

# **Licenciatura en Gestión Tecnológica**

## **Programación Avanzada 2**

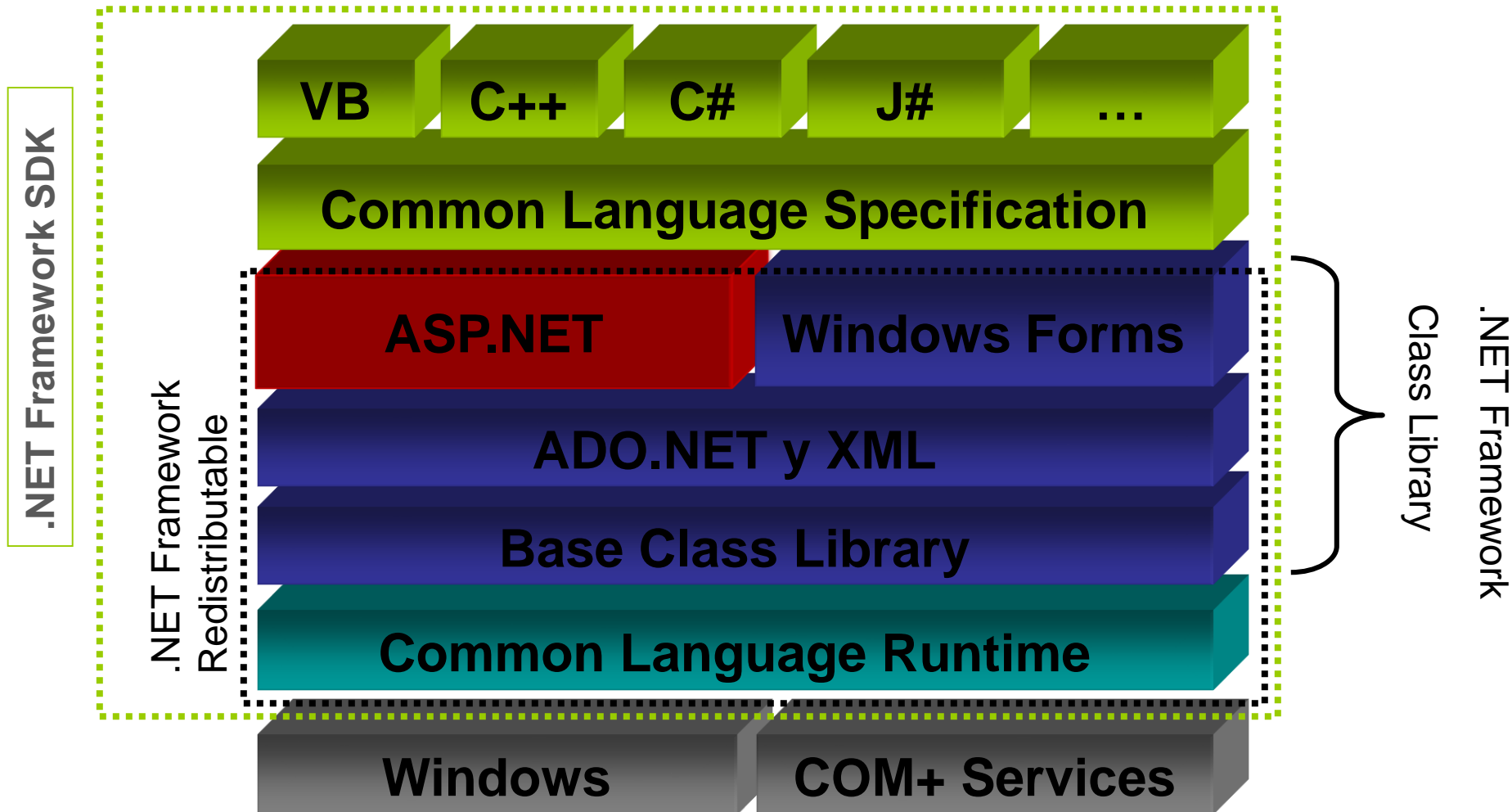


## **Introducción a ASP.NET**

**Ing. Mariano Juiz**

# Introducción

## Arquitectura del .NET Framework



# Introducción

## Páginas web

Una aplicación o sitio web es un conjunto de páginas que se transmiten por medio del protocolo HTTP de un servidor al cliente y viceversa

HTTP define como los navegadores y los servidores Web se comunican uno con otro. Esta basado en texto y es transmitido sobre conexiones TCP

El cliente solicita la página mediante un Request y el servidor web responde mediante un Response

### Request

```
GET /inicio.html HTTP/1.1
Accept: */*
Accept-Language:...
Accept-Encoding:...
If-Modified-Since:...
If-None-Match:...
User-Agent: Mozilla/4.0...
Host: www.unlam.edu.ar
Connection: Keep-Alive
[blank line]
```

### Response

```
HTTP/1.1 200 OK
Server: Microsoft-IIS/7.0
Date: ...
Content-Type: text/html
Accept-Ranges: bytes
Last-Modified: ...
ETag: ...
Content-Length: 46
[blank line]
<html>
<body>
.....
</body>
</html>
```

# Introducción

## Páginas web

### Cliente

http://www.unlam.edu.ar

Internet DNS



IP= 200.47.130.101 Puerto: 80



HTTP Request

### Servidor



www.unlam.edu.ar

IP = 200.47.130.101

inicio.html

```
<html>
```

```
<body>
```

.....

```
</body>
```

```
</html>
```

HTTP Response

# Introducción

## **ASP.NET**

ASP.NET es un “Marco” (framework) para programar aplicaciones web

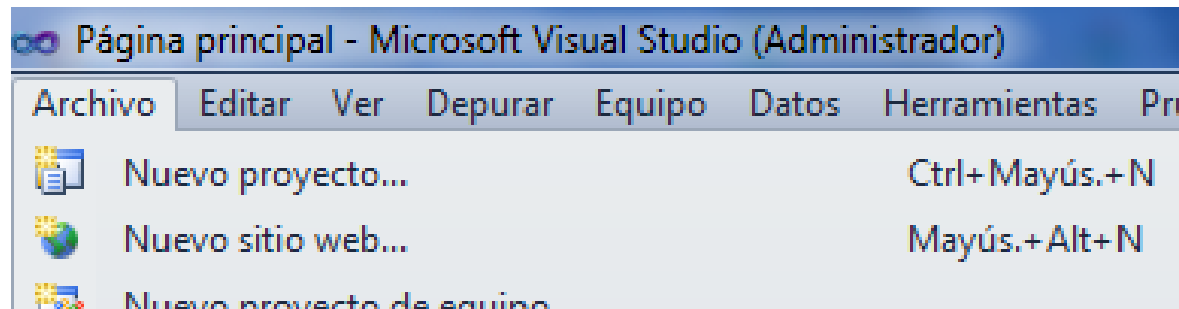
Internet Information Server (IIS), es el servidor Web de Microsoft que corre sobre plataformas Windows. Los servicios que ofrece son: FTP, SMTP, NNTP y HTTP/HTTPS

El componente fundamental de ASP.NET es el WebForm

Permite utilizar cualquier lenguaje .NET

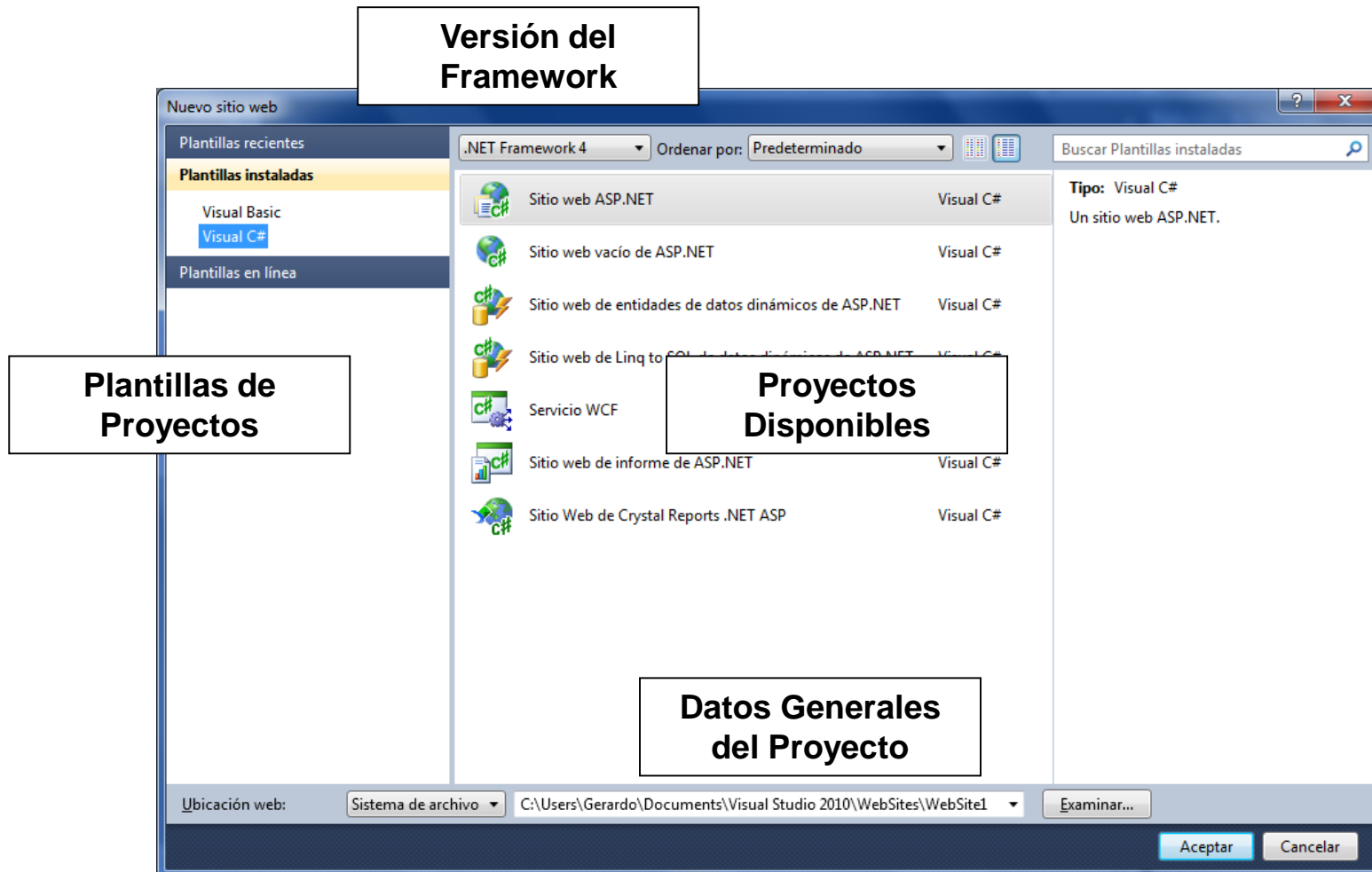
Permite crear sitios web, aplicaciones web, servicios web, controles web, entre otras cosas.

# ¿Sitio Web o Aplicación Web?



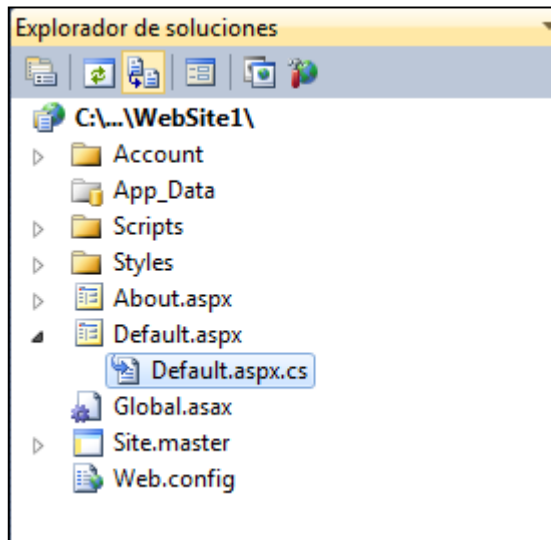
# Aplicación ASP.NET

## Sitio Web



# Aplicación ASP.NET

## Sitio Web

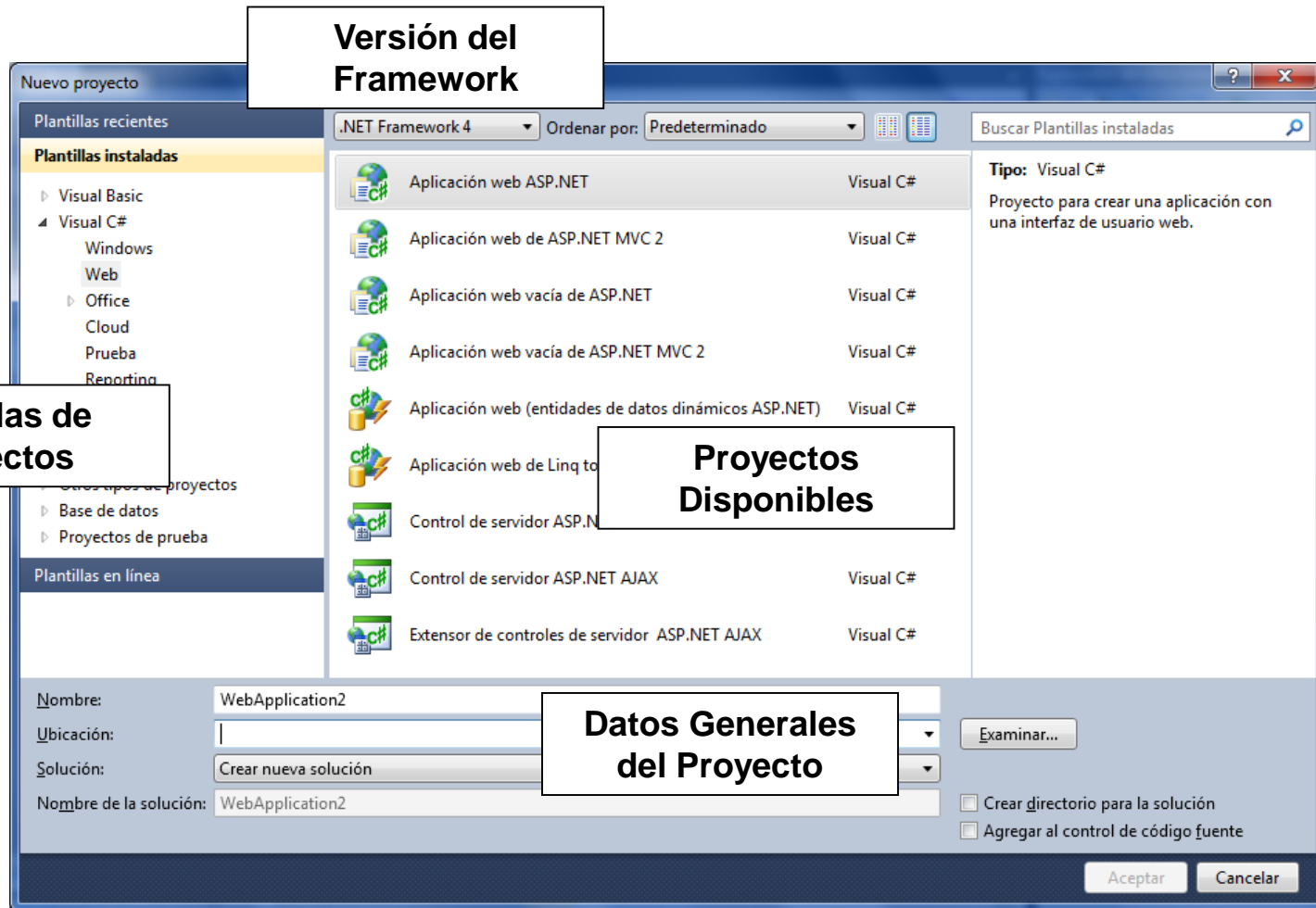


Nombre	Fecha de modifica...	Tipo
Account	28/08/2012 12:51 ...	Carpeta de archivos
App_Data	28/08/2012 12:51 ...	Carpeta de archivos
Scripts	28/08/2012 12:51 ...	Carpeta de archivos
Styles	28/08/2012 12:51 ...	Carpeta de archivos
About.aspx	28/08/2012 12:51 ...	ASP.NET Server Pa...
About.aspx.cs	28/08/2012 12:51 ...	Visual C# Source f...
Default.aspx	28/08/2012 12:51 ...	ASP.NET Server Pa...
Default.aspx.cs	28/08/2012 12:51 ...	Visual C# Source f...
Global.asax	28/08/2012 12:51 ...	ASP.NET Server A...
Site.master	28/08/2012 12:51 ...	ASP.NET Master P...
Site.master.cs	28/08/2012 12:51 ...	Visual C# Source f...
Web.config	28/08/2012 12:51 ...	XML Configuratio...



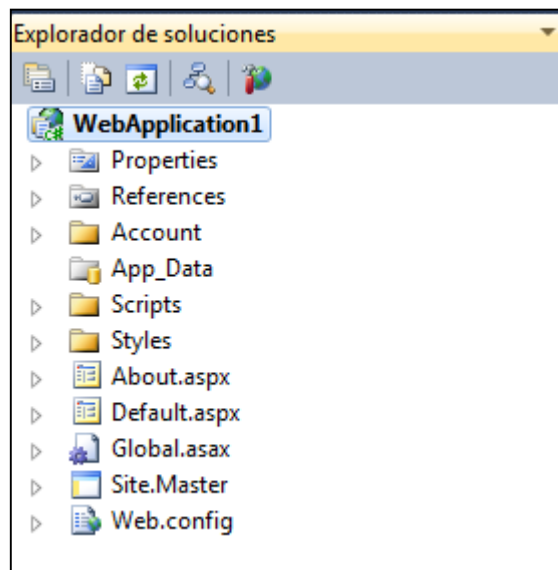
# Aplicación ASP.NET

## Aplicación Web



# Aplicación ASP.NET

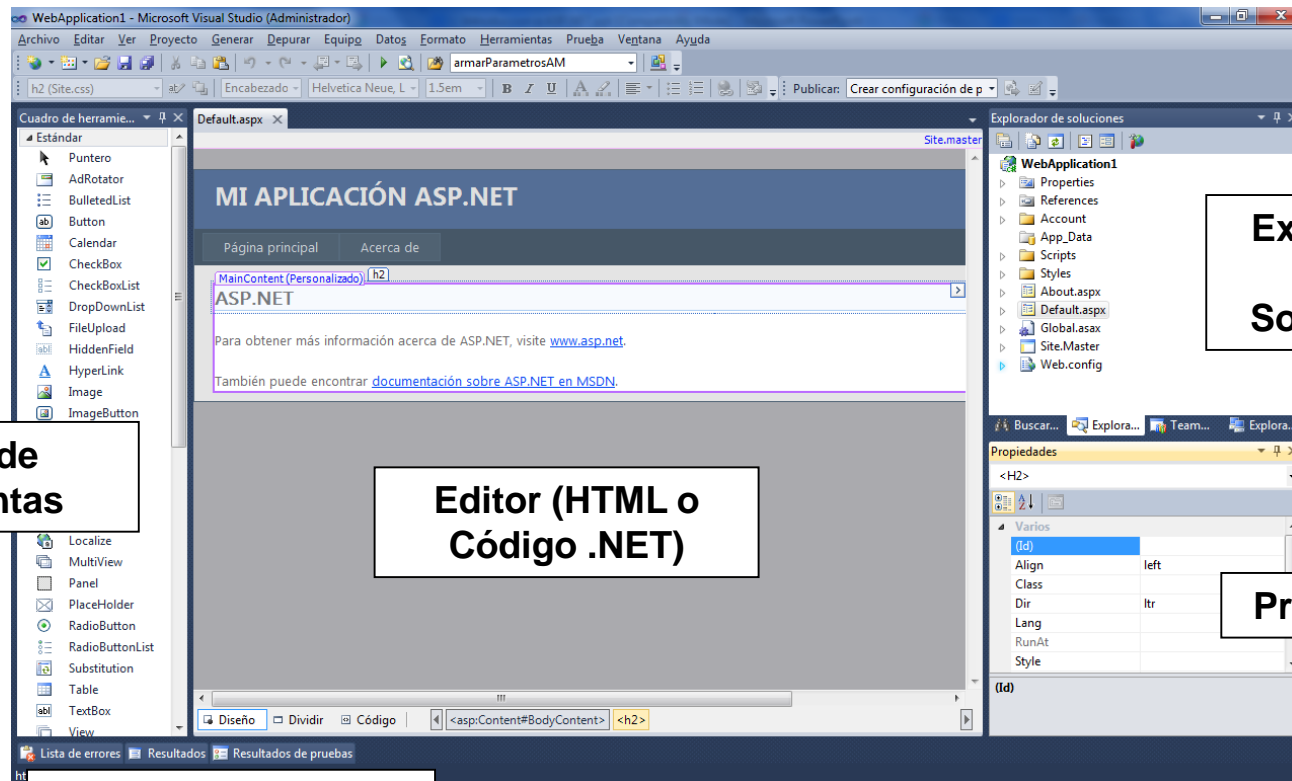
## Aplicación Web



Nombre	Fecha de modifica...	Tipo
Account	28/08/2012 12:22 ...	Carpeta de archivos
App_Data	28/08/2012 12:22 ...	Carpeta de archivos
bin	28/08/2012 12:22 ...	Carpeta de archivos
obj	28/08/2012 12:22 ...	Carpeta de archivos
Properties	28/08/2012 12:22 ...	Carpeta de archivos
Scripts	28/08/2012 12:22 ...	Carpeta de archivos
Styles	28/08/2012 12:22 ...	Carpeta de archivos
About.aspx	28/08/2012 12:22 ...	ASP.NET Server Pa...
About.aspx.cs	28/08/2012 12:22 ...	Visual C# Source f...
About.aspx.designer.cs	28/08/2012 12:22 ...	Visual C# Source f...
Default.aspx	28/08/2012 12:22 ...	ASP.NET Server Pa...
Default.aspx.cs	28/08/2012 12:22 ...	Visual C# Source f...
Default.aspx.designer.cs	28/08/2012 12:22 ...	Visual C# Source f...
Global.asax	28/08/2012 12:22 ...	ASP.NET Server A...
Global.asax.cs	28/08/2012 12:22 ...	Visual C# Source f...
Site.Master	28/08/2012 12:22 ...	ASP.NET Master P...
Site.Master.cs	28/08/2012 12:22 ...	Visual C# Source f...

# Aplicación ASP.NET

## Visual Studio (IDE)



**Cuadro de  
herramientas**

**Editor (HTML o  
Código .NET)**

**Explorador  
de  
Soluciones**

**Propiedades**

**Lista de Errores  
Lista de Resultados  
Lista de Tareas**

# Aplicación ASP.NET

## Componentes

### WebForms (Formularios Web)

Uno o más archivos con extensión **.aspx**

### Archivos Code-Behind / Code-File

Archivos asociados a WebForms que contienen código del lado del servidor Ej.: C# (\*.cs), VB.NET (\*.vb), etc.

Al usar Code-Behind, se utilizan clases parciales para vincularse con su aspx.

Las páginas heredan de System.Web.UI.Page

### Archivos de configuración con formato XML

Un archivo Web.config por c/aplicación

Un único archivo Machine.config por servidor

### Directorio BIN

Contiene el assembly de la aplicación (Ej.: MiAplic.dll)

Cero o más assemblies (Componentes externos)

# Aplicación ASP.NET

## Componentes

### Global.asax

Eventos a nivel de aplicación

### Enlaces a Servicios Web XML

Permiten a la aplicación ASP.NET enviar y recibir datos desde Servicios Web

### Archivo de Proyecto de Aplicación Web

Guardan información en formato XML de los archivos utilizados por el proyecto.  
Utilizan extensión \*.csproj o \*.vbproj

### Archivo de Solución de Aplicación ASP.NET

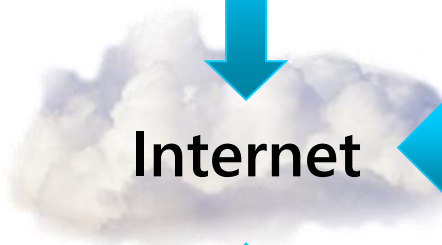
Guardan información en formato XML de los proyectos asp.net. Utilizan extensión \*.sln

# Aplicación ASP.NET

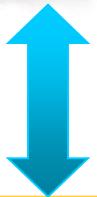
Componentes  
Cliente



Cliente



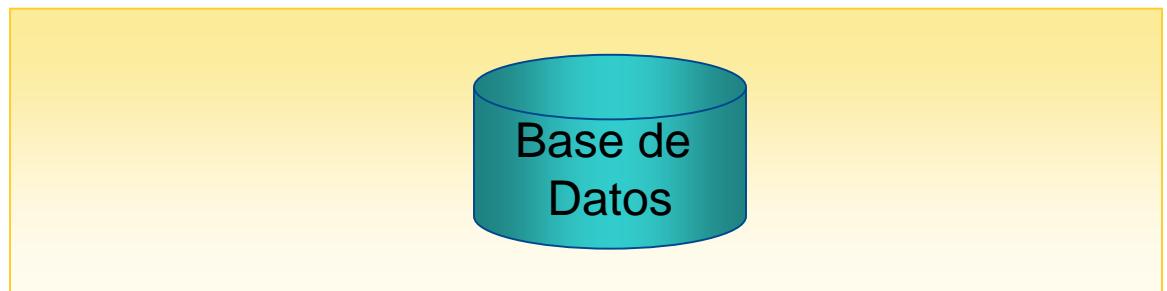
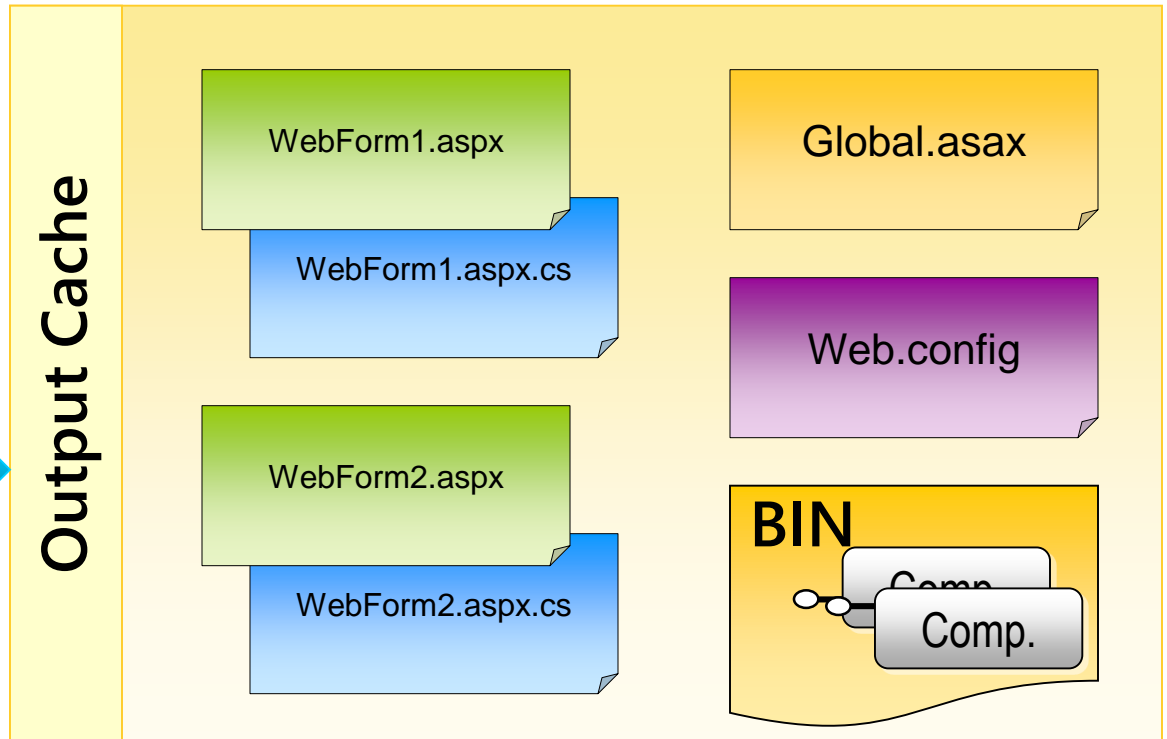
Internet



Servicios Web



ASP.NET Web Server



# Página web ASP.NET

## Introducción

- Extensión .aspx
- Atributos de página
  - Directiva @ Page

```
<%@ Page Language="c#" Codebehind="WebForm1.aspx.cs" %>
```

- Atributos de cuerpo
- Atributos de formulario

# Página web ASP.NET

## Código “En línea”

Código y contenido en el mismo archivo

Distintas secciones en el archivo para el código y HTML

```
<HTML>
<HEAD>
<SCRIPT Language="c#" runat="server">
    private void btnAceptar_Click(object sender, System.EventArgs e)
    {
        . . .
    }
</SCRIPT>
</HEAD>
<BODY>
<form id="form1" runat="server">
    <asp:Button id="btnAceptar" runat="server"/>
</form>

</BODY>
</HTML>
```

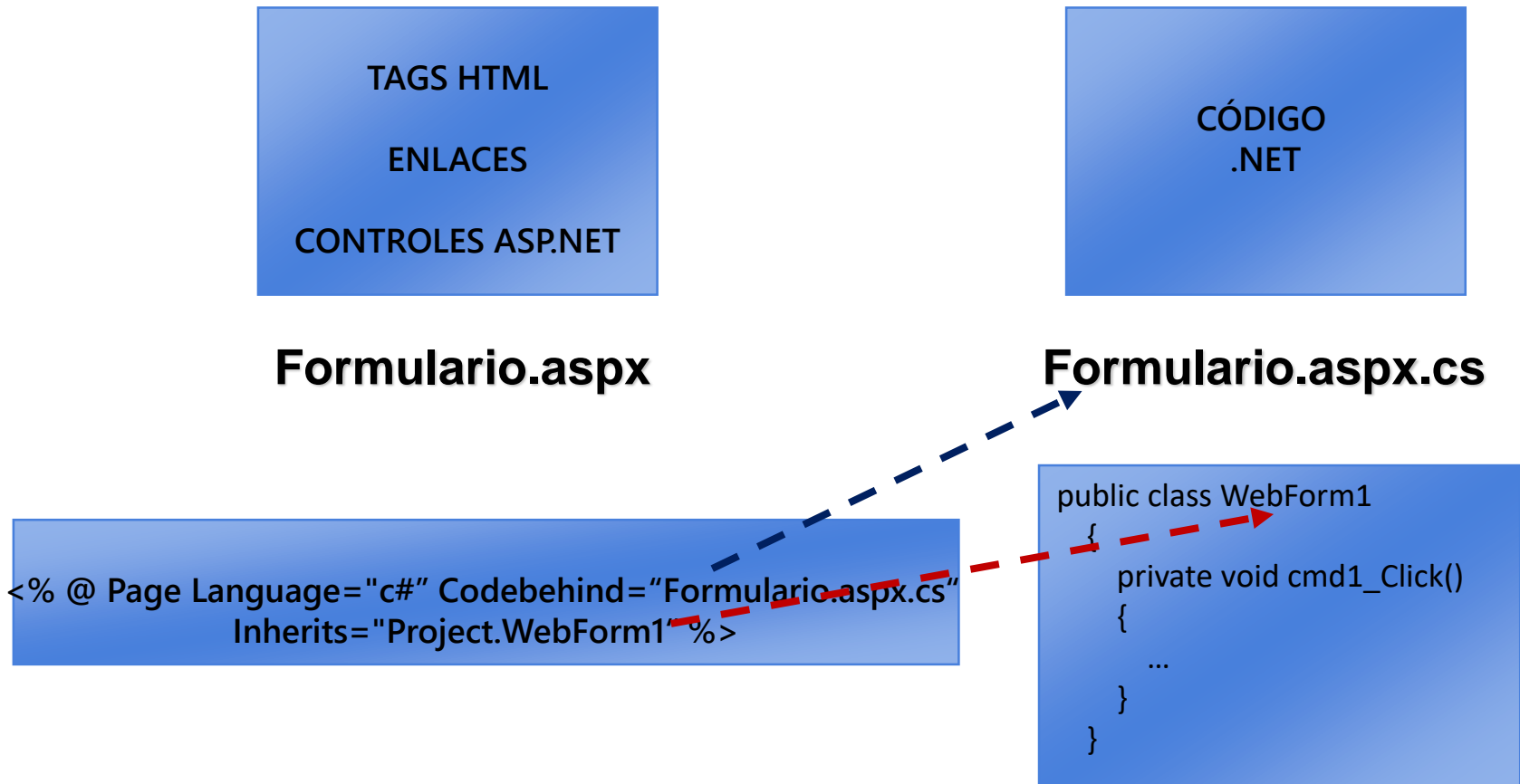
**Formulario.aspx**



# Página web ASP.NET

## Código “Subyacente” (Code-Behind)

Separación de código y contenido



# Página web ASP.NET

## Código “Mixto”

- Código embebido dentro del .aspx a la forma asp tradicional.
- Todo lo que esté entre “<% %>” se ejecutará del lado del servidor.

```
<%@ Page Language="C#" %>
<HTML>
<HEAD>
</HEAD>
<BODY>
<form id="form1" runat="server">
<div>
    <%
        if (DateTime.Now.Year == 2013)
        {
            Response.Write("Es nuestro año");
        }
        else
        {
            Response.Write("Estas en el pasado o en el futuro");
        }
    %>
</div>
</form>
</BODY>
</HTML>
```

**Formulario.aspx**

# Controles de Servidor

## Introducción

Componentes que se ejecutan en el lado del servidor

Poseen el atributo **runat="server"**

Poseen un modelo de objetos común. Ej.: todos tienen las propiedades Id y Text

Mantienen su “estado” entre postbacks al servidor (ViewState)

“Generan HTML específico según el browser cliente”

# Controles de Servidor

## Tipos de Controles de Servidor

### Controles de Servidor HTML

Por defecto, los elementos HTML no son accesibles desde código del lado del servidor

Agregando `runat="server"` y el atributo `id`, se convierten en Controles de Servidor HTML

### Controles de Servidor Web

Conocidos como WebControls

Solo accesibles del lado del servidor

Muchos tipos

- Intrínsecos
- Validación
- “Ricos”
- Del tipo lista de datos

No tienen una relación 1:1 con elementos HTML

# Controles de Servidor

## Tipos de Controles de Servidor

### Botón HTML “clásico” (No es de Servidor)

```
<INPUT type="button" value="Buscar">
```

### Control de Servidor HTML

```
<INPUT type="button" value="Buscar" id="cmdBuscar" runat="server">
```

### Control de Servidor Web

```
<asp:Button id="cmdBuscar" runat="server" Text="Buscar"/>
```

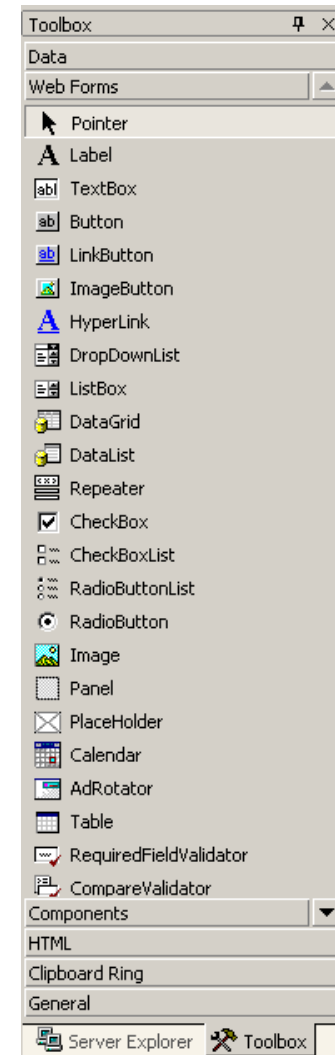
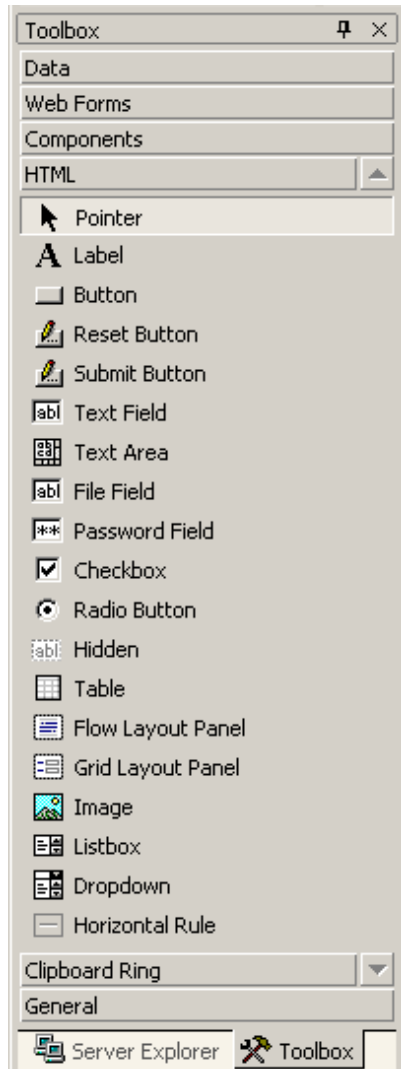
# Controles de Servidor

## Tipos de Controles de Servidor

WebControl	HTML equivalente
<asp:button>	<input type=submit>
<asp:checkbox>	<input type=checkbox>
<asp:hyperlink>	<a href="..."> </a>
<asp:image>	
<asp:imagebutton>	<input type=image>
<asp linkButton>	
<asp:label>	<span> </span>
<asp:panel>	<div> </div>
<asp:radiobutton>	<input type=radiobutton>
<asp:table>	<table> </table>
<asp:textbox>	<input type=text>
<asp:listbox>	<select size="5"> </select>

# Controles de Servidor

## Tipos de Controles de Servidor



# Controles de Servidor

## Controles HTML Servidor vs Controles ASP.NET

<b>Utilizar controles de servidor HTML si:</b>	<b>Utilizar controles de servidor Web si:</b>
<b>Se prefiere un modelo de objetos como HTML</b>	<b>Se prefiere un modelo de programación (C# o VB.NET)</b>
<b>Se está trabajando con páginas HTML existentes y se desea agregar funcionalidades de página ASP.NET Web</b>	<b>“Se está escribiendo una página que puede ser utilizada por varios navegadores”</b>
<b>El control interactuará con scripts cliente y servidor</b>	<b>Se necesitan funcionalidades específicas como un calendario o rotación de publicidad</b>
<b>El ancho de banda es limitado</b>	<b>El ancho de banda no es un problema</b>



# Controles de Servidor

## Controles HTML Servidor vs Controles ASP.NET

Ambos heredan de `System.Web.UI.Control` que se encuentra contenida en `System.Object`

Ambos guardan estados

Los controles HTML :

- Son mucho más sencillos que los controles ASP.NET
- Tienen menos propiedades y eventos (no es consistente con los controles ASP.NET)
- Son más adecuados cuando no requerimos una gran flexibilidad y queremos cargar la página lo mínimo posible
- Existen en el espacio de nombres `System.Web.UI.HtmlControls`

Los controles ASP.NET:

- Son más complejos (calendario, grilla, validación, login, etc.)
- “Se adaptan al navegador”
- Al navegador se renderizar como controles HTML
- Existen en el espacio de nombres `System.Web.UI.WebControls`

# Controles de Servidor

## Controles ASP.NET

### Sintaxis del control

```
<asp:TextBox id="TextBox1" runat="server" Text="Texto"/>
```

### HTML generado por el control

```
<input name="TextBox1" type="text" value="Texto" Id="TextBox1"/>
```

# Controles de Servidor

## Scripts de eventos del lado del cliente

- Normalmente, se utilizan únicamente con controles HTML
- Interpretado por el navegador y se ejecuta en el cliente
- No tiene acceso a los recursos del servidor
- Utiliza `<SCRIPT language="lenguaje">`

## Scripts de eventos del lado del servidor

- Utilizados tanto con controles de servidor Web como HTML
- El código se compila y ejecuta en el servidor
- Tienen acceso a recursos del servidor
- Utilizan `<SCRIPT language="vb" runat="server">` o `<SCRIPT language="cs" runat="server">`  
O bien ejecutando el evento del CodeBehind
- Se genera un POST desde el cliente hacia el servidor y una respuesta. Este ida y vuelta dentro de un mismo formulario web se llama **Postback**.

# Controles de Servidor

## Scripts de eventos del lado del servidor

Declaración de eventos en un control:

```
<asp:Button ID="btnEjemplo" runat="server" Text="Aceptar" onclick="btnEjemploClick" />
```

Atención del evento en el servidor ( code behind )

```
protected void btnEjemploClick(object sender, EventArgs e)
{
    if (!string.IsNullOrEmpty(txtEjemplo.Text))
    {
        lblEjemplo.Text = txtEjemplo.Text;
    }
}
```

# Controles de Servidor

## Postback

Mecanismo de recarga de una página a partir de una ejecución del lado del cliente

La propiedad `IsPostBack` indica si la página se está mostrando por primera vez o se está cargando como respuesta a un Postback:

```
If( !Page.IsPostBack)
{
    //Inicializar la pagina.
}
```

Para que algunos controles envíen la petición al servidor, deben setear la propiedad `AutoPostBack` en `true`

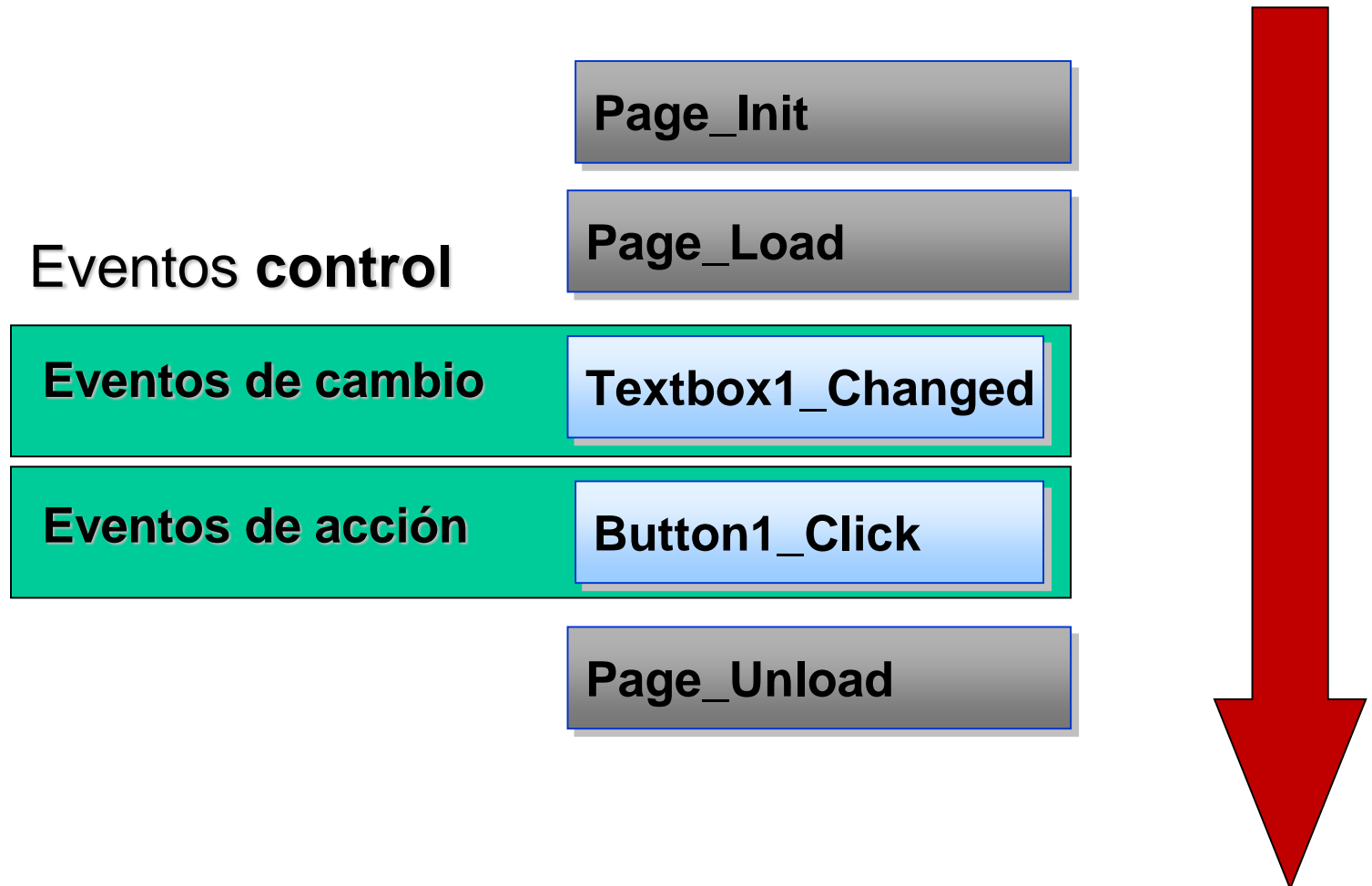
# Ciclo de Vida de una Página ASP.NET

## Sucesos mas importantes del ciclo de vida de una página

- Solicitud de Página
- Inicio
  - Se instancian los objetos de los contextos **request** y **response**
  - Se crean el árbol de controles declarado en la página .aspx
  - Se determina si la página fue llamada en modo postback o no
- Inicialización de los objetos
  - Se inicializan los objetos y la configuración de sus estados
  - Se dispara del evento **Init** en cada uno de ellos, y luego el evento Init de la página contenedora
  - Se aplican la Master Page y los Themes que afecten a la página
- Carga del View State
  - ASP.NET carga en los estados de los controles y los valores del view state
  - Se procesan los datos del post
  - Se cargan los valores en cada uno de los objetos
- Carga de la página
  - Se dispara el evento **Page\_Load**

# Ciclo de Vida de una Página ASP.NET

## Evento de una Página ASP.NET



# Ciclo de Vida de una Página ASP.NET

## Sucesos mas importantes del ciclo de vida de una página

- Eventos Post Back
  - Se ejecutan los manejadores de eventos programados por el desarrollador.
  - Se ejecuta la lógica del negocio y acceso a datos.
- Graba View State
  - Se guardan el estado de los controles
- Render
  - El lenguaje de marcas es generado (html, xml, etc).
- UnLoad
  - Los controles hijos son descargados, momento en el cual se dispara el evento **UnLoad** de cada uno de ellos y el de la página que los contiene.
  - Es responsabilidad del recolector de basura (Garbage Collector) destruir (**Dispose**) todos los objetos incluidos en el ciclo de vida de la página.



# Controles de Servidor de Usuario

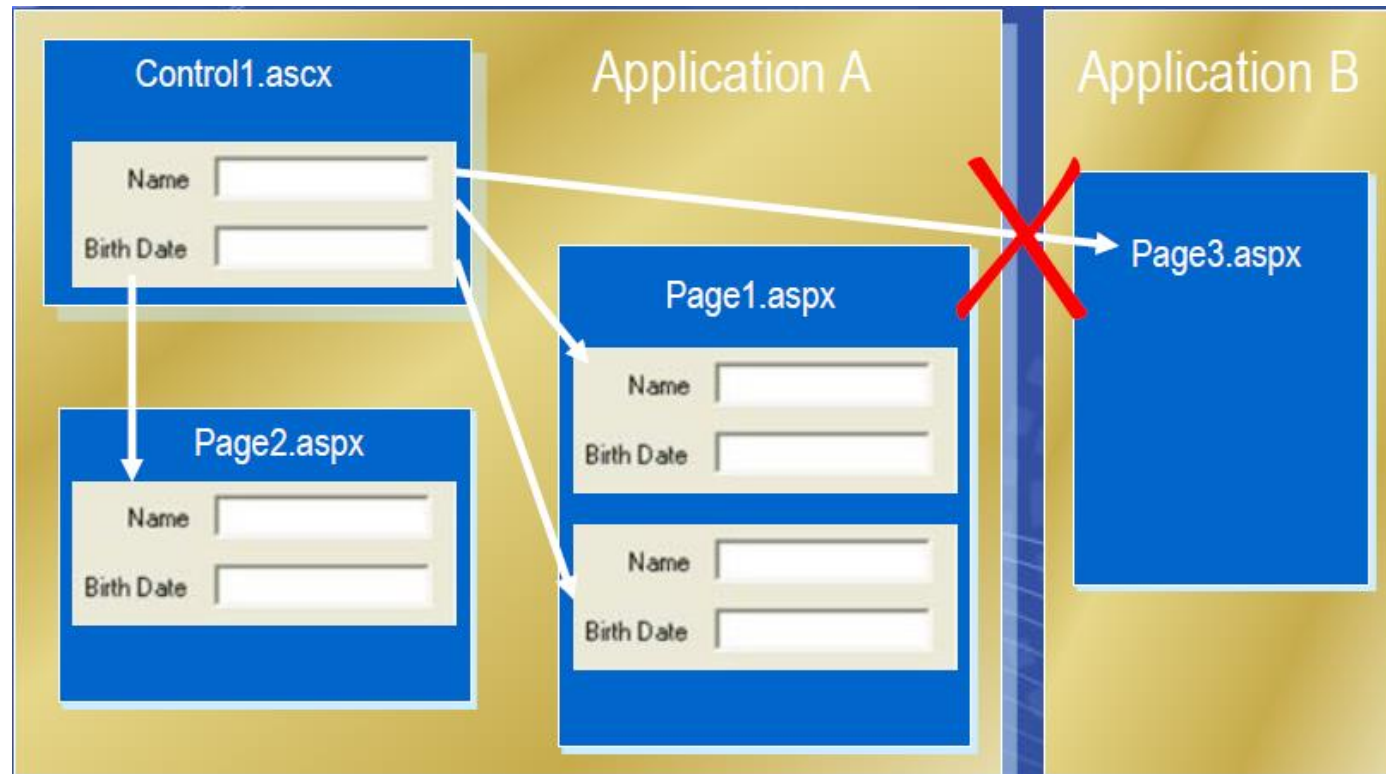
- Los **User controls** simplifican la reutilización de código e Interfaz de Usuario dentro de una aplicación web.
- Un **User Control** es un **Web Server Control** definido por el usuario con extensión “.ascx”
- Contienen HTML, pero no los tags <HTML>, <BODY>, o <FORM>
- Pueden tener también código del lado del servidor, ej c#

```
<%@ Control Language="C#" %>
```

# Controles de Servidor de Usuario

## ¿Por que utilizarlos?

- Para reutilizar interfaz de usuario y código
- Pueden estar escritos en diferentes lenguajes



# Controles de Servidor de Usuario

## Pasos

- Usar la directiva @Register para incluir un user control en una pagina ASP.NET:

```
<%@ Register TagPrefix="uc" TagName="controlTxt" Src="miTextBox.ascx" %>
```

- Insertar el user control en un Formulario Web:

```
<uc:controlTxt id="ucDireccion" runat="server"/>
```

- Debemos codificar propiedades en el user control para que pueda ser accedido:

```
public string Texto
{
    get{ return txt.Text;}
    set{ txt.Text = value;}
}
```

- Accedemos a la/s propiedad/es desde la página que contiene el user control:

```
string direccion =ucDireccion.Texto;
```

# **Licenciatura en Gestión Tecnológica**

## **Programación Avanzada 2**



# **Muchas Gracias**

**Ing. Mariano Juiz**