



Facultad de Ciencias y Tecnologías Computacionales

Desarrollo de la aplicación Picta para Smart TV Samsung

Trabajo de diploma para optar por el título de
Ingeniero en Ciencias Informáticas

Autor(es): Leydis Mariam Alarcón Sánchez

Tutor(es): Inst, Ing. Yosbel Falero Vento

Co-tutor: As, Ing. Bernardo Hernández González

Consultante: <nombre consultante (opcional de existir)>

Asesor: <nombre asesor (opcional de existir)>

La Habana, <mes> de <año>

<nombre oficial del año de la Revolución>

DECLARACIÓN DE AUTORÍA

El autor del trabajo de diploma con título “**Desarrollo de la aplicación Picta para Smart TV Samsung**”, concede a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la investigación, con carácter exclusivo. De forma similar se declara como único autor de su contenido. Para que así conste firma la presente a los ____ días del mes de _____ del año _____.

Leydis Mariam Alarcón Sánchez

Firma del Autor

Inst. Yosbel Falero Vento

Firma del Tutor

As. Bernardo Hernández González

Firma del Co-tutor

DATOS DE CONTACTO

<Curriculum e información de contacto del tutor: nombre y apellidos, títulos académicos, formación de postgrado recibida, lugar de trabajo, responsabilidades laborales asumidas, experiencia profesional, líneas de trabajo y/o investigación, correo electrónico, perfiles en redes profesionales>

<Curriculum e información de contacto del asesor: nombre y apellidos, títulos académicos, formación de postgrado recibida, lugar de trabajo, responsabilidades laborales asumidas, experiencia profesional, líneas de trabajo y/o investigación, correo electrónico, perfiles en redes profesionales>

<Curriculum e información de contacto del consultante: nombre y apellidos, títulos académicos, formación de postgrado recibida, lugar de trabajo, responsabilidades laborales asumidas, experiencia profesional, líneas de trabajo y/o investigación, correo electrónico, perfiles en redes profesionales>

AGRADECIMIENTOS

Quiero agradecer a todas las personas importantes en mi vida y darles las gracias, ellas son:

Mi mamá, por ser tan especial, amorosa y comprensiva, por ser la mejor madre que se puede desear, por guiarme en cada paso de mi vida y darme fuerzas cuando no las tenía. Gracias por secar mis lágrimas y disfrutar mis triunfos. Porque a pesar de que hoy no puedas acompañarme, sé que tu corazón esta junto al mío. Te amo.

A mi papá, por consentirme, por sus consejos y su afecto, por hacerme sentir su nena pequeña aun siendo la más grande, por no permitir que la distancia sea un impedimento para demostrar tu amor. Gracias por confiar en mí, por tu protección y tus locuras.

A mi abuelita Mami, por haberme enseñado casi todo lo que se, por cuidarme y estar siempre pendiente de mí, por cada abrazo, por cada beso y apoyarme en todo momento. Gracias por haber acompañado mis pasos con amor y darme la oportunidad de tener a alguien como tú en mi vida.

A mi tía Mary, por ese corazón tan noble, por tu cariño, tu apoyo constante y dedicación. Gracias por cuidarme como una madre, ayudarme en los momentos difíciles y ser parte de los buenos.

A mi familia en general por experimentar junto a mi mis logros y derrotas, por preocuparse por mí y darme fuerzas para seguir adelante.

A mi familia adoptiva, María, Rolando, Cristina, Eloy, Nisay y Sahir, gracias por acogerme como parte de su familia, por apoyarme siempre y estar a mi lado en momentos difíciles de mi vida, los quiero mucho y estaré eternamente agradecida por conocerlos y darme la oportunidad de ser parte de sus vidas.

A mis amigas Roxana, Adianez, Dora, Norbis, Adriana, Adisnelys y Tatiana por formar parte de varias etapas de mi crecimiento personal, por tanto, cariño y por brindarme siempre apoyo emocional.

A mis amigos, Naissa, Rosaida, Claudia, Amanda, Lorainy, Areidis, Jainy, Maivys, Aliet, Silvio, Álvaro, Adrian, Alejandro, Aldo, Carlos (Moro), Alejandro (White), Alfredo, Edgardo, Fernando, Julio y Dayron, gracias por las fiestas, las noches de desvelo, por los momentos mágicos e irrepetibles, por estar en la mejor etapa de mi vida, por ser mi apoyo sentimental y aguantar mis malcriadeces. Ustedes no son solo amigos, sino momentos que no se pueden olvidar, la universidad ha valido mas que la pena solo por darme la oportunidad de conocerlos a ustedes y poder disfrutar de tantas risas y llantos con personas verdaderamente especiales. Espero que el fin de la vida universitaria no sea una despedida, sino el comienzo de mucho tiempo de amistad.

A Leonel, Raydel, Dairon y Mauricio, por ayudarme tanto a superar este proceso, y por ser tan buenas amistades.

A mis tutores Yosbel y Bernardo, por haberme ayudado en cada momento que los he necesitado, durante todo el proceso de desarrollo de la tesis, sin su ayuda no habría sido posible realizar mi sueño.

DEDICATORIA

A mi mamá Mercy Sánchez Rego, la persona que más amo y amaré siempre, y a pesar de no estar a mi lado, sigue siendo mi motor impulsor y mi guía. A ella le debo la persona que soy hoy.

A mi abuela Miriam y mi tía Mariela, por estar siempre a mi lado y quererme como su hija, por apoyarme siempre y darme los mejores consejos.

RESUMEN

En la actualidad la tendencia hacia la convergencia entre Internet y el televisor está marcada por los líderes de la industria tecnológica. Una de las formas de transmisión de información más usuales en la Internet son los contenidos audiovisuales, por tanto, los sitios o plataformas dedicados a la gestión de este tipo de contenido son ampliamente visitados por los usuarios. El presente trabajo contiene la investigación basada en el desarrollo de aplicaciones para el sistema operativo Tizen de Smart TV Samsung. Contando con que en Cuba existen muchos televisores de este tipo, se propone mejorar la experiencia del usuario implementando la aplicación Picta con las especificaciones del SDK de Samsung. Para el desarrollo de la investigación se empleó la metodología de desarrollo AUP-UCI, posibilitando así la documentación de cada una de las etapas de vida del desarrollo del software. Además, se definieron las tecnologías y herramientas adecuadas para concretar la implementación de la aplicación propuesta. Como resultado se generó la aplicación Picta personalizada para Smart TV Samsung, la cual se validó mediante las pruebas unitarias, de aceptación y rendimiento, contribuyendo a mejorar la experiencia de los usuarios con el servicio.

Palabras claves: Internet, plataformas, Contenidos audiovisuales, Smart TV Samsung, aplicación Picta.

ABSTRACT

The expected result is to obtain the Picta application customized for the Tizen Operating System of Samsung Smart TV, which will contribute to improve the user experience with the service. This proposed solution will be implemented taking into account that this TV brand is one of the most widespread in the country. This is in addition to the fact that a significant part of the Cuban society in general, from the informatization process of the society, has some access to the Internet, even with economically attractive offers for national navigation, where Picta's domain is located and where it has preference over the traffic. The proposed solution will make it possible for users to avoid the need to search the Internet for the same contents accessible from Picta, thus contributing to reduce international traffic and its costs.

Key words: Internet, platforms, audiovisual content, Samsung Smart TV, Picta application.

TABLA DE CONTENIDOS

INTRODUCCIÓN	13
CAPÍTULO I: FUNDAMENTACIÓN TEÓRICA	19
1.1 Conceptos fundamentales	19
1.1.1 Smart TV	19
1.1.2 Sistemas Operativos.....	20
1.1.3 Sistema Operativo Tizen	21
1.1.4 Aplicaciones en Tizen	21
1.1.5 Tecnología Streaming.....	23
1.2 Soluciones existentes	24
1.2.1 Netflix.....	25
1.2.2 YouTube	26
1.2.3 Twitch	26
1.2.4 teVeo	27
1.2.5 Primicia.....	28
1.2.6 Observaciones	¡Error! Marcador no definido.
1.3 Caracterización de las tecnologías, herramientas y metodología.....	30
1.3.1 Herramientas utilizadas	30
1.3.2 Lenguaje de Modelado Unificado 2.5	30
1.3.4 Samsung TV SDK.....	31
1.4 Lenguajes para el modelado y desarrollo	32
1.4.1 UML	32
1.4.2 HTML5.....	33
1.4.3 CCS3	33
1.4.5 XML	34
1.5 Frameworks	35
1.5.1 Tizen.NET.....	35
1.5.2 Node.js	36
1.5 Metodología de desarrollo	37
CAPÍTULO II: DISEÑO E IMPLEMENTACION DE LA PROPUESTA DE SOLUCION.....	41
2.1 Modelo del dominio	41
2.1.2 Descripción de conceptos.....	41
2.2 Levantamiento de requisitos.....	42
2.2.1Requisitos Funcionales.....	42
2.2.2 Requisitos no funcionales.....	44
2.2.3 Historias de usuario	45
2.2.4 Plan de iteración	50
2.3 Arquitectura de software.....	51
2.3.1 Patrón arquitectónico.....	52
2.4 Modelo de diseño	53

2.4.1 Diagrama de clases del diseño.....	53
2.3.2 Patrones de diseño.....	56
Conclusiones del capítulo.....	59
CAPÍTULO III: IMPLEMENTACIÓN Y VALIDACIÓN DE LA SOLUCIÓN PROPUESTA.....	60
3.1 Modelo de Implementación.....	60
3.1.1 Diagrama de componentes.....	60
3.3 Métodos de pruebas.....	62
3.3.1 Pruebas unitarias.....	62
3.3.2 Pruebas de Caja Blanca.....	63
3.3.3 Pruebas de aceptación.....	67
3.3.1 Pruebas de Caja Negra.....	67
3.3.4 Pruebas de rendimiento.....	71
Conclusiones del capítulo.....	74
CONCLUSIONES FINALES.....	75
RECOMENDACIONES.....	76
REFERENCIAS BIBLIOGRÁFICAS.....	77
ANEXOS.....	80

ÍNDICE DE TABLAS

Tabla 1: Características principales de las plataformas analizadas.....	30
Tabla 2: Historia de usuario RF7	47
Tabla 3: Historia de usuario RF8	48
Tabla 4: Historia de usuario RF9	49
Tabla 5: Historia de usuario RF10	49
Tabla 6: Historia de usuario RF11	50
Tabla 7: Plan de iteración	51
Tabla 8: Prueba de Camino Básico #1	65
Tabla 9: Prueba de Camino Básico #2	66
Tabla 10: Caso de Prueba de Aceptación #1	68
Tabla 11: Caso de Prueba de Aceptación #2	69
Tabla 12: Caso de Prueba de Aceptación #3	69
Tabla 13: Caso de Prueba de Aceptación #4	70
Tabla 14: Caso de Prueba de Aceptación #5	70
Tabla 15: Resultados de la Prueba de Carga y Estrés.	73
Tabla 16: Historia de Usuario RF12.....	80
Tabla 17: Historia de Usuario RF13.....	81
Tabla 18: Historia de Usuario RF14.....	82

ÍNDICE DE FIGURAS

Figura 1: Modelo del dominio.....	41
Figura 2: Arquitectura Cliente-Servidor.....	52
Figura 3: Diagrama de clases del diseño Series.....	54
Figura 4: Diagrama de clases del diseño Películas.	55
Figura 5: Diagrama de clases del diseño Musicales.....	56
Figura 6: Patrón de diseño controlador.....	57
Figura 7: Patrón de diseño creador	58
Figura 8: Patrón de diseño experto.....	58
Figura 9: Diagrama de Componentes.....	61
Figura 10: Diagrama de despliegue.....	62
Figura 11: hideLoading.	64
Figura 12: Resultados de las Pruebas Unitarias.....	66
Figura 13: Resultados de las Pruebas de Aceptación.	71
Figura 14: Diagrama de clases de Documentales.	83
Figura 15: Diagrama de clases de Cartelera.	83
Figura 16:Aval de aceptación.	85

OPINIÓN DEL(OS) TUTOR(ES)

<Contenido de la opinión de los tutores>

AVAL DEL CLIENTE

<Contenido del aval del cliente sobre la solución desarrollada>

INTRODUCCIÓN

Los artefactos tecnológicos tienen dos funciones principales: reproducir y producir. La tecnología digital, en sus distintos soportes, permite reproducir toda clase de mensajes y contenidos; pero, por fortuna, también permite crear o producir contenidos y desarrollar nuevas formas de representación y, lo que es mejor, nuestros propios contenidos. Así, una de las principales funciones de los medios tecnológicos o herramientas digitales (para el caso es lo mismo) es la de permitir su uso como medio de expresión y comunicación (Moreno 2015).

Los contenidos audiovisuales actualmente constituyen una de las principales formas de transmisión de información en la sociedad, donde cada individuo puede identificar sus preferencias y disfrutar de ellas gracias a la variedad de contenidos disponibles. Estos poseen dos componentes: la información que contiene y el medio a través del cual se transmite. Muchos años han pasado desde el comienzo del desarrollo para la transmisión de señales audiovisuales a distancia. Los primeros ensayos de televisión se iniciaron de forma paralela en Estados Unidos y Europa en los años 30 del siglo XX. Ya en la década de los 50, la televisión llegaría a los principales países del mundo. En un principio, sólo se retransmitía durante ciertas horas al día y todos los contenidos eran en directo (Sánchez Márquez 2015).

Según un artículo sobre tecnologías de la información de la revista IESE, “en los últimos años el sector audiovisual está cambiando a pasos agigantados: supresión de la publicidad en TVE, proceso de digitalización de la radio, televisión por telefonía móvil, fragmentación de la audiencia con la implantación de la televisión digital terrestre (TDT), nuevo marco legal...” Se ha desarrollado un sector totalmente nuevo, debido a que la era digital ha abierto las puertas a una de las mayores revoluciones en los métodos de producción y distribución de los medios (López 2013).

En un contexto marcado por una profunda crisis económica, la industria audiovisual se ha visto en la necesidad de distribuir sus contenidos más allá de los soportes tradicionales, precisamente cuando su modelo de negocio ya se encontraba consolidado. Antes, el núcleo del negocio de una cadena de televisión comercial consistía en diseñar una oferta de

contenidos atractiva para conseguir audiencia y vender tiempo publicitario. Sin embargo, actualmente, su área de acción se extiende más allá de la pantalla del televisor, invadiendo otras como el ordenador o los dispositivos móviles (Rodríguez et al. 2013).

El espacio de la televisión por Internet es joven y la competencia está madura, por lo que la innovación es crucial. Un pilar clave de este proyecto es el sistema de recomendación que ayuda a nuestros miembros encontrar videos para ver en cada sesión. La utilidad y la temporalidad, están intrínsecamente ligados al hecho de que la visualización de contenidos audiovisuales en línea conduce a un cambio de actitud entre los usuarios.

La pantalla de televisión es un dispositivo altamente aceptado y presente en todos los hogares al igual que Internet, y por ende es un excelente canal para captar un mayor número de usuarios; junto con esto existen una serie de condicionantes y limitantes que exigen un profundo análisis del comportamiento de la interfaz que se provea a fin de lograr una alta aceptación por parte del público en general. El producto Smart Tv es una experiencia multimedia visualmente compleja con mucho contenido para el usuario. Esta experiencia incluye funcionalidades y servicios como video llamadas, streaming de series de Tv y películas, redes sociales, servicios meteorológicos o financieros, descarga de aplicaciones, entre otros (Medina et al. 2011).

Actualmente existen aplicaciones y sistemas informáticos que centran sus servicios en la publicación y reproducción de contenidos audiovisuales, y de reproducción directa y al aire (streaming), tales son los casos de la aplicación de Netflix y la red social Twitch. Netflix se encarga de exhibir producciones de terceros y, desde el año 2013, de distribuir y producir contenidos audiovisuales propios como series, películas y documentales. Por situarse en internet, como la principal plataforma de distribución de sus servicios y contenidos es considerada una empresa OTT1 (Over the top o encima de la nube), lo que le ha posibilitado también una importante expansión internacional e incorporar innovaciones en la prestación de su servicio de video online (Ruiz 2017).

Por otra parte, la plataforma de vídeo en directo Twitch, desde 2014 propiedad de Amazon, se ha erigido como una de las principales aglutinadoras mundiales del consumo audiovisual, supone un escaparate codiciado, para que las empresas de videojuegos o software anuncien en ella sus productos. Aunque ahora, todo tipo de empresas relacionadas con la tecnología en general se están anunciando en ella. Asimismo, es el canal por el que los principales campeonatos de deportes electrónicos retransmiten sus eventos. Es un espacio donde muchos usuarios anónimos intentan ganarse la vida emitiendo sus partidas en directo. De esta manera, los conceptos de "streaming" (como producción de contenidos en directo) adquiere un significado propio si se los asocia específicamente con esta plataforma (Gutiérrez Lozano and Cuartero Naranjo 2020).

La brecha digital creada entre Cuba y el resto del mundo obligó a utilizar tecnologías alternativas para preservar y fortalecer los logros de la Revolución en el campo de la educación. Una de las acciones realizadas en este sentido consiste en la fabricación de equipos y soportes audiovisuales, convertirse en un elemento complementario de la labor educativa del maestro, y así velar por el uso racional de dichos medios. El nuevo potencial de los televisores inteligentes se debe a la inclusión, en sus circuitos, de pequeños ordenadores como los que se utilizan en teléfonos móviles, tabletas o incluso en portátiles ordenadores. Hay varias empresas transnacionales que actualmente están desarrollando este tipo de televisión; entre ellos se encuentran los mundialmente conocidos LG, Google y Samsung (Rosales Rosa et al. 2014).

Picta es la plataforma cubana de contenidos audiovisuales creada por la Universidad de Ciencias Informáticas. La plataforma informática de entretenimiento dispone de series de televisión, películas y documentales entre otras propuestas. Por norma, todos los usuarios pueden buscar y reproducir materiales audiovisuales. Una vez realizado el registro podrán acceder a otras funciones como descargar, comentar, votar y denunciar una publicación determinada. Dentro de sus novedades ofrece el servicio de televisión en vivo a través de streaming de video. Además, puede suscribirse a uno o muchos canales. Dentro de los dispositivos que acceden a Picta están los Smart TV, sobre todo los Smart TV Samsung. Picta a pesar de contar con una

aplicación web, cuando la misma es instalada en algunos televisores o Smart TV, no se ajusta a ellos, por lo tanto, el usuario no puede instalar la aplicación, no tiene acceso al contenido multimedia, como consecuencia la experiencia de los usuarios que cuenten con Smart TV Samsung no es buena. Como parte de la estrategia del Departamento de Informatización en conjunto con Etecsa, teniendo en cuenta que en Cuba existe una gran cantidad de televisores de este modelo, debido a la preferencia de los usuarios de esta marca por encima de otras, se desea mejorar la experiencia de usuario implementando la aplicación Picta con las especificaciones del SDK de Samsung.

Partiendo de la situación planteada se deriva el siguiente **Problema científico**: ¿Cómo mejorar la experiencia de usuario de la aplicación Picta para Smart TV Samsung?

La investigación se centra en Aplicaciones para el sistema operativo Tizen de Smart TV Samsung, lo cual constituye el **Objeto de estudio**, enmarcado en el **Campo de acción**: Aplicación de servicios de streaming para el Sistema Operativo Tizen de Smart TV Samsung.

Se define como **Objetivo general**: Desarrollar la aplicación Picta para el Sistema Operativo Tizen de Smart TV Samsung.

Para dar cumplimiento a los objetivos se trazó el siguiente plan de tareas de la investigación:

1. Fundamentación teórico-metodológico de las plataformas audiovisuales
2. Caracterización del sistema operativo en el que se va a desarrollar la aplicación.
3. Desarrollo de la aplicación Picta para dispositivos móviles con sistemas operativo Tizen
4. Selección de las herramientas adecuadas para el desarrollo la aplicación Picta para Smart TV Samsung con sistema operativo Tizen.
5. Elaboración de la documentación necesaria según la metodología de desarrollo seleccionada.

6. Análisis de diversas plataformas de streaming para Smart TV.
7. Validación de la solución mediante la aplicación de pruebas funcionales, de rendimiento, de usabilidad e integración.

Métodos Teóricos

- ✓ **Método histórico – lógico:** Este método se utilizó para analizar la trayectoria y evolución histórica de plataformas compatibles con Smart TV que permitan el consumo de contenidos audiovisuales, así como establecer las tendencias generales que rigen el funcionamiento y desarrollo de las mismas.
- ✓ **Método análisis – sintético:** Este método se utilizó con el objetivo de descomponer el proceso de consumo de contenidos audiovisuales desde los Smart TV, procesar los fundamentos y las teorías relacionadas a este proceso, lo que permite extraer los aspectos significativos que sustentan la propuesta de solución.
- ✓ **Método de Modelación:** Este método permitió la abstracción de objetos de la vida real a modelos informáticos que permiten el desarrollo de la solución propuesta.
- ✓ **Análisis documental:** Este método se utiliza para la selección y recolección de la documentación asociada al tema abordado, con el objetivo de extraer la información necesaria para lograr los resultados esperados.

Se plantea como **resultado esperado** obtener la aplicación Picta personalizada para el Sistema Operativo Tizen de Smart TV Samsung, lo cual contribuirá a mejorar la experiencia de los usuarios con el servicio. Esta propuesta de solución se implementará teniendo en cuenta que esta marca de TV constituye una de las más extendidas en el país. Esto se une además al hecho de que una parte significativa de la sociedad cubana en general, a partir del proceso de informatización de la sociedad, dispone de alguna vía de acceso a internet, incluso con ofertas económicamente atractivas para la navegación nacional, donde se encuentra el

dominio de Picta y donde tiene preferencia sobre el tráfico. La propuesta de solución posibilitará que los usuarios no tengan la necesidad de buscar en internet los mismos contenidos accesibles desde Picta, contribuyendo a disminuir el tráfico internacional y sus costos.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

En este capítulo se abordarán los principales conceptos sobre el desarrollo de aplicaciones para el sistema Operativo Tizen de Smart TV Samsung, así como se realizará el estudio de la bibliografía para detectar elementos que formarán parte de la propuesta de solución.

1.1 Conceptos fundamentales

Para obtener una mejor comprensión del dominio del problema, a continuación, son descritos conceptos asociados al mismo.

1.1.1 Smart TV

La Smart TV es una televisión conectada a Internet que, por su implicación en hardware y software, permite el acceso a múltiples fuentes de videos en una sola plataforma. Las Industrias involucradas y sus expertos señalan repetidamente que se avecina una revolución en la forma de vivir la televisión en donde se multiplicarán las oportunidades de creación de valor y negocio en el futuro cercano. Además, Combina TV e internet, y provee más contenido de video (noticias, juegos etc.), y multimedia. Cambiará los patrones actuales de ver la televisión, y permitirá la convergencia de varios servicios de comunicación. En este tipo de dispositivo podremos realizar nuevas actividades desde lo más cotidiano, como utilizar el Web browser (buscador de internet), el navegador, acceder a las redes sociales como Facebook y Twitter, y a un mundo de aplicaciones en los mercados. Asimismo, la tecnología de las Smart TV nos permite interactuar más fácilmente con otros dispositivos tales como tablets, Smartphone o hasta las mismas computadoras(Carpinteyro 2012).

El Smart TV es una plataforma cuyo concepto se basa en la integración de internet y las características Web a la televisión digital. Cada año los fabricantes de Smart TV integran nuevas funcionalidades a sus dispositivos de manera que permiten una mejor experiencia de usuario. Actualmente entre las funcionalidades que contienen los Smart TV se encuentra el reconocimiento de gestos, el reconocimiento de voz, la recomendación de visualización y el social tv. La interconectividad de una Smart TV es su principal fortaleza. En virtud de una noción de Steve Jobs que planteó que la gente use sus televisores de pantalla ancha para entretenerse. Lo cual no aísla la existencia del televisor como un artefacto desconectado de

los demás dispositivos. Samsung, la empresa líder en el mercado de televisión, ha apostado por un sistema operativo diferente. En él permite a los usuarios disfrutar de un contenido diferenciado con tecnología de alta definición y un diseño sofisticado (López Chiu and Vega Layme 2019).

De lo anterior se puede inferir que un televisor inteligente o Smart TV, es un televisor que cumple su función principal, pero con características adicionales muy avanzadas, que permiten al usuario disfrutar del acceso a diversas aplicaciones, juegos, videos y a internet, como si se tratara de una computadora inmersa en un televisor.

1.1.2 Sistemas Operativos

La mayor parte de los equipos electrónicos como las computadoras, celulares, y otros equipos que usamos en la actualidad funcionan con los microchips determinados como procesadores en él se incluyen sistemas operativos, los sistemas operativos son fundamentales por que permiten interactuar con los usuarios haciendo un manejo favorable en su entorno, estos sistemas operativos son fundamentales, eficaces permiten que los programas y aplicaciones funcionen correctamente (Monterrubio-Hernandez 2019).

Los sistemas operativos requieren obtener continuamente actualizaciones necesarias para brindar su correcto funcionamiento en los servicios que requiere, sin afectar el funcionamiento existente, cabe mencionar que constantemente se actualiza la versión del Sistema operativo, realizando cambios en el entorno gráfico, y otros servicios que permiten facilitar su uso y que favorezca las necesidades del usuario, hacen que este sea más sencillo y eficaz (Monterrubio-Hernandez 2019).

Podemos considerar de gran importancia los sistemas operativos debido que permiten interactuar y darle órdenes al computador. Sin un sistema operativo el computador sería inútil. No se tendría la plataforma que soporta los programas que permiten hacer cartas, escuchar música, navegar por internet o enviar un correo electrónico.

1.1.3 Sistema Operativo Tizen

Tizen OS es un sistema operativo ligero desarrollado por Samsung con la ayuda de Intel y la Linux Fundación a partir de un sistema operativo abandonado por Nokia e Intel, llamado MeeGo, un sistema operativo para móviles que Nokia abandonó cuando éste decidió pasar del sistema Linux al sistema de Microsoft Windows IPhone 7. Tizen OS fue desarrollado desde comienzo con el objetivo de poder ser utilizado sobre varios dispositivos fundamentalmente diferentes, como móviles, wearables (relojes inteligentes) y televisiones. Tizen OS está basado en el kernel de Linux y opera con herramientas de código abierto, para ser desarrollado por todos que lo deseen. Su SDK (Kit de Desarrollo Software) está construido sobre componentes de código abierto (Strasser 2018)

Sus creadores han puesto un gran esfuerzo en que sea un sistema personalizable y configurable por cualquiera que lo desee, permitiéndose diseñar perfiles de manejo del sistema o incluso personalizarlo para usuarios específicos. Gran parte de este permite el desarrollo de aplicaciones y manejo de interfaz con HTML5, permitiendo que se integre fácilmente con aplicaciones web o plataformas que estén desarrolladas con tecnologías web. Igualmente, Tizen OS es compatible con Android, brindándole la posibilidad de usar una gran cantidad de las aplicaciones de este (Salazar Serna and Escobar Ceballos 2016).

1.1.4 Aplicaciones en Tizen

El desarrollo e implementación de aplicación para todo tipo de utilidades sobre las televisiones de hoy en día era un paso esperado. Durante los últimos años todo tipo de actividades que previamente se podían realizar únicamente sobre un ordenador personal han sido trasladados a nuestros dispositivos móviles y tablets, como es navegar por internet, escribir correos electrónicos e incluso jugar juegos. Era de esperar que todo esto acabaría siendo parte de nuestras televisiones como un estándar(Strasser 2018). Tizen.NET es una forma nueva y emocionante de desarrollar aplicaciones para el sistema operativo Tizen, que se ejecuta en 50 millones de dispositivos Samsung, incluidos televisores, dispositivos portátiles, teléfonos móviles y muchos otros dispositivos IoT en todo el mundo.

Es posible desarrollar dos tipos de aplicaciones para Tizen, la misma dependerá del grado de dependencia para el dispositivo (ANTONIO 2017).

-Aplicaciones tipo web

-Aplicaciones nativas

Aplicaciones nativas

Las aplicaciones nativas son consideradas aplicaciones convencionales móviles, en otras palabras, están implementadas en el lenguaje nativo del propio terminal, por lo que todos los recursos de éste estarán accesibles para poder sacar el máximo partido la aplicación. Estas residen en el dispositivo, suelen desarrollarse con lenguajes de programación que requieren una compilación anterior y se instalan directamente en el smartphone, a través de un medio de distribución de aplicaciones. De este modo, se consigue una mayor difusión de la aplicación y es más accesible para los usuarios. Además, al terminar la instalación de la aplicación, el usuario dispondrá de un acceso directo de ésta para poder lanzarla de una forma rápida y sencilla. Cabe destacar también, que este tipo de aplicaciones reciben las notificaciones y las actualizaciones automáticas al instante, es decir, éstas pueden funcionar, aunque el teléfono no esté conectado a Internet, así los usuarios pueden acceder a la información de la aplicación en cualquier momento (Román López 2014).

El modelo de aplicación Tizen Native maneja el ciclo de vida de la aplicación y los eventos del sistema. El ciclo de vida de la aplicación nativa de Tizen es manejado por la API de la aplicación (en aplicaciones móviles y portátiles). Proporciona funciones para administrar el bucle de eventos principal, los eventos de cambio de estado de la aplicación y los eventos básicos del sistema.

Aplicaciones web

Una aplicación web consta de HTML, JavaScript y CSS combinados en un paquete, que se puede instalar en el dispositivo Tizen. Un paquete de aplicación web incluye todos los archivos de soporte que necesita la aplicación web. Por lo tanto, una aplicación web puede ejecutarse sin ningún recurso externo adicional o conectividad de red después de la instalación. El modelo

de aplicación es compatible con un amplio conjunto de funciones W3C/HTML5 estándar, que incluyen varias API de JavaScript, así como marcas HTML y funciones CSS adicionales. Estas funciones, junto con las API de dispositivos Tizen y la compatibilidad con el marco de la interfaz de usuario, se pueden usar para crear aplicaciones web enriquecidas en una variedad de categorías, como contactos, mensajería, acceso a la información del dispositivo, multimedia, gráficos y juegos. También se puede diseñar una aplicación web que sea muy similar a una nativa, aunque realmente no lo sea. Para este tipo de aplicaciones el problema que se presenta es que los lenguajes web no tienen todavía compatibilidad con todas las funciones nativas (API) (Román López 2014).

En cambio, una habilidad que se destaca sobre las aplicaciones web en comparación con las nativas, es su gran versatilidad, debido a que son multiplataforma, lo que significa que se puede ejecutar en cualquier sistema operativo, basta con que el terminal disponga de un navegador. Esto conlleva a que no es necesario tener que volver a programar, por lo tanto, esto disminuye los problemas que traen las aplicaciones nativas creadas para un único sistema operativo que requieren atención constante, lo que implica un aumento de los costes. Asimismo, si comparamos éstas con las nativas, las aplicaciones web requieren un tiempo invertido relativamente corto, debido a que la programación en HTML 5 es mucho más simple y esto hace que se necesiten menos requisitos. Y las ventajas siguen aumentando, ya que, en caso de actualización, no es necesario que el usuario deba actualizar la aplicación, pues al ser una página web, siempre accederá a la versión más actualizada y, además, al no ser descargable desde los medios de distribución de aplicaciones, no es necesario pasar ningún proceso de validación (Román López 2014).

1.1.5 Tecnología Streaming

Es una tecnología (también conocida como media-flujo) que permite la distribución de archivos multimedia (principalmente audio y video) a través de la red de computadores en tiempo real. Estos son cargados en un buffer (cola de pedidos) de datos donde se puedan reproducir al mismo tiempo que se descargan y luego se descartan, sin quedar almacenados en el disco duro del cliente. Esta tecnología nació en el año 1995 junto con la aparición de RealAudio 1.0,

un formato que permite justamente reproducir el audio en tiempo real; pero no se hizo popular sino hasta el año 2000 cuando el ancho de banda se hizo lo suficientemente barato y rápido para gran parte de la población. Antes de la aparición de esta tecnología, la reproducción de contenido multimedia a través de Internet implicaba descargar completamente el archivo al disco duro local. Como estos archivos multimedia son de gran tamaño, su descarga y acceso como paquetes completos resultaba un mecanismo lento (González 2017).

El streaming año a año se ha ganado su lugar. Se puede observar su crecimiento al ver las ganancias generadas por sus producciones y como ha superado a las salas de cine. Este aumento del streaming da cuenta de cómo las prácticas sociales mediadas por las tecnológicas modifican nuestras formas de consumir, ver y producir cultura audiovisual. Pasamos de las salas de cine a la comodidad de nuestras casas. Este traspaso surge por el costo de las suscripciones: al ser más accesibles, en términos monetarios, más personas acceden al servicio generándose así un aumento del consumo y producciones en este formato. Se puede decir que hemos pasado a incorporarlas en nuestras vidas porque nos han facilitado disfrutar de una película, serie o documental en nuestros televisores o celulares de forma inmediata y sin necesidad de movilizarnos (Llanes 2021).

Las plataformas streaming son tan importantes e influyentes hoy en día que incluso en algunas casas está desconectado el cable de la antena del televisor. Cada vez son más numerosos los usuarios que no consumen la televisión en directo, siendo los únicos contenidos que ven aquellos que están disponibles en este tipo de servicios. Permite poder disfrutar en cualquier momento del contenido deseado, sin necesidad de esperar que llegue la hora de emisión. Además, tanto las películas como las series, los documentales y otros tipos de contenidos se reproducen con una calidad excelente que supera a la de la televisión.

1.2 Soluciones existentes

En la actualidad existen disímiles sitios que brindan servicios de hospedaje de contenidos audiovisuales, en los cuales los usuarios pueden interactuar con los contenidos expuestos. Entre ellos podemos encontrar: Netflix, YouTube, Twitch, TeVeo y Primicia.

1.2.1 Netflix

Netflix se configura en un nuevo modelo de creación y distribución de contenidos audiovisuales que recoge aprendizajes de los modelos clásicos televisivos con las lógicas de funcionamiento del internet y el entretenimiento *streaming*. De hecho, esta corporación pasó de definirse como una red de televisión global a proclamarse como el principal servicio de entretenimiento por internet en el mundo. Con 200 millones de suscriptores y presencia en 190 países, esta empresa creada en 1997 como una plataforma de distribución de contenidos audiovisuales de terceros, actualmente orienta gran parte de su estrategia de programación y diferenciación hacia contenidos propios bajo la estrategia de contenido original o *Netflix originals* (Heredia-Ruiz et al. 2021).

Uno de los principales diferenciadores de Netflix es su amplio catálogo de contenidos, que se renueva permanentemente con el lanzamiento de nuevas series y producciones, y que optimiza la experiencia de consumo del usuario, Aunque Netflix es considerada según perspectivas contemporáneas como Televisión Distribuida por internet, su propósito es construir y mantener un flujo televisivo permanente. Esta plataforma *streaming* ofrece un catálogo variado que el usuario puede navegar a través de internet con total libertad y control. “Los miembros pueden ver todo lo que quieran, en cualquier momento, en cualquier lugar, en cualquier pantalla conectada a Internet. Los miembros pueden reproducir, pausar y reanudar la visualización, todo sin comerciales ni compromisos (Heredia-Ruiz, Quirós-Ramírez and Quiceno-Castañeda 2021). Con la fragmentación de audiencias y la multiplicidad de pantallas la plataforma ha adoptado una estrategia expansiva para estar presentes en Smart TV, móviles y las tabletas, ordenadores, consolas de videojuegos e incluso reproductores multimedia y Blue Ray (Agustín-Lacruz and Gómez-Díaz 2022) .

En los televisores Samsung, las aplicaciones se pueden instalar a través de Smart Hub. El televisor trae algunas aplicaciones preinstaladas, aunque tiene muchas otras aplicaciones en su tienda que se deben descargar para ser usadas. Netflix es considerada entre las aplicaciones de Tizen Smart TV más interesantes disponibles en el Samsung Smart Hub. Tener la aplicación de Netflix instalada en tu televisor es prácticamente imprescindible, ya que es uno de

los servicios de en streaming más apreciado para películas y series, con una mezcla de hits de ayer y de hoy y contenido exclusivo.

1.2.2 YouTube

YouTube fue fundada en 2005, es actualmente la segunda red social con más seguidores activos. Cada día se visualizan a través de ella cinco mil millones de vídeos, cada minuto se cargan 300 horas de contenido y las personas que crean piezas audiovisuales ya ascienden a 50 millones. Como otras redes sociales con actividad prosumidora, YouTube ofrece posibilidades para que los ciudadanos puedan acceder directamente a sus canales informativos auto elaborados, sin necesidad de contar con los medios convencionales. Además, brinda la oportunidad de que los activistas políticos produzcan y difundan sus propios mensajes (Rodríguez-Breijo et al. 2018).

Se podría decir que YouTube es una especie de televisión en Internet bajo demanda, aunque incluye emisiones en directo y la posibilidad de realizar los Hangouts de Google+ en directo. De hecho, el término “televisión”, aunque no se suele utilizar en relación a YouTube, viene a cuento por su integración cada más fuerte con la televisión “tradicional” a través de las Smart tv y dispositivos externos con acceso a Internet que, conectados a un televisor tradicional permiten disfrutar de YouTube como si fuera un canal de TV más (Cedeño Cantos 2021).

YouTube es una aplicación que no debería faltar en ningún televisor, la misma no es de pago obligatorio. La versión de YouTube para Samsung Smart TV es bastante aceptable, con soporte para subtítulos y fácil de controlar con el control remoto del televisor. El contenido de YouTube no necesita presentación: encontrará de todo, desde auténticas obras maestras hasta contenido difícil de digerir. Este es el TOP de las aplicaciones de Samsung Smart TV.

1.2.3 Twitch

Twitch, la plataforma de live streaming de Amazon, es la red social que más creció el pasado año. De hecho, según sus propios números, llegó a superar el trillón de minutos vistos. Y, en la actualidad, se afirma que la plataforma de Amazon acumula casi 10 millones de canales

activos y una media diaria de 30 millones de visitantes. Pero la cifra más llamativa, aquella que está haciendo de Twitch la plataforma de streaming por excelencia, es la media de tres millones de espectadores concurrentes diarios, que aumentan hasta los 4,7 millones en su momento de máxima audiencia. La plataforma se define como "el lugar donde millones de personas se reúnen en directo a diario para charlar, interactuar y crear juntos su propio entretenimiento". Tal ha sido el crecimiento de Twitch, que ya se le compara con su máximo competidor: YouTube (Sánchez Lara 2021).

Twitch y los podcasts son fenómenos marginales y la televisión les saca 16 vueltas de ventaja", las televisiones tradicionales tienen puesto el foco de atención en la irrupción de esta plataforma. Twitch está captando a gran parte de los jóvenes y podría convertirse en su nueva televisión. Twitch y la televisión son dos mundos paralelos, pero podrían aprovechar las oportunidades que ofrece cada plataforma para impulsar su trabajo (Sánchez Lara 2021).

Sin lugar a duda el sitio más popular de streaming de juegos en la actualidad es Twitch, ya que se puede ver como otros retransmiten sus partidas en directo. Las distintas marcas de televisores más punteras que existen en la actualidad difieren bastante en instalar la app de Twitch o no. Las marcas que se conocen como claves en las Smart TV no incorporan la posibilidad de utilizar la aplicación del servicio de streaming de juegos más utilizado. Samsung dejó de lado la aplicación de Twitch en 2019 y no la ha vuelto a colocar entre las elegidas de su sistema operativo Tizen, por lo que no se puede disfrutar de ella de manera directa. Se necesita de un Chromecast o TV Box con Android para compartir la pantalla del móvil en el Smart TV.

1.2.4 teVeo

teVeo es una aplicación web que ofrece el servicio de audio y vídeo bajo demanda del Instituto Cubano de Radio y Televisión (ICRT) es decir, permite al usuario el acceso a contenidos multimedia de forma personalizada, dando la posibilidad de solicitar y visualizar los contenidos en el momento deseado (TeVeo, 2012).

Características de teVeo:

1. El contenido disponible en el sitio incluye los tipos: audio y vídeo.
2. Los archivos son organizados según su contenido, actualización y popularidad en bloques como: Recomendaciones, Lo más reciente y Lo más popular en teVeo.
3. Permite realizar búsqueda simple mostrando la información de los archivos de audio y vídeos, cuenta además con un servicio RSS, para publicar las últimas actualizaciones del sitio.
4. Cuenta con permisos limitados a la institución, sin opciones para establecer contacto con los administradores, comentar, o compartir archivos.

1.2.5 Primicia

La Plataforma de Televisión Informativa PRIMICIA surge con la intención de transmitir informaciones que integren textos, imágenes, videos y audios en forma de noticias, a través de un canal de televisión que transmita constante y automáticamente. Este producto informático se desarrolla en la Universidad de las Ciencias Informáticas desde hace varios años, esto permite la obtención de nuevas versiones cada vez más completas y atractivas para los usuarios. A diferencia de los sistemas similares que existen a nivel mundial, PRIMICIA utiliza para lograr sus propósitos tecnologías comunes y que requieren el mínimo de prestaciones especiales para la transmisión del canal y evitan generar altos costos por conceptos de equipamiento tecnológico (Cabrera 2012).

Los objetivos concretos del sistema son:

1. Desarrollar una herramienta de software que permita realizar la transmisión de un canal de televisión.
2. Desarrollar una herramienta de software que permita realizar la administración del canal a través de la web.
3. Desarrollar una herramienta de software que permita redactar las noticias que serán transmitidas, utilizando textos, imágenes, videos y audios.
4. Desarrollar una herramienta de software que permita publicar las noticias existentes en el sistema, agrupadas por bloques o temáticas.

5. Desarrollar una herramienta de software que permita integrar infocintas a la transmisión del canal.
6. Desarrollar una herramienta de software que permita gestionar los archivos multimedia que se utilizarán en la redacción de noticias.
7. Desarrollar una herramienta de software que permita asegurar la seguridad del sistema mediante la gestión de usuarios, grupos y permisos (Cabrera 2012).

En el estudio de los sistemas homólogos anteriores se identificaron como principales ventajas:

1. Netflix es un servicio de streaming por suscripción que les permite a sus miembros ver series y películas en un dispositivo con conexión a internet. En función del plan que tengas, también puedes descargar series y películas en un dispositivo con iOS, Android, o Windows 10 y verlas sin conexión a internet.
2. La Interfaz de YouTube es muy amigable y basta con tocar un par de botones para publicar un buen video. Además, las cámaras y gadgets para producir video son cada vez más portables y cómodos.
3. La plataforma Twitch con más tráfico de usuarios» Mejor implementación para smartphones y tablets. Más variedad de contenido. *Engagement* para conseguir suscriptores.
4. Para lograr sus objetivos, PRIMICIA utiliza tecnología convencional que requiere mínimas particularidades en la transmisión de canales y no incurre en altos costos en conceptos de equipamiento técnico.
5. TeVeo permite ver información sobre archivos de audio y video y realizar búsquedas simples. También hay un servicio RSS que publica las últimas actualizaciones del sitio web.

A pesar de estas ventajas las soluciones existentes, no resuelven el problema de la presente investigación, aun así, se tomaron como referencia para el diseño y la presentación de la información.

Plataforma	Software	Servicios Streaming	Integración con Smart TV Samsung
------------	----------	---------------------	----------------------------------

Netflix	Privativo	si	si
YouTube	Privativo	si	si
Twitch	Privativo	si	no
Primicia	Libre	si	no
TeVeo	Libre	si	no

Tabla 1: Características principales de las plataformas analizadas.

1.3 Caracterización de las tecnologías, herramientas y metodología

Desarrollar un software implica que sea necesario definir las tecnologías, herramientas y metodología de desarrollo que se van a emplear para lograr el objetivo planteado. A continuación, se exponen las herramientas a utilizar, la metodología de desarrollo seleccionada y un análisis de las tecnologías actuales para el consumo de contenidos desde los Smart TV Samsung.

1.3.1 Herramientas utilizadas

1.3.2 Lenguaje de Modelado Unificado 2.5

El Lenguaje Unificado de Modelado (*UML, Unified Modeling Language*) fue adoptado en noviembre de 1997 por *OMG (Object Management Group)* como una de sus especificaciones y desde entonces se ha convertido en un estándar de facto para visualizar, especificar y documentar los modelos que se crean durante la aplicación de un proceso de software. UML ha ejercido un gran impacto en la comunidad del software, tanto a nivel de desarrollo como de investigación (Hernández et al.).

1.3.3 Visual Paradigm 8.0

En la Universidad de las Ciencias Informáticas (UCI) se ha estandarizado el uso del Visual Paradigm for UML en su distribución libre como herramienta CASE para el modelado de los procesos de desarrollo de software que en ella se llevan a cabo, dado por la gran cantidad de ventajas que posee, las cuales están en concordancia con los intereses y políticas establecidas

en la institución. Entre sus principales características se encuentran que es multiplataforma, posee interoperabilidad, facilita la colaboración en equipo y brinda apoyo al ciclo de vida completo del desarrollo de software (Hernández, Peña, Valdés and Cornelio).

Para el modelado se utilizará Visual Paradigm for UML 8.0 por ser una herramienta profesional que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación. Agiliza la construcción de aplicaciones con calidad y a un menor coste. Posibilita la generación de bases de datos, transformación de diagramas de Entidad-Relación en tablas de base de datos, así como ingeniería inversa de bases de dato (Hernández, Peña, Valdés and Cornelio).

1.3.4 Samsung TV SDK

Mejor conocido como Tizen IDE o Tizen Studio, proporciona las herramientas para realizar el desarrollo de aplicaciones para televisores inteligentes Samsung basados en Tizen. Este SDK es una versión del IDE Eclipse, que contiene una interfaz de usuario intuitiva y funciones fáciles de usar para el desarrollador. Esto incluye todas las diferentes tecnologías basadas en la web, además de simuladores y emuladores de televisión Tizen, y la generación automática de los paquetes de la aplicación, facilitando la instalación y el testeado de esta aplicación sobre dispositivos reales (Strasser 2018).

El SDK de Samsung ofrece tres ventajas principales, tales como la edición de código fácil, edición visual fácil y empaquetado rápido. Esto hará que sus obras se desarrollen con rapidez y mayor facilidad.

El Samsung Smart TV SDK organiza los archivos en los proyectos, proporcionando un panel explorador de proyectos y documentos, y una navegación por pestañas para poder acceder a los documentos del proyecto con facilidad. Se pueden crear automáticamente aplicaciones en blanco, con todos los archivos que una aplicación básica necesita. Después de crear una aplicación, se puede probar con el validador de HTML y el Samsung Smart TV SDK emulador, que recrea el ambiente de la televisión en el ordenador. El Samsung Smart TV SDK permite

editar intuitivamente los aspectos visuales de las aplicaciones. El modo visual muestra una vista previa de la aplicación. Se puede hacer clic y arrastrar los componentes comunes de aplicaciones tales como botones, casillas de verificación, etc. En la ventana de edición, se puede cambiar el tamaño y organizar los componentes e inspeccionar las propiedades específicas de cada componente en el panel de Propiedades(Jaldo-Asenjo 2017).

Los proyectos desarrollados para la Smart TV de Samsung son aplicaciones basadas en Web, pero en lugar de necesitar de un servidor donde ubicar la aplicación, se procede a empaquetar el código dentro de una aplicación que se va a ejecutar en la televisión. Las aplicaciones son un tipo especial de páginas Web que se implementan sobre un navegador, el cual se ejecuta sobre la pantalla del televisor.

1.4 Lenguajes para el modelado y desarrollo

1.4.1 UML

El Lenguaje Unificado de Modelado (UML, por sus siglas en inglés) es un lenguaje de modelado visual que se usa para especificar, visualizar, construir y documentar artefactos de un sistema de software. Captura decisiones y conocimiento sobre los sistemas que se deben construir. Se usa para entender, diseñar, hojear, configurar, mantener, y controlar la información sobre tales sistemas. Está pensado para usarse con todos los métodos de desarrollo, etapas del ciclo de vida, dominios de aplicación y medios.

La formalización de los diagramas del UML permite que cada modelo de sistemas se refine, admitiendo la inclusión y la refinación de las relaciones entre los elementos, chequeando la consistencia interna de cada uno de los elementos, y verificando la interconexión entre los elementos. UML surge como una herramienta de gran aceptación cuando es necesario soportar el diseño y la implementación de una solución automatizada, que subyace en un modelo de gestión de cualquier sistema. Para ello se debe tener la documentación apropiada para su desarrollo y su mantenimiento subsiguiente o eventuales modificaciones. Lo anterior

resulta deseable y debe tenerse en cuenta en las representaciones visuales del sistema para su adecuada operación y un mejor entendimiento de los diseños(Silva et al. 2018).

1.4.2 HTML5

No es simplemente una nueva versión del lenguaje de marcación HTML, sino una agrupación de diversas especificaciones concernientes a el desarrollo web. Es decir, HTML 5 no se limita sólo a crear nuevas etiquetas, atributos y eliminar aquellas marcas que están en desuso o se utilizan inadecuadamente, sino que va mucho más allá. Pretende proporcionar una plataforma con la que desarrollar aplicaciones web más parecidas a las aplicaciones de escritorio, donde su ejecución dentro de un navegador no implique falta de recursos o facilidades para resolver las necesidades reales de los desarrolladores. Para ello se están creando unas APIs que permitan trabajar con cualquiera de los elementos de la página y realizar acciones que hasta hoy era necesario realizar por medio de tecnologías accesorias (Alvarez et al. 2017).

La base principal de las aplicaciones actuales desarrolladas para los sistemas de Tizen, el lenguaje de etiquetas más utilizados al día de hoy, que no solamente permite un desarrollo simple y flexible, sino también permite el desarrollo de multiplataforma. A partir de una serie de cambios, la misma aplicación sería capaz de utilizarse sobre móviles y relojes inteligentes que también hacen uso de Tizen OS (Strasser 2018).

1.4.3 CCS3

Un lenguaje de diseño gráfico para definir y desarrollar la forma de presentación de un documento estructurado, como puede ser nuestra base desarrollada en HTML5. CSS3 está diseñado principalmente para marcar los diferentes sectores del contenido de un documento estructurado y la forma de presentación de este, características como colores y fuentes (Strasser 2018).

La tecnología CSS está diseñada para marcar la separación del contenido de las páginas web y la forma de presentación de estas. Lo que genera múltiples beneficios, como: presentar el documento final en diferentes estilos (pantalla, voz, impresión).

1.4.4 JavaScript

JavaScript es un lenguaje de programación que se utiliza principalmente para crear páginas web dinámicas. Una página web dinámica es aquella que incorpora efectos como texto que aparece y desaparece, animaciones, acciones que se activan al pulsar botones y ventanas con mensajes de aviso al usuario. Técnicamente, JavaScript es un lenguaje de programación interpretado, por lo que no es necesario compilar los programas para ejecutarlos. En otras palabras, los programas escritos con JavaScript se pueden probar directamente en cualquier navegador sin necesidad de procesos intermedios (Pérez 2019).

Un lenguaje ligero e interpretado, orientado a objetos, conocido como el lenguaje script para páginas web, pero también usado en muchos entornos sin navegador. Utilizado para la interacción con la aplicación y la manipulación de las diferentes etiquetas de la base HTML, con el fin de poder cambiar o actualizar estas diferentes partes del archivo por individual (Strasser 2018).

1.4.5 XML

Se trata de un meta-lenguaje que permite definir etiquetas personalizadas para descripción y organización de datos. Es un estándar internacionalmente conocido para la representación de información estructurada en las aplicaciones web, permitiendo que esta información pueda ser almacenada, transmitida y procesada por muy diversos tipos de aplicaciones y dispositivos. Utilizada para la estructuración y el uso de la base de datos de nuestro proyecto. Para almacenar y procesar toda la información relacionada con las diferentes películas que deseamos mostrar en nuestra aplicación, incluido imágenes y enlaces a los diferentes tráileres (Strasser 2018).

El marcaje XML se usa para estructurar precisamente todos los elementos de los textos en los procesos de edición contemporánea de los artículos y otros tipos de documentos. Cada elemento está definido por una etiqueta. El XML consiste en poner etiquetas para indicarles a las bases de datos en dónde se encuentran los contenidos. Toda esa información no es para los lectores, sino para las computadoras que buscan y muestran la información. A partir de los textos marcados es posible extraer metadatos del artículo (Nieto Ramírez 2017).

La aplicación es una página Web que consiste en HTML, CSS, y JavaScript, y que se ejecuta en televisores Samsung. La página HTML muestra la estructura de la aplicación, el CSS se encarga del estilo, y el archivo JavaScript controla el comportamiento de la aplicación. Además de esto, para poder ejecutar la página en el televisor se necesita de un fichero de configuración “config.xml” que contiene información sobre el entorno operativo y la versión de la aplicación.

1.5 Frameworks

El desarrollo de software está marcado por la creciente tendencia de utilizar marcos de trabajo que faciliten el desarrollo de aplicaciones de escritorio o web. Un marco de trabajo se puede considerar como una aplicación genérica incompleta y configurable a la que se añaden las últimas piezas para construir una aplicación concreta. Un marco de trabajo se puede considerar como una aplicación genérica incompleta y configurable a la que se añaden las últimas piezas para construir una aplicación concreta (Díaz et al. 2020).

1.5.1 Tizen.NET

Tizen .NET permite desarrollar aplicaciones Tizen para que se ejecuten en dispositivos Samsung, incluidos televisores, dispositivos portátiles, dispositivos móviles y otros dispositivos IoT. Tizen .NET permite compilar aplicaciones .NET con Xamarin.Forms y Tizen .NET Framework. Samsung admite la plataforma Tizen .NET. Xamarin.Forms permite crear fácilmente interfaces de usuario, mientras que tizenFX API proporciona interfaces al hardware que se encuentra en los dispositivos modernos de TV, móvil, portátil e IoT (Microsoft 2022).

El SDK depende de las últimas tecnologías web como JavaScript, CSS, HTML5 y el empaquetado de widgets W3C. De hecho, Samsung ha establecido Tizen.Net, que es una nueva aplicación multiplataforma que se ha integrado con Visual Studio (Aguirre Benavides 2019).

1.5.2 Node.js

Es una plataforma de ejecución la cual permite ejecutar JavaScript de parte del servidor que al combinarse con Express.js que es un *framework* que funciona sobre Node.js permitiendo generar enrutamiento para trabajar sobre el protocolo HTTP. *Node* es un entorno de ejecución orientado a eventos asíncronos de JavaScript diseñado para construir aplicaciones de red escalables, si no hay trabajo por hacer Node.js permanece dormido hasta que haya una conexión para que se active una llamada al servicio, todo esto es posible ya que se usa JavaScript del lado de servidor (Haro et al. 2019).

NodeJS es un entorno en tiempo de ejecución multiplataforma, de código abierto, que permite trabajar con JavaScript en la capa del servidor, con arquitectura orientada a eventos y basado en el motor V8 de Google, este motor está diseñado para correr en el browser y ejecutar el código JavaScript de una manera rápida (Gómez García 2018).

1.5.4 Vue.js

Vue.js es un *framework* progresivo de JavaScript, desarrollado por Evan You, para la construcción de interfaces web interactivas. Su principal característica es que está diseñado para adaptarse incrementalmente al proyecto hasta lograr sofisticadas aplicaciones SPA al combinarse con librerías de soporte. Es un sistema de modularización con un Core que permite desarrollar componentes de interfaz de usuario a través de JavaScript y es enriquecido por un sinnúmero de librerías. Este framework se “enmarca en las arquitecturas de componentes con una gestión de modelos basada en el patrón MVVM”, permite manipular los estados de aplicación. Sus librerías se encuentran divididas y bien acotadas para cumplir funciones específicas evitando así incluir todas las excesivas funcionalidades desde un inicio, como es el caso de otros framework (Olivo Silva 2019).

1.5 Metodología de desarrollo

Una metodología es un conjunto integrado de técnicas y métodos que permite abordar de forma homogénea y abierta cada una de las actividades del ciclo de vida de un proyecto de desarrollo. Es un proceso de software detallado y completo. Definen artefactos, roles y actividades, junto con prácticas y técnicas recomendadas. Comprende los procesos a seguir sistemáticamente para idear, implementar y mantener un producto software desde que surge la necesidad del producto hasta que cumplimos el objetivo por el cual fue creado (Maida and Pacienza 2015). El principal objetivo del proceso de desarrollo es promover la calidad del software en cada una de las fases x las que se desplaza a través de una comunicación efectiva, realizando verificaciones inmediatas y llevando a cabo una buena planificación y control.

Las metodologías ágiles presentan como principal particularidad la flexibilidad, los proyectos en desarrollo son subdivididos en proyectos más pequeños, incluye una comunicación constante con el usuario, son altamente colaborativos y es mucho más adaptable a los cambios. De hecho, el cambio de requerimientos por parte del cliente es una característica especial, así como también las entregas, revisión y retroalimentación constante (Montero et al. 2018).

1.5.1 Metodología AUP-UCI

El Proceso Unificado Ágil de Scott Ambler o *Agile Unified Process (AUP)* en inglés es una versión simplificada del Proceso Unificado de *Rational (RUP)*. Este describe de una manera simple y fácil de entender la forma de desarrollar aplicaciones de software de negocio usando técnicas ágiles y conceptos que aún se mantienen validos en RUP. La Universidad de las Ciencias Informáticas (UCI) ha definido una variación de la metodología AUP en unión con el modelo CMMI-DEV v 1.3, para que pueda emplearse en los proyectos productivos de la Universidad de las Ciencias Informáticas(UCI) (Sánchez 2015).

Disciplinas de la Metodología Variación AUP-UCI

Modelado de negocio: Destinada a comprender los procesos de negocio de una organización. Se define cómo funciona el negocio que se desea informatizar para tener garantías de que el

software desarrollado va a cumplir su propósito. Para modelar el negocio se proponen las siguientes variantes: Casos de Uso del Negocio (CUN), Descripción de Proceso de Negocio (DPN) y Modelo Conceptual (MC). A partir de las variantes anteriores se condicionan cuatro escenarios para modelar el sistema en la disciplina Requisitos. La investigación se centra en la disciplina de Modelado de negocio que describe la metodología variación AUP-UCI(Sánchez 2015).

Requisitos: El esfuerzo principal en la disciplina Requisitos es desarrollar un modelo del negocio del sistema que se va a construir. Esta disciplina comprende la administración y gestión de los requisitos funcionales y no funcionales del producto. Existen tres formas de encapsular los requisitos: Casos de Uso del Sistema (CUS), Historias de usuario (HU) y Descripción de requisitos por proceso (DRP), agrupados en cuatro escenarios condicionados por el Modelado de negocio.

Análisis y diseño: Los requisitos pueden ser refinados y estructurados para conseguir una comprensión más precisa de estos, y una descripción que sea fácil de mantener y ayude a la estructuración del sistema (incluyendo su arquitectura). Además, en esta disciplina se modela el sistema y su forma (incluida su arquitectura) para que soporte todos los requisitos, incluyendo los requisitos no funcionales. Los modelos desarrollados son más formales y específicos que el de análisis.

Implementación: En la implementación, a partir de los resultados del Análisis y Diseño se construye el sistema.

Pruebas internas: Se verifica el resultado de implementación probando cada construcción, incluyendo tanto las construcciones internas como intermedias, así como las versiones finales a ser liberada. Se deben desarrollar artefactos de prueba como: diseños de de casos de prueba, listas de chequeo y de ser posible componentes de prueba ejecutables para automatizar las pruebas.

Pruebas de liberación: Diseñadas y ejecutadas por una entidad certificadora de la calidad externa, a todos los entregables de los proyectos antes de ser entregados al cliente para su aceptación.

Pruebas de Aceptación: Es la prueba final antes del despliegue del sistema. Su objetivo es verificar que el software está listo y que puede ser usado por usuarios finales para ejecutar aquellas funciones y tareas para las cuales el software fue construido (Sánchez 2015).

Descripción de los escenarios

A partir de que el Modelado de negocio propone tres variantes a utilizar en los proyectos (CUN, DPN o MC) y existen tres formas de encapsular los requisitos (CUS, HU, DRP), surgen cuatro escenarios para modelar el sistema en los proyectos, manteniendo en dos de ellos el MC, quedando de la siguiente forma:

Escenario No 1: Aplica a los proyectos que hayan evaluado el negocio a informatizar y como resultado obtengan que puedan modelar una serie de interacciones entre los trabajadores del negocio/actores del sistema (usuario), similar a una llamada y respuesta respectivamente, donde la atención se centra en cómo el usuario va a utilizar el sistema.

Escenario No 2: Aplica a los proyectos que hayan evaluado el negocio a informatizar y como resultado obtengan que no es necesario incluir las responsabilidades de las personas que ejecutan las actividades, de esta forma modelarían exclusivamente los conceptos fundamentales del negocio.

Escenario No 3: Aplica a los proyectos que hayan evaluado el negocio a informatizar y como resultado obtengan un negocio con procesos muy complejos, independientes de las personas que los manejan y ejecutan, proporcionando objetividad, solidez, y su continuidad.

Escenario No 4: Aplica a los proyectos que hayan evaluado el negocio a informatizar y como resultado obtengan un negocio muy bien definido. El cliente estará siempre acompañando al equipo de desarrollo para convenir los detalles de los requisitos y así poder implementarlos, probarlos y validarlos(Sánchez 2015).

AUP-UCI fue seleccionada debido a que se centra en equipos de desarrollo pequeños, como el presente caso que está integrado solamente por un programador de mediana experiencia.

Además, brinda una gestión de cambios ágil que hace que el equipo de desarrollo se adapte a nuevas condiciones, que en este caso los requisitos pueden cambiar a decisión del cliente. Otro punto a su favor es la forma en que se planifica el proyecto y realiza la estimación del tiempo, pues se cuenta con poco tiempo para el desarrollo de la aplicación. Como factor más determinante de esta selección es que esta es la metodología utilizada actualmente en la UCI. Finalmente, dentro de los escenarios que posee se seleccionó el escenario cuatro debido a que negocio está bien definido y el cliente siempre acompaña al equipo de desarrollo

1.6 Conclusiones Parciales

- Se identificó conceptos relacionados con el dominio del problema. El estudio de las soluciones existentes demuestra que no pueden ser utilizadas para dar solución al problema planteado, ya que sus funcionalidades responden al negocio para el que están diseñadas. Además, no son compatibles en su totalidad con los servicios que va a brindar la plataforma Picta. A pesar de esto se tomaron como referencia para el diseño y la presentación de la información.
- Las tecnologías escogidas para el desarrollo de la aplicación, fueron determinadas por las características de éstas, se determinó utilizar para la implementación de la aplicación el lenguaje de programación java scrip, HTML5 como sistema de gestión de plantilla y CCS3 como gestión de entorno.
- Se determinó hacer uso de UML como lenguaje de modelado por su estrecha integración con la metodología adoptada, AUP-UCI en su escenario 4. Como herramienta de modelado se utilizó Visual Parading v8.0 para la generación de artefactos y como herramienta IDE Visual Studio Code v4.6. Las herramientas elegidas favorecen la soberanía tecnológica que ofrece el país y el entorno de desarrollo en el que se produce este resultado.

CAPÍTULO 2: ANÁLISIS Y DISEÑO DE LA PROPUESTA DE SOLUCION.

En este capítulo se realiza una descripción del modelo de dominio y se muestran los conceptos principales y su relación. Se dan a conocer los requisitos funcionales y no funcionales. Se describe la arquitectura del sistema y se argumentan los patrones de diseño a utilizar.

2.1 Modelo del dominio

Un modelo de dominio es una representación de las clases conceptuales del mundo real, no de componentes de software. No se trata de un conjunto de diagramas que describen clases software, u objetos software con responsabilidades (García-Holgado et al. 2022).

El modelo de dominio captura los tipos más importantes de objetos en el contexto del sistema. Los objetos del dominio representan las cosas que existen o los eventos que suceden en el entorno en el que trabaja el sistema. Muchos de los objetos del dominio o clases pueden obtenerse de una especificación de requisitos o mediante la entrevista con los expertos del dominio. Se muestra el Modelo de Dominio de la aplicación web Picta en la Figura 1:

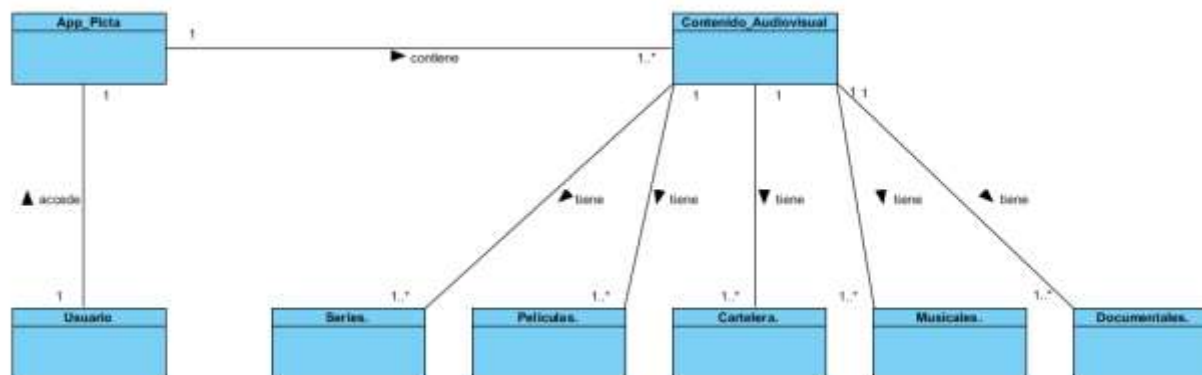


Figura 1: Modelo del dominio

2.1.2 Descripción de conceptos

- Usuario: Es el encargado buscar y reproducir materiales audiovisuales.

- Contenido_Audiovisual: Son los diferentes tipos de audiovisuales que se pueden escoger.
- Aplicación_Picta: plataforma cubana de contenidos audiovisuales.
- Musical: contiene temas musicales o cortos que tienen una canción.
- Serie: Es una obra de carácter audiovisual, manteniendo cada una de ellas una unidad y continuidad argumental o al menos temática con los episodios anteriores y posteriores. Puede constar de una o más temporadas.
- Documentales: Audiovisual que tienen como característica un carácter informativo y poseen situaciones reales.
- Películas: Espacio donde podemos encontrar el audiovisual conocido como películas cinematográficas
- Cartelera: Espacio donde se puede encontrar la lista de canales del país que tiene detalles del orden y el contenido de su transmisión.

2.2 Levantamiento de requisitos

La Ingeniería de requisitos proporciona el mecanismo apropiado para entender lo que el cliente quiere, analizar las necesidades, evaluar la factibilidad, negociar una solución razonable, especificar la solución sin ambigüedades, validar la especificación, y administrar los requisitos conforme éstos se transforman en un sistema operacional (Núñez and Muñoz).

2.2.1 Requisitos Funcionales

Los requisitos funcionales son aquellos directamente relacionados con las funciones o las reacciones que el sistema debe proporcionar (Cardozzo 2016).

A la hora de analizar un requisito es de suma importancia conocer que son caracterizados por ser especificados por escrito, posible de probar o verificar, conciso, completo, no ambiguo y consistente. Estos muestran las especificaciones de software. Los requisitos funcionales identificados fueron los siguientes:

AF. Series

RF1. Mostrar series

RF2. Seleccionar serie
RF3. Seleccionar capítulo
RF4. Reproducir capítulo
RF5. Picture in picture
RF6. Seleccionar otro video

AF. Películas

RF7. Mostrar películas
RF8. Seleccionar película
RF9. Reproducir película
RF10. Picture in picture
RF11. Seleccionar otro video

AF. Musicales

RF12. Mostrar musicales
RF13. Seleccionar musical
RF14. Reproducir musical
RF15. Picture in picture
RF16. Seleccionar otro video

AF. Documentales

RF17. Mostrar documentales
RF18. Seleccionar documental
RF19. Reproducir documental
RF20. Picture in picture
RF21. Seleccionar otro video

AF. Cartelera

RF22. Mostrar listado canales
RF23. Mostrar detalles del canal

2.2.2 Requisitos no funcionales

Los requisitos no funcionales definirán las propiedades ambientales y las restricciones de implementación relacionadas con el desempeño del producto software, deben ser considerados como parte del proceso de desarrollo de software debido a que representan características de usabilidad, flexibilidad, desempeño, operatividad y seguridad del producto solicitado (Buitrón et al. 2018).

Los requisitos no funcionales con los que debe contar el sistema son:

RNF01. Requisitos de usabilidad:

- Las interfaces gráficas implementadas por el sistema deben concebirse con un ambiente sencillo y de navegación fácil para el usuario, debido a que El televisor tiene una interfaz de usuario (IU) más simple que un dispositivo móvil, se debe tener en cuenta la experiencia del usuario al diseñar la interfaz de usuario.

RNF02. Requisitos de interfaz:

- La aplicación debe tener el logo de Picta en una de las esquinas superiores.
- Se utilizarán los colores azul oscuro y blanco.
- Diseño de la interfaz debe ser agradable e informativa para el usuario.

RNF03. Requisitos de eficiencia:

- El tiempo de demora del sistema en cada transición debe ser menor de quince (15) segundos aproximadamente.

Requisitos de accesibilidad:

- Interfaces fáciles de usar por personas que tienen poca experiencia en el manejo de dispositivos.

RNF04. Extensibilidad:

- El sistema debe ser fácilmente extensible. Se le debe poder añadir nuevas funcionalidades de una manera relativamente fácil o mejorar las ya existentes sin afectar a su funcionamiento.

RNF05. Requisitos de software:

Contar con un dispositivo Smart TV Samsung con sistema operativo Tizen OS 4.0 en adelante.

RNF06. Requisitos de hardware:

- El sistema para su correcta ejecución requiere de 4 GB o más de RAM.
- El sistema deberá disponer de 6 GB o más de espacio en disco.

2.2.3 Historias de usuario

Las historias de usuario se usan, en el contexto de la ingeniería de requisitos ágil, como una herramienta de comunicación que combina las fortalezas de ambos medios: escrito y verbal. Describen, en una o dos frases, una funcionalidad de software desde el punto de vista del usuario, con el lenguaje que éste emplearía. El foco está puesto en qué necesidades o problemas soluciona lo que se va a construir. Las historias de usuario son una herramienta que agiliza la administración de requisitos, reduciendo la cantidad de documentos formales y tiempo necesarios (Menzinsky et al. 2018).

Las HU se clasifican según:

La prioridad en el negocio:

- **Alta:** Se le otorga a las HU que resultan funcionalidades fundamentales en el desarrollo del sistema, a las que el cliente define como principales para el control integral del sistema.
- **Media:** Se le otorga a las HU que resultan para el cliente como funcionalidades a tener en cuenta, sin que estas tengan una afectación sobre el sistema que se esté desarrollando.

- **Baja:** Se le otorga a las HU que constituyen funcionalidades que sirven de ayuda al control de elementos asociados al equipo de desarrollo, a la estructura y no tienen nada que ver con el sistema en desarrollo.

El riesgo en su desarrollo:

- **Alto:** Cuando en la implementación de las HU se consideran la posible existencia de errores que conlleven a la inoperatividad del código.
- **Medio:** Cuando pueden aparecer errores en la implementación de la HU que puedan retrasar la entrega de la versión.
- **Bajo:** Cuando pueden aparecer errores que serán tratados con relativa facilidad sin que traigan perjuicios para el desarrollo del proyecto.

Las HU son representadas mediante tablas divididas por las siguientes secciones:

Número: Indica el id del requisito funcional.

Usuario: Indica que usuario utilizara este requisito.

Nombre de Historia: Nombre del requisito funcional.

Prioridad en Negocio: Representa que requisitos deben implementarse primero. La prioridad se establece como: alta, media o baja, según la importancia de este requisito para el negocio.

Riesgo en Desarrollo: Determina la complejidad de desarrollar el requisito.

Iteración Asignada: Determina en que sprint se comienza a implementar el requisito según su prioridad.

Tiempo estimado: Es el tiempo estimado que se demorará el desarrollo de la HU.

Tiempo real: Representa el tiempo que se demoró en realidad el desarrollo de la HU.

Programador Responsable: Persona encargada de implementar el requisito.

Descripción: Describe de forma breve que espera el usuario de esta funcionalidad.

Prototipo: un modelo de simulación interactiva y real de cómo funcionará una aplicación.

En la descripción de las historias de usuario se exhiben las más notables del proyecto. En este caso se analizarán las historias de usuario de los requisitos que tiene una vista (Películas).

HISTORIA DE USUARIO

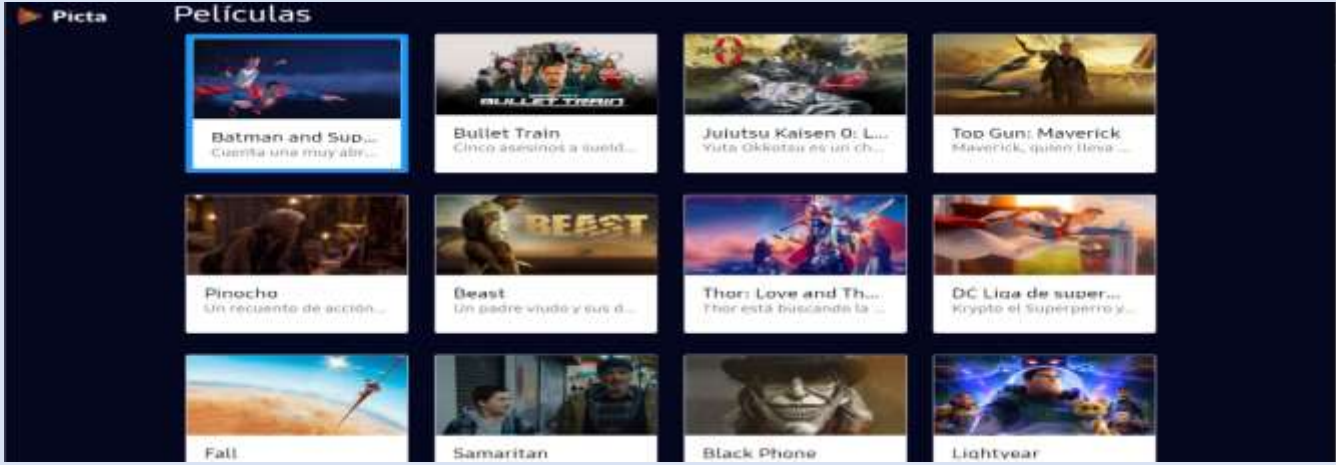
Numero: 7	Usuario: Cliente
Nombre de la historia: Mostrar películas	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Tiempo estimado: 6 días	Iteración asignada: 1
Programador responsable: Leydis Mariam Alarcón Sánchez	
Descripción: Le permite al usuario poder ver la lista de películas que se encuentran en la plataforma.	
Prototipo de interfaz:	
	

Tabla 2: Historia de usuario RF7

HISTORIA DE USUARIO	
Numero: 8	Usuario: Cliente
Nombre de la historia: Seleccionar película	
Prioridad en negocio: Alta	Riesgo en desarrollo: Medio
Tiempo estimado: 4 días	Iteración asignada: 1
Programador responsable: Leydis Mariam Alarcón Sánchez	
Descripción: Le permite al usuario elegir una película.	
Prototipo de interfaz:	

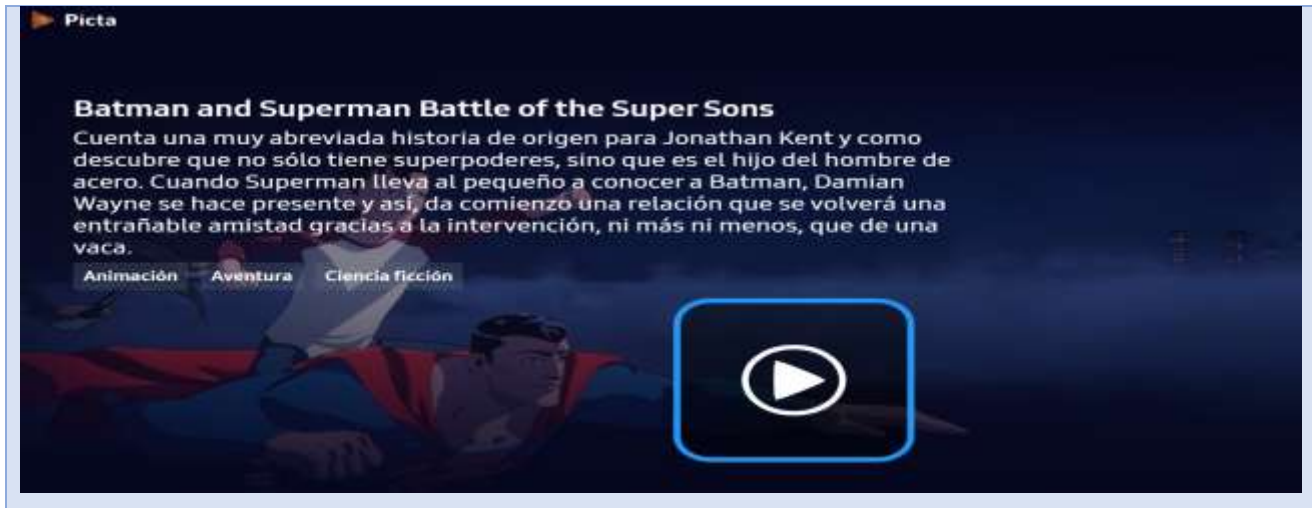


Tabla 3: Historia de usuario RF8

HISTORIA DE USUARIO	
Numero: 9	Usuario: Cliente
Nombre de la historia: Reproducir película	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Tiempo estimado: 5 días	Iteración asignada: 1
Programador responsable: Leydis Mariam Alarcón Sánchez	
Descripción: Le permite al usuario reproducir la película.	
Prototipo de interfaz:	

Tabla 4: Historia de usuario RF9

HISTORIA DE USUARIO	
Numero: 10	Usuario: Cliente
Nombre de la historia: picture in picture	
Prioridad en negocio: Media	Riesgo en desarrollo: Alta
Tiempo estimado: 3 días	Iteración asignada: 2
Programador responsable: Leydis Mariam Alarcón Sánchez	
Descripción: permite ver el video en una pequeña pantalla emergente mientras navega en la plataforma.	
Prototipo de interfaz:	
 <p>El prototipo de interfaz muestra una cuadrícula de películas con los siguientes títulos: Batman and Superman Battle of the Sun (2022), Bullet Train (2022), Jujutsu Kaisen 0: La película (2022), Top Gun: Maverick (2022), Pinocho (2022), Beast (2022). En la parte inferior derecha, se muestra una pantalla emergente con el texto "Volver a la pantalla" y un ícono de "X" para cerrar.</p>	

Tabla 5: Historia de usuario RF10

HISTORIA DE USUARIO	
Numero: 11	Usuario: Cliente
Nombre de la historia: Seleccionar otro video	
Prioridad en negocio: Media	Riesgo en desarrollo: medio

Tiempo estimado: 3 días	Iteración asignada: 3
Programador responsable: Leydis Mariam Alarcón Sánchez	
Descripción: Permite elegir otro video.	
Prototipo de interfaz:	
 <p>The screenshot shows a video player interface. On the left is a large movie poster for 'Bullet Train' featuring Brad Pitt. Below the poster are genre tags: 'Acción', 'Comedia', and 'Thriller'. On the right, there is a 'Videos recomendados' section with a 'REPRODUCCIÓN AUTOMÁTICA' toggle. The recommended videos include 'Venom' (2754 vistas), 'Animales fantásticos: Los crímenes de Grindelwald' (1715 vistas), and 'Beast'. A small video player is visible at the bottom right of the recommendations.</p>	

Tabla 6: Historia de usuario RF11

2.2.4 Plan de iteración

Luego de identificar las Historias de usuario se procede a realizar el plan de iteraciones, en el cual estarán contenidas las HU en el orden a realizar por cada iteración según su nivel de prioridad, así como la descripción y el total de semanas que durarán cada una de estas. A continuación, se muestra el plan de iteraciones:

Iteración	Descripción	Orden de la HU a implementar	Duración total(semána)
1	En la primera iteración se tiene como principal objetivo implementar las HU que poseen prioridad Alta, las cuales representan funcionalidades sensibles	Mostrar película Seleccionar película Reproducir película	2.1

	del sistema que inciden críticamente en el funcionamiento del mismo.		
2	En esta iteración se implementarán las HU que cuentan con una prioridad Media al igual que las anteriores son de gran importancia en cuanto a su incidencia en el funcionamiento del producto final, pero muchas dependen de la implementación de las Historias de usuario anteriores.	Picture in Picture	0.7
3	En esta última iteración se implementarán las HU que cuentan con una prioridad Baja, que, aunque cuenten con esta prioridad, son importantes para garantizar un funcionamiento completo de la aplicación.	Seleccionar otro video	0.4

Tabla 7: Plan de iteración

2.3 Arquitectura de software

La arquitectura del software de un programa o sistema de cómputo es la estructura o las estructuras del sistema, que incluyen los componentes del software, las propiedades visibles externamente de esos componentes y las relaciones entre ellos (Cabrera 2017). La correcta selección y aplicación de una arquitectura en el diseño de un proyecto es fundamental, pues permite organizar el desarrollo, promover la reutilización de código, facilitar el mantenimiento y modificaciones posteriores.

2.3.1 Patrón arquitectónico

Un patrón arquitectónico, expresa una organización estructural para los sistemas de software, proporciona un conjunto de subsistemas predefinidos, especifica sus responsabilidades e incluye reglas y directrices para organizar la relación entre ellos. Los patrones arquitectónicos son plantillas para arquitecturas de software concretas. La selección de un patrón arquitectónico es, por lo tanto, una decisión de diseño fundamental en el desarrollo de un sistema de software (Rodríguez Peña and Silva Rojas 2016).

Para el desarrollo de la solución propuesta se escoge el patrón arquitectónico Cliente-Servidor, el cual es un modelo para el desarrollo de sistemas de información en el que las transacciones se dividen en procesos independientes que cooperan entre sí para intercambiar información, servicios o recursos. Se denomina cliente al proceso que inicia el diálogo o solicita los recursos y servidor al proceso que responde a las solicitudes (Morera and Polini 2010).



Figura 2: Arquitectura Cliente-Servidor.

- Cliente: Programa ejecutable que participa activamente en el establecimiento de las conexiones. Envía una petición al servidor y se queda esperando por una respuesta. Su tiempo de vida es finito una vez que son servidas sus solicitudes, termina el trabajo.
- Servidor: Es un programa que ofrece un servicio que se puede obtener en una red. Acepta la petición desde la red, realiza el servicio y devuelve el resultado al solicitante.

Al ser posible implantarlo como aplicaciones de programas, puede ejecutarse en cualquier sistema donde exista TCP/IP y junto con otros programas de aplicación. El servidor comienza su ejecución antes de comenzar la interacción con el cliente (Lizama et al. 2016).

Para implementar la funcionalidad que se requiere se ha pensado en desarrollar un entorno basado en un esquema cliente-servidor, donde el cliente es la aplicación instalada en la TV y el servidor una interfaz (en forma de servicios web, por ejemplo) que proporciona los datos requeridos a la aplicación cliente.

2.4 Modelo de diseño

El modelo de diseño es un modelo de objetos que describe la realización física de los casos de uso centrándose en cómo los requisitos funcionales y no funcionales, junto con otras restricciones relacionadas con el entorno de implementación, tienen impacto en el sistema (Larman 2005).

2.4.1 Diagrama de clases del diseño

Un diagrama de clases sirve para visualizar las relaciones entre las clases que involucran el sistema, los cuales responden al cumplimiento de los requisitos funcionales. Los diagramas de clases del diseño son subproductos del modelo de diseño proporcionando una perspectiva estática que representa el diseño estructural del sistema, mostrando un conjunto de clases y sus atributos (Larman 2005).

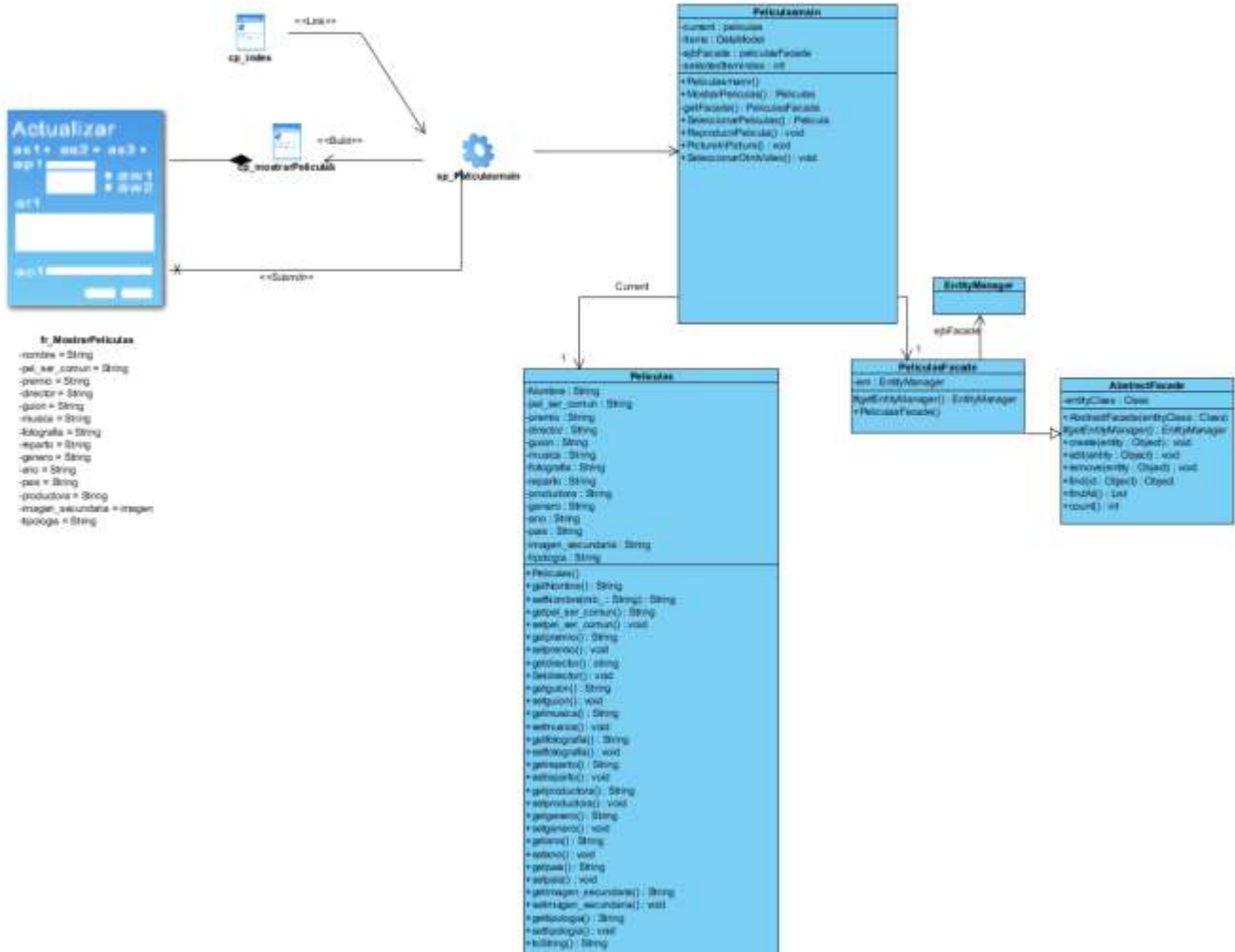


Figura 4: Diagrama de clases del diseño Películas.

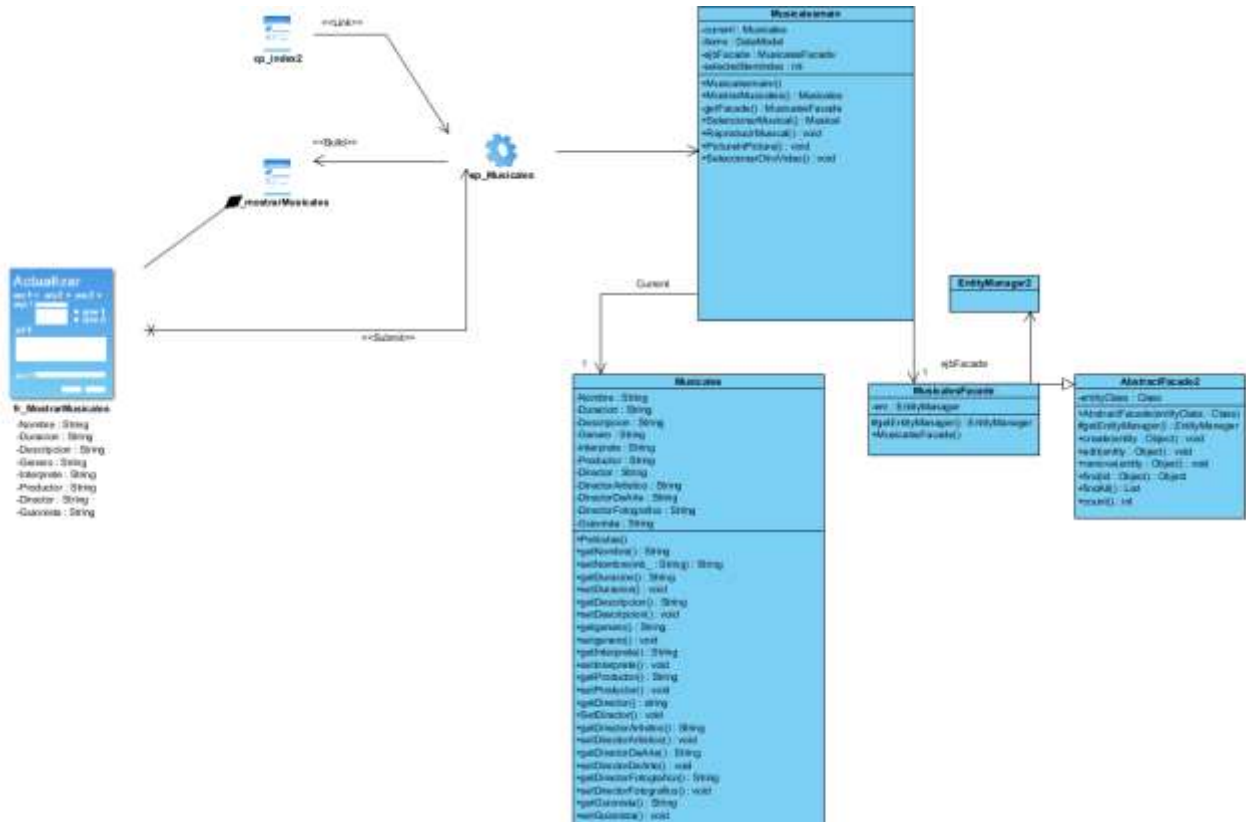


Figura 5: Diagrama de clases del diseño Musicales.

2.5 Patrones de diseño

Se puede decir que un patrón de diseño es la solución a un problema de diseño, el cual debe haber comprobado su efectividad resolviendo problemas similares en el pasado, también tiene que ser reutilizable, por lo que se deben poder usar para resolver problemas parecidos en contextos diferentes (Blancarte 2016). Existen clasificaciones para estos patrones entre los que se encuentran los Patrones Generales de Software para Asignar Responsabilidades y los Patrones del Grupo de los Cuatro. A continuación, se presentan los patrones de diseño que son utilizados para el desarrollo de la solución:

Patrones Generales de Software para Asignar Responsabilidades

Los patrones generales de software para asignar responsabilidades (por sus siglas en inglés GRASP) permiten asignar responsabilidades a objetos de forma tal que estos conozcan el rol que les corresponde (Larman 2005).

- Alta cohesión: Es un principio evaluativo que aplica un diseñador mientras evalúa todas las decisiones de diseño. Indica la relación que existe entre los elementos de un mismo módulo. Es la medida de la relación funcional de los elementos de un módulo. El objetivo es organizar estos elementos de manera que los que tengan una mayor relación a la hora de realizar una tarea pertenezcan al mismo módulo, y los elementos no relacionados, se encuentren en módulos separados.
- Bajo acoplamiento: Impulsa la asignación de responsabilidades de manera que su localización no incremente el acoplamiento hasta un nivel que nos lleve a los resultados negativos que puede producir un acoplamiento alto. Es el grado de interdependencia entre los módulos. Un buen diseño se caracteriza por un acoplamiento mínimo, es decir, unos módulos tan independientes los unos de los otros como sea posible.
- Controlador: Proporciona guías acerca de las opciones generalmente aceptadas y adecuadas para manejar eventos. Es conveniente utilizar la misma clase controlador para todos los eventos del sistema de un caso de uso, de manera que es posible manejar la información acerca del estado del caso de uso en el controlador.



Figura 6: Patrón de diseño controlador

- Creador: Guía la asignación de responsabilidades relacionadas con la creación de objetos, una tarea muy común. La intención básica del patrón es encontrar un creador que necesite conectarse al objeto creado en alguna situación.

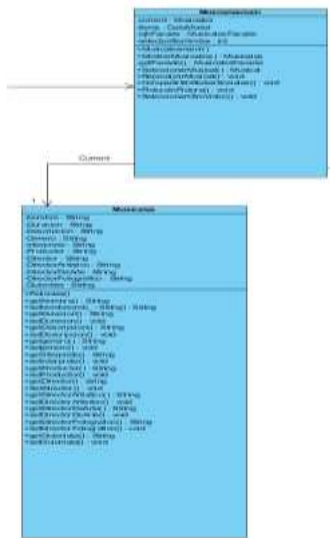


Figura 7: Patrón de diseño creador

- Experto: Se utiliza con frecuencia en la asignación de responsabilidades; es un principio de guía básico que se utiliza continuamente en el diseño de objetos. Expresa la intuición común de que los objetos hacen las cosas relacionadas con la información que tienen (Larman 2005).

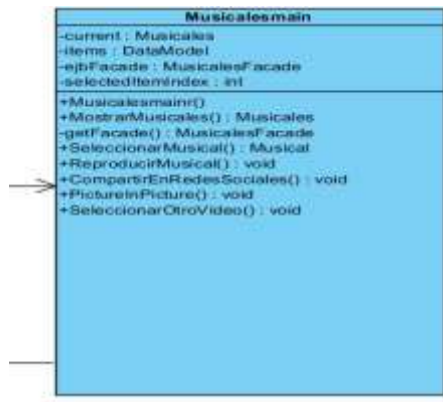


Figura 8: Patrón de diseño experto

Patrones Grupo de los Cuatro (GOF)

Los patrones del Grupo de los Cuatro (por sus siglas en inglés GOF14) se dividen en tres categorías: patrones de creación, patrones estructurales y patrones de comportamiento (Larman 2005). A continuación, se describen los patrones que se evidencian en la solución:

Adaptador (del inglés Adapter): convierte la interfaz de una clase en otra distinta que es la que esperan los clientes. Permite que cooperen clases que de otra manera no podrían por tener interfaces incompatibles.

Conclusiones Parciales del capítulo

- La elaboración del modelo de dominio permitió un mejor entendimiento de los conceptos principales en el entorno del consumo de contenidos audiovisuales desde el Smart TV con sistema operativo Tizen.
- El sistema desarrollado cumple con todos los requisitos funcionales y no funcionales de la aplicación Picta, donde las historias de usuario permitieron una detallada descripción del comportamiento de cada requisito.
- Con el propósito de representar la organización de la aplicación se desarrollaron los diagramas de clases; mostrando la relación entre clases, atributos e interfaces.
- El uso del patrón arquitectónico Cliente-Servidor, en conjunto con el uso de patrones de diseño GRASP Y GOF contribuyeron a una adecuada organización de la aplicación.

CAPÍTULO 3: IMPLEMENTACIÓN Y VALIDACIÓN DE LA SOLUCIÓN PROPUESTA

En el presente capítulo se hace un análisis de los resultados obtenidos durante el desarrollo de la solución propuesta. Se realiza el diagrama de componentes para describir la organización del sistema y sus relaciones, así como el modelo de despliegue. Además, se especifican las diferentes métricas para la validación de los requisitos y el diseño.

3.1 Modelo de Implementación

El modelo de implementación describe cómo los elementos del modelo del diseño se implementan en términos de componentes, como ficheros de código fuente ejecutables. Este modelo refiere también cómo se organizan los componentes de acuerdo con los mecanismos de estructuración (Larman 2005).

3.1.1 Diagrama de componentes

Un diagrama de componentes muestra dependencias entre los componentes, que no son más que una unidad física de implementación con interfaces bien definidas, pensada para ser utilizada como parte reemplazable de un sistema. Puede mostrar un sistema configurado, con la selección de componentes usados para construirlo o un conjunto de componentes disponibles (una biblioteca de componentes) con sus dependencias (Rumbaugh et al. 2007).

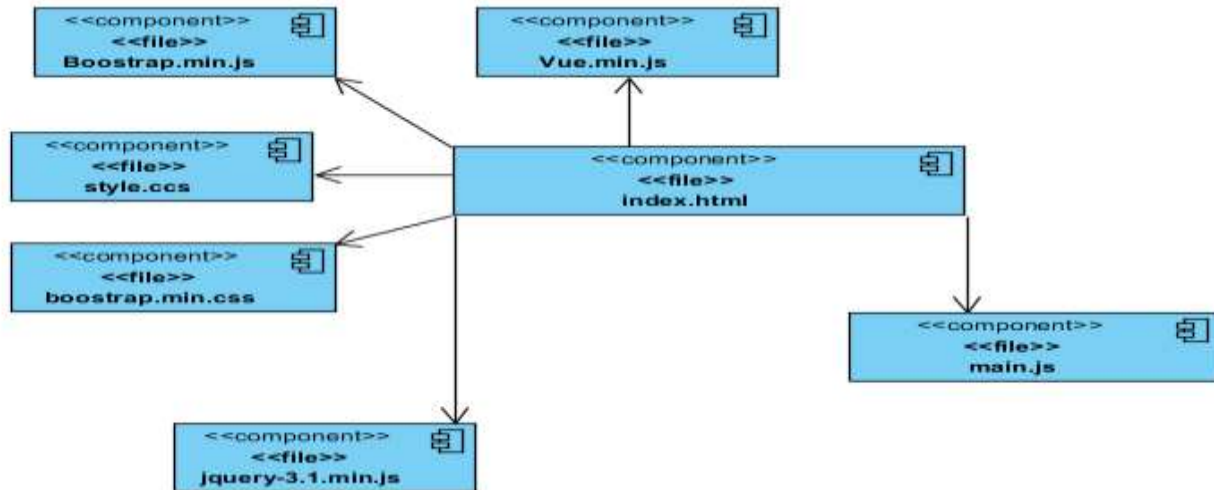


Figura 9: Diagrama de Componentes.

A continuación, se muestra el diagrama de componentes donde se aprecia la división del sistema en componentes y las dependencias que se generan entre los mismos. Primeramente, se presenta una plantilla Index.html la cual utiliza Bootstrap.min.js como framework de javascript para generar validaciones, style.css y Bootstrap.min.css como gestor de entorno u JQuery.js y Vue.min.js como librería de javascript. Esta index se relaciona con un archivo main.js el cual contiene toda la lógica de negocio de la propuesta de solución.

3.2 Diagrama de despliegue:

El Diagrama de despliegue es un diagrama estructurado que muestra la arquitectura del sistema desde el punto de vista del despliegue (distribución) de los artefactos del software en los destinos de despliegue. Los artefactos representan elementos concretos en el mundo físico que son el resultado de un proceso de desarrollo. Ejemplos de artefactos son archivos ejecutables, bibliotecas, archivos, esquemas de bases de datos, archivos de configuración entre otros (Sarmiento 2013).

Descripción de los nodos físicos del sistema:

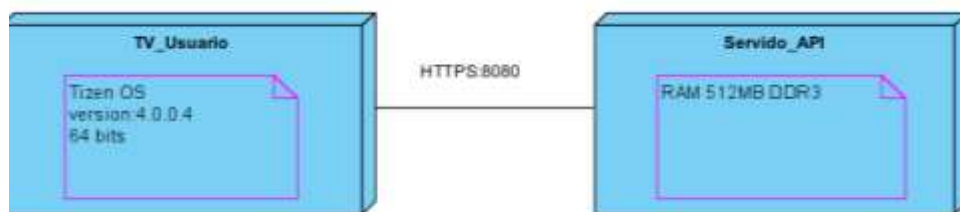


Figura 10: Diagrama de despliegue.

TV_Usuario: donde se encuentra instalada la aplicación.

Servidor_API: Aplicación que se comunica con la plataforma Picta y consume los servicios que brinda esta.

3.3 Métodos de pruebas

La prueba de caja negra se refiere a las pruebas que se llevan a cabo en la interfaz del software. Una prueba de caja negra examina algunos aspectos fundamentales de un sistema con poca preocupación por la estructura lógica interna del software. La prueba de caja blanca del software se basa en el examen cercano de los detalles de procedimiento. Las rutas lógicas a través del software y las colaboraciones entre componentes se ponen a prueba al revisar conjuntos específicos de condiciones y/o bucles (Pressman 2007).

3.3.1 Pruebas unitarias

Las pruebas unitarias son la forma de comprobar el correcto funcionamiento de una unidad de código y hoy en día se convirtieron en una parte fundamental del desarrollo de software, ayudando a elevar la calidad del producto con beneficios como el de prevenir errores en etapas tempranas del desarrollo, legibilidad de código, confiabilidad en el producto, entre otros ahorrando tiempo y dinero en los proyectos (Bedoya Alzate 2021).

3.3.2 Pruebas de Caja Blanca

La prueba de caja blanca, en ocasiones llamada prueba de caja de vidrio, es una filosofía de diseño de casos de prueba que usa la estructura de control descrita como parte del diseño a nivel de componentes para derivar casos de prueba (Pressman 2007).

Al usar los métodos de prueba de caja blanca, puede derivar casos de prueba que:

- Garanticen que todas las rutas independientes dentro de un módulo se revisaron al menos una vez.
- Revisen todas las decisiones lógicas en sus lados verdadero y falso.
- Ejecuten todos los bucles en sus fronteras y dentro de sus fronteras operativas.
- Revisen estructuras de datos internas para garantizar su validez.

Para la generación de casos de prueba de unidad, se decidió utilizar la técnica de camino básico:

La técnica del camino básico, consiste en verificar el código de nuestros sistemas de manera que comprobemos que todo funciona correctamente, es decir, se debe verificar que todas las instrucciones del programa se ejecutan por lo menos una vez (MOUSE 2017).

Los pasos para desarrollar la prueba del camino básico son:

- Confeccionar el grafo de flujo
- Calcular la complejidad ciclomática
- Determinar el conjunto básico de caminos independientes Esta métrica se calcula sobre un grafo y se puede realizar mediante tres formas distintas:

1. $V(G) = R.$

2. $V(G) = E - N + 2$

3. $V(G) = P + 1$

Conociendo que:

- G: Grafo de flujo (grafo)
- R: El número de regiones contribuye a estimar el valor de la complejidad ciclomática.
- E: Número de aristas

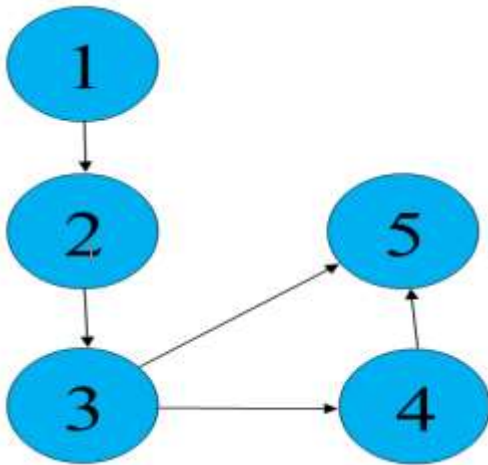
- $V(G)$: Complejidad ciclomática
- N : Número de nodos del grafo
- P : Número de nodos predicados incluidos en el grafo. Una vez calculada la complejidad ciclomática, el valor obtenido representa el límite superior de pruebas que deberán aplicarse.

Se analizará el método `loadDocumentales ()` que está contenido en el archivo `main.js` el cual tiene como función principal cargar todos los documentales desde la API.

```
hideLoading() {
  this.isLoading = false;
  if (this.columnSelect == 0) {
    this.cards[this.indexSelect].loading = false;
    this.cards[this.indexSelect].select = false
    this.indexSelect = 0;
    this.cards[this.indexSelect].select = true
  }
  this.options[this.indexOption].select = false
  this.options[this.indexOption].loading = false;
},
```

Figura 11: `hideLoading`.

Cuando se genera el diagrama de flujo relacionado con el procesamiento anterior, se determina la complejidad ciclomática. Esta es una métrica de software útil porque proporciona una medida cuantitativa de la complejidad lógica del programa. El valor calculado como complejidad del ciclo determina el número de rutas independientes en el conjunto subyacente del programa y da un límite superior al número de pruebas que se deben realizar.



Cálculo de la complejidad ciclomática:

$$V(g) = (5 - 5) + 2 = 2 \text{ (I)}$$

$$V(g) = 1 + 1 = 2 \text{ (II)}$$

$$V(g) = 2 \text{ (III)}$$

Como la fórmula $V(g)(II) = 2$ existen al menos dos caminos por los cuales se puede llegar a un nodo final.

Camino básico 1: 1, 2, 5.

Camino básico 2: 1, 2, 3, 4, 5.

CASO DE PRUEBA CAMINO BÁSICO #1	
Descripción:	Este método se encarga de comprobar si los datos han sido cargados y cambia los valores de la interfaz para adaptarse si no hay datos de carga.
Condición de ejecución:	Siempre se ejecuta una vez sea llamado por otro método de carga.
Entrada:	No tiene datos de entrada.
Resultados esperados:	Cambios en la interfaz gráfica.
Resultado:	Satisfactorio

Tabla 8: Prueba de Camino Básico #1

CASO DE PRUEBA CAMINO BÁSICO #2

Descripción:	Este método se encarga de comprobar si los datos han sido cargados y cambia los valores de la interfaz para adaptarse si no hay datos de carga.
Condición de ejecución:	Siempre se ejecuta una vez sea llamado por otro método de carga.
Entrada:	No tiene datos de entrada.
Resultados esperados:	Cambios en la interfaz gráfica.
Resultado:	Satisfactorio

Tabla 9: Prueba de Camino Básico #2

Resultados:

Se realizaron un total de 10 pruebas, de las cuales 9 de ellas fueron satisfactorias representando un 90% de las mismas, una de ellas fue no satisfactorias para un 10%, estas fueron mitigadas al realizar la segunda iteración de las pruebas.



Figura 12: Resultados de las Pruebas Unitarias.

3.3.3 Pruebas de aceptación

La prueba de aceptación es generalmente desarrollada y ejecutada por el cliente en conjunto con el equipo de desarrollo. Esta tiene como objetivo determinar cómo el sistema satisface sus criterios de aceptación, validando los requisitos identificados para el desarrollo del software, incluyendo la documentación y procesos de negocio. Está considerada como la fase final del proceso, para crear un producto confiable y apropiado para su uso (Training 2017).

3.3.1 Pruebas de Caja Negra

Las pruebas de caja negra, también llamadas pruebas de comportamiento, se enfocan en los requerimientos funcionales del software; es decir, las técnicas de prueba de caja negra le permiten derivar conjuntos de condiciones de entrada que revisarán por completo todos los requerimientos funcionales para un programa (Pressman 2007).

Las pruebas de caja negra intentan encontrar errores en las categorías siguientes:

- Funciones incorrectas o faltantes.
- Errores de interfaz.
- Errores en las estructuras de datos o en el acceso a bases de datos externas.
- Errores de comportamiento o rendimiento.
- Errores de inicialización y terminación.

Técnica de prueba: Partición equivalente

Esta técnica divide el dominio de entrada de un programa en clases de datos a partir de las cuales pueden derivarse casos de prueba. El método se esfuerza por definir un caso de prueba que descubra ciertas clases de errores, reduciendo así el número total de casos de prueba que deben desarrollarse. Una clase de equivalencia representa un conjunto de estados válidos y no válidos para las condiciones de entrada (SCRIBD. 2018).

CASO DE PRUEBA DE ACEPTACIÓN

Código:	Historia de Usuario: 7
CP-01	

Nombre: Mostrar películas

Descripción: Le permite al usuario poder ver la lista de películas que se encuentran en la plataforma.			
<i>Acción a probar:</i>	<i>Datos de entrada:</i>		<i>Resultados esperados:</i>
Mostrar películas	132,1,2000, Cuba, https://www.picta.cu/imagen/img_8m9oFp7.jpeg , 1		<ol style="list-style-type: none"> 1. Se Verifican los datos de entrada. 2. Se muestra la película.
<i>Evaluación de la prueba:</i> Satisfactoria			

Tabla 10: Caso de Prueba de Aceptación #1

CASO DE PRUEBA DE ACEPTACIÓN			
Código:	Historia de Usuario:8		
CP-02			
<i>Nombre:</i> Seleccionar películas			
Descripción: Le permite al usuario elegir una película			
<i>Acción a probar:</i>	<i>Datos de entrada:</i>		<i>Resultados esperados:</i>
Seleccionar películas	132,1,2000, Cuba, https://www.picta.cu/imagen/img_8m9oFp7.jpeg , 1		<ol style="list-style-type: none"> 1. Se Verifican los datos de entrada. 2. Se selecciona la película.

<i>Evaluación de la prueba:</i> Satisfactoria

Tabla 11: Caso de Prueba de Aceptación #2

CASO DE PRUEBA DE ACEPTACIÓN			
Código: CP-03	Historia de Usuario:9		
<i>Nombre:</i> Reproducir películas			
Descripción: Le permite al usuario reproducir la película.			
<i>Acción a probar:</i>	<i>Datos de entrada:</i>	<i>Resultados esperados:</i>	
Reproducir películas	132,1,2000, Cuba, https://www.picta.cu/imagen/img_8m9oFp7.jpeg , 1	1.	Se Verifican los datos de entrada.
		2.	Se reproduce la película.
<i>Evaluación de la prueba:</i> Satisfactoria			

Tabla 12: Caso de Prueba de Aceptación #3

CASO DE PRUEBA DE ACEPTACIÓN			
Código: CP-05	Historia de Usuario:10		
<i>Nombre:</i> Picture in picture			
Descripción: permite ver el video en una pequeña pantalla emergente mientras navega en la plataforma.			
<i>Acción a probar:</i>	<i>Datos de entrada:</i>	<i>Resultados esperados:</i>	

Picture in picture	132,1,2000, Cuba, https://www.picta.cu/imagen/img_8m9oFp7.jpeg , 1	1.	Se Verifican los datos de entrada.
		2.	Se reproduce el video en la ventana emergente.

Evaluación de la prueba: 2 No Satisfactoria

Tabla 13: Caso de Prueba de Aceptación #4

CASO DE PRUEBA DE ACEPTACIÓN			
Código:	Historia de Usuario:13 CP-06		
Nombre:	Seleccionar otro video		
Descripción:	Permite elegir otro video.		
<i>Acción a probar:</i>	<i>Datos de entrada:</i>	<i>Resultados esperados:</i>	
Seleccionar otro video	726,2,2019, Corea, https://www.picta.cu/imagen/img_EAMJ22q.jpeg , 1	1.	Se Verifican los datos de entrada.
		2.	Se selecciona otro video
<i>Evaluación de la prueba: Satisfactoria</i>			

Tabla 14: Caso de Prueba de Aceptación #5

Resultados:

Se realizaron un total de 20 casos de pruebas de caja negra (pruebas de aceptación), de ellos 4 tuvieron resultados no satisfactorios, lo cual representa el 20% del total de casos de prueba de caja negra realizados. Mientras los 16 casos de prueba restantes tuvieron resultados satisfactorios para un 80% del total. Los errores detectados a través de la aplicación de los casos de pruebas fueron mitigados después de 2 iteraciones de prueba.



Figura 13: Resultados de las Pruebas de Aceptación.

3.3.4 Pruebas de rendimiento

Se necesita que el rendimiento de la aplicación sea el esperado, que esté disponible cuando se necesite, que soporte la cantidad de usuarios esperada y que cuando se encuentre bajo condiciones de uso extremas responda sin ningún contratiempo, son las cualidades que todo cliente desea en su software, llegando al punto en el que esto se convierte en una necesidad que debe ser resuelta por el equipo encargado de desarrollar el software. Para esto, se cuenta con un conjunto de pruebas agrupadas dentro de la categoría llamada *performance* o rendimiento (Pilicita Garrido et al. 2021).

Las pruebas de *performance* pueden clasificarse en varios grupos, entre los cuales podemos encontrar: pruebas de carga, que sirven para evaluar el comportamiento de un aplicativo al simular un número de usuarios concurrentes según los esperados en la vida real durante cierto período de tiempo. Pruebas de capacidad, donde se busca el punto de quiebre del producto, determinando la cantidad máxima de usuarios que puede soportar el sistema sin que se vea afectado. Las pruebas de estrés, donde se chequea la robustez y confiabilidad del producto desarrollado saturando el aplicativo hasta puntos extremos para facilitar la configuración de alarmas cuando se llegue a algunos límites (Pilicita Garrido, Borja López and Gutiérrez Constante 2021).

A la aplicación se le realizó una prueba de rendimiento de tipo carga y estrés. Se utilizó la herramienta Apache JMeter en su versión 5.5, diseñando pruebas de cargas de comportamientos funcionales y la medición del rendimiento. El ambiente de prueba está conformado por:

Hardware de prueba (PC Emulador de Tizen):

1-Sistema Operativo Windows 11 Home

2-Microprocesador: 11th Gen Intel(R) Core (TM) i5-1135G7 @ 2.40GHz 2.40 GHz

3-Memoria RAM: 8 GB

4-Disco Solido: 250 GB

5-Tarjeta de video: Intel(R) Iris(R) Xe Graphics

Software instalado en la PC:

1-Tipo de servidor web: Apache 5.5.

2-Plataforma: SO, Windows.

3-Tizen Studio 4.6

Luego de haber definido el hardware se configuran los parámetros del Apache JMeter logrando un total de 20, 40 y 100 usuarios conectados concurrentemente. En la tabla 15 se puede observar los resultados obtenidos por el sistema.

Parámetros que componen la tabla para una mejor interpretación:

- Usuarios: Total de usuarios.
- Muestras: El número de peticiones.
- Min: El mínimo tiempo transcurrido en mili-segundos para las muestras de la URL dada.
- Max: El máximo tiempo transcurrido en un mili-segundos para las muestras de la URL dada.
- % Error: Porcentaje de las peticiones con errores.
- Rendimiento: Rendimiento medido en base a peticiones por segundo/minuto/hora.
- Kb/s Recibidos: Rendimiento medido en Kbytes por segundos.

Usuarios	Muestra	Media	Min	Max	Línea 90%	% Error	Rend	Kb/s
20	50	2029	195	5767	1845	0.00%	2.6/sec	18.1
40	100	3012	109	6527	2984	0.00%	3.0/sec	15.8
100	200	3427	74	6792	2081	0.00%	3.8/sec	97.2

Tabla 15: Resultados de la Prueba de Carga y Estrés.

Resultados:

El informe de Apache JMeter arrojó que la aplicación es capaz de responder a 50 peticiones de 20 usuarios conectados simultáneamente en un tiempo promedio de 2029 milisegundos con un error de 0.00%, a 100 peticiones de 40 usuarios conectados simultáneamente en un

tiempo promedio de 3012 milisegundos con un error de 0.00% y 200 peticiones de 100 usuarios conectados simultáneamente en un tiempo promedio de 3427 milisegundos con un error de 0.00% esto evidencia que el sistema puede responder correctamente a las peticiones realizadas.

Conclusiones del capítulo

- En el modelo de implementación, la realización del diagrama de componentes permitió comprender mejor las dependencias en cuanto a librerías, apreciándose la interacción entre los componentes. El diagrama de despliegue ayudó a entender mejor la relación entre los componentes físicos que intervienen en el sistema final.
- Se realizaron las pruebas de caja negra, usando la variante de partición equivalente, donde quedaron definidos los diseños de casos de pruebas para comprobar las funcionalidades del sistema además de verificar las entradas y respuestas del mismo cumpliendo todas con los resultados esperados.
- Se aplicó el Procedimiento de Pruebas Caja Blanca usando la Técnica del Camino Básico, con el objetivo de intentar descubrir la presencia de errores, mejorar propiedades como la seguridad y el uso eficiente del sistema.
- Se realizaron pruebas de rendimiento con el objetivo de determinar el rendimiento del sistema bajo una carga de trabajo definida utilizando diferentes tipos de pruebas tales como pruebas de carga y de estrés.

CONCLUSIONES FINALES

El estudio realizado de los principales conceptos relacionados con el objeto de estudio de la investigación permitió elaborar el marco teórico de la misma, teniendo como resultado la propuesta de solución de la problemática planteada.

Se desarrolló la aplicación web Picta para el Sistema Operativo Tizen de Smart TV Samsung con las especificaciones de su SDK.

Las herramientas y tecnologías seleccionadas permitieron la obtención de una amplia documentación, el desarrollo de una solución siguiendo estándares utilizados internacionalmente, además de sentar las bases para futuras versiones.

El proceso de desarrollo de la solución propuesta fue guiado por AUP-UCI, la metodología de desarrollo seleccionada, quedando documentada cada etapa del ciclo de vida. Los artefactos generados servirán como base de futuras actualizaciones de la aplicación.

La realización de pruebas de Integración, Caja Negra, Rendimiento y Aceptación, Caja Blanca demostró el correcto funcionamiento de la aplicación Picta para Smart TV Samsung con sistema operativo Tizen.

Las encuestas aplicadas a los especialistas corroboran que la aplicación propuesta en este trabajo de diploma cumple con los objetivos propuestos, es decir, garantizar la accesibilidad y mejorar la experiencia de usuario.

RECOMENDACIONES

A partir de los resultados de este trabajo de diploma surgieron las siguientes recomendaciones:

- Incrementar el número de funcionalidades a la aplicación Picta, ya que la plataforma brinda más funcionalidades que las implementadas en la solución.
- Agregar funcionalidades a la solución enfocadas a la interacción con las redes sociales: Facebook, Twitter e Instagram.

REFERENCIAS BIBLIOGRÁFICAS

- AGUIRRE BENAVIDES, C. D. Estudio de la integración de YII2 y REACT. JS para la elaboración de un sistema de tipificación para la cruz roja ecuatoriana junta provincial de Imbabura. 2019.
- AGUSTÍN-LACRUZ, C. AND R. GÓMEZ-DÍAZ. En el laberinto de las plataformas VOD: Un estudio comparativo de Netflix, Amazon Prime Video, HBO, Movistar+ Lite, Filmin y Disney+. 2022.
- ALVAREZ, M., D. GALEANO AND J. J. R. D. H. E. C. E. S. S. G. T.-G. M.-L.-H. P. PEÑA Manual del lenguaje de marcación de HTML5 2017.
- ANTONIO, R. M. F. DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICA 2017.
- BEDOYA ALZATE, E. Implementación de pruebas unitarias 2021.
- BLANCARTE, O. J. M., MÉXICO DF Introducción a los Patrones de Diseño 2016.
- BUITRÓN, S. L., B. L. FLORES-RIOS AND F. J. J. I. R. C. D. I. PINO Elicitación de requisitos no funcionales basada en la gestión de conocimiento de los stakeholders 2018, 26(1), 142-156.
- CABRERA, G. A. H. J. T. I. Y. A. Arquitectura de software para la construcción de un sistema de cuadro de mando integral como herramienta de inteligencia de negocios 2017, 5(2), 143-152.
- CABRERA, L. D. PLATAFORMA DE TELEVISIÓN INFORMATIVA PRIMICIA V1. 7. Universidad de las Ciencias Informáticas, 2012.
- CARDOZZO, D. R. *Desarrollo de Software: requisitos, estimaciones y análisis*. Edtion ed.: IT Campus Academy, 2016. ISBN 1530088615.
- CARPINTEYRO, B. B. Servicios Basados en Localización y Personalización de la Televisión Inteligente (Smart TV): Estrategias y Aplicaciones-Edición Única 2012.
- CEDEÑO CANTOS, N. F. Análisis del periodismo deportivo digital en la transmisión de contenidos en el medio social youtube, año 2021. BABAHOYO: UTB, 2021, 2021.
- DÍAZ, D. C., L. E. R. GÓMEZ, L. A. FLORES, G. L. KOW, et al. SOLUCIÓN A LOS PROBLEMAS FRECUENTES CON LOS MARCO DE TRABAJO PHP SOLUTION TO THE FREQUENT PROBLEMS WITH THE FRAMEWORK PHP 2020.
- GARCÍA-HOLGADO, A., A. VÁZQUEZ-INGELMO AND F. GARCÍA-PEÑALVO Modelo de Dominio 2022.
- GÓMEZ GARCÍA, D. E. Desarrollo del sistema de requisiciones para la empresa hidroeléctrica Abanico SA Aplicando el entorno de programación Node. js. Escuela Superior Politécnica de Chimborazo, 2018.
- GONZÁLEZ, A. Tecnología Streaming 2017.
- GUTIÉRREZ LOZANO, J. F. AND A. J. Á. R. I. D. C. CUARTERO NARANJO, 50, 159-175. El auge de Twitch: nuevas ofertas audiovisuales y cambios del consumo televisivo entre la audiencia juvenil 2020.
- HARO, E., T. GUARDA, A. O. Z. PEÑAHERRERA AND G. N. J. R. I. D. S. E. T. D. I. QUIÑA Desarrollo backend para aplicaciones web, servicios web restful: Node. js vs spring boot 2019, (E17), 309-321.
- HEREDIA-RUIZ, V., A. C. QUIRÓS-RAMÍREZ AND B. E. QUICENO-CASTAÑEDA Netflix: Catálogo de contenido y flujo televisivo en tiempos de big data. Revista de Comunicacion, 2021, 20(1), 117-136.
- HERNÁNDEZ, L. R. B., D. M. PEÑA, O. R. VALDÉS AND O. M. CORNELIO Extensión de la herramienta Visual Paradigm for UML para la evaluación y corrección de Diagramas de Casos de Uso Plugin of Visual Paradigm for UML tool for evaluation and correction of Use Case Diagram.
- JALDO-ASENJO, J. Ampliación tv-learning: aprendizaje a distancia con smart tvs 2017.

- LARMAN, C. *Applying UML and patterns: an introduction to object-oriented analysis and design and iterative development*. Edtion ed.: Pearson Education India, 2005. ISBN 8177589792.
- LIZAMA, O., G. KINDLEY, J. J. MORALES AND A. J. U. T. F. S. M. GONZALES *Redes de Computadores: Arquitectura Cliente-Servidor* 2016, 1-8.
- LLANES, M. S. J. P. *El impacto de los servicios de streaming* 2021, (1), 89-99.
- LÓPEZ CHIU, C. A. AND F. Á. VEGA LAYME *Desarrollo de aplicaciones para SMART TV* 2019.
- LÓPEZ, S. R. J. R. Y. P. *Los Contenidos Audiovisuales En Internet Y Su Impacto En La Televisión| Audiovisual content on the Internet and its impact on Television* 2013, 17(2_83), 333-341.
- MAIDA, E. G. AND J. PACIENZIA *Metodologías de desarrollo de software* 2015.
- MEDINA, G. N., E. GAITE AND D. GÁNDARA. *Avances sobre aplicaciones optimizadas para Smart Tv: "Tv-Widgets"*. In *XIII Workshop de Investigadores en Ciencias de la Computación*. 2011.
- MENZINSKY, A., G. LÓPEZ, J. PALACIO, M. SOBRINO, et al. *Historias de usuario* 2018.
- MICROSOFT. *Microsoft Ignite*. In., 2022.
- MONTERO, B. M., H. V. CEVALLOS AND J. D. J. E. R. M. D. I. CUESTA *Metodologías ágiles frente a las tradicionales en el proceso de desarrollo de software* 2018, 2(17), 114-121.
- MONTERRUBIO-HERNANDEZ, E. J. C.-C. S. B. C. D. L. E. P. I. *Sistema Operativo* 2019, 1(1).
- MORENO, I. J. A. D. I. E. *Para qué sirven las TIC* 2015, 240(2), 33-37.
- MORERA, J. D. D. M. AND S. C. J. U. POLINI *Implementación de un Servidor FTP utilizando el modelo Cliente/Servidor mediante el uso de Sockets en Lenguaje C Unix con el fin de mejorar los tiempos de respuesta en la Red* 2010, (24), 83-89.
- MOUSE, J. *Caja blanca. Obtenido de Prueba de camino básico*. In., 2017.
- NIETO RAMÍREZ, E. J. G. Y. O. D. M. *XML: un nuevo lenguaje, una nueva necesidad* 2017, 85(9), 0-0.
- NÚÑEZ, I. B. AND M. L. C. MUÑOZ *La Ingeniería de Requisitos como pilar fundamental en el desarrollo exitoso de proyectos de software Requirements engineering as a cornerstone in the successful development of software projects*.
- OLIVO SILVA, M. A. *Desarrollo del geoportal para la gestión del sistema de información ALPA de la ESPOCH utilizando el framework Vue. js*. Escuela Superior Politécnica de Chimborazo, 2019.
- PÉREZ, J. E. *introduccion a JavaScript*. In., 2019.
- PILICITA GARRIDO, A., Y. BORJA LÓPEZ AND G. GUTIÉRREZ CONSTANTE *Rendimiento de MariaDB y PostgreSQL* 2021.
- PRESSMAN, R. *Ingeniería de Software. Un enfoque práctico*. edited by M.G. HILL. Edtion ed., 2007.
- RODRÍGUEZ-BREIJO, V., J. GALLARDO-CAMACHO AND J. J. P. D. L. I. SIERRA-SÁNCHEZ *Información política en los vídeos que son tendencia en YouTube España* 2018, 27(5), 1041-1050.
- RODRÍGUEZ, J. J. V., M.-J. DÍAZ-GONZÁLEZ AND M. G. TORRE. *Contenidos periodísticos audiovisuales en aplicaciones móviles de televisión. In Notícias e Mobilidade: o jornalismo, na era dos dispositivos móveis. Labcom-Laboratório de Comunicação e Conteúdos Online* 2013, p. 219-238.
- RODRÍGUEZ PEÑA, A. D. AND L. G. J. R. C. D. I. M. SILVA ROJAS *Arquitectura de software para el sistema de visualización médica Vismedic* 2016, 8(1), 75-86.
- ROMÁN LÓPEZ, V. *Estudio sobre el desarrollo de aplicaciones con soporte multimedia en plataformas para móviles basadas en web*. 2014.
- ROSALES ROSA, E., A. A. FÍRVIDA DONÉSTEVEZ, M. GONZÁLEZ MUÑO AND A. P. FUENTES. *Smart TV with free technologies in support of teaching-learning process. In IFIP international conference on open source systems*. Springer, 2014, p. 147-152.

- RUIZ, V. H. J. C. R. L. D. C. Revolución Netflix: desafíos para la industria audiovisual 2017, (135), 275-295.
- RUMBAUGH, J., I. JACOBSON AND G. BOOCH *El Lenguaje Unificado de Modelado. Manual de Referencia*. Edtion ed., 2007.
- SALAZAR SERNA, R. A. AND N. O. ESCOBAR CEBALLOS Elección de un SO adecuado para el funcionamiento de la plataforma SIRUTP 2016.
- SÁNCHEZ LARA, C. M. El auge de las plataformas audiovisuales: Netflix y Twitch 2021.
- SÁNCHEZ MÁRQUEZ, A. Desarrollo de aplicaciones para Smart TV usando la tecnología Android TV. 2015.
- SÁNCHEZ, T. R. J. L. H. S. Metodología de desarrollo para la Actividad productiva de la UCI 2015.
- SARMIENTO, J. UML: Diagrama de Despliegue. In., 2013.
- SCRIBD. Obtenido de Casos de Prueba - Particion Equivalente. In., 2018.
- SILVA, A., E. LEDEZMA, J. CASTORENA, A. DOMÍNGUEZ, et al. Utilidad del Lenguaje Unificado de Modelado (UML) en el desarrollo de software profesional dentro del sector empresarial y educativo 2018, 56.
- STRASSER, C. C. Tizen Development 2018.
- TRAINING, L. A. Obtenido de Profesionales del testing 2017.

ANEXOS

Anexo 1: Historias de usuarios.

HISTORIA DE USUARIO	
Numero: 12	Usuario: Cliente
Nombre de la historia: Mostrar musicales	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Tiempo estimado: 6 días	Iteración asignada: 1
Programador responsable: Leydis Mariam Alarcón Sánchez	
Descripción: Le permite al usuario poder ver la lista de canciones que se encuentran en la plataforma	
Prototipo	de
	
	interfaz:

Tabla 16: Historia de Usuario RF12

HISTORIA DE USUARIO	
Numero: 13	Usuario: Cliente
Nombre de la historia: Seleccionar Musical	
Prioridad en negocio: alta	Riesgo en desarrollo: Alta

Tiempo estimado: 3 días	Iteración asignada: 2
Programador responsable: Leydis Mariam Alarcón Sánchez	
Descripción: Permite al usuario elegir el musical de su preferencia.	
Prototipo	de
interfaz:	
	

Tabla 17: Historia de Usuario RF13.

HISTORIA DE USUARIO	
Numero: 9	Usuario: Cliente
Nombre de la historia: Reproducir musical	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Tiempo estimado: 5 días	Iteración asignada: 1
Programador responsable: Leydis Mariam Alarcón Sánchez	
Descripción: Le permite al usuario reproducir el musical.	

Prototipo de interfaz:



Tabla 18: Historia de Usuario RF14.

Anexo 2: Diagramas de clases del diseño.

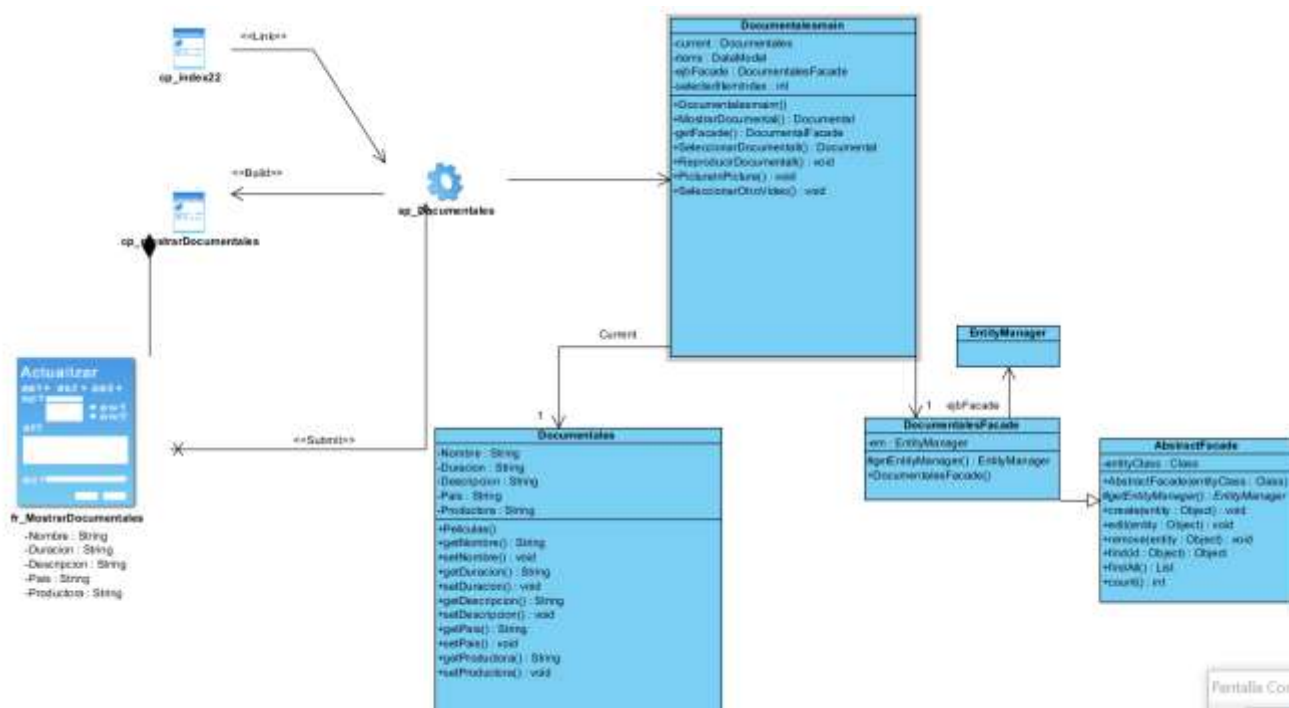


Figura 14: Diagrama de clases de Documentales.

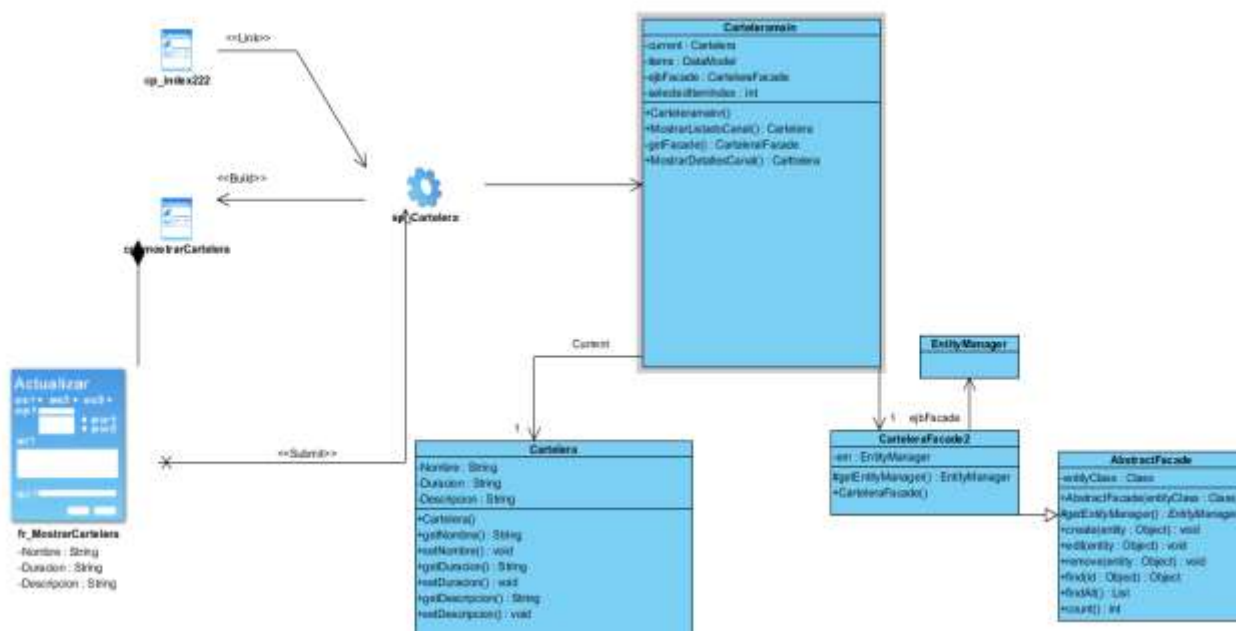


Figura 15: Diagrama de clases de Cartelera.

Anexo 3:

Encuesta para los usuarios pertenecientes a Cuba que tienen un acercamiento con la aplicación Picta para Smart TV Samsung.

Estimado usuario, la encuesta siguiente tiene el objetivo de conocer su opinión acerca del correcto funcionamiento de la aplicación Picta y cómo contribuye a mejorar la experiencia de los usuarios que acceden a sus servicios. Necesitamos su colaboración respondiendo con total precisión y sinceridad, este proceso es totalmente anónimo.

Datos generales:

Sexo: F__ M__

Marque con una X el indicador que se ajuste en cada caso.

1- ¿Entiendo lo que hace la aplicación Picta para Smart TV Samsung?

Si__ No__

2- ¿Sé cómo usar la aplicación Picta para Smart TV Samsung?

Si__ No__

3- ¿Por qué empezaste a usar la aplicación?

4- ¿Qué tan útil es la aplicación para ti y por qué?

5- Cuéntanos sobre tu experiencia con la aplicación Picta para Smart TV?

6- ¿Cómo calificarías la facilidad de uso de la interfaz de la aplicación?

7- Teniendo en cuenta que has utilizado mucho la interfaz de la aplicación, ¿cuál es la probabilidad de que la recomiendes a tus amigos y colegas?

8- ¿Cómo calificarías tu experiencia general con nuestra aplicación?

9- ¿Hay algo que cambiarías sobre cómo funciona?

10- ¿Hay algún comentario o sugerencia que pueda ayudarnos a mejorar la experiencia de usuario y la interfaz?

Anexo 4: Aval de Aceptación



DIRECCIÓN GENERAL DE TECNOLOGÍA
DIRECCIÓN DE INFORMATIZACIÓN

La Habana, 24 de noviembre de 2022
"Año 64 de la Revolución"

ACTA DE ACEPTACIÓN

De una parte, la Dirección de Informatización, en lo sucesivo DIN, de la Universidad de las Ciencias Informáticas, representado en este acto por: Ing. Yosbel Falero Vento, y de otra parte la estudiante: Leydis Mariam Alarcón Sánchez.

Primero: Que en cumplimiento de los requisitos funcionales han sido efectuadas las implementaciones correspondientes.

CONSIDERANDO: Que los hitos realizados han sido desarrollados con la calidad requerida y bajo las condiciones pactadas y aprobadas por las partes.

CONSIDERANDO: Que los hitos que se han ejecutado cumplen con los requerimientos establecidos.

POR TANTO: Las partes acuerdan formalizar mediante la presente Acta, la aceptación del producto:

"Desarrollo de la aplicación Picta para Smart TV Samsung"

Y para que así conste, se extiende la presente Acta en dos (2) ejemplares, rubricados por las partes.


Estudiante Leydis Mariam Alarcón Sánchez


Ing. Yosbel Falero Vento

Entrega

Recibe



Universidad de las Ciencias Informáticas
Carretera San Antonio Km 2 ½ Torrens, Boyero,
La Habana, Cuba.
Teléfono: +53 7 837 7020 y +53 7 837 7021

Figura 16:Aval de aceptación.