

AJAX

ASP.NET AJAX

Navegación acelerada

Sigla de **Asynchronous JavaScript y XML**, **AJAX** es una tecnología que permite interactuar con aplicaciones web mucho más rápidamente. Esto se debe al envío de pequeñas cantidades de información entre el servidor web y el cliente web. De este modo se reduce el tráfico sobre la conexión, y el aplicativo resulta más veloz, en comparación con un mismo programa que no use esta tecnología. Si no utilizamos **AJAX**, el circuito normal de una página **ASP.NET** es que cada vez que se refresca o se realiza un "post back", ésta es transmitida por completo nuevamente al navegador del usuario. Por el contrario, utilizando **AJAX** sólo se actualiza la porción de página necesaria.

ASP.NET AJAX combina el poder de **ASP.NET** en el servidor con las técnicas de ejecución del lado del cliente de **AJAX**. Es una nueva tecnología para desarrollos web impulsada por Microsoft que integra librerías de script del lado del cliente, es decir, sobre el browser, con aplicaciones hechas en **ASP.NET 2.0**. Gracias a esto se pueden desarrollar rápidamente páginas complejas con buenas respuestas en la comunicación entre el aplicativo, que corre en el navegador del cliente, y el servidor web; todo esto, con unos pocos controles que se utilizan del lado del servidor.

Con **ASP.NET AJAX**, los programadores pueden optar por alternativas de desarrollo. Podrán decidir programar del lado del cliente o del lado del servidor, o ambos.

En el caso de la programación del lado del servidor, los desarrolladores que estén familiarizados con el modelo pueden incluir funcionalidad **AJAX** a sus aplicaciones envolviendo los controles del servidor dentro de un nuevo control llamado **UpdatePanel**, que permite a estos controles ser actualizados sin realizar un "post back". Esto es realmente tan simple como arrastrar el **UpdatePanel** sobre la página en cuestión.

Por su parte, aquellos programadores que estén acostumbrados a trabajar con mucho **JavaScript** sobre el cliente podrán crear aplicaciones **AJAX** que se ejecuten del lado del cliente, trayendo información de servidores remo-

tos para actualizar la página web con la respuesta deseada. La respuesta puede ser un HTML "snippet" o un documento XML, lo que permite actualizaciones simples o complejas sobre la página en desarrollo.

Arquitectura AJAX

En la figura 1 podemos ver la iteración típica de un aplicativo web. Primero, desde un browser se hace una petición a una página; por ejemplo, el usuario escribe la URL. Cuando el requerimiento llega al servidor, se realizan los procesos de la página, pero al completarse se envía al browser del cliente nuevamente la página completa. Debido a este comportamiento, no es posible que el usuario siga interactuando con la página, ya que debe esperar a que se arme nuevamente en su browser. Y así deberá continuar mientras trabaje con esa aplicación web.

Ahora, si vemos la figura 2, donde se muestran las iteraciones de un aplicativo hecho con **AJAX**, podremos observar que deja a un lado la forma de refrescar por completo la página sobre el browser del usuario. Inicialmente, todo el HTML es recuperado y las llamadas posteriores al servidor darán los datos actualizados en XML o Javascript Object Notation (JSON), o snippets de HTML actualizarán la página de manera incremental. Las llamadas asincrónicas en segundo plano pueden acceder a **Web Services** o actualizar páginas sin que el usuario deba realizar una pausa.

Ejemplo

Veamos un simple ejemplo que demuestra lo explicado en esta nota. Recuerde que para realizar este tipo de aplicaciones debe descargar e instalar el framework **AJAX**, sobre el .NET Framework 2.0.

La idea del ejemplo es poner en una página dos labels que contengan la hora del servidor, uno programado con **AJAX** y otro que no lo esté. Pondremos un botón que hará únicamente un "post back" al servidor. Entonces, cuando esta página funcione, deberíamos notar que el label programado con **AJAX** actualiza la hora y el segundo no lo hace.

El siguiente es el código HTML:

```
<body>
  <form id="form1" runat="server">
    <asp:ScriptManager id="ScriptManager1" runat="server" />
    <div>
      <asp:UpdatePanel id="UpdatePanel1" runat="server">
        <ContentTemplate>
          <asp:Label id="lblConAjax" runat="server" Text="Label" />
          <asp:Button id="Button1" runat="server" Text="Button" />
        </ContentTemplate>
      </asp:UpdatePanel>
      <asp:Label id="lblSinAjax" runat="server" Text="Label" />
    </div>
  </form>
</body>
```

El código en el behind de la página será:

```
Protected Sub Page_Load(ByVal sender As Object,
    ByVal e As System.EventArgs)
    Handles Me.Load
    lblConAjax.Text = Date.Now
    lblSinAjax.Text = Date.Now
End Sub
```

A medida que hagamos clic en el botón, sólo se actualizará la hora del primer label y no la del segundo, debido a que no se carga nuevamente toda la página, sino solamente el área necesaria, es decir, aquellos controles que estén dentro del **UpdatePanel**.



Figura 1. Forma típica de iterar un aplicativo web

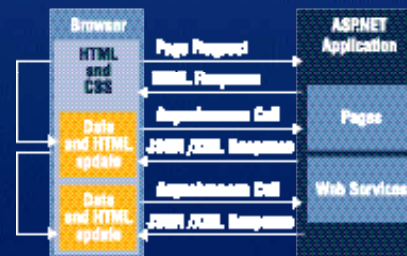


Figura 2. Forma típica de iterar un aplicativo AJAX

Hasta aquí hemos visto las características de **ASP.NET AJAX** y las grandes ventajas que ofrece, además de un ejemplo práctico para comprobar la utilidad de esta gran tecnología, que se está imponiendo en el mundo de Internet.

José Luis Olivarez
Gerente de Desarrollo
División Consultoría
BS - Buffa Sistemas