

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 Orientada a la
Web II

Grupo: GP1

Profesor(a): Ing.

Autor:

1. Brandon Gonzalo Altamirano Avendaño

León, Nicaragua, 31 de Agosto del 2025.

“¡A la Libertad por la Universidad!”

Acerca del proyecto

Este proyecto es una práctica de MVC en .NET 6. La idea era aprender a crear controladores, recibir parámetros desde la URL y mostrar resultados dinámicos en las vistas.

Se hicieron dos controladores: Ejercicio1 para jugar con cadenas y mostrar un perfil personal, y Ejercicio2 para buscar nombres dentro de una lista.

Controladores

Ejercicio1

Este controlador es como un pequeño laboratorio para trabajar con texto. Tiene tres cosas principales:

Index()

Es básicamente la página de inicio del controlador. Solo devuelve la vista Index.cshtml con un mensaje de prueba.

Convertir(string cadena)

Recibe cualquier cadena que pongas en la URL, la pone en mayúsculas y la invierte.

Por ejemplo, si escribes hola, te devuelve ALOH.

Está pensado para ver cómo se pueden recibir datos desde la URL y manipularlos en C#.

Comparar(string cadena1, string cadena2)

Compara dos cadenas que le pases.

Devuelve true si son iguales y false si no.

Sirve para entender cómo hacer comparaciones simples usando parámetros.

Perfil()

Muestra una vista Perfil.cshtml con mis datos personales: nombre, edad, carrera y una foto.

Es solo para practicar cómo mostrar datos dinámicos en la vista con Razor.

Ejercicio2

Este controlador es más de listas y búsqueda. Tiene una lista privada de nombres y permite buscar de dos formas:

BuscarPosicion(int posicion)

Le das un número y devuelve el nombre que está en esa posición dentro de la lista.

Si te sales del rango de la lista, devuelve "Posición inválida".

Sirve para entender cómo trabajar con arrays o listas en C#.

BuscarCadena(string nombre)

Busca un nombre específico dentro de la lista.

Devuelve la posición donde lo encontró o -1 si no existe.

Es como practicar búsquedas dentro de una colección.

Vistas

Index.cshtml

Simple, solo tiene un `<p>Test</p>`. Es más que todo para comprobar que el controlador funciona.

Perfil.cshtml

Muestra mis datos personales.

Usa Razor para mostrar información como el nombre, la edad, la carrera y una foto.

Así se ve cómo se mezclan datos del controlador con HTML en la vista.

```
@{  
    ViewData["Title"] = "Perfil";  
}
```

```
<h1>@ViewData["Title"]</h1>  
<p>Nombre: Brandon Gonzalo Altamirano Avendaño</p>  
<p>Edad: 27</p>
```

```
  
<p>Carrera: Ingeniería en Sistemas de la Información</p>
```

Cómo funciona todo junto

Pones la URL en el navegador, por ejemplo:

Ejercicio1/Convertir?cadena=hola → devuelve la cadena invertida y en mayúsculas.

Ejercicio2/BuscarCadena?nombre=Luis → te dice en qué posición está el nombre en la lista.

El controlador recibe los datos, hace su magia en C# y devuelve algo:

Texto plano (Convertir, BuscarPosicion)

Booleano (Comparar)

Vista HTML (Perfil o Index)

Las vistas muestran los resultados y puedes probar que todo funcione desde el navegador.

Cosas que aprendí haciendo esto

Cómo crear controladores en MVC.

Cómo recibir parámetros desde la URL y usarlos en métodos.

Cómo devolver datos simples o vistas completas según lo que necesites.

Cómo mezclar C# con Razor para mostrar información en la página.

La diferencia entre métodos que devuelven texto, booleanos o vistas.