

PROGRAMACIÓN 3: PROYECTO FINAL

PROYECTO: FILMASTER



Luis Javier Espinal Duran 2023-1055
2/12/24



Nombre:

Luis Javier

Apellido:

Espinal Duran

Matrícula:

2023-1055

Materia:

Programación 3

Profesor/a:

Kelyn Tejada

Fecha:

2/12/24

Índice

Nombre del proyecto de software:	3
Tecnologías aplicadas:	3
Lenguajes aplicados:	3
Objetivo del proyecto.....	3
Alcance del proyecto.....	4
Cronograma del proyecto:.....	4
Definición del primer Release:	5
Metodología Scrum:	6
Definir tareas a ejecutar:	6
Definición de equipo de trabajo:	6
Herramientas usadas:	6
Definir las épicas:	7
Ceremonias de Scrum:	7
Historias de usuario:	8
Plan de Pruebas:	9
Lista de requerimientos funcionales y no funcionales:.....	9
Criterios de aceptación y rechazo de pruebas:	9
Herramientas de prueba:	10
Plantillas para casos de pruebas:	10
Cronograma de Ejecución de Pruebas:	11
Equipos de pruebas y responsabilidades:	11
Conclusión:.....	12
Bibliografías:.....	12





Nombre del proyecto de software:

Filmaster




Modern UI/UX Movie Catalog
Web Application

Tecnologías aplicadas:

-  React Typescript
-  Tailwind CSS
-  Firebase Hosting
-  Firestore Database

Lenguajes aplicados:

-  JavaScript
-  HTML
-  TypeScript

Objetivo del proyecto

El objetivo de Filmmaster es proporcionar a los usuarios una experiencia moderna e intuitiva para explorar, organizar y disfrutar de una extensa colección de películas. La aplicación está diseñada para satisfacer las necesidades de cinéfilos y aficionados al cine, permitiéndoles buscar películas de manera eficiente, descubrir detalles ricos sobre sus títulos favoritos y personalizar su experiencia a través de herramientas como listas de seguimiento, favoritos y reseñas. Además, busca fomentar la interacción y la participación de los usuarios al permitirles calificar y reseñar películas, convirtiéndose en una plataforma que no solo organiza contenido, sino que también genera comunidad en torno al cine.

Alcance del proyecto

El sistema Filmmaster cubre la funcionalidad completa de un catálogo de películas accesible desde cualquier dispositivo gracias a su diseño responsivo. Permite a los usuarios buscar películas por título, género o miembros del reparto, proporcionando una experiencia personalizada al marcar películas como favoritas y crear listas de seguimiento. El sistema también incluye la capacidad de interactuar con el contenido mediante la calificación y reseña de películas. Técnicamente, se apoya en tecnologías modernas como React, TypeScript y Tailwind CSS para la interfaz de usuario, y en Firebase Hosting y Firestore Database para el almacenamiento y acceso rápido a los datos. Sin embargo, su alcance no incluye funcionalidades como transmisión de películas ni características avanzadas de redes sociales, manteniéndose enfocado en la organización y descubrimiento de contenido cinematográfico.

Cronograma del proyecto:

Actividad	Fecha de inicio	Fecha de fin	Responsable
Planificación del proyecto	26 de noviembre	26 de noviembre	Product Owner
Diseño de la interfaz (UI/UX)	27 de noviembre	28 de noviembre	Desarrollador
Configuración de herramientas y stack	27 de noviembre	28 de noviembre	Desarrollador
Desarrollo del componente de búsqueda	29 de noviembre	30 de noviembre	Desarrollador
Integración con Firestore Database	29 de noviembre	30 de noviembre	Desarrollador
Implementación de listas personalizadas	1 de diciembre	2 de diciembre	Desarrollador
Pruebas y depuración	1 de diciembre	2 de diciembre	Product Owner
Revisión y entrega final	2 de diciembre	2 de diciembre	Product Owner

Este cronograma sirvió para garantizar que la aplicación esté lista y funcional dentro del plazo establecido, asignando roles o actividades específicas al/los responsables/s para maximizar la eficiencia del trabajo.

Definición del primer Release:

Definición del primer Release:

Filmaster ofrecerá una experiencia básica pero funcional para que los usuarios exploren y gestionen un catálogo de películas.

El sistema permitirá a los usuarios buscar películas por título, género, visualizar detalles clave como sinopsis, fecha de lanzamiento, duración y calificación, así como marcar películas como favoritas y crear listas de seguimiento personalizadas.

También incluirá la funcionalidad para que los usuarios califiquen y reseñen películas.

Requerimientos funcionales:

1. **Exploración y búsqueda:** El sistema permitirá filtrar películas por título, género o miembros del reparto.
2. **Detalles de películas:** Los usuarios podrán acceder a información detallada, incluyendo sinopsis, fecha de lanzamiento, duración y calificación.
3. **Favoritos y listas personalizadas:** Se podrán marcar películas como favoritas y crear listas de seguimiento personalizadas.
4. **Calificaciones y reseñas:** Los usuarios podrán agregar una calificación y escribir reseñas para las películas.

Requerimientos no funcionales

1. **Rendimiento:** Las búsquedas y el acceso a los datos serán rápidos gracias a la integración con Firestore Database.
2. **Escalabilidad:** La arquitectura del sistema permitirá añadir nuevas funcionalidades en futuras versiones.
3. **Interfaz moderna:** Usará React y Tailwind CSS para ofrecer un diseño visual atractivo y fácil de usar.
4. **Seguridad:** La información de los usuarios será gestionada de manera segura a través de Firebase Authentication.

Metodología Scrum:

Definir tareas a ejecutar:

- ✚ Configurar entorno de desarrollo (React + Tailwind + Firebase).
- ✚ Implementar barra de búsqueda (componente de React).
- ✚ Diseñar interfaz responsiva para la lista de películas.
- ✚ Interacción de usuarios.
- ✚ Configurar Firestore para almacenar datos.
- ✚ Crear sistema de favoritos en Firebase.

Definición de equipo de trabajo:

Producto Owner (PO): Luis Espinal

- ✚ Priorizar las funcionalidades.
- ✚ Definir las historias de usuario y criterios de aceptación.

Scrum Master: Luis Espinal

- ✚ He de asegurar que el proyecto siga las prácticas Scrum.
- ✚ Organizar las reuniones y mantiene el flujo de trabajo.

Desarrollador: Luis Espinal

- ✚ Implementa las funcionalidades definidas en el backlog.
- ✚ Realiza pruebas y asegura la calidad del producto.

Herramientas usadas:

- ✚ Jira

Definir las épicas:

- ✚ Búsqueda y exploración de películas.
- ✚ Interacciones del usuario.
- ✚ Optimización y rendimiento.

Ceremonias de Scrum:

Sprint Planning (26 de noviembre)	Daily Standups (27 de noviembre al 2 de diciembre)	Sprint Review (2 de diciembre)	Sprint Retrospective (2 de diciembre)
Asignar las historias de usuario al sprint desde el Backlog.	Revisar el tablero Active Sprint.	Ver el tablero y validar que todas las historias en “Realizadas” cumplan con los criterios de aceptación.	Reflexionar sobre qué funcionó y qué mejorar.
Estimar las tareas y prioriza las que más impacto tienen.	Actualizar el estado de las tareas.		
	Tomar notas sobre impedimentos.		

Historias de usuario:

ID	Historia de Usuario	Criterios de Aceptación	Puntos
1	Como usuario, quiero buscar películas por título para encontrarlas fácilmente.	Debo poder escribir un título y obtener resultados relevantes en menos de 2 segundos.	3
2	Como usuario, quiero filtrar películas por género.	Al seleccionar un género, solo deben mostrarse las películas correspondientes.	2
3	Como usuario, quiero ver detalles de una película.	Al hacer clic en una película, debe mostrarse la sinopsis, fecha de lanzamiento, duración y calificación.	3
4	Como usuario, quiero marcar películas como favoritas.	Debo poder hacer clic en un ícono para agregar películas a una lista de favoritos.	3
5	Como usuario, quiero crear listas personalizadas.	Debo poder crear y nombrar listas para organizar películas.	5
6	Como usuario, quiero agregar calificaciones a películas.	Debo poder seleccionar una calificación de 1 a 5 estrellas y que se registre en la base de datos.	3
7	Como usuario, quiero reseñar películas.	Debo poder escribir comentarios sobre las películas y que se guarden.	5
8	Como usuario, quiero que la aplicación sea responsiva.	La interfaz debe ajustarse correctamente en dispositivos móviles, tabletas y computadoras.	8
9	Como usuario, quiero buscar actores específicos.	Debo poder buscar películas donde haya participado un actor en particular.	3
10	Como usuario, quiero que la aplicación cargue rápido.	La aplicación debe cargar en menos de 3 segundos, incluso con 50 elementos en pantalla.	5

Plan de Pruebas:

Lista de requerimientos funcionales y no funcionales:

Funcionales:

- ✚ Buscar películas por título.
- ✚ Filtrar películas por género.
- ✚ Mostrar detalles de una película.
- ✚ Marcar películas como favoritas.
- ✚ Crear listas personalizadas.
- ✚ Calificar y reseñar películas.

No Funcionales:

- ✚ La aplicación debe ser responsiva en todos los dispositivos.
- ✚ Carga inicial de la página en menos de 3 segundos con 50 elementos en pantalla.
- ✚ Respuesta a consultas de búsqueda en menos de 2 segundos.

Criterios de aceptación y rechazo de pruebas:

Aceptación:

- ✚ Todas las funcionalidades cumplen con los criterios definidos.
- ✚ La interfaz es completamente responsiva y funcional en diferentes navegadores y dispositivos.
- ✚ El tiempo de carga cumple con el estándar especificado.

Rechazo:

- ✚ Funcionalidades incompletas o con errores.
- ✚ Interfaz no responsiva o diseño inconsistente.
- ✚ Retrasos significativos en el tiempo de respuesta.

Herramientas de prueba:

Xray (Jira Test Management Plugins):

Xray permiten crear, gestionar y ejecutar casos de prueba directamente en Jira y facilita la integración de pruebas con historias de usuario, además de generar reportes claros.

Selenium (automatización):

Para pruebas funcionales automatizadas en la interfaz. Es fácil de integrar con frameworks de JavaScript y soporta pruebas en múltiples navegadores.

Plantillas para casos de pruebas:

Campo	Descripción
ID del Caso de Prueba	
Título	
Descripción	
Historia Asociada	
Precondiciones	
Pasos de la Prueba	
Resultado Esperado	
Resultado Real	
Estado	
Prioridad	
Evidencia	

Cronograma de Ejecución de Pruebas:

Fecha	Actividad	Tipo de Prueba	Responsable
28 de noviembre	Preparación de casos de prueba	Planeación	Luis Espinal
29 de noviembre	Ejecución de pruebas manuales	Manual (funcionales)	Luis Espinal
30 de noviembre	Validación de criterios no funcionales	Manual (rendimiento)	Luis Espinal
1 de diciembre	Automatización de pruebas críticas	Automatizadas (funcionales)	Luis Espinal
2 de diciembre	Revisión y análisis de resultados	Revisión final	Luis Espinal

Equipos de pruebas y responsabilidades:

- ❖ Luis Espinal (Yo) será el encargado de todas las fases de pruebas, desde la creación de casos hasta la ejecución y revisión.

Automatización de Pruebas:

- ❖ Escribir y mantener scripts de prueba usando Selenium.
- ❖ Diseñar pruebas automatizadas para flujos repetitivos o críticos.

Gestión de Defectos:

- ❖ Registrar defectos detectados durante pruebas manuales o automatizadas.
- ❖ Priorizar y asignar defectos según su severidad.

Conclusión:

El desarrollo de este sistema de gestión de películas garantiza funcionalidad, diseño y experiencia del usuario. Utilizando metodologías ágiles como Scrum, se han implementado características que responden a las necesidades iniciales del usuario, también sientan las bases para futuras expansiones y mejoras. Funciones como la búsqueda avanzada, la creación de listas personalizadas, y la capacidad de marcar favoritos para ofrecer una experiencia personalizada y eficiente.

El uso de tecnologías modernas como React, Firebase y herramientas para pruebas automatizadas como Selenium asegura que el sistema sea escalable, responsivo y de calidad. La integración de procesos de pruebas y gestión ágil mediante Jira refuerza el compromiso con cada etapa del desarrollo.

Bibliografías:

<https://github.com/JaviTheLovely/Proyecto-Final-Pro3-Filmaster>

<https://luis-final-proyect.atlassian.net/jira/software/projects/SCRUM/summary>