

Proyecto #1 – TV Track

Valor: 27.5%

Fecha de entrega: 26 de marzo antes de las 6:00pm.

Grupos: Máximo 2 personas.

Elementos Generales

Enviar dentro de un .txt en Moodle el enlace al proyecto en Jira donde se organizó el proyecto y otro al repositorio del proyecto donde la versión final que se revisará deberá estar en el branch master. Este repositorio deberá tener no solamente el código fuente sino también una documentación técnica del proyecto: en un archivo PDF, Portada, Índice, decisiones de diseño, desarrollo donde se explique el cómo utilizar y ejecutar el programa, así como las diferentes interacciones de este, Análisis de Resultados/Aprendizajes/Conclusiones. El repositorio deberá estar organizado según lo visto en clases.

En la misma carpeta deberá haber un documento de planificación el cual será conocido como entregable #1 este deberá ser inicialmente adjuntado en Moodle una semana después de la publicación de este encabezado bajo la tarea Entregable #1. En este documento deberá estar en formato PDF, con Portada, índice, división del proyecto desde tareas grandes a subtareas más pequeñas (epic – proyecto –, features – funcionalidades a implementar –, PBIs – elementos de trabajo, un PBI puede estar compuesto de varios PBIs – tareas de implementación), bibliografía en caso de ser necesaria, conclusiones/Análisis de aprendizaje.

Problema por desarrollar

Utilice los principios de la programación orientada a objetos para desarrollar un sistema de gestión que permita a los usuarios registrar y gestionar las series, películas y etc que están viendo, proporcionando estadísticas y recomendaciones personalizadas, utilizando principios como Clean Code, MVC, POO, etc dentro del lenguaje C#. Recuerde esta es la primera iteración del proyecto por lo que su código debe ser extensible y será base para el proyecto #2. El sistema deberá permitir la gestión de usuarios, búsquedas de series y películas, recomendaciones basadas en lo que vio y estadísticas de una manera eficiente, intuitiva y organizada.

Dentro de los requisitos del sistema se encuentran los siguientes:

- Gestión de Usuarios: el sistema deberá tener dos tipos usuarios, usuario y administrador – con sus respectivos privilegios de uso –. Los usuarios administradores podrán editar, ver reportes, estadísticas agregar y modificar películas. Los usuarios comunes podrán tener sus diferentes playlist así como un historial de contenido visto, podrá brindar calificaciones y reseñas.
- El sistema deberá permitir filtrar contenido por nombre, género o plataforma de streaming.
- Los usuarios podrán marcar episodios o películas como vistos o parcialmente vistos, así como recomendaciones basadas en sus gustos e historial de usuario. Estos gustos podrían ser por género en esta iteración.
- Se generarán estadísticas en tiempo de visualización.

- Reportes: Se deberán generar los reportes desde el rol de administrador que tome en cuenta cantidad de usuarios en el último mes suscritos, interacción de los usuarios con la aplicación y por último géneros y series más populares en un determinado plazo de tiempo.

Recuerde utilizar el patrón Modelo-Vista-Controlador y que cualquiera de sus capas puede ser intercambiables. Para esta primera iteración se manejará mediante carga de archivos (csv, json) con la información básica del sistema, cada archivo deberá contener al menos 100 registros. Los reportes podrían ser mostrados en esta iteración mediante pantallas.

Recuerde que, en la segunda iteración de este proyecto, algunas especificaciones podrían cambiar, para esto se recomienda seguir Clean Code y buenas prácticas como, por ejemplo:

- Nombres significativos
- Métodos cortos y enfocados (Single Responsibility)
- Comentarios y documentación del código
- Manejo de Excepciones.
- SOLID

Aspectos por evaluar

Uso de herencia, polimorfismo, encapsulamiento, aspectos básicos de la programación orientada a objetos, SOLID, patrones de diseño (como mínimo 1 patrón de diseño), MVC, reutilización de código, mantenibilidad, escalabilidad, robustez, clean code, y demás conceptos vistos en clase.

Se solicita la utilización de Winforms para el desarrollo de los views.

El uso de Git será evaluado en base a la revisión del historial, así mismo el uso de Jira como herramienta de seguimiento del proyecto.

Las buenas prácticas de codificación serán evaluadas mediante el uso de Sonar Analyzer.

Se rebajarán puntos en caso de no seguir las instrucciones como por ejemplo formatos de envío, el plagio será evaluado y en caso de ser detectado se rebajarán los puntos respectivos.

El uso de un diagrama de Gantt con el entregable 1, para la estructuración de las tareas y el cumplimiento de este tendrá un valor adicional de 10% siempre y cuando la funcionalidad sea mayor al 80%

Escala

Por Evaluar	Valor
Entregable #1	10
Documentación final	10
Uso de patrones	10
SOLID, POO, Mantenibilidad, Clean Code	20
Interacciones básicas	45
Uso de Jira y Git	5
Total	100%

Rebajos mínimos

- El no anotarse en el Excel y participar en las actividades de daily meeting conllevara al rebajo de 5 puntos
- El no contar con sonar o alguna herramienta de revisión de código conlleva al rebajo de 10 puntos
- El no contar con archivos con al menos 100 registros conlleva al rebajo de 20 puntos. Solo los administradores podrían ser máximo 5.
- Todo proyecto deberá ser revisado en clases.

Anexos

Para el desarrollo del Entregable #1 pueden utilizar el siguiente patrón, LA INFORMACION AQUÍ MOSTRADA ES UN EJEMPLO DE ESTRUCTURA:

Epic1: Sistema de Ferretería los 3 patitos

Feature1.1: Reportes y Análisis

Este consiste en el análisis de los datos para la generación de los siguientes reportes: Ventas diarias entre las fechas, Ventas Mensuales, Ventas Anuales, etc...

Product Backlog Item 1.1.1: Generación de Reportes de ventas diarias

Descripción: Como desarrollador, se necesita del análisis de los datos que se encuentran en ventas para la generación de un reporte de ventas diarias dentro del sistema.

Criterio de Aceptación: El sistema genera una ventana con la información de las ventas diarias dado un rango de fechas, donde se validan fechas invalidas.

Tareas:

- Creación de Clase para cálculo de ventas.
- Creación de ventana para muestra de datos.
- Agregar ventana al menú principal de reportes.

Product Backlog Item 1.1.2: Descripción

Descripción Detallada

Criterio de Aceptación

Tareas: