



Escuela de Sistemas y Tecnologías

Transparencias de ANALISTA DE SISTEMAS
Edición 2017 – Materia: Diseño de Aplicaciones Web

TEMA: ASP.NET Avanzado

Plantel y Contactos

➤ **Bedelía:**

- Mail: bedeliasistemas@bios.edu.uy

➤ **Encargado de Sucursal:**

- Pablo Castaño
- Mail: pablocasta@bios.edu.uy

Recursos

➤ Recursos Imprescindibles:

- Sitio Web de material
(comunicarse con Bedelía por usuario/contraseña).
- Transparencias del Curso.
- Contar con el software necesario

Agenda

- ☐ Controles de Navegación
- ☐ Controles de Datos
- ☐ Autenticación y Autorización

Controles de Navegación

Controles de Navegación

- El sistema de navegación de sitios ofrece a los desarrolladores un mecanismo para definir un mapa del sitio y una API para dicho mapa que se debe consultar mediante código.
- Los controles web de navegación son:
 - SiteMapPath
 - TreeView
 - Menu
- Facilitan la transformación total o parcial del mapa del sitio en un elemento de navegación de la interfaz de usuario.

Navegación de Sitios (1)

- Se pueden utilizar las características de exploración del sitio de ASP.NET con el fin de proporcionar una manera coherente a los usuarios para que exploren el sitio.
- Cuando un sitio crece y cuando se mueven sus páginas, puede resultar complicado administrar todos los vínculos.
- La navegación de sitios de ASP.NET permite almacenar los vínculos de todas las páginas en una ubicación central y representar estos vínculos en listas o menús de desplazamiento en cada página incluyendo un control de servidor Web específico.

Navegación de Sitios (2)

- Con ASP.NET puede crear una solución de exploración consistente y fácil de administrar para el sitio.
- La navegación de sitios de ASP.NET cuenta con las características siguientes:
 - **Mapas del sitio:**
 - Puede utilizar un mapa del sitio para describir la estructura lógica del mismo.
 - A continuación, puede administrar la exploración de la página modificando el mapa del sitio cuando se agregan o se eliminan páginas en lugar de modificar los hipervínculos en todas las páginas Web.

Navegación de Sitios (3)

(cont.):

- **Controles ASP.NET:**

- Puede utilizar estos controles para mostrar los menús de exploración en las páginas Web.
- Los menús de exploración se basan en el mapa del sitio.

- **Controles de programación:**

- Puede utilizar la exploración del sitio de ASP.NET en el código para crear controles de exploración personalizados o modificar la ubicación de la información que se muestra en un menú de desplazamiento.

Navegación de Sitios (4)

(cont.):

- **Reglas de acceso:**

- Puede configurar reglas de acceso que muestren u oculten un vínculo en el menú de desplazamiento.

- **Proveedores de mapas de sitio personalizados:**

- Puede crear proveedores de mapas de sitio personalizados que le permitan trabajar con su propio servidor del mapa del sitio.

Exploración del Sitio (1)

- Con la exploración del sitio de ASP.NET, puede describir el diseño del sitio como una jerarquía.
- Por ejemplo, una empresa ficticia dedicada al manejo informático podría tener un sitio compuesto por ocho páginas diseñadas de la manera siguiente:

Inicio

Productos

Hardware

Software

Servicios

Training

Consultoría

Soporte

Exploración del Sitio (2)

- Como primer paso, se deberá crear un mapa o una representación del sitio.
 - Puede describir la jerarquía del sitio en un archivo XML.
- Después de crear un mapa, puede mostrar la estructura de exploración en una página ASP.NET utilizando un control de exploración del sitio.

Mapa del Sitio

- El proveedor de mapas de sitio de ASP.NET predeterminado carga los datos del mapa del sitio como un documento XML.
- Se almacena en caché como datos estáticos cuando la aplicación se inicia.
- Las características de exploración del sitio de ASP.NET dependen de las notificaciones del archivo para mantener los datos de desplazamiento actualizados.
- Cuando se modifica un archivo de mapa del sitio, ASP.NET vuelve a cargar los datos del mapa.

Controles Exploración de Sitio

- Se debe mostrar la estructura de exploración del sitio en las páginas ASP.NET, para que los usuarios puedan desplazarse por el sitio con facilidad.
- Se puede establecer la exploración en las páginas de forma sencilla utilizando cualquiera de los siguientes controles:
 - **SiteMapPath:** muestra una ruta de desplazamiento que indica al usuario la ubicación de la página actual y muestra los vínculos como una ruta de retorno a la página principal.
 - **TreeView:** muestra una estructura de árbol, o de menú, que los usuarios pueden recorrer para llegar a diferentes páginas del sitio.
 - **Menu:** muestra un menú expansible que los usuarios pueden recorrer para llegar a diferentes páginas del sitio.

Control SiteMapPath

- Puede utilizar el control **SiteMapPath** para crear la exploración del sitio sin código y sin enlaces de datos explícitos.
- El control puede leer y representar la información del mapa del sitio automáticamente.
- Permite a los usuarios desplazarse hacia atrás desde la página actual a las páginas superiores en la jerarquía del sitio. No le permite desplazarse hacia delante desde la página actual a otra página más profunda en la jerarquía.

Controles TreeView o Menú

- Los usuarios pueden abrir los nodos y desplazarse directamente a una página determinada.
- Estos controles no leen el mapa del sitio directamente, puesto que ya lo hace el control **SiteMapPath**.
- En su lugar, puede agregar un control **SiteMapDataSource** a una página que pueda leer el mapa del sitio.
- A continuación, puede enlazar el control **TreeView** o **Menu** al control **SiteMapDataSource**, lo que hace que el mapa del sitio se represente en la página.

API de Exploración del Sitio

- Cuando la aplicación Web se ejecuta, ASP.NET expone un objeto **SiteMap** que refleja la estructura del mapa del sitio. Todos los miembros del objeto **SiteMap** son estáticos.
- El objeto **SiteMap**, a su vez, expone una colección de objetos **SiteMapNode** que contienen las propiedades de cada nodo del mapa.
- Puede utilizar los objetos **SiteMap**, **SiteMapNode** y **SiteMapProvider** en su propio código para recorrer la estructura del mapa del sitio o crear un control personalizado para mostrar los datos del mapa del sitio.
- No puede escribir en el mapa del sitio, pero puede modificar sus nodos en la instancia del objeto.

Controles de Datos

SqlDataSource (1)

- El control **SqlDataSource** manipula datos con muy poco (o ningún) código. Se basa en el manejo de clases ADO.NET.
- Permite utilizar un control Web para tener acceso a los datos ubicados en:
 - BDs de Microsoft SQL Server
 - BDs de Oracle
 - Orígenes de datos OLE DB y ODBC
- Se puede utilizar el control **SqlDataSource** con otros controles que muestran datos como:
 - **GridView**
 - **FormView**
 - **DetailsView**

SqlDataSource (2)

- Para ello:
 - Debe proporcionar una cadena de conexión para conectarse a la BD.
 - Debe definir los procedimientos almacenados (SP) o instrucciones SQL que funcionen con sus datos.
- En tiempo de ejecución, el **SqlDataSource** abre automáticamente la conexión a la BD, ejecuta el SP o la instrucción SQL, devuelve los datos seleccionados y cierra la conexión.

SqlDataSource (3)

- Cuando se configura un control **SqlDataSource**:
 - Establecer la propiedad **ProviderName** en el tipo de base de datos (el valor predeterminado es **System.Data.SqlClient**).
 - La propiedad **ConnectionString** en una cadena de conexión que incluya la información necesaria para conectarse a la base de datos.
- El contenido de la cadena de conexión difiere según el tipo de BD al que tiene acceso el control de origen de datos.

SqlDataSource (4)

- En lugar de establecer las cadenas de conexión en tiempo de diseño como valores de propiedad en el control **SqlDataSource**, puede almacenarlas de forma centralizada como parte de los valores de configuración de la aplicación mediante el elemento de configuración **connectionStrings**.
- Se pueden especificar un máximo de cuatro comandos (consultas SQL) para **SqlDataSource**:
 - SelectCommand
 - UpdateCommand
 - DeleteCommand
 - InsertCommand

SqlDataSource (5)

- El control de origen de datos ejecuta los comandos cuando se llama a sus métodos **Select**, **Update**, **Delete** o **Insert** correspondientes.
- El método **Select** se llama automáticamente al llamar al método **DataBind** de la página o de un control enlazado al control de origen de datos.
- Si el control de origen de datos debe ejecutar un comando, también puede llamar explícitamente a cualquiera de los cuatro métodos.

SqlDataSource (6)

- El control **SqlDataSource** puede devolver datos de dos maneras:
 - Como un objeto **DataSet**
 - Como un lector de datos de ADO.NET
- Se puede especificar cuál de los dos formatos se debe devolver, estableciendo la propiedad **DataSourceMode** del control.
- El control **SqlDataSource** puede almacenar en caché los datos que ha recuperado. De forma predeterminada no está habilitado. Se puede habilitar estableciendo la propiedad **EnableCaching**.

SqlDataSource (7)

- Para filtrar datos, debe estar habilitado el almacenamiento en caché para el control **SqlDataSource** y estar especificado un conjunto de datos devueltos por una consulta Select. En este caso se puede filtrar los datos sin necesidad de volver a ejecutar la consulta.
- El control **SqlDataSource** admite una propiedad **FilterExpression** que permite especificar criterios de selección que se aplican a los datos mantenidos por el control de origen de datos

SqlDataSource (8)

- También se pueden agregar parámetros a la expresión de filtro creando objetos **FilterParameters**, que proporcionan valores en tiempo de ejecución para dicha expresión.
- El control **SqlDataSource** admite solicitudes de ordenación del control enlazado cuando **DataSourceMode** se establece en **DataSet**.

ObjectDataSource (1)

- El control **ObjectDataSource** de ASP.NET representa un objeto con funciones de recuperación de datos y de actualización.
- El control **ObjectDataSource** actúa como una interfaz de datos para controles enlazados a datos como los controles **GridView**, **FormView** o **DetailsView**.
- El control **ObjectDataSource** admite una arquitectura de tres niveles y proporciona un modo de enlazar controles de datos a un objeto de nivel medio.
- Establecer la propiedad **TypeName** del control **ObjectDataSource** para especificar el nombre de la clase que se va a utilizar como objeto de origen

ObjectDataSource (2)

- Si el objeto devuelto al control **ObjectDataSource** por el objeto de datos de origen es un objeto **DataSet** o **Datatable**, el control **ObjectDataSource** admite el filtrado mediante la sintaxis de la propiedad **Expression** de la clase **DataColumn**.
- Filtrar permite exponer sólo las filas que coincidan con un criterio de búsqueda determinado, sin tener que volver a consultar el origen de datos con nuevos criterios de selección.

Repeater (1)

- El control **Repeater** representa una lista de sólo lectura de un conjunto de registros devueltos desde un origen de datos.
- El control no especifica un diseño integrado. En su lugar, se crea el diseño utilizando plantillas.
- Cuando se ejecuta la página, el control **Repeater** recorre cada uno de los registros del origen de datos y representa un elemento para cada registro.
- Dado que no dispone de una apariencia predeterminada, se puede utilizar para crear varios tipos de listas (ej: diseño de tabla, lista delimitada por comas, lista con formato XML).

Repeater (2)

- Plantillas admitidas por el control **Repeater**:
 - **ItemTemplate**: Contiene elementos HTML y controles que se representan una vez por cada elemento del origen de datos.
 - **AlternatingItemTemplate**: Contiene elementos HTML y controles que se representan una vez por cada dos elementos del origen de datos.
 - **HeaderTemplate** y **FooterTemplate**: Contiene el texto y los controles que se representan al inicio y al final de la lista, respectivamente.

Repeater (3)

- Cuando se agregan controles en una plantilla se debe utilizar la sintaxis de enlace de datos, para enlazar el control individual a un campo devuelto por el origen de datos.
- **Eventos:**
 - **ItemCommand:** Se produce en respuesta a los clics que se hagan en controles tipo botón. El **Repeater** permite incrustar controles **Button**, **LinkButton** o **ImageButton** en una plantilla de elemento, y obtener una notificación cuando se haga clic en el botón.

Autenticación y Autorización

Definición

- **Autenticación** es el proceso que se lleva a cabo para validar a un usuario mediante sus credenciales:
 - El usuario ingresa sus credenciales (usuario y contraseña) y el proceso verifica que la información sea correcta.
- **Autorización** es el proceso que se realiza luego a la autenticación y establece cuales son los permisos que el usuario tiene sobre la aplicación una vez que la misma lo validó:
 - Se le asignan permisos para las funcionalidades de la aplicación a los usuarios autenticados.