ASP.NET CORE CON C#

5. LOS MODELOS

Para tener el patrón completo, incluiremos los modelos. Son clases donde encapsulamos todo el código y operaciones sobre los datos que maneje la aplicación. Son la parte más importante porque contienen toda la "lógica". Además esta es otra forma de pasar datos desde el controlador a su vista.

Ahora, al asociar la vista al controlador, indicaremos que ésta es una vista "tipada". De esta manera, le indicamos el tipo de objeto que le pasaremos y no tendremos que hacerlo manualmente

Ejemplo: Pasaremos al ejemplo del tema anterior el nombre como un objeto String

Podemos pasar valores desde una acción del controlador a su vista:

```
return View("nombreVista", objeto);
```

Si la vista es la correspondiente a la acción no es necesario pasar su nombre, basta con pasar el objeto.

En las vistas podemos incluir código de programación para recoger y mostrar resultados o datos. Para ello utilizamos

La sintaxis actual para los modelos varía un poco con respecto a la tradicional de Visual Studio para las aplicaciones Windows y las anteriores versiones de ASP.NET:

- No se declaran los campos, se declaran directamente los procedimientos de propiedad sin código.

- No se definen constructores, el objeto se instancia con dichos procedimientos de propiedad

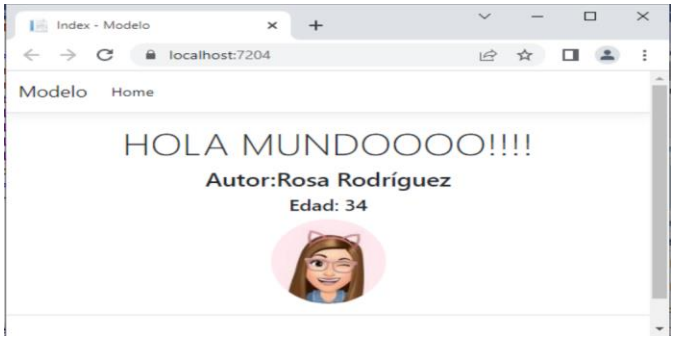
Para evitar que nos avise de que los campos por referencia no deben ser nulos, doble click en el proyecto

```
<PropertyGroup>
  <TargetFramework>net6.0</TargetFramework>
  <Nullable>enable</Nullable>
  <ImplicitUsings>enable</ImplicitUsings>
</PropertyGroup>
```

Y cambiamos la propiedad Nullable a disable

O bien le añadimos una interrogación al campo

Ejemplo: Proyecto nuevo con un modelo persona con nombre, edad e imagen



Razor

Es un motor de visualización que procesa el contenido de ASP.NET y busca instrucciones, normalmente para insertar contenido dinámico en la salida que se envía a un navegador. Permite mezclar código HTML y código de servidor sin tener que marcar donde empieza y acaba este último:

- Expresiones de línea: comienzan con el símbolo @ para escribir código c# y html

```
<h2>@DateTime.Now</h2>
```

- Para trabajar con modelos: para determinar que un objeto de ese tipo le es pasado a la vista desde una acción de un controlador

```
@model nombreSitio.Models.NombreModelo


@Model.propiedad: uso de la propiedad del objeto

@Model.propiedad.método(propiedad..)
```

- Bloques de código: los bloques comienzan con @{ y terminan con }. El código será interpretado cuándo la vista sea renderizada.

Para declarar variables

```
@{
int x=3; String nombre="Rosa";
}
```

Para mostrar texto dentro de un bloque de código

```
@{
String nombre="Rosa";
int edad=34;
if (edad>34)
<p>@nombre, eres muy joven</p>
}

Se pueden definir funciones en esos bloques
```

Bloque de código con una única instrucción

```
@if(DateTime.IsLeapYear(DateTime.Now.Year) )
{
    @DateTime.Now.Year @:is a leap year.
}
else {
    @DateTime.Now.Year @:is not a leap year.
}
```

```
@foreach (var item in Model)
{
    <tr>
        <td>@item.Nombre</td>
        <td>@item.Apellidos</td>
        <td> @item.Telefono</td>
    </tr>
}
```