Proyecto 2, Mejoras I Cuatrimestre 2025

Arce Barquero Oscar Daniel, Facultad de Ingeniería informática
Miranda Chacón Ivania, Facultad de Ingeniería informática
Universidad Politécnica Internacional

Técnicas de Programación

Mora Umaña Luis Felipe

Abril de 2025

Contenido

Introducción

En el desarrollo de aplicaciones de gestión de contenido audiovisual como lo son plataformas como Disney, Netflix, entre otras, se ha experimentado un gran avance debido a la creciente demanda de plataformas que permiten organizar, calificar y visualizar películas y series.

El proyecto inicial nació como una solución web basada en el patrón MVC, utilizando bases de datos relacionales y tecnologías propias de ASP.NET Core.

Siguiendo las indicaciones actuales, se planteó una evolución sustancial del proyecto.

Este documento analiza los cambios técnicos y funcionales, y propone una arquitectura más modular, mantenible y orientada a la experiencia de usuario, adaptada a entornos de escritorio modernos.

Comparativa y Evolución del Proyecto

En el primer proyecto, se implementó un sistema basado en el patrón MVC (Modelo-Vista-Controlador), orientado a una aplicación web. Se utilizaron bases de datos relacionales para almacenar la información, incorporando datos reales tanto de usuarios como de contenido (películas y series). La estructura del proyecto estaba centrada en tecnologías web y la lógica de negocio se organizaba mediante controladores, vistas y modelos.

En contraste, el nuevo proyecto presenta una evolución significativa tanto en arquitectura como en funcionalidades. A continuación, se detallan los principales cambios y mejoras.

Cambios Técnicos y Funcionales

Nos enfocaremos en cambiar la arquitectura de este, y ahora se adoptará un enfoque de aplicación de escritorio (Windows Forms) usando .NET 6+.

Además de reemplazará el uso de bases de datos tradicionales por archivos JSON para el almacenamiento persistente de datos, y se organiza el código bajo principios de POO (Programación Orientada a Objetos) y Clean Code, separando responsabilidades mediante servicios, repositorios e interfaces.

Incorporación de Netwatch

Se integrará NetWatch para el monitoreo del sistema en tiempo real, permitiendo observar el rendimiento, errores y comportamiento del programa en ejecución.

Nuevas Funciones Administrativas

Se investigará para lograr que el administrador ahora tendrá la capacidad de generar reportes en formato PDF, incluyendo estadísticas de uso, contenido más visto, y actividad de los usuarios y también va a tener la posibilidad de visualizar y filtrar datos de forma más eficiente mediante formularios personalizados.

Interacciones del Usuario

Los usuarios ahora podrán tener la posibilidad de comentar películas y series, agregando una capa de interacción social al sistema y también van a poder ver y gestionar su historial de contenido visto y sus calificaciones.

Testing y calidad de software

Se trabajará en la incorporación de un sistema de testing con mocks, permitiendo pruebas unitarias que simulan el comportamiento de servicios y repositorios sin necesidad de datos reales. Esto asegura que los módulos funcionen correctamente de forma independiente.

Análisis

Evolución Arquitectónica

El cambio más representativo es la transición de una aplicación web a una aplicación de escritorio basada en Windows Forms, utilizando .NET 6+. Esta transformación implica una modificación total en la forma en que se gestionan las interfaces y la persistencia de datos, optando ahora por el uso de archivos JSON en lugar de bases de datos relacionales, lo cual permite un entorno más ligero, portable y con menor complejidad en su despliegue.

Mejora de la Estructura Interna

A nivel de arquitectura del código, se adopta una estrategia centrada en principios de Programación Orientada a Objetos (POO) y Clean Code, organizando el sistema a través de servicios, repositorios e interfaces. Esto facilita el mantenimiento, la escalabilidad del sistema y permite una mejor separación de responsabilidades, alineándose con buenas prácticas de desarrollo profesional.

Incorporación de Herramientas de Monitoreo

Se propone la integración de NetWatch para monitoreo en tiempo real del comportamiento de la aplicación. Esto permitirá a los desarrolladores y administradores observar métricas clave como rendimiento, errores y uso del sistema, anticipando problemas y mejorando la estabilidad general de la aplicación.

Funciones Administrativas Avanzadas

Se potencia el rol del administrador, quien ahora podrá generar reportes, visualizar estadísticas como el contenido más visto o la actividad de los usuarios, y utilizar formularios personalizados para filtrar y analizar información. Esto mejora la toma de decisiones y eleva la utilidad de la plataforma a nivel organizacional.

Interacción del Usuario

La experiencia del usuario se enriquece mediante la posibilidad de comentar contenido, gestionar su historial de visualización y administrar sus calificaciones. Estas nuevas funciones añaden una dimensión social al sistema, promoviendo el compromiso y la personalización de la experiencia.

Testing y Calidad del Software

Finalmente, se trabaja en la implementación de un sistema de testing basado en mocks, permitiendo ejecutar pruebas unitarias sobre componentes aislados del sistema. Esto garantiza una mayor fiabilidad del código y un desarrollo orientado a la calidad desde etapas tempranas.

Conclusión

La evolución de este proyecto nos permite identificar cómo una solución tecnológica puede adaptarse a nuevas demandas y paradigmas de desarrollo. La migración hacia una arquitectura modular, basada en buenas prácticas de ingeniería de software, no solo mejora el rendimiento y la mantenibilidad del sistema, sino que también incrementa su valor funcional para usuarios y administradores.

La incorporación de diversas herramientas, la interacción social entre usuarios y la capacidad de monitorear y probar el sistema de forma automatizada consolidan este proyecto como una solución madura, moderna y escalable.

Bibliografía

- Microsoft. (s.f.). *Guía de escritorio (.NET para Windows Forms)*. Microsoft Learn. https://learn.microsoft.com/es-es/dotnet/desktop/winforms/overview/?view=netdesktop-9.0
- Microsoft. (s.f.). *Novedades de Windows Forms para .NET 6*. Microsoft Learn. https://learn.microsoft.com/es-es/dotnet/desktop/winforms/whats-new/net60?view=netdesktop-9.0
- Gavilán, L. (2018, 10 de octubre). *Fundamentos de Mocks (Pruebas Unitarias)*. Gavilanch Blog. https://gavilanch.wordpress.com/2018/10/10/fundamentos-de-mocks-pruebas-unitarias/
- García, J. (2013, mayo 7). *Lo mejor que puedes leer para aprender diseño orientado a objetos*. Javier Garzás Blog. https://www.javiergarzas.com/2013/05/mejores-libros-orientacion-objetos.html
- CodeStack. (2020, 9 de abril). *MOCK en UNIT TEST para tratar con DEPENDENCIAS EN C#* [Video]. YouTube. https://www.youtube.com/watch?v=pgCMTsVlHrg
- Microsoft. (2023). *.NET documentation*. https://learn.microsoft.com/en-us/dotnet/
- Unit Testing Best Practices. (n.d.). *Microsoft Learn*. https://learn.microsoft.com/en-us/dotnet/core/testing/unit-testing-best-practices