Herramientas de Programación 2

Tema Nº2:ASP.NET CORE MVC

Indicador de logro Nº2:Construye la estructura de un proyecto web usando las partes de la estructura del proyecto aprendido.

**TEMA 01 Teoría de los**

Imagen que contiene Icono

Descripción generada automáticamente

**TEMA Nº2:**

ASP.NET Core MVC

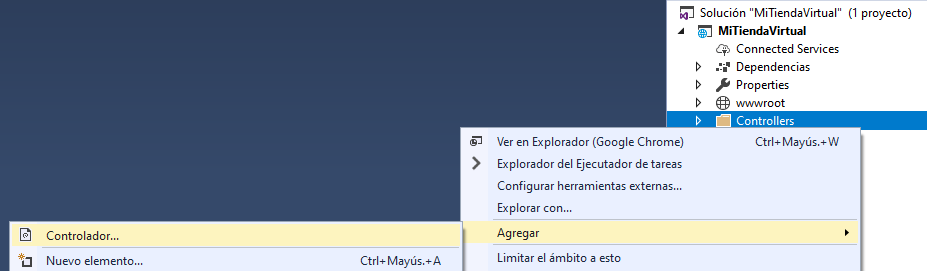
**Subtema 2.1:**

EL CONTROLADOR

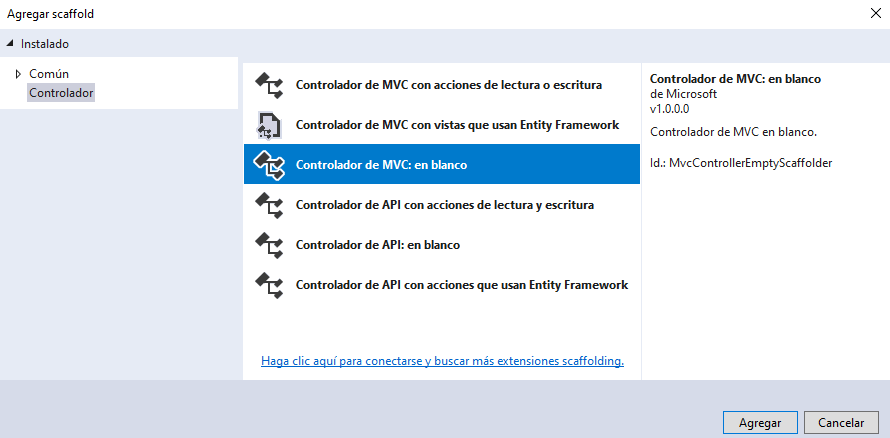
La capa del controlador gestiona las peticiones de los usuarios. Es responsable de responder la información solicitada con la ayuda tanto del Modelo como de la Vista.

Paso 1. Crear Controladores.

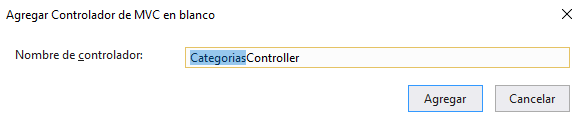
1. Abrimos la solución creada en el taller anterior.
2. Hacemos click derecho en la carpeta ***Controllers*** y seleccionamos la opción: ***Agregar > Controlador…***



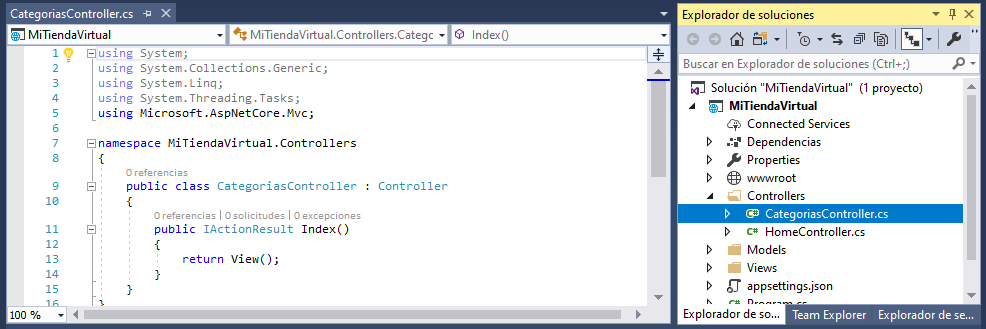
1. Seleccionamos la opción ***Controlador de MVC*** en blanco y presionamos en el botón ***Agregar***.



1. Le ponemos el nombre ***CategoriasControlle***r y presionamos en el botón ***Agregar***.



1. Observamos que se ha creado el archivo ***CategoriasController.cs***



1. Modificamos el código del método ***Index()*** de acuerdo a:

public class CategoriasController : Controller

{

//Url: dominio:puerto/Categorias

//Url: dominio:puerto/Categorias/Index

public String Index()

{

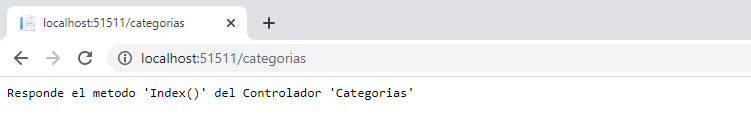
String mensaje = "Responde el metodo 'Index()' del Controlador 'Categorias'";

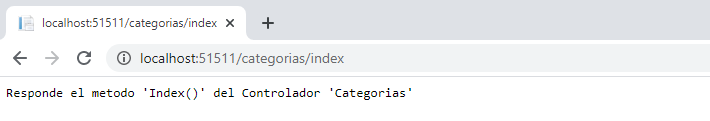
return mensaje;

}

}

1. Ejecutamos la aplicación, cambiamos la Url para que responda el método que acabamos de modificar.





1. Al mismo Controlador le agregamos el método ***MostrarCategoria()*** de acuerdo a:

//GET

//dominio:puerto/Categorias/MostrarCategoria/123

public String MostrarCategoria(int Id)

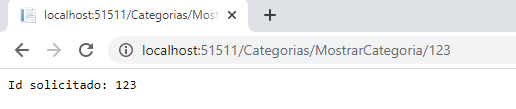
{

String mensaje = "Id solicitado: " + Id.ToString();

return mensaje;

}

1. Ejecutamos la aplicación.



**Subtema 2.2:**

EL MODELO

Es la representación de la información con la cual el sistema opera, por lo tanto, gestiona todos los accesos a dicha información, tanto las consultas como actualizaciones, implementando también los privilegios de acceso que se hayan descrito en las especificaciones de la aplicación (lógica de negocio).

Paso 2. Gestionar datos desde el Modelo.

1. Modificamos el código del método ***MostrarCategoria()*** de acuerdo a:

public String MostrarCategoria(int Id)

{

String mensaje = "Categoria solicitada:\n\n";

using (TiendaVirtualDBContext BD = new TiendaVirtualDBContext())

{

var listaCategorias = (from c in BD.Categoria

where c.Id == Id

select new Categoria

{

Id = c.Id,

Nombre = c.Nombre,

Activo = c.Activo

}).ToList();

if (listaCategorias.Count > 0)

{

Categoria categoria = listaCategorias.First();

String activo = categoria.Activo.GetValueOrDefault() ? "Si" : "No";

mensaje += "Id : " + categoria.Id.ToString() + "\n";

mensaje += "Nombre: " + categoria.Nombre + "\n";

mensaje += "Activo: " + activo;

}

else

{

mensaje = "El Producto no existe en la BD.";

}

}

return mensaje;

}

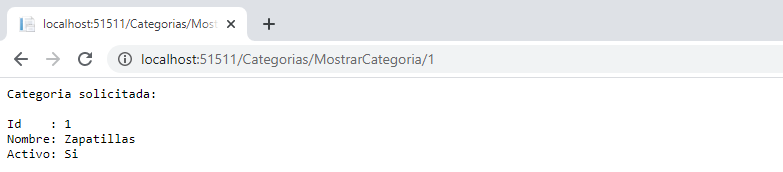
1. En la parte superior del archivo del controlador agregamos:

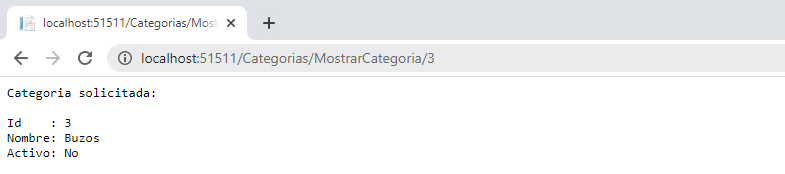
using MiTiendaVirtual.Models;

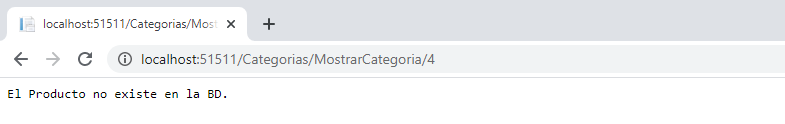
1. Antes de ejecutar la aplicación verificamos las categorías que tenemos en la BD.



1. Ejecutamos la aplicación.



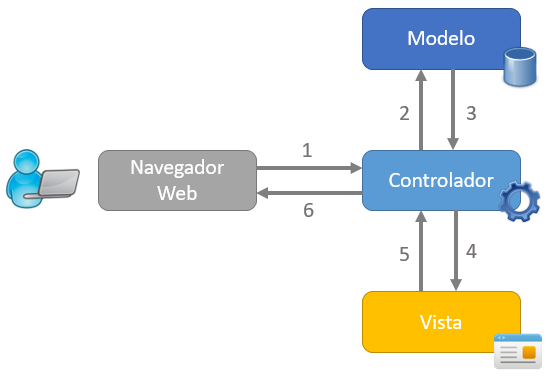




**Subtema 2.3:**

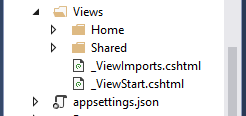
LA VISTA

Presenta el Modelo (información y lógica de negocio) en un formato adecuado para interactuar (usualmente la interfaz de usuario) por tanto requiere de dicho Modelo la información que debe representar como salida.

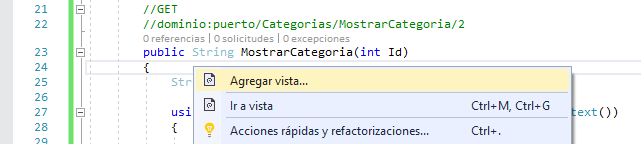


Paso 3. Diseñar Vistas con datos solicitados y devolverlas al usuario.

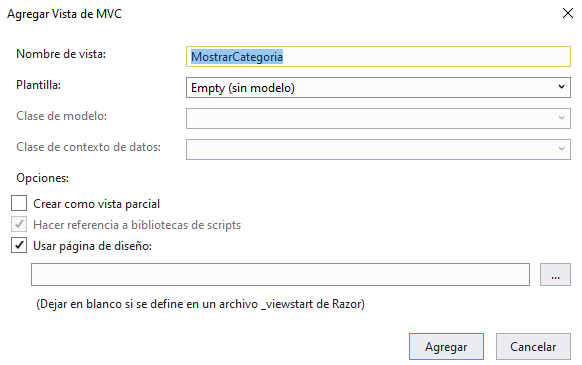
1. Primero verificamos que no existen Vistas para ningún método del Controlador ***Categorias***.



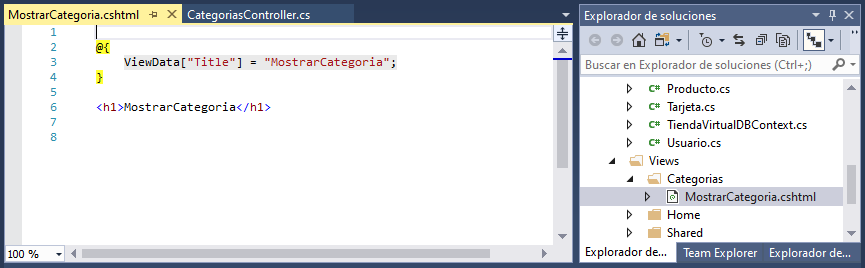
1. Vamos a crear una vista para el método ***MostrarCategoria().***
2. Hacemos click derecho dentro del código del método ***MostrarCategoria()*** y seleccionamos la opción: ***Agregar vista…***



1. En el formulario ingresamos las opciones tal cual se muestra en la figura y presionamos en el botón ***Agregar***.



1. Observamos que la Vista ha sido creada.



1. Actualizamos el código de la Vista de acuerdo a:

@model MiTiendaVirtual.Models.Categoria

@{

ViewData["Title"] = "MOSTRAR CATEGORIA";

}

<div>

<h4>CATEGORIA</h4>

<hr />

<dl class="row">

<dt class="col-sm-2">Id</dt>

<dd class="col-sm-10">@Html.DisplayFor(model => model.Id)

</dd>

</dl>

<dl class="row">

<dt class="col-sm-2">Nombre</dt>

<dd class="col-sm-10">@Html.DisplayFor(model => model.Nombre)

</dd>

</dl>

<dl class="row">

<dt class="col-sm-2">Id</dt>

<dd class="col-sm-10">@ViewData["activo"]

</dd>

</dl>

</div>

1. Modificamos el código del método MostrarCategoria() de acuerdo a:

public IActionResult MostrarCategoria(int Id)

{

Categoria categoria = new Categoria();

using (TiendaVirtualDBContext BD = new TiendaVirtualDBContext())

{

var listaCategorias = (from c in BD.Categoria

where c.Id == Id

select new Categoria

{

Id = c.Id,

Nombre = c.Nombre,

Activo = c.Activo

}).ToList();

if(listaCategorias.Count > 0)

{

categoria = listaCategorias.First();

ViewData["activo"] = categoria.Activo.GetValueOrDefault() ? "Si" : "No";

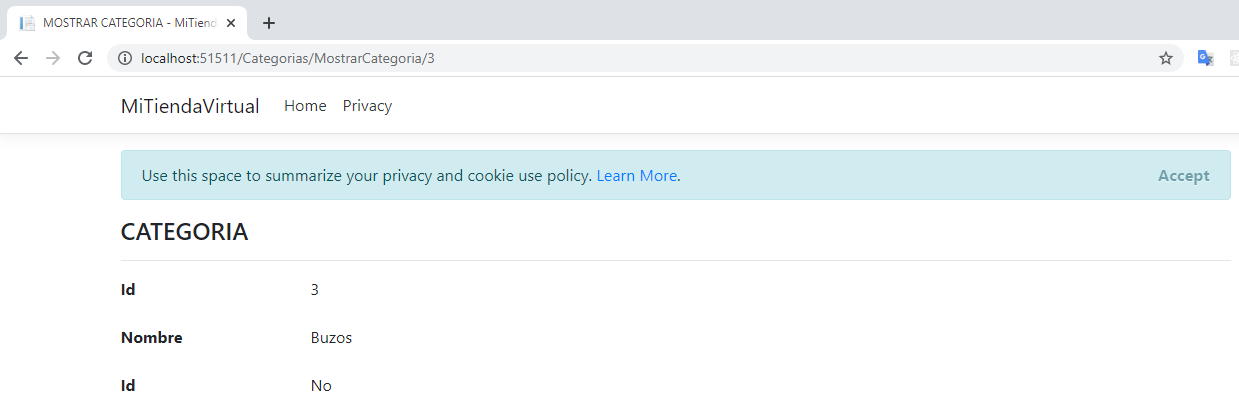
}

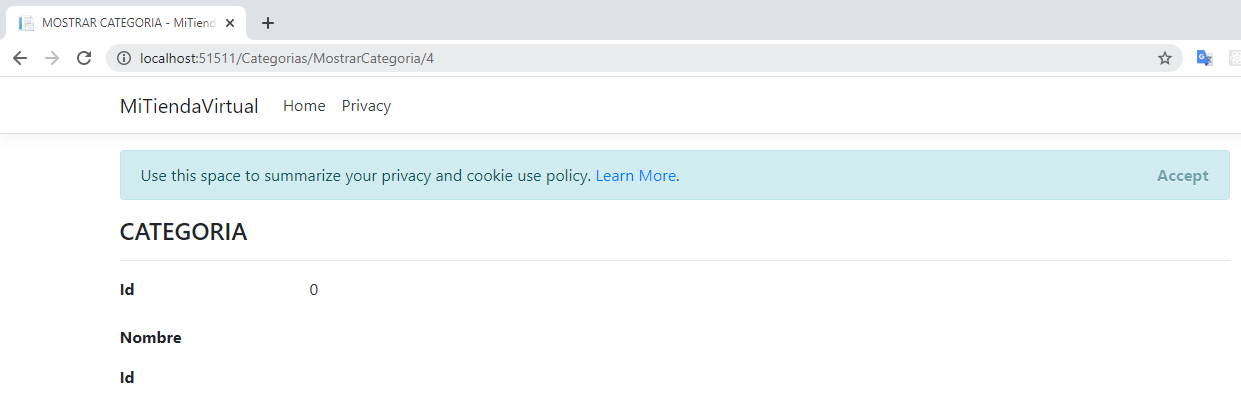
}

return View(categoria);

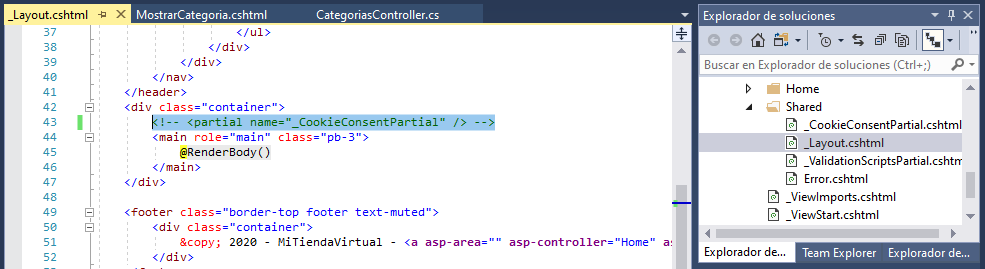
}

1. Ejecutamos la aplicación:

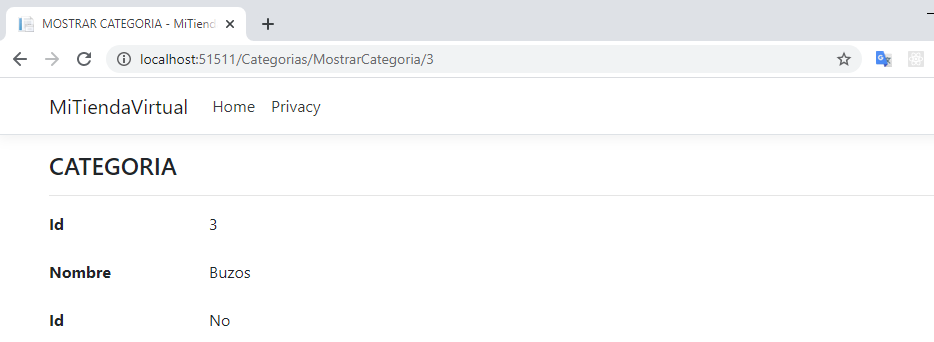


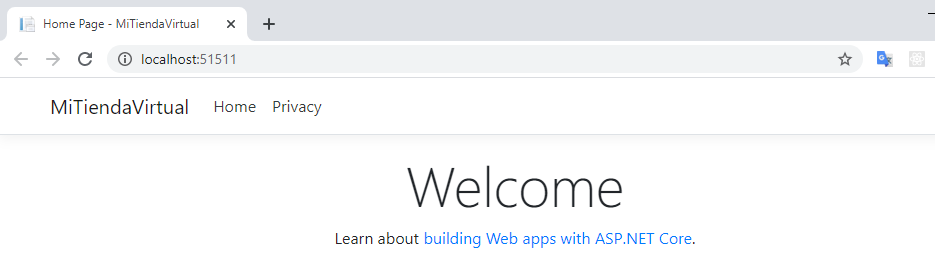


1. Vamos a eliminar el mensaje que se encuentra debajo de la cabecera (la que hace mención a las cookies)
2. Abrimos el archivo ***\_Layout.cshtml.***
3. Comentamos el código de la línea Nro 43.



1. Ejecutamos la aplicación.

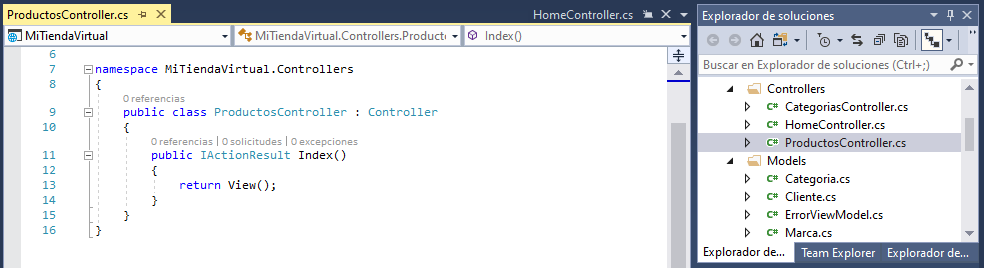




1. Verificamos que el mensaje ya no se muestra en ninguna de las páginas de la aplicación.

Paso 4. Ejercicio: Mostrar la lista de Productos.

1. Creamos el controlador ***ProductosController.cs.***



1. Actualizamos el código del método ***Index()*** de acuerdo a:

public class ProductosController : Controller

{

//GET

//dominio:puerto/Productos

//dominio:puerto/Productos/Index

public IActionResult Index()

{

List<Producto> listaProductos = new List<Producto>();

using (TiendaVirtualDBContext BD = new TiendaVirtualDBContext())

{

listaProductos = (from p in BD.Producto select p).ToList();

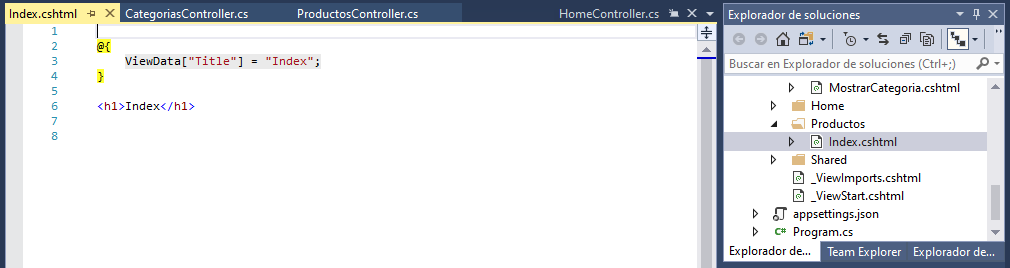
}

return View(listaProductos);

}

}

1. Creamos una vista para el método Index().



1. Actualizamos el código de la Vista de acuerdo a:

@model IEnumerable<MiTiendaVirtual.Models.Producto>

@{

ViewData["Title"] = "PRODUCTOS";

}

<h1>Lista de Productos</h1>

<table class="table">

<thead>

<tr>

<th>Id</th>

<th>Nombre</th>

<th>Precio</th>

<th>Destacado</th>

<th>Activo</th>

</tr>

</thead>

<tbody>

@foreach (var item in Model)

{

<tr>

<td>@item.Id</td>

<td>@item.Nombre</td>

<td>@item.Precio</td>

<td>@item.Destacado</td>

<td>@Html.DisplayFor(modelItem => item.Activo)</td>

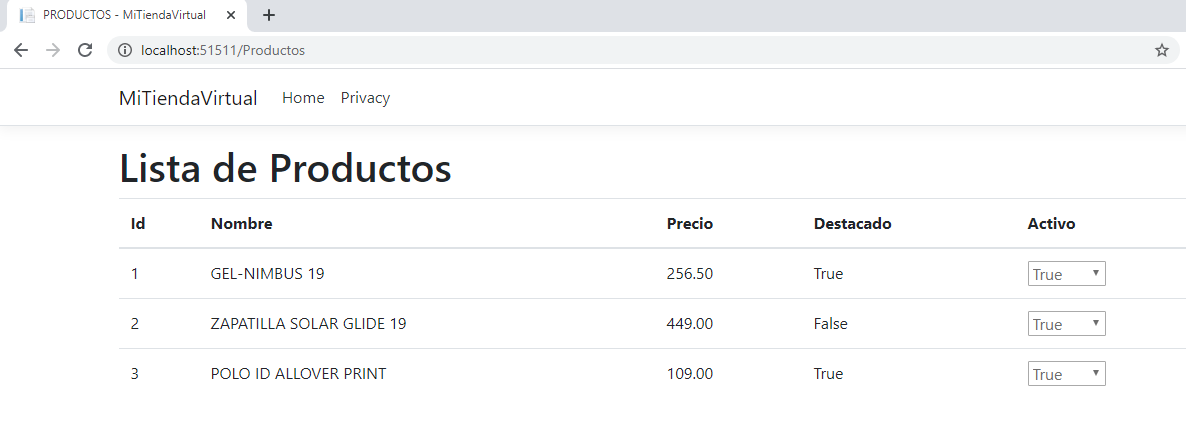
</tr>

}

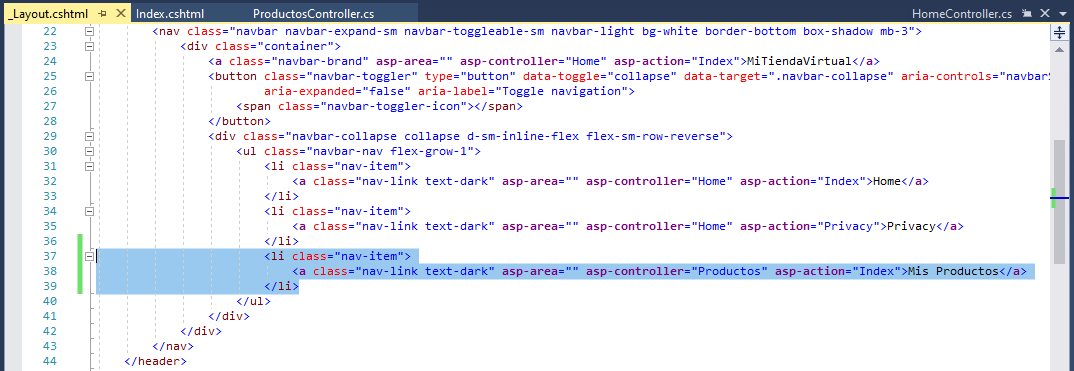
</tbody>

</table>

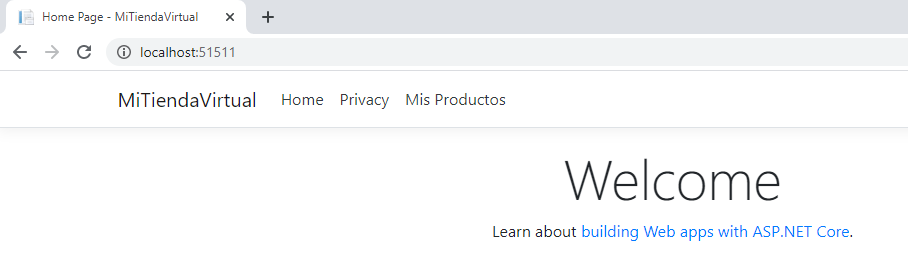
1. Ejecutamos la aplicación.



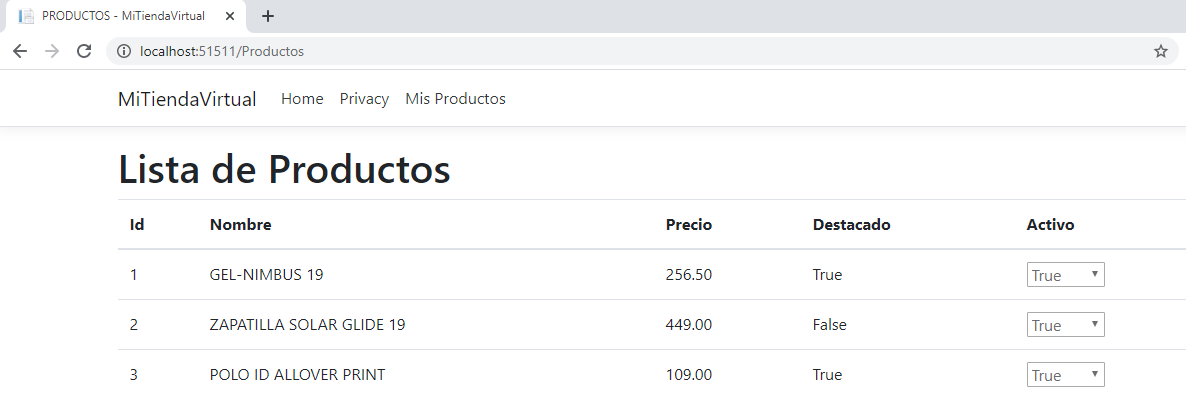
1. Vamos agregar una opción en el menú que nos permita dirigir a la lista de Productos.
2. Abrimos el archivo ***\_Layout.cshtml*** y agregamos la tres líneas que se muestran:



1. Ejecutamos la aplicación.



1. Presionamos en la opción de menú ***Mis Productos.***



**ACTIVIDAD VIRTUAL:**

Ingresa a la plataforma virtual. Revisa y analiza el tema desarrollado en esta sesión, luego desarrolla las siguientes actividades propuestas:

1. CUESTIONARIO TÉCNICO
   * 1. Crear un Controlador (con su respectiva Vista) que permita mostrar los Pedidos que existen en la BD. Debe tener las siguientes columnas:

* Id del Pedido
* Fecha (sin hora) del Pedido
* Monto total del Pedido
* Estado del Pedido

Ordenar ascendentemente por el campo Fecha del Pedido.

* + 1. Crear un Controlador (con su respectiva Vista) que permita mostrar las Tarjetas que existen en la BD. Debe tener las siguientes columnas:
* Id de la Tarjeta
* Marca
* Numero

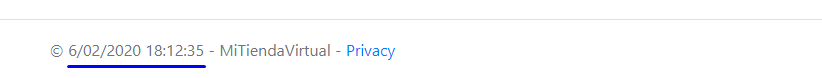
Ordenar ascendentemente por el campo Marca.

* + 1. Crear un Controlador (con su respectiva Vista) que permita mostrar los Clientes que existen en la BD. Debe tener las siguientes columnas:
* Id del Cliente
* Nombres y Apellidos (concatenado)
* Dni
* Teléfono
* Correo
* Dirección

Ordenar ascendentemente por el campo Nombres y Apellidos.

* + 1. Modificar el pie de página de tal forma que en vez del año se muestre la fecha y hora del servidor web.





* + 1. Investigar la forma de agregar en la lista de Productos la columna Categoría, esta columna debe mostrar el nombre de la Categoría a la que pertenece cada Producto.



1. CONCLUSIONES DE LA EXPERIENCIA

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_