

# Programación Visual en C#.NET

## Guion de la práctica 8

### OBJETIVOS

- Aprender a crear y trabajar con aplicaciones web
- Aprender a crear y trabajar con servicios web

### TEMPORIZACIÓN

Recogida del enunciado:	Semana del 28 de noviembre
Desarrollo de la práctica:	<b>3 semanas</b>
Fecha de entrega:	Semana del 19 de diciembre
Fecha límite de entrega:	Semana del 19 de diciembre

## **PRÁCTICA 8**

### **Aplicaciones y servicios web**

#### **TABLA DE CONTENIDOS**

<b>INTRODUCCIÓN .....</b>	<b>2</b>
<b>12.1 Creación de la base de datos .....</b>	<b>2</b>
<b>12.2 Creación del sitio web.....</b>	<b>4</b>
<b>12.3 Añadir fotos a nuestra base de datos .....</b>	<b>8</b>
<b>12.4 Borrar Fotos.....</b>	<b>11</b>
<b>12.5 Creación de un servicio web y su cliente .....</b>	<b>12</b>
<b>12.6 Autenticación, autorización y roles.....</b>	<b>15</b>
12.6.3 Ocultar opciones del menú .....	17
<b>12.7 AJAX .....</b>	<b>18</b>
<b>12.8 Publicar el sitio web y el servicio web.....</b>	<b>19</b>

## Introducción

Una parte importante de *MS Visual Studio* (o, en su defecto, *MS Visual Studio Community*) es su capacidad para crear aplicaciones distribuidas basadas en la web. *MS Visual Studio* permite crear la interfaz de usuario de la aplicación mediante páginas que muestran formularios web, y crear componentes utilizando servicios web XML.

- Formularios web: tecnología de ASP.NET que permite crear interfaces de usuario en aplicaciones web, para clientes que obtengan acceso a la aplicación tanto desde un explorador web tradicional como desde un dispositivo móvil.
- Servicios web WCF: son componentes que se ejecutan en el servidor, y suelen incluir lógica de negocio. Al igual que los componentes tradicionales, los servicios web encapsulan funciones específicas, y se les puede llamar desde diferentes programas. Sin embargo, están disponibles a través de protocolos web, lo que les hace compatibles con programas que se ejecutan en diferentes lenguajes, en diferentes equipos e, incluso, en diferentes sistemas operativos.

El objetivo de esta práctica es crear una aplicación web que permita administrar una base de datos para almacenar fotos. Esta aplicación podrá ser accedida desde un explorador, permitiendo realizar las operaciones de mostrar, añadir y borrar fotos. Estas operaciones serán proporcionadas, en una primera fase, por una aplicación web (puntos 12.1 a 12.4), y en una segunda (punto 12.5), las operaciones de añadir y borrar pasarán a ser proporcionadas por un servicio web que será accedido desde la aplicación web. En los puntos restantes, se implementará la autenticación, la autorización, AJAX y se publicará la aplicación web en un servidor de aplicaciones.

La operación de mostrar estará disponible para todos los usuarios que accedan al sitio web, la operación de añadir solo estará disponible para los usuarios autenticados, y la operación de borrar, para los usuarios autenticados que sean administradores.

### 12.1 Creación de la base de datos

Para la realización de esta práctica, deberá tener instalado *Microsoft SQL Server*. Deberá crear la base de datos en la que almacenará su álbum de fotos, haciendo uso del *script* que se le ha proporcionado junto con el enunciado de esta práctica (archivo *Práctica-8\_C#\_recursos.txt*). Para ello siga los siguientes pasos:

1. Vaya a *Inicio > Ejecutar*, y abra la línea de órdenes (mediante la orden **cmd**) y escriba la siguiente orden:

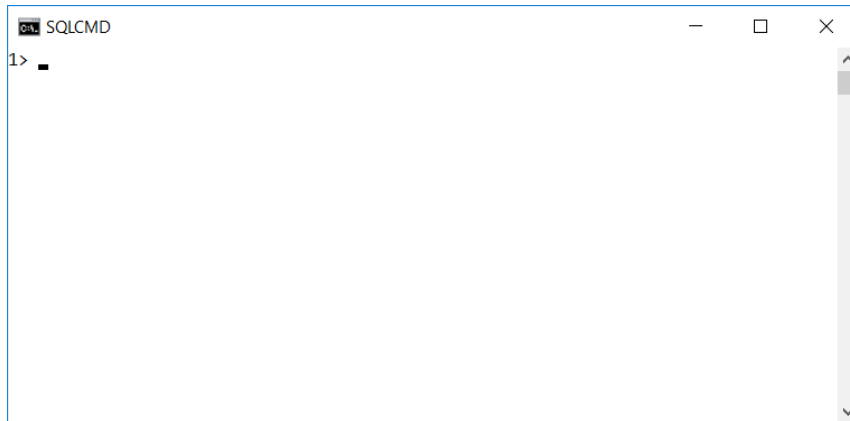
```
> sqlcmd -S localhost\SqLExpress
```

**Nota:** S MAYÚSCULA. Pudiera darse el caso de que el ejecutable *SQLCMD.EXE* no se encontrara exactamente en la misma ruta en su máquina. Compruébelo.

Si se ha ejecutado correctamente salte al punto 2. Si la ejecución de la orden ha retornado un error pruebe a ejecutarla de nuevo desde la carpeta donde se encuentra el fichero. Teclee las siguientes ordenes:

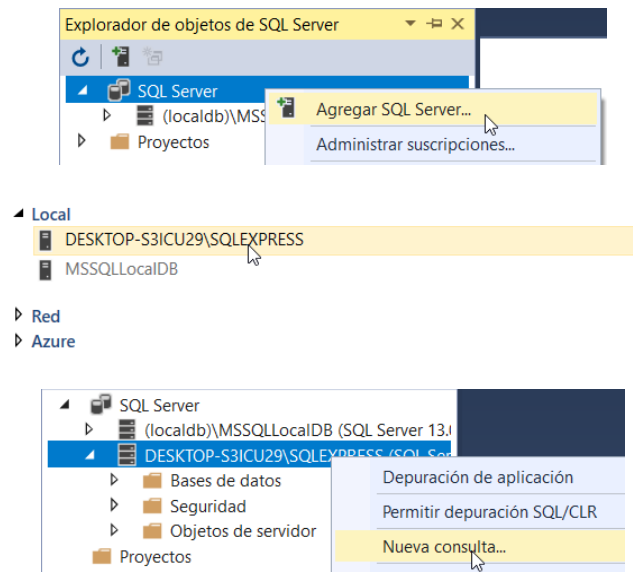
```
> cd "C:\Program Files\Microsoft SQL Server\Client SDK\ODBC\130\Tools\Binn"  
> sqlcmd -S localhost\SqLExpress
```

2. Abra el *script sql* adjunto con un editor de texto cualquiera, copie su contenido en la línea de órdenes (que tendrá un aspecto como el de la ventana a continuación), y pulse *Entrar*:



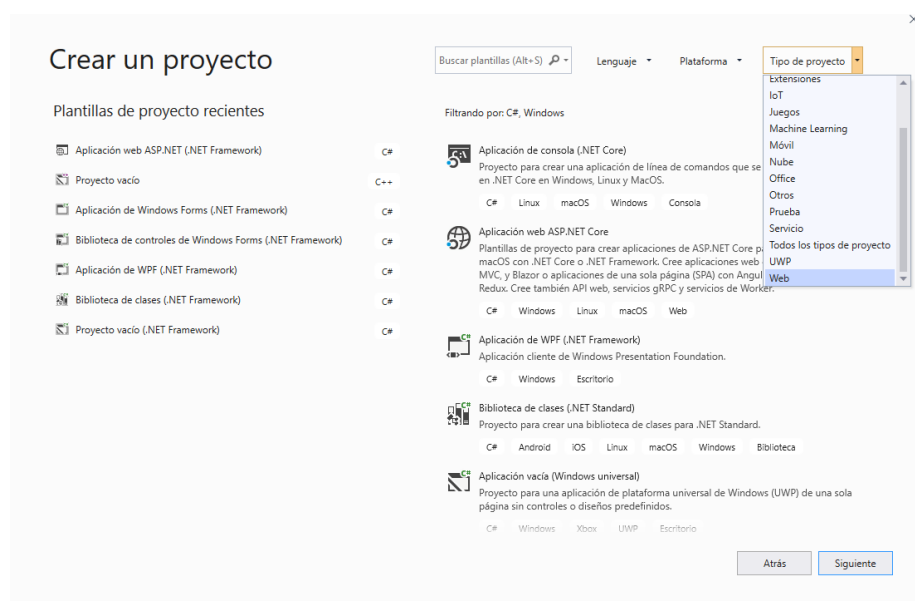
3. La base de datos ya está creada, se llama **fotos** y contiene una tabla **album** con las siguientes columnas:
  - **foto**: es una cadena de caracteres de longitud 30. En ella, almacenará el nombre del archivo que contiene la imagen, incluyendo la ruta relativa (fotos/vacaciones.jpg, fotos/archivo0405.jpg, ...)
  - **nombre**: es una cadena de caracteres de longitud 30 en la que almacenará el nombre de la foto (vacaciones, verano2017, cuadro de Van Gogh, etc.).
  - **autor**: contendrá el nombre del creador de la foto/cuadro. Será también una cadena de caracteres de longitud máxima 30.
  - **descripcion**: contendrá un breve texto describiendo la imagen. Como máximo, admitirá 400 caracteres.

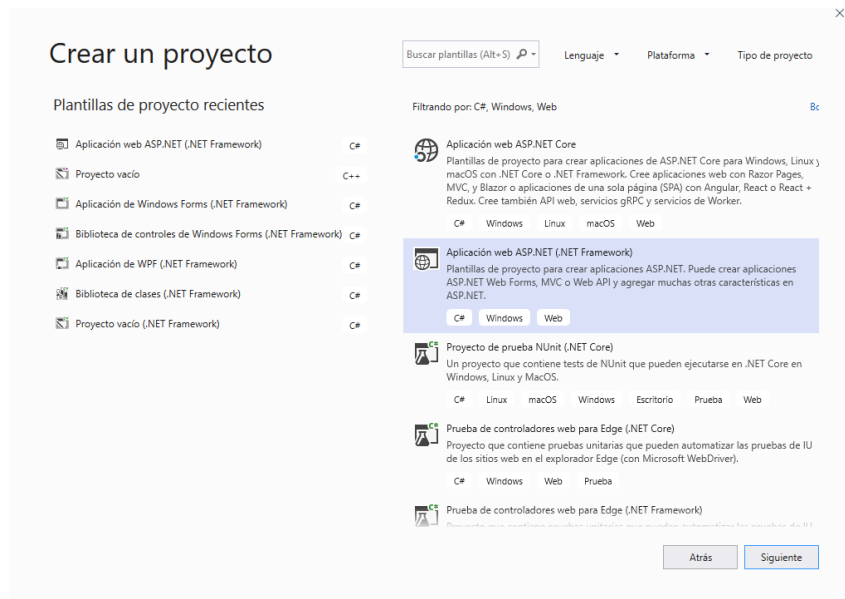
La base de datos puede también ser creada desde *MS Visual Studio* por medio del asistente *Explorador de objetos de SQL Server* al que tendrá que agregar el *SQL Server* anteriormente instalado para poder realizar una consulta que será el *script* anterior.



## 12.2 Creación del sitio web

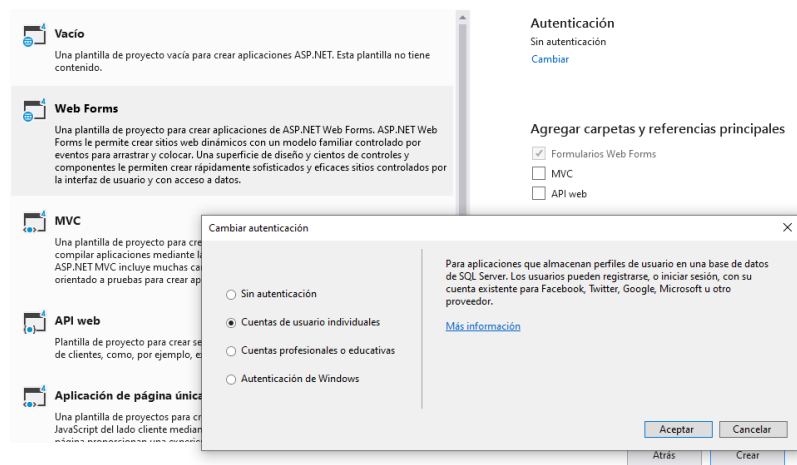
Para realizar este apartado, abra el *MS Visual Studio* (*MS Visual Studio profesional*, *MS Visual Studio Community*, etc. En cualquier caso y de ahora en adelante nos referiremos a este entorno como VS). Cree un nuevo sitio web (Seleccione: *Crear un Proyecto*, *Lenguaje C#*, *Plataforma Windows* y *Tipo de proyecto Web*) utilizando la plantilla **Aplicación web ASP.NET (.NET Framework)**, localizado en una carpeta **SitioWeb** (nombre del proyecto) de su sistema de archivos. Se sugiere crear una solución **Práctica8**, en una carpeta diferente del proyecto. Si en su instalación de Visual Studio no aparece la plantilla **Aplicación web ASP.NET (.NET Framework)** actívela siguiendo las indicaciones del anexo al final del documento.





Seleccione las opciones que muestra la figura siguiente: *Web Forms*, autenticación: *Cuentas de usuario individuales* y haga clic en *Aceptar*:

### Crear una aplicación web ASP.NET



Al crear el nuevo proyecto, podrá ver en el explorador de soluciones el formulario *Default.aspx* creado por defecto. Este formulario, utilizando la página maestra *Site.master*, dará lugar a la página web en la que se mostrarán las fotos de su álbum. Para que la pestaña de esta página muestre el título “Mostrar - Álbum de fotos”, edite *Default.aspx* y asigne a su atributo **Title** la subcadena “Mostrar”; después, edite *Site.master* y asigne a su etiqueta `<title>` el resto de la cadena, esto es, “Álbum de fotos”, reemplazando a la existente.

Eche un vistazo a la página *Site.master*, la cual, además del título, contendrá un menú (o barra de opciones) para navegar por las páginas de contenido, así como un inicio de

sesión. Si no es así, tendrá que crear esta infraestructura manualmente. La primera opción de esa barra es:

```
<a class="navbar-brand" runat="server"
    href="/">Nombre de la aplicación</a>
```

Sustituya esta línea por la siguiente, con el fin de establecer su propio logo en la página *Site.master*. Añada al proyecto una carpeta *images* para guardar las imágenes de esta aplicación.

```
<asp:Image ID="Logo" runat="server" ImageUrl="~/images/miLogo.png"
    BorderStyle="None" />
```

Elimine del cuerpo de la página *Default.aspx*, que se encuentra entre las etiquetas **Content**, el contenido que no sea necesario. El resultado será análogo al siguiente:

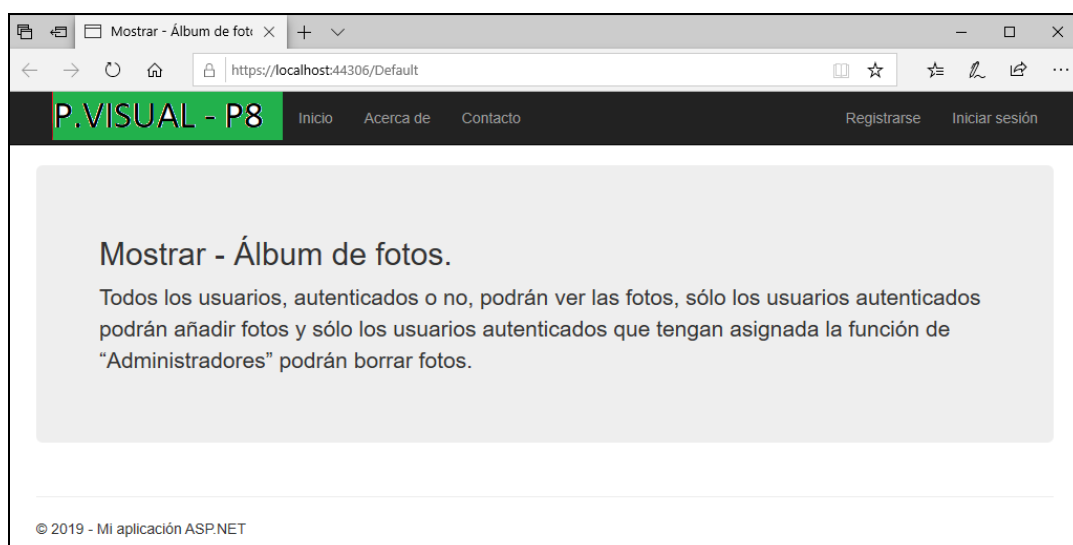
```
<%@ Page Title="Mostrar" ... %>

<asp:Content ID="BodyContent" ContentPlaceHolderID="MainContent" runat="server">

    <div class="jumbotron">
        <h2><%= Title %> - Álbum de fotos.</h2>
        <p>
            Todos los usuarios, autenticados o no, podrán ver las fotos, sólo los
            usuarios autenticados podrán añadir fotos y sólo los usuarios autenticados
            que tengan asignada la función de "Administradores" podrán borrar fotos.
        </p>
    </div>

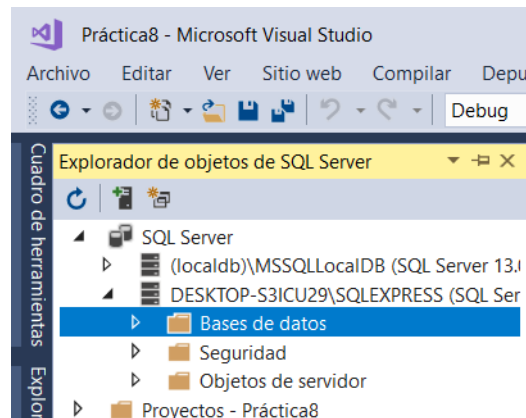
    <div class="row">
        <div class="col-md-4">

        </div>
    </div>
</asp:Content>
```



La autenticación se realizará por medio del proveedor de autenticación de formularios utilizando una base de datos SQL Server (las clases que proporcionan la interfaz para realizar este proceso, son proporcionadas por la biblioteca de ASP.NET). De esto nos ocuparemos en el apartado 12.6.

Vaya al explorador de objetos de SQL Server (si no está visible: *Ver > Explorador de objetos de SQL Server*) y verifique en el nodo *Bases de datos* la existencia de *fotos*. El servidor de base de datos será **Microsoft SQL Server Express**. Si no se visualiza, intente añadirlo haciendo clic con el botón secundario del ratón en el nodo *SQL Server*.



Añada a la página *Default.aspx* (entre las etiquetas **Content** correspondientes al cuerpo de la página) un control **SqlDataSource**. En su menú de tareas (al cual se accede haciendo clic sobre la flecha que aparece junto a dicho control, en la vista de diseño), escoja la opción *Configurar origen de datos* y pulse en *Nueva conexión*. Con ello, elegirá la fuente de datos:

Origen de datos: Microsoft SQL Server (SqlClient)

Nombre del servidor: .\sqlexpress

Autenticación: de Windows

Nombre de la base de datos: fotos

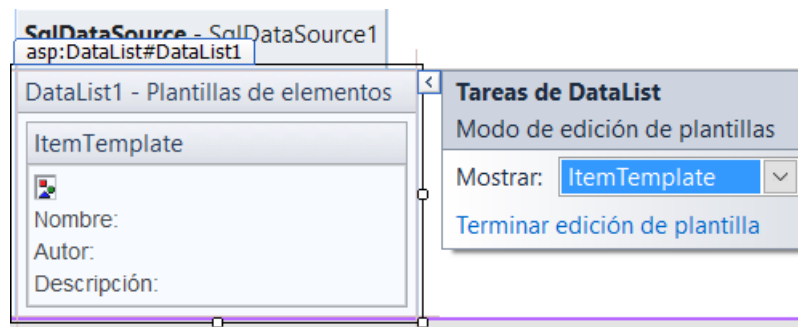
Continúe pulsando *Siguiente* y configure la sentencia SELECT para que permita recuperar todas las columnas de la tabla *álbum* y, además, ordene las filas seleccionadas de la tabla por *autor*.

Cuando finalice la configuración del control **SqlDataSource**, observe que en el fichero de configuración *web.config* hay dos cadenas de conexión: la correspondiente a la base de datos de usuarios (local) y la correspondiente a la base de datos *fotos* (remota).

A continuación, añada desde la caja de herramientas, un objeto de tipo **DataList** y, desde su lista de tareas, haga que su fuente de datos sea la anterior (*SqlDataSource1*).

Modifique el objeto anterior (mediante la opción *Editar plantillas*) para que quede como se muestra en la siguiente figura: cada elemento de la lista *DataList1* contiene un control **Image**, para mostrar la foto, y tres controles **Label** para mostrar los datos nombre, autor y descripción de la foto.





Para ello, abra la lista de tareas programadas de *DataList1*, haga clic en *Editar plantillas* > *Mostrar: ItemTemplate*, añada un control imagen (*Caja de herramientas* > *Estándar* > *Image*), abra su lista de tareas (*la del control imagen*), haga clic en *Editar enlace de datos*, seleccione **ImageUrl**, clic en *Actualizar esquema*, *Enlace a campo* y enlázelo con el campo *foto* de nuestra base de datos (opcionalmente, cuando vea los resultados más adelante, puede establecer un ancho para la imagen si lo desea, propiedad **Width**). Siguiendo un proceso análogo añada el resto de elementos (de tipo **Label**) y enlázelos con los campos correspondientes de la base de datos. Haga clic en *Aceptar* y, finalmente, haga clic en *Terminar edición de plantilla* de lista de tareas programadas de *DataList1*.

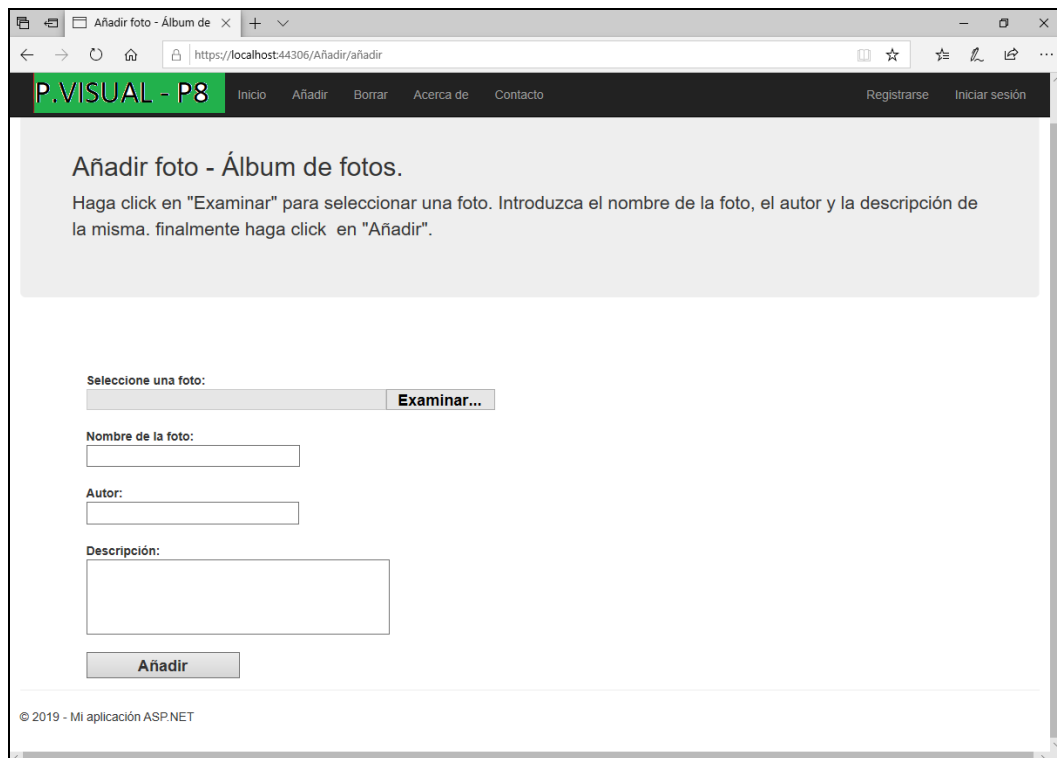
### Compile y ejecute para ver el resultado.

Si todo ha ido bien debería verse una web en blanco ya que aún no ha añadido datos a su tabla. Si se produjese algún error revise la configuración de SQL.

Configure la web con el título, color, y demás características que considere necesarias. Eche una ojeada a las hojas de estilo que se incluyeron cuando generó el proyecto (carpeta *Content*). Observe también que a partir de Visual Studio 2013 se utiliza para el diseño el *framework Bootstrap*.

## 12.3 Añadir fotos a nuestra base de datos

Añada una nueva página web (elemento de tipo *Formulario Web con página maestra*) a su sitio web que utilice como plantilla la página maestra *Site.master*. Llámela *añadir.aspx*, póngale un título, “Añadir foto”, y añada la pestaña correspondiente, “Añadir”, al menú de navegación. Cree una estructura para esta página análoga a *Default.aspx*.



Cree el formulario anterior usando controles **TextBox** para la entrada de datos, y un control de tipo **FileUpload** para cargar archivos en el servidor. Cree una carpeta “fotos” en su sitio web (como subcarpeta de la carpeta del sitio) para guardar las fotos que se carguen. Ponga nombre a las cajas de texto: *ct\_Nombre*, *ct\_Autor*, *ct\_Descripcion*, y llame al objeto **FileUpload** *sa\_SubirArchivo*. Por último, añada un botón *bt\_Añadir*. Limite el tamaño máximo de las cajas de texto, acorde con el tamaño de los parámetros en la base de datos.

### Compile y ejecute para ver el resultado.

Añada un método *bt\_Añadir\_Click* que responda al evento **Click** del botón *Añadir*. Escriba el siguiente código, añadiendo lo que falte:

```
protected void bt_Añadir_Click(object sender, EventArgs e)
{
    // La carpeta raíz del sitio web es "~/\"
    string sRutaSitioWeb = ...; // ver documentación de Page.Server

    // Subimos los archivos al servidor
    sa_SubirArchivo.PostedFile.SaveAs(sRutaSitioWeb +
                                     "fotos/" + sa_SubirArchivo.FileName);

    //Guardamos la información en la base de datos
    string foto = "fotos/" + sa_SubirArchivo.FileName;
    string nombre = ...;
    string autor = ...;

    string descripcion = ct_Descripcion.Text;
    if (descripcion.Length == 0)
        descripcion = "-- Sin Descripción --";
}
```

```

string sconexión = ...; // ver clase ConfigurationManager
SqlConnection conexión = new SqlConnection(sconexión);

string sOrdenSQL = "insert into album (foto, nombre, autor, " +
    " descripcion) values (@FOTO,@NOMBRE,@AUTOR,@DESCRIPCION) ";

SqlCommand ordenSQL = new SqlCommand(sOrdenSQL, conexión);

ordenSQL.Parameters.AddWithValue("@FOTO", foto);
ordenSQL.Parameters.AddWithValue("@NOMBRE", nombre);

// Idem para autor, descripcion...
// ...

ordenSQL.Connection.Open();
ordenSQL.ExecuteNonQuery();
ordenSQL.Connection.Close();

this.Response.Redirect("Default.aspx");
}

```

### Compile y ejecute para ver el resultado.

Añada en la página *añadir.aspx*, a la derecha de las cajas de texto, un objeto de tipo **Label**. Por ejemplo:

Nombre de la foto:  \* Error: Escriba el nombre!!

Estas etiquetas serán utilizadas para notificar al usuario la validez de los datos introducidos. Llámelas *et\_Nombre*, *et\_Autor*, *et\_Archivo* y con el editor de propiedades, póngalas sin texto inicial y en color rojo.

Para evitar que el usuario pueda aceptar el formulario sin haber rellenado los datos, añadiremos la propiedad de sólo lectura *FormularioCorrecto*, que retornará **true** si todos los datos han sido introducidos correctamente. Añada el código que falta:

```

protected bool FormularioCorrecto
{
    get
    {
        bool bCorrecto = true;

        // Volvemos a dejar en blanco los comentarios
        et_Nombre.Text = "";
        et_Autor.Text = "";
        et_Archivo.Text = "";

        // Comprobamos el formulario
        if (sa_SubirArchivo.FileName.Length == 0)
        {
            et_Archivo.Text = "* Error: Seleccione un archivo!!";
            bCorrecto = false;
        }

        // Comprobar también el Nombre y el Autor
        // ...
    }
}

```

```
        return bCorrecto;
    }
}
```

Usando la propiedad anterior, modifique el método *bt\_Añadir\_Click* para que suba el archivo y escriba en la base de datos los datos recogidos en el formulario sólo cuando el formulario esté correctamente rellenado.

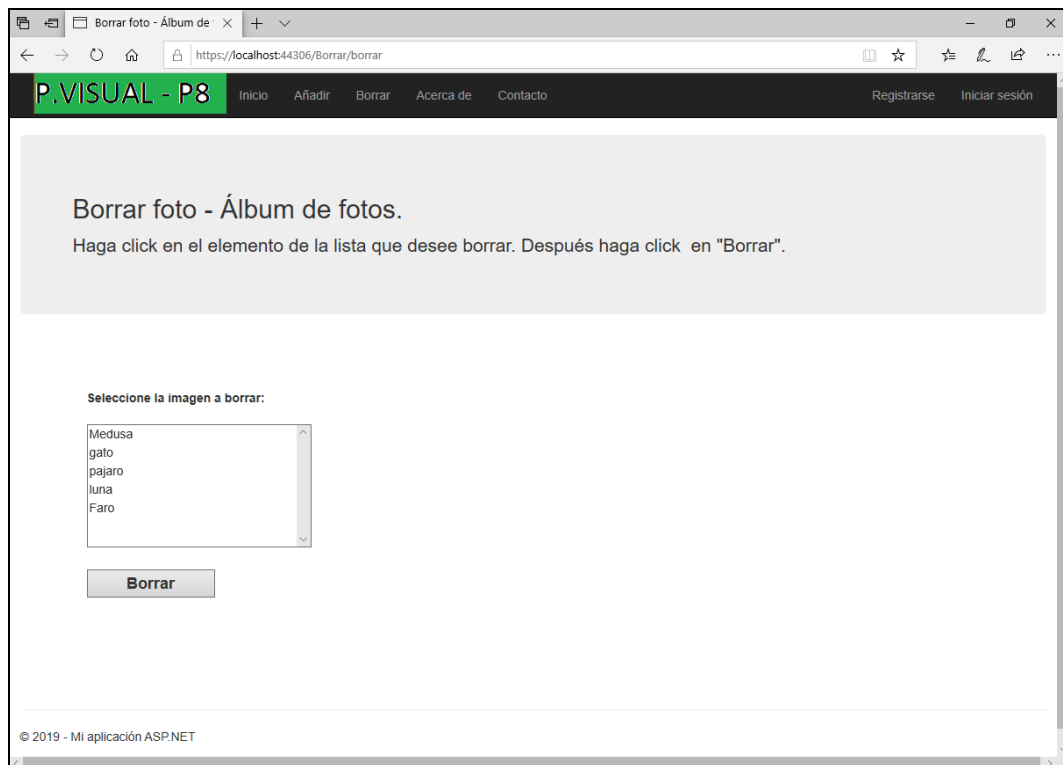
**Compile y ejecute para ver el resultado.**

## 12.4 Borrar Fotos

Añada otra página web al sitio web (elemento de tipo *Formulario Web con página maestra*) que utilice como plantilla la página maestra *Site.master*; llámela *borrar.aspx*, póngale un título y añada la pestaña correspondiente, “Borrar”, al menú de navegación. Cree una estructura para esta página análoga a *añadir.aspx*.

Inserte en ella un objeto de tipo **SqlDataSource** y configúrelo: elija la cadena de conexión que ya hay creada y cree la sentencia *select* para que seleccione los campos “nombre” y “foto” de la tabla “Album” de la base de datos. Estos campos serán utilizados por el **ListBox** que va a insertar a continuación (el campo “nombre” lo utilizará para mostrar las opciones del **ListBox** y el campo “foto” como valor de las opciones seleccionadas).

Añada un objeto **ListBox**, *ls\_Nombres*, y configúrelo de forma que se enlace con el origen de datos introducido en el paso anterior, para que muestre los nombres de las fotos y *para que tenga* como valor la entrada *foto* de la tabla *álbum* de la base de datos. Finalmente, ponga debajo de la lista un botón con el texto “Borrar” y llámelo *bt\_Borrar*. El aspecto deberá ser parecido al siguiente:



Añada el método *bt\_Borrar\_Click* que responda al evento **Click** del botón borrar y complételo con el código siguiente, añadiendo lo que falta:

```
protected void bt_Borrar_Click(object sender, EventArgs e)
{
    // La carpeta raíz del sitio web es "~/\"
    // ...

    // Borramos la imagen de la carpeta fotos
    File.Delete(sRutaSitioWeb + ls_Nombres.SelectedValue);

    // Borramos la ruta de la imagen de la base de datos
    SqlConnection conexión = ...; // ver clase ConfigurationManager

    string sOrdenSQL = "DELETE from album where foto = @FOTO";

    SqlCommand ordenSQL = ...;
    ordenSQL.Parameters.AddWithValue(...);

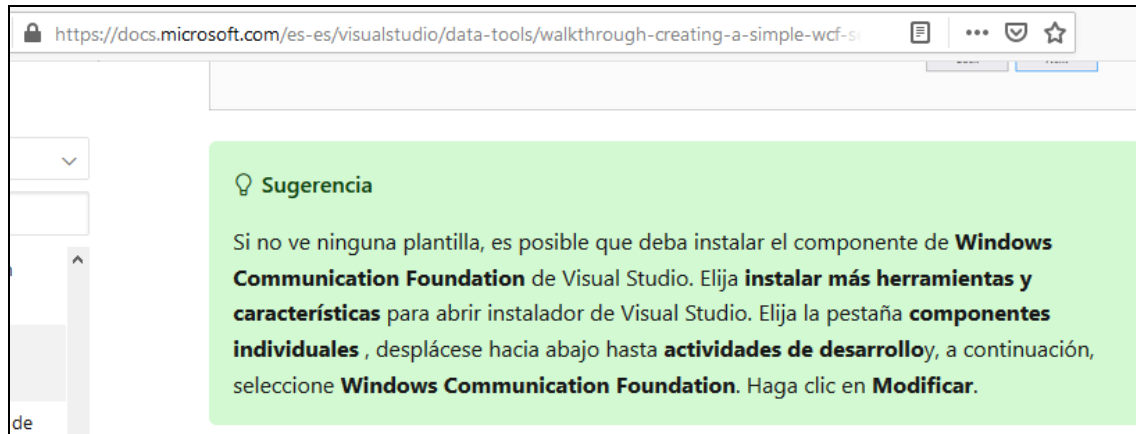
    // ...

    this.Response.Redirect("Default.aspx");
}
```

## 12.5 Creación de un servicio web y su cliente

En este apartado, vamos a crear un servicio web con el fin de separar la presentación, de la capa de acceso a datos (lógica de negocio). El servicio web se encargará de acceder a la base de datos para borrar o añadir registros. Desde el Sitio Web se sustituirán estas operaciones por una llamada al servicio. Para ello, añade a la solución *Práctica8* un nuevo proyecto (clic con el botón secundario del ratón sobre el **nombre de la solución** >

Agregar > Nuevo proyecto > Visual C# > Tipo de proyecto: servicio > plantilla **Aplicación de servicios WCF**. Si no aparece esta plantilla es posible que necesite ampliar la instalación de *Visual Studio*. En ese caso, siga los siguientes pasos recomendados por Microsoft.



Puede buscar “WCF” en la ventana de búsqueda de Visual Studio y aparecerá el instalador. Una vez actualizado Visual Studio, agregue un proyecto **Aplicación de servicios WCF** a la solución y localícelo en una carpeta *ServicioWeb* de su sistema de archivos. Después, establezca este proyecto como *proyecto de inicio* (clic con el botón secundario del ratón sobre el nombre del proyecto > **Establecer como proyecto de inicio**).

Tendrá que especificar la cadena de conexión con la base de datos en el archivo de configuración *Web.config*. Puede copiarla desde el fichero *Web.config* del *SitioWeb* y situarla al comienzo del archivo *Web.config* del *ServicioWeb*, dentro de la etiqueta `<configuration>`.

Abra el archivo *IService1.cs*, contrato del servicio, y elimine las firmas de los métodos existentes y el contrato de datos que no vamos a utilizar. Se quedará solamente la interfaz *IService1* vacía. A continuación, añada a la interfaz *IService1* las firmas de los métodos indicadas a continuación:

```
[OperationContract]
void BorrarFoto(string foto);

[OperationContract]
void AñadirFoto(string foto, string nombre,
               string autor, string descripcion);
```

Abra el archivo *Service1.svc.cs* elimine los métodos existentes y defina los métodos especificados en el contrato:

```
public void BorrarFoto(string foto)
{
    // Borramos de la base de datos la ruta de la foto
    // seleccionada tal y como lo hacíamos en el sitio web
```

```
}

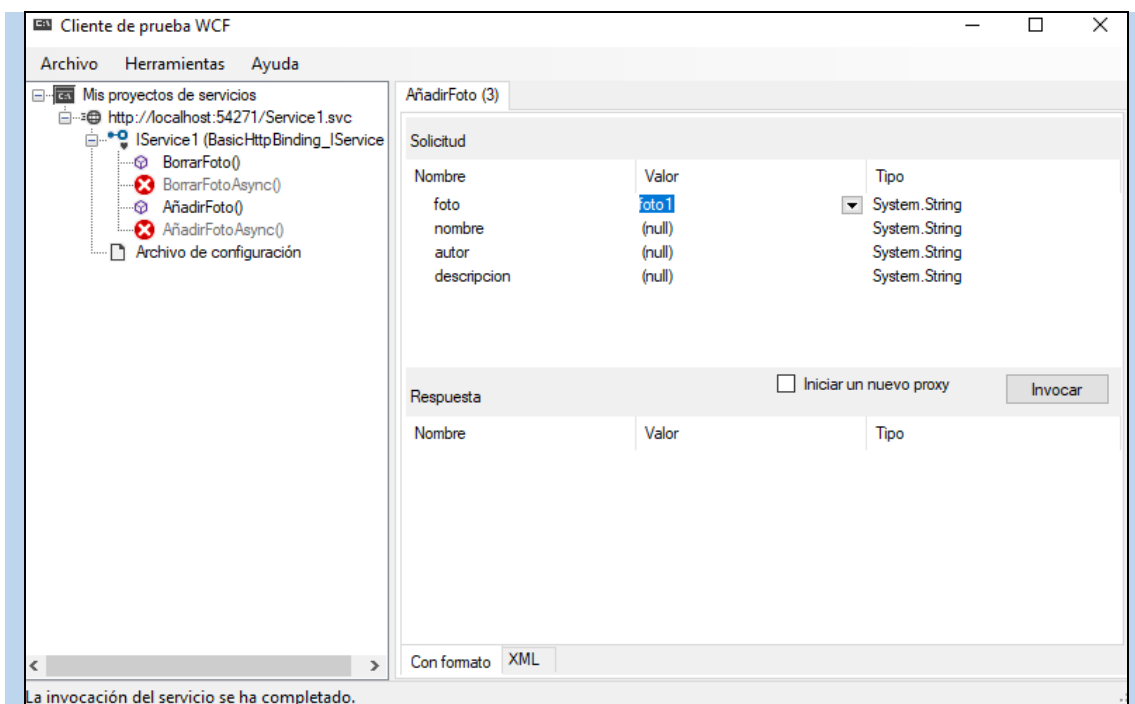
public void AñadirFoto(string foto, string nombre,
                      string autor, string descripcion)
{
    // Añadimos una entrada en la base de datos tal y como lo
    // hacíamos en el sitio web
}
```

Estos métodos sólo proporcionarán el servicio que permitirá añadir o borrar los registros de la base de datos vinculados con las fotos; esto es, *AñadirFoto* no subirá los archivos al servidor y ninguno de ellos redireccionará a la web principal, ya que estas operaciones serán realizadas por el sitio web que acceda a este servicio.

Abra en el editor el fichero *Service1.svc* (clic con el botón secundario > *Abrir con...*).

### Compile y ejecute el servicio web para ver el resultado.

Pruebe los métodos anteriores, añadiendo entradas a la base de datos y eliminándolas a través de la interfaz que crea VS (doble click en los métodos *AñadirFoto()* y *BorrarFoto()* en el menú de la izquierda, asignar un valor en la columna *Valor* y pulsar *Invocar*.). Para borrar una entrada, deberá introducir los datos que introdujo en el campo *foto*.



Puede mostrar los datos de la tabla a través del *Explorador de objetos de SQL Server* y verificar que las filas se añaden y se borran.

**Nota:** Una vez ejecutado el servicio web tome nota del URL en la parte superior izquierda de la interfaz del servicio web; por ejemplo, en el caso de la imagen anterior sería:

`http://localhost:54271/Service1.svc`

Esto es necesario porque cuando se crea el servicio web sobre el sistema de archivos, VS utiliza el servidor integrado en ASP.NET, no IIS. Por lo tanto, no cierre esta instancia de cliente de prueba WCF para que no se cierre el servidor.

Para desarrollar la interfaz que hace uso del servicio, vuelva a establecer el proyecto *SitioWeb* como *proyecto de inicio*. Después, haga clic con el botón derecho del ratón sobre el nombre del proyecto *SitioWeb* y seleccione la orden *Agregar referencia de servicio*; se muestra el diálogo *Agregar referencia de servicio*. Escriba en la caja *Dirección* la ruta en la que almacenó su servicio web (`http://...`), haga clic en el botón *Ir* y asigne en la caja *Espacio de nombres* el nombre *AlbumFotos*.

*[Nota: Si por algún motivo ajeno a la práctica, tuviera que cerrar VS entre este punto y el apartado 12.7 (incluido este último), al ejecutarlo de nuevo lo primero que se debe hacer es establecer el servicio web como proyecto de inicio, compilar, abrir en el editor el fichero Service1.svc (clic con el botón secundario > Abrir con...) y ejecutar para dejarlo abierto. Después, al volver al sitio web, que ya podrá comunicar con el servicio sin problema. La referencia al servicio web no es necesario crearla de nuevo.]*

Modifique el método *bt\_Borrar\_Click* de *borrar.aspx.cs* así:

```
protected void bt_Borrar_Click(object sender, EventArgs e)
{
    // La carpeta raíz del sitio web es "~/\"
    string sRutaSitioWeb = ...;
    // Borrar la imagen
    // ...
    AlbumFotos.Service1Client servicio = new AlbumFotos.Service1Client();
    servicio.BorrarFoto(ls_Nombres.SelectedValue);

    this.Response.Redirect("Default.aspx");
}
```

El método ha  
funcionado, llamándolo  
desde el sitioweb.

Si se rechaza expresamente la  
conexión deberá hacerse lo siguiente:

Crear un nuevo método exactamente  
igual, abrir su svc.cs y .svc, para  
obtener la ventana de prueba al  
servicio.

Modifique el método *bt\_Añadir\_Click* de forma análoga para que añada una foto al álbum, haciendo uso del servicio implementado.

Una vez llegados aquí solo habría  
que poner de nuevo el sitio web  
como inicio de proyecto y llamar a  
los métodos.

**Compile y ejecute la aplicación web para ver el resultado.**

## 12.6 Autenticación, autorización y roles

Cuando Visual Studio creó el sitio web ya creó una carpeta *Account* con las páginas para trabajar con la identidad de un usuario y otra *App\_Data* para almacenar la base de datos de los usuarios. Por lo tanto, ejecute el sitio web, haga clic en *iniciar sesión* e intente registrar un nuevo usuario.

Eche ahora una ojeada a la base de datos que almacena la identidad de los usuarios para ver su composición y a las cadenas de conexión en *Web.config*. En la cadena de conexión de la base de datos de usuarios (llamada *aspnet-SitioWeb*) podrá comprobar que esta base de datos está ubicada en el servidor local, *localDb*, a diferencia de la base



de datos *fotos* que se encuentra en el servidor *sqlexpress*. Puede usar como referencia el número que aparece en el nombre de la bbdd de usuarios para buscarla en el servidor local y realizar una consulta que compruebe que el usuario con el que se acaba de registrar se ha añadido correctamente.

La aplicación deberá estar configurada para que todos los usuarios, autenticados o no, puedan ver las fotos, para que sólo los usuarios autenticados puedan añadir fotos y para que sólo los usuarios autenticados que tengan asignada la función de “Administradores” puedan borrar fotos. Una vez autenticada la identidad del usuario, el proceso de autorización determina si tiene acceso a un recurso específico.

La autorización resulta sencilla de implementar si organiza las páginas en carpetas. De esta forma los usuarios, dependiendo de si se han autenticado o no y dependiendo de su función (rol), podrán acceder a unos u otros recursos. Según esto, añada dos nuevas carpetas, *añadir* y *borrar*, y aloje las páginas correspondientes en las mismas.

Esta nueva organización en carpetas le obligará a modificar algunas rutas, por ejemplo, en el menú de la página maestra o en las redirecciones (una ruta a una página es de la forma *~/.../página.aspx*; por ejemplo *~/Default.aspx*; *~* representa a la carpeta del sitio).

La autorización para acceder o no a un recurso se consigue incluyendo en la carpeta que contiene ese recurso un archivo de configuración secundario denominado *Web.config* (Botón derecho del ratón desde la carpeta *añadir/borrar* y pulsar **Agregar/Nuevo elemento/Archivo de configuración web**). Un ejemplo del contenido de este archivo puede ser el siguiente:

```
<?xml version="1.0" encoding="utf-8"?>
<configuration>
  <system.web>
    <authorization>
      <deny users="?" />
    </authorization>
  </system.web>
</configuration>
```

El código anterior deniega el acceso a los usuarios no autenticados. Tomando este código como ejemplo, añada los ficheros de configuración para las carpetas *añadir* y *borrar*.

### Compile y ejecute el *sitio web* para ver el resultado.

En Visual Studio 2013 o superior, para crear usuarios y roles se utiliza *ASP.NET Identity*. El sistema de autorización no cambia, por lo que para crear las reglas de acceso escribiremos manualmente los archivos de configuración comentados anteriormente.

*ASP.NET Identity* es el sistema de identidad actual de ASP.NET que se introdujo con Visual Studio 2013. Está basado en la especificación OWIN (*Open Web Interface for .NET*) definida por miembros del propio grupo de ASP.NET, cuya finalidad es desacoplar la dependencia entre el host de la aplicación web y la propia aplicación. Recordemos que todo ASP.NET se apoyaba en **System.Web**, que es parte del *.Net Framework*, orientado a *WebForms*. Después surgió ASP.NET MVC, un *framework* de desarrollo web no orientado a *WebForms* y no incluido en el monolito *.Net Framework*, lo que permitió una evolución completamente independiente, aunque su ejecución

seguía dependiendo de **System.Web**, lo que obligaba a alojar una aplicación web en IIS, consecuencia del acoplamiento existente entre la aplicación web, *.Net Framework* y el servidor de aplicaciones. De esta experiencia nace OWIN con la intención de crear la mayor independencia posible entre esos componentes de un desarrollo web, así como facilitar la autenticación por medio de proveedores externos muy conocidos, como lo son Google y FaceBook, utilizados por millones de personas en todo el mundo.

Cuando Visual Studio creó el sitio web, creó una serie de carpetas y una serie de archivos; nos fijaremos en dos archivos: *IdentityModels.cs* y *Startup.cs*. El primero contiene una serie de clases destinadas a la administración de usuarios y el segundo, una clase *Startup* que define el punto de entrada a la aplicación. Abra el archivo, *IdentityModels.cs*, y copie en el mismo el código que se le ha proporcionado en el archivo de recursos junto con el enunciado de esta práctica, para añadir funcionalidad adicional para administrar usuarios y roles. Esta funcionalidad adicional añade la clase *userManager* (si es que no está), modifica la clase *ApplicationUser* y añade las clases *ApplicationRole* y *RoleManager*, del espacio de nombres *SitioWeb.Models*, si *SitioWeb* fue el nombre elegido cuando creó el proyecto.

Ahora, utilizando la funcionalidad aportada por las clases anteriores crearemos un usuario “admin” con perfil “Administradores” y contraseña “admin.p8” cuando se inicie la aplicación. Para ello, complete el método *Configuration* de la clase *Startup* así:

```
public void Configuration(IApplicationBuilder app)
{
    ConfigureAuth(app);

    // Crear el rol "Administradores" si no existe.
    ApplicationRole rolAdmin = null;
    if (!ApplicationRole.FindRole("Administradores"))
        rolAdmin = ApplicationRole.CreateRole("Administradores");
    // Crear un usuario y añadirlo al rol "Administradores".
    var user = ApplicationUser.CreateUser("admin@uah.es", "admin.p8");
    if (user != null)
    {
        // Escriba aquí la línea de código que añada
        // el usuario "user" al rol "rolAdmin".
    }
}
```

### Compile y ejecute el *sitio* web para ver el resultado.

Si todo ha ido bien, habrá observado que la base de datos se ha reconstruido y tiene un solo usuario, el añadido en el código anterior. Esto sucedió porque las nuevas clases añadidas modificaron el modelo de entidades correspondiente al contexto *ApplicationDbContext*. Si ahora añade nuevos usuarios, observará que todo funciona como esperaba.

#### 12.6.3 Ocultar opciones del menú

Como la pestaña “Borrar” sólo está disponible para los administradores, haga que se muestre sólo cuando el usuario autenticado sea un administrador. Para ello:

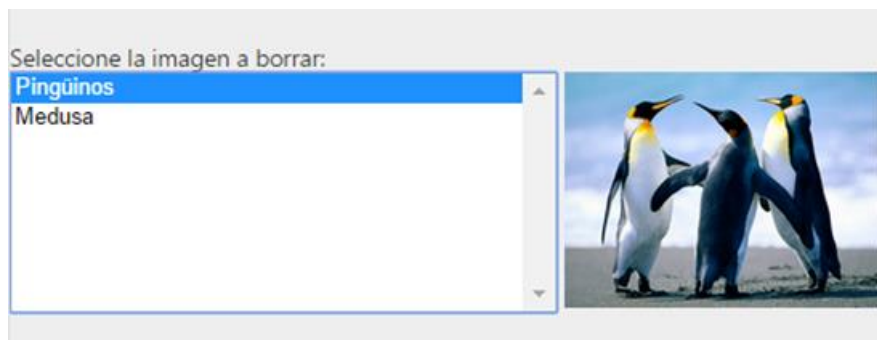
```
<li><a id="liBorrar" runat="server" href="~/borrar/borrar.aspx"
visible="false">Borrar</a></li>
```

Y cuando se cargue página *Site.master* (evento **Load**):

```
if (HttpContext.Current.User.IsInRole("Administradores"))
{
    liBorrar.Visible = true;
}
```

## 12.7 AJAX

Añada en la página *borrar.aspx*, a la derecha de la lista de fotos, un control **Image** que muestre una versión en miniatura de la foto seleccionada de la lista:



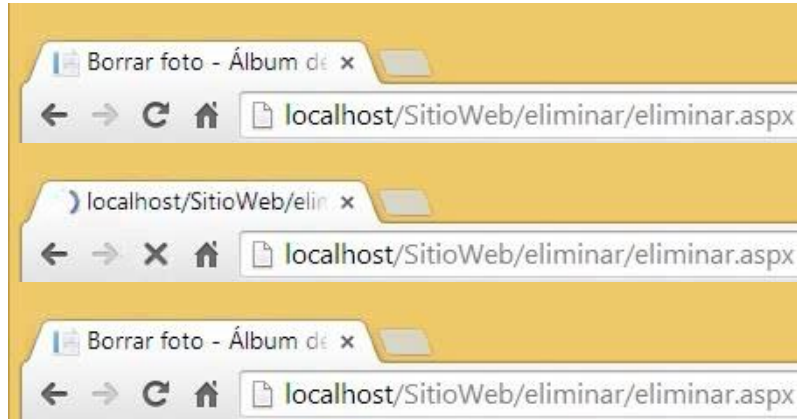
Para que la imagen no presente problemas de alineación con el ListBox utilice como contenedor un elemento de tipo `<table>` donde el ListBox y la imagen deben estar contenidos en la misma fila y distinta columna. Las etiquetas HTML mostradas a continuación representan una tabla de una fila y dos columnas:

```
<table>
  <tr> ← Comienzo de fila
    <td> ← Comienzo de columna
      ...
    </td>
    <td>
      ...
    </td>
  </tr>
</table>
```

Configure también el evento correspondiente para que se muestre la foto seleccionada en el ListBox en cada momento. Tenga en cuenta que para evitar sobrecargar el servidor, los eventos se gestionan por defecto en el momento de la pulsación del botón, por lo que la imagen no se visualizaría hasta que se pulsara el botón. Para que el evento se gestione en el momento de la selección de la imagen, active `AutoPostBack` en la lista de propiedades del ListBox.

**Compile y ejecute el sitio web para ver el resultado.**

Observe la pestaña “Borrar foto” de la página web mostrada en el explorador. El icono animado (icono situado a la izquierda de la pestaña) indica que se recarga totalmente la página cada vez que selecciona una imagen de la lista.



Para que la interacción con esta página sea fluida, utilice un control **UpdatePanel** de ASP.NET AJAX para que sólo se actualice el control **Image** sin necesidad de que se recargue toda la página cada vez que selecciona una imagen de la lista. Pista: el contenido del control **UpdatePanel** será sólo la imagen (utilice la etiqueta `<ContentTemplate>` asociada a un `UpdatePanel` para delimitar la imagen). Desde la ventana de propiedades definirá en su colección Triggers un *trigger asíncrono* para interceptar el evento que la lista genera cuando se selecciona un nuevo elemento.

```
<asp:UpdatePanel . . . . .  
    <ContentTemplate>  
        <asp:Image . . . . .  
    </ContentTemplate>  
</asp:UpdatePanel>
```

**El resto de la práctica es voluntaria.**

## 12.8 Publicar el sitio web y el servicio web

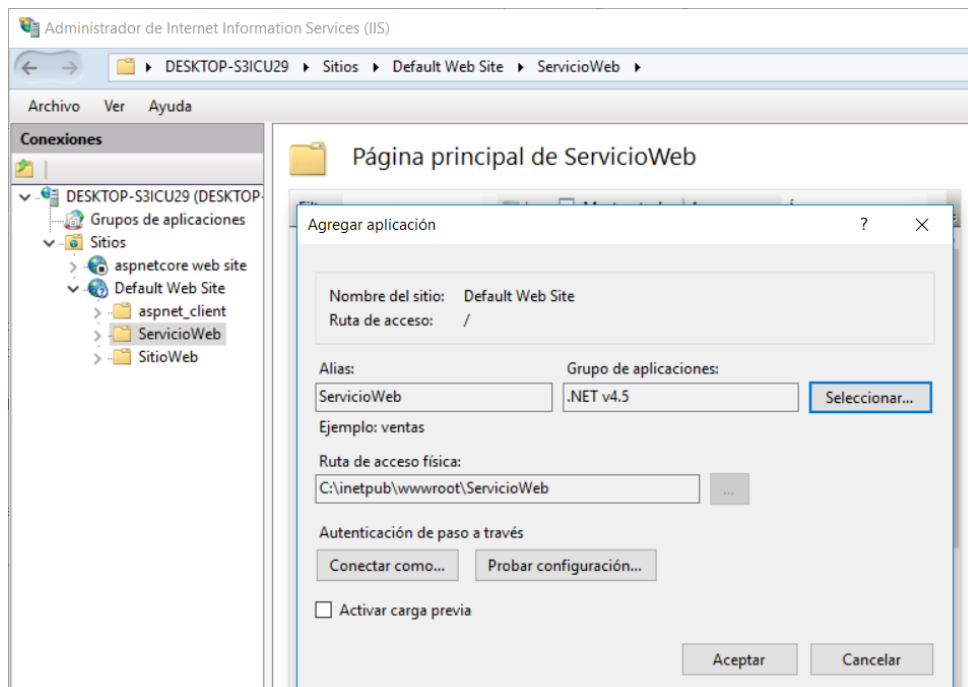
El hecho de haber creado tanto el servicio web como su cliente en el sistema de archivos local, impide acceder a ellos desde una máquina remota. En este apartado veremos cómo desplegar ambos componentes en el servidor de aplicaciones IIS de Windows.

Este apartado sólo podrá realizarlo si tiene instalado el servidor de aplicaciones IIS. Para comprobarlo, abra su explorador y acceda a la URL `http://localhost`. Si lo tiene instalado se le mostrará una página de presentación.

Si IIS no está instalado, instálelo siguiendo el documento *Instalación de IIS* publicado en la página web de la asignatura.

Si su sistema operativo no admite la instalación de IIS puede utilizar una máquina virtual con un sistema operativo que sí lo admita.

Cumplidos los requisitos anteriores, para la realización de este apartado deberá instalar en IIS (carpeta *c:\inetpub\wwwroot*) el servicio web y el sitio web construidos en los apartados anteriores. Copie las carpetas *ServicioWeb* y *SitioWeb* en la carpeta *c:\inetpub\wwwroot* y, por medio del administrador de IIS (puede llegar a él a través de inicio de Windows, escribiendo en la caja de búsqueda *inetmgr* o *IIS*), configúrelos para que funcionen correctamente. El primer paso es convertir ambas carpetas en aplicaciones de IIS del grupo ASP.NET v4.0 o superior: compruebe que desde *IIS* se visualizan las carpetas que copió (desplegando el menú situado a la izquierda (*sitios>Default Web Site*); a continuación, click en el botón derecho del ratón sobre cada una de las carpetas y seleccione *Convertir en aplicación* estableciendo *.NET v4.5* para el grupo de aplicaciones. Otra forma más segura, sería ejecutar la orden *Publicar sitio web* del menú *Compilar*.



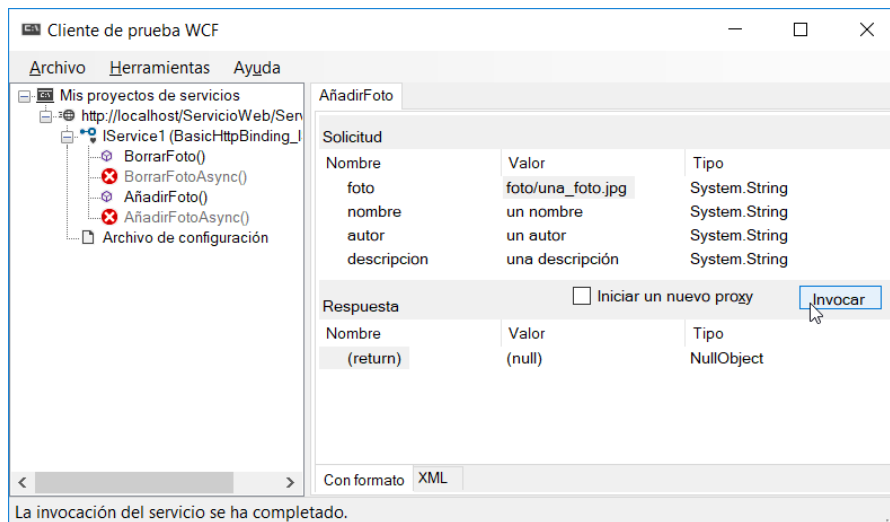
Habilite para el *servicio web* la autenticación anónima y para el *sitio web* la autenticación anónima y mediante formularios. A continuación, vamos a probar el funcionamiento primero del servicio y después el sitio web.

Abra el servicio web desde el entorno de desarrollo ejecutado como administrador, si es que no lo tiene abierto (*Continuar sin código* -> *Archivo* > *Abrir* > *Sitio web*).



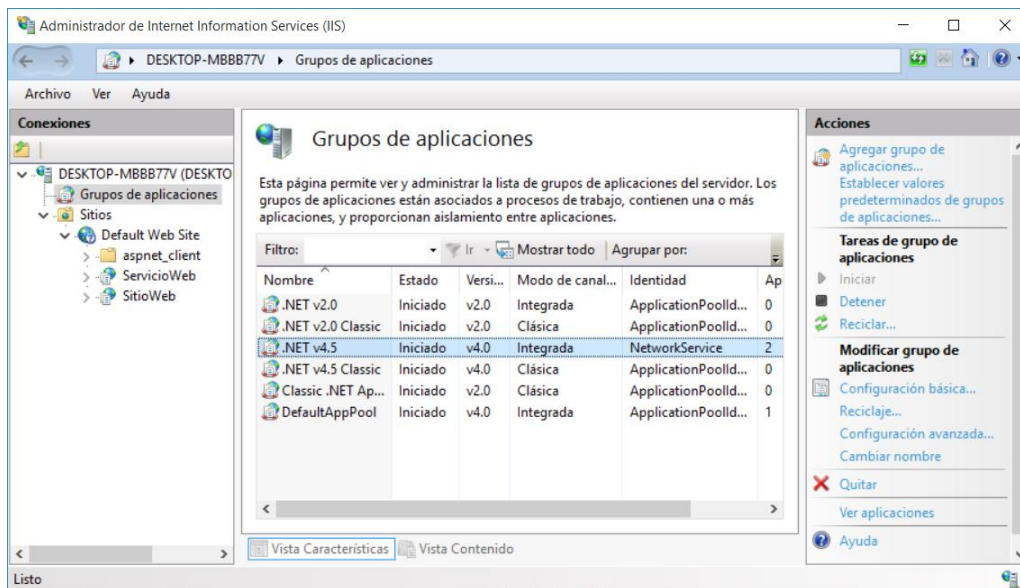
Abra en el editor el fichero *Service1.svc*.

**Compile y ejecute el servicio web para ver el resultado.**

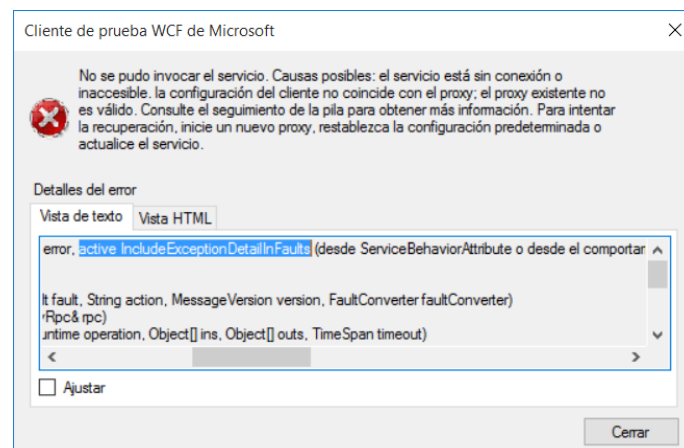


Pruebe los servicios *AñadirFoto* y *BorrarFoto* que ofrece el servicio WCF.

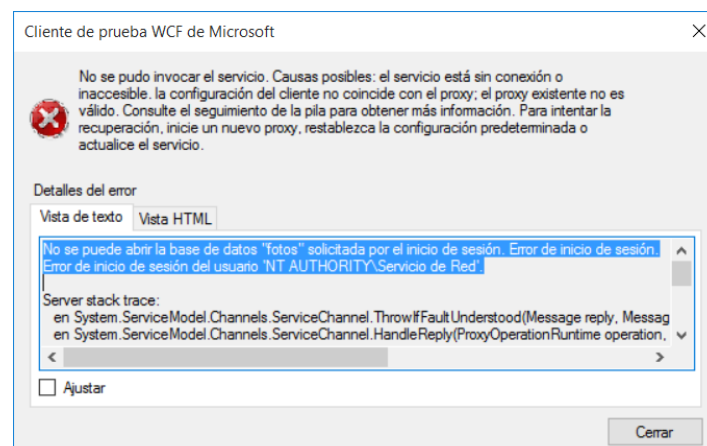
Si al hacer clic en el botón *Invocar* se muestra un error, supervise que la aplicación pertenece al grupo de aplicaciones adecuado (diríjase al *Administrador de IIS* y haga clic con el botón secundario del ratón en el nodo *ServicioWeb*; después, seleccione *Administrar aplicación* > *Configuración avanzada* > *Grupo de aplicaciones* > *.NET 4.5*), supervise también las características del grupo de aplicaciones seleccionado, mediante el menú situado en la parte superior izquierda de la pantalla (especialmente la *Identidad*; puede utilizar como identidad *NetworkService*):



Vuelva a ejecutar la aplicación. Seguramente mostrará de nuevo una página de error cuando intente invocar cualquiera de las dos operaciones ofrecidas por el *servicio web*, por ejemplo, como el de la figura siguiente:



Para obtener información acerca del error, realice las operaciones indicadas en el mensaje: abra el archivo de configuración de la aplicación, *web.config*, y lea los comentarios de la sección *<behavior>*, y proceda en consecuencia. Después vuelva a ejecutar la aplicación. Si se muestra este otro error:





*No se puede abrir la base de datos “fotos” solicitada por el inicio de sesión.  
Error de inicio de sesión del usuario “NT AUTHORITY\Servicio de Red”.*

El usuario bajo cuya identidad se ejecuta la aplicación no está autorizado para acceder a la base de datos. Para solucionarlo, deberá dar permisos al usuario especificado en el mensaje de error para que pueda acceder a la base de datos (rol *db\_owner*). Puede hacerlo mediante un script (puede verlo al final del script que utilizó para generar la base de datos). Si la ejecución del script le reporta un mensaje indicando que el usuario *NT AUTHORITY\Servicio de Red* no existe en la base de datos, pruebe a ejecutar de nuevo el script, pero en esta ocasión cambie el usuario especificado en el script por el especificado en el mensaje de error que generó el servicio web; léase también el apartado “Obtener acceso a los recursos de ASP.NET” del capítulo “Formularios web”) o mediante la utilidad *SQL Server Management Studio*.

Asegúrese también de que el servicio web puede ser accedido desde IIS (no desde el servidor local de ASP.NET). Para ello, abra su explorador y escriba la dirección:

<http://localhost/ServicioWeb/Service1.svc>

A continuación, abra el sitio web desde el entorno de desarrollo (ejecute VS como administrador), si es que no lo tiene abierto (*Continuar sin código -> Archivo > Abrir > Sitio web > carpeta del sitio web*).

Borre en el sitio web la carpeta, *Connected Services*, correspondiente a la referencia de servicio establecida anteriormente y establézcala de nuevo al servicio web ubicado ahora en IIS.

Posiblemente será notificado de que este proyecto usa *SQL Server Express Local DB* para administrar la base de datos de usuarios y que con IIS se recomienda utilizar *SQL Server Express*. Acepte esta recomendación.

### **Compile y ejecute el sitio web para ver el resultado.**

Si aparece el siguiente error de compilación “*No se guardó la solución. Guarde la solución antes de administrar los paquetes NuGet*”, vaya a *Archivo* pulse *Guardar todo* y compile de nuevo.

Asegúrese de que el sitio web sea accedido desde IIS (no desde el servidor local de ASP.NET). Para ello, abra su explorador y escriba la dirección:

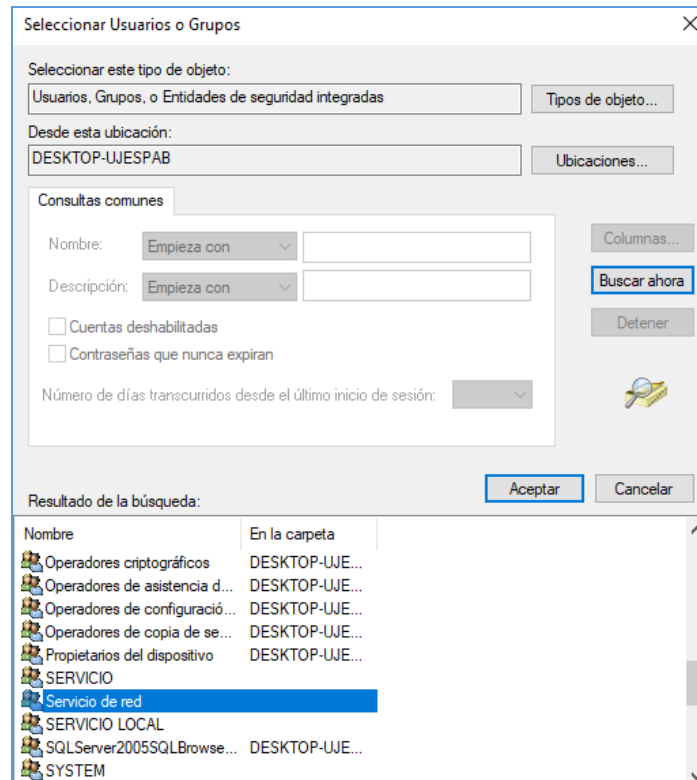
<http://localhost/SitioWeb/Default.aspx>

Solucione cualquier otro error que se le pueda presentar. Por ejemplo, relativo al grupo de aplicaciones, a la ruta donde se deben almacenar las fotos, a los permisos de la carpeta *App\_Data* donde debe almacenarse la base de datos, o de la carpeta *fotos* donde deben almacenarse las fotos, etc.

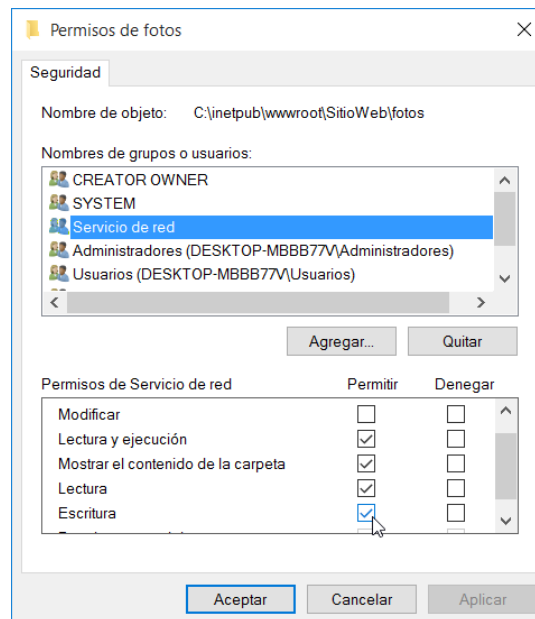
Por ejemplo, si al ejecutar la aplicación desde su explorador se produce un error “Acceso denegado a la ruta ...”, seguramente será porque el usuario utilizado por ASP.NET no está autorizado para obtener acceso al recurso solicitado: carpeta



*App\_Data* o *fotos*. ASP.NET tiene una identidad de proceso base (normalmente {MACHINE}\ASPNET en IIS 5 o Network Service en IIS 6 e IIS 7, y la identidad del grupo de aplicaciones configurada en IIS 7.5 y superiores). Para conceder a este usuario acceso a una carpeta o a un archivo, edite desde el explorador de Windows los permisos de la carpeta. Por ejemplo, haga clic con el botón secundario del ratón en la carpeta o en el archivo, elija *Propiedades* y seleccione la ficha *Seguridad*. Haga clic en *Editar* > *Agregar* (después en *Opciones avanzadas* > *Buscar ahora*) para agregar el usuario o grupo adecuado (*Servicio de red*). Seleccione la cuenta de ASP.NET y active las casillas según el acceso deseado (escritura o *también control total*).



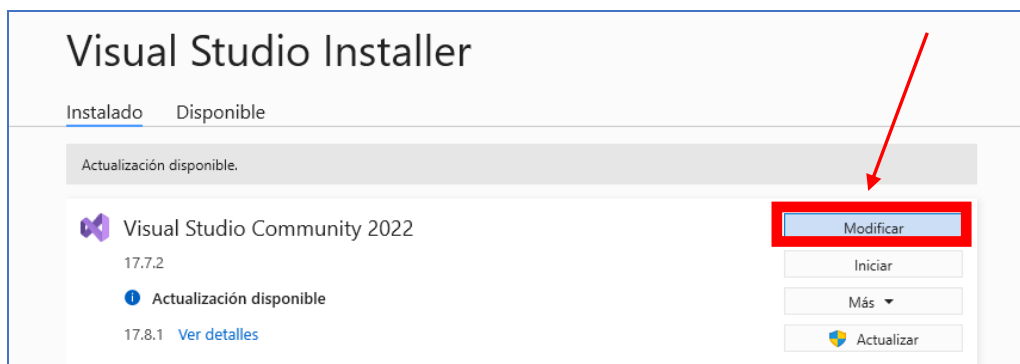
También puede realizar este proceso desde el administrador de IIS, seleccionando la carpeta de la que se quieren modificar los permisos.

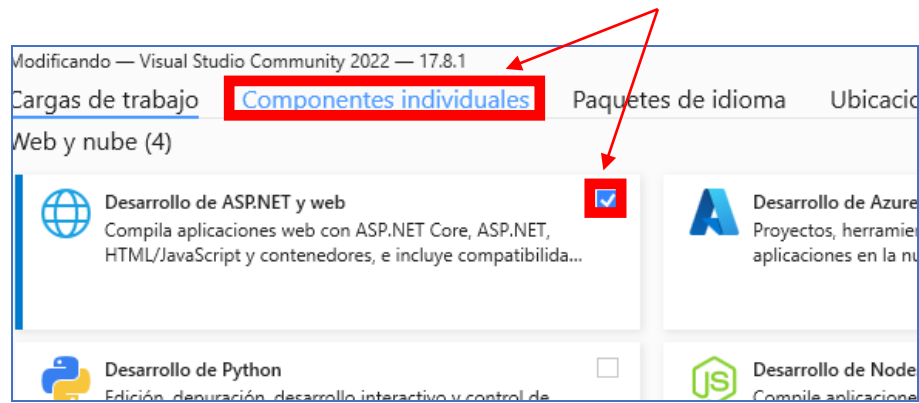


Finalmente, pruebe a registrar nuevos usuarios para verificar que esta operación también se puede realizar. Si copió las aplicaciones en *wwwroot*, probablemente obtendrá un error (acceso denegado) porque el archivo de base de datos es de sólo lectura o bien, simplemente no se puede acceder a la base de datos del usuario. Si es así, proceda con la carpeta *App\_Data* (y con el fichero de base de datos, si procede) de la misma forma que lo hizo con la carpeta *fotos*. Este sistema de seguridad se haría extensible a otras carpetas si fuera necesario.

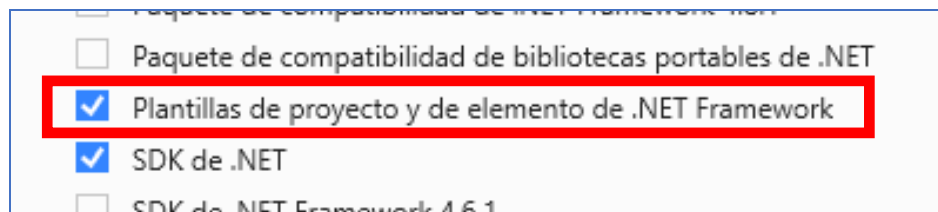
## ANEXO

Ejecute Visual Studio Installer y pulse *Modificar*.





Desactive y vuelva a activar la opción *Desarrollo de ASP.NET y web* y a continuación seleccione la pestaña *Componentes Individuales*.



Seleccione la opción *Plantillas de proyecto y de elementos de .NET Framework* o bien en idioma inglés, si fuera el caso, *.NET Framework Project and item templates* y pulse *Modificar*.

Ya podrá utilizar Visual Studio y seleccionar el tipo de proyecto *Aplicación web ASP.NET (.NET Framework)*.