**UNIVERSIDAD NACIONAL AUTÓNOMA DE NICARAGUA**

**UNAN – LEÓN**

**FACULTAD DE CIENCIAS Y TECNOLOGÍA**

**INGENIERÍA EN SISTEMAS DE LA INFORMACIÓN**

****

**AÑO LECTIVO: 2025**

**SEMESTRE: II**

**Componente Curricular: programación web 2.**

**Grupo: 1**

**Documentación tarea 1 (Vistas)**

**Profesor(a): Lic. Juan Carlos Leyton Briones**

**Autor:**

**Josías Javier Sosa Salinas.**

León, Nicaragua, 2025.

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

**Introducción**

**Descripción del Proyecto**

Este es un proyecto de prueba de una **aplicación .NET MVC** desarrollada para explorar diversas funcionalidades de programación. La aplicación se basa en la plantilla estándar de .NET + MVC, a la que se le han añadido archivos y métodos personalizados para ampliar sus capacidades.

**Aplicación**

La aplicación utiliza la estructura predeterminada generada con el comando **dotnet new mvc -o NombreDeProyecto**. A esta base se le han incorporado varios archivos adicionales creados durante el desarrollo del proyecto.

**Componentes y Funcionalidades Principales**

**Controlador Ejercicio1.cs**

Este controlador contiene métodos para manipular y comparar cadenas de texto.

* **Convertir(cadena)**: Este método invierte y convierte a mayúsculas una cadena de texto que recibe de la URL. Por ejemplo, al ingresar **.../Ejercicio1/Convertir?cadena=silla** en el navegador, la respuesta será **"ALLIS"**.
* **Comparar(cadena1, cadena2)**: Compara dos cadenas de texto recibidas a través de la URL. Devuelve **true** si son idénticas y **false** si son diferentes. Por ejemplo, al usar **.../Ejercicio1/Comparar?cadena1=silla&cadena2=silla**, el resultado será **true**.

**Controlador Ejercicio2.cs**

Este controlador trabaja con una lista predefinida de nombres de personas y ofrece los siguientes métodos:

* **Método BuscarPosicion(posición)**: Este método busca y devuelve un nombre de la lista basándose en un índice numérico (posición) proporcionado. Si el índice es válido, como **5**, el método retornará el nombre en esa posición, que en el ejemplo es **"Jorge"**. Si el índice está fuera del rango de la lista, el método maneja el error de forma segura, devolviendo la cadena **"Posición inválida"**. Puedes usarlo navegando a la URL **.../Ejercicio2/BuscarPosicion?posicion=5**.
* **Método BuscarCadena(nombre)**: Funciona como un buscador que encuentra y devuelve el índice de un nombre específico en la lista. Por ejemplo, para el nombre **"Lucía"**, el método devolverá **6**. Si el nombre no se encuentra, el método devuelve **-1** para indicar que no hubo coincidencias. Se accede a este método a través de la URL **…/Ejercicio2/BuscarCadena?nombre=Lucía**.

**Vista Ejercicio1/Index.cshtml**

La vista **Index.cshtml** es una página web sencilla creada para fines de prueba. Su única función es mostrar un mensaje que confirma que la aplicación está funcionando correctamente. Para visualizarla, simplemente navega a **.../Ejercicio1** o **.../Ejercicio1/Index** en tu navegador.