Herramientas de Programación 2

Tema Nº4:SERVICIOS WEB RESTFULL (PARTE 2)

Indicador de logro Nº4:Reconoce los componentes de la arquitectura orientada a servicios (SOA) verificando el consumo de servicios web en Postma.

**TEMA 01 Teoría de los**

Imagen que contiene Icono

Descripción generada automáticamente

**TEMA Nº4:**

SERVICIOS WEB RESTFULL (PARTE 2)

**Subtema 4.2:**

ASP.NET Core API.

Actualmente la mayoría de empresas utilizan API REST para la creación de servicios profesionales. Esto se debe a que es un estándar lógico y eficiente para la creación de servicios web que se invoca a través del protocolo HTTP, lo que hace posible que se puedan hacer peticiones desde cualquier lenguaje de programación. Grandes empresas como Facebook, Twitter, LinkedIn, Youtube, etc., exponen gran parte de su funcionalidad a través de API’s permitiendo de este modo extender su funcionalidad.

ASP.NET Web API Core es un framework para crear servicios basados HTTP, incluido en ASP.NET Core, a los que se puede acceder desde diferentes aplicaciones y plataformas, como web, windows, móvil, etc.

ASP.NET Core es un nuevo framework de código abierto y multiplataforma, creado y diseñado desde cero por Microsoft, para la creación de aplicaciones modernas y conectadas a internet que permite desarrollar y ejecutar aplicaciones en Windows, Mac y Linux

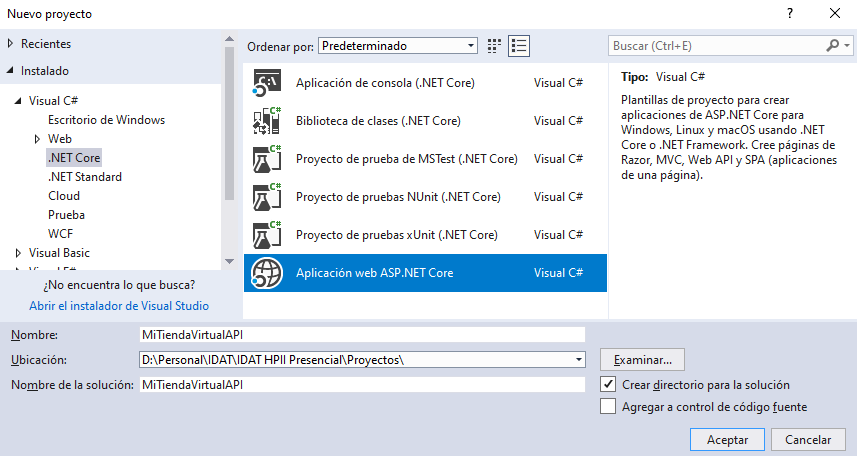


Paso 1. Crear un proyecto ASP.NET Core API en Visual Studio.

1. Abrimos Visual Studio.
2. Seleccionamos la opción: *Archivo > Nuevo > Proyecto…*
3. En el formulario seleccionamos el tipo de proyecto:

*Visual C# > .NET Core > Aplicación web ASP.NET Core*

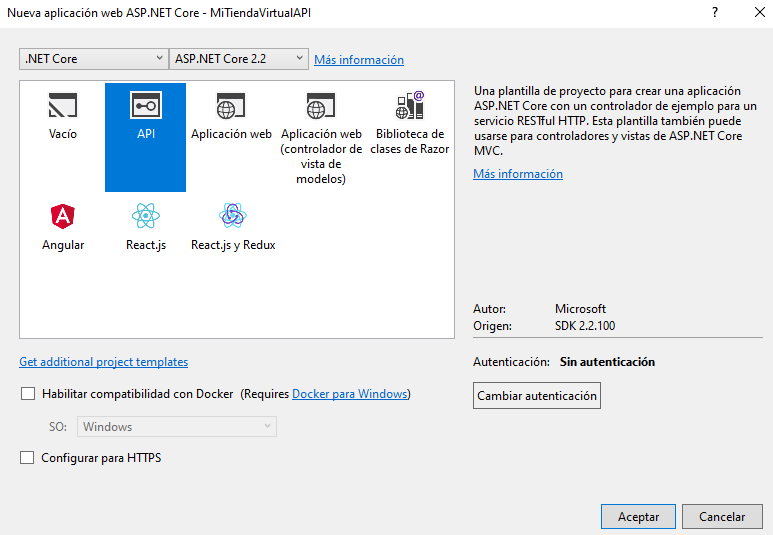
1. Le ponemos el nombre *MiTiendaVirtualAPI* y presionamos en el botón *Aceptar*.



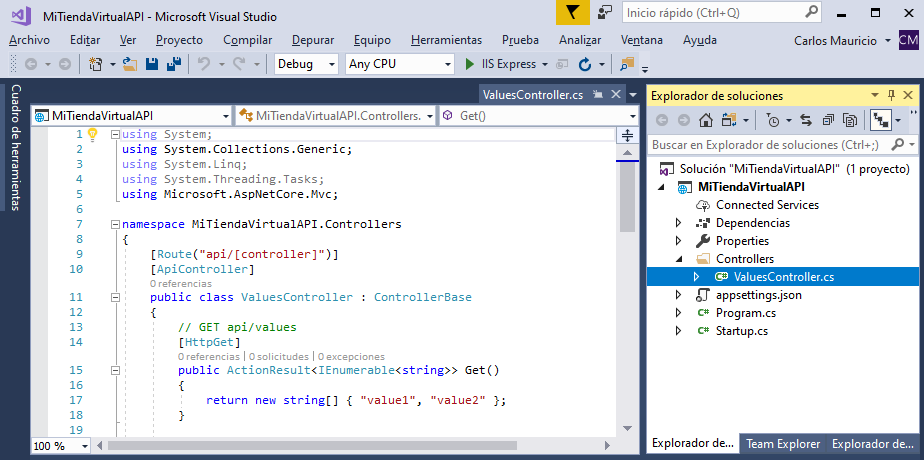
1. En el nuevo formulario seleccionamos la opción: *API.*

Tener cuidado en la versión del .NET Core, debemos seleccionar la opción: *ASP.NET Core 2.2*.

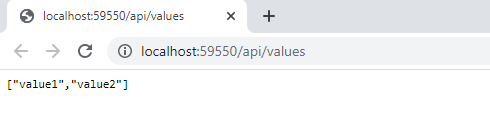
No debe estar seleccionado ninguno de los dos CheckBox.



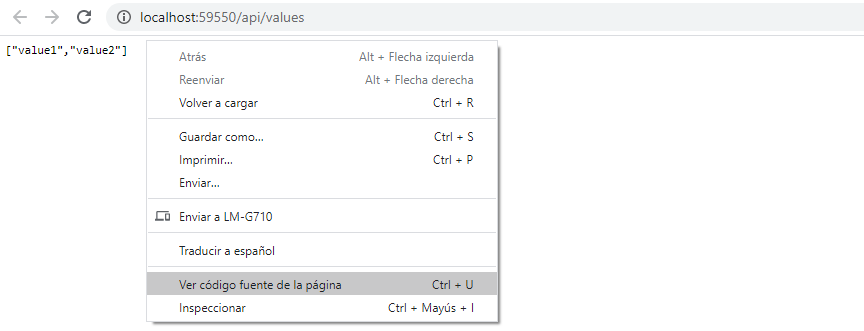
1. Presionamos en el botón Aceptar.
2. Verificamos la estructura del proyecto (API).

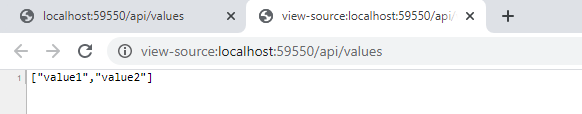


1. Ejecutamos la aplicación, para ellos seleccionamos la opción del menú: IIS Express.
2. Se abrirá una ventana el explorador web con el siguiente contenido.

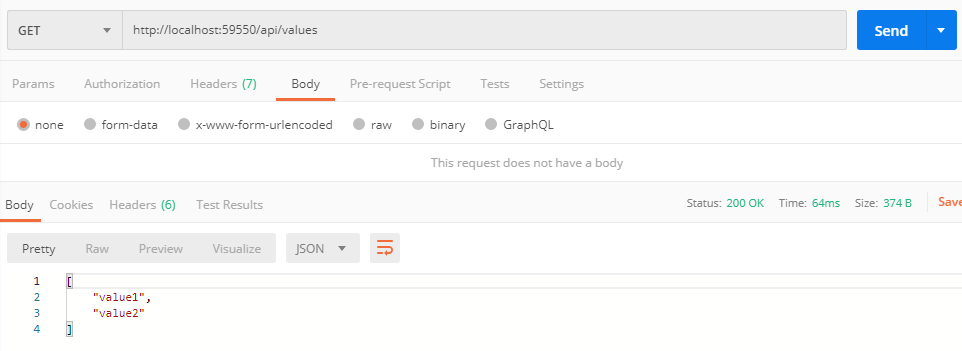


1. Sin embargo no es una página web (HTML) tal cual, sino un arreglo de cadenas de texto con dos elementos, para verificar esto hacemos click derecho dentro de la página y seleccionamos la opción: *Ver código fuente de la página*.





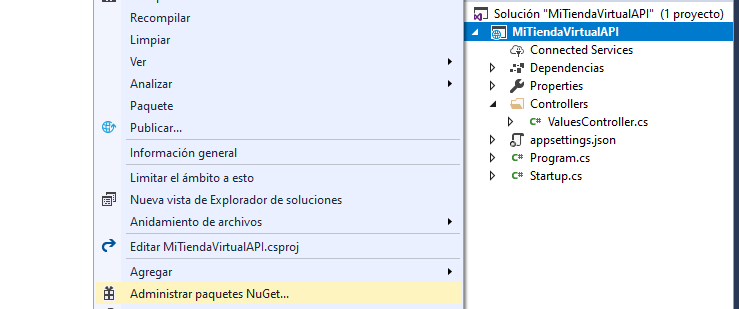
1. Lo mismo podemos obtener consultando el servicio API desde el Postman.



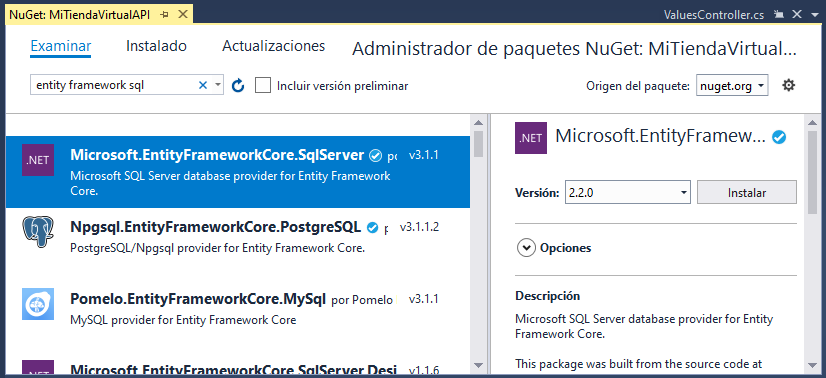
1. Culminamos la ejecución del servicio cerrando el explorador web.

Paso 2. Crear el Modelo de la aplicación a partir de la BD.

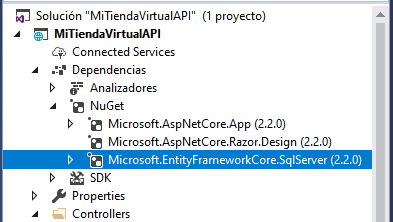
1. Antes de crear las entidades (clases) del Modelo debemos agregar a nuestra aplicación las librerías de Entity Framework Core. Seleccionamos la opción: *Administrar paquetes NuGet.*



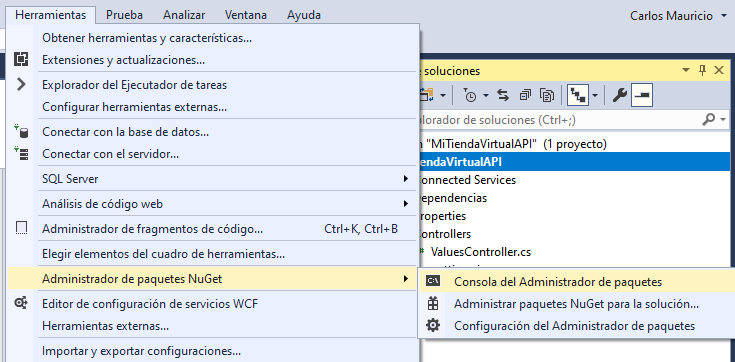
1. En el formulario, dentro de la pestaña *Examinar*, en el cuadro de búsqueda ingresamos el texto: *entity framework sql.*
2. De la lista resultante seleccionamos: *Microsoft.EntityFrameworkCore.SqlServer.*
3. Seleccionamos la versión *2.2.0*
4. Presionamos en el botón *Instalar*.

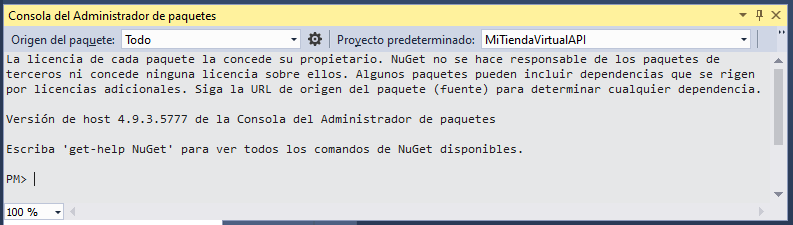


1. Verificamos el paquete instalado.



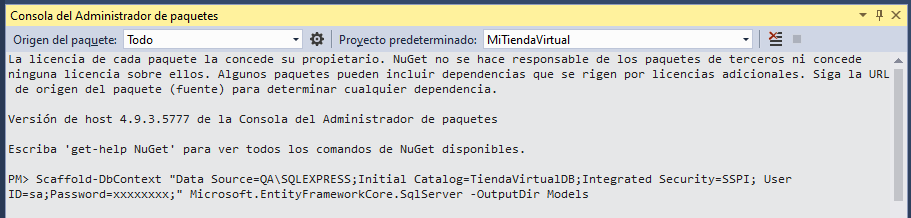
1. Ahora si vamos a generar el Modelo y el Contexto de Datos de nuestra aplicación, desde el menú *Herramientas* seleccionamos la opción: *Consola del Administrador de paquetes*



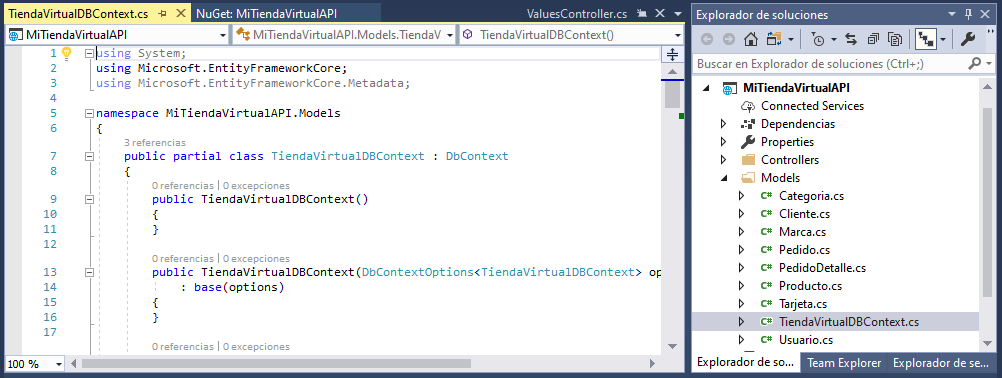


1. Ingresamos el siguiente comando:

Scaffold-DbContext "Data Source=QA\SQLEXPRESS;Initial Catalog=TiendaVirtualDB;Integrated Security=SSPI; User ID=sa;Password=\*\*\*\*\*\*\*\*;" Microsoft.EntityFrameworkCore.SqlServer -OutputDir Models

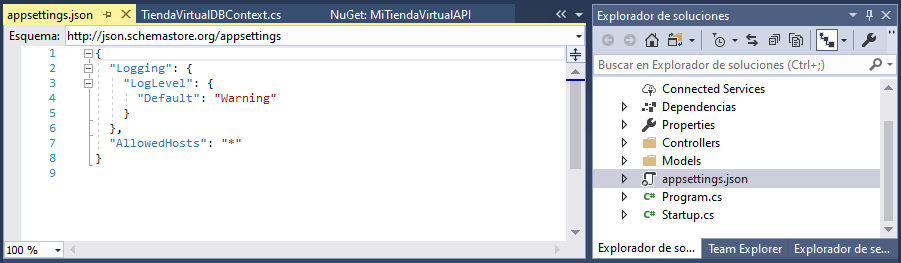


1. Debemos acondicionar la cadena de conexión de acuerdo a la configuración de la BD en SQL Server.
2. Presionamos la tecla *Enter* para ejecutar el comando.
3. Verificamos que se han creado las entidades (clases) del Modelo y su correspondiente Contexto de Datos.



Paso 3. Configurar la cadena de conexión a la BD.

1. Para agregar una cadena de conexión abrimos el archivo *appsettings.json*.



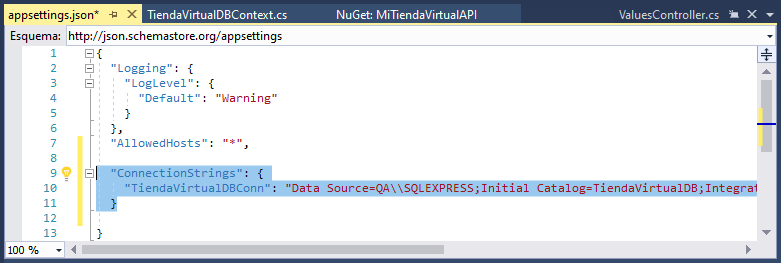
1. Agregamos la siguiente línea de código:

"ConnectionStrings": {

"TiendaVirtualDBConn": "Data Source=QA\\SQLEXPRESS;Initial Catalog=TiendaVirtualDB;Integrated Security=SSPI; User ID=sa;Password=\*\*\*\*\*\*\*"

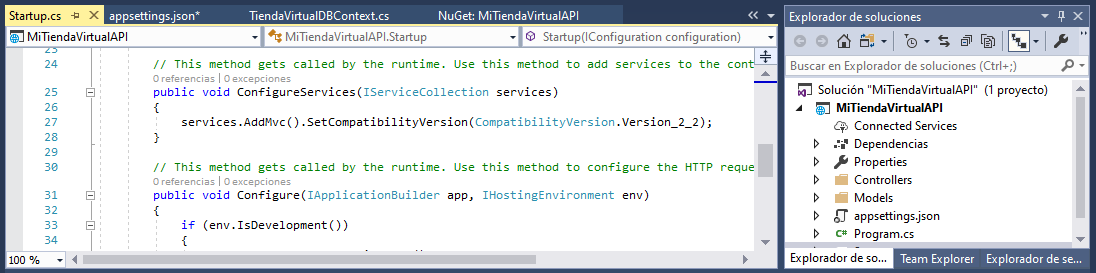
}

Recordar que se debe acondicionar la cadena de conexión de acuerdo a la configuración de la BD en SQL Server.



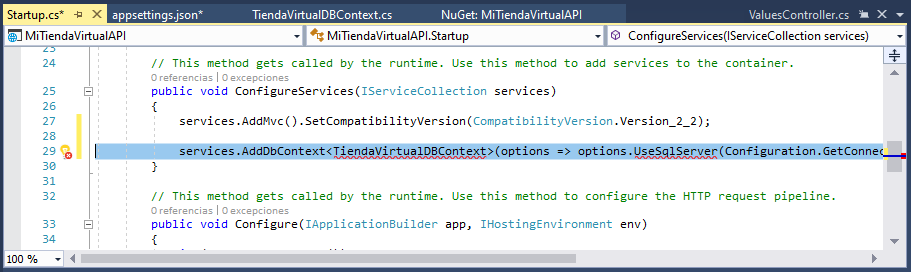
1. Observar que el nombre de la cadena de conexión es *TiendaVirtualDBConn*.
2. Finalmente para optimizar el acceso a la BD con Entity Framework Core vamos a registrar el Contexto de Datos como un Servicio dentro de nuestra aplicación (esto aplica el patrón de diseño denominado Inyección de Dependencias).

Abrimos el archivo *Startup.cs*



1. Agregamos la siguiente línea de código:

services.AddDbContext<TiendaVirtualDBContext>(options => options.UseSqlServer(Configuration.GetConnectionString("TiendaVirtualDBConn")));

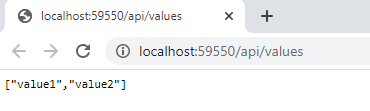


1. Observamos que se muestran errores, para corregirlos debemos agregar en la parte superior de este mismo archivo las referencias a dos componentes de la aplicación:

using MiTiendaVirtualAPI.Models;

using Microsoft.EntityFrameworkCore;

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



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. Modificar el método *Get(int id)* del Controlador *ValuesController.cs* de tal forma que devuelva un objeto de la clase *Usuario*, este objeto no debe obtenerse de la BD sino debe crearse manualmente dentro de la método. Por ejemplo:

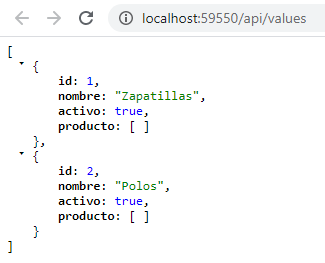


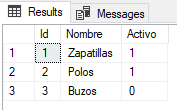
* + 1. Modificar el método *Get()* de tal forma que devuelva todos los usuarios de la tabla *Usuario*. Por ejemplo:



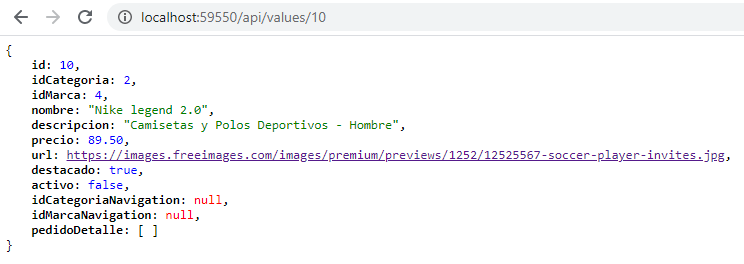
En Chrome, para visualizar información en formato Json debe añadir una extensión, por ejemplo *JSONView*.

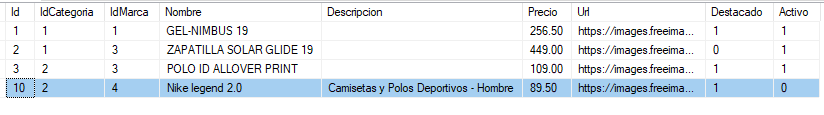
* + 1. Modificar el método *Get()* de tal forma que devuelva todos las categorías activas de la tabla *Categoria*. Por ejemplo:





* + 1. Modificar el método *Get(int id)* de tal forma que devuelva un Producto de la BD, se debe enviar el Id del Producto que se solicita. Por ejemplo:





* + 1. Investigar cual es la forma correcta de consumir servicios de tipo GET, POST, PUT y DELETE. A estos tipos se les conoce como verbos HTTP.

1. CONCLUSIONES DE LA EXPERIENCIA

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

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

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

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

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

**ACTIVIDAD** **VIRTUAL**:

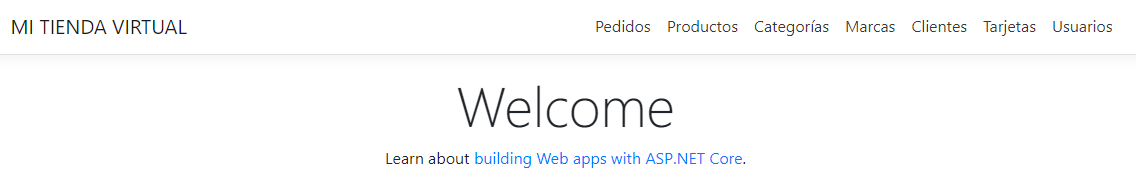
Ingresa a la plataforma virtual. Revisa y analiza el tema desarrollado en esta sesión, luego desarrolla las siguientes actividades propuestas:

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

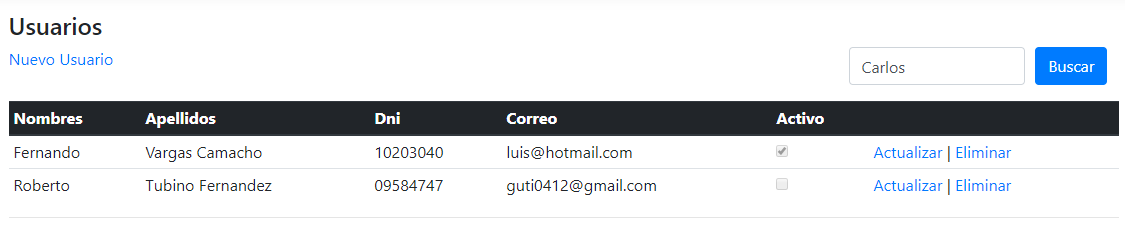
* Marca
* Categoría
* Producto
  + 1. Crear con Scaffold Controladores y vistas para el mantenimiento (CRUD) de las siguientes entidades (Tablas):
* Cliente
* Tarjeta

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:

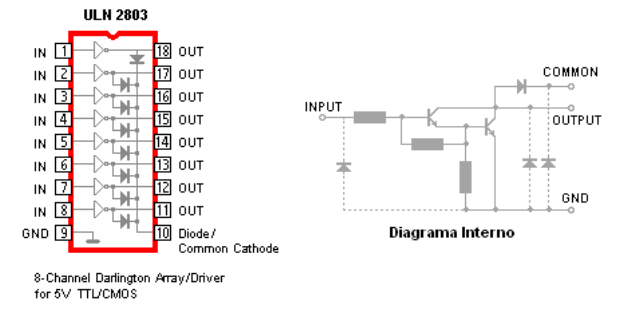


**Subtema 1.2:**

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.

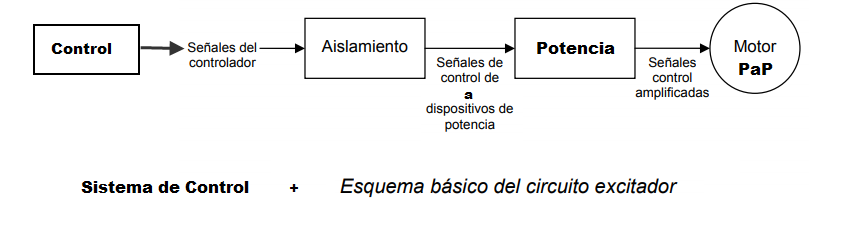
Por lo tanto si deseamos mayores corrientes debemos utilizar otros drivers o implementar uno concomponentes discretos , es decir con transistores de potencia.



**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.