

Capacitación MVC

Día 1



PUESTO N°8



NIVEL 3



EMPRESA INFORMÁTICA
2000



EXPORTACIÓN
DE SERVICIOS

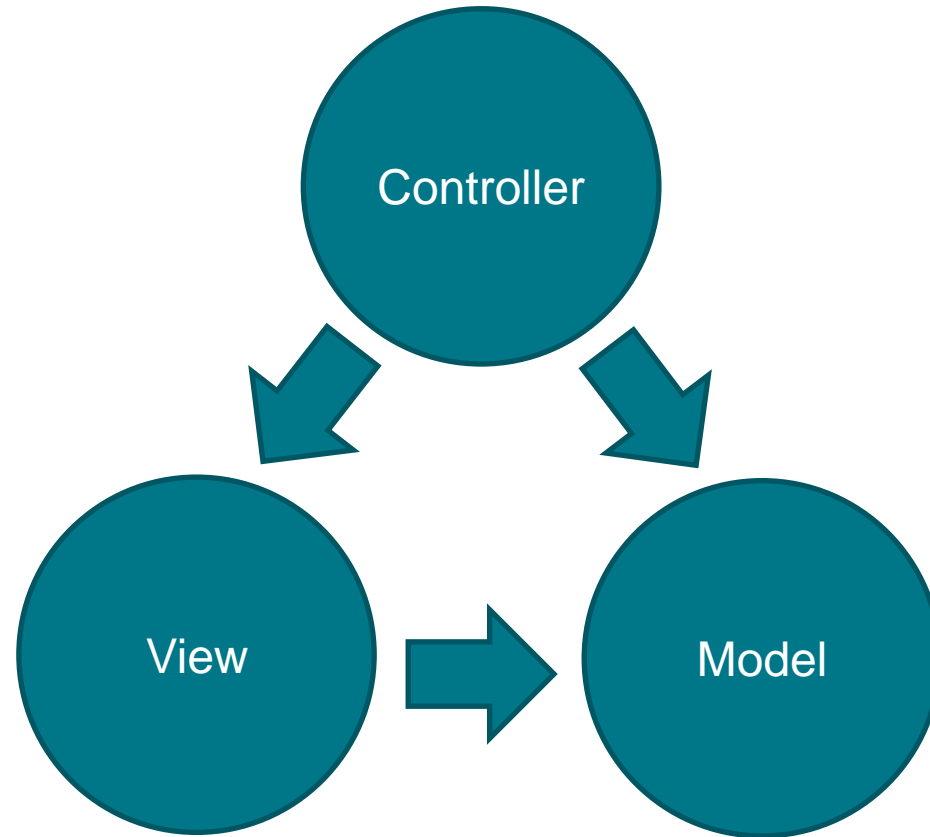


Agenda

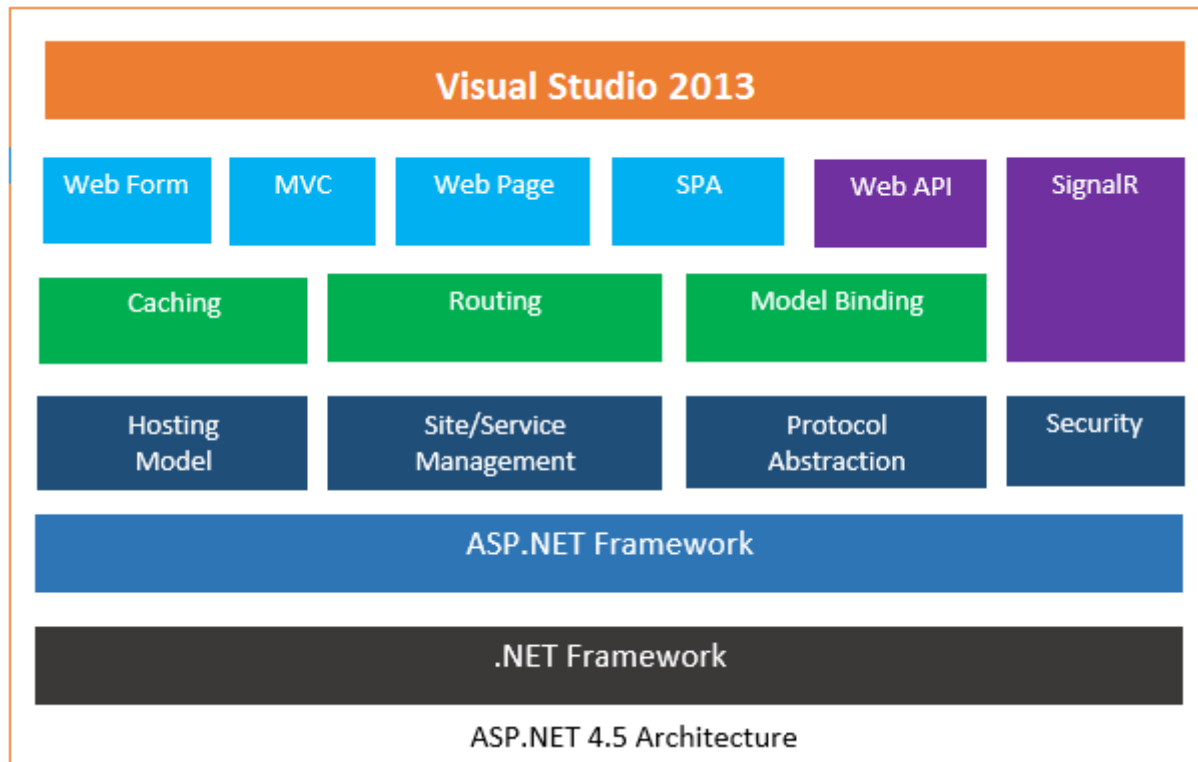
- Introducción a MVC
 - Patrón MVC
 - Asp.net MVC – Estructura
- Generando URLs
- Routing
- Action Results – Resultado de acciones?
- Pasando datos de un Controlador a una vista
- Html Helpers - ¿Que son? ¿Cuáles hay?
- Pasando parámetros a un Action
- Dando formato a nuestra aplicación
- Display Templates – Personalizando los tipos de datos.



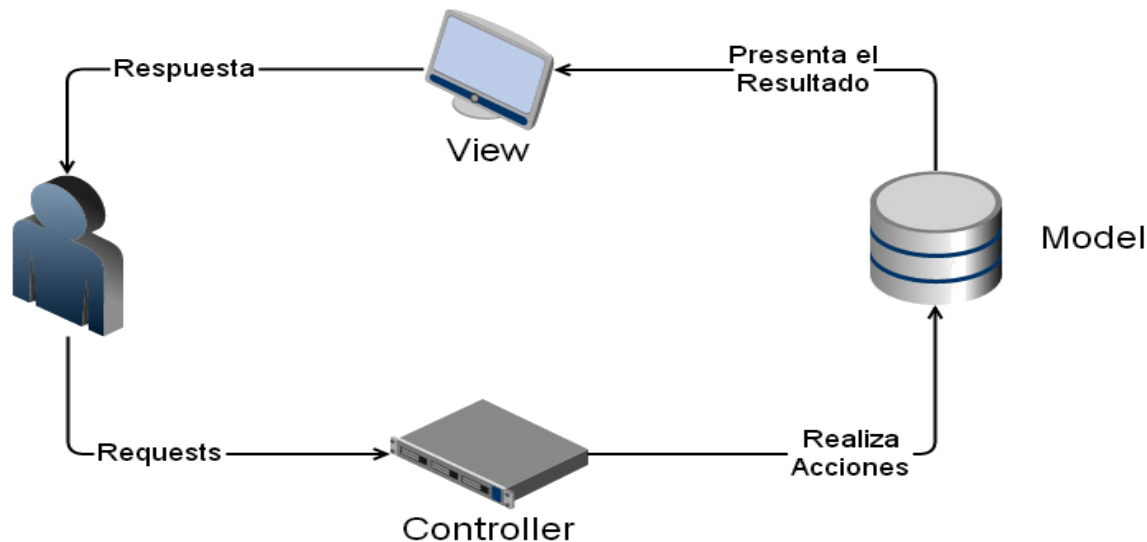
Patrón MVC



ASP.NET MVC - Estructura



ASP.NET MVC – Patrón aplicado al framework web



- **Controller:** Es el encargado de recibir los inputs y realizar acciones acordes
- **View:** Componente encargado de presentar los resultados del controller
- **Model:** Dominio de la aplicación

La estructura de un proyecto MVC es la siguiente:

➤ **Controller**

- ***[Nombre Controller] + “Controller”***

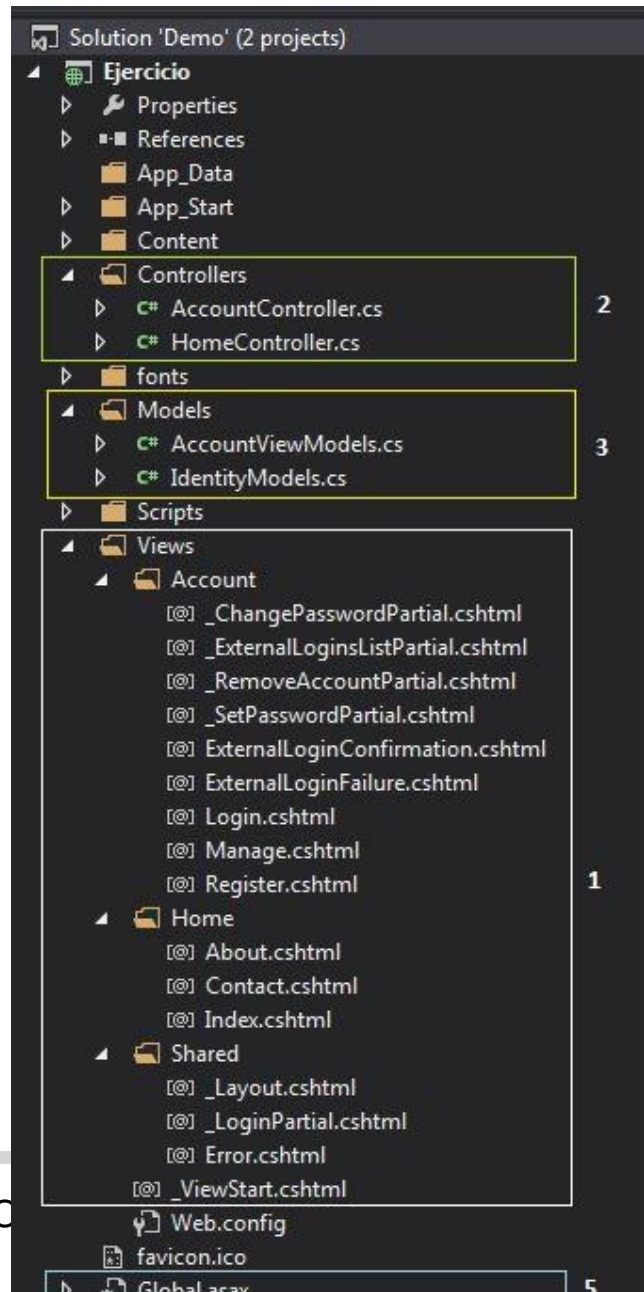
➤ **Views**

- ***[Nombre Controller]***
 - *[Nombre Vista]*
- **Share**
 - Vistas Comunes
- Web.Config (de vistas)

➤ **Model** (Puede no estar o puede tener modelo de vistas)



Introducción a MVC - Asp.net MVC – Estructura



➤ MVC localiza las Vistas usando una convención de nombres y directorios:

1. Views \ *[Nombre Controller]* \ *[Nombre Vista]*

2. Views \ Shared \ *[Nombre Vista]*





Demo

“Nuevo proyecto y Estructura”



Práctica

“Hola Mundo”

Razor

- Compacto, expresivo y fluido
- Fácil de aprender
- No es un nuevo lenguaje
- Compatibilidad con Intellisense
- Compatible con Test Unitarios



Code expressions

1 - Razor evalúa la expresión y el resultado es mostrado.

```
<h1>Listing @items.Length items.</h1>
```

2 - Las expresiones son automáticamente encodeadas.

3 - Utilizar @@ cuando hay código que no debe ser tratado como identificadores, como ocurre con cuentas de Twitter: @aspnet -> @@aspnet

```
<p>
```

```
    You should follow
```

```
    @@aspnet
```

```
</p>
```

Code blocks

1 - Requieren del caracter @ seguido de { }

```
@{  
    string s = "One line of code.";   
    ViewBag.Title "Another line of code";  
}
```

Server Side Comments

```
@*  
This is a multiline server side comment.  
@if (showMessage) {  
<h1>@ViewBag.Message</h1>  
}  
All of this is commented out.  
*@
```

Generando URLs

- Existen varias formas de generar urls en una aplicación MVC
 - Escribiendo la a mano (Para nada recomendado).
 - Usando el ActionLink helper
 - *@Html.ActionLink("Texto del Link", "Index", "About")*
 - Usando el UrlHelper y su metodo action
 - *UrlHelper helper = new UrlHeper(this.Request.RequestContext);*
 - *urlHelper.Action("Index","About");*

Estos clases generan urls del estilo

<http://ejemplo.com/About/Index>

Significado de esto

ROUTING



Routing

`http://example.com/users/edit/5`

controller id
action

```
public static void RegisterRoutes(RouteCollection routes) {  
    routes.IgnoreRoute("{resource}.axd/{*pathInfo}");  
  
    routes.MapRoute(  
        "Default",  
        "{controller}/{action}/{id}",  
        new { controller = "Home", action = "Index",  
              id = UrlParameter.Optional }  
    );  
}  
  
protected void Application_Start()  
{  
    RegisterRoutes(RouteTable.Routes);  
}
```

Route name URL with parameters Parameter defaults Ignore route Register routes at application start



Action Results – Resultado de acciones?

➤ Que es un Action Result?

- Un action result es una forma de encapsulamiento que tiene el framework para manejar el Reponse de una llamada desde un browser por ejemplo.

➤ Tipos de Action Result

- ContentResult
- EmptyResult
- FileResult
- HttpStatusCodeResult
- JavaScriptResult
- JsonResult
- RedirectResult
- RedirectToRouteResult
- ViewResultBase
 - ViewResult
 - PartialViewResult



Ejercicio Práctico – Creación de una Home Page



- El objetivo de este ejercicio es crear una aplicación MVC usando el template de MVC 5.
- Crear una Home Page con 2 links uno para Películas y otro para Géneros.
- Crear los controllers apropiados
- Crear las vistas apropiadas.

Pasando datos de un Controlador a una vista

- Para Pasar datos de un controlador a una vista existen varias formas, pero la idea es poder acceder a los datos del modelo en la vista.
- Las diferentes formas son:
 - ViewBag.
 - Clases
 - MVVM – Model View ViewModel.
 - JQuery using Json objects.
- A su vez usando MVVM o Clases se pueden tipar las views con lo cual para que esa view se renderice es necesario pasarle una instancia de esa clase



Html Helpers - ¿Que son? ¿Cuáles hay?

- Que es un HtmlHelper?
- Algunos tipos de Helpers definidos en el framework
 - Html.CheckBox `<input type="checkbox" />`
 - Html.DropDownList `<select></select>`
 - Html.Hidden `<input type="hidden" />`
 - Html.Label `<label for="" />`
 - Html.ListBox `<select></select>` or `<select multiple></select>`
 - Html.Password `<input type="password" />`
 - Html.Radio `<input type="radio" />`
 - Html.TextArea `<textarea></textarea>`
 - Html.TextBox `<input type="text" />`
- Existen variantes de estos cuando la vista esta tipada como DisplayFor
TextBoxFor, EditorFor, etc...



Html Helpers

```
@{  
    List<String> lista = new List<String>() { "Madrid", "Barcelona" };  
}  
@Html.DropDownList("Ciudades", new SelectList(lista), "Elige")
```

Formulario2

Index



Audi ▼
Audi
Mercedes
Opel

© 2013 - My ASP.NET MVC Application

```
<select id="Ciudades" name="Ciudades"><option>Madrid</option>  
<option>Barcelona</option>  
</select>
```



Data Annotations

Atributo Required

[Required]

```
public string FirstName { get; set; }
```

Este atributo dispara un error de validación si la propiedad es null o vacía

Atributo StringLength

[Required]

[StringLength(160)]

```
public string LastName { get; set; }
```

Este atributo generará un error de validación si el usuario ingresa un valor cuya longitud exceda el valor detallado entre paréntesis.

[Required]

[StringLength(160, MinimumLength=3)]

```
public string LastName { get; set; }
```

el parámetro MinimumLength es opcional.



Data Annotations

Atributo RegularExpression

```
[RegularExpression(@"[A-Za-z0-9._%+-]+@[A-Za-z0-9.-]+\.[A-Za-z]{2,4}")]  
public string Email { get; set; }
```

```
[RegularExpression(@"[A-Za-z0-9._%+-]+@[A-Za-z0-9.-]+\.[A-Za-z]{2,4}",  
ErrorMessage="Email doesn't look like a valid email address.")]  
public string Email { get; set; }
```

Permite asegurar que el valor ingresado por el usuario tendrá un determinado formato, como un valor válido de email.

Atributo Range

Permite establecer un mínimo y un máximo para un valor numérico. Puede ser utilizado con integers y doubles y hay otra sobrecarga para decimals:

```
[Range(35,44)]  
public int Age { get; set; }
```

```
[Range(typeof(decimal), "0.00", "49.99")]  
public decimal Price { get; set; }
```



Ejemplo practico – Lista de Géneros y Peliculas



- Continuando con el ejercicio anterior
- Agregar a la vista de géneros una grilla con todos los géneros.
- Utilizar el patrón de MVVM adaptado a MVC, para lo cual hay que agregar un ViewModel para dicha vista.

Pasando parámetros a un Action

- Como se dijo anteriormente MVC es desconectado y se maneja usando los verbos de HTTP.
- Como se pueden obtener los valores pasados por un post:
 - Usando los Actions de los controllers pero agregando dos cosas:
 - Atributos [HttpPost] o Get
 - Agregar parametros al action

```
1. //  
2. // GET: /Store/Browse?genre=Disco  
3. public string Browse(string genre)  
4. {  
5.     string message = HttpUtility.HtmlEncode("Store.Browse, Genre = " + genre)  
6.     ;  
7.     return message;  
8. }
```

- Leer valores directamente del Query string
- Usando Model binders o crear el propio.

```
string codeValue = "";  
if(Request.QueryString["code"] != null)  
{  
    codevalue = Request.QueryString["code"]  
}
```



Ejemplo practico – Buscando los géneros de acuerdo a su nombre.



- Completando el ejemplo anterior agreguemos la posibilidad de buscar por un nombre los géneros.
- Realizar lo mismo para las películas.

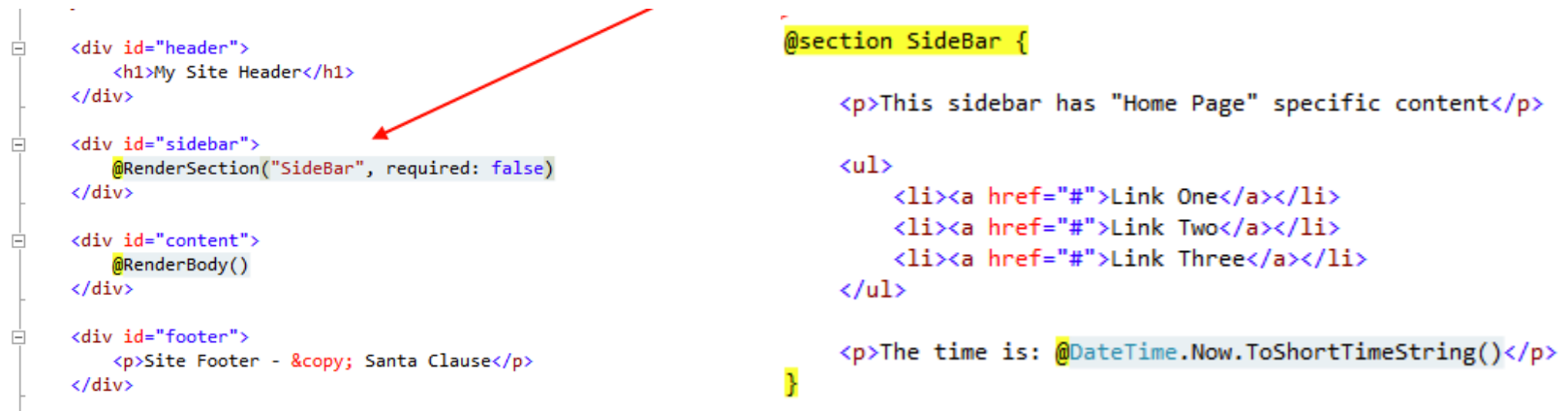
Dando formato a nuestra aplicación

Layouts

- Sirven para darle formato a las paginas en común.
- Reemplazan las Master Pages de Asp.Net WebForms.
- Se utilizan en las paginas de la siguiente forma
`@{ Layout = "~/Views/Shared/_Layout.cshtml"; }`
- Los layouts se pueden heredar.
- En general van en la Carpeta View/Shared ya que son compartidos.

Sections

- Una forma de dividir los layouts para proporcionar diferentes funcionalidades.
- Requieren un nombre y pueden ser requeridos o no.
- Se usan utilizando el @Section dentro de la pagina.
- Un ejemplo de section es el Body de la pagina y la barra de navegación.



```
<div id="header">
  <h1>My Site Header</h1>
</div>

<div id="sidebar">
  @RenderSection("SideBar", required: false)
</div>

<div id="content">
  @RenderBody()
</div>

<div id="footer">
  <p>Site Footer - &copy; Santa Clause</p>
</div>
```

```
@section SideBar {

  <p>This sidebar has "Home Page" specific content</p>

  <ul>
    <li><a href="#">Link One</a></li>
    <li><a href="#">Link Two</a></li>
    <li><a href="#">Link Three</a></li>
  </ul>

  <p>The time is: @DateTime.Now.ToShortTimeString()</p>
}
```



Ejemplo practico – Listar películas de acuerdo al genero



- Completando el ejercicio anterior se pide generar un listado de películas por genero.
- La misma debe aparecer cuando un se hace click en un genero.
- Debe tener un estilo diferente al de géneros.
 - Agregar un layout para géneros
 - Agregar un layout para Películas diferente al de géneros.
 - Utilizar secciones en el layout para separar la parte del header de HTML y del body.

Display Templates – Personalizando los tipos de datos.

- Los Display Templates son mecanismos para mostrar la información de una forma específica y unificada en todas las paginas de la aplicación.
- Para utilizar los display templates por defecto de la aplicación existen las siguientes formas:
- Html Helper
 - DisplayFor()
 - DisplayForModel()
 - EditorFor()
 - EditorForModel()
- En la propiedad de la clase utilizar el atributo “UIHint” y el nombre del template.
- El nombre del template se llamado igual que el tipo a mostrar.
- A su vez es posible generar todos los Display templates que se necesiten. Los mismos deben estar ubicados en Views\Shared\Display para que sean accesibles a todos lados.
- Ejemplos
 - ISBN
 - CUIT
 - Fecha



Ejemplo practico – Mostrar detalles de una película.



- Para finalizar el ejercicio agregar al mismo la posibilidad de ver los detalles de una película.
- Utilizar display templates para mostrar y formatear las diferentes secciones
 - Por ejemplo
 - La fecha
 - El Titulo



¿Preguntas?



ARGENTINA

Clay 2954

Capital Federal (C1426DLD)

tel: 54+11+5299 5400

Belgrano 133 - Piso 2

Bahía Blanca,

Buenos Aires (B8000IJC)

San Martín 902 - 1º Piso - Oficina 6

Paraná,

Entre Ríos (E3100AAT)

Calle 48 N°1165

La Plata,

Buenos Aires (1900)

tel: 54+11+5299 5400

BRASIL

Cardoso de Melo 1470 – 8, Vila Olimpia

San Pablo (04548004)

tel: 55+11+3045 2193

URUGUAY

Roque Graseras 857

Montevideo (11300)

tel: 598+2+7117879

USA

12105 Sundance Ct.

Reston (20194)

tel:+703 842 9455

www.hexacta.com

