

Tema 11. AJAX en .net

Herramientas Avanzadas para el Desarrollo de Aplicaciones

1

Introducción a AJAX

AJAX= Asynchronous JavaScript and XML

- Cura para “enfermedades comunes” de las aplicaciones basadas en navegador
 - Usa el objeto **XmlHttpRequest** para recuperar datos del servidor de forma asíncrona y **JavaScript** para actualizar el contenido de la página.
 - Elimina el parpadeo y usa el ancho de banda de manera más eficiente.
- Se utiliza en Google Suggests, Google Maps, Microsoft Virtual Earth, Outlook Web Access... etc

http DESVENTAJAS

- Período demasiado largo al refrescar la página
- El usuario debe esperar a que la petición (request) finalice para seguir interactuando con la aplicación.
- SOLUCIÓN: Manejo de la petición de un modo asíncrono
- ***Llamadas asíncronas que no bloquean la interfaz de usuario mientras se producen.***

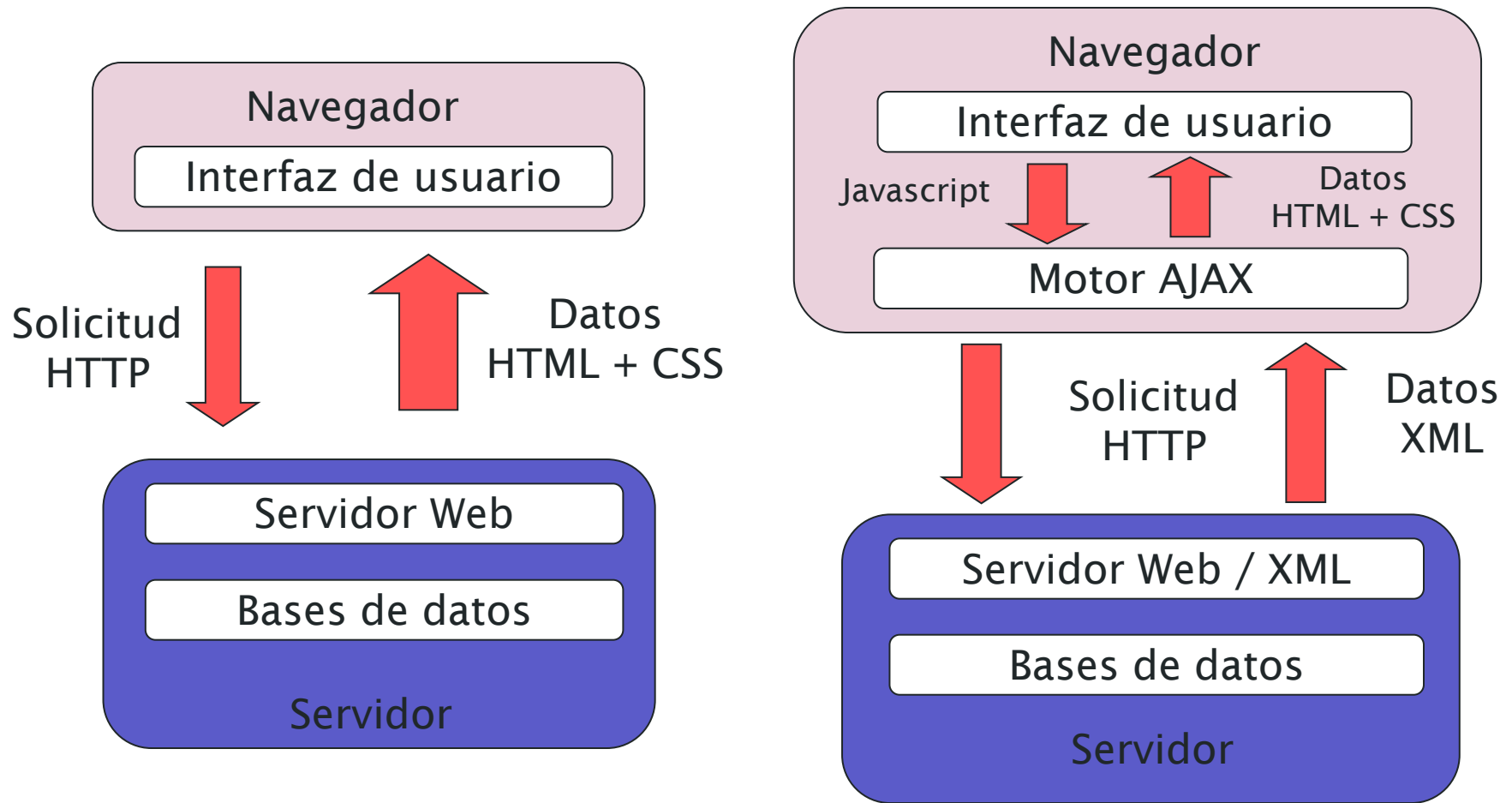


- Tecnología de Microsoft para desarrollar aplicaciones Web basadas en AJAX
- Provee librerías script en el cliente que incorporan tecnologías **JavaScript** y HTML dinámico (**DHTML**), y las integra con la plataforma de desarrollo basada en **ASP.NET**
- También incluye componentes básicos en el servidor para soportar esas llamadas asíncronas del cliente
- Utilizando ASP.NET AJAX, se puede **mejorar la experiencia de usuario y la eficiencia de las aplicaciones Web.**

¿Por qué usar AJAX?

- **Eficiencia mejorada** ejecutando partes significantes de una página Web en el navegador.
- **Elementos de IU familiares**, típicas de una aplicación de escritorio, como ventanas pop-up, indicadores de progreso y tooltips.
- **Actualizaciones parciales de la página** que refrescan sólo las partes de la página Web que han cambiado.
- **Soporte para los navegadores más populares** y utilizados, como Microsoft Internet Explorer, Mozilla Firefox, y Apple Safari.

Modelo Clásico de aplicaciones web vs AJAX



Una aplicación Ajax

- Se elimina la naturaleza *start-stop-start-stop* de la interacción de la Web introduciendo un intermediario — **un motor Ajax** — entre el usuario y el servidor.
- En lugar de cargar una página Web, al inicio de la sesión, el navegador carga el **motor Ajax** — escrito en JavaScript , usualmente en un marco oculto.

Motor Ajax

- Este motor es responsable de presentar la interfaz al usuario y de comunicar con el servidor de parte del usuario.
- Permite que la interacción del usuario con la aplicación sea **asíncrona**.
- **Cada acción del usuario** que normalmente generaría una petición HTTP se convierte en una **llamada JavaScript al motor Ajax**.

Motor Ajax (II)

- Cualquier respuesta a una acción del usuario que no requiera volver al servidor es manejada por el motor.
 - como validación de datos simple, editar datos en memoria, y incluso alguna navegación
- Si el motor necesita algo del servidor para elaborar la respuesta el motor hace estas peticiones de manera **asíncrona**, usualmente utilizando XML, sin paralizar la interacción del usuario con la aplicación.

ASP.Net

2

Creación de una
aplicación Web
AJAX

Controles de servidor ASP.net AJAX

- Consisten en código de cliente y de servidor que se integran para producir el comportamiento AJAX.
- **ScriptManager**
 - Registra el script de la librería de microsoft AJAX en la página. Esto permite al script de cliente admitir características como la representación parcial de páginas y las llamadas a servicios web. El control **ScriptManager** es necesario para utilizar los demás controles.

?

X

Nuevo proyecto

▶ Reciente

◀ Instalado

▶ Plantillas

▶ Visual Basic

▶ Visual C#

Escritorio de Windows

◀ Web

Visual Studio 2012

▶ Office/SharePoint

Cloud

LightSwitch

Prueba

Reporting

Silverlight

WCF

Workflow

▶ Visual C++

▶ Visual F#

SQL Server

Python

TypeScript

▶ Otros tipos de proyectos

Proyectos de modelado

▶ En línea

.NET Framework 4.5

Ordenar por: Predeterminado

Buscar en la Plantillas instalado (Ctrl+E)

Aplicación web ASP.NET

Visual C#

Tipo: Visual C#

Una plantilla de proyecto para crear aplicaciones ASP.NET. Puede crear aplicaciones ASP.NET Web Forms, MVC o Web API y agregar muchas otras características en ASP.NET.

Haga clic aquí para buscar plantillas en línea.

Nombre: WebApplication2

Ubicación: c:\users\irene\documents\visual studio 2013\Projects

Nombre de la solución: WebApplication2

Examinar...

☒ Crear directorio para la solución

☐ Agregar al control de código fuente

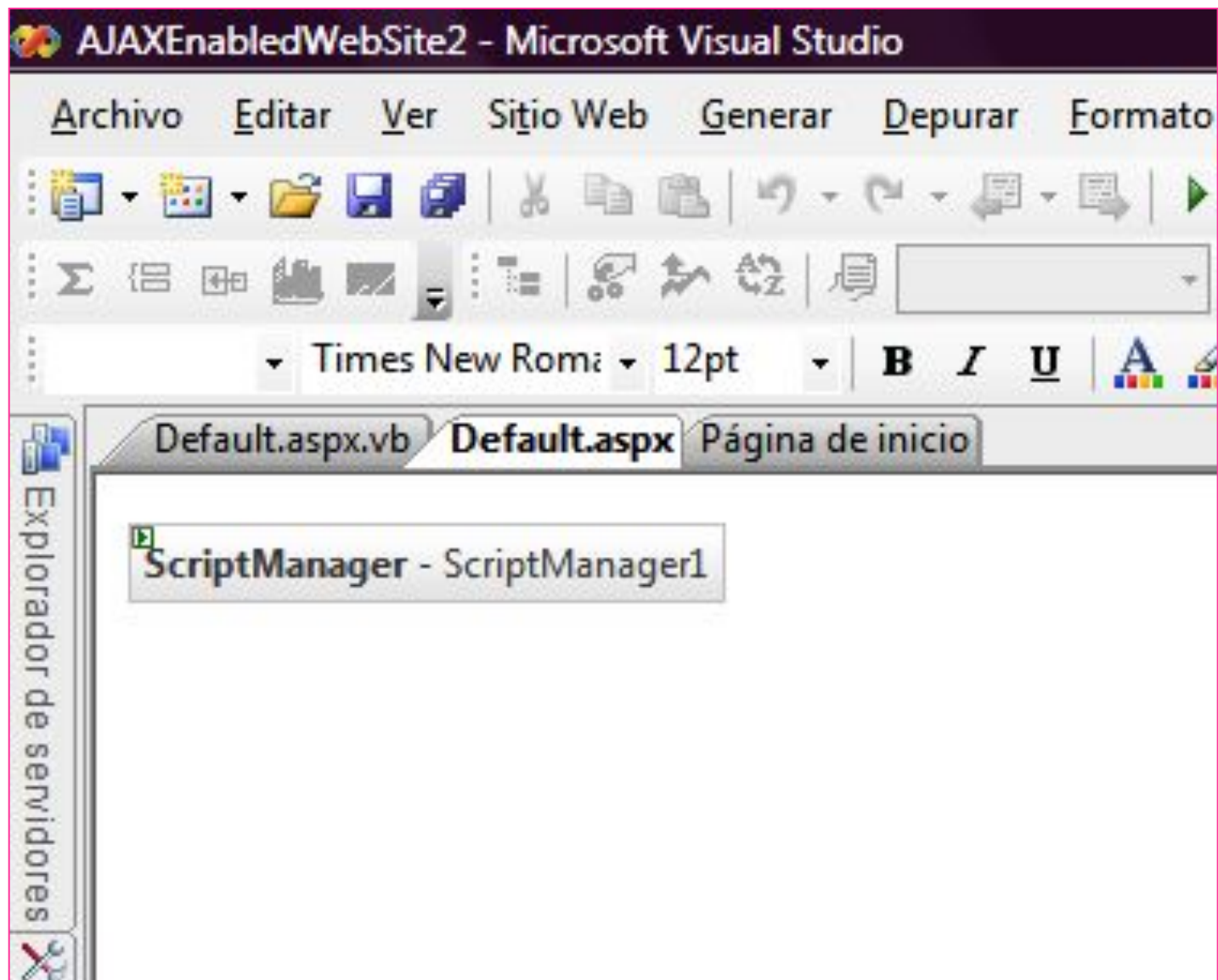
Aceptar

Cancelar

Default.aspx

```
<%@ Page Language="VB" AutoEventWireup="true" CodeFile="Default.aspx.vb" Inherits="_Default" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN" "http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title>Untitled Page</title>
</head>
<body>
    <form id="form1" runat="server">
        <asp:ScriptManager ID="ScriptManager1" runat="server" />
        <div>
        </div>
    </form>
</body>
</html>
```



Controles AJAX

- UpdatePanel
 - Permite refrescar las partes seleccionadas de la página, en lugar de refrescar la página completa utilizando un postback síncrono.
- UpdateProgress
 - Provee información de estado sobre actualizaciones parciales de la página en controles UpdatePanel.
- Timer
 - Ejecuta postbacks en intervalos de tiempo definidos. Puede utilizarse para enviar la página completa, o usarse junto al control UpdatePanel para realizar actualizaciones parciales de la página en un intervalo definido.

Ejercicio

- ▶ En un formulario Web vamos a crear dos listas desplegadas (listBox) a las que vamos a insertar elementos desde un cuadro de texto.
- ▶ La primera lista se actualizará SIN recarga de la página.
- ▶ La segunda lista se actualizará normalmente.

Añadir elemento

elemento 1

añadir sin recarga

Añadir con recarga

no se recarga la página...

elemento 1

se recarga la página...

elemento 1

Código asociado

```
protected void Button1_Click(object sender, EventArgs e)
```

```
{
```

```
}
```

```
protected void Button2_Click(object sender, EventArgs e)
```

```
{
```

```
}
```

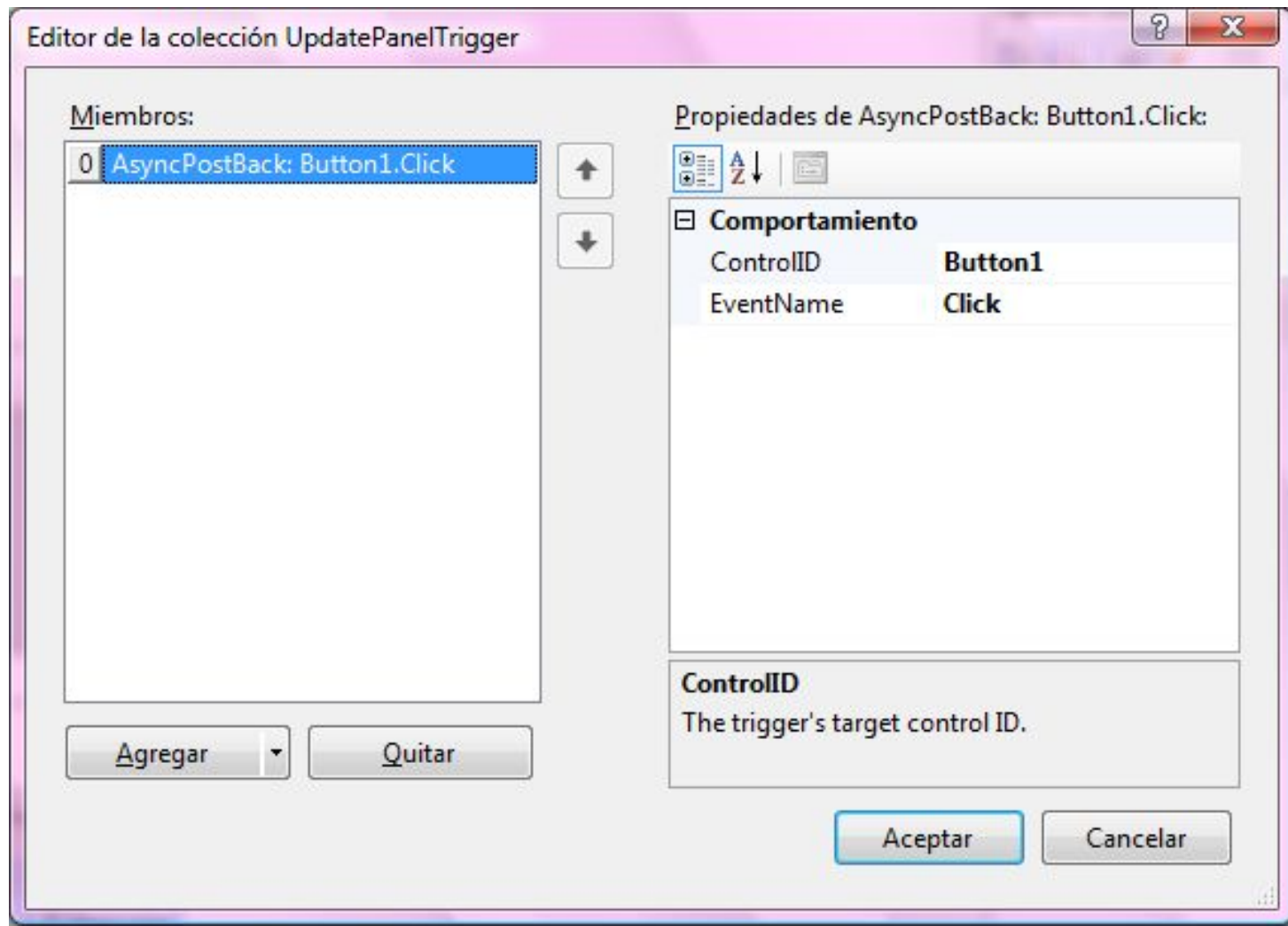
Código asociado

```
protected void Button1_Click(object sender, EventArgs e)
{
    ListBox1.Items.Add(TextBox1.Text);
}
```

```
protected void Button2_Click(object sender, EventArgs e)
{
    ListBox2.Items.Add(TextBox1.Text);
}
```

Update Panel

- Propiedad Triggers



Ejercicio



Ejercicio

- ▶ Vamos a añadir ahora un control UpdateProgress.
- ▶ En este control añadimos el texto “Actualizando...” y la imagen indicator.gif (cv).
- ▶ Vamos a añadir un retardo de 2 segundos para hacerlo realista.

Añadir elemento

elemento 2

añadir sin recarga

Añadir con recarga

Actualizando... 

no se recarga la página...

elemento 1

se recarga la página...

elemento 1

Código a añadir...

```
protected void Button1_Click(object sender, EventArgs e)
{
    ...
    System.Threading.Thread.Sleep(2000);
}
```


Código a añadir...

```
protected void Button1_Click(object sender, EventArgs e)
{
    ListBox1.Items.Add(TextBox1.Text);
    System.Threading.Thread.Sleep(2000);
}
```

ASP.Net

ASP.net Ajax
Control Toolkit

3

ASP.NET AJAX Control Toolkit

- Está desarrollado en base a ASP.NET AJAX y contiene una serie de controles Web y extendedores con los que podremos utilizar las avanzadas características de ASP.NET AJAX .

ASP.NET AJAX Control Toolkit

- Vamos a distinguir entre controles Web y extendedores, donde los primeros tienen una entidad por sí mismos, mientras que los segundos únicamente añaden un comportamiento a un control Web existente.
- Estos controles van desde un simple botón con una alerta asociada, hasta un complejo panel que podemos arrastrar por la pantalla;
 - en ambos casos, mandando y recogiendo información entre el cliente y el servidor sin ningún tipo de recarga de página.
- Su uso hará que nuestra Web sea mucho más atractiva, potente y efectiva.

Pasos instalación toolkit

1. Descargar el instalador en:
<https://www.devexpress.com/products/ajax-control-toolkit/>



What's New in AJAX Control Toolkit

Free to use. More Dependable than Ever Before

Pasos instalación toolkit

- Instalarlo (es un exe): lo que hace es añadir una librería DLL al visual studio con los controles nuevos

ASP.NET AJAX Control Toolkit v18.1.1



CodePlex BSD License Agreement

New BSD License (BSD)

Copyright (c) 2012-2018, CodePlex Foundation
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following

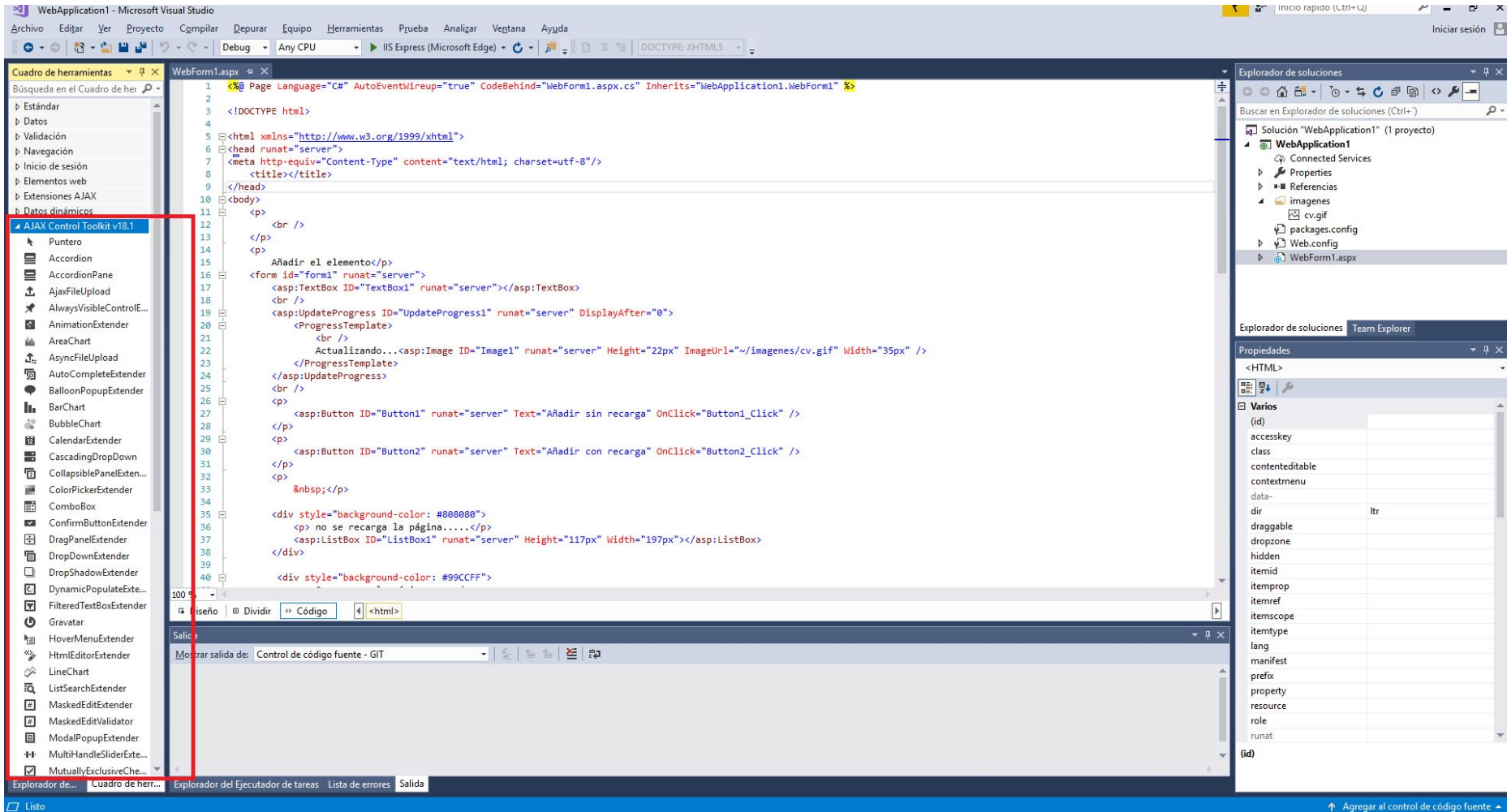
The [CodePlex BSD License Agreement](#) governs the use of this distribution.

I have read, understand and agree to all of its terms.

Accept

Decline

Comprobar instalación



EJEMPLOS DE CONTROLES AJAX

- <http://www.ajaxcontroltoolkit.com/>

AlwaysVisible extender

- El extendedor **AlwaysVisibleControl** permite colocar un control en la página Web que permanece visible y “flotante” sobre el contenido de la página mientras hacemos scroll o redimensionamos la ventana del navegador.
- Cualquier control puede tener asociado este extendedor.

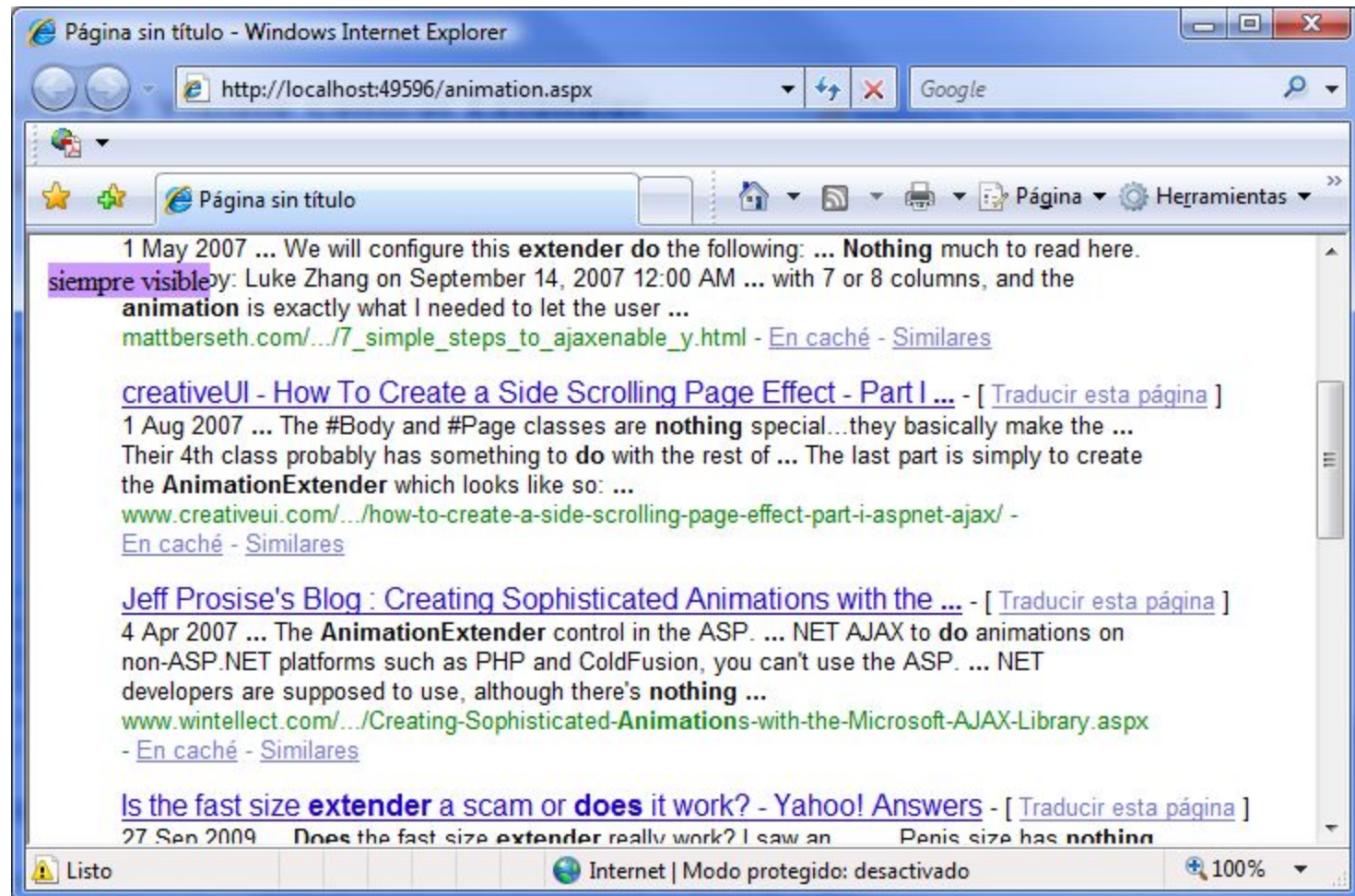
Exercise



Ejercicio

- Añadir un panel estándar, añadirle color de fondo y borde y el texto “siempre visible”.
- Copiar de internet una página web para que haya texto suficiente para hacer scroll.
- Añadir el extendedor alwaysvisible al panel.

AlwaysVisibleControl extender



código

```
<asp:ScriptManager ID="ScriptManager1" runat="server">  
</asp:ScriptManager>
```

```
<asp:Panel ID="Panel1" runat="server" BackColor="#CC99FF">  
    siempre visible  
</asp:Panel>
```

```
<asp:AlwaysVisibleControlExtender  
    ID="Panel2_AlwaysVisibleControlExtender"  
    runat="server" Enabled="True" TargetControlID="Panel1">  
</asp:AlwaysVisibleControlExtender>
```

CollapsiblePanel extender

- Con el CollapsiblePanel conseguiremos que cualquier control ASP.NET pueda ser maximizado o minimizado a nuestro antojo.
- Distinguiremos entre el contenido, que será el control que vaya a cerrarse y abrirse (por ejemplo un Panel) y el controlador, que será el control sobre el que deberemos hacer clic para cerrar y/o abrir el contenido.
- El estado del contenido (abierto o cerrado) es guardado a lo largo de los postbacks, por lo que permanecerá igual cuando recarguemos una página.

Ejercicio



Ejercicio

- Crear un formulario web con un panel estándar que será la cabecera, y otro panel que será el contenido a expandir/contrair.
- En la cabecera añadimos una imagen y una etiqueta que inicialmente dirá “(mostrar detalles...)”.
- En el contenido escribimos texto y añadimos una imagen. **IMPORTANTE, este panel tendrá height=0**
- Añadimos un collapsiblepanel extender sobre el panel de contenido

Vista diseño

asp:Panel#Panel2

Información sobre mi ciudad (Mostrar detalles...)



Mi nombre es Joe. Vivo en un pueblo a las afueras

blabla



blabla

blabla

Código: Panel Titulo

```
<asp:Panel ID="Panel2" runat="server" BackColor="#FFCCFF"
    BorderColor="#993399" BorderStyle="Dashed" Height="28px"
    Width="648px">
```

Información sobre mi ciudad

```
<asp:Label ID="Label1" runat="server" Text="Label">
```

(Mostrar detalles...)

</asp:Label>

```
<asp:Image ID="Image1" runat="server" Height="24px"
ImageUrl="~/flechader.jpg" Width="23px" />
```

</asp:Panel>

Código: Panel contenido

```
<asp:Panel ID="Panel3" runat="server" BackColor="#FF99FF"  
  Height="0px" Width="652px">
```

```
  Mi nombre es Joe. Vivo en un pueblo a las afueras<br />
```

```
  <br />
```

```
  blabla<br />
```

```
  <br />
```

```
  <asp:Image ID="Image2" runat="server" Height="177px"
```

```
    ImageUrl="~/pueblonevado.jpg" Width="273px" />
```

```
  <br />
```

```
  blabla<br />
```

```
  <br />
```

```
  blabla
```

```
</asp:Panel>
```

Código: CollapsiblePanel

```
<asp:CollapsiblePanelExtender  
  ID="Panel3_CollapsiblePanelExtender"  
  runat="server" Enabled="True"
```

```
  TargetControlID="Panel3"
```

```
  ExpandControlID="Panel2"
```

```
  CollapseControlID="Panel2" Collapsed=true
```

```
  TextLabelID="Label1" ExpandedText="Ocultar detalles"  
  CollapsedText="Mostrar detalles"
```

```
  ImageControlID=Image1 ExpandedImage="~/flechaiz.jpg"  
  CollapsedImage="~/flechader.jpg" SuppressPostBack=true>
```

```
</asp:CollapsiblePanelExtender>
```

Panel de contenido

Panel cabecera

Ejecución

Información sobre mi ciudad (Mostrar detalles)



Información sobre mi ciudad (Ocultar detalles)



Mi nombre es Joe. Vivo en un pueblo a las afueras

blabla



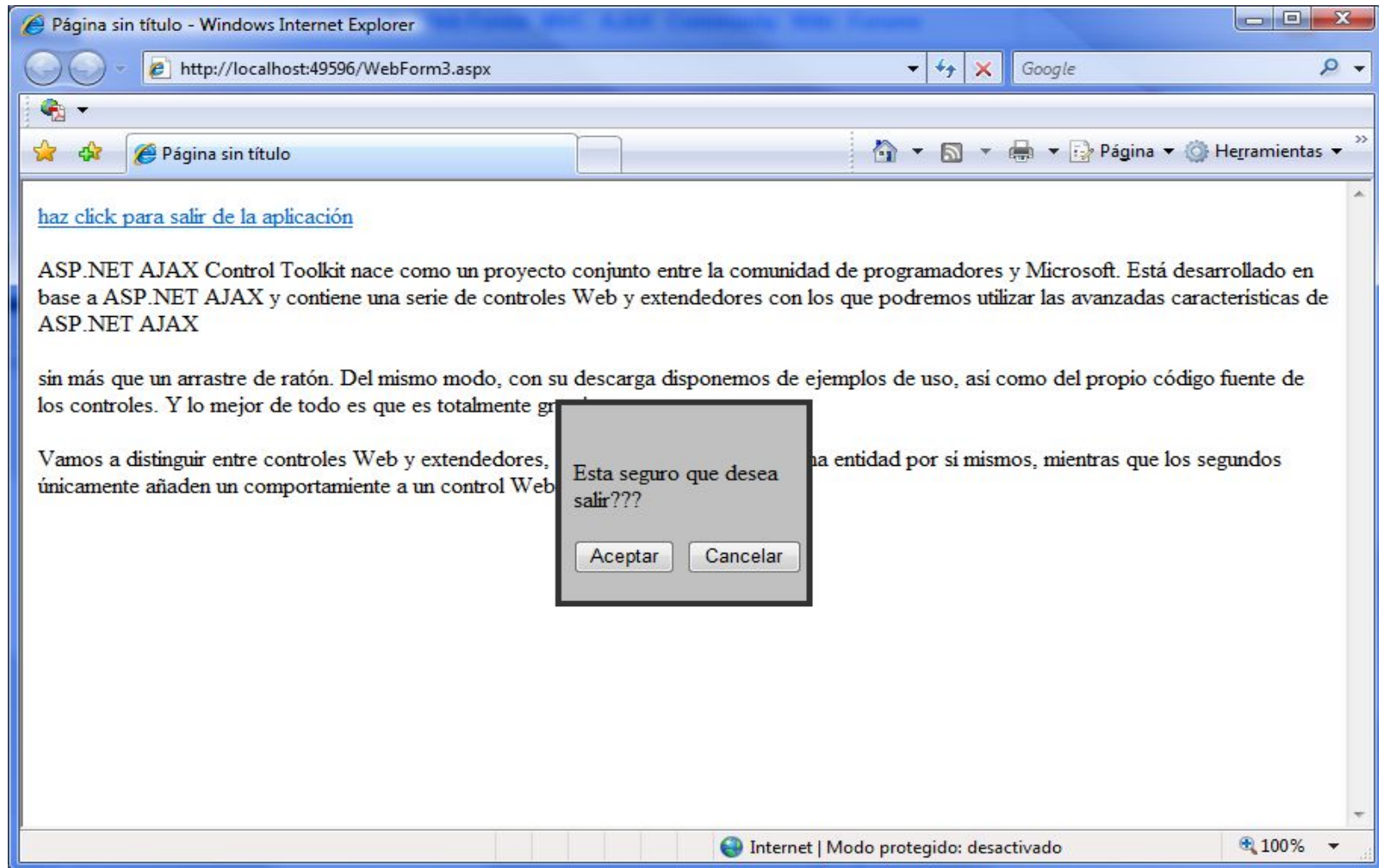
blabla

blabla

ModalPopup

- Con este extensor, conseguimos el efecto de mostrar contenido deshabilitando la interacción con el resto de la página.
- Podemos emular el efecto del famoso “`window.open(...)`” de javascript sin necesidad de salir de la página en que estamos ni de abrir una nueva ventana del navegador.

En ejecución



Código Panel

```
<asp:Panel ID="Panel1" runat="server" BackColor="Silver"
  BorderColor="#333333"
  BorderStyle="Solid" Width="162px" Style="display:none">
  <br />
  <br />
  &nbsp; Esta seguro que desea&nbsp;&nbsp;&nbsp;
  <br />
  &nbsp; salir???<br />
  <br />
  &nbsp;
  <asp:Button ID="Button1" runat="server" Text="Aceptar" />
  &nbsp;
  <asp:Button ID="Button2" runat="server" Text="Cancelar" />
  <br />
</asp:Panel>
```

Código

```
<asp:ModalPopupExtender  
ID="LinkButton1_ModalPopupExtender" runat="server"  
DynamicServicePath="" Enabled="True"  
TargetControlID="LinkButton1" PopupControlID="Panel1">  
</asp:ModalPopupExtender>
```


SlideShow extender

- Imágenes con slideshow



arbol

Prev Next Play



Ejercicio

- Crear un formulario web con los siguientes controles:
 - AjaxScriptManager
 - Control Imagen
 - Label
 - Tres botones (Prev, Next, Play)
 - Extensor SlideShow en la imagen

Código (I)

```
<asp:Image ID="Image1" runat="server" Height="300"  
    Style="border: 1px solid black; width: auto" ImageUrl="images/Creek.jpg"  
    AlternateText="Rio"/>
```

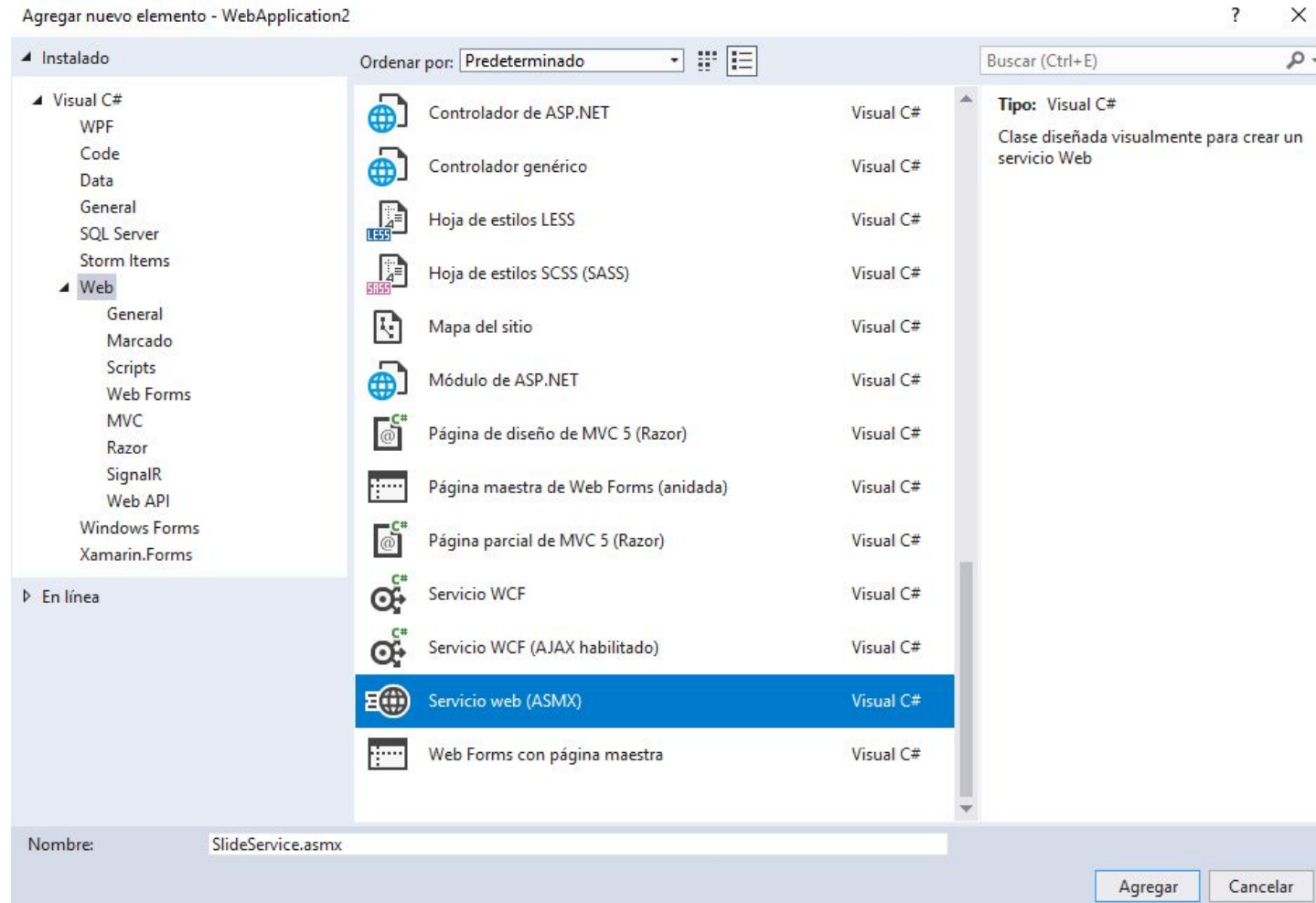
```
<asp:SlideShowExtender ID="Image1_SlideShowExtender" runat="server"  
    Enabled="True"  
    SlideShowServicePath="SlideService.asmx"  
    TargetControlID="Image1"  
    SlideShowServiceMethod="GetSlides"  
    AutoPlay="true"  
    ImageTitleLabelID="Label1"  
    NextButtonID="Button2"  
    PlayButtonID="Button3"  
    PreviousButtonID="Button1"  
    PlayButtonText="Play"  
    StopButtonText="Stop">  
</asp:SlideShowExtender>  
<br />
```

Necesitamos un
servicio Web

Código (II)

```
</div>  
  <asp:Label ID="Label1" runat="server"></asp:Label>  
  <br />  
  
  <asp:Button ID="Button1" runat="server" Text="Prev"  
Font-Size=Larger/>  
  
  <asp:Button ID="Button2" runat="server" Text="Next"  
Font-Size=Larger/>  
  
  <asp:Button ID="Button3" runat="server" Text="Play"  
Font-Size=Larger/>  
</div>
```

Añadir elemento Webservice



Añadir elemento Webservice

[\[System.Web.Script.Services.ScriptService\]](#)

```
public class SlideService : System.Web.Services.WebService
{
    [WebMethod]
    public AjaxControlToolkit.Slide[] GetSlides() {
        AjaxControlToolkit.Slide[] myslides=new AjaxControlToolkit.Slide[4];

        myslides[0]=new AjaxControlToolkit.Slide("images/Creek.jpg","Rio","Rio muy bonito");
        myslides[1] = new AjaxControlToolkit.Slide("images/Dock.jpg", "puerto", "puerto azul");
        myslides[2] = new AjaxControlToolkit.Slide("images/Garden.jpg", "jardin", "jardin margaritas");
        myslides[3] = new AjaxControlToolkit.Slide("images/Tree.jpg", "arbol", "arbol marron");

        return myslides;
    }
}
```

PasswordStrength

- El PasswordStrength es otro extendedor del TextBox.
- Con él podremos mostrar al usuario el nivel de fortaleza que tiene la contraseña que está escribiendo, en base a unos parámetros típicos de fortaleza que podemos configurar nosotros.



Ejercicio

- Crear un formulario web con los siguientes controles:
 - AjaxScriptManager
 - TextBox1
 - Label1
 - TextBox2
 - Label2
 - Extensor PasswordStrength a cada textbox

Diseño

AjaxScriptManager - AjaxScriptManager1

Indicadores de texto

[Label1]

Indicadores con barra estado

[Label2]

Ejecución

Indicadores de texto

Seguridad: Muy baja

8 more characters

Indicadores con barra estado



2 more characters

Código (I)

```
<asp:TextBox ID="TextBox1" runat="server"></asp:TextBox>
<asp:PasswordStrength ID="TextBox1_PasswordStrength" runat="server"
    Enabled="True" TargetControlID="TextBox1"
    DisplayPosition="RightSide"
    StrengthIndicatorType="Text"
    PreferredPasswordLength="10"
    PrefixText="Seguridad:"
    TextStrengthDescriptions="Muybaja;Debil;Media;Fuerte;Excelente"
    MinimumLowerCaseCharacters="0"
    MinimumSymbolCharacters="0"
    HelpStatusLabelID="Label1"
    RequiresUpperAndLowerCaseCharacters="false">
</asp:PasswordStrength>
</p>
<asp:Label ID="Label1" runat="server"></asp:Label>
```

Código (II)

```
<asp:TextBox ID="TextBox2" runat="server"></asp:TextBox>
<asp>PasswordStrength ID="TextBox2_PasswordStrength" runat="server"
    Enabled="True" TargetControlID="TextBox2"
    DisplayPosition="RightSide"
    StrengthIndicatorType="BarIndicator"
    PreferredPasswordLength="10"
    PrefixText="Seguridad:"
    TextStrengthDescriptions="Muy baja;Debil;Media;Fuerte;Excelente"
    MinimumNumericCharacters="1"
    MinimumSymbolCharacters="1"
    HelpStatusLabelID="Label2"
    RequiresUpperAndLowerCaseCharacters="true"
    BarBorderCssClass="barBorder"
    BarIndicatorCssClass="barInternal">
</asp>PasswordStrength>
```

CSS

```
<head>
  <style>
    .barBorder
    {
      border: solid 1px red;
      width: 300px;
    }
    .barInternal
    {
      background: yellow;
    }
  </style>
</head>
```

ConfirmButton

- Con el ConfirmButton conseguimos una sencilla funcionalidad. Lo asignaremos a un Button, LinkButton o HyperLink, de modo que cuando se haga clic sobre éste, el navegador nos muestre una ventana de confirmación.
 - Si se elige aceptar, el botón o enlace funciona normalmente.
 - De lo contrario, no se ejecutará el comportamiento definido. Opcionalmente se puede definir un script en el cliente llamándolo mediante la propiedad OnClientCancel.
- Útil para pedir confirmación al usuario al navegar a una página o borrar un link etc.

Código

```
<asp:Button ID="Button1" runat="server"  
    Text="Button"  
    onclick="Button1_Click" />
```

```
<asp:ConfirmButtonExtender  
    ID="Button1_ConfirmButtonExtender"  
    runat="server"  
    ConfirmText="Seguro que deseas aceptar?"  
    Enabled="True"  
    TargetControlID="Button1">  
</asp:ConfirmButtonExtender>
```

Al pinchar en el botón...

