

## Librerías de cliente

Ya en capítulos pasados se trabajó con las librerías de cliente más utilizadas en el mercado, entre ellas el conjunto Prototype+Script.aculo.us y Yahoo! UI Library. En este capítulo se va a hacer un repaso por otras librerías alternativas que van ganando mercado de a poco.

La mayoría tiene la funcionalidad básica similar e incorpora algunas funciones propias o controles ricos con distinto grado de uso y metodología de implementación.

Recuérdese que todas estas librerías se ejecutan directamente en el cliente, o sea, en el navegador Web.

### Adobe Spry

---

Spry es una librería JavaScript de Adobe pensada para la creación de Aplicaciones Ricas de Internet con AJAX. Su sitio Web se puede encontrar en [www.adobe.com/go/spry](http://www.adobe.com/go/spry), donde hay documentación, descarga y ejemplos.

El objetivo de la librería es utilizar AJAX pero abstrayendo al programador o el diseñador de cómo se realiza la comunicación con el servidor o cómo se realizan los controles ricos. La librería utiliza una metodología de trabajo que requiere poco código JavaScript y agrega atributos nuevos a los elementos XHTML que todos conocen.

Adobe Dreamweaver CS3 incorpora la librería Spry en forma transparente para crear controles ricos y enlazar controles a fuentes de datos asincrónicas en XML. También incorpora comportamientos y efectos nuevos que hacen uso de la librería Spry que ya viene incorporada con el entorno (fig. 1).

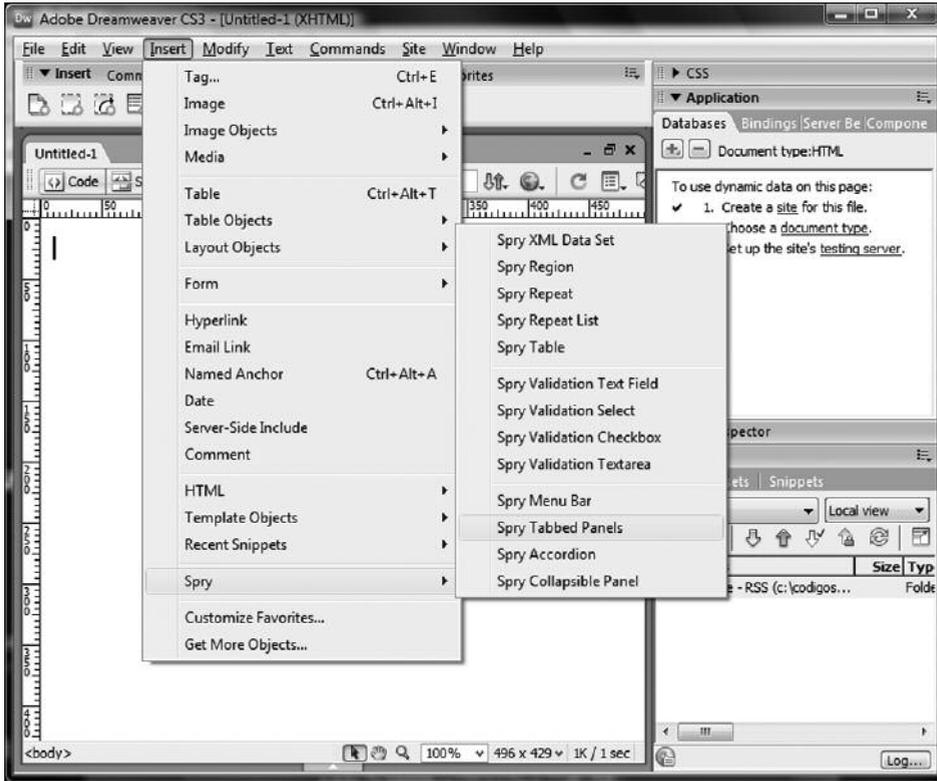


Fig. 1. Spry está incorporada en Dreamweaver CS3, donde se podrán crear controles ricos y enlazarlos a XML en vista Diseño.

## Spry Widgets

En la librería de Adobe se incluyen varios *widgets* o controles ricos listos para utilizar con una sola línea de JavaScript. Un *widgets* se compone de:

1. Estructura del *widget*. Es un código HTML que define la composición y el contenido del control.
2. Comportamiento del *widget*. Es un código JavaScript que instancia el *widget*, define su comportamiento y captura eventos.
3. Estilos del *widget*. Son clases CSS que definen la visualización del control rico.

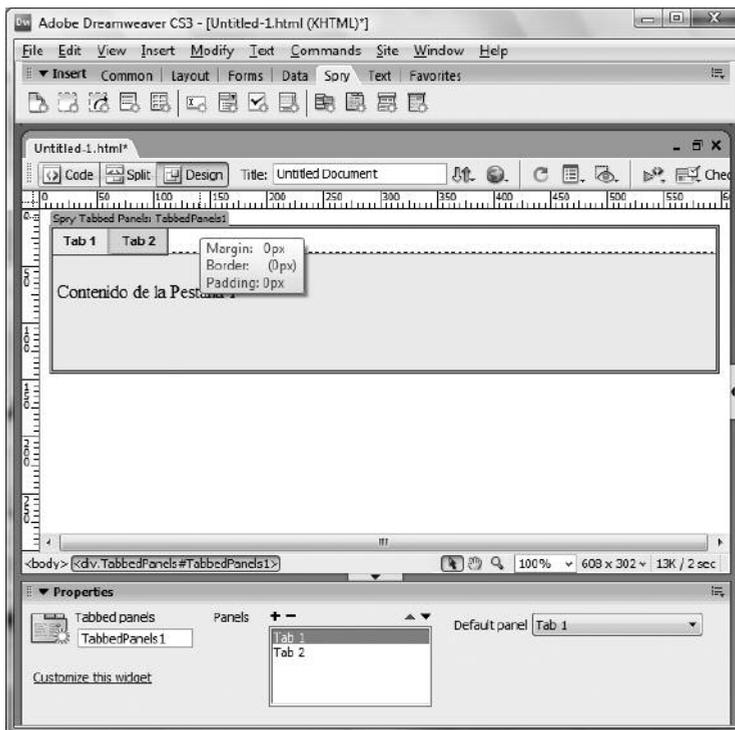
El objetivo de la librería es que la comunidad de desarrolladores cree *widgets* y controles nuevos bajo el mismo esquema utilizando la base de Spry.

Los controles disponibles hasta la versión 1.5 son:

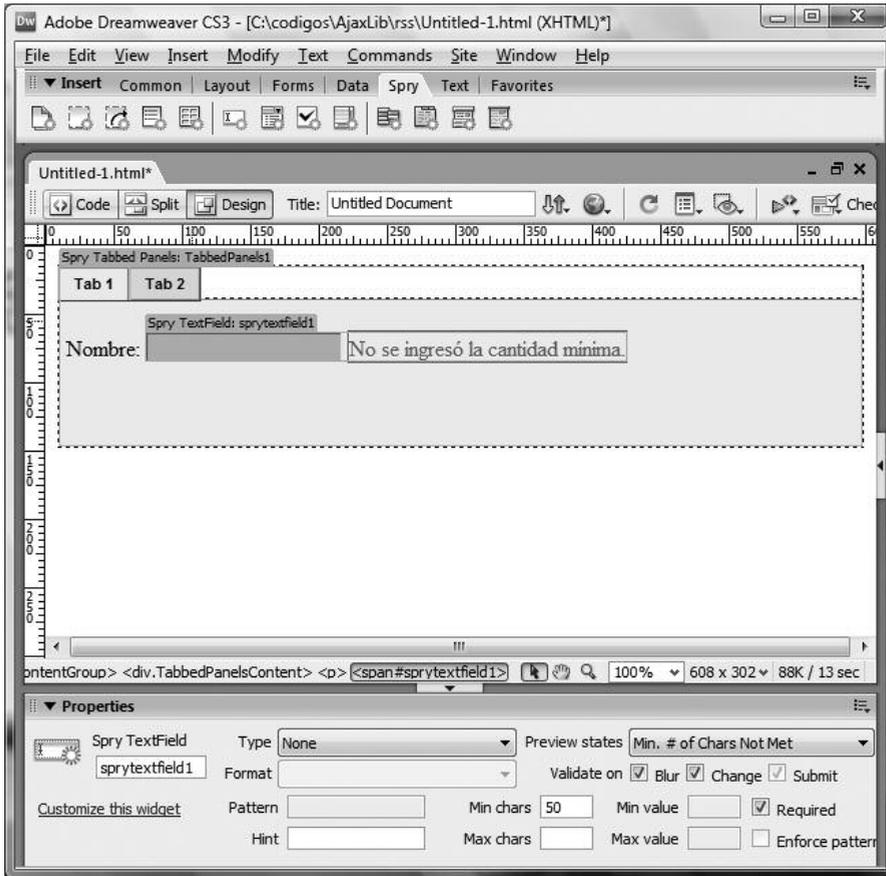
- **Accordion**: permite agrupar distintos contenidos con un título y abrir y colapsar uno por uno exclusivamente con un efecto de acordeón.

- **Collapsible Panel:** permite incluir un contenido con un título, y que al presionar sobre el título se pueda ver y colapsar el contenido.
- **Tabbed Panels:** permite agrupar varias ventanas de información bajo las famosas pestañas o tabs.
- **Menu Bar:** permite crear un menú desplegable, similar a los disponibles en aplicaciones de escritorio.
- **Validation Text Field:** unido a un campo de texto, se encarga de validar el contenido y mostrar un mensaje de error cuando la validación falla.
- **Validation Text Area:** similar al anterior, pero con un campo de texto multilínea (etiqueta TextArea).
- **Validation Select:** similar a los anteriores, pero valida una lista de selección.
- **Validation CheckBox:** valida que un campo de selección simple esté activado.

Todos estos controles pueden ser configurados y previsualizados desde Adobe Dreamweaver CS3, incluso por cualquier diseñador sin conocimientos de JavaScript (figs. 2 y 3).



**Fig. 2.** Se observan las clásicas pestañas implementadas desde Dreamweaver CS3. También se las puede hacer totalmente con JavaScript y DIV.



**Fig. 3.** Los validadores permiten mostrar mensajes de error y estilos CSS distintos cuando falla algún requisito mínimo, como ser obligatorio o tratarse de un *e-mail* válido. Se puede validar al hacer el envío o cuando el usuario sale de foco del control.

## XML DataSets

Adobe Spry soporta un subset del lenguaje de consultas XPath que permite navegar por un XML de manera más sencilla que con la especificación DOM. De esta manera, un XML con información se transforma en un objeto conocido como DataSet para la librería que puede enlazarse a cualquier elemento HTML.

Aquí se ve un ejemplo de cómo se crea un DataSet:

```
<script type="text/javascript" src="includes/xpath.js"></script>
<script type="text/javascript" src="includes/SpryData.js"></script>

<script type="text/javascript">
```

```
var dsClientes = new Spry.Data.XMLDataSet("clientes.xml",
"clientes/cliente");
</script>
```

El segundo parámetro es una consulta XPath que indica qué leerá del XML, de la etiqueta clientes, todas aquellas etiquetas cliente que existan y esa colección de nodos cliente será considerada como el registro que se ha de iterar.

Una vez que el DataSet está cargado (lo trae vía XMLHttpRequest) se lo puede enlazar en forma automática con casi cualquier etiqueta XHTML (hay algunas restricciones) por medio de un atributo propio de Spry. Por ejemplo:

```
<div id="divClientes" spry:region="dsClientes">
  <table>
    <tr>
      <th>Código</th>
      <th>Nombre</th>
      <th>Teléfono</th>
    </tr>
    <tr spry:repeat="dsClientes">
      <td>{codigo}</td>
      <td>{nombre}</td>
      <td>{telefono}</td>
    </tr>
  </table>
</div>
```

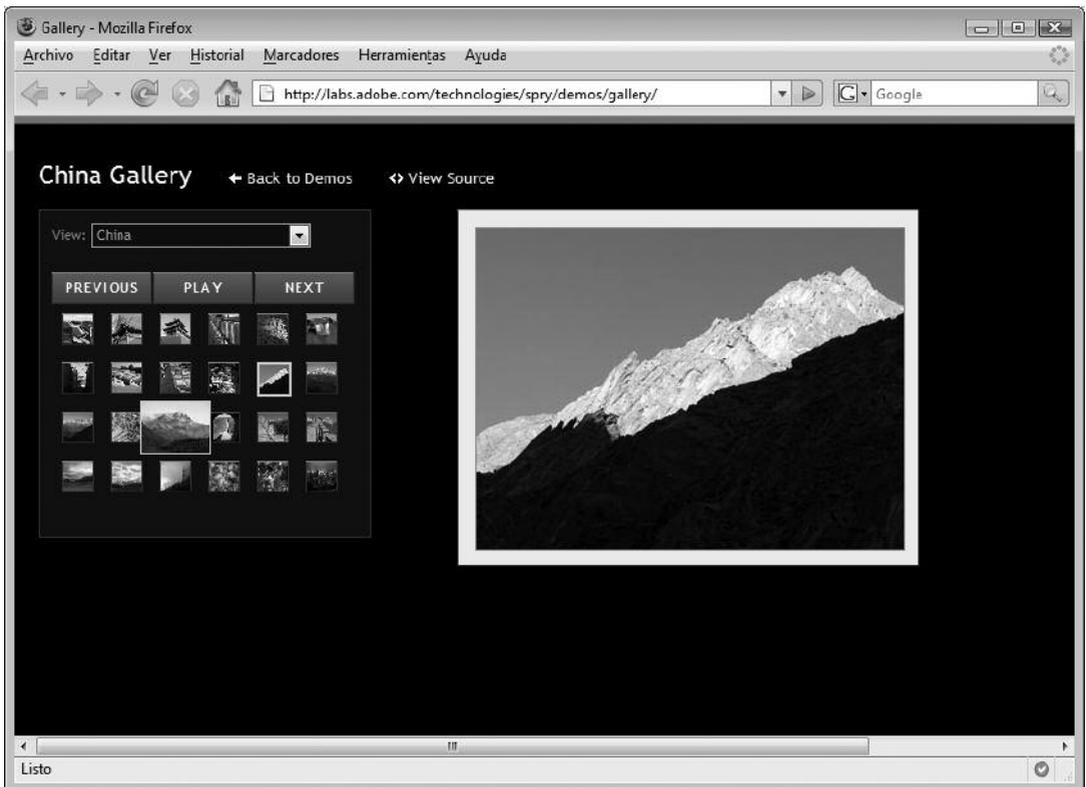
En el vocabulario de Spry el divClientes se conoce como una región dinámica. Cuando se crea el DataSet la fila TR que contiene la instrucción spry:repeat se repetirá por cada nodo definido. Asimismo, {codigo} se reemplazará por una subetiqueta (o atributo, si se lo especifica al estilo {@codigo}) llamada código, que exista dentro de cada cliente.

Estos atributos con prefijo spry: son XHTML válidos y no tienen problemas con los navegadores. Recuérdese que la repetición de los TR y el reemplazo por los datos del XML se realizan en el cliente, vía JavaScript. La librería no funciona sobre el servidor que es lo que uno supone cuando ve por primera vez una página que utiliza Spry (fig. 4).

Con muy poco código también se puede crear una página que utilice el modelo Maestro-Detalle, por ejemplo, para mostrar el detalle de un cliente y los productos disponibles cuando el usuario selecciona un cliente de una lista.

Entre los atributos disponibles que permiten trabajar con datos se encuentran:

Atributo	Descripción
spry:repeat	Repite la etiqueta actual por cada nodo del XMLDataSet.
spry:repeatchildren	Igual al anterior, sólo que repite el nodo XHTML hijo del actual.
spry:if	Permite mostrar la etiqueta actual sólo si se cumple la condición indicada en este atributo.
spry:choose	Equivale a un <i>switch</i> o <i>case</i> que permite mostrar sólo un hijo de la etiqueta. Cada hijo tendrá que incluir la condición para ser mostrado, en un atributo spry:when o spry:default.
spry:state	Define que el contenido de la etiqueta actual aparezca sólo cuando la aplicación esté en un cierto estado que nosotros definimos, por ejemplo: esperando, cargando, finalizado.
spry:hover	Especifica el nombre de una clase CSS para cuando el usuario posa el cursor sobre el elemento.
spry:select	Especifica el nombre de una clase CSS para cuando el usuario selecciona un elemento.



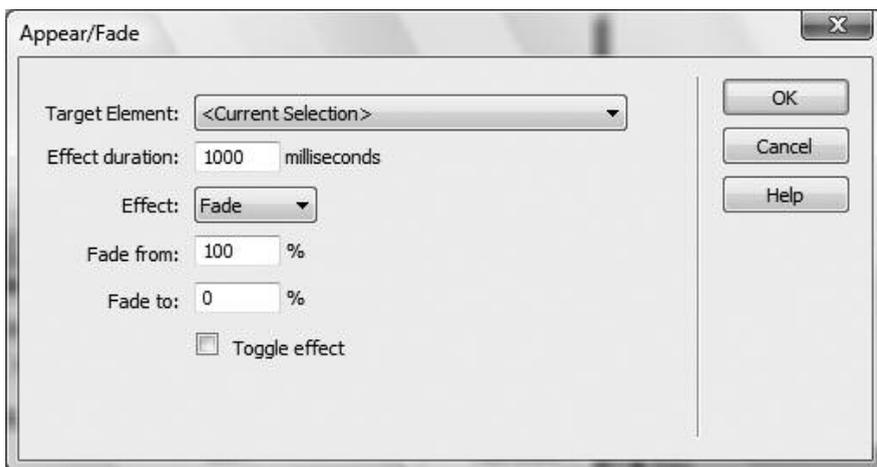
**Fig. 4.** En el sitio Web de Spry se encontrarán varios ejemplos simples y muy vistosos sobre cómo enlazar XHTML a datos XML.

Cuando estamos en un repeat u otra instrucción que itera entre varios registros, podemos utilizar algunas de las opciones siguientes en cualquier nodo XHTML:

Expresión	Descripción
{etiqueta}	Muestra el valor de una etiqueta del nodo en el cual se itera.
{@etiqueta}	Muestra el valor que tiene un atributo del nodo en el que se itera.
{ds_RowID}	El ID del registro actual.
{ds_RowNumber}	El índice del registro actual empezando en 0.
{ds_RowNumberPlus1}	El índice del registro actual empezando en 1.
{ds_RowCount}	La cantidad total de filas.
{ds_SortColumn}	El nombre de la columna por la cual se está ordenando.
{ds_EvenOddRow}	Devuelve 'even' (par) u 'odd' (impar), según el índice del registro actual.

## Efectos Spry

Adobe Spry también presenta una galería de efectos que se pueden aplicar en casi cualquier elemento XHTML. Los efectos que incluye la librería son: Fade, Highlight, Blind Up, Blind Down, Grow, Shake y Squish. Según la misma documentación de Spry, para los efectos se utilizó la misma nomenclatura y algo de las técnicas de los efectos de Script.aculo.us trabajando en conjunto con el creador de esta última librería (fig. 5).



**Fig. 5.** Se pueden aplicar efectos Spry desde código JavaScript o en Dreamweaver CS3 seleccionando en vista diseño.

## Microsoft AJAX Library

---

Microsoft no se quiso quedar atrás en el mundo de AJAX y lanzó una serie de herramientas para trabajar con esta plataforma. Entre ellas se encuentran los siguientes *frameworks*:

- **Microsoft ASP.NET AJAX Extensions:** es una librería de servidor que permite ejecutar aplicaciones ASP.NET utilizando tecnología AJAX.
- **ASP.NET AJAX Control Toolkit:** es una colección de código abierto de controles ricos que se pueden incorporar a ASP.NET AJAX.
- **Microsoft AJAX Library:** es una librería de cliente (la misma que utiliza la librería de servidor) que permite integrarse a PHP o cualquier otra plataforma.

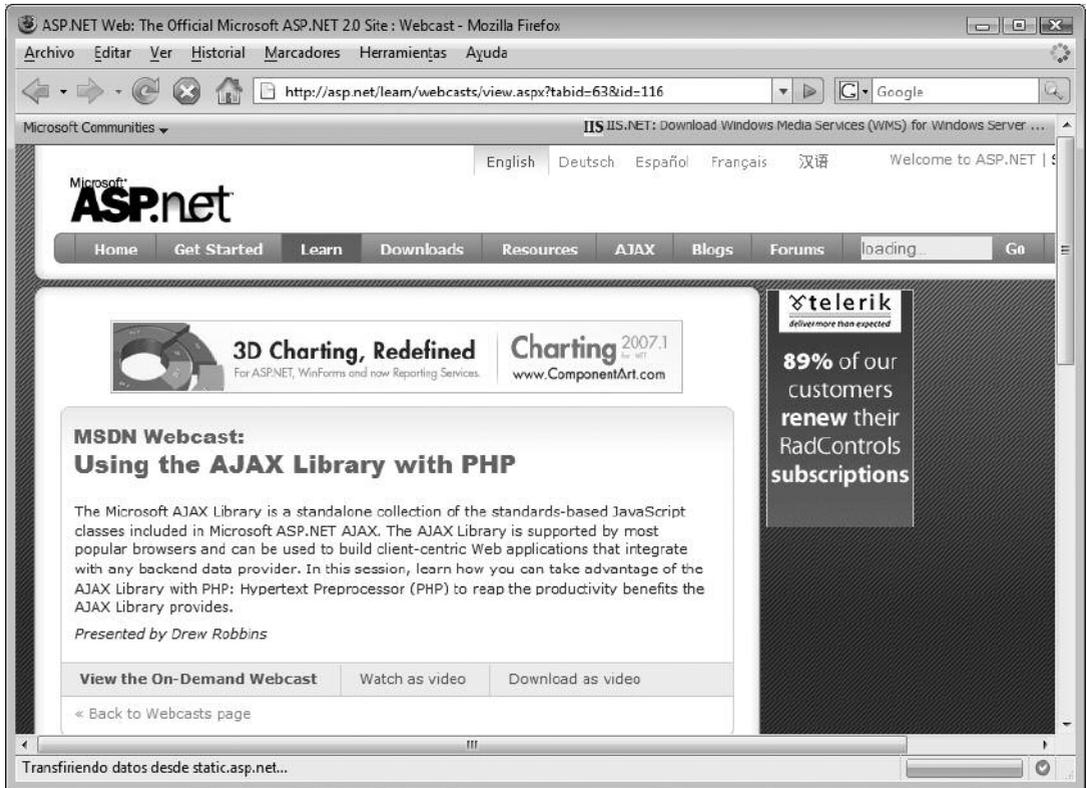
La librería de cliente, entonces, se puede descargar como archivos JavaScript y no requiere que se utilice la plataforma de Microsoft en el servidor. La librería de cliente se puede descargar desde [ajax.asp.net/downloads](http://ajax.asp.net/downloads).

Hay un proyecto *open source*, auspiciado por el mismo Microsoft, para enlazar esta librería con PHP de manera sencilla. Se llama PHP Library for the Microsoft Ajax y se puede descargar de [codeplex.com/phpmsajax](http://codeplex.com/phpmsajax).

La librería de cliente es compatible con todos los navegadores conocidos del mercado e incorpora a JavaScript algunos conceptos que son propios del framework .NET, en lenguajes como C# o Visual Basic, incluidos soporte de JSON y *WebServices*. Entre las características que incorpora se encuentran (fig. 6):

- Soporte de creación de clases y herencia.
- Soporte de espacio de nombres o *namespaces* bajo los cuales existen y se agrupan las clases.
- Modificadores de acceso. Si bien en JavaScript no se puede definir como privado un atributo, se toma la convención de que si el atributo comienza con guión bajo (`_`) éste es privado y no se permite su acceso.
- Soporte de creación de interfaces.
- Soporte de enumeraciones.
- API de Reflection para analizar el contenido de una clase.
- Agrega métodos conocidos en .NET a clases de JavaScript, por ejemplo:
  - El método `parse` a los Boolean y Number.
  - Los métodos `startsWith`, `endsWith`, `lTrim`, `rTrim`, `format`, `localeFormat` a los Strings.
  - `add`, `clone`, `contains`, `indexOf`, `foreach`, `insert`, `remove`, `get_length` a los Arrays

- createError que equivale a throw new Exception en .net.
- El método toString a las fechas.



**Fig. 6.** Microsoft ofrece un video donde explica cómo integrar fácilmente su librería de cliente con código PHP.

## jQuery

jQuery es una librería bastante nueva en el mercado de AJAX, pero que promete cumplir con la misma funcionalidad de Prototype+Script.aculo.us, aunque de manera más liviana, rápida y sencilla de codificar.

Su sitio Web oficial es jquery.com y ofrece en 20 Kb toda su funcionalidad compatible con todos los navegadores conocidos del mercado.

Entre las funciones que incluye se pueden mencionar:

- Soporte del evento ready que se ejecuta cuando el DOM se encuentra completamente cargando sin las imágenes.
- Manejo de eventos sencillo mediante funciones.

- Efectos visuales con poco código.
- Soporte de cadena de ejecución con la posibilidad de ejecutar múltiples instrucciones en una sola línea.
- Manipulación de documentos con CSS 1, 2 y 3.
- Soporte de algunos estándares de Prototype como \$ y \$\$ incorporando expresiones XPath.
- Soporte de nuevos *plugins*.

Algunos ejemplos con jQuery (fig. 7):



Fig. 7. jQuery quiere convertirse en el reemplazante del binomio Prototype y Scriptaculous.

```
// Trae todos los radiobuttons seleccion
var radio = $("input[@type=radio][@checked]");

// Le agrega la clase CSS texto a todos los párrafos
$("p").addClass("texto");
```

```

// Pone a todos los links para que se abran en ventana nueva
$("a").attr({ target: "_blank", title: "Se abrirá en ventana nueva"
});

// Agrega una imagen luego de cada fin de párrafo
$("p").after("<img src='fin.jpg' /> ");

// Define la función a ejecutarse en el clic del botón
$("btnEnviar").click(function() { alert("Gracias por enviar los
datos") });

// Anima un div
$("divContenido").fadeIn("slow");

// Hace una petición AJAX
$.ajax({
  type: "POST",
  url: "guardar.php",
  data: "nombre=Juan&apellido=Gomez",
  success: function(mensaje){
    alert(mensaje);
  }
});

// Hace algo si es Internet Explorer
if ($.browser.msie) {

}

// Itera entre una colección
$.each( [2, 4, 6], function(i, total) { alert(i) });

// Busca todos los párrafos que sean de clase "copete",
// les agrega la clase "deportes" y los muestra con un efecto lento
$("p.copete").addClass("deportes").show("slow");

```

Es una librería interesante con bastante futuro, que intenta que los desarrolladores AJAX puedan hacer más cosas con menos código.

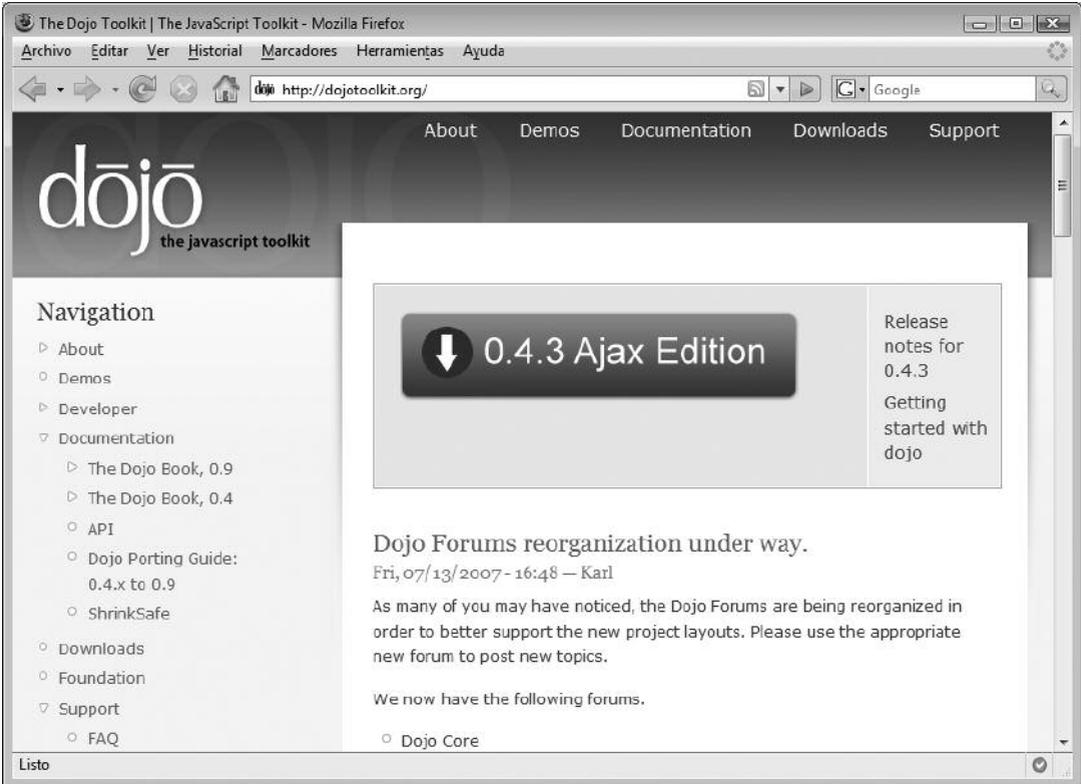
## Dojo Toolkit

---

Dojo es un *framework* muy completo para realizar Aplicaciones Ricas de Internet con JavaScript. Una de sus desventajas es que es uno de los más pesados a la hora de cargar código JavaScript, pero a su favor tiene su gran funcionalidad, que incorpora como un nuevo modelo de eventos y facilidades de escritura y utilitarios para las tareas más

comunes en un desarrollo de este tipo. Su sitio oficial es [dojotoolkit.org](http://dojotoolkit.org), donde se encontrarán ejemplos y la documentación oficial, además de la librería (fig. 8).

En realidad, Dojo Toolkit surgió como la unión de varios *frameworks* y librerías que se unieron bajo un mismo paraguas de trabajo.



**Fig. 8.** Dojo es una librería muy completa que incorpora la funcionalidad de muchas otras en una sola.

## Dijit

Sobre Dojo hay una librería de *widgets* o controles ricos conocida como Dijit. Esta librería incluye controles agrupados en categorías:

Controles de formulario:

- Auto Completer.
- Inline Edit Box.
- TextBox (con validación y formato).
- Buttons (de varios tipos).
- Dropdown Calendar.

- Number Spinner.
- Resizeable Text Area.
- Select.
- Slider.

#### Controles de disposición:

- Content Pane.
- Title Pane.
- Layout Container.
- Tab Container.
- Page Container.
- Split Container.
- Accordion Container.
- Dialog.

#### Controles de comando:

- ToolBar.
- Menús Popup.

#### Controles de asistencia al usuario:

- Progress Bar.
- Tooltip.

#### Controles avanzados:

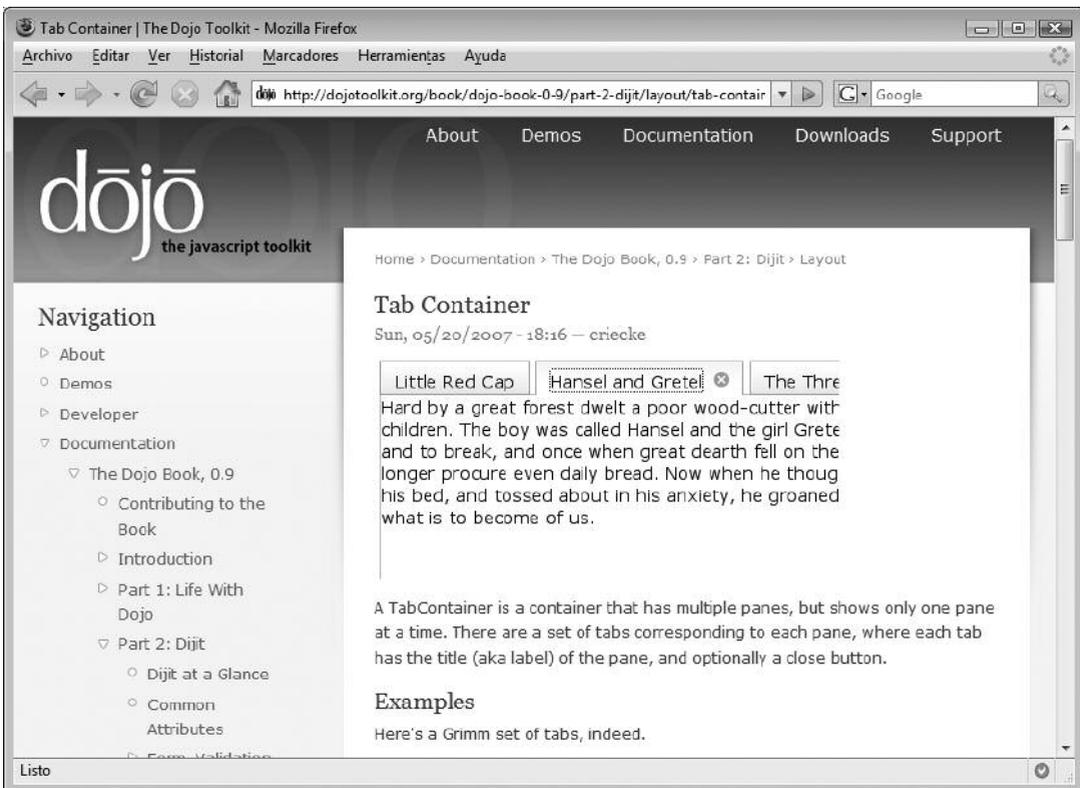
- Color Palette.
- Tree.
- Rich Text Editor.
- Grid.
- Toggler.

Todos los *widgets* soportan temas para combinar estilos y colores, y contienen una lista de atributos que todos comparten en común y hacen más sencilla la tarea de programarlos.

## Módulos

Entre los módulos de funcionamiento que incorpora, se pueden mencionar (fig. 9):

- *Charts*: permite crear gráficos estadísticos.
- *Offline*: permite crear aplicaciones desconectadas junto a Google Gears.
- *Storage*: permite almacenar información en el cliente.
- *Drag and Drop*.
- Animación.
- Botón de Atrás.
- Internacionalización y múltiples idiomas.



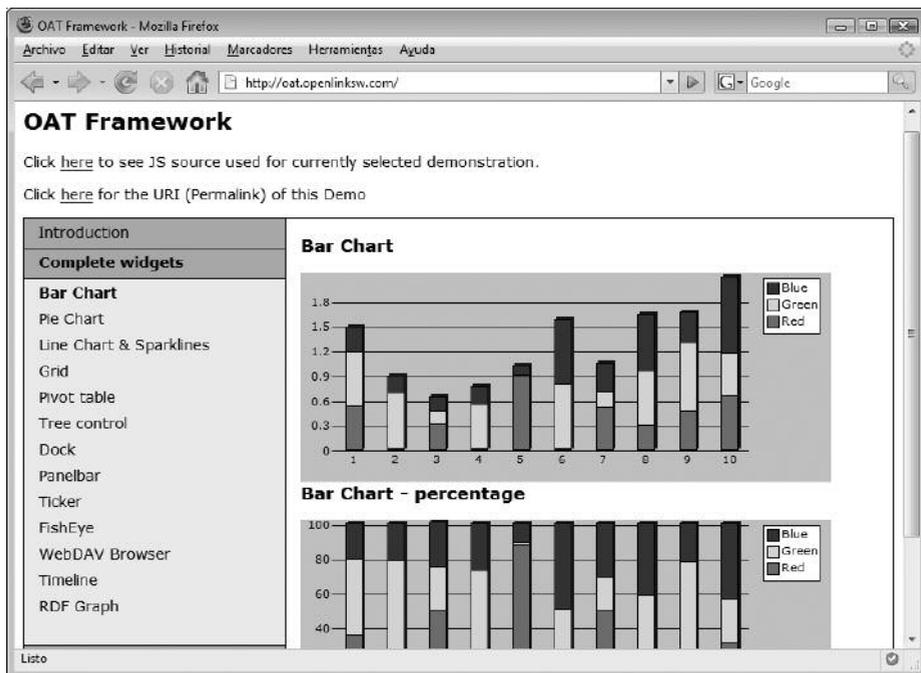
**Fig. 9.** Dojo Toolkit es la librería que agrupa más controles ricos compatibles con AJAX.

## Otras librerías

Ahora haremos un repaso final a otras librerías JavaScript (figs. 10 y 11).



**Fig. 10.** moo.FX es una librería ultraliviana que permite hacer efectos visuales con sólo 3 Kbytes de JavaScript.



**Fig. 11.** Algunas librerías, como OAT Framework, permiten crear *charts* o gráficos estadísticos desde JavaScript.

### **Moo.fx – moofx.mad4milk.net**

Librería ultraliviana que permite hacer efectos y acordeones con sólo 3 Kb.

### **OAT Framework – oat.openlinksw.com**

Completa librería que contiene controles ricos y creación de gráficos estadísticos con SVG y otras tecnologías.

### **Aflax – aflax.org**

Librería que permite crear aplicaciones Flash en SWF desarrollando toda la interfaz desde JavaScript.

### **Rico – openrico.org**

Antigua librería para AJAX que permite crear controles ricos con facilidad.

### **Number Formatting Library - xaprb.com**

Permite darle formato a un número según un patrón dado.

### **Mochikit – mochikit.com**

Librería general que agrega mucha funcionalidad a JavaScript y AJAX.

## **Librerías para PHP**

Las librerías de servidor para AJAX tienen la ventaja de que, en general, nos abstraen del JavaScript que se conecta al servidor. Para ello, desde código de servidor se configura cierta información y luego se puede hacer uso de ella desde JavaScript, como si fueran objetos locales de JavaScript.

Muchas de estas librerías tienen la desventaja de que se pierde el control de bajo nivel de la petición y de lo que sucede en la comunicación cliente-servidor.

Se empezará por analizar las más utilizadas en PHP.

### **SAJAX**

---

Simple AJAX Toolkit for PHP (SAJAX) fue una de las primeras librerías para PHP que incorporaron funcionalidades de AJAX a este lenguaje. Se puede descargar gratis de [www.modernmethod.com/sajax](http://www.modernmethod.com/sajax) y la idea es muy sencilla.

Desde PHP se hacen algunos includes obligatorios y luego se definen funciones que se exportan a AJAX. Estas funciones, entonces, estarán disponibles para su ejecución directamente desde JavaScript. La librería se encargará de hacer la petición XMLHttpRequest, pasar los parámetros y ejecutar la función finalmente en PHP.

Por ejemplo:

```
<?
    require("Sajax.php");

    function sumar($a, $b) {
        return $a * $b;
    }

    sajax_init();
    // Exportamos la función a JavaScript
    sajax_export("sumar");
    sajax_handle_client_request();

?>
<html>
<head>
    <title>Suma AJAX</title>
    <script>
    <?
    sajax_show_javascript();
    ?>

    function recibirResultado(z) {
        document.getElementById("z").value = z;
    }

    function hacerSuma() {
        var a, b;

        a = document.getElementById("a").value;
        b = document.getElementById("b").value;
        // Llamamos a la función de PHP con un prefijo _x
        // Y le agregamos un último llamado que es la función
        // callback
        x_sumar(a, b, recibirResultado);
    }
    </script>

</head>
<body>
    <input type="text" name="a" id="a" value="2" size="3">
    *
    <input type="text" name="b" id="b" value="3" size="3">
    =
```

```
<input type="text" name="z" id="z" value="" size="3">
<input type="button" name="check" value="Hacer Suma"
onclick=" hacerSuma (); return false;">
</body>
</html>
```

## XAJAX

---

XAJAX es similar a SAJAX con la diferencia de que implementa la funcionalidad en clases PHP compatible con PHP 4 y 5. Se puede descargar desde [www.xajaxproject.org](http://www.xajaxproject.org).

XAJAX incorpora un objeto llamado `xajaxReponse`, que permite que desde PHP se le den instrucciones a JavaScript para que modifique algo del HTML. Por ejemplo:

```
require_once("xajax.inc.php");
$xajax = new xajax();
$xajax->registerFunction("myFunction");

function suma($a, $b) {
    $respuesta = new xajaxResponse();
    $z = $a + $b;
    // Le damos instrucciones al JavaScript desde PHP
    $respuesta->addAssign("divContenido","innerHTML", "La suma es
    $z");
    $respuesta->addAssign("z","value", $z);
    return $respuesta;
}
```

## PAJAX

---

PAJAX es otra librería para PHP que va un poco más allá que las anteriores.

Este *framework* permite crear clases en PHP con la lógica de negocios que se desea aplicar, y luego, desde JavaScript simplemente instanciar esas clases y ejecutar sus métodos sin preocuparse por la comunicación con el servidor.

Por ejemplo:

```
<?
class Calculadora extends PajaxRemote {
    function suma($x, $y) {
```

```

        return $x + $y;
    }

    function multiplicacion($x, $y) {
        return $x * $y;
    }

    ...
}
?>

```

Desde el XHTML se agrega un include:

```

<script type="text/javascript"
    src="pajax_import.php?Calculadora">
</script>

```

y luego simplemente se instancia el objeto y se lo usa como si fuera local de JavaScript.

```

window.onload = {
    var calculadora = new Calculadora();
    alert(calculadora.suma(2, 3));
}

```

La librería está para su descarga en [www.auberger.com/pajax](http://www.auberger.com/pajax)

## Librerías para ASP.NET

Ahora se analizarán las librerías disponibles para la tecnología ASP.NET de Microsoft.

### Ajax.NET Professional

---

Ésta fue la primera librería que surgió para conectar ASP.NET con tecnología AJAX de manera fácil desde el servidor. Funciona en las versiones del .NET Framework 1.1 y 2.0 o superior.

Su sitio Web es [www.ajaxpro.info](http://www.ajaxpro.info) y sus funcionalidades permiten ejecutar desde JavaScript métodos de una página ASPX sin ocuparse de las peticiones al servidor.

Se debe descargar el paquete e instalar en la carpeta **bin** del proyecto el archivo **AjaxPro.dll**, si se usa .NET 1.1, o **AjaxPro.2.dll**, si se utiliza .NET 2.0 o superior.

En el archivo de configuración **web.config** se debe incluir lo siguiente:

```
<?xml version="1.0" encoding="utf-8" ?>
<configuration>
  <system.web>
    <httpHandlers>
      <add verb="POST,GET" path="ajaxpro/*.ashx" type="AjaxPro.Ajax-
HandlerFactory, AjaxPro.2"/>
    </httpHandlers>
  </system.web>
</configuration>
```

Luego, se debe definir un atributo ante cada método que se desea llamar desde JavaScript; por ejemplo, en C#:

```
[AjaxPro.AjaxMethod]
public int Suma(int a, int b)
{
  return a+b;
}
```

o desde Visual Basic:

```
<AjaxPro.AjaxMethod> _
Public Function Suma(ByVal a as Integer, ByVal b as Integer) as
Integer
  Return a+b
End Function
```

Por último se agrega una línea en el evento **Page\_Load**

```
AjaxPro.Utility.RegisterTypeForAjax(typeof(_Default));
```

El paso siguiente es conocer el espacio de nombres o nombre del proyecto donde se está trabajando, y el nombre de la clase de la página, por ejemplo, **\_Default**. Ahora, desde JavaScript sólo se debe utilizar:

```
MiProyecto._Default.Suma(a, b, mostrarResultado);
```

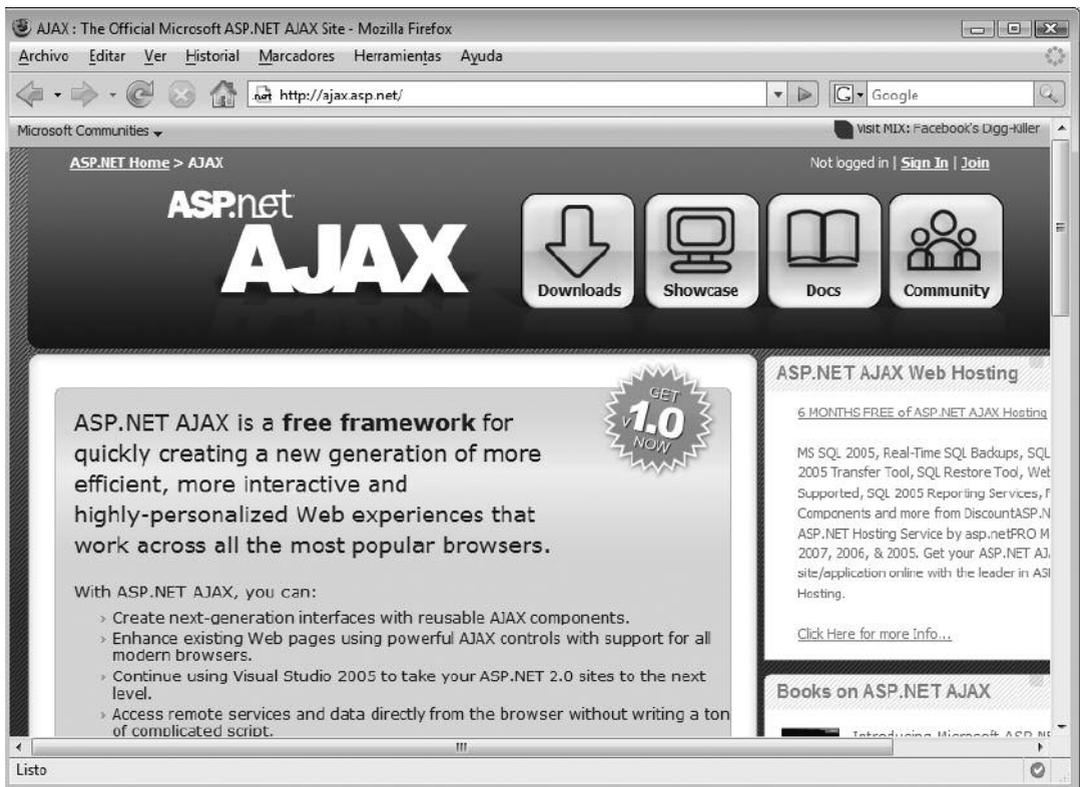
## Microsoft ASP.NET AJAX Extensions

Microsoft lanzó su paquete de agregado a la tecnología ASP.NET para la creación de aplicaciones AJAX y realmente permite que se desarrolle una aplicación RIA sin ningún esfuerzo. Primero se lo conoció como Microsoft Atlas y luego se lo llamó Microsoft ASP.NET AJAX Extensions. Su sitio oficial es [ajax.asp.net](http://ajax.asp.net).

Con soporte para Visual Studio y Visual Web Developer Express (la versión gratuita) es posible crear aplicaciones AJAX en vista diseño sin tratar con JavaScript.

Para ello, la librería incorpora unos controles de servidor no visuales que se integran con el *framework* de ASP.NET para que cualquier página ASP.NET 2.0 o superior se pueda convertir en AJAX insertando sólo un nuevo control en el formulario Web.

Esto permite migrar sitios no AJAX en pocos minutos y garantizar toda la misma funcionalidad que tenían antes (figs. 12 y 13).



**Fig. 12.** Microsoft ASP.NET AJAX da un paso adelante en la creación de sitios ricos, sin necesidad de programar nada de JavaScript, utilizando controles de servidor.

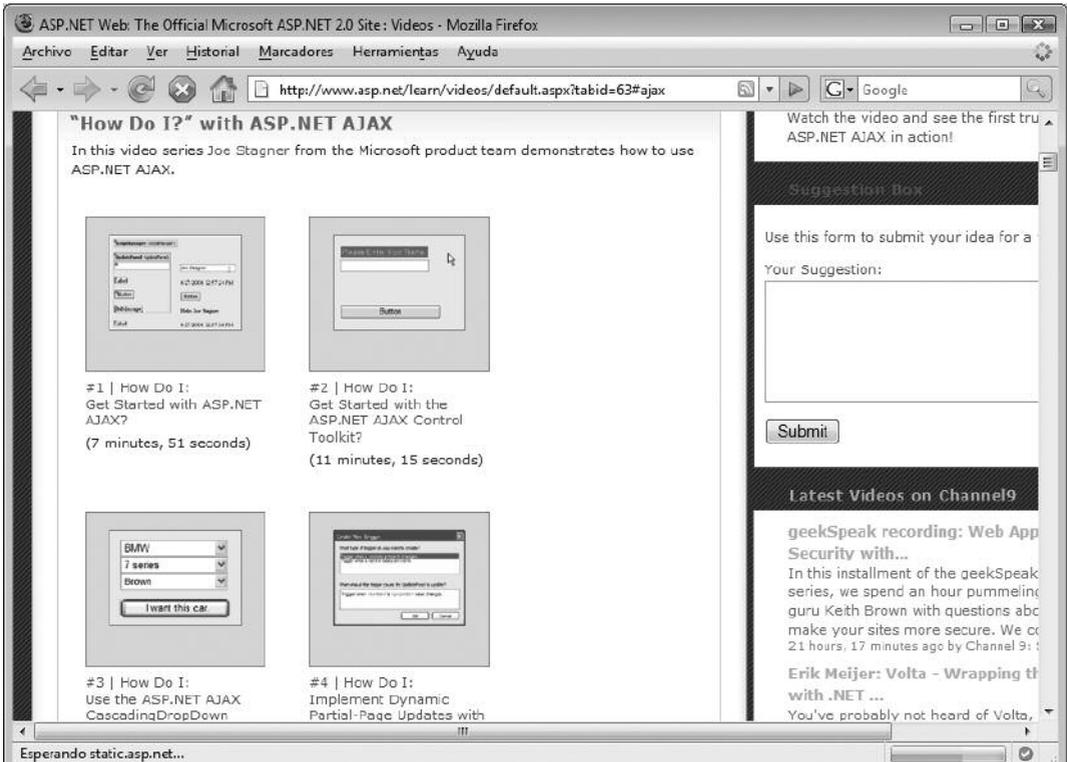


Fig. 13. Microsoft ofrece decenas de videos de cómo utilizar su librería de AJAX para ASP.NET.

## Controles

Al paquete de controles básicos de ASP.NET se incorporan luego de instalar las extensiones:

Control	Descripción
ScriptManager	Es un control obligatorio que se debe insertar en el ASPX que generará todos los JavaScripts necesarios.
UpdatePanel	Permite insertar contenido y controles, y todos ellos harán <i>postbacks</i> asincrónicos vía AJAX sin ningún otro cambio. También es posible definir que otros eventos fuera del panel invoquen peticiones AJAX en forma automática.
UpdateProgress	Al asignar un UpdateProgress a un UpdatePanel se verá un cartel de Cargando cuando el panel se actualice vía AJAX.
TimerControl	Permite realizar una petición al servidor cada x segundos y ejecutar un evento en el servidor.
ScriptManagerProxy	Permite ejecutar AJAX en aplicaciones que usen páginas maestras ( <i>MasterPages</i> ).

La versión más simple de este *framework* es incluir un `ScriptManager` al inicio de un `ASPX` y luego un `UpdatePanel` encerrando todo el contenido. De manera automática toda la página pasará a actualizarse vía `AJAX` en lugar de hacer *postbacks* sincrónicos al servidor. Los eventos en el servidor se seguirán ejecutando de igual forma y no se debe cambiar nada más. Para optimizar la transferencia, en `UpdatePanel` sólo debería encerrarse el contenido que se actualiza en las peticiones.

Por ejemplo:

```
<asp:Form runat="server">
  <ajax:ScriptManager id="script1" runat="server" />
  <ajax:UpdatePanel runat="server" id="update1">
    <asp:Button runat="server" onclick="PulsaBoton" text="Re-
cibir AJAX" />
    <asp:Label runat="server" id="etiqueta" />
  </ajax:UpdatePanel>
</asp:Form>
```

Por ejemplo, la función `PulsaBoton`, podría hacer:

```
etiqueta.Text = "Hola AJAX"
```

Y cuando el usuario pulse el botón, se realizará una petición `AJAX` automática, que dará como resultado que en la etiqueta se vea el texto `Hola AJAX`.

## Control Toolkit

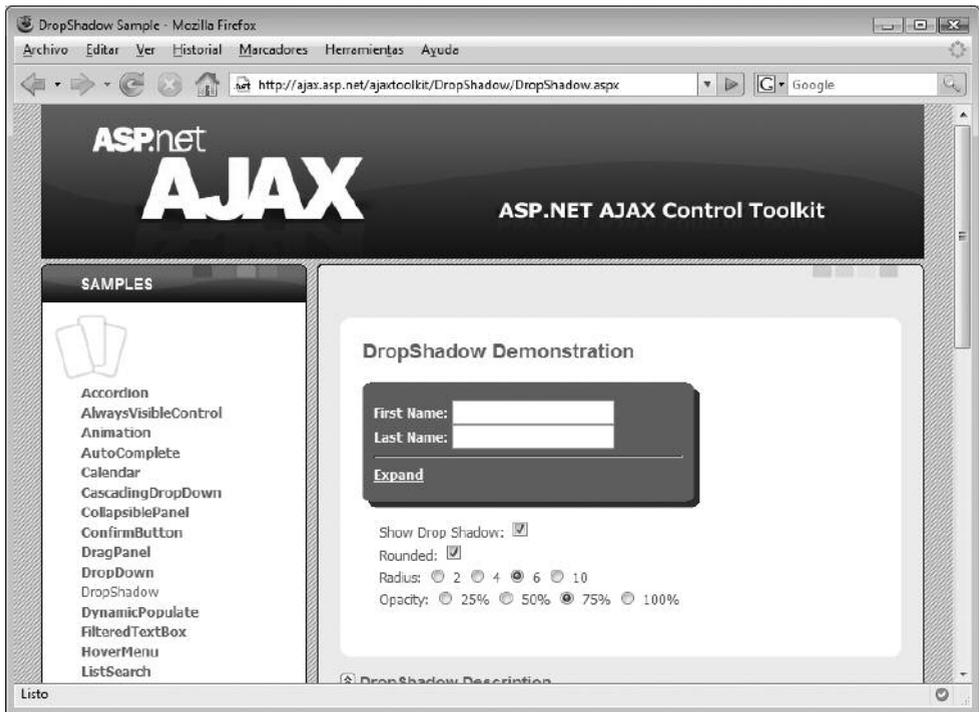
Además del paquete inicial, hay un paquete de controles visuales que agregan características de sitios ricos de Internet a una aplicación `Web ASP.NET`. Son todos controles de servidor que se pueden arrastrar y soltar en `Visual Studio` o `Visual Web Developer` y automáticamente estarán funcionando con `XHTML`, `CSS` y `JavaScript`. Algunos de ellos son adaptadores que modifican el comportamiento de otros controles básicos, como un `TextBox` con un `AutoComplete`.

Los controles son de código abierto y se actualizan en forma constante. Se puede ver un ejemplo de cada control en [ajax.asp.net/ajaxtoolkit](http://ajax.asp.net/ajaxtoolkit) (fig. 14).

Al momento de escribir este libro la lista es:

Accordion	MutuallyExclusiveCheckBox
AlwaysVisibleControl	NoBot
Animation	NumericUpDown
AutoComplete	PagingBulletedList
Calendar	PasswordStrength

- CascadingDropDown
- CollapsiblePanel
- ConfirmButton
- DragPanel
- DropDown
- DropShadow
- DynamicPopulate
- FilteredTextBox
- HoverMenu
- ListSearch
- MaskedEdit
- ModalPopup
- MutuallyExclusiveCheckBox
- NoBot
- NumericUpDown
- PagingBulletedList
- PasswordStrength
- PopupControl
- Rating
- ReorderList
- ResizableControl
- RoundedCorners
- Slider
- SlideShow
- TextBoxWatermark
- ToggleButton
- UpdatePanelAnimation
- ValidatorCallout



**Fig. 14.** Con AJAX Control Toolkit se podrán tener controles visuales ricos con sólo arrastrarlos y configurarlos en el IDE.

## Otros lenguajes

Se puede consultar una lista de *frameworks* y librerías actualizadas para cada lenguaje en:

### **Java**

[ajaxpatterns.org/Java\\_Ajax\\_Frameworks](http://ajaxpatterns.org/Java_Ajax_Frameworks)

### **ColdFusion**

[ajaxpatterns.org/Coldfusion\\_Ajax\\_Frameworks](http://ajaxpatterns.org/Coldfusion_Ajax_Frameworks)

### **.NET**

[ajaxpatterns.org/DotNet\\_Ajax\\_Frameworks](http://ajaxpatterns.org/DotNet_Ajax_Frameworks)

### **Python**

[ajaxpatterns.org/Python\\_Ajax\\_Frameworks](http://ajaxpatterns.org/Python_Ajax_Frameworks)

### **PHP**

[ajaxpatterns.org/PHP\\_Ajax\\_Frameworks](http://ajaxpatterns.org/PHP_Ajax_Frameworks)

### **Ruby**

[ajaxpatterns.org/Ruby\\_Ajax\\_Frameworks](http://ajaxpatterns.org/Ruby_Ajax_Frameworks)

