

Tema 9; Aplicaciones Web y Capa de Interfaz

Aplicaciones web vs escritorio

Simple creación de interfaz de usuario.

Distribución de actualizaciones más fácil y rápido y menos costoso.

Procesamiento distribuido.

La web provee protocolos estándar (HTTP, HTML, XML) para facilitar aplicaciones n-capa.

Tecnología dependiente del servidor

La ventaja principal radica en la seguridad que tiene el programador sobre su código

Qué es ASP.NET?

Plataforma de para construir Aplicaciones Web y Servicios Web que funcionan bajo IIS.

Aplicaciones Web ASP.NET

Combinación de archivos, páginas, manejadores, módulos y código ejecutable que puede invocarse desde un directorio virtual.

Se dividen en varias páginas web.

Comparten un conjunto de recursos y opciones de configuración común.

Una aplicación web sólo existe en un Directorio Virtual, es un recurso compartido identificado por un alias que representa la localización física en el servidor.

Formularios Web

Técnicas para Rápido Desarrollo de Aplicaciones (RAD).

La aplicación se desarrolla para un servidor web.

Los usuarios interactúan con la aplicación a través de un navegador.

Proporcionan una aproximación orientada a: Objetos, eventos y gestión de estado

Programación del lado del servidor para manejar eventos del lado del cliente: pueden ejecutarse, virtualmente, sobre cualquier navegador compatible con HTML

Visual Studio dispone de su propio servidor de desarrollo, por lo que no necesitamos tener instalado IIS (Internet Information Server).

Code-inline vs Code-behind

Único archivo ("Code-inline")

Archivos separados ("Code-behind")

Páginas maestras

Las páginas maestras permiten crear un diseño coherente para las páginas de la aplicación.

Se puede definir el aspecto, el diseño y el comportamiento estándar que desea que tengan todas las páginas (o un grupo de páginas) de la aplicación en una sola página maestra.

Realización de cambios de diseño en una sola ubicación.

Reutilización de la interfaz de usuario

Master pages definen el contenido común y los contenedores de contenido, hacen referencia a las paginas maestras y llenan a los contenedores con su contenido.

Desde la perspectiva del usuario, da como resultado una única página.

La dirección URL de esta página es la de la página de contenido.

Secuencia de control IIS (En tiempo de ejecución)

por primera vez se carga la maestra y la página de contenido

por segunda vez ya no hace falta compilar la página maestra solo la de contenido, la página maestra se va actualizando en las demás de contenido.

Controles de servidor

Tienen propiedades que pueden ser establecidas declarativamente o mediante programación.

Soporte para características avanzadas: enlace de datos, plantillas..
Se emplea el prefijo asp: junto con el atributo runat="server".

Tipos de controles ASP.NET

Label	Se utiliza para mostrar texto dinámico (cambiamos sus propiedades a través del código del servidor)
Hyperlink	Muestra un enlace a otra página
TextBox.	Permite introducir texto al usuario. Propiedad Textmode valores: Single, Multiline o Password
Image.	Sirve para mostrar una imagen en la página web
Button .	Se usa en un Formulario web para crear un control de tipo submit (evento OnClick) o un control de comando tipo botón (evento OnCommand)
LinkButton	Apariencia de hipervínculo pero funciones de control de un botón (envío formulario)
ImageButton.	Muestra una imagen que maneja eventos tipo click
Checkbox.	Sirven para añadir casillas de verificación a una página
RadioButton.	Crea un botón de radio individual en la página
DropDownList.	Proporciona un buen método para que los usuarios elijan elementos de una lista en un espacio pequeño
ListBox.	Propiedad SelectionMode: Single, Múltiple. Cada elemento se crea con un control ListItem
CheckBoxList.	Permite presentar una lista de opciones pudiendo el usuario seleccionar varias
RadioButtonList.	Permite crear una lista de botones de radio de opciones excluyentes
Panel .	Puede usarse como contenedor de otros controles
Table, TableRow, TableCell.	Permiten crear dinámicamente una tabla mediante programación

Modelo de eventos

Un evento es un mensaje que envía un objeto cuando ocurre una acción.

En una aplicación, se debe traducir el mensaje en una llamada a un método del código.

En las páginas Web ASP.NET, los eventos asociados a los controles de servidor se originan en el cliente pero los controla la página ASP.NET en el servidor Web.

La información del evento se captura en el cliente y se transmite un mensaje al servidor mediante un envío HTTP.

La página debe interpretar el envío para determinar el evento ocurrido y, a continuación, llamar al método apropiado del código del servidor para controlar dicho evento.

.NET Framework define un tipo especial que proporciona la funcionalidad de un puntero a función.

Objeto evento

El objeto del evento se conoce como remitente.

El objeto que captura el evento es el receptor.

En las comunicaciones de eventos, el remitente del evento no sabe qué objeto o método recibirá los eventos que provoca.

Se necesita un intermediario entre el origen y el receptor.

El enlace entre el mensaje del evento y un método específico se lleva a cabo utilizando un delegado de eventos.

Clase delegado

es una clase que puede guardar una referencia a un método

tiene un prototipo y puede guardar referencias únicamente a los métodos que coinciden con su prototipo.
equivale a un puntero a función con seguridad.

los delegados de evento de .NET Framework tienen dos parámetros, el origen que provocó el evento y los datos del evento

Los delegados personalizados sólo son necesarios cuando un evento genera datos de evento.

Los delegados son de multidifusión, pueden guardar referencias a más de un método de control de eventos.

Manejadores de eventos

El prototipo de los métodos manejadores de eventos deben coincidir con el prototipo del delegado EventHandler.

los manejadores de evento en ASPnet devuelven void

tienen dos parámetros: Objeto que lanza el evento y Argumentos del evento (información específica del evento)

Crear una instancia del delegado utilizando una referencia al método controlador de eventos.

Cuando se llama a la instancia de delegado, ésta, a su vez, llama al método controlador de eventos.

Agregar la instancia de delegado al evento: cuando se provoca el evento, se llama a la instancia de delegado y a su método de controlador eventos asociado.

Asociar el manejador al control

Seleccionar el evento de la ventana de propiedades del control

se hace cuando se están creando controles mediante programación.

crear una instancia del delegado EventHandler, a la que se debe pasar la dirección del método al que se va a enlazar.

Después agregar el objeto delegado a la lista de métodos a los que se llama cuando se produce el evento.

Tipos de eventos

Eventos de envío (PostBack). Son los eventos lanzados por controles que emiten un envío (post) al servidor de manera inmediata para procesar el Web Form.

Eventos de caché (NO-PostBack). Estos eventos se producen en la vista y serán procesados en el servidor cuando se envíe la información mediante un evento de envío.

Eventos postback vs no-postback

Una página se carga después de cada petición: fenómeno conocido comoPostBack (=envío).

Eventos no-PostBack (o cached): la información se envía en el siguiente evento postback.

Los eventos se guardan en una cola en el cliente hasta que un evento postback ocurre.

Button, Link Button y Image Button causan eventos postback.

TextBox, DropDownList, ListBox, RadioButton y CheckBox, proveen eventos cached.

Sin embargo podemos sobrecargar este comportamiento en los controles para poder realizar eventos postback, cambiando la propiedad AutoPostBack a true.

Eventos de página

Las páginas ASP.NET provocan eventos de ciclos de vida como Init, Load, PreRender y otros.

De manera predeterminada, los eventos de página se pueden enlazar a los métodos utilizando la convención de nomenclatura Page_nombreDeEvento.

Las páginas ASP.NET enlazan automáticamente los eventos de páginas a los métodos que tienen el nombre Page_evento.

Este enlace automático lo configura el atributo AutoEventWireup de la directiva @ Page, cuyo valor predeterminado es true. Si establece en false, la página no busca automáticamente los métodos Page_evento

Importante: Los métodos de control de eventos de página no requieren ningún argumento. Estos eventos (como Load) pueden aceptar los dos argumentos estándar, pero en estos argumentos no se pasa ningún valor.

Maquetación con CSS

CSS es una herramienta que permite definir la presentación de un documento

Permite crear un conjunto de estilos en una única localización

Las páginas a las que se aplica la misma hoja de estilos tendrán las mismas fuentes, colores y tamaño Proporcionan una estética homogénea en todas las páginas de un sitio web

Maquetar una página web es pasar el diseño a código HTML

Cada Explorador renderiza de manera distinta cada documento HTML

Divs

Las capas, layouts o divs son la misma cosa con distinto nombre, para tener un concepto mental de lo que son, podemos imaginarlos como contenedores o bloques donde podemos meter lo que queramos dentro a los que se le asigna un ancho, alto y posición, de esta manera se van a ir posicionando consiguiendo la estructura que queremos.

Los elementos DIV pueden centrarse utilizando los atributos `margin-left: auto; margin-right: auto;`

La diferencia entre una clase (.) y un bloque (#) es que el bloque es único e irrepetible en la página, en cambio si lo convertimos en una clase podremos usarlo cuantas veces queramos.

En las estructuras clásicas con tablas, podíamos utilizar tamaños en porcentaje. Aunque aquí, también podemos utilizar porcentajes, el dibujo de la página puede no verse como se esperaba.

Método Response.Redirect

Navega a otra página por medio del código.

Los hipervínculos responden a eventos click mostrando la página especificada en la propiedad `NavigateURL` del control.

Si quieres capturar un click en el código, debes usar los eventos de un `LinkButton` o `ImageButton` y utilizai

Utilizar los métodos de la clase `HttpServerUtility` para codificar los datos

`Response.Redirect("WebForm2.aspx?par1 =" + Server.UrlEncode("&hola "));`

Paso de parámetros a en la URL. QueryString

Almacena los datos en la colección `QueryString`

Los caracteres utilizados deben ser caracteres permitidos en una URL

La información es visible a los usuarios en la barra del navegador

Los usuarios pueden modificar la información provocando errores inesperados

Muchos navegadores imponen un límite en la longitud de la URL

`&` (para separar múltiple query strings)

`+` (alternativa para representar un espacio)

`#` (especifica un marcador en una página web) Solución:

UrlEncode

reemplaza los caracteres especiales por secuencias de escape

Objeto Request

proporciona información sobre la petición HTTP del cliente que ha provocado la carga de la página actual.

Información sobre el cliente (`Request.Browser`, `Request.Browser.IsMobileDevice`, `Request.Browser.Id`)

Cookies (`Request.Browser.Cookies`)

Parámetros pasados a la página:

`if (Request.QueryString["par1"] != "") Label1.Text = Request.QueryString["par1"];`