**Un dibujo animado con letras

Descripción generada automáticamente con confianza media**

**BUTTERPOP - DIGITAL DREAM**

**INTEGRADORA II**

**R E P O R T E T É C N I C O**

**TÉCNICO SUPERIOR UNIVERSITARIO EN TECNOLOGÍAS DE LA INFORMACIÓN ÁREA DESARROLLO DE SOFTWARE MULTIPLATAFORMA**

*P R E S E N T A*

**ANDREA DOMÍNGUEZ ZENTENO**

**EMANUEL GARCIA CAPOTE**

**JESÚS MIGUEL ROSALES MURILLO**

**JOEL GONZÁLEZ CRUZ**

**JOSÉ JULIÁN MARTÍNEZ DE LA CRUZ**

ASESORA DE LA ORGANIZACIÓN: DRA. MORAMAY RAMÍREZ HERNÁNDEZ

ASESORA ACADÉMICA: MTRA. SILVIA MURILLO PAZARÁN

ORGANIZACIÓN: “DIGITAL DREAM”  
GENERACIÓN: ENERO 2023-DICIEMBRE 2023

CUATRIMESTRE DE TÉRMINO: SEPTIEMBRE-DICIEMBRE 2024

**ÍNDICE**

# **INTRODUCCIÓN**

Cinemas AJEM es una empresa mexicana líder en entretenimiento dedicada al desarrollo y operación de complejos múltiples de exhibición cinematográfica. Con más de 23 años en el mercado, se posiciona en el top 10 de las mejores cadenas cinematográficas del mundo, contando con 335 complejos y 2,898 pantallas en 98 ciudades de la República Mexicana.

El departamento de ventas de Cinemas AJEM se encarga no solo del manejo y control de todas las ventas registradas dentro del cine, sino que también participa en diversas actividades con el objetivo de promover la compra de sus servicios y productos por parte de los clientes, tales como la venta de comida, boletería, dulcería, etc.

Respecto a estas operaciones, existe un problema relacionado al apartado de boletería, ya que con la llegada de las plataformas de streaming, su público fuente prefiere ver películas vía online en lugar de acudir al cine, lo que reduce considerablemente sus ventas.

Basándose en este problema surgirá la necesidad de incrementar sus ventas por medio de una aplicación multiplataforma amigable, segura, estable y fácil de usar que permitirá a los usuarios, que no lograron ver la película en su momento, o que prefieran volver a disfrutar de ella desde la comodidad de su casa, la compra/renta de las películas disponibles en su cartelera.

La implementación de esta aplicación podría aumentar considerablemente las ventas de las proyecciones del cine, ya que, además de permitir la compra/renta de películas de manera fácil para los usuarios, también brindará una experiencia única a los clientes permitiéndoles opinar, organizar sus películas en listas de reproducción, además les permitirá acumular puntos para canjearlos por cupones o descuentos, motivando así una visita próxima a sus sucursales.

Debido a la necesidad mencionada anteriormente, la participación de desarrolladores técnicos universitarios en el área de desarrollo de software multiplataforma de la Universidad Tecnológica de Tecámac será la adecuada, ya que poseerán los conocimientos necesarios en diseño, codificación, almacenamiento y manejo de datos para desarrollar la aplicación de manera efectiva y brindar una solución a la reducción de ventas del cine.

El presente documento tendrá como objetivo mostrar, explicar y analizar el proceso de realización de la app. Estará dividido en tres capítulos de gran importancia, ya que cada uno proporcionará información detallada sobre cómo se desarrollará la aplicación multiplataforma.

El capítulo uno se enfocará en la recolección y refinamiento de los requisitos sugeridos por el cliente. Esto incluirá identificar y comprender sus necesidades, establecer los roles de usuario para la aplicación y definir los requerimientos funcionales y no funcionales del sistema. Esta sección servirá como base para determinar los requerimientos de software y hardware necesarios para la realización de la aplicación. Además, se presentarán los diagramas de caso de uso, secuencia y actividad para visualizar el funcionamiento del sistema y la interacción de los usuarios con cada una de las actividades de este.

En el capítulo dos se abordará el diseño y desarrollo del proyecto. Aquí se diseñará la base de datos con las tablas necesarias para gestionar cada parte del sistema, como películas, clientes, listas, etc. También se creará un diccionario de datos y se diseñarán las pruebas necesarias para asegurar el correcto funcionamiento de la aplicación. Además, se realizará el maquetado de cada una de las interfaces.

Finalmente, el capítulo tres se centrará en la implementación y pruebas de la aplicación. Se documentará todo lo relacionado con las interfaces diseñadas en el capítulo dos y se llevarán a cabo las pruebas para garantizar el funcionamiento adecuado del sistema.

# **OBJETIVOS**

OBJETIVO GENERAL

Desarrollar una aplicación multiplataforma destinada a mejorar la experiencia de los clientes, permitiéndoles la compra y renta las películas de la cartelera, de modo que aquellos usuarios que no pudieron verlas en el cine puedan disfrutarlas en cualquier momento o durante el período de renta, además el desarrollo de esta aplicación podría incrementar significativamente las ventas del cine.

OBJETIVOS ESPECÍFICOS

Realizar la base de datos que almacene la información relacionada con los administradores, clientes y el catálogo de películas, lo cual será fundamental para la codificación y elaboración de las interfaces.

Diseñar las interfaces de los administradores y clientes con un diseño práctico y atractivo, para posteriormente llevar a cabo las pruebas.

Probar la funcionalidad de las interfaces y registrar los errores que puedan ocurrir, con el objetivo de solucionarlos rápidamente y entregar un trabajo de calidad.

# **PROGRAMA Y CRONOGRAMA**

******

|  |
| --- |
| **UNIVERSIDAD TECNOLÓGICA DE TECÁMAC** |
| ***DIVISIÓN TIC*** |
| ***PROGRAMA DE ESTADÍAS PROFESIONALES*** |

|  |  |  |
| --- | --- | --- |
| ***PROGRAMA DE TRABAJO*** |  | |
|  | | *FECHA: dd/mm/aaaa* |
|  |  |  |

*DATOS DEL O DE LA* *ESTUDIANTE*

|  |  |
| --- | --- |
| NOMBRE: | Domínguez Zenteno Andrea  Garcia Capote Emanuel  González Cruz Joel  Martínez de la Cruz José Julián  Rosales Murillo Jesús Miguel |
| DIVISIÓN: | Tecnologías de la Información y Comunicación |
| CARRERA: | Técnico Superior Universitario en Tecnologías de la Información Área Desarrollo de Software Multiplataforma |
| MATRÍCULA: | 2523260004  2523260069  2523260021  2523260022  2523260068 |
| GENERACIÓN: | Enero-Abril 2023 |

*ASESOR ACADÉMICO O ASESORA ACADÉMICA*

|  |  |
| --- | --- |
| NOMBRE: | LDA. Sandra Raquel López Arce Soroa |
| CARGO: | Profesora de Asignatura |

*DATOS DE LA ORGANIZACIÓN*

|  |  |
| --- | --- |
| NOMBRE DE LA ORGANIZACIÓN: | Universidad Tecnológica De Tecámac |
| DEPARTAMENTO: | Tecnologías de la Información y Comunicación |
| ÁREA: | Tecnologías de la Información Área Desarrollo de Software Multiplataforma |
| DIRECCIÓN: | Carretera Federal México - Pachuca Km 37.5, 55749 Estado de México, State of Mexico |
| TELÉFONO: | 55-64-99-76-32 |
| E-MAIL: | dtic@uttecamac.edu.mx |

*ASESOR DE LA ORGANIZACIÓN O ASESORA DE LA ORGANIZACIÓN*

|  |  |
| --- | --- |
| NOMBRE: | Silvia Murillo Pazaran |
| CARGO: | Profesora de Tiempo Completo |

*PERÍODO*

|  |  |
| --- | --- |
| DURACIÓN: | 15 semanas |
| FECHA DE INICIO: | 07 de mayo de 2024 |
| FECHA DE TERMINACIÓN: | 15 de agosto de 2024 |
| HORARIO: | 10am a 6pm |

*PROYECTO*

|  |  |
| --- | --- |
| NOMBRE: | BUTTERPOP |
| DESCRIPCIÓN: | El presente proyecto consiste en la elaboración de una aplicación multiplataforma que sea auxiliar a la empresa Cinemas AJEM, con el fin de incrementar sus ventas y sea de alternativa a los clientes como una forma de ver películas en estreno. |
| OBJETIVO GENERAL: | Desarrollar una aplicación multiplataforma destinada a mejorar la experiencia de los clientes, permitiéndoles la compra y renta las películas de la cartelera, de modo que aquellos usuarios que no pudieron verlas en el cine puedan disfrutarlas en cualquier momento o durante el período de renta, además el desarrollo de esta aplicación podría incrementar significativamente las ventas del cine. |
| OBJETIVOS ESPECÍFICOS: | Realizar la base de datos que almacene la información relacionada con los administradores, clientes y el catálogo de películas, lo cual será fundamental para la codificación y elaboración de las interfaces.  Diseñar las interfaces de los administradores y clientes con un diseño práctico y atractivo, para posteriormente llevar a cabo las pruebas.  Probar la funcionalidad de las interfaces y registrar los errores que puedan ocurrir, con el objetivo de solucionarlos rápidamente y entregar un trabajo de calidad. |
| ALCANCE(S): | La aplicación podrá usarla administradores de Cinemas AJEM para la gestión de películas en estreno de temporada. De igual forma será adaptada a los usuarios que quieran una alternativa de rentar y/o comprar películas. |
| META(S): | La aplicación cumplirá en su totalidad a fin de cuatrimestre con las expectativas del cliente, ayudando a la incrementación de sus ventas. |
| RECURSOS: | Hardware: Computadoras, memorias USB y router con conexión a internet.  Software: Android Studio, Visual Studio, PostgreSQL, Testlink, MantisBT. |

*PLAN DE TRABAJO*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| ACTIVIDAD | | DESCRIPCIÓN | SEMANA | | FECHAS | |
| INICIO | TÉRMINO | INICIO | TÉRMINO |
|  |  | Requerimientos del cliente | 1 | 1 | 07/05/2024 | 14/05/2024 |
|  |  | Especificar Alcance del proyecto | 1 | 1 | 07/05/2024 | 14/05/2024 |
|  |  | Preparación de hardware y software | 1 | 1 | 07/05/2024 | 14/05/2024 |
|  |  | Diagrama de Casos de Uso | 2 | 3 |  |  |
|  |  | Especificación Casos de Uso | 2 | 3 |  |  |
|  |  | Modelado de base de datos | 3 | 4 |  |  |
|  |  | Programacion de aplicacion | 5 | 8 |  |  |
|  |  | Diseño de interfaces | 8 | 10 |  |  |
|  |  | Entrega de avance del proyecto | 13 | 13 |  |  |
|  |  | Implementación y pruebas | 13 | 14 |  |  |
|  |  | Entrega de documentación final | 14 | 15 |  |  |

*CRONOGRAMA DE ACTIVIDADES (PROGRAMA)*

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| # | ACTIVIDADES | CONTROL | Mes 1 | | | | | Mes 2 | | | | Mes 3 | | | | Mes 4 | |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
|  | Exploración | PROG. |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| REAL |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | Iniciación | PROG. |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| REAL |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | Producto | PROG. |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| REAL |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | Estabilización | PROG. |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| REAL |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | Pruebas | PROG. |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| REAL |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

*FIRMAS*

|  |  |
| --- | --- |
| Lda. Silvia Murillo Pazaran  *ASESORA O ASESOR DE LA ORGANIZACIÓN* | |
| Ángel Omar Flores Hernández  *ESTUDIANTE* | Lda. Sandra Raquel López Arce Soroa  *ASESORA ACADÉMICA O ASESOR ACADÉMICO*] |

# **MARCO TEÓRICO**

Una aplicación multiplataforma es una pieza de software diseñada para funcionar en múltiples sistemas operativos y dispositivos sin la necesidad de diseñar versiones diferentes para cada plataforma. Esto significa que el mismo código fuente puede ser usado en diferentes entornos, tales como: Android, iOS, Windows, macOS, Lunix, entre otros.

En los años 90's, estas aplicaciones surgen como respuesta a la necesidad de las empresas de alcanzar una audiencia más amplia, reducir costos y tiempo de desarrollo. Algunas de las razones por las cuales se usan aplicaciones multiplataforma son.

1. Diversidad de dispositivos y sistemas operativos

Con la proliferación de dispositivos móviles y una amplia variedad de sistemas operativos, presenta el desafío de desarrollar aplicaciones nativas para cada una de las plataformas, lo que implica disponer de múltiples equipos de desarrollo, lo que significaba mayores costos y cantidad de tiempo para las empresas en desarrollar dicha aplicación para cada plataforma.

1. Eficiencia en el desarrollo

Desarrollar una aplicación para cada plataforma es costoso, lento e ineficiente. Las aplicaciones multiplataforma permiten escribir un único código fuente y utilizarlo en múltiples entornos dependiendo de las necesidades y capacidades del dispositivo. Esto acelera en gran medida el desarrollo, facilita el mantenimiento y reduce los costos de la aplicación.

1. Tecnologías web y móviles

El desarrollo de lenguajes, frameworks y herramientas han hecho posible la creación de aplicaciones eficientes en diversas plataformas. Estas tecnologías permiten a los desarrolladores acceder a características nativas de cada sistema operativo con un solo código fuente.

1. Demanda del mercado

Los consumidores esperan que las aplicaciones estén disponibles en sus dispositivos y estos mismos pueden variar de persona a persona. Las empresas desean llegar a audiencias cada vez más grandes, por lo que necesitan asegurarse de que sus aplicaciones estén disponibles para la mayor cantidad de dispositivos y plataformas posibles para así cumplir con los objetivos planteados por la propia empresa.

Larraín (2023) expone que "La gente ve más contenidos y conoce las plataformas, eso es lo positivo, pero lo negativo es que por un tiempo estuvieron los cines cerrados y eso produjo un cambio cultural en cómo la gente se relaciona con la idea de ir al cine. Hoy en día las personas prefieren consumir contenido de entretenimiento mediante plataformas de streaming a través de sus dispositivos móviles ya que además de la comodidad, le brindan una experiencia al usuario." La experiencia del usuario es fundamental en este tipo de aplicaciones. Una interfaz amigable, intuitiva y segura puede mejorar significativamente la satisfacción del usuario y fomentar su lealtad.

El área de desarrollo de software multiplataforma es justamente, el área que permite crear este tipo de aplicaciones que funcionan en diversos dispositivos y sistemas operativos, optimizando los recursos y alcanzando a una mayor audiencia. Herramientas como Xamarin son populares en este ámbito, proporcionando soluciones eficientes y de alta calidad.

En cuanto a la atracción y retención audiencia, en la industria del entretenimiento, es crucial diversificar los productos y servicios para atraer a diferentes segmentos de clientes. Ofrecer descuentos y cupones a través de la plataforma es una estrategia efectiva para lograrlo.

Para desarrollar una aplicación multiplataforma, eficiente, atractiva y escalable se hará uso de las siguientes herramientas y tecnologías:

Apache NetBeans

Es un entorno de desarrollo, plataforma de herramientas y marco de aplicación.  
Se utilizará este entorno de desarrollo debido a su completo entorno de ejecución y asistencias de depuración.

Visual Studio Code

Es un editor de código fuente liviano pero potente que se ejecuta en el escritorio y está disponible para Windows, macOS y Linux. Viene con un soporte integrado para JavaScript TypeScript y Node.js y tiene un rico ecosistema de extensiones para otros lenguajes y tiempos de ejecución, como C++, C#, Java, Python, PHP, Go y .Net.  
Se hará uso de esta herramienta debido a su flexibilidad, simpleza y su soporte para diferentes lenguajes de programación.

Android Studio

Es el IDE oficial para el desarrollo de Android e incluye todo lo necesario para compilar Apps para Android.

Se emplea este entorno de desarrollo debido a que se debe garantizar la compatibilidad con dispositivos Android además de facilitar su desarrollo para las mismas.

Xamarin

Es un marco de interfaz de usuario de código abierto. Xamarin.forms permite a los desarrolladores compilar aplicaciones en Android, iOS y Windows desde un código base compartido. Xamarin.forms permite a los desarrolladores crear interfaces de usuario XAML con código subyacente en C#. Estas interfaces se representan como controles nativos con mejor rendimiento en cada plataforma.

Se utilizará este Framework ya que es requisito que la aplicación sea multiplataforma, usando como entorno de desarrollo Android Studio para posteriormente ser compilada a diversos sistemas operativos móviles de manera sencilla, rápida y eficiente.

git

Es un sistema de control de versiones distribuido, cada desarrollador tiene una copia integral del mismo. A diferencia de los sistemas de control de versiones centralizados los DVCS necesitan una conexión constante a un repositorio central. Git es un sistema de control de versiones distribuido más popular y se utiliza para el desarrollo de proyectos de código abierto.

Se hará uso de esta herramienta para facilitar y agilizar un flujo colaborativo para el equipo, además de mejorar el versionado del proyecto y aumentar la eficiencia de correcciones, implementaciones y reducir tiempos de desarrollo.

MongoBD

Es una base de datos de documentos que ofrece una gran escalabilidad y flexibilidad y un modelo de consultas e indexación avanzada.

Se hará uso de este gestor de base de datos no relacional para poder almacenar grandes volúmenes de datos cuya información no sea sensible.

PostgreSQL

Es un potente sistema de base de datos relacional de objetos de código abierto con más de 35 años de desarrollo activo que le ha ganado una sólida reputación por su confiabilidad solidez de funciones y rendimiento.

Se emplea este gestor de base de datos relacional para tener un control preciso de aquellos registros que requieran ser protegidos ya que estos pueden presentar información sensible.

StarUML

Es un sofisticado modelador de software destinado a soportar un modelado ágil y conciso.  
Se utilizará este programa para modelar diagramas de casos de uso, de secuencia, actividades, entre otros, para sustentar el desarrollo del proyecto.

Testlink

Es un sistema de gestión de pruebas basado en la web que facilita el control de calidad del software. Ofrece soporte para casos de prueba, conjuntos de pruebas, planes de pruebas, proyectos de prueba y gestión de usuarios, así como diversos informes y estadísticas.

Se usará este programa en la fase de pruebas para así garantizar la calidad, funcionalidad y consistencia de comportamiento del proyecto, para así hacer las respectivas correcciones o robustecer el proyecto o ciertas partes de este.

MantisBT

Es un software que constituye una solución completa para gestionar tareas en un equipo de trabajo. Es una aplicación OpenSource que se utiliza para probar soluciones automatizadas, llevando un registro histórico de las alteraciones y gestionando equipos de trabajo de forma remota.

Se emplea este software para tener un control de las fallas que pueda presentar la aplicación y tener un registro de dichas fallas y cómo se podrían solucionar.

Java

Es un lenguaje de programación de alto nivel, compilado y caracterizado por su enfoque en la orientación a objetos, lo que facilita la creación de aplicaciones modulares y fomenta la reutilización de código.

Se utilizará este lenguaje debido a su potencia y consistencia en la sintaxis además de permitir el uso de las Java Server Pages.

JavaScript

Es un lenguaje de programación interpretado, dialecto del estándar ECMAScript. Se define como orientado a objetos, basado en prototipos, imperativo, débilmente tipado y dinámico.  
Se utilizará este lenguaje en el front-end o del lado del cliente para aportar un mayor dinamismo a la aplicación además de reducir costos de procesamiento y energéticos a los servidores que se vayan a emplear.

HTML 5

El Hypertext Markup Language (HTML), en su quinta edición (HTML5), es el lenguaje de marcado estándar utilizado en la construcción y diseño de websites. Debido a su versatilidad, estabilidad y soporte universal.

Se hará uso de este lenguaje para definir la estructura de cada una de las interfaces de la aplicación debido a su modelo de cajas.

CSS 3

Cascade Style Sheet u "Hojas de Estilo en Cascada", se trata de la tercera revisión del lenguaje de hoja de estilos utilizado para describir la presentación de un documento escrito en HTML o XML (incluyendo formatos XML como SVG, XHTML o MathML). El CSS3 es una actualización significativa de la versión anterior, CSS2, e introduce una serie de nuevas características y mejoras.

Se utilizará esta tecnología para definir estilo a las interfaces y estas resulten ser agradables y atractivas para los usuarios, además de que esta presenta diversas opciones y funciones integradas que desencadenan en una enorme variedad de diseños.

JSP

JavaServer Pages (JSP) es una tecnología que ayuda a los desarrolladores de software a generar contenido web dinámico, interactivo, y basado en HTML, XML o cualquier otro tipo de documento dinámico.

PHP

Es un lenguaje de programación de código abierto, ampliamente utilizado por desarrolladores web y es el fundamento de muchas plataformas robustas. Proporciona una forma eficiente y eficaz de desarrollar sitios web dinámicos e interactivos.  
Se utilizará este lenguaje en el lado del back-end debido a su tipado dinámico y rápida implementación en el mismo.

Node.js

Es un entorno de ejecución de JavaScript que se utiliza para desarrollar aplicaciones de servidor. Impulsado por el motor V8 de Google, Node.js utiliza un modelo de programación orientado a eventos y entradas/salidas (I/O) no bloqueantes, lo que lo hace ligero y eficiente, perfecto para aplicaciones en tiempo real con intercambio intenso de datos a través de dispositivos distribuidos.

Se utilizará este lenguaje debido a que su sintaxis es idéntica a la de JavaScript por lo que su implementación en el lado del back-end o del servidor será rápida, eficiente y potente.

Figma

Figma es un editor de gráficos vectorial y una herramienta de generación de prototipos, principalmente basada en la web, con características off-line adicionales habilitadas por aplicaciones de escritorio en macOS y Windows.

Se usará este programa para realizar el maquetado de diseño de interfaces.

# **METODOLOGÍA**

Dentro del ámbito del desarrollo de aplicaciones móviles, elegir una metodología se toma de gran importancia, debido a que el equipo se puede sincronizar y planificar las fases del proyecto con mayor facilidad.

La metodología implementada a este proyecto es Mobile-D dada la preferencia en aplicaciones móviles y que está enfocada a equipos de menor tamaño para lograr ciclos de desarrollo muy rápidos. En el proyecto, es conveniente tener este tipo de metodología ágil para lapsos de tiempo cortos.

Las fases de Mobile-D se basan en un modelo de planificación y entrega:

Fase de exploración: Aquí se define el alcance del proyecto con planificación y atención a los conceptos básicos del proyecto. La aplicación se enfoca al mundo del entretenimiento, por lo que conocer las funcionalidades cercanas a los usuarios no es complejo.

Fase de iniciación: En esta fase, se preparan los materiales, software y hardware para empezar el proyecto, teniendo un plazo de planificación y el resto de trabajo.

Fase de producto: De igual manera, se tiene un plazo de planeación como retroalimentación de lo anterior y definición de al menos una prueba que verifique el funcionamiento del proyecto. Seguido, se tiene un último plazo de trabajo enfocado en la implementación, equivalente al código y diseño de la aplicación por módulo.

Fase de estabilización: Se realiza la integración del proyecto, es decir, módulos separados que deben juntarse para tener una versión completa del proyecto. Así mismo, se entrega documentación del proyecto en donde se describan y exhiban los pasos y descripciones de todo el desarrollo de la aplicación.

Fase de pruebas: Una vez pasada una prueba general de la aplicación, se llega a un plazo de testeo hasta llegar a una versión adecuada a las necesidades del cliente y funcionalidades requeridas por el mismo. Se corrigen fallos, se documentan, pero no se agrega más funcionalidades.

CAPÍTULO 1 RECOLECCIÓN Y REFINAMIENTO DE REQUISITOS

## 1.1 Necesidades del cliente

El departamento de ventas de Cinemas AJEM necesita una aplicación multiplataforma llamada "ButterPop" que abarque tanto la compra y renta de las películas más recientes de su cartelera, como la interacción con los usuarios, brindándoles una experiencia amigable. Por lo que proponen que la misma cuente con dos interfaces clave para lograr esto.

La primera interfaz está destinada a administradores y superadministradores. Permite la gestión eficiente del catálogo de películas del cine. Los administradores pueden introducir información como la imagen relacionada a la película, el género, duración, título, clasificación, y precio.

La segunda interfaz está diseñada para mejorar la experiencia de los usuarios. Permitiéndoles registrarse, consultar el catálogo de películas, la compra/renta de estas, la creación de listas de reproducción o colecciones de sus películas favoritas o películas pendientes por ver, puntuar y comentar, así como permitirles la obtención de puntos.

El acceso a ambas interfaces depende de un mismo login, sin embargo, para acceder como administradores deben hacerlo por medio de usuarios previamente creados de manera manual por los superadministradores.

## 1.2 Identificación y definición de roles de usuario

Dentro de la aplicación "ButterPop" existen dos roles de usuario fundamentales para el correcto funcionamiento de esta.

1. Cliente: Este usuario se tendrá que registrar por medio de correo electrónico y contraseña, e iniciar sesión para poder rentar, comentar, puntuar o guardar una película en alguna lista. Sobre las listas, el usuario podrá crear diferentes dependiendo de sus necesidades. Para poder rentar una película, el cliente tendrá que vincular una forma de pago para poder adquirir la película.
2. Administrador: El administrador será el responsable de la gestión de la aplicación "pendiente" de manera que pueda manipular la base de datos para la gestión de películas, usuarios y las rentas que el usuario realice.

## 1.3 Requerimientos Funcionales

|  |
| --- |
| Registro de clientes |
| El sistema deberá permitir el registro de clientes interesados con correo y contraseña como parte del registro. |

|  |
| --- |
| Inicio de sesión de clientes |
| El sistema deberá permitir el inicio de sesión de clientes con su correo y contraseña previamente registradas. |

|  |
| --- |
| Renta de películas |
| El sistema deberá permitir la renta de películas en estreno a usuarios registrados en la aplicación siempre y cuando vinculen un método de pago. |

|  |
| --- |
| Comentar y puntuar |
| El sistema deberá permitir a los usuarios con cuenta puntuar las películas, así como comentarlas. |

|  |
| --- |
| Gestionar listas |
| El sistema deberá permitir a los usuarios visualizar, crear, eliminar o actualizar listas personalizables en donde puedan guardar películas. |

|  |
| --- |
| Inicio de sesión de administradores |
| El sistema deberá permitir a los administradores iniciar sesión con un correo y contraseña previamente registrada en la base de datos. |

|  |
| --- |
| Gestionar películas |
| El sistema deberá permitir a los administradores agregar, actualizar y eliminar películas. |

## 1.4 Requerimientos No Funcionales

|  |
| --- |
| Encriptar datos |
| El sistema deberá encriptar datos sensibles tanto del administrador como el cliente, por ejemplo, contraseñas e información bancaria. |

|  |
| --- |
| Tiempo de respuesta |
| El sistema deberá tener un tiempo de respuesta entre cada actividad de máximo cinco segundos. |

|  |
| --- |
| Diferenciación administrador y cliente |
| El sistema deberá diferenciar el inicio de sesión entre un cliente y un administrador. |

|  |
| --- |
| Diseño |
| El sistema deberá tener un diseño responsivo, agradable a la vista del cliente en cuanto a color, tipografía y acorde a la aplicación. |

|  |
| --- |
| Usabilidad |
| El sistema deberá ser de fácil navegación a los usuarios. |

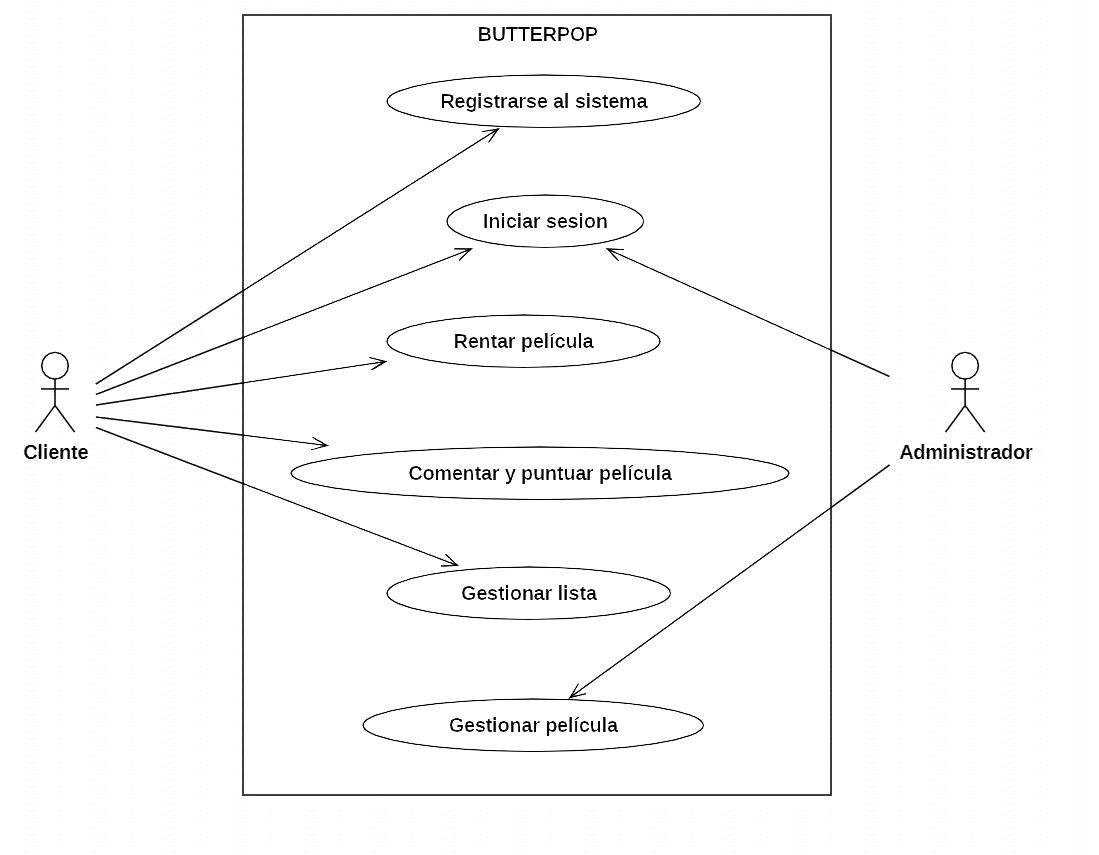
## 1.5 Requerimientos De Hardware

* Computadoras con mínimo de 16gb RAM y 1TB de almacenamiento. Con monitor de mínimo 22 pulgadas y procesador Intel Core i5 6ta Gen. Para uso general en el proyecto, programación, diseño y documentación.
* Memorias USB de mínimo 8gb de almacenamiento de la marca Kingston. Se utilizarán como medio de guardado de archivos móviles.
* Router o modem de cualquier compañía que ofrezca velocidad de internet de mínimo 61,82 Mb/s.

## 1.6 Requerimientos De Software

* Conexión a internet por medio del router con mínima velocidad de 61,82 Mb/s.
* PostgreSQL: Será el gestor de base de datos del proyecto, dada la estructura que facilita a las bases de datos relacionales.
* Android Studio: Para la realización de la aplicación, tanto diseño como programación con modelo MVC.
* Apache NetBeans: Para la escritura de etiquetado HTML, CSS y JavaScript.
* Testlink: Gestor de casos de prueba en función de requisitos del cliente.
* MantisBT: Funcionará como gestor de reportes de errores o fallos de funcionalidad u ortográficos.

## 1.7 Diagrama General De Casos De Uso



## 1.8 Especificación De Los Casos De Uso

|  |  |  |
| --- | --- | --- |
| **CU-0001** | **Registrar usuario** | |
| **Versión** | 1.0 (15/05/24) | |
| **Autores** | Martínez de la Cruz José Julián | |
| **Fuentes** |  | |
| **Dependencias** | Ninguno | |
| **Descripción** | El sistema deberá comportarse tal cual se describe en el siguiente caso de uso cuando el usuario registra una cuenta en la aplicación multiplataforma | |
| **Precondición** | El usuario deberá visualizar la pantalla principal y la aplicación debe estar activa | |
| **Secuencia normal** | **Paso** | **Acción** |
| 1 | El usuario da clic en el botón “Registrarse” |
| 2 | El sistema muestra el formulario de registro |
| 3 | El usuario ingresa su nombre |
| 4 | El usuario ingresa su correo electrónico |
| 5 | El usuario ingresa su contraseña |
| 6 | El usuario confirma su contraseña |
| 7 | El sistema valida los datos |
| 8 | El usuario da clic en el botón “Registrarme” |
| **Postcondición** | Se guardan los datos de la nueva cuenta en la base de datos y el sistema manda al usuario a la interfaz principal | |
| **Excepciones** | **Paso** | **Acción** |
| 7 | Si existe falta de datos, invalidación de datos o el correo electrónico ya ha sido registrado anteriormente, el sistema muestra el mensaje “Tu cuenta no pudo crearse con éxito” y manda al usuario a la interfaz de registrar cuenta, a continuación, este caso de uso queda sin efecto. |

|  |  |  |
| --- | --- | --- |
| **CU-0002** | **Iniciar sesión** | |
| **Versión** | 1.0 (15/05/24) | |
| **Autores** | Martínez de la Cruz José Julián | |
| **Fuentes** |  | |
| **Dependencias** | Ninguno | |
| **Descripción** | El sistema deberá comportarse tal cual se describe en el siguiente caso de uso cuando el usuario inicia sesión en la aplicación multiplataforma | |
| **Precondición** | El usuario debe contar con una cuenta previamente registrada | |
| **Secuencia normal** | **Paso** | **Acción** |
| 1 | El usuario da clic en el botón “Iniciar sesión” |
| 2 | El sistema muestra el formulario de inicio de sesión |
| 3 | El usuario ingresa su correo electrónico |
| 4 | El usuario ingresa su contraseña |
| 5 | El sistema valida los datos |
| 6 | El usuario da clic en el botón “Acceder” |
| **Postcondición** | El sistema manda al usuario a la interfaz principal | |
| **Excepciones** | **Paso** | **Acción** |
| 4 | Si el usuario ingresa datos no válidos, el sistema cancela el inicio de sesión, y muestra el mensaje “Correo electrónico o contraseña incorrectos” , a continuación, este caso de uso queda sin efecto. |
| 6 | Si el sistema no encuentra una cuenta registrada, el sistema cancela su ingreso, y muestra el mensaje “Parece que la cuenta ingresada no existe. Vuelve a intentarlo” , a continuación, este caso de uso queda sin efecto. |

|  |  |  |
| --- | --- | --- |
| **CU-0003** | **Rentar Película** | |
| **Versión** | **1.1 (30/05/2024)** | |
| **Autores** | **González Cruz Joel** | |
| **Actores** | **Cliente** | |
| **Dependencias** | **Ninguna** | |
| **Descripción** | El sistema le permitirá al usuario rentar una o más películas de su elección durante 30 días para que el cliente pueda poseerlas durante ese tiempo, posterior al plazo, el acceso a la película se revocará. | |
| **Precondición** | El cliente deberá haber iniciado sesión | |
| **Flujo normal** | **Paso** | **Acción** |
| 1 | El usuario selecciona la película que desee rentar. |
| 2 | El sistema muestra la información de la película además del tráiler y costo de la renta. |
| 3 | El usuario da click en el botón "Rentar". |
| 4 | El sistema mostrará un formulario de método de pago para rentar la película. |
| 5 | El usuario llena la información con sus datos bancarios. |
| 6 | El sistema valida la información ingresada. |
| 7 | El sistema muestra un resumen de la película a rentar, bajo que cuenta se va a rentar y con que método de pago se hará la compra. |
| 8 | El usuario verificará la información y dará click en el botón "Rentar por 30 días" |
| 9 | El sistema realizará el cobro bajo el concepto "BUTTERPOP – renta – {nombre de la película}, 30 días" |
| 10 | El sistema mostrará un mensaje: "Transacción realizada exitosamente, revisa tu lista privada ‘mis películas rentadas’ o haz click aquí para ver la película que acabas de rentar". |
| **Postcondición** | El link de la película se agregará a una lista privada de "mis películas rentadas" al cliente | |
| **Flujo alternativo** | **Paso** | **Acción** |
| 5.1 | En caso de que algún campo este vacío el sistema mostrará que todos los campos son requeridos |
| 6.1 | En caso de los datos ingresados no se puedan validar mostrára un mesaje indicando que ocurrío un problema al verificar los datos |
| 9.1 | En caso de que el método de pago no posea con los fondos suficientes, el sistema cancelará la transacción y notificara al usuario de que sus fondos son insuficientes |
| 9.2 | En caso de que el sistema no pueda completar la transacción, mostrará al cliente un mensaje de error: "No se ha podido realizar la operación, comprueba el estado de tu método de pago" |

|  |  |  |
| --- | --- | --- |
| **CU-0004** | **Comentar película** | |
| **Versión** | 1.0 (15/05/24) | |
| **Autores** | Rosales Murillo Jesús Miguel | |
| **Fuentes** |  | |
| **Dependencias** | Ninguno | |
| **Descripción** | El sistema deberá permitir a los usuarios comentar las películas que hayan visto en su respectiva sección y poder visualizar los comentarios de cada película. | |
| **Precondición** | El usuario deberá haber visto la película en cuestión previamente y encontrarse en el apartado de películas. | |
| **Secuencia normal** | **Paso** | **Acción** |
| 1 | El usuario deberá darle clic en el botón comentar. |
| 2 | El sistema muestra la interfaz del apartado comentar. |
| 3 | El usuario rellenará con sus palabras la caja de comentario. |
| 4 | El usuario deberá darle clic en guardar. |
| 5 | El sistema le mostrará el mensaje de comentario insertado. |
| **Postcondición** | Los comentarios ingresados se almacenan en la base de datos para que cualquiera pueda visualizarlos. | |
| **Excepciones** | **Paso** | **Acción** |
| 7 | Si el usuario no ha visto la película, no podrá comentar respecto a ella, apareciéndose un mensaje de error “No es posible comentar si no has visto la película”. |

|  |  |  |
| --- | --- | --- |
| **CU-0004** | **Puntuar película** | |
| **Versión** | 1.0 (15/05/24) | |
| **Autores** | Rosales Murillo Jesús Miguel | |
| **Fuentes** |  | |
| **Dependencias** | Ninguno | |
| **Descripción** | El sistema deberá permitir a los usuarios puntuar las películas que hayan visto en su respectiva sección y poder visualizar las puntuaciones de cada película. | |
| **Precondición** | El usuario deberá haber visto la película en cuestión previamente y encontrarse en el apartado de películas. | |
| **Secuencia normal** | **Paso** | **Acción** |
| 1 | El usuario deberá darle clic en el botón puntuar. |
| 2 | El sistema muestra la interfaz del apartado puntuar. |
| 3 | El usuario puntuará de acuerdo a su preferencia (De 0 a 5 puntos). |
| 4 | El usuario deberá darle clic en guardar. |
| 5 | El sistema le mostrará el mensaje de puntuación guardada. |
|  | 6 | El sistema regresará al usuario a la interfaz de películas con su puntuación guardada. |
| **Postcondición** | Las puntuaciones guardadas se almacenan en la base de datos para que cualquiera pueda visualizarlas. | |
| **Excepciones** | **Paso** | **Acción** |
| 7 | Si el usuario no ha visto la película, no podrá dar su puntuación respecto a ella, apareciéndose un mensaje de error “No es posible puntuar si no has visto la película”. |

|  |  |  |
| --- | --- | --- |
| **CU-0005** | **Gestionar lista** | |
| **Versión** | 1.0 (15/05/24) | |
| **Autores** | Domínguez Zenteno Andrea | |
| **Fuentes** |  | |
| **Dependencias** | Ninguno | |
| **Descripción** | El sistema deberá comportarse tal cual se describe en el siguiente caso de uso cuando el usuario pueda ver, crear, eliminar una lista de reproducción. | |
| **Precondición** | El usuario deberá previamente iniciar sesión o estar en la interfaz de películas. | |
| **Secuencia normal** | **Paso** | **Acción** |
| 1 | El usuario da click en “Mis listas”. |
| 2 | El sistema muestra la interfaz de “Mis listas”. |
| 3 | El usuario da click en el icono “+”. |
| 4 | El sistema despliega una pestaña para el nombre. |
| 5 | El usuario ingresa el nombre de su lista. |
| 6 | El usuario da click en “Crear”. |
| 7 | El sistema sube el nombre de la lista. |
| 8 | El sistema muestra la interfaz de “Mis listas”. |
| 9 | El usuario da click en el icono de “Editar”. |
| 10 | El sistema despliega una pestaña para actualizar el nombre. |
| 11 | El usuario ingresa el nuevo nombre de su lista. |
| 12 | El usuario da click en “Renombrar”. |
| 13 | El sistema actualiza el nombre de la lista. |
| 14 | El sistema muestra la interfaz de “Mis listas”. |
| 15 | El usuario da click en el icono de “Basura”. |
| 16 | El sistema muestra un mensaje de confirmación: “¿Estás seguro de eliminar esta lista?” |
| 17 | El usuario da click en “Confirmar”. |
| **Postcondición** | Se guardan los datos de la nueva lista en la base de datos y el sistema manda al usuario la interfaz “Mis listas”. | |
| **Excepciones** | **Paso** | **Acción** |
| 6 | Si ya existe una lista con el mismo nombre, el sistema debe mandar un aviso: “Ya tienes una lista con ese nombre, prueba con otro”. |
| 12 | Si el usuario da click en “Cancelar”, no hay cambios en los datos de la lista. |
| 17 | Si el usuario da click en “Cancelar”, no se elimina la lista. |

|  |  |  |
| --- | --- | --- |
| **CU-0006** | **Gestionar películas** | |
| **Versión** | 1.0 (15/05/24) | |
| **Autores** | Garcia Capote Emanuel | |
| **Fuentes** |  | |
| **Dependencias** | Ninguno | |
| **Descripción** | El sistema permitirá a los administradores agregar, eliminar o actualizar una película. | |
| **Precondición** | El administrador deberá de iniciar sesión previamente. | |
| **Secuencia normal** | **Paso** | **Acción** |
| 1 | Administrador da click en “Películas”. |
| 2 | El sistema muestra interfaz de películas. |
| 3 | El administrador da click en “+”. |
| 4 | El sistema muestra un menú para ingresar datos de la película. |
| 5 | El administrador ingresa “Nombre de la película”, “Descripción” y una “Imagen”. |
| 6 | El administrador da click en “Agregar”. |
| 7 | El sistema procesa los datos. |
| 8 | El sistema muestra la interfaz de “Películas”. |
| 9 | El administrador da click en el icono de “Editar”. |
| 10 | El sistema muestra un menú para ingresar datos de la película. |
| 11 | El administrador ingresa “Nombre de la película”, “Descripción” o una “Imagen”. |
| 12 | El administrador da click en “Confirmar”. |
| 13 | El sistema actualiza la base de datos. |
| 14 | El sistema muestra la interfaz de “Películas” con la película actualizada. |
| 15 | El administrador da click en el icono de “Borrar”. |
| 16 | El sistema muestra un mensaje de confirmación: “¿Estás seguro de eliminar la película?” con la película actualizada. |
| 17 | El administrador da click en “Confirmar”. |
| 18 | El sistema muestra la interfaz de “Películas” con la película eliminada. |
| **Postcondición** | Se guardan los datos en la base de datos y el sistema muestra los cambios hechos a clientes y administradores. | |
| **Excepciones** | **Paso** | **Acción** |
| 6 | Si algún dato proporcionado por el administrador no corresponde al establecido, se cancela el proceso. |
| 6 | Si el administrador da click en “Cancelar” la información se descarta. |
| 12 | Si el administrador da click en “Cancelar” la información de la película queda igual. |
| 17 | Si el administrador da click en “Cancelar” la película se queda en la base de datos. |

|  |  |  |
| --- | --- | --- |
| **CU-0007** | **Gestionar películas** | |
| **Versión** | 1.0 (15/05/24) | |
| **Autores** | Garcia Capote Emanuel | |
| **Fuentes** |  | |
| **Dependencias** | Ninguno | |
| **Descripción** | El sistema permitirá a los usuarios filtrar y ver rentas de distintas películas | |
| **Precondición** | El usuario deberá visualizar la pantalla principal, la aplicación debe estar activa y la película tendrá que estar disponible. | |
| **Secuencia normal** | **Paso** | **Acción** |
| 1 | El usuario dará clic en el botón filtrar |
| 2 | El sistema muestra el formulario de los distintos filtros |
| 3 | El usuario elegirá el tipo de filtro |
| 4 | El usuario dará clic en aplicar |
| 5 | El sistema muestra mensaje de confirmación |
| 6 | El usuario confirma |
| 7 | El sistema muestra películas |
| 8 | El usuario da clic en la película agradado |
| **Postcondición** | Se guardan los datos del filtro para próxima gestión de películas. | |
| **Excepciones** | **Paso** | **Acción** |
| 6 | Si existe una indecisión del tipo de filtros puede cancelar y se quedara como esta la cuenta |

**Diseño de pruebas basado en Casos de Uso**

Las pruebas basadas en casos de uso son una de las técnicas de diseño de pruebas de software que se centran en los casos de uso definidos para el sistema. Esta técnica es ampliamente utilizada porque permite realizar las pruebas basándose en cómo los usuarios reales interactúan con el sistema, asegurando que se prueben las funcionalidades más importantes desde la perspectiva del usuario. Además, al diseñar las pruebas basándose en los casos de uso, se garantiza que todos los requisitos funcionales se prueben adecuadamente. El motivo más importante es que facilita la identificación de defectos que pueden no ser evidentes con pruebas unitarias.

La implementación de esta técnica debe realizarse mediante un proceso estructurado que incluye las siguientes etapas:

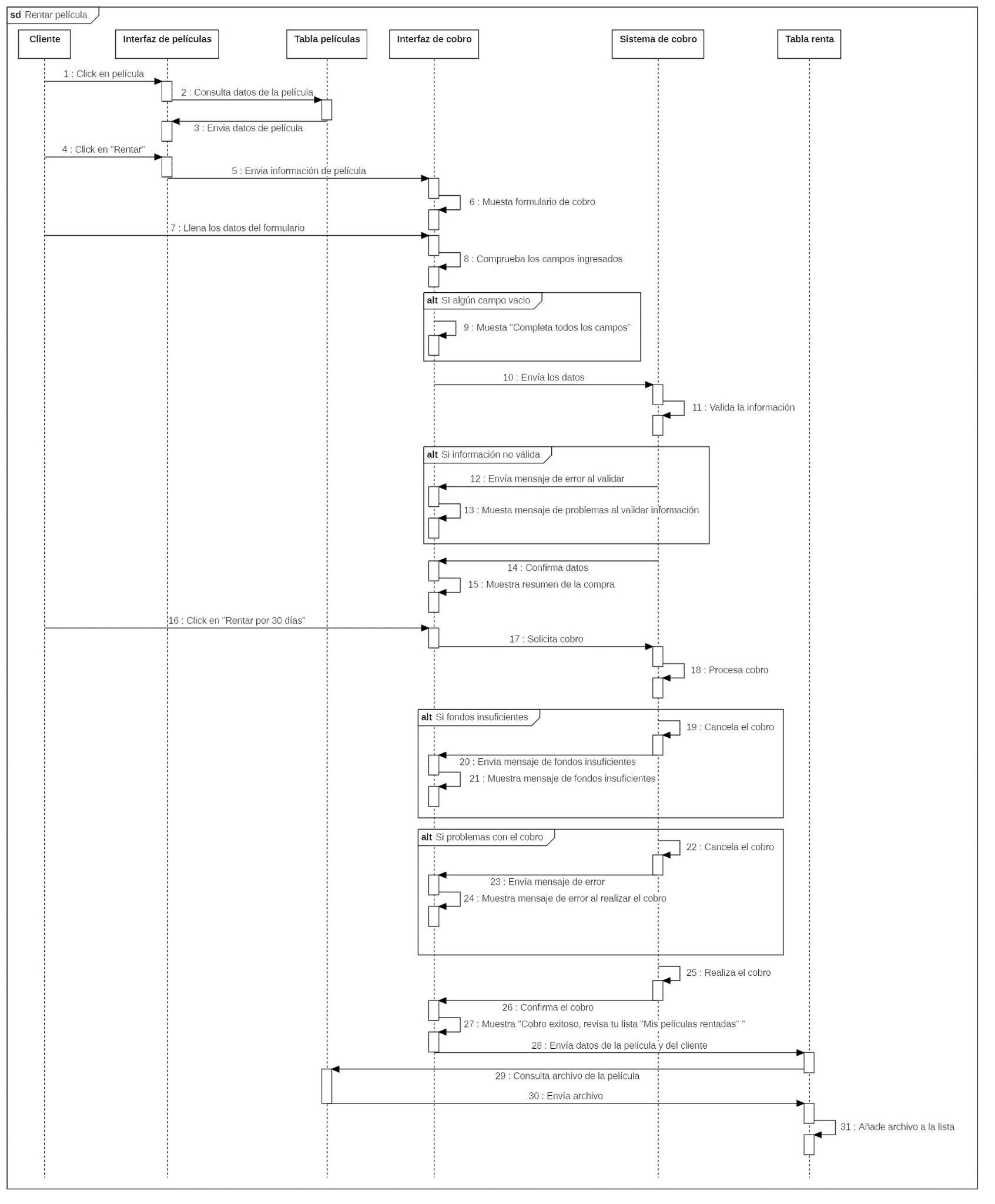
1. Identificación de Casos de Uso: Durante la fase de análisis de requisitos, se identifican y documentan los casos de uso del sistema. Cada caso de uso describe una secuencia de acciones que un usuario realiza para lograr un objetivo particular.
2. Descomposición de Casos de Uso: Cada caso de uso se descompone en sus pasos individuales o escenarios, incluyendo los pasos principales y las variaciones o excepciones. Esto ayuda a identificar todos los posibles caminos que un usuario puede seguir.
3. Diseño de Casos de Prueba: Para cada paso o escenario dentro de un caso de uso, se diseñan casos de prueba específicos. Estos casos de prueba detallan las entradas, las acciones a realizar y los resultados esperados. Es importante cubrir tanto los flujos normales como las excepciones.
4. Ejecución de Casos de Prueba: Los casos de prueba diseñados se ejecutan en el entorno de prueba. Durante la ejecución, se verifica que el sistema se comporte conforme a lo esperado en cada paso del caso de uso.
5. Documentación de Resultados: Los resultados de las pruebas se documentan, incluyendo cualquier defecto encontrado. Esta información se utiliza para realizar ajustes y correcciones en el sistema.
6. Revisión y Retroalimentación: Los resultados y los defectos encontrados se revisan con el equipo de desarrollo y otros stakeholders. Esta retroalimentación es crucial para mejorar la calidad del sistema y asegurar que se cumplan los requisitos del usuario.

1.9 Diagramas de secuencia

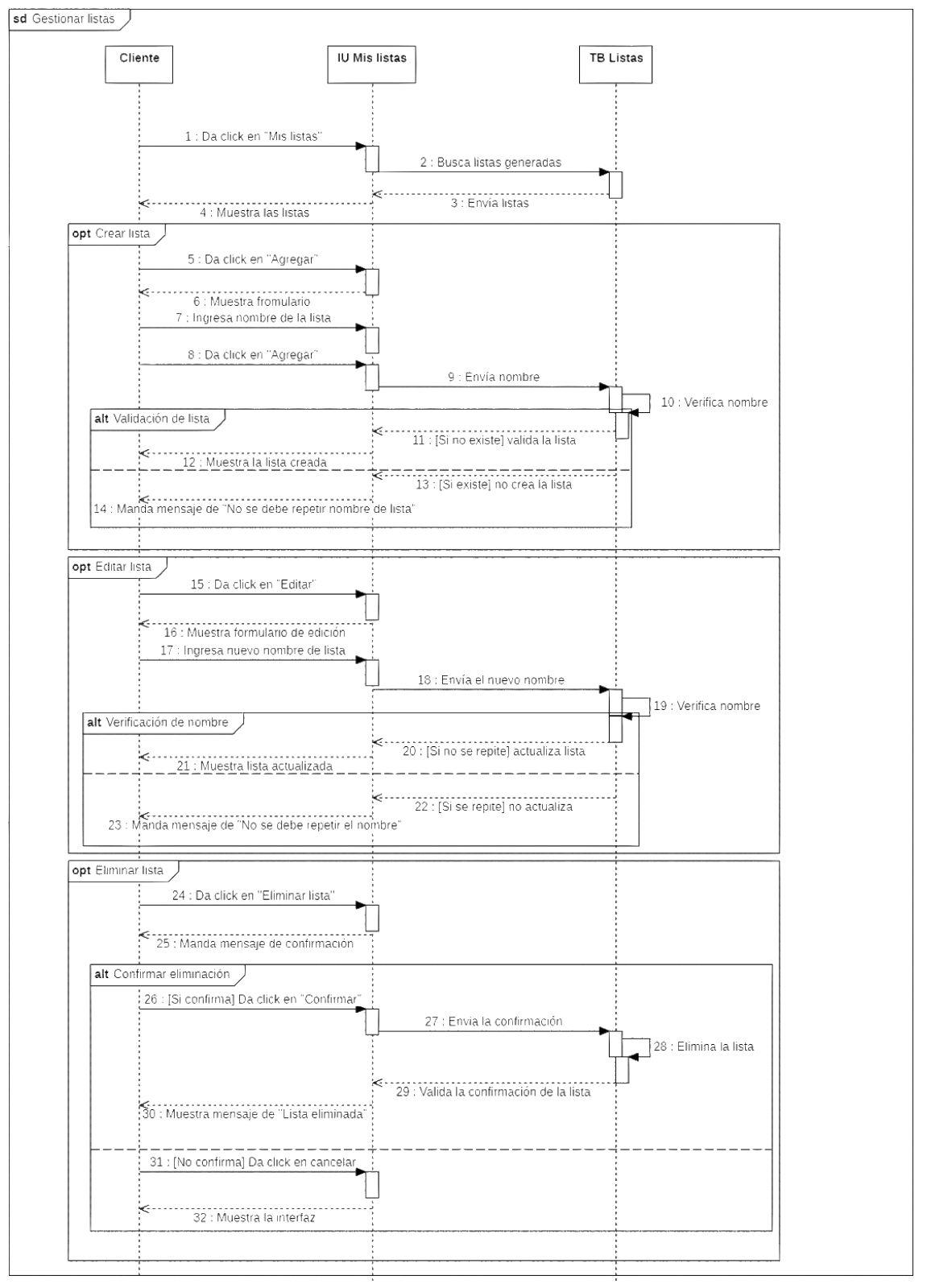
Inicio de sesión

Registrar usuarios

Rentar película



Gestionar lista



Rentar película

Puntuar película

Comentar película

1.10 Diagrama de actividades