**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**

**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: 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](#_Toc169027233)

[**ABSTRACT** 2](#_Toc169027234)

[**INTRODUCCIÓN** 3](#_Toc169027235)

[**OBJETIVOS** 5](#_Toc169027236)

[**PROGRAMA Y CRONOGRAMA** 6](#_Toc169027237)

[**MARCO TEÓRICO** 10](#_Toc169027238)

[**METODOLOGÍA** 15](#_Toc169027239)

[**CAPÍTULO 1: RECOLECCIÓN Y REFINAMIENTO DE REQUISITOS** 16](#_Toc169027240)

[1.1 Necesidades del cliente 16](#_Toc169027241)

[1.2 Identificación y definición de roles de usuario 16](#_Toc169027242)

[1.3 Requerimientos funcionales 17](#_Toc169027243)

[1.4 Requerimientos no funcionales 18](#_Toc169027244)

[1.5 Requerimientos De Hardware 19](#_Toc169027245)

[1.6 Requerimientos De Software 19](#_Toc169027246)

[1.7 Diagrama General De Casos De Uso 20](#_Toc169027247)

[1.8 Especificación De Los Casos De Uso 21](#_Toc169027248)

[1.9 Diagramas de Secuencia 27](#_Toc169027249)

[1.10 Diagrama de Actividades 35](#_Toc169027250)

[**CAPÍTULO 2: DISEÑO Y DESARROLLO DEL PROYECTO** 36](#_Toc169027251)

[2.1 Diagrama de la Base de Datos (relacional) 36](#_Toc169027252)

[2.2 Diccionario de Datos 36](#_Toc169027253)

[2.3 Diseño de Pruebas 38](#_Toc169027254)

[2.4 Diseño de las Interfaces (Maquetas) 44](#_Toc169027255)

# **RESUMEN**

# **ABSTRACT**

# **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  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 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.

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 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.

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 desarolladores 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.

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.

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.

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.

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.

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

**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.

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.

**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, funcionabilidad 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 empleará este software para tener un control de las fallas que pueda presentar la aplicación y tener un registro de dichas fallas y como se podrían solucionar.

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

**JavaScript**

Es un lenguaje de programación ligero, interpretado, de dialecto del estándar ECMAScript con funciones de primera clase. Si bien es más conocido como un lenguaje de secuencias de comandos para páginas web, y es usado en muchos entornos fuera del navegador, es un lenguaje de programación basado en prototipos, multiparadigma, de un solo hilo, dinámico, con soporte para programación orientada a objetos, imperativo, declarativo, débilmente tipado y dinámico.

Se utilizará este lenguaje del lado del cliente para aportar un mayor dinamismo a la aplicación además de reducir costos de procesamiento y energéticos a o los servidores que se vayan a emplear.

**JSP**

La tecnología Java Server Pages permite generar contenido Web dinámico como, por ejemplo, archivos HTML, DHTML, XHTML y XML, para incluirlos en una aplicación Web. Los archivos JSP son una forma de implementar contenido de páginas dinámico del lado del servidor. Los archivos JSP permiten a un servidor Web como, por ejemplo, Apache Tomcat, añadir contenido dinámicamente a las páginas HTML antes de enviarlas al navegador que las solicita.

Cuando se despliega un archivo JSP en un servidor Web que proporciona un motor de servlets, se procesa previamente en un servlet que se ejecuta en el servidor. Esto contrasta con JavaScript™ en el lado del cliente (dentro de códigos <SCRIPT>), que se ejecuta en un navegador. Una página JSP resulta ideal para tareas para tareas cuya ejecución es más adecuada en el servidor como, por ejemplo, acceder a bases de datos

**CSS3**

Hojas de Estilo en Cascada (del inglés Cascading Style Sheets) o CSS es el lenguaje de estilos utilizado para describir la presentación de documentos HTML o XML (incluyendo varios lenguajes basados en XML como SVG, MathML o XHTML). CSS describe cómo debe ser renderizado el elemento estructurado en la pantalla, en papel, en el habla o en otros medios. Es muy usado para establecer el diseño visual de los documentos web e interfaces de usuario escritas en HTML La última versión de este lenguaje, CSS3, incrementó significativamente el alcance de las especificaciones y el progreso de los diferentes módulos de CSS comenzó a mostrar varias diferencias, lo que hizo más efectivo desarrollar y publicar recomendaciones separadas por módulos.

**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 entorno de ejecución 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.

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

Se empleará 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.

# **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**

## 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. |

**Figura 1.3.1** Requisito funcional 1

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

**Figura 1.3.2** Requisito funcional 2

|  |
| --- |
| **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. |

**Figura 1.3.3** Requisito funcional 3

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

**Figura 1.3.4** Requisito funcional 4

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

**Figura 1.3.5** Requisito funcional 5

|  |
| --- |
| **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. |

**Figura 1.3.6** Requisito funcional 6

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

**Figura 1.3.7** Requisito funcional 7

## 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. |

**Figura 1.4.1** Requisito no funcional 1

**Figura 1.4.2** Requisito no funcional 2

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

**Figura 1.4.3** Requisito no funcional 3

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

**Figura 1.4.4** Requisito no funcional 4

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

**Figura 1.4.5** Requisito no funcional 5

## 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

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

|  |  |  |
| --- | --- | --- |
| **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

|  |  |  |
| --- | --- | --- |
| **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

|  |  |  |
| --- | --- | --- |
| **CU-0003** | **Rentar película** | |
| **Versión** | 1.1 (30/05/24) | |
| **Autores** | González Cruz Joel | |
| **Actores** | Cliente | |
| **Dependencias** | Ninguna | |
| **Descripción** | El sistema le permitirá al usuario rentar una o más películas de su elección durante 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. | |
| **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 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 | |
| **Excepciones** | **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 mostrará un mensaje indicando que ocurrió 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" |

**Figura 1.8.3** Rentar película

|  |  |  |
| --- | --- | --- |
| **CU-0004** | **Comentar película** | |
| **Versión** | 1.0 (15/05/24) | |
| **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. |
| **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”. |

**Figura 1.8.4** Comentar

|  |  |  |
| --- | --- | --- |
| **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 con 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”. |

**Figura 1.8.5** Puntuar película

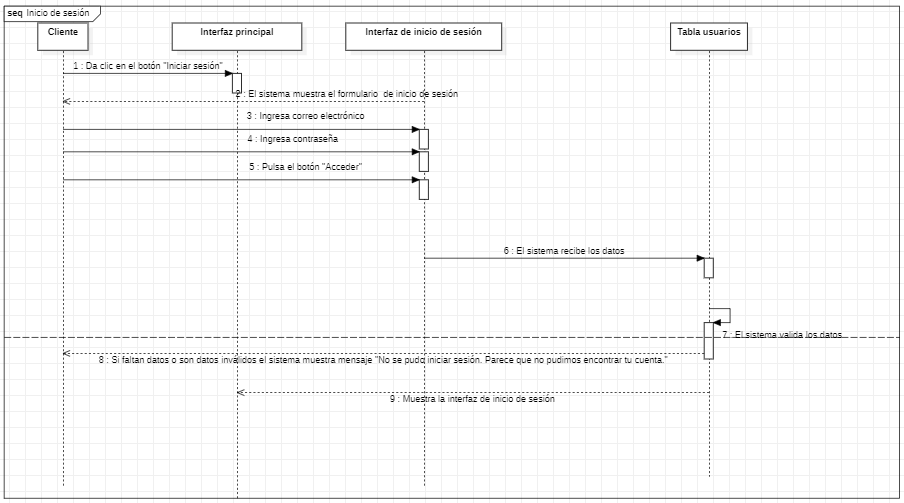
|  |  |  |
| --- | --- | --- |
| **CU-0005** | **Gestionar lista** | |
| **Versión** | 1.1 (15/05/24) | |
| **Autores** | Domínguez Zenteno Andrea | |
| **Fuentes** |  | |
| **Dependencias** | Ninguno | |
| **Descripción** | El sistema deberá comportarse tal cual se describe en el siguiente caso de uso 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 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. |

**Figura 1.8.6** Gestionar película

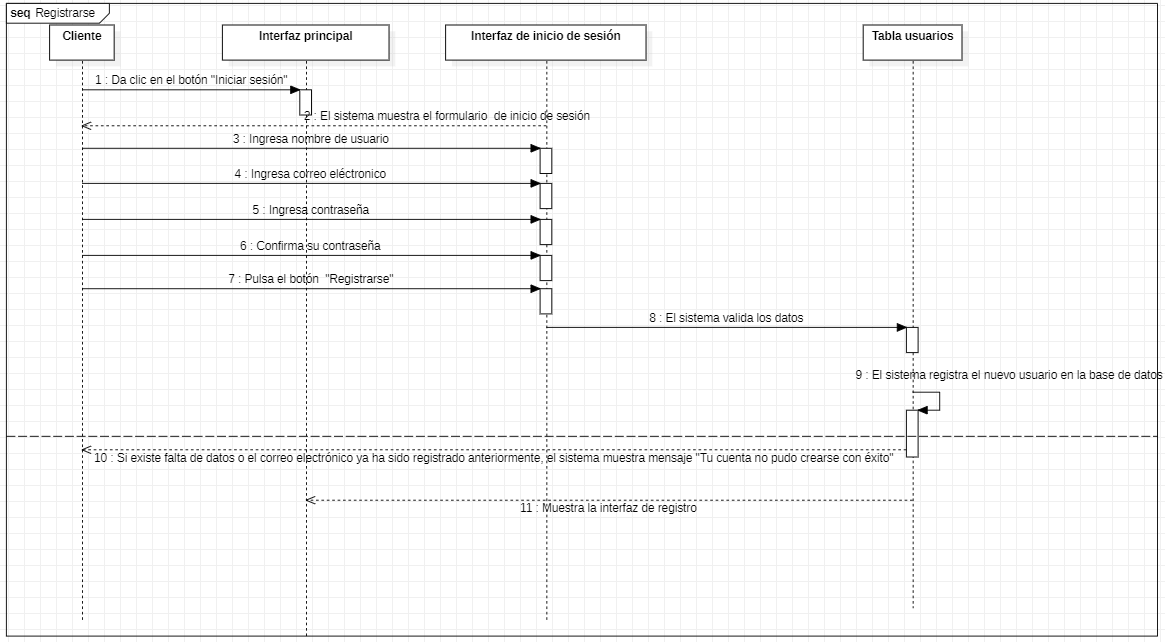
|  |  |  |
| --- | --- | --- |
| **CU-0006** | **Gestionar películas** | |
| **Versión** | 1.0 (15/05/24) | |
| **Autores** | Garcia Capote Emanuel | |
| **Fuentes** |  | |
| **Dependencias** | Ninguno | |
| **Descripción** | El sistema permitirá a los administradores agregar, eliminar o actualizar una película. | |
| **Precondición** | El administrador deberá de iniciar sesión previamente. | |
| **Secuencia normal** | **Paso** | **Acción** |
| 1 | Administrador da 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



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



**Figura 1.9.2** Diagrama de secuencia para registrarse

Tabla

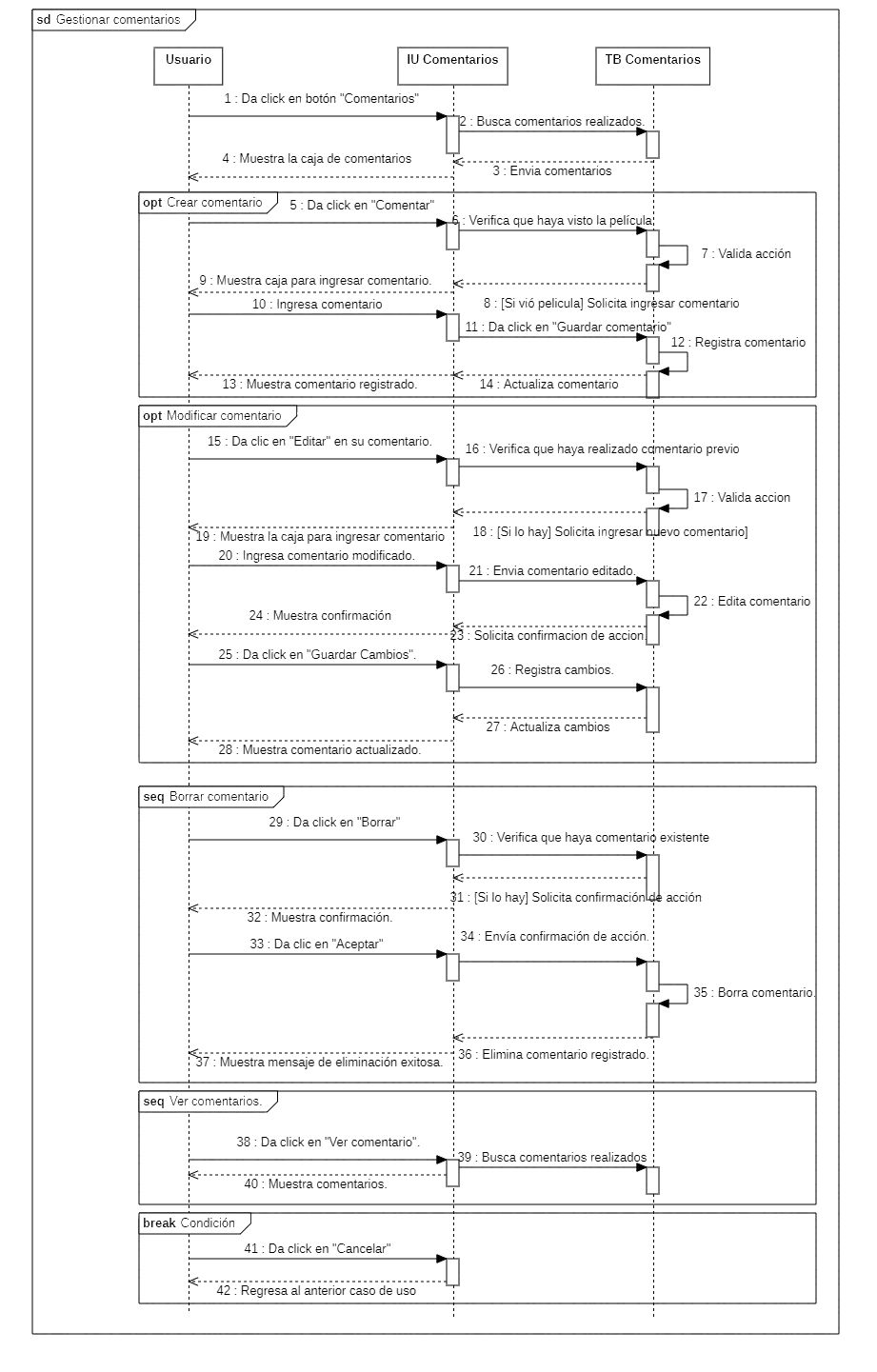
Descripción generada automáticamente

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

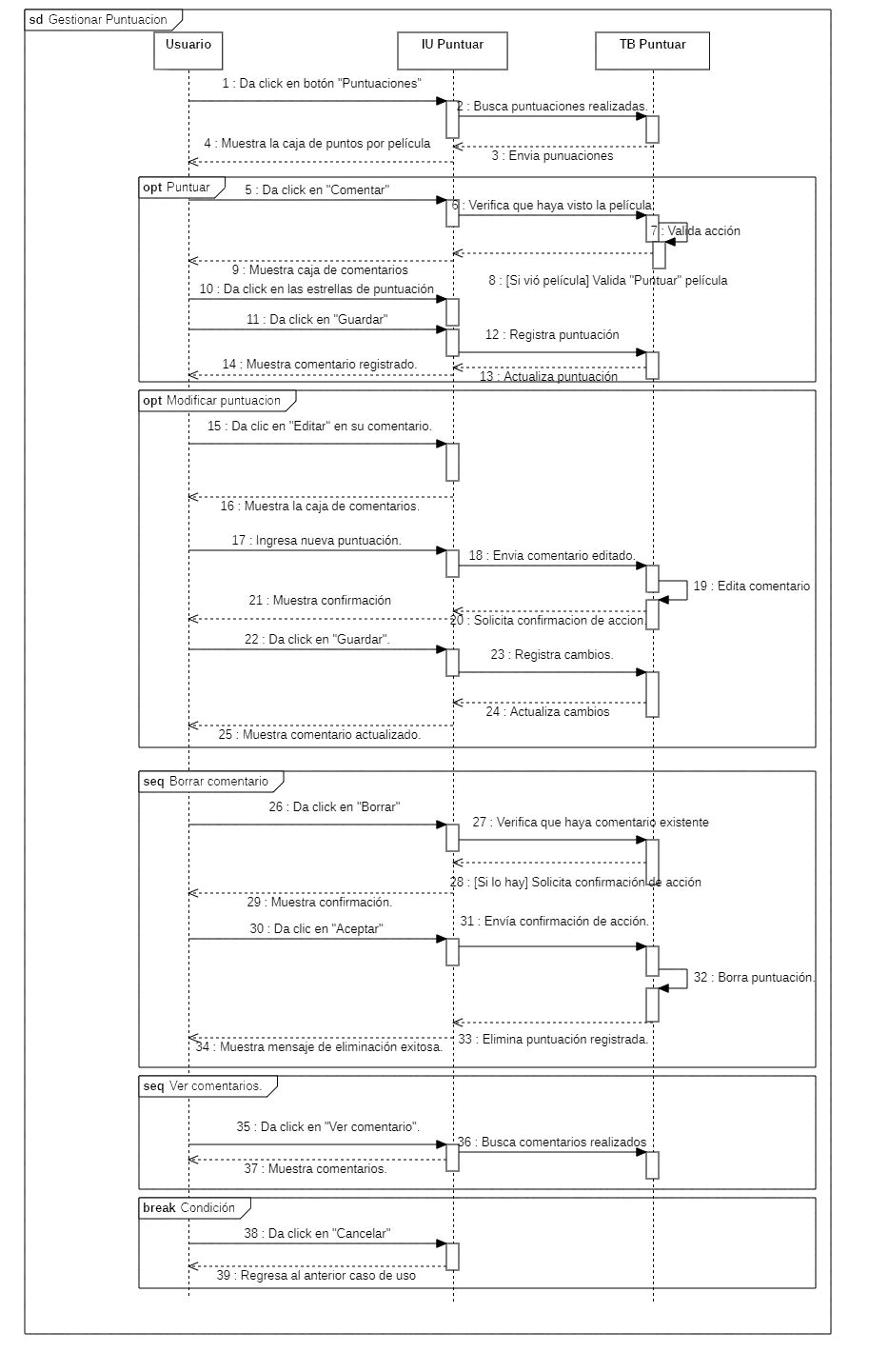
Tabla

Descripción generada automáticamente

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



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



**Figura 1.9.6** Diagrama de secuencia para gestionar puntuación

## 1.10 Diagrama de Actividades

Diagrama

Descripción generada automáticamente

**Figura 1.10.1** Diagrama de actividades

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

## 2.1 Diagrama de la Base de Datos (relacional)

## 2.2 Diccionario de Datos

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| TB PELICULA | | | | |
| Campo | Tipo | Extensión | Restricción | Observaciones |
| id\_pelicula | serial | N/A | PK | Llave primaria de película |
| correo\_admin | varchar | 30 | FK | Llave foránea del administrador que gestiona la película de la tabla administrador |
| 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 | float | N/A | N/A | Duración de la película (HH.MM) |
| archivo | bytea | N/A | N/A | Archivo de la película |
| precio | float | N/A | N/A | Precio de la renta de la película |
| portada | bytea | N/A | N/A | Portada de la película |
|  |  |  |  |  |
| TB RENTA | | | | |
| campo | tipo | Extensión | Restricción | Observaciones |
| id\_renta | serial | N/A | PK | Llave primaria de renta |
| correo\_cliente | 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 pelicula |
| fecha\_renta | timestamp | N/A | N/A | Fecha y hora en la que se registra la renta |
|  |  |  |  |  |
|  |  |  |  |  |
| TB COMENTAR | | | | |
| campo | tipo | Extensión | Restricción | Observaciones |
| id\_comentario | serial | N/A | PK | Llave primaria de comentar |
| correo\_cliente | 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 - 5.0) |
|  |  |  |  |  |
| 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 pelicula |
|  |  |  |  |  |
| TB CLIENTE | | | | |
| campo | tipo | Extensión | Restricción | Observaciones |
| correo\_cliente | varchar | 50 | PK | Llave primaria del cliente |
| contrasena | varchar | 20 | N/A | Hash de la contraseña del cliente |
| usuario | varchar | 30 | N/A | Nombre de usuario del cliente |
| no\_cuenta | varchar | 16 | N/A | Número de cuenta bancario del cliente |
|  |  |  |  |  |
|  |  |  |  |  |
| 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 |
|  |  |  |  |  |
| TB ADMINISTRADOR | | | | |
| campo | tipo | Extensión | Restricción | Observaciones |
| correo\_admin | serial | N/A | PK | Llave primaria del administrador |
| contrasena | varchar | 20 | N/A | Hash de la contraseña del administrador |

## 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**

|  |  |  |  |
| --- | --- | --- | --- |
| **CONDICIONES DE ENTRADA** | **REGLA EURÍSTICA** | **CLASES VÁLIDAS** | **CLASES NO VÁLIDAS** |
| **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.** ALMENOS UNA MINUSCULA | **7.** SIN MINUSCULAS |
| **8.** ALMENOS UNA MAYUSCULA | **9.** SIN MAYUSCULAS |
| **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 |

CLASES VÁLIDAS

|  |  |  |
| --- | --- | --- |
| **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 |

CLASES NO VÁLIDAS

|  |  |  |
| --- | --- | --- |
| **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 |

**CASOS DE PRUEBA**

|  |  |  |  |
| --- | --- | --- | --- |
| **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 |  |

**ELIMINAR LISTA**

**TABLAS DE CONSIGNACIÓN**

|  |  |  |  |
| --- | --- | --- | --- |
| 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 |

CLASES VÁLIDAS

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

CLASES NO VÁLIDAS

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

**CASOS DE PRUEBA**

|  |  |  |  |
| --- | --- | --- | --- |
| 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 |  |

## 2.4 Diseño de las Interfaces (Maquetas)

Interfaz de usuario gráfica

Descripción generada automáticamente con confianza media

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

Interfaz de usuario gráfica, Texto, Aplicación, Chat o mensaje de texto

Descripción generada automáticamente

**Figura 2.4.1** Interfaz de inicio de sesión

Interfaz de usuario gráfica, Texto, Aplicación, Chat o mensaje de texto

Descripción generada automáticamente

**Figura 2.4.2** Interfaz de registro de usuario

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

**Figura 2.4.3** Interfaz principal (cartelera)

Captura de pantalla de un celular

Descripción generada automáticamente

**Figura 2.4.4** Interfaz selección de película

Imagen de la pantalla de un celular

Descripción generada automáticamente

**Figura 2.4.5** Interfaz renta de película

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

**Figura 2.4.6** Interfaz del perfil (Mis películas)

Una captura de pantalla de un celular

Descripción generada automáticamente

**Figura 2.4.7** Interfaz película rentada

Interfaz de usuario gráfica

Descripción generada automáticamente

**Figura 2.4.8** Interfaz Mis listas

Interfaz de usuario gráfica, Texto, Aplicación, Chat o mensaje de texto

Descripción generada automáticamente

**Figura 2.4.8** Interfaz crear lista

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

**Figura 2.4.9** Interfaz lista creada

Captura de pantalla de un celular

Descripción generada automáticamente

**Figura 2.4.10** Interfaz de editar lista

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

**Figura 2.4.11** Interfaz eliminar lista

Captura de pantalla de un celular

Descripción generada automáticamente

**Figura 2.4.12** Interfaz de mi perfil