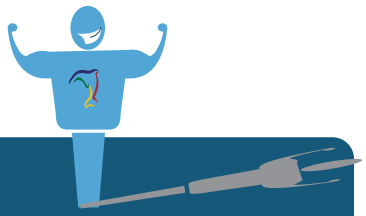
autentia

Avenida de Castilla,1 - Edificio Best Point - Oficina 21B
28830 San Fernando de Henares (Madrid)
tel./fax: +34 91 675 33 06
info@autentia.com - **www.autentia.com**

¿**Qué ofrece** Autentia Real Business Solutions S.L?

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

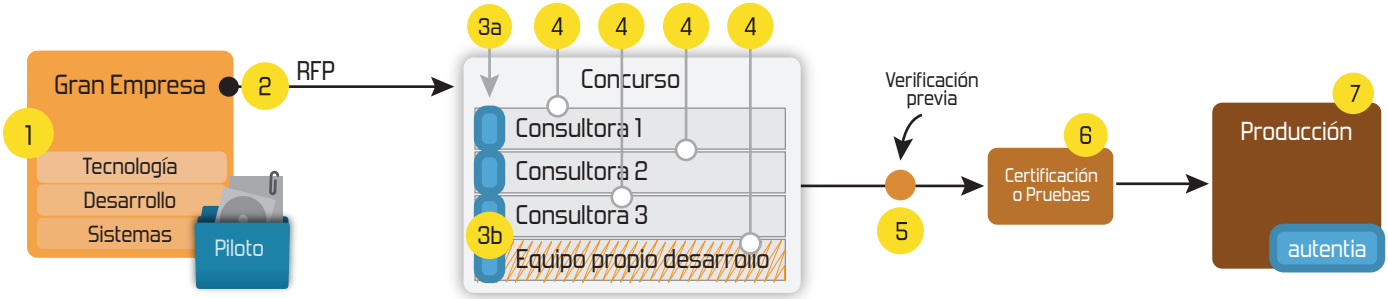
1. Desarrollo de componentes y proyectos a medida



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
- 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
- Tareas programadas (Quartz)
- Gestor documental (Alfresco)
- Inversión de control (Spring)
- BPM (jBPM o Bonita)
- Generación de informes (JasperReport)
- ESB (Open ESB)

Compartimos nuestro conocimiento en:
www.adictosaltrabajo.com

Para más información visítenos en:
www.autentia.com

Hosting Patrocinado por enREDados.com

Tutorial desarrollado por: Luis Paniagua

neva-es.com

Descargar este documento en formato PDF authforms.pdf

Curso Web J2EE
Curso Avanzado en Desarrollo Web con J2EE
www.eps.mondragon.edu/caj2ee

Certificados SSL VeriSign
Servidor seguro con SSL de 128 bits
Descargar ahora la guía gratis.
www.verisign.es

Servidores Linux
Servidores rack o pedestal, hasta 4 procesadores Xeon o Itanium2
www.caton.es

Anuncios Goooooogle Anunciarse en este sitio

Autenticación basada en formularios (IIS y .Net).

1. - Introducción.

En este manual vamos a crear una zona privada para nuestra web alojada en un servidor IIS y con el .Net FrameWork instalado, para con unas líneas en ASP .Net validemos los datos insertados por los usuarios en el correspondiente formulario de acceso.

Resulta útil, sobre todo cuando no tenemos acceso a la configuración de usuarios y privacidad de las carpetas en el disco duro del servidor en el que se aloja nuestra web, y queremos hacer una sección privada en la que restringir el acceso a los usuarios anónimos que navegan por nuestro site, mediante la solicitud y validación de login y password. También puede utilizarse para que las herramientas de administración de nuestro site estén protegidas contra el acceso de usuarios ajenos a la administración.

Se explicarán la configuración del web.config de la aplicación y del directorio privado y dos formas para validar el acceso de los usuarios. La más sencilla con los usuarios validos configurados en el web.config. La otra, un poco más compleja, mediante una consulta a una base de datos SQL para localizar a los usuarios validos.

Los usuarios se validarán en el correspondiente formulario de autenticación, dicha autenticación utiliza cookies del navegador para comprobar, según navegamos, los permisos de acceso al directorio, subdirectorios y ficheros de la zona privada.

Una vez configurado el directorio privado no podemos acceder al contenido de este salvo que dispongamos de la autenticación adecuada almacenada en una cookie.

2.- ¿Qué necesitamos?

Para activar la autenticación de formularios básica en una aplicación, realizaremos los siguientes pasos:

- Configurar el modo de autenticación modificando la sección authentication en el web.cofig de la aplicación.
- Denegar el acceso a los usuarios anónimos en uno o más directorios de la aplicación, mediante la modificación de la sección authorization en los archivos web.config de los directorios privados
- Creara una página de registro con un formulario que permita la identificación de usuarios.
- Para la validación contra la base de datos, necesitaremos un servidor de base de datos en el que configuraremos una tabla de usuarios y un procedimiento almacenado.

Para empezar vamos a crear un directorio, PRIVADO, y dentro de este una página default que nos muestre un "HOLA MUNDO PRIVADO" para validar que vemos el directorio en nuestra web.

3.- Modificando el web.config.

Con el directorio privado que hemos creado, y desde el cual podremos colgar todas nuestras herramientas administrativas del site, como el acceso a descargas o imágenes personales, tenemos que realizar una serie de modificaciones en el web.config

En este tutorial no nos vamos a parar en explicar al detalle todas las opciones que se puede usar para la crear un acceso privado, solo nos ocuparemos de las opciones que vamos a usar.

El fichero web.config se encuentra localizado en el directorio raíz de la aplicación web con la que estamos trabajado. Al abrirlo veremos que dentro de la etiqueta system.web se encuentran diferentes opciones, las que ocuparan nuestra atención son las entradas authentication y authorization, ya que son en las que realizaremos las modificaciones.

En el caso de que no exista este archivo web.config podremos crearlo, pero siempre en el raíz de la aplicación.

3.1.- authentication

El archivo web.config contendrá el siguiente código:

```
<configuration>
```

```
 <system.web>
  <authentication mode="Forms">
   <forms name="FormsAuthDB.nevaAuth" loginUrl="login.aspx" protection="All" timeout="10" path="/">
    <credentials passwordFormat="Clear">
     <user name="adm" password="myadm"/>
    </credentials>
   </forms>
  </authentication>
 </system.web>
<configuration>
```

Como se puede observar, y parece obvio, el modo de autenticación de nuestro directorio será mediante un formulario, quedando esto reflejado en la propiedad mode="Forms"

Dentro de la sección forms se definen el funcionamiento de nuestro formulario:

- **name**: el nombre de la cookie de navegador que contiene la entrada de autenticación. Si no especificamos uno se llamará .ASPXAUTH
  Recomendamos que si usamos varias aplicaciones en nuestro servidor web se especifique un nombre por aplicación.
- **loginUrl**: indicaremos la página a la que se dirigirá al usuario, de modo automático, cuando se precise su identificación. Si no especificamos una página, por defecto se redirige a login.aspx situada en el raíz de la aplicación web.
- **timeout**: es el tiempo, en minutos, que la cookie tiene validez, siendo de 30 minutos el valor por defecto.
- **protection**: especifica la forma en que se protegen los datos de la cookie. Los valores son All, None, Encryption y Validation, siendo All el valor que se toma por defecto.
  Por omisión las cookies sé encriptan con DES o TripleDES, dependiendo de la capacidad del servidor.

El elemento credentials nos servirá para configurar los usuarios que tendrán acceso a nuestro directorio privado. Esto es útil si no tenemos un listado largo de usuarios y que no queremos o nos es necesario realizar labores de mantenimiento sobre estos.

Este elemento es bastante simple de entender ya que se creará una entrada por cada uno de los usuarios que tendrán acceso a las aplicaciones o elementos descargables que tengamos en nuestra zona privada. En estos configuraremos el nombre de usuario y la password.

Si deseamos disponer de una mayor seguridad, por si algún curioso lograra acceder al servidor y ver las password podríamos encriptarlos utilizando los algoritmos hash SHA1 y MD5, para ello tendríamos que cambiar el valor de passwordFormat por SHA1 o MD5. Pero la forma de encriptar este password y acceder a ellas lo trataremos en otra ocasión.

4.- Configurando el directorio

Bien, ya tenemos listo el sistema de autenticación, hemos creado el directorio que deseamos hacer privado y preparada una página que nos servirá de bienvenida a la zona privada.

Ahora solo nos falta hacer que este sea privado. Para ello será necesario crear un fichero web.config en el raíz de nuestro directorio privado. En este web.config configuraremos la sección authorization, donde denegaremos el acceso a los usuarios anónimos.

El contenido del web.config será el siguiente.

```
<configuration>
 <system.web>
  <authorization>
   <deny users="?"/>
  </authorization>
 </system.web>
</configuration>
```

Dentro de la sección authorization, se permite las siguientes entradas:

- **deny** para denegar usuarios
- **allow** para permitir el acceso a usuarios.

Ambas entradas tienen la propiedad users donde podemos especificar:

- ? todos los usuarios anónimos.
- * todos los usuarios, tanto anónimos como autenticados.
- Lista usuarios, se permite especificar un listado de usuarios separados por comas, que tendrán permitido o denegado el acceso según donde se especifique

5.- La página de Login.

Bien ya está todo listo, hemos configurado el directorio, le hemos dado las indicaciones a la aplicación para que nos autentique a los usuarios, hemos definido la autorización del directorio, pero no falta hacer logo, ya que sin esto no veremos nunca ese HOLA MUNDO PRIVADO que hemos preparado.

Pues manos a la obra. Tenemos que crear un formulario web, pero no voy a explicar como hacerlo, se supone que esto ya sabemos hacerlo, pero si no tenemos ganas de golpear teclas, descarga aquí el zip con todo el código.

Este formulario web tendrá los siguientes elementos:

- Una caja de textos, con su correspondiente título donde pidamos el nombre de usuario.
- Una caja de texto, y su titulo, donde pidamos la password, esta tendrá la propiedad textmode=Password.
- Un checbox, con su titulo para que el usuario marque que le recuerde la password en el futuro.
- Un boto de enviar, y opcional otro de borrar.
- Una etiqueta para mostrar errores.

Opcionalmente podemos insertar el objeto que nos valide que el usuario a insertado los datos de forma correcta en el formulario.

Ya hemos creado, pintado y dado color a nuestra pantalla de login. Todo al gusto, no me meto.

Como apuntaba al principio de este tutorial, la intención era enseñar dos métodos de validación, una por la configuración de usuarios en el web.config y otra mediante consultas a base de datos.

5.1.- Usuarios en web.config.

Ya se ha explicado donde y como se configurar, si no lo tienes claro pincha aquí. Y ahora tenemos que hacer que cuando el usuario pulse el botón enviar se nos valide y de acceso a este al directorio o nos deniegue el acceso en caso de error.

Para eso incluiremos el siguiente código en el fichero VB de nuestro formulario:

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
  'Validación con usuarios en el WebConfig.
  If IsValid Then
    If System.Web.Security.FormsAuthentication.Authenticate(tbLogin.Text, tbPassword.Text) Then
      System.Web.Security.FormsAuthentication.RedirectFromLoginPage(tbLogin.Text, chkPassword.Checked)
    Else
      lblError.Visible = True
      lblError.Text = "Nombre de usuario y/o contraseña no validos."
    End If
  End If
End Sub
```

Con IsValid comprobamos si el formulario está correcto, esto era opcional.

En la siguiente validación empieza lo duro, ya que con el método Autenticate de la clase FormsAuthentication, hacemos que se localice en el web.config el nombre de usuario y la contraseña introducidas. Devolverá un true o false si lo encuentra o no en la sección credentials.

En el caso de que la respuesta sea errónea, pintamos el correspondiente mensaje de error en la etiqueta reservada para esto.

Si hemos acertado con el usuario y la contraseña entra en juego el método RedirectFromLoginPage, al que le pasamos el usuario y el valor del check para que en el futuro nos pregunte o no por nuestra identificación. Este nos devolverá a la url que hemos intentado acceder dentro de nuestro directorio privado antes de que se nos mostrara la página de login.

5.2.- Usuarios en base de datos.

Ya hemos vista la forma simple de hacer login con nuestros usuarios en el web.config. Ahora lo podemos complicar un poco más utilizando la base de datos, si es que tenemos disposición de hacerlo.

Supongamos que tenemos un SQL Server instalado y que tenemos acceso a el. Tendremos que configurar una tabla, donde tengamos un campo login y otro password como mínimo, el resto de campos a elección. No explicaremos como crear la tabla y el procedimiento almacenado, que reservado para otro tutorial. Si queréis evitar crearlo manualmente, pincha aquí y dentro del zip encontrarás el Script SQL que genera la tabla con esta información mínima y un procedimiento almacenado para buscar el usuario.

Vamos a usar la mismas página aspx que en el ejemplo anterior, pero en esta ocasión añadimos un botón más que nos servirá para llamar a una función diferente.

En esta ocasión necesitaremos importar las siguientes clases para el manejo de base de datos:

```
Imports System.data
Imports System.Data.SqlClient
```

Tras pulsar el botón de validar en la Base de datos, se ejecutará el siguiente código:

```
Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
  'Validación con usuarios en Base de datos.
  If IsValid Then
    If DBAuthenticate(tbLogin.Text, tbPassword.Text) > 0 Then
      System.Web.Security.FormsAuthentication.RedirectFromLoginPage(tbLogin.Text, chkPassword.Checked)
    End If
  End If
End Sub
```

Como se puede observar este código es similar al anterior. IsValid nos valida el que el formulario sea correcto

En la siguiente comprobación se hace una llamada a la función, que tendremos que incluir, para realizar la validación contra la base de datos.

La función queda como sigue, se ha de prestar atención a los comentarios para comprender que hace el código:

```
Function DBAuthenticate(ByVal strUsername As String, ByVal strPassword As String) As Integer
  Dim conMyData As SqlConnection
  Dim cmdSelect As SqlCommand
  Dim parmReturnValue As SqlParameter
  Dim intresult As Integer

  'Conexión a la base de datos
  conMyData = New SqlConnection(ConfigurationSettings.AppSettings("nevaCnnString"))
  'Preparamos la llamada al Procedimiento Almacenado
  cmdSelect = New SqlCommand("DBAuthenticate", conMyData)
```

```
  cmdSelect.CommandType = CommandType.StoredProcedure

  parmReturnValue = cmdSelect.Parameters.Add("RETURN_VALUE", SqlDbType.Int)
  parmReturnValue.Direction = ParameterDirection.ReturnValue
  'Asignamos valores a los parámetros del procedimiento almacenado.
  cmdSelect.Parameters.Add("@username", strUsername)
  cmdSelect.Parameters.Add("@password", strPassword)

  Try 'Control de errores
    conMyData.Open()
    'Ejecutamos el Procedimiento almacenado
    cmdSelect.ExecuteNonQuery()
    'Recogemos el resultado del Procedimiento almacenado.
    intresult = cmdSelect.Parameters("RETURN_VALUE").Value
    'Cerramos conexión
    conMyData.Close()

    If intresult < 0 Then 'Se ha producido error
      lblError.Visible = True
      If intresult = -1 Then 'Nombre de usuario no valido.
        lblError.Text = "Nombre de usuario no registrado."
      Else 'intResult = -2 Error en la password
        lblError.Text = "Password incorrecta."
      End If
    End If
    Return intresult
  Catch ex As Exception 'Se produce error en el uso del procedimiento
    lblError.Visible = True
    lblError.Text = "Error al intentar identificar al usuario: "
    lblError.Text &= ex.Message
  End Try
End Function
```

Sin entrar en detalles de manejo de base de datos, la función prepara el acceso a la base de datos, y el uso del procedimiento almacenado, le asigna los valores nombre de usuario y contraseña a los parámetros del procedimiento y lo ejecuta.

El procedimiento almacenado se encarga de localizar en la tabla en nombre de usuario, y comprobar la password, en caso de ser correctos devuelve el ID correspondiente en la tabla al registro. Este procedimiento controla dos posibles errores, es simple, si no localiza el nombre de usuario devolverá el error -1 y si encuentra el nombre de usuario pero la contraseña no coincide devuelve un -2.

Esta misma función toma como valor el resultado del procedimiento almacenado por lo que al evaluarla desde la pulsación del botón, solamente en caso de que el resultado sea mayor a 0 se redirigirá al usuario a la página o directorio que solicitó, de igual modo que sucedía con la validación del usuario configurado en el web.config, mediante el método RedirectFromLoginPage, que espera recibir nuevamente el nombre de usuario y y el resultado del check que nos evitará la inserción de estos datos en el futuro mediante el control de la cookie.

6.- La prueba.

Si todo está bien y se han seguido los pasos indicados, nuestro directorio estará protegido contra usuarios anónimos. De modo que si llamamos al directorio privado desde le navegador, http://miservidor/privado nos responderá la página de login que indicamos en el web.config, y tras identificarnos correctamente accederemos a la página que habíamos solicitado dentro del directorio privado.

Con esto ya se puede configurar diferentes herramientas administrativas que gestionen el site, o un lugar donde guardes tus fotos personales y que solo quieres que algunos tenga acceso.

7.- Notas.

En la validación por base de datos, no se ha mencionado ya que ase asume que sabemos acceder a base de datos, pero es necesario configurar en el web.config del raíz de la aplicación la cadena de conexión a la base de datos. Esta cadena de conexión es usada en la línea

conMyData = New SqlConnection(ConfigurationSettings.AppSettings("nevaCnnString"))

¿Cómo decirse entre configurar los usuarios en el Web.config o en la base de datos? Todo dependerá del número de usuarios que se tiene previsto mantener, si queremos mantener de 3 a 10 usuarios, nos decantaremos por la solución web.config, pero si el número es mayor y encima se permitirá el crecimiento de usuarios la mejor solución es la base de datos

---

Si desea contratar formación, consultoria o desarrollo de piezas a medida puede contactar con

Creatividad Internet

Autentia S.L. Somos expertos en:
J2EE, C++, OOP, UML, Vignette, Creatividad ..
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	
	Enviar

Otros Tutoriales Recomendados (También ver todos)

Nombre Corto	Descripción
AdRotator en Página ASP .Net	Ismael Caballero nos cuenta como utilizar el control AdRotator en una página ASP .Net
Aplicaciones con el framework de Microsft .NET	Ejemplo de desarrollo de una aplicación con el framework de Microsft .NET (creación de un servicio deEncuestas Web)
Ado y Net	En este tutorial podeis descubrir como funciona ADO y las nuevas diferencias en arquitectura .Net .
WebDav con IIS	En este tutorial podemos aprender como gestionar los contenidos de nuestro servidor Web sin necesidad de utilizar FTP.
Instalar un Web en tu máquina Windows	Si trabajais habitualmente en plataforma Windows, en este tutorial podreis encontrar todos los pasos necesarios para instalar un servidor Web en vuestra propia máquina. Esto puede ser el primer paso para crear una Intranet.
Seguridad en Tomcat	Os mostramos como proteger de un modo básico el acceso a recursos dentro de vuestro servidor de componentes Tomcat
Automatización de procesos con Scripts en Windows	En este tutorial aprendereis a simplificar la actualización automática de ficheros de vuestro Web, utilizando comandos ftp.
Activar SSL en IIS	Os mostramos comos activar el soporte de https en IIS, creando vuestros propios certificados autofirmados, usando OpenSSL
Administracion Web de MySQL	Os mostramos como instalar en vuestro PC phpMySQL, un potente gestor Web de MySQL utilizado en la mayoría de los Hostings.
Mailings con asp y Base de Datos	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).

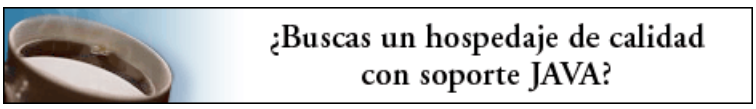
Nota: Los tutoriales mostrados en este Web tienen como objetivo la difusión del conocimiento.

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 Opimizado 800X600