

¿Qué ofrece Autentia Real Business Solutions S.L?

Somos su empresa de **Soporte a Desarrollo Informático**.
 Ese apoyo que siempre quiso tener...

1. Desarrollo de componentes y proyectos a medida



2. Auditoría de código y recomendaciones de mejora

3. Arranque de proyectos basados en nuevas tecnologías

1. Definición de frameworks corporativos.
2. Transferencia de conocimiento de nuevas arquitecturas.
3. Soporte al arranque de proyectos.
4. Auditoría preventiva periódica de calidad.
5. Revisión previa a la certificación de proyectos.
6. Extensión de capacidad de equipos de calidad.
7. Identificación de problemas en producción.



4. Cursos de formación (impartidos por desarrolladores en activo)

Spring MVC, JSF-PrimeFaces /RichFaces,
 HTML5, CSS3, JavaScript-jQuery

Gestor portales (Liferay)
 Gestor de contenidos (Alfresco)
 Aplicaciones híbridas

Tareas programadas (Quartz)
 Gestor documental (Alfresco)
 Inversión de control (Spring)

Control de autenticación y
 acceso (Spring Security)
 UDDI
 Web Services
 Rest Services
 Social SSO
 SSO (Cas)

JPA-Hibernate, MyBatis
 Motor de búsqueda empresarial (Solr)
 ETL (Talend)

Dirección de Proyectos Informáticos.
 Metodologías ágiles
 Patrones de diseño
 TDD

BPM (jBPM o Bonita)
 Generación de informes (JasperReport)
 ESB (Open ESB)



Tutorial desarrollado por:
[Ismael Caballero, 2004.](#)

Contacto: icaballero@carixma.com



Descargar este documento en formato PDF [clasesAsp.pdf](#)

[Firma en nuestro libro de Visitas](#)

[Master Java J2ee Oracle](#)

100% alumnos ya trabajan. Nuevo temario de Struts + J2ME.

[WebSphere Solved](#)

From J2EE to Web Services, we demystify the middle tier for you

[WebSphere Plugin](#)

visual rules: Plugin for fast Development of Business Rules

[Dirección de Empresas](#)

Estudie su MBA en una escuela de prestigio internacional en España

Creación y utilización de clases en páginas ASP

1- INTRODUCCIÓN

En el presente tutorial se explicará la **implementación de una clase utilizando la tecnología ASP (Active Server Pages) de Microsoft**, así como la manera de **instanciar y aplicar sus métodos en páginas ASP**.

Las páginas Active Server Pages (ASP) es una tecnología de [Microsoft](#) que precisa de un servidor, para ejecutar páginas web generadas dinámicamente. Estas páginas web suelen estar asociadas al servidor [Internet Information Server](#) (IIS).

Las páginas se suelen construir mezclando código de [scripts del lado del servidor](#) (incluyendo acceso a base de datos) con [HTML](#). Por ejemplo:

```
<% if request("Variable1") = 1 then %>
    <b>Variable 1 es igual a uno</b>
<% else %>
    <b>Variable 1 es distinta a uno</b>
<%end if%>
```

Para este tutorial, crearemos y utilizaremos una clase que servirá para registrar las acciones realizadas, a modo de "log" de eventos, simulando un control de acciones de un **call center** de una empresa para atender a sus clientes. Para ello utilizaremos una base de datos MS Access para guardar los registros, la funcionalidad de acceso a la base de datos e inserción de datos se implementará en la clase generada.

También mostraremos como instanciar la clase creada y como utilizar sus métodos, desde páginas asp de un proyecto web.

Utilizando clases en nuestros proyectos y desarrollos, encapsularemos las funcionalidades más utilizadas o importantes de nuestros proyectos, permitiéndonos reutilizar múltiples métodos, ahorrándonos tiempo de codificación y de depuración, obteniendo soluciones más robustas, fiables y seguras.

Manos a la obra....

2- Definición de la base de datos utilizada en este proyecto

Como he comentado anteriormente, para el ejemplo de este tutorial, vamos a utilizar una base de datos MS Access, que denominaremos **dbClases.mdb**, que crearemos en la carpeta "bd", dentro de la carpeta de nuestro proyecto web.

Una vez creada la base de datos en blanco, el primer paso es darle permisos de lectura y escritura, para que se puedan consultar e insertar registros. Para ello utilizaremos el sistema de administración de nuestro sitio web, que suelen incluir en los planes de hosting o bien directamente, seleccionando con el botón derecho del ratón la carpeta "bd" seleccionaremos la opción de "compartir y seguridad...", activando las opciones de compartir y de permitir modificar los datos por los usuarios de la red.

A continuación crearemos las tablas que vamos a utilizar:

- TAcciones
- TClientes
- TProductos
- TTipoAcciones
- TAlertas

En las siguientes imágenes se presentan los campos de cada una de ellas y las relaciones finales existentes entre ellas:

TAcciones : Tabla						
	IdAccion	IdTipoAccion	IdCliente	IdProducto	DescripcionAccion	FechaAccion
	1	3	1	1	aaa	09/08/2006
	2	2	2	2	bbb	09/08/2006
	3	4	3	3	ccc	09/08/2006
▶	(Autonumérico)					

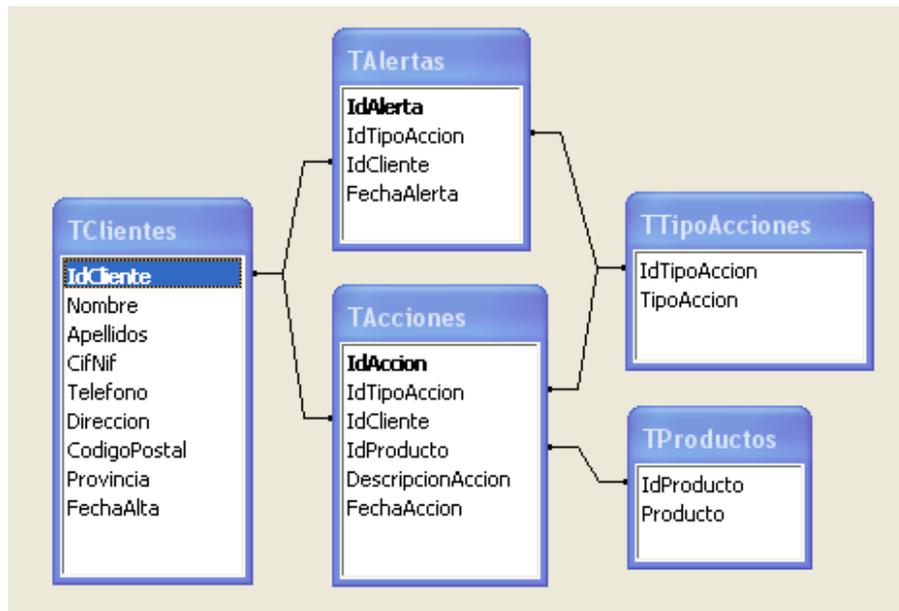
TClientes : Tabla									
	IdCliente	Nombre	Apellidos	CiNif	Telefono	Direccion	CodigoPostal	Provincia	FechaAlta
+	1	Ismael	Caballero	11111111	111111111	c/carixma 1	11111	Madrid	07/08/2006
+	2	a	a	11111112	111111112	c/carixma 2	11111	Madrid	07/08/2006
+	3	b	b	11111113	111111113	c/carixma 3	11111	Madrid	07/08/2006
+	4	c	c	11111114	111111114	c/carixma 4	11111	Madrid	07/08/2006
▶	(Autonumérico)								

TProductos : Tabla	
IdProducto	Producto
1	Producto tipo 1
2	Producto tipo 2
3	Producto tipo 3
4	Producto tipo 4
5	Producto tipo 5
6	Producto tipo 6
▶	

TTipoAcciones : Tabla	
IdTipoAccion	TipoAccion
1	Alta de cliente
2	Actualización de datos personales
3	Actualización de datos bancarios
4	Baja de cliente
5	Contratación de un producto
6	Cambio de producto
7	Cancelación de producto
8	Renovación de producto
9	Consulta
10	Sugerencia
11	Reclamación
▶	

TAlertas : Tabla				
	IdAlerta	IdTipoAccion	IdCliente	FechaAlerta
✎	1	3	3	09/08/2006
*	(Autonumérico)	0	0	

Relaciones:



Para finalizar, nos faltaría rellenar las tablas de TTipoAcciones, TProductos y TCientes con los valores iniciales.

3- Desarrollo del proyecto

En este apartado mostraremos los códigos de nuestras páginas ASP y de la clase creada, así como las explicaciones a los aspectos más relevantes de cada una de ellas.

Página de la Clase: clase_log.asp

Esta será la página que la se encargue de registrar las acciones realizadas por los clientes, así como de controlar los casos especiales, por ejemplo, en los casos de baja del cliente o de cancelación de algún producto, registrar una "alerta" para que sea gestionada por el departamento de bajas o comercial de la empresa, todo este proceso será "transparente" al operario del call center, que sólo se encargará de registra la acción realizada por el cliente desde el formulario de registro de acciones (**formulario.asp**)

El código de la clase es el siguiente:

```

<%
Class Acciones
private Conexion

'Constructor de la conexión
Public Sub Init_Connection()
'Conexión a base de datos Access
basedatos = Server.MapPath("bd/bdClases.mdb")
set Conexion = Server.CreateObject("ADODB.Connection")
conexion.Open "Provider=Microsoft.Jet.OLEDB.4.0;Data Source=" & basedatos & ";"

End Sub

'Destructor de la conexión
Public Sub Close_Connection()
on error resume next
Conexion.Close
End Sub

'Método para registrar la acción
Public Function Register_Action (IdAction, IdCliente, IdProducto, sDescription)
Init_Connection

'Comprobamos el tipo de acción para añadir una alerta para el dpto. comercial o de bajas
if (cLng(IdAction) = 4) or (cLng(IdAction) = 7) then 'Baja de cliente o cancelación de producto
sQuery = "insert into TAlertas (IdTipoAccion, IdCliente, FechaAlerta) values ("
sQuery = sQuery & ", " & IdCliente & ", " & date() & ")"
'Ejecutamos la sentencia SQL correspondiente
Conexion.Execute sQuery
end if
  
```

```
'Registramos la acción en la base de datos
sQuery = "insert into TAcciones (IdTipoAccion, IdCliente, IdProducto, DescripcionAccion, FechaAccion) values ("
sQuery = sQuery & IdAction & ", " & IdCliente & ", " & IdProducto & ", " & sDescription & ", " & date() & """)"
'Ejecutamos la sentencia SQL correspondiente
Conexion.Execute sQuery

'Devolvemos el código de error
Register_Action = err.number

Close_Connection
End Function
End Class
%>
```

Para definir una clase, debemos comenzar por escribir al inicio de la página:

Class Nombre_de_la_clase en este caso: Class Acciones

Y finalizar la página con **End Class**

En la parte interior irán los métodos y funcionalidades correspondientes a cada parte del proyecto.

Cabe destacar el código de conexión a base de datos, donde se indica la ruta donde se encuentra la base de datos "dbClases.mdb".

Los métodos de **Init_Connection ()** y **Close_Connection ()** utilizados para instanciar y abrir la conexión a la base de datos y para cerrarla y destruirla.

También encontramos la función **Register_Action** que recibe los parámetros del formulario, los gestiona y los inserta en las tablas correspondientes.

Esta función devuelve el número de error, que se comprobará en la página registrar.asp para conocer si los datos de la acción han sido procesados correctamente o ha existido algún error y no se han podido procesar de manera satisfactoria.

Se ha incluido un sistema de comprobación de la acción, para consultar si son acciones críticas. de baja del cliente o del producto, para avisar al departamento comercial o de bajas.

Página del formulario de acciones: formulario.asp

Esta página presentará los campos para registrar las acciones de los clientes, pudiendo seleccionar el tipo de acción que se desea realizar, el cliente el producto y una descripción de la acción realizada, una vez rellenados todos los campos existe un botón de "Enviar" que enviará los datos a la página de registrar.asp para proceder a guardar estos datos en las tablas de nuestra base de datos.

Este formulario, sería la principal herramienta de trabajo de nuestros operarios del call center.

El código de esta página es el siguiente:

Nota: Para este ejemplo no se han incluido sistemas de validación de los campos.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
<head>
<title>Formulario Ejemplo Clases con ASP</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
</head>
<body>
<%
'Conexión a base de datos Access
basedatos = Server.MapPath("bd/bdClases.mdb")
Set conexion = Server.CreateObject("ADODB.Connection")
conexion.Open "Provider=Microsoft.Jet.OLEDB.4.0;Data Source=" & basedatos & ";"

sQuery = "select * from TCientes order by Nombre"
set MiRS=Server.CreateObject("ADODB.RecordSet")
MiRS.Open sQuery, Conexion, 3

sQuery = "select * from TTipoAcciones order by TipoAccion"
set MiRS2=Server.CreateObject("ADODB.RecordSet")
MiRS2.Open sQuery, Conexion, 3

sQuery = "select * from TProductos order by Producto"
set MiRS3=Server.CreateObject("ADODB.RecordSet")
MiRS3.Open sQuery, Conexion, 3
%>
```

```

<br>
<table width="600" border="0" cellpadding="2" cellspacing="0">
<form name="frm" method="post" action="registrar.asp">
<tr><td colspan="2" align="center"><b>FORMULARIO DE REGISTRO DE ACCIONES</b><br><br></td></tr>
<tr><td colspan="2" align="left">Por favor rellene los siguientes datos para registrar la acción del
cliente:<br><br></td></tr>
<tr>
<td width="140" align="left"><b>Tipos de Acción:</b></td>
<td>
<select name="IdAccion">
<% do while not MiRS2.eof %>
<option value="<%= MiRS2("IdTipoAccion") %>"><%= MiRS2("TipoAccion") %></option>
<%
MiRS2.Movenext
Loop
%>
</select>
</td>
</tr>
<tr>
<td width="140"><b>Clientes:</b></td>
<td>
<select name="IdCliente">
<% do while not MiRS.eof %>
<option value="<%= MiRS("IdCliente") %>"><%= MiRS("Apellidos") %>, <%= MiRS("Nombre") %></option>
<%
MiRS.Movenext
Loop
%>
</select>
</td>
</tr>
<tr>
<td width="140"><b>Productos:</b></td>
<td>
<select name="IdProducto">
<% do while not MiRS3.eof %>
<option value="<%= MiRS3("IdProducto") %>"><%= MiRS3("Producto") %></option>
<%
MiRS3.Movenext
Loop
%>
</select>
</td>
</tr>
<tr>
<td valign="top"><b>Descripción de la Acción:</b></td>
<td><textarea name="Descripcion" cols="35" rows="5"></textarea></td>
</tr>
<tr><td colspan="2">&nbsp;</td></tr>
<tr><td colspan="2" align="center"><input type="submit" name="registrar" value="Registrar
Acción"><br></td></tr>
</form>
</table>
<%
MiRS.Close
MiRS2.Close
Conexion.Close
Set MiRS = Nothing
Set MiRS2 = Nothing
Set Conexion = Nothing
%>
</body>
</html>

```

Página de registro de acciones: registrar.asp

Esta página nos servirá para instanciar la clase asp, que se encargará de gestionar y guardar los datos de cada acción, que devolverá un código de error que comprobaremos para Mostrar al usuario si el procesamiento de la información ha sido correcto o ha existido algún problema.

El código de esta página es el siguiente:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
```

```

<html>
<head>
<title>Formulario Ejemplo Clases con ASP</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
</head>
<body>
<!-- #include file="clase_log.asp" -->
<%
'Registramos la acción realizada en el log de acciones
set objAcciones = new Acciones
'Los parámetros a enviar son: Identificador del tipo de acción, Identificador del Cliente, Identificador del Producto y
Descripción de la acción realizada
sControl = objAcciones.Register_Action (request("IdAccion"), request("IdCliente"), request("IdProducto"),
trim(replace(request("Descripcion"),"", "")))
set objAcciones = Nothing

if err.number = 0 then
%>
<table width="600" border="0" cellspacing="2" cellpadding="0">
<tr><td><b>REGISTRO DE ACCIONES</b></td></tr>
<tr><td><strong>Su acción ha sido registrada correctamente.</strong></td></tr>
</table>
<%
else
%>
<table width="600" border="0" cellspacing="2" cellpadding="0">
<tr><td><b>REGISTRO DE ACCIONES</b></td></tr>
<tr><td><strong>Su acción no ha podido ser registrada correctamente, por favor, compruebe la información e
inténtelo de nuevo.</strong></td></tr>
</table>
<%
end if
%>
</body>
</html>

```

Cabe destacar el modo de instanciar la clase y de llamar a la función Register_Action:

Antes de crear el objeto objAcciones, debemos incluir la clase, para ello utilizamos un include al archivo:

```

<!-- #include file="clase_log.asp" -->

```

A continuación creamos el objeto y hacemos la llamada a la función:

```

<%
'Registramos la acción realizada en el log de acciones
set objAcciones = new Acciones
'Los parámetros a enviar son: Identificador del tipo de acción, Identificador del Cliente, Identificador del Producto y
Descripción de la acción realizada
sControl = objAcciones.Register_Action (request("IdAccion"), request("IdCliente"), request("IdProducto"),
trim(replace(request("Descripcion"),"", "")))
set objAcciones = Nothing
%>

```

4- CONCLUSIÓN

En este tutorial se ha desarrollado una aplicación utilizando páginas ASP, en ellas se ha creado una clase y se ha mostrado la forma para instanciarla y utilizarla en otra página web, para registrar acciones, mejorando nuestra estructura del proyecto y el mantenimiento de éste.

Utilizando un sistema de clases se desarrolla con una estructura de 3 capas:

1. **Capa de Datos:** Correspondiente a la base de datos MS Access.
2. **Capa de Negocio:** Correspondiente a la clase desarrollada, que contendrá los métodos y funciones más importantes de la aplicación.
3. **Capa de Presentación:** Correspondiente a las páginas web que instancian y utilizan los métodos de la clase.

Por último, en este tutorial se ha puesto la mejor intención para presentar la creación y utilización de clases en páginas ASP, pero puedo haberme equivocado en algo, si alguien desea aportar datos, o colaborar, o corregirme. Estaré encantado de esperar su respuesta en icaballero@carixma.com







[Puedes opinar sobre este tutorial aquí](#)

Recuerda

que el personal de [Autentia](#) te regala la mayoría del conocimiento aquí compartido ([Ver todos los tutoriales](#))

¿Nos vas a tener en cuenta cuando necesites consultoría o formación en tu empresa?

¿Vas a ser tan generoso con nosotros como lo tratamos de ser con vosotros?

info@autentia.com

Somos pocos, somos buenos, estamos motivados y nos gusta lo que hacemos

Autentia = Soporte a Desarrollo & Formación

[Autentia S.L.](#) Somos expertos en:
J2EE, Struts, JSF, C++, OOP, UML, UP, Patrones de diseño ..
y muchas otras cosas

Nuevo servicio de notificaciones

Si deseas que te enviemos un correo electrónico cuando introduzcamos nuevos tutoriales, inserta tu dirección de correo en el siguiente formulario.

Subscribirse a Novedades	
e-mail	<input type="text"/>
<input type="button" value="Enviar"/>	

Otros Tutoriales Recomendados ([También ver todos](#))

Nombre Corto

[Mailings con asp y Base de Datos](#)

[AdRotator en Página ASP .Net](#)

[Mailings con ASP y ficheros](#)

[Patrones de GRASP](#)

Descripción

Ismael Caballero describe el proceso de realización de un sistema para el envío de correos (mailings) de forma automática utilizando un sistema de base de datos (Microsoft SQL Server).

Ismael Caballero nos cuenta como utilizar el control AdRotator en una página ASP .Net

Ismael Caballero nos muestra una variación de su tutorial para enviar mailings con tecnología asp y ficheros (sin necesidad de base de datos)

Os presentamos una introducción a los patrones de asignación de responsabilidades y su relación con el proceso unificado.

Nota: Los tutoriales mostrados en este Web tienen como objetivo la difusión del conocimiento.

Los contenidos y comentarios de los tutoriales son responsabilidad de sus respectivos autores.

En algún caso se puede hacer referencia a marcas o nombres cuya propiedad y derechos es de sus respectivos dueños. Si algún afectado desea que incorporemos alguna reseña específica, no tiene más que solicitarlo.

Si alguien encuentra algún problema con la información publicada en este Web, rogamos que informe al administrador rcanales@adictosaltrabajo.com para su resolución.

[Patrocinados por enredados.com Hosting en Castellano con soporte Java/J2EE](#)



www.AdictosAlTrabajo.com Optimizado 800X600