ITLA

DEIVID TAVERAS

2023-0979

KELYN TEJADA BELLIARD

SISTEMA DE GESTION DE RESERVAS Y VENTAS DE BOLETOS CINE

PROGRAMACION III C3-2025

INDICE:

Introducción del Proyecto ----------3

Objetivo Detallado del Proyecto ----------3

Equipo Scrum----------4

Herramientas Utilizadas----------6

Épicas del Proyecto----------6

Épicas del Proyecto----------6

Ceremonias Scrum----------8

Lista de Requerimientos Funcionales y No Funcionales----------8

Criterios de Aceptación y Rechazo de Pruebas----------11

Herramientas de Pruebas----------13

Equipos de Pruebas y Responsabilidades----------14

Plan de Automatización de Pruebas----------14

Estrategia----------15

Introducción del Proyecto

El presente trabajo tiene como finalidad desarrollar y documentar un sistema web para la Gestión de Reservas y Ventas de Boletos de un cine, aplicando la metodología ágil Scrum. La idea surge debido a la necesidad de los cines modernos de ofrecer a los usuarios una plataforma rápida, organizada y fácil de usar para reservar boletos, seleccionar horarios y escoger asientos sin necesidad de hacer largas filas.

Como estudiantes de tecnología y desarrollo de software, es fundamental aprender a planificar, diseñar y construir sistemas utilizando metodologías ágiles, ya que estas se aplican ampliamente en la industria actual. Por esa razón, este proyecto no solo busca construir un sistema funcional en C# y .NET, sino también practicar la organización del trabajo mediante Scrum, utilizando herramientas reales como Jira y GitHub.

A lo largo de este documento se detallan cada uno de los elementos que conforman un proyecto Scrum: el objetivo del sistema, el alcance, la definición del equipo, las épicas, historias de usuario, actividades del cronograma y las ceremonias del marco de trabajo. Todo esto con el propósito de mostrar una visión completa de cómo se desarrolla un software profesional desde cero.

Objetivo Detallado del Proyecto

El objetivo principal del proyecto es desarrollar un sistema web que permita a los usuarios consultar la cartelera del cine, seleccionar una película, elegir un horario, ver la disponibilidad de asientos y realizar una reserva de manera rápida y segura. Además, el sistema tendrá un módulo administrativo que permitirá gestionar películas, salas, horarios y reportes de ventas.

De manera detallada, este proyecto busca:

- Facilitar el acceso a información actualizada de las películas en cartelera.

- Automatizar el proceso de reserva para evitar filas físicas y mejorar la experiencia del cliente.
- Permitir a los administradores controlar la información del cine desde un panel interno.
- Implementar un sistema seguro con autenticación de usuarios.
- Garantizar un diseño responsivo para que el sistema funcione tanto en computadoras como en dispositivos móviles.
- Aplicar buenas prácticas de programación usando C#, .NET y Entity Framework Core.
- Organizar el trabajo del equipo siguiendo los principios de Scrum y sus ceremonias.

En general, el sistema permitirá una interacción más fluida entre el cine y los usuarios, al mismo tiempo que representa un ejercicio práctico del ciclo de vida de desarrollo de software en un ambiente ágil.

Equipo Scrum

Scrum se basa en equipos pequeños y multidisciplinarios. Para este proyecto se definieron los siguientes roles:

Product Owner (PO)

El Product Owner es la persona responsable de que el proyecto tenga una dirección clara. Su rol es esencial porque entiende las necesidades del negocio y se asegura de que el producto final entregue valor. Entre sus funciones en este proyecto están:

- Gestionar el Product Backlog.
- Definir las historias de usuario.
- Priorizarlas según importancia y necesidad.
- Aclarar dudas del equipo durante el desarrollo.
- Validar que cada incremento desarrollado cumpla con los requerimientos.

El PO debe ser una persona organizada, comunicativa y capaz de tomar decisiones rápidamente para no detener el avance del equipo.

Scrum Master

El Scrum Master funciona como una guía dentro del equipo. No es un jefe, sino un facilitador que asegura que Scrum se implemente correctamente. En este proyecto sus responsabilidades incluyen:

- Dirigir las ceremonias Scrum (Sprint Planning, Daily, Review y Retro).
- Quitar impedimentos que afecten el avance del equipo.
- Promover la colaboración y el trabajo en equipo.
- Ayudar a que todos entiendan y apliquen los valores de Scrum.

Este rol necesita habilidades de liderazgo, comunicación y resolución de problemas.

Desarrollador Backend (.NET)

El desarrollador backend se encarga de toda la lógica interna del sistema. Algunas de sus tareas incluyen:

- Crear la estructura de la base de datos.
- Programar los modelos y entidades con Entity Framework.
- Construir las API para el login, cartelera, reservas y administración.
- Implementar la autenticación y autorización con JWT.
- Hay que asegurar que las reglas de negocio se cumplan correctamente.

Este rol requiere dominio de C#, .NET, SQL y conocimientos de arquitectura de software.

Desarrollador Frontend

El frontend es la parte visual del sistema. El estudiante encargado de este rol se ocupa de:

- Crear las pantallas de login y registro.
- Diseñar y mostrar la cartelera de películas.
- Desarrollar el mapa de asientos interactivo.
- Integrar el frontend con las APIs del backend.
- Crear las vistas del panel administrativo.

Debe manejar HTML, CSS, JavaScript y tecnologías web integradas con .NET.

QA Tester

El QA es quien verifica la calidad del software antes de entregarlo. Entre sus tareas están:

- Crear casos de prueba para cada historia de usuario.
- Detectar errores en la interfaz o en las APIs.
- Validar que los criterios de aceptación se cumplan.
- Usar herramientas de prueba como Postman.
- Reportar los bugs encontrados en Jira.

Este rol requiere atención al detalle y pensamiento crítico.

Herramientas Utilizadas

En un entorno ágil, las herramientas son clave para organizarse. Para este proyecto se usaron:

Jira

Sirve para gestionar el Product Backlog, las épicas, historias de usuario y los sprints. Permite ver el progreso en tiempo real.

GitHub

Controla las versiones del código y permite que varios desarrolladores trabajen al mismo tiempo.

Visual Studio 2022

IDE principal para escribir el código en C# y .NET.

SQL Server

Base de datos usada por el backend para guardar usuarios, películas, reservas, etc.

Postman

Herramienta para probar las APIs creadas en el backend.

Épicas del Proyecto

Las épicas representan grupos grandes de funcionalidades. Para este sistema se definieron:

ÉPICA 1: Gestión de Usuarios y Autenticación

Incluye todo lo relacionado con registro, login y seguridad.

ÉPICA 2: Gestión de Cartelera

Para visualizar películas, horarios, formatos y detalles.

ÉPICA 3: Sistema de Reservas

Permite al usuario seleccionar asientos y recibir un ticket.

ÉPICA 4: Administración del Cine

Incluye CRUD de películas, horarios y salas.

ÉPICA 5: Reportes y Métricas del Sistema

Permite ver ventas, asientos vendidos, reservas por día, etc.

Ceremonias Scrum

Sprint Planning

Reunión donde se seleccionan las historias que el equipo desarrollará en el sprint. El Product Owner explica las prioridades y el equipo estima el esfuerzo en puntos. Se generan tareas técnicas para cada historia.

Daily Scrum

Reunión diaria de 15 minutos donde cada miembro responde:

1. ¿Qué hice ayer?
2. ¿Qué haré hoy?
3. ¿Tengo algún impedimento?

Esto ayuda a mantener al equipo alineado.

Sprint Review

Es una demostración del trabajo terminado al Product Owner. Aquí se valida si las historias cumplen los criterios de aceptación.

Sprint Retrospective

El equipo reflexiona sobre el sprint para identificar:

- Qué se hizo bien
- Qué no salió como se esperaba
- Qué se puede mejorar

Es esencial para el aprendizaje continuo.

Lista de Requerimientos Funcionales y No Funcionales

Requerimientos Funcionales (RF)

Relacionados directamente con las historias de usuario.

RF1 – Registro de usuario (HU1)

El sistema debe permitir crear nuevos usuarios almacenando correo, nombre y contraseña.

RF2 – Inicio de sesión (HU2)

El sistema debe validar credenciales y permitir el acceso con autenticación.

RF3 – Visualización de cartelera (HU3)

El usuario debe poder ver las películas disponibles en la plataforma.

RF4 – Detalles de película (HU4)

El sistema debe mostrar horarios, sinopsis, duración y formato.

RF5 – Mapa de asientos (HU5)

El usuario debe visualizar los asientos disponibles, ocupados o reservados.

RF6 – Reserva de asiento (HU6)

El sistema debe guardar una reserva y verificar disponibilidad.

RF7 – Generación de ticket (HU7)

El ticket debe incluir información de película, sala, asiento y fecha.

RF8 – CRUD de películas (HU8)

El administrador puede agregar, modificar o eliminar películas.

RF9 – Gestión de salas y horarios (HU9)

El administrador debe definir nuevas salas y horarios sin duplicados.

RF10 – Reportes de ventas (HU10)

El sistema debe mostrar tickets vendidos y permitir exportación a PDF.

Requerimientos No Funcionales (RNF)

RNF1 – Rendimiento

- La cartelera debe cargar en menos de 3 segundos.
- Las solicitudes a la API no deben superar 1.5 segundos promedio.

RNF2 – Seguridad

- Cifrado de contraseñas (bcrypt o similar).
- Autenticación con JWT.
- Acceso restringido a módulos administrativos.

RNF3 – Usabilidad

- Interfaz intuitiva, clara y responsive.
- Compatible con celulares y computadoras.

RNF4 – Disponibilidad

- El sistema debe estar disponible un 95% del tiempo.

RNF5 – Escalabilidad

- El sistema debe permitir agregar más salas, horarios y películas sin modificar estructuras básicas.

Criterios de Aceptación y Rechazo de Pruebas

Los criterios definen cuándo una prueba se considera exitosa.

Criterios de aceptación

- Todas las funcionalidades deben ejecutarse sin errores visibles.
- Los datos deben registrarse correctamente en la base de datos.
- Debe cumplirse con el resultado esperado descrito en el caso de prueba.
- La interfaz debe responder sin tiempos excesivos.
- No deben existir brechas de seguridad evidentes.

Criterios de rechazo

- La funcionalidad no cumple lo esperado.
- Aparecen errores o fallas durante la prueba.
- La interfaz muestra comportamientos incorrectos.
- El sistema no valida datos o genera información incorrecta.
- El rendimiento está por debajo de lo esperado.

Herramientas de Pruebas

Postman (Pruebas de API)

- Permite validar endpoints del backend (.NET).
- Soporta pruebas automatizadas con scripts.
- Fácil de compartir colecciones entre testers.

SQL Server Management Studio (Validación de Base de Datos)

- Permite analizar si los datos se almacenan correctamente.
- Revisa consistencia en las tablas y relaciones.

Jira (Gestión de pruebas)

- Permite documentar casos de pruebas.
- Facilita el seguimiento de bugs y asignación.

Selenium (opcional para automatización UI)

- Realiza pruebas automáticas en navegadores.
- Permite simular interacciones del usuario.

Visual Studio Test Explorer

- Útil para pruebas unitarias con NUnit o xUnit.

Cronograma de Ejecución de Pruebas

ETAPA	ACTIVIDAD	TIPO	DURACIÓN ESTIMADA	RESPONSABLE
SEMANA 1	Pruebas del módulo de usuarios	Manual	2 días	QA
SEMANA 1	Pruebas API de autenticación	Manual/Automatizada	1 día	QA + Backend

SEMANA 2	Pruebas de cartelera y detalles	Manual	2 días	QA
SEMANA 2	Pruebas de reserva y mapa de asientos	Manual/Automatizada	3 días	QA + Frontend
SEMANA 3	Pruebas del módulo administrativo	Manual	3 días	QA
SEMANA 3	Pruebas de reportes	Manual	2 días	QA
SEMANA 4	Pruebas de rendimiento	Manual	1 día	QA
SEMANA 4	Pruebas de seguridad	Manual	1 día	QA
SEMANA 4	Automatización final	Automatizada	2 días	QA Automation
SEMANA 4	Corrección de errores	—	2 días	Equipo completo

Equipos de Pruebas y Responsabilidades

QA Tester (Principal)

- Diseñar y ejecutar casos de pruebas.
- Reportar bugs en Jira.
- Validar criterios de aceptación.

QA Automation

- Crear scripts de automatización para API y UI.
- Mantener el repositorio de pruebas automatizadas.

Backend Developer

- Dar soporte técnico al QA.
- Resolver defectos relacionados con la API.

Frontend Developer

- Corregir problemas de interfaz detectados en pruebas.
- Ayudar al QA en validación visual.

Product Owner

- Aceptar o rechazar las pruebas finales.
- Revisar resultados y validar cumplimiento del negocio.

Plan de Automatización de Pruebas

Pruebas que se automatizarán

- Login y registro (Postman o NUnit).
- Consulta de cartelera.
- Ver disponibilidad de asientos.
- Crear una reserva.
- Validar generación de ticket.
- Pruebas UI básicas con Selenium.

Estrategia

1. Crear scripts iniciales en Postman para validar endpoints.

2. Implementar pruebas unitarias en backend con NUnit.

3. Automatizar pruebas críticas de interfaz con Selenium WebDriver.

4. Integrar las pruebas en GitHub Actions (opcional).