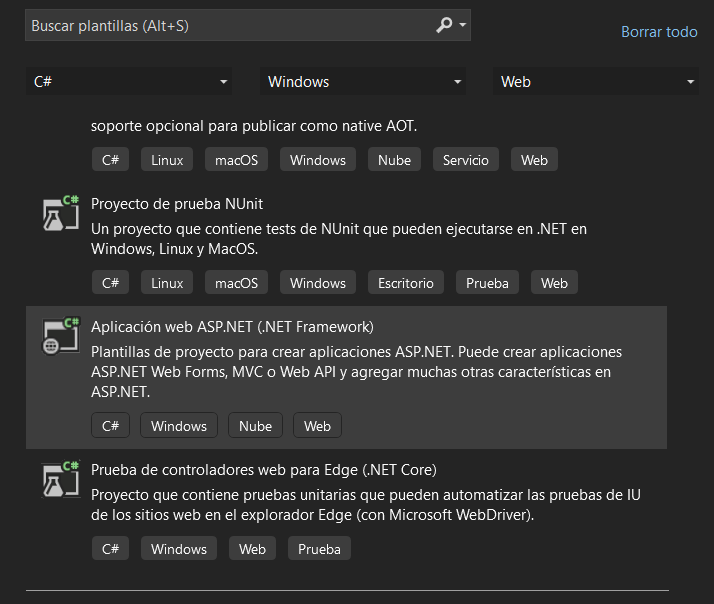
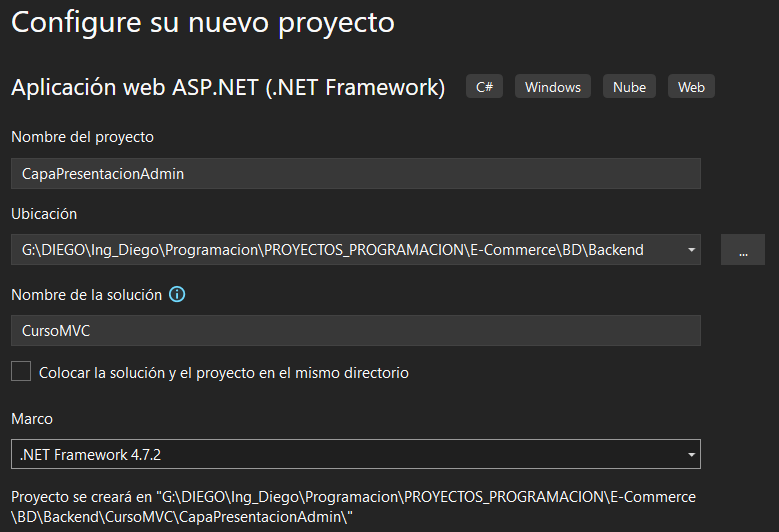
**CREACIÓN DEL BACKEND**

En C# creamos un nuevo proyecto de tipo Aplicación web ASP.NET (.NET Framework), elegimos la ruta donde se alojará, y en el marco elegimos uno que sea igual o superior al 4.5. Nombramos a la Solución como: “CursoMVC”.



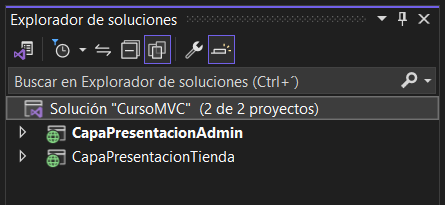




Sobre Solución “CursoMVC” (el nombre que le dimos a la solución), clic derecho, agregar, Nuevo proyecto y lo llamaremos “CapaPresentacionTienda” y elegimos las mismas configuraciones que en la hoja anterior.

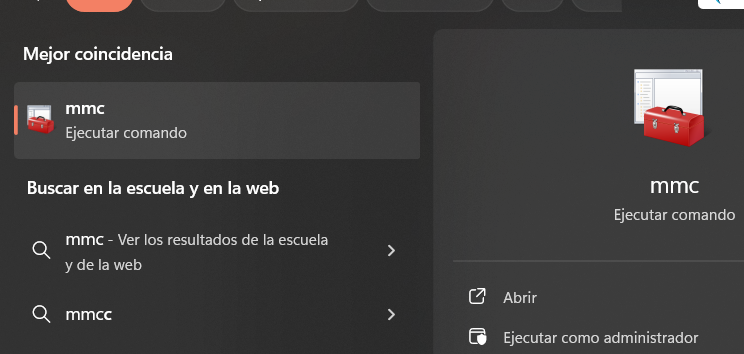
Se crean dos porque una es la perspectiva (modulo) desde el lado del admin y la otra desde el lado del cliente.

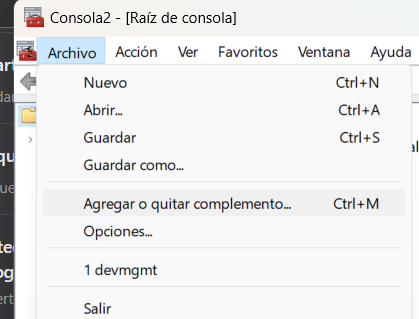
Debe quedar así:

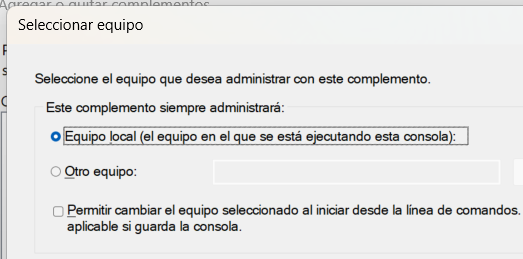
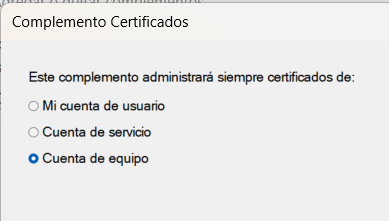


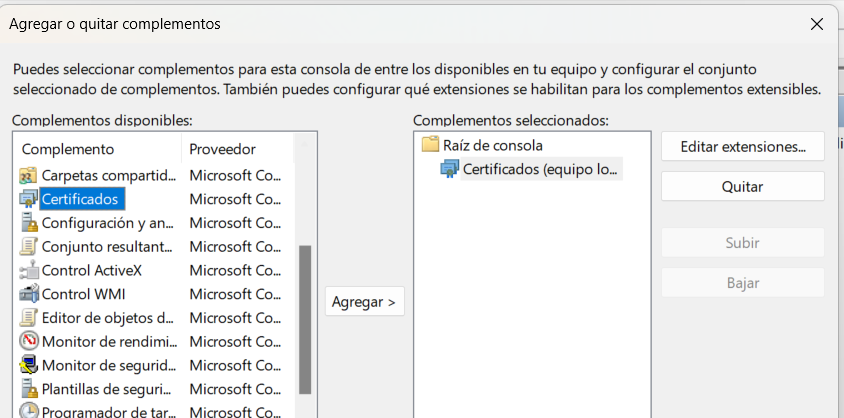
Para eliminar las restricciones SSL que nos puede presentar el equipo al ejecutar el aplicativo debemos:

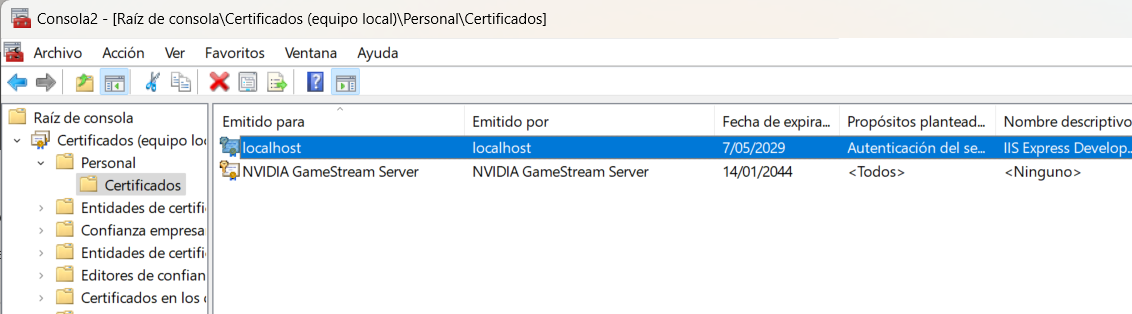
Clic en buscar, escribimos mmc, archivo, agregar o quitar complemento, Certificados, Agregar, aceptar, cuenta de equipo, siguiente, equipo local, finalizar, aceptar. Este se encontrará ahora dentro de Raiz de consola, Certificados, Personal, Certificados.





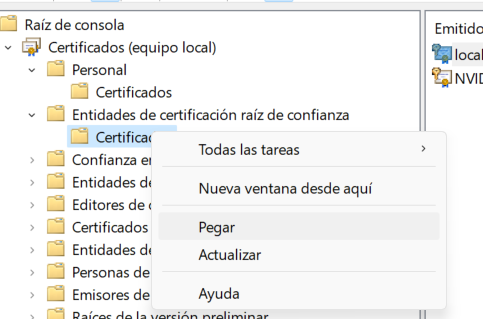




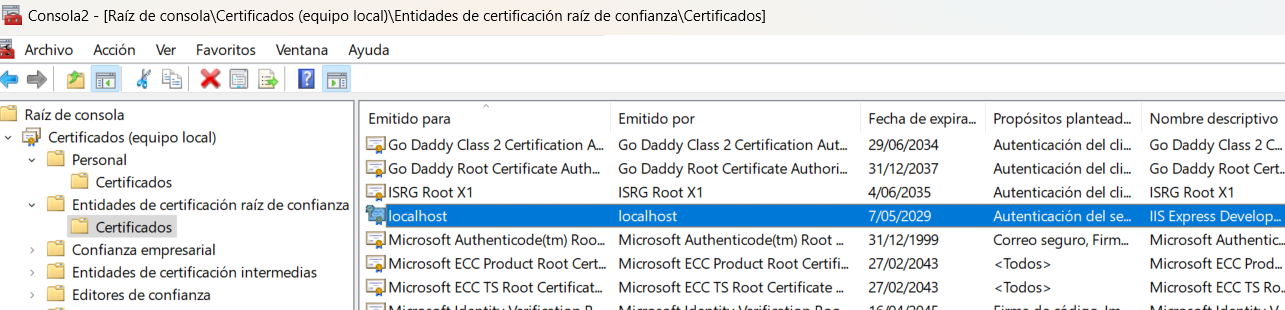


Como se puede ver tiene una fecha de expiración, para que eso no sea así debemos:

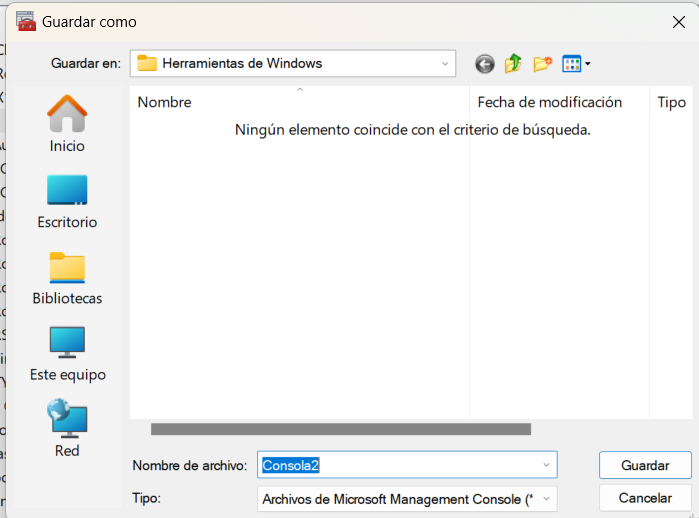
Lo copiamos el que acabamos de crear “localhost”, y dentro de Trusted Root Certificates o Entidades de Certificación Raíz de Confianza, lo pegamos:



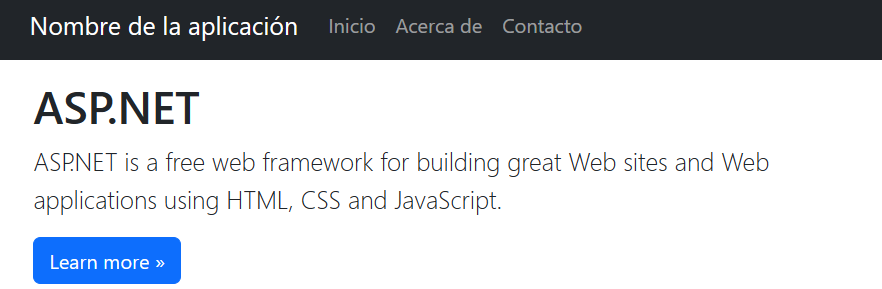
Y quedará así:



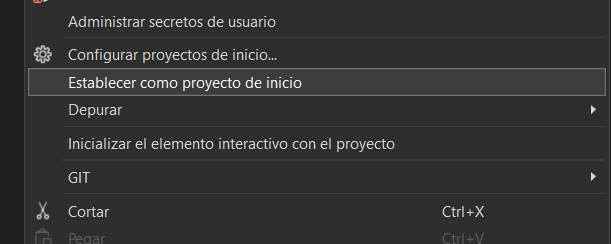
Guardamos y cerramos:

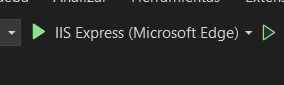


Y nos debe salir esto al ejecutar el aplicativo:

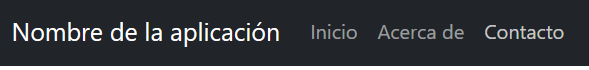


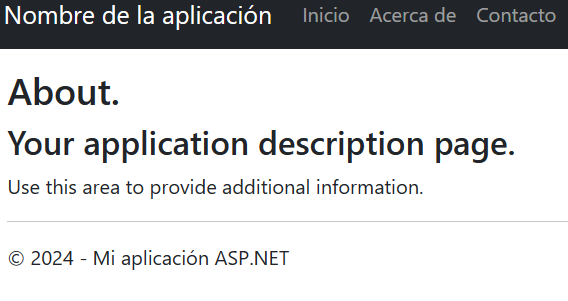
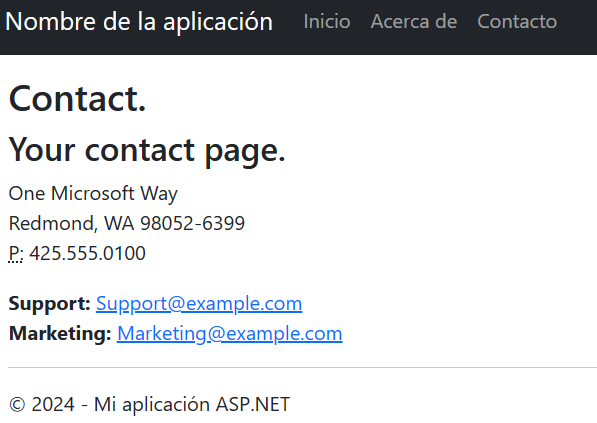
Sobre CapaPresentacionAdmin, le damos clic derecho, Establecer como proyecto de inicio, y le damos ejecutar para abrir la anterior ventanas en el navegador





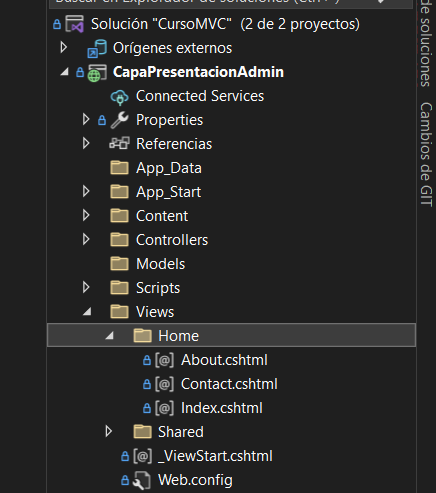
Aquí nos va a mostrar los formularios que por defecto crea la plantilla MVC:



Estas tres vistas las encontramos para edición en:

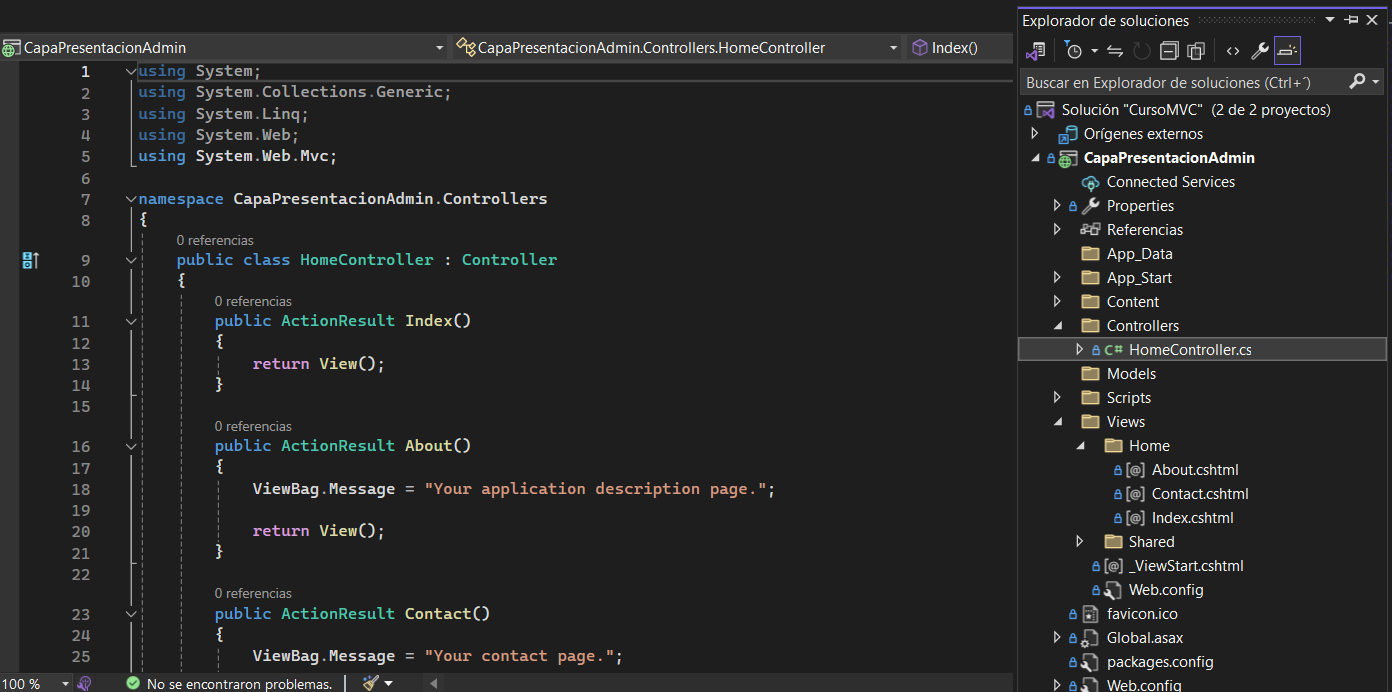
Explorador de soluciones, CapaPresentacionAdmin, Views, Home, (aquí están about, contact e index como paginas html, suceptibles a modificación)



El que dice que mostrar (el que controla) es el archivo llamado:

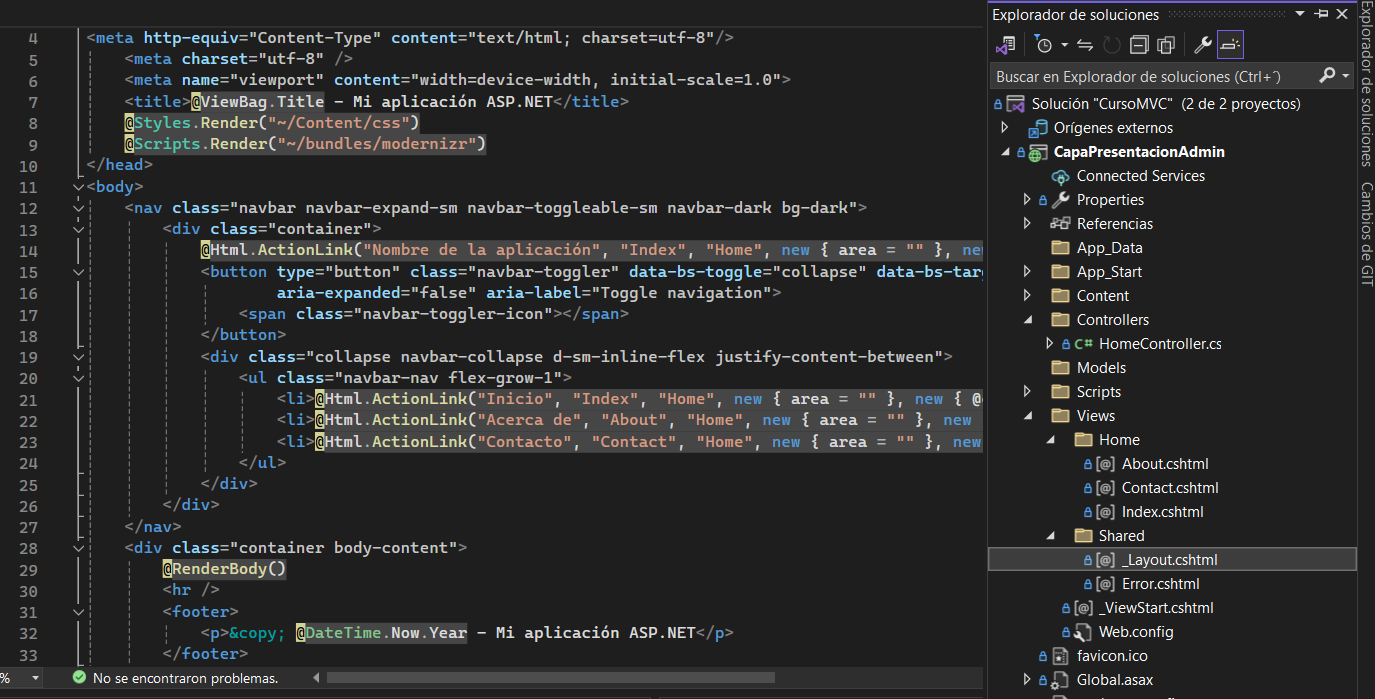
HomeController.cs

Este se compone de estos tres métodos, los cuales son del tipo ActionResult, los cuales devuelven una acción, en este caso, retornar una vista.



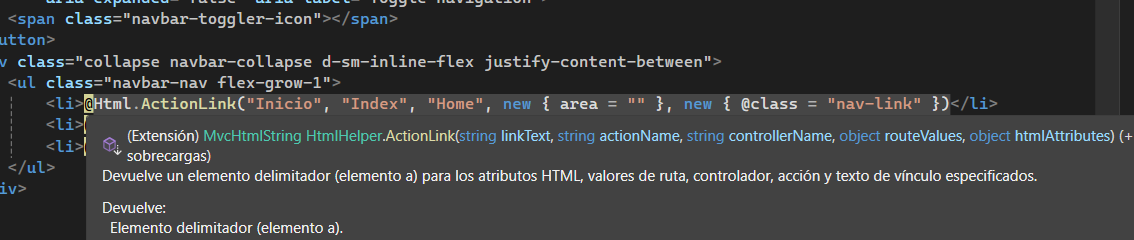
El llamado de estos métodos se hace a través de CapaPresentacionAdmin, Views, Shared en el apartado Layout.

Es diferente la vista en HTML a como está construido



Si quisiéramos modificarlo debemos tener en cuenta

1. El texto que va a mostrarse.
2. Nombre de la acción.
3. Nombre del controlador.



**1**

**3**

**2**

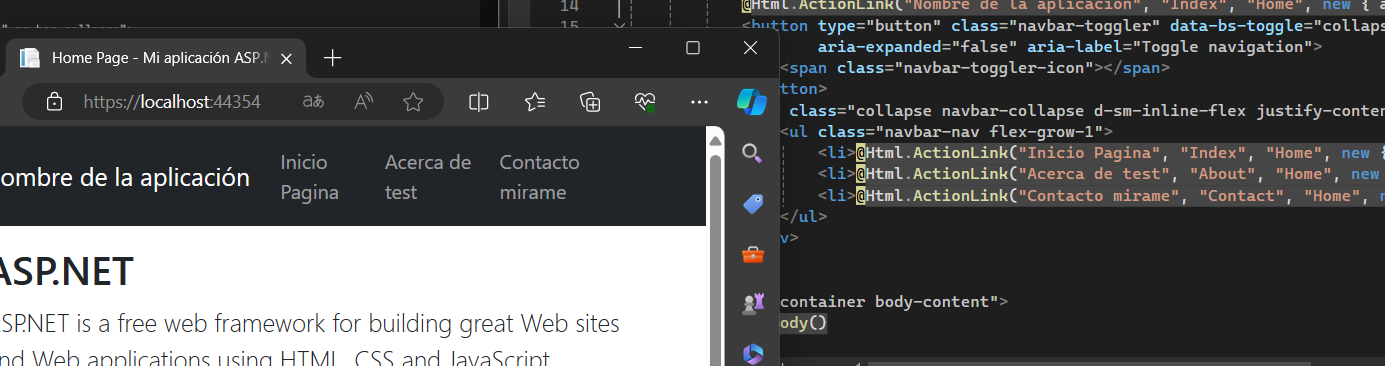
**3**

**2**

**1**

Como nos damos cuenta, lo modificamos y ahora se ve así

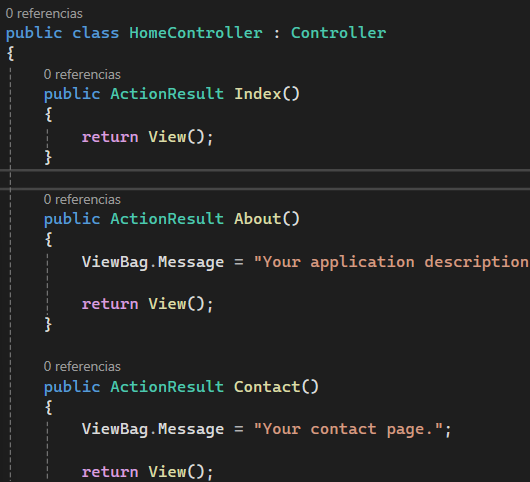
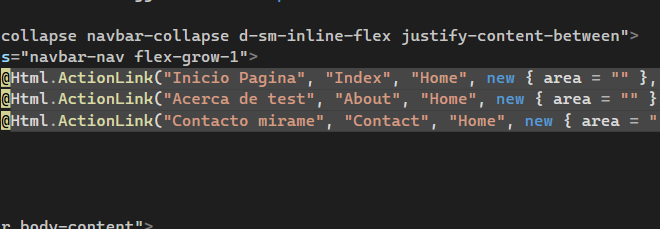
Inicio > Inicio Página. Acerca > Acerca de test. Contacto > Contacto mírame.



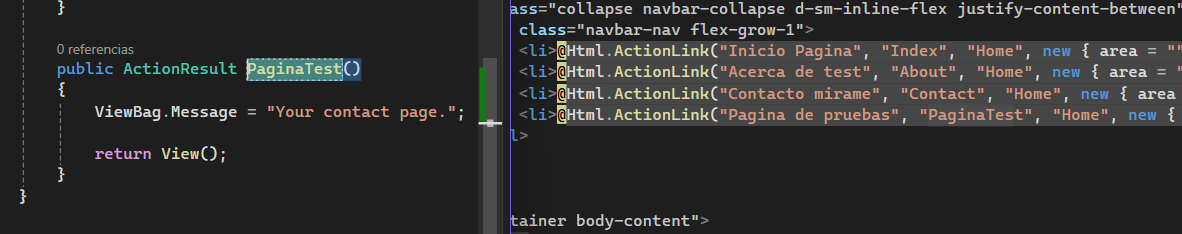
Podemos modificar ese texto, pero no el 2) ni el 3).

El 2) viene del nombre de los métodos ActionResult escritos en el Controller. Esos deben seguir igual.

El 3) viene ligado al HomeController.cs



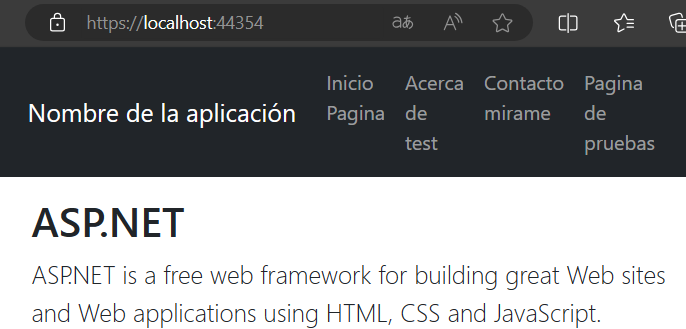
Agregando nueva vista: En el controller agregamos un nuevo método llamado PaginaTest (copiando y pegando los que están mas arriba), y en el Layout agregamos igual otro item dentro del html



**Layout.cshtml**

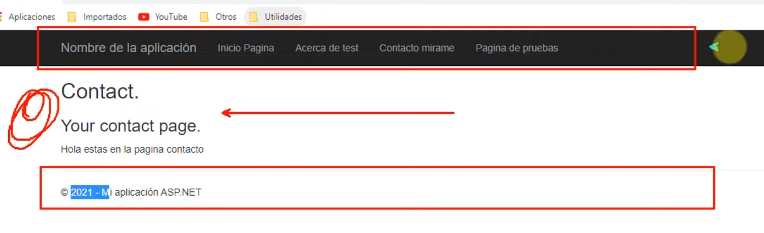
**HomeController.cs**

Lo cual se vería así:



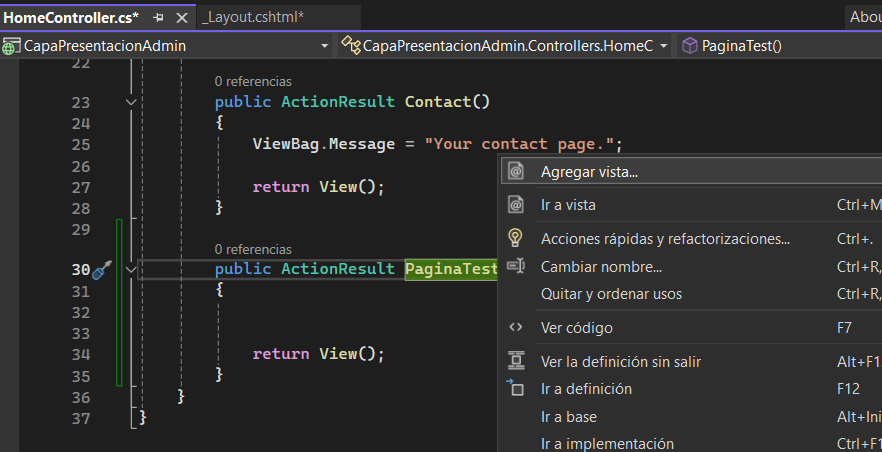
**Hemos agregado “Página de pruebas”**

Nuestro aplicativo tiene el diseño de una Master Page, lo cual significa que todas las vistas se verán igual en su parte superior e inferior porque esto viene definido en el Layout; lo único que cambia es el contenido del medio, el cual podemos variarlo o modificarlo dentro de

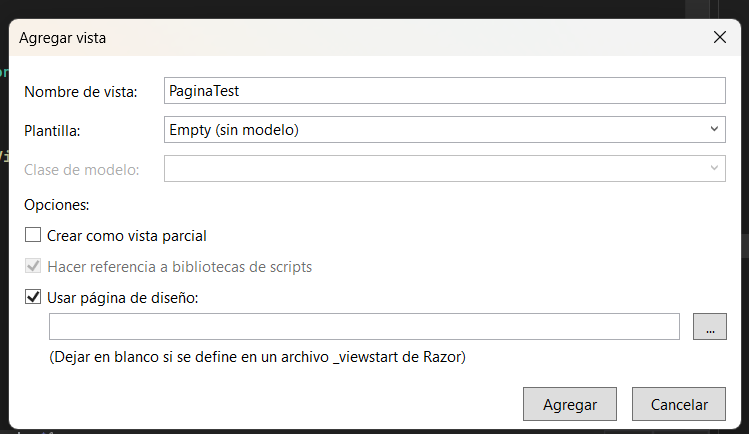


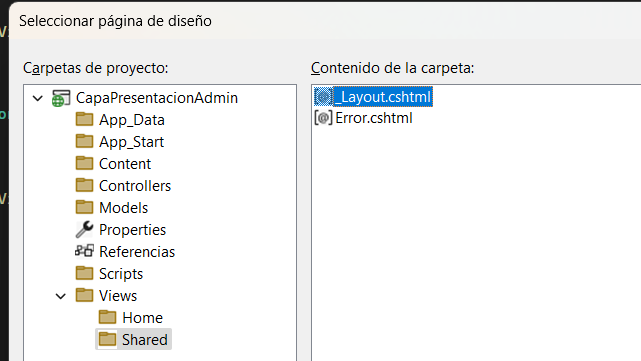
Este tipo de configuración se puede hacer gracias a que dentro del Layout hay un apartado de @RenderBody(), el cual permite hacer variar el contenido del medio, pero manteniendo siempre el resto igual de la master page.

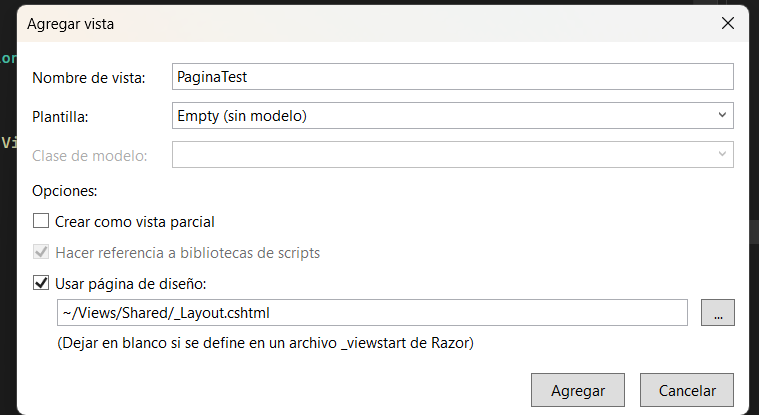
Para modificar lo que se ve dentro de la nueva pestaña, debemos editar el método en el HomeController.cs, ya que de igual forma este presentaba error. Para ello debemos sobre el nombre del método (PaginaTest) lo seleccionamos, clic derecho, agregar vista, vista de MVC 5, solo modificamos “usar pagina de diseño” que es para seguir utilizando la plantilla de la master page (mantener cabecera y footer), para ello, Views, Shared, Layout.cshtml (que es la master page)



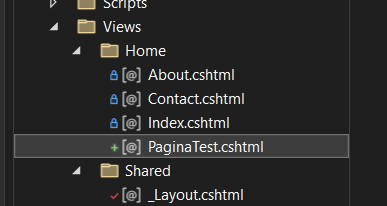
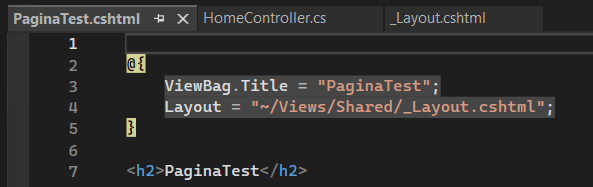




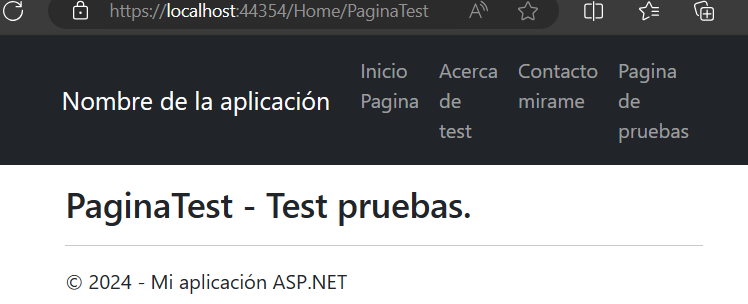




Ahora se agregó este archivo PaginaTest.cshtml

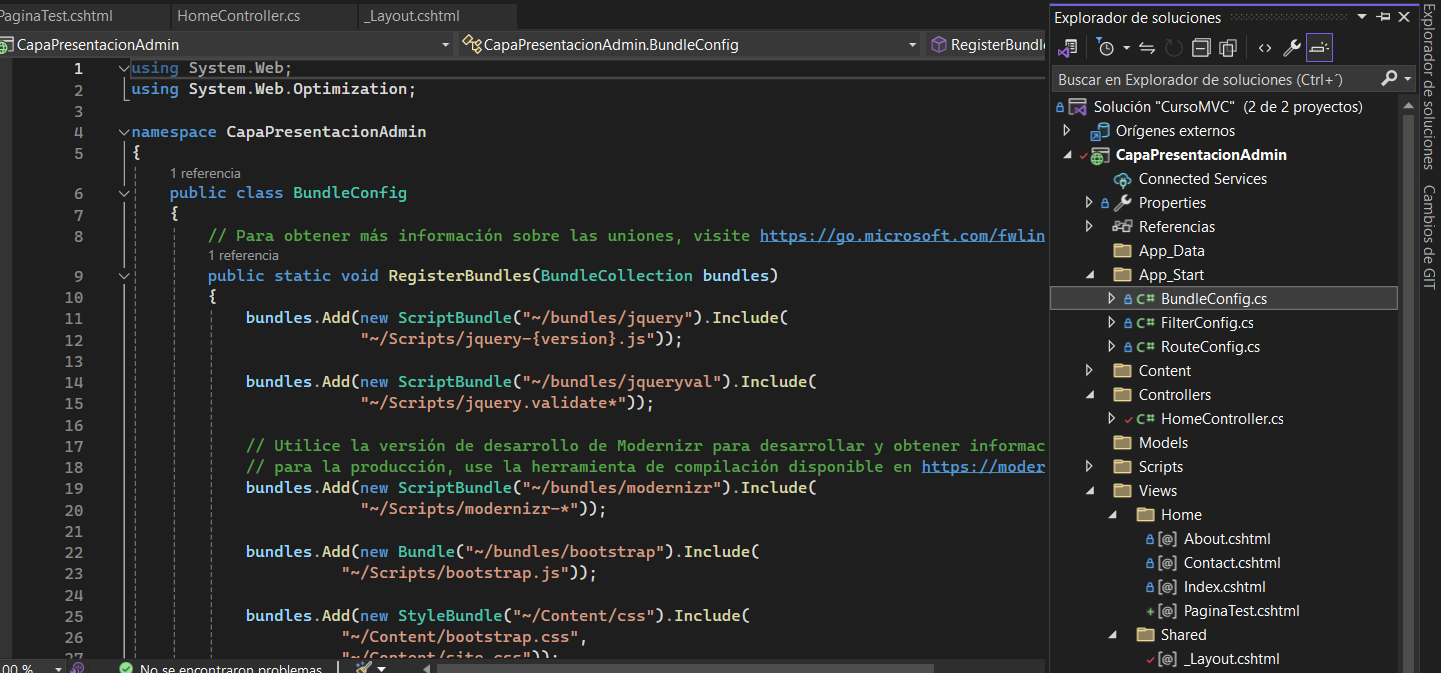
 

Si ejecutamos la aplicación, vemos que ya podemos acceder a la nueva pestaña “Pagina de pruebas”, y esta mantiene la estructura de la Master Page.

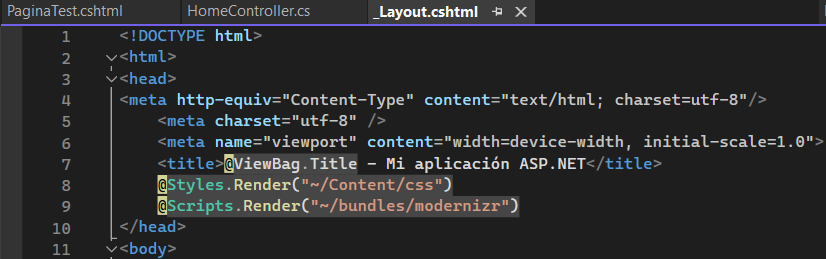


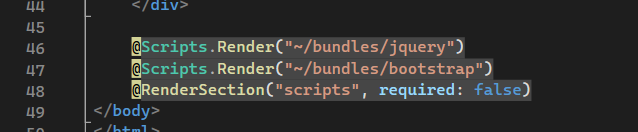
Los estilos y bundles los encontramos en la carpeta: App\_Start.

Estos son invocados en el archivo Layout.cshtml



**Aquí se invocan a los estilos y bundles en Layout.**





Debemos descargar las plantillas a usar, las cuales se encuentran en:

<https://startbootstrap.com>

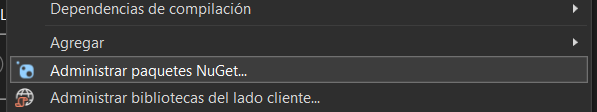
y las plantillas en especifico son estas tres:

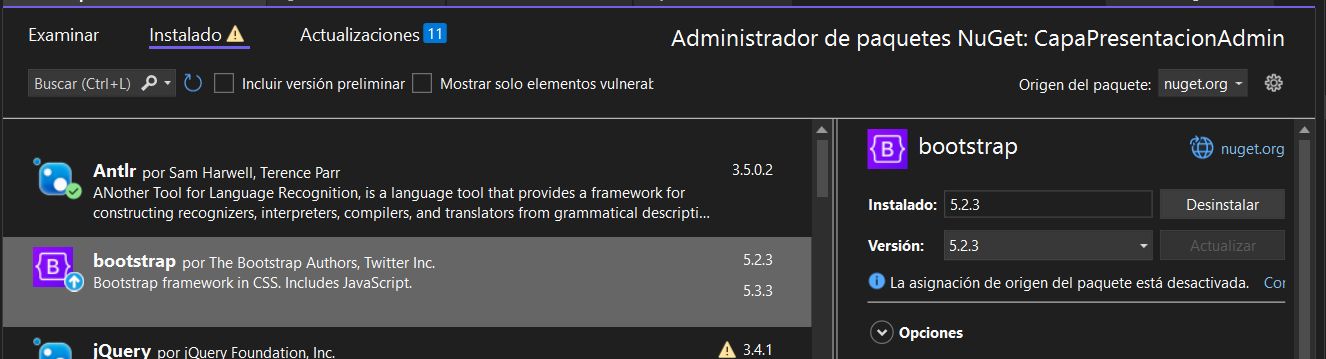
<https://startbootstrap.com/template/sb-admin>

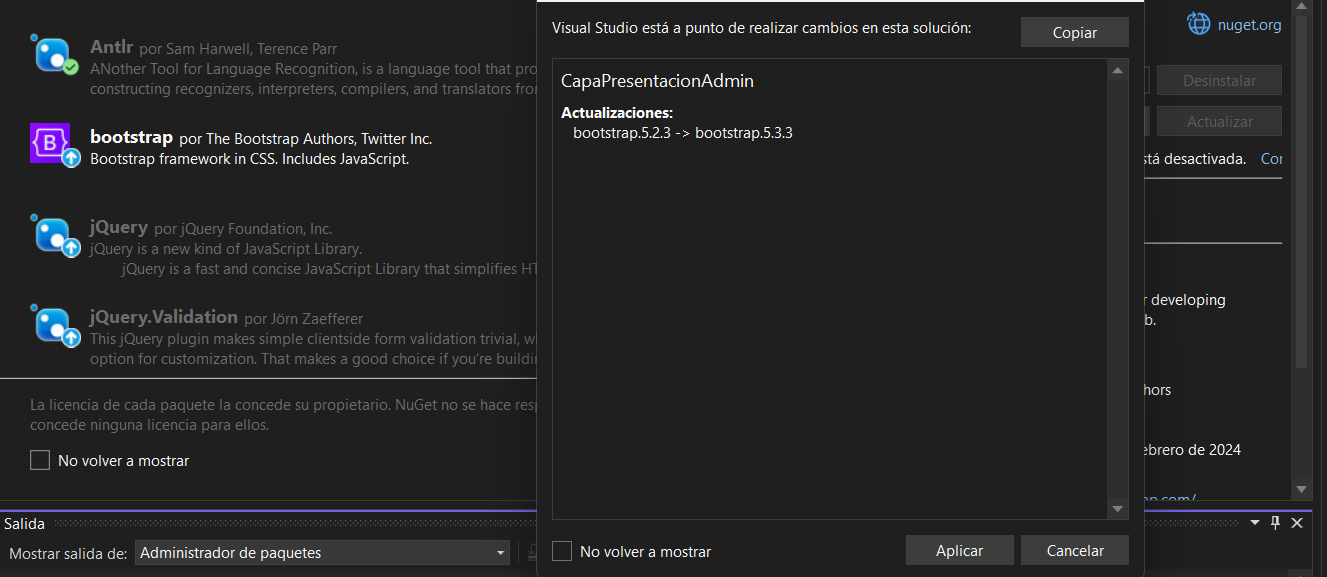
<https://startbootstrap.com/template/shop-homepage>

<https://startbootstrap.com/template/shop-item>

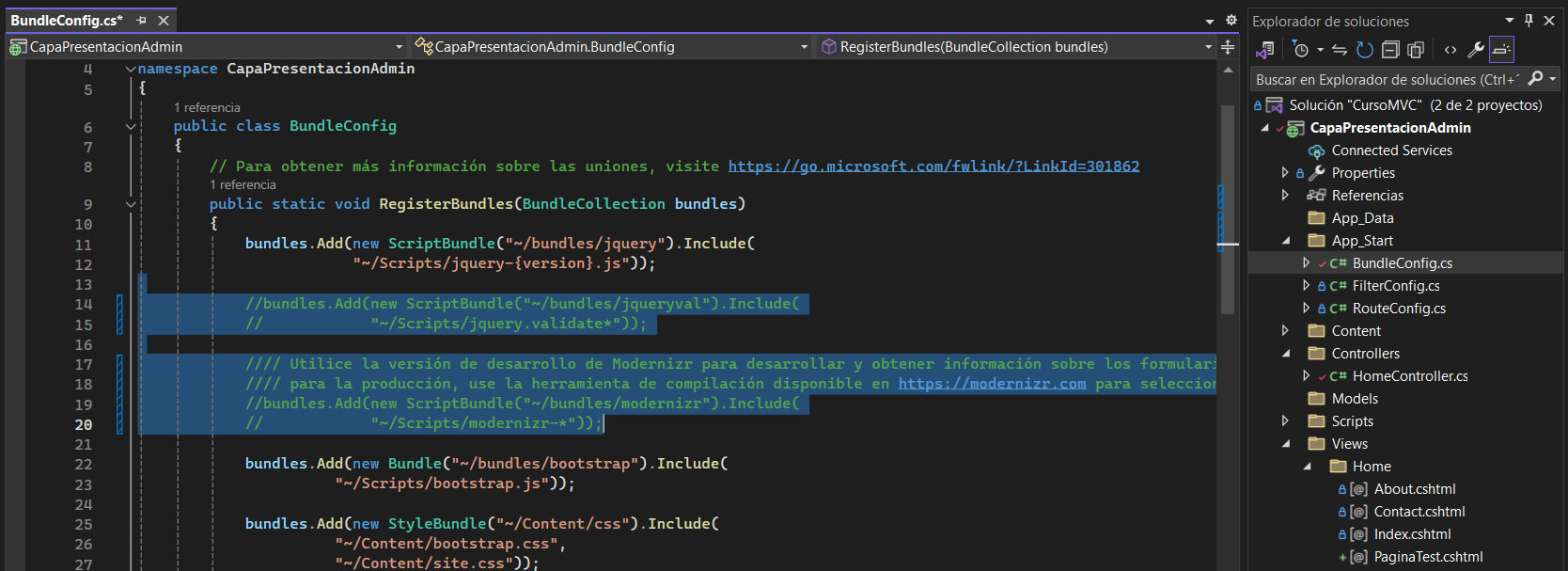
Todas ellas manejan Bootstrap 5, para ello debemos validar en el visual studio que versión de Bootstrap tenemos instalada, para ello nos vamos a CapaPresentacionAdmin, clic derecho, agregar paquetes NuGet…, en Instalado buscamos Bootstrap, y lo mas seguro es que sea una versión inferior, en caso de ser así, solo debemos seleccionar la ultima versión mas estable, y darle actualizar. **Lo mismo lo debemos hacer para CapaPresentacionTienda.**





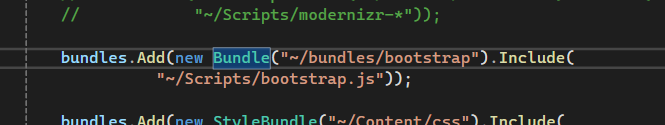


En los Bundles comentamos las siguientes líneas:



Así mismo debemos modificar dos líneas de código, la 22 y la 23, cambiando Bundle por ScritpBundle; y bootstrap.js por bootstrap.bundle.js

*Como estaba*



*Como debe quedar*

