

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!”

Indice

GUÍA DE APRENDIZAJE 2 — Vistas en .NET 6 – Visor Seguro de Contenido de e926 (Artista DarkAlcaline)	1
I. Datos Generales	1
II. Competencia del Componente Curricular	1
III. Introducción.....	1
IV. Arquitectura del Proyecto.....	2
V. Funcionalidad Principal	2
VI. Flujo de la Aplicación	2
VII. Archivos Clave	3
VIII. Decisiones de Diseño	3
IX. Flujo de Datos.....	3
X. Pruebas y Evidencias	3
XI. Mejoras Futuras	4
XII. Conclusión	5

GUÍA DE APRENDIZAJE 2 — Vistas en .NET 6 – Visor Seguro de Contenido de e926 (Artista DarkAlcaline)

I. Datos Generales

- Facultad: Ciencias y Tecnología
- Carrera: Ingeniería en Sistemas de Información
- Modalidad: Diurna
- Componente Curricular: Programación Orientada a la Web II
- Unidad: Programación con ASP.NET
- Tema: Vistas en .NET 6
- Tiempo requerido: 2 horas presencial, 4 horas no presencial
- Profesor: Juan Carlos Leyton Briones
- Año académico: 2025

II. Competencia del Componente Curricular

El estudiante desarrolla aplicaciones web modulares y escalables, aplicando el patrón MVC en .NET 6, integrando datos desde APIs externas y diseñando vistas dinámicas que permitan búsquedas, detalles y navegación de información.

III. Introducción

El proyecto consiste en una aplicación web basada en ASP.NET Core MVC que funciona como galería segura de contenido del artista “DarkAlcaline” utilizando la API de e926.net, un repositorio de imágenes y arte con control de contenido seguro (SFW). Se aprovecha la arquitectura MVC para separar lógica, datos y presentación. La app está pensada para un público menor de 13 años, por lo que integra filtros estrictos de contenido y tags, mostrando solo información segura y relevante.

IV. Arquitectura del Proyecto

El proyecto se basa en una estructura MVC estándar de ASP.NET Core:

Controladores: Contienen la lógica de negocio y orquestan las peticiones a la API.

Modelos: Representan los datos obtenidos de e926 y permiten su manipulación en C#.

Vistas: Generan la interfaz de usuario utilizando Razor Pages, Bootstrap y plantillas reutilizables.

Servicios: Implementan el consumo de la API de e926 mediante HttpClient, manejo de cabeceras y filtrado de datos.

Archivos estáticos: Recursos CSS y JS ubicados en wwwroot.

V. Funcionalidad Principal

- 1. Galería filtrada:** Solo muestra publicaciones etiquetadas con el tag darkcaline, forzando rating seguro.
- 2. Detalle de publicaciones:** Presenta imagen optimizada, datos básicos y tags filtrados.
- 3. Filtros de contenido:** Elimina tags sensibles y refuerza rating seguro.
- 4. Diseño responsivo:** Interfaz adaptable a dispositivos móviles.
- 5. Uso de API pública:** Consumo de datos en tiempo real desde e926 con parámetros de paginación.

VI. Flujo de la Aplicación

- 1. Inicio:** Consulta a la API con el tag darkcaline.
- 2. Procesamiento de datos:** Filtrado de tags prohibidos.
- 3. Presentación:** Vista Index muestra galería con imágenes.
- 4. Detalle:** Vista Details presenta datos filtrados y seguros.

VII. Archivos Clave

- **Program.cs:** Configura servicios, rutas y HttpClient.
- **Services/E926Api.cs:** Lógica de conexión a la API, URLs y filtrado.
- **Models/PostDtos.cs:** Clases que representan la estructura de datos.
- **Controllers/PostsController.cs:** Controlador que gestiona rutas y búsqueda.
- **Views/Shared/_Layout.cshtml:** Plantilla base.
- **Views/Posts/Index.cshtml:** Vista de galería.
- **Views/Posts/Details.cshtml:** Vista de detalle.

VIII. Decisiones de Diseño

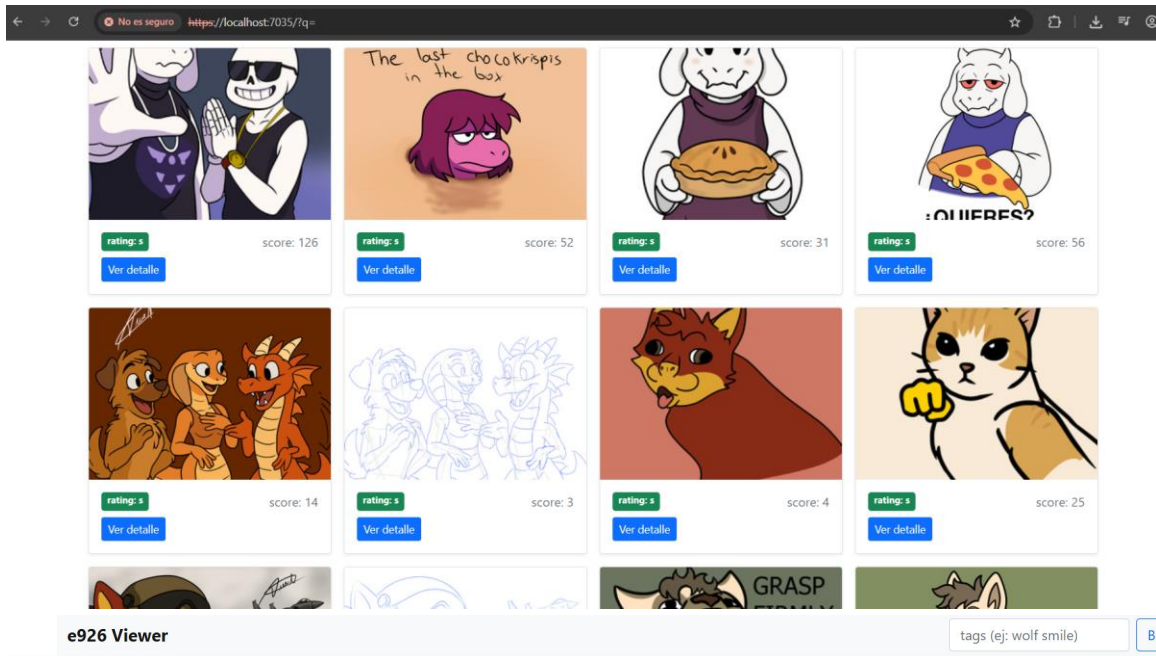
- **Foco en un único artista:** Solo se muestran publicaciones de darkalcaline.
- **Seguridad para público infantil:** Tags prohibidos y rating seguro.
- **Arquitectura escalable:** Separación clara de capas.
- **Rendimiento y uso responsable de la API:** Paginación y User-Agent personalizado.

IX. Flujo de Datos

1. Usuario accede a /Posts.
2. Controlador llama al servicio con tags fijos.
3. Servicio consume API, filtra tags y rating.
4. Modelos representan los datos.
5. Vista renderiza galería segura.

X. Pruebas y Evidencias

- **Listado general:** Publicaciones de darkalcaline.
- **Detalle:** Imagen grande, metadatos y tags filtrados.
- **Seguridad:** Todo rating:s y sin contenido sensible.
- **Rendimiento:** Paginación con 24 resultados por página.



Post #5632295

Rating: s

Score: 126 (129 / -3)

Dimensiones: 2500 x 3200 (png)

Tags (general): anthro, black_clothing, black_shirt, black_topwear, bone, clothing, cool_pose, duo, eyewear, fashion, female, hip_hop_style, male, parody, pose

Artista: darkcaline

Personajes: sans_(undertale), toriel

Especies: animated_skeleton, bovid, caprine, goat, humanoid, mammal, undead

Volver

XI. Mejoras Futuras

- Memoria caché.
- Sistema de favoritos.
- Estadísticas del artista.

- Soporte para otros artistas.
- Internacionalización.

XII. Conclusión

El proyecto demuestra cómo integrar una API pública en ASP.NET Core MVC con prácticas profesionales: separación de capas, vistas dinámicas y seguridad en datos. La solución está optimizada para mostrar contenido seguro de un artista, garantizando un entorno apto para menores y reforzando habilidades en programación orientada a la web.