

**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: Programacion WEB II**

**Grupo: 1**

**Sub-Grupo: 1**

**Profesor:** Juan Carlos Leyton Briones

**Autor:**

1. Diego Rafael Mairena López

León, Nicaragua, 2025.

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

# **GUÍA DE APRENDIZAJE 1 — Programación con ASP.NET Core MVC (Introducción a MVC)**

## **I. Datos generales**

- Facultad: Ciencias y Tecnología
- Carrera: Ingeniería en Sistemas de Información
- Componente: Programación Orientada a la Web II
- Unidad: Programación con ASP .NET
- Tema: Introducción a MVC
- Año: 2025

## **II. Competencia**

Crear una aplicación web modular y escalable aplicando el patrón MVC en ASP.NET Core.

## **III. Introducción**

Este ejercicio introduce el patrón MVC en ASP.NET Core mediante la creación de una aplicación con controladores y vistas que reciben y procesan parámetros desde la URL. Se trabaja con .NET 6 SDK, enrutamiento por convención y acciones simples que: transforman texto, comparan cadenas, renderizan una vista de perfil y buscan elementos en un arreglo.

## **IV. Requisitos previos**

1. Windows 10/11 o Linux/macOS.
2. .NET 6 SDK instalado.
3. Editor/IDE: Visual Studio 2022, Visual Studio Code o Rider.
4. Navegador actualizado.

## V. Creación del proyecto MVC

1. Crear y ejecutar el proyecto:

```
dotnet new mvc -n GuiaMVC
```

```
cd GuiaMVC
```

```
dotnet run
```

2. Abrir la URL <https://localhost:xxxx> que aparece en consola.

3. Estructura de carpetas:

GuiaMVC/Controllers, Models, Views, wwwroot, Program.cs, appsettings.json.

## VI. Implementación solicitada

Se crean dos controladores: Ejercicio1Controller y Ejercicio2Controller, y una vista Perfil.cshtml.

### Código de Ejercicio1Controller.cs

```
using Microsoft.AspNetCore.Mvc;
```

```
namespace GuiaMVC.Controllers
```

```
{
```

```
    public class Ejercicio1Controller : Controller
```

```
    {
```

```
        [HttpGet]
```

```
        public IActionResult Convertir(string texto)
```

```
        {
```

```
            if (string.IsNullOrEmpty(texto))
```

```
                return Content("Proporciona ?texto=...");
```

```
            string mayus = texto.ToUpperInvariant();
```

```
            string inverso = new string(mayus.Reverse().ToArray());
```

```
            return Content(inverso);
```

```
        }
```

```
        [HttpGet]
```

```
        public IActionResult Comparar(string a, string b)
```

```
        {
```

```

        if (a is null || b is null)
            return Content("Proporciona ?a=...&b=...");

        bool iguales = string.Equals(a, b, StringComparison.OrdinalIgnoreCase);
        return Content(iguales ? "Iguales" : "Diferentes");
    }

    [HttpGet]
    public IActionResult Perfil()
    {
        ViewData["Nombre"] = "Diego Rafael Mairena López";
        ViewData["Edad"] = 20;
        ViewData["Carrera"] = "Ingeniería en Sistemas de Información";
        ViewData["FotoUrl"] = Url.Content("~/images/perfil.jpg");
        return View();
    }
}
}
}

```

### Código de Perfil.cshtml

```

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

<div style="max-width:680px;margin:auto;padding:24px;border:1px solid
#eee;border-radius:12px">
    <h1>@ViewData["Title"]</h1>
    
    <p><strong>Nombre:</strong> @ViewData["Nombre"]</p>
    <p><strong>Edad:</strong> @ViewData["Edad"]</p>
    <p><strong>Carrera:</strong> @ViewData["Carrera"]</p>
</div>

```

## Código de Ejercicio2Controller.cs

```
using Microsoft.AspNetCore.Mvc;

namespace GuiaMVC.Controllers
{
    public class Ejercicio2Controller : Controller
    {
        private readonly string[] _items = new[]
        {
            "manzana", "banana", "cereza", "durazno", "uva", "mango", "pera"
        };

        [HttpGet("Ejercicio2/buscar/indice/{pos:int}")]
        public IActionResult BuscarPorIndice(int pos)
        {
            if (pos < 0 || pos >= _items.Length)
                return Content($"Índice fuera de rango. Usa 0..{_items.Length - 1}");

            return Content(_items[pos]);
        }

        [HttpGet("Ejercicio2/buscar/texto")]
        public IActionResult BuscarPorTexto(string q)
        {
            if (string.IsNullOrEmpty(q))
                return Content("Proporciona ?q=...");

            var coincidencias = _items
                .Select((s, i) => new { s, i })
                .Where(x => x.s.Contains(q, StringComparison.OrdinalIgnoreCase))
                .Select(x => $"{x.i}:{x.s}")
                .ToArray();

            return coincidencias.Length == 0
                ? Content("No encontrada")
                : Content("Encontrada(s): " + string.Join(", ", coincidencias));
        }
    }
}
```

## Código de Program.cs

```
var builder = WebApplication.CreateBuilder(args);
builder.Services.AddControllersWithViews();
var app = builder.Build();

if (!app.Environment.IsDevelopment())
{
    app.UseExceptionHandler("/Home/Error");
    app.UseHsts();
}

app.UseHttpsRedirection();
app.UseStaticFiles();
app.UseRouting();
app.UseAuthorization();

app.MapControllerRoute(
    name: "default",
    pattern: "{controller=Home}/{action=Index}/{id?}");

app.Run();
```

## Pruebas de la aplicación

- /Ejercicio1/convertir?texto=hola mundo → ODNUM ALOH
- /Ejercicio1/comparar?a=perro&b=PERRO → Iguales
- /Ejercicio1/perfil → Muestra datos y foto
- /Ejercicio2/buscar/indice/3 → durazno
- /Ejercicio2/buscar/texto?q=na → banana, mango

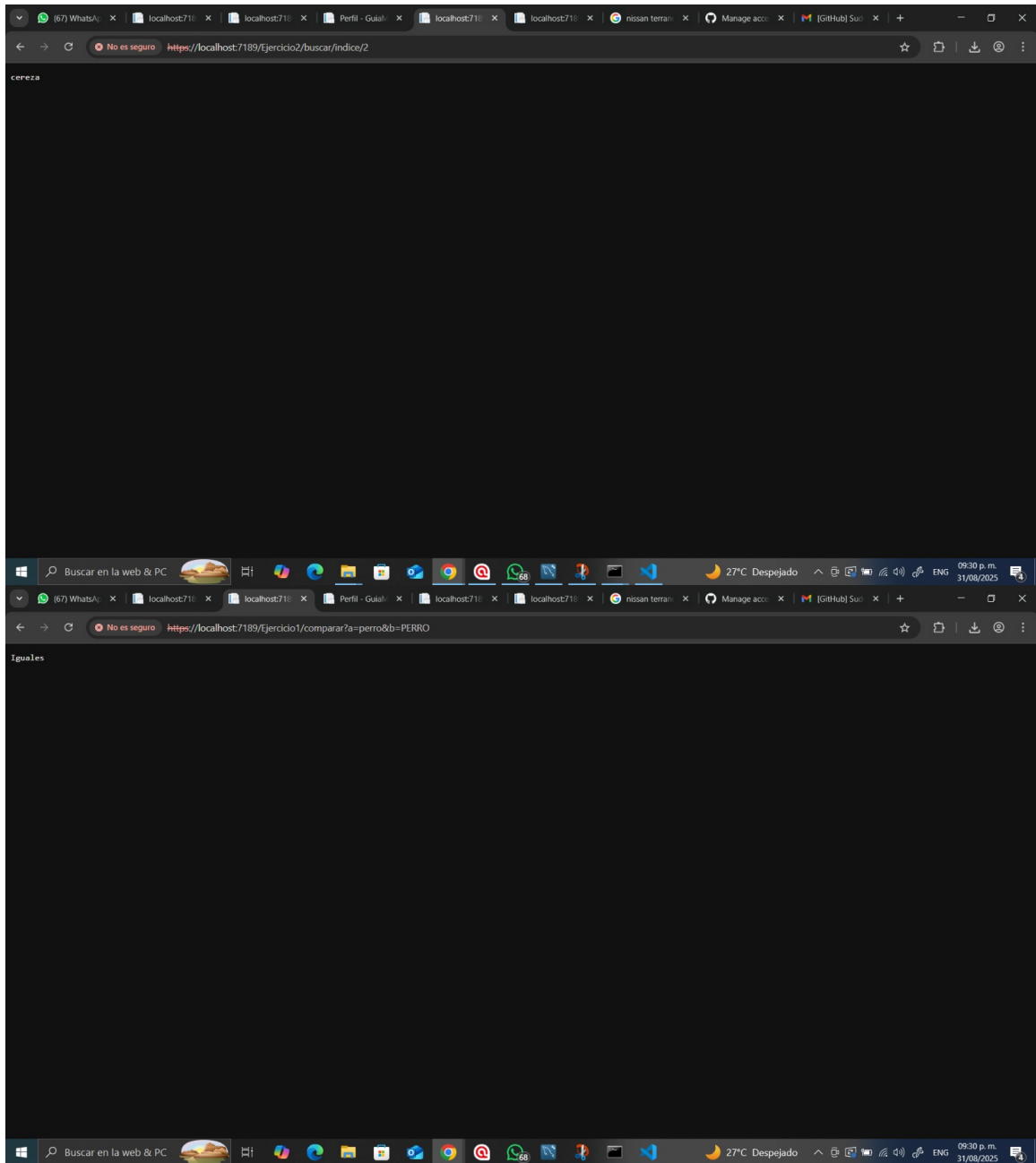
## Perfil



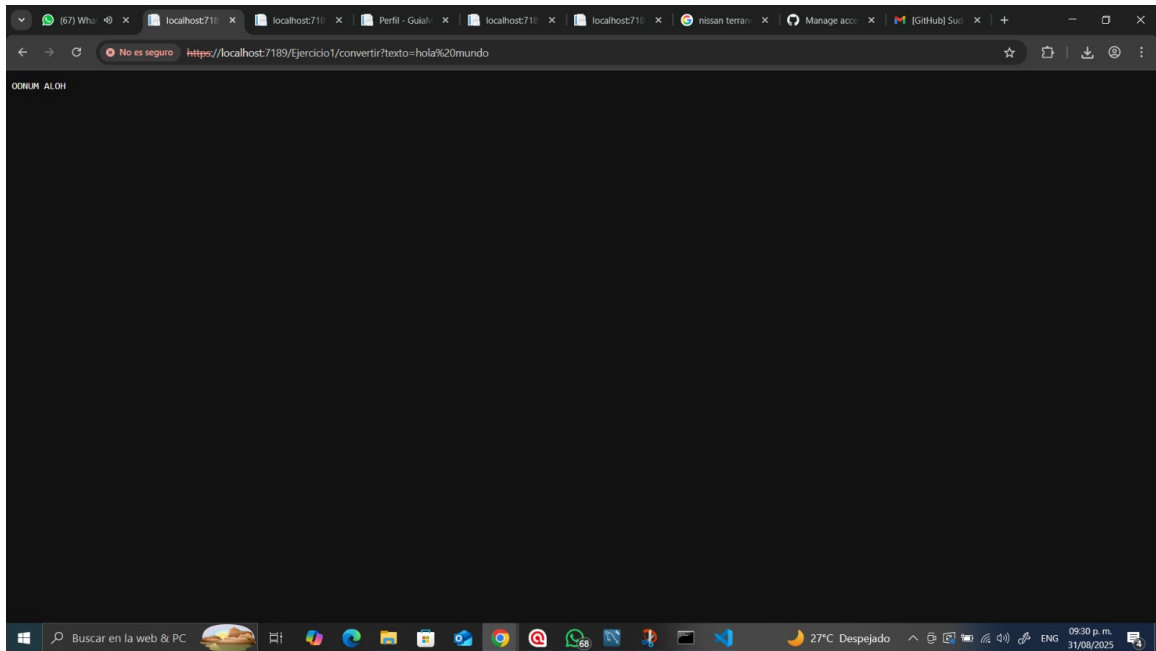
**Nombre:** Diego Rafael Mairena López

**Edad:** 20

**Carrera:** Ingeniería en Sistemas de Información







## Conclusión

Se creó una aplicación ASP.NET Core MVC básica que demuestra el uso de controladores, acciones, vistas y parámetros. Esta guía sirve como base para proyectos más complejos.