

UNIVERSIDAD MAYOR DE SAN ANDRÉS
FACULTAD CIENCIAS PURAS Y NATURALES
CARRERA DE INFORMÁTICA



PROYECTO DE GRADO

**“ENTRETENIMIENTO EN VUELO ORIENTADO A
DISPOSITIVOS MÓVILES ANDROID
CASO: LÍNEA AÉREA AMASZONAS S.A.”**

**PARA OPTAR EL TÍTULO DE LICENCIATURA EN INFORMÁTICA
MENCIÓN INGENIERÍA DE SISTEMAS INFORMÁTICOS**

POSTULANTE : UNIV. MARIO ROBERTO CARLO MIRANDA
TUTOR METODOLÓGICO : M.SC. ROSA FLORES MORALES
ASESOR : LIC. MARCELO GERMAN ARQUIPA CHAMBI

LA PAZ – BOLIVIA
2017



**UNIVERSIDAD MAYOR DE SAN ANDRÉS
FACULTAD DE CIENCIAS PURAS Y NATURALES
CARRERA DE INFORMÁTICA**



LA CARRERA DE INFORMÁTICA DE LA FACULTAD DE CIENCIAS PURAS Y NATURALES PERTENECIENTE A LA UNIVERSIDAD MAYOR DE SAN ANDRÉS AUTORIZA EL USO DE LA INFORMACIÓN CONTENIDA EN ESTE DOCUMENTO SI LOS PROPÓSITOS SON ESTRICTAMENTE ACADÉMICOS.

LICENCIA DE USO

El usuario está autorizado a:

- a) visualizar el documento mediante el uso de un ordenador o dispositivo móvil.
- b) copiar, almacenar o imprimir si ha de ser de uso exclusivamente personal y privado.
- c) copiar textualmente parte(s) de su contenido mencionando la fuente y/o haciendo la referencia correspondiente respetando normas de redacción e investigación.

El usuario no puede publicar, distribuir o realizar emisión o exhibición alguna de este material, sin la autorización correspondiente.

TODOS LOS DERECHOS RESERVADOS. EL USO NO AUTORIZADO DE LOS CONTENIDOS PUBLICADOS EN ESTE SITIO DERIVARA EN EL INICIO DE ACCIONES LEGALES CONTEMPLADOS EN LA LEY DE DERECHOS DE AUTOR.

DEDICATORIA

A Dios y mi mamita Rosy; que me motivaron al ingreso a tan linda carrera; a mi papi Mario que debe estar orgulloso desde el cielo.

Especialmente a mi esposita Paola Lima; que me alentó a culminar esta carrera, dándome fuerzas día a día.

A mi Tía Esperanza; que económicamente me apoyo en mis estudios universitarios

A mi hijita Soly, mi familia y amigos, que no dudaron de mi capacidad para la culminación de esta etapa de mi vida, apoyándome, en especial a Victor Limachi.

AGRADECIMIENTOS

A Dios por ser mi guía, estando siempre a mi lado, dándome bendiciones en mis estudios universitarios.

Mis agradecimientos a todas las personas

A mi madre, mi esposa, mi hija y mi familia por su amor, confianza y comprensión.

A la Lic. Rosa Flores, quien como Docente Tutor fue la guía con un gran aporte por sus motivaciones, sus consejos, observaciones, comprensión y por el tiempo invertido al momento de evaluar el trabajo.

Al Lic. Marcelo Aruquipa Chambi, quien como Docente Asesor, me dio su apoyo incondicional, realizando observaciones pertinentes y correcciones oportunas, por lo que pude seguir adelante con el presente proyecto.

Agradecimiento a los Licenciados de la Carrera de Informática, por los conocimientos transmitidos durante el tiempo de mi formación.

A la Línea Aérea Amazonas S.A. por permitirme desarrollar el presente Proyecto de Grado, en especial a Erick Silberstein por guiarme en el desarrollo del Sistema.

A todas las personas que hicieron posible culminar el presente trabajo de manera satisfactoria, a mis compañeros, a mis amigos con quienes compartimos alegrías y tristezas.

Muchas Gracias....

ÍNDICE DE CONTENIDO

RESUMEN	ix
ABSTRACT.....	x
CAPITULO I	1
1 MARCO REFERENCIAL	1
1.1 Introducción	1
1.2 Antecedentes.....	2
1.2.1 Antecedentes de la institución.....	2
1.2.2 Antecedentes del proyecto	2
1.3 Planteamiento del problema.....	6
1.4 Objetivos.....	7
1.4.1 Objetivo general.....	7
1.4.2 Objetivos específicos.....	7
1.5 Justificación	8
1.5.1 Justificación tecnológica.....	8
1.5.2 Justificación económica.....	8
1.5.3 Justificación social	8
1.6 Alcances y Límites.....	9
1.6.1 Alcances.....	9
1.6.2 Límites	9
1.7 Aportes	9
1.7.1 Aporte Tecnológico	9
1.8 Metodologías y Herramientas.....	10
1.8.1 Metodologías.....	10
1.8.2 Herramientas.....	10
CAPITULO II	12
2. MARCO TEÓRICO	12
2.1 Metodología y arquitectura	12
2.1.1 Método de desarrollo ágil Scrum	12
2.2 Arquitectura clientes servidor	13
2.2.1 Servidor pesado, cliente ligero.....	13

2.3	Tecnologías de software.....	16
2.3.1	Android.....	16
2.3.2	MVC.....	17
2.3.3	InVision.....	18
2.3.4	Visual Paradigm.....	18
2.3.5	Eclipse.....	19
2.3.6	Android SDK.....	19
2.3.7	JSON.....	19
2.3.8	Volley Librería de Google.....	20
2.3.9	PHP.....	20
2.3.10	GenyMotion.....	21
2.3.11	Wowza Streaming Engine.....	21
2.4	Componentes de una aplicación android.....	22
2.4.1	Actividades.....	22
2.4.2	Intents.....	24
2.4.3	Servicios.....	25
2.4.4	Content providers.....	25
2.5	La tecnología Streaming.....	26
2.5.1	Componentes.....	26
2.5.2	Protocolo RTSP.....	27
2.6	Costos y/ o beneficios.....	27
2.6.1	Método de COCOMO II.....	27
2.6.2	Estimación con COCOMO II.....	28
2.6.3	Modelo Post-Arquitectura.....	29
2.7	Proceso de calidad del desarrollo de software.....	31
2.7.1	ISO 9126.....	31
2.7.2	Buenas prácticas W3C (<i>World Wide Web Consortium</i>).....	35
2.7.3	Técnica de prueba.....	41

CAPITULO III 42

3 MARCO APLICATIVO 42

3.1	Introducción.....	42
3.2	Pre Game Etapa de análisis.....	43
3.2.1	Recopilación de requerimientos funcionales.....	43
3.2.2	Recopilación de requerimientos no funcionales.....	49
3.2.3	Cronograma de trabajo.....	51
3.2.4	Identificación de riesgos.....	52
3.2.5	Iteraciones.....	53
3.2.6	Primera iteración.....	53
3.2.7	Segunda iteración.....	54
3.2.8	Tercera iteración.....	54

3.3	Game	55
3.3.1	Identificación de roles	55
3.3.2	Caso de uso general del sistema	56
3.3.3	Diagrama de componentes	61
3.3.4	Modelo Conceptual	62
3.3.5	Modelo de Servicio	64
3.3.6	Diseño de pantallas	65
3.3.7	Desarrollo de módulos	67
3.4	Post-Game. Pruebas	71
3.4.1	Fase de ejecución de pruebas	72
3.4.2	Pruebas de ejecución	77
3.4.3	Pruebas de calidad de software	78
3.4.4	Costo de desarrollo	83
CAPITULO IV		90
4	CONCLUSIONES Y RECOMENDACIONES.....	90
4.1	Conclusiones.....	90
4.2	Recomendaciones	91
5	BIBLIOGRAFÍA	92

INDICE DE FIGURAS

FIGURA 2.1.1 PROCESO DE DESARROLLO SCRUM	12
FIGURA 2.2.2 ARQUITECTURA DEL SISTEMA	15
FIGURA 2.3.3 MODELO VISTA CONTROLADOR	17
FIGURA 2.4.4 CICLO DE VIDA DE UNA ACTIVIDAD EN ANDROID	23
FIGURA 2.6.5 ARQUITECTURA DE COCOMO II	29
FIGURA 3.2.1 PLANIFICACIÓN DE ITERACIONES	52
FIGURA 3.3.1 CASO DE USO GENERAL DEL SISTEMA	56
FIGURA 3.3.2 D.S. BUSCAR GÉNERO DE PELÍCULA.....	58
FIGURA 3.3.3 D.S. REPRODUCIR UNA PELÍCULA	59
FIGURA 3.3.4 BUSCAR GÉNERO DE CANCIONES	60
FIGURA 3.3.5 REPRODUCIR CANCIÓN.....	61
FIGURA 3.3.6 DIAGRAMA DE COMPONENTES.....	61
FIGURA 3.3.7 MODELO DE SERVICIOS	65
FIGURA 3.3.8 INTERFAZ DE INGRESO.....	66
FIGURA 3.3.9 INTERFAZ DE SELECCIÓN DE TIPO.....	66
FIGURA 3.3.10 INTERFAZ DE DESPLIEGUE DE GÉNEROS DE PELÍCULA	67
FIGURA 3.3.11 SELECCIÓN DE PELÍCULA.....	67

ÍNDICE DE TABLAS

TABLA 2.7-1 AGRUPACIÓN Y TÉCNICAS DE PRUEBAS	41
TABLA 3.1-1 TABLA DE ACTIVIDADES SEGÚN SCRUM	42
TABLA 3.2-1 RECOPIACIÓN DE REQUERIMIENTOS.....	43
TABLA 3.2-2 PLANTILLA DE HISTORIA DE USUARIO	44
TABLA 3.2-3 BÚSQUEDA DE GÉNERO DE PELÍCULAS	45
TABLA 3.2-4 BÚSQUEDA DE PELÍCULAS	45
TABLA 3.2-5 REPRODUCCIÓN DE PELÍCULAS	45
TABLA 3.2-6 BÚSQUEDA DE GÉNERO DE CANCIONES	46
TABLA 3.2-7 BÚSQUEDA DE CANCIONES	46
TABLA 3.2-8 REPRODUCCIÓN DE CANCIÓN	47
TABLA 3.2-9 BÚSQUEDA DE LIBRO.....	47
TABLA 3.2-10 VISUALIZACIÓN DEL CONTENIDO DE UN LIBRO	47
TABLA 3.2-11 LECTURA DE REVISTA DE LA EMPRESA.....	48
TABLA 3.2-12 PUBLICIDAD DE LA EMPRESA.	48
TABLA 3.2-13 PLANTILLA DE FORMULACIÓN DE REQUISITOS NO FUNCIONALES	49
TABLA 3.2-14 PUBLICIDAD DE LA EMPRESA.	49
TABLA 3.2-15 R.N.F. SISTEMA OPERATIVO.	50
TABLA 3.2-16 R.N.F. RENDIMIENTO.....	50
TABLA 3.2-17 R.N.F. ESCALABILIDAD	50
TABLA 3.2-18 R.N.F. IDIOMA.....	51
TABLA 3.2-19 R.N.F. INTERNET.	51
TABLA 3.2-20 CRONOGRAMA DE TRABAJO POR ITERACIÓN	51
TABLA 3.2-21 ANÁLISIS DE RIESGOS	52
TABLA 3.2-22 TAREAS DE LA PRIMERA ITERACIÓN	53
TABLA 3.2-23 TAREAS DE LA SEGUNDA ITERACIÓN.....	54
TABLA 3.2-24 TAREAS DE LA TERCERA ITERACIÓN	55
TABLA 3.3-1 IDENTIFICACIÓN DE ROLES DE USUARIOS	55
TABLA 3.3-2 PLANTILLA DE REALIZACIÓN DE CASOS DE USO	57
TABLA 3.3-3 CU #1 BUSCAR GÉNERO DE PELÍCULA	57
TABLA 3.3-4 CU #2 REPRODUCIR UNA PELÍCULA.....	58

TABLA 3.3-5 CU #3 BUSCAR GÉNERO DE CANCIONES	59
TABLA 3.3-6 CU #4 REPRODUCIR CANCIÓN	60
TABLA 3.3-7 DISEÑO DE TABLAS	62
TABLA 3.4-1 FORMULARIO DE ENCUESTA	73
TABLA 3.4-2 – RESULTADOS DE ENCUESTA	74
TABLA 3.4-3 RESULTADOS ENCUESTA	76
TABLA 3.4-4 PRUEBA DE INGRESO AL SISTEMA	77
TABLA 3.4-5 PRUEBA DE SELECCIÓN DE PELÍCULAS	77
TABLA 3.4-6 PRUEBA DE REPRODUCCIÓN DE TEMAS MUSICALES	78
TABLA 3.4-7 VALORES DE AJUSTE DE COMPLEJIDAD.....	78
TABLA 3.4-8 AJUSTE DE COMPLEJIDAD DEL PUNTO FUNCIÓN.....	79
TABLA 3.4-9 RESULTADOS DE EVALUACIÓN DE EJECUCIÓN DEL SISTEMA.....	80
TABLA 3.4-10 RESULTADOS DE EVALUACIÓN SEGÚN ENCUESTA A LOS USUARIOS.....	81
TABLA 3.4-11 RESULTADOS DE EVALUACIÓN DE EFICIENCIA DEL SISTEMA	82
TABLA 3.4-12 FACTORES	83
TABLA 3.4-13 MÓDULOS DEL SISTEMA.....	84
TABLA 3.4-14 CONTEO DE LINEAS	85
TABLA 3.4-15 TAMAÑO DE SLOC	85
TABLA 3.4-16 FACTOR EXPONENCIAL	86
TABLA 3.4-17 ESFUERZO NOMINAL	87
TABLA 3.4-18 NIVELES DE MULTIPLICADORES DE ESFUERZO	87
TABLA 3.4-19 ESFUERZO ESTIMADO POR MODULO	88
TABLA 3.4-20 ESFUERZO ESTIMADO DEL SISTEMA TOTAL PM.....	88

RESUMEN

Los servicios que se ofrecen en las aeronaves, generalmente no son personalizados hacia los pasajeros y por ende no siempre están satisfechos.

Presentar servicios de entretenimiento acordes a los gustos y preferencias del pasajero que realizan viajes de un punto geográfico a otro en una aeronave, en especial en aviones de la Línea Aérea Amazonas S.A, no siempre han sido considerados prioritarios por los directivos de la empresa.

Ante esta problemática, se propone establecer una política de recreación basada en el uso de la tecnología orientada a que los pasajeros puedan viajar y sentirse cómodos y distraídos de problemas personales hasta llegar a su destino mirando videos, revistas o publicidad de la empresa durante el vuelo.

Para lograr este objetivo, se desarrolló una aplicación móvil en plataforma android y tablets vía *wifi* local con conexión a la intranet desde donde el pasajero selecciona una película o un tema musical para su reproducción, permitiendo que el pasajero se sienta distraído, aplicando la tecnología *streaming*.

Se utilizó para ello la metodología de desarrollo Scrum junto a tecnología Cliente-Servidor elaborado con herramientas de libre uso, logrando un producto estable y robusto que ha sido sometido a pruebas tanto por los administradores del sistema, así como los pasajeros de la Línea Aérea Amazonas S.A.. Este producto final, se constituye como una de las primeras herramientas ofertadas por Amazonas y permitirá ser considerado como un hito en el uso de tecnologías móviles orientadas a los pasajeros en nuestro medio; además de ser considerado como una herramienta para la fidelización del uso de los servicios de transporte de la empresa.

Palabras clave: entretenimiento, *wifi*, *streaming*, intranet, android.

ABSTRACT

The services they offer on the aircraft, are generally not customized for the passengers and for the effort they are not always satisfied.

To present entertainment services according to the tastes and preferences of the passengers that travel from one geographical point to another in an aircraft, especially in airplanes of the Airline Amazonas SA, have not been considered a priority by company executives.

Faced with this problem, it is proposed to establish a recreation policy based on the use of technology aimed at the passengers to travel and feel comfortable and distracted from personal problems until arriving at their destination watching videos, books, magazines and advertising the company during the flight.

To achieve this goal, developed a mobile application on the android platform and tablets via local wi-fi with connection to the intranet from where the passenger selects a movie or a musical theme for its reproduction, allowing the passenger to feel distracted, applying streaming technology.

It was used for Scrum development methodology coupled with Client-Server technology crafted with free wrench tools using a stable and robust product that had been subjected to testing by both system administrators as well as airline passengers. This final product, constitute one of the first tools of Amazonas applications and are used as a milestone in the use of mobile technologies oriented to the passengers in our midst; in addition to being considered as a tool for the fidelization of the use of the transport services of the company.

Keywords: entertainment, wifi, streaming, intranet, android.

CAPITULO I

MARCO REFERENCIAL

1.1 Introducción

Muchas veces las organizaciones no entraron en la etapa de cambio hacía la era tecnológica sin darse cuenta que es un riesgo absolutamente grande de fracaso debido a las amenazas del mercado y su imposibilidad de competir, es de ahí que, todos los Sistemas de Información basados en cualquier plataforma se están convirtiendo rápidamente en un componente necesario para el éxito empresarial en el entorno global y dinámico actual.

Atrás han quedado los magos, caricaturistas y pequeñas bandas de música que en los años 50 acompañaban a los pasajeros en los vuelos más lujosos.

En realidad, corría el año 1921 cuando un avión mostró por primera vez una película a sus pasajeros (aunque por esta época era cine mudo). A partir de 1931 se hicieron muy populares las transmisiones de radio en vivo. Fue en 1948 cuando se mostró la primera película con sonido (a pesar de que casi no se apreciara por el ruido del propio avión), todo ello en aerolíneas norteamericanas.

Pero no fue hasta los años 60 cuando nacieron los sistemas de entretenimiento en vuelo como tales que, entre funciones, cumplían la de disimular los retrasos de los vuelos para tranquilizar a los pasajeros, aunque fuera sin auriculares.

Un buen sistema de entretenimiento en vuelo es una de las cuestiones que determinan el buen servicio al pasajero que da una aerolínea. De hecho, es uno de los recursos que usan como reclamo para atraer a sus pasajeros, ya que el sistema moderno de entretenimiento en vuelo permite convertir el viaje en una buena experiencia y ayuda a que el tiempo pase más rápido, sobre todo, en viajes de larga distancia. El sistema de entretenimiento en vuelo tiene tanta importancia que en la actualidad se han creado premios.

El propósito del presente proyecto es desarrollar un Software de Aplicación que permita visualizar películas y reproducir canciones, aprovechando los dispositivos móviles android y tablets, con el fin de entretener a los pasajeros en el trayecto de su viaje; y de esta manera incrementar un servicio de calidad a la Línea Aérea para aumentar la cantidad de pasajeros transportados.

1.2 Antecedentes

1.2.1 Antecedentes de la institución

La Línea Aérea Amazonas S.A. inicia sus actividades en 1999 con un personal que no superaba los 10 empleados. Hoy en día, llegó a más de 1.000.000 pasajeros transportados y 1.400.000 Kg de carga depositadas en destino, la empresa cuenta con una flota de aeronaves certificadas para operar en aeropuertos de gran altura y pistas no preparadas.

La empresa Amazonas S.A. es una empresa en franco crecimiento y ha ido expandiéndose en el transcurrir de los años hasta convertirse en una de las más importantes empresas del sector aeronáutico en el país. Actualmente la empresa tiene oficinas en las ciudades de La Paz, Santa Cruz, Cochabamba, Sucre, Uyuni y Rurrenabaque. Teniendo vuelos hacia otros países como ser Chile, Paraguay, Uruguay y Perú. Además, cuenta con agentes de venta a nivel Internacional.

En la actualidad por la clase de aviones que cuenta CRJ-200 (capacidad de 49 pasajeros) no existe algún entretenimiento hacia el pasajero, solamente se brinda refrigerio como ser agua, café, café con leche y jugo frutal.

1.2.2 Antecedentes del proyecto

En el ámbito local no se desarrollaron proyectos similares, pero se puede citar noticias relacionadas al proyecto en el ámbito aeronáutico:

Los sistemas de entretenimiento en vuelo, en el pasado

“Posiblemente se puede tomar a broma, pero el primer intento de entretenimiento en vuelo fue subir un piano a bordo. Es obvio que no se trata de un avión convencional, sino del Hindenburg, el dirigible alemán que terminó con la era de los zepelines al estrellarse. El interior del Hindenburg, para entretener a los pasajeros en los viajes que suponían días, contaba con el citado piano, un bar, una sala de fumadores, un comedor y un salón.

El entretenimiento en vuelo se transformó en el servicio de catering y en la proyección de películas. Las azafatas han sido las encargadas de este catering hasta nuestros días, y la proyección se hacía en la parte frontal de la cabina.

La primera película reproducida en vuelo es ‘Howdy Chicago’, lo cual se hizo en 1921. Y se tuvo que esperar hasta 1963 para empezar a ver tomas de auriculares en los asientos, los cuales eran neumáticos. A partir de 1979 se empezaron a sustituir por auriculares electrónicos. Y las pantallas individuales vinieron en 1988 de la mano del Boeing 747, hechas por Airvision para Northwest Airlines.” (Perez, 2017).

El entretenimiento actual: pantallas individuales y LRUs

Lo común en los aviones es que cada asiento tenga su propia pantalla individual. Que los pasajeros controlen lo que aparece con un mando o con una pantalla táctil. La idea es que cada pasajero pueda ver lo que quiera en la pantalla, nada de una retransmisión para todo el mundo. Incluso esto abre la puerta a la personalización, usando sus nombres o datos para que la aerolínea ofrezca sus servicios (Perez, 2017).

Este sistema, que hace posible que cada pasajero vea lo que quiera, está basado en un servidor (un ordenador pequeño) ubicado en el avión. Se llama LRU, *Line Replaceable Unit*, y es el responsable de controlar todas esas pantallas gracias a conexiones *Ethernet*. Las series y películas se alojan en otros LRU con discos duros. Y el contenido

se actualiza con fibra óptica, a través de un dispositivo que contiene las películas cifradas por los estudios.

Todo esto se vuelve un poco caótico cuando cada aerolínea crea su propio sistema de entretenimiento. Las operadoras están muy interesadas en mantener su propia marca, así que suelen crear un software exclusivo que funcione sólo en sus aviones. También cierran por separado los acuerdos con las productoras para tener contenido. Así que, por eso, cada sistema de entretenimiento puede ser un mundo (Perez, 2017).

Lo que no se diferencia tanto es quien es el responsable de montar todas las piezas. La mayoría de aerolíneas usan los servicios de Panasonic Avionics para montar las pantallas, los LRU y todas las conexiones intermedias. Según Panasonic, más de 300 aerolíneas utilizan sistemas de Panasonic Avionics. Esto implica que Panasonic estaría ofreciendo entretenimiento en vuelo para más de 500 millones de pasajeros (Perez, 2017).

El futuro del entretenimiento en vuelo

“Sin embargo, las aerolíneas están muy interesadas en eliminar todas esas pantallas. Esta es una época en la que cada pasajero tiene un *smartphone*, una tablet o un ordenador al viajar. Y todo ese equipo que se despliega en los aviones pesa, el peso implica mayor gasto de combustible. Y el mayor consumo de *jet fuel* aumenta los gastos de tener un avión en el aire.

Así que la idea de las aerolíneas es ofrecer todo este contenido a través de la red *WiFi on-board*. Cuando el pasajero se conecta al *WiFi* del avión, el LRU dará acceso a todo ese contenido. Y esto también abre las puertas a integrar conexiones a Internet.” (Perez, 2017).

Respecto a los posibles temores a fallos de seguridad que esto pueda generar, está controlado. Desde que el Sistema de Entretenimiento pudiera estar implicado en un accidente en 1998, la aplicación en vuelo se aísla del resto del avión. Así, en caso de

fallo, no afectaría a la seguridad o funcionalidad de un avión. Y un hacker no podría tomar el control del avión con un fallo del Sistema de Entretenimiento.

“Iberia ofrece *wifi* en vuelos directos entre Costa Rica y España, y compañías de EE.UU. empiezan a dar en la región. Iberia ofrece conexión *wifi* y GSM para Internet y llamadas desde teléfonos celulares en los vuelos directos entre San José y Madrid.

Las aerolíneas están ampliando sus servicios de conectividad a Internet y de entretenimiento para sus pasajeros durante los vuelos comerciales a diferentes destinos. *American Airlines*, Delta, Iberia y *United Airlines* confirmaron que ya brindan Internet, mediante tecnología inalámbrica *wifi*, especialmente en rutas domésticas en Estados Unidos (EE. UU.), y que refuerzan el servicio aumentando la velocidad y la cobertura, incluyendo a la región de América Latina y el Caribe. Avianca anunció que espera brindar este tipo de conectividad durante el 2017. Los esfuerzos se dirigen a atender las necesidades, exigencias y expectativas de los usuarios, que requieren cada vez más comunicación para negocios y asuntos personales, así como en el campo del entretenimiento, superando diversos tipos de limitaciones.” (Cordero, 2016).

A nivel internacional se pueden mencionar trabajos afines y son los siguientes:

- Sistemas de entretenimiento en vuelo, con el Título: “eX1”, realizado por la compañía Panasonic, en el año 2007.
- Sistemas de entretenimiento en vuelo, con el Título: “eX2”, realizado por la compañía Panasonic, en el año 2009.
- Sistemas de entretenimiento en vuelo, con el Título: “eX3”, realizado por la compañía Panasonic, en el año 2012.
- Sistemas de entretenimiento en vuelo, con el Título: “eXLite”, realizado por la compañía Panasonic, en el año 2013.
- Sistemas de entretenimiento en vuelo, con el Título: “eXW”, realizado por la compañía Panasonic, en el año 2015 – ofrece a los pasajeros de Aeroméxico un interfaz personalizado familiar, fácil de usar, que ha sido

diseñado para una experiencia de visualización óptima. El interfaz está respaldado por la mayor librería de juegos, audio y video de la industria.

- Sistemas de entretenimiento en vuelo, con el Título: “Entice”, realizado por la compañía GEE, en el año 2016 - tiene una inmensa cantidad de contenido de casi 10.000 horas de programación con un archivo sin precedente de programas y películas populares de todas partes del mundo. También cuentan con algoritmos que tienen la capacidad de crear perfiles que proporcionan recomendaciones de contenido relevantes y el disfrute continuo entre vuelos. Aparte de lo antes mencionado, las aerolíneas pueden beneficiarse de la información proveniente de los reportes y análisis de uso, permitiéndoles colocar anuncios publicitarios y ofertas más acertadas de acuerdo al perfil que genera pasajero.
- Sistema de entretenimiento a bordo, con el Título: “*LATAM Entertainment*”, realizado por la Línea Aérea LATAM, en el año 2016 - es una aplicación móvil que consiste en sistema de entretenimiento inalámbrico a bordo para dispositivos personales, a través del cual los pasajeros pueden ver más de 50 películas, 42 capítulos de series, además de explorar la ruta en el mapa de vuelo, todo utilizando sus dispositivos electrónicos personales como teléfonos inteligentes, tablets y laptops de manera gratuita. En el futuro, se agregarán contenidos de noticias y un *e-reader*, entre otros.

1.3 Planteamiento del problema

En el área del transporte aéreo surge la necesidad de mejorar y automatizar en lo posible todas las áreas laborales, debido a que el costo es elevado para realizar un servicio eficaz por la demanda que significa el contrato de personas con una mayor experiencia en el área aeronáutica.

Muchas Líneas Aéreas Internacionales están pensando brindar el servicio de entretenimiento en vuelo en dispositivos móviles, pues al eliminar sus pantallas de entretenimiento que tienen en la parte trasera de sus asientos, bajaría un buen porcentaje

de sus costos por el hecho que a menor peso en el avión mayor ahorro de combustible, ya que el jet fuel tiene un costo elevado.

Actualmente la Línea Aérea Amazonas S.A. no cuenta con un servicio de entretenimiento en vuelo debido a que la infraestructura de sus aeronaves no lo permite; esto toma incidencia cuando el pasajero toma la elección en varias líneas aéreas, y este es un factor más que hace que la empresa reduzca la captación de pasajeros en temporada baja.

1.4 Objetivos

1.4.1 Objetivo general

Analizar, Diseñar e Implementar un Software de aplicación orientado a móviles Android que mejore la calidad de vuelo de los pasajeros de la Línea Aérea Amazonas S.A., facilitando el acceso a contenido multimedia.

1.4.2 Objetivos específicos

Para desarrollar una aplicación funcional se pretende cumplir los siguientes objetivos específicos:

- Implementar un servidor multimedia capaz de distribuir la información a la cantidad de pasajeros en vuelo.
- Gestionar altas, bajas y modificaciones de archivos multimedia en el servidor como ser: video, audio.
- Mejorar la satisfacción y entretenimiento del pasajero dentro de la aeronave
- Obtener una elección eficaz de una metodología ágil para el desarrollo de la aplicación.
- Aplicar el uso de técnicas para el manejo de streaming
- Desarrollar un servicio web para la transferencia de información de los archivos multimedia hacia el dispositivo Android.

1.5 Justificación

1.5.1 Justificación tecnológica

Con el pasar de los años van apareciendo innovadores equipos informáticos, como ser servidores computacionales de un tamaño reducido y poco peso, de fácil instalación dentro de un avión.

El celular se volvió una herramienta de comunicación muy indispensable, el uso de redes sociales da la necesidad a la persona de tener un *Smartphone*. Por esta razón el apps podrá ser aplicado por la mayoría de los celulares de los pasajeros.

1.5.2 Justificación económica

En este proyecto una de las partes importantes es la justificación económica ya que al elegir la plataforma de desarrollo Android la implementación de aplicaciones no tiene costo, al aplicar herramientas de tipo “*Open Source*”, se evita el pago de licencias de uso y la aplicación instalada puede ser distribuida libremente por el servidor multimedia.

Por otra parte, el costo del servidor debe ser mínimo porque no requiere mayores requisitos para su ejecución.

A respecto del uso de archivos multimedia, el costo de compra es muy bajo con respecto a años anteriores, por ejm: costo por una película actual es de \$us 3.-

1.5.3 Justificación social

Está por demás decir, el alcance del aplicativo es para todo usuario que cuente con celular Android, con el apoyo de las características tecnológicas accesibles en estos dispositivos móviles. La instalación es muy sencilla y la conexión con el servidor multimedia es de manera automática.

1.6 Alcances y Límites

1.6.1 Alcances

El presente trabajo abarca únicamente a los pasajeros en vuelo pertenecientes a la Línea Aérea Amaszonas S.A.

El software de aplicación es funcional en celulares *smartphone*, tablets con sistema Android con versión 4.0 como mínimo.

1.6.2 Límites

El software de aplicación de entretenimiento en vuelo para la empresa AMASZONAS S.A. siguiendo los objetivos trazados, tiene como limitaciones:

- Móviles de los pasajeros con tecnología Iphone y Blackberry
- Equipos de computación portátiles

1.7 Aportes

1.7.1 Aporte Tecnológico

El solo hecho que implica de la no existencia de una aplicación Android en Bolivia para una Línea Aérea mediante Wi-Fi ya constituye un aporte que se utilizará como punto de partida para el desarrollo de futuras aplicaciones similares.

Este trabajo también puede ser muy bien aplicado en la Domotica, de esta manera personalizar los archivos multimedia que una persona cuenta en su casa.

Las librerías Android a utilizar son actuales, estas herramientas son necesarias de conocer para desarrollar trabajos en este Sistema Operativo.

1.8 Metodologías y Herramientas

1.8.1 Metodologías

Para el desarrollo integral de este proyecto se fundamentará en el estudio del método sistemático porque el sistema implementado para la Línea Aérea Amaszonas S.A. requiere la integración de varios componentes.

El tipo de investigación será de desarrollo tecnológico con enfoque cuantitativo, tomando en cuenta las siguientes particularidades (Muñoz, 2011).

- Obtener conocimiento útil para resolver un problema concreto de las necesidades de la sociedad.
- El diseño inicial no es único y definitivo, pues no hay una solución correcta para un problema, siempre es posible mejorar y modificar un diseño.
- Que sea factible desde el punto de vista tecnológico, si se dispone de los conocimientos necesarios de diseño, operacionalidad y materialización.
- Los métodos e instrumentos propios de las disciplinas técnicas e ingenieriles y áreas afines que se utilizan están en constante evolución y se renuevan para adaptarse a los conocimientos científicos en los que se apoyan.

Para el desarrollo del prototipo del proyecto se utilizará la metodología ágil SCRUM y el patrón de diseños MVC (Modelo Vista Controlador)

1.8.2 Herramientas

Para el desarrollo de software se aplicará las siguientes herramientas, explicada en profundidad en el Marco Teórico.

- Para la descripción de actividades, funcionalidades y análisis de requerimientos se utilizará las Normas y estándares UML ver 2.5 (Lenguaje Unificado de Modelado).
- Invision, para realizar prototipo de la aplicación antes de su programación para resolver el problema de cómo mostrar la nueva funcionalidad al usuario

- *Visual Paradigm UML* para documentar con diagramas UML el desarrollo de la apps.
- *Xampp*, para la administración de los archivos planos mediante php y mysql.
- SDK (*Software Development Kit – Kit de Desarrollo de Software*) de Android.
- *Android Development Tools* con soporte API para Eclipse.
- Librería *Volley* de Google.
- Entorno IDE de programación Eclipse.
- Wowza - servidor streaming



CAPITULO II

MARCO TEÓRICO

2.1 Metodología y arquitectura

2.1.1 Método de desarrollo ágil Scrum

“Scrum es un proceso ágil para desarrollar software que fue aplicado por primera vez por Ken Schwaber y Jeff Sutherland., quienes lo documentaron en detalle en el libro *Agile Software Development with Scrum*. Esta metodología centra su atención en las actividades de Gerencia y no especifica prácticas de Ingeniería. Fomenta el surgimiento de equipos autodirigidos cooperativos y aplica inspecciones frecuentes como mecanismo de control.

Scrum parte de la base de que los procesos definidos funcionan bien sólo si las entradas están perfectamente definidas y el ruido, ambigüedad o cambio es muy pequeño. Por lo tanto, resulta ideal para proyectos con requerimientos inestables, ya que fomenta el surgimiento de los mismos.” (Palacios, 2014).

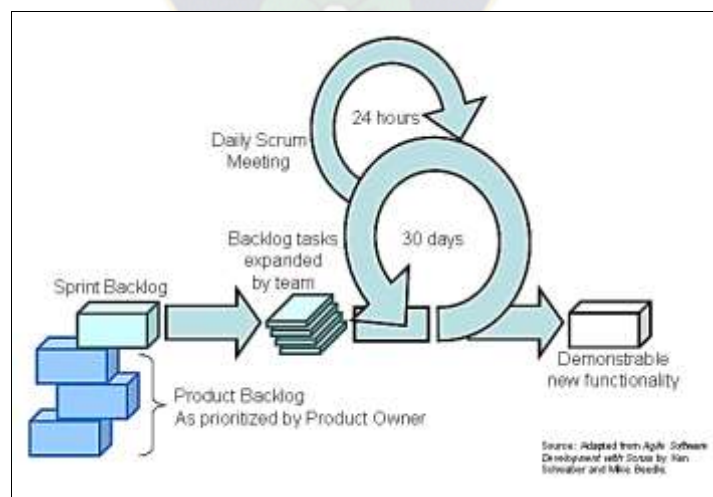


Figura 2.1.1 Proceso de desarrollo Scrum
Fuente: (Palacios, 2014)

El ciclo de vida según la figura 2.1.1 definido por Scrum es incremental iterativo y se caracteriza por ser muy adaptable.

Entre sus principales características están:

- Equipos autodirigidos
- Utiliza reglas para crear un entorno ágil de administración de proyectos
- No prescribe prácticas específicas de ingeniería
- Los requerimientos se capturan como ítems de la lista *Product Backlog*

El producto se construye en una serie de *Sprints* de un mes de duración.

El trabajo a ser realizado en un proyecto Scrum es listado en el *Product Backlog*, que es una lista de todos los cambios requeridos sobre un producto.

Los proyectos se realizan durante una serie de iteraciones de un mes de duración llamadas *Sprints*. Al comienzo de cada *Sprint* tiene lugar una *Sprint Planning Meeting* durante la cual el *Product Owner* prioriza el *Product Backlog* y el Scrum Team selecciona las tareas que serán completadas durante el *Sprint* que va a comenzar. Esas tareas son removidas del *Product Backlog* para ser llevadas al *Sprint Backlog* (Palacios, 2014).

Durante el *Sprint* el equipo se mantiene en contacto a través de las *Daily Meetings*. Y al final del *Sprint* debe mostrar la funcionalidad completa en la *Sprint Review Meeting*.

2.2 Arquitectura clientes servidor

El sistema se basa en una arquitectura de modelo cliente-servidor. Una de las características del modelo cliente/servidor es que permite balancear la potencia de cálculo aplicada hacia el lado servidor o hacia el lado cliente, para el proyecto se tomará la alternativa:

2.2.1 Servidor pesado, cliente ligero

El lado cliente se emplea sólo para el nivel de presentación (en nuestro caso el móvil) y el lado servidor se encarga de ejecutar la aplicación. En este tipo de esquemas se

puede disponer, incluso, de clientes sin disco duro. Otra de las ventajas de esta opción es que ofrecen una mayor seguridad frente a intentos de acceso indebido.

Además de las ventajas proporcionadas por la arquitectura utilizada en particular, la elección del modelo cliente-servidor se basa en dos características del sistema que se deben respetar:

- La Ley de Propiedad Intelectual. Al almacenar los contenidos en el servidor del avión y asegurar que no se van a almacenar en los dispositivos de los pasajeros, se evitan problemas con la Ley de Propiedad Intelectual. Se tomará las precauciones necesarias en las capas de seguridad que se vean convenientes (*firewall, proxy, gateway, etc.*) para asegurar la integridad e inviolabilidad de los contenidos multimedia.
- La baja capacidad de almacenamiento de determinados dispositivos. Algunos dispositivos, como teléfonos móviles o tablets antiguas, poseen poco espacio de almacenamiento de datos. Por esta razón, también resulta conveniente que los contenidos prestados a los usuarios permanezcan almacenados en los servidores de los proveedores de contenidos y que se reproduzcan en los dispositivos de forma online y sin llegar a descargarse.

Se define una arquitectura de sistemas, donde se muestran las máquinas o dispositivos implicados en el sistema y los diferentes protocolos utilizados para la comunicación entre todos ellos, y una arquitectura de componentes, donde se reflejan las diferentes tecnologías de las que se hace uso en cada máquina o dispositivo.

La figura 2.2.2 ilustra tanto la arquitectura de sistemas como la de componentes:

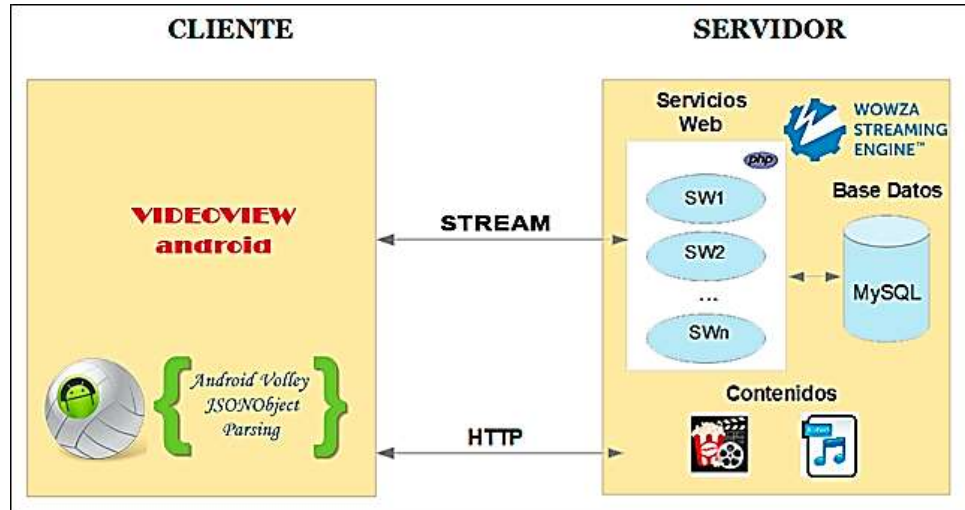


Figura 2.2.2 Arquitectura del sistema

Fuente: [Elaboración Propia]

A continuación, se comprobará los principios en los que se basa la arquitectura y las ventajas que proporciona en relación al software de aplicación (Hevia, 2010):

a) Contratos de servicio estandarizados. Con la ayuda del estándar WSDL, la interfaz de entrada/salida a los servicios ofrecidos por el servidor acerca a los contenidos queda perfectamente delimitado.

b) Servicios con bajo acoplamiento. El acoplamiento entre Amazonas y el pasajero de estos servicios es muy bajo, lo que permite modificar los servicios del servidor sin que el pasajero sufra ninguna consecuencia. Nótese que sólo ha de respetarse el protocolo estándar RTSP, será posible cambiar el servidor Apache por un servidor Tomcat y utilizar Java como lenguaje de programación para los servicios web.

c) Abstracción. Gracias al bajo acoplamiento descrito en el apartado anterior y la interfaz de entrada/salida del WSDL comentado en el inciso a), los servicios web funcionarán como una caja negra para sus consumidores: al cliente solo le importa hacerles llegar unos datos RTSP para recibir la respuesta en el mismo estándar, no le interesa cómo esté implementado interiormente el servicio.

d) Reusabilidad. Este concepto hace referencia a crear los servicios de forma que entre sí se puedan reutilizar. De momento, no es el caso este proyecto aplicativo, pero es una ventaja de la que puede beneficiarse en el futuro.

e) Autonomía. Los servicios del software a realizar tendrán un alto grado de control sobre el entorno en el que se ejecutan y sobre la lógica que implementan, como lo demuestra el hecho de que en la propia función que realiza el servicio se ejecutan las llamadas requeridas a la base de datos para ofrecer el servicio.

f) Sin estado. Los servicios de la aplicación no necesitan mayor cantidad de información y ésta le llega a través de los parámetros de entrada enviados por el usuario. Por tanto, la escalabilidad del sistema no se ve comprometida.

g) Capacidad de descubrimiento. Aunque no se utiliza el estándar UDDI, los servicios web poseerán metadatos en su cabecera para hacerse visibles a los usuarios.

h) Composición. Este concepto define la capacidad de un servicio para formar parte de un servicio más complejo. Como se ha indicado, los servicios del sistema no reutilizará otros servicios ya creados, pero podrían hacerlo. Del mismo modo, se podrían construir servicios de más alto nivel a partir de los ya existentes.

i) Interoperabilidad. No es aplicable a este proyecto, ya que todos los servicios están desarrollados en un mismo lenguaje (PHP) y en un mismo servidor (Apache).

2.3 Tecnologías de software

2.3.1 Android

Android es un sistema operativo basado en el núcleo Linux. Fue diseñado principalmente para dispositivos móviles con pantalla táctil, como teléfonos inteligentes, tablets, o tabléfonos; y también para relojes inteligentes, televisores y automóviles. Inicialmente fue desarrollado por Android Inc., empresa que Google respaldó económicamente y más tarde, en 2005, la compró. El primer móvil con el

sistema operativo Android fue el HTC *Dream* y se vendió en octubre de 2008. Actualmente, los dispositivos Android se venden más que los sistemas iOS y Windows Phone juntos (Ben, 2005).

La aplicación está desarrollada en este sistema operativo, aunque cabe la posibilidad de que en un futuro se pueda adaptar a otros (iOS y Windows Phone). Además, para la implementación se usará como editor Android Studio, entorno desarrollado para la plataforma Android. Está basado en el software IntelliJ IDEA de JetBrains, es publicado de forma gratuita a través de la Licencia Apache 2.0, y está disponible para las plataformas Microsoft Windows, Mac OS X y GNU/Linux (Ben, 2005).

2.3.2 MVC

El patrón de arquitectura MVC (Modelo Vista Controlador) es un patrón que define la organización independiente del Modelo (Objetos de Negocio), la Vista (interfaz con el usuario u otro sistema) y el Controlador (controlador del *workflow* de la aplicación). De esta forma, se divide el sistema en tres capas donde se tiene la encapsulación de los datos, la interfaz o vista por otro lado y por último la lógica interna o controlador (Ver figura 2.3.3).

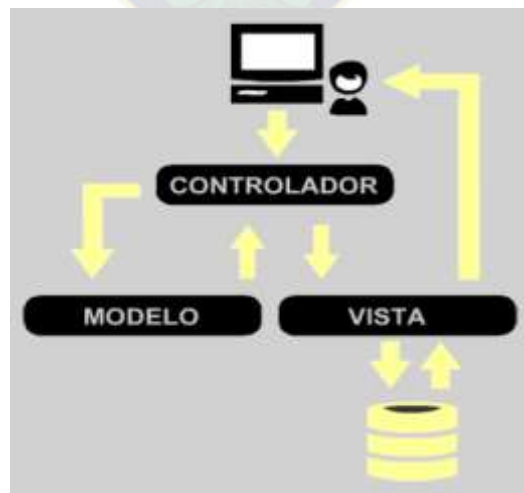


Figura 2.3.3 Modelo Vista Controlador
Fuente: (Ruiz, 2013)

2.3.3 InVision

Es un programa para prototipar apps muy parecido a Marvel, aunque más potente en algunos aspectos. Es parecido a Marvel en el sentido que Marvel parte de la idea que el usuario suba sus pantallas ya diseñadas y a partir de ahí poder añadirle diferentes interactividades mediante *gestures* y diferentes transiciones entre pantallas. Así, el funcionamiento básico de InVision es que el usuario diseñe previamente unas pantallas con otro programa para hacer prototipos, luego suba esos prototipos a la plataforma y empiece a añadirle interactividades de manera que esas pantallas queden entrelazadas entre sí.

InVision, además, ofrece muchas funcionalidades relacionadas con trabajar en equipo. En este sentido, se puede considerar como una herramienta de gestión de proyectos. Como los prototipos son una genial herramienta de comunicación entre un equipo y/o entre diseñador-cliente, contar con un programa de prototipado que facilite la comunicación entre los miembros de este equipo y con el cliente es una gran idea.

Ofrece la posibilidad de hacer una presentación en vivo y en directo desde InVision con la herramienta “*LiveShare*”, que permite convertir la pantalla en una pizarra para que se pueda anotar cualquier cosa que surja durante la reunión. Se puede también enviar un enlace con el prototipo a nuestros clientes y equipo para que puedan interactuar con el prototipo (Llensa, 2015).

2.3.4 Visual Paradigm

Visual Paradigm para UML es una herramienta para desarrollo de aplicaciones utilizando modelado UML ideal para Ingenieros de Software, Analistas de Sistemas y Arquitectos de sistemas que están interesados en construcción de sistemas a gran escala y necesitan confiabilidad y estabilidad en el desarrollo orientado a objetos (Morente, 2017).

2.3.5 Eclipse

Eclipse es un entorno de desarrollo compuesto por herramientas de programación de código abierto multiplataforma. Se le añaden plugins para programar en Java por ejemplo (*Java Development Tools*, JDT), o, en el caso del presente proyecto, Android (*Android Development Tools*, ADT).

En concreto se utilizará el Bundle ADT, que contiene los componentes esenciales de desarrollo Android. Diseñado para ampliar las características de Eclipse y permitir configurar y codificar nuevos proyectos Android rápidamente (Giraldo, 2014).

2.3.6 Android SDK

El Android SDK (Android Software Development Kit, contiene herramientas necesarias para y gratuitas para el desarrollo Android en Eclipse. Combinado con el Bundle ADT antes mencionado forman la combinación perfecta para este tipo de desarrollo.

El SDK contiene emuladores y un depurador bastante potente. Además incorpora la herramienta Manager, que permite descargar e incorporar librerías de la versión deseada al proyecto, así como todo tipo de documentación deseada.

2.3.7 JSON

(*JavaScript Object Notation* - Notación de Objetos de JavaScript) es un formato ligero de intercambio de datos. Leerlo y escribirlo es simple para humanos, mientras que para las máquinas es simple interpretarlo y generarlo. Está basado en un subconjunto del Lenguaje de Programación JavaScript, *Standard ECMA-262 3rd Edition* - Diciembre 1999. JSON es un formato de texto que es completamente independiente del lenguaje pero utiliza convenciones que son ampliamente conocidos por los programadores de la familia de lenguajes C, incluyendo C, C++, C#, Java, JavaScript, Perl, Python, y muchos otros. Estas propiedades hacen que JSON sea un lenguaje ideal para el intercambio de datos.

JSON está constituido por dos estructuras:

Una colección de pares de nombre/valor. En varios lenguajes esto es conocido como un objeto, registro, estructura, diccionario, tabla hash, lista de claves o un arreglo asociativo. (Standard, s.f.)

Una lista ordenada de valores. En la mayoría de los lenguajes, esto se implementa como arreglos, vectores, listas o secuencias.

Estas son estructuras universales; virtualmente todos los lenguajes de programación las soportan de una forma u otra. Es razonable que un formato de intercambio de datos que es independiente del lenguaje de programación se base en estas estructuras.

2.3.8 Volley Librería de Google

Es una librería que se puede integrar a nuestro proyecto Android para hacer más fáciles y rápidas las operaciones de red como: Descarga de imágenes, peticiones o envío de datos entre cliente y servidor (Arroyo, 2014).

Volley automáticamente programa todas las solicitudes de red. Significa que Volley se hace cargo de todas las solicitudes de red que su aplicación ejecuta para ir a buscar la respuesta o la imagen de la web.

Volley ofrece disco transparente y el almacenamiento en caché de memoria.

Volley ofrece una potente API solicitud de cancelación. Esto significa que usted puede cancelar la solicitud o puede establecer bloques o ámbitos de peticiones para cancelar.

Volley proporciona potentes capacidades de personalización.

Volley proporciona herramientas de depuración y rastreo.

2.3.9 PHP

El lenguaje PHP (*HyperText Preprocessor*) es un lenguaje de código abierto normalmente dedicado al desarrollo web y que puede ser incrustado en lenguaje

HTML. Utilizado también en servidores para gestionar llamadas http, lo que es perfecto para alcanzar los objetivos de este proyecto (Gomez A. , 2014).

2.3.10 GenyMotion

Genymotion es un emulador de Android que aprovecha la arquitectura x86 para dar una forma fluida y rápida a los distintos dispositivos Android que es capaz de emular. Bastante más rápido que el emulador nativo de Android, además es compatible con Windows y Linux, lo que será necesario para el proyecto, pues habrá fase de codificación y pruebas en Linux, con el fin de probar los scripts php del servidor en localhost, como se explicará más adelante (Genymotion, s.f.).

Posee una interfaz simple, capaz de soportar distintas funcionalidades, sin olvidar su principal objetivo, los desarrolladores.

2.3.11 Wowza Streaming Engine

Wowza Streaming Engine (conocida como Wowza Media Server en versiones anteriores a la versión 4) es un software de servidor de medios de *streaming* unificado desarrollado por Wowza Media Systems. El servidor se utiliza para la transmisión en directo y bajo demanda de vídeo, audio y aplicaciones RIA (*Rich Internet Applications*) a través de redes IP. El servidor es una aplicación desarrollada en Java para los siguientes sistemas operativos: Linux, Mac OS X, Solaris, Unix y Windows (Wowza, s.f.).

Puede transmitir a múltiples tipos de clientes y dispositivos de reproducción de forma simultánea, incluyendo el reproductor de Adobe Flash, Microsoft Silverlight player, Apple QuickTime Player y dispositivos iOS (*iPad, iPhone, iPod Touch*), teléfonos móviles 3GPP (Android, BlackBerry OS, Symbian, etc), IPTV set-top boxes (Amino, Enseo, Roku, Streamit y otros), y las consolas de juegos como Wii, Xbox y PS3.

Es compatible con protocolos de transmisión estándar. En el lado de reproducción, estos incluyen RTMP (y las variantes RTMPS, RTMPT, RTMPE, RTMPTE), HDS,

HLS, MPEG DASH, RTSP, *Smooth Streaming*, y MPEG-TS (*unicast* y *multicast*). En el lado de retransmisión en directo el servidor Tecnología puede retransmitir vídeo y audio a través de RTP, RTSP, RTMP, MPEG-TS (*unicast* y *multicast*). También es compatible con los flujos de entrada a través de los protocolos RTSP y WOWZ de dispositivos Android y iOS móviles que funcionan con la aplicación de codificación móvil Wowza GoCoder.

Para el *streaming* bajo demanda, *Wowza Streaming Engine* puede transmitir varios tipos de archivos de audio y vídeo. Tipos de archivo admitidos incluyen MP4 (contenedor QuickTime - Mp4, .f4v, mov, m4a, m4v, .mp4a, .mp4v, .3gp, y .3g2), FLV (Flash Video - flv), y el contenido MP3 (Mp3).

2.4 Componentes de una aplicación android

Una aplicación en Android está compuesta por varios componentes que permiten estructurar de manera lógica el funcionamiento integral de la aplicación. Estos componentes se interrelacionan de manera conjunta para obtener el correcto funcionamiento de la aplicación. A continuación, se describe de manera breve cada uno de los componentes (Revelo, 2014):

2.4.1 Actividades

Una actividad es la representación visual e interactiva de una aplicación. Esta se puede interpretar como una máquina de estados que está permanentemente pendiente de las acciones del usuario. Aunque el programador no puede controlar la forma en que se iniciará, si puede decidir qué acciones se ejecutarán en cada estado.

Las actividades son representadas por la clase *Activity*, y en ella se encuentran todos los métodos que representan su ciclo de vida. Cada transición entre estados representa un método de retrolamada (*callback*) sobre la actividad. La figura 2.4.4 representa las respectivas transiciones entre estados a través de estos métodos:

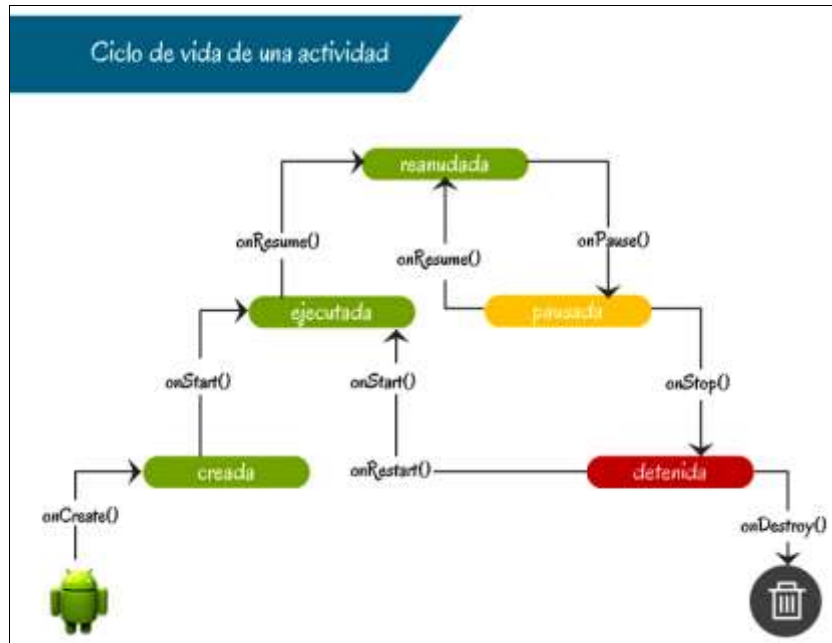


Figura 2.4.4 Ciclo de vida de una actividad en Android

Fuente: (Revelo, 2014)

- **Creación:** Cuando el usuario presiona sobre el icono de la aplicación en su dispositivo, se dispara el método `onCreate()` es ejecutado inmediatamente para cargar el *layout* de la actividad principal en memoria.
- **Ejecución-Reanudación:** Después de haber sido cargada la actividad se hace la transición hacia su ejecución de manera secuencial a través de los métodos `onStart()` y `onResume()`. Aunque `onStart()` hace visible la actividad, es `onResume()` quien le transfiere el foco para que interactúe con el usuario.
- **Pausa:** Una actividad está en pausa cuando se encuentra parcialmente visible en la pantalla. Un ejemplo sería cuando se abren diálogos que toman el foco superponiéndose a la actividad. El método llamado para la transición hacia la pausa es `onPause()`.
- **Detención:** Una actividad está detenida cuando no es visible en la pantalla, pero aún se encuentra en memoria y en cualquier momento puede ser reanudada. Cuando una aplicación es enviada a segundo plano se ejecuta el método

onStop(). Al reanudar la actividad, se pasa por el método onRestart() hasta llegar al estado de ejecución y luego al de reanudación.

- **Destrucción:** Cuando la actividad ya no existe en memoria se encuentra en estado de destrucción. Antes de pasar a destruir la aplicación se ejecuta el método onDestroy(). Es común que la mayoría de actividades no implementen este método, a menos que deban destruir procesos en segundo plano (como servicios).

Aunque la aplicación puede tener varias actividades en su estructura, es necesario definir una actividad principal. Para hacerlo se debe especificar en el AndroidManifest un componente `<activity>`, con un componente hijo `<intent-filter>`. Dentro de este componente se deben declarar dos nuevos componentes, `<action>` y `<category>`. El primero tendrá asignado el elemento enumerado MAIN y el segundo, el elemento LAUNCHER.

2.4.2 Intents

Un Intent es el elemento básico de comunicación entre los distintos componentes Android. Se pueden entender como los mensajes o peticiones que son enviados entre los distintos componentes de una aplicación o entre distintas aplicaciones para comunicarse entre sí y compartir datos. Estos se pueden asemejar a los links entre páginas web.

Los Intents pueden ser implícitos o explícitos. Un ejemplo de Intent implícito se presenta cuando se desea compartir un sitio web en alguna red social. En ese momento, Android ofrece una lista para elegir que aplicación abrir entre todas las aplicaciones sociales disponibles que están instaladas en el teléfono, en este caso no se especificó una aplicación, simplemente se mostraron todas las aplicaciones que podrían responder a este tipo de acción. Contrario a esto, un Intent explícito se da cuando se invoca una actividad con un tipo específico de clase para una acción determinada.

2.4.3 Servicios

Un servicio es una entidad que ejecuta instrucciones en segundo plano sin que el usuario lo note en la interfaz. Son muy utilizados para realizar acciones de larga duración mientras las actividades muestran otro tipo de información. Por ejemplo, guardar información en una base de datos, escuchar música mientras se ejecuta la aplicación, administrar conexiones de red, etc. Los Servicios extienden de la clase base *Service* y no son afectados por las transiciones en el ciclo de vida de una actividad. (Revelo, 2014)

Debido a que las actividades tienden por defecto a ejecutarse en un mismo hilo llamado *UI thread*, esto hace que una instrucción no se pueda ejecutar hasta que se termine el proceso actual. Es por esta razón que se requieren los servicios, para permitir ejecutar instrucciones de larga duración en otro hilo simultáneo y así no alterar las respuestas de la interfaz.

2.4.4 Content providers

Un *Content Provider* es una interfaz que permite intercambiar información persistente y estructurada entre dos aplicaciones. Cada aplicación tiene protegida su información, de modo que están aisladas de los contextos de otras aplicaciones. Aunque los *Intents* permiten intercambiar pequeños datos entre aplicaciones, un *Content Provider* extiende esta funcionalidad para que un determinado grupo de datos esté disponible para distintas aplicaciones.

Los *Content Providers* extienden la clase *ContentProvider* para implementar los métodos de la interfaz, las aplicaciones no acceden a este interfaz directamente, sino que lo hacen a través de la clase *ContentResolver*. Este sistema permite acceder a datos almacenados en el sistema de ficheros, bases de datos *SQLite* o cualquier otra fuente de datos a través de un sistema unificado.

Los Content Providers son excelentes para trabajar en segundo plano cargando información. Esta ventaja permite proporcionar una excelente experiencia al usuario debido a que no habrá interferencia entre las consultas a la base de datos y las respuestas de la interfaz (Revelo, 2014).

2.5 La tecnología Streaming

El *streaming* simplemente es la tecnología que permite ver un archivo de audio o video directamente desde internet en una página o aplicación móvil sin descargarlo previamente a nuestro dispositivo. Se visualiza a medida que va descargando al PC, Tableta o *Smartphone*.

Una transmisión de *streaming* nunca queda almacenada en el equipo del usuario, razón por la cual la hace un poco más segura para evitar plagio o que capturen el archivo de audio/video.

Además, permite ver u oír transmisiones en vivo y en directo a través de reproductores específicos o en una página web a través de flash player o HTML5. (Cehis, 2016)

2.5.1 Componentes

Se debe contar con equipos (hardware) que permitan reproducir el video desde el servido y enviarla al cliente de *streaming* para que se pueda visualizar desde Wi-Fi

Un computador

Para que el Servidor comience a reproducir es necesario contar con un equipo PC con ciertas características.

Procesador Intel Core i5 doble nucleo 2,4Ghz o superior. Recomendado Intel Core i7.

4GB Memoria ram, recomendado 8GB o más.

20GB Disco Duro. En caso de guardar copia de la transmisión.

Sistema operativo Windows 7 o superior.

Conexión a Wi-Fi

Es el factor más importante que define la calidad de la imagen o audio que experimentara el cliente. Se requiere una conexión *wifi* en el punto donde se originará la señal. Dicha conexión debe cumplir unos requerimientos mínimos.

Se sugiere que para una transmisión de audio se cuente con una conexión no inferior a 1Mbps de subida (UPLOAD). En vídeo se recomienda 3Mbps de UPLOAD para una buena calidad de imagen.

2.5.2 Protocolo RTSP

RTSP (*Real Time Streaming Protocol*) desarrollado por la IETF y publicado en 1998 como RFC 2326, es un protocolo para el uso en sistemas de *streaming* de medios que permite a un cliente para controlar remotamente un servidor de *streaming* de medios de comunicación, la emisión de comandos VCR-como por ejemplo "obra" y "pausa", y permitir el acceso basado en el tiempo a los archivos en un servidor.

Algunos servidores RTSP uso de RTP como protocolo de transporte para el audio real de datos de vídeo. Muchos servidores RTSP uso de IDT de propiedad de RealNetworks como el protocolo de transporte.

RTSP puede ser transportado a través de UDP o TCP (RTSPU y protocolos RTSPT).

El puerto por defecto para RTSP es de 554. (AmericanDominios, s.f.)

2.6 Costos y/ o beneficios

2.6.1 Método de COCOMO II

Hoy en día las compañías de software desarrollan diferentes programas de software en paralelo, que es una tarea muy compleja. Los responsables del proyecto gestionan los diferentes procesos de desarrollo de software basados en restricciones como el tiempo, el costo y el número de personal.

Calcular el tiempo, el costo y el número de personal es un trabajo muy tedioso para los directores de proyectos de las empresas de software en las primeras etapas de la planificación y seguimiento.

COCOMO es uno de los mejores modelos para estimar el costo y el tiempo en meses-persona de un proyecto de software.

A continuación, se describe una visión general de los modelos COCOMO que incluye extensiones y modelos independientes, y describe las metodologías subyacentes y la lógica detrás de los modelos y cómo pueden ser utilizados en conjunto para apoyar grandes necesidades de estimación del sistema de software. Concluye con una discusión para unificar estos diversos modelos en un proceso más simplificado que derivará en una nueva metodología completa y fácil de usar (Gomez, Lopez, Migiani, & Itazu).

2.6.2 Estimación con COCOMO II

La estructura de COCOMO II, como se visualiza en la figura 2.6.5 se basa en modelos que asumen que se progresa a lo largo de un desarrollo de tipo espiral para consolidar los requisitos y la arquitectura, reduciendo el riesgo; tales modelos son:

- Modelo de Composición de Aplicaciones.
- Modelo de Diseño Temprano.
- Modelo de Arquitectura Tardía.



Figura 2.6.5 Arquitectura de COCOMO II
Fuente: (Gomez, Lopez, Migiani, & Itazu)

COCOMO II utiliza variables establecidas en función de una medida de cinco factores de escala:

PREC Precedencia.

FLEX Flexibilidad de desarrollo.

RESL Resolución de Arquitectura / Riesgos.

TEAM Cohesión de equipo.

PMAT Madurez del proceso.

2.6.3 Modelo Post-Arquitectura

Para realizar las estimaciones COCOMO II utiliza como medida puntos de objeto, puntos de función o líneas de código, basándose en el diseño lógico del sistema. COCOMO II posee 17 multiplicadores de costos, cada uno de los cuales debe ser estimado:

RELY Fiabilidad.

DATA Tamaño de la Base de Datos.

CPLX Complejidad.

RUSE Reutilización requerida.

DOCU Documentación.

TIME Restricción tiempo de ejecución.

STOR Restricción de almacenamiento principal.

PVOL Volatilidad plataforma.

ACAP Capacidad del analista.

PCAP Capacidad del programador.

AEXP Experiencia de aplicaciones.

PEXP Experiencia plataforma.

LTEX Experiencia del lenguaje y herramienta.

PCON Continuidad del personal.

TOOL Uso de herramientas software.

SITE Desarrollo Multi-lugar.

SCED Planificación requerida.

Una característica importante a destacar es su modelo de reutilización y sus características de auto calibración.

Como se ve, muchos de sus parámetros de configuración son subjetivos, por lo tanto la exactitud de la estimación depende en gran medida de la experiencia de la persona que la realiza, además es muy importante la cantidad de proyectos anteriores (con

características similares al nuevo proyecto) porque ayudaría a obtener datos más precisos y partir de una base sólida.

COCOMO II pasó a ser una familia de modelos de productividad, estimación y toma de decisiones.

Algunos de los modelos se consideran en desarrollo, es decir que requieren todavía estudios para validarlos y calibrarlos.

Es el modelo de estimación más detallado y se aplica cuando la arquitectura del proyecto está completamente definida. Este modelo se aplica durante el desarrollo y mantenimiento de productos de software incluidos en las áreas de Sistemas Integrados, Infraestructura y Generadores de Aplicaciones (Gomez, Lopez, Migiani, & Itazu).

El esfuerzo nominal se ajusta usando 17 factores multiplicadores de esfuerzo. El mayor número de multiplicadores permite analizar con más exactitud el conocimiento disponible en las últimas etapas de desarrollo, ajustando el modelo de tal forma que refleje fielmente el producto de software bajo desarrollo. La fórmula para el cálculo del esfuerzo es la siguiente:

$$PM_{estimado} = PM_{nominal} \times \prod_{i=1}^{17} EM_i$$

2.7 Proceso de calidad del desarrollo de software

2.7.1 ISO 9126

La organización internacional para la estandarización ISO fue creada en 1946 con el fin de facilitar el comercio internacional, la coordinación internacional y la unificación de estándares industriales promoviendo una serie de simples patrones de estándares que deberán ser reconocidos y respetados. ISO 9126 fue originalmente desarrollado en

1991 para proporcionar un esquema para la evaluación de la calidad del software y así refinarlo en un periodo de 10 años.

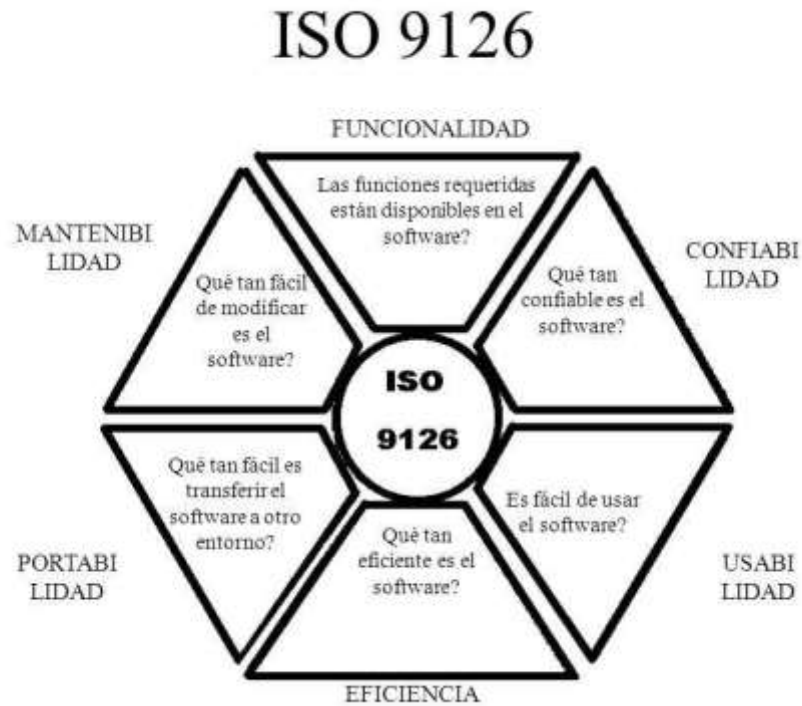


Figura 2.7.1 Etapas de la ISO 9126

Muchos estudios criticaron la ISO 9126 por no recomendar requerimientos específicos de calidad en vez de definir un esquema general para la evaluación de calidad del software. Se cree que este es de hecho una de las fortalezas y así es más adaptable y puede ser usado a través de varios sistemas incluso sistemas de aprendizaje virtual. El producto original definió seis características del producto (ver Figura 2.7.1) estas características son divididas en un número de sub-características.

C1. Funcionalidad

En este grupo se conjunta una serie de atributos que permiten calificar si un producto de software maneja en forma adecuada el conjunto de funciones que satisfagan las necesidades para las cuales fue diseñado. Para este propósito se establecen los siguientes atributos:

- Adecuación. Se enfoca a evaluar si el software cuenta con un conjunto de funciones apropiadas para efectuar las tareas que fueron especificadas en su definición.
- Exactitud. Este atributo permite evaluar si el software presenta resultados o efectos acordes a las necesidades para las cuales fue creado.
- Interoperabilidad. Permite evaluar la habilidad del software de interactuar con otros sistemas previamente especificados.
- Conformidad. Evalúa si el software se adhiere a estándares, convenciones o regulaciones en leyes y prescripciones similares.
- Seguridad. Se refiere a la habilidad de prevenir el acceso no autorizado, ya sea accidental o premeditado, a los programas y datos.

C2. Confiabilidad

Aquí se agrupan un conjunto de atributos que se refieren a la capacidad del software de mantener su nivel de ejecución bajo condiciones normales en un periodo de tiempo establecido. Las subcaracterísticas que el estándar sugiere son:

- Nivel de Madurez. Permite medir la frecuencia de falla por errores en el software.
- Tolerancia a fallas. Se refiere a la habilidad de mantener un nivel específico de funcionamiento en caso de fallas del software o de cometer infracciones de su interfaz específica.
- Recuperación. Se refiere a la capacidad de restablecer el nivel de operación y recobrar los datos que hayan sido afectados directamente por una falla, así como al tiempo y el esfuerzo necesarios para lograrlo.

C3. Usabilidad

Consiste de un conjunto de atributos que permiten evaluar el esfuerzo necesario que deberá invertir el usuario para utilizar el sistema.

- **Comprensibilidad.** Se refiere al esfuerzo requerido por los usuarios para reconocer la estructura lógica del sistema y los conceptos relativos a la aplicación del software.
- **Facilidad de Aprender.** Establece atributos del software relativos al esfuerzo que los usuarios deben hacer para aprender a usar la aplicación.
- **Operabilidad.** Agrupa los conceptos que evalúan la operación y el control del sistema.

C4. Eficiencia

Esta característica permite evaluar la relación entre el nivel de funcionamiento del software y la cantidad de recursos usados. Los aspectos a evaluar son:

- **Comportamiento con respecto al Tiempo.** Atributos del software relativos a los tiempos de respuesta y de procesamiento de los datos.
- **Comportamiento con respecto a Recursos.** Atributos del software relativos a la cantidad de recursos usados y la duración de su uso en la realización de sus funciones.

C5. Mantenibilidad

Se refiere a los atributos que permiten medir el esfuerzo necesario para realizar modificaciones al software, ya sea por la corrección de errores o por el incremento de funcionalidad. En este caso, se tienen los siguientes factores:

- **Capacidad de análisis.** Relativo al esfuerzo necesario para diagnosticar las deficiencias o causas de fallas, o para identificar las partes que deberán ser modificadas.
- **Capacidad de modificación.** Mide el esfuerzo necesario para modificar aspectos del software, remover fallas o adaptar el software para que funcione en un ambiente diferente.
- **Estabilidad.** Permite evaluar los riesgos de efectos inesperados debidos a las modificaciones realizadas al software.

- **Facilidad de Prueba.** Se refiere al esfuerzo necesario para validar el software una vez que fue modificado.

C6. Portatilidad

En este caso, se refiere a la habilidad del software de ser transferido de un ambiente a otro, y considera los siguientes aspectos:

- **Adaptabilidad.** Evalúa la oportunidad para adaptar el software a diferentes ambientes sin necesidad de aplicarle modificaciones.
- **Facilidad de Instalación.** Es el esfuerzo necesario para instalar el software en un ambiente determinado.
- **Conformidad.** Permite evaluar si el software se adhiere a estándares o convenciones relativas a portatilidad.
- **Capacidad de reemplazo.** Se refiere a la oportunidad y el esfuerzo usado en sustituir el software por otro producto con funciones similares.

2.7.2 Buenas prácticas W3C (*World Wide Web Consortium*)

Para poner un poco de orden y criterio, el W3C ha lanzado *Mobile Web Best Practices* 1.0 o algo así como las "Buenas Prácticas para el Desarrollo Web en Dispositivos Móviles". W3C quiere ofrecer un conjunto de pautas que ayuden a proporcionar una mejor experiencia a los usuarios de dispositivos móviles. (Quevedo, 2006)

Navegabilidad

URI. Intentar que la *Uniform Resource Identifier* de entrada al sitio sea tan corta como pueda.

Barra de navegación. Ofrecer en la cabecera de la página sólo la navegación mínima necesaria.

Recursos externos. Intentar que los enlaces a recursos externos sean los mínimos posibles.

Navegación. Ofrecer mecanismos de navegación consistentes.

Equilibrio de enlaces. Valorar el tener muchos vínculos en una página y que el usuario tenga que seguir muchos enlaces hasta encontrar lo que busca.

Teclas de acceso. Proporcionar atajos de teclado (accesskeys) a los enlaces de navegación y a las funciones más usadas.

Enlaces. Identificar de forma clara el destino de cada enlace. No cambiar el formato de los enlaces a menos que sepa que el dispositivo es compatible con la modificación.

Mapas de imagen. No utilizar mapas de imagen a menos que sepa que el dispositivo los soporta. En cualquier caso, pensar en vías alternativas para poder mostrar la información.

Ventanas emergentes (Pop-Up). No utilizar ventanas emergentes. No cambiar la ventana actual sin informar al usuario.

Control del usuario

Recarga automática. No hacer que las páginas se recarguen automáticamente cada cierto tiempo, a menos que se informe al usuario de ello y se ofrezca una forma para poder detener dicha acción.

Redireccionamiento. No utilizar código que redirija automáticamente las páginas. En lugar de eso, configure el servidor para que encargue de redireccionarlas.

Pulsaciones de teclas. Intentar que las teclas que tenga que pulsar el usuario sean las mínimas necesarias.

Contenido

Adecuación. Comprobar que el contenido es apropiado para su uso en un contexto móvil.

Claridad. Usar un lenguaje claro y simple.

Limitación. Limitar el contenido al que el usuario solicita.

Títulos de las páginas. Cree títulos de páginas cortos pero descriptivos.

Consistencia. Cerciorarse de que el contenido es consistente cuando se accede desde diferentes dispositivos.

Jerarquía semántica. Verificar que el contenido más importante de la página aparece antes que el contenido secundario.

Imágenes

No usar imágenes para lograr el posicionamiento de elementos, crear espacios, etc.

Tamaño de imágenes. No utilizar imágenes que no puedan mostrarse en el dispositivo. Evite imágenes grandes o detalles de alta resolución. Definir el tamaño de las imágenes en el marcado si poseen una medida específica.

Imágenes de fondo. Cuando se utilice imágenes de fondo, verificar que el contenido sigue siendo legible.

Escalado de imágenes. Si se deben escalar las imágenes a una medida específica, redimensionar desde el servidor.

Color

Uso del color. Comprobar que la información transmitida por medio del color sigue estando disponible sin él.

Contraste del color. Comprobar que los colores de primer plano y del fondo contrastan lo suficiente.

Diseño

Página limitada. Dividir la página en partes cuyos tamaños sean fáciles de utilizar en el dispositivo.

Peso limitado. Asegurar de que el peso total de las páginas es el adecuado para las limitaciones de memoria del dispositivo.

Desplazamiento. Limitar el desplazamiento de la página a una dirección, a menos que el desplazamiento secundario pueda evitarse.

Medidas. No usar medidas definidas en píxeles. No utilizar medidas absolutas en los valores de los atributos del marcado ni en los valores de las hojas de estilos.

Fuentes. No confiar en la compatibilidad de las fuentes tipográficas que se declaran en los estilos.

Capacidades. Aprovechar las capacidades del dispositivo para ofrecer una experiencia de uso óptima.

Pruebas. Hacer pruebas tanto con dispositivos reales como con emuladores.

Tecnologías y marcado

Marcado válido. Crear documentos que sean válidos con los estándares y tecnologías del W3C.

Minimice. Usar un marcado conciso y eficaz.

Marcos. No usar conjuntos de marcos.

Estructura. Usar las características del lenguaje de marcado para indicar la estructura lógica del documento. Hacer uso de un marcado semántico.

Tablas

No utilizar tablas a menos que sepa que el dispositivo es compatible con ellas.

No utilizar tablas anidadas.

No usar las tablas para la maquetación

Cuando sea posible, ofrecer alternativas a la presentación de datos tabulados.

Scripts, objetos, *applets* y plug-ins: no usar elementos incrustados en las páginas a menos que sepa que van a funcionar en el dispositivo. En cualquier caso, ofrecer alternativas para los usuarios que no puedan verlos.

Alternativas no textuales. Ofrecer alternativas textuales para cada elemento no textual.

Compatibilidad con la codificación de caracteres. Verificar que el contenido está codificado con un juego de caracteres que va a ser soportado por el dispositivo.

Codificación de caracteres. Indicar el juego de caracteres que está utilizando.

Mensajes de error. Ofrecer mensajes de error informativos y significativos, con los mecanismos de navegación necesarios para salir del error y volver a la información útil.

Cookies. No confiar en que las cookies estén siempre disponibles.

Caché. Intentar guardar en memoria la información de las respuestas HTTP.

Formato del contenido. Cuando sea posible, enviar el contenido en un formato preferido y que sepa que va a ser compatible con el dispositivo.

Incompatibilidades. Ser prudente a la hora de trabajar con implementaciones deficientes.

Hojas de estilos

Usar hojas de estilos para controlar la presentación, a menos que el dispositivo no sea compatible con ellas.

Organizar los documentos de tal forma que sean legibles sin hojas de estilos.

Intentar que las hojas de estilo sean pequeñas (en peso).

Formularios

Valores por defecto. Ofrecer valores preseleccionados por defecto y evitar la introducción libre de texto cuando sea posible.

Modo de entrada. Si el dispositivo es compatible, especificar una forma por defecto de insertar texto, idioma y/o método de introducción.

Orden de tabulación. Crear un orden lógico entre los enlaces, controles de los formularios y objetos.

Etiquetado de controles. Etiquetar adecuadamente todos los controles de los formularios y asociar explícitamente cada etiqueta a su control.

Posición de los controles. Colocar las etiquetas de tal forma que se muestren correctamente y en correspondencia con los controles de formulario a los que se refieren.

2.7.2.1 Pruebas de aceptación

Se enfocan en la aceptación de los criterios previstos en un contrato de desarrollo de software, acordado entre la fábrica de software y el cliente.

2.7.3 Técnica de prueba

Se puede evidenciar en la tabla 2.7-1 que existen varias técnicas de prueba.

Tabla 2.7-1 Agrupación y técnicas de pruebas

Agrupación	Técnicas
Técnicas de caja negra	Partición de equivalencia Análisis del valor límite Tablas de decisión Máquinas de estado finito Grafo causa efecto Prueba de dominios
Técnicas de caja blanca	Basadas en el flujo de control Basadas en el flujo de los datos Mutantes
Técnicas según quién hace la prueba	Pruebas de aceptación Pruebas alfa y beta Pruebas de usuario Pruebas en pares
Técnicas basadas en la experiencia	Prueba <i>ad hoc</i> Conjetura de errores Testing exploratorio



CAPITULO III

MARCO APLICATIVO

3.1 Introducción

Este capítulo trata sobre el análisis, diseño y desarrollo de la aplicación móvil siguiendo las normativas de la metodología Scrum y desarrollo de artefactos que serán utilizadas como línea orientadora del Proyecto de Grado.

Tabla 3.1-1 Tabla de actividades según SCRUM

SCRUM	Fase	
Análisis (Pre - Game)	Arranque	Recopilación de requerimientos Cronograma de trabajo Identificación de riesgos Iteraciones
Construcción (Game)	Análisis	Identificación de actores Historias de usuario Diagrama de caso de uso Gestión de usuarios Gestión de películas Gestión de canciones Gestión de revistas Gestión de publicidad
	Diseño	Diagrama de secuencia Diagrama de componentes Modelo conceptual Modelo de servicios Diseño de pantallas (mockup)
	Construcción	Diseño de pantallas Construcción de la base de datos física. Desarrollo de módulos
Pruebas (Post Game)	Pruebas	Pruebas de ejecución Pruebas de calidad de software

Scrum trabaja bajo una lista de tareas ordenadas denominadas Sprint (iteraciones), iterativamente el desarrollo del software va generando productos entregables.

3.2 Pre Game Etapa de análisis

3.2.1 Recopilación de requerimientos funcionales

Para la determinación de las necesidades que serán resueltas, se realizaron reuniones y así establecer el Sprint Backlog inicial, que serán los requerimientos del sistema.

Tabla 3.2-1 Recopilación de requerimientos

Id	Descripción	Prioridad
R1	Lista de películas	Media
R2	Búsqueda de películas	Media
R3	Reproducción de una película	Media
R4	Búsqueda por género de canciones	Media
R5	Búsqueda de una canción	Media
R6	Reproducción de una canción	Media
R7	Búsqueda de un libro	Media
R8	Presentación del contenido de un libro	Media
R9	Presentación del contenido de la revista de la empresa	Media
R10	Presentación de la publicidad de la empresa	Media

Para elaborar la tabla 3.2-1, se realizaron entrevistas tanto al personal como a los pasajeros de la empresa de aviación Amazonas Línea Aérea, logrando identificar diez requerimientos, asignando prioridad a cada una de ellas para su posterior desarrollo, esta lista constituye el Sprint Backlog inicial, que se constituirá como los requerimientos del sistema.

Adicionalmente se identificaron otras necesidades propias para el mejor funcionamiento de la aplicación móvil.

Entonces a partir de la recopilación de los requerimientos, las tablas siguientes muestran en detalle una descripción de cada requerimiento.

Dado que se sigue la metodología Scrum, los requisitos están definidos a través de historias de usuario. Las historias de usuario se constituyen como los requisitos funcionales y debe responder principalmente a tres preguntas:

¿Quién se beneficia?

¿Qué se quiere?

¿Cuál es el beneficio?

Además, estas historias son pequeñas e independientes de unas a otras lo que permite desarrollar productos funcionales en periodos de tiempo reducidos. Otra de las características del Scrum, es que en cada iteración del proyecto los requerimientos pueden sufrir modificaciones, se agreguen o se quiten más historias según la evolución del proyecto; cambios que se ajustan de manera inmediata por el equipo de desarrollo.

Para la formulación de las historias de usuario, se utilizará la tabla 3.2-2:

Tabla 3.2-2 Plantilla de Historia de usuario

#. Nombre de la historia	
Como [usuario] quiero [obtener algo] para [conseguir algo]	
Prioridad:	1 - 10 (min - max)
Criterio de aceptación	
Criterio que hace que se cumpla la historia de usuario correctamente.	

Esta historia refleja la necesidad del pasajero de tener a disposición un listado de los géneros de películas que la aplicación ofrece según criterios de selección, según tabla 3.2-3

Tabla 3.2-3 Búsqueda de género de películas

1. Busco genero de películas	
Como Pasajero quiero ver los géneros de películas para escoger el que más me gusta	
Prioridad:	10
Criterio de aceptación	
La aplicación mostrara un listado de todos los géneros de películas	

Una vez que el pasajero selecciona el género de película, la aplicación debe ser capaz de ofrecer un listado de películas relacionadas al género seleccionado, según la tabla 3.2-4

Tabla 3.2-4 Búsqueda de películas

2. Busco una película	
Como Pasajero quiero ver las películas de un género para escoger una película	
Prioridad:	10
Criterio de aceptación	
La aplicación mostrara un listado de todas las películas de un genero	

Finalmente y luego de haber seleccionado la película, mediante streaming la aplicación reproducirá la película en el dispositivo móvil del pasajero, según tabla 3.2-5

Tabla 3.2-5 Reproducción de películas

3. Reproduzco una película	
Como Pasajero quiero reproducir una película para entretenerme en el vuelo	
Prioridad:	10
Criterio de aceptación	
La aplicación reproducirá la película seleccionada	

Al igual que las películas, otra oferta que presenta la aplicación se relaciona con los temas musicales a disposición del pasajero. Entonces de igual forma se ofrece al pasajero seleccionar el género de temas musicales según tabla 3.2-6

Tabla 3.2-6 Búsqueda de género de canciones

4. Busco genero de canciones	
Como Pasajero quiero ver los géneros de canciones para escoger el que más me gusta	
Prioridad:	10
Criterio de aceptación	
La aplicación mostrara un listado de todos los géneros de canciones	

Luego de haber seleccionado el género, la aplicación presenta un listado de canciones que pertenecen al género seleccionado. La cantidad de canciones ofertadas a los pasajeros estará en directa dependencia de la capacidad de almacenamiento del servidor, según tabla 3.2-7

Tabla 3.2-7 Búsqueda de canciones

5. Busco una canción	
Como Pasajero quiero ver las canciones de un género para escoger una canción	
Prioridad:	10
Criterio de aceptación	
La aplicación mostrara un listado de todas las canciones de un genero	

Y finalmente la aplicación reproduce la canción de forma inmediata en el dispositivo del pasajero a través de la intranet según tabla 3.2-8

Tabla 3.2-8 Reproducción de canción

6. Reproduzco una canción	
Como Pasajero quiero reproducir una canción para entretenerme en el vuelo	
Prioridad:	10
Criterio de aceptación	
La aplicación reproducirá la canción seleccionada	

Otro sector apreciado por los pasajeros de la línea aérea Amazonas, es el de la lectura, por lo que la aplicación ofrece un conjunto de e-books que están a consideración de los usuarios. Entonces mediante una lista se ofrece los e-books almacenados en el servidor de la nave. Este requerimiento se formula en la tabla 3.2-9

Tabla 3.2-9 Búsqueda de libro

7. Busco un libro	
Como Pasajero quiero ver el listado de libros para escoger una libro específico	
Prioridad:	10
Criterio de aceptación	
La aplicación mostrara un listado de todos los libros	

Una vez seleccionado el libro acorde al gusto del pasajero, la aplicación despliega el contenido que es recuperado desde el servidor de la nave, según tabla 3.2-10

Tabla 3.2-10 Visualización del contenido de un libro

8. Veo un libro	
Como Pasajero quiero ver un libro para entretenerme en el vuelo	
Prioridad:	10
Criterio de aceptación	
La aplicación mostrara el libro seleccionado	

Muchas veces el pasajero solamente desea ver la revista de la empresa donde están algunas ofertas de temporadas y servicios que ofrece. Entonces la aplicación debe estar en condiciones de mostrar una revista digital con la información y campañas de la empresa.

Por razones de velocidad en la carga de contenidos, los contenidos deben ser sometidos a compresión de datos y otros procesos de disminución de peso, tareas que realizarán los administradores, ver tabla 3.2-11

Tabla 3.2-11 Lectura de revista de la empresa

9. Veo la revista de la empresa	
Como Pasajero quiero ver la revista para informarme de sus viajes	
Prioridad:	10
Criterio de aceptación	
La aplicación mostrara la revista de la empresa	

Generalmente el público femenino que requiere los servicios de la empresa se enfoca en las campañas publicitarias que ofrece la línea aérea, entonces la aplicación también ofrece a este sector un módulo de despliegue con la publicidad de diversas empresas que realizan intercambios comerciales con la línea aérea y otras campañas propias de la misma empresa según la tabla 3.2-12

Tabla 3.2-12 Publicidad de la empresa.

10. Veo la publicidad de la empresa	
Como Pasajero quiero ver la publicidad para entretenerme en el vuelo	
Prioridad:	10
Criterio de aceptación	
La aplicación mostrara la publicidad de la empresa	

3.2.2 Recopilación de requerimientos no funcionales

Este tipo de necesidades se constituyen como requerimientos no funcionales (R.N.F.) y no son de cumplimiento obligatorio para el logro de los objetivos ya que dependen generalmente de inversiones económicas o de mejoramiento respecto al funcionamiento del hardware requerido por la aplicación.

Para la formulación de los requerimientos no funcionales, se utilizará la tabla 3.2-13

Tabla 3.2-13 Plantilla de formulación de Requisitos no funcionales

#. Nombre del requisito no funcional
Descripción:
Descripción del requisito

En las siguientes tablas se presentan los requerimientos no funcionales identificados para el mejor desempeño de la aplicación móvil de entretenimiento.

De la interfaz, la aplicación debe presentar una interfaz sencilla y muy fácil de utilizar de forma que el pasajero se sienta “atraído” por la aplicación, como muestra la tabla 3.2-14

Tabla 3.2-14 Publicidad de la empresa.

1. La interfaz será fácil de usar.
Descripción:
La interfaz será sencilla, solo con la información necesaria en todo momento.

Del sistema operativo, la aplicación solo trabajará con aquellos dispositivos que tengan como plataforma de funcionamiento Android, tabla 3.2-15

Tabla 3.2-15 R.N.F. Sistema operativo.

2. Sistema operativo
Descripción:
La aplicación estará aplicada a todas las versiones de android

El rendimiento de la aplicación, es decir el tiempo de carga y recuperación de los contenidos debe ser mínimo, estableciendo para ello un periodo de al menor 6 segundos, ver tabla 3.2-16

Tabla 3.2-16 R.N.F. Rendimiento.

3. Rendimiento
Descripción:
La aplicación no ha de tardar más de 6 segundos en cada operación que haga el usuario a través del Wifi

Dadas las características de la aplicación, el mismo debe estar abierto a nuevas modificación y mejoras para que el tiempo de vida de la aplicación contemple el mayor tiempo de funcionamiento posible, ver tabla 3.2-17

Tabla 3.2-17 R.N.F. Escalabilidad

4. Escalable
Descripción:
Las funcionalidades del software han de poder ser modificadas sin afectar al resto de funciones.

La aplicación y dada la naturaleza de los pasajeros que abordan las naves de la empresa Amazonas, debe estar en condiciones de poder cambiar de idiomas, ya sea en español o en inglés, aunque también se considera la posibilidad de incrementar nuevos idiomas como el francés, alemán y chino mandarín; este último grupo de pasajero se va

incrementando con el transcurrir de los años desde el año 2013. (Según registro de venta de pasajes aéreos de la empresa), según tabla 3.2-18

Tabla 3.2-18 R.N.F. Idioma

5. Idioma
Descripción:
La app ha de estar traducida en inglés y español

Finalmente, la aplicación solo debe requerir para la conexión al servidor de la intranet, señal wi-fi sin conexión al servicio de Internet, según tabla 3.2-19

Tabla 3.2-19 R.N.F. Internet.

6. Internet
Descripción:
La aplicación no necesitará de internet para su funcionamiento

3.2.3 Cronograma de trabajo

El cronograma de trabajo se definió en base al ciclo de vida de la metodología SCRUM, constituidos por las tres etapas, la descripción detallada del cronograma de trabajo para el proyecto se muestra en la tabla 3.2-20

Tabla 3.2-20 Cronograma de trabajo por iteración

Id	Nombre	Inicio	Fin	Duración
1	Primera iteración	13/03/2017	29/03/2017	13
2	Segunda iteración	30/03/2017	12/04/2017	10
3	Tercera iteración	13/04/2017	28/06/2017	55

Esta planificación de forma gráfica se muestra en la figura 3.2.1

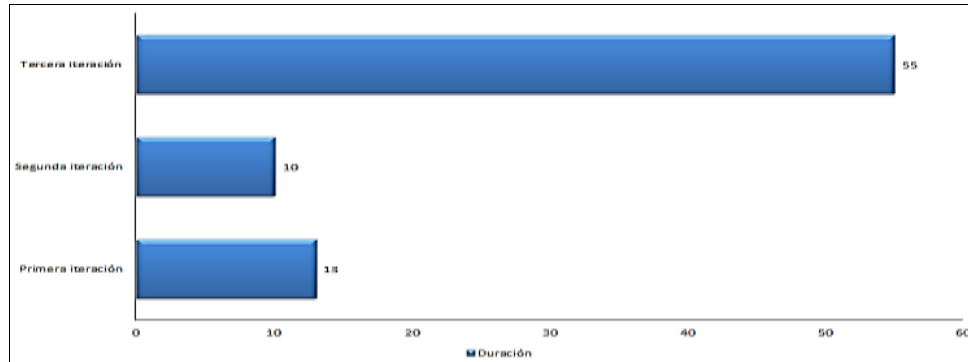


Figura 3.2.1 Planificación de Iteraciones

3.2.4 Identificación de riesgos

Un riesgo es la probabilidad de que ocurra algo adverso:

- a) Riesgo de proyecto, que afecta a la calendarización o recursos del proyecto.
- b) Riesgo de producto, que afectan a la calidad o rendimiento del software en desarrollo.
- c) Riesgo de negocio, que afectan a la organización que desarrolla o suministra el software.

Tabla 3.2-21 Análisis de riesgos

Tipo de riesgo	Riesgo	Tipo	Probabilidad	Efecto	Estrategia
a)	Incumplimiento de fechas	Proyecto	Media	Tolerable	Ajustar el cronograma.
b)	Cambios de requerimiento de cliente	Proyecto producto	Moderada	Tolerable	Realizar revisiones constantes
c)	No existe infraestructura necesaria	Proyecto	Baja	Tolerable	Solicitar equipos con anticipación.

La tabla 3.2-21 refleja los riesgos identificados que se pudieran presentar en el desarrollo del sistema de “Entretenimiento en vuelo orientado a dispositivos móviles Android”.

Se consideran tres riesgos relacionadas con el incumplimiento de entrega del producto, los cambios en los requerimientos del cliente y la falta de infraestructura y la estrategia a seguir deberá ser evaluada en cada sprint inicial, en la reunión diaria.

3.2.5 Iteraciones

Para la elaboración de las iteraciones, se planificó realizar tres iteraciones, a su vez cada iteración contempla un conjunto de tareas a ser elaboradas.

Las tablas 3.2-22, 3.2-23 y 3.2-24 muestran las actividades que se realizan por iteración.

3.2.6 Primera iteración

Durante la primera iteración se desarrollará los elementos relacionados con la pila de requerimientos identificados por los actores del sistema, establecimiento de los casos de uso de los procesos principales y diagramas de secuencia, según tabla 3.2.22

Tabla 3.2-22 Tareas de la primera iteración

		Iteración 1	Inicio 13/03/2017	Duración 13 días
Id	Tarea	Tipo	Días de trabajo	Estado
1.1	Recopilación de requerimientos	Arranque	2	Hecho
1.2	Cronograma de trabajo	Arranque	1	Hecho
1.3	Identificación de riesgos	Arranque	1	Hecho
1.4	Planificación de iteraciones	Arranque	1	Hecho
1.5	Identificación de actores	Análisis	1	Por hacer
1.6	Historias de usuario	Análisis	0.5	Por hacer
1.7	Diagrama de caso de uso general	Análisis	0.5	Por hacer
1.8	H.U. usuarios	Análisis	1	Por hacer
1.9	H.U. películas	Análisis	1	Por hacer
1.10	H.U. canciones	Análisis	1	Por hacer
1.11	H.U. libros	Análisis	1	Por hacer
1.12	H.U. revistas	Análisis	1	Por hacer
1.13	H.U. publicidad	Análisis	1	Por hacer

3.2.7 Segunda iteración

Esta segunda iteración desarrollará el análisis y el diseño de los procesos de gestión de usuarios, de proyectos y el establecimiento de la base de datos que se consideran para la elaboración del proyecto, según tabla 3.2-23

Tabla 3.2-23 Tareas de la segunda iteración

		Iteración	Inicio	Duración
		2	30/03/2017	10 días
Id	Tarea	Tipo	Días de trabajo	Estado
2.1	Diagrama de secuencia	Diseño	2	Por hacer
2.2	Diagrama de componentes	Diseño	2	Por hacer
2.3	Modelo conceptual	Diseño	2	Por hacer
2.4	Modelo de servicios	Diseño	2	Por hacer
2.5	Diseño de pantallas (mockup)	Diseño	2	Por hacer

3.2.8 Tercera iteración

La iteración está referida a la fase de construcción del proyecto de entretenimiento contemplando como primera actividad el diseño físico que serán desplegadas en la aplicación móvil, la implementación de la base de datos, el desarrollo de cada uno de los módulos planificadas en la iteración 1 (Usuarios, películas, canciones, libros, revistas y publicidad).

Se incluye también en esta iteración la realización de las pruebas de ejecución a las que será sometido el producto final bajo responsabilidad de los pasajeros de la línea aérea Amazonas S.A. y finalmente establecer la calidad del software, según tabla 3.2-24

Tabla 3.2-24 Tareas de la tercera iteración

		Iteración	Inicio	Duración
		3	22/06/2014	55 días
Id	Tarea	Tipo	Días de trabajo	Estado
3.1	Diseño de pantallas	Construcción	5	Por hacer
3.2	Construcción de la base de datos	Construcción	5	Por hacer
3.3	Desarrollo de módulos	Construcción	15	Por hacer
3.4	Pruebas de ejecución	Pruebas	20	Por hacer
3.5	Pruebas de calidad de software	Pruebas	10	Por hacer

3.3 Game

3.3.1 Identificación de roles

Para identificar a los usuarios que serán los directos beneficiarios de la aplicación, se realizó una reunión preliminar en principio con los administradores de la línea aérea Amazonas S.A., quienes expresaron que la aplicación de entretenimiento es sería de gran beneficio para los pasajeros que utilizan este medio de transporte.

Tabla 3.3-1 Identificación de roles de usuarios

Usuarios	Descripción
Técnico en sistemas (Admin)	Encargado de sistema, será encargado de tener en perfecto funcionamiento la aplicación móvil para los pasajeros. Adicionalmente está encargado de la gestión de los contenidos de la aplicación móvil. Por último está bajo su responsabilidad el funcionamiento adecuado de los equipos de hardware necesarios para que la aplicación móvil no presente problemas de acceso desde los diversos dispositivos móviles vía Wi-Fi.
Pasajero (Usuario)	Es el usuario final que ejecuta la aplicación móvil desde su dispositivo inteligente. Los requerimientos, solicitudes y sugerencias permitirán la construcción de la aplicación móvil. Puede acceder a la aplicación de entretenimiento desde la aeronave en la que se transporta, ya que cada una de las mismas cuenta con un servidor para brindar los servicios que los pasajeros soliciten.

También se realizó entrevistas a los pasajeros que frecuentan esta Línea Aérea.

Los actores serán en el sistema considerados como usuarios, mientras que los administradores serán los responsables del manejo del sistema para que se tenga un correcto funcionamiento, según tabla 3.3-1

3.3.2 Caso de uso general del sistema

A partir de la recopilación de requerimientos, lo que se constituyen como requerimientos funcionales, se presenta en este apartado el diagrama de casos de uso de forma general como una visión global de los usuarios que interactúan con el sistema.

La ventaja de la figura 3.3.1 radica en la facilidad de ser interpretado y permite mostrar al cliente, con gran rapidez lo que hace el sistema y quien está implicado en él.

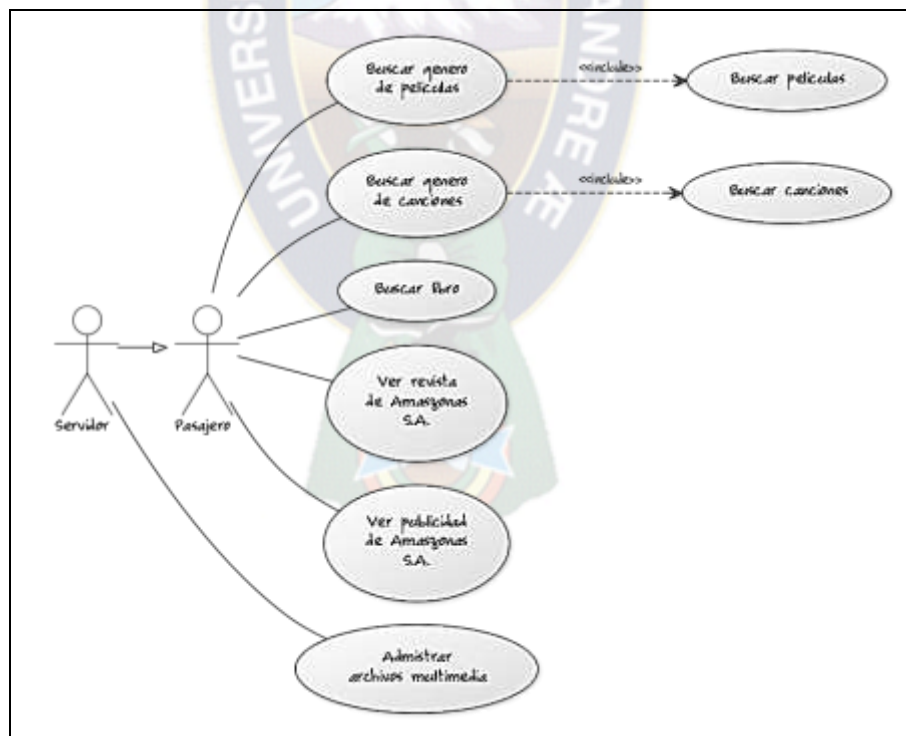


Figura 3.3.1 Caso de uso general del sistema

Fuente: [Elaboración Propia]

Un caso de uso es una descripción de las actividades que deben ser realizadas por un proceso y lograr un resultado.

En las tablas siguientes, se detalla casos de uso más relevantes para el análisis del sistema de “Entretenimiento en vuelo orientado a dispositivos móviles android” para la Línea Aérea Amazonas S.A., casos de uso presentados en el diagrama anterior y que son parte de los procesos del sistema a desarrollar incluyendo los diagramas de secuencia para cada uno de los mismos.

Para la formulación de los casos de uso, se utilizará la siguiente plantilla para mantener una misma estructura. (Ver tabla 3.3-2)

Tabla 3.3-2 Plantilla de realización de casos de uso

CU #00	
Actor:	Actor o actores implicados en el caso de uso
Precondición:	Condición a satisfacer antes de empezar el caso de uso.
Disparador:	Condición que provoca el caso de uso.
Escenario principal	
Secuencia de acciones e interacciones entre el actor y el sistema.	
Extensiones	
Alternativas y tratamiento de situaciones no normales.	

3.3.2.1 CU #1. Buscar género de película (Ver tabla 3.3-3)

Tabla 3.3-3 CU #1 Buscar género de película

CU #01 Buscar genero de películas	
Actor:	Pasajero
Precondición:	Ninguna
Disparador:	El pasajero quiere ver los géneros de películas
Escenario principal	
<ol style="list-style-type: none"> 1. El pasajero pide a la aplicación que quiere ver los géneros de películas ofrecidos por la Línea Aérea Amazonas. 2. El sistema muestra los tipos de géneros de películas disponibles. 	
Extensiones	
<p>2a. No hay conexión con el Servidor Streaming</p> <p>2a1. El sistema indica al pasajero que no hay conexión disponible.</p> <p>2a2. Se vuelve al paso 1.</p>	

El diagrama de secuencia según figura 3.3.2 corresponde a la funcionalidad de Buscar género de película, la que se encarga de realizar la búsqueda de género.

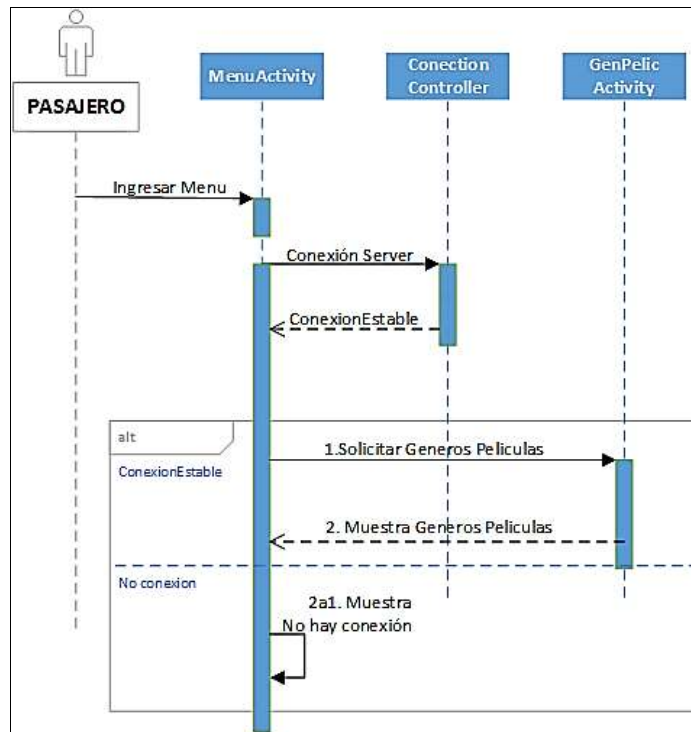


Figura 3.3.2 D.S. Buscar género de película

3.3.2.2 CU #2. Reproducir una película (ver tabla 3.3-4)

Tabla 3.3-4 CU #2 Reproducir una película

CU #02 Reproducir una película	
Actor:	Pasajero
Precondición:	CU #01
Disparador:	El pasajero quiere ver una película de su elección
Escenario principal	
<ol style="list-style-type: none"> 1. El pasajero indica a la aplicación que quiere reproducir una película. 2. El sistema reproduce la película indicada. 	
Extensiones	
2a. No hay conexión con el Servidor Streaming	

El diagrama de secuencia según figura 3.3.3 corresponde a la funcionalidad de Reproducir película, la que se encarga de realizar la reproducción de la película

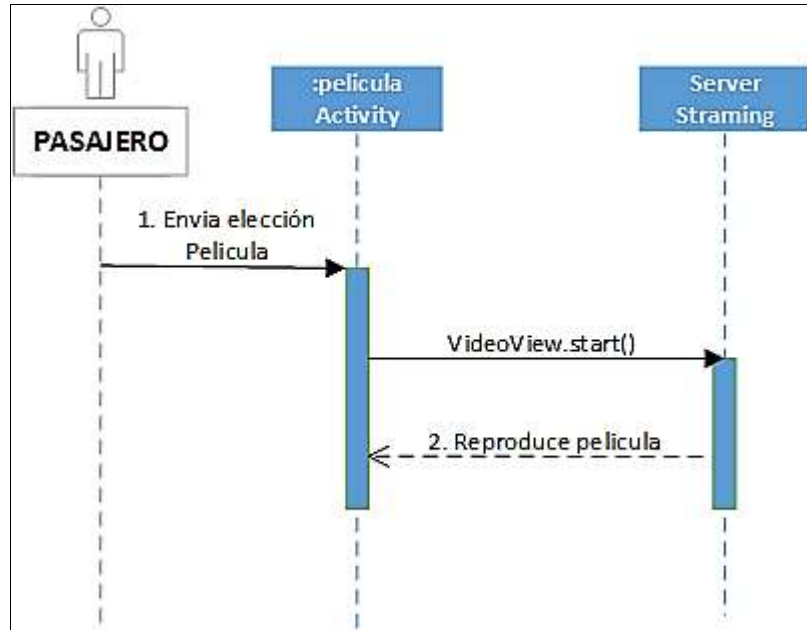


Figura 3.3.3 D.S. Reproducir una película

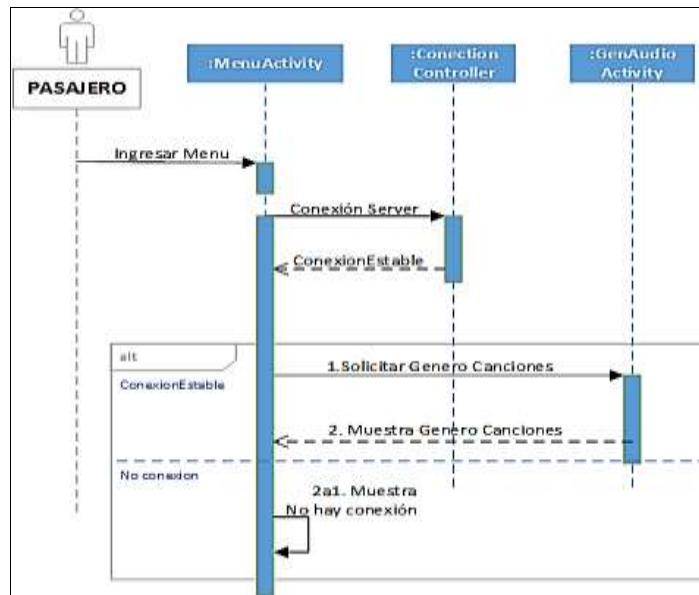
3.3.2.3 CU #3. Buscar género de canciones (ver tabla 3.3-5)

Tabla 3.3-5 CU #3 Buscar género de canciones

CU #01 Buscar genero de canciones	
Actor:	Pasajero
Precondición:	Ninguna
Disparador:	El pasajero quiere ver los géneros de canciones
Escenario principal	
<ol style="list-style-type: none"> 1. El pasajero pide a la aplicación que quiere ver los géneros de canciones ofrecidos por la Línea Aérea Amazonas. 2. El sistema muestra los tipos de géneros de canciones disponibles. 	
Extensiones	
2a. No hay conexión con el Servidor Streaming 2a1. El sistema indica al pasajero que no hay conexión disponible. 2a2. Se vuelve al paso 1.	

El diagrama de secuencia según figura 3.3.4 corresponde a la funcionalidad de buscar género de la canción.

Figura 3.3.4 Buscar género de canciones



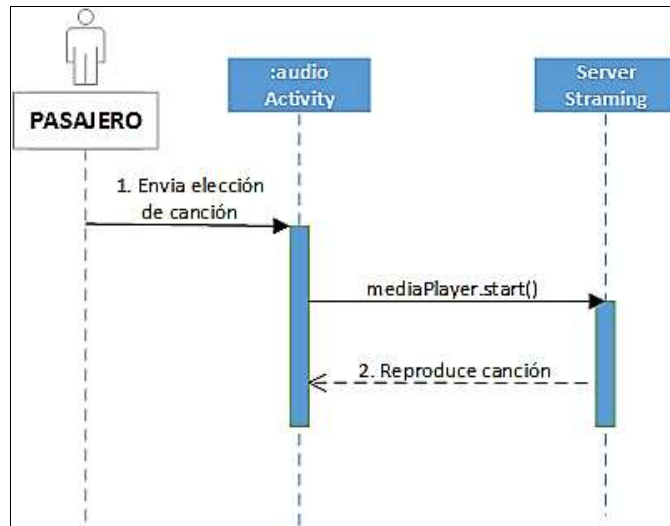
3.3.2.4 CU #4. Reproducir canción (ver tabla 3.3-6)

Tabla 3.3-6 CU #4 Reproducir canción

CU #04 Reproducir una canción	
Actor:	Pasajero
Precondición:	CU #03
Disparador:	El pasajero quiere escuchar una canción de su elección
Escenario principal	
<ol style="list-style-type: none"> 1. El pasajero indica a la aplicación que quiere reproducir una canción. 2. El sistema reproduce la canción indicada. 	
Extensiones	
2a. No hay conexión con el Servidor Streaming	

El diagrama de secuencia según figura 3.3.5 corresponde a la funcionalidad de reproducir la canción.

Figura 3.3.5 Reproducir canción



3.3.3 Diagrama de componentes

El diagrama de componentes representa la estructura de hardware/software donde será almacenada el sistema. La figura 3.3.6 muestra esta relación entre los componentes utilizados para el desarrollo de la solución.

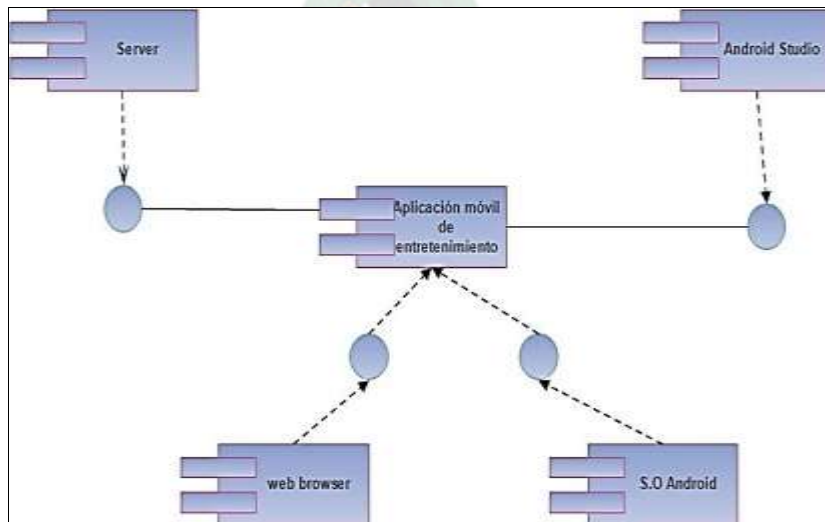


Figura 3.3.6 Diagrama de componentes

3.3.4 Modelo Conceptual

La aplicación en sí, no utiliza una base de datos, solamente requiere de archivos planos que son extraídos del servidor web mediante JSON.

Por las características y el tipo de funcionamiento del producto software en desarrollo, se considera el diseño de la tabla 3.3-7 generada con el phpMyAdmin de MySQL en el servidor web.

Tabla 3.3-7 Diseño de tablas

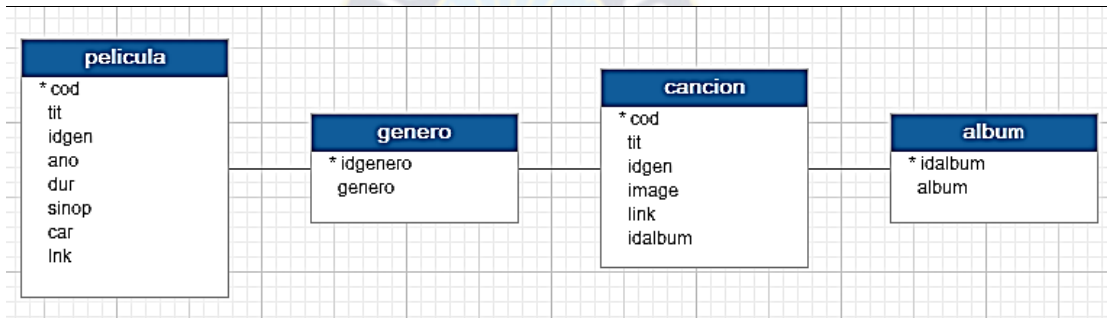


TABLA ÁLBUM

#	Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Extra
<input type="checkbox"/> 1	<u>idalbum</u>	int(10)			No	Ninguna	AUTO_INCREMENT
<input type="checkbox"/> 2	<u>album</u>	char(255)	utf8_general_ci		Sí	NULL	

TABLA CANCION

#	Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	E
1	<u>cod</u>	int(10)			No	0	
2	<u>tit</u>	char(255)	utf8_general_ci		Sí	NULL	
3	<u>idgen</u>	int(10)			Sí	NULL	
4	<u>image</u>	char(255)	utf8_general_ci		Sí	NULL	
5	<u>link</u>	char(255)	utf8_general_ci		Sí	NULL	
6	<u>idalbum</u>	int(10)			Sí	NULL	

TABLA GENERO

#	Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Extra
1	<u>idgenero</u>	int(3)			No	Ninguna	AUTO_INCREMENT
2	<u>genero</u>	char(255)	utf8_general_ci		Sí	NULL	

TABLA PELICULA

#	Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Extra
1	<u>cod</u>	int(10)			No	Ninguna	AUTO_INCREMENT
2	<u>tit</u>	char(50)	utf8_general_ci		Sí	NULL	
3	<u>idgen</u>	int(10)			Sí	NULL	
4	<u>ano</u>	int(4)			Sí	NULL	
5	<u>dur</u>	int(3)			Sí	NULL	
6	<u>sinop</u>	char(255)	utf8_general_ci		Sí	NULL	
7	<u>car</u>	char(255)	utf8_general_ci		Sí	NULL	
8	<u>lnk</u>	char(255)	utf8_general_ci		Sí	NULL	

El código sql para la generación de las tablas se presenta a continuación:

```

CREATE TABLE `album` (
  `idalbum` int(10) NOT NULL COMMENT 'id de album musical',
  `album` char(255) CHARACTER SET utf8 COLLATE utf8_general_ci NULL DEFAULT NULL
  COMMENT 'Nombre del album',
  PRIMARY KEY (`idalbum`)
)
CREATE TABLE `cancion` (
  `cod` int(10) NOT NULL DEFAULT 0,
  `tit` char(255) CHARACTER SET utf8 COLLATE utf8_general_ci NULL DEFAULT NULL
  COMMENT 'titulo de cancion',
  `idgen` int(10) NULL DEFAULT NULL COMMENT 'id genero',
  `image` char(255) CHARACTER SET utf8 COLLATE utf8_general_ci NULL DEFAULT NULL
  COMMENT 'enlace de imagen de artista',
  `link` char(255) CHARACTER SET utf8 COLLATE utf8_general_ci NULL DEFAULT NULL
  COMMENT 'enlace de audio',

```

```

`idalbum` int(10) NULL DEFAULT NULL COMMENT 'id de album musical',
PRIMARY KEY (`cod`)
)
CREATE TABLE `genero` (
`idgenero` int(3) NOT NULL COMMENT 'id genero de pelicula',
`genero` char(255) CHARACTER SET utf8 COLLATE utf8_general_ci NULL DEFAULT NULL
COMMENT 'descripcion de género de pelicula o música',
PRIMARY KEY (`idgenero`)
)
CREATE TABLE `pelicula` (
`cod` int(10) NOT NULL COMMENT 'llave primaria',
`tit` char(50) CHARACTER SET utf8 COLLATE utf8_general_ci NULL DEFAULT NULL
COMMENT 'titulo de la pelicula',
`idgen` int(10) NULL DEFAULT NULL COMMENT 'id del género al cual pertenece la pelicula',
`ano` int(4) NULL DEFAULT NULL COMMENT 'año de creacion',
`dur` int(3) NULL DEFAULT NULL COMMENT 'duracion de la pelicula',
`sinop` char(255) CHARACTER SET utf8 COLLATE utf8_general_ci NULL DEFAULT NULL
COMMENT 'sinopsis de la pelicula',
`car` char(255) CHARACTER SET utf8 COLLATE utf8_general_ci NULL DEFAULT NULL
COMMENT 'enlace de cartel',
`lnk` char(255) CHARACTER SET utf8 COLLATE utf8_general_ci NULL DEFAULT NULL
COMMENT 'enlace de pelicula',
PRIMARY KEY (`cod`)
)

```

3.3.5 Modelo de Servicio

El esquema que se utilizará para la instalación de un servicio de "*video streaming*" tiene dos actividades fundamentales bien diferenciadas:

- La elaboración de contenidos en un formato digital utilizando procedimientos de compresión.
- La distribución de los contenidos por la red a los usuarios finales.

Para la elaboración de contenidos existe una primera fase de captura de audio-video, contenidos grabados (archivos multimedia), y una fase de compresión en la que se trata separadamente el audio y el video. El resultado de esta actividad es un fichero multimedia o una corriente (*streaming*).

La distribución de contenidos es realizada por el Servidor especializado (*servidor de streaming*), almacena y/o distribuye los contenidos a los pasajeros, según la figura 3.3.7

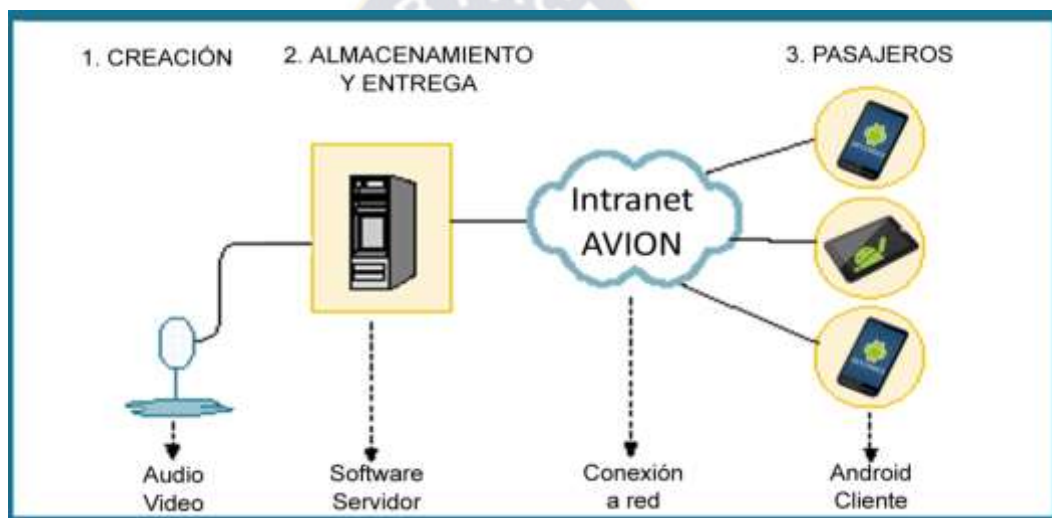


Figura 3.3.7 Modelo de Servicios

Fuente: [Elaboración Propia]

El servidor proporcionará el siguiente tipo de contenido: Difusión (*broadcast*) a varios clientes de un mismo contenido almacenado previamente en el servidor.

El sistema de difusión tiene analogías con los canales de TV.

3.3.6 Diseño de pantallas

En este apartado se muestran algunos diseños de pantallas realizados como parte de la aplicación de entretenimiento de la línea aérea Amaszonas.

El desarrollo de cada una de las interfaces fue elaborado con Android Studio utilizando enteramente la codificación bajo esta plataforma de desarrollo.

Cuando el pasajero aborda una nave, y luego de haber despegado del aeropuerto, el pasajero puede ingresar a la aplicación sin ninguna restricción y con solo presionar sobre el botón de ingreso, la aplicación ya presenta opciones para la selección de canciones, películas, revista de la empresa o publicidad. Las figuras 3.3.8 y 3.3.9 muestran esta situación.



Figura 3.3.8 Interfaz de ingreso
Fuente: [Elaboración Propia]

Selección de tipo de solicitud.

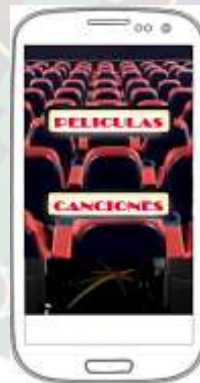


Figura 3.3.9 Interfaz de selección de tipo
Fuente: [Elaboración Propia]

Si el pasajero por ejemplo selecciona la opción de películas, entonces la aplicación móvil presenta otra interface mostrando opciones de categorías como: Acción, romántica, comedia, terror entre otros. (Ver figura 3.3.10)



Figura 3.3.10 Interfaz de despliegue de géneros de película
Fuente: [Elaboración Propia]

De acuerdo a su selección entonces se presentará un listado de archivos listos para ser visualizados en el dispositivo móvil (Ver figura 3.3.11)

Reproducción de un video en el dispositivo móvil.



Figura 3.3.11 Selección de película
Fuente: [Elaboración Propia]

3.3.7 Desarrollo de módulos

A continuación, se presenta un fragmento de la codificación de los módulos relevantes del sistema de entretenimiento desarrollado.

ADAPTADOR DE MENU POST

```
package com.example.jefe.entretenimientoz8;
import java.io.Serializable;

public class Post {
    // Atributos
    private String id;
    private String titulo;
    private String genero;
    private String imagen;
    public Post() {
    }
    public Post(String id, String titulo, String genero, String imagen) {
        this.id = id;
        this.titulo = titulo;
        this.genero = genero;
        this.imagen = imagen;
    }
    public String getId() {
        return id;
    }
    public void setId(String id) {
        this.id = id;
    }
    public String getTitulo() {
        return titulo;
    }
    public void setTitulo(String titulo) {
        this.titulo = titulo;
    }
    public String getGenero() {
        return genero;
    }
    public void setGenero(String genero) {
        this.genero = genero;
    }
    public String getImagen() {
        return imagen;
    }
    public void setImagen(String imagen) {
        this.imagen = imagen;
    }
}

```

ADAPTADOR DE MENU POST ADAPTER

```
package com.example.jefe.entretenimientoz8;
import android.content.Context;
import android.graphics.Bitmap;
import android.util.Log;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ArrayAdapter;
import android.widget.ImageView;

```



```

import android.widget.TextView;
import com.android.volley.Request;
import com.android.volley.RequestQueue;
import com.android.volley.Response;
import com.android.volley.VolleyError;
import com.android.volley.toolbox.ImageLoader;
import com.android.volley.toolbox.ImageRequest;
import com.android.volley.toolbox.JsonObjectRequest;
import com.android.volley.toolbox.NetworkImageView;
import com.android.volley.toolbox.Volley;
import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;
import java.util.ArrayList;
import java.util.List;

public class PostAdapter extends ArrayAdapter {

    // Atributos
    private String URL_BASE = "http://xxxxxxxxxxxxxxxx";
    private static final String URL_JSON = "/xxxxx.php";
    private static final String TAG = "PostAdapter";
    ListaPost items;
    public PostAdapter(Context context) {
        super(context, 0);

        // Añadir petición GSON a la cola
        MySocialMediaSingleton.getInstance(getContext()).addToRequestQueue(
            new GsonRequest<ListaPost>(
                URL_BASE+URL_JSON,
                ListaPost.class,
                null,
                new Response.Listener<ListaPost>() {
                    @Override
                    public void onResponse(ListaPost response) {
                        items = response;
                        notifyDataSetChanged();
                    }
                },
                new Response.ErrorListener() {
                    @Override
                    public void onErrorResponse(VolleyError error) {
                        Log.d(TAG, "Error Volley:"+
error.getMessage());
                    }
                }
            )
        );
    }

    @Override
    public int getCount() {
        return items != null ? items.getItems().size() : 0;
    }

    @Override
    public View getView(int position, View convertView, ViewGroup parent) {

        LayoutInflater inflater =
LayoutInflater.from(parent.getContext());

```

```

// View auxiliar
View listItemView;
//Comprobando si el View no existe
if (null == convertView) {
    //Si no existe, entonces inflarlo
    listItemView = inflater.inflate(
        R.layout.post,
        parent,
        false);
} else listItemView = convertView;

// Obtener el item actual
Post item = items.getItems().get(position);
// Obtener Views
TextView textoId = (TextView) listItemView.
    findViewById(R.id.textoId);
TextView textoTitulo = (TextView) listItemView.
    findViewById(R.id.textoTitulo);
TextView textoGenero = (TextView) listItemView.
    findViewById(R.id.textoGenero);
NetworkImageView imagenPost = (NetworkImageView) listItemView.
    findViewById(R.id.imagenPost);
// Actualizar los Views
textoId.setText(item.getId());
textoTitulo.setText(item.getTitulo());
textoGenero.setText(item.getGenero());
// Petición el image loader
ImageLoader imageLoader=
MySocialMediaSingleton.getInstance(getApplicationContext()).getImageLoader();
// Petición
imagenPost.setImageUrl(URL_BASE +item.getImagen(), imageLoader);
return listItemView;
}
}

```

REPRODUCTOR DE VIDEO STREAM

```

/**
 * Created by RCARLO on 1/6/2017.
 */
package com.example.rcarlo.pruebamediaoplayer;
import android.media.MediaPlayer;
import android.media.MediaPlayer.OnPreparedListener;
import android.net.Uri;
import android.os.Bundle;
import android.app.Activity;
import android.app.ProgressDialog;
import android.util.Log;
import android.widget.MediaController;
import android.widget.VideoView;
import com.androidbegin.videostreamtutorial.R;
public class VideoViewActivity extends Activity {
    // Declare variables
    ProgressDialog pDialog;
    VideoView videoview;
    // Video URL
    String VideoURL = URL;

    @Override

```

```

protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    // Get the layout from video main.xml
    setContentView(R.layout.videoview_main);
    // Find your VideoView in your video_main.xml layout
    videoview = (VideoView) findViewById(R.id.VideoView);
    // Execute StreamVideo AsyncTask

    // Create a progressbar
    pDialog = new ProgressDialog(VideoViewActivity.this);
    // Set progressbar title
    pDialog.setTitle("IFE Amazonas");
    // Set progressbar message
    pDialog.setMessage("Buffering...");
    pDialog.setIndeterminate(false);
    pDialog.setCancelable(false);
    // Show progressbar
    pDialog.show();
    try {
        // Start the MediaController
        MediaController mediacontroller = new MediaController(
            VideoViewActivity.this);
        mediacontroller.setAnchorView(videoview);

        Uri video = Uri.parse(VideoURL);
        videoview.setMediaController(mediacontroller);
        videoview.setVideoURI(video);
    } catch (Exception e) {
        Log.e("Error", e.getMessage());
        e.printStackTrace();
    }

    videoview.requestFocus();
    videoview.setOnPreparedListener(new OnPreparedListener() {

        public void onPrepared(MediaPlayer mp) {
            pDialog.dismiss();
            videoview.start();
        }
    });
}
}

```

3.4 Post-Game. Pruebas

Según la planificación realizada, esta fase de la metodología está orientada principalmente a las pruebas a las que debe ser sometida tanto la aplicación, así como pruebas de calidad del software.

3.4.1 Fase de ejecución de pruebas

La fase de ejecución de pruebas es el periodo de tiempo en el ciclo de vida de desarrollo software, durante el cual los componentes de un producto software son ejecutados y el producto software es evaluado para determinar si los requisitos han sido satisfechos.

3.4.1.1 Tamaño de la población.

Según el periódico La Razón, la línea aérea Amazonas redujo su flujo de pasajeros transportados en un 2015, obteniendo un decrecimiento del 12%. En 2014 transportó a 477.408 personas y el año pasado la cifra bajó a 420.898 usuarios. (Villca, 2016).

Entonces la línea transporta aproximadamente 1150 pasajeros por día, cálculo realizado por simple aritmética (por día se transporta $477408 / 365$).

3.4.1.2 Tamaño de la muestra.

Para la determinación de la muestra, se utilizó el método estadístico explicado de manera detallada en (Morales, 2012) y establecido por la fórmula:

$$\text{Tamaño de la muestra} = \frac{N * (\alpha_c * 0,5)^2}{1 + (e^2 * (N - 1))}$$

Dónde: α_c = Valor del nivel de confianza (varianza).

Nivel de confianza, es el riesgo que aceptamos de equivocarnos al presentar nuestros resultados (también se puede denominar grado o nivel de seguridad), el nivel habitual de confianza es del 95%.

e = Margen de error

Margen de error, es el error que estamos dispuestos a aceptar de equivocarnos al seleccionar nuestra muestra; este margen de error suele ponerse en torno a un 3%.

N = Tamaño Población (universo)

Entonces realizando un cálculo, realizado mediante una aplicación en Microsoft Excel, según Ilustración 3.4.1-1

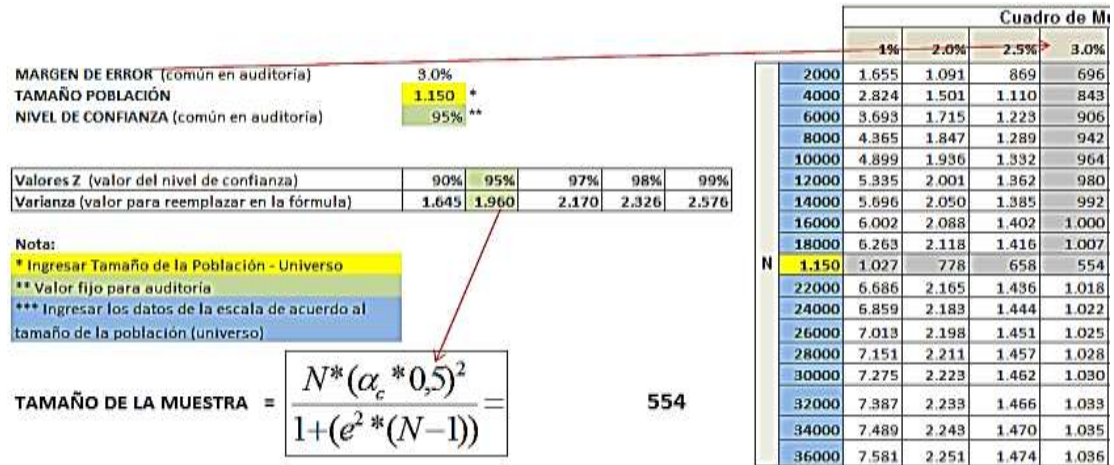


Ilustración 3.4.1-1

Se determina que el tamaño de la muestra es de 554 encuestas. El formulario de encuesta se muestra en la tabla 3.4-1

Tabla 3.4-1 Formulario de encuesta

		S	N
		I	O
Navegabilidad	La URI de entrada a la aplicación es demasiado larga para ser escrita ?	<input type="checkbox"/>	<input type="checkbox"/>
	La aplicación posee un menú de selección de opciones?	<input type="checkbox"/>	<input type="checkbox"/>
	La aplicación reproduce de manera inmediata el libro, música o video seleccionado?	<input type="checkbox"/>	<input type="checkbox"/>
	Puede navegar por la aplicación en horizontal o vertical?	<input type="checkbox"/>	<input type="checkbox"/>
	Puede encontrar de forma rápida lo que usted busca?	<input type="checkbox"/>	<input type="checkbox"/>
	Los enlaces pueden ser identificados de forma intuitiva?	<input type="checkbox"/>	<input type="checkbox"/>
	La aplicación posee ventanas innecesarias?	<input type="checkbox"/>	<input type="checkbox"/>
Control del usuario	La aplicación posee una recarga automática?	<input type="checkbox"/>	<input type="checkbox"/>
	Cuando usted selecciona una opción, ésta le redirige a otra opción?	<input type="checkbox"/>	<input type="checkbox"/>
	Las pulsaciones en las teclas responden adecuadamente?	<input type="checkbox"/>	<input type="checkbox"/>
Contenido	En contenido que se le presenta es apropiado y se adapta a su dispositivo móvil?	<input type="checkbox"/>	<input type="checkbox"/>
	Usted puede entender los mensajes de forma clara y simple?	<input type="checkbox"/>	<input type="checkbox"/>
	La aplicación le presenta lo que usted solicita y no otros contenidos?	<input type="checkbox"/>	<input type="checkbox"/>

Imágenes	Las imágenes que se presentan en la aplicación son adecuadas y se adaptan a su dispositivo?	<input type="checkbox"/>	<input type="checkbox"/>
	El fondo de la aplicación afecta a su contenido?	<input type="checkbox"/>	<input type="checkbox"/>
	Las imágenes se presentan en forma adecuada y adaptada en su dispositivo?	<input type="checkbox"/>	<input type="checkbox"/>
Color	Los colores utilizados en la aplicación afectan a su visión?	<input type="checkbox"/>	<input type="checkbox"/>
	Los colores de primer plano y del fondo contrastan lo suficiente?	<input type="checkbox"/>	<input type="checkbox"/>
Diseño	La aplicación posee muchas páginas?	<input type="checkbox"/>	<input type="checkbox"/>
	La aplicación se carga de manera rápida en su dispositivo ?	<input type="checkbox"/>	<input type="checkbox"/>
	Algunas ventanas de la aplicación se muestran de forma parcial?	<input type="checkbox"/>	<input type="checkbox"/>
	Los textos utilizados tienen un adecuado tamaño?	<input type="checkbox"/>	<input type="checkbox"/>
Hojas de estilos	Las páginas de la aplicación tienen una misma presentación?	<input type="checkbox"/>	<input type="checkbox"/>
Páginas	Puede ingresar a la aplicación aun sin indicar su elección?	<input type="checkbox"/>	<input type="checkbox"/>
	La aplicación exige que ingrese sus datos personales?	<input type="checkbox"/>	<input type="checkbox"/>
	Los botones de la aplicación poseen la suficiente información?	<input type="checkbox"/>	<input type="checkbox"/>

3.4.1.3 Periodo de Encuesta

El periodo para la elaboración de estas encuestas a los 554 pasajeros de la Línea Aérea se realizó en fecha 15 de junio al 20 de junio de 2017.

Los resultados obtenidos se muestran en la tabla 3.4-2

Tabla 3.4-2 – Resultados de encuesta

Navegabilidad		SI	NO
		La dirección web de entrada a la aplicación es demasiado larga para ser escrita?	507
La aplicación posee un menú de selección de opciones?	493	61	
La aplicación reproduce de manera inmediata la revista, publicidad, música o video seleccionado?	498	56	
Puede navegar por la aplicación en horizontal o vertical?	493	61	
Puede encontrar de forma rápida lo que usted busca?	518	36	
Los enlaces pueden ser identificados de forma intuitiva?	506	48	
La aplicación posee ventanas innecesarias?			
	36	518	

Control del usuario	La aplicación posee una recarga automática?	483	71
	Cuando usted selecciona una opción, ésta le redirige a otra opción?	488	66
	Las pulsaciones en las teclas responden adecuadamente?	502	52
Contenido	En contenido que se le presenta es apropiado y se adapta a su dispositivo móvil?	486	68
	Usted puede entender los mensajes de forma clara y simple?	504	50
	La aplicación le presenta lo que usted solicita y no otros contenidos?	509	45
Imágenes	Las imágenes que se presentan en la aplicación son adecuadas y se adaptan a su dispositivo?	505	49
	El fondo de la aplicación afecta a su contenido?	516	38
	Las imágenes se presentan en forma adecuada y adaptada en su dispositivo?	515	39
Color	Los colores utilizados en la aplicación afectan a su visión?	55	499
	Los colores de primer plano y del fondo contrastan lo suficiente?	508	46
Diseño	La aplicación posee muchas páginas?	490	64
	La aplicación se carga de manera rápida en su dispositivo ?	514	40
	Algunas ventanas de la aplicación se muestran de forma parcial?	483	71
	Los textos utilizados tienen un adecuado tamaño?	507	47
Estilo	Las páginas de la aplicación tienen una misma presentación?	492	62
Páginas	Puede ingresar a la aplicación aun sin indicar su elección?	35	519
	La aplicación exige que ingrese sus datos personales?	51	503
	Los botones de la aplicación poseen la suficiente información?	509	45

Cuyos resultados totales se muestran en la Tabla 3.4-3

Tabla 3.4-3 Resultados encuesta

FACTOR	Positivas	Negativas
Navegabilidad	504.7	49.3
Control de usuario	491.0	63.0
Contenido	499.7	54.3
Imágenes	512.0	42.0
Color	503.5	50.5
Diseño	498.5	55.5
Hojas de Estilo	492.0	62.0
Páginas	510.3	43.7
	%	0.90517
		0.09483

Cuyo gráfico se muestra en la Ilustración 3.4.1-2

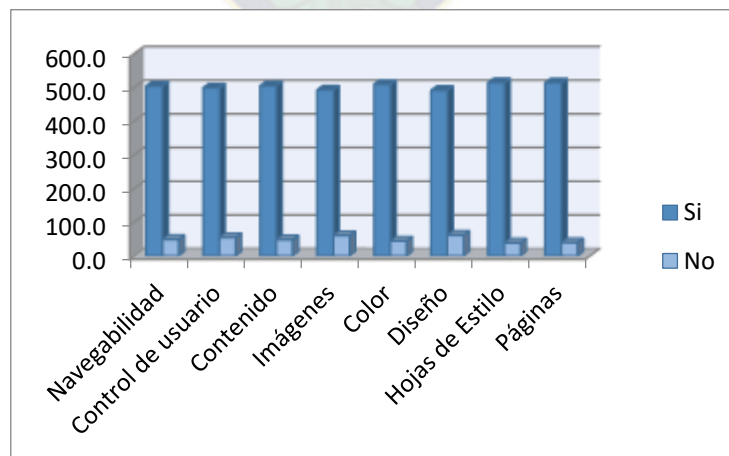


Ilustración 3.4.1-2 Resultado en barras

Por tanto es posible concluir que la aplicación tiene una aceptación según las normas de buena prácticas establecidas por la W3C. (Quevedo, 2006)

SI	90.5%
NO	9.4%

3.4.2 Pruebas de ejecución

Esta etapa considera principalmente las pruebas ejecutadas con el sistema de los diferentes módulos.

También se considera la calidad del software las normas ISO 9126 descrita en el marco teórico.

Para el ingreso al sistema (ver tabla 3.4-4)

Tabla 3.4-4 Prueba de ingreso al sistema

ID	Descripción	Pasos	Resultados Esperados	Resultados Actuales	Status
1	Probar el botón de ingreso a la aplicación.	Ingresar a la aplicación.	Ingresar al sistema	Se pudo ingresar al sistema	OK

Para seleccionar género de las películas (ver tabla 3.4-5)

Tabla 3.4-5 Prueba de selección de películas

ID	Descripción	Pasos	Resultados Esperados	Resultados Actuales	Status
1	Mostrar al pasajero las opciones de género de películas en un interfaz amigable en el dispositivo móvil.	Presionar sobre el botón de tipo de género de película.	Disponer de un listado de películas pertenecientes a un mismo género.	Se presenta una listado de películas.	OK
2			No se tiene el listado de películas por la falta de conexión al servidor	Sin conexión.	ERROR

Para los temas musicales (ver tabla 3.4-6)

Tabla 3.4-6 Prueba de reproducción de temas musicales

ID	Descripción	Pasos	Resultados Esperados	Resultados Actuales	Status
1	Mostrar al pasajero las opciones de género de temas musicales con una interfaz amigable en el dispositivo móvil.	Presionar sobre el botón de tipo de música.	Disponer de un listado de temas musicales pertenecientes a un mismo género.	Se presenta una listado de temas musicales.	OK
2			No se tiene el listado de temas musicales por la falta de conexión al servidor streaming de la aeronave.	Sin conexión.	ERROR

3.4.3 Pruebas de calidad de software

3.4.3.1 Funcionalidad

Para cuantificar la funcionalidad, en principio se determina los valores de punto función ponderando un conjunto de catorce factores en base a la siguiente escala de valores presentadas en la tabla 3.4-7

Valores de ajuste de la complejidad

Tabla 3.4-7 Valores de Ajuste de complejidad

0	1	2	3	4	5
No influencia	Incidencia	Moderado	Medio	Significativo	Esencial

La empresa cuenta con un personal en la unidad de sistemas y la parte administrativa de 25 personas.

La calificación del software fue elaborada por veinte personas funcionarios de la Línea Aérea Amazonas, lo que establece un índice del 80% del personal involucrado en el manejo de la aplicación.

Se ha establecido que el número de líneas elaboradas en la aplicación es de 3260.

La tabla 3.4-8 muestra la cuantificación de catorce factores que se utilizan para proceder al ajuste.

Tabla 3.4-8 Ajuste de complejidad del Punto Función

	Factor	Valor
1	Requiere el Sistema de copias de seguridad y de recuperación de datos fiables	5
2	Se requiere de comunicación de datos	5
3	Existen funciones de procesamiento distribuido	4
4	Es crítico el rendimiento	4
5	Se ejecuta el Sistema en un entorno operativo existente	5
6	Requiere el Sistema la entrada de datos de forma interactiva, mostrando múltiple pantallas u operaciones.	4
7	Se actualizan los archivos maestros de forma automática	5
8	Son complejas las entradas, salidas o peticiones de usuario	4
9	Es complejo el procesamiento interno	4
10	Es compleja la utilización del Sistema	4
11	Se ha diseñado el código para ser reutilizable	5
12	Está incluida en el diseño la instalación	4
13	Se ha diseñado para facilitar los cambios y sea de fácil uso para el usuario	4
14	Se diseñó el Sistema para soportar múltiples instalaciones en diferentes departamentos	4
	ΣFi	61

Para calcular puntos: $PF = \text{Cuenta Total} \times [R(t) + 0.01 \times \Sigma Fi]$

Dónde: Cuenta Total=Suma de todas las entradas obtenidas

0.01=Error de confiabilidad de Sistema

ΣFi = Suma de factores de complejidad

Cálculos:

$$PF = 163 (0.65 + 0.01 * 61) = 205.3$$

$$PF \text{ M\u00e1ximo} = 163 (0.65 + 0.01 * 70) = 220$$

$$\text{Funcionalidad} = (PF / PF \text{ M\u00e1ximo}) * 100 = (205.3 / 220) * 100 = 93.3$$

Entonces la funcionalidad del Sistema es de 93.4%

3.4.3.2 Fiabilidad

La fiabilidad se refiere al punto en que se puede esperar que el programa lleve a cabo su funci\u00f3n predefinida con la exactitud requerida y est\u00e1 dado por:

$$1 - (\text{n\u00famero de errores} / \text{n\u00famero de l\u00edneas de c\u00f3digo})$$

La tabla 3.4-9 muestra los resultados de pruebas de corridas realizadas.

Tabla 3.4-9 Resultados de evaluaci\u00f3n de ejecuci\u00f3n del sistema

Tiempo de servicio a	Transacciones realizadas b	Fallas encontradas c	Probabilidad fallo bajo demanda d = c / b	Tiempo medio entre fallos e = a / c
8 hrs	70	52	0.074	0.15 hrs.
16 hrs	54	34	0.629	0.47 hrs.

El promedio de errores encontrados es de $124 / 2 = 62$ errores en las dos pruebas de ejecuci\u00f3n del software.

$$\text{Fiabilidad} = 1 - (62 \text{ errores} / 3260 \text{ l\u00edneas de c\u00f3digo})$$

$$\text{Fiabilidad} = 0.988 * 100$$

$$\text{Fiabilidad} = 98\% \text{ (Significa que en 98\% el sistema es fiable)}$$

3.4.3.3 Facilidad de mantenimiento

Es el esfuerzo necesario para localizar y arreglar un error en el programa y está dada por la siguiente ecuación matemática:

$$= 1 - 0.1 * (\text{número medio de días - hombre por corrección})$$

$$\text{Facilidad de mantenimiento} = 1 - 0.1 (2 - 1 \text{ persona por corrección})$$

$$\text{Facilidad de mantenimiento} = 0.9 * 100$$

Facilidad de mantenimiento = 90 % (Significa que en un 90% es fácil de mantener)

3.4.3.4 Eficiencia

Para evaluar este factor de calidad, se consideraron los atributos correspondientes a la característica de eficiencia compuesto por la comprensibilidad del sistema, mecanismos de ayuda, interface del sistema, y los aspectos de exploración.

Tabla 3.4-10 Resultados de evaluación según encuesta a los usuarios

CARACTERÍSTICA	Usr1	Usr2	Usr3	Usr4	Usr5	Usr6	Usr7	Usr8	Usr9	Usr10
Comprensibilidad del sistema	90	88	95	80	95	95	90	88	89	95
Mecanismos de ayuda y retroalimentación	75	78	85	91	88	90	75	80	85	80
Aspectos de la interfaz	92	95	95	90	85	86	88	90	88	92
Aspectos de exploración	84	90	90	88	74	90	89	92	84	90
Errores	10	5	10	15	20	10	15	20	20	5
	Usr11	Usr12	Usr13	Usr14	Usr15	Usr16	Usr17	Usr18	Usr19	Usr20
Comprensibilidad del sistema	94	88	98	90	95	98	90	88	94	95
Mecanismos de ayuda y retroalimentación	89	89	85	98	88	90	75	95	87	90
Aspectos de la interfaz	92	95	95	90	94	86	98	97	88	92
Aspectos de exploración	89	90	97	94	74	90	93	92	84	95
Errores	5	10	7	9	0	3	8	2	7	9

Como se observa en la tabla 3.4-10, se tienen veinte evaluaciones y los resultados promedios se muestran en la tabla 3.4-11:

Tabla 3.4-11 Resultados de evaluación de eficiencia del sistema

CARACTERÍSTICA	TOTAL
Comprensibilidad del sistema	90.5
Mecanismos de ayuda y retroalimentación	82.7
Aspectos de la interfaz	90.1
Aspectos de exploración	87.1
Errores	13

Los resultados obtenidos en base a un cuestionario resuelto tanto por los usuarios administradores (Dos) así como los pasajeros de la Línea Aérea (Dieciocho), se consideró un total de 20 encuestas.

De acuerdo a los resultados obtenidos, la eficiencia es:

$$\text{Eficiencia} = \text{promedio} (90.5, 82.7, 90.1, 87.1, (100-13)) = 87.48\%$$

Este resultado indica que el Sistema elaborado logra un servicio con una eficiencia del 87.5%, lo que se interpreta como un rendimiento satisfactorio al momento de generar informes de salida, reportes y de ser comprensible en el manejo, ya que estas fueron diseñadas con elementos simples que reducen el tiempo de proceso.

3.4.3.5 Flexibilidad

La flexibilidad es el coste de modificación del producto cuando cambian sus especificaciones, y puede ser calculada a partir de la siguiente fórmula:

$$1 - 0.05 \text{ (número medio de días – hombre por cambio)}$$

$$\text{Flexibilidad} = 1 - 0.05 (2 - 1)$$

$$\text{Flexibilidad} = 0.95 * 100$$

$$\text{Flexibilidad} = 95\% \text{ (Significa que en un 95\% es flexible el Sistema)}$$

3.4.4 Costo de desarrollo

Este apartado trata sobre la estimación del costo de desarrollo del sistema. Se determina utilizando el modelo COCOMO II.

Estimación de costos. COCOMO II. Modelo post-arquitectura.

El esfuerzo nominal se ajusta usando 17 factores multiplicadores de esfuerzo. El mayor número de multiplicadores permite analizar con más exactitud el conocimiento disponible en las últimas etapas de desarrollo, ajustando el modelo de tal forma que refleje fielmente el producto de software bajo desarrollo. La fórmula para el cálculo del esfuerzo es la siguiente:

$$PM_{\text{estimado}} = PM_{\text{nominal}} \times \text{Productoria}(EM_i)$$

Los factores que se deben considerar se muestran en la tabla 3.4-12

Tabla 3.4-12 Factores

	Factor
Producto	RELY
	DATA
	CPLX
	RUSE
	DOCU
Plataforma	TIME
	STOR
	PVOL
Personal	ACAP
	PCAP
	PCON
	AEXP
	PEXP

	LTEX
Proyecto	TOOL
	SITE
	SCED

La determinación de las líneas de código por módulo desarrollado se muestra en la tabla 3.4-13 y luego de haber realizado el conteo utilizando una herramienta CASE LOC Counter GUI, cada módulo tiene las siguientes líneas desarrolladas.

Punto 1.

Identificar los módulos que conforman el sistema, asignarles un número y un nombre.

Tabla 3.4-13 Módulos del sistema

#	Módulo
1	Administración
2	Selección
3	Reproducción

Luego se establece el factor exponencial de escala B considerando 5 factores W_j (PREC, FLEX, RESL, TEAM y PMAT), considerándolos como nominal.

Punto 2.

Realizar el conteo de líneas por cada uno de los módulos. Ver tabla 3.4-14

Tabla 3.4-14 Conteo de líneas

#	Módulo	SLOC
1	Administración	453
2	Selección	352
3	Reproducción	432

Punto 3

Determinar el tamaño en SLOC del Sistema. Ver tabla 3.4-15

Tabla 3.4-15 Tamaño de SLOC

#	Módulo	SLOC
1	Administración	453
2	Selección	352
3	Reproducción	432
		1237

Punto 4.

Calcular el Factor Exponencial de Escala (B). (Ver tabla 3.4-16)

$$B = 1.01 + 0.01 \times \text{Sumatoria } W_j$$

Tabla 3.4-16 Factor exponencial

	Factor	VLO	LO	NOM	HI	VHI	XHI
Proyecto	PREC	6,2	4,96	3,72	2,48	1,24	0
	FLEX	5,07	4,05	3,04	2,03	1,01	0
	RESL	7,07	5,65	4,24	2,03	1,41	0
	TEAM	5,48	4,38	3,29	2,19	1,1	0
	PMAT	7,8	6,24	4,68	3,12	1,56	0

Considerando los valores como nominal, entonces se obtiene:

$$B = 1.01 + 0.01 \times (3.72+3.04+4.24+3.29+4.68)$$

$$B = 1.1997$$

Punto 5.

Calcular el Esfuerzo Nominal requerido para desarrollar el sistema. (PM_{nominal})

$$PM_{\text{nominal}} = A * KSLOC^B$$

A es una constante 2.94

$$PM_{\text{nominal}} = 2.94 * 1.237^{1.1997}$$

$$PM_{\text{nominal}} = 3.79$$

Luego determinar la Productividad nominal

$$Productividad_{\text{Nominal}} = KSLOC / PM_{\text{nominal}}$$

$$Productividad_{\text{Nominal}} = 1237 / 3.79$$

$$Productividad_{\text{Nominal}} = 325.99$$

Punto 6.

Calcular el esfuerzo nominal por módulo según tabla 3.4-17

$$Productividad_{\text{Nominal, modulo}} = SLOC_{\text{modulo}} / Productividad_{\text{Nominal}}$$

Tabla 3.4-17 Esfuerzo nominal

#	Módulo	SLOC	PM nominal
1	Administración	453	1.389606259
2	Selección	352	1.079782347
3	Reproducción	432	1.325187425
		1237	

Punto 7.

Analizar las características de cada módulo y asignar nivel determinado para los multiplicadores de esfuerzo. Ver tabla 3.4-18

Tabla 3.4-18 Niveles de multiplicadores de esfuerzo

#	Módulo	SLOC	RELY	DATA	DOCU	CPLX	RUSE	TIME	STOR	PVOL	ACAP	AEXP	PCAP	PEXP	LTEX	PCON	TOOL	SCED	SITE	EAF
1	Administración	453	1	1	1	1	1	1	1	0.8	0.8	0.8	1	1	1	1	0.9	1	1	0.53
2	Selección	352	1	1	1	1	1	1	1	0.8	0.8	0.8	0.8	0.9	0.9	1	0.9	1	1	0.39
3	Reproducción	432	1	1	0.8	1	1	1	1	0.8	0.8	0.8	1.1	1.0	1.0	1	0.9	1	1	0.64

Punto 8.

Obtener factor de ajuste del Esfuerzo EAF. El factor EAF es la productoria de los 17 factores por cada módulo.

Punto 9.

Calcular el esfuerzo estimado por modulo, multiplicando cada PMnominal por cada EAF. Ver tabla 3.4-19

Tabla 3.4-19 Esfuerzo estimado por modulo

#	Módulo	SLOC	PM nominal	EAF	Pmestimado,module
1	Administración	453	1.389606259	0.5390955	0.74913048
2	Selección	352	1.079782347	0.39285399	0.4241968
3	Reproducción	432	1.325187425	0.64081961	0.84920609
		1237			

Punto 10.

Sumar los valores calculados para determinar el Esfuerzo Estimado del Sistema Total PM estimado. Ver tabla 3.4-20

Tabla 3.4-20 Esfuerzo estimado del Sistema Total PM

Módulo	SLOC	PM nominal	EAF	Pmestimado,module
Administración	453	1.389606259	0.5390955	0.74913048
Selección	352	1.079782347	0.39285399	0.4241968
Reproducción	432	1.325187425	0.64081961	0.84920609
	1237			2.02253337

Punto 11.

Determinar el tiempo de desarrollo estimado del proyecto TDEV.

$$TDEV = [3.0 * PM^{(0.33 + 0.2 * (B - 1.01))}] * (SCED\%) / 100$$

Considerando que SCED como 100% compresión/expansión del cronograma

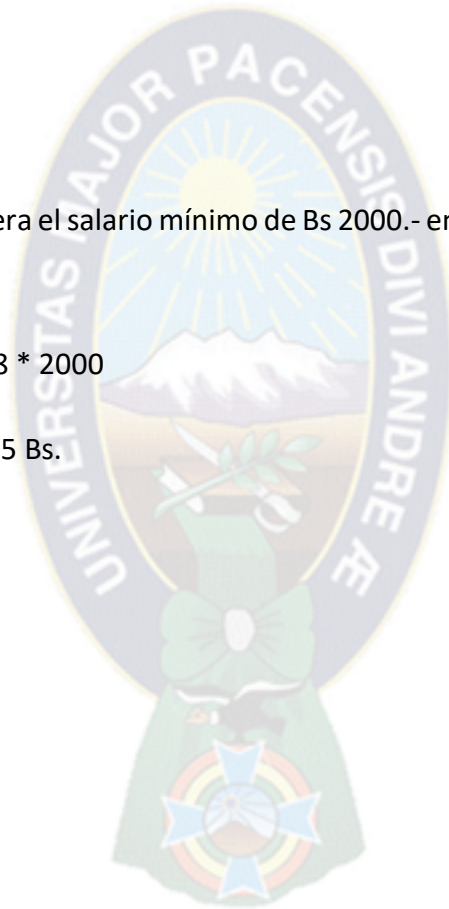
$$TDEV = 3.88 \text{ meses.}$$

Punto 12.

Finalmente se considera el salario mínimo de Bs 2000.- entonces el costo ce desarrollo de la aplicación es de

$$\text{Costo Estimado} = 3.88 * 2000$$

$$\text{Costo estimado} = 7775 \text{ Bs.}$$



CONCLUSIONES Y RECOMENDACIONES

4.1 Conclusiones

Se logró analizar, diseñar e implementar un Software de aplicación orientado a móviles Android mejorando la calidad de vuelo de los pasajeros de la Línea Aérea Amazonas S.A., pudiendo obtener mayor fidelización del cliente con este simple aplicativo y mantener contentos y entretenidos a los clientes.

Se logró implementar un servidor multimedia capaz de distribuir la información a la cantidad de pasajeros en el vuelo.

La gestión de los archivos multimedia cuenta con módulos desarrollados fáciles de ser manejados y a cargo del administrador del aplicativo.

La metodología ágil utilizada estableció de manera directa y concreta el desarrollo del aplicativo logrando obtener un producto software de calidad.

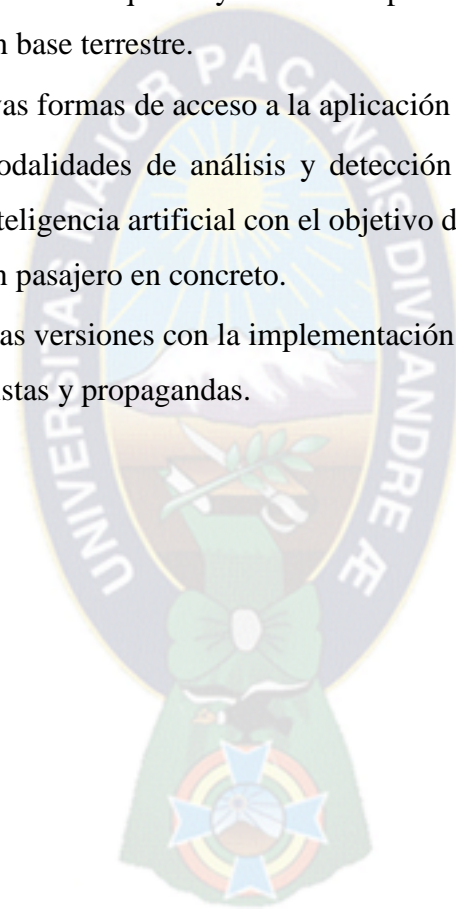
Según las encuestas realizadas se estableció que de los pasajeros que tuvieron una experiencia con la aplicación, hubo un 90% de aceptación y conformidad.

Se logró desarrollar y administrar web *services* asociados a la programación android. Para el intercambio de datos entre los servicios web, se manejó como formato de transferencia de datos el formato JSON.

4.2 Recomendaciones

A partir de los resultados obtenidos, se han establecido algunas áreas desde donde pueden realizarse nuevos trabajos futuros.

- Ampliar el sistema para la coordinación e intercambio de información con otros sistemas y unidades que hayan sido implementados en la Línea Aérea Amazonas en base terrestre.
- Elaborar nuevas formas de acceso a la aplicación por ejemplo interfaz de voz
- Incorporar modalidades de análisis y detección de satisfacciones utilizando técnicas de inteligencia artificial con el objetivo de conocer el comportamiento e interés de un pasajero en concreto.
- Ampliar nuevas versiones con la implementación de otros idiomas tanto en las películas, revistas y propagandas.



BIBLIOGRAFÍA

- AmericanDominios. (s.f.). *Que son los protocolos RTSPU, RTSPT, MMSU y MMST*.
Obtenido de <http://www.americandominios.com/conta/knowledgebase/233/Que-son-los-protocolos-RTSPU-RTSPT-MMSU-y-MMST.html>
- Arroyo, F. J. (11 de marzo de 2014). *Como añadir Volley (libreria de Android)*.
Obtenido de <https://www.adictosaltrabajo.com/tutoriales/android-volley/>
- Bahit, E. (2009). *POO y MVC en PHP*. Creative Commons Atribución NoComercial-SinDerivadas 3.0.
- Ben, E. (2005). *Google Buys Android for Its Mobile Arsenal*. Bloomberg Businessweek.
- Boehm, B., Clark, B., Horowitz, E., Westland, C., Madachy, R., & Selby, R. (1995). *The COCOMO 2.0 Software Cost Estimation Model*.
- Booch, G., Rumbaugh, J., & Jacobson, I. (1999). *Lenguaje Unificado de Modelado*.
- Calderon Macias, F. R. (2016). *El Estándar ISO y su Aportación al Proceso de Calidad del Desarrollo de Software*.
- Cehis. (04 de febrero de 2016). *Que es y para que sirve el streaming*. Obtenido de <http://cehis.net/sitio/ayuda-video-streaming/asistencia-y-soporte/base-de-conocimiento-faq/ayuda-video-streaming/que-es-y-para-que-sirve-el-streaming>
- Cordero, C. (24 de junio de 2016). *Aerolíneas amplían servicios de Internet y entretenimiento en los vuelos*. Obtenido de http://www.nacion.com/economia/empresarial/American-Avianca-Copa-Delta-Iberia-United-Volaris-Internet-aerolineas_0_1568643203.html

- Genymotion. (s.f.). *Genymotion*. Obtenido de <https://www.genymotion.com/>
- Giraldo, S. S. (03 de abril de 2014). *Eclipse es un programa informático compuesto por un conjunto*. Obtenido de <https://prezi.com/s0iiyrafxxui/eclipse-es-un-programa-informatico-compuesto-por-un-conjunto/>
- Gomez, A. (2014). *Aplicacion Android para la empresa Travelling-Service*. Madrid.
- Gomez, A., Lopez, M. d., Migiani, S., & Itazu, A. (s.f.). *COCOMO - UN MODELO DE ESTIMACION DE PROYECTOS DE SOFTWARE*.
- Herederero, C. d. (2004). *Informática y comunicaciones en la empresa*. ESIC.
- Hevia, A. (2010). *Principios de la Arquitectura Orientada a Servicios*. Obtenido de <https://andreshevia.com/2010/02/27/principios-de-soa/>.
- ISO/IEC9126. (2001). *International Organization for Standardization*. Recuperado el 10 de Marzo de 2017, de <https://www.iso.org/standard/22749.html>
- Llensa, E. (2015). *Análisis del programa para prototipar apps: InVision*. Obtenido de <http://www.ubicuostudio.com/es/disenio-grafico/programa-prototipar-apps-invision/>
- Madachy, R. (s.f.). *COCOMO II - Constructive Cost Model*. Recuperado el 20 de Marzo de 2017, de <http://csse.usc.edu/tools/cocomoii.php>
- Morales, P. (2012). *Estadística aplicada a las Ciencias Sociales - Tamaño necesario de la muestra*. Obtenido de www.up.es/personal/peter/investigacion/Tama%F1oMuestra.pfd
- Morente, J. A. (30 de enero de 2017). *5 herramientas gratuitas que nos permiten crear diagramas*. Obtenido de <http://www.unir.net/ingenieria/revista/noticias/5-herramientas-gratuitas-que-nos-permiten-crear-diagramas/549201645031/>
- Muñoz, C. (2011). *Cómo elaborar y asesorar una investigación de tesis Cap.6*.

- Navarro, R. (2006). *REST Vs. Web Services*. Obtenido de <http://users.dsic.upv.es/~rnavarro/NewWeb/docs/RestVsWebServices.pdf>
- Palacios, J. (2014). *Gestión de proyectos Scrum Manager*.
- Perez, D. (05 de marzo de 2017). *Cómo funciona el sistema de entretenimiento en vuelo de un avión*. Obtenido de <http://omicronno.lespanol.com/2017/03/sistema-entretenimiento-en-vuelo-avion/>
- Quevedo, J. R. (28 de 06 de 2006). *Qweos*. Obtenido de <http://qweos.net/blog/2006/06/28/guias-practicas-para-profesionales-web-buenas-practicas-en-web-movil/>
- Revelo, J. (29 de agosto de 2014). *¿Cuales son los Componentes de una Aplicación Android?* Obtenido de <http://www.hermosaprogramacion.com/2014/08/android-app-componentes/>
- Ruiz, Z. (27 de noviembre de 2013). *El patrón de arquitectura modelo vista controlador*. Obtenido de https://prezi.com/ret8n_fyoulg/el-patron-de-arquitectura-modelo-vista-controlador-es-una/
- Schwaber, K., & Sutherland, J. (2013). *La Guía de Scrum. La Guía Definitiva de Scrum: Las Reglas del Juego*.
- Standard, J. D. (s.f.). *Introducción a JSON*. Obtenido de <http://www.json.org/json-es.html>
- Villca, C. (27 de 02 de 2016). Aerolíneas mueven en promedio 7.693 pasajeros por día en el país . *La Razón*.
- Wowza. (s.f.). *Wowza Streaming Engine*. Obtenido de <http://www.wowza.com/products/streaming-engine/documentation>