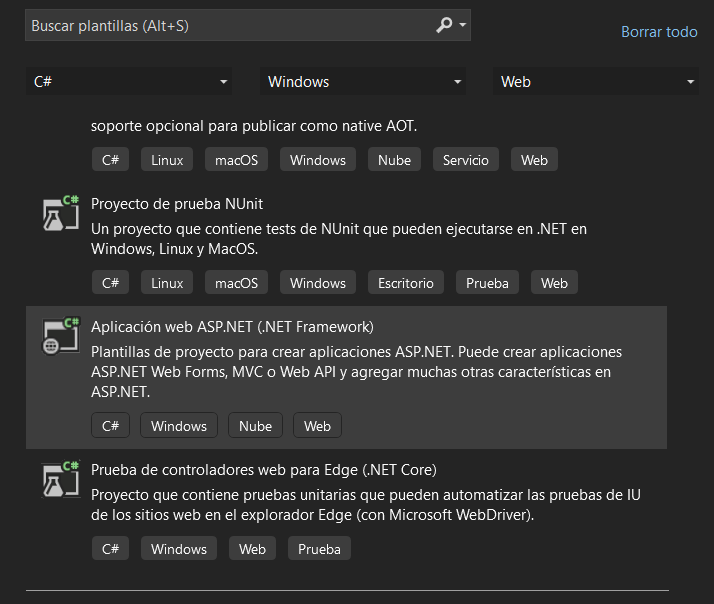
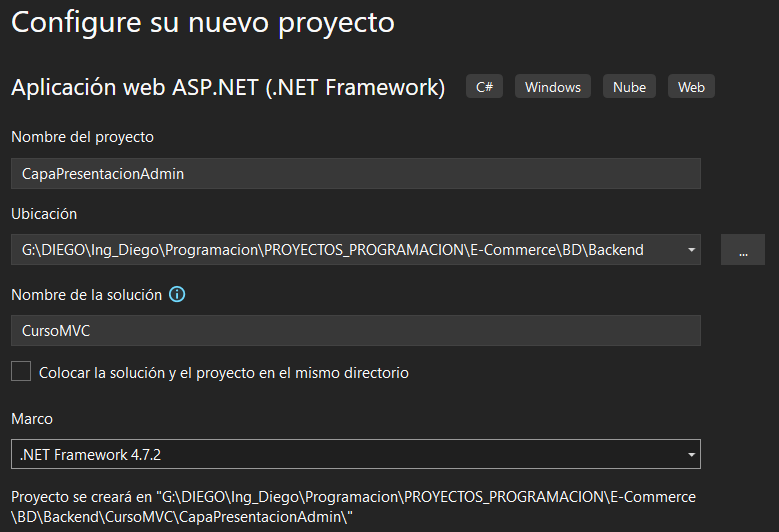
**CREACIÓN DEL BACKEND**

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”.



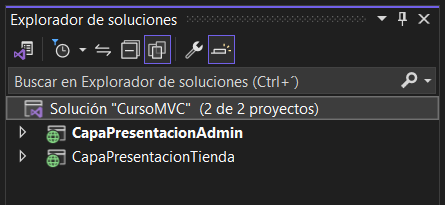




Sobre Solución “CursoMVC” (el nombre que le dimos a la solución), clic derecho, agregar, Nuevo proyecto y lo llamaremos “CapaPresentacionTienda” y elegimos las mismas configuraciones que en la hoja anterior.

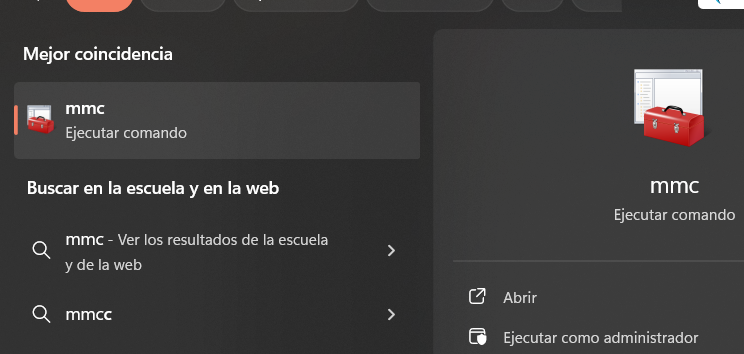
Se crean dos porque una es la perspectiva (modulo) desde el lado del admin y la otra desde el lado del cliente.

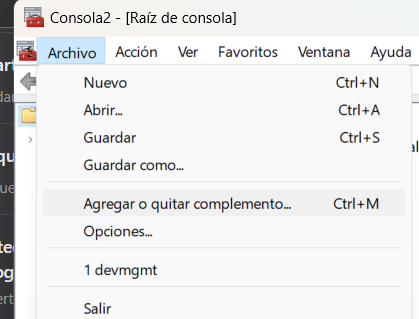
Debe quedar así:

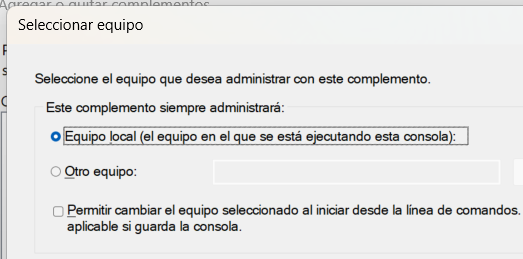
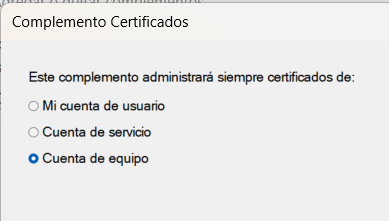


Para eliminar las restricciones SSL que nos puede presentar el equipo al ejecutar el aplicativo debemos:

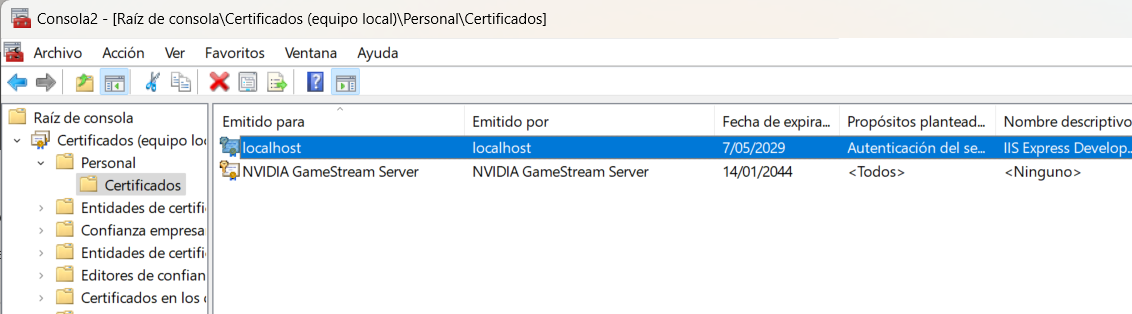
Clic en buscar, escribimos mmc, archivo, agregar o quitar complemento, Certificados, Agregar, aceptar, cuenta de equipo, siguiente, equipo local, finalizar, aceptar. Este se encontrará ahora dentro de Raiz de consola, Certificados, Personal, Certificados.





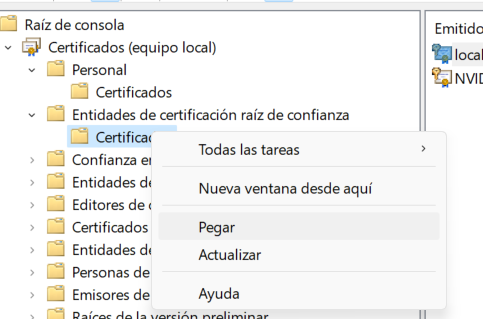




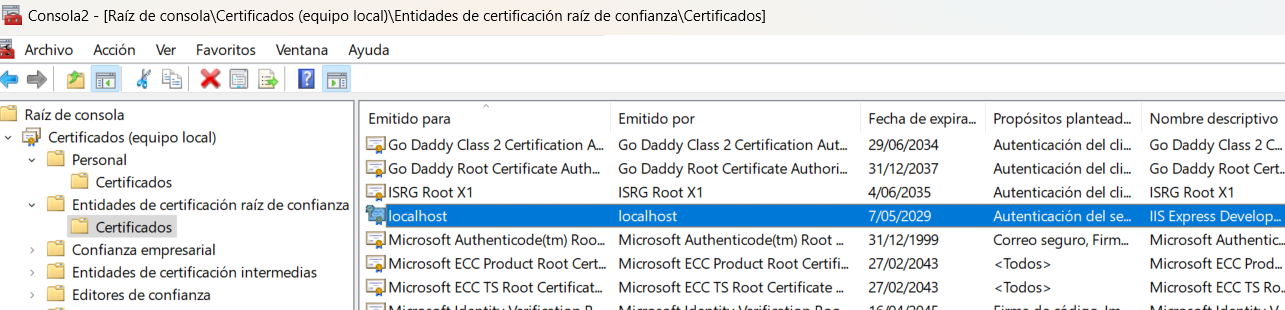


Como se puede ver tiene una fecha de expiración, para que eso no sea así debemos:

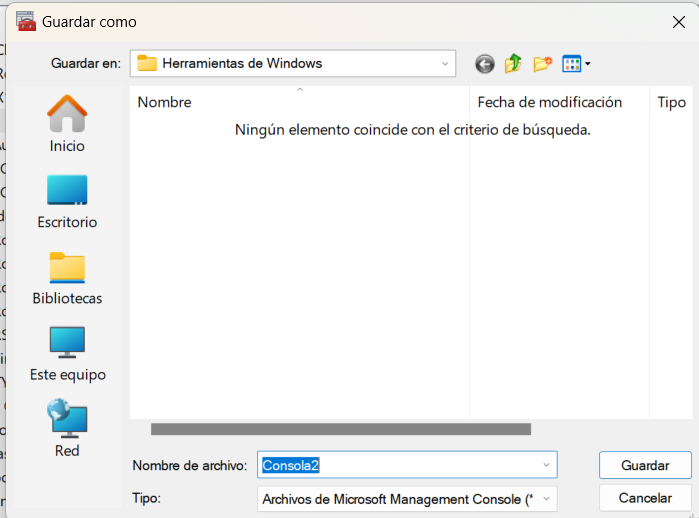
Lo copiamos el que acabamos de crear “localhost”, y dentro de Trusted Root Certificates o Entidades de Certificación Raíz de Confianza, lo pegamos:



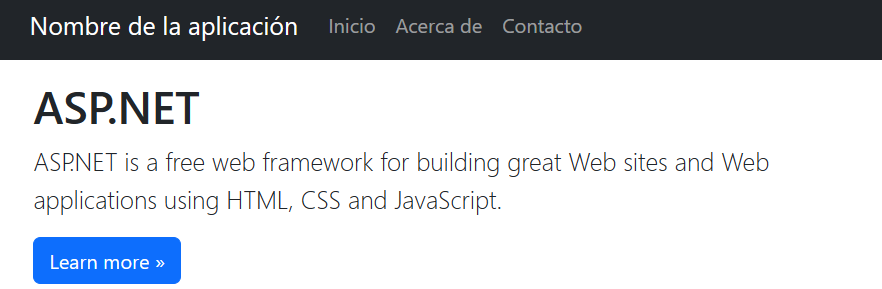
Y quedará así:



Guardamos y cerramos:

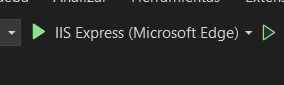


Y nos debe salir esto al ejecutar el aplicativo:

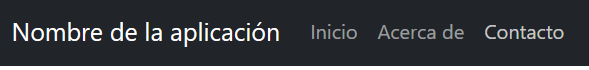


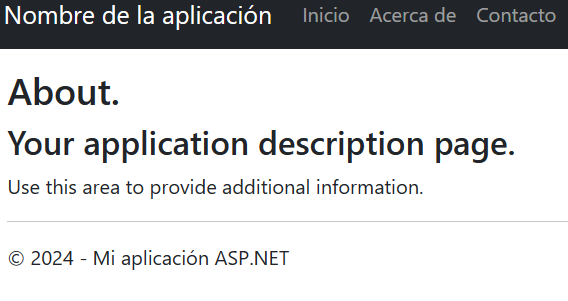
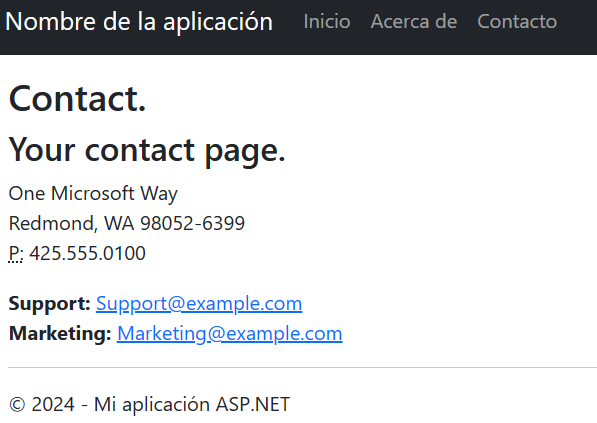
Sobre CapaPresentacionAdmin, le damos clic derecho, Establecer como proyecto de inicio, y le damos ejecutar para abrir la anterior ventanas en el navegador





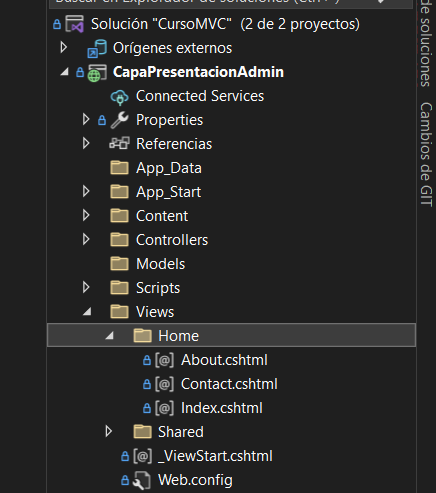
Aquí nos va a mostrar los formularios que por defecto crea la plantilla MVC:



Estas tres vistas las encontramos para edición en:

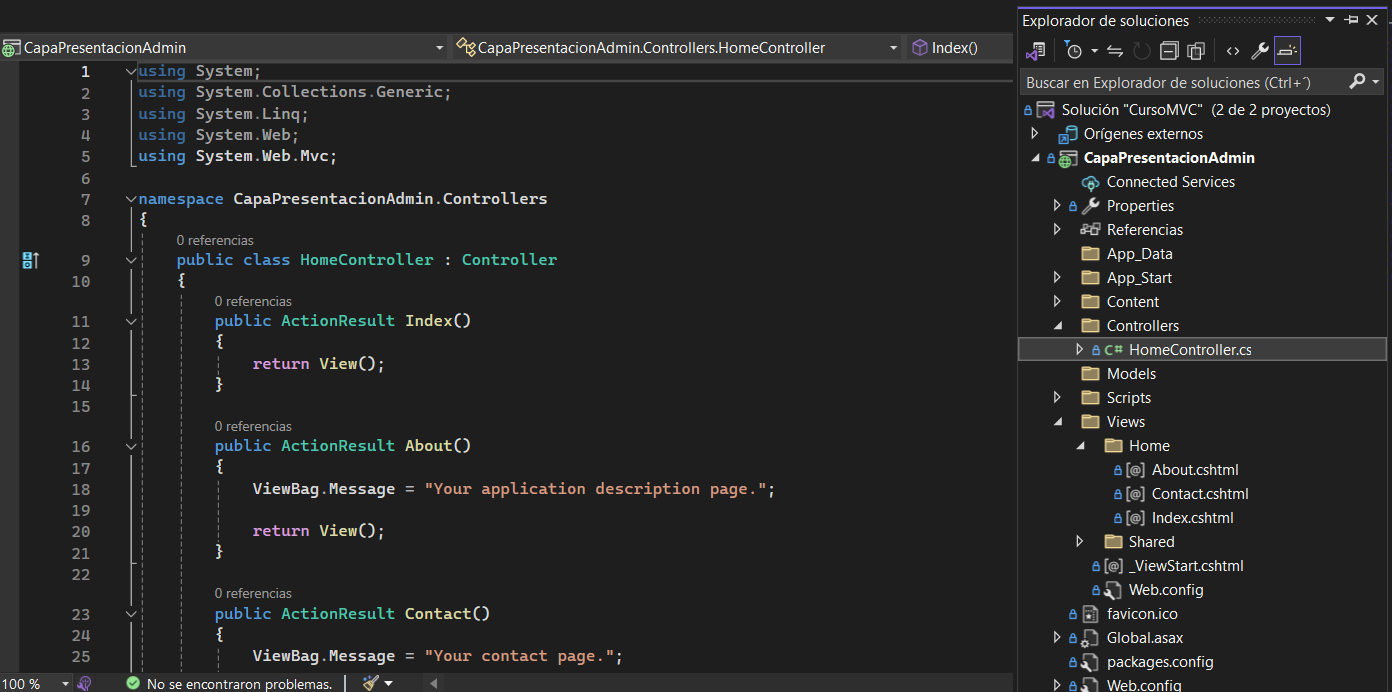
Explorador de soluciones, CapaPresentacionAdmin, Views, Home, (aquí están about, contact e index como paginas html, suceptibles a modificación)



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

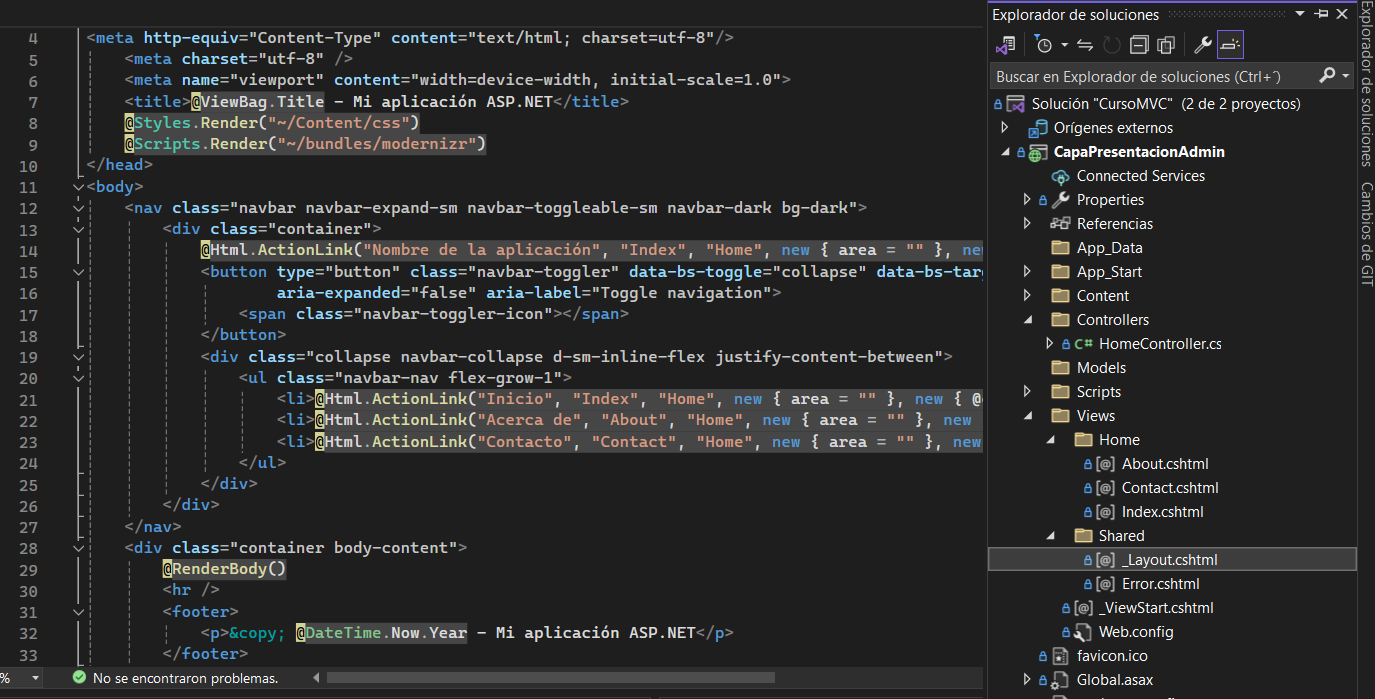
HomeController.cs

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



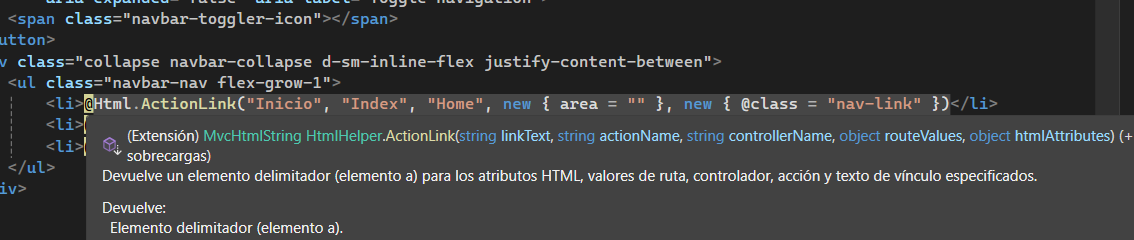
El llamado de estos métodos se hace a través de CapaPresentacionAdmin, Views, Shared en el apartado Layout.

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**

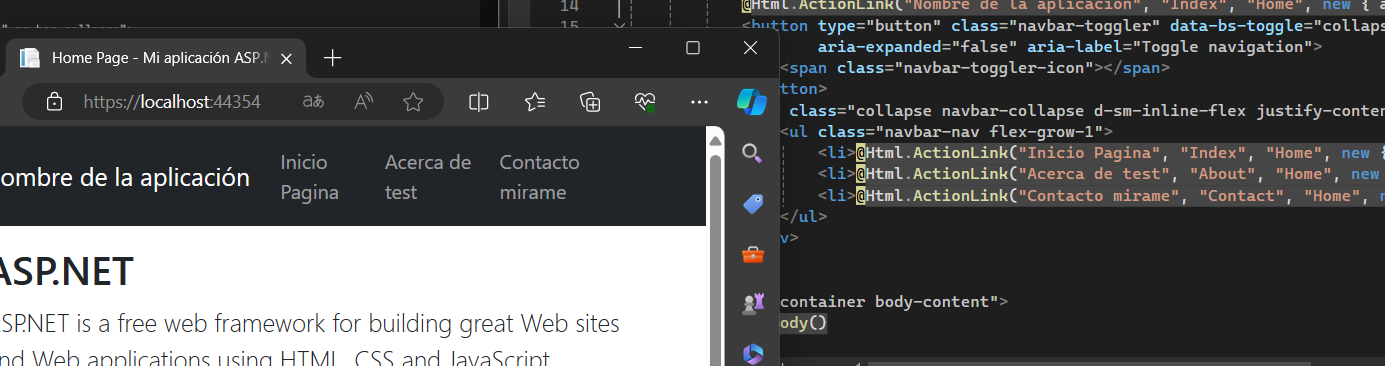
**3**

**2**

**1**

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

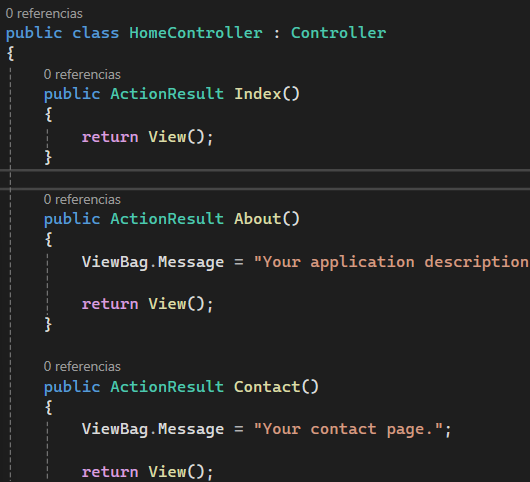
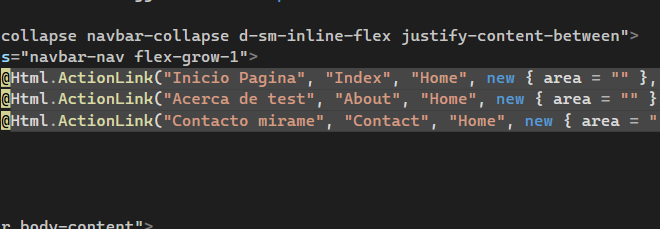
Inicio > Inicio Página. Acerca > Acerca de test. Contacto > Contacto mírame.



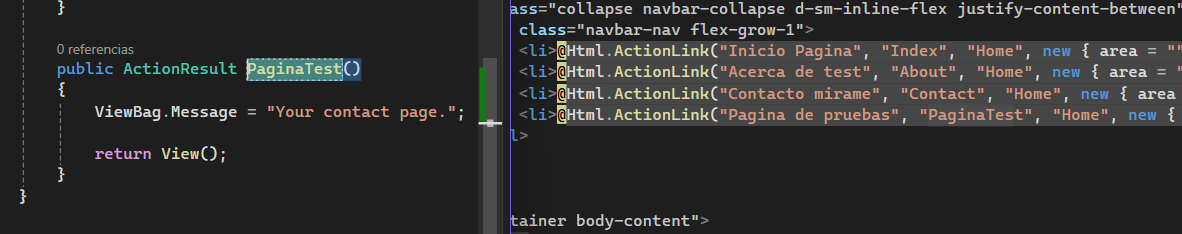
Podemos modificar ese texto, pero no el 2) ni el 3).

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

El 3) viene ligado al HomeController.cs



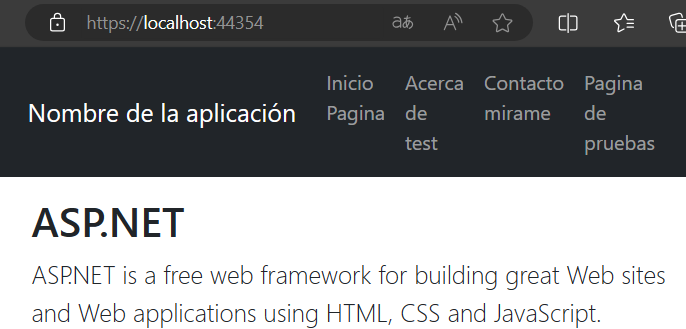
Agregando nueva vista: En el controller agregamos un nuevo método llamado PaginaTest (copiando y pegando los que están mas arriba), y en el Layout agregamos igual otro item dentro del html



**Layout.cshtml**

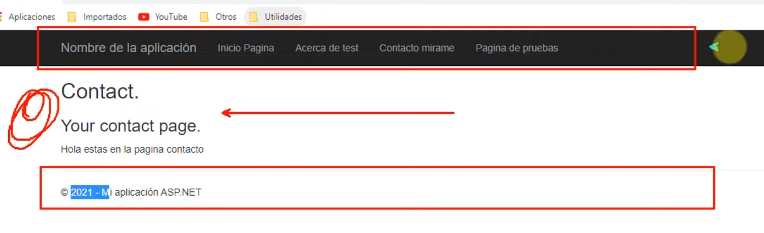
**HomeController.cs**

Lo cual se vería así:



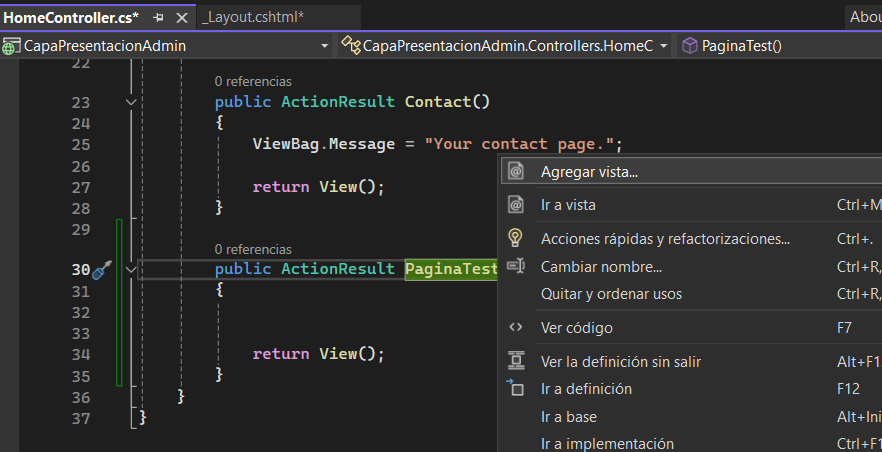
**Hemos agregado “Página de pruebas”**

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

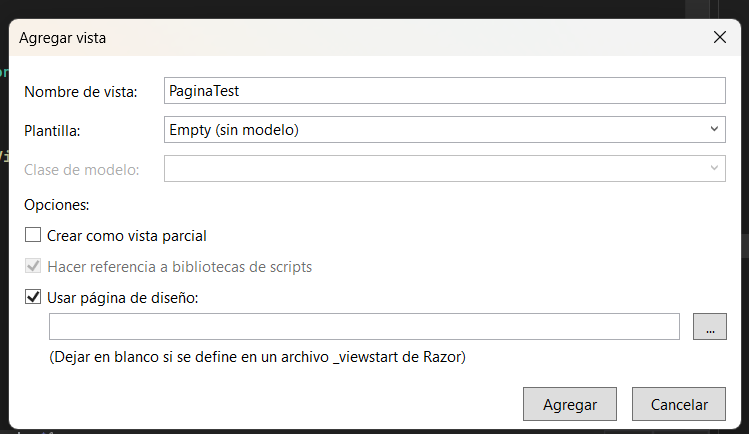


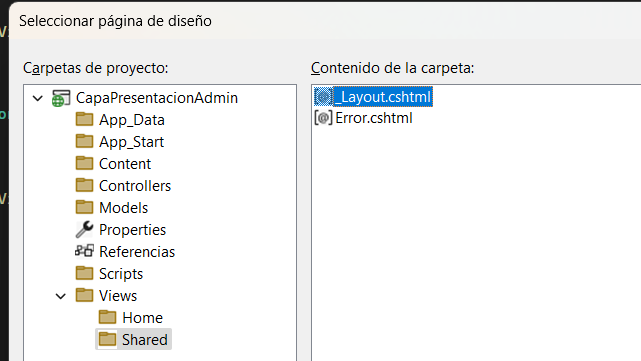
Este tipo de configuración se puede hacer gracias a que dentro del Layout hay un apartado de @RenderBody(), el cual permite hacer variar el contenido del medio, pero manteniendo siempre el resto igual de la master page.

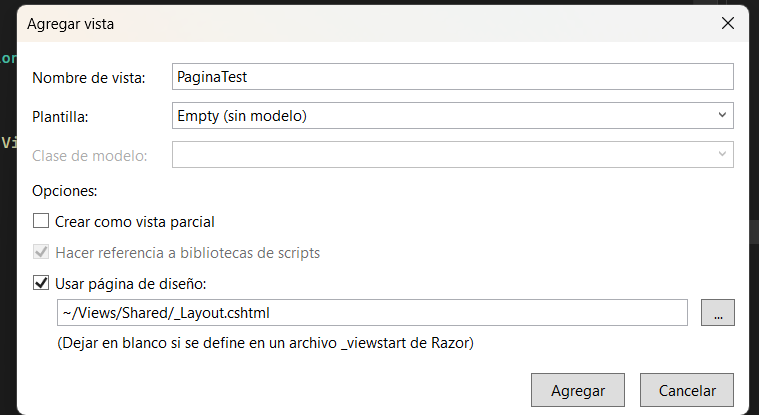
Para modificar lo que se ve dentro de la nueva pestaña, debemos editar el método en el HomeController.cs, ya que de igual forma este presentaba error. Para ello debemos sobre el nombre del método (PaginaTest) lo seleccionamos, clic derecho, agregar vista, vista de MVC 5, solo modificamos “usar pagina 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)



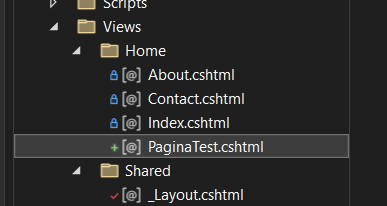
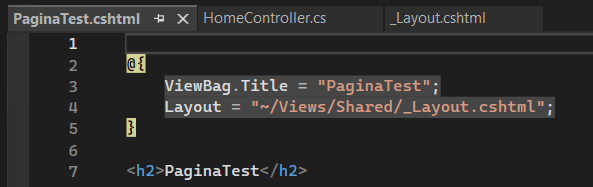








Ahora se agregó este archivo PaginaTest.cshtml

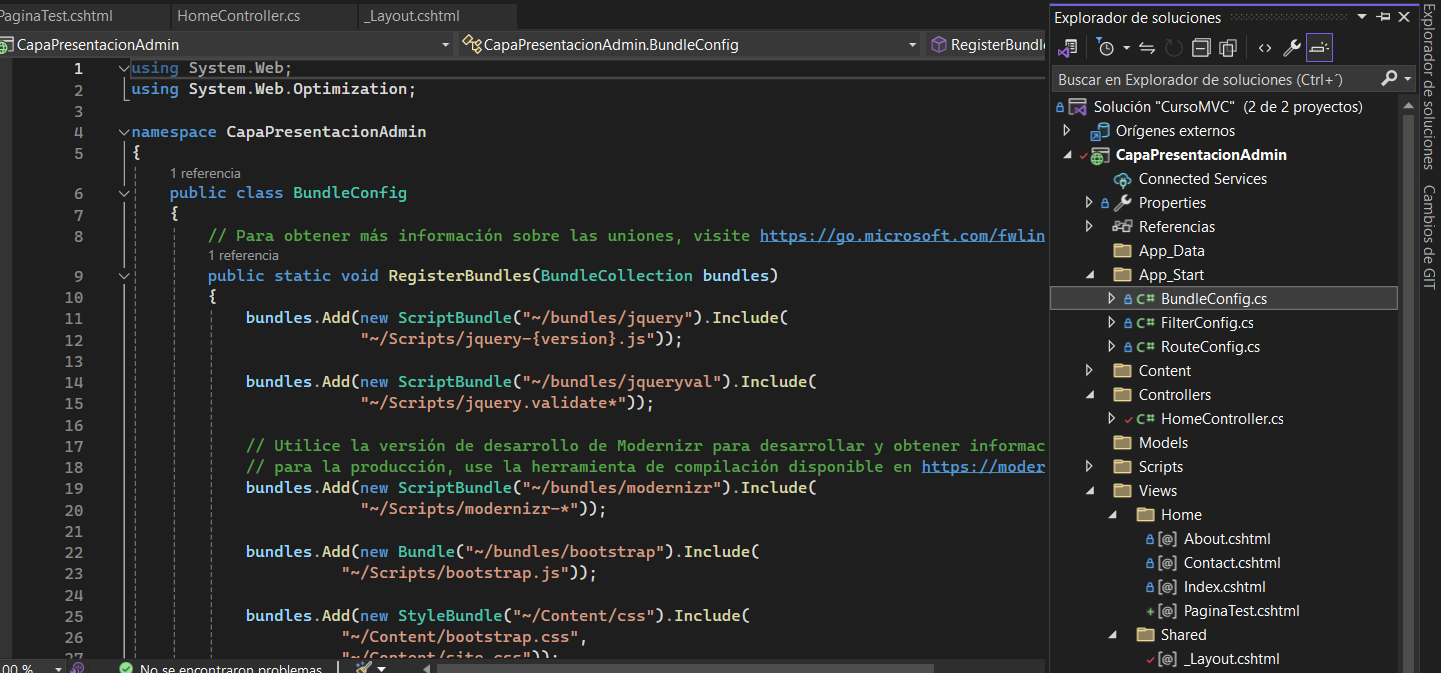
 

Si ejecutamos la aplicación, vemos que ya podemos acceder a la nueva pestaña “Pagina de pruebas”, y esta mantiene la estructura de la Master Page.

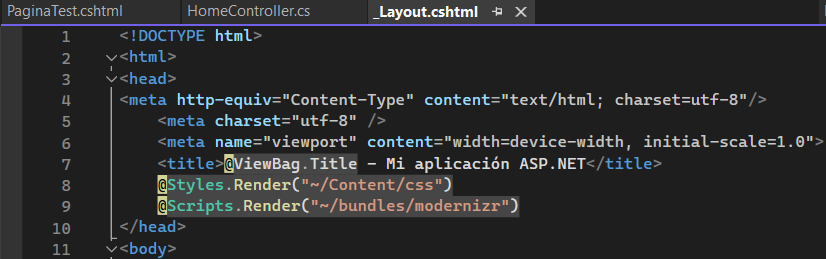


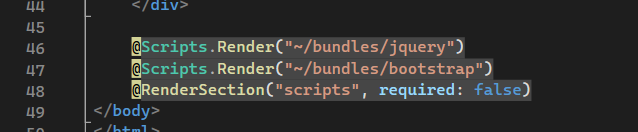
Los estilos y bundles los encontramos en la carpeta: App\_Start.

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>

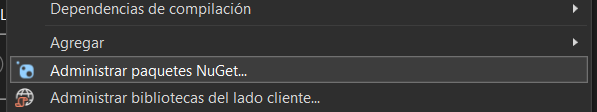
y las plantillas en especifico son estas tres:

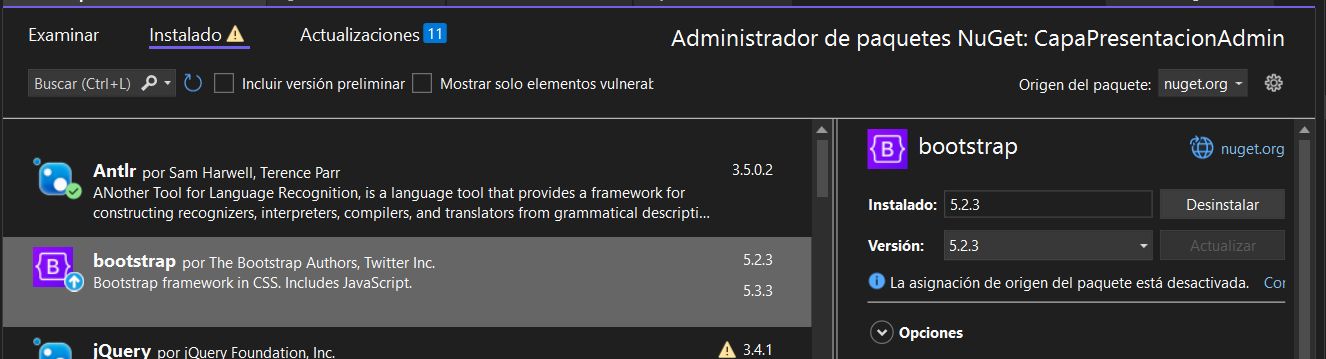
<https://startbootstrap.com/template/sb-admin>

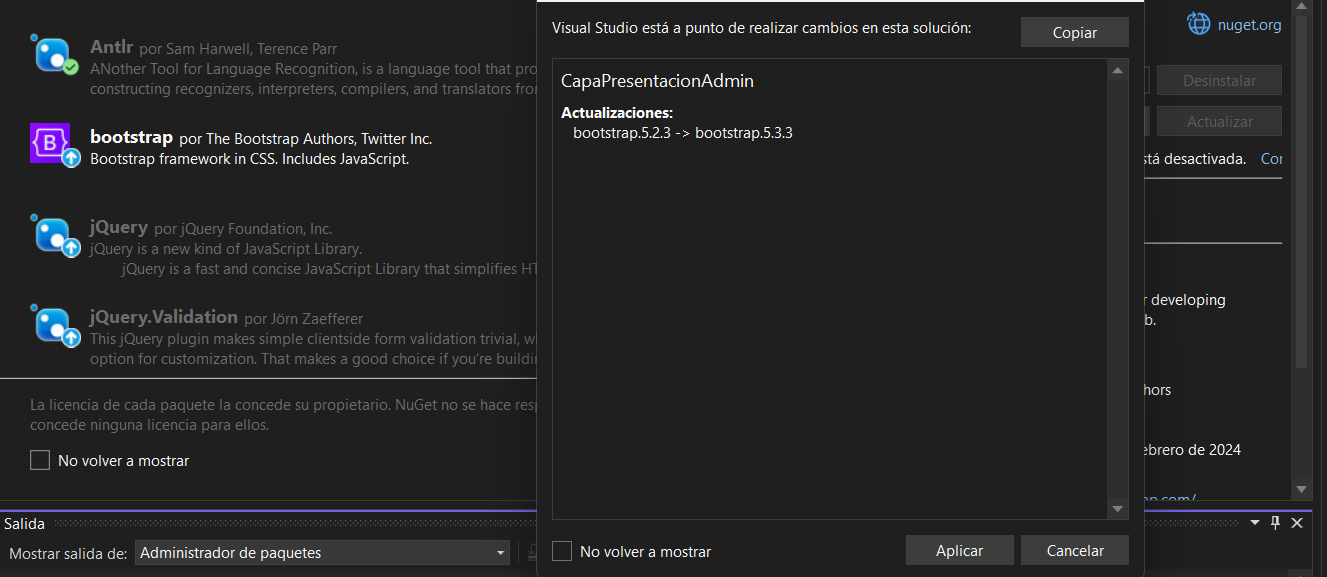
<https://startbootstrap.com/template/shop-homepage>

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

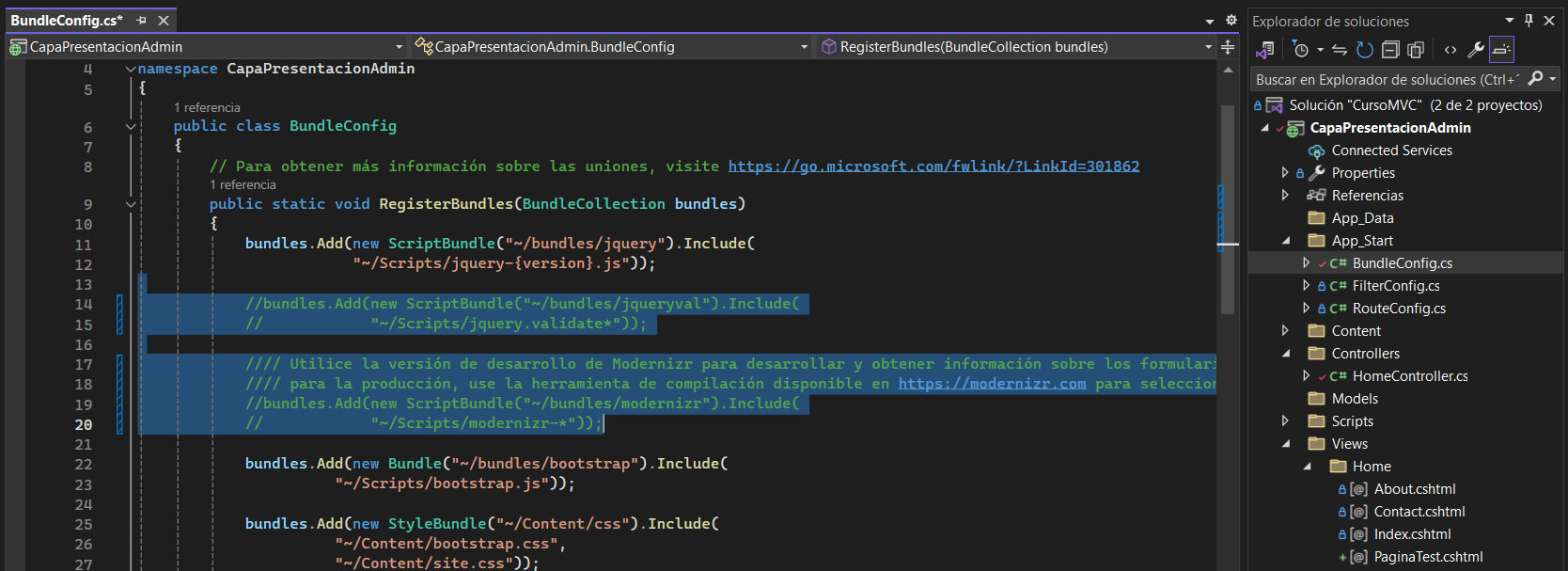
Todas ellas manejan Bootstrap 5, para ello debemos validar en el visual studio que versión de Bootstrap tenemos instalada, para ello nos vamos a CapaPresentacionAdmin, clic derecho, agregar paquetes NuGet…, en Instalado buscamos Bootstrap, y lo mas seguro es que sea una versión inferior, en caso de ser 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:

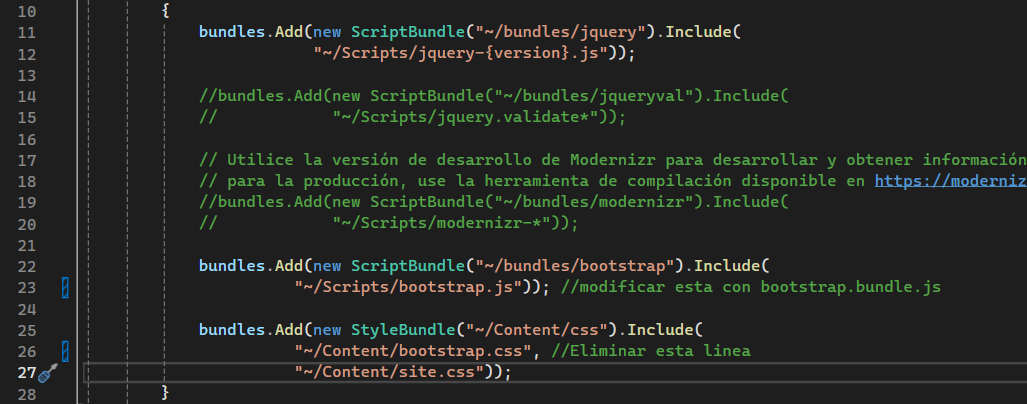


Es posible que presente la “s” del Content/Site.css en minúscula, esta debe quedar en mayúsculas para no presentar errores mas adelante.

Así mismo debemos modificar la línea 23 por "~/Scripts/bootstrap.bundle.js" , eliminar la línea 26 ("~/Content/bootstrap.css",).

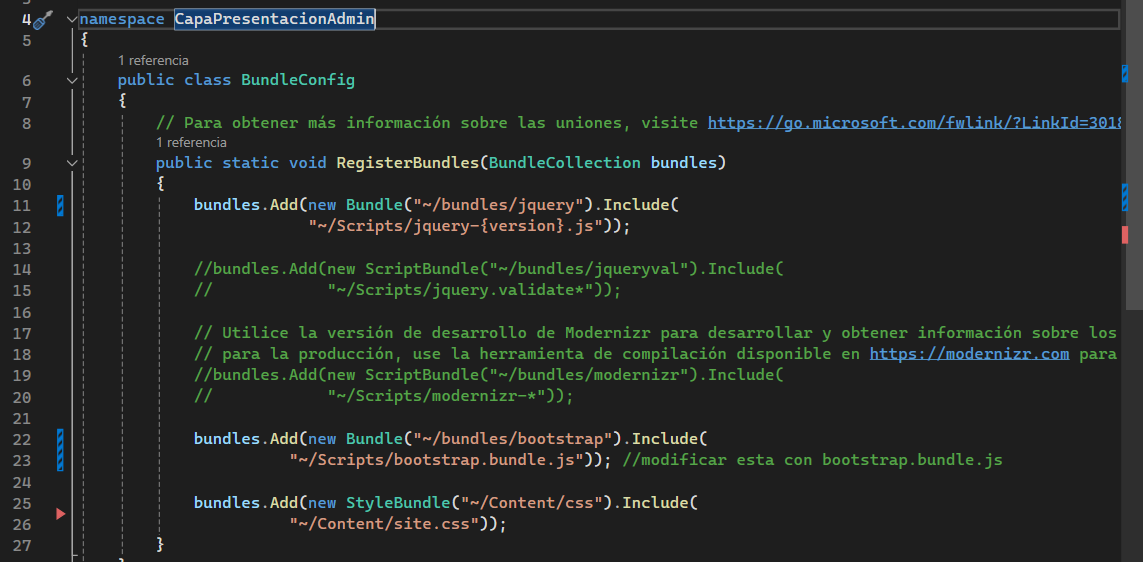
En la línea 11 y 22 debemos eliminar la palabra ScriptBundle y solo dejar Bundle, esto porque en las versiones iguales o superiores a Bootstrap 5, ya no es necesario colocarlo, en versiones previas si.

*Como estaba*

**

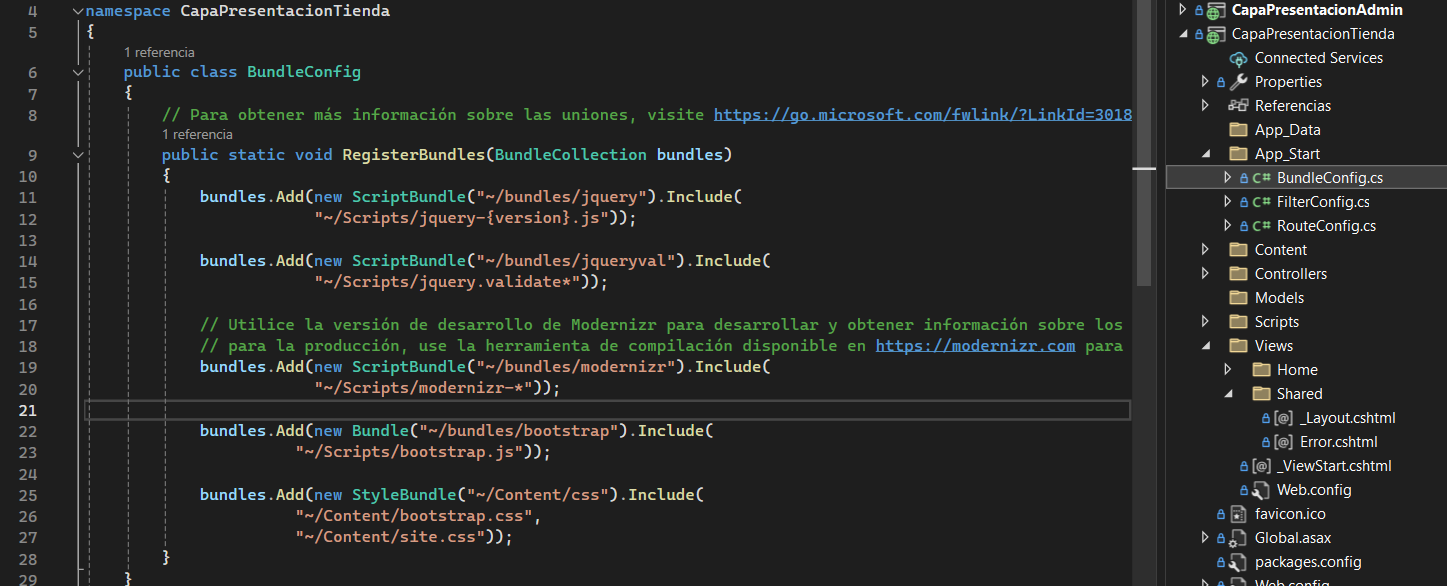
Es posible que presente la “s” del Content/Site.css en minúscula, esta debe quedar en mayúsculas para no presentar errores mas adelante.

*Como debe quedar*



Es posible que presente la “s” del Content/Site.css en minúscula, esta debe quedar en mayúsculas para no presentar errores mas adelante.

Ahora debemos hacer lo mismo con la CapaPresentacionTienda. Pasando de esto:



Es posible que presente la “s” del Content/Site.css en minúscula, esta debe quedar en mayúsculas para no presentar errores mas adelante.

A esto:

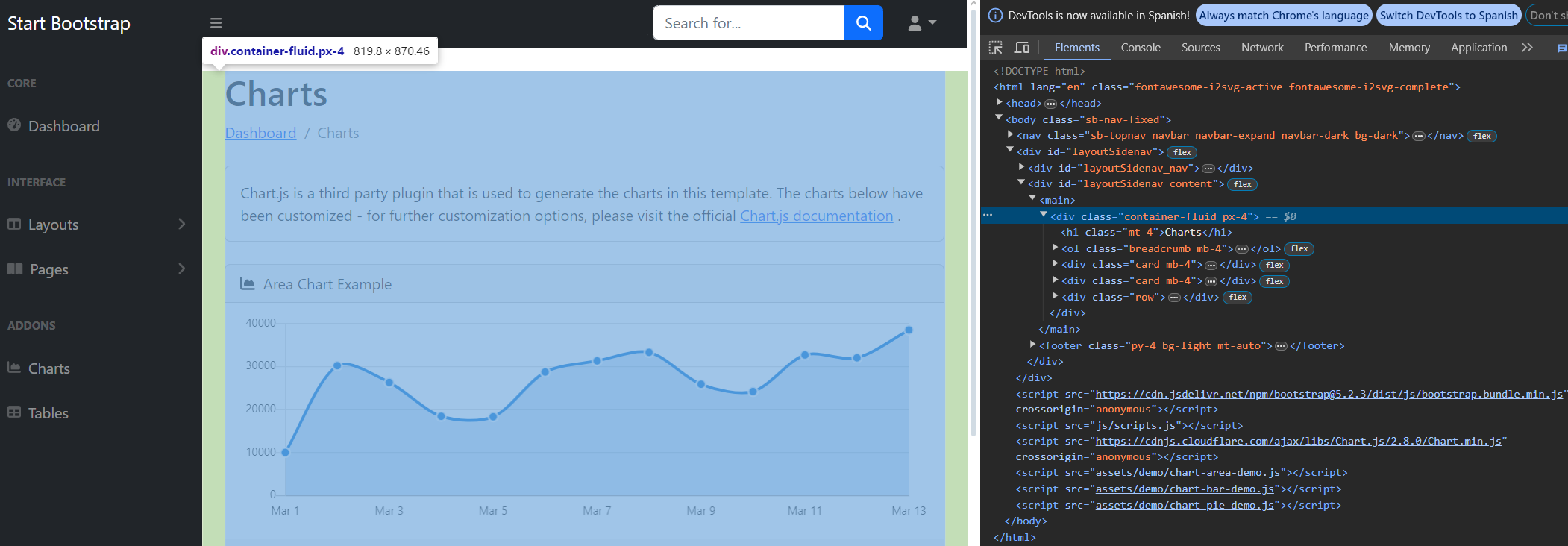


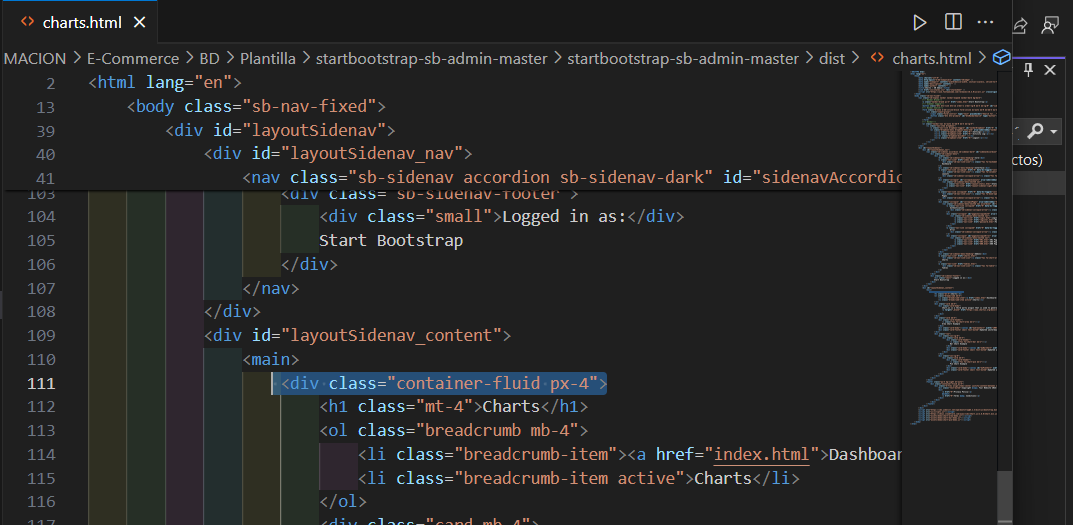
Es posible que presente la “s” del Content/Site.css en minúscula, esta debe quedar en mayúsculas para no presentar errores mas adelante.

Empezando a usar la plantilla.

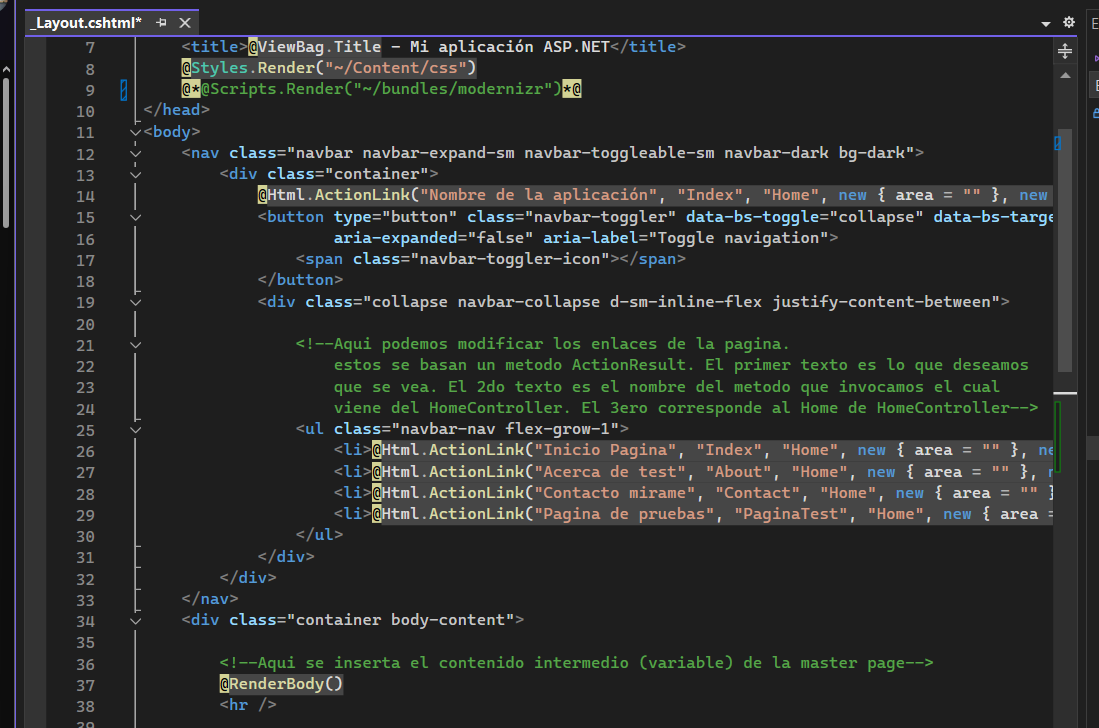
Como primer paso debemos buscar la plantilla descargada a usar. La abrimos con el navegador y vemos que nombre tiene el contenedor variable de la master page en el html. En este caso este se llama **class=”container-fluid px-4"**

Sobre este haremos las modificaciones pertinentes. Abriendo el html con visual studio code, buscaremos la clase container-fluid px-4 todo lo ubicado en este div es lo que cambia

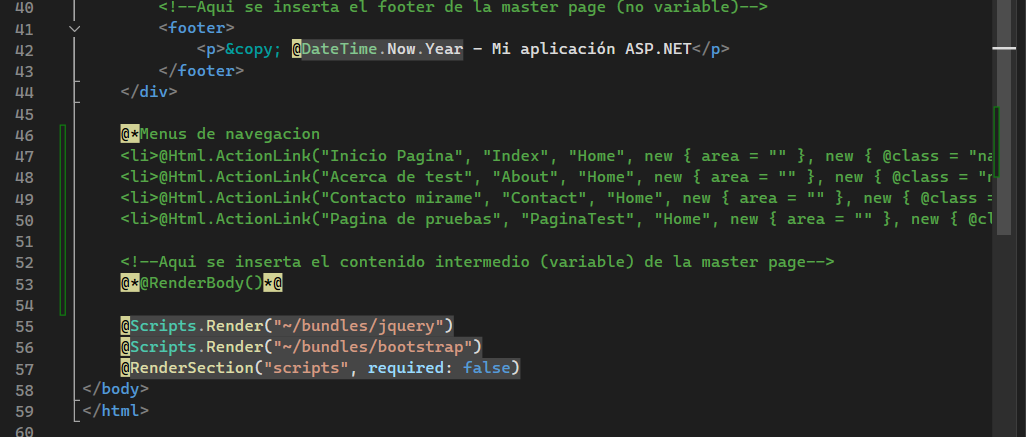




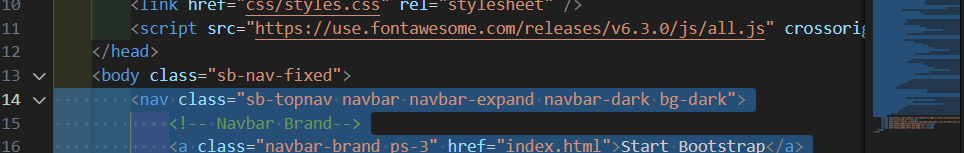
Todo el contenido que está dentro del dody de el archivo layout de CapaPresentacionAdmin, lo borramos, teniendo mucho cuidado de copiar mas abajo la referencia de la master page y de la barra de navegación:

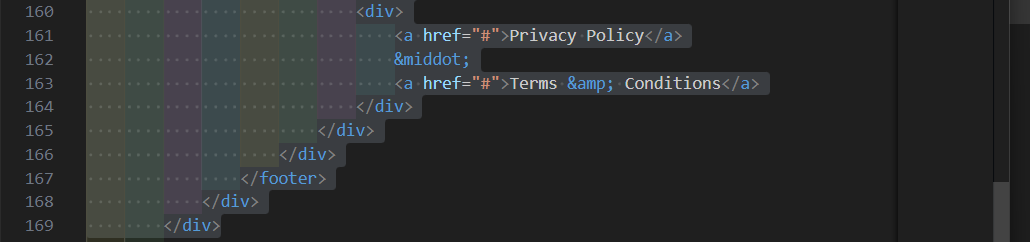


Lo pegaremos al final así:

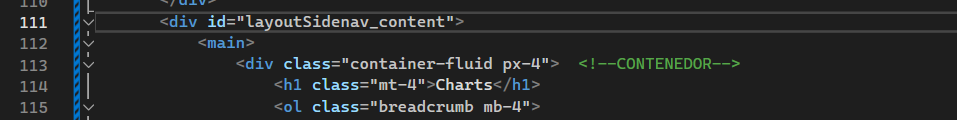


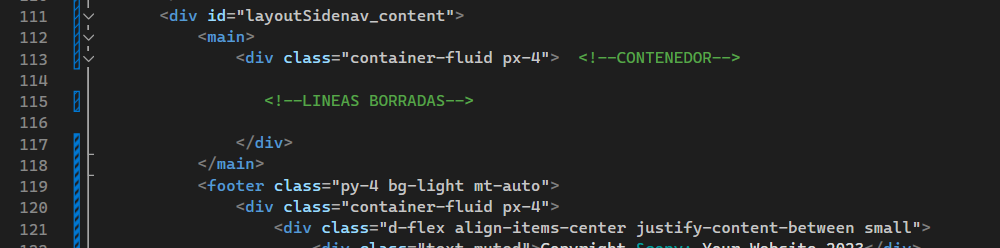
Ahora vamos al VSCode del html de la plantilla, y lo copiamos desde el nav hasta el ultimo div, así:



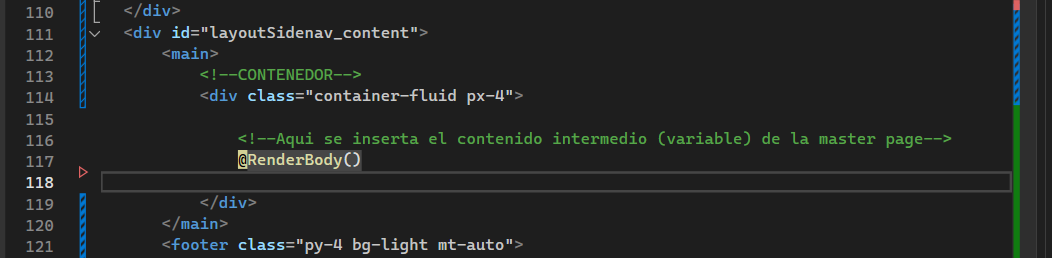


Y lo pegamos en el Visual Studio en el body del Layout

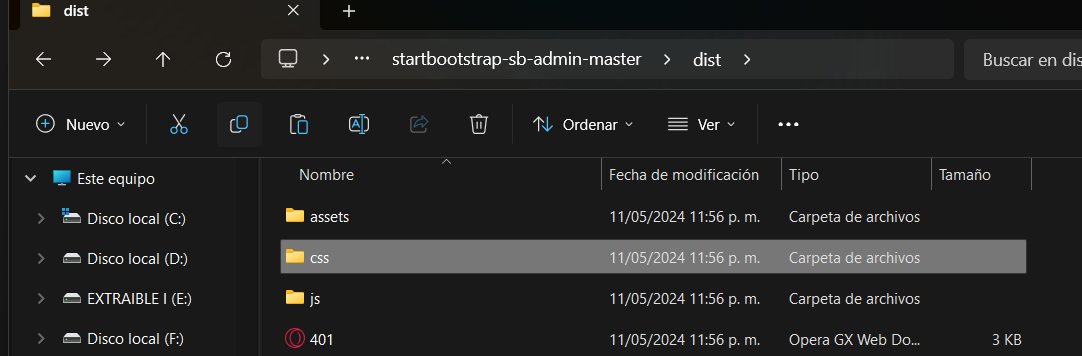
Ahora debemos ubicar el contenedor “container-fluid px-4”, para posteriormente borrar todo lo que hay dentro de el, debemos dejarlo como la segunda imagen de aquí:

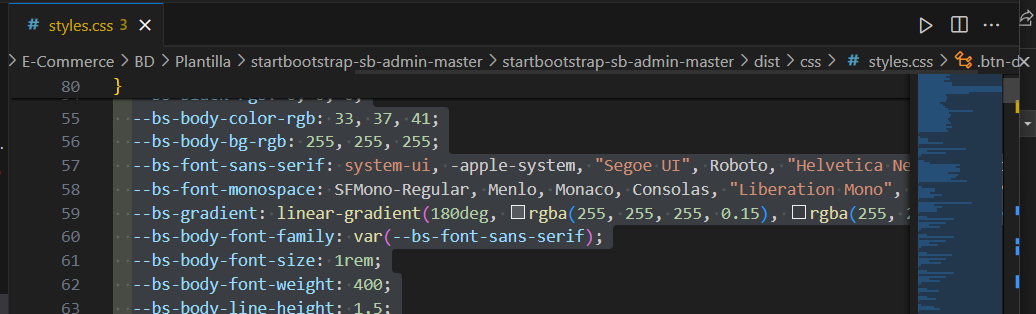


Una vez eliminado lo anterior, pegamos ahí dentro la referencia a la master page que habíamos hecho copia al final del documento, y debe quedar así:

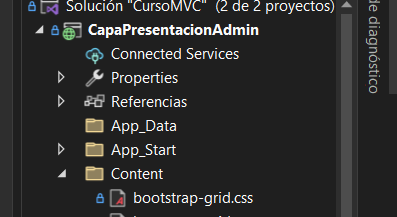


Ahora debemos copiar los estilos de la plantilla, para ello abrimos los estilos con el visual studio code y copiamos todo para posteriormente pegarlo en

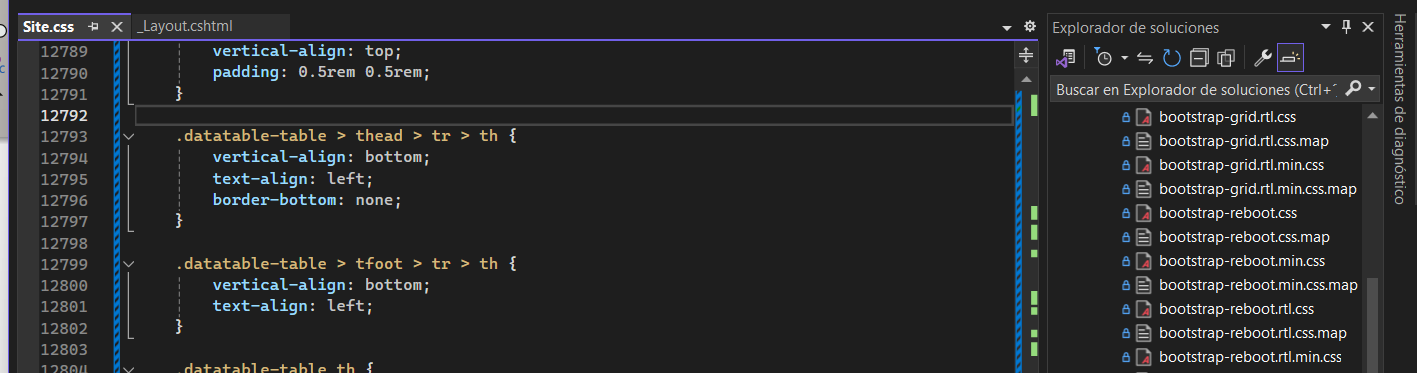




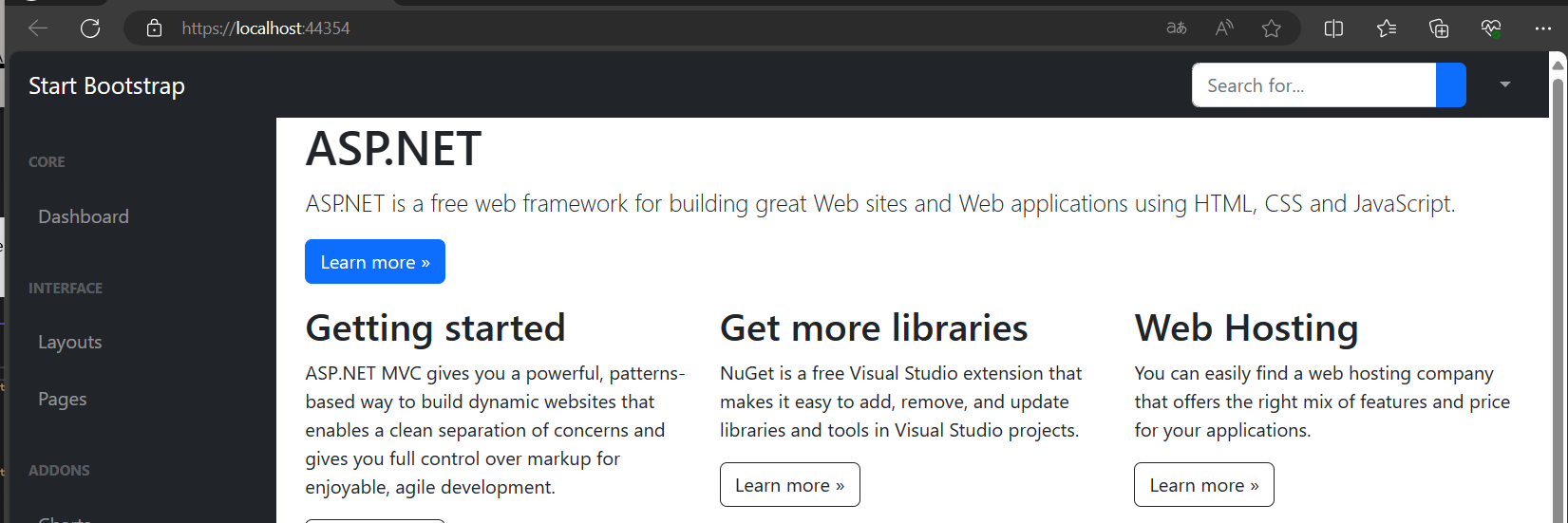
Lo pegamos en el archivo site.css que está dentro de la carpeta Content de CapaPresentacionAdmin

Borramos todo lo que haya dentro del archivo Site.css ya que son estilos por defecto, y pegamos todo lo que se extrajo del css de la plantilla, así:

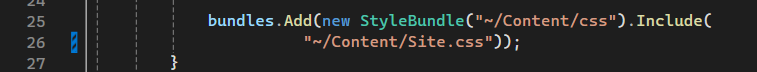


Si todo salió bien la apariencia de la aplicación debe quedar como la siguiente imagen,



en caso de que no sea así, debemos revisar la referencia que se hace en el BundleConfig.cs al final del documento, ya que debe aparecer así la S en mayúscula y no en minúscula:

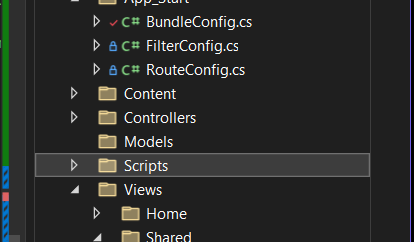
correcto: "~/Content/Site.css"));



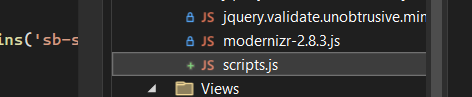
Incorrecto: "~/Content/site.css"));



Ahora debemos copiar el js de la plantilla al proyecto, para ello vamos a la carpeta contenedora de la plantilla, abrimos la carpeta js, y copiamos el archivo scripts.js ahora dentro del visual studio buscamos la carpeta Scritps, clic derecho y pegar.



Este ya seria el archivo pegado



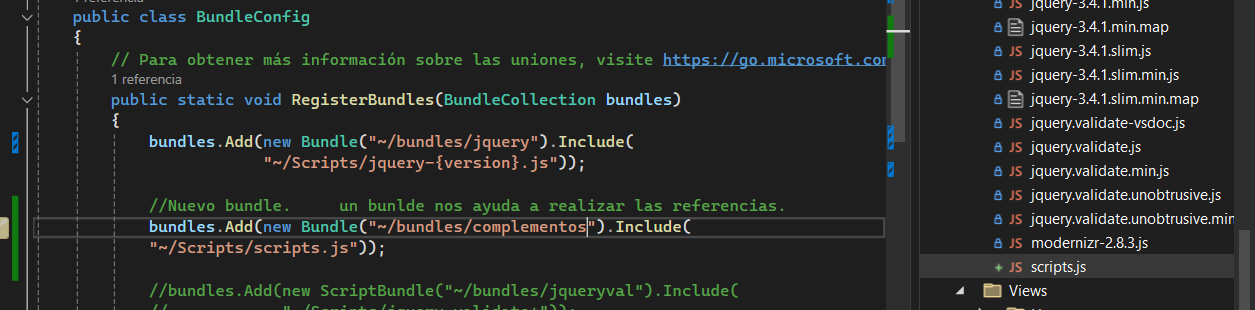
Ahora debemos hacerle una referencia a ese script con el bundle, para ello, dentro del archivo BundleConfig.cs debemos agregar un nuevo bundle, para hacer esto copiamos y pegamos uno que ya esté allí y lo modificamos, para que tenga este contenido:

//Nuevo bundle. un bunlde nos ayuda a realizar las referencias.

bundles.Add(new Bundle("~/bundles/complementos").Include(

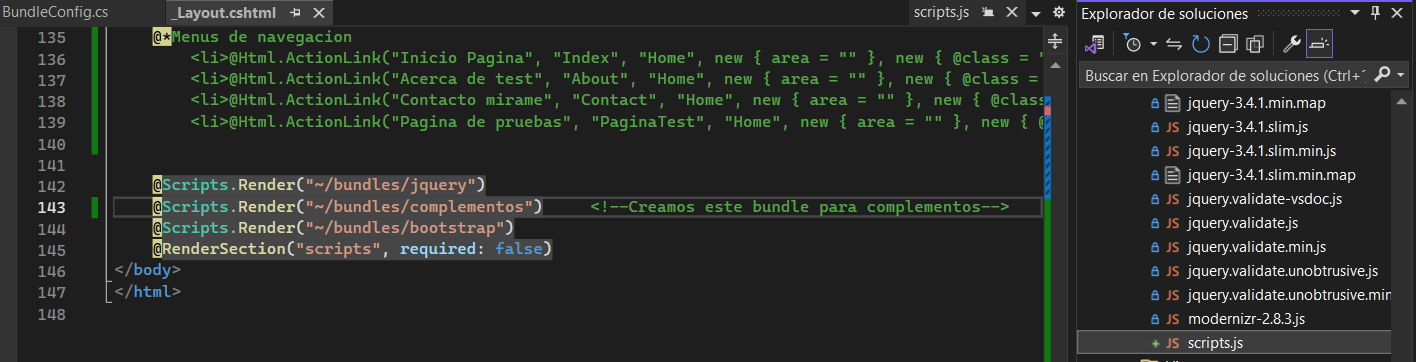
"~/Scripts/scripts.js"));

Con el */Scripts/scripts.js”));* lo estamos referenciando al js que trajimos de la plantilla, el cual lo podemos ver en esta imagen.

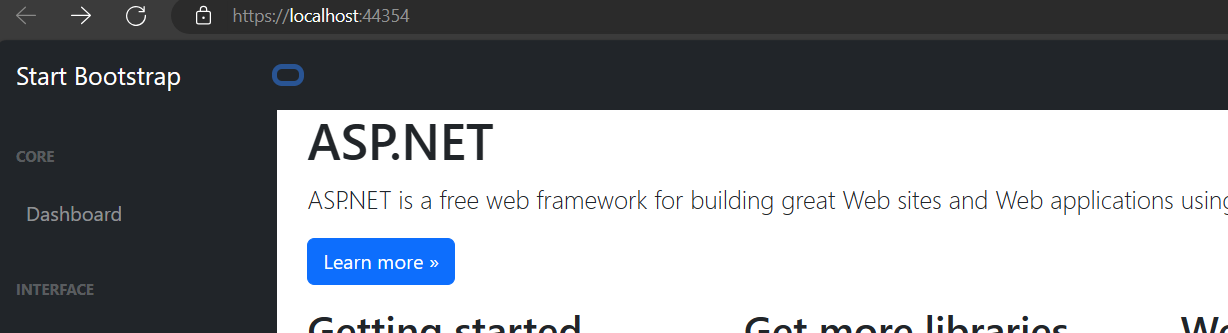


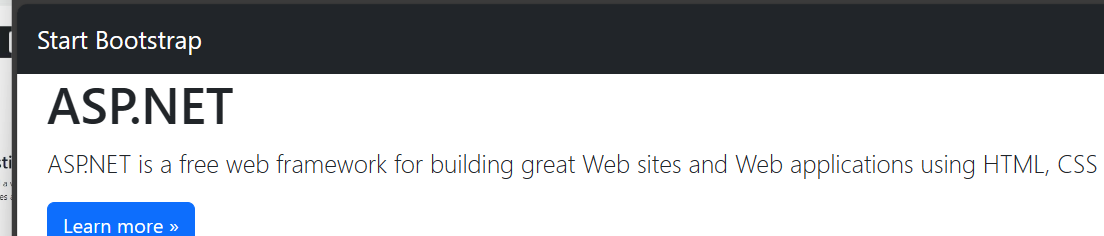
Ahora dentro del Layout, al final del documento creamos otro Script.Render para referenciar al “complementos” que acabamos de crear, este debe tener esta estructura:

@Scripts.Render("~/bundles/complementos")

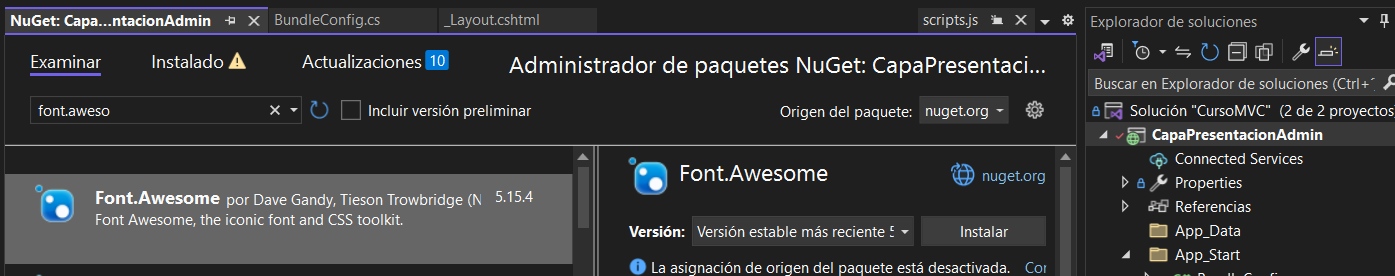


Con esto ya tendríamos activo el botón que oculta o muestra la barra del lateral izquierdo:





Para arreglar el tema de los íconos, perfil y similares, debemos ir sobre la CapaPresentacionAdmin, clic derecho, Administrar Paquetes de NuGet…, Examinar, buscamos: “Font.Awesome”, y lo instalamos.



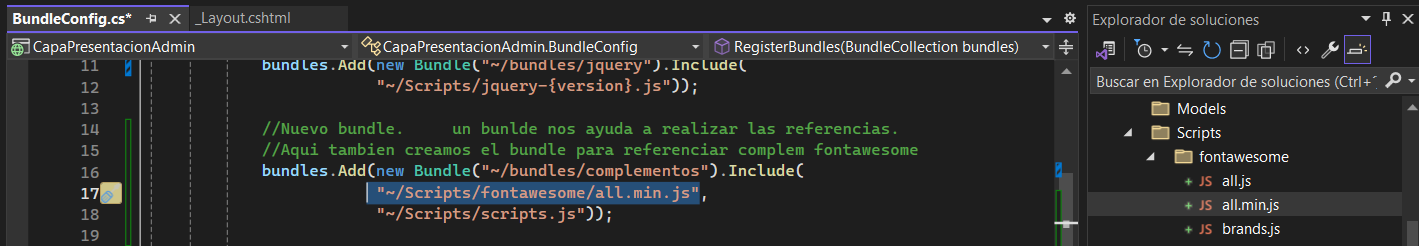
Ahora dentro del BundleConfig.cs debemos crear una referencia mas para complementos, esto lo hacemos dentro de la misma que ya habíamos creado para referenciar el .js de la plantilla; ya modificado se vería así;

bundles.Add(new Bundle("~/bundles/complementos").Include(

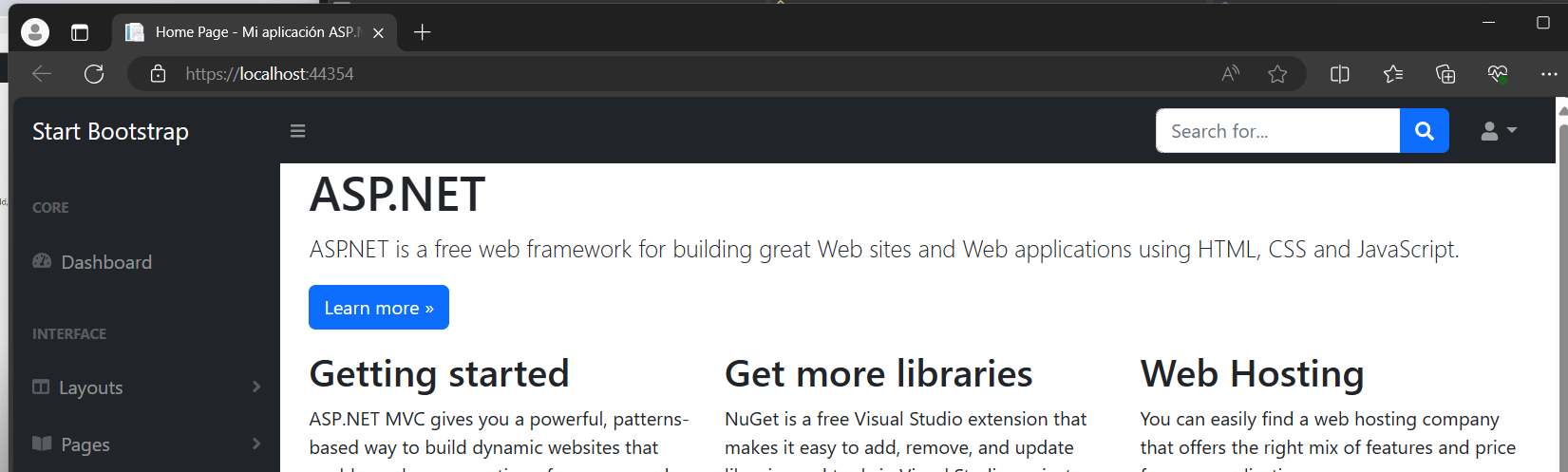
"~/Scripts/fontawesome/all.min.js",

"~/Scripts/scripts.js"));

Lo amarillo es lo que creamos en esta ocasión. El FontAwesome quedo instalado en Scripts, Fontawesome, y solo nos interesa referenciar el all.min.js



Ejecutando la app, ya se podremos ver los iconos y similares, que anteriormente no estaban



Si ingreso a otra de las paginas que creamos (Pagina test) veremos que se mantiene la master page, con sus estilos y sus iconos.

