



UNIVERSIDAD CENTRAL DE VENEZUELA

FACULTAD DE CIENCIAS

ESCUELA DE COMPUTACIÓN

LABORATORIO DE REDES MÓVILES E INALÁMBRICAS

**DESARROLLO DE UNA APLICACIÓN
MÓVIL EN ANDROID DE M-LEARNING
PARA LA ESCUELA DE COMPUTACIÓN
DE LA UNIVERSIDAD CENTRAL DE VENEZUELA.**

Trabajo Especial de Grado presentado ante la Ilustre

Universidad Central de Venezuela por el bachiller

Gabriel Gaslonde (C.I. V-20.799.141)

para optar por el título de Licenciado en Computación

Tutor:

Profesor Fernando Crema

Ciudad Universitaria de Caracas, Octubre de 2015

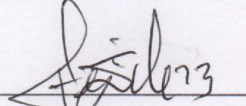
ACTA

Quienes suscriben, miembros del Jurado designado por el Consejo de Escuela de Computación de la Facultad de Ciencias de la Universidad Central de Venezuela, para examinar el Trabajo Especial de Grado titulado: Desarrollo de una aplicación móvil en Android de M-learning para la Escuela de Computación de la Facultad de Ciencias de la UCV.

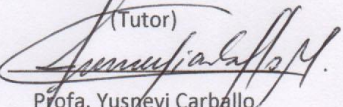
Presentado por el bachiller Gabriel Andres Gaslonde Gioia, portador de la cédula de identidad V.-20.799.141 a los fines de optar al título de Licenciado en Computación, dejan constancia de lo siguiente:

Dicho trabajo, leído por cada uno de los miembros del jurado, se fijó el día, para que su autor lo defendiera en forma pública en la Escuela de Computación, mediante una presentación oral de su contenido, luego de lo cual se respondieron las preguntas formuladas. Finalizada la defensa pública del Trabajo Especial de Grado, el jurado decidió aprobarlo con la nota de VEINTE (20) puntos.

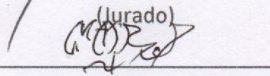
En fe de lo cual se levanta la presente Acta, en Caracas a los veintidós días del mes de Octubre del 2015, dejando constancia de que actuó como coordinador del jurado el Profesor Fernando Crema.


Prof. Fernando Crema

(Tutor)


Profa. Yusreysi Carballo,

(Jurado)


Prof. Miguel Astor

(Jurado)



Universidad Central de Venezuela

Facultad de Ciencias

Escuela de Computación

Laboratorio de Redes Móviles e Inalámbricas

DESARROLLO DE UNA APLICACIÓN MÓVIL EN ANDROID DE M-LEARNING PARA LA ESCUELA DE COMPUTACIÓN DE LA UNIVERSIDAD CENTRAL DE VENEZUELA.

Autor:

Gabriel Gaslonde,
ggaslonde@gmail.com

Tutor:

Prof Fernando Crema,
fernando.crema@ciens.ucv.ve

Fecha:

22 de Octubre de 2015

RESUMEN

Los profesores de la Escuela de Computación de la Universidad Central de Venezuela actualmente no poseen una aplicación móvil propia de la Escuela que puedan utilizar y que se adecuen a algunas de sus necesidades. Entre las necesidades de algunos de los profesores están una aplicación que pueda reproducir videos y que pueda desplegar documentos, pero a su vez haga un uso eficiente de datos ya que las conexiones del país con respecto a la tasa de subida y descarga de datos no siempre tienen buen rendimiento. Por otro lado resulta atractivo para cualquier estudiante el poder tener acceso a una aplicación móvil que refleje el contenido de las materias que cursa en un semestre determinado. Para dar solución a las necesidades de los profesores y estudiantes se propone este Trabajo Especial de Grado (TEG) cuya finalidad es el desarrollo de una aplicación móvil para el sistema operativo Android, que pueda desplegar documentos y reproducir videos, haciendo un uso eficiente de datos para aligerar el impacto sobre la plataforma tecnológica. Así como también el desarrollo de tecnologías de base de datos, servidor de manejo de contenidos y servidor de *stream* que den soporte a la aplicación móvil.

Palabras Clave: Aplicación móvil, *Stream*, Android, aprendizaje, enseñanza, m-learning, e-learning.

Índice de contenidos

Introducción.....	1
Capítulo 1: Problema de Investigación	4
1. Planteamiento del Problema	4
2. Objetivos	5
2.1 Objetivo General	5
2.2 Objetivos Específicos.....	5
3. Justificación e Importancia	5
Capítulo 2: Marco Conceptual	7
2.1 <i>E-Learning</i>	7
2.2 <i>Mobile Learning</i>	8
2.2.1 Características del Mobile Learning [Universidad,2013]	8
2.2.2 Ventajas del <i>Mobile Learning</i> [Universidad,2013].....	9
2.2.3 Retos para su uso en educación tradicional	10
2.3 Android	11
2.3.1 Arquitectura Android	11
2.4 <i>Streaming</i>	14
2.5 RTMP “ <i>Real Time Messaging Protocol</i> ”	15
2.5.1 Funcionamiento	16
2.6 Servidor web	16
2.7 nginx.....	17
2.7.1 Arquitectura nginx	18
2.8 PHP	19
2.9 RED5.....	20
2.9.1 Características.....	21
2.9.2 Ventajas [Gong,2008]:	22
2.9.3 Desventajas [Gong,2008]:.....	22
2.10 Bases de datos	22
2.11SQL	24

2.12 ACID.....	24
2.13 MariaDB	24
2.14 <i>Android Studio</i>	25
2.14.1 Proyecto y estructura del archivo	25
2.14.2 Sistema de construcción Android	27
2.15 Método de desarrollo de la plataforma de aprendizaje.....	27
2.15.1 Metodología de Roger Pressman.....	28
2.15.2 Análisis de los requisitos del software.....	29
2.15.3 Diseño	30
2.15.4 Generación de Código:.....	31
2.15.5 Pruebas	31
2.15.6 Mantenimiento	32
Capítulo 3: Marco Aplicativo.....	33
3.1 Abordando la Problemática	33
3.2 Análisis de los requerimientos del software.....	34
3.2.1 Especificación de los casos de uso de la aplicación móvil	36
3.2.2 Especificación de los casos de uso de la aplicación de administración	39
3.2.3 Modelado del sistema.....	42
3.3 Diseño	46
3.3.1 Diseño de datos.....	46
3.3.2 Diseño arquitectónico y procedimental.....	49
3.3.3 Diseño de la interfaz	52
3.4 Generación de código	61
3.4.1 Módulo de ingreso de la aplicación móvil	61
3.4.2 Módulo de registro de la aplicación móvil.....	62
3.4.3 Módulo de materias de la aplicación móvil	63
3.4.4 Módulo de documentos de la aplicación móvil	64
3.4.5 Módulo de videos de la aplicación móvil.....	64
3.4.6 Módulo de inicio de la aplicación de administración	65
3.4.7 Módulo de agregar materia de la aplicación de administración	66
3.4.8 Módulo de administrar materia de la aplicación de administración.....	67
4. Pruebas	68

4.1 Pruebas de la aplicación móvil EduKT.....	68
4.2 Pruebas de la aplicación de administración de contenidos.....	73
4.3 Pruebas de rapidez de descarga de los videos.....	77
5. Resultados.....	77
5.1 Se desarrolló de una aplicación móvil con tecnología Android	77
5.2 Se desarrolló de una página web que permite la administración de contenidos.....	78
5.3 Se configuró un servidor de <i>streaming</i> RED5 para el manejo de contenido multimedia.....	78
5.4 Se diseñó una base de datos SQL para información de usuario y sistema	79
Conclusiones	80
Trabajos Futuros	81
Bibliografía	82

Índice de figuras

Figura 1: Estructura proyecto Android.....	26
Figura 2: Gradle de Android.....	27
Figura 3: Proceso de desarrollo de aplicaciones.....	28
Figura 4: Casos de uso nivel 1 Aplicación móvil.....	35
Figura 5: Casos de uso nivel 1 Aplicación de Administración.....	35
Figura 6: Casos de uso nivel 2 Aplicación de Administración.....	36
Figura 7: Diagrama de componentes de la aplicación móvil.....	43
Figura 8: Diagrama de componentes de la aplicación de administración.....	43
Figura 9: Diagrama de entidad relación de la aplicación de móvil.....	45
Figura 10: Diagrama de entidad relación de la aplicación de administración.....	45
Figura 11: Arquitectura de la aplicación móvil.....	50
Figura 12: Arquitectura de la aplicación administración.....	51
Figura 13: Interfaz de inicio.....	53
Figura 14: Interfaz de las materias.....	54
Figura 15: Interfaz de los contenidos.....	55
Figura 16: Interfaz que despliega documentos.....	56
Figura 17: Interfaz que despliega videos.....	57
Figura 18: Interfaz de inicio aplicación de administración.....	58
Figura 19: Interfaz agregar materia aplicación de administración.....	59
Figura 20: Interfaz administrar materia aplicación de administración.....	60

Índice de Tablas y Gráficos

Tabla 1: Especificación del caso de uso Iniciar sesión.....	37
Tabla 2: Especificación del caso de uso Registrar usuario.....	37
Tabla 3: Especificación del caso de uso Ver materia.....	38
Tabla 4: Especificación del caso de uso Ver video.....	38
Tabla 5: Especificación del caso de uso Ver documento.....	39
Tabla 6: Especificación del caso de uso Agregar materia.....	39
Tabla 7: Especificación del caso de uso Eliminar materia.....	40
Tabla 8: Especificación del caso de uso Administrar materia.....	40
Tabla 9: Especificación del caso de uso Cargar video.....	41
Tabla 10: Especificación del caso de uso Cargar documento.....	41
Tabla 11: Especificación del caso de uso Eliminar contenido.....	42
Tabla 12: Especificación de la tabla contenido.....	47
Tabla 13: Especificación de la tabla usuario.....	47
Tabla 14: Especificación de la tabla usuario_materia.....	47
Tabla 15: Especificación de la tabla materia.....	48
Gráfico 1: Dificultad de uso de la aplicación móvil.....	69
Gráfico 2: Utilidad de los mensajes de la aplicación móvil.....	69
Gráfico 3: Módulo de PDF de la aplicación móvil.....	70
Gráfico 4: Calidad de los videos de la aplicación móvil.....	70
Gráfico 5: Rapidez de carga de videos de la aplicación móvil.....	71
Gráfico 6: Recomendar la aplicación móvil.....	72
Gráfico 7: Dificultad de uso aplicación de administración.....	73
Gráfico 8: Utilidad de los mensajes aplicación de administración.....	74
Gráfico 9: Manipulación de contenidos de la aplicación de administración.....	74
Gráfico 10: Rapidez de recursos agregados a la aplicación de administración.....	75
Gráfico 11: Recomendar la aplicación de administración.....	76

Introducción

El uso de herramientas informáticas para el apoyo del aprendizaje han causado un gran impacto en la educación actual. La informática, quizás es uno de los avances más grandes en los últimos tiempos, ha adquirido una importancia sin precedente en nuestras vidas de tal manera que constituye, en la actualidad, uno de los medios de expansión, distribución y obtención de información con más presencia a escala mundial. Es por ello que resulta atractivo implementar herramientas informáticas para el apoyo del aprendizaje del estudiantado explotando esta área en continua expansión.

Esto nos lleva a introducir el término de *M-Learning*, como la modalidad educativa que facilita la construcción del conocimiento, la resolución de problemas de aprendizaje y el desarrollo de destrezas o habilidades diversas de forma autónoma y ubicua gracias a la mediación de dispositivos móviles portables [Brazuelo,Gil,Domingo,2011].

El dispositivo móvil es una herramienta estrella para el uso educativo, ya que el mismo se encuentra integrado a la vida cotidiana de las personas. Estudios indican que estos artefactos se han convertido en lo primero que se usa al despertar y lo último que se observa antes de dormir [Universidad,2013].

Los dispositivos móviles poseen características de un gran potencial educativo tales como:

- Conectividad.
- Geolocalización.
- Grabación.
- Creación y publicación de información.
- Ubicuidad.

En la actualidad, no existe una aplicación móvil, que puedan utilizar los profesores y estudiantes en la Escuela de Computación para apoyar el proceso de aprendizaje y que se adecuen a sus necesidades. Es por ello que el objetivo de este Trabajo Especial de Grado es desarrollar una plataforma de aprendizaje móvil, para apoyar el proceso de aprendizaje profesor-estudiante facilitando la consulta de los contenidos relacionados con determinadas asignaturas. Con la perspectiva de integrarse al Portal de sitios web de asignaturas (ASIG) de la Facultad de Ciencias de la Universidad Central de Venezuela.

El conjunto de objetivos a resolver dentro del marco de la plataforma de aprendizaje móvil incluyen:

1. Desarrollar una aplicación móvil con tecnología **Android**.
2. Desarrollar un manejador de contenidos usando un servidor **nginx** como plataforma.
3. Integrar un servidor de *streaming* **RED5** para el manejo de contenido multimedia (audio y video).
4. Diseñar base de datos relacional SQL que permite almacenar la información del usuario y del sistema. Para operar sobre un sistema de servidor bases de datos **MariaDB**.

Se ha estructurado el presente documento de la siguiente forma:

Capítulo 1: En este capítulo se aborda el Problema de la Investigación y Diseño, que es el objeto de estudio del presente Trabajo Especial de Grado, cuya finalidad es desarrollar una plataforma de aprendizaje móvil para la Escuela de Computación de la Facultad de Ciencias de la UCV. Además, se expone el objetivo general y los objetivos específicos de la tesis, así mismo como la justificación e importancia del presente trabajo.

Capítulo 2: En este capítulo se aborda el Marco Conceptual, el cual cubre todas las necesidades conceptuales que apoyan al desarrollo de una plataforma de aprendizaje

móvil, se expone la información técnica del sistema operativo Android de la aplicación móvil, también información asociada al servidor de contenidos nginx, del servidor de base de datos MariaDB y del servidor de *streaming* RED5. Adicionalmente se exponen conceptos asociados a *streaming*, servidor web y bases de datos.

Capítulo 3: En este capítulo se aborda el Marco Aplicativo, el cual expone detalladamente, la metodología seleccionada aplicada a cada una de las fases del desarrollo de la plataforma de aprendizaje móvil de la Escuela de Computación, Facultad de Ciencias de la UCV. Además se detalla cada uno de los procesos y fases utilizados, haciendo referencia a la investigación realizada y a las herramientas involucradas.

Pruebas: En esta sección se desarrollan los resultados en base a gráficos de las distintas preguntas de las encuestas de la aplicación móvil y de la aplicación de administración.

Resultados: En esta sección se desarrollan los resultados del Trabajo Especial de Grado (TEG), que son relacionados con cada objetivo específico que fue planteado al inicio del trabajo.

Conclusiones: En esta sección se desarrollan las conclusiones del Trabajo Especial de Grado(TEG), que son relacionados con el objetivo general de la investigación, la necesidad o problema del usuario.

Trabajos Futuros: En esta sección se desarrollan las recomendaciones que aportan una sugerencia a trabajos futuros en base a este Trabajo Especial de Grado.

Capítulo 1: Problema de Investigación

En este capítulo se abarca las bases del presente Trabajo Especial de Grado (TEG), tal como es el planteamiento del problema, el objetivo general y los objetivos específicos de la tesis, así como también la justificación e importancia del presente trabajo.

1. Planteamiento del Problema

Algunos de los miembros del cuerpo docente de la Escuela de Computación de la Facultad de Ciencias utilizan diferentes aplicaciones para dar poder compartir el contenido de las materias que imparten a sus estudiantes. Algunas de estas aplicaciones permiten compartir documentos pero otras no permiten compartir videos. No existe una propuesta apropiada que pueda solventar este problema, entonces esto ha llevado a que los profesores utilicen o recurran a otras vías menos eficientes para compartir el contenido de las materias que imparten. Dichas soluciones van desde enviar contenidos por correo electrónico, hasta utilizar páginas web de las materias y, en última instancia, acudir a plataformas de *E-learning* como Moodle.

Algunos profesores creen que en Moodle no pueden desplegar videos desde la misma aplicación, esto se erige en un muro en el cual los docentes se encuentran atrapados ante la necesidad de poder compartir, por ejemplo, sus clases a través de videos en el caso de que las tuviesen grabadas.

Existen aplicaciones móviles (algunas no gratuitas) que dan solución a estos problemas pero muchas deben hacer uso de datos, lo cual causa un impacto negativo para nosotros debido a que las redes de nuestro país no siempre presentan el rendimiento mínimo requerido por estas aplicaciones y además representan un coste adicional elevado para los que utilizan la aplicación.

Actualmente los estudiantes no poseen una herramienta que les permita tener acceso a todo el contenido de sus materias a través de una aplicación móvil.

Con el fin de dar respuesta a estas limitaciones y servir como apoyo para el aprendizaje de los estudiantes, se emprende este Trabajo Especial de Grado.

2. Objetivos

2.1 Objetivo General

- Desarrollar una plataforma de aprendizaje móvil para apoyar el proceso de enseñanza y aprendizaje en la Escuela de Computación de la Facultad de Ciencias de la UCV.

2.2 Objetivos Específicos

1. Desarrollar una aplicación móvil con tecnología **Android**.
2. Desarrollar una página web que permita la administración de contenidos de la plataforma de aprendizaje con el uso de un servidor **nginx**.
3. Configurar servidor de *streaming* **RED5** para el manejo de contenido multimedia (audio y video).
4. Diseñar base de datos relacional *SQL* que permite almacenar la información del usuario y del sistema. Para operar sobre un sistema de servidor bases de datos **MariaDB**.

3. Justificación e Importancia

Ante el auge que han venido tomando las plataformas de aprendizaje en línea mejor conocidas como *E-Learning*, las cuales han sido adoptadas con mayor fuerza por distintas Universidades a nivel mundial, se propone la implementación de una plataforma de aprendizaje móvil o *M-Learning*, que sirva como una herramienta opcional, se espera en un futuro imprescindible, para el aprendizaje por parte de los alumnos de la Escuela de Computación. Tomando como punto de partida la inexistencia en Venezuela de una herramienta móvil que apoye al proceso de educación y que a su vez facilite el proceso de enseñanza al profesorado de la Escuela de Computación. Dicha propuesta se basa en el desarrollo de una aplicación móvil que

despliegue contenidos multimedia como videos y audios, que permita a los estudiantes aprovechar la ubicuidad que aportan las tecnologías móviles, permitiendo al alumnado prácticamente aprender en infinidad de lugares en cualquier momento. Así mismo, se busca que la página web de la plataforma de aprendizaje, posibilite al profesorado administrar contenidos como crear o eliminar materias, subir contenidos multimedia y documentos en la medida de sus necesidades.

Capítulo 2: Marco Conceptual

En este capítulo se cubren todas las necesidades conceptuales que apoyan al desarrollo de una plataforma de aprendizaje.

2.1 *E-Learning*

Se entiende como desarrollo del proceso de formación a distancia, basado en el uso de las tecnologías de información y telecomunicaciones o que posibilitan un aprendizaje interactivo, flexible y accesible a cualquier receptor potencial [Cabrero,2006].

El concepto de *E-Learning* (o de otros similares como tele-formación, educación virtual, cursos online, enseñanza flexible, educación web, docencia en línea, entre otros) es un método de formación que consiste en el diseño, puesta en práctica y evaluación de un curso o plan de formación desarrollado a través de redes de dispositivos y que puede definirse como una educación o formación ofrecida a individuos que están geográficamente separados o que interactúen en tiempos diferidos, de forma asíncrona, del docente empleando los recursos informáticos y de telecomunicaciones. Lo característico del *E-Learning* es que el proceso formativo tiene lugar totalmente o en parte a través de una especie de un entorno virtual de aprendizaje en el cual tiene lugar la interacción profesor-alumno así como las actividades de los estudiantes con los materiales de aprendizaje [Cabrero,2006].

En síntesis, *E-Learning* se refiere a la utilización de las tecnologías de Internet para ofrecer un conjunto de propuestas que permitan incrementar el conocimiento y la práctica educativa [Rosenberg,2001].

2.2 Mobile Learning

En la actualidad, *E-Learning* se está expandiendo rápidamente por todo el sistema educativo impregnando tanto a la educación formal como a la no formal, tanto en la enseñanza presencial como en la formación a distancia. También al campo de la educación superior, de personas adultas, de los ámbitos empresariales y ocupacional, de las enseñanzas medias, entre otros, ofrecen sus cursos no sólo en las modalidades tradicionales de aulas físicas, sino también a través de lo que se conoce como aulas virtuales [O'Malley, Vavoula, G., Glew, J.P., Taylor, J., Sharples, M., & Lefrere, P, 2003].

Sin embargo, cada día se hace más común el uso del dispositivo móvil a nivel mundial, es decir, el mismo está plenamente integrado en la vida cotidiana de las personas, prácticamente se puede asegurar, que es lo último que se mira antes de ir a dormir y lo primero después de levantarse. Además, no se debe dejar de lado su ubicuidad, se puede disponer del dispositivo móvil las 24 horas del día, por lo tanto, es factible aprender en casa, en el parque e infinidad de lugares. Es por esto, que en la actualidad el dispositivo móvil constituye una herramienta estrella para el uso educativo, dando lugar a una nueva metodología de aprendizaje conocida como *mobile learning* que se cree, en un futuro no tan lejano reemplazará por completo a la metodología *E-learning*, *mobile learning*, no es otra cosa que *E-learning* a través de dispositivos móviles de comunicación [Quinn, 2000].

Se puede conceptualizar el *mobile learning* como la modalidad educativa que facilita la construcción del conocimiento, la resolución de problemas de aprendizaje y el desarrollo de destrezas o habilidades diversas de forma autónoma y ubicua gracias a la mediación de dispositivos móviles portables [Brazuelo, 2011].

2.2.1 Características del Mobile Learning [Universidad, 2013]

- Ubicuo: Posibilidad de acceso sin importar lugar ni momento.
- Flexible: Es adaptable a la necesidad de cada usuario o aprendiz.

- **Portable:** Los dispositivos móviles permiten que puedan ser trasladados con mayor facilidad dependiendo de la movilidad del usuario.
- **Inmediato:** Los dispositivos móviles que se llevan siempre consigo mismo pueden acceder a ellos en cualquier momento y por lo tanto también a estas herramientas.
- **Motivante:** Resulta para el estudiantado es atractivo el uso de aplicaciones en el móvil.
- **Accesible:** En comparación con otras herramientas tecnológicas los costos tienden a ser más bajos.
- **Activo:** Involucra de mayor manera al alumno con el aprendizaje.
- **Conectividad a internet:** Permite el uso de obtener información de la red.
- **Acceso a aplicaciones:** Permite la utilización de diversas aplicaciones, para el aprendizaje, producción de contenido, etc.
- **Sensores multifunción:** Dispone de herramientas tales como acelerómetro, GPS, cámara, que pueden enriquecer los procesos de aprendizaje.
- **Personales:** Son propios de cada usuario, crea una relación personal hacia el mismo.
- **Pantalla táctil:** Permite una interacción diferente con la aplicación, como agrandar o disminuir el tamaño de una imagen.

2.2.2 Ventajas del *Mobile Learning* [Universidad,2013]

- Permite el acceso a la información cuándo y dónde sea necesario.
- Podría favorecer un aprendizaje centrado en el alumno y en el contexto.
- Permite la multifuncionalidad de las actividades, bajo distintos sensores (video, acelerómetros, cámara, etc).
- Aumenta la motivación del alumno al utilizar una aplicación en un dispositivo al cual está familiarizado en su uso.
- Permite la utilización de juegos como apoyo a la enseñanza.
- Permiten una evaluación formativa y sumativa.

2.2.3 Retos para su uso en educación tradicional

Pantallas pequeñas: Actualmente las pantallas de los teléfonos inteligentes, y las tabletas tienen un tamaño mucho mayor. Lo importante es adaptar los contenidos y actividades a un tamaño de pantalla estándar.

Introducción de datos en la interfaz del usuario: Existen algunas soluciones como el reconocimiento de voz y el reconocimiento de trazos de escritura natural.

Fuente de distracción: Es importante dejar claro al inicio de la actividad las indicaciones de su uso, ya que el uso del dispositivo móvil puede involucrar otras actividades que no corresponden a las de la aplicación del aprendiz, las cuales puedan interferir en el proceso de enseñanza.

Las capacidades de cálculo y procesamiento son limitadas. Con las innovaciones tecnológicas en los teléfonos inteligentes, cada vez aumentan más las capacidades de cálculo y procesamiento.

Autonomía de la batería: Actualmente las baterías son poco duraderas, aunque se están creando baterías de carga más prolongada. Las tabletas tienen mayor duración de la batería que los teléfonos inteligentes.

Limitación en el almacenamiento: Con las posibilidades que ofrece el *cloud computing* (nube) pueden permitirse gran cantidad de almacenamiento de contenido fuera del dispositivo, sin la necesidad de consultar todos los contenidos localmente en los dispositivos [M Sharples, J Taylor, G Vavoula,2005.].

2.3 Android

En los últimos años los teléfonos móviles han experimentado una gran evolución, desde los primeros terminales, grandes y pesados, pensados sólo para hablar por teléfono en cualquier parte, a los últimos modelos, con los que el término “medio de comunicación” se queda bastante pequeño. Es así como nace Android, el cual es un sistema operativo y una plataforma, basado en Linux para dispositivos móviles con pantalla táctil, como teléfonos inteligentes o tabletas, y así como también para relojes inteligentes, televisores y automóviles. Android permite programar en un entorno de trabajo (*framework*) de Java, aplicaciones sobre una máquina virtual Dalvik (una variación de la máquina de Java con compilación en tiempo de ejecución). Además, en lo que lo diferencia de otros sistemas operativos, es que cualquier persona que sepa programar puede crear nuevas aplicaciones, *widgets*, o incluso, modificar el propio sistema operativo, dado que Android es de código libre, por lo que sabiendo programar en lenguaje Java, va a ser muy fácil comenzar a programar en esta plataforma.

Fue desarrollado por Android Inc., empresa que en el 2005 fue comprada por Google, aunque no fue hasta 2008 cuando se popularizó, gracias a la unión al proyecto de *Open Handset Alliance*, un consorcio formado por 48 empresas de desarrollo hardware, software y telecomunicaciones, que decidieron promocionar el software libre. Pero ha sido Google quien ha publicado la mayor parte del código fuente del sistema operativo, gracias al *software* Apache, que es una fundación que da soporte a proyectos software de código abierto [Tomás,2015].

2.3.1 Arquitectura Android

Núcleo Linux: El núcleo de Android está formado por el sistema operativo Linux versión 2.6. Esta capa principalmente proporciona servicios tales como seguridad,

manejo de memoria, multiprocesos, pila de protocolos y soporte de drivers para dispositivos.

Runtime de Android: Está basado en el concepto de máquina virtual que utiliza Java. Dado limitaciones que existen en los dispositivos donde podría correr Android (poca memoria y procesamiento limitado) no fue posible utilizar una máquina virtual de Java. Google tomó la decisión de crear una nueva máquina virtual “Dalvik”, que se adecuara mejor a distintas limitaciones .

Una de las características de Dalvik, es facilitar la optimización de recursos los cuales son: que ejecuta ficheros Dalvik (dex), que es un formato optimizado para ahorrar memoria. Además, está basada en registros. Cada aplicación se ejecuta en su propio proceso Linux como una instancia de la máquina virtual Dalvik, delegando al núcleo de Linux funciones de hilos y manejo de memoria a bajo nivel [Tomás,2015]..

Librerías nativas: Incluye un conjunto de librerías de C/C++ usadas en varios componentes de Android. Se encuentran compiladas en código nativo al procesador. Muchas de estas librerías utilizan proyectos de código abierto, entre las librerías tenemos [Tomás,2015].:

- *Media FrameWork:* Librería basada en *Packet Video's Open CORE:* soporta *codecs* para la reproducción y grabación de diferentes formatos de audio y video.
- *-WebKit:* Soporta un navegador Web del navegador Android.
- *-SQLite:* Potente y ligero motor de base de datos relacionales disponible para todas las aplicaciones.
- *-SSL:* Proporciona servicios de encriptación “*Secure Socket Layer*”.

Entorno de aplicación: Proporciona una plataforma de libre desarrollo para aplicaciones con grandes riquezas e innovaciones.

Esta capa ha sido diseñada para simplificar la reutilización de componentes. Las aplicaciones pueden publicar diferentes capacidades para que otras aplicaciones puedan hacer uso de ellas, bajo este mecanismo permite a los usuarios reemplazar componentes .

Una de las mayores fortalezas del entorno de aplicación de Android es que se aprovecha el lenguaje de programación Java. El SDK de Android no ofrece todo lo disponible para su estándar de entorno de ejecución Java (JRE), pero es compatible con una fracción muy importante de la misma.

Los servicios más importantes que incluye son:

- *Views*: Extenso conjunto de vistas.
- *Resource Manager*: Proporciona acceso a los recursos que no son precisamente códigos.
- *Activity Manager*: Maneja el ciclo de vida de las aplicaciones y proporciona un sistema de navegación entre ellas.
- *Notification Manager*: Permite a las aplicaciones mostrar ciertas alertas que pueden ser personalizadas en la barra de estado.
- *Content Providers*: Mecanismo simple para permitir el acceso a los datos de otras aplicaciones.

Aplicaciones: Este nivel está formado por todo un conjunto de aplicaciones instaladas en una máquina virtual Android. Todas las aplicaciones deben correr en la máquina virtual Dalvik para garantizar la seguridad y funcionalidad del sistema.

Por lo general las aplicaciones Android se escriben en Java. Para desarrollar aplicaciones en Java podemos utilizar el Android SDK. Existe otra opción que es desarrollar aplicaciones con C o C++, para esto podemos utilizar el *Android NDK* “*Native Development Kit*” [Tomás,2015].

2.4 Streaming

Tecnología de transmisión a través de redes de medios continuos (principalmente audio y vídeo), donde no existe descarga de información a un disco local, sino que se envía la información a través de la red y el cliente la reproduce en tiempo real al recibirla [Ministerio,2008]. El término *stream*, refiere al proceso de entregar la data de forma continua.

Este tipo de tecnologías basa su funcionamiento en un buffer de datos que va almacenando lo que se va descargando desde el usuario para luego mostrarle el material que está siendo descargado, esto se opone al mecanismo tradicional de descarga de archivos que requiere que el usuario descargue por completo los archivos para poder acceder a su contenido [Sciara,2004].

La ventaja que existe en los clientes que reproducen *streaming* es principalmente poder reproducir una secuencia de video o sonido, mientras que se descarga con una conexión de Internet, como por ejemplo reproducir data como una película, sin la necesidad que esta se haya transmitido en su totalidad [Ministerio,2008].

Según el Instituto Superior de Formación y Recursos en Red para el Profesorado (2008). El *streaming* puede ser de dos tipos dependiendo de la tecnología instalada en el servidor:

- Descarga progresiva. Se produce en servidores web que disponen de *Internet Information Server (IIS)*, Apache, Tomcat, etc. El archivo de vídeo o audio solicitado por el cliente es liberado por el servidor como cualquier otro archivo utilizando el protocolo HTTP. Sin embargo, si el archivo ha sido especialmente empaquetado para *streaming*, al ser leído por el reproductor cliente, se iniciará en *streaming* en cuanto se llene el buffer.

- Transmisión por secuencias. Se produce en servidores multimedia que disponen de un software especial para gestionar óptimamente el *streaming* de audio y vídeo: *Windows Media Server*, *Flash Communication Server*, etc. La utilización de un servidor multimedia ofrece múltiples ventajas frente al servidor web. Las más destacadas son:
 - Mayor rapidez en la visualización de este tipo de contenidos.
 - La comunicación entre servidor/cliente se puede realizar por protocolos alternativos al HTTP. Tiene el inconveniente del bloqueo impuesto por *Firewalls* pero tiene la ventaja de una mayor rapidez.
 - Mejor gestión del procesador y ancho de banda de la máquina del servidor ante peticiones simultáneas de varios clientes del mismo archivo de audio o vídeo.
 - Control predefinido sobre la descarga que pueden realizar los clientes: autenticada, filtrada por IP, sin almacenarla en la caché del cliente, etc.
 - Mayor garantía de una reproducción ininterrumpida gracias al establecimiento de una conexión de control inteligente entre servidor y cliente.
 - Posibilidad de distribución de transmisiones de audio y vídeo en directo.

2.5 RTMP “Real Time Messaging Protocol”

Es un protocolo que provee un servicio de múltiples envíos de mensajes de forma bidireccional, intencionalmente fue creado para llevar paralelamente *streams* de videos, audio y mensajes con datos, con información de tiempo asociada. La implementación típicamente asigna diferentes prioridades a diferentes clases de mensajes, el cual afecta el orden de cómo es cada mensaje es encolado, para bajar la exigencia a la capa de transporte cuando este está exigido [H. Parmar, Ed. M. Thornburgh, Ed.,2012].

2.5.1 Funcionamiento

RTMP es un protocolo basado en TCP, por lo tanto mantiene conexiones persistentes y permite comunicaciones de baja latencia. Para entregar datos de *streams* suavemente y transmitir tanta información como sea posible. Principalmente el protocolo señala que se debe separar el *stream* en fragmentos y que el tamaño de dichos fragmentos debe ser dinámicamente negociado entre el cliente y el servidor, en algunos casos el servidor coloca un fragmento fijo. Los fragmentos de diferentes *streams* pueden ser intercalados y multiplexados en una sola conexión. Cabe destacar que el *overhead* es muy bajo ya que es solo de 1 byte el encabezado de cada fragmento. Sin embargo, realmente los fragmentos individuales normalmente no se intercalan. Entonces, intercalar y multiplexar se realiza a nivel de paquetes, con RTMP los paquetes atraviesan una gran cantidad de diferentes canales siendo intercalados de alguna manera para asegurar que cada canal cumpla con su ancho de banda, latencia y cualquier medida de calidad de servicio.

El protocolo define una cantidad de canales virtuales por donde los paquetes son enviados y recibidos, los cuales operan independientemente de otros canales. Durante una sesión RTMP, muchos canales pueden ser activados de forma simultánea en cualquier momento. Cuando la data viaja cifrada, se genera un encabezado para el paquete. Ese encabezado especifica el id del canal por el cual fue enviado, una estampa de tiempo de cuando fue generado y el tamaño del carga útil del paquete. Seguido al encabezado está el contenido del carga útil, el cual se encuentra fragmentado de acuerdo al tamaño negociado cuando fue enviado [H. Parmar,2012].

2.6 Servidor web

Es un programa que se encarga de procesar una aplicación del lado del servidor, realizando diferentes tareas como conexiones unidireccionales o bidireccionales, síncronas o asíncronas con el cliente y generando una respuesta en algún lenguaje o aplicación del lado del cliente.

Principalmente almacena documentos HTML, imágenes, videos, textos, presentaciones, y en general todo tipo de información, además se encarga de enviar estas informaciones a los clientes.

2.7 nginx

Es un servidor http de código abierto, libre y de alto rendimiento, así como también un servidor proxy reverso y un servidor proxy IMAP/POP3. Igor Sysoev inició el desarrollo de nginx en 2002, con el primer lanzamiento público en 2004. Actualmente nginx gestiona alrededor del 12.18% de los sitios web que se encuentran activos en todos los dominios. Además es conocido por su alto rendimiento, estabilidad, simple configuración y bajo consumo de recursos . Posee otras características que lo hacen estar en el top de funcionalidades de servidores web como:

- Balanceo de carga.
- *Caching*.
- Acceso y control de ancho de banda.
- Puede integrar de forma eficiente una gran cantidad de aplicaciones.

nginx es bastante personalizable para cualquier ocasión, provee características claves necesarias para este fin, las cuales son:

- Manejo de concurrencia.
- Manejo de latencia.
- Provee *SSL (secure sockets layer)*.
- Contenido estático.
- Comprensión.
- *Caching*.
- Emparejamiento de conexiones y solicitudes.

- HTTP media *streaming* desde una capa de aplicación hacia una más eficiente capa de servidor web.
- Se integra también con soluciones *NoSQL*.

El código base de nginx esta originalmente escrito en el lenguaje de programación C, que es soportado por varias arquitecturas y sistemas operativos, incluyendo Linux, FreeBSD, Solaris, Mac OS X, AIX y Microsoft Windows [nginx,2015].

2.7.1 Arquitectura nginx

La arquitectura de nginx se compone principalmente de dos elementos los cuales interactúan entre sí y con un conjunto de recursos para dar respuesta a diferentes solicitudes. Estos elementos son el *Master*, el cual es un único proceso que corre en un solo hilo de ejecución y los *Worker* que son varios procesos que corren en diferentes hilos de ejecución .

nginx ejecuta una gran cantidad de procesos en memoria, en donde existe un único proceso *Master* y muchos procesos *Worker*. También existen un par de procesadores de propósito especial, específicamente un manejador de cache y un cargador de cache. Todos los procesadores son de un solo hilo de ejecución en la versión 1.x de nginx. Los procesos principalmente utilizan mecanismos de compartición de memoria para la comunicación entre procesos. El proceso *Master* se ejecuta como un usuario *root*. El manejador de cache, cargador de cache y los *workers* se ejecutan en un usuario no privilegiado .

El proceso *Master* es responsable de una serie de tareas:

- Lectura y validación de configuración.
- Creación, vinculación y finalización de sockets.
- Comenzar, terminar y mantener un número de procesos *workers* configurados.
- Reconfigurar sin la necesidad de interrumpir el servicio.
- Controlar sin parar las actualizaciones de binarios.
- Reabrir los archivos de registros.

- Compilar scripts de Perl embebidos.

Los procesos *workers* aceptan, manejan y procesan conexiones de los clientes, proveen proxy reverso y funcionalidad de filtrado. De manera de monitorear el comportamiento de una instancia de nginx, el sistema administrador debería mantener un ojo sobre los procesos *workers* y que dichos procesos reflejen las operaciones que día a día se realizan en un servidor web.

El procesos de cargador de cache es responsable de chequear los artículos que se encuentran en cache y surtir a la base de datos de nginx con metadatos de cache. Esencialmente, este proceso prepara instancia nginx para trabajar con archivos que ya han sido cargados en disco en una especial estructura de directorio. Este traversa los directorios, chequea el contenido de cache de los metadatos, actualiza las entradas relevantes en memoria y finaliza cuando todo está limpio y listo para usarse.

El proceso de manejador cache es responsable por la caducidad y nulidad. Él está en memoria durante la operación normal de nginx y él es reiniciado por el proceso maestro en caso de que se presente alguna falla [nginx,2015].

Un concepto interesante de nginx frente a otros servidores web, es el hecho de no utilizar un nuevo hilo para cada nueva solicitud, ya que esto en un esquema con miles de solicitudes por segundo provocaría un decaimiento de rendimiento. En vez de ello nginx permite que cada *Worker* pueda multiplexar una gran cantidad de solicitudes dando así una solución a este problema común de otras arquitecturas.

2.8 PHP

PHP (acrónimo recursivo de **PHP: Hyper text Preprocessor**) es un lenguaje de código abierto muy popular especialmente adecuado para el desarrollo web y que puede ser incrustado en HTML.

El gran parecido que posee PHP con los lenguajes más comunes de programación estructurada, como C y Perl, permiten a la mayoría de los programadores crear aplicaciones complejas con una curva de aprendizaje muy

corta. También les permite involucrarse con aplicaciones de contenido dinámico sin tener que aprender todo un nuevo grupo de funciones.

A diferencia de *JavaScript* lo que distingue a PHP es que el código es ejecutado en el servidor, generando HTML y enviándolo al cliente. El cliente recibirá el resultado de ejecutar el *script*, aunque no sabrá el código subyacente que era. El servidor web puede ser configurado incluso para que procese todos los ficheros HTML con PHP, por lo que hay manera que los usuarios puedan saber que se tiene debajo de la manga.

PHP entre sus funcionalidades permite [Comunidad PHP,2015]:

- Autenticación HTTP.
- Manejo de *Cookies*.
- Manejo de sesiones.
- Manejo de *XForms*.
- Manejo de carga de archivos.
- Manejo de conexiones.
- Conexiones persistentes a Bases de Datos.
- Funcionamiento de línea de comandos.
- Recolección de basura.

2.9 RED5

Red5 es un servidor *Open Source* para entregar contenido en *streaming* en Flash. Para ello utiliza el protocolo RTMP (*Real Time Messaging Protocol*) con lo cual se puede transmitir contenido en tiempo real.

Utiliza la sintaxis de *Action Script Communication* con la cual se pueden desarrollar aplicaciones de comunicación en tiempo real.

El proyecto RED5 fue realizado el 31 de agosto del 2005 por el grupo de *Open Source Flash community*. A partir de esta fecha se agregaron funcionalidades al proyecto, tales como [Gong,Gregoire,Rossi(2008)]:

- *Streaming* de Video.
- *Shared Object*.
- Implementación de RTMPT.

2.9.1 Características

Este servidor está escrito enteramente en Java y tiene las siguientes características:

- *Streaming* de Audio y Video (FLV y MP3).
- Facilidad para guardar *Streaming* del cliente.
- *Shared Objects*.
- Publicación *Live Streaming*.
- Soporte para AMF o *Flash Remoting*.

En términos más sencillos, gracias al soporte de transmisión de datos en tiempo real se podría implementar [Gong,2008]::

- Chats multiusuario en tiempo real.
- *Streaming* de video y Mp3, pudiéndose reproducir parte del vídeo sin necesidad de descargar el vídeo completo.
- Transmisión de señal de televisión en tiempo real.
- Chats con soporte de audio y vídeo, con la cual los usuarios pueden transmitir el contenido de su webcam.
- Facilidad para implementar pantallas compartidas, en la cual múltiples usuarios pueden interactuar en una misma pantalla.
- Grabación de contenido en formato FLV.
- Protección de derechos de Autor (DRM) pues no se transmiten los archivos físicos, sino que se envía contenido en *streaming*.

- Análisis y reportes de datos en tiempo real.

2.9.2 Ventajas [Gong,2008]:

- Es libre.
- La instalación es sencilla.
- Multiplataforma.
- Es muy estable, si un cliente genera una excepción el servidor solo falla en la conexión en la cual se generó y todos los demás clientes se mantienen sin problemas.

2.9.3 Desventajas [Gong,2008]:

- Falta de información sobre la creación de proyectos para los usuarios novatos.
- Para trabajar con funciones de clases java se debe modificar el jar con el que se ejecuta el servidor de red5.

2.10 Bases de datos

Se pueden definir las bases de datos como una colección de datos relacionados [Silberschatz,Korth,Sudarshan,2002].

Con base a la definición anterior una base de datos "BD" se define como: un almacén o repositorio centralizado (de cualquier tamaño y complejidad) de datos organizados y relacionados lógicamente para un propósito específico (aportar información de diversa índole dependiendo del ámbito de la BD), que permite la sencilla manipulación y obtención de los mismos [Elmasri,Navathe,2007].

Se puede afirmar que las BD se clasifican en:

Según la variabilidad de los datos en:

- Estáticas: solo permiten la lectura y almacenamiento de datos.

- Dinámicas: aquellas donde la información almacenada se modifica con el tiempo, es decir que la información se puede actualizar, añadir, eliminar, entre otros.

Y según el modelo de datos que utilicen en:

- BD Orientada a objetos (BDOO): son aquellas que siguen el paradigma de programación orientada a objetos (POO), las entidades que se utilizan en él son equivalentes a las usadas en este tipo de BD, con la diferencia de que en los programas orientados a objetos éstas desaparecen en el momento en que termina su ejecución mientras que las de este estilo de BD persisten. En éstas también se incorporan los conceptos de encapsulación (ocultar información al resto de los objetos para impedir accesos incorrectos o conflictos), herencia (propiedad a través de la cual los objetos heredan comportamientos de otros objetos con una jerarquía mayor), y polimorfismo (propiedad de aplicar una operación a diferentes tipos de objetos, la operación recibe el mismo nombre pero diferentes parámetros.). Las BDOO son utilizadas normalmente cuando se necesita manipular tipos de datos complejos.
- BD Relacional: cumple con las reglas del modelo de datos relacional, el cual consiste de un conjunto de tablas que representan relaciones entre los datos. La forma en que se almacenan los datos en ésta no tiene relevancia.
- BD Jerárquica: es aquella que almacena su información en una estructura de jerarquía. Aquí, los datos se organizan de manera análoga a un árbol visto desde la raíz hasta las hojas y se representan mediante registros llamados nodos que contienen campos llamados atributos. Los nodos padres pueden tener varios hijos pero cada hijo sólo tiene un padre, el nodo que no tenga se denomina nodo raíz. Las relaciones se representan a través de lazos, enlaces o ligas. Son útiles en el caso de aplicaciones que manejen una gran cantidad de información compartida. La desventaja principal es que no se garantiza la inexistencia de duplicidad de los datos. Este tipo de BD ya no es comúnmente utilizado, incluso se puede decir que es obsoleto.

- BD de Red: es similar a la jerárquica, ya se sabe que un nodo padre puede tener múltiples hijos pero en este modelo se añade que un nodo hijo puede tener múltiples padres (no se puede en la jerárquica). Esto ayudo a disminuir la duplicidad de los datos, sin embargo, no se logró un manejo totalmente eficiente de la información. También es considerada obsoleta.

2.11 SQL

El lenguaje de consulta estructurado o SQL (por sus siglas en inglés *structured query language*) es un lenguaje declarativo de acceso a bases de datos relacionales que permite especificar diversos tipos de operaciones en estas. Una de sus características es el manejo del álgebra y el cálculo relacional permitiendo efectuar consultas con el fin de recuperar de una forma sencilla información de interés de una base de datos, así como también hacer cambios sobre ella.

2.12 ACID

- Atomicidad: es la propiedad que asegura que la operación se ha realizado o no, y por lo tanto ante un fallo del sistema no puede quedar a medias.
- Consistencia: Integridad. Es la propiedad que asegura que sólo se empieza aquello que se puede acabar. Por lo tanto se ejecutan aquellas operaciones que no van a romper las reglas y directrices de integridad de la base de datos.
- Aislamiento: es la propiedad que asegura que una operación no puede afectar a otras. Esto asegura que la realización de dos transacciones sobre la misma información sean independientes y no generen ningún tipo de error.
- Durabilidad: es la propiedad que asegura que una vez realizada la operación, ésta persistirá y no se podrá deshacer aunque falle el sistema.

2.13 MariaDB

MariaDB es un sistema manejador de bases de datos relacionales SQL que nace como un reemplazo a MySQL que pasó a ser privativa. MariaDB se inclina para

ser la elección lógica por profesionales en el área que buscan robustez, escalabilidad y seguro servidor SQL. Para llevar a cabo ello, posee una fundación que de forma cercana y cooperativa con una gran comunidad de usuarios y desarrolladores de espíritu libre y de software de código abierto [Comunidad Mariadb,2015].

Características que ofrecen sobre MySQL:

- Más motores de almacenamiento.
- Mayor velocidad de consultas.
- Extensiones.
- Mejores pruebas de calidad.
- Código realmente abierto.
- Menos errores y alertas.

2.14 Android Studio

Es la herramienta que se seleccionó para desarrollar la aplicación en Android. Es el ambiente integrado de desarrollo oficial para el desarrollo de aplicaciones Android, basado en IntelliJ IDEA, esta herramienta ofrece:

- Un sistema de construcción basado en un *Gradle* flexible.
- Construir variables y generación múltiple de archivos apk.
- Plantillas de código que ayudan a construir características comunes de aplicaciones.
- Un editor de diseños rico que permiten la edición de temas arrastrando y arrojando.
- Herramientas *"lint"* para detectar problemas de desempeño, usabilidad, compatibilidad de versiones.

2.14.1 Proyecto y estructura del archivo

Por defecto, *Android Studio* muestra los archivos del proyecto en el *"Android Project view"*. Esta vista muestra una versión resumida de la estructura real del proyecto, proveyendo un rápido acceso a las fuentes de archivos claves.

- Muestra la más importante fuente de directorios al tope del nivel de la jerarquía del módulo.
- Agrupa los archivos de construcción de todos los módulos en una carpeta común.
- Agrupa todos los archivos de manifiesto de cada módulo en una carpeta común.

La siguiente imagen tomada del sitio oficial de los desarrolladores Android [Developers Android,2015] muestra la estructura de un proyecto Android:

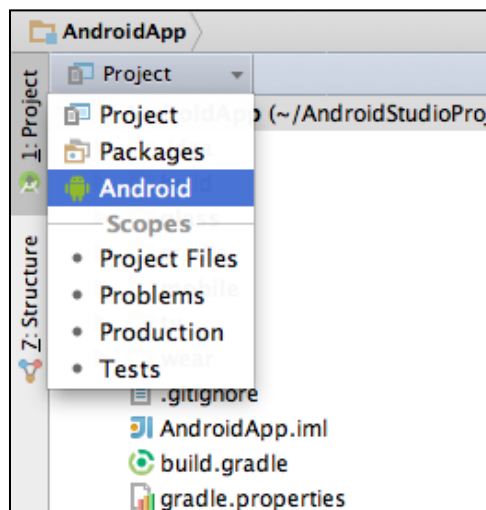


Figura 1: Estructura proyecto Android.

La siguiente imagen tomada del sitio oficial de los desarrolladores Android [Developers Android,2015] muestra el *Gradle* de un proyecto Android:

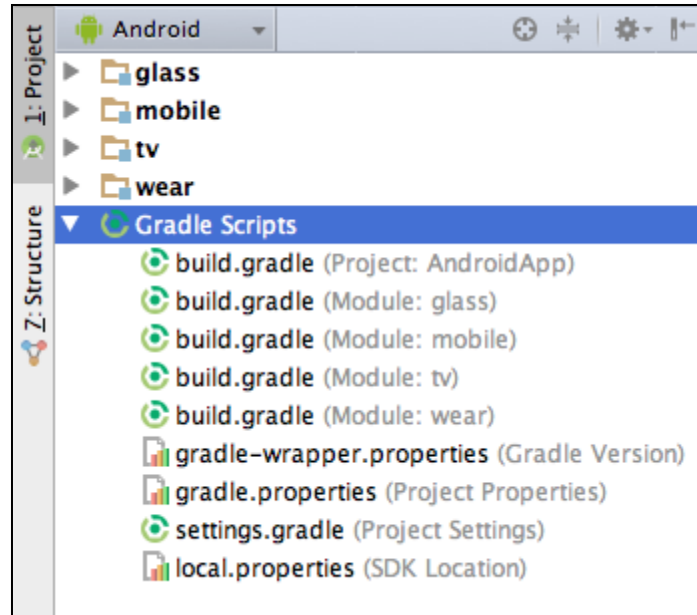


Figura 2: *Gradle* de Android.

2.14.2 Sistema de construcción Android

El sistema de construcción Android se usa para construir, probar, ejecutar y empaquetar las aplicaciones. Este sistema reemplaza el sistema usado con Eclipse, puede ser ejecutado como una herramienta integrada desde el menú de Android Studio e independiente de la línea de comando. Se puede utilizar las características de este sistema de construcción para:

- Personalizar, configurar y ampliar el proceso de construcción.
- Crear múltiples APK para la aplicación con diferentes características usando el mismo proyecto y sus módulos.
- Reutilizar código y recursos a través de conjuntos de fuentes.

2.15 Método de desarrollo de la plataforma de aprendizaje

Desde el análisis de metodologías que se realizó en el trabajo de seminario, se seleccionó la siguiente metodología para el desarrollo de la plataforma de aprendizaje.

2.15.1 Metodología de Roger Pressman

De acuerdo con Roger Pressman, las etapas metodológicas a llevar a cabo para el desarrollo de Aplicaciones de Software, se establecen de la siguiente manera:

- Análisis.
- Diseño.
- Codificación.
- Prueba.
- Mantenimiento.

La siguiente figura muestra el proceso de desarrollo de aplicaciones de software [Barrios,Romero,2015]:

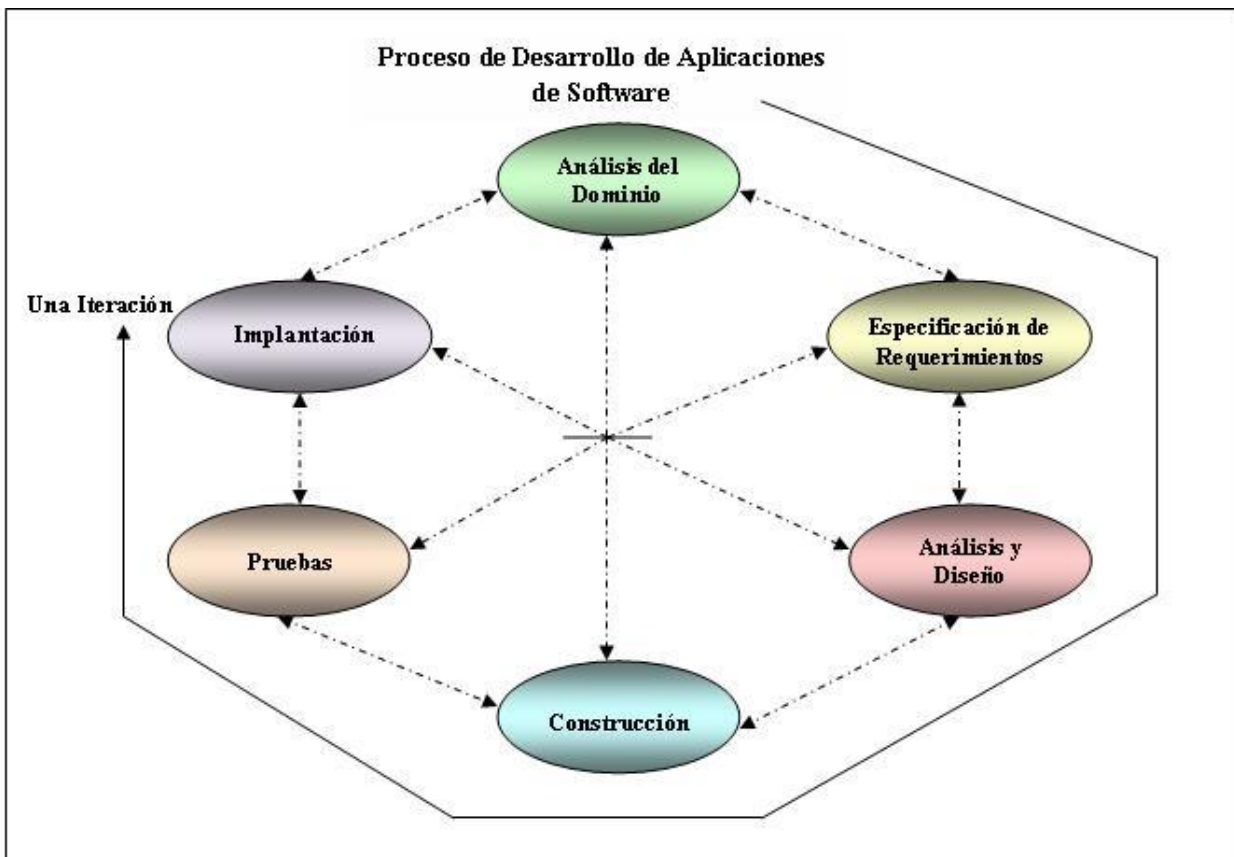


Figura 3: Proceso de desarrollo de aplicaciones.

2.15.2 Análisis de los requisitos del software

El proceso de reunión de requisitos se intensifica y se centra especialmente en el software. Dentro del proceso de análisis, es fundamental que a través de una colección de requerimientos funcionales y no funcionales, el desarrollador o desarrolladores del software comprendan completamente la naturaleza de los programas que deben construirse para desarrollar la aplicación, la función requerida, comportamiento, rendimiento e interconexión. Es de suma importancia que antes de empezar a codificar los programas, se tenga una completa y plena comprensión de los requisitos del software.

Pressman establece que la tarea del análisis de requisitos es un proceso de descubrimiento, refinamiento, modelado y especificación. Se refina en detalle el ámbito del software, y se crean modelos de los requisitos de datos, flujo de información y control, y del comportamiento operativo. Se analizan soluciones alternativas y se asignan a diferentes elementos del software. El análisis de requisitos permite al desarrollador o desarrolladores especificar la función y el rendimiento del software, indica la interfaz del software con otros elementos del sistema y establece las restricciones que debe cumplir el software.

El análisis de requisitos del software puede dividirse en cinco áreas de esfuerzo, que son [Barrios,2015]:

- Reconocimiento del problema: Reconocer los elementos básicos del problema tal y como los perciben los usuarios finales.
- Evaluación y síntesis: Definir todos los objetos de datos observables externamente, evaluar el flujo y contenido de la información, definir y elaborar todas las funciones del software, entender el comportamiento del software en el contexto de acontecimientos que afectan al sistema.
- Modelado: Crear modelos del sistema con el fin de entender mejor el flujo de datos y control, el tratamiento funcional y el comportamiento operativo y el contenido de la información.

- Especificación: Realizar la especificación formal del software.
- Revisión: Un último chequeo general de todo el proceso.

2.15.3 Diseño

Según Pressman, el diseño del software es realmente un proceso de muchos pasos pero que se clasifican dentro de uno mismo. En general, la actividad del diseño se refiere al establecimiento de las estructuras de datos, la arquitectura general del software, representaciones de interfaz y algoritmos. El proceso de diseño traduce requisitos en una representación de software.

El diseño es el primer paso en la fase de desarrollo de cualquier producto o sistema de ingeniería. De acuerdo con Pressman, el objetivo del diseño es producir un modelo o representación de una entidad que se va a construir posteriormente.

El diseño, es la primera de las tres actividades técnicas que implica un proceso de ingeniería de software; estas etapas son diseño, codificación y pruebas. Generalmente la fase de diseño produce un diseño de datos, un diseño arquitectónico, un diseño de interfaz, y un diseño procedimental.

El diseño de datos esencialmente se encarga de transformar el modelo de dominio de la información creada durante el análisis. En el diseño arquitectónico se definen las relaciones entre los principales elementos estructurales del programa. Para una herramienta de software basada en el desarrollo e implementación de ambientes virtuales éste es un aspecto fundamental dado que en esta representación del diseño se establece la estructura modular del software que se desarrolla. El diseño de interfaz describe cómo se comunica el software consigo mismo, con los sistemas que operan con él, y con los operadores que lo emplean [Barrios,2015].

2.15.4 Generación de Código:

Esta actividad consiste en traducir el diseño, en una forma legible por la máquina. La generación de código se refiere tanto a la parte de generación de los ambientes virtuales, como a la parte en la cual se añadirá comportamiento a estos ambientes. Por ejemplo, el lenguaje de programación VRML 2.0 es un lenguaje de modelado en 3D en el cuál se dibuja por medio de generar código de programación de formato y marcado para especificar las características del objeto u objetos que se van agregando a un mundo o entorno virtual. El comportamiento de las escenas virtuales es decir, su funcionalidad, se puede construir a través de algún otro lenguaje de programación, como clases Java o *scripts* especificados en *JavaScript*. Todas estas actividades implican generar código [Barrios,2015].

2.15.5 Pruebas

Una vez que se ha generado código, comienzan las pruebas del software o sistema que se ha desarrollado. De acuerdo con Pressman, el proceso de pruebas se centra en los procesos lógicos internos del software, asegurando que todas las sentencias se han comprobado, y en los procesos externos funcionales, es decir, la realización de las prueba para la detección de errores. En el caso de una herramienta de software, es necesario tener etapas de pruebas tanto para la parte funcional del software, como para la parte aplicativa del mismo.

Se requiere poder probar el software con aplicaciones reales que puedan evaluar el comportamiento del software, con el fin de proporcionar retroalimentación a los desarrolladores. Es sumamente importante que durante el proceso de desarrollo no se pierda el contacto con los interesados o solicitantes del desarrollo de software, de esta manera los objetivos de proyecto se mantendrán vigentes y se tendrá una idea clara de los aspectos que tienen que probarse durante el periodo de pruebas [Barrios,2015].

2.15.6 Mantenimiento

El software indudablemente sufrirá cambios, y habrá que hacer algunas modificaciones a su funcionalidad. Es de suma importancia que el software de calidad pueda adaptarse con fines de acoplarse a los cambios de su entorno externo. Por medio de la documentación apropiada y atinada del software se pueden presentar las vías para el mantenimiento y modificaciones al mismo [Barrios,2015].

Capítulo 3: Marco Aplicativo

En este capítulo se describe un resumen sobre la problemática de la Escuela de Computación de la Universidad Central de Venezuela, así como las fases de desarrollo de la plataforma de aprendizaje aplicadas a la solución planteada, haciendo uso de diferentes herramientas como un servidor de contenidos nginx, un servidor de *stream* red5, un servidor de bases de datos MariaDB, servicios web en PHP y *Android Studio*.

3.1 Abordando la Problemática

Actualmente los profesores de la Escuela de Computación de la Universidad Central de Venezuela, no tienen disponible una solución de aplicación móvil propia de la Escuela que les sirva como apoyo al proceso de enseñanza. Los profesores buscan de una aplicación que pueda desplegar videos de una manera eficiente y mostrar también documentos.

Por otro lado los estudiantes, se encuentran en cierta necesidad de poder hacer uso de una aplicación móvil que puedan llevar siempre consigo y que refleje el contenido de las materias que cursan.

Existe un reto agregado al desarrollo de una aplicación móvil en Venezuela, ya que esta no debe causar un gran impacto sobre la plataforma tecnológica del país con respecto a la velocidad de conexión de transferencia de datos que no siempre poseen un buen rendimiento. Es por ello que la aplicación debe de utilizar tecnologías que puedan aligerar este impacto, tecnologías como *stream* para la reproducción de los videos en la aplicación.

Resultaría interesante para algunos profesores que no se trate solo de una aplicación móvil, si no dé también una plataforma tecnológica que de soporte a dicha aplicación móvil, que incluya recursos como tecnologías de servidor que permiten el funcionamiento de la aplicación consultando los contenidos de la aplicación y su base de datos y además la existencia de una página web que permita la administración de la aplicación.

3.2 Análisis de los requerimientos del software

Resulta de gran importancia analizar los requerimientos funcionales y no funcionales del software a desarrollar, para que así el desarrollador entienda la naturaleza de los programas que debe construir para dar ofrecer una solución a estas necesidades.

Dentro del contexto de la problemática y en reuniones con los usuarios finales se desarrolló la siguiente lista de requerimientos funcionales:

- La aplicación móvil debe permitir el registro de diferentes usuarios.
- La aplicación móvil debe permitir iniciar sesión a diferentes usuarios.
- La aplicación móvil debe dar acceso a todas las materias que posea el sistema.
- La aplicación móvil debe permitir reproducir videos.
- La aplicación móvil debe permitir desplegar documentos.
- Se debe implementar una aplicación que permita la administración de contenidos de la aplicación móvil.
- La aplicación de administración debe permitir crear y eliminar materias, así como también de cargar contenidos, como videos o documentos y eliminarlos.

Y una lista de requerimientos no funcionales:

- La aplicación móvil debe hacer uso de tecnologías eficientes para la transmisión de datos.
- El desarrollo de las interfaces de la aplicación sea consistente.
- Las contraseñas deben almacenarse cifradas.
- La aplicación móvil debe ser desarrollada para el sistema operativo Android.
- Desarrollo de una aplicación minimalista, sin muchos elementos que causen distracciones.

Tomando en cuenta la lista anterior de requerimientos, entonces se procede a elaborar los siguientes diagramas de casos de uso de la aplicación móvil (ver Figura 4) y aplicación de administración (ver Figura 5 y 6) que definen y elaboran las principales funciones del software así como los roles que participan:

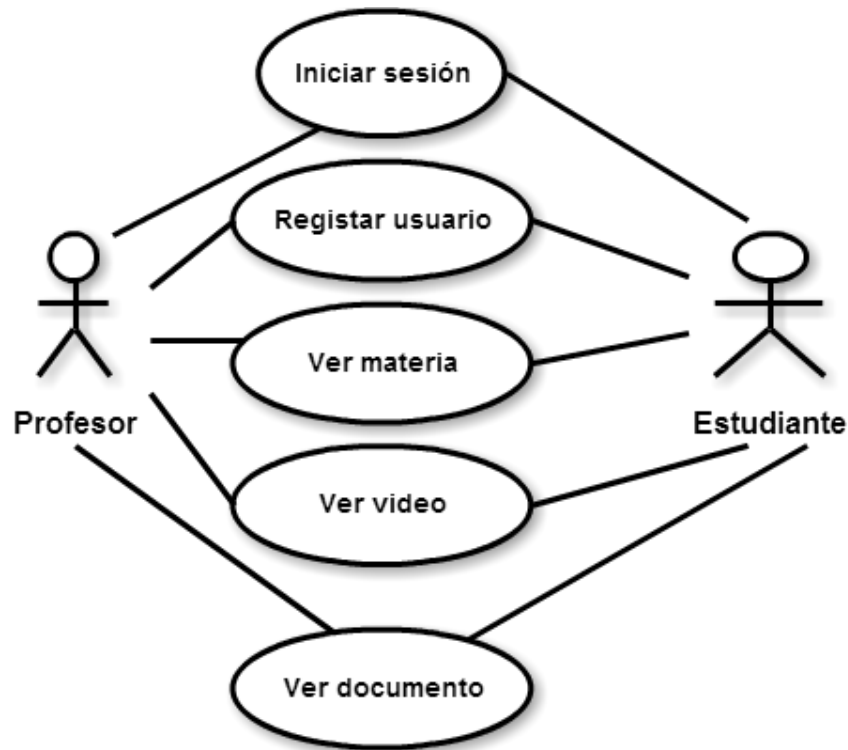


Figura 4: Casos de uso nivel 1 Aplicación móvil.

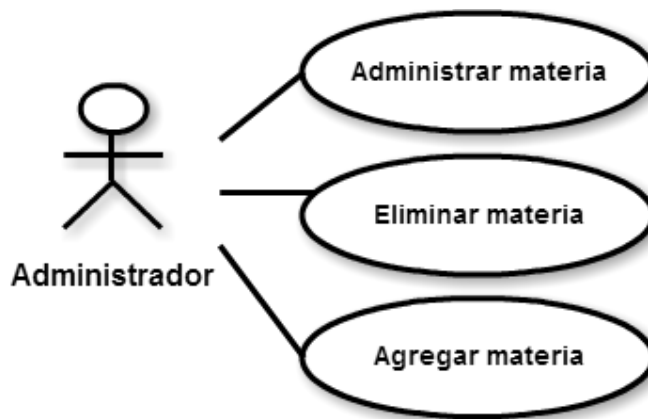


Figura 5: Casos de uso nivel 1 Aplicación de Administración.

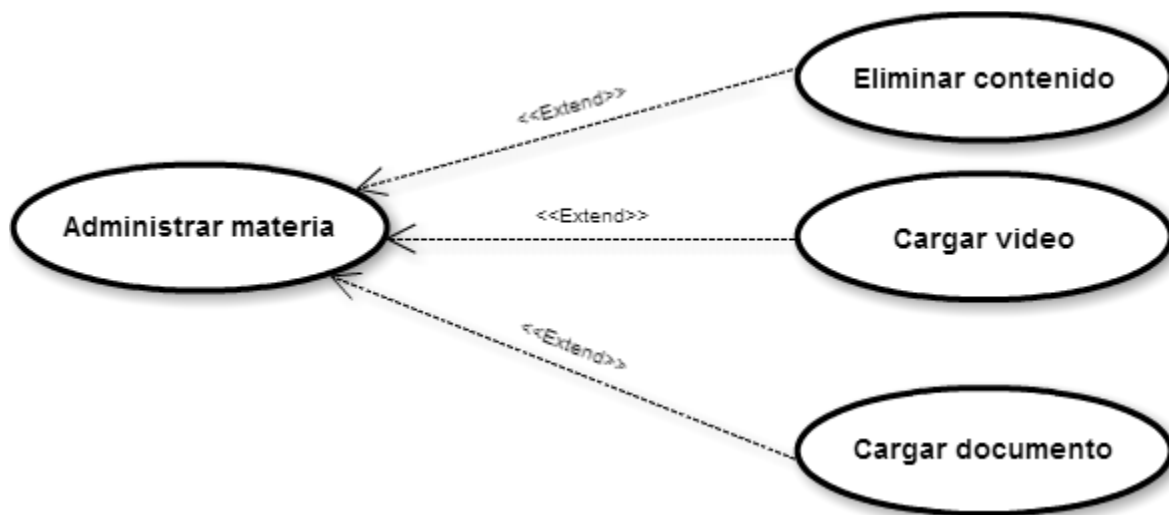


Figura 6: Casos de uso nivel 2 Aplicación de Administración.

3.2.1 Especificación de los casos de uso de la aplicación móvil

A continuación se presenta la especificación de los principales casos de usos, de la aplicación móvil, bajo cuadros que establecen la información para cada uno de ellos.

Caso de uso:	Iniciar sesión.
Actores involucrados:	Profesor/Estudiante.
Precondiciones:	-Poseer un usuario y contraseña.
Postcondiciones:	-Ingreso satisfactorio al sistema.
Escenario principal:	-El usuario ingresa usuario y contraseña. -El sistema procede a comparar los datos ingresados frente a la base de datos. -Los datos ingresados son correctos, se procede al ingreso al sistema.
Escenario alternativo:	-Los datos ingresados no son correctos. -Se muestra un mensaje de error al usuario.

Requisitos especiales:	No aplica.
Frecuencia:	Alta.

Tabla 1: Especificación del caso de uso Iniciar sesión.

Caso de uso:	Registrar usuario.
Actores involucrados:	Profesor/Estudiante.
Precondiciones:	No aplica.
Postcondiciones:	-Usuario registrado exitosamente.
Escenario principal:	-El sistema solicita que el usuario ingrese información asociada al registro. -El usuario llena la información y procede enviarla al sistema. -El sistema verifica en base de datos de poder realizar el registro. -El sistema notifica al usuario que el registro fue realizado con éxito.
Escenario alternativo:	-El usuario no llena la información de forma correcta o el nombre de usuario ya existe. -El sistema envía un mensaje de error.
Requisitos especiales:	No aplica.
Frecuencia:	Media.

Tabla 2: Especificación del caso de uso Registrar usuario.

Caso de uso:	Ver materia.
Actores involucrados:	Profesor/Estudiante.
Precondiciones:	-Ingresar al sistema.
Postcondiciones:	-Se despliega el contenido de la materia.
Escenario principal:	-El usuario selecciona desplegar las materias del

	<p>cargadas en el sistema.</p> <p>-El sistema despliega las opciones de materias del sistema.</p> <p>-El usuario selecciona ingresar a una materia.</p> <p>-El sistema le muestra el contenido de dicha materia.</p>
Escenario alternativo:	No aplica.
Requisitos especiales:	No aplica.
Frecuencia:	Alta.

Tabla 3: Especificación del caso de uso Ver materia.

Caso de uso:	Ver video.
Actores involucrados:	Profesor/Estudiante.
Precondiciones:	<p>-Ingresar al sistema.</p> <p>-Ingresar a la materia.</p>
Postcondiciones:	-Se despliega el video al usuario.
Escenario principal:	<p>-El usuario selecciona desplegar los videos cargados en el sistema.</p> <p>-El sistema despliega las opciones de videos de esa materia.</p> <p>-El usuario selecciona la calidad del video.</p> <p>-El usuario selecciona visualizar un video.</p> <p>-El sistema le despliega el video.</p>
Escenario alternativo:	No aplica.
Requisitos especiales:	No aplica.
Frecuencia:	Media.

Tabla 4: Especificación del caso de uso Ver video.

Caso de uso:	Ver documento.
Actores involucrados:	Profesor/Estudiante.
Precondiciones:	-Ingresar al sistema.

	-Ingresar a la materia.
Postcondiciones:	-Se despliega el documento al usuario.
Escenario principal:	-El usuario selecciona desplegar los documentos cargados en el sistema. -El sistema despliega las opciones de documentos de esa materia. -El usuario selecciona visualizar un documento. -El sistema le despliega el documento.
Escenario alternativo:	No aplica.
Requisitos especiales:	No aplica.
Frecuencia:	Media.

Tabla 5: Especificación del caso de uso Ver documento.

3.2.2 Especificación de los casos de uso de la aplicación de administración

A continuación se presenta la especificación de los principales casos de usos, de la aplicación de administración, bajo cuadros que establecen la información para cada uno de ellos:

Caso de uso:	Agregar materia.
Actores involucrados:	Administrador.
Precondiciones:	No aplica.
Postcondiciones:	-Materia agregada al sistema.
Escenario principal:	-El usuario selecciona la opción de agregar materia. -El sistema muestra la información que debe ser llenada por el usuario para agregar la materia. -El usuario llena la información y la envía al sistema. -El sistema procesa la información y registra la materia al sistema.
Escenario alternativo:	No aplica.

Requisitos especiales:	No aplica.
Frecuencia:	Baja.

Tabla 6: Especificación del caso de uso Agregar materia.

Caso de uso:	Eliminar materia.
Actores involucrados:	Administrador.
Precondiciones:	-Seleccionar una materia.
Postcondiciones:	-Se elimina la materia del sistema.
Escenario principal:	-El usuario selecciona una materia y presiona la opción de eliminar materia. -El sistema elimina la materia del sistema.
Escenario alternativo:	No aplica.
Requisitos especiales:	No aplica.
Frecuencia:	Baja.

Tabla 7: Especificación del caso de uso Eliminar materia.

Caso de uso:	Administrar materia.
Actores involucrados:	Administrador.
Precondiciones:	-Seleccionar una materia.
Postcondiciones:	No aplica.
Escenario principal:	-El usuario selecciona una materia y presiona la opción administrar materia. -El sistema procesa la información y despliega las opciones de administración de dicha materia.
Escenario alternativo:	No aplica.
Requisitos especiales:	No aplica.
Frecuencia:	Media.

Tabla 8: Especificación del caso de uso Administrar materia.

Caso de uso:	Cargar video.
Actores involucrados:	Administrador.

Precondiciones:	-Seleccionar administrar materia.
Postcondiciones:	-Video agregado al sistema.
Escenario principal:	-El usuario selecciona una ruta de video y presiona la opción subir contenido. -El sistema procede a cargar el video al sistema. -El video es cargado de forma exitosa.
Escenario alternativo:	-El video no cumple alguna de las restricciones. -Se muestra un error al usuario y el video no es cargado.
Requisitos especiales:	No aplica.
Frecuencia:	Media.

Tabla 9: Especificación del caso de uso Cargar video.

Caso de uso:	Cargar documento.
Actores involucrados:	Administrador.
Precondiciones:	-Seleccionar administrar materia.
Postcondiciones:	-Documento agregado al sistema.
Escenario principal:	-El usuario selecciona una ruta del documento y presiona la opción subir contenido. -El sistema procede a cargar el documento al sistema. -El documento es cargado de forma exitosa.
Escenario alternativo:	-El documento no cumple alguna de las restricciones. -Se muestra un error al usuario y el documento no es cargado.
Requisitos especiales:	No aplica.
Frecuencia:	Media.

Tabla 10: Especificación del caso de uso Cargar documento.

Caso de uso:	Eliminar contenido.
--------------	---------------------

Actores involucrados:	Administrador.
Precondiciones:	-Seleccionar administrar materia.
Postcondiciones:	-Contenido eliminado del sistema.
Escenario principal:	-El sistema despliega todos los contenidos de la materia. -El usuario selecciona el contenido que desea eliminar y presiona eliminar contenido. -El sistema elimina el contenido seleccionado.
Escenario alternativo:	No aplica.
Requisitos especiales:	No aplica.
Frecuencia:	Media.

Tabla 11: Especificación del caso de uso Eliminar contenido.

3.2.3 Modelado del sistema

Con el fin de tener un mejor entendimiento del comportamiento del software a desarrollar se debe crear modelos del sistema, que permitan entender mejor el flujo de datos y control, el comportamiento operativo y el contenido de la información. Bajo esta premisa, entonces se elabora un diagrama de componentes, tanto de la aplicación móvil (ver Figura 7) como también de la aplicación de administración (ver Figura 8):

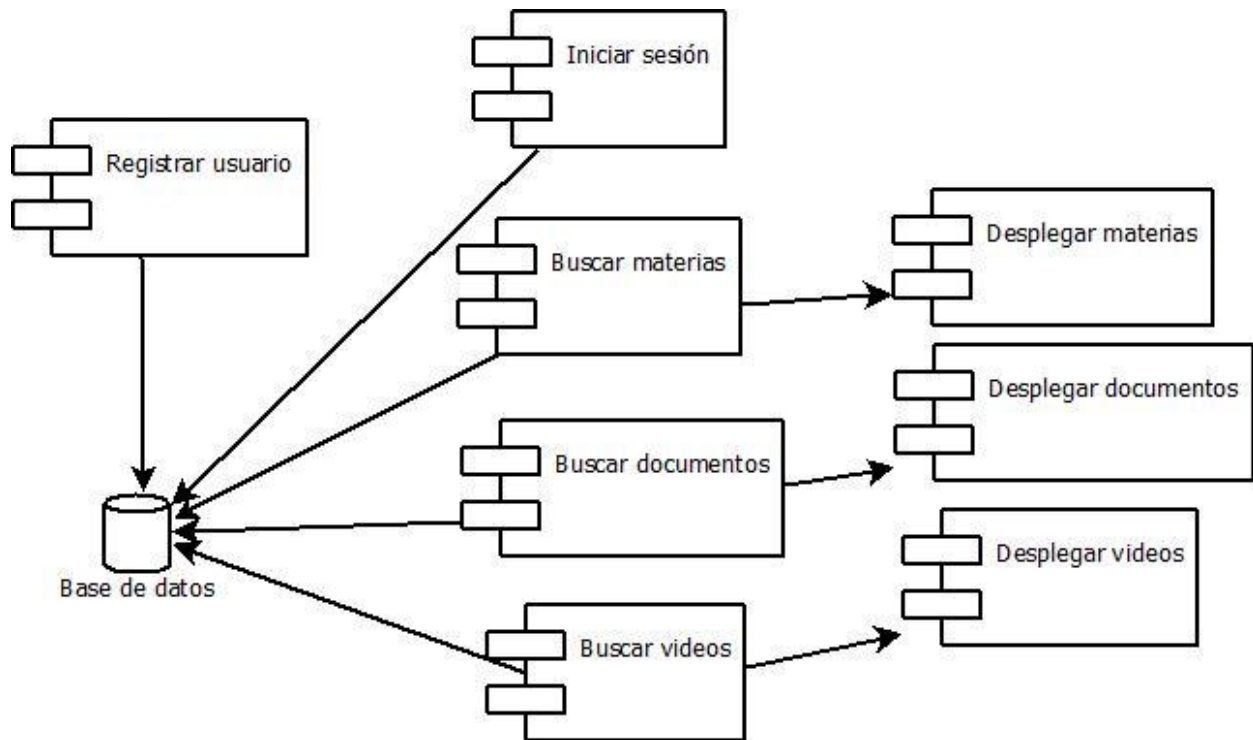


Figura 7: Diagrama de componentes de la aplicación móvil.

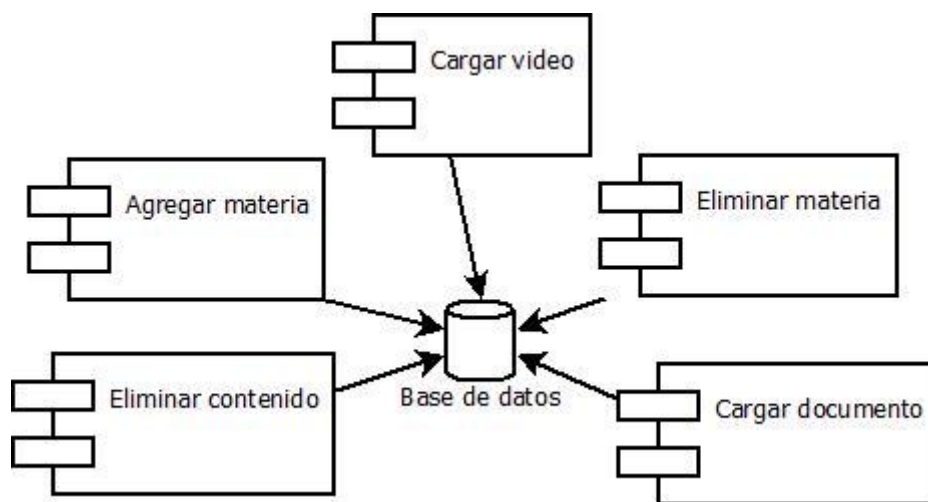


Figura 8: Diagrama de componentes de la aplicación de administración.

Del diagrama de componentes de la aplicación móvil se puede observar un conjunto de componentes que trabajan en conjunto para dar una respuesta y otros componentes que trabajan de forma independiente.

En el diagrama de componentes de la aplicación de administración no se observa alguna relación entre los componentes para dar una respuesta, solo una serie de componentes que trabajan de forma independiente.

Bajo el diagrama de componentes de ambas aplicaciones una similitud, la cual es que ambas aplicaciones tienen un gran impacto de actividad sobre la estructura de base de datos de las aplicaciones. Entonces cobra gran importancia el cómo debe ser el diseño de las bases de datos o la base de datos de las aplicaciones, para que con ello las operaciones que se hagan sobre ellas sean lo más eficientes y significativas para los sistemas.

Existen también otros tipos de diagramas que aportan otro tipo de información, uno de estos diagramas es el diagrama de entidad relación, que permiten comprender cómo se relacionan los datos existentes, entonces se elaboraron dos diagramas de entidad relación tanto como para la aplicación móvil y la aplicación de administración y se presentan a continuación:

En la siguiente Figura 9 se visualiza el diagrama de entidad relación de la aplicación de móvil:

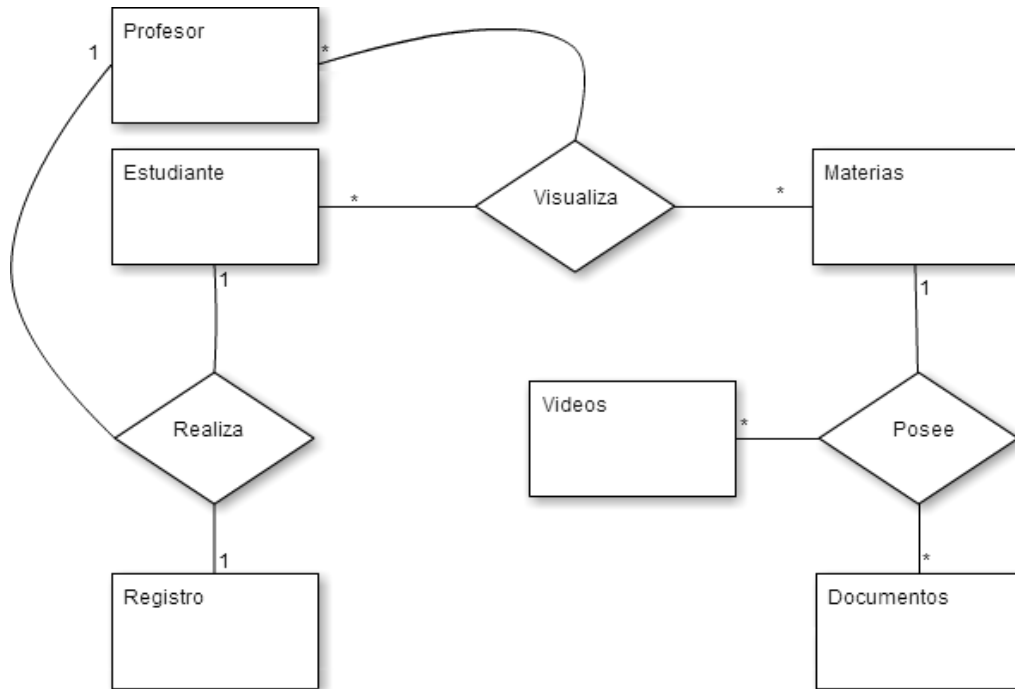


Figura 9: Diagrama de entidad relación de la aplicación de móvil.

En la siguiente Figura 10 se visualiza el diagrama de entidad relación de la aplicación de administración:

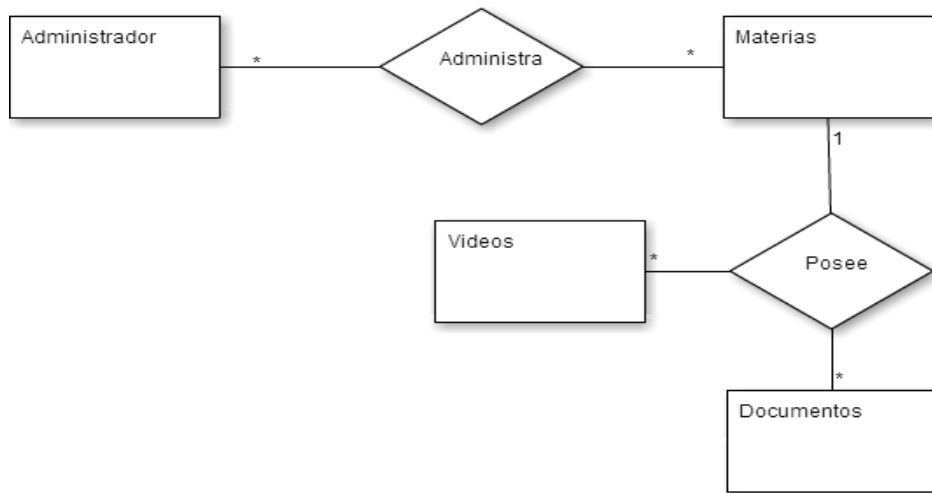


Figura 10: Diagrama de entidad relación de la aplicación de administración.

A partir de los diagramas de entidad relación se puede observar que existe una gran relación entre ambos diagramas, ya que una aplicación es complemento de la otra, pero entre esos elementos que comparten son las materias que poseen videos y documentos, mientras que en una aplicación son visualizados desde la otra aplicación son administrados.

Además se pueden observar los participantes en cada diagrama entidad relación, en el diagrama de entidad relación de la aplicación móvil, los participantes son Profesor y Estudiante, mientras tanto los participantes del diagrama de entidad relación de la aplicación de administración hay un único participante denominado Administrador.

3.3 Diseño

Según la metodología escogida el diseño es un proceso de muchos pasos, pero esta actividad se resume en el establecimiento de estructuras de datos como un diseño de datos, la arquitectura general de software como un diseño arquitectónico, representaciones de interfaz como un diseño de interfaz y finalmente los algoritmos como un diseño procedimental.

3.3.1 Diseño de datos

En un diseño de datos se indican las estructuras de datos que deben ser utilizadas por el sistema o los sistemas. Para llevar a cabo esta aplicación móvil se considera el siguiente modelo de datos:

Comprende de una estructura de base de datos, la cual posee las siguientes tablas:

- El contenido del sistema como "contenido".
- La información de los usuarios como "usuario".
- Que usuario se relaciona con una materia como "usuario_materia".
- La información de las materias como "materia".

Los siguiente recuadros describen la estructura de cada tabla:

Campo	Tipo de dato	Descripción
idcontenido	Integer	Clave primaria de tabla que identifica de forma única al registro.
tipo	Varchar<30>	El tipo de contenido que identifica al registro si es "video" o "documento".
ruta	Varchar<300>	La ruta o el nombre del archivo del registro.
idmateria	Integer	Clave foránea que identifica a cual materia pertenece el registro de contenido.

Tabla 12: Especificación de la tabla contenido.

Campo	Tipo de dato	Descripción
idusuario	Integer	Clave primaria de tabla que identifica de forma única al registro.
username	Varchar<35>	El nombre de usuario.
passwd	Varchar<350>	La contraseña de usuario.
privilegioBorrar	Bit	Si o no posee privilegios especiales.
correo	Varchar<200>	El correo del usuario.

Tabla 13: Especificación de la tabla usuario.

Campo	Tipo de dato	Descripción
idusuario	Integer	Clave foránea, pertenece al id del usuario.
idmateria	Integer	Clave foránea, pertenece al id de la materia.

Tabla 14: Especificación de la tabla usuario_materia.

Campo	Tipo de dato	Descripción
idmateria	Integer	Clave primaria de tabla que identifica de

		forma única al registro.
nombre	Varchar<50>	Nombre de la materia.
Codigo_CE	Varchar<20>	Código de control de estudios de la materia.
tipo_materia	Varchar<30>	El tipo de materia si es electiva, optativa o obligatoria.
semestre	Varchar<20>	El semestre al cual pertenece la materia.

Tabla 15: Especificación de la tabla materia.

Una de las primeras actividades que toma realmente importancia dado al análisis de fases anteriores, es sin duda la creación de la estructura de la base de datos que va dar soporte a la aplicación móvil y a su aplicación de administración.

Se crea entonces una base de datos denominada "estudio", en un sistema de bases de datos MariaDB, con la estructura mencionada anteriormente en el diseño de datos.

Esta base de datos, va ser la encargada de soportar todas las operaciones que se realizan mediante la plataforma de aprendizaje, entre estas operaciones se puede mencionar:

- Registro de usuarios.
- Corroborar datos al ingreso del sistema.
- Listado de materias en el sistema.
- Listado de documentos en el sistema.
- Listado de videos en el sistema.
- Registro de nuevas materias.
- Eliminar materias existentes.
- Registro de contenido de las materias.
- Eliminar contenido de las materias.

Cabe destacar que es la base de datos que será usada por ambos sistemas, tanto como por la aplicación móvil y la aplicación de administración.

No se dió el uso ningún programa que permitiera manipular la base de datos, si no que se manipula directamente por consola y sentencias SQL, ya que no se consideró necesario por no ser un proyecto tan complicado.

La escogencia de MariaDB sobre otros sistemas de base de datos, es porque MariaDB ha tenido gran protagonismo entre la comunidad de desarrolladores y principalmente por su licencia GPL que permite la libertad de usar, estudiar, compartir y modificar el software.

3.3.2 Diseño arquitectónico y procedimental

El diseño arquitectónico lo compone la estructura de los datos y los componentes del programa que se requieren para construir un sistema informático. Constituye el estilo arquitectónico que tendrá el sistema, la estructura, las propiedades de sus componentes que comprende al sistema y las interrelaciones que tienen lugar entre todos los componentes arquitectónicos del sistema [Barrios,2015].

Para el desarrollo de las aplicaciones, tanto como la aplicación móvil y la aplicación de administración tienen sus respectivas arquitecturas , pero con ciertas similitudes ya que una es complemento de la otra.

El diseño de arquitectura de la aplicación móvil que se observa en la Figura 9 es el siguiente:



Figura 11: Arquitectura de la aplicación móvil.

El diseño procedimental de esta arquitectura es el siguiente:

- El diseño de arquitectura propuesto se compone en una plataforma Android, tecnologías Java y PHP.
- Se desarrolla una aplicación móvil que contiene la interfaz gráfica. Allí se podrán registrar tanto profesores como alumnos, se podrá acceder a los diferentes cursos, acceder a los contenidos, reproducción de video y audio *on demand*.
- Se implementa un servidor de *streaming* con red5 que permita la gestión de contenido multimedia (video y audio), utilizando el protocolo RTMP.
- Se comunica la aplicación Android con el servidor de *streaming* a través de la librería de Java Vitamio.
- Se implementa el servidor web nginx el cual va a contener el *backend* de la aplicación para la gestión de manejo de contenidos e información de usuario .Del lado servidor se desarrollará en PHP.
- Para el almacenamiento de los datos de usuario y datos del sistema, se implementará un servidor de base de datos relacional MariaDB.

El diseño de arquitectura de la aplicación de administración que se observa en la Figura 10 es el siguiente:



Figura 12: Arquitectura de la aplicación administración.

El diseño procedimental de esta arquitectura es el siguiente:

- El diseño de arquitectura propuesto se compone en los lenguajes de programación como HTML y PHP.
- Se desarrolla una aplicación web que contiene la interfaz gráfica. Allí se permite la administración del contenido de la aplicación móvil, como crear o eliminar cursos/materias, agregar o eliminar contenidos como documentos, videos y audios.
- Se implementa un servidor de *streaming* con red5 que permite la gestión de contenido multimedia (video y audio), donde la aplicación de administración permite cargar tanto videos como audios al servidor de *streaming*.
- Se implementa el servidor web nginx el cual va a contener el *backend* de la aplicación para la gestión de manejo de contenidos, donde la aplicación de administración permite cargar documentos a este servidor.
- Para el almacenamiento de los datos de usuario y datos del sistema, se implementará un servidor de base de datos relacional MariaDB, donde la aplicación de administración refleja todos los cambios asociados al contenido de la aplicación móvil.

3.3.3 Diseño de la interfaz

Básicamente el diseño de interfaz debe especificar cómo se comunica el software consigo mismo, con los sistemas que operan junto con él y con los operadores y usuarios que lo emplean [Barrios,2015].

Entonces el diseño de interfaz, se puede dividir en tres partes, atacando la primera de ellas es como el software se comunica consigo mismo, esto en la aplicación móvil sucede bajo una de las funcionalidades de Java para Android, las cuales son los *"Intent"* los cuales permiten comunicar diferentes piezas de software con otras. Los *"Intent"* no solo permiten el paso de mensajes entre diferentes piezas de software, si no también permiten cambiar de una vista a otra, también permiten cancelar la acción que se realizó.

El software de administración de la aplicación, se comunica consigo mismo usando variables de sesión de PHP, que permiten compartir un gran conjunto de variables a través de una sesión de PHP.

En segundo lugar es como el software se comunica con los sistemas que operan junto con él, esto tanto en la aplicación móvil y en la aplicación de administración, ocurre mediante servicios web en PHP, estos servicios van a permitir a diferentes componentes del software comunicarse bajo un esquema de petición y respuesta. Esto se refiere a que un componente realiza una petición bajo algún o algunos parámetros a un servicio web, el servicio web se ejecuta comunicándose con otro sistema y prepara una respuesta al componente que inicio la petición.

En tercer lugar es como el software se comunica con los usuarios y los operadores que lo emplean, esto hace referencia al diseño de interfaz de usuario, que ocurre como un método iterativo de prototipos, que son presentados al usuario final y este ejerce una retroalimentación hacia el prototipo que le fue presentado, haciendo así que ocurran modificaciones o mejoras.

Se han generado diferentes prototipos de la aplicación móvil y la aplicación de administración, pero estos no son cambios significativos, es por ello que no se ha guardado imágenes de estos prototipos previos. Entre estos cambios no significativos, están por ejemplo cambio de tipo de letra, color de letra, cambio de color de fondo y cambio de localización de elementos de la interfaz. Siempre se ha preservado la estructura inicial de los elementos de la interfaz.

Tanto la aplicación móvil y la aplicación de administración presentaron diferentes prototipos, pero son las interfaces finales que son presentados a continuación:

En la Figura 13, se muestra la interfaz de inicio de la aplicación móvil, la cual posee una etiqueta de bienvenida, dos textos editables para ingresar usuario y contraseña, dos botones, el primero permite el ingreso a la siguiente interfaz de la aplicación y el segundo acceder al registro y finalmente una imagen distintiva de la Universidad Central de Venezuela como es una caricatura del reloj de plaza del rectorado.

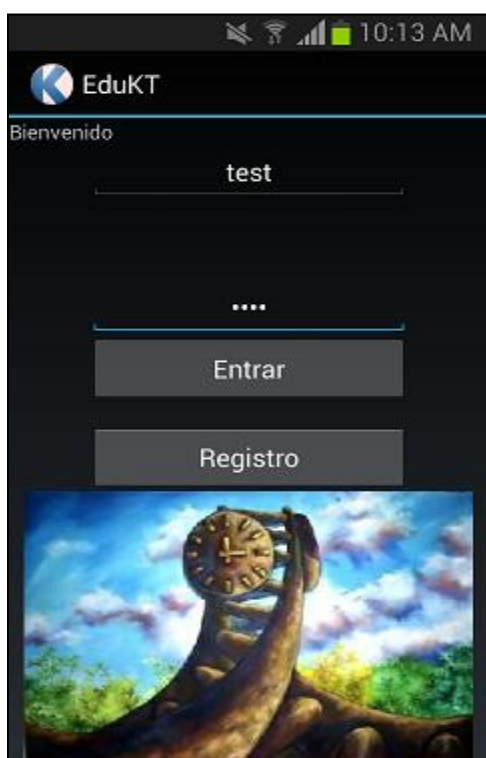


Figura 13: Interfaz de inicio.

En la Figura 14, se muestra la interfaz de la aplicación móvil de las materias que posee la aplicación, a través de seis listas desplegables, el usuario puede desplegar cada lista e ingresar a la materia que seleccione que lo llevara a otra interfaz de la aplicación.



Figura 14: Interfaz de las materias.

En la Figura 15, se muestra la interfaz de la aplicación móvil de los contenidos de una materia, esta interfaz posee la funcionalidad de seleccionar documento o seleccionar video a través de listas desplegables, así como también la calidad de los videos.

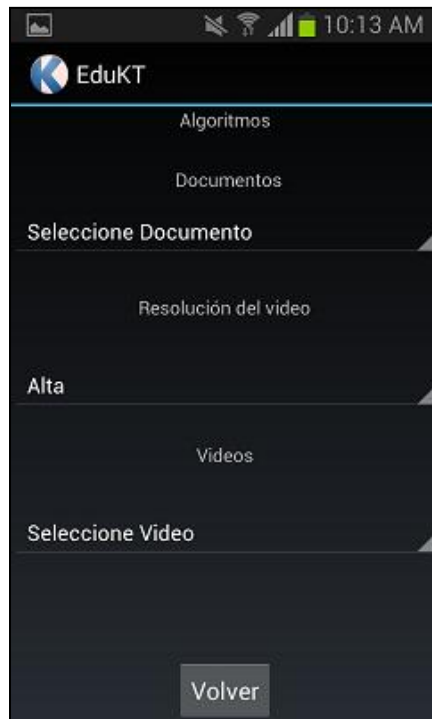


Figura 15: Interfaz de los contenidos.

En la Figura 16, se muestra la interfaz que despliega los documentos, una vez seleccionado el documento en la interfaz de la Figura 15, entonces este es descargado al dispositivo móvil y la aplicación permite visualizar ese contenido mostrándole las opciones para abrir este tipo de archivos que posee el dispositivo móvil.

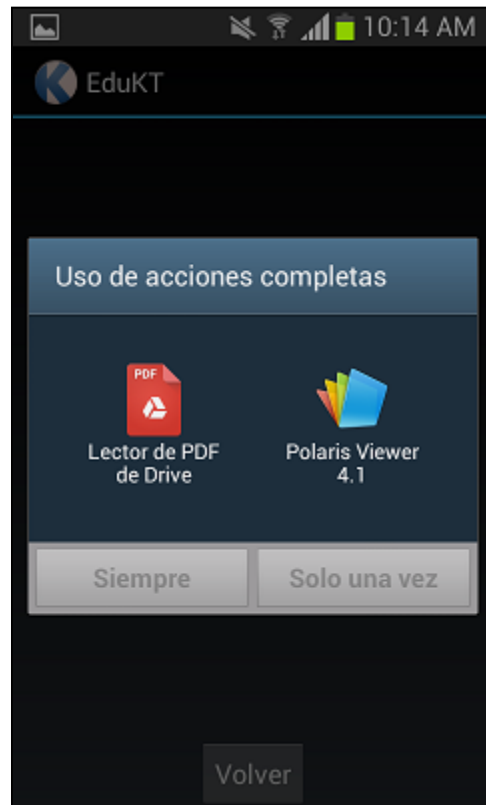


Figura 16: Interfaz que despliega documentos.

En la Figura 17 se muestra la interfaz que despliega los videos, una vez seleccionado el video en la interfaz de la Figura 15, entonces este se obtiene por *streaming* al dispositivo móvil y la interfaz posee un reproductor que capta este *streaming* y lo reproduce

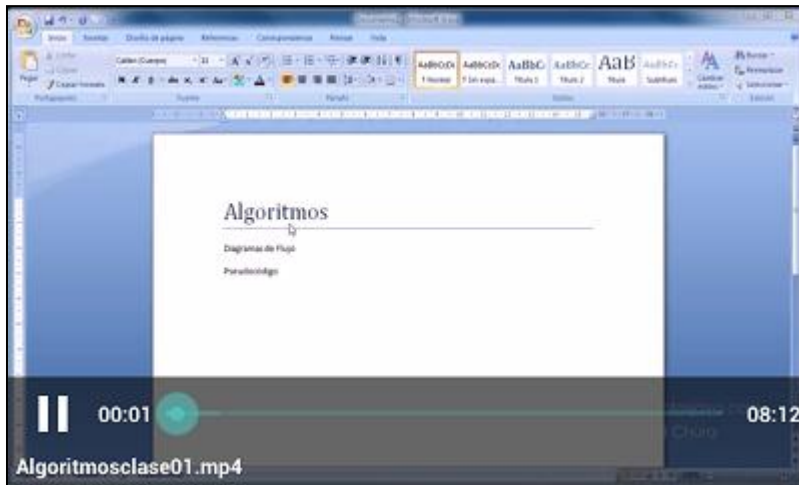


Figura 17: Interfaz que despliega videos.

En la Figura 18, se muestra la interfaz de inicio de la aplicación móvil que posee un *combo box* que permite seleccionar cada una de las materias que están cargadas a la aplicación de administración, posee también dos botones, el primero permite administrar una materia seleccionada y el segundo permite agregar una nueva materia.



Figura 18: Interfaz de inicio aplicación de administración.

En la Figura 19, se muestra una interfaz de la aplicación de administración que permite agregar una materia a la aplicación, posee tres cuadros para introducir el texto de información correspondiente de la materia, así como también una serie de botones de radio para seleccionar el semestre de la materia, finalmente posee dos botones, uno que permite agregar la materia y el otro que permite volver a la interfaz de la Figura 18.

Agregar materia

Nombre materia:

Código materia:

Tipo de materia:

Semestre de materia:

Primero Segundo Tercero Cuarto Quinto Electiva

Agregar

Volver

Figura 19: Interfaz agregar materia aplicación de administración.

En la Figura 20, se muestra una interfaz de la aplicación de administración que permite administrar una materia de la aplicación, posee dos radio botones que permiten seleccionar que tipo de contenido se selecciona, un *combo box* que muestra las materias cargadas en la aplicación, también esta interfaz posee cinco botones, el primero "Seleccionar archivo" permite seleccionar un archivo para ser subido a la aplicación, el segundo "Subir" permite subir el archivo que ha sido previamente seleccionado, el tercero "Eliminar contenido" permite eliminar un contenido que haya sido seleccionado dentro del *combo box*, el cuarto "Eliminar materia" elimina la materia la cual se está administrando y el quinto "Volver" permite volver a la interfaz de la Figura 18.



Figura 20: Interfaz administrar materia aplicación de administración.

3.4 Generación de código

Esta es sin duda la fase más importante ya que gracias esta se llevaran a cabo las tareas para la que está diseñado el software. La generación de código prácticamente se encarga de trasladar el diseño planteado hacia una forma legible por la máquina.

Para la construcción de los siguientes módulos de la aplicación móvil, se utiliza tres tecnologías, la primera de ellas es el servidor de contenidos nginx que permitirá ejecutar un servicio web, la segunda es el servicio web en si en PHP, y la tercera es *Android Studio* que permite la construcción de la interfaz y su controlador.

3.4.1 Módulo de ingreso de la aplicación móvil

Se arranca con *Android Studio*, que permite diseñar la interfaz que va constar de los siguientes elementos:

- Un "*Label*" que es una etiqueta que da la bienvenida al sistema.
- Dos "*Edit text*" que permiten capturar texto del usuario tales como usuario y contraseña.
- Una imagen que va servir de adorno a la interfaz.
- Dos botones, el primero que permite el ingreso al sistema, y el segundo el acceso al registro.

Desde el controlador en Java, se va a capturar un evento asociado al botón de ingreso, el cual verificara que el usuario haya escrito su usuario y contraseña en los "*Edit text*". Este controlador va a ejecutar un servicio web en PHP que permitirá corroborar si efectivamente el usuario y contraseña existen en el sistema de base de datos, el servicio web simplemente da una respuesta de verdadero o falso. Si es verdadero el ingreso es satisfactorio y el usuario pasa a otra interfaz, si es falso el ingreso es incorrecto y se le muestra un mensaje al usuario.

Cabe destacar, que el controlador envía la contraseña bajo un hash de sha256 al servicio web en PHP, ya que las contraseñas se almacenan en base de datos bajo dicho hash.

También desde el controlador en Java, se va manejar un evento asociado al botón de registro, de ser pulsado por el usuario, se mostrara al usuario a la interfaz del registro.

3.4.2 Módulo de registro de la aplicación móvil

El objetivo de este módulo, es permitir el registro de nuevos usuarios a la aplicación móvil.

Con *Android Studio*, se diseña la interfaz del registro que consta de los siguientes elementos:

- Un *"Label"* que es una etiqueta, la cual indica lo que el usuario debe hacer.
- Cuatro *"Edit text"* que permiten capturar texto del usuario tales como usuario, contraseña, clave especial y correo.
- Dos botones, uno permite ejecutar el registro y el otro permite volver al ingreso.

Desde el controlador en Java, se captura el evento cuando es pulsado el botón de "Registrarse", este controlador va verificar que el usuario haya llenado de forma correcta los *"Edit text"*, de ser así va ejecutar un servicio web en PHP, que va permitir crear un nuevo registro en la base de datos con los datos introducidos por el usuario, al terminar el registro el controlador enviara un mensaje por la interfaz al usuario, que efectivamente el usuario ha sido creado.

También desde el controlador en Java, se captura el evento de la pulsación del botón "Volver al inicio", esto permitirá volver al ingreso de la aplicación.

Cabe destacar, que la contraseña es enviada al servicio web en PHP, bajo un hash de sha256, para garantizar la privacidad de la contraseña ante el sistema de bases de datos.

3.4.3 Módulo de materias de la aplicación móvil

El objetivo de este módulo, es desplegar al usuario las distintas materias que posee la aplicación móvil.

Con *Android Studio*, se diseña la interfaz del módulo de materias de la aplicación móvil, esta interfaz consta de los siguientes elementos:

- Un "*Label*" etiqueta que indica en que sección se encuentra el usuario.
- Seis "*Spinners*" o listas desplegables, que posee las materias cargadas al sistema por semestre.
- Un botón "Salir", que permite volver al ingreso del sistema.

Inicialmente desde el controlador en Java y apenas se carga la interfaz al usuario ocurre un evento automático, el cual ejecuta un servicio web en PHP, que va a devolver un listado de todas las materias que están cargadas en el sistema, esta lista es pasada a cada lista desplegable por semestre para ser mostrada al usuario.

Desde el mismo controlador Java, permite capturar el evento de cuando el usuario desea desplegar alguna lista y seleccionar una opción de materia de la lista, cuando la selecciona el sistema le muestra al usuario la interfaz correspondiente a la materia que selecciono.

También desde el controlador Java, captura un evento asociado a la pulsación del botón "Salir", que le permitirá salir del sistema mostrando la interfaz correspondiente al ingreso del sistema.

3.4.4 Módulo de documentos de la aplicación móvil

El objetivo de este módulo es listar y desplegar documentos en formato PDF al usuario.

Con *Android Studio*, se diseña la interfaz de usuario que posee los siguientes elementos:

- Dos "*Label*" o etiquetas, la primera indica que materia se encuentra el usuario la segunda aparece "Documentos".
- Un "*Spinner*" o lista desplegable, que posee los diferentes documentos cargados en el sistema.
- Un botón "Volver", que permite volver la anterior interfaz del sistema.

Inicialmente desde el controlador Java, ocurre un evento automatizado el cual se encarga de ejecutar un servicio web en PHP que va devolver una lista de todos los documentos que se encuentran cargados en el sistema de la materia seleccionada. Esta lista se pasa a la lista desplegable para que pueda ser mostrada al usuario.

Desde el mismo controlador Java, permite capturar el evento de cuando el usuario desea desplegar la lista y seleccionar una opción de documento de la lista, cuando la selecciona el sistema despliega una interfaz mientras se descarga al dispositivo el documento seleccionado, una vez descargado permite abrirlo con alguna aplicación que permita visualizar documentos en formato PDF que posea el dispositivo.

También desde el controlador Java, captura un evento asociado a la pulsación del botón "Volver", que le permitirá volver la interfaz anterior perteneciente a donde se seleccionan las materias.

3.4.5 Módulo de videos de la aplicación móvil

El objetivo de este módulo es listar y desplegar videos al usuario. Este módulo hace uso de un servidor de *stream* red5 para los videos usando el protocolo RTMP.

Con *Android Studio*, se diseña la interfaz de usuario, la cual posee los siguientes elementos:

- Tres "*Label*" o etiquetas, la primera indica que materia se encuentra el usuario la segunda aparece "Videos" y la tercer muestra "Resolución del video".
- Dos "*Spinners*" o listas desplegables, la primera que muestra las distintas resoluciones en las cuales se puede mostrar el video y la segunda que posee los diferentes videos cargados de la materia.

Inicialmente desde el controlador Java, ocurre un evento automatizado el cual se encarga de ejecutar un servicio web en PHP que va devolver una lista de todos los videos que se encuentran cargados en el sistema de la materia seleccionada. Esta lista se pasa a la lista desplegable para que pueda ser mostrada al usuario.

Desde el mismo controlador Java, permite capturar el evento de cuando el usuario desea desplegar la lista y seleccionar una opción de video de la lista, cuando la selecciona el sistema despliega una nueva interfaz, esta interfaz recibe como parámetro también la calidad de resolución que el usuario selecciono, la nueva interfaz posee un reproductor de videos, el controlador entonces haciendo uso de la librería Vitamio, busca el recurso RTMP seleccionado y se lo pasa al reproductor para su reproducción. El usuario puede en cualquier momento adelantar, pausar, detener o retroceder el video a través del reproductor.

Los siguientes módulos hicieron uso de un servidor nginx y los lenguajes de programación de HTML y PHP.

3.4.6 Módulo de inicio de la aplicación de administración

El objetivo de este módulo es mostrarle al usuario todas las opciones que puede realizar desde la aplicación de administración.

Con HTML se diseña la interfaz de usuario, la cual posee los siguientes elementos:

- Un título, el cual muestra el título de la interfaz.
- Un formulario, el cual presenta un cuadro de opciones.
- Dos botones, el primero permite ir a la sección de administración de materias y el segundo permite ir a la sección de agregar materias.

Desde el controlador en HTML, se captura la pulsación del botón de administrar materias, el cual toma la información de la materia seleccionada en el cuadro de opciones redirecciona al usuario a una interfaz de usuario perteneciente a otra sección del sistema.

También desde el controlador en HTML, se captura la pulsación del botón de agregar materia, el cual redirecciona al usuario a una interfaz de usuario perteneciente a otra sección del sistema.

3.4.7 Módulo de agregar materia de la aplicación de administración

El objetivo de este módulo, es permitir al usuario agregar una nueva materia al sistema.

Con HTML se diseña la interfaz de usuario, la cual posee los siguientes elementos:

- Un título, el cual muestra el título de la interfaz.
- Un formulario con tres entradas de texto, la primera para el nombre de la materia, la segunda para el código de la materia y la tercera para el tipo de materia.
- Dos botones "Agregar", el cual permite al pulsarlo agregar la materia bajo los datos escritos anteriormente y "Volver", el cual permite direccionar a la anterior interfaz o la interfaz de inicio del sistema.

Desde el controlador HTML, se captura la pulsación del botón "Agregar", se toman los datos introducidos por el usuario en el formulario y son enviados a otra

página la cual realiza el proceso de registro de la materia en la base de datos y redirecciona al usuario nuevamente a la interfaz de agregar materia.

También desde el controlador HTML, se captura la pulsación del botón "Volver", el cual redirecciona al usuario a la interfaz de inicio del sistema.

3.4.8 Módulo de administrar materia de la aplicación de administración

El objetivo de este módulo, es permitir al usuario administrar una materia, administrar en tal sentido de cargar al sistema videos y documentos de la materia o si no también eliminar videos y documentos previamente cargados.

Con HTML se diseña la interfaz de usuario la cual posee los siguientes elementos:

- Un formulario, que en forma de cuadro de opciones presenta el contenido que ha sido previamente cargado a la materia.
- Dos botones de radio, que permiten seleccionar que tipo de contenido será cargado (documento o video).
- Cinco botones, el primero "Seleccionar archivo", el segundo "Subir", el tercero "Eliminar contenido", el cuarto "Volver" y el quinto "Eliminar materia".

Desde el controlador HTML, se captura la pulsación del botón "Seleccionar archivo", el cual despliega una nueva interfaz que le permite al usuario seleccionar un archivo de su sistema.

Desde el controlador PHP, se captura la pulsación del botón "Subir", entonces se procede a verificar los botones de radio para ver cómo y dónde se debe cargar el archivo seleccionado. Independientemente de la selección del usuario se verifica que el archivo sea de un tamaño apropiado, luego si el usuario selecciono la opción de video entonces el archivo debe ser cargado al servidor de videos red5 bajo la carpeta de dicha materia. Por otra parte si el usuario selecciono la opción de documento, entonces

el archivo debe ser cargado al servidor de contenidos nginx. Finalmente se procede a crear el registro en base de datos del contenido cargado al sistema.

Desde el controlador PHP, se captura la pulsación de "Eliminar contenido", se busca en el cuadro la opción seleccionada por el usuario a eliminar, la opción puede ser un video o un documento, si es un video se procura a eliminar el video seleccionado en la carpeta de la materia dentro del servidor de videos red5, si es un documento entonces se elimina el documento dentro del servidor de contenidos nginx. Finalmente se procede a eliminar el registro en base de datos el contenido seleccionado.

Desde el controlador en PHP, se captura la pulsación del botón "Eliminar materia", entonces se procede a realizar una consulta sobre la base de datos, lo cual devuelve todo el contenido entre documentos y videos que posee la materia, con esta información se procede a eliminar uno a uno tanto documentos y videos de dicha materia (tanto del servidor de contenidos nginx y del servidor de *streamred5*), luego se procede a eliminar los registros de bases de datos correspondientes. Finalmente se elimina la materia de los registros de base de datos y se redirecciona a la interfaz de principal.

Desde el controlador en HTML, se captura la pulsación del botón "Volver", el cual redirecciona al usuario a la interfaz de inicio del sistema.

4. Pruebas

Se realizan una serie de pruebas de usabilidad de los sistemas en forma de encuestas a diferentes usuarios con diferentes características. Además se realizó una prueba de rapidez a la carga de los videos de la aplicación móvil.

4.1 Pruebas de la aplicación móvil EduKT

Se realiza una encuesta a 5 personas de edad comprendida entre los dieciocho y veintiocho años que actualmente se encuentran cursando estudios universitarios de

pregrado o recientemente graduados, estas encuestas poseen diferentes preguntas y sus resultados se exponen a continuación:

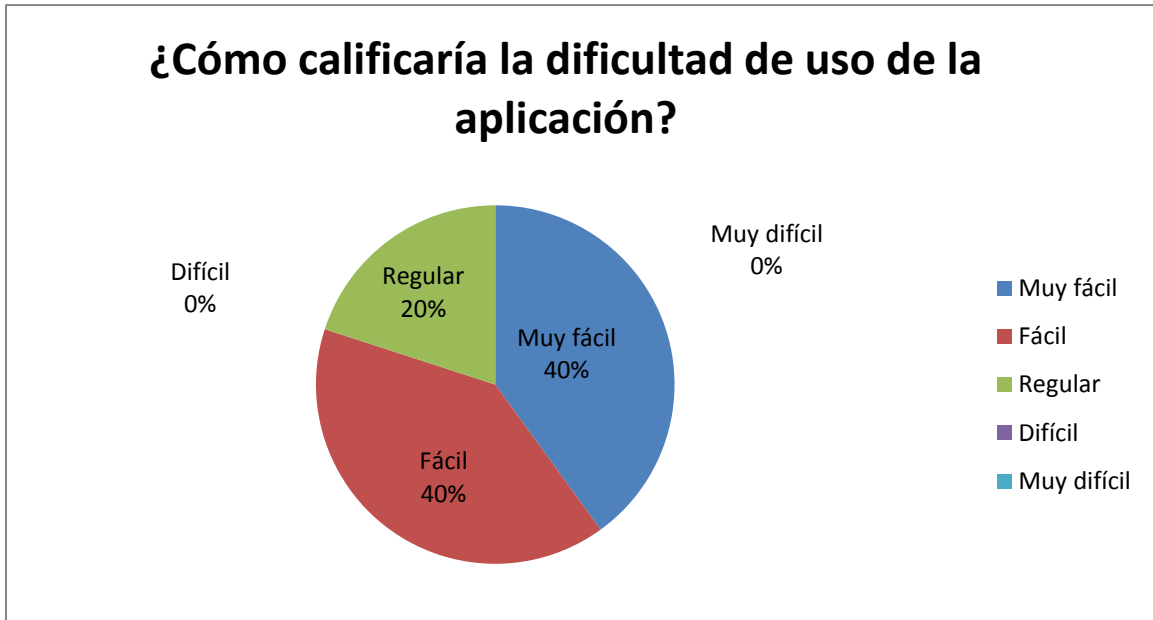


Gráfico 1: Dificultad de uso aplicación móvil.

A la totalidad de los encuestados les pareció de muy fácil a regular el uso de esta aplicación. A ninguno le pareció muy difícil ni siquiera difícil.

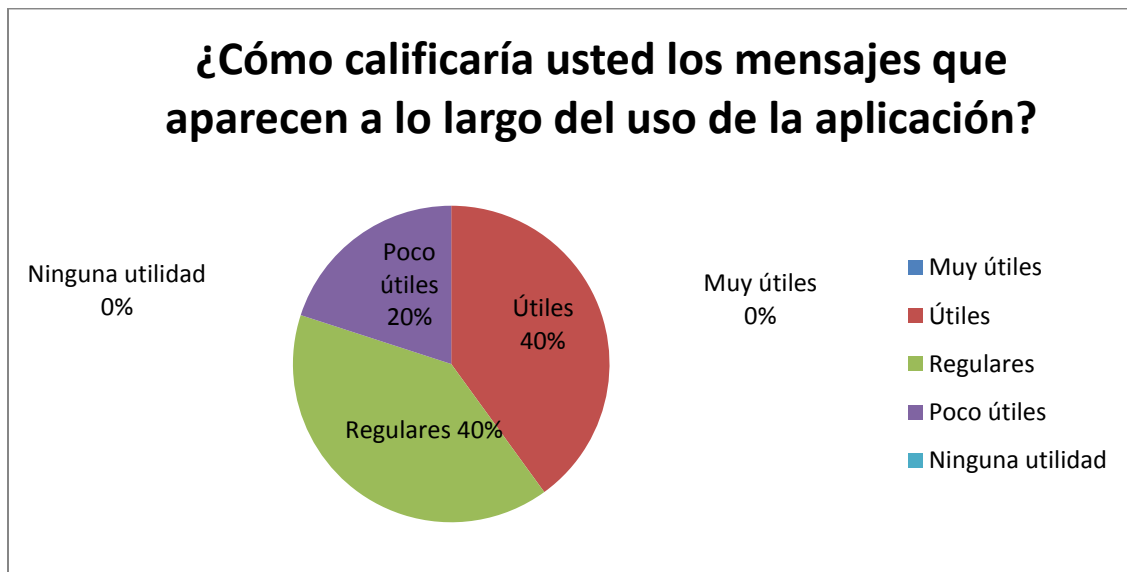


Gráfico 2: Utilidad de los mensajes de la aplicación móvil.

La gran mayoría de los encuestados respondió que los mensajes orientadores a lo largo de la aplicación fueron de regular a útiles.

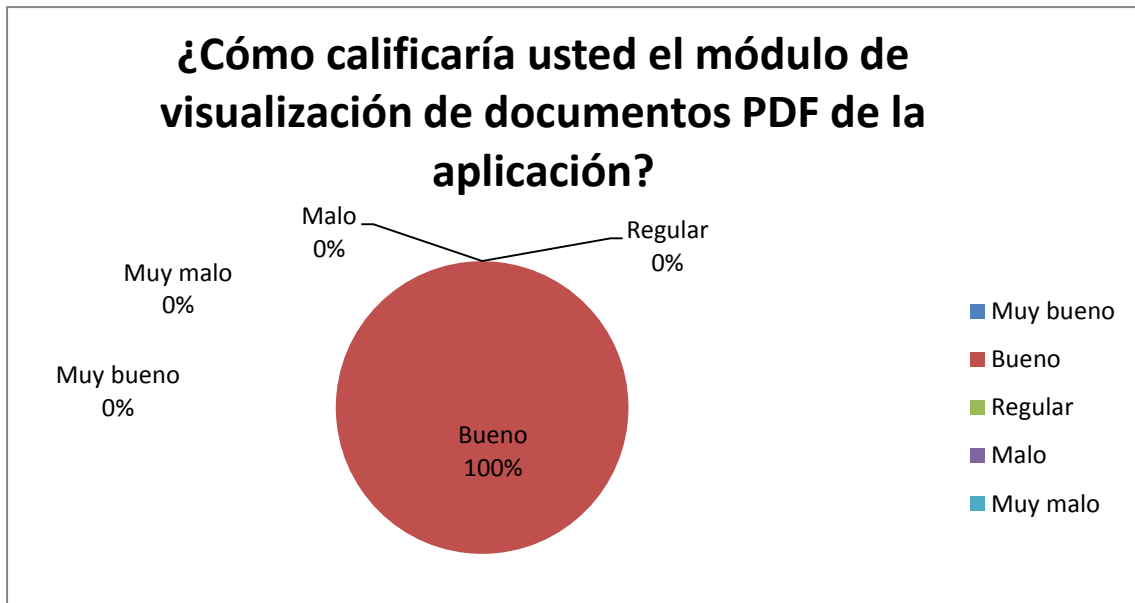


Gráfico 3: Módulo de PDF de la aplicación móvil.

Definitivamente el módulo de visualización de documentos PDF es **bueno**.

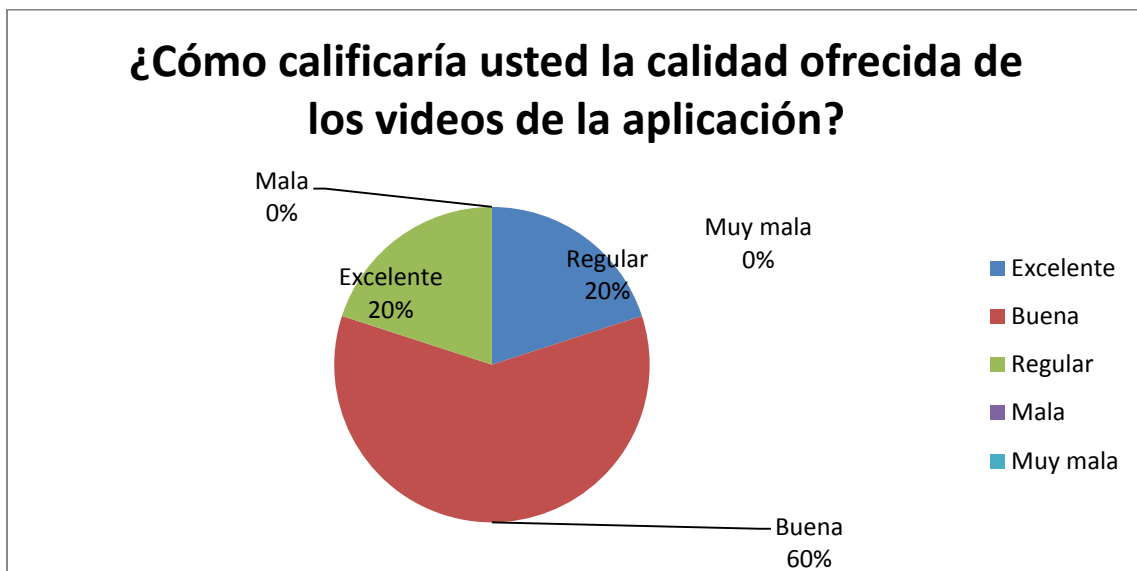


Gráfico 4: Calidad de los videos de la aplicación móvil.

A la gran mayoría de los encuestados le pareció que la calidad de los videos es de buena a excelente.

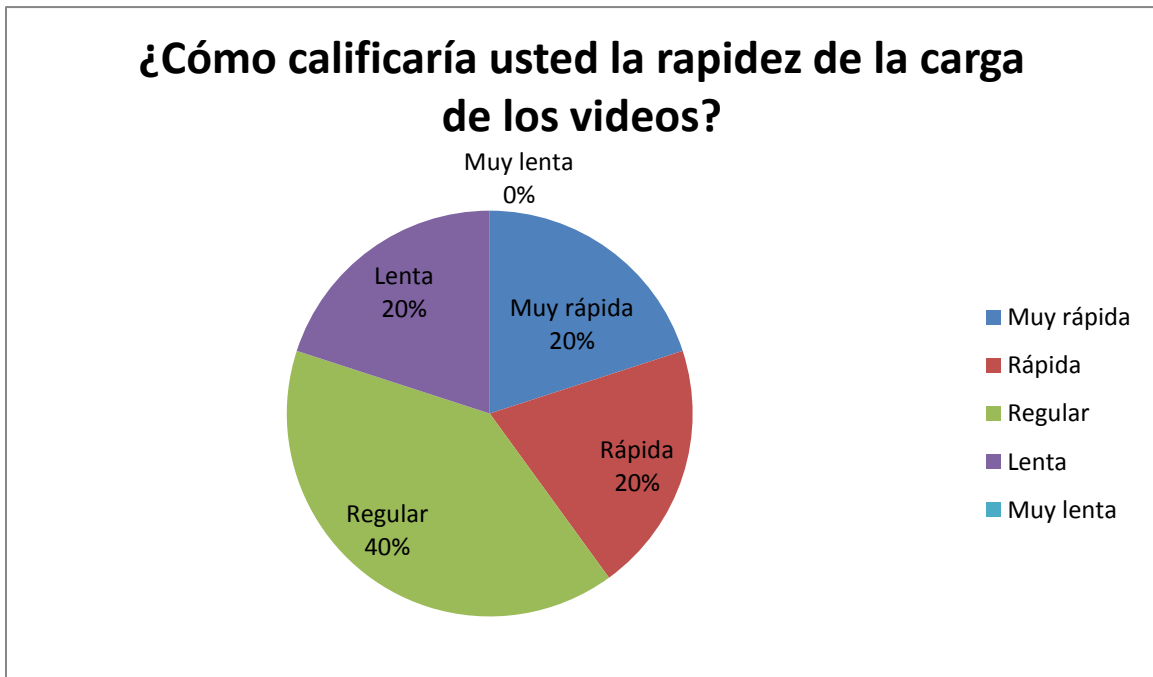


Gráfico 5: Rapidez de carga de videos de la aplicación móvil.

La gran mayoría de los encuestados estuvo de acuerdo a que la rapidez de carga de los videos estaba entre regular y muy rápida. Lo cual es bastante aceptable.



Gráfico 6: Recomendar la aplicación móvil.

Tomando en cuenta la opinión de la totalidad de los encuestados acerca de esta aplicación sería recomendable.

Se recogen igual algunas de las opiniones de los encuestados en cuanto al porque recomendarían esta aplicación, que son las siguientes:

- "Sí, porque es una forma de incentivar a los estudiantes y acercarlos mucho más al aprendizaje."
- "Sí, dado que se centraliza la información y no hay que buscar entre diferentes sistemas existentes que actualmente se utilizan (Dropbox, Google Drive, email, Youtube, etc) para encontrar el material."
- "La recomendaría, aunque un diseño más atractivo sería agradable."
- "Sí, ya que ofrece la facilidad de poder acceder a la información académica de las materias mediante un medio diferente no antes desarrollada."
- "Sí, se ve útil. Aunque hay que hacerle algo de tratamiento a la interfaz hasta que se vea mejor y que sea más intuitivo."

4.2 Pruebas de la aplicación de administración de contenidos

Se realiza una encuesta a 5 personas de edad comprendida entre los veinticuatro y sesenta años que actualmente poseen experiencia de al menos un año como profesor o preparador, estas encuestas poseen diferentes preguntas y sus resultados se exponen a continuación:

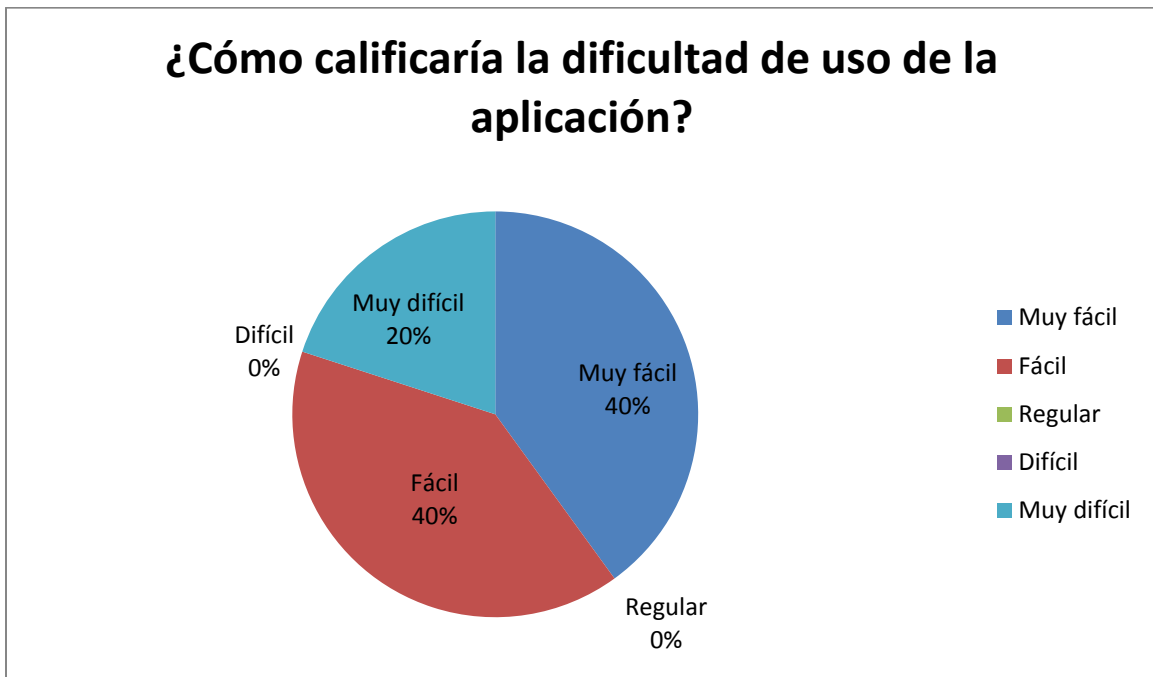


Gráfico 7: Dificultad de uso de la aplicación de administración.

A la gran mayoría de los encuestados le pareció que esta aplicación es de fácil a muy fácil uso.

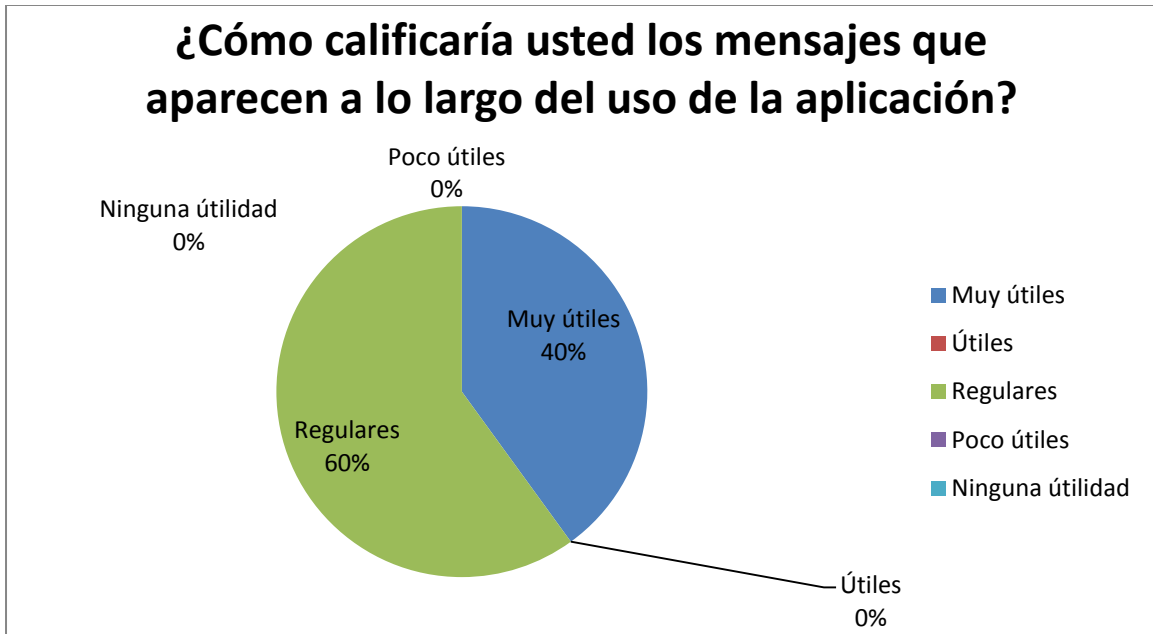


Gráfico 8: Utilidad de los mensajes de la aplicación de administración.

La totalidad de los encuestados respondió que los mensajes orientadores a lo largo de la aplicación fueron de regular a muy útiles.

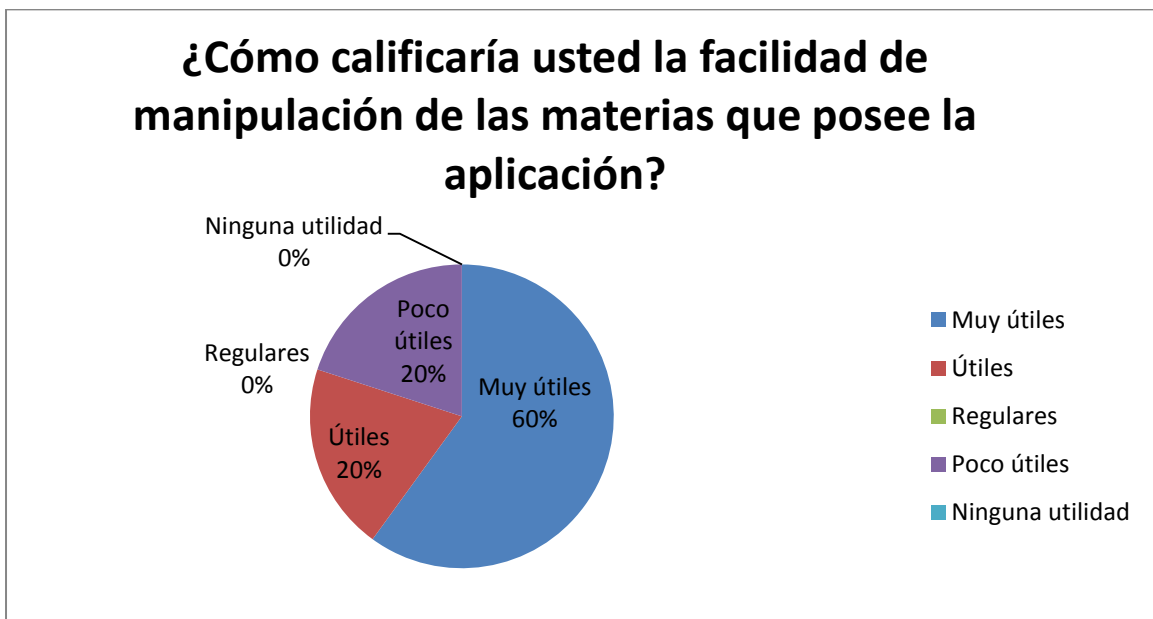


Gráfico 9: Manipulación de contenidos de la aplicación de administración.

La gran mayoría de los encuestados se pronunció a través de la facilidad de la manipulación de las materias incluidas en esta aplicación.

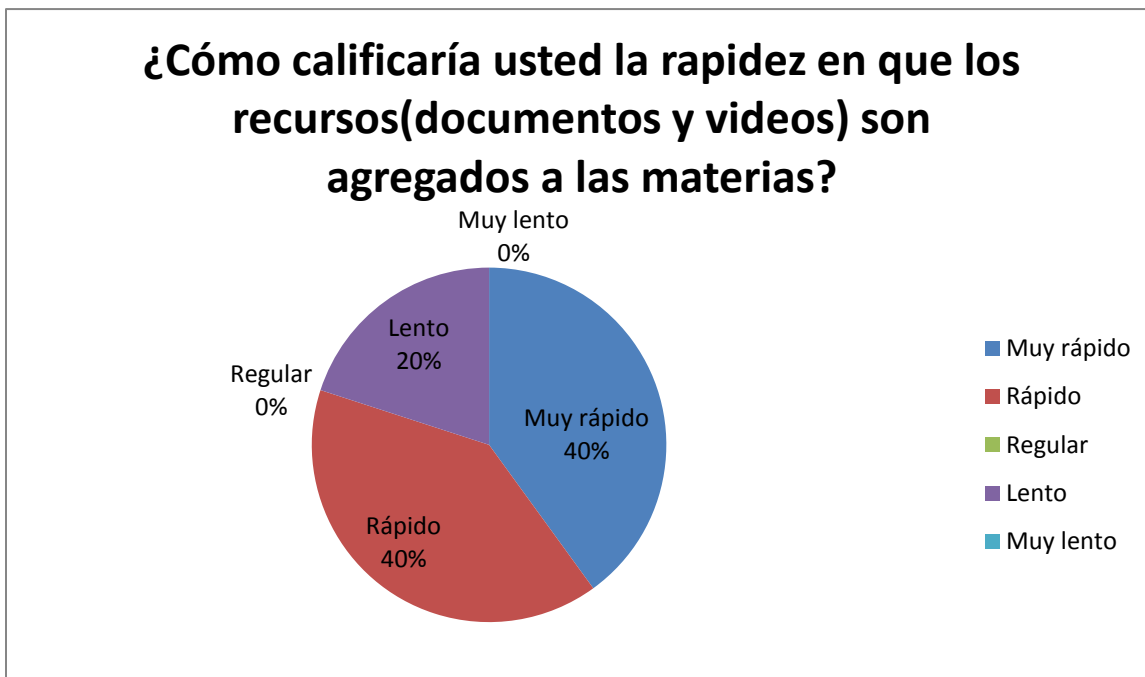


Gráfico 10: Rapidez de recursos agregados de la aplicación de administración.

La gran mayoría de los encuestados se inclinó hacia la rapidez de como los recursos son agregados a las materias.

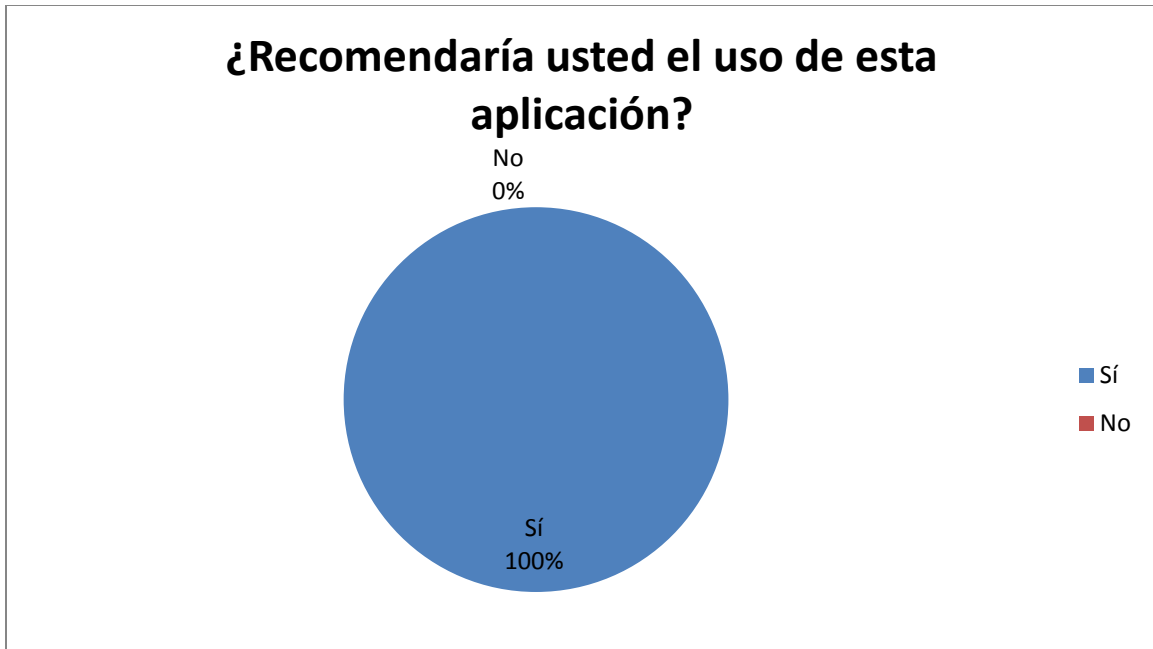


Gráfico 11: Recomendar la aplicación de administración.

La totalidad de los encuestados se pronunció a favor de recomendar esta aplicación.

Se recogen igual algunas de las opiniones de los encuestados en cuanto al porque recomendarían esta aplicación, que son las siguientes:

- "Sí, Lo recomendaría porque facilita la comunicación profesor alumno, el dar accesibilidad por a los estudiantes del material de estudio (videos, audios, presentaciones, textos) pertinente a cada asignatura. Contribuye al proceso de enseñanza aprendizaje. Usa la telefonía móvil con fines docentes. Una herramienta más."
- "Sí, sin embargo mejoraría ciertos elementos de la interfaz: Elementos de la interfaz podrían ser más intuitivos. Ej.: Auto-selección, filtrado de tipos de archivos durante carga, carga dinámica al seleccionar asignaturas. Validaciones de datos, advertencias podrían mejorar (Ej. validaciones de códigos de asignatura, menú desplegable en opciones como tipo de asignatura, entre otros). "
- "Sí lo recomendaría el uso de esta aplicación porque ayuda a gestionar las materias en cuanto al contenido, y es una gran utilidad al tener acceso a una app móvil y también a un panel administrativo donde el Profesor puede subir los contenidos de la materia."

- "Sí, siempre son necesarias herramientas como estas que permitan acercar al estudiante y al profesor para mejorar intercambio de conocimientos y a su vez mejorar el perfil del egresado."
- "Sí, me parece que puede ser muy útil al momento de compartir información con estudiantes, ya que todo lo compartido quedaría almacenado en el repositorio y todo estará en un mismo lugar."

4.3 Pruebas de rapidez de descarga de los videos

Se realizaron pruebas en el Laboratorio de Redes Móviles e Inalámbricas de la Escuela de Computación de la Facultad de Ciencias de la Universidad Central de Venezuela y entonces se observó mayor rapidez en la descarga de los videos usando esta aplicación en comparación con la velocidad de descarga de otras aplicaciones de la web, mientras que en la aplicación móvil se descargaban y reproducían los videos en las otras aplicaciones de la web ni siquiera se empezaban a transmitir. Se pudo apreciar la rapidez y el poco tiempo que demoraba los videos en cargarse e iniciar su reproducción. Esta observación no se cuantificó objetivamente porque no se dieron las condiciones ideales para realizar las pruebas correspondientes.

5. Resultados

Se cumplieron los objetivos específicos que se plantearon como justificación del presente trabajo haciendo énfasis en una mejora a futuro en cuanto a la interfaz en los módulos de administración de contenidos y posiblemente en la aplicación móvil. En base a los objetivos específicos planteados tenemos los siguientes resultados:

5.1 Se desarrolló de una aplicación móvil con tecnología Android

Se desarrolló una aplicación móvil con el lenguaje de programación Java y la plataforma Android, que abarcó las principales funcionalidades (desplegar documentos y videos) para la cual fue planteada, además se desarrolló sus interfaces procurando que fuese minimalista. Sin embargo se recogieron diferentes opiniones de los usuarios que utilizaron la aplicación y sugirieron que se realicen diferentes mejoras a la interfaz de forma de que esta luzca más atractiva en su uso y además de que su uso fuese más intuitivo.

La aplicación móvil, con todas las ventajas que da esta vía de comunicación, facilita la transmisión de contenidos de diferentes asignaturas, desde el profesor a sus alumnos, siempre y cuando se encuentren dentro de la facultad. Por ahora, la aplicación no funciona fuera de la facultad y si no se está conectado a una red interna. Sin embargo, esto no es una limitante dado el deficiente estado de la salida de la red interna de la Universidad Central de Venezuela y en este sentido, esta aplicación es una alternativa real al problema que suscita este escenario.

Para el funcionamiento de esta aplicación móvil se desarrollaron servicios web en PHP que permitieron comunicar la aplicación móvil con los otros componentes de la aplicación como lo son el servidor de *streaming* RED5, el servidor de contenidos nginx y el servidor de base de datos MariaDB.

5.2 Se desarrolló de una página web que permite la administración de contenidos

Se desarrollo una página web con los lenguajes de programación HTML, *JavaScript* y PHP, la cual cumple con las principales funcionalidades como lo son agregar/eliminar materias, agregar/eliminar contenidos. Fue elaborada con la intención de ofrecer una solución de administración primaria que puede ser objeto de otro Trabajo Especial de Grado para su posterior mejora.

5.3 Se configuró un servidor de *streaming* RED5 para el manejo de contenido multimedia

Se configuró un servidor de *streaming* RED5, habilitando sobre él una aplicación denominada por el servidor como "OflaDemo", la cual permite compartir contenido como videos o audios por medio de una distribución de carpetas en el servidor. Esta aplicación "OflaDemo" va permitir la consulta de diferentes videos desde la aplicación móvil por medio del protocolo RTPM.

Mediante pruebas realizadas en el Laboratorio de redes móviles e inalámbricas de la Escuela de Computación de la Universidad Central de Venezuela, se determino que la carga de los videos es relativamente rápida con respecto al uso de datos tradicionales.

Este quizás es uno de los hallazgos más importantes de este Trabajo Especial de Grado.

5.4 Se diseñó una base de datos SQL para información de usuario y sistema

Se diseñó e implementó una base de datos relacional SQL en un sistema de bases de datos MariaDB. La estructura inicial de esta base de datos descrita en capítulos anteriores, permitió la interacción de la aplicación móvil con información importante de esta aplicación y de los usuarios que operan con ella.

Conclusiones

El presente Trabajo De Grado permite abordar la primera plataforma de *m-learning* para la Escuela de Computación de la Universidad Central de Venezuela. A su vez, no es solo una herramienta de consulta de contenidos móvil, sino también una aplicación que permite la administración de la herramienta. Con la culminación de esta plataforma se obtienen las siguientes conclusiones:

- Resulta de gran importancia de que exista una herramienta que los profesores puedan utilizar, sirviéndoles como apoyo al proceso de aprendizaje de los estudiantes.
- Se diseñó una plataforma tecnológica que permite facilitar el proceso aprendizaje a través de *m-learning* para los profesores y estudiantes para la Escuela de Computación de la Facultad de Ciencias de la UCV.
- La carga de videos es relativamente rápida con respecto al uso de datos tradicionales.
- A través de la herramienta de administración se logra de forma dinámica cargar y eliminar diferentes documentos o videos relacionados con distintas asignaturas.
- Los usuarios finales se mostraron satisfechos con el uso de la aplicación y se puede.

Trabajos Futuros

A continuación se presentan una serie de recomendaciones, para el uso y expansión de la plataforma de aprendizaje:

- Promover la instalación de la aplicación para que sea accesible a usuarios potenciales, profesores y estudiantes.
- El uso de la aplicación dará pie para que se elaboren las mejoras necesarias. Por ejemplo que la aplicación permita compartir la información bidireccional de profesor a estudiante, de estudiante a profesor y entre estudiantes.
- Se sugiere a los profesores el empleo de alguna herramienta que les facilite la creación de videos para así nutrir al sistema.
- Que la aplicación posea otros elementos educativos como el foro, el chat, las encuestas o cuestionarios, wikis, entre otros.
- Integrar la aplicación a una red distinta a la interna de tal manera que se pueda acceder al contenido desde cualquier lugar.
- Mejorar interfaz de las aplicaciones una vez que ya se tiene instalado toda la plataforma se podría realizar un trabajo especial de grado con el objetivo de mejorar estos aspectos.
- Intentar agregar videos con subtítulos en el servidor Red5 de tal manera de reutilizar material ya hecho por universidades extranjeras.

Bibliografía

Brazuelo Grund, Francisco, y Gallego Gil, Domingo J. (2011) Mobile Learning. Los dispositivos móviles como recurso educativo. Sevilla: Editorial MAD, S.L.

BARRIOS, CARLOS E. LEÓN, JOSÉ. ROMERO, YENNY.. (2015). Metodología de Roger Pressman. Mayo 2015, de Wiki espacios Sitio web: <https://sisteminformacii.wikispaces.com/METODOLOG%C3%8DA+DE+PRESSMAN+-+2DA+PARTE>.

Cabrero, Julio. (2006). Bases pedagógicas del e-learning. Revista de Universidad Sociedad del Conocimiento, Vol 3, N1, 2,5.

Comunidad Mariadb. (2015). Información general de Mariadb. Mayo 2015, de Mariadb.org Sitio web: <https://mariadb.com>.

Comunidad PHP. (2015). ¿Qué es PHP?. Mayo 2015, de PHP Sitio web: <http://php.net/manual/es/intro-what-is.php>.

Daniel Rijo Sciara. (2004). Monografía para el curso de postgrado: “Codificación de Imágenes y Video”. Mayo 2015, de Universidad de la República Montevideo Sitio web: <http://iie.fing.edu.uy/ense/asign/codif/material/monografias/2004-01.pdf>.

Developers Android. (2015). Android Studio Overview. Mayo 2015, de Developers Android Sitio web: <http://developer.android.com/tools/studio/index.html>.

Elmasri, R., & Navathe, S. (2007). Sistemas de Bases de Datos. Conceptos Fundamentales. México: Addison-Wesley Iberoamericana.

H. Parmar, Ed. M. Thornburgh, Ed. (2012). Especificación RTMP. Mayo 2015, de Adobe Sitio web: http://www.images.adobe.com/content/dam/Adobe/en/devnet/rtmp/pdf/rtmp_specification_1.0.pdf.

Jesús Tomás. (2015). Arquitectura Android. Mayo 2015, de Universidad Politécnica de Valencia Sitio web: <http://www.androidcurso.com/index.php/tutoriales-android/31-unidad-1-vision-general-y-entorno-de-desarrollo/99-arquitectura-de-android>.

Ministerio Español para la educación . (2008). ¿Qué es el streaming?. Mayo 2015, de Ministerio de Educación, Política Social y Deporte. Instituto Superior de Formación y Recursos en Red para el Profesorado Sitio web: <http://www.ite.educacion.es/formacion/materiales/107/cd/video/video0103.html>.

M Sharples, J Taylor, G Vavoula . (2005). Towards a Theory of Mobile Learning . Abril 2015, de University of Birmingham, UK, 2 The Open University, UK Sitio web: <http://www.mlearn.org.za/CD/papers/Sharples-%20Theory%20of%20Mobile.pdf>.

nginx. (2015). Wiki documentation. Mayo 2015, de nginx Sitio web: <https://www.nginx.com/resources/wiki/>.

O'Malley et al., 2003, Vavoula, G., Glew, J.P., Taylor, J., Sharples, M., &Lefrere, P. . (2003). MOBIlearn WP4 – Guidelines for learning/teaching/tutoring in a mobile environment.. Abril 2015, de HAL archives ouvertes Sitio web: <https://hal.archives-ouvertes.fr/hal-00696244/document>.

Quinn, C. . (2000). mLearning: Mobile, Wireless, in your Pocket Learning. Mayo 2015, de Line Zine Sitio web: www.linezine.com/2.1/features/cqmmwiyp.htm.

ROSENBERG, M.J. (2001): e-Learning. Strategies for delivering knowledge in the Digital Age. New York, McGraw-Hill.

Steven Gong, Paul Gregoire, Daniel Rossi,(2008) Especificación Red5. Mayo 2015 de Red 5 Open Source Sitio web red5.googlecode.com/svn/.../red5-reference-1.0.pdf .

Silberschatz, Korth&Sudarshan (2002). FUNDAMENTOS DE BASES DE DATOS. Cuarta edición. España: McGraw-Hill Inc.

Universidad Politécnica de Madrid. (2013). Guía para la implantación del MOBILE LEARNING . Mayo 2015, de Gobierno de tele-educación Sitio web: http://serviciosgate.upm.es/docs/asesoramiento/guia_implementation_movil.pdf .