**Un dibujo animado con letras

Descripción generada automáticamente con confianza media**

**BUTTERPOP**

**INTEGRADORA**

**R E P O R T E T É C N I C O**

PARA OBTENER EL TÍTULO DE

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

*P R E S E N T A*

**ANDREA DOMÍNGUEZ ZENTENO**

**JESÚS MIGUEL ROSALES MURILLO**

**JOEL GONZÁLEZ CRUZ**

**JOSÉ JULIÁN MARTÍNEZ DE LA CRUZ**

ASESORA DE LA ORGANIZACIÓN: DRA. MORAMAY RAMÍREZ HERNÁNDEZ

ASESORA ACADÉMICA: LDA. SANDRA RAQUEL LÓPEZ ARCE SOROA

ORGANIZACIÓN: “DIGITAL DREAM S.A. DE C.V.”  
GENERACIÓN: ENERO 2023-DICIEMBRE 2024

CUATRIMESTRE DE TÉRMINO: SEPTIEMBRE-DICIEMBRE 2024

**ÍNDICE**

[**RESUMEN** 1](#_Toc173233069)

[**ABSTRACT** 2](#_Toc173233070)

[**INTRODUCCIÓN** 3](#_Toc173233071)

[**OBJETIVOS** 5](#_Toc173233072)

[**PROGRAMA Y CRONOGRAMA** 6](#_Toc173233073)

[**MARCO TEÓRICO** 10](#_Toc173233074)

[**METODOLOGÍA** 15](#_Toc173233075)

[**CAPÍTULO 1 RECOLECCIÓN Y REFINAMIENTO DE REQUISITOS** 16](#_Toc173233076)

[1.1 Necesidades del cliente 16](#_Toc173233077)

[1.2 Identificación y definición de roles de usuario 16](#_Toc173233078)

[1.3 Requerimientos funcionales 17](#_Toc173233079)

[1.4 Requerimientos no funcionales 17](#_Toc173233080)

[1.5 Requerimientos de hardware 18](#_Toc173233081)

[1.6 Requerimientos de software 18](#_Toc173233082)

[1.7 Diagrama general de casos de uso 19](#_Toc173233083)

[1.8 Especificación de los casos de uso 20](#_Toc173233084)

[1.9 Diagramas de secuencia 28](#_Toc173233085)

[1.10 Diagrama de actividades 34](#_Toc173233086)

[**CAPÍTULO 2 DISEÑO Y DESARROLLO DEL PROYECTO** 35](#_Toc173233087)

[2.1 Diagrama de la base de datos (relacional) 35](#_Toc173233088)

[2.2 Diccionario de datos 35](#_Toc173233089)

[2.3 Diseño de pruebas 38](#_Toc173233090)

[2.4 Diseño de las interfaces (maquetas) 50](#_Toc173233092)

[**CAPÍTULO 3 IMPLEMENTACION Y PRUEBAS** 64](#_Toc173233093)

# **RESUMEN**

El presente documento tiene como finalidad mostrar el desarrollo del proyecto de renta de películas digitales “BUTTERPOP”, en este documento se desglosa sus partes, tales como: los objetivos del proyecto, alcance, herramientas empleadas para su desarrollo, metodología empleada, especificación de requisitos, diseño y desarrollo del proyecto, así como su implementación.

La aplicación BUTTERPOP está destinada a la renta de películas digitales, mientras el usuario tenga una renta activa podrá ver la película cuantas veces desee hasta que su renta expire, dicha renta podrá hacerse durante 1 hasta 4 semanas. También el usuario podrá crear listas de reproducción de películas, las cuales podrán estar rentadas por el usuario o no. Por último, los usuarios que hayan rentado alguna película podrán comentar y puntuar la película para que otros usuarios puedan ver los comentarios y las puntuaciones.

La finalidad de la aplicación es ofrecer una alternativa de renta de películas para que los consumidores tengan la elección de decidir al disfrutar de sus películas favoritas en el cine o en la comodidad de su hogar. Las películas disponibles en “BUTTERPOP” dependen de la cartelera vigente de Cinemas AJEM.

Aquí mismo se entra en detalle sobre que herramientas se emplean para su especificación de requisitos, desarrollo, implementación, pruebas, despliegue y mantenimiento futuro. Esto con la finalidad de dar claridad en los pasos del desarrollo, partiendo de una interfaz amigable para el usuario.

BUTTERPOP tiene interfaces amigables para facilitar la interacción del usuario con todas las funcionalidades de la aplicación, asegurando que el producto final tenga portabilidad, usabilidad, mantenibilidad, fiabilidad, eficiencia, seguridad y compatibilidad con sistemas operativos móviles tales como Android e iOS, asegurando la calidad de la aplicación “BUTTERPOP”.

# **ABSTRACT**

The purpose of this document is to show the development of the digital film rental project "BUTTERPOP", in this document all of its parts are well specified, such as: the objectives of the project, scope, tools used for its development, methodology used, specification of requirements, design and development of the project, as well as its implementation.

The BUTTERPOP application is intended for the rental of digital films, while the user has an active rent where they can watch movies as many times as they want until their rent expires, this rent can be made for 1 to 4 weeks. The user will also be able to create movie playlists, which can be rented by the user or not. Finally, users who have rented a movie will be able to comment and rate the movie so that other users can see all the comments and ratings from other users.

The purpose of the application is to offer a movie rental alternative so that consumers have the choice to decide when enjoying their favorite movies in the cinema or in the comfort of their home. The movies available in "BUTTERPOP" depend on the current Cinemas AJEM billboard.

Right here we go into detail about what tools are used for your requirements specification, development, implementation, testing, deployment and future maintenance. This in order to give clarity in the steps of development, starting from a user-friendly interface.

BUTTERPOP has user-friendly interfaces to facilitate user interaction with all the functionalities of the application, ensuring that the final product has portability, usability, maintainability, reliability, efficiency, security and compatibility with mobile operating systems such as Android and iOS, ensuring the quality of the "BUTTERPOP" application.

# **INTRODUCCIÓN**

Cinemas AJEM será 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 posicionará 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 encargará no solo del manejo y control de todas las ventas registradas dentro del cine, sino que también participará 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, existirá 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 innovadora, amigable, segura e intuitiva que permita a los clientes de Cinemas AJEM comprar y rentar las películas más recientes de su cartelera desde la comodidad de sus hogares. La aplicación proporcionará una experiencia de usuario única al ofrecer funcionalidades como la creación de listas de reproducción, acumulación de puntos para descuentos, y la opción de dejar opiniones sobre las películas. Con el objetivo de revertir la tendencia a la baja en ventas debido a la popularidad de las plataformas de streaming, esta aplicación se posicionará como una solución integral para incrementar significativamente las ventas del cine y atraer a los usuarios que no lograron ver las películas en el cine o desean volver a disfrutar de ellas en cualquier momento durante el período de renta.

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**

|  |
| --- |
| **Un dibujo animado con letras  Descripción generada automáticamente con confianza mediaUNIVERSIDAD 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  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  2523260021  2523260022  2523260068 |
| GENERACIÓN: | Enero 2023 – Diciembre 2024 |

*ASESORA ACADÉMICA*

|  |  |
| --- | --- |
| NOMBRE: | 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: | Desarrollo de Software Multiplataforma |
| DIRECCIÓN: | Carretera Federal México – Pachuca Km 37.5, 55749 Estado de México |
| TELÉFONO: | 55 6499 7632 |
| E-MAIL: | ditc@uttecamac.edu.mx |

*ASESORA DE LA ORGANIZACIÓN*

|  |  |
| --- | --- |
| NOMBRE: | Moramay Ramírez Hernández |
| CARGO: | Profesora de Tiempo Completo |

*PERÍODO*

|  |  |
| --- | --- |
| DURACIÓN: | 15 semanas |
| FECHA DE INICIO: | 01 de mayo de 2024 |
| FECHA DE TERMINACIÓN: | 15 de agosto de 2024 |
| HORARIO: | 10:00 am a 6:00 pm |

*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 innovadora, amigable, segura e intuitiva que permita a los clientes de Cinemas AJEM comprar y rentar las películas más recientes de su cartelera desde la comodidad de sus hogares. La aplicación proporcionará una experiencia de usuario única al ofrecer funcionalidades como la creación de listas de reproducción, acumulación de puntos para descuentos, y la opción de dejar opiniones sobre las películas. Con el objetivo de revertir la tendencia a la baja en ventas debido a la popularidad de las plataformas de streaming, esta aplicación se posicionará como una solución integral para incrementar significativamente las ventas del cine y atraer a los usuarios que no lograron ver las películas en el cine o desean volver a disfrutar de ellas en cualquier momento durante el período de renta. |
| 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 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 |
| 1 |  | Recolección de Refinamiento y Requisitos |  |  |  |  |
|  | 1.1 | Requerimientos del Cliente | 1 | 1 | 07/05/2024 | 10/05/2024 |
|  | 1.2 | Especificar Alcance del Proyecto | 1 | 1 | 07/05/2024 | 13/05/2024 |
|  | 1.3 | Preparación de Hardware y Software | 1 | 1 | 07/05/2024 | 14/05/2024 |
|  | 1.4 | Diagrama de Casos de Uso | 2 | 2 | 15/05/2024 | 18/05/2024 |
|  | 1.5 | Especificación de Casos de Uso | 2 | 3 | 17/05/2024 | 24/05/2024 |
|  | 1.6 | Modelado de Base de Datos | 3 | 4 | 27/05/2024 | 05/06/2024 |
|  | 1.7 | Programación de Aplicación | 5 | 8 | 10/06/2024 | 21/06/2024 |
|  | 1.8 | Diseño de Interfaces | 8 | 10 | 22/06/2024 | 06/07/2024 |
|  | 1.9 | Entrega de Avance de Proyecto | 13 | 13 | 07/07/2024 | 10/07/2024 |
|  | 1.10 | Implementación y Pruebas | 13 | 14 |  |  |
|  | 1.11 | 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 |
| 1 | Exploración | PROG. |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| REAL |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 2 | Iniciación | PROG. |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| REAL |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 3 | Producto | PROG. |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| REAL |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 4 | Estabilización | PROG. |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| REAL |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 5 | Pruebas | PROG. |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| REAL |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

FIRMAS

|  |  |
| --- | --- |
| Dra. Moramay Ramírez Hernández  *ASESORA DE LA ORGANIZACIÓN* | |
| Jose Julian Martinez de la Cruz  *ESTUDIANTE* | Lda. Sandra Raquel López Arce Soroa  ASESORA ACADÉMICA |

# **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, Linux, 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:

* **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.

* **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 reutilizarlo 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.

* **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.

* **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.

Para el desarrollo de este proyecto se utilizarán diversos lenguajes de programación y herramientas de diseño y prueba que permitirán la implementación de la lógica del cliente y las funcionalidades esenciales de la aplicación multiplataforma, entre las cuales se incluyen:

* **Visual Studio**

Microsoft Visual Studio es un entorno de desarrollo integrado (IDE, por sus siglas en inglés) para Windows y macOS. Es compatible con múltiples lenguajes de programación, tales como C++, C#, Fortran, Visual Basic .NET, F#, Java, Python, Ruby y PHP, al igual que entornos de desarrollo web, como ASP.NET MVC, Django, etc.

Visual Studio se emplea para abarcar todo el ciclo de vida del desarrollo de software, generalmente de aplicaciones complejas escritas en C#, C++, F#, Python, entre otros. Este entorno de desarrollo ofrece herramientas de depuración avanzadas, diseño de interfaces, integración con bases de datos, herramientas de colaboración, herramientas de testing y de publicación.

Se hará uso de este entorno para desarrollar la aplicación, compilar y depurarla.

* **Xamarin**

Es un marco de interfaz de usuario de código abierto. Xamarin.froms permite a los desarrolladores compilar aplicaciones en Xamarin, Android y xamarin.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.

Xamarin se usa para desarrollar aplicaciones móviles multiplataforma con un solo código fuente con un rendimiento cercano al de una aplicación nativa, permite el desarrollo de interfaces de usuario, otorga acceso a funcionalidades del dispositivo móvil, ofrece herramientas de depuración y herramientas de testing.

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.

* **C#**

Es un lenguaje de programación multiparadigma desarrollado por Microsoft, que evoluciona de la familia de lenguajes C, toma lo mejor de los lenguajes C y C++, los cuales se asemejan mucho a lenguajes de alto nivel de abstracción como Java y JavaScript. Forma parte de la plataforma .NET de Microsoft.

Este lenguaje de programación se utiliza principalmente para el desarrollo aplicaciones de escritorio, desarrollo web, de aplicaciones móviles, desarrollo de videojuegos, servicios en la nube y de backend, desarrollo empresarial, y scripts de automatización en entornos Windows.

Se usará este lenguaje de programación para desarrollar un código general, el cual posteriormente será compilado por Xamarin para Android e iOS.

* **XAML**

El lenguaje de marcado de aplicaciones extensible (XAML) es un lenguaje declarativo. En concreto, XAML puede inicializar objetos y establecer propiedades de objetos mediante una estructura de lenguaje que muestra relaciones jerárquicas entre varios objetos y una convención de tipo de respaldo que admite la extensión de tipos. Puedes crear elementos visibles de la interfaz de usuario en el marcado XAML declarativo. A continuación, puedes asociar un archivo de código subyacente distinto para cada archivo XAML que puede responder a eventos y manipular los objetos que originalmente declares en XAML.

Este lenguaje de marcado se utiliza para definir interfaces de usuario en aplicaciones de escritorio y móviles, permite también la vinculación de datos, además de uso de recursos y estilos.

Se empleará este lenguaje de marcado para diseñar las interfaces de la aplicación.

* **SQLite**

Es un sistema de gestión de bases de datos relacional compatible con ACID, está escrita en C. A diferencia de los sistemas de gestión de bases de datos cliente-servidor, el motor de SQLite no es un proceso independiente con el que el programa principal se comunica. En lugar de eso, la biblioteca SQLite se enlaza con el programa pasando a ser parte integral del mismo. El programa utiliza la funcionalidad de SQLite a través de llamadas simples a subrutinas y funciones. Esto reduce la latencia en el acceso a la base de datos, debido a que las llamadas a funciones son más eficientes que la comunicación entre procesos.

SQLite sirve para almacenar información relacional de forma local, prototipado y desarrollo de pruebas de almacenamiento, almacenamiento en aplicaciones móviles, de escritorio y web.

Se empleará este gestor de base de datos para almacenar y procesar la información necesaria para la ejecución de la aplicación "BUTTERPOP".

* **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 el desarrollo de proyectos de código abierto.

Este sistema de control de versiones permite participar en un flujo colaborativo, manipular el historial de cambios, además de la creación y fusión de ramas.

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.

* **Visual Studio Code**

Es un editor de código fuente desarrollado por Microsoft para Windows, Linux, macOS y Web. Incluye soporte para la depuración, control integrado de Git, resaltado de sintaxis, finalización inteligente de código, fragmentos y refactorización de código. También es personalizable, por lo que los usuarios pueden cambiar el tema del editor, los atajos de teclado y las preferencias. Es gratuito y de código abierto,1​2​ aunque la descarga oficial está bajo software privativo e incluye características personalizadas por Microsoft.

Visual Studio Code posee de ciertas características como herramientas de depuración, extensiones que aumentan las capacidades de este editor de código, integración de un sistema de control de versiones como git, autocompletado inteligente y remarcado de sintaxis, terminales integradas, entre otros.

Se empleará este editor de código para la edición de archivos ligeros como archivos markdown o md y asistencia en resolución de conflictos de merge en git.

* **StarUML**

Es un sofisticado modelador de software destinado a soportar un modelado ágil y conciso.

Este software de modelado se usa para desarrollar diagramas UML (Unified Modeling Language), los cuales se emplean para diseñar software, aplicar ingeniería inversa, ingeniería directa, entre otros tipos de requerimientos de documentación en proyectos de software.

Se utilizará este programa para modelar diagramas de casos de uso, de secuencia, actividades, entre otros, para sustentar el desarrollo del proyecto.

* **Figma**

Figma es un editor de gráficos vectoriales y una herramienta de generación de prototipos, principalmente basada en la web, con características offline adicionales habilitadas por aplicaciones de escritorio en macOS y Windows. Permite a los diseñadores colaborar en tiempo real, lo cual es ideal para equipos distribuidos geográficamente. Esta herramienta es conocida por su capacidad de mantener un flujo de trabajo eficiente gracias a sus funciones de coedición y comentarios en vivo. Además, Figma soporta la creación de componentes reutilizables, lo que facilita la consistencia en el diseño de la interfaz de usuario.

Figma se utiliza para diseñar interfaces de usuario, colaborar en tiempo real, prototipado interactivo, diseño vectorial, usar bibliotecas de diseño.

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

* **Dia**

Dia es una aplicación informática de propósito general para la creación de diagramas, creada originalmente por Alexander Larsson,1​ y desarrollada como parte del proyecto GNOME. Está concebido de forma modular, con diferentes paquetes de formas para diferentes necesidades.

Se usará este programa para modelar el modelo relacional de la base de datos de "BUTTERPOP".

# **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. Nuestra 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**

El primer capítulo contiene la definición y especificación de los usuarios que usarán la aplicación, así como los requisitos funcionales y no funcionales del mismo, además del software y hardware necesario para desarrollar dicha aplicación. También se especifican los procesos generales de cada caso de uso con su respectivo diagrama de secuencia en cada caso.

## 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.

* 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.
* 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

Los requerimientos funcionales son aquellas descripciones del sistema que tienen una funcionalidad para satisfacer al usuario.

* 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

Los requerimientos no funcionales son aquellas descripciones que determinan como debe comportarse el sistema en cuestión de calidad y características.

* 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 1TB. 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 de nuestro proyecto dada la estructura que facilita a las bases de datos relacionales.
* Visual Studio: Para el desarrollo general de la aplicación, así como su compilación y depuración.
* Xamarin: Para la compilación de la aplicación en plataformas móviles como Android e iOS.
* 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

La figura 1.7.1 representa el diagrama general de casos de uso que muestran las gestiones principales de la aplicación.

Diagrama

Descripción generada automáticamente

**Figura 1.7.1** Diagrama general de casos de uso para la aplicación ButterPop.

## 1.8 Especificación de los casos de uso

La figura 1.8.1 muestra las secuencias del registro de un usuario.

|  |  |  |
| --- | --- | --- |
| **CU-0001** | **Registrar usuario** | |
| **Versión** | 1.0 (15/05/24) | |
| **Autores** | Martínez de la Cruz José Julián | |
| **Actores** | Cliente | |
| **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** |
| 7.1 | 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. |

**Figura 1.8.1** Registrar usuario

La figura 1.8.2 muestra las secuencias del inicio de sesión de un usuario.

|  |  |  |
| --- | --- | --- |
| **CU-0002** | **Iniciar sesión** | |
| **Versión** | 1.0 (15/05/24) | |
| **Autores** | Martínez de la Cruz José Julián | |
| **Actores** | Cliente, Administrador | |
| **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** |
| 4.1 | 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. |
| 6.1 | 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. |

**Figura 1.8.2** Iniciar sesión

La figura 1.8.3 muestra las secuencias y gestión de rentar una película.

|  |  |  |
| --- | --- | --- |
| **CU-0003** | **Rentar película** | |
| **Versión** | 2.0 (18/07/24) | |
| **Autores** | González Cruz Joel | |
| **Actores** | Cliente | |
| **Dependencias** | Ninguna | |
| **Descripción** | El sistema le permitirá al usuario rentar una o más películas de su elección durante 1 hasta 4 semanas 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. | |
| **Secuencia 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 usuario selecciona la cantidad de semanas de renta, estos pueden ser: 1, 2, 3, o 4 semanas |
| 7 | El sistema valida la información ingresada. |
| 8 | El sistema genera la información necesaria para completar la renta. Para calcular el precio de renta lo hará como la sumatoria del precio, sobre la semana cuadrada menos 1 |
| 9 | 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. |
| 10 | El usuario verificará la información y dará click en el botón "Rentar" |
| 11 | El sistema realizará el cobro bajo el concepto "BUTTERPOP - renta - {nombre de la película}, {cantidad de semanas} semanas" |
| 12 | El sistema creará un registro de información sobre la operación realizada y lo almacenará en una base de datos. |
| 13 | 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 | |
| **Excepciones** | **Paso** | **Acción** |
| 5.1 | En caso de que exista información bancaria del cliente, se rellenarán los datos del formulario automáticamente |
| 5.2 | En caso de que algún campo este vacío el sistema mostrará que todos los campos son requeridos |
| 7.1 | En caso de los datos ingresados no se puedan validar mostrará un mensaje indicando que ocurrió un problema al verificar los datos |
| 11.1 | 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 |
| 11.2 | 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" |

**Figura 1.8.3** Rentar película

La figura 1.8.4 muestra las secuencias y gestión de comentar una película.

|  |  |  |
| --- | --- | --- |
| **CU-0004** | **Comentar película** | |
| **Versión** | 1.1 (23/06/24) | |
| **Autores** | Rosales Murillo Jesús Miguel | |
| **Actores** | Cliente | |
| **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. |
| Modificar comentario | |
| 6 | El caso de uso comienza cuando el usuario clickea en la opción “Editar” en su comentario. |
| 7 | El sistema muestra la interfaz del apartado comentar con una caja para ingresar texto. |
| 8 | El usuario ingresará su nuevo comentario en la caja de comentario. |
| 9 | El usuario deberá darle clic en “Guardar”. |
| 10 | El sistema guarda el comentario modificado |
| 11 | El sistema muestra el comentario modificado. |
| Eliminar comentario | |
| 12 | El caso de uso comienza cuando el usuario clickea en la opción “Borrar” en la sección “Editar” de su comentario. |
| 13 | El sistema muestra la interfaz de eliminar comentario, donde se le preguntará “¿Está seguro de querer borrar su comentario?”. |
| 14 | El usuario deberá seleccionar la opción “BORRAR”. |
| 15 | El sistema eliminará el comentario. |
| 16 | El sistema actualizará los comentarios. |
| **Postcondición** | Los comentarios ingresados se almacenan en la base de datos para que cualquiera pueda visualizarlos. | |
| **Excepciones** | **Paso** | **Acción** |
| 6.1 | 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”. |
| 7.1 | Si el usuario selecciona “Cancelar”, no se realizarán cambios en su comentario. |
| 12.1 | Si el usuario selecciona “Cancelar”, no se borrará su comentario. |

**Figura 1.8.4** Comentar

La figura 1.8.5 muestra las secuencias y gestión de puntuar una película.

**Figura 1.8.5** Puntuar película

|  |  |  |
| --- | --- | --- |
| **CU-0005** | **Puntuar película** | |
| **Versión** | 1.1 (23/06/24) | |
| **Autores** | Rosales Murillo Jesús Miguel | |
| **Actores** | Cliente | |
| **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 con su preferencia (De 0 a 3 estrellas). |
| 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. |
| Modificar puntuación. | |
| 7 | El caso de uso comienza cuando el usuario clickea en la opción “Editar” en su comentario. |
| 8 | El sistema muestra la interfaz del apartado comentar con una caja para ingresar su nueva puntuación. |
| 9 | El usuario ingresará su nueva puntuación en la caja de comentarios. |
| 10 | El usuario deberá darle clic en “Guardar”. |
| 11 | El sistema guarda la puntuación modificada |
| 12 | El sistema muestra la puntuación modificada |
| **Postcondición** | Las puntuaciones guardadas se almacenan en la base de datos para que cualquiera pueda visualizarlas. | |
| **Excepciones** | **Paso** | **Acción** |
| 7.1 | 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”. |

La figura 1.8.6 muestra las secuencias y gestión una lista.

|  |  |  |
| --- | --- | --- |
| **CU-0006** | **Gestionar lista** | |
| **Versión** | 1.1 (15/05/24) | |
| **Autores** | Domínguez Zenteno Andrea | |
| **Actores** | Cliente | |
| **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 clic en “Mis listas”. |
| 2 | El sistema muestra la interfaz de “Mis listas”. |
| 3 | El usuario da clic 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 clic en “Crear”. |
| 7 | El sistema sube el nombre de la lista. |
| 8 | El sistema muestra la interfaz de “Mis listas”. |
| Actualizar lista | |
| 9 | El usuario da clic 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 clic en “Renombrar”. |
| 13 | El sistema actualiza el nombre de la lista. |
| 14 | El sistema muestra la interfaz de “Mis listas”. |
| Eliminar lista | |
| 15 | El usuario da clic en el icono de “Basura”. |
| 16 | El sistema muestra un mensaje de confirmación: “¿Estás seguro de eliminar esta lista?” |
| 17 | El usuario da clic 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** |
| 6.1 | 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”. |
| 12.1 | Si el usuario da clic en “Cancelar”, no hay cambios en los datos de la lista. |
| 17.1 | Si el usuario da clic en “Cancelar”, no se elimina la lista. |

La figura 1.8.7 muestra las secuencias y gestión de una película.

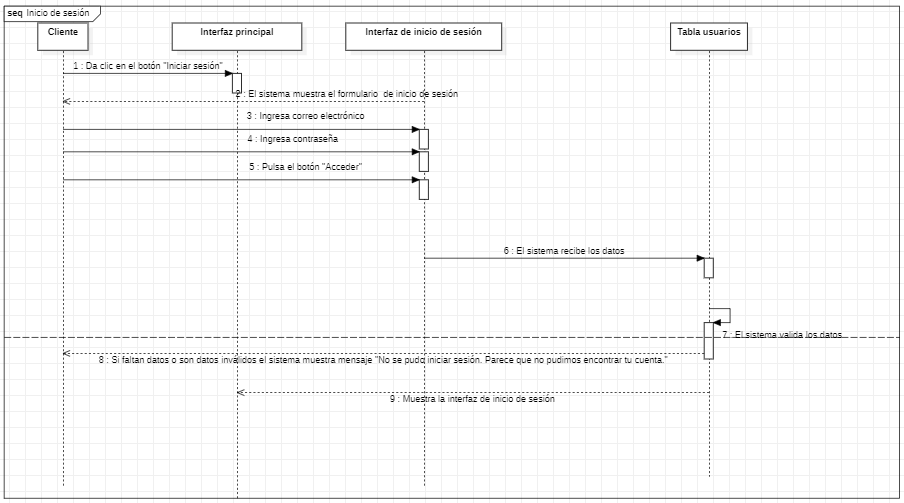
**Figura 1.8.6** Gestionar película

|  |  |  |
| --- | --- | --- |
| **CU-0007** | **Gestionar películas** | |
| **Versión** | 1.0 (15/05/24) | |
| **Autores** | Martínez de la Cruz José Julián | |
| **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 clic en “Películas”. |
| 2 | El sistema muestra interfaz de películas. |
| 3 | El administrador da clic 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 clic en “Agregar”. |
| 7 | El sistema procesa los datos. |
| 8 | El sistema muestra la interfaz de “Películas”. |
| 9 | El administrador da clic 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 clic 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 clic 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 clic 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** |
| 6.1 | Si algún dato proporcionado por el administrador no corresponde al establecido, se cancela el proceso. |
| 6.2 | Si el administrador da clic en “Cancelar” la información se descarta. |
| 12.1 | Si el administrador da clic en “Cancelar” la información de la película queda igual. |
| 17.1 | Si el administrador da clic en “Cancelar” la película se queda en la base de datos. |

**Figura 1.8.7** Gestionar película

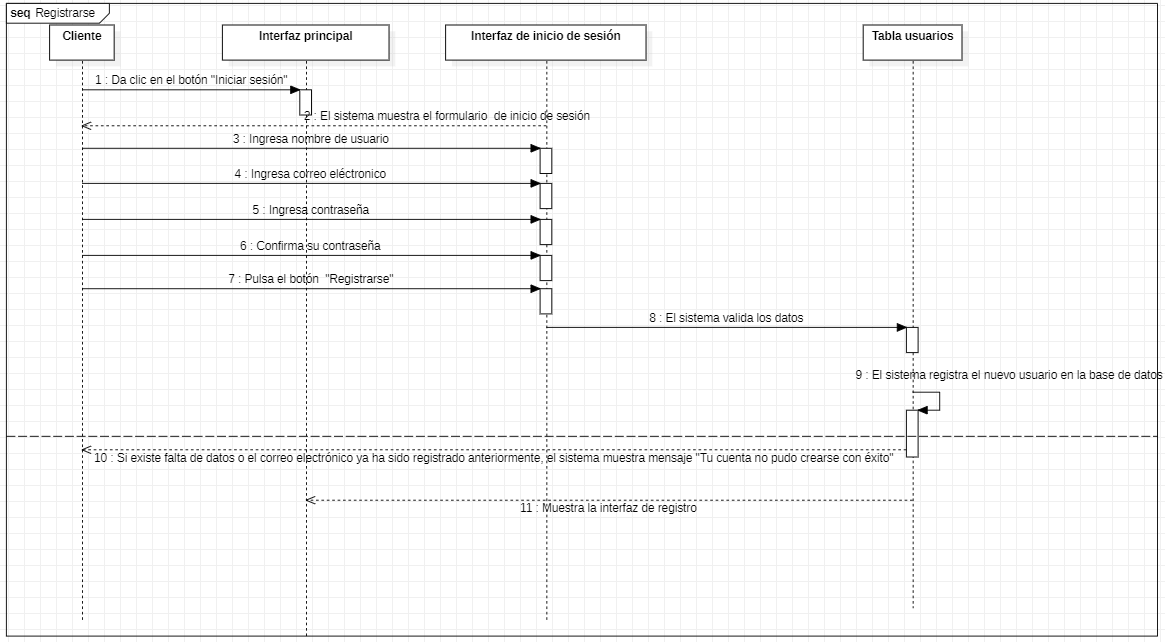
## 1.9 Diagramas de secuencia

La figura 1.9.1 muestra el diagrama de secuencia del caso de uso “Iniciar sesión”



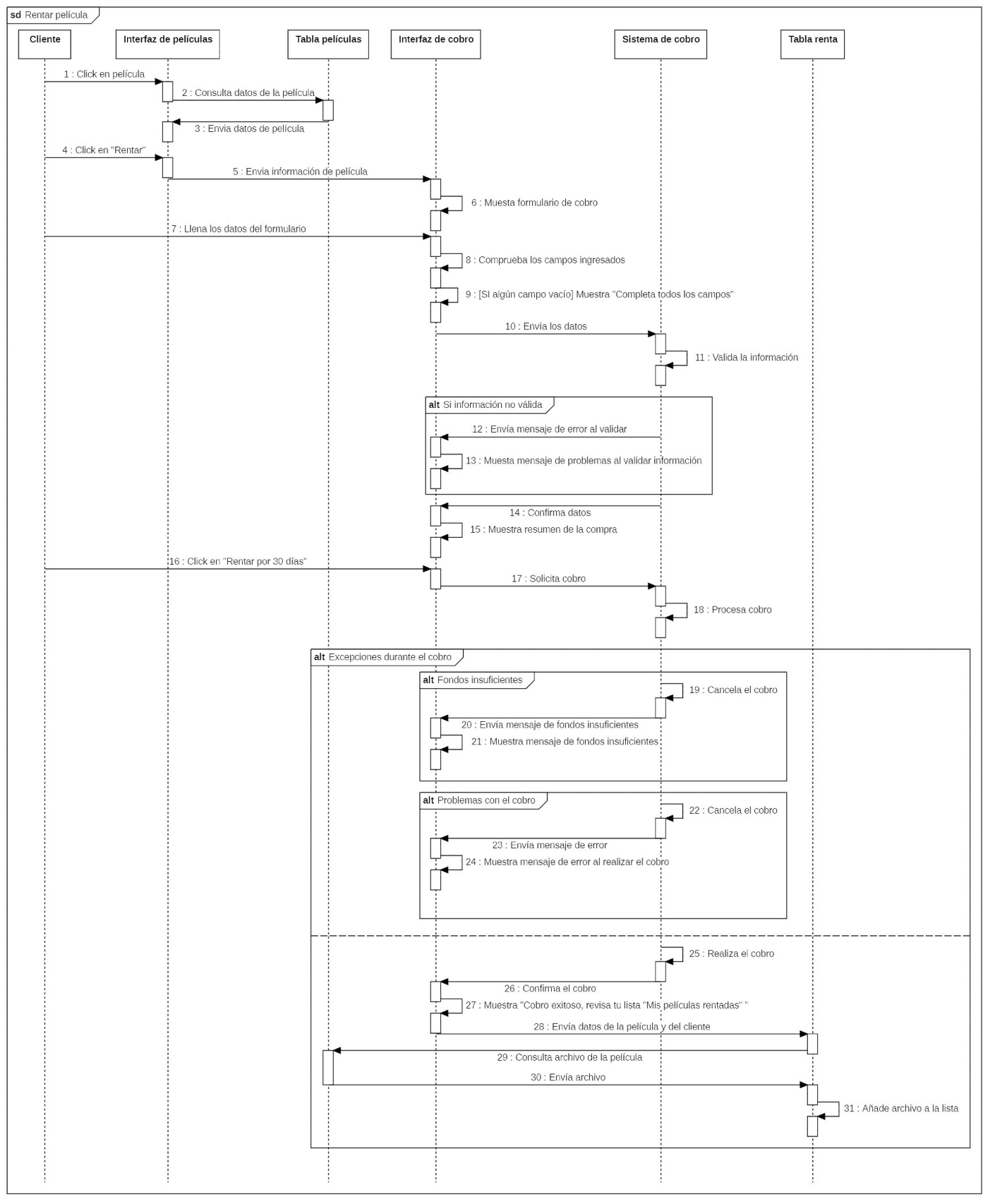
**Figura 1.9.1** Diagrama de secuencia para iniciar sesión

La figura 1.9.2 muestra el diagrama de secuencia del caso de uso “Registrar usuario”



**Figura 1.9.2** Diagrama de secuencia para registrarse

La figura 1.9.3 muestra el diagrama de secuencia del caso de uso “Rentar película”



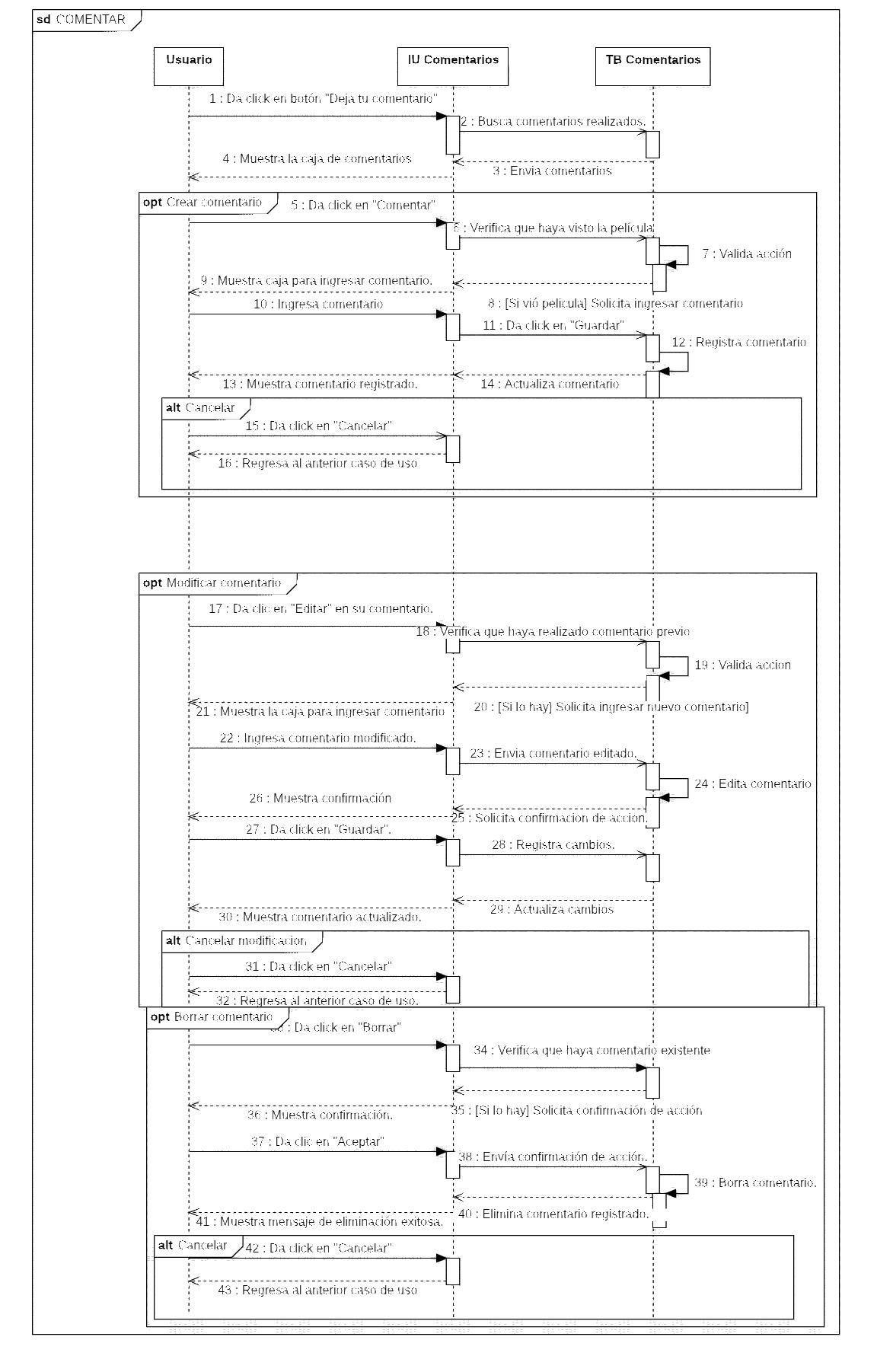
**Figura 1.9.3** Diagrama de secuencia para rentar película

La figura 1.9.4 muestra el diagrama de secuencia del caso de uso "Gestionar listas"

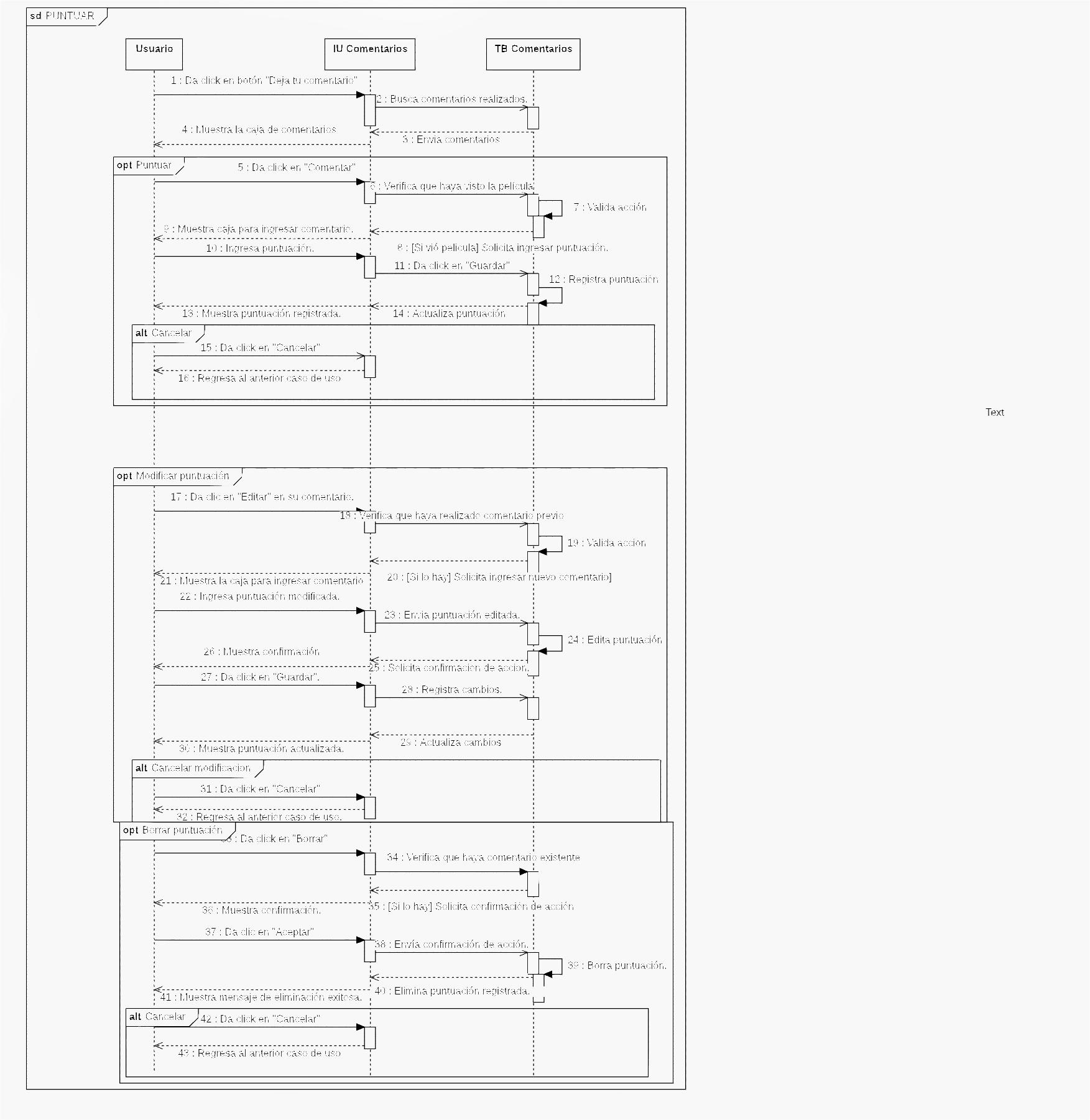
Tabla

Descripción generada automáticamente

**Figura 1.9.4** Diagrama de secuencia para gestionar listas

La figura 1.9.5 muestra el diagrama de secuencia del caso de uso “Gestionar comentarios”

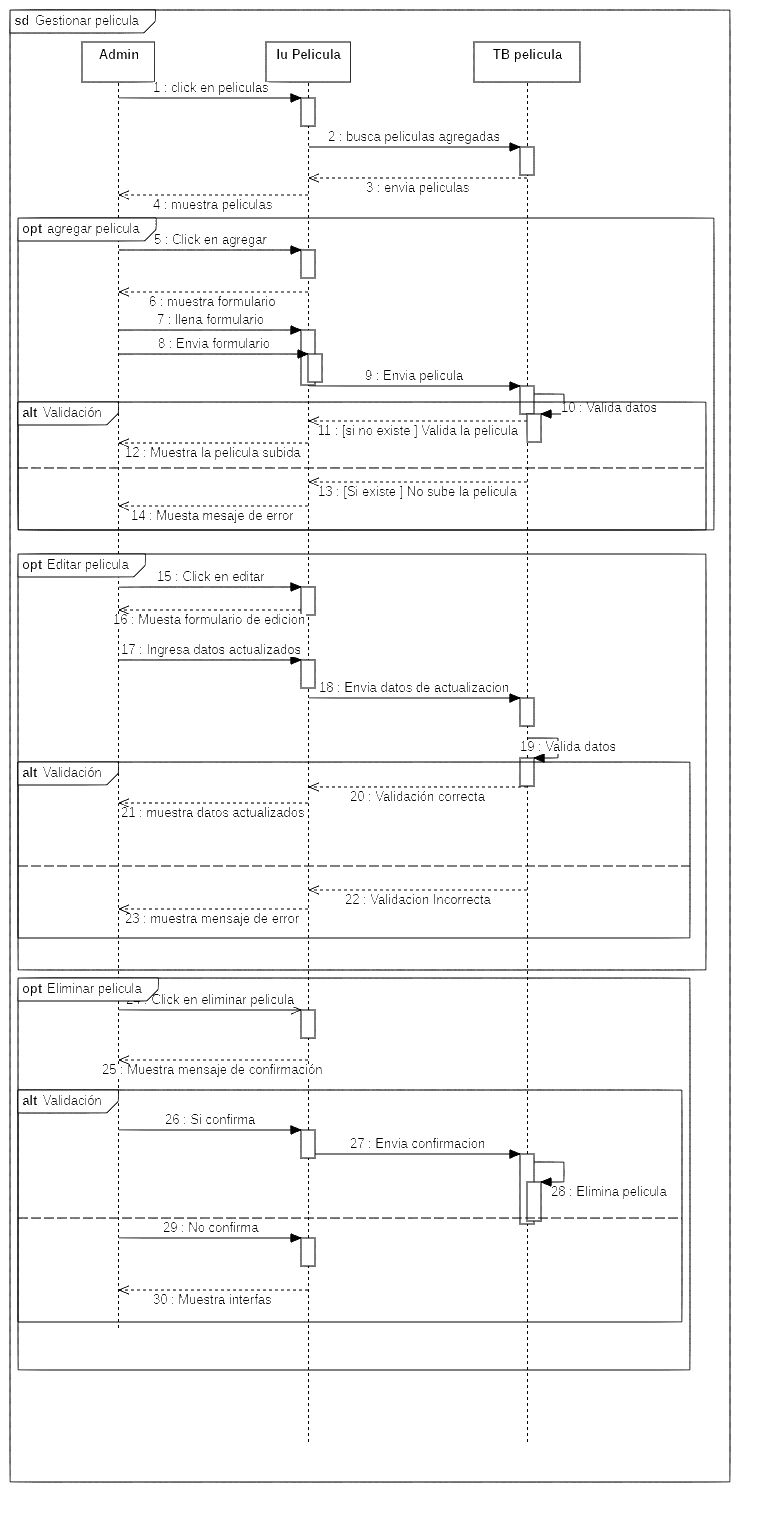
**Figura 1.9.5** Diagrama de secuencia para gestionar comentarios

La figura 1.9.6 muestra el diagrama de secuencia del caso de uso “Gestionar puntuación”

**Figura 1.9.6** Diagrama de secu a figura 1.9.6 muestra el diagrama de secuencia del caso de uso “Gestionar puntuación”

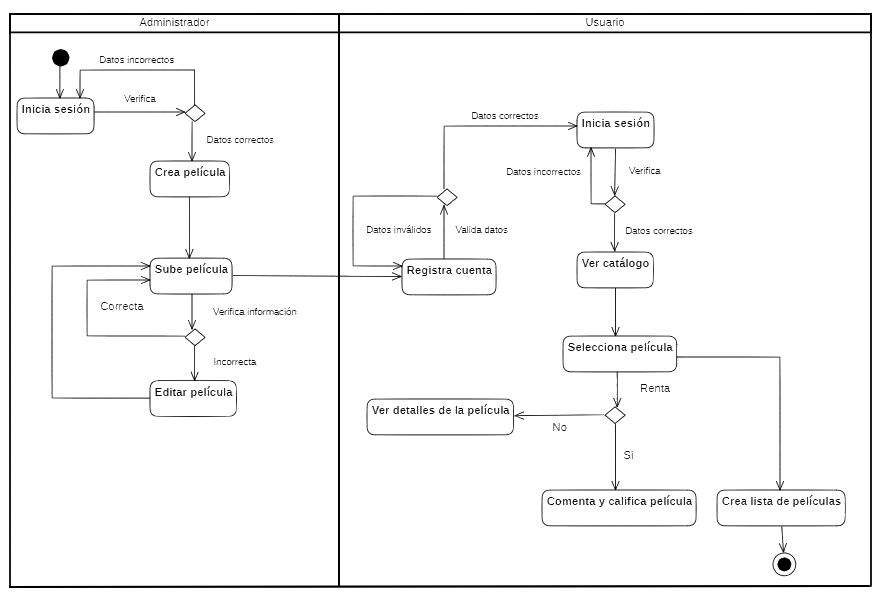
encia para gestionar puntuación

La figura 1.9.7 muestra el diagrama de secuencia del caso de uso “Gestionar película”



**Figura 1.9.7** Diagrama de secu a figura 1.9.7muestra el diagrama de secuencia del caso de uso “Gestionar pelicula”

## 1.10 Diagrama de actividades

La figura 1.10.1 muestra el diagrama de actividades de la aplicación

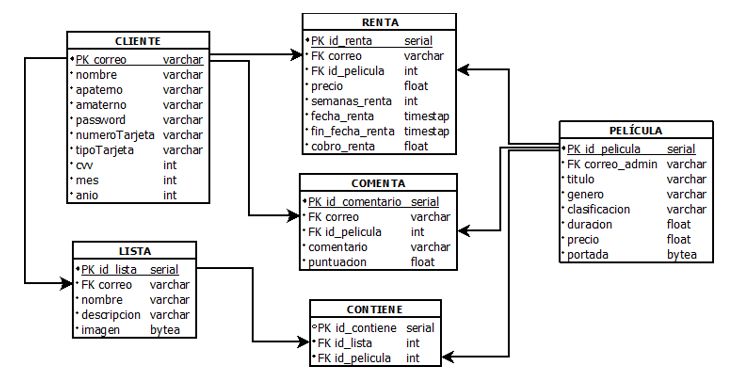
**Figura 1.10.1** Diagrama de actividades

# **CAPÍTULO 2 DISEÑO Y DESARROLLO DEL PROYECTO**

El segundo capítulo contiene la especificación de la base de datos, tales como sus tablas y su respectivo diccionario de datos para cada tabla, además se plantea un plan de pruebas con cada caso de prueba, por último, también se presenta el diseño de interfaces que contiene la aplicación.

## 2.1 Diagrama de la base de datos (relacional)

La figura 2.1.1 muestra el modelo relacional de la base de datos empleada para la aplicación "BUTTERPOP"



**Figura 2.1.1** Modelo de base de datos relacional

## 2.2 Diccionario de datos

El diccionario de datos son todas aquellas tablas donde especifican el nombre del campo, el tipo, la extensión, restricción y observaciones de cada tabla en la base de datos.

La figura 2.2.1 muestra el diccionario de datos de la tabla "pelicula"

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **TB PELÍCULA** | | | | |
| **Campo** | **Tipo** | **Extensión** | **Restricción** | **Observaciones** |
| id\_pelicula | serial | N/A | PK | Llave primaria de película. |
| titulo | varchar | 50 | N/A | Título de la película. |
| genero | varchar | 30 | N/A | Género de la película. |
| clasificacion | varchar | 30 | N/A | Clasificación de la película. |
| duracion | int | N/A | N/A | Duración de la película en minutos. |
| precio | float | N/A | N/A | Precio de la renta de la película. |
| portada | bytea | N/A | N/A | Portada de la película. |

**Figura 2.2.1** Diccionario de datos de la tabla “pelicula”

La figura 2.2.2 muestra el diccionario de datos de la tabla "renta"

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **TB RENTA** | | | | |
| **Campo** | **Tipo** | **Extensión** | **Restricción** | **Observaciones** |
| id\_renta | serial | N/A | PK | Llave primaria de renta. |
| correo | varchar | 50 | FK | Llave foránea del cliente que renta la película de la tabla cliente. |
| id\_pelicula | int | N/A | FK | Llave foránea de la película de la tabla película. |
| precio | float | N/A | N/A | Precio de la película |
| semanas\_renta | int | N/A | N/A | Semanas de renta de la película |
| fecha\_renta | timestamp | N/A | N/A | Fecha y hora en la que se registra la renta. |
| fin\_fecha\_renta | timestamp | N/A | N/A | Fecha y hora en la que se finaliza la renta. |
| cobro\_renta | float | N/A | N/A | Cobro de renta de película. |

**Figura 2.2.2** Diccionario de datos de la tabla “renta”

La figura 2.2.3 muestra el diccionario de datos de la tabla "comentar"

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **TB COMENTAR** | | | | |
| **Campo** | **Tipo** | **Extensión** | **Restricción** | **Observaciones** |
| id\_comentario | serial | N/A | PK | Llave primaria de comentar. |
| id\_pelicula | int | N/A | FK | Llave foránea de película. |
| correo | varchar | 50 | FK | Llave foránea del cliente de la tabla cliente. |
| comentario | varchar | 100 | N/A | Contenido del comentario. |
| puntuacion | float | N/A | N/A | Puntuación de la película (0.0 – 3.0). |

**Figura 2.2.3** Diccionario de datos de la tabla “comentar”

La figura 2.2.4 muestra el diccionario de datos de la tabla "contiene"

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **TB CONTIENE** | | | | |
| **Campo** | **Tipo** | **Extensión** | **Restricción** | **Observaciones** |
| id\_contiene | serial | N/A | PK | Llave primaria de contiene. |
| id\_lista | int | N/A | FK | Llave foránea de la lista de la tabla lista. |
| id\_pelicula | int | N/A | FK | Llave foránea de la película de la tabla película. |

**Figura 2.2.4** Diccionario de datos de la tabla “contiene”

La figura 2.2.5 muestra el diccionario de datos de la tabla "cliente"

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **TB CLIENTE** | | | | |
| **Campo** | **Tipo** | **Extensión** | **Restricción** | **Observaciones** |
| correo | varchar | 50 | PK | Llave primaria del cliente. |
| nombre | varchar | 50 | N/A | Nombre del cliente. |
| apaterno | varchar | 50 | N/A | Apellido paterno del cliente. |
| amaterno | varchar | 50 | N/A | Apellido materno del cliente. |
| password | varchar | 20 | N/A | Hash de la contraseña del cliente. |
| numeroTarjeta | varchar | 16 | N/A | Número de tarjeta del cliente |
| tipoTarjeta | varchar | 50 | N/A | Tipo de tarjeta del cliente. |
| cvv | int | N/A | N/A | Clave de seguridad de la tarjeta del cliente. |
| mes | int | N/A | N/A | Mes de vencimiento de la tarjeta del cliente. |
| anio | int | N/A | N/A | Año de vencimiento de la tarjeta del cliente. |

**Figura 2.2.5** Diccionario de datos de la tabla “cliente”

La figura 2.2.6 muestra el diccionario de datos de la tabla "lista"

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **TB LISTA** | | | | |
| **Campo** | **Tipo** | **Extensión** | **Restricción** | **Observaciones** |
| id\_lista | serial | N/A | PK | Llave primaria de lista. |
| correo\_cliente | varchar | 50 | FK | Llave foránea del cliente de tabla cliente. |
| nombre | varchar | 30 | N/A | Nombre de la lista. |
| descripcion | varchar | 100 | N/A | Descripción de la lista. |
| imagen | bytea | N/A | N/A | Imagen de la lista. |

**Figura 2.2.6** Diccionario de datos de la tabla “lista”

## 2.3 Diseño de pruebas

Diseño de pruebas por partición de equivalencia es una técnica utilizada en pruebas de software para dividir el dominio de entrada de un programa en clases de equivalencia. Cada clase representa un conjunto de valores de entrada que se espera que el software trate de manera similar. Las pruebas se realizan seleccionando al menos un valor de cada clase, con el objetivo de reducir el número de casos de prueba necesarios mientras se mantiene una cobertura efectiva del comportamiento del sistema.

**INICIAR SESIÓN / REGISTRARSE**

**TABLAS DE CONSIGNACIÓN**

Figura 2.3.1.1 muestra la tabla de consignación para el plan de pruebas “iniciar sesión” y “registrarse”

|  |  |  |  |
| --- | --- | --- | --- |
| **Condiciones de entrada** | **Regla heurística** | **Clases validas** | **Clases no validas** |
| **Credenciales de inicio de sesión** | Valor específico | **1.** Contraseña = 8 caracteres | **2.** Contraseña > 8 caracteres |
| **3.** Contraseña < 8 caracteres |
| Conjunto | **4.** Al menos un número | **5.** Sin números |
| **6.** Al menos una minúscula | **7.** Sin minúsculas |
| **8.** Al menos una mayúscula | **9.** Sin mayúsculas |
| **10.** Un símbolo | **11.** Sin símbolo |
| Booleano | **12.** Correo válido | **13.** Correo inválido |
| **14.** Existe en bd | **15.** No existe en bd |
| **16.** Contraseña correcta | **17.** Contraseña incorrecta |

**Figura 2.3.1.1** Tabla de consignación para “iniciar sesión” y “registrarse”

CLASES VÁLIDAS

Figura 2.3.1.2 muestra la tabla de clases válidas para el plan de pruebas “iniciar sesión” y “registrarse”

|  |  |  |
| --- | --- | --- |
| **No. Caso** | **Credenciales de inicio** | **Clases válidas** |
| 1 | **Correo:** [Jjulianmtz321@gmail.com](mailto:Jjulianmtz321@gmail.com)  **Contraseña:** juLian@1 | 1, 4, 6, 8, 10, 12, 14, 16 |

**Figura 2.3.1.2** Clases válidas para “iniciar sesión” y “registrarse”

CLASES NO VÁLIDAS

Figura 2.3.1.3 muestra la tabla de clases inválidas para el plan de pruebas “iniciar sesión” y “registrarse”

|  |  |  |
| --- | --- | --- |
| **No. Caso** | **Credenciales de inicio** | **Clases inválidas** |
| 2 | **Correo:** [Jjulianmtz321@gmail.com](mailto:Jjulianmtz321@gmail.com)  **Contraseña:** Juli@n789 | 2, 4, 6, 8, 10, 12, 14, 17 |
| 3 | **Correo:** [alanfugas@gmail.com](mailto:alanfugas@gmail.com)  **Contraseña:** Al@n4 | 3, 4, 6, 8, 10, 12, 14, 17 |
| 4 | **Correo:** [elvale@gmail.com](mailto:elvale@gmail.com)  **Contraseña:** Val@ntin | 1, 5, 6, 8, 10, 12, 14, 17 |
| 5 | **Correo:** [elvale@gmail.com](mailto:elvale@gmail.com)  **Contraseña:** VAL@NT1N | 1, 4, 7, 8, 10, 12, 14, 17 |
| 6 | **Correo:** [elvale@gmail.com](mailto:elvale@gmail.com)  **Contraseña:** val@nt1n | 1, 4, 6, 9, 10, 12, 14, 17 |
| 7 | **Correo:** [elvale@gmail.com](mailto:elvale@gmail.com)  **Contraseña:** Valent1n | 1, 4, 6, 8, 11, 12, 14, 17 |
| 8 | **Correo:** [elvale@gmail.com](mailto:elvale@gmail.com)  **Contraseña:** 12345678 | 1, 4, 7, 9, 11, 12, 14, 17 |
| 9 | **Correo:** [elvalegmail.com](mailto:elvale@gmail.com)  **Contraseña:** Juli@n7 | 1, 4, 6, 8, 10, 13, 15, 17 |
| 10 | **Correo:** [elvale1@gmail.com](mailto:elvale1@gmail.com)  **Contraseña:** Juli@n7 | 1, 4, 6, 8, 10, 15, 17 |
| 11 | **Correo:** [Jjulianmtz321@gmail.com](mailto:Jjulianmtz321@gmail.com)  **Contraseña:** Juli@n7 | 1, 4, 6, 8, 10, 14, 17 |

**Figura 2.3.1.3** Clases inválidas para “iniciar sesión” y “registrarse”

**CASOS DE PRUEBA**

Figura 2.3.1.4 muestra la tabla de casos de prueba para el plan de pruebas “iniciar sesión” y “registrarse”

|  |  |  |  |
| --- | --- | --- | --- |
| **No. Caso** | **Credenciales de inicio de sesión** | **Resultados esperados** | **Resultados reales** |
| 1 | **Correo:** [Jjulianmtz321@gmail.com](mailto:Jjulianmtz321@gmail.com)  **Contraseña:** juLian@1 | El sistema verifica las credenciales de inicio de sesión y manda al usuario a la interfaz principal |  |
| 2 | **Correo:** [Jjulianmtz321@gmail.com](mailto:Jjulianmtz321@gmail.com)  **Contraseña:** Juli@n789 | El sistema muestra mensaje “Contraseña no cumple con los requisitos” y regresa al usuario a la interfaz de inicio de sesión |  |
| 3 | **Correo:** [alanfugas@gmail.com](mailto:alanfugas@gmail.com)  **Contraseña:** Al@n4 | El sistema muestra mensaje “Contraseña no cumple con los requisitos” y regresa al usuario a la interfaz de inicio de sesión |  |
| 4 | **Correo:** [elvale@gmail.com](mailto:elvale@gmail.com)  **Contraseña:** Val@ntin | El sistema muestra mensaje “Contraseña no cumple con los requisitos” y regresa al usuario a la interfaz de inicio de sesión |  |
| 5 | **Correo:** [elvale@gmail.com](mailto:elvale@gmail.com)  **Contraseña:** VAL@NT1N | El sistema muestra mensaje “Contraseña no cumple con los requisitos” y regresa al usuario a la interfaz de inicio de sesión |  |
| 6 | **Correo:** [elvale@gmail.com](mailto:elvale@gmail.com)  **Contraseña:** val@nt1n | El sistema muestra mensaje “Contraseña no cumple con los requisitos” y regresa al usuario a la interfaz de inicio de sesión |  |
| 7 | **Correo:** [elvale@gmail.com](mailto:elvale@gmail.com)  **Contraseña:** Valent1n | El sistema muestra mensaje “Contraseña no cumple con los requisitos” y regresa al usuario a la interfaz de inicio de sesión |  |
| 8 | **Correo:** [elvale@gmail.com](mailto:elvale@gmail.com)  **Contraseña:** 12345678 | El sistema muestra mensaje “Contraseña no cumple con los requisitos” y regresa al usuario a la interfaz de inicio de sesión |  |
| 9 | **Correo:** [elvalegmail.com](mailto:elvale@gmail.com)  **Contraseña:** Juli@n7 | El sistema muestra mensaje “El correo ingresado no es un correo valido” y regresa al usuario a la interfaz de inicio de sesión |  |
| 10 | **Correo:** [elvale1@gmail.com](mailto:elvale1@gmail.com)  **Contraseña:** Juli@n7 | El sistema muestra mensaje “No se pudo localizar la cuenta ingresada” y regresa al usuario a la interfaz de inicio de sesión |  |
| 11 | **Correo:** [Jjulianmtz321@gmail.com](mailto:Jjulianmtz321@gmail.com)  **Contraseña:** Juli@n7 | El sistema muestra mensaje “Contraseña incorrecta” y regresa al usuario a la interfaz de inicio de sesión |  |

**Figura 2.3.1.4** Casos de prueba para “iniciar sesión” y “registrarse”

**RENTAR PELÍCULA**

**TABLAS DE CONSIGNACIÓN**

Figura 2.3.2.1 muestra la tabla de consignación para el plan de pruebas “rentar película”

|  |  |  |  |
| --- | --- | --- | --- |
| **Condiciones de entrada** | **Regla heurística** | **Clases validas** | **Clases no validas** |
| Número de la tarjeta | Valor específico | 1. Número de caracteres de la tarjeta = 16 | 1. Caracteres de la tarjeta > 16 |
| 1. Caracteres de la tarjeta < 16 |
| Fecha de expiración | Booleano | 1. Fecha de expiración vigente | 1. Fecha de expiración no vigente |
| Clave de seguridad | Booleano | 1. Clave de seguridad valida | 1. Clave de seguridad invalida |
| Fondos de la tarjeta | Booleano | 1. Fondos de la tarjeta suficientes | 1. Fondos de la tarjeta insuficientes |

**Figura 2.3.2.1** Tabla de consignación para “rentar película”

CLASES VÁLIDAS

Figura 2.3.2.2 muestra la tabla de clases válidas para el plan de pruebas “rentar película”

|  |  |  |
| --- | --- | --- |
| No. Caso | Valores ingresados | Clases validas |
| 1 | **Número de tarjeta**: 6754 6785 2345 9800  **Fecha de expiración**: 06/28  **Clave de seguridad**: 347  **Fondos de la tarjeta**: Suficientes | 1, 4, 6, 8 |

**Figura 2.3.2.2** Clases válidas para “rentar película”

CLASES NO VÁLIDAS

Figura 2.3.2.3 muestra la tabla de clases inválidas para el plan de pruebas “rentar película”

|  |  |  |
| --- | --- | --- |
| No. Caso | Valores ingresados | Clases no validas |
| 2 | **Número de tarjeta**: 5630 4585 4762 5987  **Fecha de expiración**: 07/25  **Clave de seguridad**: 478  **Fondos de la tarjeta**: Insuficientes | 1, 4, 6, 9 |
| 3 | **Número de tarjeta**: 4576 7589 3485 0054  **Fecha de expiración**: 06/30  **Clave de seguridad**: 04  **Fondos de la tarjeta**: Suficientes | 1, 4, 7, 8 |
| 4 | **Número de tarjeta**: 3287 2098 2683 5983  **Fecha de expiración**: 02/27  **Clave de seguridad**: 22  **Fondos de la tarjeta**: Insuficientes | 1, 4, 7, 9 |
| 5 | **Número de tarjeta**: 4327 8523 5932 3478  **Fecha de expiración**: 03/22  **Clave de seguridad**: 349  **Fondos de la tarjeta**: Suficientes | 1, 5, 6, 8 |
| 6 | **Número de tarjeta**: 2348 7475 6385 8043  **Fecha de expiración**: 12/12  **Clave de seguridad**:584  **Fondos de la tarjeta**: Insuficientes | 1, 5, 6, 9 |
| 7 | **Número de tarjeta**: 1295 9230 9572 9054  **Fecha de expiración**: 09/28  **Clave de seguridad**: 01  **Fondos de la tarjeta**: Suficientes | 1, 5, 7, 8 |
| 8 | **Número de tarjeta**: 7823 4782 3579 0121  **Fecha de expiración**: 05/24  **Clave de seguridad**: -457  **Fondos de la tarjeta**: Insuficientes | 1, 5, 7, 9 |
| 9 | **Número de tarjeta**: 1295 2467 3468 2476 3498 54  **Fecha de expiración**: 07/28  **Clave de seguridad**: 789  **Fondos de la tarjeta**: Suficientes | 2, 4, 6, 8 |
| 10 | **Número de tarjeta**: 4567 4577 4564 1209 12  **Fecha de expiración**: 04/25  **Clave de seguridad**: 456  **Fondos de la tarjeta**: Insuficientes | 2, 4, 6, 9 |
| 11 | **Número de tarjeta**: 3456 5678 7980 7987 6  **Fecha de expiración**: 05/27  **Clave de seguridad**: 4253  **Fondos de la tarjeta**: Suficientes | 2, 4, 7, 8 |
| 12 | **Número de tarjeta**: 1234 5678 9012 3456 7890  **Fecha de expiración**: 07/27  **Clave de seguridad**: 0727  **Fondos de la tarjeta**: Insuficientes | 2, 4, 7, 9 |
| 13 | **Número de tarjeta**: 7463 2655 4654 3732 64  **Fecha de expiración**: 01/01  **Clave de seguridad**: 956  **Fondos de la tarjeta**: Suficientes | 2, 5, 6, 8 |
| 14 | **Número de tarjeta**: 4563 5683 5845 6365 8378 546  **Fecha de expiración**: 07/08  **Clave de seguridad**: 094  **Fondos de la tarjeta**: Insuficientes | 2, 5, 6, 9 |
| 15 | **Número de tarjeta**: 8734 6582 3745 6275 4273 852 3  **Fecha de expiración**: 10/23  **Clave de seguridad**: 5587456347  **Fondos de la tarjeta**: Suficientes | 2, 5, 7, 8 |
| 16 | **Número de tarjeta**: 1111 2222 3333 4444 5555  **Fecha de expiración**: 11/22  **Clave de seguridad**: 3456  **Fondos de la tarjeta**: Insuficientes | 2, 5, 7, 9 |
| 17 | **Número de tarjeta**: 1234 5678 90  **Fecha de expiración**: 05/29  **Clave de seguridad**: 848  **Fondos de la tarjeta**: Suficientes | 3, 4, 6, 8 |
| 18 | **Número de tarjeta**: 5534 9873 66  **Fecha de expiración**: 12/30  **Clave de seguridad**: 672  **Fondos de la tarjeta**: Insuficientes | 3, 4, 6, 9 |
| 19 | **Número de tarjeta**: 095  **Fecha de expiración**: 09/26  **Clave de seguridad**: 2346 4576 9038 2759  **Fondos de la tarjeta**: Suficientes | 3, 4, 7, 8 |
| 20 | **Número de tarjeta**: 4587 3482 1039 439  **Fecha de expiración**: 03/31  **Clave de seguridad**: 4053  **Fondos de la tarjeta**: Insuficientes | 3, 4, 7, 9 |
| 21 | **Número de tarjeta**: 3568 04  **Fecha de expiración**: 01/24  **Clave de seguridad**: 987  **Fondos de la tarjeta**: Suficientes | 3, 5, 6, 8 |
| 22 | **Número de tarjeta**: 0112 3581 3471 123  **Fecha de expiración**: 10/07  **Clave de seguridad**: 278  **Fondos de la tarjeta**: Insuficientes | 3, 5, 6, 9 |
| 23 | **Número de tarjeta**: 3738 2538 746  **Fecha de expiración**: 08/22  **Clave de seguridad**: -000  **Fondos de la tarjeta**: Suficientes | 3, 5, 7, 8 |
| 24 | **Número de tarjeta**: 3845 7247 38  **Fecha de expiración**: 07/00  **Clave de seguridad**: 123456  **Fondos de la tarjeta**: Insuficientes | 3, 5, 7, 9 |

**Figura 2.3.2.3** Clases inválidas para “rentar película”

**CASOS DE PRUEBA**

Figura 2.3.2.4 muestra la tabla de casos de prueba para el plan de pruebas “rentar película”

|  |  |  |  |
| --- | --- | --- | --- |
| No. Caso | Información de la tarjeta | Resultados esperados | Resultados reales |
| 1 | **Número de tarjeta**: 6754 6785 2345 9800  **Fecha de expiración**: 06/28  **Clave de seguridad**: 347  **Fondos de la tarjeta**: Suficientes | El sistema realiza el cobro y muestra un mensaje de operación exitosa |  |
| 2 | **Número de tarjeta**: 5630 4585 4762 5987  **Fecha de expiración**: 07/25  **Clave de seguridad**: 478  **Fondos de la tarjeta**: Insuficientes | El sistema muestra un mensaje indicando que los fondos de la tarjeta son insuficientes |  |
| 3 | **Número de tarjeta**: 4576 7589 3485 0054  **Fecha de expiración**: 06/30  **Clave de seguridad**: 04  **Fondos de la tarjeta**: Suficientes | El sistema muestra un mensaje diciendo que ocurrió un problema al realizar el cobro |  |
| 4 | **Número de tarjeta**: 3287 2098 2683 5983  **Fecha de expiración**: 02/27  **Clave de seguridad**: 22  **Fondos de la tarjeta**: Insuficientes | El sistema muestra un mensaje diciendo que ocurrió un problema al realizar el cobro |  |
| 5 | **Número de tarjeta**: 4327 8523 5932 3478  **Fecha de expiración**: 03/22  **Clave de seguridad**: 349  **Fondos de la tarjeta**: Suficientes | El sistema muestra un mensaje diciendo que ocurrió un problema al realizar el cobro |  |
| 6 | **Número de tarjeta**: 2348 7475 6385 8043  **Fecha de expiración**: 12/12  **Clave de seguridad**:584  **Fondos de la tarjeta**: Insuficientes | El sistema muestra un mensaje diciendo que ocurrió un problema al realizar el cobro |  |
| 7 | **Número de tarjeta**: 1295 9230 9572 9054  **Fecha de expiración**: 09/28  **Clave de seguridad**: 01  **Fondos de la tarjeta**: Suficientes | El sistema muestra un mensaje diciendo que ocurrió un problema al realizar el cobro |  |
| 8 | **Número de tarjeta**: 7823 4782 3579 0121  **Fecha de expiración**: 05/24  **Clave de seguridad**: -457  **Fondos de la tarjeta**: Insuficientes | El sistema muestra un mensaje diciendo que ocurrió un problema al realizar el cobro |  |
| 9 | **Número de tarjeta**: 1295 2467 3468 2476 3498 54  **Fecha de expiración**: 07/28  **Clave de seguridad**: 789  **Fondos de la tarjeta**: Suficientes | El sistema muestra un mensaje diciendo que ocurrió un problema al realizar el cobro |  |
| 10 | **Número de tarjeta**: 4567 4577 4564 1209 12  **Fecha de expiración**: 04/25  **Clave de seguridad**: 456  **Fondos de la tarjeta**: Insuficientes | El sistema muestra un mensaje diciendo que ocurrió un problema al realizar el cobro |  |
| 11 | **Número de tarjeta**: 3456 5678 7980 7987 6  **Fecha de expiración**: 05/27  **Clave de seguridad**: 4253  **Fondos de la tarjeta**: Suficientes | El sistema muestra un mensaje diciendo que ocurrió un problema al realizar el cobro |  |
| 12 | **Número de tarjeta**: 1234 5678 9012 3456 7890  **Fecha de expiración**: 07/27  **Clave de seguridad**: 0727  **Fondos de la tarjeta**: Insuficientes | El sistema muestra un mensaje diciendo que ocurrió un problema al realizar el cobro |  |
| 13 | **Número de tarjeta**: 7463 2655 4654 3732 64  **Fecha de expiración**: 01/01  **Clave de seguridad**: 956  **Fondos de la tarjeta**: Suficientes | El sistema muestra un mensaje diciendo que ocurrió un problema al realizar el cobro |  |
| 14 | **Número de tarjeta**: 4563 5683 5845 6365 8378 546  **Fecha de expiración**: 07/08  **Clave de seguridad**: 094  **Fondos de la tarjeta**: Insuficientes | El sistema muestra un mensaje diciendo que ocurrió un problema al realizar el cobro |  |
| 15 | **Número de tarjeta**: 8734 6582 3745 6275 4273 852 34  **Fecha de expiración**: 10/23  **Clave de seguridad**: 5587456347  **Fondos de la tarjeta**: Suficientes | El sistema muestra un mensaje diciendo que ocurrió un problema al realizar el cobro |  |
| 16 | **Número de tarjeta**: 1111 2222 3333 4444 5555  **Fecha de expiración**: 11/22  **Clave de seguridad**: 3456  **Fondos de la tarjeta**: Insuficientes | El sistema muestra un mensaje diciendo que ocurrió un problema al realizar el cobro |  |
| 17 | **Número de tarjeta**: 1234 5678 90  **Fecha de expiración**: 05/29  **Clave de seguridad**: 848  **Fondos de la tarjeta**: Suficientes | El sistema muestra un mensaje diciendo que ocurrió un problema al realizar el cobro |  |
| 18 | **Número de tarjeta**: 5534 9873 66  **Fecha de expiración**: 12/30  **Clave de seguridad**: 672  **Fondos de la tarjeta**: Insuficientes | El sistema muestra un mensaje diciendo que ocurrió un problema al realizar el cobro |  |
| 19 | **Número de tarjeta**: 095  **Fecha de expiración**: 09/26  **Clave de seguridad**: 2346 4576 9038 2759  **Fondos de la tarjeta**: Suficientes | El sistema muestra un mensaje diciendo que ocurrió un problema al realizar el cobro |  |
| 20 | **Número de tarjeta**: 4587 3482 1039 439  **Fecha de expiración**: 03/31  **Clave de seguridad**: 4053  **Fondos de la tarjeta**: Insuficientes | El sistema muestra un mensaje diciendo que ocurrió un problema al realizar el cobro |  |
| 21 | **Número de tarjeta**: 3568 04  **Fecha de expiración**: 01/24  **Clave de seguridad**: 987  **Fondos de la tarjeta**: Suficientes | El sistema muestra un mensaje diciendo que ocurrió un problema al realizar el cobro |  |
| 22 | **Número de tarjeta**: 0112 3581 3471 123  **Fecha de expiración**: 10/07  **Clave de seguridad**: 278  **Fondos de la tarjeta**: Insuficientes | El sistema muestra un mensaje diciendo que ocurrió un problema al realizar el cobro |  |
| 23 | **Número de tarjeta**: 3738 2538 746  **Fecha de expiración**: 08/22  **Clave de seguridad**: -000  **Fondos de la tarjeta**: Suficientes | El sistema muestra un mensaje diciendo que ocurrió un problema al realizar el cobro |  |
| 24 | **Número de tarjeta**: 3845 7247 38  **Fecha de expiración**: 07/00  **Clave de seguridad**: 123456  **Fondos de la tarjeta**: Insuficientes | El sistema muestra un mensaje diciendo que ocurrió un problema al realizar el cobro |  |

**Figura 2.3.2.4** Casos de prueba para “rentar película”

**COMENTAR / PUNTUAR**

**TABLAS DE CONSIGNACIÓN**

Figura 2.3.3.1 muestra la tabla de consignación para el plan de pruebas “comentar” y “puntuar”

|  |  |  |  |
| --- | --- | --- | --- |
| **Condiciones de entrada** | **Regla heurística** | **Clases validas** | **Clases no validas** |
| **COMENTAR Y DAR PUNTUACIÓN.** | Valor específico | 1. Comentario <= 100 caracteres | 1. Comentario > 100 caracteres |
| 1. Puntuación = 0-3 |
| **RENTA** | Booleano | 1. Película rentada | 1. Película no rentada. |

**Figura 2.3.3.1** Tabla de consignación para “comentar” y “puntuar”

CLASES VÁLIDAS

Figura 2.3.3.2 muestra la tabla de clases válidas para el plan de pruebas “comentar” y “puntuar”

|  |  |  |
| --- | --- | --- |
| **NO. CASO** | **COMENTAR / PUNTUAR** | **CLASES VÁLIDAS** |
| 1 | **Comentario:** “¡Excelente película, muero de ganas de poder volver a verla!”  **Puntuación:** 3 Estrellas. | 1, 3 4. |

**Figura 2.3.3.2** Clases válidas para “comentar” y “puntuar”

CLASES NO VÁLIDAS

Figura 2.3.3.3 muestra la tabla de clases inválidas para el plan de pruebas “comentar” y “puntuar”

|  |  |  |
| --- | --- | --- |
| **NO. CASO** | **COMENTAR / PUNTUAR** | **CLASES INVÁLIDAS** |
| 2 | **Comentario:** (Si supera los 100 caracteres) | 2, 5 |

**Figura 2.3.3.3** Clases inválidas para “comentar” y “puntuar”

**CASOS DE PRUEBA**

Figura 2.3.3.4 muestra la tabla de casos de prueba para el plan de pruebas “comentar” y “puntuar”

|  |  |  |  |
| --- | --- | --- | --- |
| **NO. CASO** | **COMENTAR / PUNTUAR** | **RESULTADOS ESPERADOS** | **RESULTADOS REALES** |
| 1 | **Comentario:** (De <= 100 caracteres)  **Puntuación:** 1 estrella. | El sistema verifica la cantidad de caracteres que no sobrepase los 100, verifica la puntuación ingresada y actualiza los comentarios una vez ingresados el comentario. |  |
| 2 | **Comentario:** (De = 100 caracteres)  **Puntuación:** 2 estrellas. | El sistema verifica la cantidad de caracteres que no sobrepase los 100, verifica la puntuación ingresada y actualiza los comentarios una vez ingresados el comentario. |  |
| 3 | **Comentario:** (> 100 caracteres)  **Puntuación:** 3 estrellas. | El sistema muestra mensaje “Comentario excede la cantidad permitida de caracteres” y lo regresa a la caja de comentarios. |  |

**Figura 2.3.3.4** Casos de prueba para “comentar” y “puntuar”

**ELIMINAR LISTA**

**TABLAS DE CONSIGNACIÓN**

Figura 2.3.4.1 muestra la tabla de consignación para el plan de pruebas “eliminar lista”

|  |  |  |  |
| --- | --- | --- | --- |
| **Condiciones de entrada** | **Regla heurística** | **Clases válidas** | **Clases no válidas** |
| Listas | Booleano | 1. Lista en la base de datos | 1. No está en la base de datos |

**Figura 2.3.4.1** Tabla de consignación para “eliminar lista”

CLASES VÁLIDAS

Figura 2.3.4.2 muestra la tabla de clases válidas para el plan de pruebas “eliminar lista”

|  |  |  |
| --- | --- | --- |
| **No. Caso** | **Lista** | **Clases válidas** |
| 1 | **Lista:** Está en la base de datos | 1 |

**Figura 2.3.4.2** Clases válidas para “eliminar lista”

CLASES NO VÁLIDAS

Figura 2.3.4.3 muestra la tabla de clases inválidas para el plan de pruebas “eliminar lista”

|  |  |  |
| --- | --- | --- |
| **No. Caso** | **Lista** | **Clases inválidas** |
| 2 | **Lista:** No está en la base de datos | 2 |

**Figura 2.3.4.3** Clases inválidas para “eliminar lista”

**CASOS DE PRUEBA**

Figura 2.3.4.4 muestra la tabla de casos de prueba para el plan de pruebas “eliminar lista”

|  |  |  |  |
| --- | --- | --- | --- |
| **No. Caso** | **Lista** | **Resultados esperados** | **Resultados reales** |
| 1 | Lista “Mis favoritos”  (Base de datos) | Se elimina la lista de la interfaz del usuario y de la base de datos |  |
| 2 | Lista “Para ver con la familia”  (No está en la base de datos) | Manda mensaje de error |  |

**Figura 2.3.4.4** Casos de prueba para “eliminar lista”

**GESTIONAR PELICULA**

**TABLAS DE CONSIGNACIÓN**

Figura 2.3.5.1 muestra la tabla de consignación para el plan de pruebas “Gestionar película”

|  |  |  |  |
| --- | --- | --- | --- |
| condiciones de entrada | Regla Heurística | Clases Validas | Clases No validas |
| subir película | Rango | título de la película<=50 caracteres | titulo>50 caracteres |
| titulo nulo |
| Rango | Descripción <=500 caracteres | descripción>500 caracteres |
| descripción nula |
| Booleana | Imagen existe en la BD | imagen no existente en la BD |
| Imagen en formato JPG | Imagen en formato diferente |
| Imagen nula |

**Figura 2.3.1.1** Tabla de consignación para “Gestionar película”

Clases Validas

Figura 2.3.5.2 muestra la tabla de clases válidas para el plan de pruebas “Gestión de Películas”

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No Caso | Titulo<=50 | descripción <=500 | Imagen en BD | Formato de Imagen |
| 1 | titulo=40 | descripción=250 | true | Png |
| 2 | titulo=50 | descripción=350 | true | Png |
| 3 | titulo=22 | descripción=500 | true | Png |
|  |  |  |  |  |

**Figura 2.3.5.6** Clases válidas para “Gestionar Película”

Clases No validas

Figura 2.3.5.3 muestra la tabla de clases No válidas para el plan de pruebas “Gestión de Películas”

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No Caso | Titulo<=50 | descripción <=500 | Imagen en BD | Formato de Imagen |
| 4 | titulo=60 | descripción=450 | true | png |
| 5 | titulo=60 | descripción=600 | true | png |
| 6 | titulo=60 | descripcion=600 | false | png |
| 7 | titulo=60 | descripcion=600 | false | jpg |
| 8 | titulo=50 | descripcion=700 | true | png |
| 9 | titulo=50 | descripcion=500 | false | jpg |
| 10 | titulo=50 | descripcion=500 | true | jpg |
| 11 | titulo=45 | descripcion=450 | false | png |

**Figura 2.3.5.7** Clases Noválidas para “Gestionar Película”

## 2.4 Diseño de las interfaces (maquetas)

La figura 2.4.1 muestra la interfaz de bienvenida a la aplicación



**Figura 2.4.1** Interfaz de bienvenida a la aplicación

La figura 2.4.2 muestra la interfaz de inicio de sesión



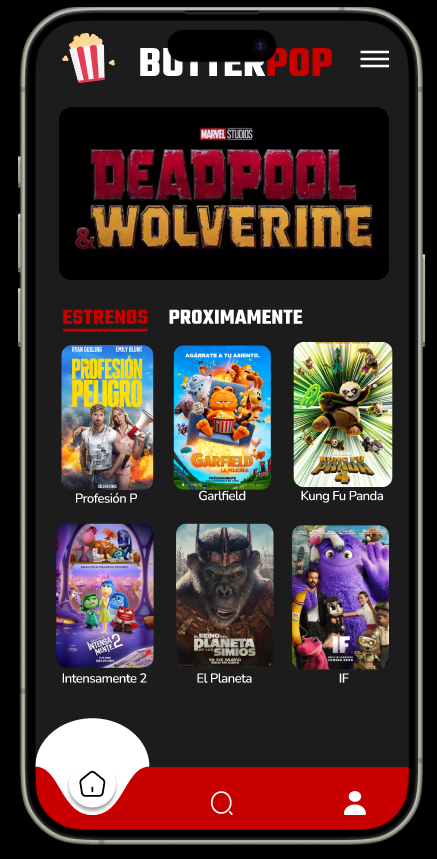
**Figura 2.4.2** Interfaz de inicio de sesión

La figura 2.4.3 muestra la interfaz de registrar cuenta



**Figura 2.4.3** Interfaz de registro de usuario

La figura 2.4.4 muestra la interfaz de principal (cartelera)



**Figura 2.4.4** Interfaz de principal (cartelera)

La figura 2.4.5 muestra la interfaz de selección de película



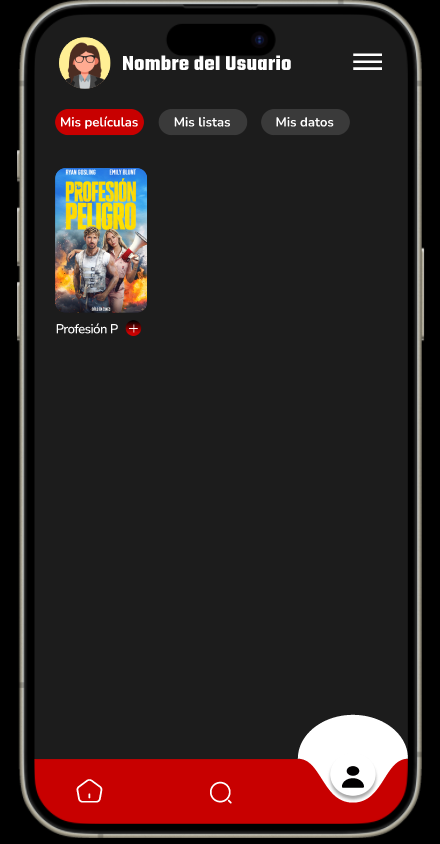
**Figura 2.4.5** Interfaz selección de película

La figura 2.4.6 muestra la interfaz de renta de película



**Figura 2.4.6** Interfaz renta de película

La figura 2.4.7 muestra la interfaz de "mis películas"



**Figura 2.4.7** Interfaz del perfil (mis películas)

La figura 2.4.8 muestra la interfaz de película rentada



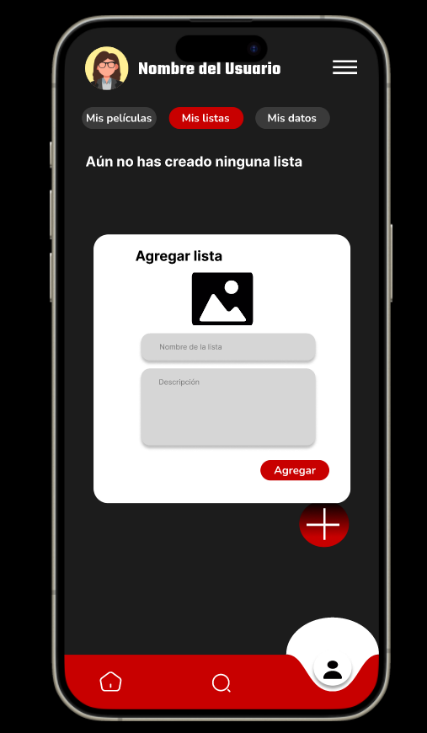
**Figura 2.4.8** Interfaz película rentada

La figura 2.4.9 muestra la interfaz de "mis listas"



**Figura 2.4.9** Interfaz mis listas

La figura 2.4.10 muestra la interfaz de crear lista



**Figura 2.4.10** Interfaz crear lista

La figura 2.4.11 muestra la interfaz de lista creada



**Figura 2.4.11** Interfaz lista creada

La figura 2.4.12 muestra la interfaz de editar lista



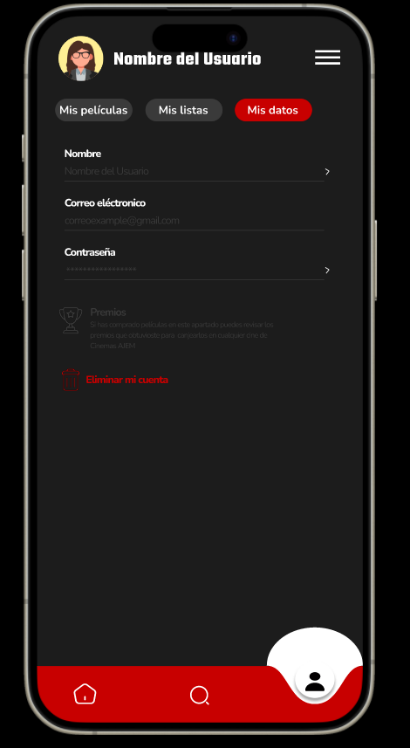
**Figura 2.4.12** Interfaz editar lista

La figura 2.4.13 muestra la interfaz de eliminar lista



**Figura 2.4.13** Interfaz eliminar lista

La figura 2.4.14 muestra la interfaz de "mi perfil"



**Figura 2.4.14** Interfaz de mi perfil

# **CAPÍTULO 3 IMPLEMENTACION Y PRUEBAS**

En el capitulo 3 se presenta la documentación de interfaces de la aplicación “ButterPop”

La figura 3.1.1 muestra la interfaz de registro de clientes, donde se solicitan los datos necesarios para que el cliente realice con éxito su registro en la aplicación



**Figura 3.1.1** Interfaz de registro de clientes

La figura 3.1.2 se muestra la tabla donde se describen detalladamente cada uno de los elementos de la interfaz de registro. En esta tabla, se especifican las funciones y características de cada campo requerido para completar el registro

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No. Elemento | Elemento | Interacciones | Tipo de dato de entrada | Campo que afecta |
| 1 | PanGestureRecognizer \_PanUpdate | Al deslizar el dedo en el control switch se pone de color rojo la parte seleccionada y muestra el formulario ya sea para iniciar sesión o registrarse. |  |  |
| 2 | OnText1Tapped | Al presionar el texto en el control switch se pone de color rojo la parte seleccionada y muestra el formulario ya sea para iniciar sesión o registrarse. |  |  |
| 3 | txtNombre | Entrada de tipo texto que permite al usuario ingresar su nombre. | String | Nombre |
| 4 | txtApaterno | Entrada de tipo texto que permite al usuario ingresar su apellido paterno. | String | Apaterno |
| 5 | txtAmaterno | Entrada de tipo texto que permite al usuario ingresar su apellido materno. | String | Amaterno |
| 6 | txtEmailR | Entrada de tipo email que permite al usuario ingresar su correo electrónico. | String | Correo |
| 7 | txtPassR | Entrada de tipo password que permite al usuario ingresar una contraseña. | String | Password |
| 8 | btnVerPassword | Al presionar el botón, la aplicación llama al método "btnVerPassword\_Clicked", el cual se encarga de mostrar los caracteres ocultos de la contraseña. Este método permite al usuario ver la contraseña ingresada. |  |  |
| 9 | BtnRegistrar | Al presionar el botón,9 la aplicación llama al método "btnRegistrar\_Clicked", el cual se encarga de validar que los campos ingresados por el usuario no estén vacíos y estén correctamente completados para posteriormente almacenar la información en la base de datos en la table Clientes. Posteriormente muestra la interfaz de inicio de sesión. |  |  |

**Figura 3.1.2** Tabla de elementos de la interfaz de registro de clientes

La figura 3.2.1 muestra la interfaz de creación de listas



**Figura 3.6.1** Interfaz de creación de listas

La figura 3.2.2 muestra la tabla donde describe los elementos de la interfaz de edición de lista. La interfaz Perfil.xaml contiene un frame que conecta a la base SQLite ButterPop.db3, la actualización afecta a en la tabla Lista al confirmar una actualización con el botón btnConfirmarEdicion. El campo actNombre, actDesc e imgEditar alteran la tabla Lista al cambiar los valores antes registrados

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No. Elemento | Elemento | Interacciones | Tipo de dato | Campo que afecta |
| 1 | btnImagen | Al presionar el botón, el usuario podrá ingresar una imagen. | Button |  |
| 2 | imgEditar | El usuario podrá visualizar la imagen seleccionada, previo a esto, muestra la imagen anterior. | Bytea[] | Imagen |
| 3 | actId | El entry únicamente muestra la Id correspondiente a la lista a editar. | Int | Id\_lista |
| 4 | actNombre | Permite al usuario ingresar un nuevo nombre a la lista. | String | Nombre |
| 5 | actDesc | Permite al usuario ingresar una nueva descripción a la lista. | String | Descripcion |
| 6 | btnCancelar | Al presionar el botón, el usuario podrá cancelar los cambios de la lista. | Button |  |
| 7 | btnConfirmarEdicion | Al presionar el botón, el usuario podrá confirmar los cambios de la lista. | Button |  |

**Figura 3.2.2** Tabla de elementos de la interfaz de creación de listas

La figura 3.3.1 muestra un formulario de pago donde el usuario podrá ingresar su número de tarjeta, la fecha de expiración, la clave de seguridad y la cantidad de semanas de renta que desee. Esta vista esta alojada en el archivo FormRenta.xaml. Una vez estén llenos todos los campos y el usuario de click en el botón 'Confirmar', la aplicación validará tos los campos, mostrará algunos DisplayAlert mostrando el resumen de la renta o información adicional. Cuando todos los datos hayan sido validados, la aplicación procesará los datos ingresados y creará un nuevo registro en la base de datos SQLite 'ButterPop.db3'



**Figura 3.3.1** Interfaz de formulario de cobro de renta

La figura 3.9.2 muestra una descripción de los elementos de la vista FormRenta.xaml

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| No. Elemento | Elemento | Tipo de elemento | Interacciones | Tipo de dato de entrada | Campo que afecta |
| 1 | nav\_back | Frame | Al presionarlo, la aplicación regresará a vista (pantalla) anterior |  |  |
| 2 | banner | Image | Imagen de cartel de la película previamente seleccionada | ImageSource |  |
| 3 | film\_name | Label | Título de la película previamente seleccionada | String |  |
| 4 | card\_number | Entry | Número de tarjeta bancaria, ingresada por el usuario | String |  |
| 5 | month | Entry | Mes a 2 cifras de la tarjeta bancaria, ingresada por el usuario | String |  |
| 6 | year | Entry | Año a 2 cifras de la tarjeta bancaria, ingresada por el usuario | String |  |
| 7 | cvv | Entry | Clave de seguridad de la tarjeta bancaria, ingresada por el usuario | String |  |
| 8 | output\_semanas | Label | Cantidad de semanas de renta | String |  |
| 9 | input\_slide\_semanas | Slider | Al deslizar el slide, la aplicación cambiará su valor entero desde 1 a 4 |  | semanas\_renta |
| 10 | btnConfirmCard | Button | Al presionar, la aplicación validará los campos y procesará el cobro |  |  |

**Figura 3.3.2** Tabla de elementos de la interfaz de formulario de cobro de renta

La figura 3.2.1 muestra la interfaz de gestión de películas (parte 1)



**Figura 3.4.1** Interfaz de gestión de películas (parte 1)

La figura 3.2.2 muestra la interfaz de gestión de películas (parte 2)



**Figura 3.4.2** Interfaz de gestión de películas (parte 2)

*En la figura 3.4.3 se muestra la tabla donde describe los elementos de la interfaz de gestión de películas por parte del administrador. La interfaz GestionPeliculas.xaml conecta a la base SQLite ButterPop.db3, el registro afecta a en la tabla Pelicula al confirmar un registro con el botón btnGuardarPelicula.*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No. Elemento | Elemento | Interacciones | Tipo de datos | Campo que afecta |
| 1 | imgPelicula | El elemento mostrará la imagen previamente seleccionada por el administrador que lleve la película | Byte[] | portada |
| 2 | btnSeleccionarImagen | Al dar click en el botón, la aplicación permitirá al administrador seleccionar la imagen para la portada de la película. |  |  |
| 3 | txtTitulo | Entrada de tipo texto que permite al administrador ingresar el título de la película. | String | titulo |
| 4 | txtDescripcion | Entrada de tipo texto que permite al administrador la sinopsis de la película. | String | sinopsis |
| 5 | generoPicker | Este picker permite al administrador escoger un género de entre varios que están dentro del picker. | String | genero |
| 6 | clasificacionPicker | Este picker permite al administrador escoger una clasificación de entre varias que están dentro del picker. | String | clasificacion |
| 7 | txtDuracion | Entrada de tipo numeric que permite al administrador ingresar la duración en minutos de la película. | int | duracion |
| 8 | txtPrecio | Entrada de tipo numeric que permite al administrador ingresar el precio de la película. | int | precio |
| 9 | btnGuardarPelicula | Al presionar el botón, la aplicación llama al método “btnGuardarPelicula\_Clicked”, el cual se encarga de validar que los campos ingresados por el administrador no estén vacíos y estén correctamente completados para posteriormente almacenar la información en la base de datos en la tabla Pelicula. Posteriormente muestra la interfaz de inicio de sesión. |  |  |

**Figura 3.4.2** Tabla de elementos de la interfaz de gestión de películas