado

Utilización de procedimientos almacenados para modificar bases de datos

Puede emplear un procedimiento almacenado para modificar una base de datos. Un procedimiento almacenado es un elemento de base de datos almacenado que realiza alguna operación en la base de datos. Un procedimiento almacenado contiene código SQL que puede, entre otras cosas, insertar, actualizar o eliminar registros. Los procedimientos almacenados también pueden alterar la estructura de la base de datos. Por ejemplo, puede utilizar un procedimiento almacenado para añadir una columna de tabla o incluir borrar una tabla.

Un procedimiento almacenado también puede llamar a otro procedimiento almacenado, así como aceptar entradas y devolver múltiples valores al procedimiento llamado en forma de parámetros de salida.

Un procedimiento almacenado es reutilizable en el sentido de que se puede reutilizar una versión compilada del procedimiento para ejecutar una operación de base de datos varias veces. Si sabe que una tarea de base de datos se va a ejecutar muchas veces o que distintas aplicaciones van a ejecutar la misma

tarea, el uso de un procedimiento almacenado para ejecutar dicha tarea puede agilizar las operaciones de base de datos.

Nota: Las bases de datos mySQL y Microsoft Access no admiten procedimientos almacenados.

Antes de utilizar un procedimiento almacenado para modificar una base de datos, compruebe que dicho procedimiento contiene SQL que modifique la base de datos de algún modo. Para crear y almacenar uno en la base de datos, consulte la documentación de la base de datos y un buen manual de Transact-SQL.

El procedimiento para utilizar un procedimiento almacenado varía según el modelo de servidor

Para añadir un procedimiento almacenado a una página ASP:

En Dreamweaver, abra la página que ejecutará el procedimiento almacenado.

En el panel Vinculaciones, haga clic en el botón de signo más (+) y seleccione Comando (procedimiento almacenado). Aparecerá el cuadro de diálogo Comando.

Introduzca un nombre para el comando, elija una conexión con la base de datos que contiene el procedimiento almacenado y, a continuación, elija Procedimiento almacenado en el menú emergente Tipo. Elija el procedimiento almacenado ampliando la rama Procedimientos almacenados del cuadro Elementos de base de datos, eligiendo el procedimiento almacenado en la lista y haciendo clic en el botón Procedimiento.

Introduzca los parámetros necesarios en la tabla Variables.

No es preciso que introduzca ningún parámetro para variables RETURN_VALUE.

Haga clic en Aceptar.

Después de cerrar el cuadro de diálogo, Dreamweaver insertará código ASP en la página que, cuando se ejecute en el servidor, creará un objeto de comando que ejecute un procedimiento almacenado en la base de datos. El procedimiento almacenado realizará a su vez una operación de base de datos, como insertar un registro.

De forma predeterminada, el código define la propiedad Preparado del objeto Comando como true, lo que hace que el servidor de aplicaciones reutilice una versión compilada del objeto cada vez que se ejecuta el procedimiento almacenado. Si sabe que el comando se va a ejecutar muchas veces, puede utilizar una versión compilada del objeto para agilizar las operaciones de base

de datos. Sin embargo, si el comando sólo se va a ejecutar una o dos veces, el uso de una versión compilada puede ralentizar la aplicación Web, pues el sistema tiene que detenerse para compilar el comando. Para cambiar la configuración, pase a la vista Código y establezca la propiedad Preparado como false.

Nota: No todos los proveedores de base de datos admiten comandos preparados. Si su base de datos no los admite, es posible que aparezca un mensaje de error cuando ejecute la página. Cambie a la vista Código y establezca la propiedad Preparado como false.

Si el procedimiento almacenado toma parámetros, puede crear una página que recoja los valores de los parámetros y los envíe a la página con el procedimiento almacenado. Por ejemplo, puede crear una página que utilice parámetros de URL o un formulario HTML que recoja los valores de los parámetros de los usuarios

Ejecución del procedimiento almacenado en ASP.NET

Con las páginas ASP.NET en Dreamweaver, deberá añadir a una página un comportamiento de servidor Procedimiento almacenado para ejecutar un procedimiento almacenado.

Para añadir un procedimiento almacenado a una página ASP.NET:

En Dreamweaver, abra la página que ejecutará el procedimiento almacenado.

En el panel Vinculaciones, haga clic en el botón de signo más (+) y seleccione Procedimiento almacenado.

Aparecerá el cuadro de diálogo Procedimiento almacenado

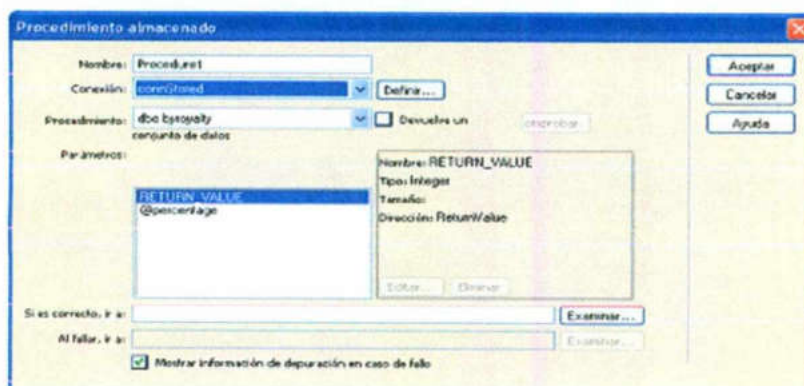


Fig. 5.32 Procedimientos almacenados en una página web

Seleccione las opciones deseadas en el cuadro de diálogo.

Para obtener más información, consulte Configuración de las opciones del cuadro de diálogo Procedimiento almacenado. Haga clic en Aceptar.

Después de cerrar el cuadro de diálogo, Dreamweaver insertará código ASP.NET en la página que, cuando se ejecute en el servidor, ejecutará un procedimiento almacenado en la base de datos. El procedimiento almacenado realizará a su vez una operación de base de datos, como insertar un registro o ejecutar una consulta.

Si el procedimiento almacenado toma parámetros, puede crear una página que recoja los valores de los parámetros y los envíe a la página con el procedimiento almacenado. Por ejemplo, puede crear una página que utilice parámetros de URL o un formulario HTML que recoja los valores de los parámetros de los usuarios

Creación de un comando que utilice SQL para editar una base de datos

Puede utilizar un objeto de comando ASP que utilice una declaración SQL para insertar, actualizar o eliminar registros de una base de datos.

Para crear un objeto de comando:

En Dreamweaver, abra la página ASP que ejecutará el comando.

Abra el panel Comportamientos del servidor (Ventana > Comportamientos del servidor), haga clic en el botón de signo más (+) y seleccione Comando.

Aparecerá el cuadro de diálogo Comando.

Introduzca un nombre para el comando, elija una conexión con la base de datos que contiene los registros que desea editar y elija la operación de edición que desea que realice el comando (Insertar, Actualizar o Eliminar).

Dreamweaver iniciará la declaración SQL basándose en el tipo de operación que usted elija. Por ejemplo, éste es el cuadro de diálogo después de elegir Insertar



Fig. 5.33 Creación de comandos para crear sentencias en SQL

Realice la declaración SQL.

Para obtener información sobre cómo escribir declaraciones SQL que modifiquen bases de datos, consulte un manual de Transact-SQL.

Utilice el área Variables para definir variables SQL.

Por ejemplo, a continuación se incluye una declaración Insertar que contiene tres variables SQL. Los valores de estas variables se obtienen a través de los parámetros de URL transferidos a la página, como se define en la columna Valor de tiempo de ejecución del área Variables

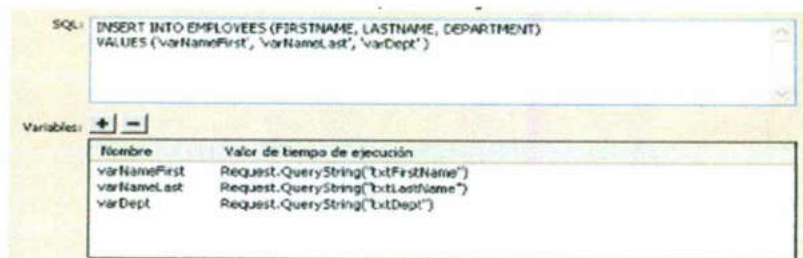


Fig. 5.34 Creación de script SQL con variables

Después de cerrar el cuadro de diálogo, Dreamweaver insertará código ASP en la página que, cuando se ejecute en el servidor, creará un comando que inserte, actualice o elimine registros de la base de datos.

De forma predeterminada, el código define la propiedad Preparado del objeto Comando como true, lo que hace que el servidor de aplicaciones reutilice una versión compilada del objeto cada vez que se ejecuta el comando. Para cambiar esta configuración, pase a la vista Código y establezca la propiedad Preparado como false.

En el ejemplo anterior, a continuación probablemente crearía una página con un formulario HTML en el que los usuarios podrán introducir datos de registros. El formulario HTML contendría tres campos de texto (txtFirstName, txtLastName, and txtDept) y un botón Enviar. Asimismo, el formulario utilizaría el método GET y enviaría los valores de campo de texto a la página que contiene el comando

Creación de una página de registro

Una aplicación Web puede contener una página que obligue a los usuarios a registrarse en su primera visita al sitio.

Nota: Dreamweaver no dispone de comportamientos de servidor de autenticación para páginas ASP.NET o PHP.

Por ejemplo, la siguiente página solicita a los usuarios primerizos que se registren:

courtesy title
 <Select>

first name* MI last name*

preferred greeting name

e-mail address*

Macromedia ID*

password*

Your password must be between 4 and 25 characters and contain at least one alpha character.

retype password*

remember me

Fig. 5.35 Ejemplo de una página de registro

Una página de registro consta de los siguientes bloques:

- Una tabla de base de datos para almacenar la información de conexión de los usuarios.
- Un formulario HTML en el que los usuarios pueden elegir un nombre de usuario y una contraseña

También puede utilizar el formulario para obtener información personal adicional de los usuarios.

- Un comportamiento de servidor Insertar registro para actualizar la tabla de la base de datos.
- Un comportamiento de servidor Comprobar nuevo nombre de usuario para comprobar que el nombre introducido por el usuario no sea utilizado por otro usuario.

Nota: Puede eliminar o cambiar las propiedades de cualquier comportamiento de servidor que añada a una página (consulte Edición de comportamientos de servidor).

Almacenamiento de la información de conexión de los usuarios

Una página de registro necesita utilizar una tabla de base de datos para almacenar la información de conexión que introducen los usuarios. Asegúrese de que la tabla de la base de datos contiene columnas de nombre de usuario y contraseña. Si desea conceder distintos privilegios de acceso a los usuarios, incluya también una columna de privilegio de acceso (consulte Almacenamiento de privilegios de acceso en la base de datos de usuarios).

Si desea establecer una contraseña común para todos los usuarios del sitio, configure la aplicación de base de datos (Microsoft Access, Microsoft SQL Server, Oracle, etc.) para introducir la contraseña de forma predeterminada en todo nuevo registro de usuario. La mayoría de las aplicaciones de base de datos permiten establecer el valor predeterminado de una columna cada vez que se crea un nuevo registro. Establezca el valor predeterminado de la contraseña.

También puede utilizar la tabla de la base de datos para almacenar otra información de utilidad sobre los usuarios.

Almacenamiento de privilegios de acceso en la base de datos de usuarios

Este elemento sólo es necesario si desea que los usuarios conectados puedan tener distintos privilegios de acceso. Si sólo desea que los usuarios se conecten, no necesita almacenar los privilegios de acceso.

Si desea que los usuarios conectado tengan distintos privilegios de acceso, asegúrese de que la tabla de la base de datos de usuario contiene una columna en la que se especifican los privilegios de acceso de cada usuario (invitado, usuario, administrador, etc.). El administrador del sitio deberá introducir en la base de datos los privilegios de acceso de cada usuario.

La mayoría de las aplicaciones de base de datos permiten establecer el valor predeterminado de una columna cada vez que se crea un nuevo registro. Defina el privilegio de acceso más habitual como valor predeterminado del sitio (por ejemplo, invitado) y cambie manualmente los casos que sean una excepción (por ejemplo, cambiar de invitado a administrador). Ahora el usuario tiene acceso a todas las páginas de administrador.

Asegúrese de que todos los usuarios de la base de datos tienen un solo privilegio de acceso (como invitado o administrador) y no varios (como "Usuario, Administrador"). Si desea establecer varios privilegios de acceso a las páginas (por ejemplo, para que todos los invitados y administradores puedan ver una página), establézcalos a nivel de la página, no de la base de datos. Para obtener más información, consulte Redireccionamiento de usuarios no autorizados a otra página.

Desconexión de usuarios

Cuando un usuario se conecta correctamente, se crea una variable de sesión que contiene el nombre de usuario correspondiente. Cuando el usuario abandona el sitio, se puede utilizar el comportamiento de servidor Desconectar usuario para borrar la variable de sesión y enviar al usuario a otra página (generalmente, una página de despedida o de agradecimiento).

Puede llamar al comportamiento de servidor Desconectar usuario cuando el usuario hace clic en un vínculo o al cargar una página dada.

Para añadir un vínculo para desconectar al usuario:

En la página, seleccione el texto o la imagen que se utilizará como vínculo.

En el panel Comportamientos de servidor, haga clic en el botón de signo más (+) y elija Autenticación de usuarios > Desconectar usuario. Aparecerá el cuadro de diálogo Desconectar usuario. Especifique la página que se debe abrir cuando el usuario haga clic en el vínculo. Generalmente, se trata de una página de despedida o agradecimiento. Haga clic en Aceptar.

Para desconectar un usuario al cargar una página concreta:

Abra la página que se cargará en Dreamweaver. Generalmente, se trata de una página de despedida o agradecimiento. En el panel Comportamientos de servidor, haga clic en el botón de signo más (+) y elija Autenticación de usuarios > Desconectar usuario. Aparecerá el cuadro de diálogo Desconectar usuario.

Seleccione la opción "Desconectar al Cargarse la página". Haga clic en Aceptar

Cómo permitir que los usuarios elijan un nombre de usuario y una contraseña

Para que los usuarios puedan elegir un nombre de usuario y una contraseña (si es aplicable), deberá añadir un formulario HTML a la página de registro.

Para permitir que los usuarios elijan un nombre de usuario y una contraseña:

Cree una nueva página (Archivo > Nuevo) y disponga la página de registro con las herramientas de diseño de Dreamweaver. Añada un formulario HTML situando el punto de inserción donde desea que aparezca el formulario y eligiendo Formulario en el menú Insertar. Se creará un formulario vacío en la página. Quizá deba activar los elementos invisibles (Ver > Ayudas visuales > Elementos invisibles) para ver los límites del formulario, que se representan por medio de líneas rojas finas.

Asigne un nombre al formulario HTML. Para ello, haga clic en la etiqueta <form>, situada en la parte inferior de la ventana de documento, para seleccionar el formulario, abra el inspector de propiedades (Ventana > Propiedades) e introduzca un nombre en el cuadro Nombre del formulario.

No es necesario que especifique los atributos action o method para indicar al formulario dónde y cómo enviar los datos del registro cuando el usuario presione el botón Enviar, ya que el comportamiento de servidor Insertar registro configura estos atributos automáticamente (consulte Actualización de la tabla de usuarios de la base de datos).

Añada campos de texto (Insertar > Objetos de formulario > Campo de texto) para permitir que el usuario introduzca el nombre de usuario y la contraseña.

El formulario también puede incluir otros objetos formulario para obtener información personal adicional. Deberá añadir etiquetas (de texto o imágenes) junto a cada objeto de formulario para indicar a los usuarios la información que deben introducir. También es recomendable alinear los objetos situándolos dentro de una tabla HTML. Para obtener más información sobre objetos de formulario, consulte Creación de formularios interactivos.

Añada al formulario un botón Enviar (Insertar > Objetos de formulario > Botón).

Si lo desea, cambie la etiqueta del botón Enviar. Para ello, seleccione el botón, abra el inspector de propiedades (Ventana > Propiedades) e introduzca un nuevo valor en el cuadro Etiqueta.

Por ejemplo, éste es el inspector de propiedades de un botón cuya etiqueta es "Register":



Fig. 5.36 Propiedades de botón

El siguiente paso consiste en añadir el comportamiento de servidor Insertar registro para insertar registros en la tabla de usuarios de la base de datos

Actualización de la tabla de usuarios de la base de datos

Para actualizar la tabla de usuarios de la base de datos deberá añadir un comportamiento de servidor Insertar registro.

Para actualizar la tabla de usuarios de la base de datos:

En el panel Comportamientos del servidor (Ventana > Comportamientos del servidor), haga clic en el botón de signo más (+) y seleccione Insertar registro del menú emergente.

Aparecerá el cuadro de diálogo Insertar registro.

Utilice los menús emergentes Conexión e Insertar en la tabla para especificar la tabla de usuarios de la base de datos.

En el cuadro de diálogo Tras insertar, ir a, introduzca la página que se debe abrir después de insertar el registro en la tabla. En el menú emergente Obtener valores de, elija el formulario HTML utilizado para obtener el nombre de usuario y la contraseña de los usuarios.

Dreamweaver elegirá automáticamente el primer formulario de la página.

Especifique el campo de la tabla de la base de datos que actualizará cada objeto del formulario. Para ello, seleccione un objeto de formulario en la lista Elementos de formulario y elija una columna de la tabla en el menú emergente Columna y el tipo de datos correspondiente en el menú emergente Enviar como.

El tipo de datos indica la clase de datos que espera recibir la columna de la tabla de la base de datos (texto, numérico, valores booleanos de casilla de verificación). Generalmente, las columnas de contraseña y nombre de usuario son de texto.

Por ejemplo, en la lista Elementos de formulario, haga clic en el campo de texto de contraseña, seleccione la columna de la tabla de base de datos en la que se almacenará la contraseña y elija el tipo de datos Texto. Repita el mismo procedimiento para cada objeto de formulario de la lista Elementos de formulario. Haga clic en Aceptar.

El último paso consiste en comprobar que el nombre de usuario no está siendo utilizado por otro usuario registrado

Comprobación de que el nombre de usuario elegido es exclusivo

Para asegurarse de que el nombre de usuario introducido no está siendo utilizado por otro usuario registrado, deberá añadir un comportamiento de servidor.

Cuando el usuario hace clic en el botón Enviar de la página de registro, el comportamiento de servidor compara el nombre utilizado con los nombres de usuario almacenados en una tabla de base de datos de usuarios registrados. Si el nombre de usuario no se encuentra en la tabla de la base de datos, el comportamiento de servidor lleva a cabo la inserción del registro con normalidad. Si el nombre de usuario ya existe, el comportamiento de servidor cancela la operación de inserción del registro y abre una nueva página en la que, generalmente, se indica al usuario que el nombre de usuario introducido ya está reservado.

Para comprobar que el nombre de usuario elegido es exclusivo:

En el panel Comportamientos del servidor (Ventana > Comportamientos del servidor), haga clic en el botón de signo más (+) y seleccione Autenticación de usuarios > Comprobar nuevo nombre de usuario en el menú emergente.

Aparecerá el cuadro de diálogo Comprobar nuevo nombre de usuario.



Fig. 5.37 Comprobar el nombre del usuario exclusivo

Seleccione las opciones deseadas en el cuadro de diálogo.

Para obtener instrucciones, consulte Configuración de las opciones del cuadro de diálogo Comprobar nuevo nombre de usuario. Haga clic en Aceptar.

Dreamweaver añade a la página de registro un comportamiento de servidor que comprueba que el nombre de usuario enviado por un visitante es exclusivo antes de añadir la información de ese visitante a la base de datos de usuarios registrados.

Creación de una página a la que sólo pueden acceder los usuarios autorizados

Una aplicación Web puede contener una página protegida a la que sólo pueden acceder los usuarios autorizados.

Nota: Dreamweaver no dispone de comportamientos de servidor de autenticación para páginas ASP.NET o PHP.

Por ejemplo, si un usuario intenta omitir la página de conexión escribiendo en el navegador el URL de la página protegida, es posible remitirle a otra página. De manera similar, si define el nivel de autorización de una página como administrador, sólo podrán acceder a ella los usuarios con privilegios de acceso de administrador. Si un usuario conectado intenta acceder a la página protegida sin los privilegios de acceso necesarios, se le remite a otra página.

También puede utilizar niveles de autorización para evaluar a los usuarios recién registrados antes de concederles acceso total al sitio. Por ejemplo, puede que desee recibir un pago antes de permitir que un usuario acceda a las páginas de miembros del sitio. Para ello, puede proteger las páginas para miembros con un nivel de autorización de miembro y conceder sólo privilegios de invitado a los usuarios recién registrados. Tras recibir el pago, puede

actualizar los privilegios de acceso del usuario a los de miembro (en la tabla de la base de datos de usuarios registrados).

Si no tiene previsto utilizar niveles de autorización, puede proteger cualquiera de las páginas del sitio añadiendo a la misma un comportamiento de servidor Restringir acceso a la página. El comportamiento de servidor envía a otra página a cualquier usuario que no se haya conectado correctamente.

Si tiene la intención de utilizar niveles de autorización, puede proteger cualquiera de las páginas del sitio por medio de los siguientes elementos:

- Una columna adicional en la tabla de la base de datos de usuarios para almacenar los privilegios de acceso de cada usuario
- Un comportamiento de servidor Restringir acceso a la página para enviar a los usuarios no autorizados a otra página

En este caso, el comportamiento de servidor remite al usuario a otra página que carece de los privilegios de acceso necesarios.

En ambos casos, puede añadir un vínculo a la página protegida para que el usuario pueda desconectarse y para borrar las variables de sesión. Para obtener más información, consulte Desconexión de usuarios.

Redireccionamiento de usuarios no autorizados a otra página

Para impedir el acceso de los usuarios no autorizados a una página, deberá añadir a la misma un comportamiento de servidor Restringir acceso a la página. Este comportamiento de servidor remite a otra página a todo usuario que intente omitir la página de conexión escribiendo directamente el URL de una página protegida en un navegador o que, estando conectado, intente acceder a una página protegida sin contar con los privilegios de acceso necesarios.

Nota: El comportamiento de servidor Restringir acceso a la página sólo permite proteger páginas HTML. No protege otros recursos del sitio, como los archivos de imagen o de audio, por ejemplo.

Si desea asignar los mismos derechos de acceso a varias páginas del sitio, puede copiarlos y pegarlos de una página a otra.

Para remitir a otra página a los usuarios no autorizados:

Abra la página que desea proteger. En el panel Comportamientos del servidor (Ventana > Comportamientos del servidor), haga clic en el botón de signo más (+) y seleccione Autenticación de usuarios > Restringir acceso a la página en el menú emergente. Aparecerá el cuadro de diálogo Restringir acceso a la página

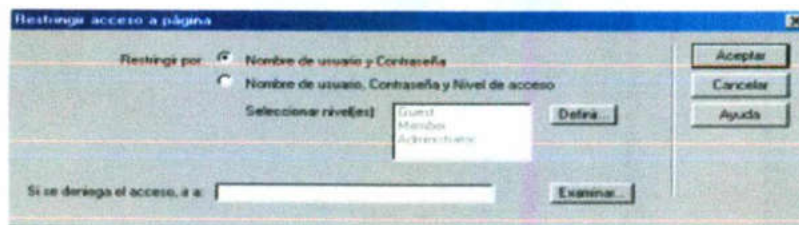


Fig. 5.38 No aceptar usuarios no autorizados

Seleccione las opciones deseadas en el cuadro de diálogo. Para obtener instrucciones, consulte Configuración de las opciones del cuadro de diálogo Restringir acceso a la página. Haga clic en Aceptar.

Dreamweaver añade un comportamiento de servidor a la página que sólo permite ver la página a los usuarios autorizados.

Para copiar y pegar los derechos de acceso de una página a otras páginas del sitio:

Abra la página protegida y seleccione el comportamiento de servidor Restringir acceso a la página que aparece en el panel Comportamientos del servidor (no el del menú emergente del signo más). Haga clic en el botón de flecha de la esquina superior derecha del panel y elija Copiar en el menú emergente.

El comportamiento de servidor Restringir acceso a la página se copia al Portapapeles del sistema.

Abra otra página que desee proteger del mismo modo. En el panel Comportamientos del servidor (Ventana > Comportamientos del servidor), haga clic en el botón de flecha situado en la esquina superior derecha y elija Pegar en el menú emergente. Repita en cada página que desee proteger

Almacenamiento de privilegios de acceso en la base de datos de usuarios

Este elemento sólo es necesario si desea que los usuarios conectados puedan tener distintos privilegios de acceso. Si sólo desea que los usuarios se conecten, no necesita almacenar los privilegios de acceso.

Si desea que los usuarios conectados tengan distintos privilegios de acceso, asegúrese de que la tabla de la base de datos de usuario contiene una columna en la que se especifican los privilegios de acceso de cada usuario (invitado, usuario, administrador, etc.). El administrador del sitio deberá introducir en la base de datos los privilegios de acceso de cada usuario.

La mayoría de las aplicaciones de base de datos permiten establecer el valor predeterminado de una columna cada vez que se crea un nuevo registro. Defina el privilegio de acceso más habitual como valor predeterminado del sitio

(por ejemplo, invitado) y cambie manualmente los casos que sean una excepción (por ejemplo, cambiar de invitado a administrador). Ahora el usuario tiene acceso a todas las páginas de administrador.

Asegúrese de que todos los usuarios de la base de datos tienen un solo privilegio de acceso (como invitado o administrador) y no varios (como "Usuario, Administrador"). Si desea establecer varios privilegios de acceso a las páginas (por ejemplo, para que todos los invitados y administradores puedan ver una página), establézcalos a nivel de la página, no de la base de datos.

Conclusión

Capítulo I

En este capítulo, hablo de la manera que se puede crear las cadenas de conexiones para poder acceder a los datos como lo permiten los DSN, existen diferentes motores de bases de datos en los que podemos hacer estas conexiones. Está por ejemplo access, que es un motor de base de datos de los mas pequeños pero muy util para aplicaciones que no requieran de muchos usuarios, otro motor de base de datos util para crear estas cadenas de conexión es SQL server el cual en tamaño es mas grande que access, soporta mayor niveles de información y permite mas fácilmente el manejo de usuarios conectados en red. Finalmente hablamos del motor de base de datos oracle este es de los mas grande, permite almacenar mayor niveles de información y conectar varios usuarios a la vez, sin problema alguno. Y por lo mismo las cadenas de conexión son mas fácil de hacer. Los actives data object (ADO) son una tecnología de fácil uso que permite acceder a las bases de datos en las paginas web.

Capítulo II

En este capítulo hablo de las técnicas que existen para poder programar una pagina web y poder administrar la información que se tiene en la base de datos. Como sabemos para la creación de una pagina web hasta el momento, es como hacer las cadenas de conexión para tener el acceso en alguna base de datos, dentro de este capítulo es bueno leerlo bien por que se explican algunas técnicas que son útiles, como la manera en que se declaran los bucles if, do while/loop para una pagina ASP. Aunque uno de los controles mas importantes al hablara de paginas ASP es el objeto response ya que este es el que nos permite mandar los datos entre variables. Y el objeto request el cual permite devolver información del usuario que ha sido enviada por los formularios.

Capítulo III

En esta sección hablo un poco sobre como es que se hace la configuración del sitio web. Desde que es lo que se necesita ser instalado en la computadora (IIS) cómo se define un directorio virtual y para que nos sirve este. Un poco también sobre seguridad y como podemos evitar que los usuarios que no pertenezcan a la organización no podrán entrar sin una clave de acceso que el administrador provee. Esta es una medida de seguridad importante para cualquier pagina. Otra de las aplicaciones que son muy útiles y que toda aquella persona que administra un web o alguna red tiene que conocer es el FTP. La cual nos permite almacenar archivos con un tamaño considerable y descargarlos en otro sitio virtual. Al configurar un servidor web es necesario hablar sobre el servicio que puede proporcionar con el SMTP.

Capítulo IV

En este capítulo, veremos la manera en la que el programa dreamweaver ayuda a la realización de las páginas ASP, desde cómo hacer conexiones DSN, hasta el desarrollo de la misma página con interfaz parecida a flash en las que podremos mostrar páginas maestro detalle, cómo hacer las páginas de búsqueda y modificaciones de una manera mas sencilla.

Bibliografía

- Jesús Bobadilla. Alejandro Alcocer
Active server pages 3 creación de aplicaciones web a traves de ejemplos
Alfaomega Ra-ma
Edición original
Año 2000 Madrid España.
- Scot Millier. Daniel mezick
Programación de Active server pages
Mc Graw Hill
Respecto a la primera edición en español
Año 1998 españa.