

¿Qué ofrece Autentia Real Business Solutions S.L?

Somos su empresa de **Soporte a Desarrollo Informático**.
 Ese apoyo que siempre quiso tener...

1. Desarrollo de componentes y proyectos a medida



2. Auditoría de código y recomendaciones de mejora

3. Arranque de proyectos basados en nuevas tecnologías

1. Definición de frameworks corporativos.
2. Transferencia de conocimiento de nuevas arquitecturas.
3. Soporte al arranque de proyectos.
4. Auditoría preventiva periódica de calidad.
5. Revisión previa a la certificación de proyectos.
6. Extensión de capacidad de equipos de calidad.
7. Identificación de problemas en producción.



4. Cursos de formación (impartidos por desarrolladores en activo)

Spring MVC, JSF-PrimeFaces /RichFaces,
 HTML5, CSS3, JavaScript-jQuery

Gestor portales (Liferay)
 Gestor de contenidos (Alfresco)
 Aplicaciones híbridas

Tareas programadas (Quartz)
 Gestor documental (Alfresco)
 Inversión de control (Spring)

Control de autenticación y
 acceso (Spring Security)
 UDDI
 Web Services
 Rest Services
 Social SSO
 SSO (Cas)

JPA-Hibernate, MyBatis
 Motor de búsqueda empresarial (Solr)
 ETL (Talend)

Dirección de Proyectos Informáticos.
 Metodologías ágiles
 Patrones de diseño
 TDD

BPM (jBPM o Bonita)
 Generación de informes (JasperReport)
 ESB (Open ESB)



[Inicio](#) [Quienes somos](#) [Tutoriales](#) [Formación](#) [Empleo](#) [Colabora](#) [Comunidad](#) [Libro de Visitas](#) [Comic](#)

[Ver cursos que ofrece Autentia](#)

[Descargar comics en PDF y alta resolución](#)



[¡NUEVO!] 2008-01-23



2008-01-21



2008-01-17



2008-01-17

Estamos escribiendo un libro sobre la profesión informática y estas viñetas formarán parte de él. Puedes opinar en la sección [comic](#).

Catálogo de servicios Autentia (PDF 6,2MB)



[En formato comic...](#)

Tutorial desarrollado por



Iván García Puebla

Puedes encontrarme en [Autentia](#)

Somos expertos en Java/J2EE

Catálogo de servicios de Autentia

[Descargar \(6,2 MB\)](#)

[Descargar en versión comic \(17 MB\)](#)

[AdictosAlTrabajo.com](#) es el Web de difusión de conocimiento de [Autentia](#).



[Catálogo de cursos](#)

Descargar este documento en formato PDF: [eventosASPNET.pdf](#)

Fecha de creación del tutorial: 2008-01-27

Eventos en ASP.NET

Las tecnologías avanzan en todos los frentes, y Microsoft saca pecho desde hace años con su Framework .NET. Una evolución interesante para el mundo de desarrollo en lenguajes propietarios que siempre aportan avances al mundo tecnológico.

Introducción

En este tutorial veremos los eventos del ciclo de vida de una aplicación web en ASP.NET. Recordemos que ASP.NET se basa en el concepto de Web Form, a diferencia de las aplicaciones de escritorio (los Window Form). Ambos tienen un ciclo de vida diferente y basan su comportamiento en eventos, pero debemos diferenciar ambos conceptos; todo lo que sigue a continuación es propio de los desarrollos web el framework .NET.

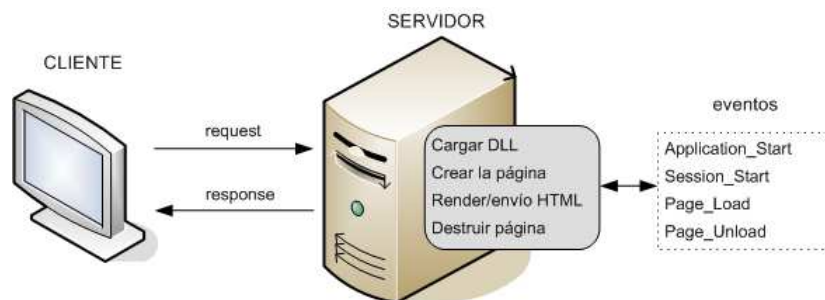
Los eventos en ASP.NET se clasifican en tres grupos:

- Eventos a nivel de aplicación. Son eventos globales de nuestra web, como pueden ser el inicio de la aplicación, finalización de una sesión, etc
- Eventos a nivel de página. Habitualmente lanzados por el usuario, como el submit de un formulario o pulsar sobre un enlace.
- Eventos a nivel de servidor. Son los eventos propios de los controles de servidor de los Web Forms

Antes de entrar en detalles, echemos un vistazo al ciclo de vida de una aplicación web y sus eventos principales.

Eventos en el ciclo de vida de una aplicación web

Una aplicación web inicia propiamente su existencia cuando un cliente se conecta a su página de inicio (es el inicio de una sesión). El servidor web Internet Information Server (IIS) levanta el proceso aspnet_wp.exe (en caso de ser necesario, podría estar ya activo con el servidor dando servicio) y carga el ensamblado DLL que contiene el código programado de la página solicitada y que se ejecuta en lado servidor. El servidor crea una instancia del Web Form solicitado, lo completa accediendo a los recursos e información necesarios, ejecuta la lógica de negocio y emite como respuesta ese Web Form en formato HTML al browser cliente. Tras esto, la instancia del Web Form y sus recursos son liberados de memoria.



Primera petición web de una aplicación.

El ciclo se repite: el usuario rellena un formulario HTML, envía los datos al servidor, éste crea un Web form, le asigna los datos enviados y procesa los eventos, envía el Web form resultante como HTML y se libera junto con sus recursos. Vemos cómo el ciclo de vida de un Web Form es muy corto.

Google

☐ Web

☒ www.adictosaltrabajo.com

Últimos tutoriales

2008-01-27
[Eventos en ASP.NET](#)

2008-01-23
[Icefaces, JBoss, Maven2 y EJB3: Parte 5](#)

2008-01-21
[Icefaces, JBoss, Maven2 y EJB3: Parte 4](#)

2008-01-20
[Crap4j, ¿es tu código difícilmente mantenible?](#)

2008-01-19
[SpringIDE, plugin de Spring para Eclipse](#)

2008-01-18
[Búsqueda de dependencias para maven](#)

2008-01-18
[Icefaces, JBoss, Maven2 y EJB3: Parte 3](#)

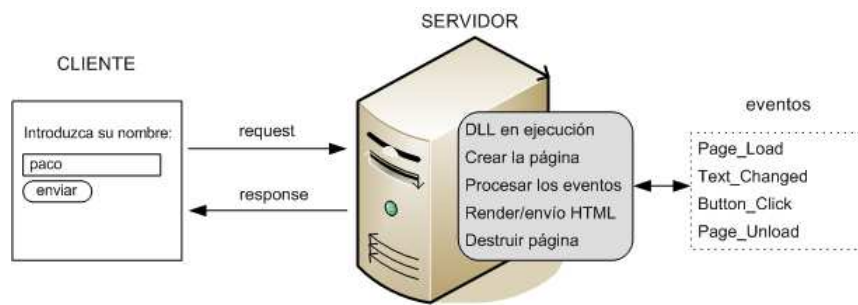
2008-01-17
[Icefaces, JBoss, Maven2 y EJB3: Parte 2](#)

2008-01-17
[Icefaces, JBoss, Maven2 y EJB3: Parte 1](#)

2008-01-17
[Como integrar tareas Ant en Maven](#)

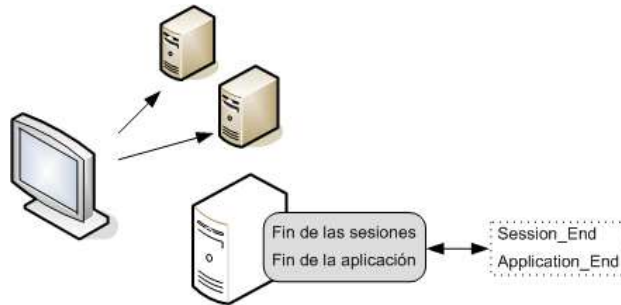
Últimas ofertas de empleo

2008-01-10
[T. Información - Analista / Programador - MADRID.](#)



Continúa la navegación por la web (e.g. proceso de un formulario)

Sin embargo, la aplicación web sigue viviendo hasta que todas las sesiones a las que está dando servicio finalizan. Tras un tiempo sin atender nuevos eventos en una sesión, el servidor destruye la sesión, y cuando no queda ninguna activa, finaliza la aplicación web (el CLR libera de memoria los recursos de la aplicación tales como el DLL cargado) y se lanza un evento Application_End.



No hay clientes activos: las sesiones expiran y la aplicación web correspondiente finaliza

Debemos aclarar que por sesión entendemos una instancia de navegador emitiendo peticiones a un servidor. Un usuario puede tener múltiples navegadores abiertos y por tanto múltiples sesiones iniciadas.

Si queremos que la información persista entre las distintas instancias volátiles y sin estado de los Web form, podemos hacer uso de dos niveles de ámbito en el servidor:

- Variables a nivel de aplicación. Es un espacio global existente en todo el ciclo de vida de la aplicación web, y accesible desde todas las sesiones a las que ésta da servicio en un determinado momento.
- Variables a nivel de sesión. La aplicación web tiene reservados tantos espacios de memoria reservados como sesiones tenga activas y reducidas a su ámbito e inaccesibles fuera de ellas (independientes entre sí).

Eventos de las aplicaciones y de las sesiones

Vamos a enumerar los manejadores de (métodos que capturan) los eventos y a ponerlos en práctica con pequeños ejemplos en C# programados con Visual Studio .NET.

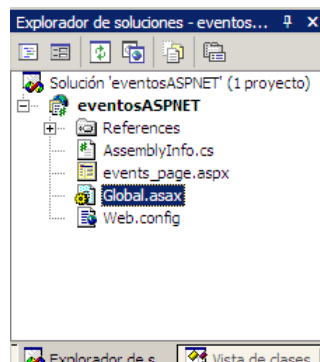
La signatura de los métodos son de la forma

protected void <nombre_manejador> (Object sender, EventArgs e)

siendo <nombre_manejador> éstos:

Manejador de evento	Acción asociada
Application_Start	Primera visita a la página inicio de la aplicación web
Application_BeginRequest	Inicio petición al servidor (solicitar una página)
Application_EndRequest	Fin de petición al servidor (la página ya ha sido solicitada)
Application_End	Todas las sesiones de usuario expiran
Application_AuthenticateRequest	Petición de autenticación
Application_Error	La aplicación lanza un error
Session_Start	Un nuevo usuario solicita una página de la aplicación web
Session_End	Un usuario deja de hacer peticiones a la aplicación web y su tiempo de sesión expira.

Los manejadores de eventos se encuentran definidos en el fichero del proyecto Global.asax:



Vamos a escribir un ejemplo que cuente el número de veces que el servidor recibe eventos de un internauta como nosotros. Para ello utilizamos los mapas que ofrece el IIS como variables de estado para almacenar objetos en memoria en sus distintos ámbitos ('Application' para aplicación y 'Session' para sesión) :

```
protected void Application_Start(Object sender, EventArgs e)
{
    //Initializations os the counters at the beginning:
    Application["activeApplications"]=0;
    Application["activeSessions"]=0;

    //Increase the respective event

    Application["activeApplications"]=(int)Application["activeApplications"]+1;
}

protected void Session_Start(Object sender, EventArgs e)
{
    //Increase the respective event
    Application["activeSessions"]=(int)Application["activeSessions"]+1;

    // Once in a session, we can send text to the browser:
    Response.Write("<p>Web Application events:
"+Application["activeApplications"]+"<br/>Active sessions:
"+Application["activeSessions"]+"</p>");
}

protected void Application_BeginRequest(Object sender, EventArgs e)
{
    Response.Write("<p><b>[Application_BeginRequest]</b>: begin of browser
request</p>");
}

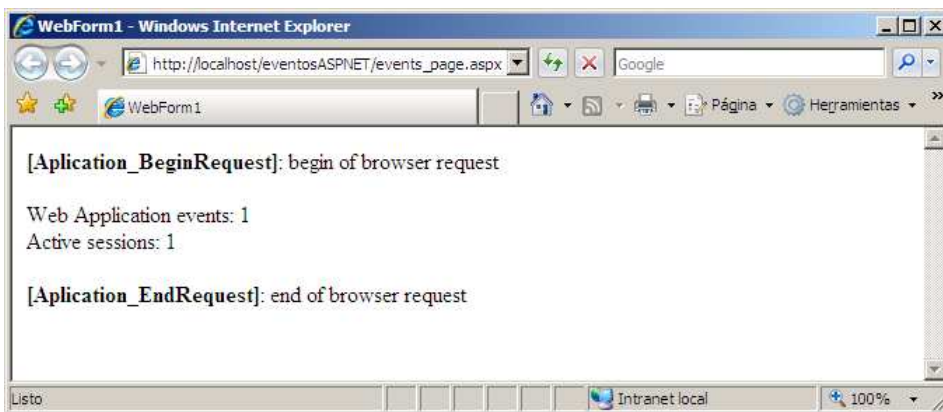
protected void Application_EndRequest(Object sender, EventArgs e)
{
    Response.Write("<p><b>[Application_EndRequest]</b>: end of browser
request</p>");
}

protected void Session_End(Object sender, EventArgs e)
{
    //the session has expired. Is this web site too boring?
    Application["activeSessions"]=(int)Application["activeSessions"]-1;
}

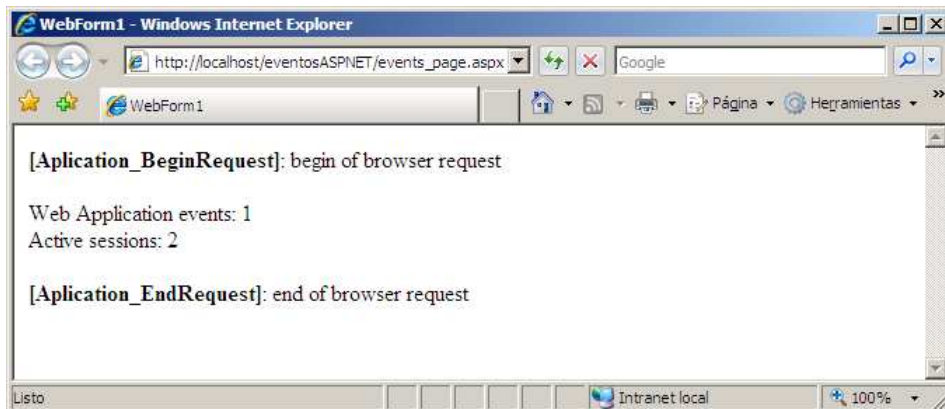
protected void Application_End(Object sender, EventArgs e)
{
    //no users in our web site

    Application["activeApplications"]=(int)Application["activeApplications"]+1;
}
```

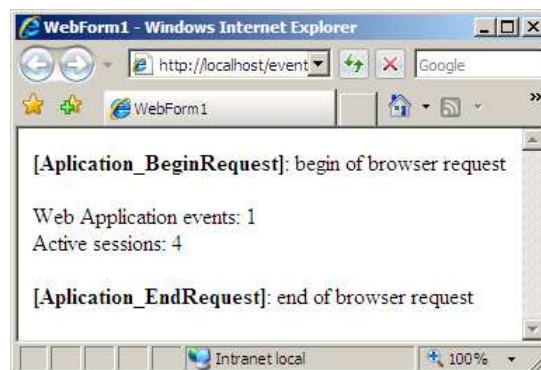
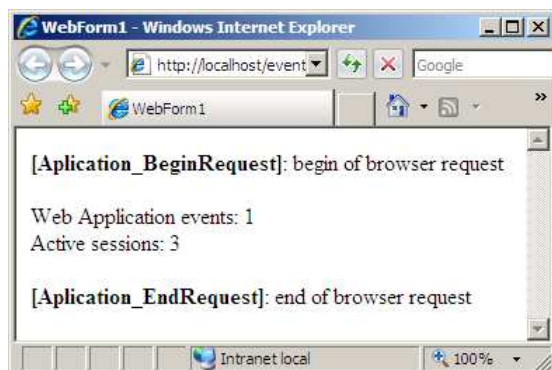
Una vez compilada la aplicación accedemos a la URL del ejemplo: http://localhost/eventosASPNET/events_page.aspx y vemos algo así:



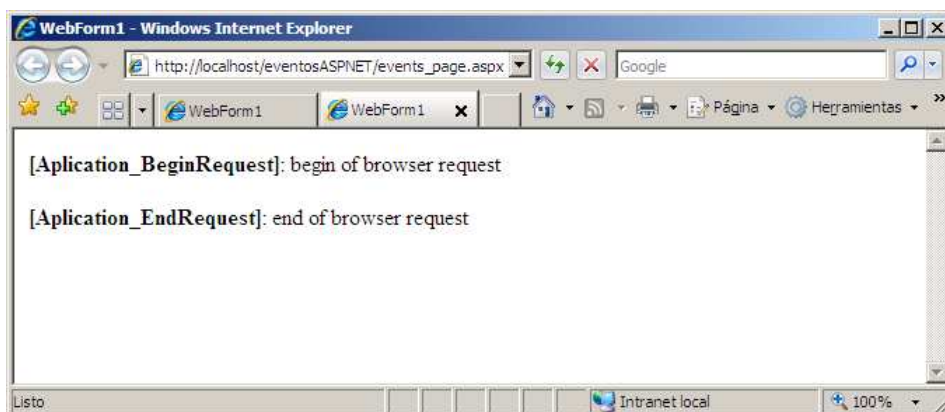
Desde otro navegador, accedemos a la misma aplicación web:



Y más!:



Ahora bien: dos pestañas de un mismo navegador Internet Explorer se interpretan como dos peticiones dentro de una misma sesión (son dos instancias de un mismo browser contenedor). Por en la siguiente captura sólo se muestran las trazas de la request (no se captura ningún evento de inicio de sesión porque ésta ya la inició la pestaña de la izquierda):



Eventos de los Web Form

Si una página la hemos programado con un Web Form, podemos utilizar los siguientes eventos para capturar una amplia variedad de situaciones:

Manejador de evento	Acción asociada
Page_Init	El servidor crea una instancia del Web Form, carga sus controles y los inicializa con lo recibido del estado de la vista (datos del browser del cliente que vienen con la request).
Page_Load	Los controles del servidor se cargan en un nuevo objeto de tipo Page. Ahora el programador puede acceder a la información de la vista y manejar los controles de la página, contenido, etc.
Page_PreRender	El objeto Page va a renderizarse en código interpretable por el browser en cuanto finalice este evento.
Page_Unload	El objeto Page deja de ser referenciable, se destruye.
Page_Disposed	La memoria ocupada por Page es liberada.
Page_Error	Evento iniciado cuando se ha producido una excepción no capturada.
Page_AbortTransaction	Una operación transaccional ha sido finalizada sin llegar a su fin.
Page_CommitTransaction	La transacción se ha completado con éxito.
Page_DataBinding	Un control de servidor se ha conectado a una fuente de datos.

Eventos de los controles de servidor

Los controles de servidor son aquellos elementos que introducimos en nuestros Web Forms y pueden lanzar acciones, e.g. botones, listas, áreas de texto, etc. Si hemos trabajado antes con ellos seguro que nos vendrá a la mente el hecho de su variedad en función de sus propósitos y acciones que se permiten al usuario. Es por esto que los eventos, concretando este razonamiento, han sido divididos en tres grupos:

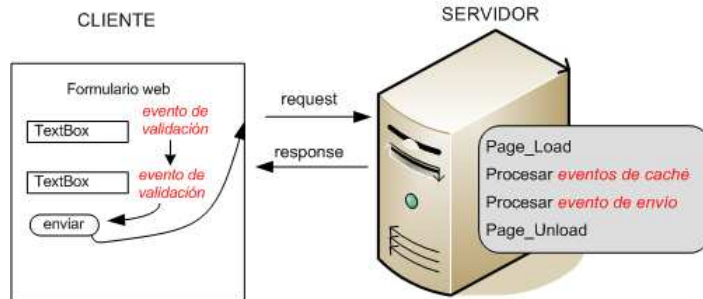
- Eventos de envío. Son los eventos lanzados por controles que emiten un envío (post) al servidor de manera inmediata para

procesar el Web Form. Por ejemplo al enviar un formulario pulsando sobre el botón de aceptación. Estos controles son: Button, Link Button e Image Button.

- Eventos de caché. Estos eventos se producen en la vista y serán procesados en el servidor cuando se envíe la información mediante un evento de envío. Por ejemplo el seleccionar un elemento de una lista provoca un cambio de estado de la misma que luego, en servidor, podremos obtener. Controles con eventos de caché son TextBox, DropDownList, CheckBox...
- Eventos de validación. Son los eventos que se procesan dentro de la página web. El ejemplo típico es la validación de que un campo de entrada está vacío o no cuando es obligatorio, usando capacidades de validación del propio componente.

En el ciclo de vida de un Web Form se aplica el siguiente algoritmo de manejo de eventos:

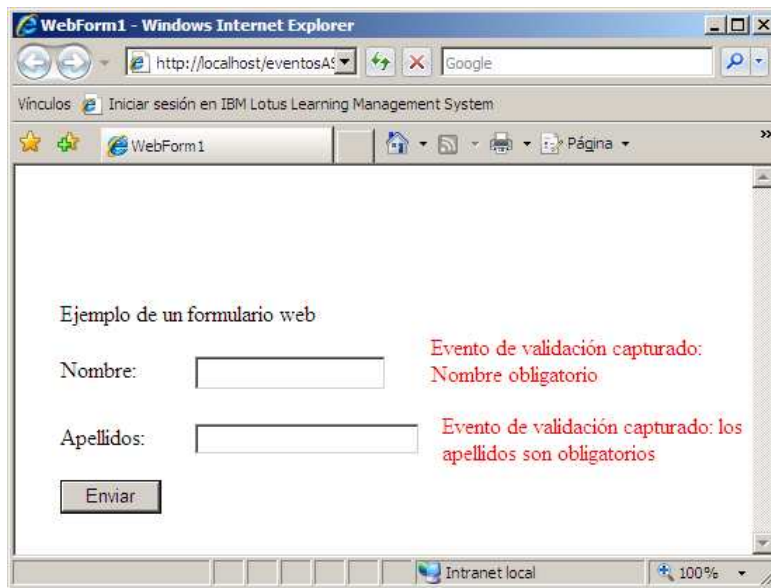
1. Los controles con validación hacen las validaciones en el propio cliente antes de que la página sea enviada al servidor (se ha tenido que producir un evento de envío previamente).
2. La página se envía y se suceden los eventos Page_Init y Page_Load. Se procesan los eventos cacheados por orden temporal y finalmente se procesa el evento de envío.
3. Nuestra lógica de negocio realiza las acciones pertinentes con los controles.



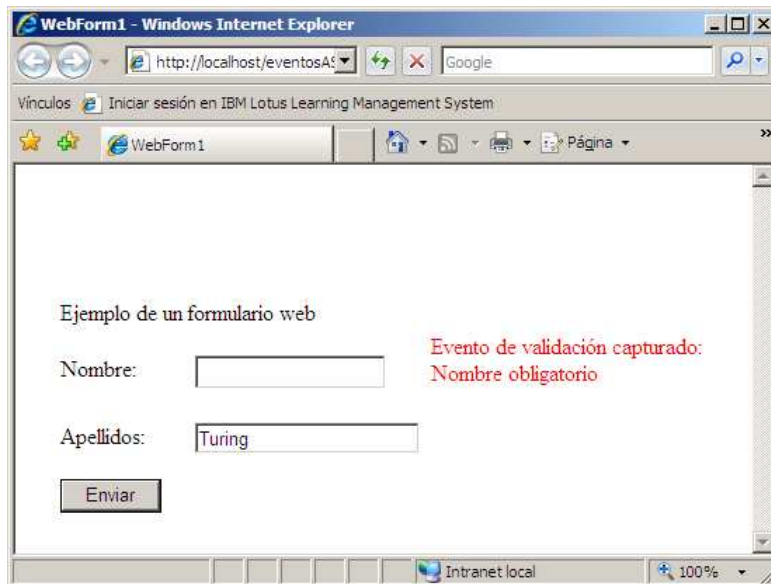
Orden de los eventos de los controles de servidor; 1º: validación, 2º: cache, 3º: envío.

Vamos a utilizar los eventos de Web Form y de controles en otro ejemplo. Conforme a la imagen anterior, tenemos un Web Form con dos campos de texto Nombre y Apellidos. Ambos son obligatorios y decidimos controlarlo en la propia página (con un componente RequiredFieldValidator). Una vez todo es correcto, se hace la request y en servidor mostramos la secuencia de manejo de eventos.

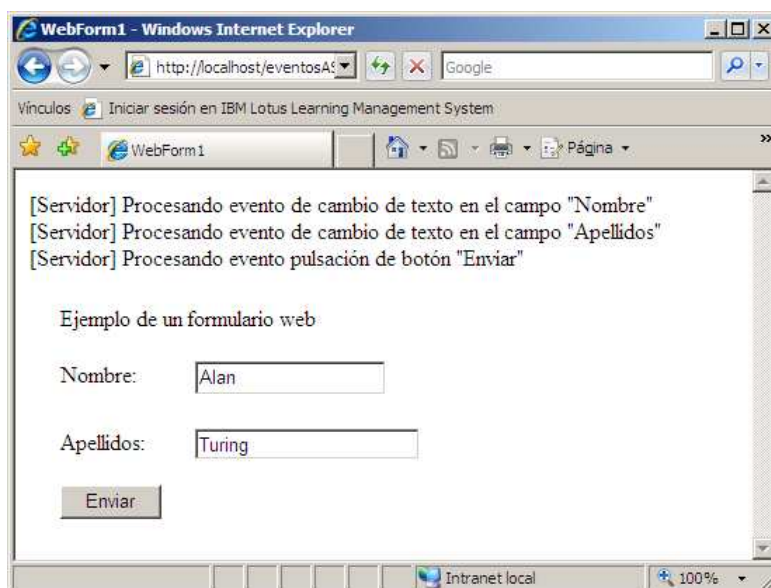
La primera imagen muestra una pulsación del botón Enviar con los campos obligatorios vacíos. el propio navegador gestiona los eventos de validación y se omite la request:



Lo mismo ocurre si dejamos un campo vacío y pretendemos hacer el envío:



Finalmente introducimos los dos campos y obtenemos por pantalla el orden por el que el servidor maneja los eventos del Web Form:



El código para esto es simple y se aloja en la clase C# `events_page.aspx.cs` asociado al Web Form `events_page.aspx`:

```
private void btnEnviar_Click(object sender,
System.EventArgs e)
{
    Response.Write("[Servidor] Procesando evento pulsación
de botón \"Enviar\"<br/>\n");
}

private void txtNombre_TextChanged(object sender,
System.EventArgs e)
{
    Response.Write("[Servidor] Procesando evento de cambio
de texto en el campo \"Nombre\"<br/>\n");
}

private void txtApellidos_TextChanged(object sender,
System.EventArgs e)
{
    Response.Write("[Servidor] Procesando evento de cambio
de texto en el campo \"Apellidos\"<br/>\n");
}
```

Conclusiones

Un buen desarrollo web con .NET comienza dominando la teoría de los eventos de nuestra aplicación, saber cuales son, donde se procesan y cómo podemos manejarlos en nuestra lógica programada. Los eventos son la interfaz aplicación/programador que permiten desarrollos rápidos de aplicaciones, característica esencial del framework .NET.

En Autentia no dejamos de lado estas tecnologías; las estudiamos, las aplicamos a nuestros proyectos y, como siempre, compartimos la información con vosotros. Deseamos que esto resulte útil para favorecer el desarrollo de software con calidad.

- Puedes opinar sobre este tutorial [haciendo clic aquí](#).
- Puedes firmar en nuestro libro de visitas [haciendo clic aquí](#).
- Puedes asociarte al grupo AdictosAlTrabajo en XING [haciendo clic aquí](#).
- Añadir a favoritos Technorati.  ADD THIS BLOG TO MY **Technorati FAVORITES**



Esta obra está licenciada bajo [licencia Creative Commons de Reconocimiento-No comercial-Sin obras derivadas 2.5](#)

Recuerda

Autentia te regala la mayoría del conocimiento aquí compartido ([Ver todos los tutoriales](#)). Somos expertos en: J2EE, Struts, JSF, C++, OOP, UML, UP, Patrones de diseño ... y muchas otras cosas.

¿Nos vas a tener en cuenta cuando necesites consultoría o formación en tu empresa?, ¿Vas a ser tan generoso con nosotros como lo tratamos de ser con vosotros?

Somos pocos, somos buenos, estamos motivados y nos gusta lo que hacemos ...

Autentia = Soporte a Desarrollo & Formación.

info@autentia.com



Servicio de notificaciones:

Si deseas que te enviemos un correo electrónico cuando introduzcamos nuevos tutoriales.

Formulario de subcripción a novedades:

E-mail

Tutoriales recomendados

Nombre	Resumen	Fecha	Visitas	pdf
AdRotator en Página ASP .Net	Ismael Caballero nos cuenta como utilizar el control AdRotator en una página ASP .Net	2004-07-17	8596	pdf
Gestión de errores con IIS	Ismael caballero nos enseña como evitar que aparezan pantallas genericas de error cuando trabajamos con IIS	2004-01-16	10114	pdf
Certificados en IIS para activación SSL	En este tutorial vamos a habilitar el soporte SSL (Secure Socket Layer, comunicación segura por https) en un servidor IIS (Internet Information Server de Microsoft).	2005-09-09	7206	pdf
Creación y utilización de clases en páginas ASP	En el presente tutorial se explicará la implementación de una clase utilizando la tecnología ASP (Active Server Pages) de Microsoft, así como la manera de instanciar y aplicar sus métodos en páginas ASP	2006-08-16	3869	pdf
Mailings con ASP y ficheros	Ismael Caballero nos muestra una variación de su tutorial para enviar mailings con tecnología asp y ficheros (sin necesidad de base de datos)	2004-01-08	8937	pdf
Activar SSL en IIS	Os mostramos como activar el soporte de https en IIS, creando vuestros propios certificados autofirmados, usando OpenSSL	2003-08-07	26455	pdf
Mailings con asp y Base de Datos	Ismael Caballero describe el proceso de realización de un sistema para el envío de correos (mailings) de forma automática utilizando un sistema de base de datos (Microsoft SQL Server).	2004-01-08	10305	pdf
Como configurar Tomcat + IIS	Os mostramos como activar el filtro ISAPI de IIS para conectarlo con Tomcat. De este modo el servidor Web IIS 5.5 de Microsoft servirá las páginas estáticas y Tomcat los JSPs y Servlets dinámicos	2005-12-11	11539	pdf
Autenticación basada en formularios (IIS y .Net).	Luis Paniagua nos enseña como realizar la configuración de un directorio privado en el servidor IIS y con .Net FrameWork instalado, al que accederemos mediante la validación de usuario en un formulario Web.	2004-07-09	12288	pdf

Nota:

Los tutoriales mostrados en este Web tienen como objetivo la difusión del conocimiento. Los contenidos y comentarios de los tutoriales son responsabilidad de sus respectivos autores. En algún caso se puede hacer referencia a marcas o nombres cuya propiedad y derechos es de sus respectivos dueños. Si algún afectado desea que incorporemos alguna reseña específica, no tiene más que solicitarlo. Si alguien encuentra algún problema con la información publicada en este Web, rogamos que informe al administrador rcanales@adictosaltrabajo.com para su resolución.