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 ORIENTADA A LA WEB II

Grupo: 1
Sub-Grupo:
Profesor(a): JUAN CARLOS LEYTON BRIONES
Autores:

1. Kenner David Rodriguez González.

León, Nicaragua, 2025.

"¡A la Libertad por la Universidad!"

PRACTICA 2 - Frutas

Kenner David Rodriguez Gonsales

CONTROLADOR FRUTAAPI, MODELOS, VISTAS Y JSON

Controlador FrutaApi

El controlador `FrutaApiController` pertenece a la capa de controladores de ASP.NET Core MVC y se encarga de gestionar información de frutas a partir de un archivo JSON o de una API. Permite listar todas las frutas, buscar por nombre y mostrar detalles individuales.

Campos y Constructor

```
public List<Fruta> lstFrutas = null;

public FrutaApiController(FrutaService frutaService)
{
    _frutaService = frutaService;
}
```

Descripción:

- `lstFrutas`: Lista pública que contiene todos los objetos `Fruta`.
- Constructor: recibe `FrutaService` a través de inyección de dependencias.
- Esto asegura que la lista de frutas esté disponible para todos los métodos del controlador.

Métodos del controlador

1. Index()

```
public async Task<IActionResult> Index()
{
    var frutas = await _frutaService.GetFrutasAsync();
    return View(frutas);
}
```

Descripción:

- No recibe parámetros.
- Retorna la vista `Index.cshtml`, pasando la lista completa de frutas.
- Sirve como página principal para mostrar todas las frutas.

2. Find(string nombre)

```
[HttpPost]
public async Task<IActionResult> Find(string nombre)
{
    var frutas = await _frutaService.SearchFrutasAsync(nombre);
    return View("Index", frutas);
}
```

Descripción:

- Recibe un parámetro `nombre` para buscar la fruta.
- Filtra las frutas cuyo nombre contenga el texto buscado (case insensitive).
- Retorna la vista `Index` mostrando solo los resultados filtrados.

3. Details(int id)

```
public async Task<IActionResult> Details(int id)
{
    var frutas = await _frutaService.GetFrutasAsync();
    var fruta = frutas.FirstOrDefault(f => f.Id == id);
    if (fruta == null)
        return NotFound();
    return View(fruta);
}
```

Descripción:

- Recibe un ID de fruta.
- Busca en `lstFrutas` la fruta con ese ID.
- Si no existe → retorna un 404 NotFound.
- Si existe → retorna la vista `Details.cshtml` mostrando los detalles de la fruta.

MODELOS

Clase `Fruta` y `FrutaContainer`

Clase Fruta

```
public class Fruta
{
    public int Id { get; set; }
    public string? Nombre { get; set; }
    public string? Genero { get; set; }
    public string? Familia { get; set; }
    public string? Orden { get; set; }
    public Nutricion? Nutriciones { get; set; }
}
```

Clase Nutricion

```
public class Nutricion
{
    public double Carbohidratos { get; set; }
    public double Proteina { get; set; }
    public double Grasa { get; set; }
    public double Calorias { get; set; }
    public double Azucar { get; set; }
}
```

Clase FrutaContainer

```
public class FrutaContainer
{
    public List<Fruta> Frutas { get; set; } = new List<Fruta>();
}
```

VISTAS

Index.cshtml

```
@model List<Tarea1.Models.Fruta>

<h2>Lista de Frutas</h2>

<form class="form-inline" method="POST" action="/FrutaApi/Find">
    <div class="form-group mx-sm-3 mb-2">
        <label class="mr-3">Buscar Fruta</label>
        <input type="text" class="form-control" required name="nombre">
    </div>
    <button type="submit" class="btn btn-primary mb-2">Buscar</button>
</form>

<div class="container">
    <div class="row">
        @foreach (var fruta in Model)
        {
            <div class="col-md-3">
                <div class="card mb-3">
                    <div class="card-body">
                        <h5 class="card-title">@fruta.Nombre</h5>
                        <p class="card-text">Género: @fruta.Genero</p>
                        <p class="card-text">Familia: @fruta.Familia</p>
                        <p class="card-text">Orden: @fruta.Orden</p>
                        <p class="card-text">Carbohidratos:
@fruta.Nutriciones?.Carbohidratos g</p>
                        <p class="card-text">Proteína:
@fruta.Nutriciones?.Proteina g</p>
                        <p class="card-text">Grasa: @fruta.Nutriciones?.Grasa
```

```
g</p>
                        <p class="card-text">Calorías:
@fruta.Nutriciones?.Calorias kcal</p>
                        <p class="card-text">Azúcar:
@fruta.Nutriciones?.Azucar g</p>
                        <a href="@Url.Action("Details", "FrutaApi", new { id
= fruta.Id })" class="btn btn-primary">Details</a>
                    </div>
                </div>
            </div>
        }
    </div>
</div>
```

JSON

Ejemplo de fruta en JSON: ```json { “Id”: 1, “Nombre”: “Manzana