

UNIVERSIDAD NACIONAL AUTÓNOMA DE NICARAGUA

UNAN – LEÓN

FACULTAD DE CIENCIAS Y TECNOLOGÍA  
INGENIERÍA EN SISTEMAS DE LA INFORMACIÓN



AÑO LECTIVO: 2025

SEMESTRE: II

Tarea 1

Componente Curricular: **PROGRAMACION  
ORIENTADA A LA WEB II**

Grupo: GP1

Profesor(a): Lic. Juan Carlos Leyton Briones

Autor:

1. Francisco José Jarquín Briceño.

León, Nicaragua, 1 de agosto de 2025.

*“¡A la Libertad por la Universidad!”*

## Introducción

Esta aplicación es una aplicación de prueba para experimentar con diferentes funcionalidades del desarrollo en .NET utilizando el modelo MVC (Model – View – Controller), por ejemplo, procesar parámetros enviados por la URL, experimentar con Razor y crear métodos dentro de este modelo.

## Aplicación

La aplicación consta de todos los elementos por defecto al crear un nuevo proyecto de .NET + MVC utilizando el comando `dotnet new mvc -o NombreDeProyecto`, a esto se le añaden varios archivos que han sido creados durante el desarrollo de la aplicación, de los cuales están:

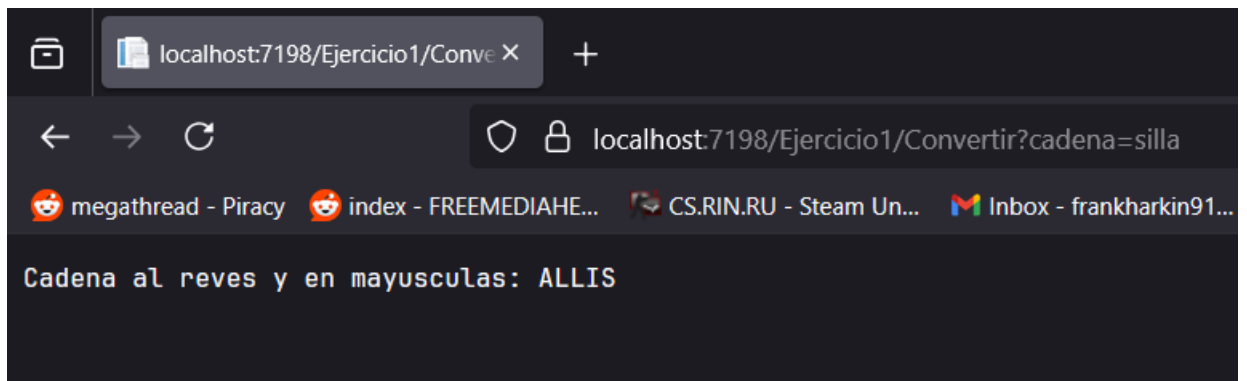
### Ejercicio1.cs

Es un controlador que posee los siguientes métodos:

- `Convertir(cadena)`: Este método convierte la cadena que se le pasa a mayúsculas e invertida.

Ejemplo: `Convertir("silla")` nos devolvería la cadena "ALLIS".

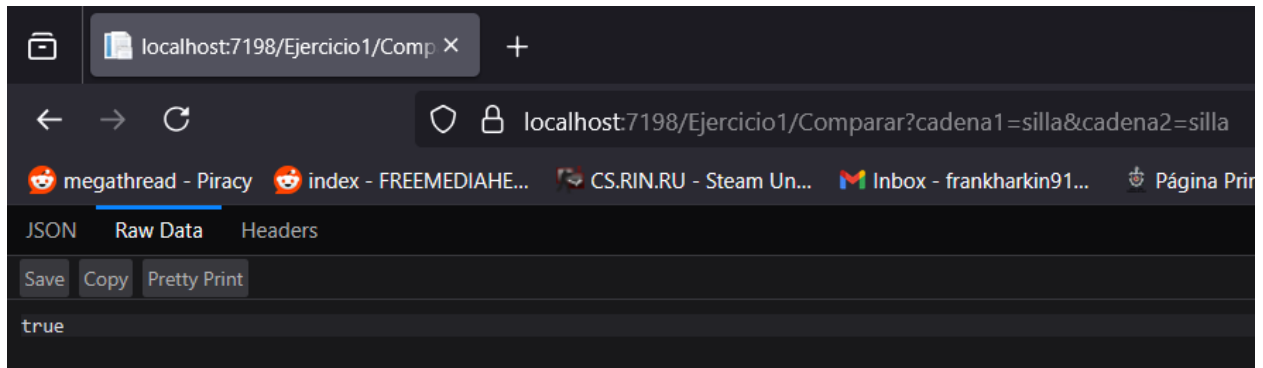
Se invoca al escribir en el URL de nuestro navegador `"/Ejercicio1/Convertir?cadena=cadena_a_convertir"`



- `Comparar(cadena1, cadena2)`: Este método compara dos cadenas y devuelve el resultado de la comparación, `true` si son iguales y `false` si son diferentes.

Ejemplo: Comparar("silla", "silla") nos devolvería true, mientras que Comparar("silla", "ola") nos devolvería false.

Se invoca al escribir en el URL de nuestro navegador ".../Ejercicio1/Comparar?cadena1=cadenaUno&cadena2=cadenaDos"



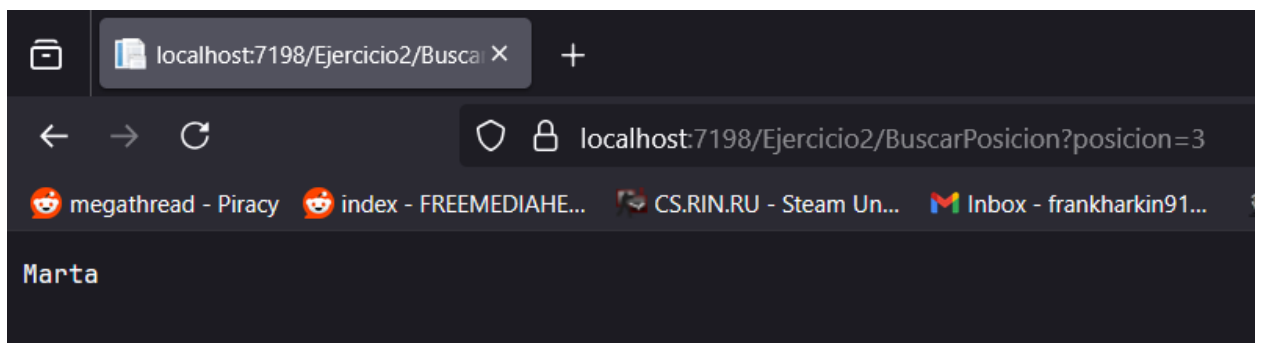
## Ejercicio2.cs

Es un controlador que posee una lista de cadenas con nombres de personas, además de los siguientes métodos:

- BuscarPosicion(posición): Es un método que retorna el nombre que se encuentra en la posición indicada, o la cadena "Posición inválida" en caso de no ser encontrada.

Ejemplo: BuscarPosicion(3) retornaría el nombre de la cadena que se encuentre en el índice 3 de la lista de nombres (nombres[3]), en este caso, "Marta".

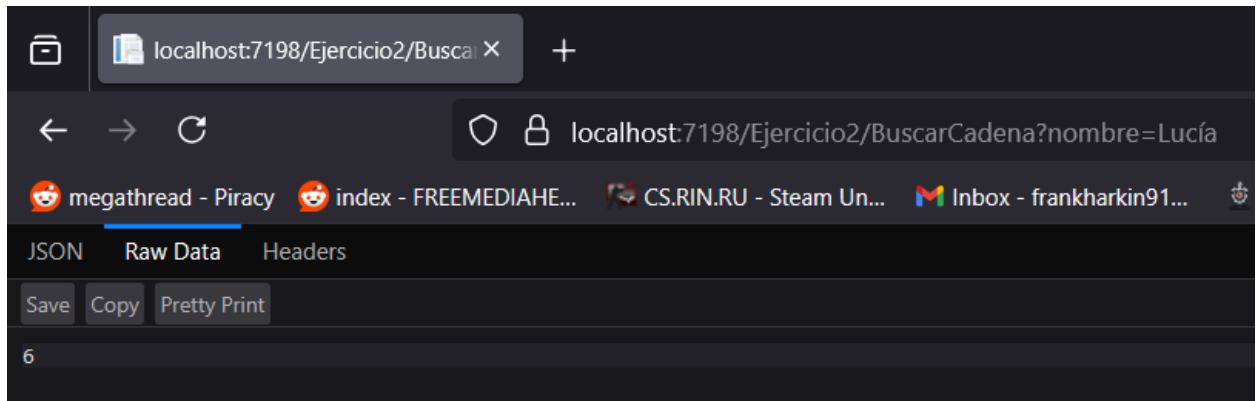
Se invoca al escribir en el URL de nuestro navegador ".../Ejercicio2/BuscarPosicion?posicion=3"



- `BuscarCadena(nombre)`: Es un método que busca el nombre proporcionado en la lista de nombres y retorna el índice del nombre, o -1 en caso de no encontrarlo.

Ejemplo: `BuscarCadena("Lucía")` retorna el índice del nombre, en este caso, 6.

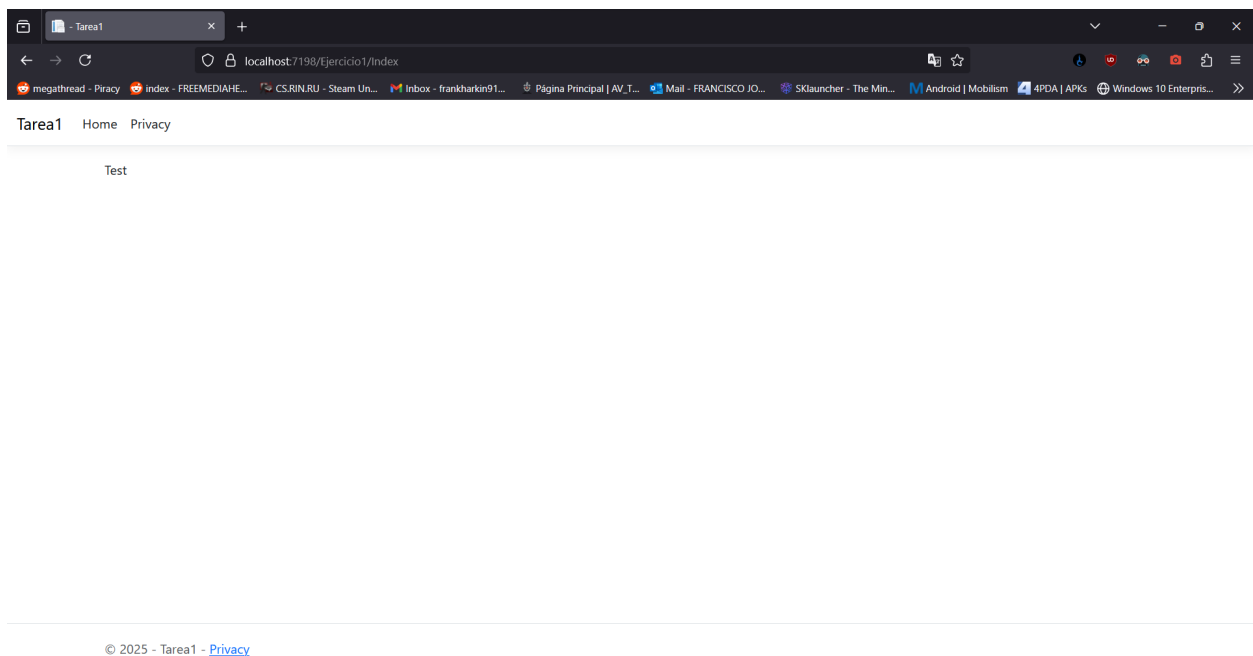
Se invoca al escribir en el URL de nuestro navegador `“.../Ejercicio2/BuscarCadena?nombre=Lucía”`



## Ejercicio1/Index.cshtml

Es una vista que solo muestra un mensaje de prueba muy básico.

Se accede a ella navegando al URL `“.../Ejercicio1”`, o al URL `“.../Ejercicio1/Index”`.



## Ejercicio1/Perfil.cshtml

Es una vista que muestra nuestro perfil, el cual incluye nuestro nombre, edad, foto de perfil y carrera en la que uno estudia.

Se accede a ella navegando al URL “.../Ejercicio1/Perfil”.

