

Desarrollador de Aplicaciones Web

Programación Web III



Departamento de Ingeniería e Investigaciones Tecnológicas

Master Pages & Controles de Navegación

Ing. Mariano Juiz
Ing. Matias Paz Wasiuchnik
Ing. Pablo Nicolás Sanchez

Agenda

1. Master Pages
2. Controles de Navegación

Master Pages

Introducción

Permite definir una estructura y elementos de interfaz comunes para el sitio web

- Cabecera de Página
- Pie de Página
- Menú

Contiene controles, códigos, etc. (como cualquier otra página)

Es donde generalmente se ubica el único formulario (form)

Tienen extensión **.master**

Se puede cambiar la Master page en el evento PreInit

Ventajas

- Mejora el Mantenimiento
- Evita la duplicación de código de estructuras y comportamiento

Content Pages

En la Master Page se debe identificar mediante el control **ContentPlaceHolder**

También puede contener contenido por defecto, por si la página derivada no necesita sobrescribir este contenido.

Una página sólo deriva de una Master Page cuando se define el atributo `MasterPageFile` en su directiva `Page`

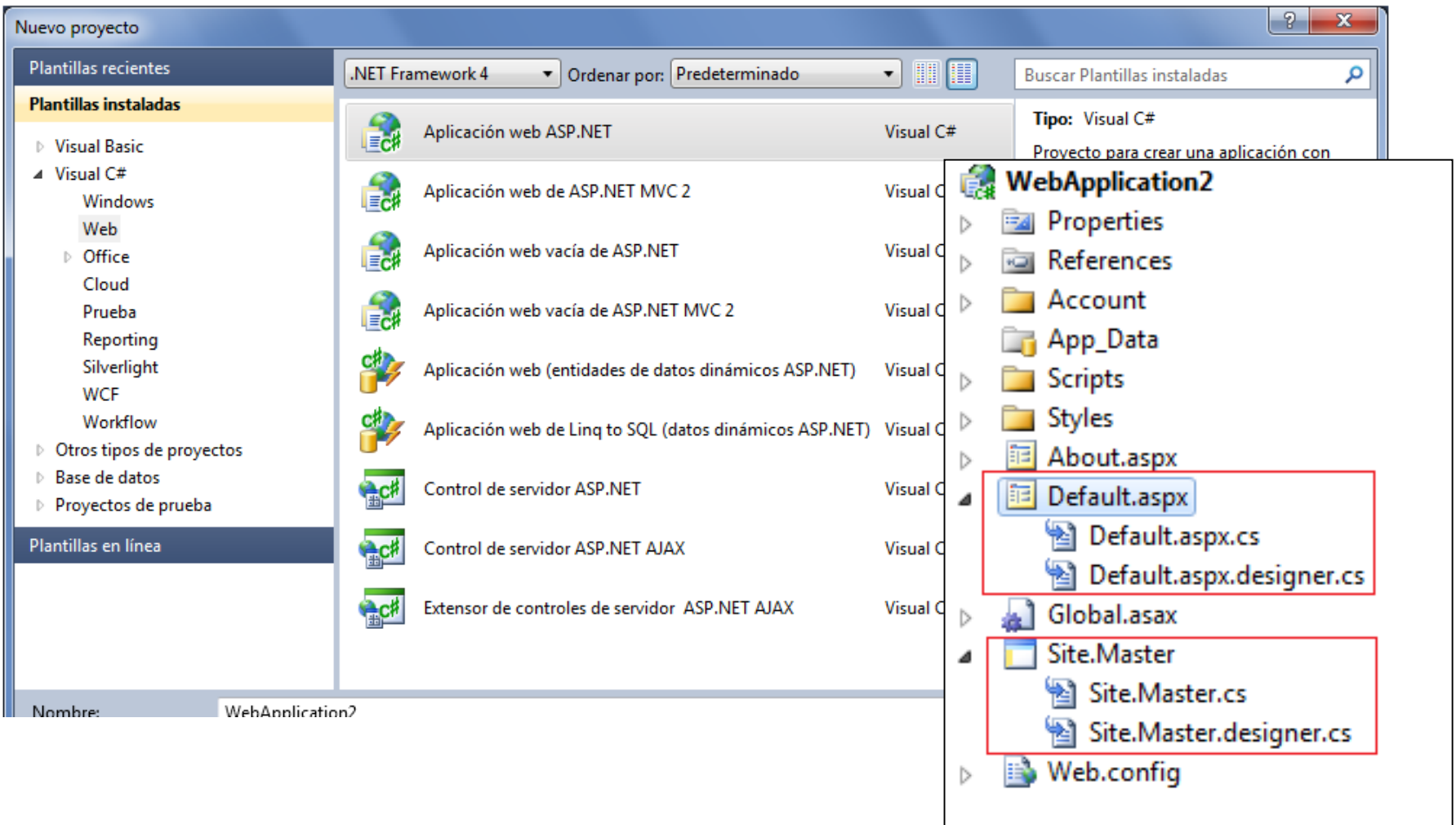
Una Content Page puede declarar controles **asp:Content** que sobre-escriben el contenido de las secciones marcadas en la Master Page.

Además, se asocia a un control **ContentPlaceHolder** de la Página Maestra a través de la propiedad **ContentPlaceHolderID**.

Debe contener marcas y controles sólo dentro de los controles Content. Puede, sin embargo, tener directivas o código del lado del servidor.

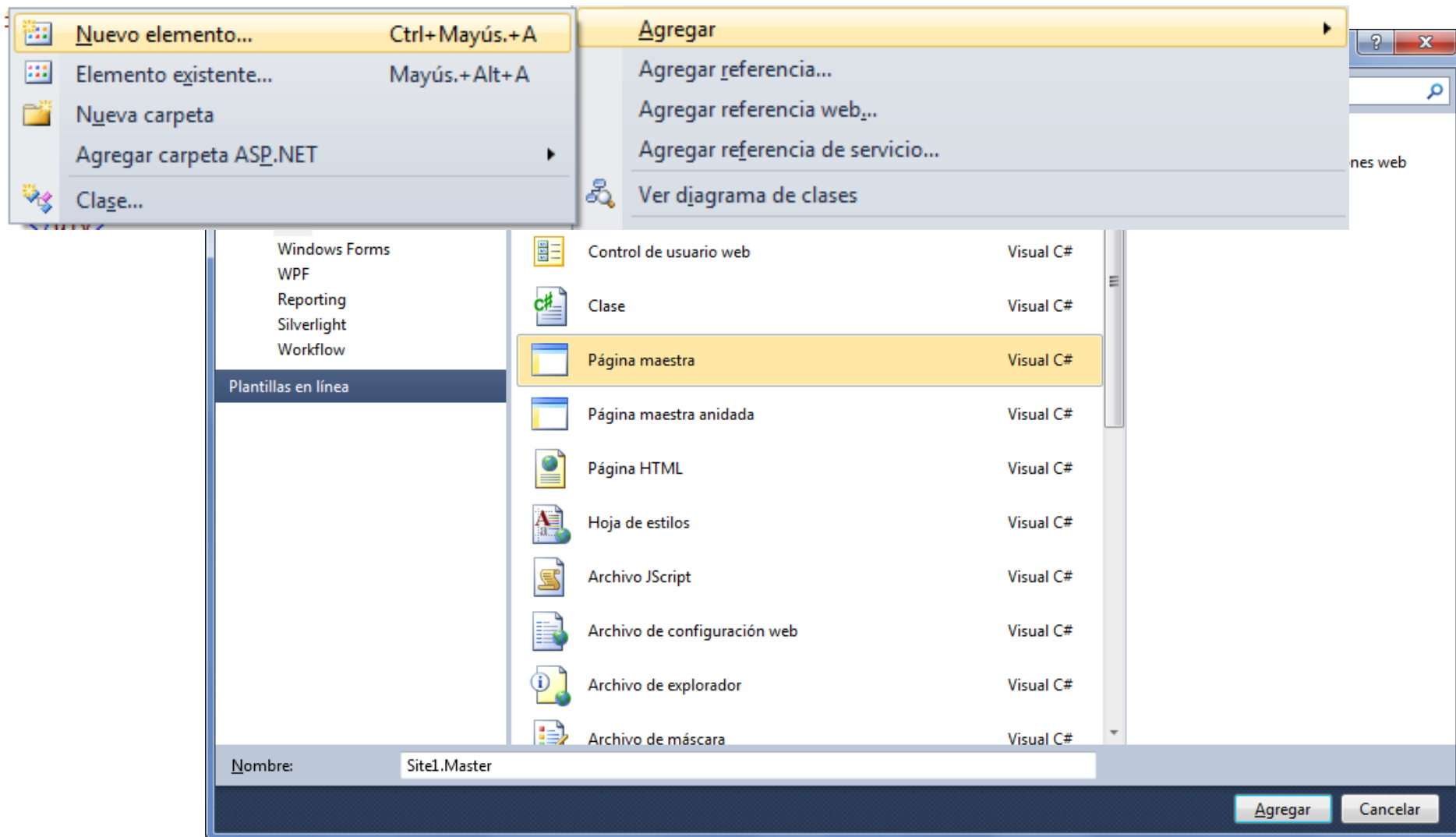
Ejemplo

Creación de una Master Page



Ejemplo

Creación de una Master Page



Ejemplo

Master Page (*.master)

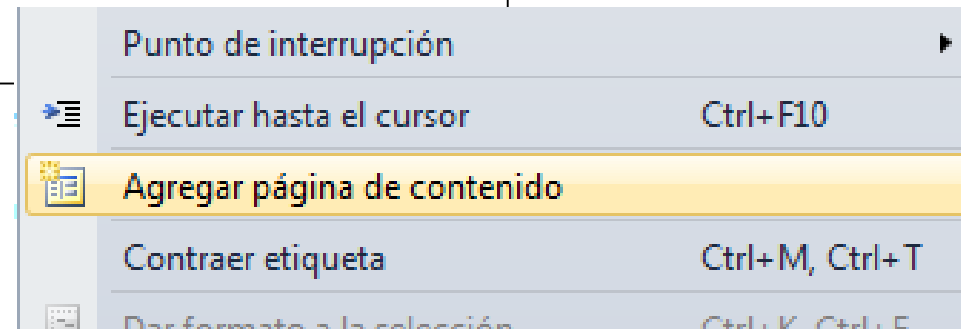
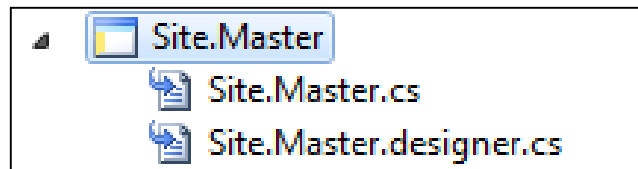
```
<%@ Master Language="C#" AutoEventWireup="true" CodeBehind="Site.master.cs" Inherits="WebApplication2.SiteMaster" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">
<head runat="server">
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
  <title></title>
  <link href="~/Styles/Site.css" rel="stylesheet" type="text/css" />
  <asp:ContentPlaceHolder ID="HeadContent" runat="server">
  </asp:ContentPlaceHolder>
</head>
<body>
  <form runat="server">
    <div class="page">
      <div class="header">
        <div class="title">
          <h1>
            Mi aplicación ASP.NET
          </h1>
        </div>
      </div>
      <div class="main">
        <asp:ContentPlaceHolder ID="MainContent" runat="server"/>
      </div>
      <div class="clear">
      </div>
    </div>
    <div class="footer">
    </div>
  </form>
</body>
</html>
```


Ejemplo

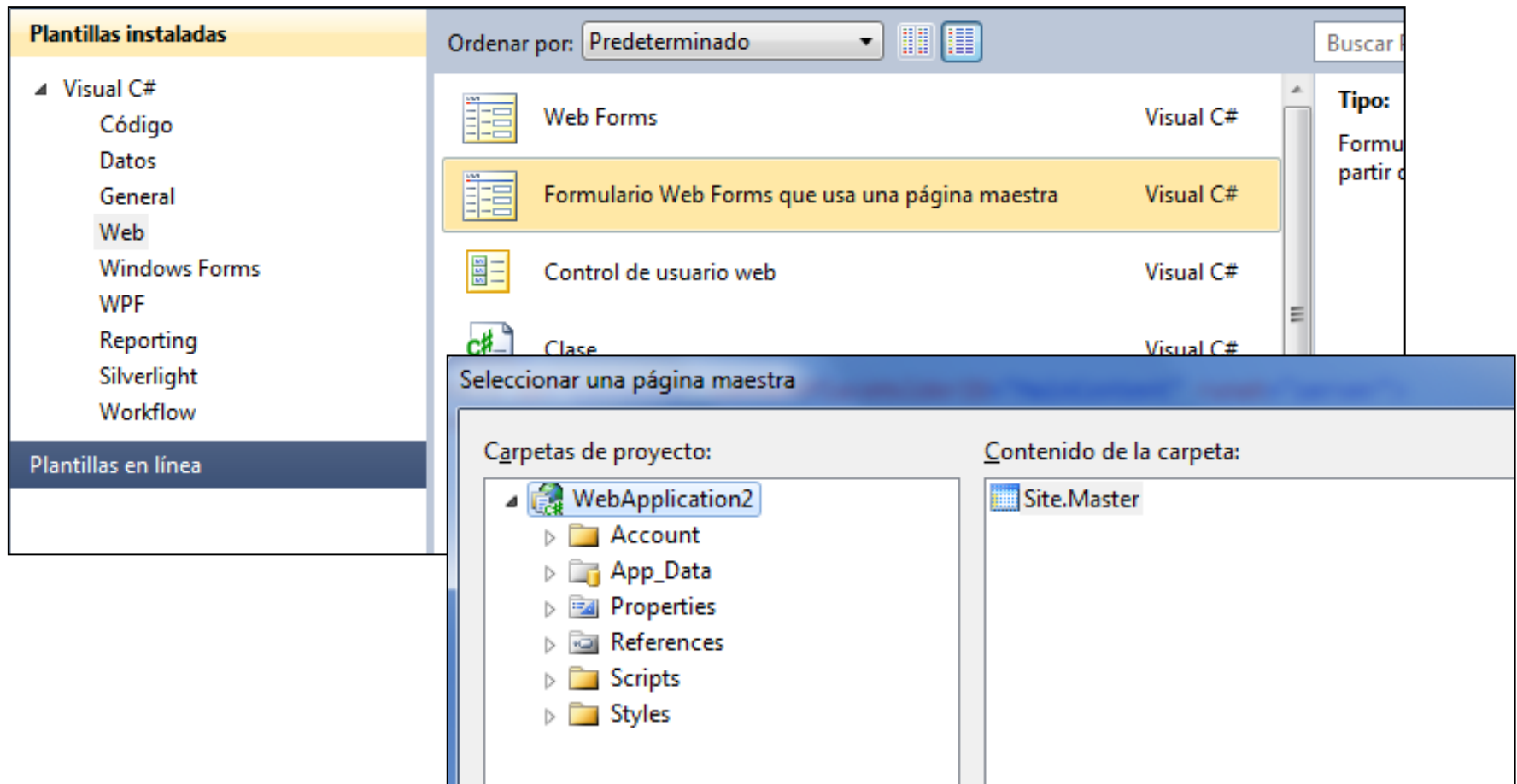
Content Page (*.aspx)

```
        </h1>
    </div>
</div>
<div class="main">
    <asp:ContentPlaceholder ID="MainContent" runat="server"/>
</div>
<div class="clear">
</div>
</div>
```



Ejemplo

Content Page (*.aspx)



Ejemplo

Content Page (*.aspx)

```
<%@ Page Title="Página principal" Language="C#" MasterPageFile="~/Site.master" AutoEventWireup="true"
    CodeBehind="Default.aspx.cs" Inherits="WebApplication2._Default" %>
```

```
<asp:Content ID="HeaderContent" runat="server" ContentPlaceHolderID="HeadContent">
</asp:Content>
```

```
<asp:Content ID="BodyContent" runat="server" ContentPlaceHolderID="MainContent">
```

```
<h2>
```

```
    ASP.NET
```

```
</h2>
```

```
<p>
```

```
    Para obtener más información acerca de ASP.NET, visite
```

```
    <a href="http://www.asp.net" title="Sitio web de ASP.NET">
```

```
        www.asp.net
```

```
    </a>.
```

```
</p>
```

```
<p>
```

```
    También puede encontrar
```

```
    <a href="http://go.microsoft.com/fwlink/?LinkID=152368"
```

```
        title="Documentación de ASP.NET en MSDN">
```

```
        documentación sobre ASP.NET en MSDN
```

```
    </a>.
```

```
</p>
```

```
</asp:Content>
```

Ejercicio 1

1 – Crear MasterPage

- a) Crear 4 secciones (Head, Menú, Contenido y Pie de Pagina).
Hacerlo con DIVs
- b) A Cada Sección ponerle un “TODO nombre_seccion”
- c) En la sección de Contenido, crear un ContentPlaceHolder

2- Crear ContentPage

- a) Crear un Formulario de Login

Accediendo a la Master Page

Es posible que la Content Page acceda a su Master Page.

Una Content Page puede crear una referencia fuertemente tipada a la Master Page mediante la directiva `<%@ MasterType %>`. Se debe especificar la ruta virtual de la página maestra, en la cabecera del archivo Content Page:

```
<%@ MasterType VirtualPath="NOMBRE_MASTERPAGE.master" %>
```

Para acceder a controles o propiedades públicas de la MasterPage, debe utilizarse la clase estática **MASTER**.

Para buscar controles, el método **FindControl** no es recomendable porque la búsqueda se realiza en tiempo de ejecución. Si el nombre del control cambia en tiempo de diseño, el código compila, y el error aparece al ejecutar el código (en tiempo de ejecución)

Accediendo a la Master Page

Site.master

```
private string pieDePagina;
```

```
public string PieDePagina
```

```
{
```

```
    get
```

```
    {
```

```
        return pieDePagina;
```

```
    }
```

```
    set
```

```
    {
```

```
        pieDePagina = value;
```

```
        this.lblPieDePagina.InnerText = value;
```

```
    }
```

```
}
```

Accediendo a la Master Page

ContentPage

Ejemplo Propiedad Pública

Master.PieDePagina = "Es un mensaje";

Ejemplo FindControl

Label lblMensaje = (Label)Master.[FindControl](#)("lblMensajeOrigen");

Ejercicio 2

1 – En la MasterPage (anteriormente creada)

- a) Agregar en la sección del menú un label “Usuario”, el cual contenga un usuario ficticio (eliminar el “TODO”)
- b) Crear una Propiedad Pública “Usuario”, que devuelva el contenido del label

2- Crear ContentPage

- a) Agregar un evento que al presionar el botón acceda al contenido del Label del Menú de la MasterPage
 - i) Mediante Propiedad Pública
 - ii) Mediante FindControl

Eventos de la MasterPage y ContentPage

Eventos de la Master Page y Content Page

No se puede administrar un evento de la MasterPage en la ContentPage

Un Evento puede dispararse en ambas Páginas (MasterPage y ContentPage). Por ejemplo, los eventos del Ciclo de Vida: INIT, LOAD, etc.

Para el Ciclo de Vida de ambas páginas, y dependiendo el evento en cuestión, existe un orden estricto de ejecución.

Eventos de la MasterPage y ContentPage

Eventos del Ciclo de Vida de la Master Page y Content Page

- 1) ContentPage: evento "PreInit".
- 2) Controles de la MasterPage: evento "Init"
- 3) Controles de la ContentPage: evento "Init"
- 4) MasterPage: evento "Init"
- 5) ContentPage: evento "Init"
- 6) ContentPage: evento "Load"
- 7) MasterPage: evento "Load"
- 8) Controles de la MasterPage: evento "Load"
- 9) Controles de la ContentPage: evento "Load"
- 10) ContentPage: evento "PreRender"
- 11) MasterPage: evento "PreRender"
- 12) Controles de la MasterPage: evento "PreRender"
- 13) Controles de la ContentPage: evento "PreRender"
- 14) Controles de la MasterPage: evento "Unload"
- 15) Controles de la ContentPage: evento "Unload"
- 16) MasterPage: evento "Unload"
- 17) ContentPage: evento "Unload"

Eventos de la MasterPage y ContentPage

Eventos de la Master Page y Content Page

¿Es posible cambiar dinámicamente (en tiempo de ejecución) una MasterPage?

Si, en el evento PRE_INIT

```
void Page_PreInit(Object sender, EventArgs e)  
{  
    this.MasterPageFile = "Master2.master";  
}
```

Ejercicio 3

1- Crear un nueva nueva MasterPage “Master2.master”

2 - Agregar una nueva página ContentPage

a) En el eventos Pre_init agregar:

```
void Page_PreInit(Object sender, EventArgs e) {  
    this.MasterPageFile = “Master1.master”; //Cambiarla en Diseño  
}
```

Anidando Master Pages

Es posible anidar Master Pages de manera similar a Content Pages

- Tienen extensión .master
- *En el encabezado debe declararse como “Master” (en lugar de “Page”) y hacer referencia a la Master Page*
- Tienen que tener un control asp.net “Content” que implementa el ContentPlaceholder de la Master Page padre.

MasterPageAnidada.Master

```
<% Master Language="C#" MasterPageFile="Site.Master"  
    CodeBehind="MasterPageAnidada.Master.cs"  
    Inherits="MasterPageAnidada" %>
```

Ejemplos

Master Page anidada (*.master)

```
<%@ Master Language="C#" MasterPageFile="~/Site.Master" AutoEventWireup="true"  
CodeBehind="NestedMasterPage1.master.cs" Inherits="WebApplication2.NestedMasterPage1" %>
```

```
<asp:Content ID="Content1" ContentPlaceHolderID="HeadContent" runat="server">  
</asp:Content>
```

```
<asp:Content ID="Content2" ContentPlaceHolderID="MainContent" runat="server">  
</asp:Content>
```

Ejemplos

Ejemplos:

SeccionVenta.Master

```
<%@ master language="C#" masterpagefile="Site.master"%>
<%@ MasterType VirtualPath="Site.master" %>

<asp:content contentplaceholderid="cphContenido" runat="server">
    <h3>Ventas </h3>

    <asp:contentplaceholder id="cphSeccionListado" runat="server"/>

    <asp:contentplaceholder id="cphSeccionDetalle" runat="server"/>
</asp:content>
```

SeccionCompra.Master

```
<%@ master language="C#" masterpagefile="Site.master"%>
<%@ MasterType VirtualPath="Site.master" %>

<asp:content contentplaceholderid="cphContenido" runat="server">
    <h3>Compras</h3>

    <asp:contentplaceholder id="cphSeccionListado" runat="server"/>

    <asp:contentplaceholder id="cphSeccionDetalle" runat="server"/>

    <asp:contentplaceholder id="cphSeccionHistorico" runat="server"/>
</asp:content>
```

Master Page obligatoria

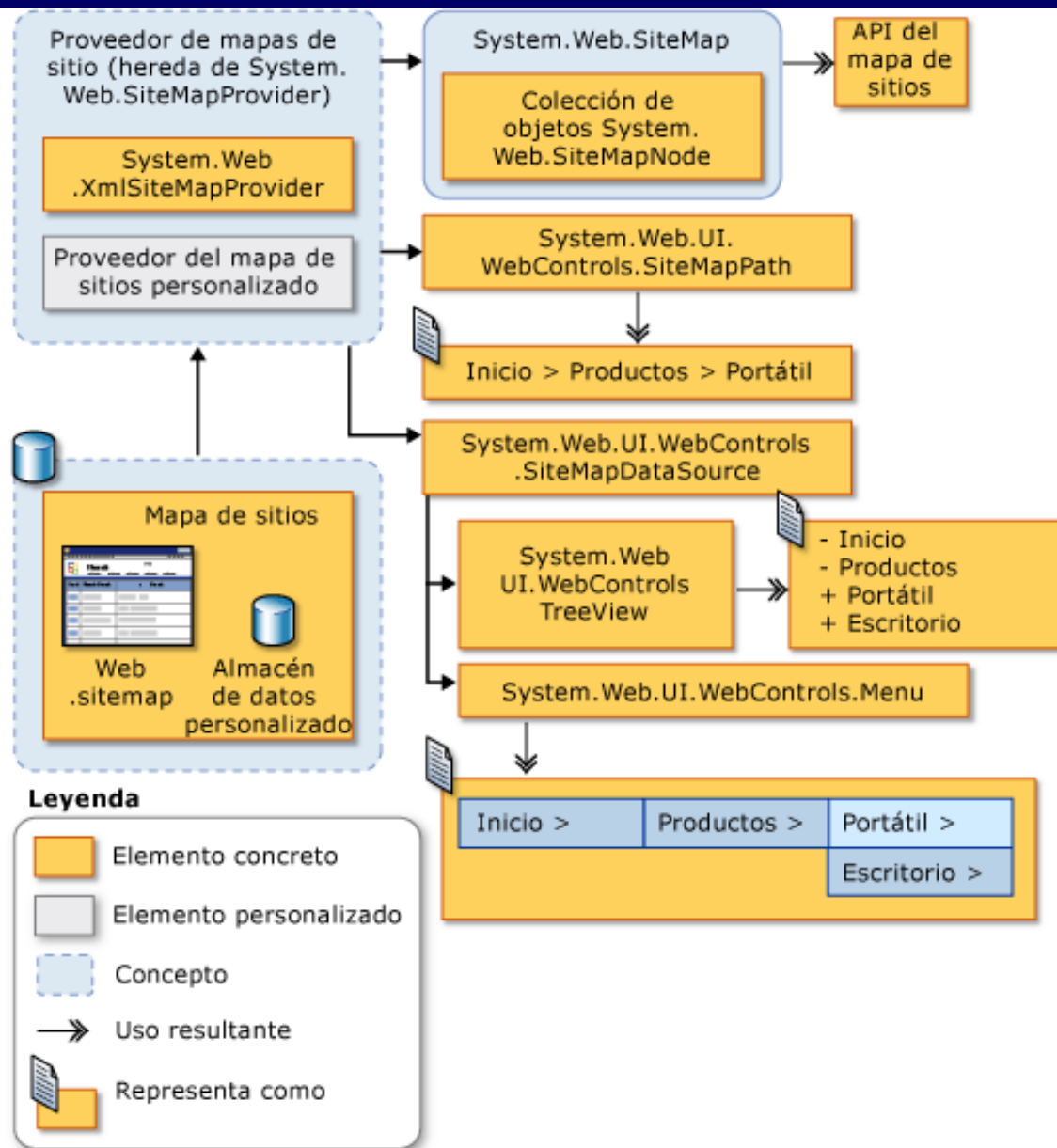
Para asegurar que todas las páginas tendrán una Master Page hay que declararlo en el archivo web.config, .

Si una página declara una Master Page con la propiedad MasterPageFile reemplaza el contenido del web.config.

```
<configuration>
  <system.web>
    <pages masterPageFile="~/MasterPage.master" />
  </system.web>
</configuration>
```


Controles de Navegación

Controles de Navegación



Controles de Navegación

Controls

Menu

TreeView

SiteMap-
DataSource

SiteMapPath

Site Navigation API

SiteMap

SiteMapNode

SiteMapNode

SiteMapNode

Providers

XmlSiteMapProvider

Other Site Map
Providers

Site Maps

Web.sitemap

Other
Data Stores

Controles de Navegación

Web.sitemap

Organiza las páginas del sitio jerárquicamente. El proveedor del mapa del sitio carga de forma predeterminada este mapa del sitio en ASP.NET.

Debe estar ubicado en el directorio raíz de la aplicación, aunque puede hacer referencia a otros proveedores o archivos de mapas de sitio situados en otros directorios siempre que esos archivos estén en la misma aplicación

Por seguridad, usar la extensión “.sitemap”. Caso contrario, alojar el archivo en la carpeta “App_Data”

El archivo incluye información de las páginas del sitio Web, como la dirección URL, el título, la descripción y la ubicación en la jerarquía de exploración

Controles de Navegación

<siteMap>

<siteMapNode title="Inicio" description="Inicio" url="~/inicio.aspx">

<siteMapNode title="Productos" description="Nuestros productos"
url="~/Productos.aspx">

<siteMapNode title="Hardware" description="Hardware"
url="~/Hardware.aspx" />

<siteMapNode title="Software" description="Software"
url="~/Software.aspx" />

</siteMapNode>

<siteMapNode title="Servicios" description="Nuestros Servicios"
url="~/Servicios.aspx">

<siteMapNode title="Capacitacion" description="Capacitación"
url="~/Capacitacion.aspx" />

<siteMapNode title="Consultoria" description="Consultoria"
url="~/Consulting.aspx" />

<siteMapNode title="Soporte" description="Soporte"
url="~/Soporte.aspx" />

</siteMapNode>

</siteMapNode>

</siteMap>

Ejercicio 1

1- Crear un web.sitemap

Primer Nivel: “Categorias”

Segundo Nivel: “Auto”, “Camionetas” y “Camiones”

Tercer Nivel: “Nuevo”, “Usados”

2 – Crear las ContentPage requeridas

Controles de Navegación

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 o a páginas superiores

Inicio > Servicios > Capacitación

El control proporciona muchas opciones para personalizar el aspecto de los vínculos

Obtiene los datos de desplazamiento de un mapa del sitio (web.sitemap)

Puede modificarse mediante programación

SiteMapDataSource

Proporciona un control de origen de datos que los controles de servidor Web y otros controles pueden utilizar para enlazarse a los datos del mapa jerárquico de un sitio.

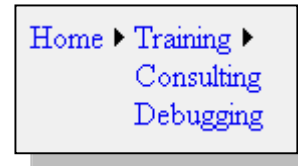
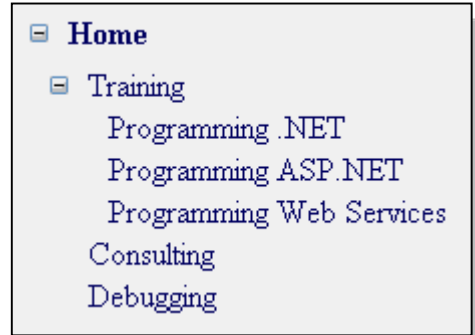
Ejercicio 2

En la MasterPage:

- 1 – Crear un SiteMapDataSource
- 2 – Crear un SiteMapNode en la sección Menú

Controles de Navegación

```
<asp:TreeView DataSourceID="SiteMap" RunAt="server" />  
<asp:Menu DataSourceID="SiteMap" RunAt="server" />  
<asp:SiteMapDataSource ID="SiteMap" RunAt="server" />
```



Web.sitemap

```
<siteMap>  
  <siteMapNode title="Home" description="" url="default.aspx">  
    <siteMapNode title="Training" url="Training.aspx">  
      description="Training for .NET developers">  
        <siteMapNode title="Programming .NET" url="Classes.aspx?id=1">  
          description="All about the .NET Framework" />  
        <siteMapNode title="Programming ASP.NET" url="Classes.aspx?id=2">  
          description="All about ASP.NET" />  
        <siteMapNode title="Programming web Services" url="Classes.aspx?id=3">  
          description="All about web services" />  
      </siteMapNode>  
    <siteMapNode title="Consulting" url="Consulting.aspx">  
      description="Consulting for .NET projects" />  
    <siteMapNode title="Debugging" url="Debugging.aspx">  
      description="Help when you need it the most" />  
  </siteMapNode>  
</siteMap>
```

Controles de Navegación

Menú

Permite crear un menú desplegable mediante dos modos de presentación: estática y dinámica.

En la **presentación estática** el menú permanece completamente desplegado todo el tiempo. Se ve la estructura completa y el usuario puede hacer clic en cualquier parte.

En la **presentación dinámica** el menú sólo tiene estáticas las partes que especifique, mientras sus elementos de menú secundarios se presentan cuando el usuario sitúa el puntero del mouse sobre el nodo primario.

Por default es un control vertical. Un buen uso del mismo hace que todos los ítems sirvan de acceso a las diferentes partes del sitio web.

Los menús se comportan en relación a una estructura de árboles normalmente organizados en diferentes niveles de una jerarquía. La primera rama es el root del menú y las siguientes son los sub menús.

Se puede personalizar el aspecto mediante estilos, temas e imágenes.

Controles de Navegación

TreeView

Está diseñado para presentar datos en una estructura jerárquica.

Los usuarios pueden abrir nodos individuales que, a su vez, pueden contener nodos secundarios.

Características:

- Enlace de datos automático
- Compatibilidad con la navegación del sitio a través de la integración con el control SiteMapDataSource.
- El texto del nodo se puede mostrar como texto seleccionable o como hipervínculos.
- Personalización del aspecto mediante estilos, temas e imágenes
- Acceso mediante programación al modelo de objetos TreeView, que permite crear árboles dinámicamente, rellenar nodos, definir propiedades, etc.
- La capacidad para mostrar una casilla al lado de cada nodo

Ejercicio 3

En la MasterPage:

- 1 – Crear un Menú en la sección Menú
- 2 – Crear un TreeView en la sección Pie de Página

Desarrollador de Aplicaciones Web

Programación Web III



Departamento de Ingeniería e Investigaciones Tecnológicas

Muchas gracias

Ing. Mariano Juiz
Ing. Matias Paz Wasiuchnik
Ing. Pablo Nicolás Sanchez