

Estructura MVC

Katherine Lissette Sanchez Vila SV161855
Gerson Ernesto Lopez Cheves LC151886



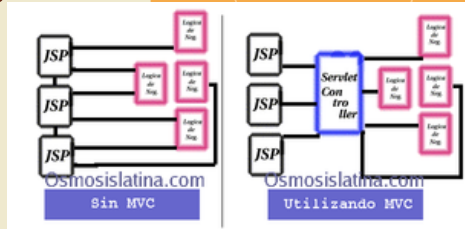
¿Que es MVC?

El MVC es un patrón de diseño arquitectónico de software, que sirve para clasificar la información, la lógica del sistema y la interfaz que se le presenta al usuario. En este tipo de arquitectura existe un sistema central o controlador que gestiona las entradas y la salida del sistema, uno o varios modelos que se encargan de buscar los datos e información necesaria y una interfaz que muestra los resultados al usuario final. Es muy usado en el desarrollo web porque al tener que interactuar varios lenguajes para crear un sitio es muy fácil generar confusión entre cada componente si estos no son separados de la forma adecuada. Este patrón permite modificar cada uno de sus componentes si necesidad de afectar a los demás.

Estructura en proyecto MVC

La estructura MVC ("Model-View-Controller") es un paradigma utilizado en diversos desarrollos de software, a través de este "Framework" se logra una división de las diferentes partes que conforman una aplicación, siendo su principal razón de ser: manutención del código fuente.

Conforme incrementan las necesidades de cualquier aplicación, la modificación a código existente se hace inminente y si no existe una clara división de uso, el código no solo se torna indecifrible sino en ocasiones impredecible debido a la mezcla de funcionalidades que pueden surgir.



A través de MVC se realiza la siguiente división :

- Model: Concentra las funcionalidades relacionadas con el Modelo de datos, esto es, el acceso y manipulación de depósitos informativos como Bases de Datos y Archivos.
- View: Se basa en el aspecto visual/gráfico que será empleado por la aplicación en cuestión.
- Controller: Empleado como un mediador entre el medio gráfico ("View") y el modelo ("Model"), coordina las acciones que son llevadas acabo entre ambos.

como realizar la configuración de rutas amigables en mvc con htaccess

Algunas URL son mejor consideradas por los motores de búsqueda tipo Google que otras. Esto ya lo habíamos comentado en nuestro manual de promoción web. Por ejemplo, URLs como estas no son muy atractivas para la promoción de las páginas:

www.dominio.com/articulos/muestra.php?id=23
www.dominio.com/pueblos/pueblo.php?nombre=torrelodones

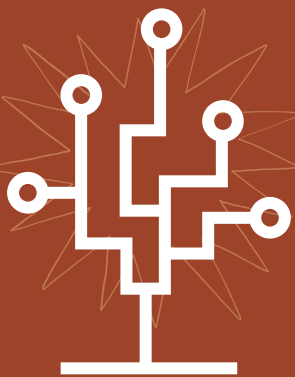
Aunque una URL con parámetros tenga contenidos específicos, los buscadores no las puntúan tan bien como si fueran los mismos contenidos con URLs sin paso de parámetros.

www.dominio.com/articulos/23.php
www.dominio.com/pueblos/torrelodones.php

Incluso, la dirección del artículo 23.php, podría ser mejor si incluyese en la propia URL alguna palabra clave, por ejemplo, si el artículo hablase sobre coches antiguos, una dirección mejor sería:

www.dominio.com/articulos/coches_antiguos.php





como hacer envios de datos a los controladores

Una parte importante de las aplicaciones web es que podamos tener persistencia a través de almacenamiento de datos en las diferentes Bases de Datos. Cuando trabajamos con la metodología MVC tenemos un aspecto que es el modelo, en este hemos de almacenar la estructura de nuestra persistencia de datos, es decir que este debe relacionarse directamente con la Base de Datos.

ViewBag

Antes de pasar a ver como se utiliza el modelo, primero debemos conocer el método ViewBag, este funciona como un empaquetador para el método ViewData de forma que podamos hacer una implementación más limpia al momento de definir los datos en el controlador.

```
public ActionResult About()
{
    ViewBag.Username = User.Identity.Username;

    ViewBag.CompanyName = "EBuy: The ASP.NET MVC Demo Site";
    ViewBag.CompanyDescription = "EBuy is the world leader in ASP.NET MVC demoing!";

    return View("About");
}
```



como hacer envios de datos a los modelos

El modelo es una entidad sumamente importante ya que es el que nos va a dar la estructura de los diferentes elementos de nuestra aplicación, por ello seguramente cuando un usuario quiera obtener datos estos vendrán de alguna estructura previamente representada por el modelo.

Si tenemos un modelo ya hecho como por ejemplo el siguiente:

```
public class CompanyInfo
{
    public string Name { get; set; }
    public string Description { get; set; }
}
```

Podemos acceder a sus propiedades haciendo una instancia en nuestro controlador de la forma que vemos en la siguiente imagen:

```
public ActionResult About()
{
    ViewBag.Username = User.Identity.Username;

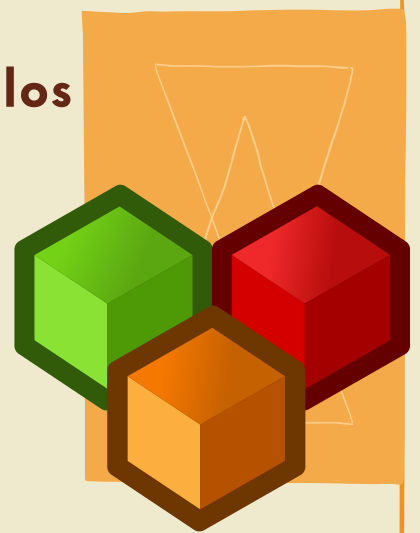
    var company = new CompanyInfo {
        Name = "EBuy: The ASP.NET MVC Demo Site",

        Description = "EBuy is the world leader in ASP.NET MVC demoing!",
    };

    return View("About", company);
}
```



La primera propiedad es bastante sencilla simplemente decimos que el nombre del usuario es el correspondiente con el sistema de autenticación del framework, luego definimos una variable la cual será una instancia de nuestro modelo y gracias a los constructores podemos hacer el establecimiento de los valores de los atributos del modelo, en este caso Name y Description por último retornamos nuestra vista.



como hacer envios de datos a los controladores y vistas

El método para pasar información de una acción de un controlador a una vista, es usando la propiedad Model. La propiedad Model no funciona como las anteriores, sino que lo que se pasa es un objeto, que se manda de la acción hacia la vista.

A diferencia de ViewData y ViewBag que existen tanto en el controlador como en la vista, el controlador no tiene una propiedad Model. En su lugar se usa una sobrecarga del método View() y se manda el objeto como parámetro:

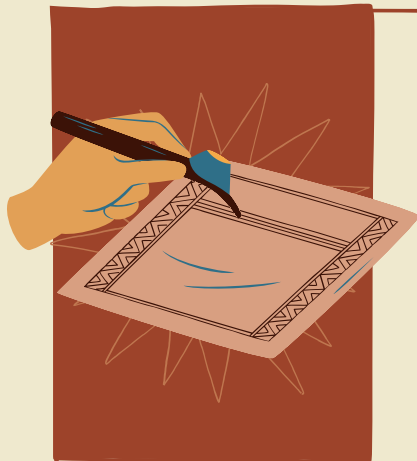
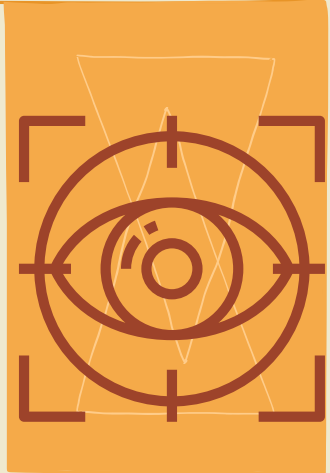
```
public class Usuario
{
    public string Nombre { get; set; }
    public string Twitter { get; set; }
    public DateTime Alta { get; set; }
}

public class HomeController : Controller
{
    public ActionResult Index3()
    {
        Usuario data = new Usuario();
        data.Nombre = "Eduard Tomàs";
        data.Twitter = "eiximenis";
        data.Alta = new DateTime(2008, 12, 10);
        return View(data);
    }
}
```

Notemos las diferencias: El controlador crea un objeto de la clase Usuario y manda ese objeto a la vista, pasándolo como parámetro a la función View. ¿Y desde la vista? Pues usamos la propiedad Model para acceder a dicho objeto:

```
@model MvcHelloWorld.Controllers.Usuario
```

```
<h2>Index3</h2>
Mi nombre es @Model.Nombre y puedes seguirme en
<a href="http://twitter.com/@Model.Twitter">Twitter</a>. <br />
Dado de alta en: @Model.Alta.ToLongDateString()
```



Como hacer envios de datos a las vistas

Ya que hemos establecido los datos en el controlador que provienen de nuestro modelo es hora de hacer que la vista muestre los datos que el usuario espera recibir luego de haber consultado el recurso.

Veamos el siguiente código:

```
@{ var company = (CompanyInfo)ViewData.Model; }
<h1>@company.Name</h1>
<div>@company.Description</div>
```


Lo primero que realizamos es igualar una variable y decirle que va a ser tipo del modelo a utilizar y adicionalmente la vinculamos con el método ViewData, todo esto en un bloque razor, luego simplemente accedemos a cada propiedad del modelo con un punto, como vemos es algo muy simple y que no lleva una complejidad mayor, esto es gracias a que el controlador hizo la mayoría del trabajo.

Finalizando el tutorial ya sabemos cómo obtener datos desde una estructura de un modelo y luego mostrarlo en la vista con la ayuda del controlador.

Aplicacion

Repositorio en Github:

<https://github.com/TheGrimReaper97/Calculo-de-notas-promedio-Online>



UNIVERSIDAD DON BOSCO
Educación Superior con Estilo Salesiano

Programa que calcule la media de tres notas

Ingresar la primera nota:

Ingresar la segunda nota:

Ingresar la tercera nota:

CALCULAR

La primera nota fue: 10
La segunda nota fue: 10
La tercera nota fue: 10
La media de las tres notas es: 10 - Excelente

volver atrás

```
File Edit Selection View Go Run Terminal Help
Promedio.html - Calculo-de-notas-promedio-Online - Visual Studio Code

EXPLORER
OPEN EDITORS 1 UNSAVED
  X Promedio.html mvc\Vistas
    JS validar.js mvc\Vistas\js
    # forms.css mvc\Vistas\css
CALCULO-DE-NOTAS-PROMEDIO...
  mvc
    Controlador
    petition.php
    Modelo
    calculo.php
    Vistas
      css
        # forms.css
        # tables.css
      img
        descarga.png
        don_bosco.jpg
        wallpaper.jpg
      js
        modernizr.custom.lis.js
        JS validar.js
        > rounded
    Promedio.html
    .gitattributes
    index.php
  > OUTLINE

mv > Vistas > Promedio.html > ...
1
2 <html>
3 <head>
4 <meta charset="utf-8" />
5 <meta name="viewport" content="width=device-width,user-scalable=no,initial-
6 scale=1.0,maximum-scale=1.0,minimum-scale=1.0" />
7 <title>Formulario de ingreso de datos</title>
8 <link rel="stylesheet" href="mvc/Vistas/css/forms.css" />
9 <script src="js/validar.js"></script>
10 <script src="js/modernizr.custom.lis.js"></script>
11 </head>
12 <body>
13 <center>
14 <br>
15 <h1> Programa que calcule la media de tres notas</h1>
16 <table>
17 <form action="mvc/Modelo/calculo.php" method="POST">
18
19 <tr><td>Ingresar la primera nota:</td><td><input type="number" min="0" max="20" name="CajaNota1"/> </t
20 <tr><td>Ingresar la segunda nota:</td><td><input type="number" min="0" max="20" name="CajaNota2"/> </t
21 <tr><td>Ingresar la tercera nota:</td><td><input type="number" min="0" max="20" name="CajaNota3"/> </t
22
23 <tr><td align="right"><input type="submit" value="Calcular"/> </td></tr>
24
25 </form>
26 </table>
27 </center>
28 </body>
29 </html>
30
31
32
33
```

```
File Edit Selection View Go Run Terminal Help
calculo.php - Calculo-de-notas-promedio-Online - Visual Studio Code

EXPLORER
OPEN EDITORS 1 UNSAVED
  X Promedio.html mvc\Vistas
    calculo.php mvc\Modelo
    # forms.css mvc\Vistas\css
CALCULO-DE-NOTAS-PROMEDIO...
  mvc
    Controlador
    petition.php
    Modelo
    calculo.php
    Vistas
      css
        # forms.css
        # tables.css
      img
        descarga.png
        don_bosco.jpg
        wallpaper.jpg
      js
        modernizr.custom.lis.js
        JS validar.js
        > rounded
    Promedio.html
    .gitattributes
    index.php
  > OUTLINE

mv > Modelo > calculo.php
1
2 <?php
3
4 $primeranota = $_POST['CajaNota1'];
5 $segundanota = $_POST['CajaNota2'];
6 $terceranota = $_POST['CajaNota3'];
7 $media = round(($primeranota+$segundanota+$terceranota)/3,0);
8 $mensaje = "";
9
10 if(($media>=0) && ($media<=3))
11 {
12     $mensaje = "Muy mal";
13 } else if (($media>=3) && ($media <=5))
14 {
15     $mensaje = "Mal";
16 }else if (($media>=6) && ($media <=7))
17 {
18     $mensaje = "Regular";
19 }else if (($media>=7) && ($media <=9))
20 {
21     $mensaje = "Bueno";
22 }else if (($media>=9) && ($media <=10))
23 {
24     $mensaje = "Excelente";
25 }
26
27 echo "La primera nota fue: <b>$primeranota</b><br>";
28 echo "La segunda nota fue: <b>$segundanota</b><br>";
29 echo "La tercera nota fue: <b>$terceranota</b><br>";
30 echo "La media de las tres notas es: <b>$media - $mensaje</b>";
31
32
33
```