

# DESARROLLO WEB CON ASP.NET



# Historia del desarrollo web

- **HTML (1991)**
  - Páginas estáticas orientadas al contenido
- **CGI (1993)**
  - (Common Gateway Interface) Ejecutables que generan HTML dinámicamente
  - Versión mejorada: ISAPI. La aplicación Web no era un ejecutable independiente, sino un plug-in. En caso de Windows se trataba de una DLL que era invocada en el propio contexto del servidor Web.
- **ASP (1996), PHP (1997), JSP (1999)**
  - Páginas HTML que incluyen código script dinámico

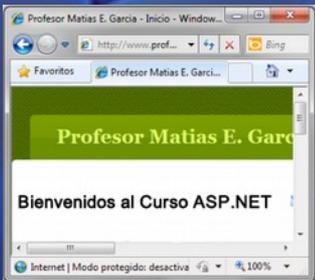
# Aplicaciones Web

## Cliente

`http://  
www.profmatiasgarci  
a.com.ar/apweb.html`

Internet  
DNS

`IP=173.237.190.67  
Puerto: 80`



## Servidor



`www.profmatiasgar  
cia.com.ar  
IP =  
173.237.190.67`

```
apweb.html
<html>
<body>
Bienvenidos al
Curso ASP.NET
</body>
</html>
```

**HTTP Request**

**HTTP Response**

# Aplicaciones Web

- Una aplicación web es un programa que genera archivos en HTML (páginas) que se transmiten por medio del protocolo HTTP de un servidor al cliente y viceversa, brindando distintas funcionalidades a un usuario final.
- El HTTP define como los navegadores y los servidores Web se comunican uno con otro.
- En una aplicación web, el componente principal es el HTML Form, que es el elemento de html que contiene los controles de la interfaz de usuario de cada página y captura la entrada de datos del usuario. El HTML Form es la porción de la página que es enviada a través del protocolo HTTP al servidor para procesar el pedido realizado por el usuario.

# Servidor Web

- Un servidor web es un programa que ejecuta una aplicación web que genera las páginas disponibles para ser visitadas por los usuarios de la red. A la computadora que ejecuta este programa también se lo llama servidor 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
- las páginas ASP tienen la extensión .ASP y son procesadas por la DLL ASP.DLL y sin embargo las páginas ASP.Net poseen la extensión .ASPX y son procesadas por el entorno de ejecución .NET Framework, que las transforma en el código intermedio, representado en este caso por la DLL ASPNET\_ISAPI.DLL.

# Hypertext Transfer Protocol (HTTP)

- HTTP es un protocolo de Internet que define la sintaxis y la semántica que deben utilizar los navegadores y los servidores Web para poder comunicarse
- Esta basado en texto y es transmitido sobre conexiones TCP
- Es un protocolo orientado a transacciones y sigue el esquema petición-respuesta entre un cliente y un servidor. Al cliente que efectúa la petición (un navegador web o un spider) se lo conoce como "user agent" (agente del usuario). A la información transmitida se la llama recurso y se la identifica mediante un localizador uniforme de recursos (URL). Los recursos pueden ser archivos, el resultado de la ejecución de un programa, una consulta a una base de datos, la traducción automática de un documento, etc.

# HTTP Request

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

# HTTP Response

HTTP/1.1 **200** OK

Server: Microsoft-IIS/5.0

Date: ...

Content-Type: **text/html**

Accept-Ranges: bytes

Last-Modified: ...

ETag: ...

Content-Length: 46

**[blank line]**

**<html>**

**<body>**

**Bienvenidos al Curso ASP.NET**

**</body>**

**</html>**

# HyperText Markup Language HTML

- Lenguaje de marcado predominante para la elaboración de páginas web que se utiliza para describir y traducir la estructura y la información en forma de texto, así como para complementar el texto con objetos tales como imágenes.
- El HTML se escribe en forma de «etiquetas», rodeadas por corchetes angulares (<, >), por ejemplo <tag> .... </tag>.
- HTML también puede describir, hasta un cierto punto, la apariencia de un documento, y puede incluir un script (por ejemplo JavaScript), el cual puede afectar el comportamiento de navegadores web y otros procesadores de HTML.



# Qué es ASP.Net?

- Las páginas ASP.Net (Active Server Pages), también llamadas páginas activas o dinámicas, son páginas que contienen código HTML, script de cliente y un código que se ejecuta en el servidor, dando como resultado código HTML. Por lo tanto al cargar una página ASP.Net en nuestro navegador, en realidad no estamos cargando la página ASP.Net como tal, sino el resultado de la ejecución de la página, es decir la salida de la página ASP.Net, que se trata de código HTML. Es decir, son páginas que se ejecutan en el servidor enviando como resultado al cliente código HTML.
- Podemos definir ASP.Net como el entorno de desarrollo incluido en la plataforma .NET de Microsoft, que nos permite desarrollar completas aplicaciones Web que se ejecutarán sobre el entorno ofrecido por el .NET Framework

# Qué es ASP.Net?

- ASP.Net es un “Marco” (framework) para programar aplicaciones web, de un modo similar al que se programan las aplicaciones windows. El componente principal son los **Web Forms** (formularios web) que permiten, entre otras cosas, separar la interfaz del usuario de la funcionalidad de la aplicación.
- ASP.Net no es un lenguaje de programación, sino que es un entorno de programación o tecnología de desarrollo, pero no es un lenguaje. De hecho ASP.Net soporta distintos lenguajes como Visual Basic.Net, C# o JScript.
- ASP.Net permite generar aplicaciones web con el paradigma de Windows

- Diseño + lógica
- Componentes gráficos

- Eventos
- Propiedades
- Estado

# Qué es ASP.Net?

VB

C++

C#

JScript

...

Common Language Specification

ASP.NET: Web Services  
and Web Forms

Windows  
Forms

ADO.NET, XML, Enterprise Services

Base Class Library

Common Language Runtime

Visual Studio.NET

# Componentes de las aplicaciones

## Web ASP.Net

- Formularios Web o páginas .ASPX: Proveen de la interfase visual. No tienen código ejecutable.
- Páginas de Code-behind: Están asociadas con cada formulario y son las que proveen del código ejecutable del lado del servidor.
- Archivos de Configuración: Son archivos que permiten configurar la aplicación, por ejemplo el archivo Web.config y el servidor por ejemplo el archivo machine.config.
- Global.asax: Es un archivo que contiene código. Este código responde a eventos que se disparan en la aplicación Web.
- Referencias a servicios web xml.
- Conectividad a bases de datos.
- Sistema de caché.

# Web Forms

- Combinación de código HTML + controles que se ejecutan en el servidor
- Pueden usarse para crear páginas Web programables que sirvan como interfaz de usuario de las aplicaciones Web.
- Extensión .aspx
- Formularios semejantes a los Windows Forms pero en Web
  - Separación lógica - diseño
  - Componentes gráficos
  - Estado
  - Eventos
  - Propiedades
- Admiten cualquier lenguaje compatible con Common Language Runtime de .NET, incluidos Visual Basic, Visual C# y JScript.NET.

# Modelo de Código de Páginas Web

- Una Página Web ASP.Net consta de dos partes:
  - Elementos visuales, que incluyen controles del servidor y texto estático.
  - Programación lógica para la página, que incluye manejador de eventos y el resto del código.
- Modelos de Página
  - Modelo de Página Single-File: el HTML y el código de programación se almacena en el mismo .aspx. El código de programación es un bloque que contiene el atributo `runat="server"`, para marcar al código como código ejecutable por ASP.NET
  - Modelo de Página Code-Behind: el HTML esta en un archivo .aspx y el código de programación en otro archivo .aspx.

# Atributos de una página

- Tres niveles de atributos

- Page, Definen funciones globales

- ```
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="inicio.aspx.cs" Inherits="inicio" %>
```

- Código, gestión de eventos del formulario

- ```
<script> .... </script>
```

- LayOut, presentación de los datos

- Body, Como se va a mostrar la página

- ```
< body>
```

- ```
....
```

- ```
</body>
```

- Form, como se procesan grupos de control

- ```
< form id="Form1" method="post" runat="server">
```

- ```
....
```

- ```
</form>
```

# Etiqueta <@ Page >

- **Language**
  - Lenguaje en el que está escrita
  - Vb, c#, jscrip
- **CodeFile**
  - Qué página contiene el código
- **Inherits**
  - Clase con la que va a trabajar
- **AutoEventWireup**
  - Si hay que especificar los procedimientos para los eventos o los toma dependiendo del nombre del procedimientos

# Etiqueta `<script>`

- Código para ejecutar eventos del cliente
- `<script runat="server">` el evento se ejecuta en el servidor
- Se recomienda programar los eventos de servidor en el archivo de código asociado para diferenciarlo del código que va a interpretar el navegador

# Etiqueta `<form>`

- No tiene que ver con el concepto de formulario.
- Puede tener varios `<form>` .. `</form>` en una web pero solo uno que se ejecute en el lado servidor
- Method, como se envían los valores al servidor
  - Post, datos se envían en parejas nombre/valor dentro de la petición http
  - Get, datos se pasan como un string
- Runat, identifica si se devuelve información al servidor
  - Runat="server", envía la información al servidor
  - No tiene, se ejecuta como un formulario HTML

# Tipos de controles servidor

- **HTML server controls**

- Son elementos HTML que contienen atributos que los hacen visibles (y programables) en un servidor. Exponen un modelo de objeto que se relacionan muy estrechamente con los elementos HTML que procesan.
- Añadir `runat="server"` para tratarlo en el modo servidor

- **Web server controls**

- Controles con más funciones incorporadas. incluyen no sólo controles de tipo formulario como botones y cuadros de texto, sino también controles con fines especiales como un calendario. Los controles de servidor Web son más abstractos que los controles de servidor HTML pues su modelo de objetos no refleja necesariamente la sintaxis HTML.
- Integrados, equivalentes a los controles HTML
- Validación, testear los datos introducidos
- Enriquecidos, funcionalidad avanzada
- Enlazados a datos
- Controles Web, presentación de datos en navegadores

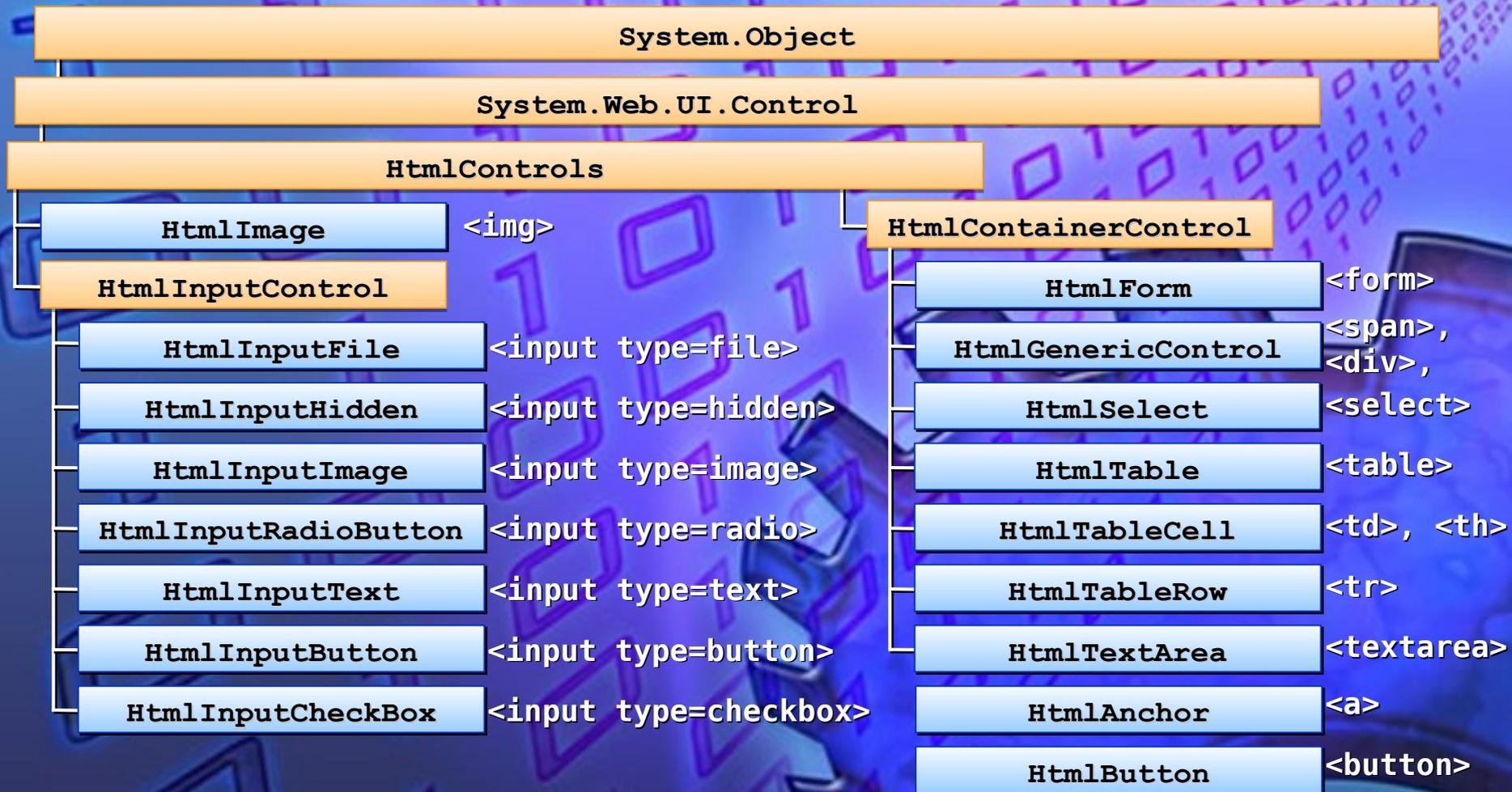
# Controles de servidor HTML

- Acceso desde el lado de servidor
- Las propiedades son todas de tipo String, por lo que solo se detecta los errores cuando se ejecuta
- Están dentro de una etiqueta `<Form ... Runat="server">`
- Namespace `System.Web.UI.HtmlControls`
- Permite actualizar rápidamente una página html con nuevas funcionalidades

```
<input type="text" id="textname" runat="server">
```

- Ofrecen las funciones siguientes:
  - Un modelo de objetos que pueda volver a programar en el servidor con las técnicas habituales orientadas a objetos.
  - Un conjunto de eventos para los que pueda escribir controles de eventos de la misma forma que lo haría en un formulario basado en cliente, con la excepción de que un evento se controla en código del servidor.
  - Mantenimiento automático del estado del control. Si el formulario realiza una acción de ida y vuelta al servidor, los valores que el usuario escriba en los controles de servidor HTML se mantendrán automáticamente cuando la página se envíe de vuelta al explorador.
  - Compatibilidad con estilos HTML 4.0 si la página de formularios Web Forms se muestra en un explorador que admite hojas de estilos en cascada.

# Controles de servidor HTML



# Controles de servidor WEB

- Creados para ASP.Net
- No tienen funcionalidad si no tienen una etiqueta `runat="server"`
- Propiedades son tipadas, permite detectar errores antes de compilar
- Namespace `System.Web.UI.WebControls`
- Ofrecen todas las funciones descritas anteriormente para los controles de servidor HTML (excepto la asignación uno a uno a elementos HTML) y estas funciones adicionales:
  - Un modelo de objetos enriquecido que proporciona capacidades de programación de tipo seguro.
  - Detección automática del explorador.
  - La capacidad de definir su propia apariencia para el control mediante plantillas.
  - La capacidad de especificar si un evento del control provoca un envío inmediato al servidor o, en su lugar, se almacena en caché y se activa cuando se envía el formulario.
  - Capacidad para pasar eventos de un control anidado (como un botón en una tabla) al control contenedor.
  - Los atributos no son los de los elementos HTML. En lugar de ello, son propiedades del control Web.

# Controles de servidor WEB

System.Object

System.Web.UI.Control

Repeater

Xml

WebControls

AdRotator

ListControl

BaseDataList

RadioButtonList

DataGrid

CheckBoxList

DataList

DropDownList

Button

ListBox

Calendar

Panel

CheckBox

Table

RadioButton

Image

HyperLink

ImageButton

TextBox

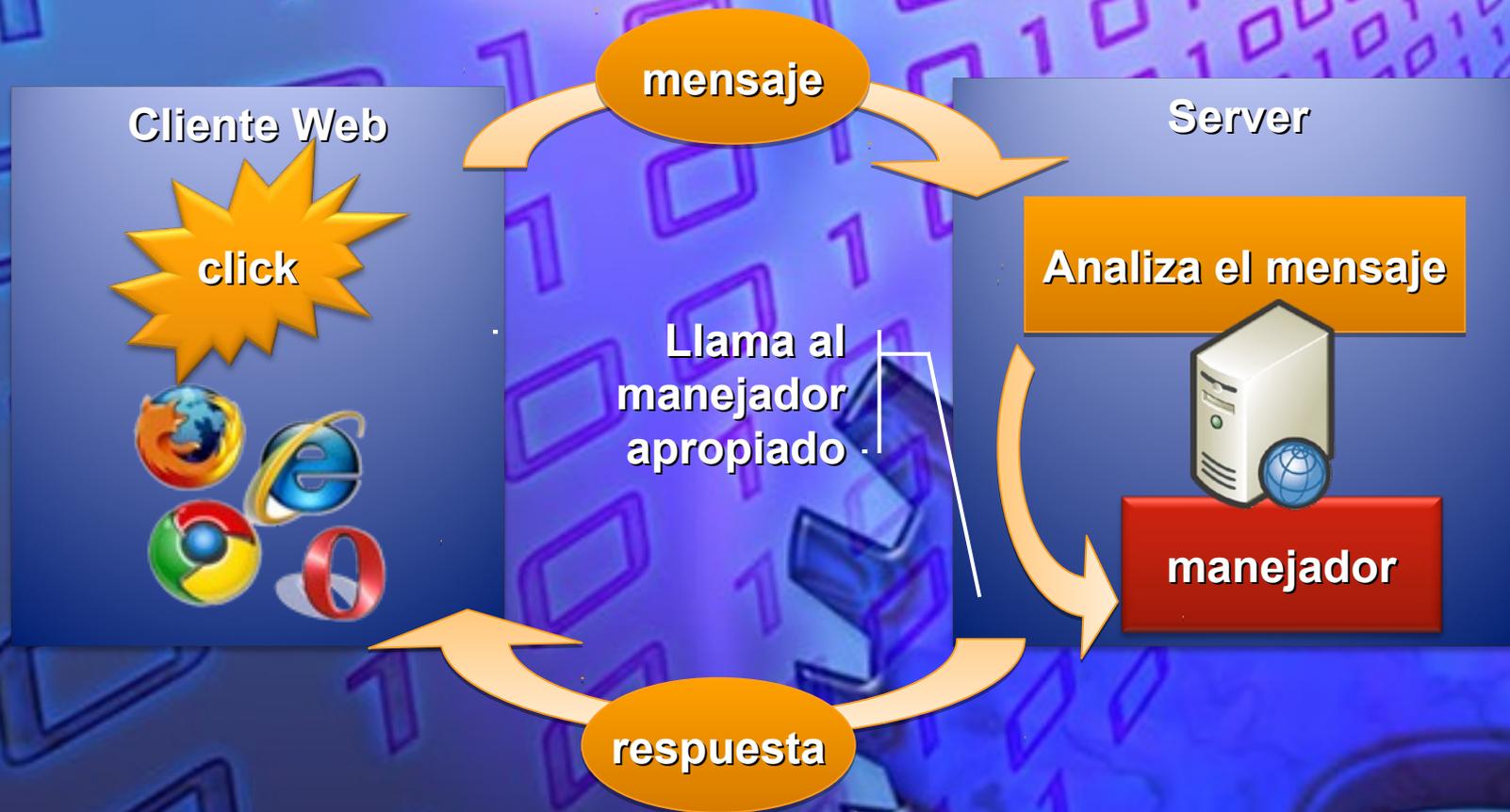
Label

# Eventos de servidor

- Los controles pueden lanzar eventos
  - Ya sean un HtmlControl o un WebControl
  - Load, Click, TextChange, SelectedItemChange...
  - Posibilidad de procesamiento:
    - En servidor
    - En cliente
    - Encolados hasta siguiente postback
- El cliente realiza una petición, el mensaje es recibido en el servidor, donde es analizado y atendido por el manejador de eventos que corresponda. Se arma la respuesta y el cliente obtiene la información que ha solicitado (ya sea una nueva página, un mensaje, etc.)

# Eventos de un webform

## Modelo de Eventos Web Form



# Eventos de un webform

- Los eventos asociados con controles ASP.NET son generados en el cliente y atendidos en el servidor. Por este motivo, ASP.NET necesita que la información para atender el evento del cliente sea transferida al servidor por medio de una petición HTTP Post.
- Para mantener el estado de los controles, se utiliza el View State. El estado de los controles se almacena en un campo oculto de la página que viaja en cada postback; si no se tiene control de su tamaño puede dificultar la navegación del sitio haciéndolo más lento. Para ello verifique el estado de la propiedad `EnableViewState` de todos los controles y utilice este mecanismo cuando sea necesario.
- un evento que se produce en el formulario web cuando es mostrado en el explorador cliente, genera un POST hacia el servidor y una respuesta. Este ida y vuelta dentro de un mismo formulario web se llama **postback**.

# Estado de un webform

- Cada vez que se pide una página se construye la página junto con todos sus controles y se destruye después de que se envíe al navegador para aumentar la capacidad del servidor.
- Los controles de servidor mantienen sus propiedades mediante dos mecanismos
  - Procesamiento automático de variables del post
  - Viewstate: variable de formulario oculta
- Soluciona el problema de que se limpien los controles cuando viaja entre el servidor y el cliente.
- Cada petición se actualiza el control

# Que es el View State

- Los controles de servidor de ASP.NET heredan de Control una propiedad denominada ViewState que les permite participar fácilmente en el mantenimiento del estado. El ViewState es un diccionario en el que se almacenan pares de nombre y valor. El área de trabajo de la página ASP.NET conserva ViewState en una variable de cadena que se envía al cliente y se recibe de vuelta de él como variable oculta. Cuando se produce la devolución, el área de trabajo de la página analiza la cadena de entrada de la variable oculta y establece la propiedad ViewState de cada control. Si en lugar de un campo privado un control utiliza ViewState para datos de propiedad, la propiedad se conservará automáticamente entre los trayectos de ida y vuelta hacia el cliente. Si una propiedad no se conserva en ViewState, es recomendable especificar su valor predeterminado en la devolución.

# Solución de problemas de rendimiento en ViewState

- Los programadores de controles deben tener en cuenta que todos los datos contenidos en ViewState recorren automáticamente un trayecto de ida y vuelta hasta el cliente. Estos trayectos suponen una carga adicional para el rendimiento, por lo que es importante hacer un uso razonable de ViewState. Enviar gran cantidad de datos a través de Internet puede generar grandes problemas en los tiempos y en el ancho de banda. Para solucionar estos problemas tenemos algunas opciones:
  - Deshabilitar el View State de los controles Web que sean necesarios configurando la propiedad `EnableViewState` en falso.
  - Deshabilitar el View State para la pagina entera configurando la propiedad `EnableViewState` a falso en al directiva `@Page`.
  - Guardar el View State en el servidor en lugar de ocultarlo en campo de formularios.

# Páginas Maestras

## ■ Definición

- Página Maestra → estructura heredable
- Página Contenido → personalizas cada página

## ■ Beneficios

- Sitio web con entorno consistente
- Ahorro de tiempo de programación
- Podemos tener varias páginas maestras en un proyecto

## ■ Añadir una página Maestra

- Añadir desde Explorador de Soluciones un nuevo archivo index.Master

## ■ Añadir una página de Contenido

- Añadir desde el explorador de soluciones un nuevo archivo del tipo WebForm activando la casilla Seleccionar página maestra

# Webgrafía & Licencia:

- Textos tomados, corregidos y modificados de diferentes páginas de Internet, tutoriales y documentos.
- Este documento se encuentra bajo Licencia Creative Commons 2.5 Argentina (BY-NC-SA), por la cual se permite su exhibición, distribución, copia y posibilita hacer obras derivadas a partir de la misma, siempre y cuando se cite la autoría del **Prof. Matías E. García** y sólo podrá distribuir la obra derivada resultante bajo una licencia idéntica a ésta.
- Autor:

*Matías E. García*

Prof. & Tec. en Informática Aplicada  
[www.profmatiasgarcia.com.ar](http://www.profmatiasgarcia.com.ar)  
[info@profmatiasgarcia.com.ar](mailto:info@profmatiasgarcia.com.ar)

