

# Tema 13. Aplicaciones Web (V)

Herramientas Avanzadas para el Desarrollo de Aplicaciones

# ASP.Net

# *1*

## ASP.net Ajax Control Toolkit

# 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: [ajaxcontroltoolkit.codeplex.com](https://ajaxcontroltoolkit.codeplex.com)

The screenshot shows the CodePlex project page for the ASP.NET AJAX Control Toolkit. The browser's address bar shows the URL <https://ajaxcontroltoolkit.codeplex.com>, which is circled in red. The page features a navigation bar with tabs for HOME, SOURCE CODE, DOWNLOADS, DOCUMENTATION, ISSUES, PEOPLE, and LICENSE. Below the navigation bar, the main heading is "AJAX Control Toolkit (maintained by DevExpress)". The "Update October 2015:" section states that version v15.1.4 is now available and provides links to get the new installer from DevExpress.com or use the ASP.NET AJAX Control Toolkit Nuget package. A "download" button, also circled in red, is prominently displayed. To the right of the button, a table lists project details: CURRENT (ASP.NET AJAX Control Toolkit v15.1.4), DATE (Mon Oct 26, 2015 at 9:00 AM), STATUS (Stable), DOWNLOADS (16.645), and RATING (4 stars, 3 ratings). Below the table, there is a section for "MOST HELPFUL REVIEWS" with two reviews visible.

ASP.NET AJAX Control Toolkit

<https://ajaxcontroltoolkit.codeplex.com>

CodePlex Project Hosting for Open Source Software

Register Sign In Search all projects

DevExpress MAINTAINED BY ASP.NET AJAX Control Toolkit

HOME SOURCE CODE DOWNLOADS DOCUMENTATION ISSUES PEOPLE LICENSE

Page Info Change History (all pages) Follow (1514) Subscribe

AJAX Control Toolkit (maintained by DevExpress)

Update October 2015:

The ASP.NET AJAX Control Toolkit v15.1.4 is now available!

Get the new installer from DevExpress.com (provides Visual Studio toolbox integration).

Or use the ASP.NET AJAX Control Toolkit Nuget package.

Learn about the v15.1.4 release here:

Blog: ASP.NET AJAX Control Toolkit - v15.1.4 - Maintenance update available

Check out these helpful articles:

» Step-by-Step Installation Guide

Update August 2015:

The ASP.NET AJAX Control Toolkit v15.1.3 is now available!

Learn about the v15.1.3 release here: [ASP.NET AJAX Control Toolkit - v15.1.3 - Maintenance update available](#)

download

CURRENT	ASP.NET AJAX Control Toolkit v15.1.4
DATE	Mon Oct 26, 2015 at 9:00 AM
STATUS	Stable
DOWNLOADS	16.645
RATING	★★★★☆ 3 ratings

Review this release

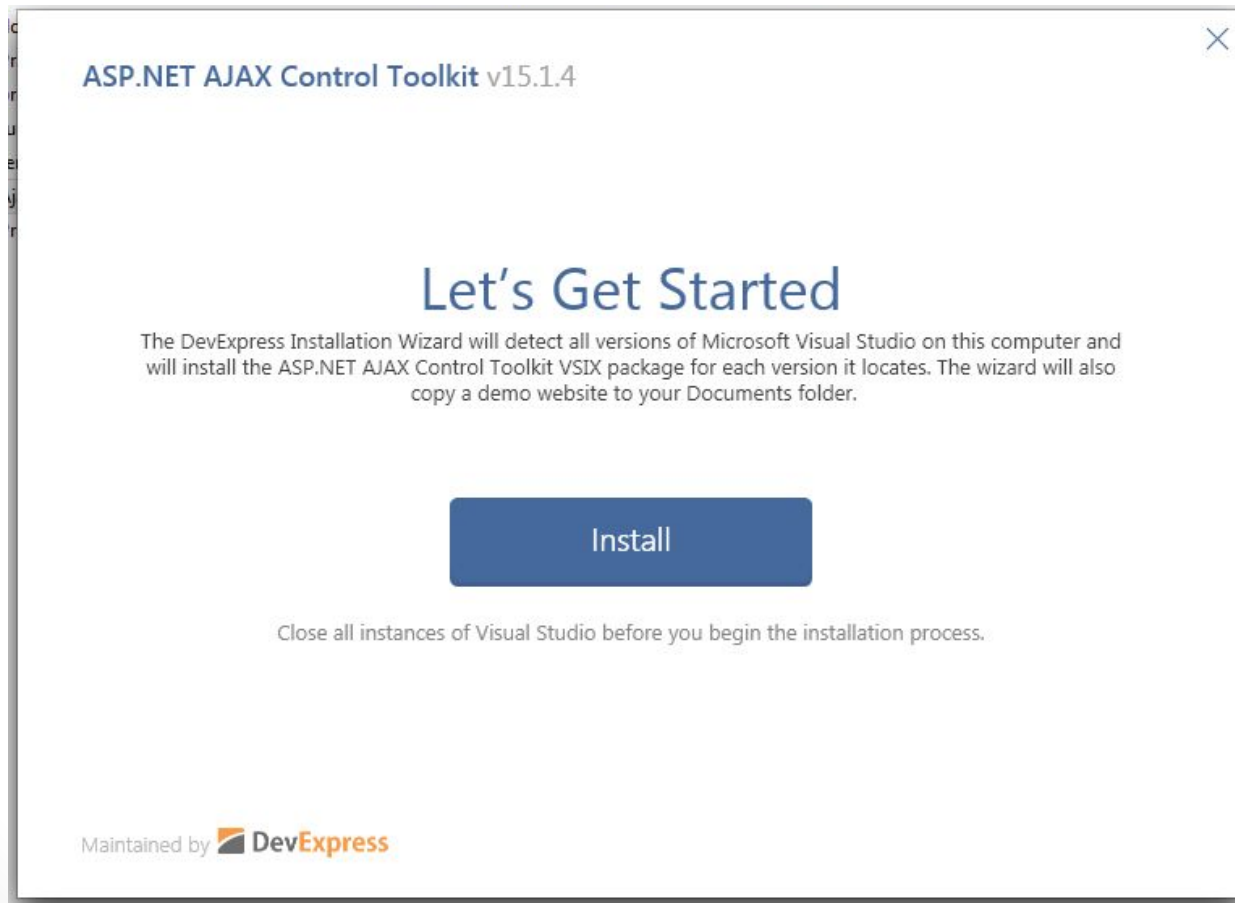
MOST HELPFUL REVIEWS

★★★★☆ Recently upgraded two web apps from .NET 2.0 to 4.5. Both were using ACT. The transition from erstwhile version of ACT to 15.1.4 was... (more)

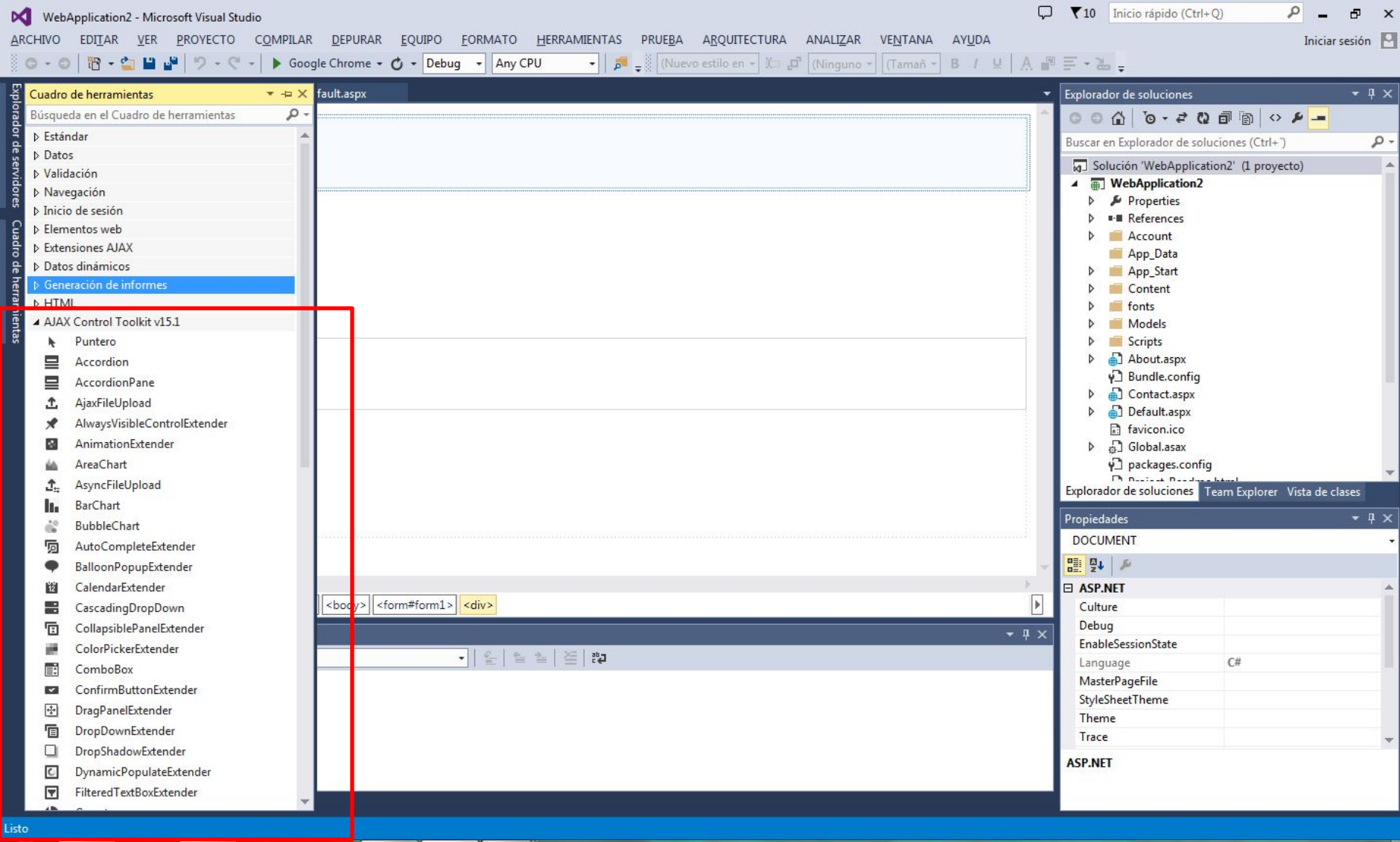
★★★★☆ An issue with MaskedEditExtender which was reported in 15.1.3, was answered as fixed

# Pasos instalación toolkit

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



# 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.

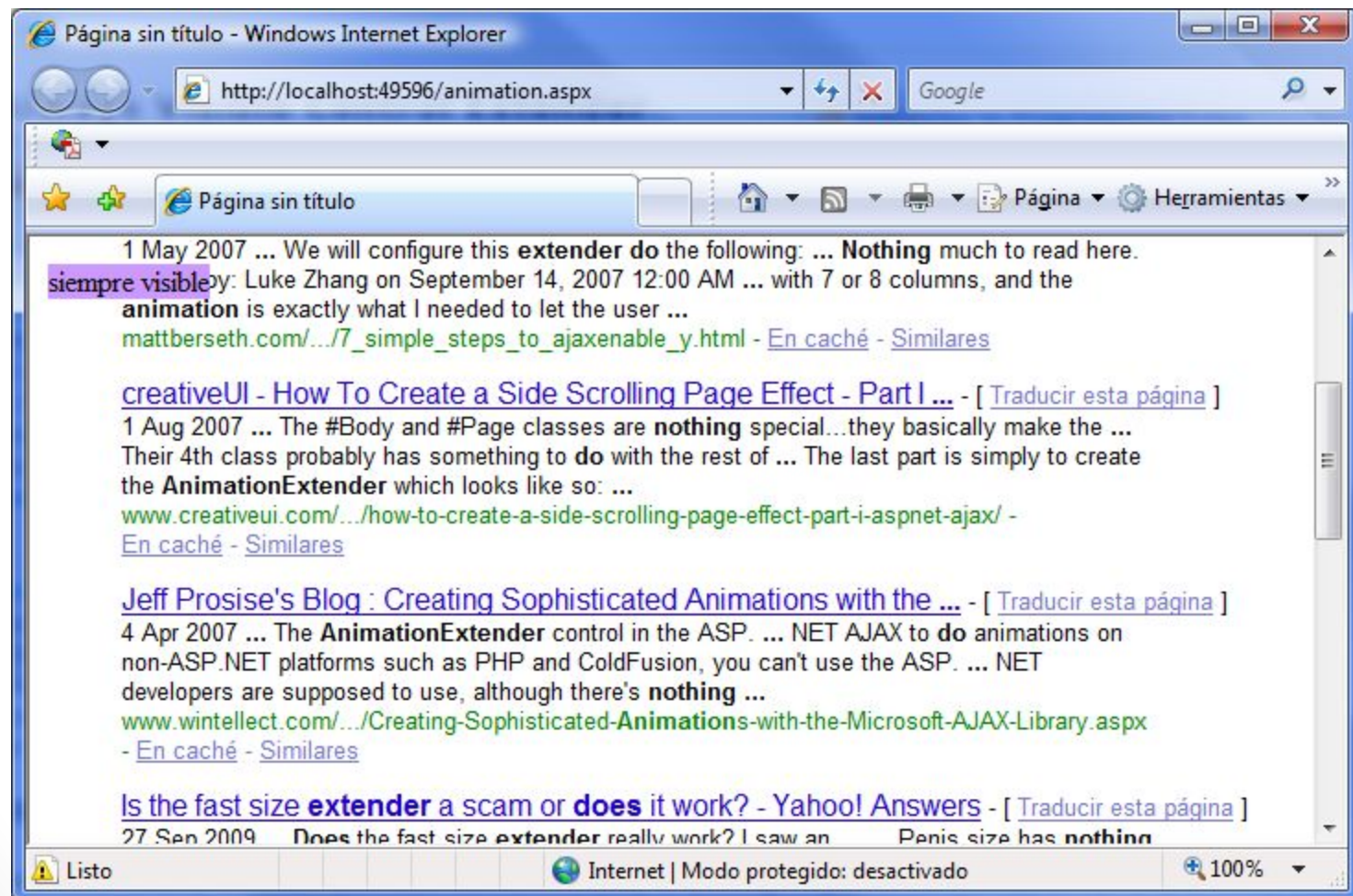
# Exercise



## Ejercicio

- Añadir un panel estandar, 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: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 estandar que será la cabecera, y otro panel que será el contenido a expandir/contraer.
- 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

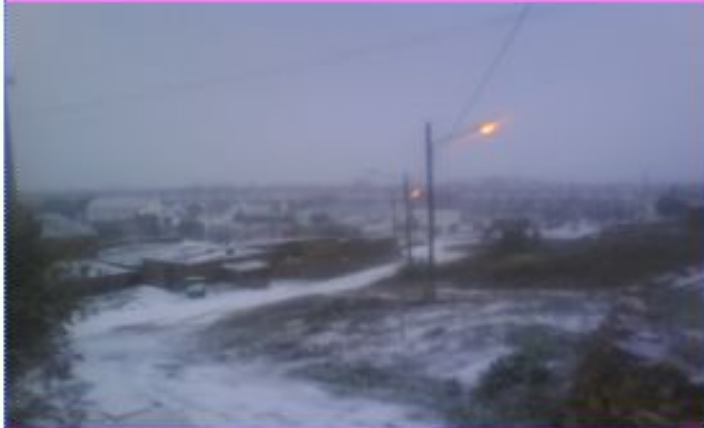
asp:Panel#Panel2

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...)

&lt;/asp:Label&gt;

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

&lt;/asp:Panel&gt;



# 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
```

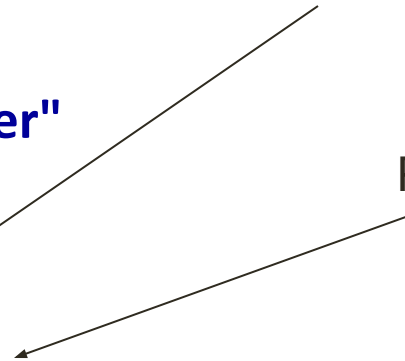
```
  TextLabelID="Label1" ExpandedText="Ocultar detalles"  
  CollapsedText="Mostrar detalles"
```

```
  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

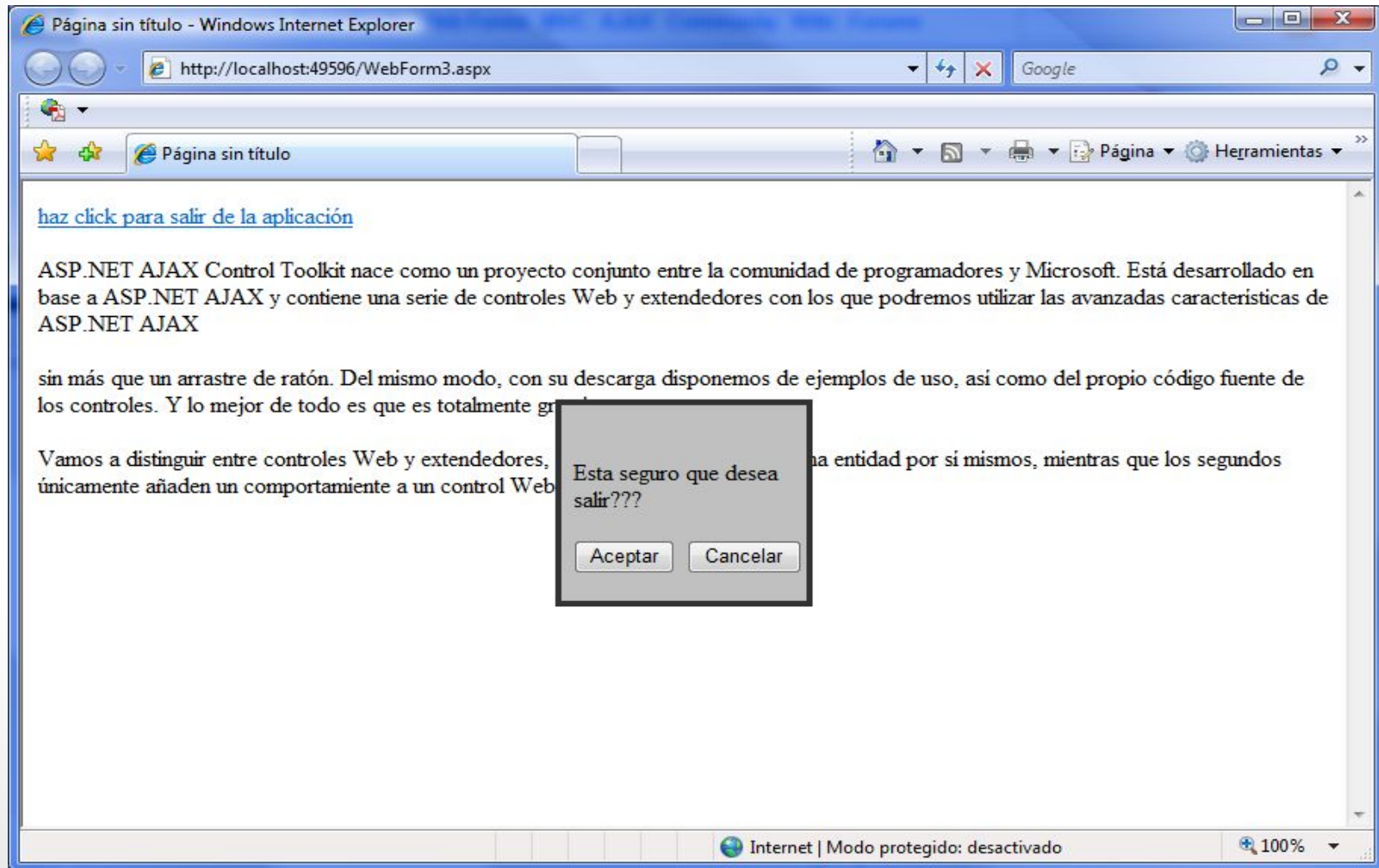
- Con este extensor, conseguimos el efecto de mostrar contenido deshabilitando la interacción con el resto de la página.
- Podemos emular el efecto del famoso “`window.open(...)`” de javascript sin necesidad de salir de la página en que estamos ni de abrir una nueva ventana del navegador.



## Ejercicio

- Crear un formulario web con los siguientes controles:
- Linkbutton
- Panel con el texto Seguro que desea salir? Y botones aceptar y cancelar
- ModalPopUp extensor

# 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)

```
<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

```
[System.Web.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  
marroñ");  
  
        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

22

Seguridad: Muy baja

8 more characters

Indicadores con barra estado

12345hy@



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.
- Util 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...

