Herramientas de Programación 2

Tema Nº3:CONTROLADOR MVC (PARTE 1)

Indicador de logro Nº3:Desarrolla las aplicaciones web con Scaffolding usando las partes del proyecto aprendido.

**TEMA 01 Teoría de los**

Imagen que contiene Icono

Descripción generada automáticamente

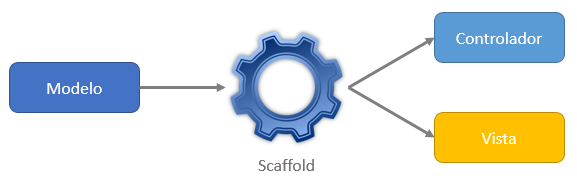
**TEMA Nº3:**

Controlador MVC (PARTE 1)

**Subtema 3.1:**

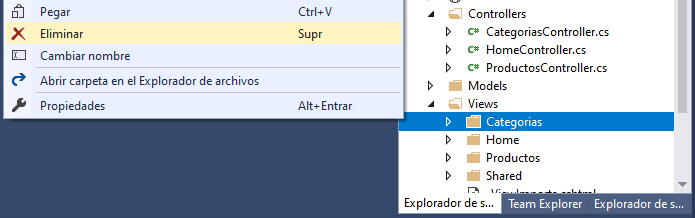
Scaffold

ASP.NET ***Scaffold*** es un framework de generación de código para aplicaciones web ASP.NET. Desde Visual Studio 2013 se incluye generadores de código preinstalados para proyectos web MVC y API. Se debe utilizar ***Scaffold*** en su proyecto cuando desea agregar rápidamente código que interactúa con el Modelo de datos. El uso de ***Scaffold*** puede reducir la cantidad de tiempo para desarrollar operaciones de datos estándar en su proyecto.

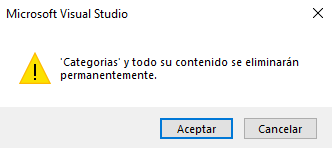


Paso 1. Acondicionar el proyecto.

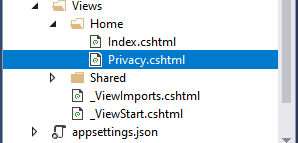
1. Abrimos la solución creada en el taller anterior.
2. Vamos a limpiar el proyecto, esto es, eliminar las vistas y controladores que creamos en el taller de la semana anterior y otros que se crearon por defecto y que no los usaremos.
3. Dentro de la carpeta ***Views*** hacemos click derecho en la carpeta ***Categorías*** y seleccionamos la opción ***Eliminar***.



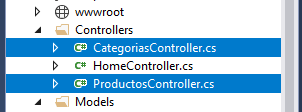
1. En mensaje de advertencia presionamos el botón ***Aceptar***.



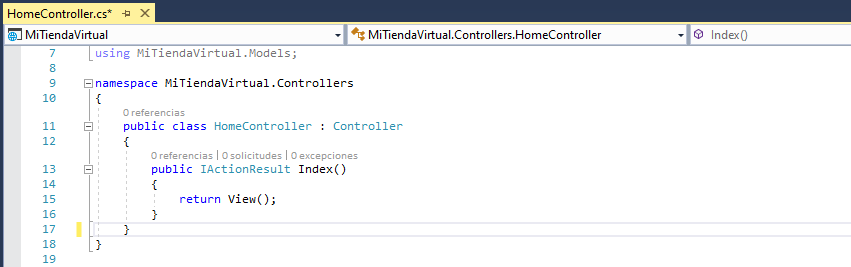
1. Del mismo modo eliminamos la carpeta ***Productos***.
2. Abrimos la carpeta ***Home***, dentro de ella eliminamos el archivo ***Privacy.cshtml***.



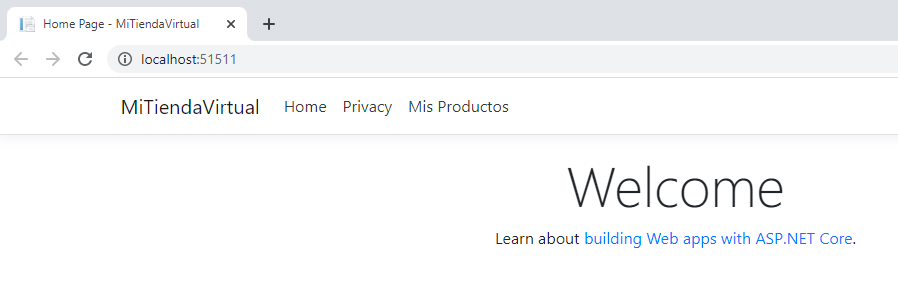
1. Dentro de la carpeta ***Controllers*** eliminamos los archivos ***CategoriasController.cs***.y ***ProductosController.cs.***



1. Abrimos el archivo ***HomeController.cs*** y eliminamos las funciones ***Privacy()*** y ***Error(),*** el archivo debe quedar de la siguiente forma:



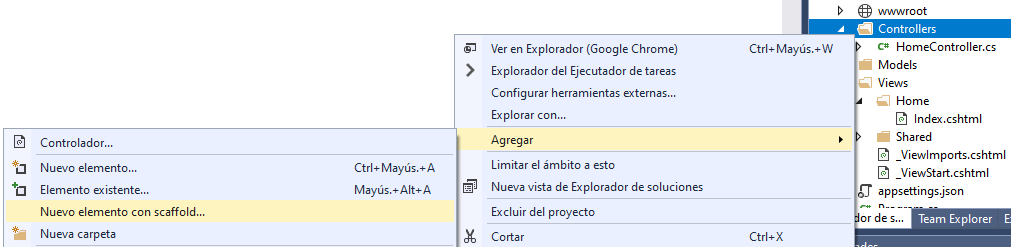
1. Ejecutamos la aplicación para verificar que no existen errores.



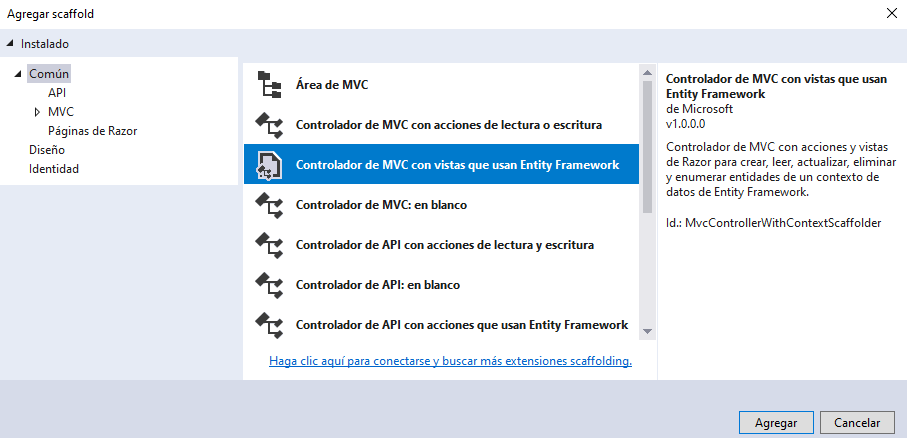
Paso 2. CRUD de la entidad (tabla) Usuario.

1. Vamos a generar, de forma automática, El Controlador y las Vistas que nos permitan hacer el mantenimiento de la taba Usuario.
2. Hacemos click derecho en la carpeta ***Controllers*** y seleccionamos la opción:

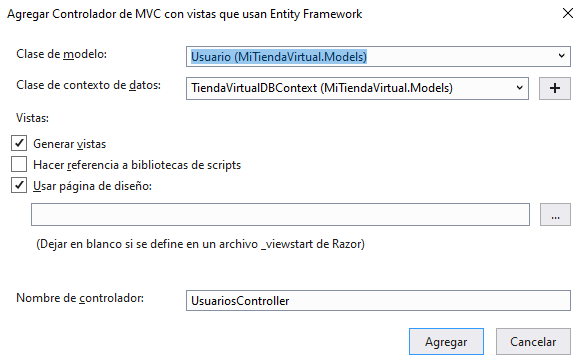
Agregar > Nuevo elemento con ***Scaffold…***



1. En el formulario siguiente seleccionar la opción ***Controlador de MVC*** con vistas que usan ***Entity Framework*** y presionar en el botón ***Aceptar***.



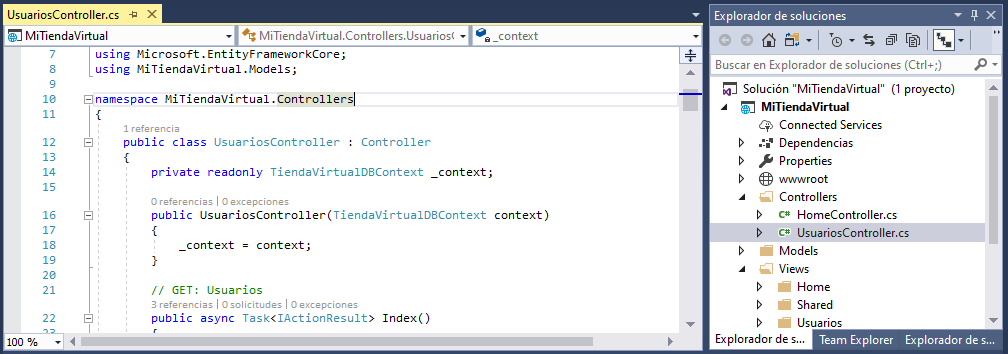
1. En el formulario siguiente seleccionamos la clase Usuario del Modelo, el controlador a crear tendrá el nombre ***UsuariosController*** (este valor es modificable). Dejamos por ahora las demás opciones como se muestran.



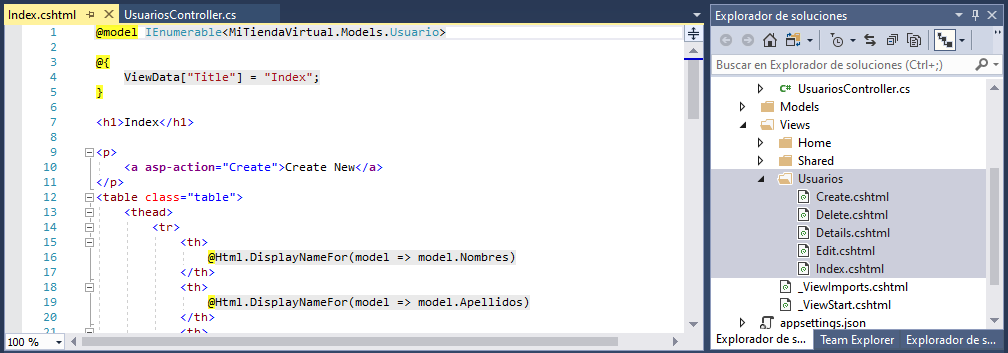
1. Presionamos el botón ***Agregar***. Scaffold empieza a generar el Controlador ***UsuariosController*** y sus vistas asociadas.



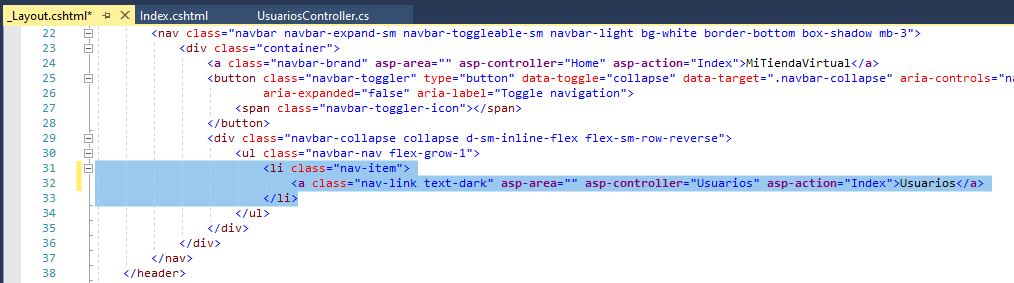
1. Verificamos que se ha creado el Controlador ***UsuariosController.cs***



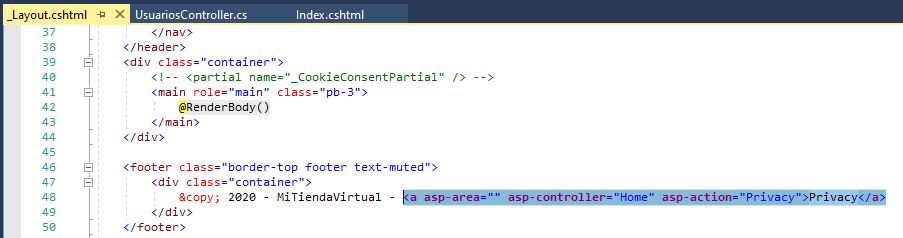
1. También verificamos que se han creado las Vistas para este nuevo Controlador.



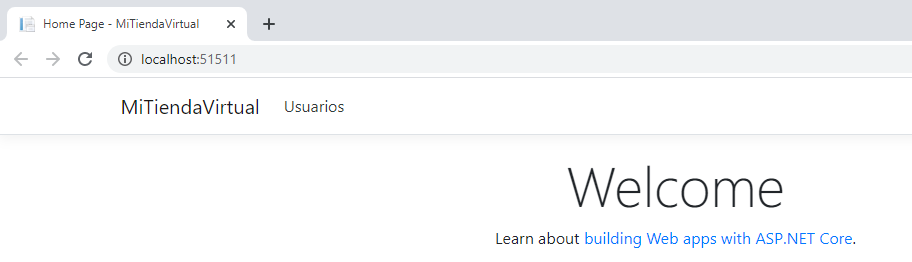
1. Abrimos el archivo ***\_Layout.cshtml,*** borramos las tres opciones de menú y agregamos uno nuevo que nos permita ver la página inicial del mantenimiento de ***Usuarios***.



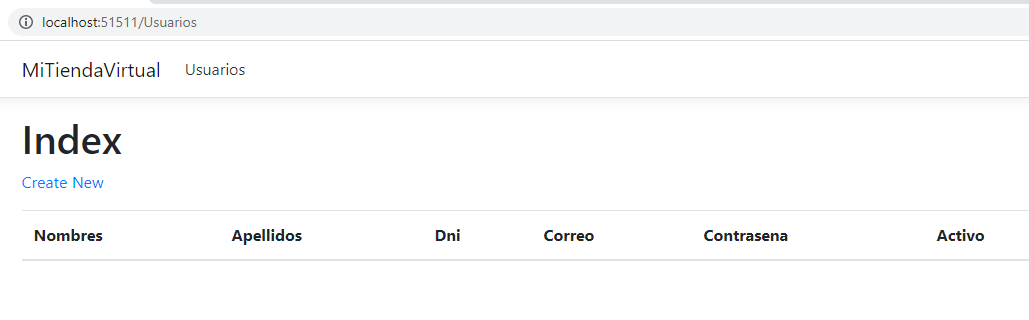
1. Borramos el link del pie de página ***(Privacy)***



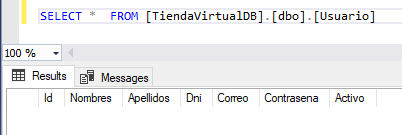
1. Ejecutamos la aplicación.



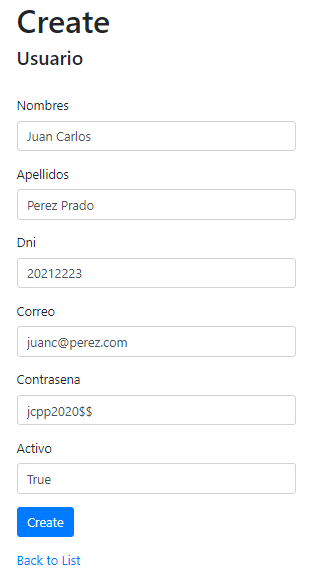
1. Seleccionamos la opción de menú ***Usuarios***.



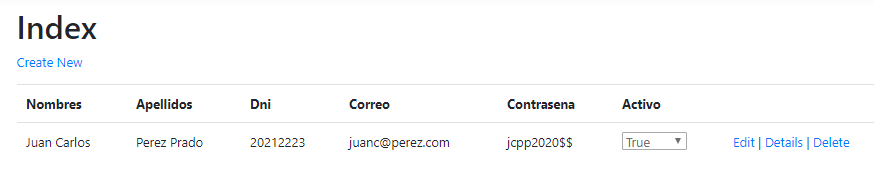
1. Observamos que no existen usuarios, verificamos con la BD.



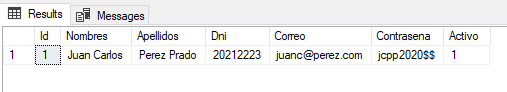
1. Más adelante vamos a hacer personalizaciones a las Vistas de tal forma que se acondicionen mejor a la funcionalidad de nuestro proyecto.
2. Para crear un nuevo Usuario hacemos click en el link ***Create New.***



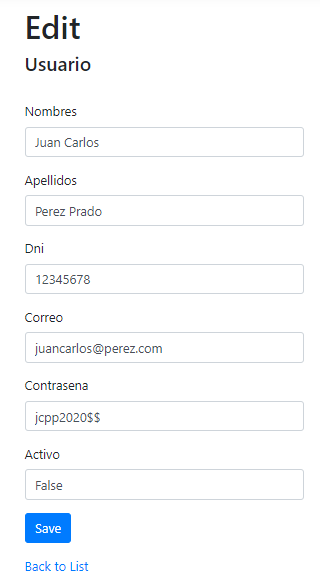
1. Presionamos en el botón ***Create***.
2. Inmediatamente se muestra en la lista de usuarios el que acabamos de crear.



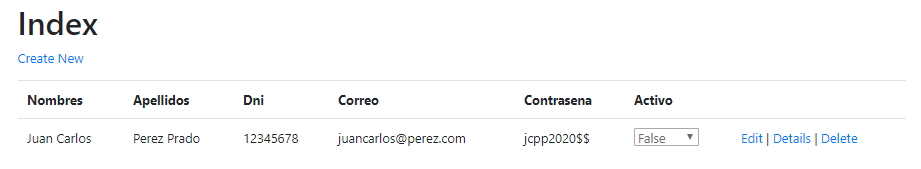
1. Verificamos en la BD.



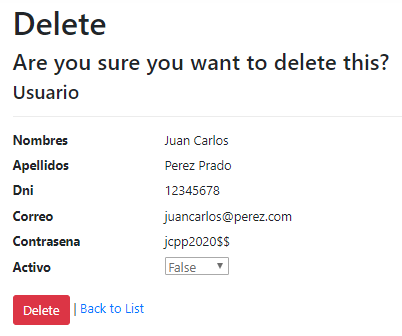
1. Para actualizar los datos del Usuario presionamos en el link Edit.
2. Modificamos alguno o todos los datos.



1. Presionamos en el botón ***Save***.



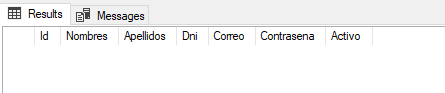
1. Para eliminar el Usuario hacemos click en el enlace ***Delete***.
2. Se mostrará una página de confirmación.



1. Si desistimos en eliminar el registro hacemos click en el link ***Back to List*** y regresaremos a la página anterior.
2. Para eliminar el registro presionamos en el botón ***Delete***.
3. Ya no se muestra el Usuario eliminado.

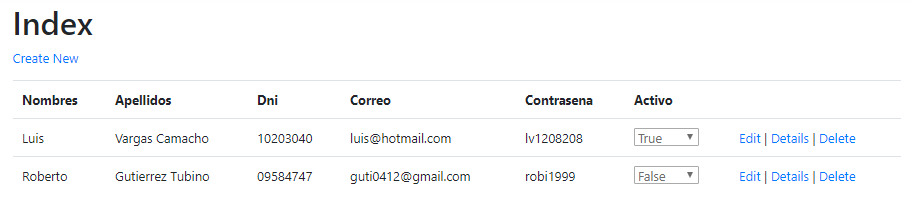


1. Verificamos en la BD.

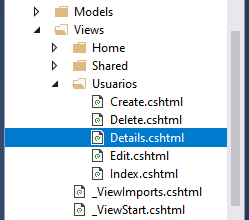


Paso 3. Personalizaciones. Eliminar la funcionalidad Details.

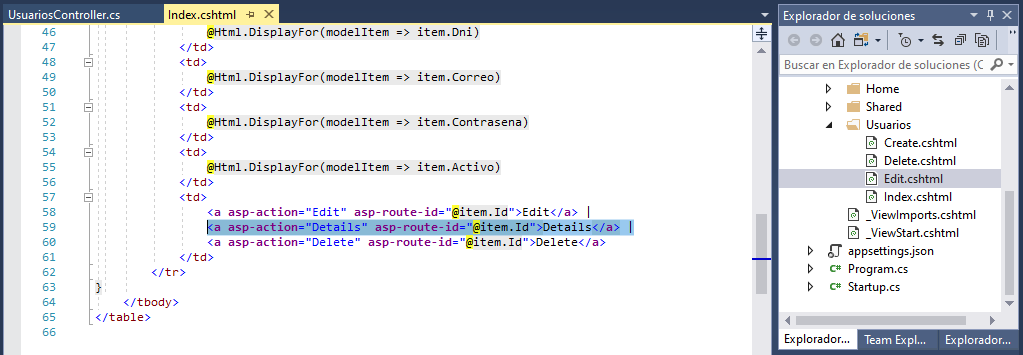
1. Por medio de la aplicación web agregamos dos usuarios.



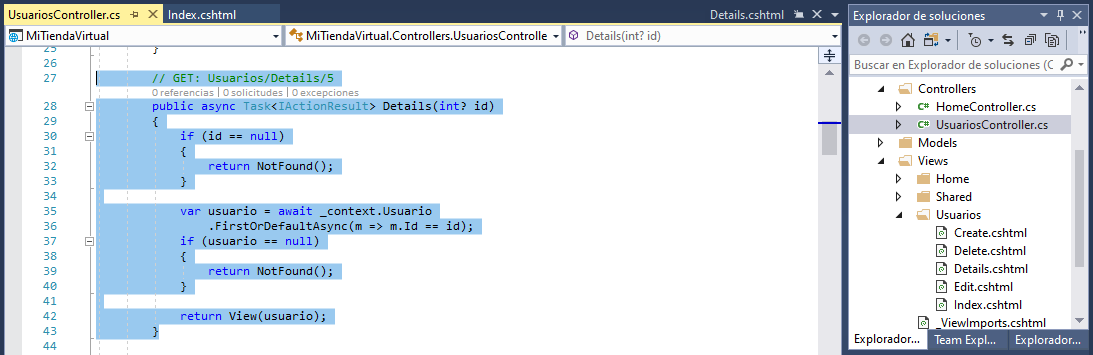
1. Cerramos la aplicación para hacer los cambios.
2. Eliminamos el archivo ***Details.cshtml.***



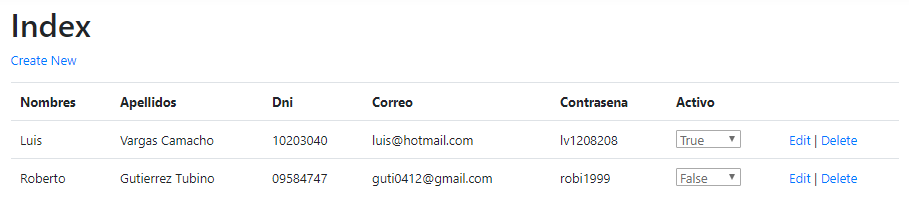
1. Abrimos el archivo ***Index.cshtml*** y borramos el código que esta seleccionado (línea 59)



1. Abrimos el archivo ***UsuariosController.cs*** y borramos el código seleccionado (que corresponde a la función ***Details())***



1. Ejecutamos la aplicación para validar los cambios.



Paso 4. Personalizaciones. Etiquetas y links.

1. Abrir el archivo ***Index.cshtml*** y cambiar el código de acuerdo a:

@model IEnumerable<MiTiendaVirtual.Models.Usuario>

@{

ViewData["Title"] = "Usuarios";

}

<h4>Usuarios</h4>

<p>

<**a** **asp-action**="Create">Nuevo Usuario</**a**>

</p>

<table class="table table-hover table-sm">

<thead class="thead-dark">

<tr>

<th>

@Html.DisplayNameFor(model => model.Nombres)

</th>

<th>

@Html.DisplayNameFor(model => model.Apellidos)

</th>

<th>

@Html.DisplayNameFor(model => model.Dni)

</th>

<th>

@Html.DisplayNameFor(model => model.Correo)

</th>

<th>

@Html.DisplayNameFor(model => model.Activo)

</th>

<th></th>

</tr>

</thead>

<tbody>

@foreach (var item in Model) {

<tr>

<td>

@Html.DisplayFor(modelItem => item.Nombres)

</td>

<td>

@Html.DisplayFor(modelItem => item.Apellidos)

</td>

<td>

@Html.DisplayFor(modelItem => item.Dni)

</td>

<td>

@Html.DisplayFor(modelItem => item.Correo)

</td>

<td>

<input type="checkbox" checked="@item.Activo.GetValueOrDefault()" disabled>

</td>

<td>

<**a** **asp-action**="Edit" **asp-route-id**="@item.Id">Actualizar</**a**> |

<**a** **asp-action**="Delete" **asp-route-id**="@item.Id">Eliminar</**a**>

</td>

</tr>

}

</tbody>

</table>

<hr />

1. Ejecutamos la aplicación y validamos los cambios.



1. Abrir el archivo ***Create.cshtml*** y cambiar el código de acuerdo a:

@model MiTiendaVirtual.Models.Usuario

@{

ViewData["Title"] = "Nuevo Usuario";

}

<h4>Nuevo Usuario</h4>

<hr />

<div class="row">

<div class="col-md-4">

<**form** **asp-action**="Create">

<**div** **asp-validation-summary**="ModelOnly" class="text-danger"></**div**>

<div class="form-group">

<**label** **asp-for**="Nombres" class="control-label"></**label**>

<**input** **asp-for**="Nombres" class="form-control" />

<**span** **asp-validation-for**="Nombres" class="text-danger"></**span**>

</div>

<div class="form-group">

<**label** **asp-for**="Apellidos" class="control-label"></**label**>

<**input** **asp-for**="Apellidos" class="form-control" />

<**span** **asp-validation-for**="Apellidos" class="text-danger"></**span**>

</div>

<div class="form-group">

<**label** **asp-for**="Dni" class="control-label"></**label**>

<**input** **asp-for**="Dni" class="form-control" />

<**span** **asp-validation-for**="Dni" class="text-danger"></**span**>

</div>

<div class="form-group">

<**label** **asp-for**="Correo" class="control-label"></**label**>

<**input** **asp-for**="Correo" class="form-control" />

<**span** **asp-validation-for**="Correo" class="text-danger"></**span**>

</div>

<div class="form-group">

<**label** **asp-for**="Contrasena" class="control-label">Contraseña</**label**>

<**input** **asp-for**="Contrasena" class="form-control" **type**="password" />

<**span** **asp-validation-for**="Contrasena" class="text-danger"></**span**>

</div>

<div class="form-group">

<**label** **asp-for**="Activo" class="control-label"></**label**>

@Html.CheckBox("Activo")

</div>

<div class="form-group">

<input type="submit" value="Guardar" class="btn btn-primary" /> |

<**a** **asp-action**="Index">Cancelar</**a**>

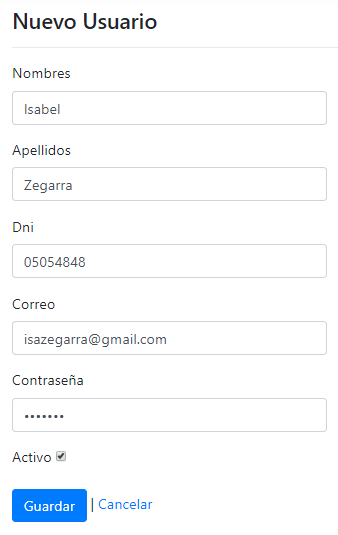
</div>

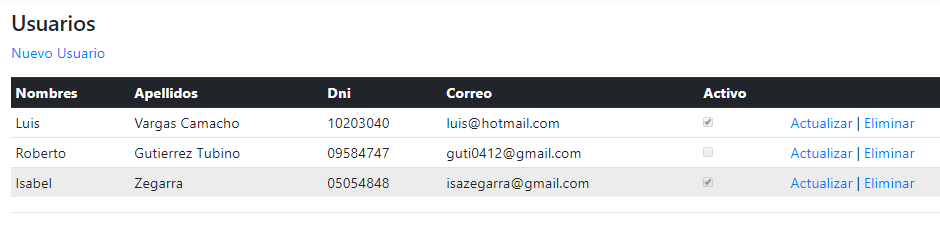
</**form**>

</div>

</div>

1. Ejecutar la aplicación y crear un nuevo Usuario para validar los cambios.





1. Abrir el archivo ***Edit.cshtml*** y cambiar el código de acuerdo a:

@model MiTiendaVirtual.Models.Usuario

@{

ViewData["Title"] = "Actualizar Usuario";

}

<h4>Actualizar Usuario</h4>

<hr />

<div class="row">

<div class="col-md-4">

<**form** **asp-action**="Edit">

<**div** **asp-validation-summary**="ModelOnly" class="text-danger"></**div**>

<**input** **type**="hidden" **asp-for**="Id" />

<div class="form-group">

<**label** **asp-for**="Nombres" class="control-label"></**label**>

<**input** **asp-for**="Nombres" class="form-control" />

<**span** **asp-validation-for**="Nombres" class="text-danger"></**span**>

</div>

<div class="form-group">

<**label** **asp-for**="Apellidos" class="control-label"></**label**>

<**input** **asp-for**="Apellidos" class="form-control" />

<**span** **asp-validation-for**="Apellidos" class="text-danger"></**span**>

</div>

<div class="form-group">

<**label** **asp-for**="Dni" class="control-label"></**label**>

<**input** **asp-for**="Dni" class="form-control" />

<**span** **asp-validation-for**="Dni" class="text-danger"></**span**>

</div>

<div class="form-group">

<**label** **asp-for**="Correo" class="control-label"></**label**>

<**input** **asp-for**="Correo" class="form-control" />

<**span** **asp-validation-for**="Correo" class="text-danger"></**span**>

</div>

<div class="form-group">

<**label** **asp-for**="Contrasena" class="control-label">Contraseña</**label**>

<**input** **asp-for**="Contrasena" class="form-control" **type**="password" **value**="@Model.Contrasena"/>

<**span** **asp-validation-for**="Contrasena" class="text-danger"></**span**>

</div>

<div class="form-group">

<**label** **asp-for**="Activo" class="control-label"></**label**>

@Html.CheckBox("Activo")

</div>

<div class="form-group">

<input type="submit" value="Guardar" class="btn btn-primary" /> |

<**a** **asp-action**="Index">Cancelar</**a**>

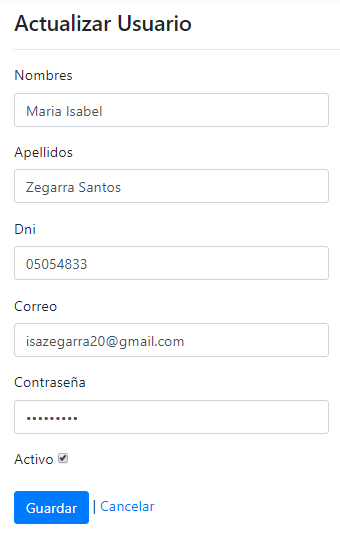
</div>

</**form**>

</div>

</div>

1. Ejecutar la aplicación y modificar los datos un Usuario para validar los cambios.



1. Abrir el archivo ***Delete.cshtml*** y cambiar el código de acuerdo a:

@model MiTiendaVirtual.Models.Usuario

@{

ViewData["Title"] = "Eliminar Usuario";

}

<h4>Eliminar Usuario</h4>

<h4>¿Está seguro de eliminar el siguiente usuario?</h4>

<div>

<hr />

<dl class="row">

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

@Html.DisplayNameFor(model => model.Nombres)

</dt>

<dd class = "col-sm-10">

@Html.DisplayFor(model => model.Nombres)

</dd>

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

@Html.DisplayNameFor(model => model.Apellidos)

</dt>

<dd class = "col-sm-10">

@Html.DisplayFor(model => model.Apellidos)

</dd>

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

@Html.DisplayNameFor(model => model.Dni)

</dt>

<dd class = "col-sm-10">

@Html.DisplayFor(model => model.Dni)

</dd>

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

@Html.DisplayNameFor(model => model.Correo)

</dt>

<dd class = "col-sm-10">

@Html.DisplayFor(model => model.Correo)

</dd>

</dl>

<**form** **asp-action**="Delete">

<**input** **type**="hidden" **asp-for**="Id" />

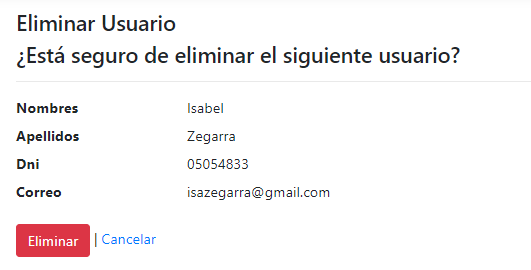
<input type="submit" value="Eliminar" class="btn btn-danger" /> |

<**a** **asp-action**="Index">Cancelar</**a**>

</**form**>

</div>

1. Ejecutar la aplicación y crear un nuevo Usuario para validar los cambios.



**ACTIVIDAD** **VIRTUAL**:

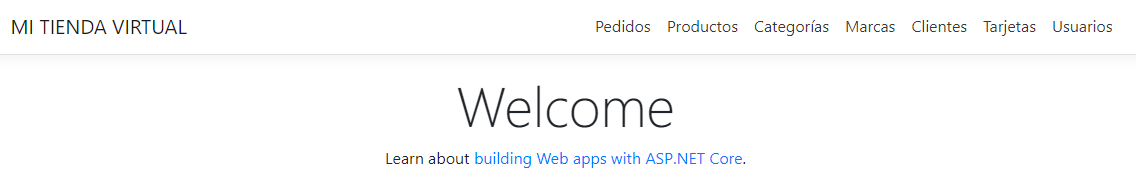
Ingresa a la plataforma virtual. Revisa y analiza el tema desarrollado en esta sesión, luego desarrolla las siguientes actividades propuestas:

1. CUESTIONARIO TÉCNICO
   * 1. Crear con Scaffold Controladores y vistas para el mantenimiento (CRUD) de las siguientes entidades (Tablas):

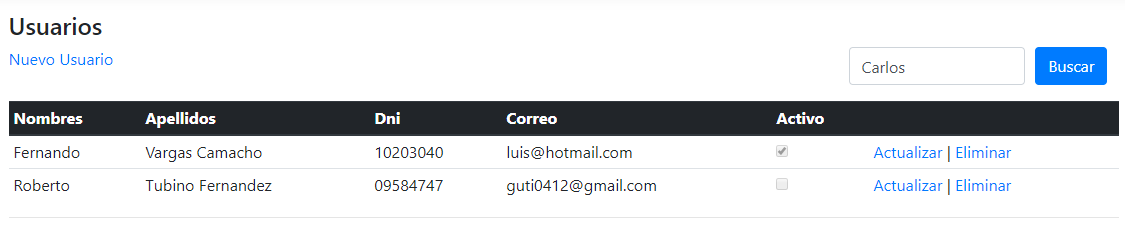
* Marca
* Categoría
* Producto
  + 1. Crear con Scaffold Controladores y vistas para el mantenimiento (CRUD) de las siguientes entidades (Tablas):
* Cliente
* Tarjeta

Estas dos entidades deben ser de solo lectura, esto es, no debe permitir crear, actualizar ni eliminar registros.

* + 1. Para los nuevos mantenimientos, desarrollar las personalizaciones de forma análoga al CRUD de Usuarios.
    2. Agregar en el menú superior links que permitan acceder a los nuevos mantenimientos. Además, alinear el menú a la derecha.



* + 1. Investigar la forma de agregar la funcionalidad de búsqueda en la lista de Usuarios, debe buscar por Nombres y Apellidos.



1. CONCLUSIONES DE LA EXPERIENCIA

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

Un motor paso a paso siempre debemos considerar su corriente de operación y los grados, como indica el ejemplo de la siguiente placa característica de la imagen mostrada:

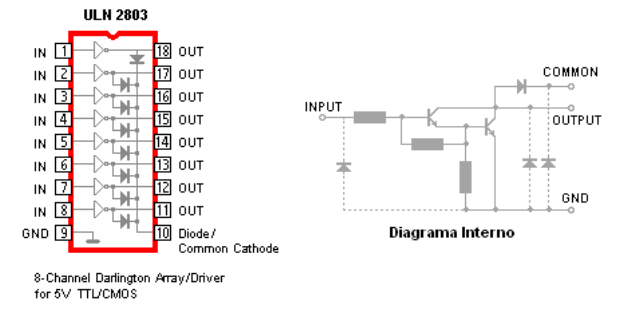


**Subtema 1.2:**

Transistores de potencia

Para operaciones con motores de bajo consumo tenemos circuitos integrados como el circuito ULN2803, el cual es un driver con transistores darlintong , con soporte de hasta 500 mA.

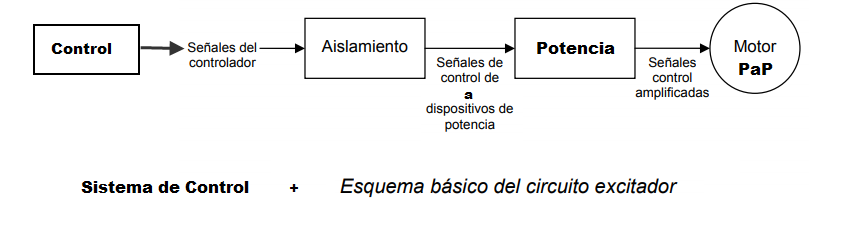
Por lo tanto si deseamos mayores corrientes debemos utilizar otros drivers o implementar uno concomponentes discretos , es decir con transistores de potencia.



**Subtema 1.3:**

Control del Motor

Las etapas para controlar un motor Paso a Paso



**Actividad:**

Los estudiantes desarrollarán ejercicios de Teoría de exponentes y radicales, propuestos por el docente. según lo aprendido en clase. a) CUESTIONARIO TÉCNICO

• ¿Qué diferencia hay entre motores paso a paso unipolares y bipolares?

• ¿Por qué es importante usar driver para motores paso a paso?

• ¿Qué características posee un motor paso a paso?

• ¿Para qué me sirve el puente H?

• Investigue sobre cómo controlar los motores paso a paso con PWM y LOGO! 8.