

# Programación Web .NET Core

Módulo 3

# **ASP.NET MVC Framework**

## **Controlador, Vista, Modelo**

# Controlador - Vista - Modelo

En el primer acercamiento a este patrón, se resolvió toda la lógica de la aplicación solamente en el **Controlador**. En la segunda etapa, se separaron las acciones y se agruparon funcionalidades en el **Controlador** y la **Vista**.

En esta tercera etapa, se implementará otro escenario, pero se abordarán los tres elementos fundamentales de este patrón, los cuales son: **Modelo**, **Vista** y **Controlador**.

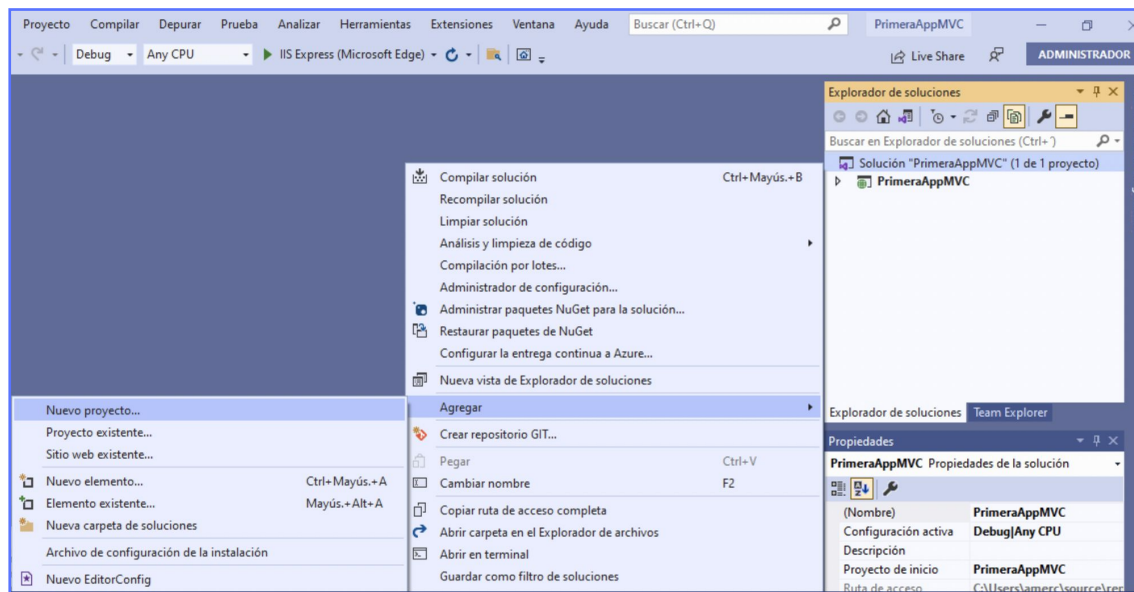
## Escenario

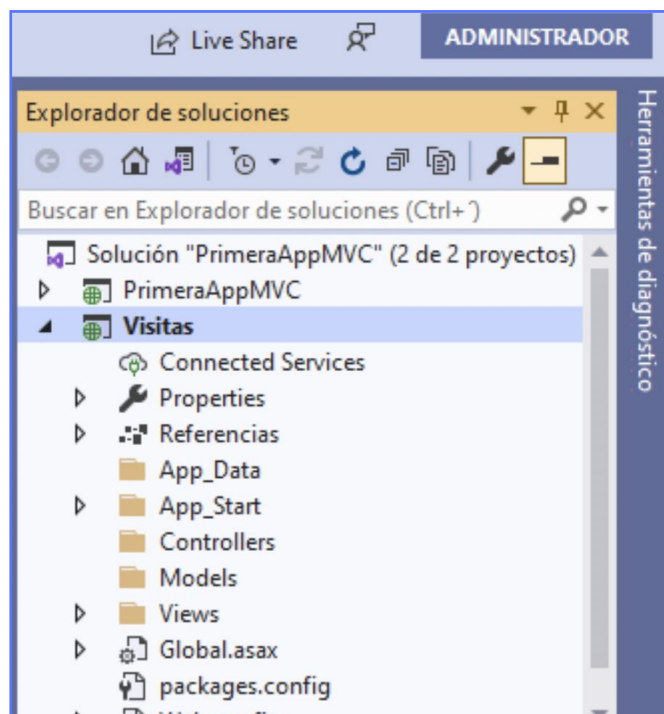
Desarrollar un libro de visitas a un sitio web. El sitio contendrá un formulario que permitirá ingresar el nombre del visitante y dejar comentarios. Los datos se almacenarán en un archivo de texto.

La página principal deberá tener dos vínculos. El primero, llevará al formulario para la carga del nombre del visitante y sus comentarios; y el segundo, deberá mostrar todos los comentarios que dejaron los visitantes.

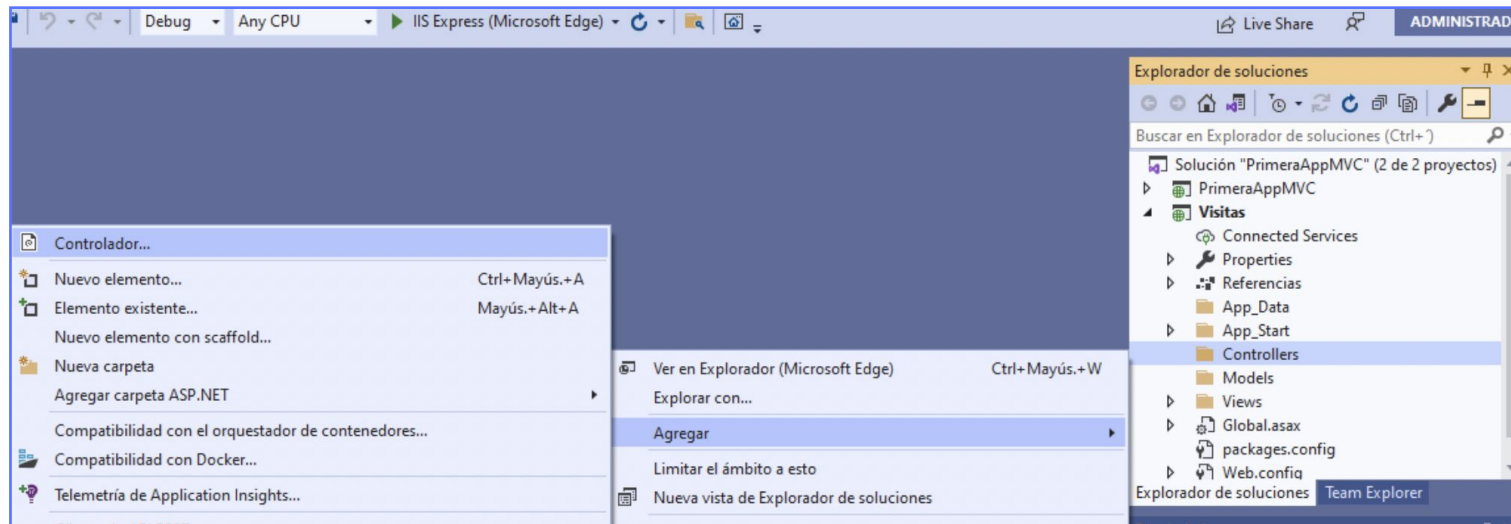


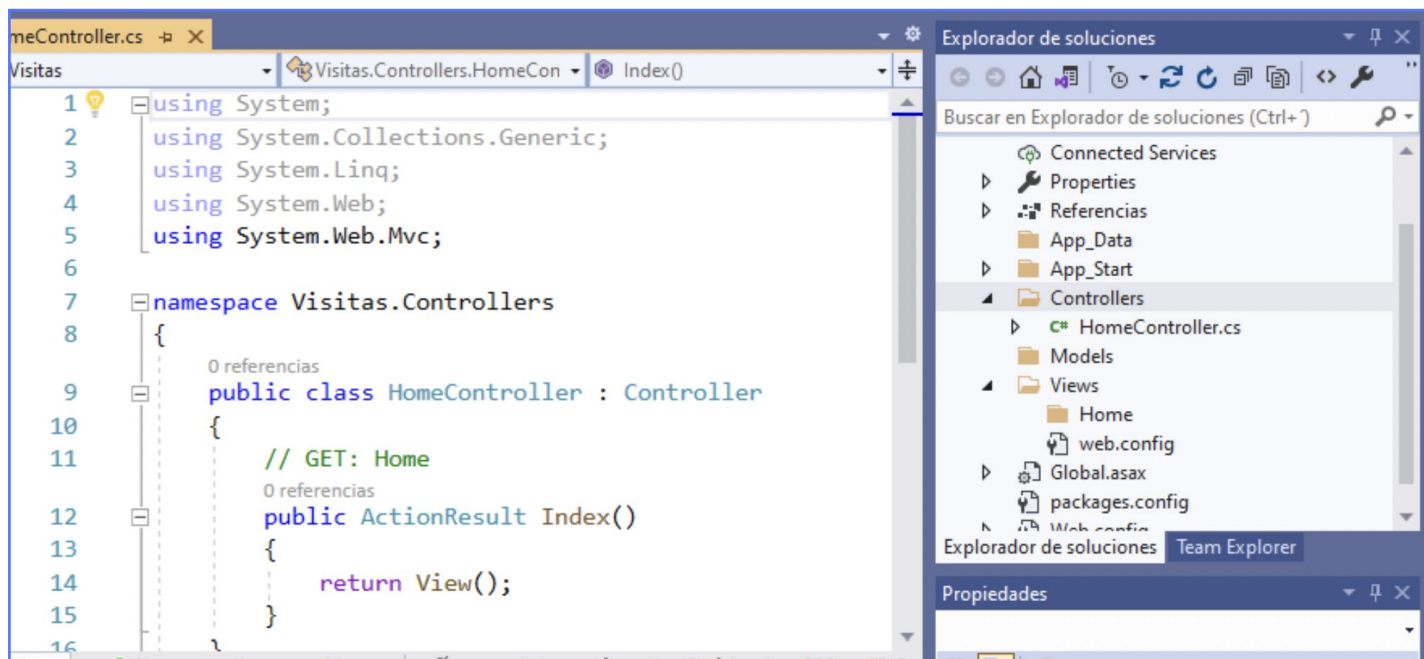
1. Podemos agregar un nuevo proyecto o crear una nueva solución aparte. En este caso agregaremos un nuevo proyecto a la solución existente.



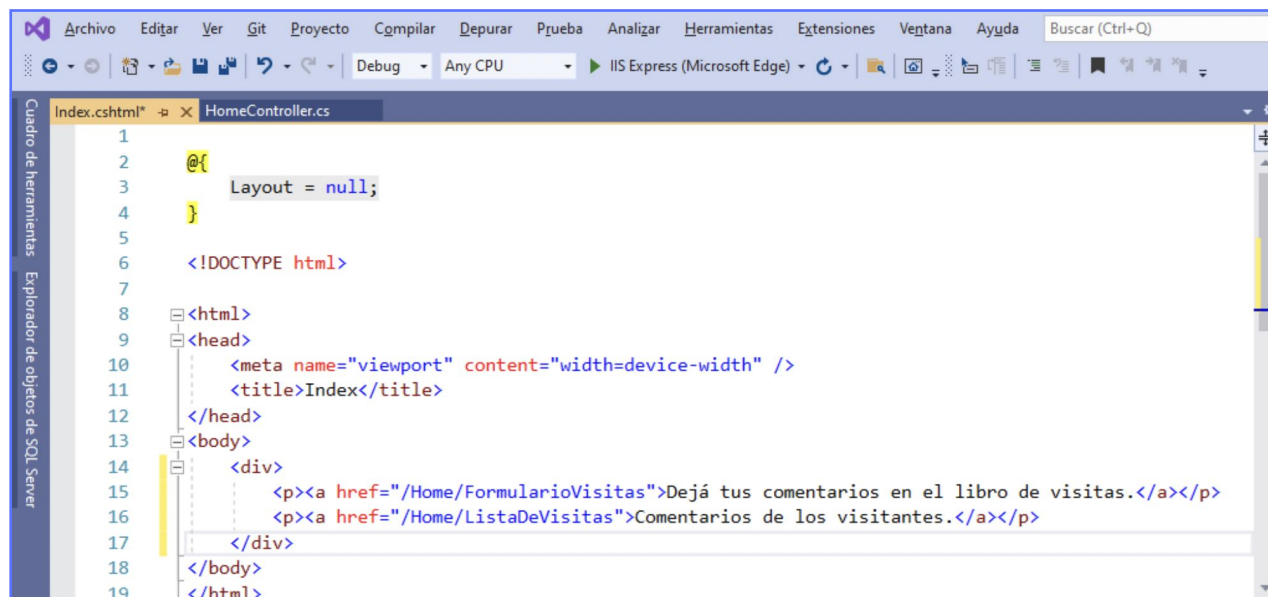


2. A continuación, se creará el **Controlador** tal como se hizo en los ejemplos anteriores.





3. Ahora generamos la vista para la página principal del sitio:

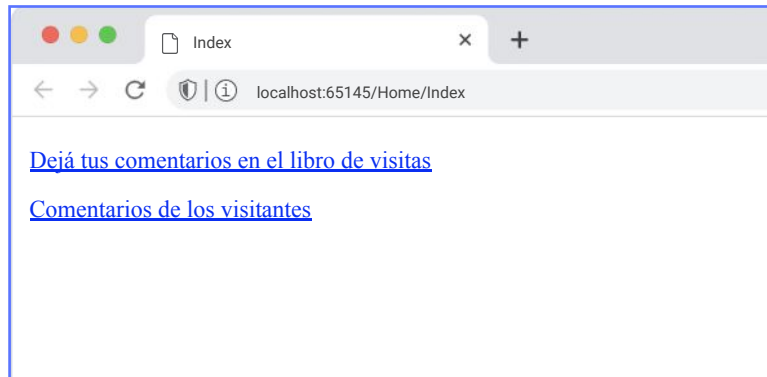


The screenshot shows the Visual Studio IDE with the 'Index.cshtml' file open in the editor. The file is part of the 'HomeController.cs' project. The code is as follows:

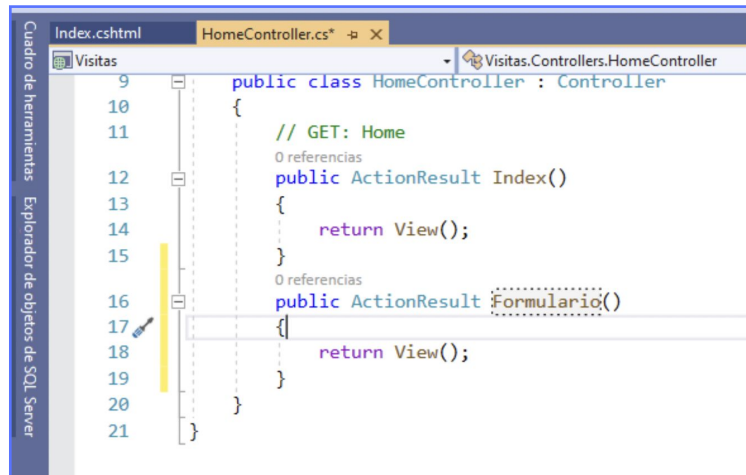
```
1 @{
2     Layout = null;
3 }
4
5 <!DOCTYPE html>
6
7 <html>
8 <head>
9     <meta name="viewport" content="width=device-width" />
10    <title>Index</title>
11 </head>
12 <body>
13     <div>
14         <p><a href="/Home/FormularioVisitas">Dejá tus comentarios en el libro de visitas.</a></p>
15         <p><a href="/Home/ListaDeVisitas">Comentarios de los visitantes.</a></p>
16     </div>
17 </body>
18 </html>
```



Si ejecutamos la aplicación podremos observar:



4. Crearemos ahora, la vista que muestre el formulario donde el visitante dejará su nombre y sus comentarios. Para eso, modifica el archivo **HomeController** y agrega otro método:



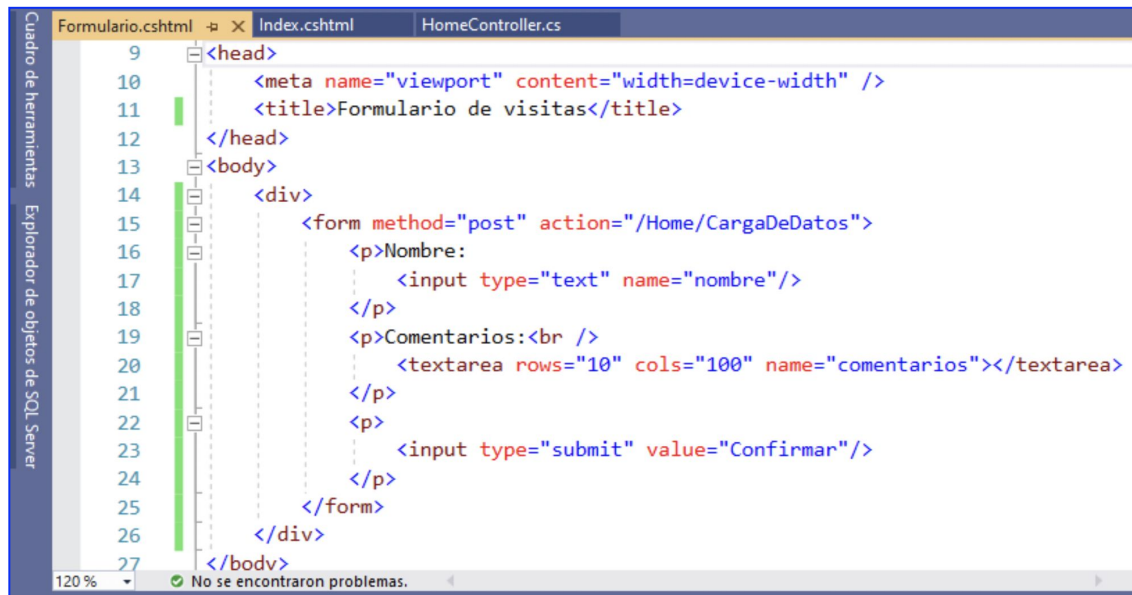
- Definimos el método **Formulario** que se ejecutará cuando el usuario ingrese la URL:  
**`http://localhost:puerto/Home/Formulario`**

Esto sucederá cuando el visitante haga clic en el primer enlace de la vista ***Index.cshtml***.

- Se debe crear la vista asociada al método **Formulario()**. En el **Explorador de soluciones**, se podrá observar que se creó un archivo llamado ***Formulario.cshtml*** en la carpeta **Home** de la carpeta **Views**.



7. Modificaremos el contenido de este archivo para que muestre el formulario HTML que permite ingresar el nombre del visitante y sus comentarios:



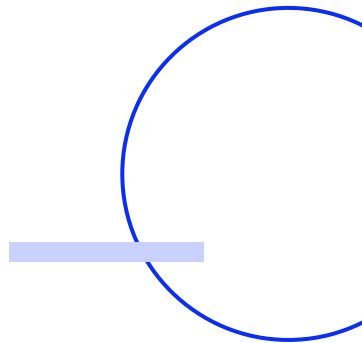
```
9 <head>
10 <meta name="viewport" content="width=device-width" />
11 <title>Formulario de visitas</title>
12 </head>
13 <body>
14 <div>
15 <form method="post" action="/Home/CargaDeDatos">
16 <p>Nombre:
17 <input type="text" name="nombre"/>
18 </p>
19 <p>Comentarios:<br />
20 <textarea rows="10" cols="100" name="comentarios"></textarea>
21 </p>
22 <p>
23 <input type="submit" value="Confirmar"/>
24 </p>
25 </form>
26 </div>
27 </body>
```

8. Codificamos un formulario HTML con un control **input** de tipo **text** y un control de tipo **textarea**. Además del botón de tipo **submit** que enviará los datos al servidor.

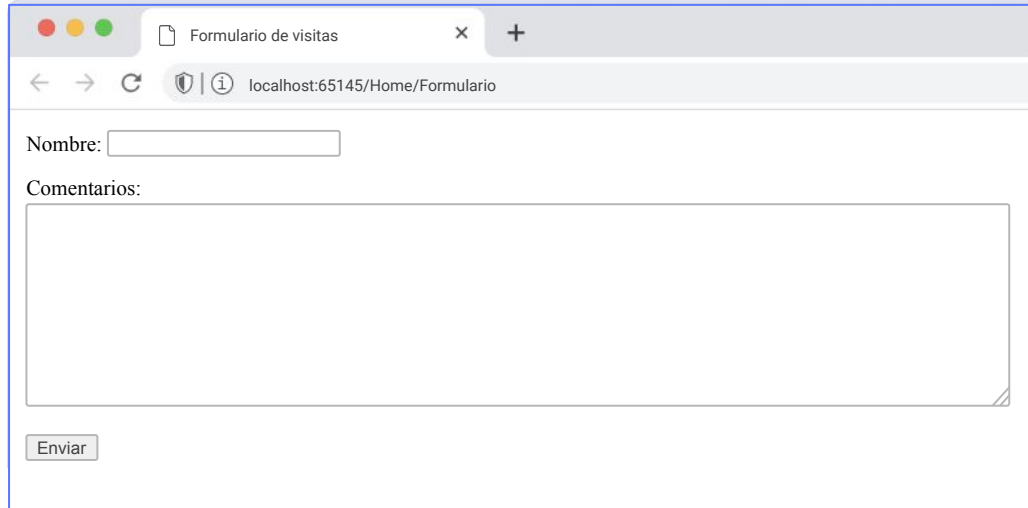
La propiedad **action** del elemento **form** indica qué URL de nuestro controlador, deberá procesar el envío de los datos del formulario que carga el usuario:

```
<formmethod="post" action="/Home/CargaDeDatos">
```

Esto significa que en el controlador llamado **HomeController** se debe definir un método cuyo nombre es **CargaDeDatos()**.



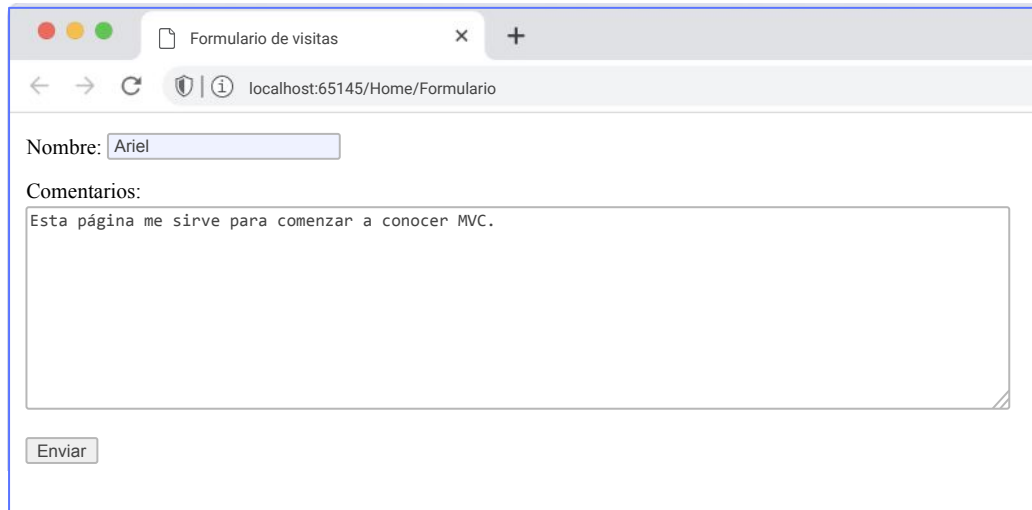
Si se ejecuta la aplicación, se puede ver el formulario:



A screenshot of a web browser window. The title bar shows a single tab labeled "Formulario de visitas". The address bar displays the URL "localhost:65145/Home/Formulario". The page content includes a label "Nombre:" followed by a text input field. Below that is a label "Comentarios:" followed by a large text area. At the bottom left of the form is a button labeled "Enviar".

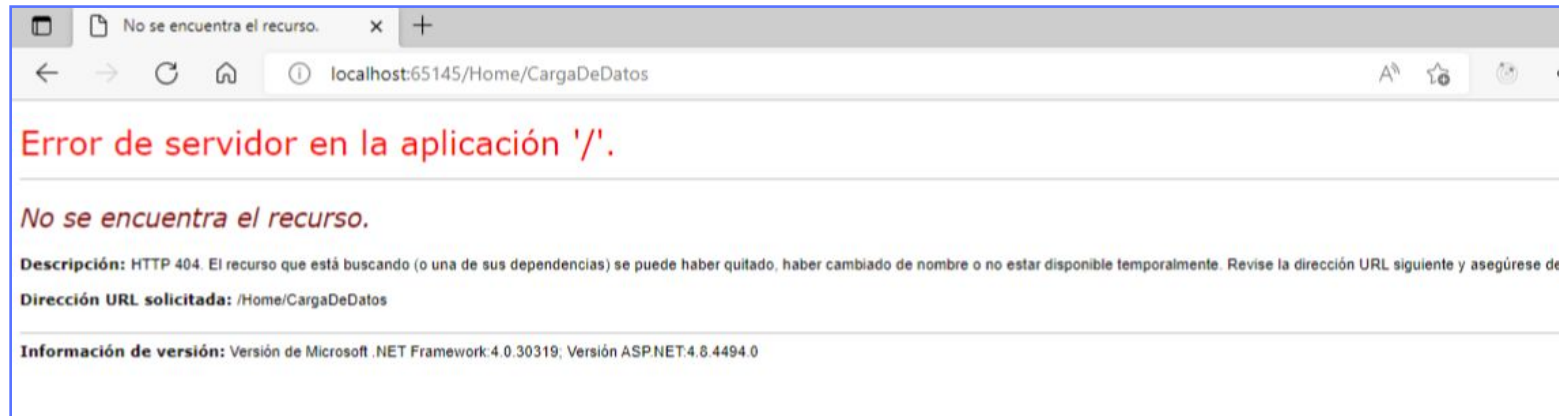


9. Se puede cargar un nombre de visitante y un comentario,

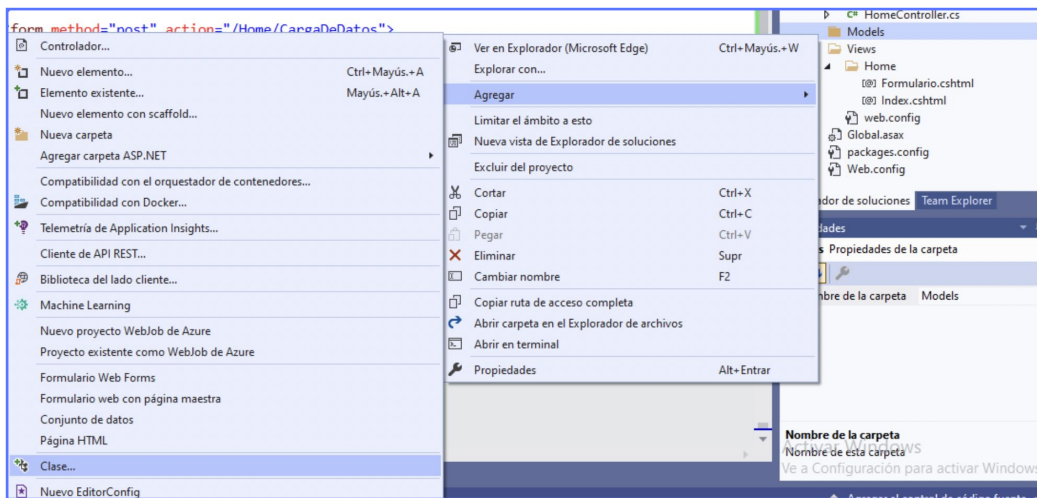


A screenshot of a web browser window. The title bar shows a single tab labeled "Formulario de visitas". The address bar displays "localhost:65145/Home/Formulario". The page content includes a label "Nombre:" followed by a text input field containing the name "Ariel". Below this is a label "Comentarios:" followed by a large text area containing the text "Esta página me sirve para comenzar a conocer MVC.". At the bottom left of the form is a button labeled "Enviar".

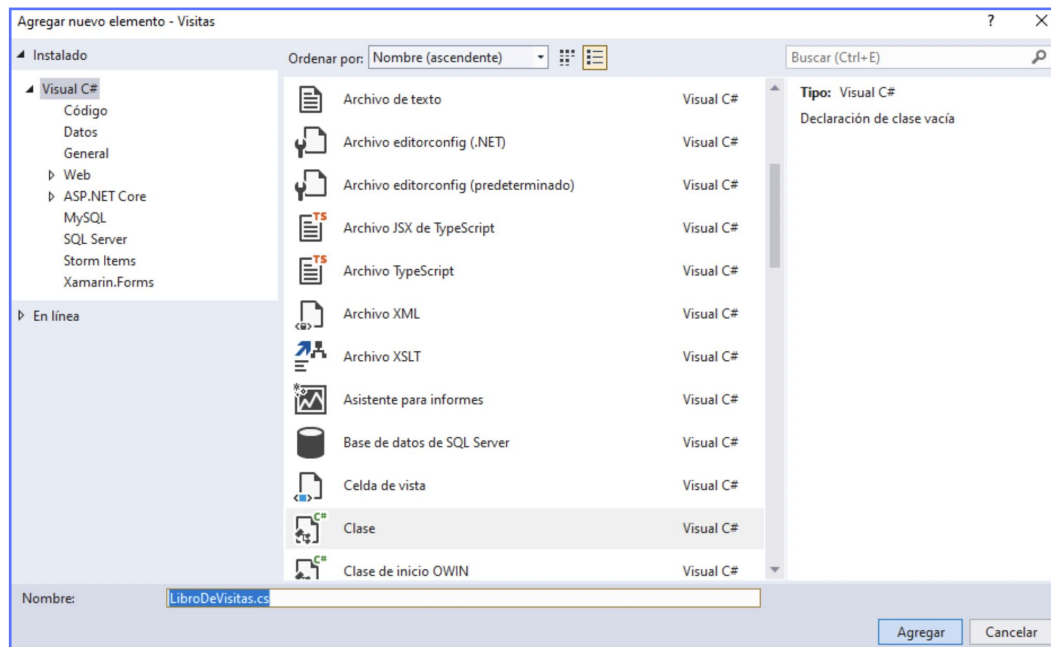
10. Pero cuando se presiona el botón **Enviar**, muestra el error 404 que devuelve el servidor:



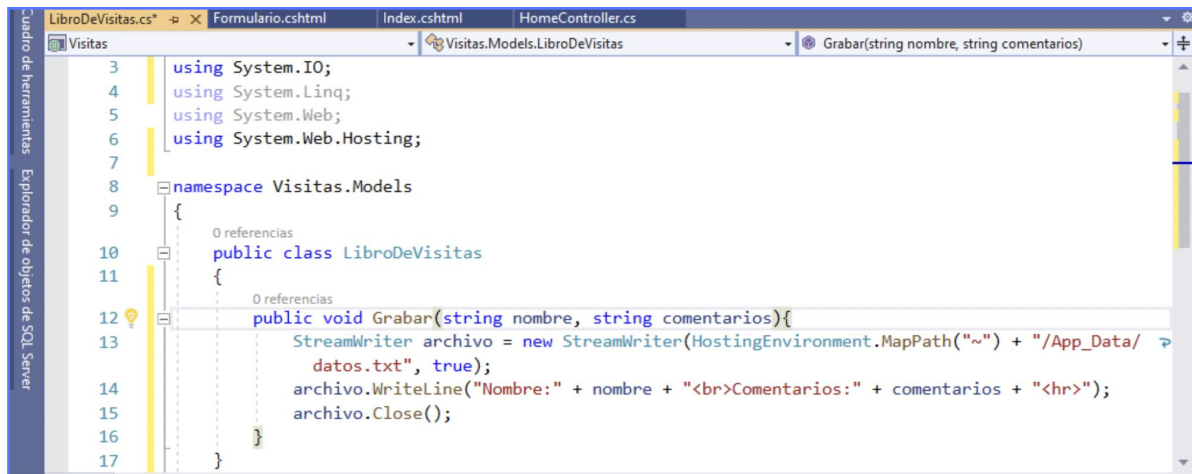
11. A continuación, crearemos el modelo que tiene como responsabilidad almacenar los datos del visitante y sus comentarios. Hacemos clic con el botón derecho del mouse sobre la carpeta **Models** y seleccionamos la opción **Agregar -> Clase...**:





12. Crearemos la clase **LibroDeVisitas.cs**:

13. Se codificará la clase para permitir guardar los datos en un archivo de texto:



The screenshot shows the Visual Studio IDE with the following details:

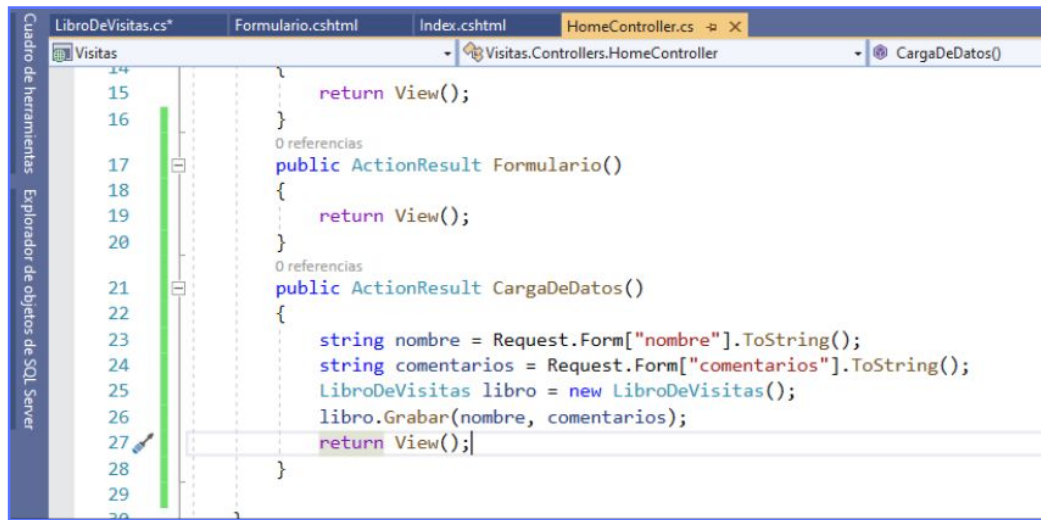
- Top Tab Bar:** LibroDeVisitas.cs\*, Formulario.cshtml, Index.cshtml, HomeController.cs
- Left Explorer:** Visitas (selected), Visitas.Models.LibroDeVisitas
- Code Editor:** Contains the implementation of the `LibroDeVisitas` class with the `Grabar` method.

```
3 using System.IO;
4 using System.Linq;
5 using System.Web;
6 using System.Web.Hosting;
7
8 namespace Visitas.Models
9 {
10     0 referencias
11     public class LibroDeVisitas
12     {
13         0 referencias
14         public void Grabar(string nombre, string comentarios){
15             StreamWriter archivo = new StreamWriter(HostingEnvironment.MapPath("~/") + "/App_Data/
16             datos.txt", true);
17             archivo.WriteLine("Nombre:" + nombre + "<br>Comentarios:" + comentarios + "<br>");
18             archivo.Close();
19         }
20     }
```

14. El método **Grabar()** recibirá dos argumentos de tipo **string** y dentro de él, se creará un objeto de la clase **StreamWriter** que se encuentra en el espacio de nombres **System.IO** (Debemos invocar el **using** correspondiente.).
15. Al constructor de la clase **StreamWriter**, le indicaremos el **path** o ruta donde se almacenará el archivo de texto llamado **datos.txt**. Utilizamos el método **MapPath()** del objeto **HostingEnvironment** que nos devolverá la ruta donde se almacena nuestro proyecto. Le concatenaremos la carpeta **App\_Data** donde se almacenará el archivo de texto.
16. En el segundo parámetro de **MapPath()** al pasarle un valor **true** indicamos que si ya existe el archivo, lo abrirá para añadir datos al final.
17. Mediante el método **WriteLine()** procederemos a grabar en el archivo de texto los parámetros **nombre** y **comentarios**.
18. Finalmente cerraremos el archivo llamando al método denominado **Close()**.



19. Ahora crearemos otra acción en el archivo llamado **HomeController** donde llamaremos al modelo para que efectúe la carga de datos. Abriremos el archivo **HomeController** y codificaremos:



```
LibroDeVisitas.cs*  Formulario.cshtml  Index.cshtml  HomeController.cs X
Visitas
14
15     return View();
16 }
17
18 public ActionResult Formulario()
19 {
20     return View();
21 }
22
23 public ActionResult CargaDeDatos()
24 {
25     string nombre = Request.Form["nombre"].ToString();
26     string comentarios = Request.Form["comentarios"].ToString();
27     LibroDeVisitas libro = new LibroDeVisitas();
28     libro.Grabar(nombre, comentarios);
29     return View();
30 }
```

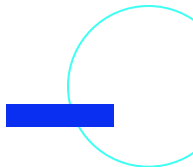
La acción **CargaDeDatos** se ejecutará cuando el visitante envía los datos del formulario de visitas.

20. Recuperaremos los datos ingresados por el visitante en los dos controles HTML:

```
string nombre = Request.Form["nombre"].ToString();  
string comentarios = Request.Form["comentarios"].ToString();
```

21. A continuación crearemos un objeto de la clase **LibroDeVisitas** y llamaremos al método **Grabar()**:

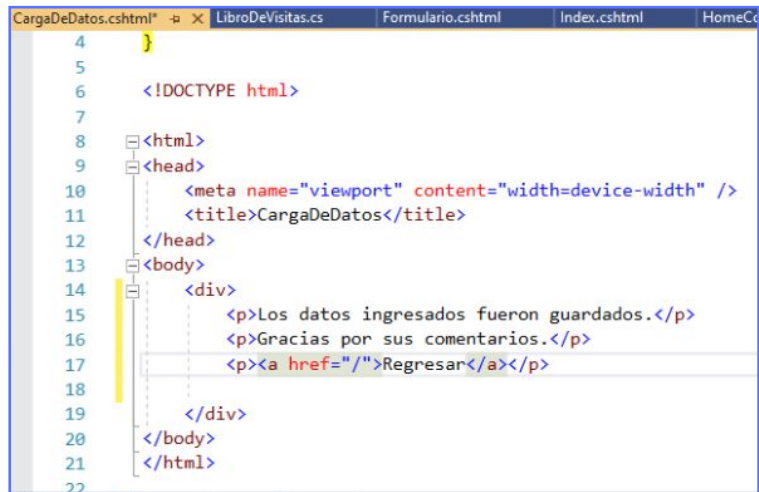
```
LibroDeVisitaslibro = new LibroDeVisitas();  
libro.Grabar(nombre, comentarios);
```



De esta manera el **Controlador** se comunicará con el **Modelo**, en este caso para solicitar que guarde los datos ingresados en el formulario.

Ahora falta crear una vista para la acción llamada **CargaDeDatos**.

22. Haremos clic con el botón derecho del mouse sobre el nombre del método **CargaDeDatos()** y seleccionaremos **Agregar vista**; luego de este paso, codificaremos el archivo cuyo nombre es **CargaDeDatos.cshtml**:



```
4      }
5
6      <!DOCTYPE html>
7
8      <html>
9      <head>
10         <meta name="viewport" content="width=device-width" />
11         <title>CargaDeDatos</title>
12     </head>
13     <body>
14         <div>
15             <p>Los datos ingresados fueron guardados.</p>
16             <p>Gracias por sus comentarios.</p>
17             <p><a href="/">Regresar</a></p>
18         </div>
19     </body>
20 </html>
21
22
```



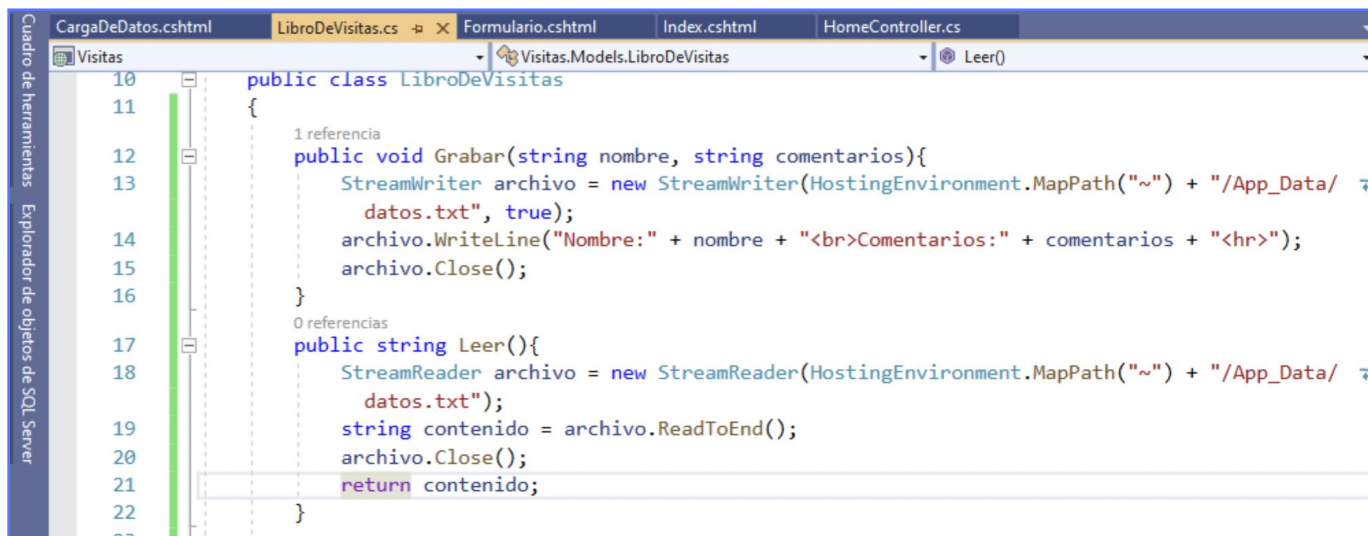
23. Seleccionemos nuevamente el archivo ***Index.cshtml*** y ejecutemos el proyecto. Cargamos los datos en el formulario y presionamos el botón ***Enviar***:



El navegador muestra la vista, pero también en el servidor se guardaron los datos ingresados en el formulario HTML. Si examinamos el contenido de la carpeta ***App\_Data*** podemos observar que se creó el archivo ***datos.txt***, y que se van agregando los comentarios y el nombre del usuario/visitante que los envía.

Nos queda resolver la página que muestre los comentarios dejados por los visitantes. Asignaremos la responsabilidad de leer el archivo de texto que le corresponde al ***Modelo***.

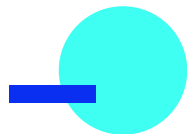
24. En la clase **LibroDeVisitas.cs** codificaremos el método **Leer**:



```
10 public class LibroDeVisitas
11 {
12     1 referencia
13     public void Grabar(string nombre, string comentarios){
14         StreamWriter archivo = new StreamWriter(HostingEnvironment.MapPath("~/") + "/App_Data/
15         datos.txt", true);
16         archivo.WriteLine("Nombre:" + nombre + "<br>Comentarios:" + comentarios + "<hr>");
17         archivo.Close();
18     }
19     0 referencias
20     public string Leer(){
21         StreamReader archivo = new StreamReader(HostingEnvironment.MapPath("~/") + "/App_Data/
22         datos.txt");
23         string contenido = archivo.ReadToEnd();
24         archivo.Close();
25         return contenido;
26     }
27 }
```



25. El método **Leer()** crea un objeto de la clase **StreamReader**. Le indicaremos el **path** donde se encuentra el archivo a leer, **datos.txt**.
26. Mediante el método **ReadToEnd()** recuperaremos todos los caracteres guardados en el archivo de texto y luego de cerrar el archivo, retornaremos un **string** teniendo todos los caracteres leídos.
27. Ahora agregaremos en el **Controlador** un método que responda al segundo vínculo de la página principal del sitio, cuyo nombre es **ListadoDeVisitas**. Pasemos a la pantalla siguiente para poder ver la imagen.





```
public ActionResult CargaDeDatos()
{
    string nombre = Request.Form["nombre"].ToString();
    string comentarios = Request.Form["comentarios"].ToString();
    LibroDeVisitas libro = new LibroDeVisitas();
    libro.Grabar(nombre, comentarios);
    return View();
}

0 referencias
public ActionResult ListadoDeVisitas()
{
    LibroDeVisitas libro = new LibroDeVisitas();
    string contenido = libro.Leer();
    ViewData["libro"] = contenido;
    return View();
}
```

28. En el método **ListadoDeVisitas()** crearemos un objeto de la clase **LibroDeVisitas** y recuperaremos el contenido almacenado llamando al método **Leer**, como vemos a continuación:

```
LibroDeVisitaslibro = new LibroDeVisitas();  
string contenido = libro.Leer();
```

29. Utilizando el diccionario **ViewData** podremos pasar datos a la vista, crearemos una entrada en el diccionario y le asignaremos el **string** recuperado desde el **Modelo**:

```
ViewData["libro"] = contenido;  
return View();
```

30. A continuación, crearemos la **Vista** para el método **ListadoDeVisitas**.
31. Codificaremos la vista cuyo nombre es **ListadoDeVisitas.cshtml** para que se muestre todos los comentarios dejados en el sitio web. Veamos la imagen de la próxima pantalla.





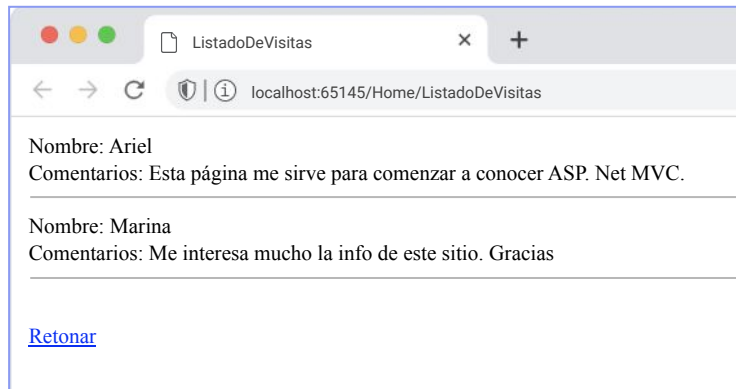
```
4      }
5
6      <!DOCTYPE html>
7
8      <html>
9      <head>
10         <meta name="viewport" content="width=device-width" />
11         <title>ListadoDeVisitas</title>
12     </head>
13     <body>
14         <div>
15             @Html.Raw(ViewData["libro"])
16             <br />
17             <a href="/">Retornar</a>
18         </div>
19     </body>
20 </html>
```

32. Ahora, utilizando **Razor** podemos recuperar los datos almacenados en el diccionario **ViewData** que cargamos en el **Controlador**:

```
<div>
    @Html.Raw(ViewData["libro"])
<br />
<a href="/">Retonar</a>
</div>
```

Debemos utilizar el método **Raw** de HTML para su correcta visualización.

El resultado de esta vista es:





**Nota:** En este momento, tenemos un sitio web que permite dejar comentarios y ver los comentarios de otros visitantes utilizando el patrón

**Modelo-Vista-Controlador.**

En un escenario más cercano a la realidad, los datos se guardarían en una base de datos. Esto lo abordaremos en los siguientes escenarios.

**¡Sigamos  
trabajando!**

