

# Web sencilla usando MVC con ASP.NET y Razor

Iris Pérez Aparicio

2º DAM

Campus FP Emprende Humanes

Crear la solución en Visual Studio	1
Definir el modelo de datos (Clase C#)	1
<b>Crear la colección en el controlador (Controller)</b>	<b>2</b>
<b>Mostrar la colección en la vista (View)</b>	<b>3</b>
Ejecutar	3
<b>Flujo de MVC</b>	<b>4</b>

## Crear la solución en Visual Studio

Abre Visual Studio 2022 crea una nueva solución de tipo Aplicación web ASP.NET Core (Modelo-Vista-Controlador) con el nombre de WebMVCSencilla

## Definir el modelo de datos (Clase C#)

Crea el archivo `Estudiante.cs` en la carpeta `Models` y añade este código:

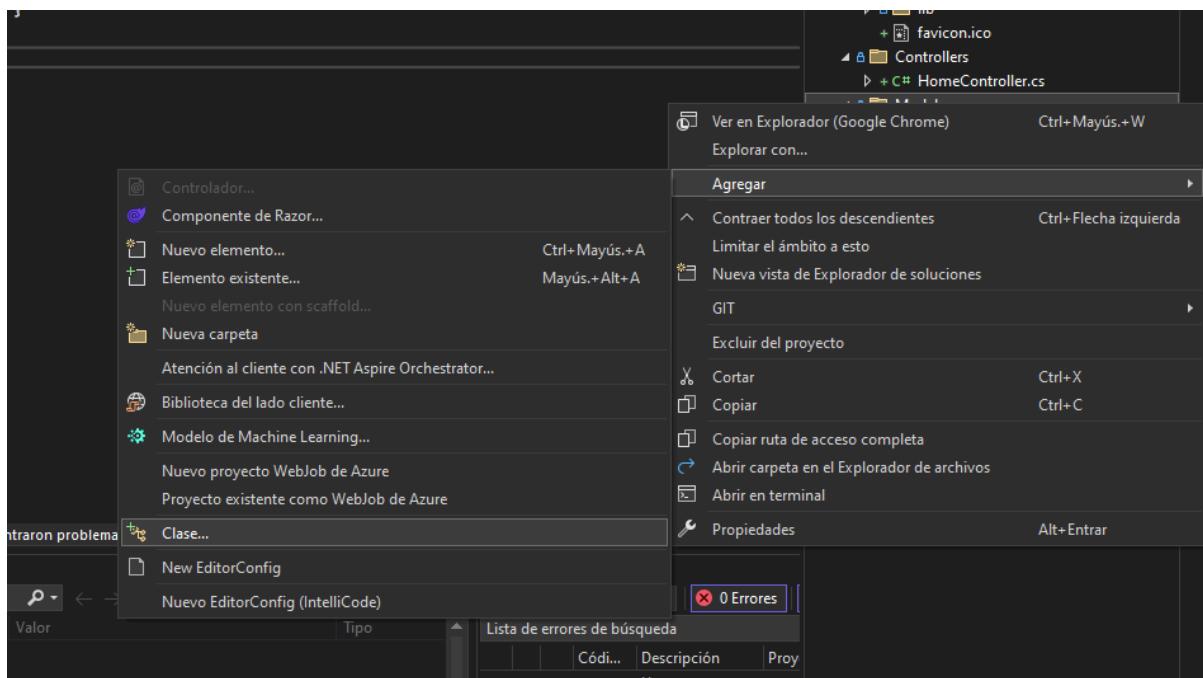
`namespace WebMVCSencilla.Models`

```
{  
    public class Estudiante  
    {  
        public int Id { get; set; }  
        public string Nombre { get; set; } = string.Empty; // Inicialización para evitar nulls  
        public string Curso { get; set; } = string.Empty;  
        public int Edad { get; set; }  
    }  
}
```

Agregamos la clase de esta manera:

Clic derecho en la carpeta `Models` > Agregar > Clase... (escribimos el nombre de la clase como nombre de archivo) `.cs`

Imagen:



## Crear la colección en el controlador (Controller)

La lógica de la aplicación se centraliza en la carpeta `Controllers`. Usaremos el controlador predeterminado, `HomeController`.

Edita el archivo `HomeController.cs` que está dentro de la carpeta `Controllers` y modifica el método `Index()` para crear la colección y pasárla a la vista:

```
// La acción Index() manejará la solicitud HTTP GET a la página de inicio.

public IActionResult Index()
{
    // 1. Crear y poblar la colección
    List<Estudiante> listaEstudiantes = new List<Estudiante>
    {
        new Estudiante { Id = 1, Nombre = "Ana García", Curso = "C# Básico", Edad = 20 },
        new Estudiante { Id = 2, Nombre = "Luis Pérez", Curso = "ASP.NET Core", Edad = 22 },
        new Estudiante { Id = 3, Nombre = "Marta Ruiz", Curso = "MVC", Edad = 21 }
    };

    // 2. Pasar la colección a la Vista ('Index.cshtml')
    // En MVC, el objeto se pasa como argumento del método View().
    return View(listaEstudiantes);
}
```

En la carpeta Controllers, archivo [HomeController.cs](#) eliminamos el método index() y lo sustituimos por este:

```
// La acción Index() manejará la solicitud HTTP GET a la página de inicio.
public IActionResult Index()
{
    // 1. Crear y poblar la colección
    List<Estudiante> listaEstudiantes = new List<Estudiante>
    {
        new Estudiante { Id = 1, Nombre = "Ana García", Curso = "C# Básico", Edad = 20 },
        new Estudiante { Id = 2, Nombre = "Luis Pérez", Curso = "ASP.NET Core", Edad = 22 },
        new Estudiante { Id = 3, Nombre = "Marta Ruiz", Curso = "MVC", Edad = 21 }
    };

    // 2. Pasar la colección a la Vista (`Index.cshtml`)
    // En MVC, el objeto se pasa como argumento del método View().
    return View(listaEstudiantes);
}
```

## Mostrar la colección en la vista (View)

Eliminamos todo el código de la vista y añadimos este:

Edita el archivo `Index.cshtml` que está dentro de la carpeta `Views/Home` y reemplaza su código por el siguiente:

```
@using WebMVCsencilla.Models

@model List<Estudiante> // ¡Define el tipo de colección que espera la vista!

@{
    ViewData["Title"] = "Listado de Estudiantes MVC";
}

<h1>@ViewData["Title"]</h1>
<p>
    Esta es una lista de estudiantes gestionada por el Controlador y pasada a la Vista.
</p>

@if (Model != null && Model.Any())
{
    <table class="table table-striped">
        <thead>
            <tr>
                <th>ID</th>
                <th>Nombre</th>
            </tr>
        </thead>
        <tbody>
            @foreach (var estudiante in Model)
            {
                <tr>
                    <td>@estudiante.Id</td>
                    <td>@estudiante.Nombre</td>
                </tr>
            }
        </tbody>
    </table>
}
```

```

<th>Curso</th>
<th>Edad</th>
</tr>
</thead>
<tbody>
@foreach (var estudiante in Model)
{
    <tr>
        <td>@estudiante.Id</td>
        <td><strong>@estudiante.Nombre</strong></td>
        <td>@estudiante.Curso</td>
        <td>@estudiante.Edad</td>
    </tr>
}
</tbody>
</table>
}
else
{
    <p class="alert alert-warning">No hay estudiantes en la colección para mostrar.</p>
}

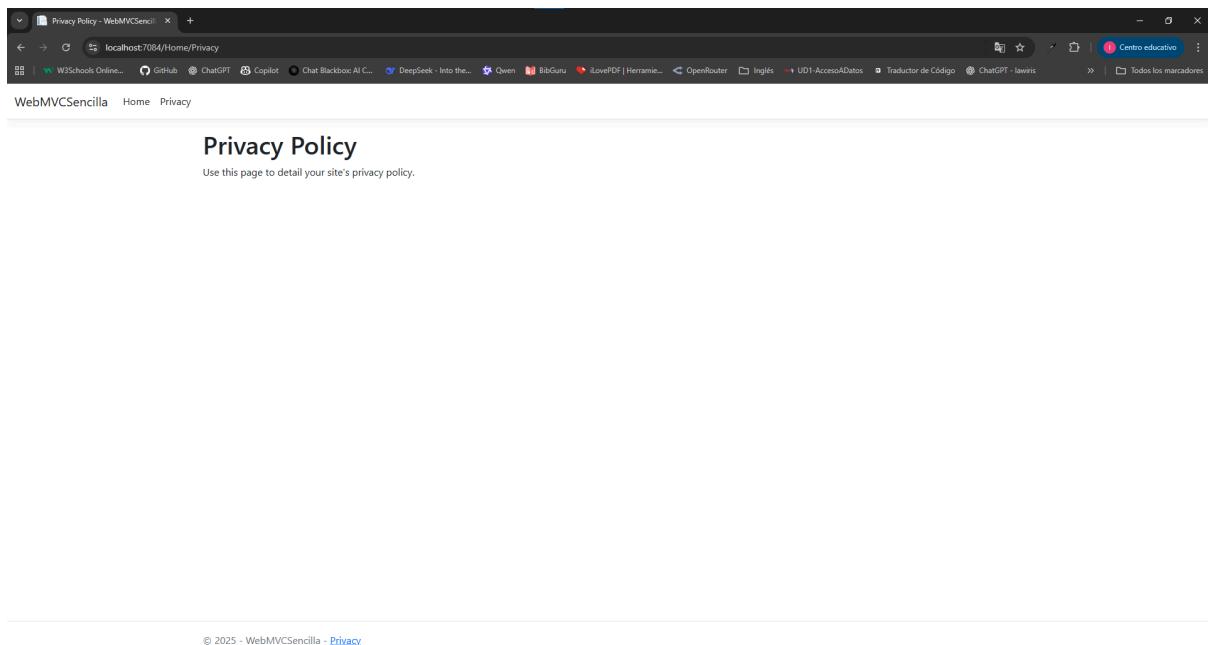
```

## Ejecutar

Guarda todos los archivos y presiona F5 para ejecutar la aplicación.

Al ejecutar la aplicación, podemos ver la página web con las siguientes secciones:

ID	Nombre	Curso	Edad
1	Ana Garcia	C# Básico	20
2	Luis Pérez	ASP.NET Core	22
3	Marta Ruiz	MVC	21



## Flujo de MVC

1. Solicitud: El usuario pide la URL `/Home/Index`.
2. Controlador: El `HomeController` recibe la solicitud.
3. Acción: El método `Index()` ejecuta la lógica (crea la colección de `Estudiante`).
4. Transferencia de Datos: El controlador pasa la colección al método `return View(listaEstudiantes);`
5. Vista: La vista `Index.cshtml` recibe la colección (gracias a la directiva `@model List<Estudiante>`) y usa la sintaxis Razor para generar el HTML final.
6. Respuesta: El HTML final se envía al navegador del usuario.

Estructura del proyecto:

