

MANUAL TÉCNICO – MovieSeriesManagement

Introducción

Esta aplicación describe la arquitectura, componentes y funcionamiento del sistema **MovieSeriesManagement**, una aplicación desarrollada en C# usando **MVC**, parte de ASP.NET para la gestión de películas y series.

Requisitos del Sistema

2.1 Software

- **Framework:** .NET Core
- **Base de Datos:** SQL Server
- **Autenticación:** ASP.NET Identity

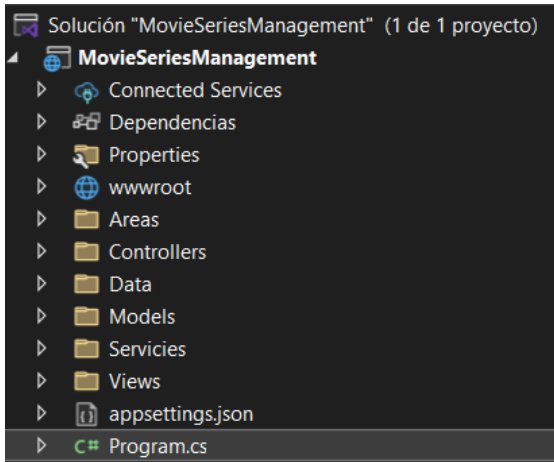
Herramientas de Desarrollo

- **IDE:** Visual Studio 2022 / Visual Studio Code / MVC
- **Gestor de Paquetes:** NuGet
- **Sistema de Control de Versiones:** Git
- **Arquitectura del Proyecto**

Patrón de Diseño

- El sistema sigue el patrón **MVC (Modelo-Vista-Controlador)** con una capa de **Repositorios** para la gestión de datos.

Estructura de Carpetas



Autenticación y Seguridad

Gestión de usuarios.

- Registro y login con validaciones.
- Asignación automática de roles ("User").
- Cierre de sesión seguro.

Funcionalidades Principales

Gestión de Usuarios

- Registro y autenticación.
- Perfiles de usuario con imagen.

Gestión de Contenido

- Búsqueda de películas y series por título, género y plataforma.
- Filtrado por tipo (película o serie).

Recomendaciones

- Basadas en el historial de visualización del usuario.
- Se almacenan en la base de datos.

Historial de Visualización

- Se almacena cada visualización del usuario.
- Se obtiene información sobre preferencias de género

Repositorios y Acceso a Datos

Implementación de Repositorios

Se utiliza el patrón **Repositorio Genérico** para la manipulación de datos.

Repositorio Base (Repository<T>)

- GetAllAsync(): Obtiene todos los registros.
- FindAsync(): Filtra registros con condiciones.
- GetByIdAsync(): Obtiene un registro por ID.
- AddAsync(), UpdateAsync(), DeleteAsync(): Manipulación de datos.

Repositorios Especializados

- ContentRepository: Búsqueda y filtrado de contenido.
- RecommendationRepository: Gestión de recomendaciones.
- ViewingHistoryRepository: Registro de historial y análisis de preferencias.