

Mi primera aplicación ASP.NET MVC 2 paso a paso – parte 1 « afelipelc Blog

Siguiendo el tutorial original : **Build your First ASP.NET MVC Application** ::

<http://www.asp.net/mvc/tutorials/getting-started-with-mvc-part1>

Conociendo los conceptos de los elementos del patrón de diseño Modelo Vista Controlador, en este tutorial pondremos en práctica los conceptos básicos de ASP.NET MVC 2 creando una aplicación básica a la que agregaremos funciones y elementos extras a los mostrados en el tutorial original que encuentras en <http://www.asp.net/mvc>.

Para saber más acerca de ASP.NET MVC, visita el sitio <http://msdn.microsoft.com/es-es/library/dd394709.aspx>.

Construyendo nuestra aplicación de colección de Películas.

Requisitos: Visual Studio 2010 y SQL Server Express 2008.

La aplicación que construiremos contara con una base de datos llamada Peliculas la cual tendrá dos tablas relacionadas. Agregaremos la función de generar reportes en formato PDF y un módulo de Administración de usuarios, funciones que agregaremos poco a poco durante el trayecto de la construcción.

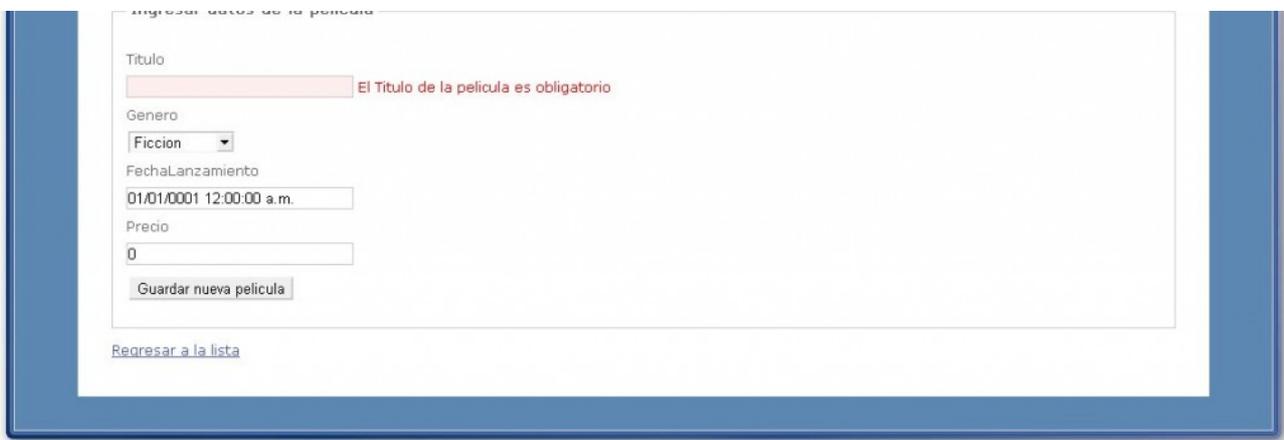
A partir de este proyecto base, podremos crear otras aplicaciones con ASP.NET MVC 2.

El resultado final de nuestra aplicación será similar a la siguiente captura. Observa que como extra, tiene una columna de CheckBoxes vinculados a una función que hace uso de AJAX.



Img 1.- Vista preliminar de como quedará nuestra aplicación

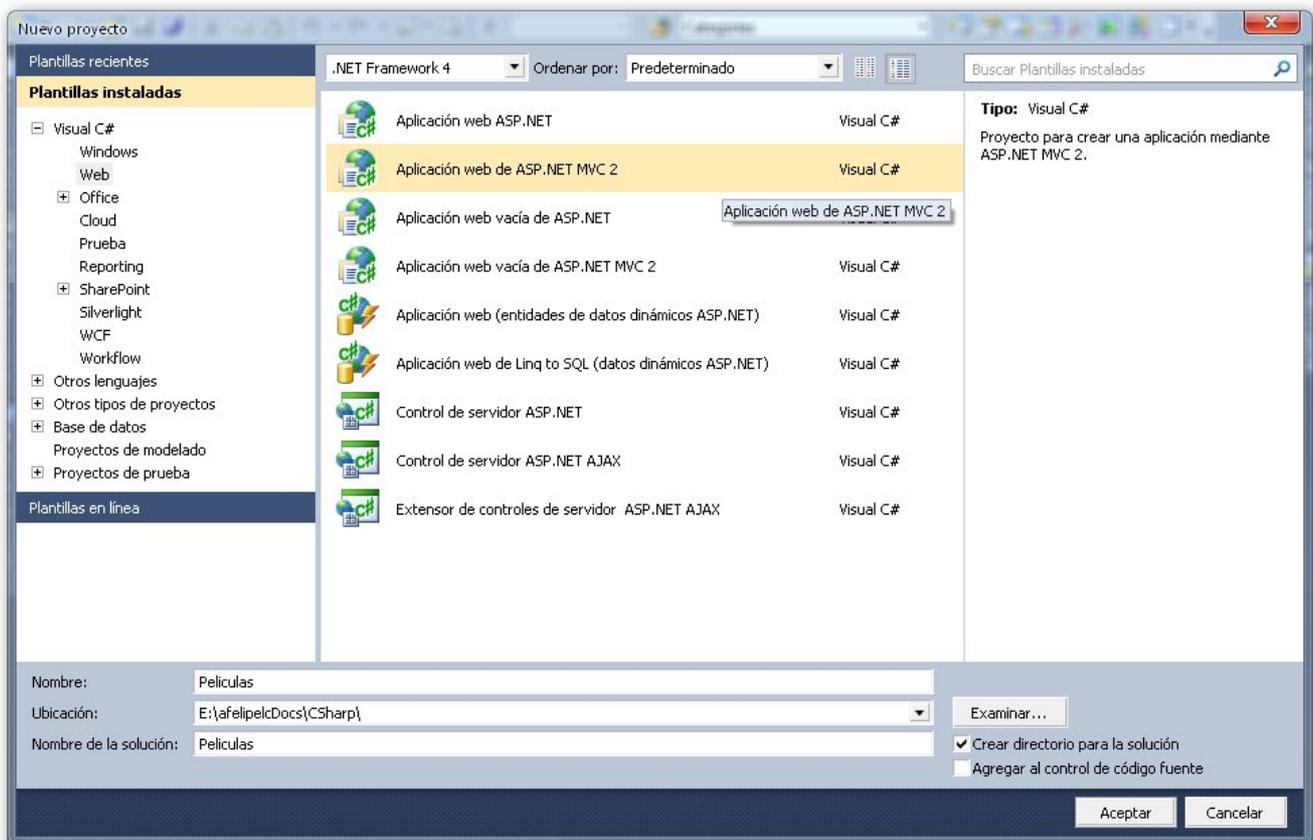




Img 2.- Vista Preliminar de como quedará nuestra aplicación.

Empezando a crear el proyecto.

En Visual Studio -> Crear nuevo Proyecto -> C# /Web elegimos Aplicación web de ASP.NET MVC 2, elegimos donde guardarlo y le ponemos el nombre de Películas.

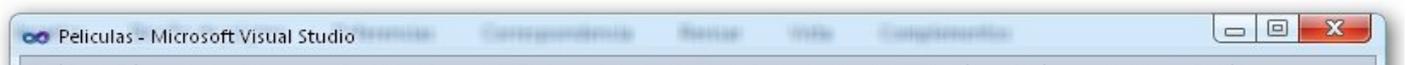


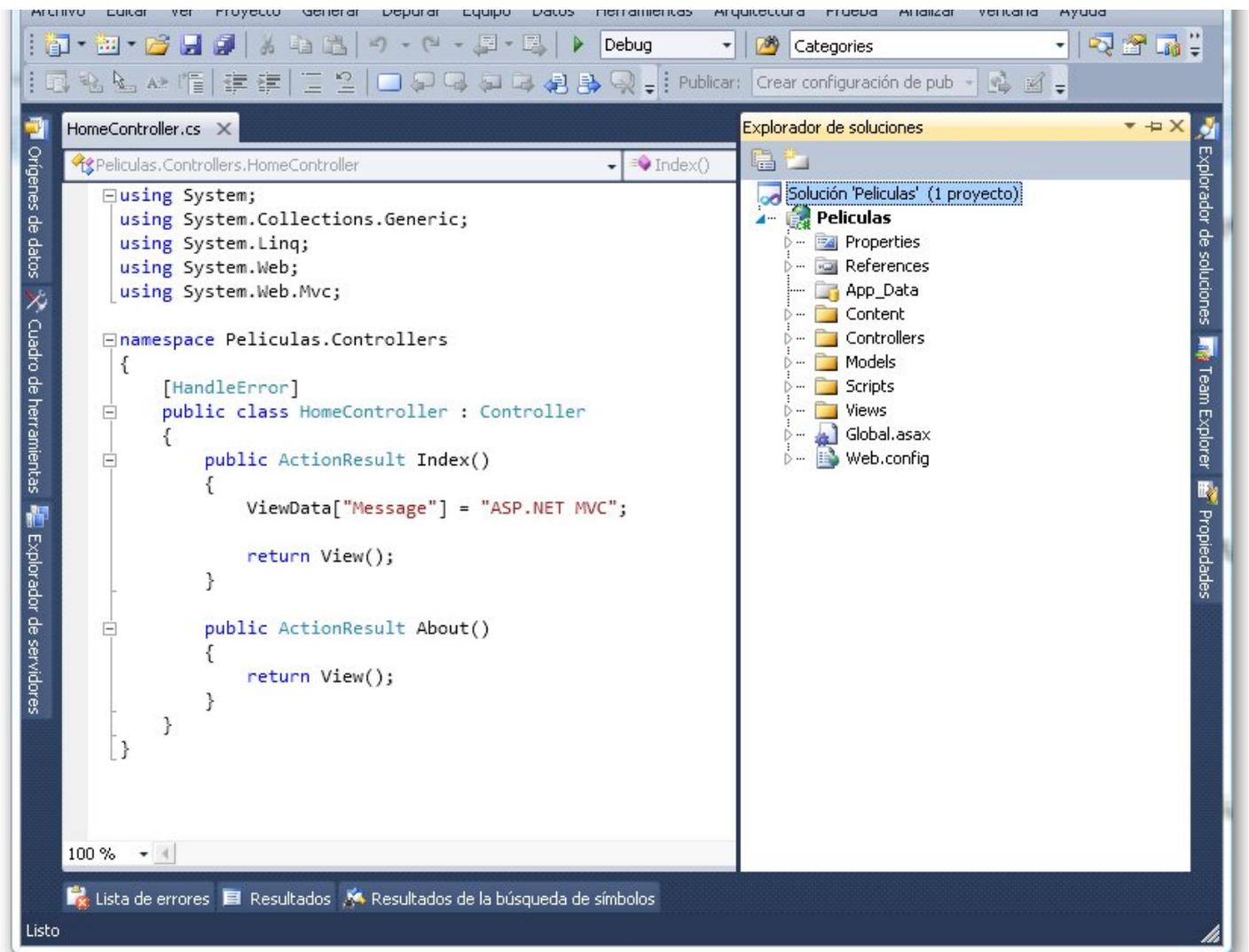
Img 3.- Creando el Proyecto

Al crear el proyecto nos pregunta si queremos crear un proyecto de prueba unitaria, elegimos **NO** y damos clic en Aceptar.

¿Qué nos genera el proyecto?

Por default nos genera la estructura de nuestra aplicación (carpeta de controladores, vistas, etc.), una Plantilla o estructura de nuestra página web llamada Site.Master, un controlador llamado HomeController que será el encargado de generar los datos para la página principal por medio de Acciones (Métodos) de la clase **ActionResult**. También nos crea un controlador con sus vistas (formulario de Login) para las sesiones de los usuarios.





Img. 4.- Estructura del proyecto y el HomeController

El controlador HomeController

Parte del código del controlador

```

//Metodo que devuelve la vista (pagina) principal del HomeController al navegador web
//Por ejemplo: http://localhost/
public ActionResult Index()
{
    //Por medio de ViewData["Parametro"] = Valor, Pasamos datos a la vista
    ViewData["Message"] = "ASP.NET MVC";

    //Devolver la vista al navegador, en este caso la vista creada es Index.aspx
    return View();
}

```

Nota: La vista `Index.aspx` es creada al crear el proyecto, cuando agreguemos nuevas acciones o controladores, debemos crear sus vistas.

Los Métodos tienen una sintaxis de la forma:

```

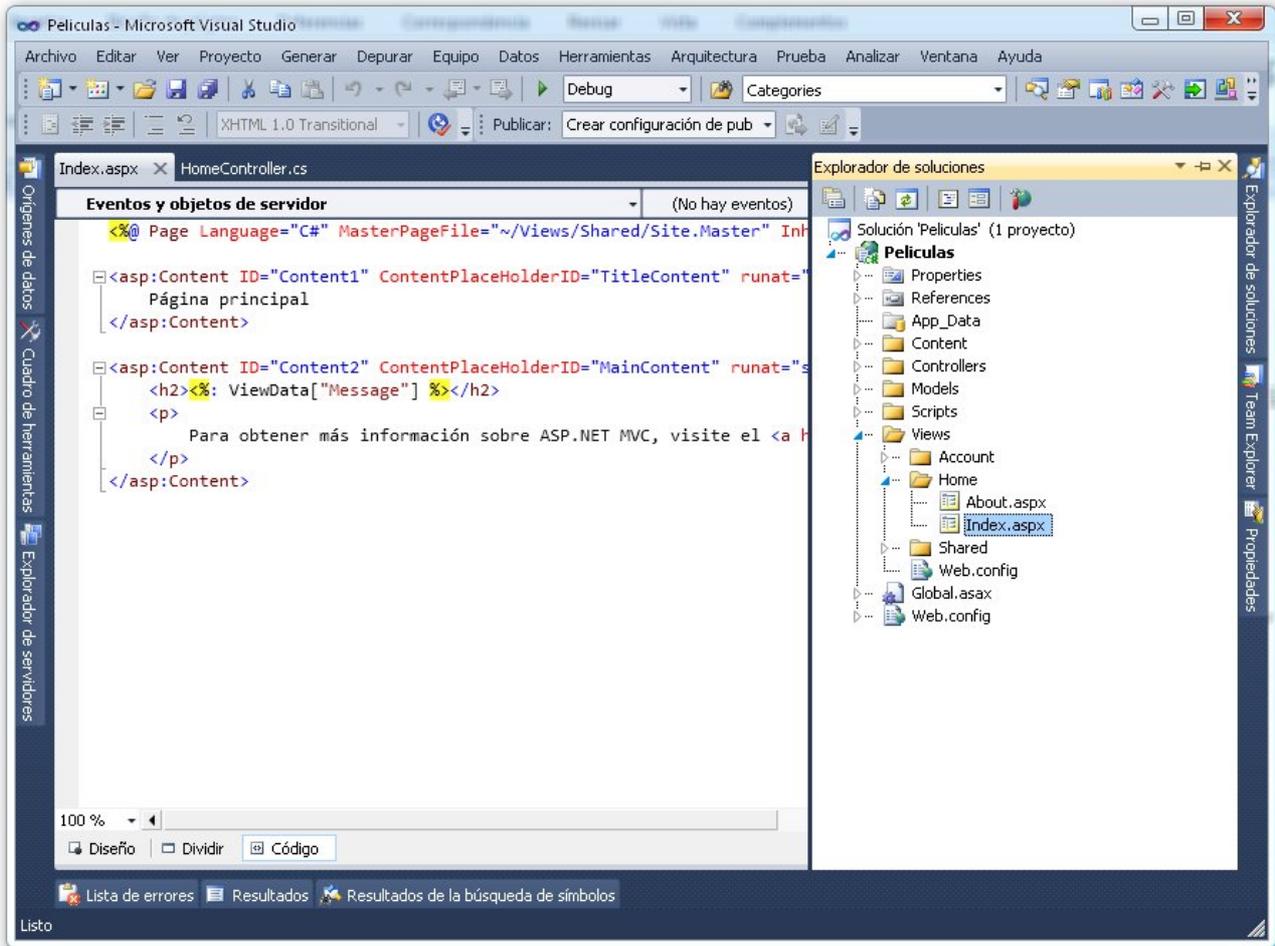
public ActionResult NombreMetodo()
{
    //Todas las sentencias que deberá ejecutar el método
    //Generar un resultado si fuera el caso

    //Devolver a la vista el resultado generado
    return View();
}

```

Más adelante veremos ejemplos de como pasar parametros a los Metodos.

El código de la vista Index.aspx generado:



Img 5.- La Vista >> Index.aspx

Si en una página asp queremos agregar un comentario, lo hacemos de la forma:

```
<% //Ejemplo de comentario en asp %>
```

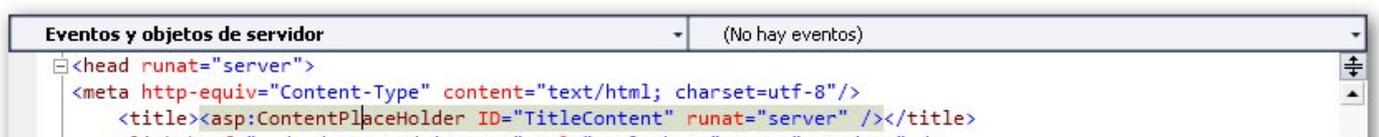
```
<asp:Content ID="Content1" ContentPlaceHolderID="TitleContent" runat="server">
Bienvenido :: Coleccion de Peliculas
<% // Todo lo que este dentro del bloque TitleContent, será el titulo de nuestra pagir
</asp:Content>
```

```
<asp:Content ID="Content2" ContentPlaceHolderID="MainContent" runat="server">
<% //Todo lo que este dentro del bloque MainContent, será lo que se muestre como conte

<%
// Imprimir en la pagina el contenido del parámetro Message enviado por el controlador

    <h2><%: ViewData["Message"] %></h2>
    <p>
        Para obtener más información sobre ASP.NET MVC, visite el <a href="http://asp.
    </p>
</asp:Content>
```

Site.Master (Plantilla, template, layout o estructura de nuestro sitio como estes acostumbrado a llamarle) Se encuentra en la carpeta Views (Vistas) / Shared (en el explorador de soluciones).

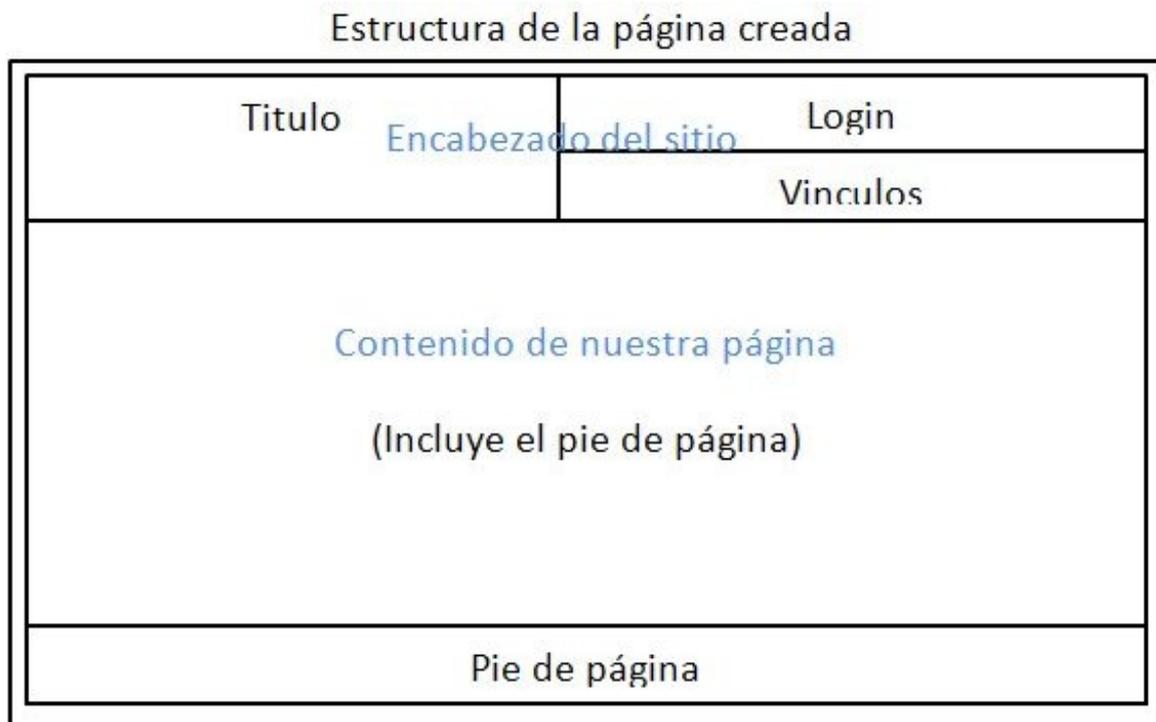


```

<link href= .../Content/Site.css rel= stylesheet type= text/css //
</head>
<body>
  <div class="page">
    <div id="header">
      <div id="title">
        <h1>Mi aplicación de MVC</h1>
      </div>
      <div id="logindisplay">
        <% Html.RenderPartial("LogOnUserControl"); %>
      </div>
      <div id="menucontainer">
        <ul id="menu">
          <li><%: Html.ActionLink("Página principal", "Index", "Home")%></li>
          <li><%: Html.ActionLink("Acerca de", "About", "Home")%></li>
        </ul>
      </div>
    </div>
  </body>
</html>

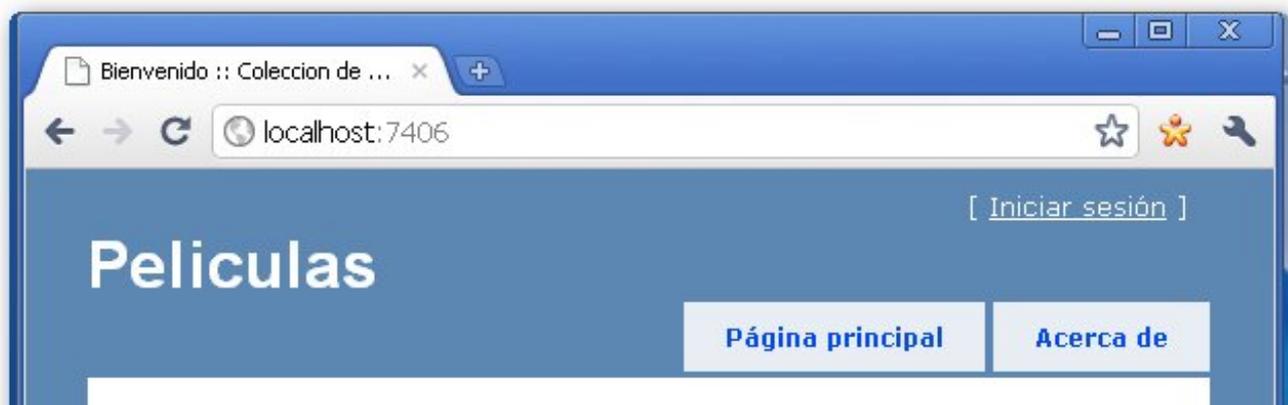
```

Img 6.- El layout **Site.Master**



Img 7.- Estructura del sitio generado al crear el proyecto.

Si ejecutamos nuestra aplicación, obtendremos:



ASP.NET MVC

Para obtener más información sobre ASP.NET MVC, visite el <http://asp.net/mvc>.

En este espacio agregaremos el contenido de nuestra pagina. Pueden ser links, etc.

Creando mi aplicacion en ASP.NET MVC 2, 2010

Img 8.- Ejecutando el proyecto...

Hasta aquí, aun no agregamos nada a nuestro proyecto , es lo que nos ha generado automáticamente, ahora...



¿Cómo se generan los vínculos?

Como se ve en el código del Site.Master, en el contenedor de vínculos, tiene:

```
<li><%: Html.ActionLink("Acerca de", "About", "Home")%></li>
```

Html.ActionLink genera los vínculos a eventos de un controlador, en este ejemplo se está creando un vinculo al método About() del HomeController que resulta ser la dirección <http://localhost/Home/About> nótese que **no se pone la palabra "Controller"**, entonces la sintaxis es:

Cuando queremos llamar a una acción en el mismo controlador en el que estamos.

```
<%: Html.ActionLink("Acceso externo", // Texto del enlace
                    "Editar",         // Acción
                    new { id=Model.Id } // Parámetros
                )
%>
```

Ejemplo:

Mientras estoy en el controlador MoviesManagerController >>> <http://localhost/MoviesManager>

```
<%: Html.ActionLink("Editar", "Edit", new { id=item.PeliculaId }) %>
<% //Genera el enlace http://localhost /MoviesManager/Edit/xNumero%>
```

```
<%: Html.ActionLink("Eliminar", "Delete", new { id=item.PeliculaId })%>
<% //Genera el enlace http://localhost /MoviesManager/Delete/xNumero %>
```

Y cuando queremos acceder a la acción de otro controlador

```
<%: Html.ActionLink(
    "Acceso Externo", //Texto del enlace
    "Editar",         // Acción
    "Productos",     // Controlador
    new { id=Model.Id } // Parámetros
) %>
```

Por Ejemplo cuando estamos en la página principal y queremos acceder a otro controlador (otro nivel)

```
<%: Html.ActionLink("Películas", "Index", "Movies")%>
<% // Genera el enlace http://localhost /Movies%>
```

Observa que para mandar a imprimir directamente en la vista, asp.net lo hace:

```
<%: Variable_a_imprimir%>
```

```
<% //Despues de abrir el codigo asp, se ponen dos puntos ":" seguido de la variable u
```

Esta es la primera parte de nuestra aplicación, posteriormente continuamos con la parte 2 donde agregamos nuestra base de datos, el Entity Data Model y los controladores...

Mi primera aplicación ASP.NET MVC 2 paso a paso – parte 1, 4.3 out of 5 based on 4 ratings