

¿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)



[Home](#) | [Quienes Somos](#) | [Empleo](#) | [Tutoriales](#) | [Contacte](#)

Tutorial desarrollado por: [Daniel Hernandez del Peso](#)

Puedes encontrarme en [Autentia](#)
Somos expertos en **Java/J2EE**
Contacta en info@autentia.com



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

Curso Web J2EE

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

Raona Ingenieria Software

Consultoría software especializada en nuevas tecnologías
www.raona.com

Formación Empresas

Consultoría de Formación Tecnologías Web
www.isthari.com

Anuncios Goooooogle

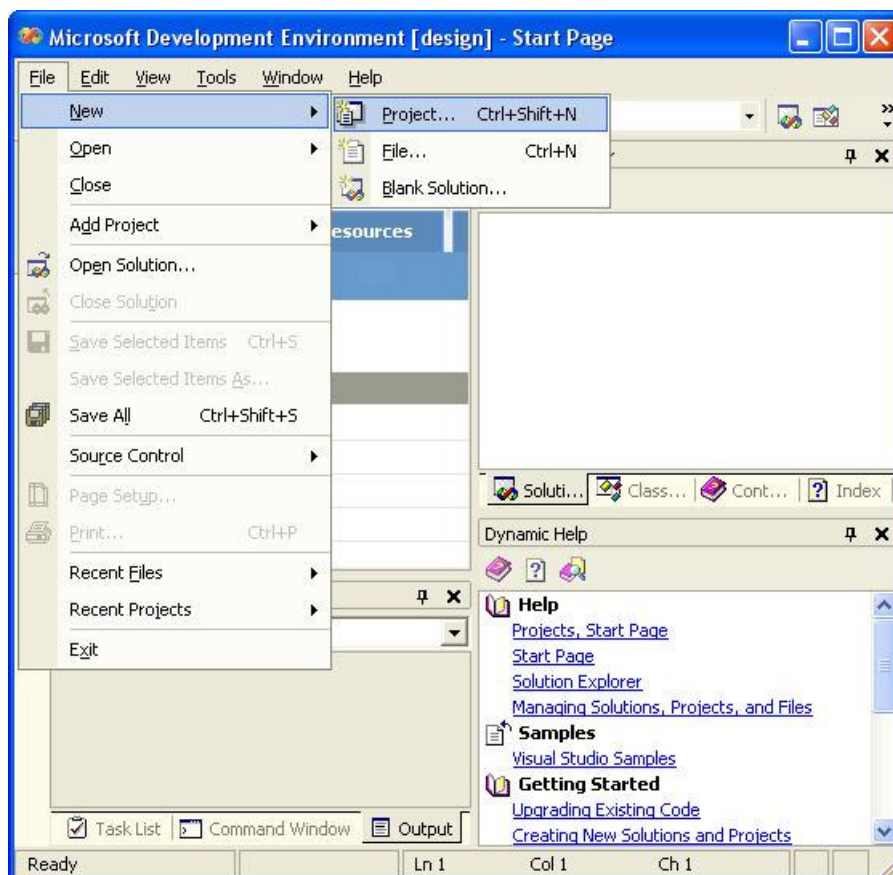
Anunciarse en este sitio

Upload de ficheros en .NET

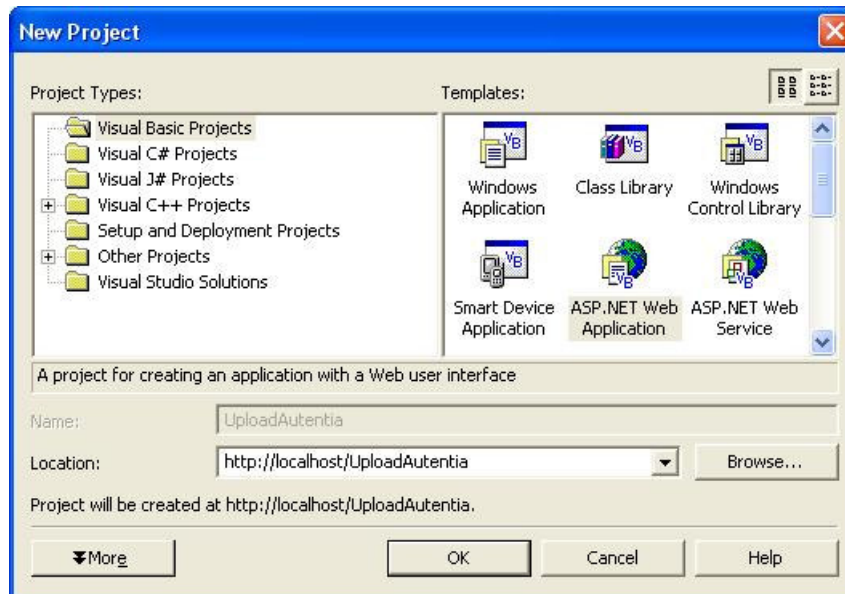
En el desarrollo de aplicaciones Web, puede resultar interesante o necesario realizar uploads (cargas) de ficheros desde el cliente al servidor. En este tutorial veremos como podemos hacerlo de forma sencilla y rápida en la plataforma de Microsoft.

Preparación del entorno

Una vez arrancado el Visual Studio, creamos un nuevo proyecto:

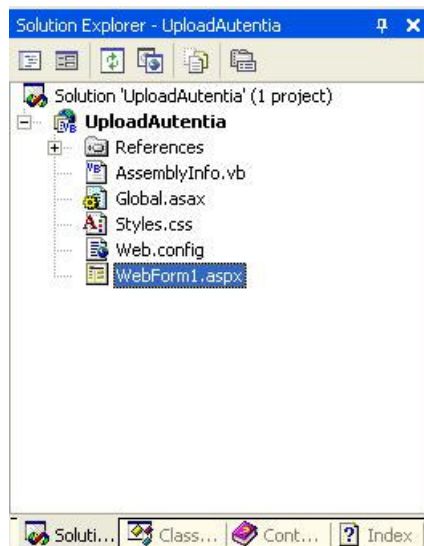


Vamos a crear un proyecto de aplicación Web ASP.NET de Visual Basic:



Y lo llamaremos, en este caso, *UploadAutentia*.

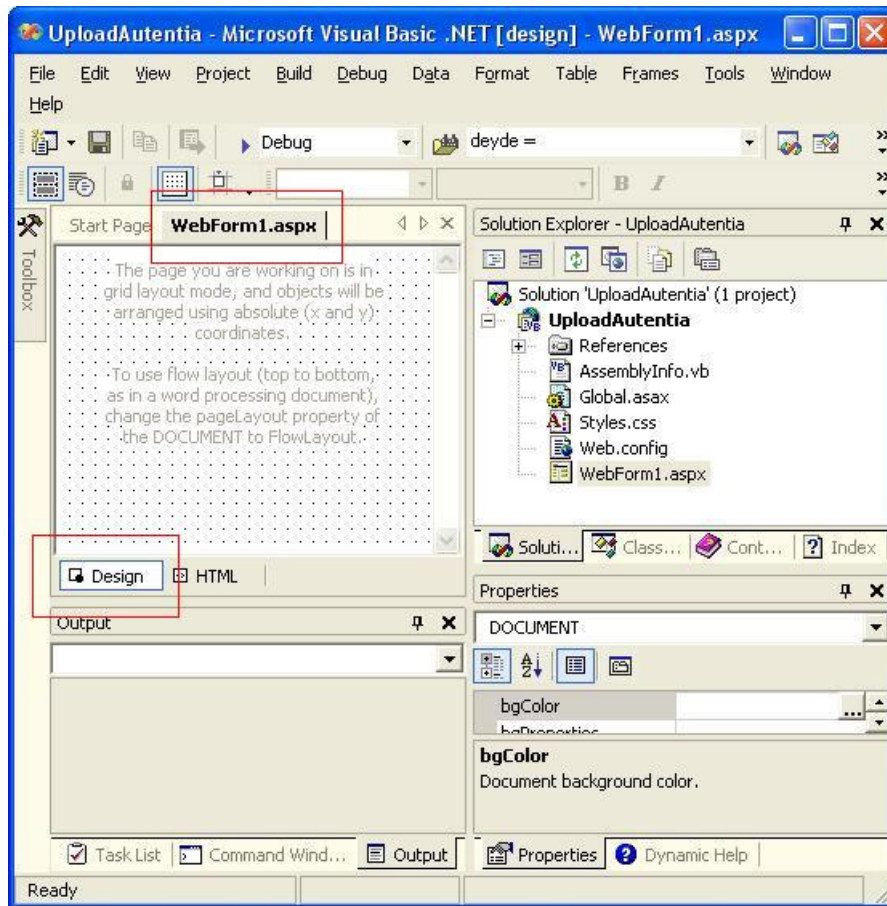
El asistente crea todos los ficheros necesarios para que la aplicación funcione correctamente:



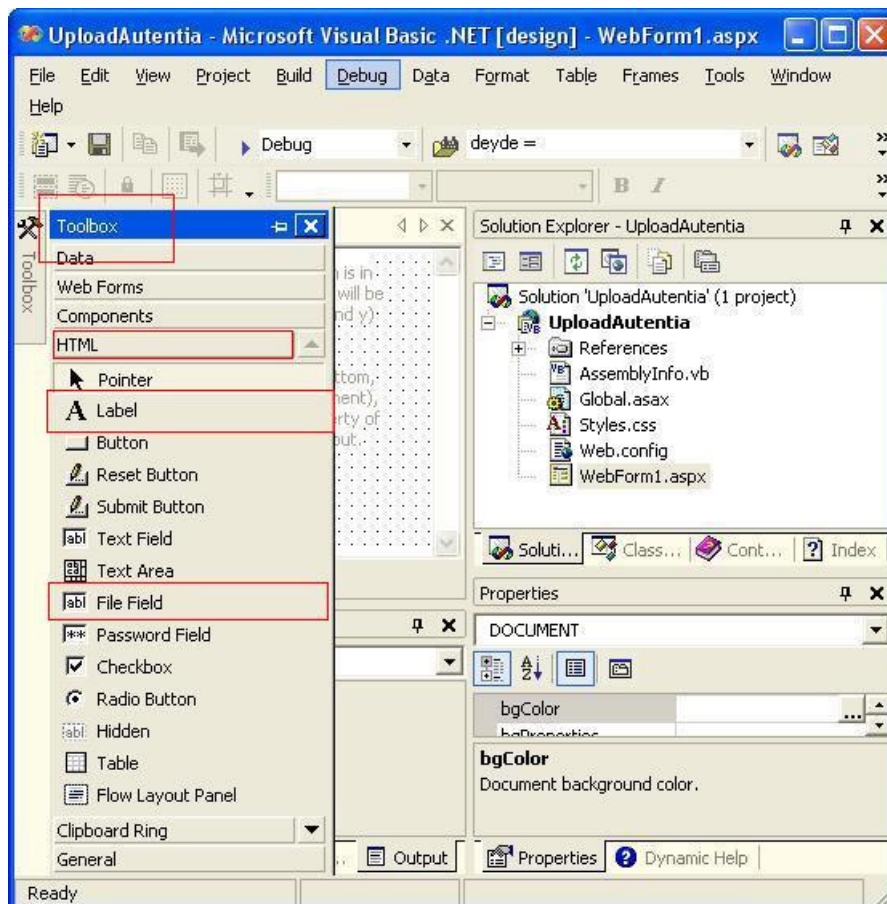
Ahora ya tenemos todo listo y podemos empezar a hacer pruebas...

Creación del formulario Web

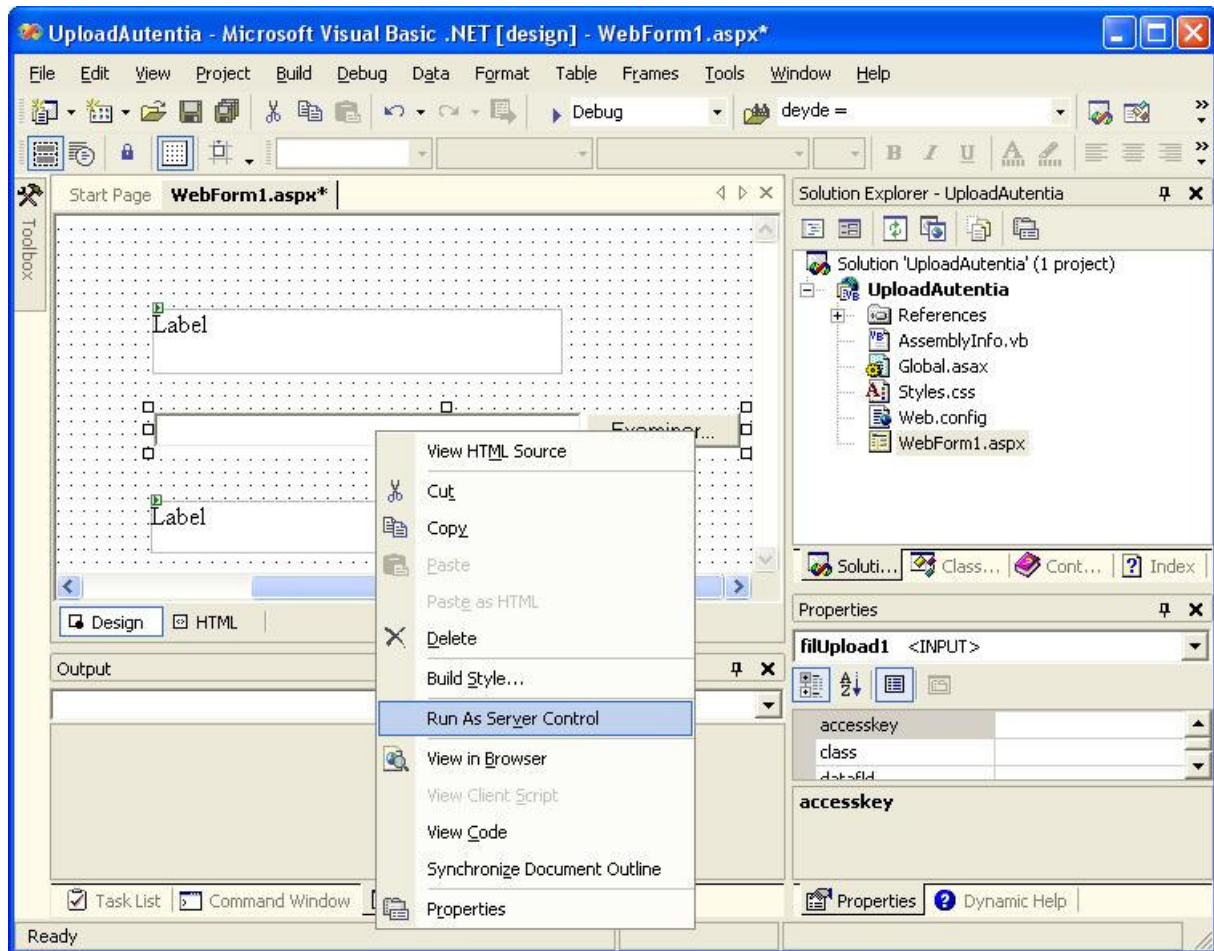
Abrimos el formulario (fichero *WebForm1.aspx*) y, en la vista de diseño



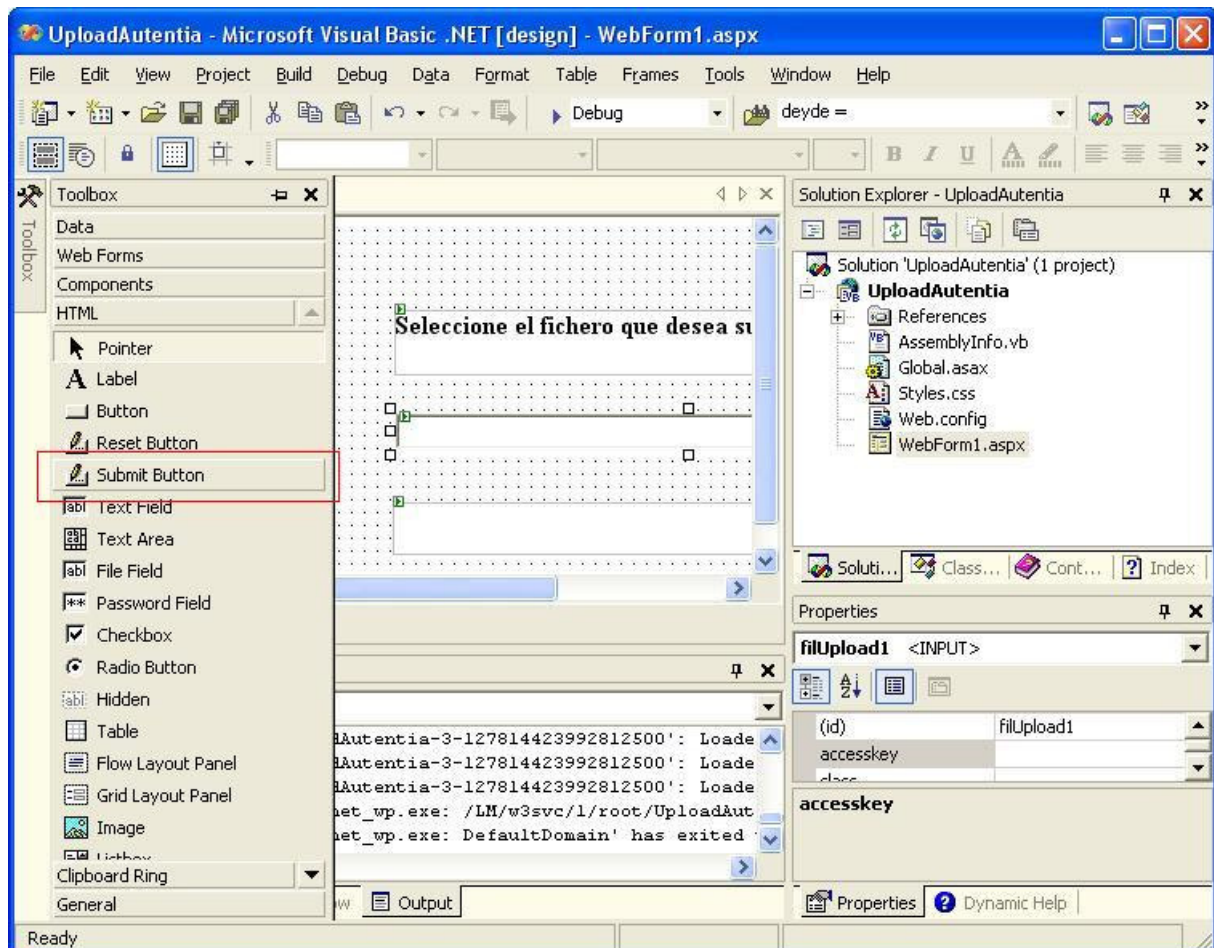
agregamos desde la lista de componentes (botón *toolbox*, se despliega al poner el puntero del ratón sobre el) un *File Field* (al que llamaremos *filUpload*) y dos *Label*, una encima y otra debajo (con nombres *lblTitulo* y *lblResultado* respectivamente) del *File Field*



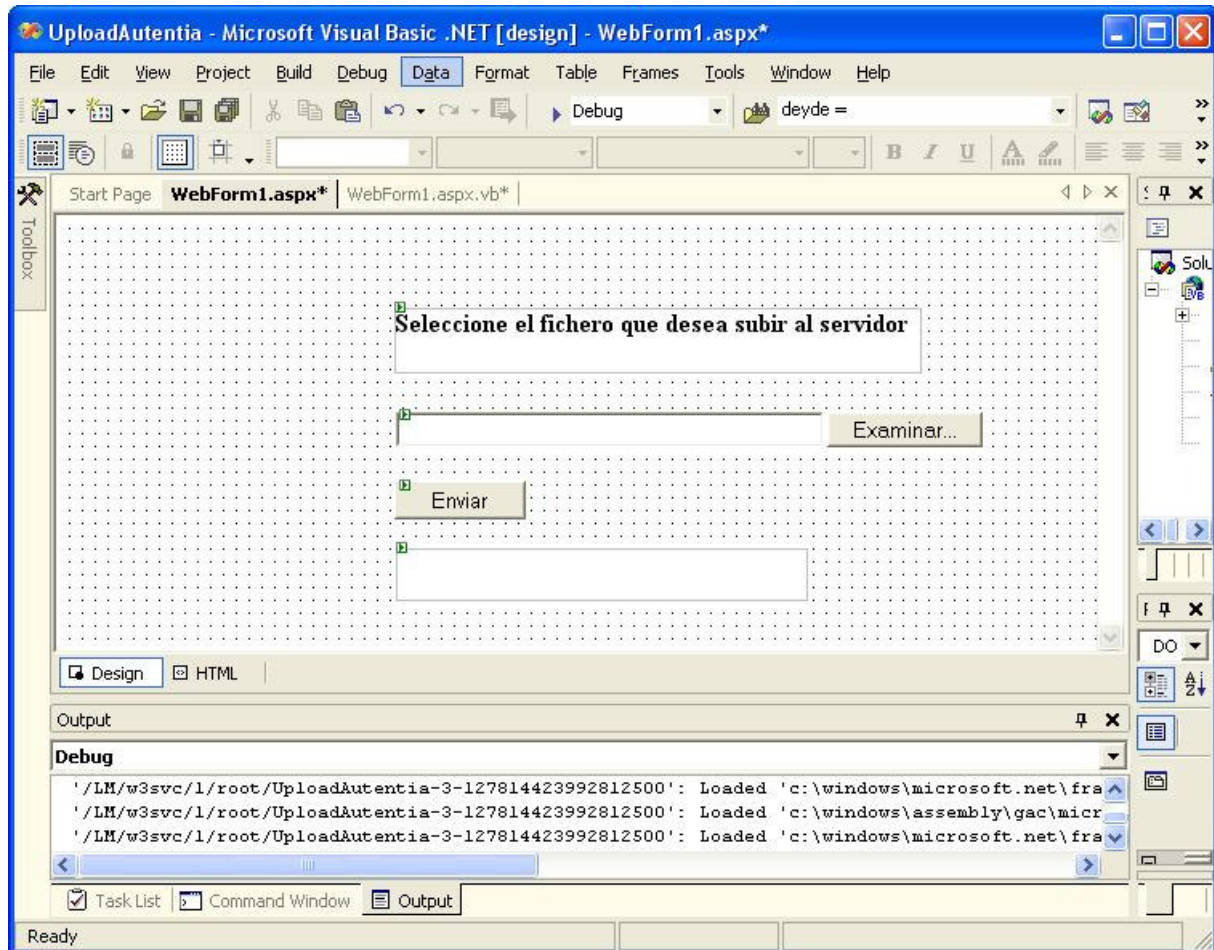
Tenemos que decir al Visual Studio que los elementos que hemos creado van a ser elementos que se ejecuten en el servidor, para ello, pinchamos sobre ellos con el botón derecho del ratón y en el menú que se despliega seleccionamos la opción *Run as Server Control*:



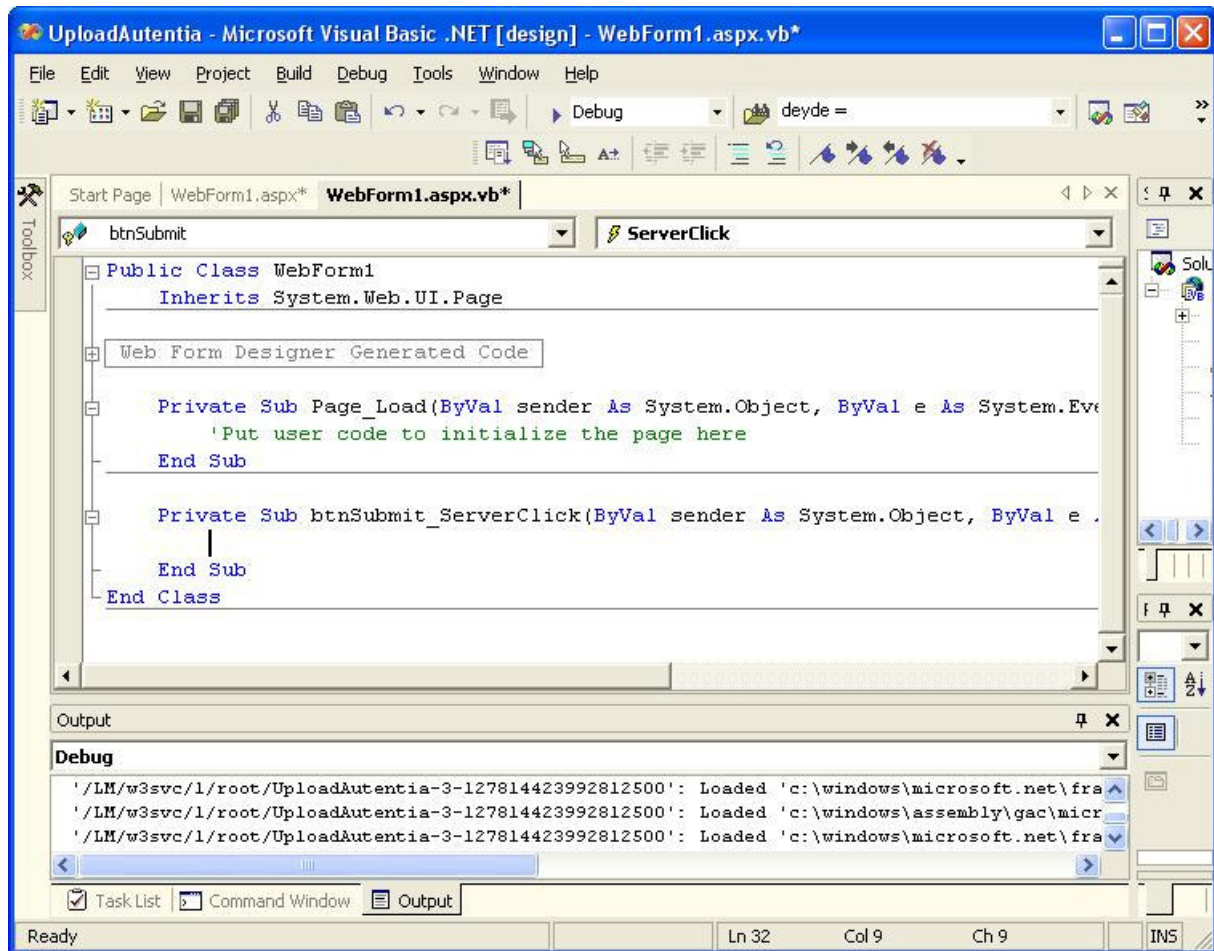
También añadiremos un *Submit Button*, al que llamaremos *btnSubmit*, y que, al igual que hicimos con los otros componentes, especificaremos que se debe ejecutar como un control en el servidor:



Este es el aspecto final que tendrá el formulario:



Ahora, hacemos doble clic sobre el botón de *submit* que hemos añadido y se abre una ventana con el código Visual Basic de la clase que manejará el funcionamiento de la página aspx (el formulario que acabamos de crear):



En ese código hay dos procedimientos: el procedimiento *Page_Load* se ejecuta cuando se carga la página, y el *btnSubmit_ServerClick* se llama cuando se hace clic sobre el botón *submit*

A continuación vamos a añadir el código necesario en ambos procedimientos para realizar el upload de ficheros

Procedimiento *Page_Load*

El código de este procedimiento será el siguiente:

```

Private pathCompletoUpload As String

Private Sub Page_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
    'Put user code to initialize the page here
    Dim pathAplicacion As String
    Dim carpetaUpload As String
    Dim directorio As String

    pathAplicacion = Request.PhysicalApplicationPath
    carpetaUpload = "upload"
    pathCompletoUpload = pathAplicacion & carpetaUpload

    directorio = Dir(pathCompletoUpload, FileAttribute.Directory)
    If (directorio = "") Then
        Mkdir(pathCompletoUpload)
    End If
End Sub

```

El código en azul corresponde a una variable global de la clase, y por tanto accesible desde todos sus procedimientos, que almacenará la ruta física en la que se guardarán los ficheros que se deseen cargar en el servidor

El código en color púrpura es el que se encarga de crear dicha ruta. Hemos elegido que en este caso se almacenen los ficheros en una carpeta llamada "upload" en el directorio donde se encuentra ubicada la aplicación, que extraemos mediante el comando **Request.PhysicalApplicationPath**

En el fragmento de código de color rojo se comprueba si la carpeta que vamos a usar existe ya, y en ese caso no se hace nada. Pero si no existe, mediante la llamada **Mkdir(pathCompletoUpload)** creamos la carpeta.

Como se ha dicho anteriormente, este método se ejecuta al cargar la página. Ahora veamos el código que se ejecuta al presionar el botón de submit.

El procedimiento btnSubmit_ServerClick

El procedimiento quedará finalmente como sigue:

```
Private Sub btnSubmit_ServerClick(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnSubmit.ServerClick

    Dim fileName As String = filUpload.PostedFile.FileName()
    fileName = System.IO.Path.GetFileName(fileName)

    Try
        filUpload.PostedFile.SaveAs(pathCompletoUpload & "\" & fileName)
        lblResultado.InnerHtml = "<b><font color=""#0000FF"">La carga del fichero
fue correcta</font></b>"

    Catch ex As Exception
        lblResultado.InnerHtml = "<b><font color=""#FF0000"">Se ha producido un
error al cargar el fichero</font></b>"
    End Try
End Sub
```

El texto en **rojo** coge el nombre del fichero que se haya enviado. Primero en la línea de código **Dim fileName As String = filUpload.PostedFile.FileName()**, cogemos el nombre del fichero del objeto del formulario (del *File Field*), que nos lo manda completo (con toda la ruta del fichero en el cliente) y luego nos quedamos sólo con el nombre del fichero propiamente dicho (con la llamada **System.IO.Path.GetFileName(fileName)**)

El código en color **azul** es el que guarda en el servidor el fichero enviado, en la carpeta creada al cargar la página.

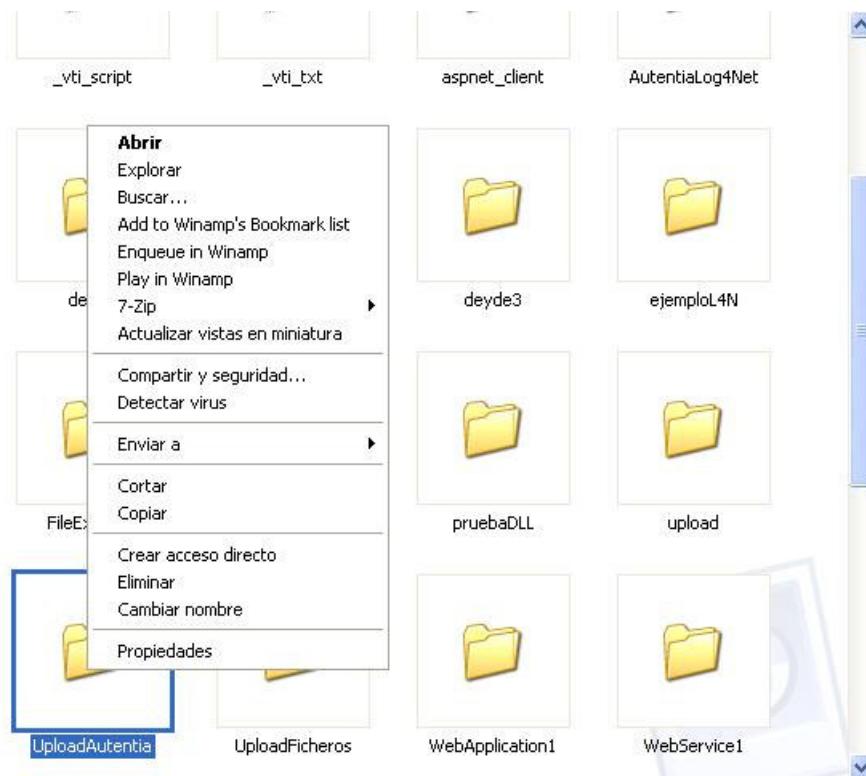
Los textos en **púrpura** y **verde** escriben en la etiqueta *lblResultado* el resultado de la operación.

Ya estamos en disposición de probar la aplicación completa...

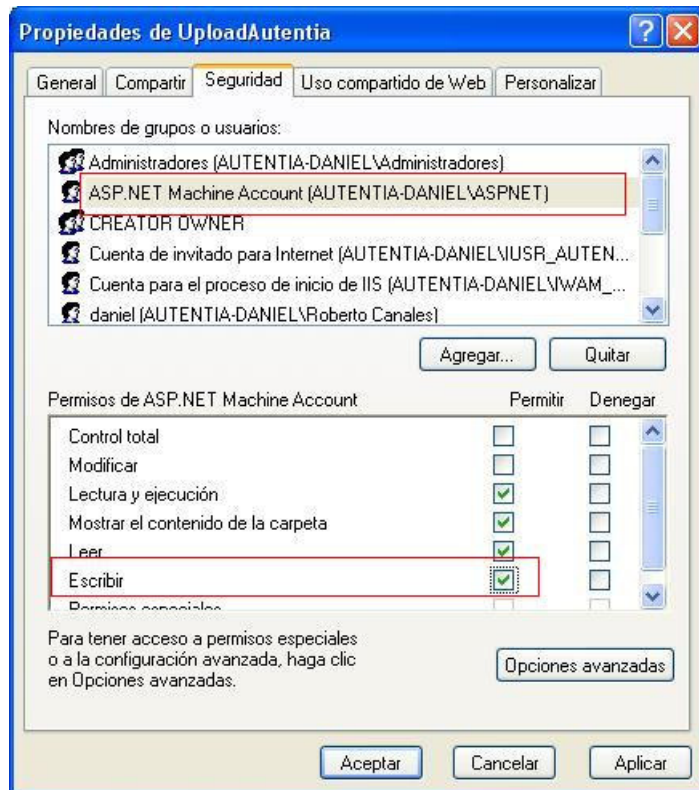
Subiendo ficheros al servidor

Antes de empezar a utilizar la aplicación, hay que realizar un paso previo sin el cual ésta no funcionará.

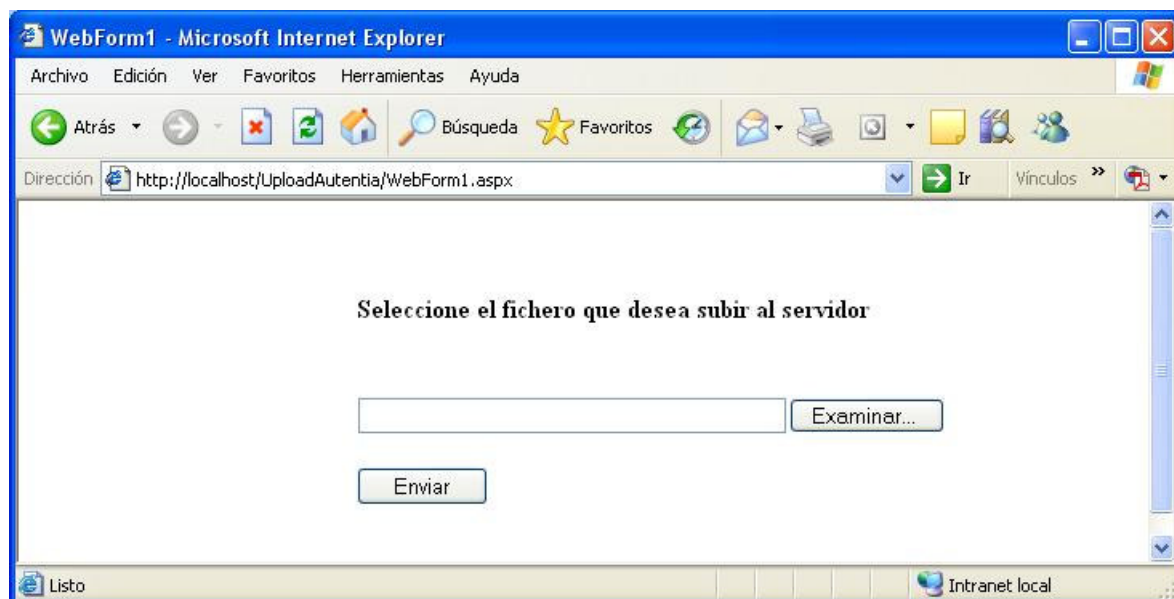
Tal como se vio en el tutorial de [log4net](#) publicado también en "Adictos al Trabajo", las aplicaciones .NET son ejecutadas en el servidor por un usuario especial, el usuario **ASPNET**. Para que la aplicación funcione, hay que dar a este usuario permisos de escritura en la carpeta en que se vayan a guardar los ficheros subidos. Para ello, pinchamos con el botón derecho sobre dicha carpeta (por defecto, **c:\inetpub\wwwroot\nombre_aplicacion**) y seleccionamos "Propiedades":



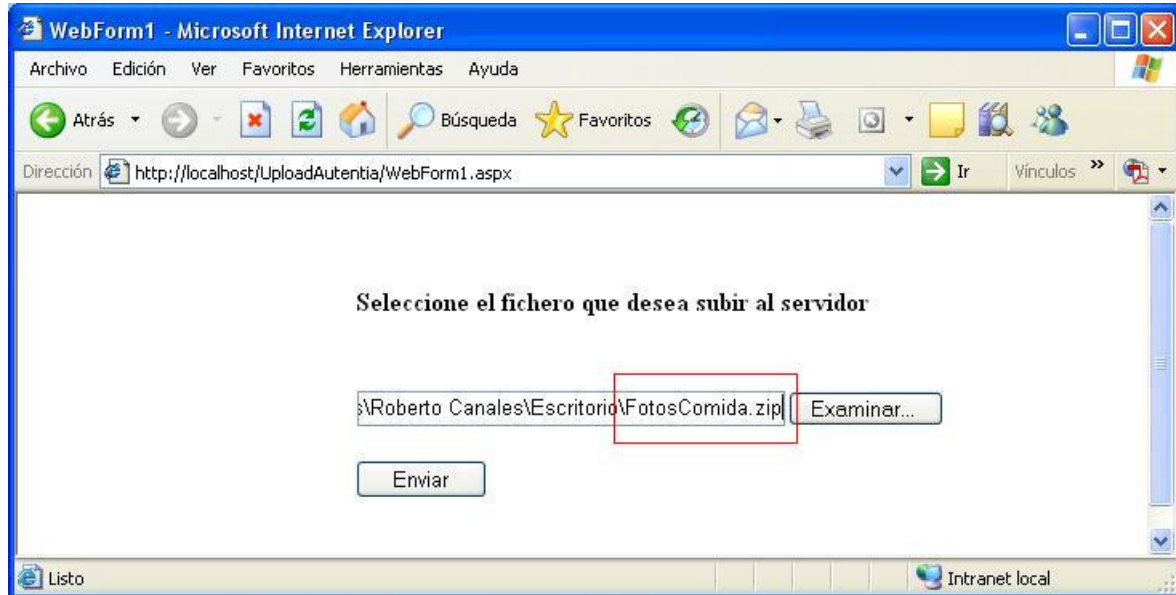
y en la pantalla que se abre, vamos a la pestaña "Seguridad" y localizamos al usuario "ASPNET" en la lista. Una vez localizado, además de todos los permisos que tenga ya asignados hay que proporcionarle permiso de escritura en la carpeta:



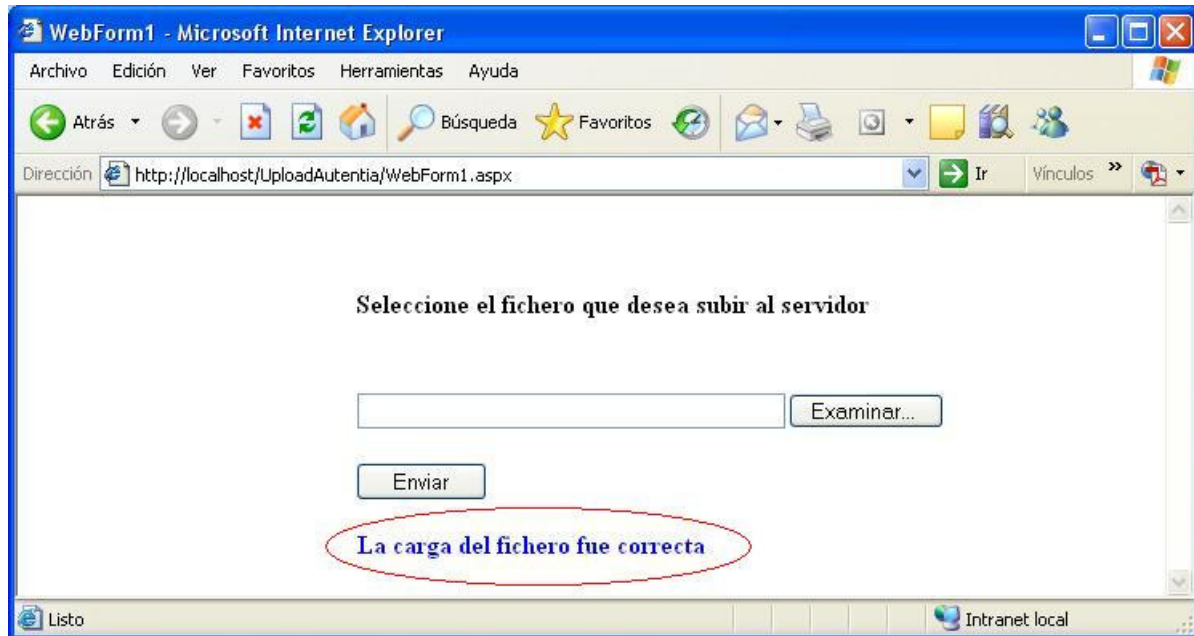
Con esto conseguiremos que se puedan grabar los ficheros, por lo que ya estamos preparados para probar la aplicación. La abrimos en el navegador...



seleccionamos el fichero...

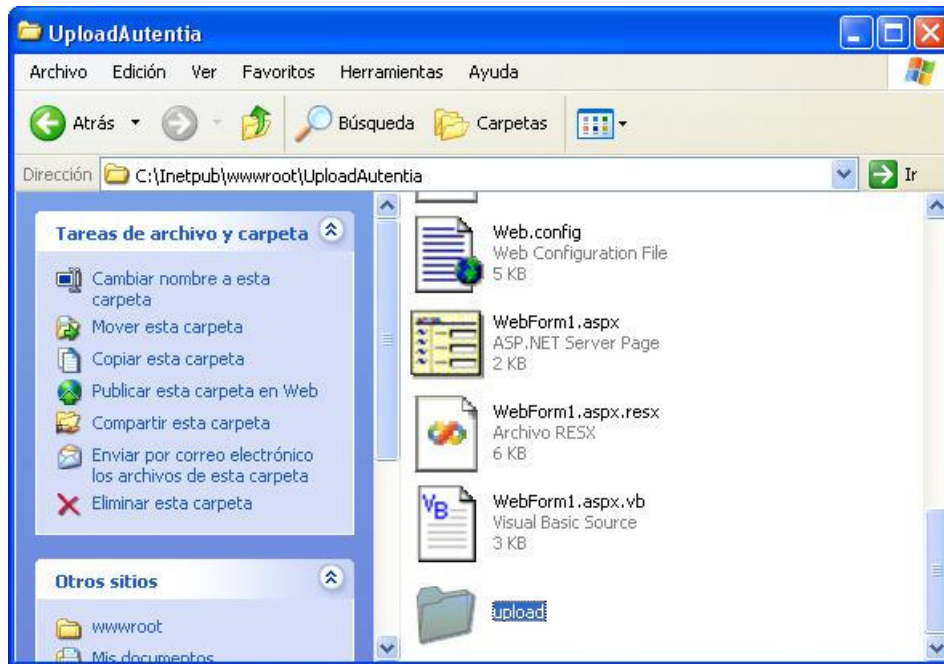


pulsamos "Enviar" y...

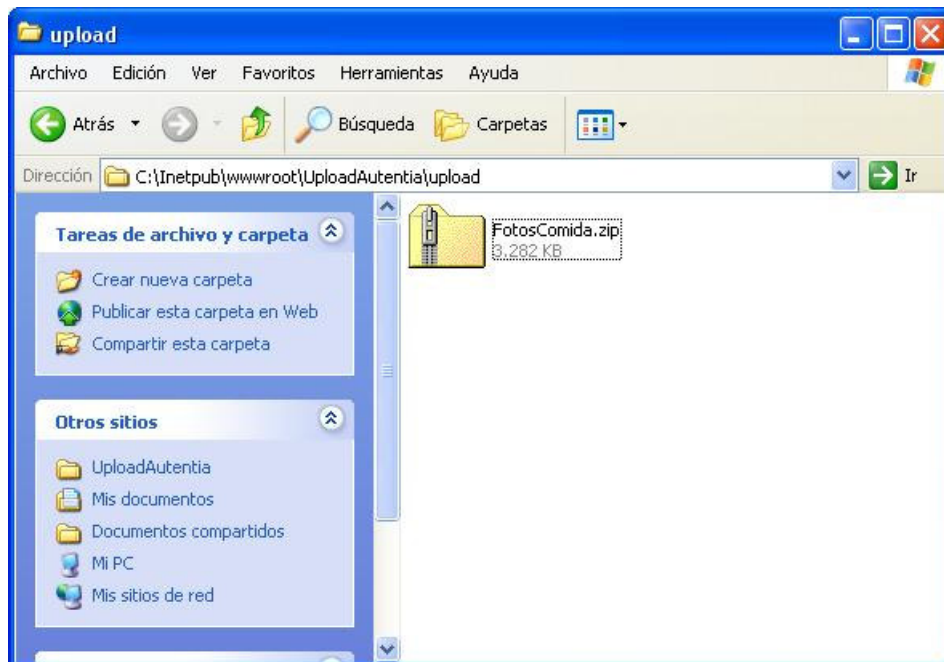


Tal como vemos, la carga se ha realizado (parece) correctamente. Comprobemos si es cierto. Vamos a la carpeta en que se ubica la aplicación (c:\inetpub\wwwroot\UploadAutentia en nuestro ejemplo)

y comprobamos que ha aparecido una carpeta "upload":



y dentro de ella...



... El fichero que hemos enviado

Pues ya lo tenemos andando... Ahora os queda a vosotros el mejorar esta aplicación para adaptarla a vuestras necesidades.

Agradecimientos

Quiero agradecer a www.4guysfromrolla.com, que me ayudó a resolver gran parte de las dudas y problemas que me surgieron...

Conclusión

Espero que este tutorial os haya servido para resolver vuestras dudas, pero si no ha sido así, ya sabéis que en mi empresa, [Autentia](#), estamos para ayudaros con vuestros desarrollos

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

soluciones reales para su negocio

[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	<input type="text"/>
	<input type="button" value="Enviar"/>

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

Nombre Corto

[Primeros programas OpenGL con Visual C++](#)

[Ado y Net](#)

[Motor de Búsqueda en Tu Web](#)

[Cliente y Servidor DCOM con MS Visual C++](#)

[Aplicaciones con el framework de Microsoft .NET](#)

[Autenticación basada en formularios \(IIS y .Net\).](#)

[Control navegación en Servlets](#)

[Upload de ficheros en Java](#)

[AdRotator en Página ASP .Net](#)

[Primer vistazo a C# en Visual Studio 2005 Beta 2](#)

Descripción

Con este tutorial, como insertar código OpenGL en un programa Visual C++, creando un sistema planetario básico.

En este tutorial podeis descubrir como funciona ADO y las nuevas diferencias en arquitectura .Net .

En este tutorial te enseñamos como configurar un buscador externo para proporcionar capacidades de búsqueda a tu Web. El buscador en este ejemplo es Atomz.

Os mostramos como construir un servidor y cliente de Automatización OLE con las facilidades que proporciona el entorno de desarrollo Microsoft Visual C++

Ejemplo de desarrollo de una aplicación con el framework de Microsoft .NET (creación de un servicio deEncuestas Web)

Luis Paniagua nos enseña como realizar la configuración de un directorio privado en el servidor IIS y con .Net Framework instalado, al que accederemos mediante la validación de usuario en un formulario Web.

Os mostramos como construir el esqueleto de una aplicación basada en Servlets y JSP, con control de navegación.

Os mostramos como enviar ficheros a un servidor Web y manipularlos en un servlet en el servidor, gracias a APIs de apache

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

Echamos un primer vistazo a C# en Visual Studio 2005 Beta 2 y os mostramos algunas diferencias respecto al lenguaje Java

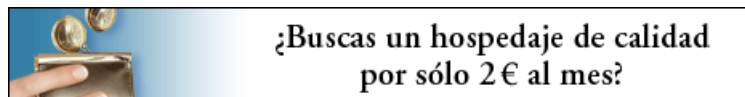
Nota: Los tutoriales mostrados en este Web tienen como objetivo la difusión del conocimiento.

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