



Universidad de San Carlos de Guatemala  
Facultad de Ingeniería  
Escuela de Ingeniería en Ciencias y Sistemas

**IMPLEMENTACIÓN DE NUEVAS TECNOLOGÍAS MÓVILES ANDROID  
PARA DIFUSIÓN DE TUTORÍAS VIRTUALES EN TIEMPO REAL**

**Brian Steve Estrada Guzmán**

**Rony Oswaldo Arredondo Torres**

Asesorado por el Ing. Edgar Estuardo Santos Sutuj

Guatemala, octubre de 2016

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA



FACULTAD DE INGENIERÍA

**IMPLEMENTACIÓN DE NUEVAS TECNOLOGÍAS MÓVILES ANDROID  
PARA DIFUSIÓN DE TUTORÍAS VIRTUALES EN TIEMPO REAL**

TRABAJO DE GRADUACIÓN

PRESENTADO A LA JUNTA DIRECTIVA DE LA  
FACULTAD DE INGENIERÍA

POR

**BRIAN STEVE ESTRADA GUZMÁN**

**RONY OSWALDO ARREDONDO TORRES**

ASESORADO POR EL ING. EDGAR ESTUARDO SANTOS SUTUJ

AL CONFERÍRSELES EL TÍTULO DE

**INGENIEROS EN CIENCIAS Y SISTEMAS**

GUATEMALA, OCTUBRE DE 2016

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA  
FACULTAD DE INGENIERÍA



**NÓMINA DE JUNTA DIRECTIVA**

DECANO	Ing. Pedro Antonio Aguilar Polanco
VOCAL I	Ing. Angel Roberto Sic García
VOCAL II	Ing. Pablo Christian de León Rodríguez
VOCAL III	Inga. Elvia Miriam Ruballos Samayoa
VOCAL IV	Br. Raúl Eduardo Ticún Córdova
VOCAL V	Br. Henry Fernando Duarte García
SECRETARIA	Inga. Lesbia Magalí Herrera López

**TRIBUNAL QUE PRACTICÓ EL EXAMEN GENERAL PRIVADO**

DECANO	Ing. Pedro Antonio Aguilar Polanco
EXAMINADOR	Ing. Oscar Alejandro Paz Campos
EXAMINADOR	Ing. José Alfredo González Díaz
EXAMINADOR	Ing. Sergio Arnaldo Méndez Aguilar
SECRETARIA	Inga. Lesbia Magalí Herrera López

**Brian Steve Estrada Guzmán**

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA  
FACULTAD DE INGENIERÍA



**NÓMINA DE JUNTA DIRECTIVA**

DECANO	Ing. Pedro Antonio Aguilar Polanco
VOCAL I	Ing. Angel Roberto Sic García
VOCAL II	Ing. Pablo Christian de León Rodríguez
VOCAL III	Inga. Elvia Miriam Ruballos Samayoa
VOCAL IV	Br. Raúl Eduardo Ticún Córdova
VOCAL V	Br. Henry Fernando Duarte García
SECRETARIA	Inga. Lesbia Magalí Herrera López

**TRIBUNAL QUE PRACTICÓ EL EXAMEN GENERAL PRIVADO**

DECANO	Ing. Murphy Olympo Paiz Recinos
EXAMINADOR	Ing. Edgar Estuardo Santos Sutuj
EXAMINADOR	Ing. Marlon Francisco Orellana López
EXAMINADOR	Ing. José Alfredo González Días
SECRETARIO	Ing. Hugo Humberto Rivera Pérez

**Rony Oswaldo Arredondo Torres**

## **HONORABLE TRIBUNAL EXAMINADOR**

En cumplimiento con los preceptos que establece la ley de la Universidad de San Carlos de Guatemala, presentamos a su consideración nuestro trabajo de graduación titulado:

### **IMPLEMENTACIÓN DE NUEVAS TECNOLOGÍAS MÓVILES ANDROID PARA DIFUSIÓN DE TUTORÍAS VIRTUALES EN TIEMPO REAL**

Tema que nos fuera asignado por la Dirección de la Escuela de Ingeniería en Ciencias y Sistemas, con fecha febrero 2015.

**Rony Oswaldo Arredondo Torres**

**Brian Steve Estrada Guzmán**

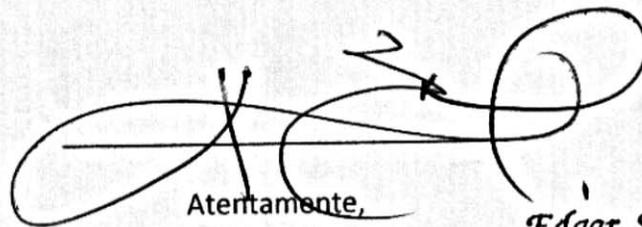
Guatemala, 15 de octubre de 2015

Ingeniero  
Marlon Antonio Pérez Türk  
Director  
Escuela de Ciencias y Sistemas  
Facultad de Ingeniería  
Universidad de San Carlos de Guatemala

Ingeniero Pérez Türk:

Me complace saludarle haciendo referencia al trabajo de graduación titulado "IMPLEMENTACIÓN DE NUEVAS TECNOLOGÍAS MÓVILES ANDROID PARA DIFUSIÓN DE TUTORÍAS VIRTUALES EN TIEMPO REAL", desarrollado por los estudiantes universitarios Rony Oswaldo Arredondo Torres con número de carné 200815456 y Brian Steve Estrada Guzmán con número de carné 200819048, que como asesor apruebo el contenido del mismo.

Para su conocimiento y efectos, sin otro particular, me suscribo.



Atentamente,  
Ing. Edgar Santos  
No. Colegiado: 5266  
Asesor

*Edgar Santos*  
INGENIERO EN CIENCIAS Y SISTEMAS  
Colegiado 5266



Universidad San Carlos de Guatemala  
Facultad de Ingeniería  
Escuela de Ingeniería en Ciencias y Sistemas

Guatemala, 4 de Noviembre del 2015

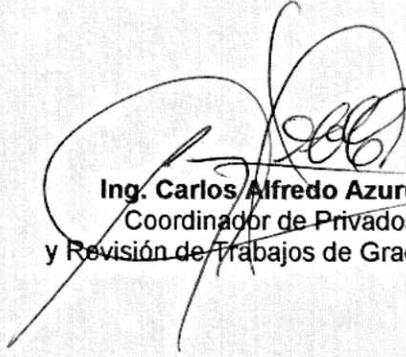
Ingeniero  
**Marlon Antonio Pérez Türk**  
Director de la Escuela de Ingeniería  
En Ciencias y Sistemas

Respetable Ingeniero Pérez:

Por este medio hago de su conocimiento que he revisado el trabajo de graduación de los estudiantes **BRIAN STEVE ESTRADA GUZMÁN** con carné **2008-19048**, y **RONY OSWALDO ARREDONDO TORRES** con carné **2008-15456**, titulado: **"IMPLEMENTACIÓN DE NUEVAS TECNOLOGÍAS MÓVILES ANDROID PARA DIFUSIÓN DE TUTORÍAS VIRTUALES EN TIEMPO REAL"**, y a mi criterio el mismo cumple con los objetivos propuestos para su desarrollo, según el protocolo.

Al agradecer su atención a la presente, aprovecho la oportunidad para suscribirme,

Atentamente,

  
**Ing. Carlos Alfredo Azurdia**  
Coordinador de Privados  
y Revisión de Trabajos de Graduación





Universidad San Carlos de Guatemala  
Facultad de Ingeniería  
Escuela de Ingeniería en Ciencias y Sistemas

Guatemala, 4 de Noviembre del 2015

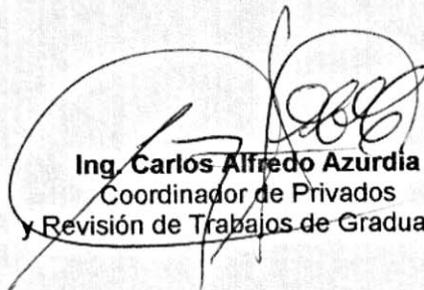
Ingeniero  
**Marlon Antonio Pérez Türk**  
Director de la Escuela de Ingeniería  
En Ciencias y Sistemas

Respetable Ingeniero Pérez:

Por este medio hago de su conocimiento que he revisado el trabajo de graduación de los estudiantes **BRIAN STEVE ESTRADA GUZMÁN** con carné 2008-19048, y **RONY OSWALDO ARREDONDO TORRES** con carné 2008-15456, titulado: **"IMPLEMENTACIÓN DE NUEVAS TECNOLOGÍAS MÓVILES ANDROID PARA DIFUSIÓN DE TUTORÍAS VIRTUALES EN TIEMPO REAL"**, y a mi criterio el mismo cumple con los objetivos propuestos para su desarrollo, según el protocolo.

Al agradecer su atención a la presente, aprovecho la oportunidad para suscribirme,

Atentamente,

  
**Ing. Carlos Alfredo Azurdia**  
Coordinador de Privados  
y Revisión de Trabajos de Graduación



E  
S  
C  
U  
E  
L  
A  
  
D  
E  
  
I  
N  
G  
E  
N  
I  
E  
R  
Í  
A  
  
E  
N  
  
C  
I  
E  
N  
C  
I  
A  
S  
  
Y  
  
S  
I  
S  
T  
E  
M  
A  
S

UNIVERSIDAD DE SAN CARLOS  
DE GUATEMALA



FACULTAD DE INGENIERÍA  
ESCUELA DE INGENIERÍA EN  
CIENCIAS Y SISTEMAS  
TEL: 24188000 Ext. 1534

*El Director de la Escuela de Ingeniería en Ciencias y Sistemas de la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala, luego de conocer el dictamen del asesor con el visto bueno del revisor y del Licenciado en Letras, del trabajo de graduación **"IMPLEMENTACIÓN DE NUEVAS TECNOLOGÍAS MÓVILES ANDROID PARA DIFUSIÓN DE TUTORÍAS VIRTUALES EN TIEMPO REAL"**, realizado por los estudiantes, BRIAN STEVE ESTRADA GUZMÁN y RONY OSWALDO ARREDONDO TORRES, aprueba el presente trabajo y solicita la autorización del mismo.*

**"ID Y ENSEÑAD A TODOS"**

*Ing. Marlon Antonio Pérez Türk*  
**Director**

**Escuela de Ingeniería en Ciencias y Sistemas**



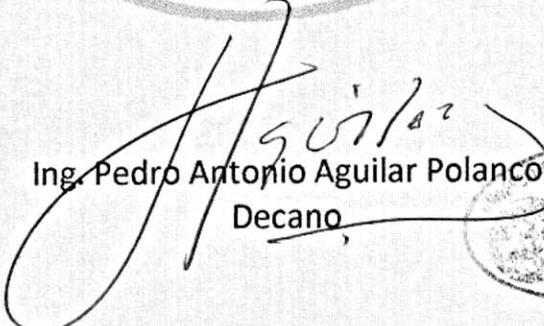
*Guatemala, 03 de octubre de 2016*



DTG. 477.2015

El Decano de la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala, luego de conocer la aprobación por parte del Director de la Escuela de Ingeniería en Ciencias y Sistemas, al Trabajo de Graduación titulado: **IMPLEMENTACIÓN DE NUEVAS TECNOLOGÍAS MÓVILES ANDROID PARA DIFUSIÓN DE TUTORÍAS VIRTUALES EN TIEMPO REAL**, presentado por los estudiantes universitarios: **Brian Steve Estrada Guzmán y Rony Oswaldo Arredondo Torres**, y después de haber culminado las revisiones previas bajo la responsabilidad de las instancias correspondientes, autoriza la impresión del mismo.

IMPRÍMASE:

  
Ing. Pedro Antonio Aguilar Polanco  
Decano



Guatemala, octubre de 2016

/gdech

## **ACTO QUE DEDICO A:**

- Dios** Por darme el don de la vida y por ser el pilar de sabiduría e incondicional amor.
- Mis padres** Edgar Arredondo y Aura Torres de Arredondo, por su amor incondicional y apoyo en estos años de preparación académica.
- Mis hermanos** Por su valiosa presencia en mi vida y por brindarme apoyo y momentos inolvidables en mi crecimiento personal y profesional.

**Rony Oswaldo Arredondo Torres**

## **ACTO QUE DEDICO A:**

- Dios** Por darme el don de la vida y por ser el pilar de sabiduría e incondicional amor.
- Mis padres** por su amor incondicional y apoyo en estos años de preparación académica.
- Mis hermanos** Por su valiosa presencia en mi vida y por brindarme apoyo y momentos inolvidables en mi crecimiento personal y profesional.

**Brian Steve Estrada Guzmán**

## **AGRADECIMIENTOS A:**

**Universidad de San  
Carlos de Guatemala**

Por ser la institución en la cual nos formamos como profesionales, entre otras cosas.

**Facultad de Ingeniería**

Por brindarnos los conocimientos básicos y avanzados para ser grandes profesionales.

**Mis amigos**

Por brindarnos su incondicional amistad y ser gran apoyo en estos años de estudios universitarios.

## ÍNDICE GENERAL

ÍNDICE DE ILUSTRACIONES.....	VII
LISTA DE SÍMBOLOS .....	IX
GLOSARIO .....	XI
RESUMEN.....	XV
OBJETIVOS.....	XVII
INTRODUCCIÓN.....	XIX
1. IDENTIFICANDO LA TEORÍA DE SISTEMAS DE INFORMACIÓN QUE SOPORTA LA INVESTIGACIÓN.....	1
1.1. ¿Qué es una teoría?.....	1
1.2. ¿Por qué es importante tener una teoría que respalde la investigación?.....	1
1.3. Teoría seleccionada .....	2
1.3.1. Sincronicidad .....	2
1.3.2. Velocidad de transmisión.....	3
1.3.3. Paralelismo .....	3
1.3.4. Conjunto de símbolos .....	3
1.3.5. Verificable.....	4
1.3.6. Reprocesable.....	4
2. TEORÍA DE SISTEMA DE INFORMACIÓN Y SU RELACIÓN CON LA TECNOLOGÍA ESCOGIDA.....	7
2.1. Esquema de MST.....	8
2.2. Comunicación.....	9
2.3. Videoconferencia.....	10

2.3.1.	Señal de audio .....	10
2.3.2.	Señal de video.....	11
2.3.3.	Flujo de video .....	11
2.3.4.	Imágenes por segundo.....	11
2.3.5.	Resolución de vídeo .....	12
2.3.6.	Pixel .....	12
2.3.7.	Relación de aspecto .....	12
2.3.8.	Compresión .....	13
2.3.9.	Tasa de bits.....	13
2.4.	Formatos multimedia.....	13
2.4.1.	Flash video (FLV) .....	13
2.4.2.	Formato contenedor .....	14
2.4.3.	Small Web Format (SWF) .....	15
2.4.4.	Sorenson codec .....	15
2.4.5.	MPEG-2 Audio Layer 3 .....	16
2.4.6.	Real Time Message Protocol .....	17
2.4.7.	Paquete RTMP .....	17
2.4.8.	Red5 Media Server .....	21
2.5.	El impacto de la teoría en Guatemala .....	21
2.5.1.	Internet .....	22
2.5.2.	Telefonía .....	23
2.5.3.	Android.....	24
2.5.4.	Google Play.....	24
3.	IDENTIFICACIÓN DEL PROBLEMA Y SOLUCIÓN QUE LA APLICACIÓN BRINDARÁ.....	25
3.1.	Antecedentes .....	25
3.2.	Descripción del problema .....	27
3.3.	Software a desarrollar .....	28

3.4.	Mercado objetivo .....	29
3.5.	<i>Benchmark</i> .....	29
3.5.1.	Cisco WebEx reuniones para Android .....	30
3.5.2.	GoToMeeting .....	31
3.5.3.	OpenMeetings .....	32
4.	DISEÑO DE LA APLICACIÓN Y SOLUCIÓN QUE LA APLICACIÓN RALIZARÁ.....	35
4.1.	Resumen de la aplicación.....	35
4.2.	Proceso básico del proyecto.....	36
4.3.	Eco.Profesor .....	36
4.3.1.	Menú.....	37
4.3.2.	Iniciar sesión.....	38
4.3.3.	Validaciones y errores .....	38
4.3.4.	Listado de cursos.....	39
4.3.5.	Ventana de video .....	40
4.4.	Eco.Aula .....	41
4.4.1.	Iniciar sesión.....	42
4.4.2.	Configurar cursos .....	42
4.4.3.	Administrar cursos .....	42
4.4.4.	Importar cursos.....	43
4.4.5.	Administrar usuarios .....	43
4.5.	Eco.Estudiante .....	43
4.5.1.	Menú.....	44
4.5.2.	Preferencias.....	45
4.5.3.	Iniciar sesión.....	47
4.5.4.	Cursos suscritos .....	48
4.6.	Suscribir curso.....	49
4.7.	Reproductor.....	51

5.	DOCUMENTACIÓN Y TUTORIAL DE PROGRAMACIÓN DE LA APLICACIÓN .....	53
5.1.	Requerimientos para desarrollo y pruebas.....	53
5.2.	Herramientas para el desarrollo .....	53
5.2.1.	Eclipse.....	53
5.2.2.	Netbeans.....	54
5.2.3.	Java.....	55
5.2.4.	Android SDK.....	55
5.2.5.	Hardware.....	55
5.2.6.	Ordenador .....	56
5.2.7.	Dispositivo con Android.....	56
5.2.8.	Servidor.....	56
5.3.	Software.....	57
5.3.1.	Webcam-capture .....	57
5.3.2.	Xuggle-Xuggler.....	58
5.3.3.	Log4j .....	59
5.3.4.	Zodiac.....	60
5.4.	Json-simple .....	60
5.4.1.	Vitatio .....	60
5.5.	Arquitectura de eco .....	61
5.6.	Aplicación Eco.Profesor .....	62
5.6.1.	Arquitectura de la aplicación .....	62
5.6.1.1.	Modelo.....	63
5.6.1.2.	Vista .....	63
5.6.1.3.	Controlador.....	64
5.7.	Aplicación Eco.Aula.....	65
5.7.1.	Arquitectura de la aplicación .....	65
5.7.1.1.	Páginas web.....	66
5.7.1.2.	Servicio web .....	67

5.8.	Aplicación Eco.Estudiante .....	67
5.8.1.	Arquitectura de la aplicación.....	68
5.8.1.1.	Vista.....	68
5.8.1.2.	Controlador .....	69
5.8.1.3.	Manifiesto de Android .....	69
	CONCLUSIONES .....	71
	RECOMENDACIONES.....	73
	BIBLIOGRAFÍA.....	75



# ÍNDICE DE ILUSTRACIONES

## FIGURAS

1.	Esquema de Sincronización de Multimedia.....	8
2.	Trama de paquete RTMP Tipo 0/1 .....	18
3.	Trama de paquete RTMP Tipo 2 .....	19
4.	Trama de paquete RTMP .....	20
5.	Flujo de la aplicación .....	37
6.	Inicio de sesión.....	39
7.	Visualización de video .....	41
8.	Diagrama de flujo de estudiante.....	44
9.	Listado de cursos en estudiante.....	45
10.	Preferencias de estudiante.....	46
11.	Inicio de sesión de estudiante .....	47
12.	Listado de cursos suscritos .....	48
13.	Buscar cursos.....	49
14.	Confirmación de suscripción .....	50
15.	Visualización de cursos .....	51
16.	Diagrama de arquitectura del software.....	62
17.	Jerarquía de vista Eco.profesor.....	64
18.	Jerarquía de vista Eco.aula .....	66

## TABLAS

I.	Tabla comparativa de medios de comunicación .....	5
II.	Requerimientos de hardware para implementación .....	57



## LISTA DE SÍMBOLOS

<b>Símbolo</b>	<b>Significado</b>
<b>Hz</b>	Hercio. Unidad de frecuencia del sistema internacional.
<b>KHz</b>	Kilohercio. 1 KHz = 1 000Hz.
<b>MB</b>	Unidad de medida de almacenamiento de información. 1 MB = 1 048 576 bytes.



## GLOSARIO

<b>Android SDK</b>	Plataforma móvil, sistema operativo orientado para dispositivos móviles basado en Linux.
<b>Base de datos (BD)</b>	Conjunto de datos que pertenecen a un contexto similar y que se almacenan para su posterior uso.
<b>Benchmark</b>	Técnica usada para medir el rendimiento de un sistema en comparación con otros sistemas similares. Es un anglicismo traducido al español como comparativa.
<b>Interfaz</b>	La conexión física entre dos sistemas de cualquier tipo dando una comunicación entre diferentes niveles.
<b>Java</b>	Lenguaje de programación orientado al paradigma de objetos el cual permite desarrollar aplicaciones de software para cualquier sistema operativo.
<b>JDK</b>	<i>Java development kit</i> por sus siglas en inglés. Es un conjunto de paquetes de software que provee herramientas de desarrollo para la creación de programas en Java.

<b>JSON</b>	Acrónimo de JavaScript Object Notation. Es un formato ligero para el intercambio de datos. Un subconjunto de la notación literal de objetos de JavaScript que no requiere el uso de XML.
<b>Log4j</b>	Librería para Java diseñada para el manejo de mensajes de error en tiempo de ejecución de una aplicación de software.
<b>NetBeans</b>	Es un entorno de desarrollo de código abierto es principalmente para el lenguaje Java. Permite escribir, depurar, compilar y ejecutar programas.
<b>RMTP</b>	<i>Real time messaging protocol</i> , protocolo para la transmisión de audio y video en tiempo real. Desarrollado por la empresa Macromedia.
<b>SDK</b>	<i>Software development kit</i> , kit completo para desarrollo de software sobre plataforma Android. Conjunto de herramientas que permiten crear aplicaciones para móviles con sistema operativo Android.
<b>Smartphone</b>	Dispositivo móvil elaborado sobre una plataforma con una mayor capacidad y conectividad que un teléfono móvil convencional.

<b>Web service</b>	Tecnología que utiliza un conjunto de protocolos y estándares que sirven para intercambiar datos entre aplicaciones.
<b>Xuggle-xuggler</b>	Librería para Java de código abierto que permite comprimir, descomprimir y manipular videos grados previamente o ya sea en tiempo real.
<b>Zodiac</b>	Librería que contiene componentes para el manejo de patrones de diseño <i>data access object</i> , <i>data transfer object</i> y <i>service-oriented architecture</i> .



## RESUMEN

En la actualidad muchos jóvenes entran a la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala (Fiusac) con la ilusión de convertirse en los nuevos profesionales del mañana. Pero muchos de estos sueños se ven truncados ya que muchas veces es muy difícil transportarse a las instalaciones centrales de la USAC diferentes motivos, como la falta de recursos económicos, residencia en lugares muy lejanos y debido al tráfico de la ciudad por las mañanas, el cual es muy pesado. Simplemente las aulas en las cuales se imparten los cursos no cuentan con un espacio físico suficiente, para abastecer la demanda de estudiantes.

Debido a que cada día son más los jóvenes que ingresan en la Fiusac se ha buscado distintas formas de facilitar el aprendizaje de los estudiantes. Sobre todo mantener la calidad y el nivel académico que caracterizan a la Facultad.

Por estas razones y por las necesidades que presenta, en este sentido, la Fiusac; en este documento se describe el proceso para desarrollar una aplicación. Esta servirá para recibir cátedras de forma virtual y en tiempo real, es decir, el catedrático estará impartiendo su clase en el salón del curso y por medio de una aplicación móvil los estudiantes podrán acceder, desde cualquier parte donde tengan acceso a internet, a ver y escuchar la clase impartida en tiempo real.



## **OBJETIVOS**

### **General**

Desarrollar una aplicación para dispositivos móviles con sistema operativo Android, que permita a los estudiantes de la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala tener acceso a recibir cátedras de forma virtual y en tiempo real.

### **Específicos**

1. Elaborar un manual/tutorial explicando de forma detallada el proceso de desarrollo de una aplicación en Android.
2. Fomentar el desarrollo de nuevos profesionales a través de la implementación de nueva tecnología en el proceso de enseñanza de la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala.
3. Dotar a la Facultad de Ingeniería de una herramienta innovadora capaz de brindarle a sus estudiantes una nueva vía de comunicación, facilitando la asistencia de los mismos a las cátedras impartidas.
4. Motivar a la Facultad de Ingeniería a tener una fuerte presencia en internet, impartiendo cursos de forma virtual.



## INTRODUCCIÓN

La Universidad de San Carlos de Guatemala (USAC) es una institución dedicada a la educación superior y dirigida al pueblo en general por ser estatal. Por tanto, es la cuna del conocimiento y debe estar a la vanguardia en los avances tecnológicos. Es por esto que los proyectos deben ser innovadores y dirigidos con proyección social, para proveer a los estudiantes de las herramientas necesarias. Es debido a esto que la propuesta de problema y solución presentada en este documento, pretende facilitar un medio de comunicación apto para la enseñanza.

El presente trabajo de graduación está compuesto por cuatro capítulos. El primer capítulo identifica y define la teoría que soporta el proyecto, así como el impacto de la misma en la República de Guatemala. Además se incluyen los conceptos teóricos que se creen convenientes para facilitar al lector el entendimiento de elementos y protocolos empleados en el desarrollo del proyecto. En el segundo capítulo se aborda el problema que se desea resolver, mediante la identificación y propuesta de solución.

En el tercer capítulo se describe la estructura del sistema presentando prototipos y descripciones de la funcionalidad. Por último, en el cuarto capítulo, se presentan todos los requerimientos para el desarrollo o continuidad del proyecto, así como la descripción del diseño y la arquitectura del sistema.



# **1. IDENTIFICANDO LA TEORÍA DE SISTEMAS DE INFORMACIÓN QUE SOPORTA LA INVESTIGACIÓN**

## **1.1. ¿Qué es una teoría?**

La teoría es una entidad abstracta que modela un sistema lógico-deductivo. Esto a partir de un conjunto de hipótesis formadas por especulación que fueron comprobadas, pretendiendo ser una explicación acertada a fenómenos observables.

Por otro lado, la teoría científica se basa en un sistema abstracto que busca establecer una descripción científica. Esto a través de un conjunto de observaciones o la realización de experimentos para comprobar una hipótesis o conjetura sobre un hecho específico.

## **1.2. ¿Por qué es importante tener una teoría que respalde la investigación?**

La importancia de la teoría radica en describir, explicar y predecir el comportamiento del fenómeno al que se refiere. Algunas otras funciones que tiene la teoría son la organización del conocimiento y la orientación de la investigación.

Las teorías tienen criterios de evaluación que son importantes para determinar la más adecuada. Entre ellos se tiene: capacidad de descripción, explicación y predicción, consistencia lógica, perspectiva, fructificación y parsimonia.

### **1.3. Teoría seleccionada**

Para este estudio de investigación se propone la teoría de sincronización de multimedia (acrónimo MST - Media Synchronicity Theory). Esta se puede describir como una innovación de la teoría de la riqueza en los medios de comunicación, desarrollada en el año 1986 por Richard L. Draft y Robert H. Lengel. Está enfocada en medir y clasificar los medios de comunicación como llamadas telefónicas, videoconferencias y correos electrónicos. Por ejemplo, podemos decir que un correo electrónico no reproduce información visual o auditiva (comúnmente es sólo texto) y por tanto es menos rico que una videoconferencia (contiene información visual y auditiva).

La teoría MST fue desarrollada por Alan R. Dennis, Robert M. Fuller y S. Valadich en 1999, y se traslada desde los medios de comunicación hacia la sincronización en los medios de comunicación como el indicador de rendimiento.

#### **1.3.1. Sincronicidad**

Según los autores de la teoría se define como: “un estado en el que los individuos están trabajando juntos al mismo tiempo con un foco común.”<sup>1</sup> La teoría MST afirma que cualquier comunicación se puede dividir en dos procesos fundamentales: conducción y convergencia.

El proceso de conducción es la transmisión y procesamiento de la información por parte del receptor para que este pueda leer, comprender e integrar para crear o modificar su paradigma sobre una determinada situación.

---

<sup>1</sup> Teoría MST.

El proceso de convergencia se da cuando, tanto el emisor del mensaje como el receptor del mismo, llegan a interpretar la información de manera que ambos pueden estar o no de acuerdo con el mensaje transmitido.

El soporte de la teoría MST se encuentra influenciado por cinco capacidades intrínsecas de los medios de comunicación:

### **1.3.2. Velocidad de transmisión**

Esta capacidad se refiere a qué tan rápido un mensaje puede llegar al receptor y ser respondido. Es la habilidad de un medio para soportar la rápida comunicación bidireccional.

### **1.3.3. Paralelismo**

Esta capacidad se refiere a la cantidad de conversaciones que pueden existir de forma simultánea. Por ejemplo, en el teléfono convencional únicamente puede existir una a la vez, en contraste con medios digitales como la mensajería instantánea que permiten múltiples. Sin embargo, cuanto mayor es el número de conversaciones, más difícil es el monitoreo y la coordinación de las mismas.

### **1.3.4. Conjunto de símbolos**

Esta capacidad se refiere a las diferentes formas en las cuales un mensaje puede ser codificado. Se debe considerar cuatro aspectos en los que un símbolo puede afectar la comunicación y el entendimiento de los mensajes. Estos son:

- Cierta información puede ser más fácilmente transmitida en un formato que en otro.
- Símbolos verbales y no verbales permiten al remitente incluir información adicional a las palabras.
- El costo de crear o procesar un mensaje que utilizan ciertos conjuntos de símbolos pueden ser muy altos y alterar la forma en que el remitente crea el mensaje o el receptor entiende el mensaje.
- La falta de símbolos verbales y no verbales afecta la percepción social del mensaje.

#### **1.3.5. Verificable**

Esta capacidad se refiere a la optimización de un mensaje antes de ser enviado. Se consideran aspectos como la edición de un mensaje mientras es enviado para asegurar que exprese correctamente el significado, sin necesidad de información adicional.

#### **1.3.6. Reprocesable**

Esta es la capacidad de recuperar y procesar nuevamente un mensaje recibido para su mejor entendimiento.

A continuación se muestra una tabla comparativa de distintos medios de comunicación y como éstos satisfacen las cinco capacidades:

Tabla I. **Tabla comparativa de medios de comunicación**

	Velocidad de transmisión	Conjunto de símbolos	Paralelismo	Verificable	Reprocesable
<b>Cara a cara</b>	Alto	Bajo-Alto	Bajo	Bajo	Bajo
<b>Videoconferencia</b>	Medio-Alto	Bajo-Alto	Bajo	Bajo	Bajo
<b>Teléfono</b>	Medio	Bajo	Bajo	Bajo	Bajo
<b>Correo</b>	Bajo	Bajo-Medio	Alto	Alto	Alto
<b>Correo de voz</b>	Bajo	Bajo	Bajo	Bajo-Medio	Alto
<b>Correo electrónico</b>	Bajo-Medio	Bajo-Alto	Medio	Alto	Alto
<b>Chat</b>	Medio	Bajo-Medio	Medio	Bajo-Medio	Bajo-Medio

Fuente: Rethinking Media Richness. *Towards a Theory of Media Synchronicity*.

[http://www.images.adobe.com/www.adobe.com/content/dam/Adobe/en/devnet/rtmp/pdf/rtmp\\_specification\\_1.0.pdf](http://www.images.adobe.com/www.adobe.com/content/dam/Adobe/en/devnet/rtmp/pdf/rtmp_specification_1.0.pdf). Consulta: mayo de 2015.



## **2. TEORÍA DE SISTEMA DE INFORMACIÓN Y SU RELACIÓN CON LA TECNOLOGÍA ESCOGIDA**

Según la teoría de sincronización de multimedia, el ambiente de comunicación que soporta alta velocidad de transmisión y bajo paralelismo fomenta la sincronidad. Esta es la clave en el proceso de convergencia. Observando la tabla 1 se determina que el mejor medio de comunicación es la conversación cara a cara. Sin embargo, la finalidad del proyecto es proveer una vía diferente de comunicación. Observando nuevamente la tabla 1 se identifica que la segunda mejor vía es la videoconferencia. Esto debido a que la valoración de las capacidades de la videoconferencia corresponden en su mayoría a las identificadas en el medio de comunicación cara a cara.

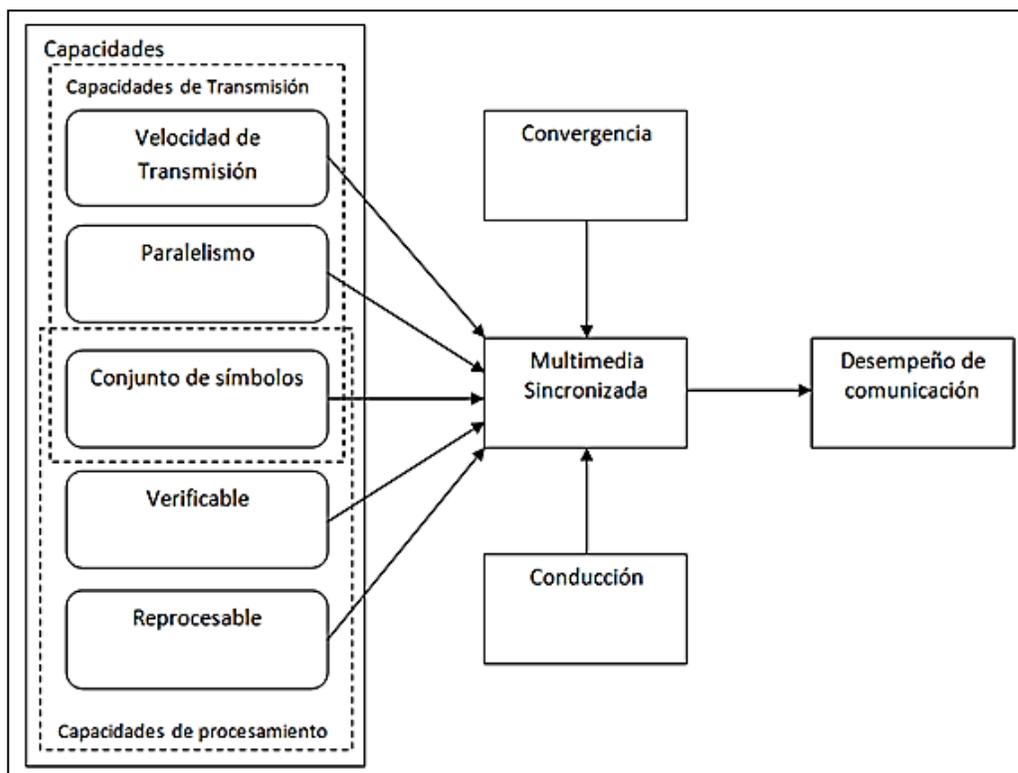
Además, la teoría indica la existencia de diversos factores en las situaciones que influyen y determinan el canal de comunicación que es efectivo para los grupos, aunque estos ejecuten tareas similares. El factor principal es si los miembros del grupo han trabajado juntos en el pasado, denominado grupo establecido. En un grupo establecido se han determinado reglas y comportamiento para cada miembro y los procesos están bien establecidos. Cuando una tarea es rutinaria el grupo se relaciona suavemente y directo a la ejecución con menos conflictos. Los grupos establecidos con normas aceptadas requieren menos comunicación altamente sincronizada.

Los estudiantes de la Fiusac, tomando algún curso, corresponden a grupos establecidos. Esto porque cuando un estudiante se agrega a un curso únicamente decimos que se está agregando un miembro al grupo, adaptándose a las reglas predefinidas.

## 2.1. Esquema de MST

En el esquema propuesto por Dennis y Valadich para la teoría “Sincronización de Multimedia”, se presenta la relación entre la capacidad de transmisión y la capacidad de procesamiento, para lograr un desempeño de comunicación adecuado.

Figura 1. Esquema de Sincronización de Multimedia



Fuente: Rethinking Media Richness. *Towards a Theory of Media Synchronicity*.

[http://wwwimages.adobe.com/www.adobe.com/content/dam/Adobe/en/devnet/rtmp/pdf/rtmp\\_specification\\_1.0.pdf](http://wwwimages.adobe.com/www.adobe.com/content/dam/Adobe/en/devnet/rtmp/pdf/rtmp_specification_1.0.pdf). Consulta: mayo de 2015.

## 2.2. Comunicación

La finalidad de la comunicación es el intercambio de información entre por lo menos dos entidades. El proceso de comunicación es una interacción entre las dos entidades que comparten un mismo conjunto de símbolos y el mismo sentido e interpretación para los estos símbolos.

Todas las formas de comunicación de forma común tiene los siguientes elementos:

- Fuente: punto desde el cual surge la información.
- Emisor: entidad que selecciona el conjunto de símbolos, codifica y envía el mensaje.
- Receptor: entidad a la que se dentista el mensaje y lo decodifica. Existen dos tipos de receptores.
  - Pasivo: solamente recibe el mensaje.
  - Activo: recibe el mensaje, interpreta, almacena e incluso intercambia su role para poder dar una respuesta.
- Conjunto de símbolos (código): conjunto de signos y reglas del lenguaje que el emisor utiliza para codificar su mensaje.
- Mensaje: información que se transmite del emisor al receptor.
- Canal: medio físico por el cual se transmite el mensaje.
- Situación: tiempo y lugar en el que se realiza el proceso de comunicación.
- Ruido: perturbaciones que sufre el mensaje mientras viaja en el canal.

## **2.3. Videoconferencia**

Es un proceso de comunicación capaz de ser implementado tanto con un canal unidireccional como bidireccional. La videoconferencia permite mantener reuniones con grupos de personas situadas físicamente en diferentes puntos alejados entre sí. Ofrece una gran variedad de ventajas al compartir desde distintas fuentes la información que se transmite.

La videoconferencia se compone de dos elementos fundamentales:

### **2.3.1. Señal de audio**

Se utiliza un micrófono que es dispositivo capaz de convertir las ondas de presión del aire en impulsos eléctricos. Luego serán convertidos nuevamente en señales audibles en unos altavoces. Los impulsos eléctricos son correspondientes a señales sonoras.

La señal que representa la voz humana tiene su relevancia en su espectro de frecuencia hasta los 10 kHz. La telefonía fija utiliza solamente los primeros 3,8 kHz, ya que con esto basta para que la voz sea comprensible, aunque no para reconocer al hablante. El espectro de frecuencias audible de un oído sano se encuentra entre las frecuencias 20 Hz y 20kHz.

El espectro de frecuencias audible se divide en tres categorías, en función de los tonos:

- Tonos graves: frecuencias comprendidas entre los 16 Hz y 256Hz
- Tonos medios: frecuencias comprendidas entre los 256 Hz y 2 kHz.
- Tonos agudos: frecuencias comprendidas entre los 2 kHz y 16 kHz

Las ondas acústicas con frecuencias que se encuentran por debajo de los 20 Hz se les llama infrasonidos. Mientras que las que se encuentra por sobre los 20 kHz se les llama ultrasonidos.

### **2.3.2. Señal de video**

La imagen de un video está formada por color y luz. La luz define la imagen entre negro y blanco, y a esta luz se le denomina luminancia.

El color se monta por encima de la luminancia y se le llama crominancia. Estos colores están definidos por dos magnitudes: saturación, que indica la cantidad de color; y el matiz, que indica el color.

### **2.3.3. Flujo de video**

El video se forma a partir de una sucesión de imágenes. Estas que son intercambiadas tan rápidamente en un recuadro limitado que el cerebro humano termina interpretando ese recuadro como un vídeo.

### **2.3.4. Imágenes por segundo**

Se refiere a la cantidad de imágenes por unidad de tiempo que son mostradas secuencialmente en el recuadro de vídeo. Para lograr la ilusión de movimiento se requieren un mínimo de 15 imágenes por segundo (fps - *frames per second*), pero los estándares varían entre 24 fps y 30 fps.

### **2.3.5. Resolución de vídeo**

El tamaño de las imágenes se mide en píxeles para vídeo digital. La resolución viene dado como el producto del ancho por el alto, medidos en píxeles, con lo que se obtiene una relación de aspecto.

Los tamaños de una pantalla pueden ser: tamaño absoluto, que son las medidas del monitor físico, generalmente en pulgadas; y el tamaño relativo, que se refiere a la cantidad de píxeles que puede mostrar. Mientras más píxeles pueda mostrar una pantalla, mayor será la calidad de imagen; siempre y cuando ésta cuente con esa cantidad de píxeles. El pixel es la unidad mínima de información, que se puede mostrar en una pantalla.

### **2.3.6. Pixel**

Es la menor unidad de información que forma parte de una imagen digital, independientemente si pertenecen a una fotografía o a un vídeo. En su mayoría los dispositivos utilizados como pantallas utilizan el modelo RGB, en el cual se compone un color a partir diferentes intensidades de los colores primarios.

Un pixel se representa por: 8 bits (escala de grises), 24 bits (es la común para el color) o 48 bits (fotografías de alta calidad).

### **2.3.7. Relación de aspecto**

Se refiere a la relación que existe entre la anchura y la altura del vídeo. La importancia en la relación de aspecto radica en la deformación de la imagen. Si una pantalla tiene una relación de aspecto diferente a la relación de aspecto de un vídeo, al escalar el vídeo se presentará la deformación en la imagen.

### **2.3.8. Compresión**

Este es un procedimiento que solo puede ser aplicado a video digital. La sucesión de imágenes que conforman un vídeo contienen gran cantidad de información redundante en tiempo, espacio y espectro. La redundancia espacial se presenta en un *frame* y el realizar este tipo de compresión se le llama intraframe y corresponde a la compresión de imágenes. Por otro lado, la redundancia temporal se dá cuando en una sucesión de imágenes existen varios puntos en los que la información, la imagen siguiente corresponde a la de la imagen anterior.

En general la compresión corresponde una pérdida en la calidad del vídeo.

### **2.3.9. Tasa de bits**

Es la cantidad de información en un flujo de vídeo en un período determinado de tiempo. Se mide en bits por segundo (bps). Mientras mayor sea la tasa de bits, mejor será la calidad del vídeo.

## **2.4. Formatos multimedia**

Son estructuras de contenido de archivos orientados a almacenar en una estructura ordenada imágenes, audio, entre otros elementos.

### **2.4.1. Flash video (FLV)**

Es un formato contenedor para transmitir video. En un inicio fue propiedad intelectual de Macromedia. Luego fue adquirido por Adobe Systems en 2005.

Existen variedad de sitios en internet que emplean este formato, como por ejemplo YouTube. Existen variedad de métodos para su reproducción, tanto por vía web como aplicación de escritorio. El método más difundido para la reproducción en web es la incrustación dentro de un archivo SWF. Este es para la reproducción de escritorio podemos utilizar cualquier reproductor que se apoye en DirectShow.

La codificación del video está regularmente en Sorenson Spark, mientras que el audio está regularmente en MP3.

Algunas de las formas de distribución estos archivos son:

- FLV autónomo
- Incrustado en un archivo SWF
- Descarga progresiva por HTTP
- *Streaming* vía RTMP

#### **2.4.2. Formato contenedor**

Este tipo de formatos almacenan información de video, audio, subtítulos, capítulos, metadatos y alguna otra información de sincronización siguiente un formato preestablecido.

Dentro de un archivo contenedor se encuentran diferentes *streams*, los cuales están almacenados de forma segmentada y alternados entre sí. Cada segmento indica el tiempo en el que debe aparecer para realizar la sincronización de los diferentes *streams*. Cada *stream* corresponde a un tipo de información definida en el párrafo anterior.

Es necesario que cada *stream* cuente con una codificación propia como por ejemplo H.264 para vídeo o MP3 para el audio.

### **2.4.3. Small Web Format (SWF)**

En un inicio era conocido como Shockwave Flash. Sin embargo, para evitar confusión con un complemento para navegadores llamado Shockwave, se modificó su nombre a Small Web Format.

La función principal de estos archivos es la creación de pequeños programas capaz de interactuar con otros componentes o con el usuario y al mismo tiempo brindar compatibilidad en diversas plataformas considerando anchos de banda bajos. Actualmente se conoce que el reproductor de estos archivos está presente en el 98 % de los equipos pertenecientes a los internautas.

Los archivos SWF pueden ser creados en el programa Adobe Flash y para ser ejecutados se puede utilizar Adobe Flash Player e incluso pueden ser encapsulados para ejecutarse de forma autónoma.

### **2.4.4. Sorenson codec**

Es conocido también como Sorenson H.263 o FLV1, pero es importante no confundirlo con el formato contenedor FLV.

Es un códec ideado por Sorenson Media Inc. y en muchas ocasiones es conocido como una implementación incompleta del estándar H.263. Este describe un códec que inicialmente se orientó a videoconferencias, optimizado

para una tasa de bits baja y un movimiento reducido. Este estándar fue desarrollado por la Unión Internacional de Comunicaciones.

El estándar H.263 soporta un conjunto limitado de resolución de video:

- 128x96
- 176x144
- 352x288
- 704x576
- 1048x1152

#### **2.4.5. MPEG-2 Audio Layer 3**

Generalmente es conocido como MP3 y fue desarrollado por el grupo Moving Picture Experts Group (MPEG). Este es un formato de compresión de audio digital. El algoritmo que emplea para realizar la compresión acepta pérdida de información para lograr la máxima reducción de tamaño.

Un *stream* MP3 está formada por varias tramas. Cada trama es independiente y cuenta con una cabecera propia, que empieza con una palabra de sincronismo para identificar el inicio. Le sigue una serie de bits que lo identifican como perteneciente al estándar MPEG. Por último la indicación si usa o no la capa 3, por tanto cada trama puede ser reproducida individualmente.

Los archivos que corresponden al estándar MPEG-2 utilizan una de las siguientes velocidades de muestreo:

- 16 kHz

- 22,05 kHz
- 24 kHz

#### **2.4.6. Real Time Message Protocol**

Es un protocolo para el *streaming* de audio, vídeo e información en internet, entre un reproductor y un servidor. Este protocolo está basado en TCP y utiliza el puerto 1935. Sin embargo, existen algunas variaciones:

- RTMPS: es RTMP sobre una conexión TLS/SSL
- RTMPE: es RTMP encriptado utilizando un mecanismo de seguridad propiedad de Adobe Systems.
- RTMPT: utiliza un encapsulamiento por HTTP para pasar los *firewalls*, utilizando el puerto 80.

Para el envío del *streaming* este protocolo de forma predeterminada lo divide en fragmentos de 64 bytes para audio, y 128 bytes para vídeo y otra información. Incluye 1 byte por fragmento como parte de su cabecera.

#### **2.4.7. Paquete RTMP**

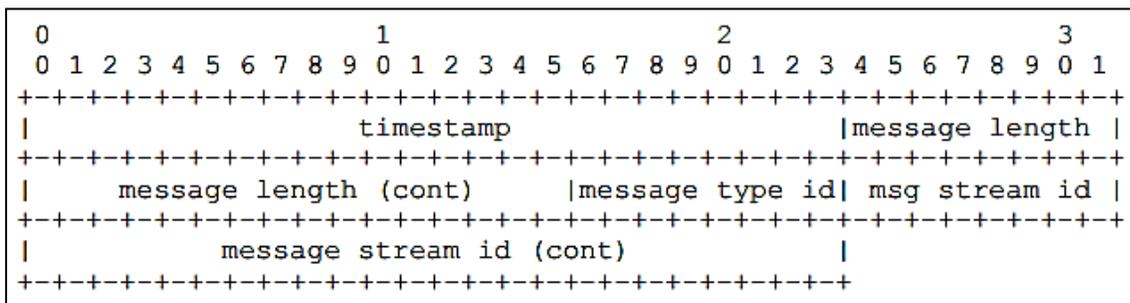
El diagrama del paquete RTMP se muestra en la figura 2. El FMT es el único conjunto de *bytes* constante en todos los paquetes.

El paquete se distribuye de la siguiente forma:

- FMT: en 2 bits identifica los cuatro tipos de *stream*.
- Tipo 0: la cabecera tiene una longitud de 11 bytes.

- Tipo 1: la cabecera tiene una longitud de 7 bytes. El *stream id* no se incluye. La información faltante debe ser obtenida de otros paquetes.
- Tipo 2: la cabecera tiene una longitud de 3 bytes. Este debe ser utilizado para *streams* con mensajes de tamaño constante, por ejemplo, para audio e información. La información faltante debe ser obtenida de otros paquetes.

Figura 2. Trama de paquete RTMP Tipo 0/1

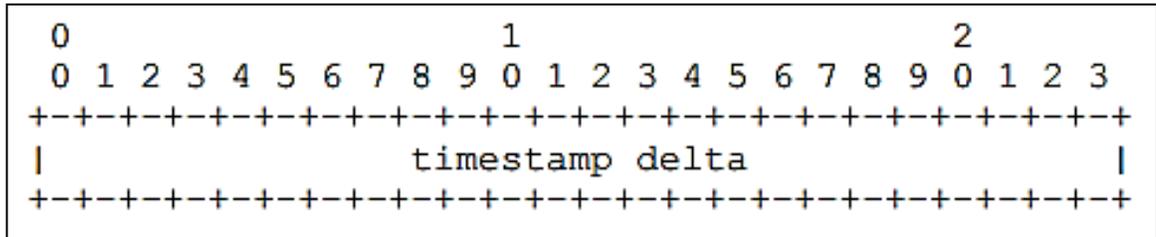


Fuente: *Especificación de RTMP 1.0*

[http://wwwimages.adobe.com/www.adobe.com/content/dam/Adobe/en/devnet/rtmp/pdf/rtmp\\_specification\\_1.0.pdf](http://wwwimages.adobe.com/www.adobe.com/content/dam/Adobe/en/devnet/rtmp/pdf/rtmp_specification_1.0.pdf). Consulta: abril de 2015.

- Tipo 3: no tiene *header*. La información faltante debe ser obtenida de otros paquetes.

Figura 3. Trama de paquete RTMP Tipo 2

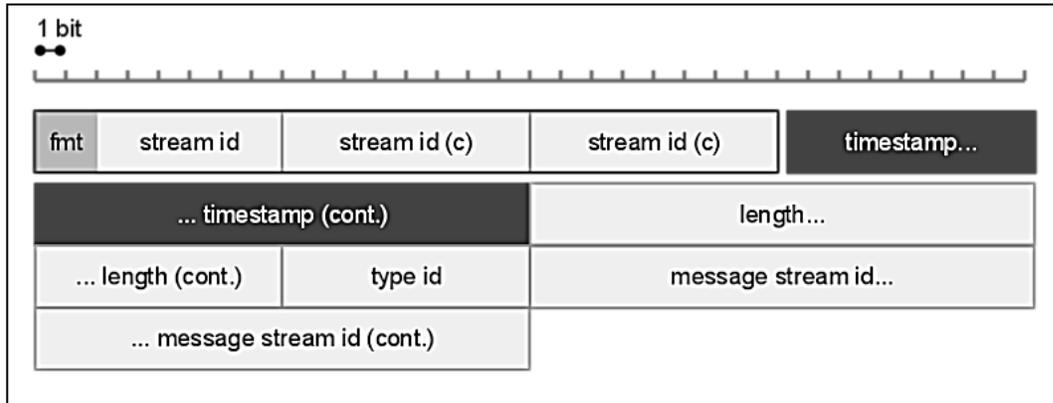


Fuente: *Especificación de RTMP 1.0*

[http://www.images.adobe.com/www.adobe.com/content/dam/Adobe/en/devnet/rtmp/pdf/rtmp\\_specification\\_1.0.pdf](http://www.images.adobe.com/www.adobe.com/content/dam/Adobe/en/devnet/rtmp/pdf/rtmp_specification_1.0.pdf). Consulta junio de 2015.

- Los valores para el *stream id* se pueden codificar:
  - 2 - 63 en 1 byte
  - 64 - 319 en 2 bytes
  - 64 - 65599 en 3 bytes
- *Timestamp*: para los tipos de *FMT* 1 y 2, es la diferencia de tiempo entre el anterior trozo y el actual. El *timestamp* es de 3 bytes y es el tiempo absoluto, siendo el mayor valor 16777215. Si este último valor está presente, indica la presencia de la extensión de *timestamp*, con el cual se puede codificar un valor de hasta 32 bits.
- *Length*: para los tipos de *FMT* 0 y 1, se envía la longitud del mensaje.
- *Type id*: para los tipos de *FMT* 0 y 1, se envía el tipo de mensaje.
- *Message stream id*: para el tipo de *FMT* 0, para los mensajes que en el mismo *chunk stream*.

Figura 4. Trama de paquete RTMP



Fuente: *Especificación de RTMP 1.0.*

[http://wwwimages.adobe.com/www.adobe.com/content/dam/Adobe/en/devnet/rtmp/pdf/rtmp\\_specification\\_1.0.pdf](http://wwwimages.adobe.com/www.adobe.com/content/dam/Adobe/en/devnet/rtmp/pdf/rtmp_specification_1.0.pdf). Consulta junio de 2015.

El protocolo se desarrolla de la siguiente forma:

- Se realiza el *handshake* entre el cliente y el servidor a través del intercambio de tres paquetes de cada lado. Los paquetes utilizados en este paso no corresponden a la estructura de los paquetes de RTMP. El procedimiento es el siguiente:
  - El cliente inicializa la conexión enviando el paquete C0 con la constante 0x03 representando la actual versión del protocolo.
  - El cliente envía el paquete C1 sin esperar recibir S0, el cual contiene 1 536 bytes con los cuatro primeros representando el *epoch timestamp*.
  - El servidor envía el paquete S1.
  - Se envían los paquetes C2 y S2 como un eco de C1 y S1 respectivamente y después de ser recibidos, se considera terminado el *handshake*.

El cliente y servidor negocian la conexión intercambiando mensajes codificados en *Action Message Format (AMF)*.

Iniciar la reproducción del video utilizando los siguientes pasos:

- El cliente envía la invocación *createStream*.
- El cliente envía un *ping*.
- El cliente envía la invocación *play* con el nombre del archivo o *streaming* como argumento.
- El servidor retorna comandos *onStatus*.
- El servidor transfiere la información del video encapsulada en mensajes RTMP.

#### **2.4.8. Red5 Media Server**

Es un servidor de multimedia que pretende ser una solución de código abierto para la transmisión de vídeos a múltiples usuarios. Está basado en el lenguaje de programación Java.

Red5 posee soporte para *NetConnection API*, *NetStream* y *SharedObject* que a la vez proporcionan una potente plataforma para RTMP. Contiene dos herramientas esenciales, el primero *alfaDemo*, el cual se puede utilizar para crear una lista de videos para transmitir con el modelo *OnDemand*. Por otro lado, también cuenta con *live*, utilizado para la transmisión en tiempo real.

#### **2.5. El impacto de la teoría en Guatemala**

Guatemala es un país en constante desarrollo y cambio, que en los últimos años se ha dado la sorpresa de introducirse al mercado tecnológico con

dispositivos como *tablets*. Además ha iniciado la importación de tecnología de punta por lo que es posible adquirir dispositivos de alto rendimiento aunque a un precio elevado. Esto ha permitido poco a poco utilizar el nuevo soporte de redes de comunicación que implementan estos dispositivos, mejorando el tiempo de transmisión física de la información.

### **2.5.1. Internet**

Las telefonías han permitido un amplio acceso a internet con precios económicos y velocidades moderadas. Incluso en los últimos dos años las empresas con iniciativa cuadruplicaron la velocidad que brindan a sus clientes sin incrementar el precio. A pesar de todo este desarrollo, la velocidad de conexión que utilizan los usuarios hacia internet. Por otro lado, las empresas que brindan su servicio en el territorio nacional han abordado satisfactoriamente las tecnologías innovadoras del mundo como por ejemplo redes 3G y 4G. Mejorando así el acceso a internet de todo tipo de dispositivo que se conecte a internet utilizando estos canales.

Según periódicos locales, en el transcurso de diez años desde que inició internet en Guatemala, la cifra de usuarios subió 3 407,7 %. Los resultados obtenidos del estudio de la ITU, para el año 2010 existían aproximadamente 2,48 millones de cibernautas.

Un estudio realizado, por varios sitios web, se determinó que para 2010. Guatemala poseía en promedio la conexión más lenta de todo latino américa, ocupando la posición 149 de 152 países que participaron.

### **2.5.2. Telefonía**

La telefonía en el territorio de Guatemala alcanzó unos exuberantes 20 millones 715 mil usuarios durante 2011. Esto se traduce como un crecimiento del 14,6 % en relación con el año anterior, según boletín estadístico de la Superintendencia de comunicaciones (SIT).

Dado el éxito de la telefonía móvil innovadora, en la cual las empresas le venden un valor agregado a los consumidores, Guatemala es adecuado para ser bombardeado por la industria con servicios de mensajes, videos, imágenes e inclusiones de funciones al celular permitiendo el desarrollo de plataformas móviles, los teléfonos inteligentes. Estos teléfonos inteligentes han dado la posibilidad de concentrar una gran cantidad de medios de comunicación en un solo dispositivo portátil y poderoso.

Con los avances tecnológicos de los últimos años surgen grandes cambios en la forma de ver el mundo y lo que representa innovación. Definitivamente se dice que la innovación más importante de la generación son los teléfonos inteligentes. A ellos se describir en como un ordenador portátil, tan portátil que son posibles cargarlos en el bolsillo. Estos dispositivos están dotados de hardware que los hace altamente usables, como por ejemplo la pantalla táctil, y procesadores extremadamente poderosos de hasta cuatro núcleos.

Estos dispositivos dieron paso a nuevos sistemas de operativos, cada uno con enfoques diferentes y un mercado objetivo creciente, entre ellos Android.

### **2.5.3. Android**

Este sistema operativo basado en Linux, está diseñado principalmente para dispositivos portátiles como teléfonos inteligentes o *tablets*. El primer dispositivo móvil equipado con Android fue el T-Mobile G1 (también conocido como HTC Dream) en octubre de 2008.

Android tiene una naturaleza de código abierto. Por tanto puede ser mejorado con contribuciones de aplicaciones desarrolladas por terceros, llegando a sobrepasar fácilmente las 700 000 aplicaciones, muchas de las cuales son gratuitas. Las aplicaciones que pueden ser instaladas en Android las podemos encontrar en diversidad de tiendas virtual como Google Play, App Store o Samsung Apps.

### **2.5.4. Google Play**

Es la principal tienda de software en línea para aplicaciones de *Android*, desarrollada y dirigida por Google. Esta aplicación le permite a los usuarios la búsqueda, descarga e instalación de aplicaciones de pago o gratuitas.

Los usuarios tienen la posibilidad de ponderar las aplicaciones con una escala de uno a cinco, ingresar comentarios o añadir capturas de pantalla. Esto para ayudar a otros usuarios a seleccionar las mejores opciones.

### **3. IDENTIFICACIÓN DEL PROBLEMA Y SOLUCIÓN QUE LA APLICACIÓN BRINDARÁ**

En este capítulo se pondrá al lector al tanto del problema identificado y a la solución que se pretende dar con el desarrollo del sistema. Se presentan además trabajos de investigación previos, que aunque no se desarrollaron para solucionar el problema identificado, si tienen alguna relación. Además también se aborda la búsqueda de aplicaciones que tiene cierta funcionalidad requerida, para utilizarlos como punto de comparación y un mínimo de resultados esperados.

#### **3.1. Antecedentes**

Cuando se hace referencia a antecedentes sobre investigaciones y aplicaciones que implementen transferencia de video *streaming* se cuenta con un marco histórico que ha llevado a poder, en la actualidad, desarrollar una aplicación que permita tener en un dispositivo móvil con sistema operativo *android* reproducir video en tiempo real. Este desarrollo realizado con el fin de promover la modernización de la figura; cuenta con antecedentes de otros trabajos de graduación que se han desarrollado en la última década y han sido una gran herramienta de apoyo al avance en el conocimiento de cómo realizar una plataforma *streaming* y las aplicaciones que lo implementan.

A continuación se presentará una breve descripción de dichos trabajos de investigación en orden jerárquico de fechas de realización de los trabajos. Estos encontrados en el catálogo de tesis de la biblioteca de la USAC.

- *Sistemas Streaming y sus Aplicaciones*: trabajo de graduación desarrollado por Genser Daniel Mayorga Elías en Guatemala, julio de 2006. En este trabajo de investigación se da una rica explicación acerca de la forma de comunicación *streaming*, se presenta su origen y evolución a través del tiempo. Explica los tipos de comunicación y las tecnologías que los soportan. Muestra varios ejemplos sobre aplicaciones que implementan este tipo de comunicación y la forma en la cual un negocio puede aprovechar la evolución de este.
- *Implementación de la plataforma para streaming de video en tiempo real, a partir de tecnologías libres*, trabajo de graduación desarrollado por Axel Omar Meza Arrecis, noviembre de 2012. En este trabajo de tesis se explica de forma detallada cómo montar una plataforma completa para transmisión de datos *streaming* utilizando únicamente aplicaciones de código abierto para su propósito. En la primera parte del trabajo explica los elementos generales necesarios para iniciar con el montaje de la plataforma de transmisión *streaming*. Luego se explica de forma detallada cómo instalar un servidor de transmisión de datos utilizando como base el servidor red5 y para finalizar se muestra un pequeño ejemplo de transmisión de datos.

Como se pudo observar ambos trabajos de investigación son complementarios. En el primero de ellos se da una introducción al tipo de comunicación *streaming* mostrando sus orígenes y usos principales. En el segundo trabajo, ya se muestran aplicaciones y un ejemplo de transmisión de datos.

Ahora con el crecimiento de las telefonías en Guatemala y el uso de dispositivos móviles como principal medio de comunicación

### **3.2. Descripción del problema**

En la actualidad al formar parte de los estudiantes que integran la Fiusac dificultades, que muchas veces, un estudiante promedio enfrenta al momento de asistir a recibir una cátedra. Siendo estos, por ejemplo, problemas de transporte para asistir en ciertos horarios, ya sea porque el curso es en un horario muy temprano o muy tarde por la noche. En horario que ya está disponible el transporte público colectivo. Otro problema identificado es que algunos cursos se reciben en aulas que no cuentan con el espacio suficiente, para la cantidad de estudiantes que desean asistir a recibir el curso. Por lo que muchos tienen que quedarse de pie fuera del aula, para recibir su curso lo que dificulta el aprendizaje correcto, para cualquier estudiante universitario.

Observando estos problemas en la Fiusac y tomando en cuenta el gran auge que ha tenido la telefonía móvil y el uso de celulares inteligentes (*smartphone*) con sistema operativo Android, se ha decidido crear un sistema que permita a los estudiantes tener acceso a recibir cursos de manera remota en tiempo real desde sus dispositivos móviles. Logrando con esto elaborar una propuesta concreta para solucionar los problemas expuestos con anterioridad.

Con el gran crecimiento del uso de la tecnología y siendo parte de los estudiantes de la Escuela de Ciencias y Sistemas de la Facultad se tiene la gran responsabilidad de proponer soluciones innovadoras que permitan a la Facultad modernizarse y sobre todo seguir preparando cada día más y mejores profesionales.

### **3.3. Software a desarrollar**

La propuesta formal que se tiene para solucionar el problema planteado anteriormente, consiste en desarrollar una aplicación de software que permita a los estudiantes de la Fiusac tener acceso de forma remota y en tiempo real a los cursos impartidos en dicha Facultad. Esto en forma más concreta y la aplicación permitirá a los estudiantes visualizar desde cualquier lugar en donde tengan acceso a internet conectarse a una página, en la cual se encontrarán los *links* de acceso a los cursos que se están impartiendo en el horario establecido.

La solución consiste en crear la aplicación cliente para móviles. El servidor encargado de gestionar los videos en curso y la aplicación que se instalará en los ordenadores de los catedráticos. Esto para transmitir el curso en forma de videoconferencia.

El diseño general de la aplicación será realizado de forma flexible lo que permitirá implementar funcionalidades futuras. Esto es un chat en el cual los estudiantes puedan ingresar dudas al catedrático y este les responda en tiempo real mejorando aún más la interacción con los catedráticos. Otra mejora podría ser la implementación de la pizarra virtual en la que los estudiantes puedan participar de manera más activa de la clase ya que, el catedrático podrá realizar actividades en las cuales se requiera de la participación de todos los estudiantes.

Con esto se tiene como objetivo futuro implementar esta tecnología a nivel Universidad, para que sean aún más los estudiantes beneficiados con la implementación de la tecnología y así promover educación de calidad y objetiva y expandir el prestigio de la USAC.

### **3.4. Mercado objetivo**

Es cada miembro estudiantil de la Fiusac, la cual es una de las unidades académicas con mayor población de la Universidad. La sede de esta Facultad se encuentra en la ciudad universitaria, zona 12 de la ciudad de Guatemala. Se utilizan aproximadamente siete edificios para impartir las cátedras. Se imparten ocho grados, ocho posgrados y catorce maestrías.

La población estudiantil de la Facultad está alrededor de 12 000 estudiantes de un total 99 000 estudiantes en la Universidad, llevándose esta facultad el primer lugar. Estos estudiantes serán los mayormente beneficiados del presente proyecto, con un único requerimiento de contar con un teléfono inteligente que funcione con el sistema operativo Android y conexión a internet. La aplicación estará expuesta al público, por lo que los estudiantes la podrán descargar, instalar y utilizar fácilmente. Además no tendrá costo alguno por su adquisición o uso.

USAC es un representante de su pueblo, por tanto, se espera incrementar la calidad de las cátedras impartidas en los centros de la misma, para que en el futuro estas implementaciones se pueda realizar fácilmente en cualquier otra institución educativa.

### **3.5. Benchmark**

La tecnología multimedia tiene ya varios de existencia y repetidas ocasiones ha sido atacado el problema de las reuniones entre grupos sociales a larga distancia mediante el uso de videoconferencias.

Algunas de las opciones que se detallan a continuación. En cierta medida tienen relación con la aplicación que se pretende desarrollar, pero que de una u otra forma no abarca integralmente todo los requerimientos. Cabe mencionar que a pesar de las similitudes que las aplicaciones tengan, no son iguales a nuestra propuesta, pues esta solución se diseñó para el problema específico de la Fiusac.

### **3.5.1. Cisco WebEx reuniones para Android**

Permite a los participantes asistir, programar o iniciar las reuniones desde su dispositivo Android. En las reuniones se puede apreciar la difusión de datos simultáneos, teleconferencia y audio. Las principales características de esta aplicación son:

- Videoconferencia móvil:
  - 2 vías de alta calidad, multipunto de vídeo con intercambiador de cámara.
  - Conmutación de video activada por voz
  - Pantalla completa de video
  - Visualización de contenido y video de forma simultánea.
  
- Capacidades de la cuenta
  - Planificar, iniciar y cancelar una reunión
  - Invitar a otras personas a una reunión
  - Trasladar las capacidades del presentador a otro participante

- Capacidades de reuniones
  - Unirse a conferencia por los siguientes medios: invitación por correo electrónico, identificador de reunión, dirección web, listas de reuniones o pantalla inicial de Android.
  - VoIP integrada de audio y conferencias web en wi-fi y 3G/4G.
  - Ver lista de asistentes.
  - Ver el contenido compartido con anotaciones.
  - Privado o chat en grupo.

### **3.5.2. GoToMeeting**

Permite mantener un número ilimitado de reuniones con un máximo de veinticinco participantes. Iniciar la reunión y compartir la pantalla con tan solo un clic. Las características de esta aplicación son las siguientes:

- Trabajar cara a cara con alta definición
- Conferencia a través de teléfono o micrófono
- Identificación de hablante.
- Compartir pantalla completa o una aplicación específica
- Grabar las sesiones
- Compartir control de teclado y del ratón para editar en forma cooperativa
- Cambiar al instante presentadores
- Dibujar y hacer marcas en la pantalla

### 3.5.3. OpenMeetings

Con esta aplicación se presenta una alternativa de código abierto la cual permite tener acceso a videoconferencias, mensajes instantáneos y un manejo fácil de un escritorio virtual.

Esta aplicación es desarrollada por la organización Apache y está orientada a la comunidad que gusta de aplicaciones de calidad y con disponibilidad para modificar el código fuente para adaptarla al gusto personal. Las principales características ofrecidas por esta aplicación son:

- Audio y videoconferencia: la aplicación permite hacer uso de esta característica en cuatro modalidades diferentes: audio y video, solamente audio, solamente video y solamente imágenes.
- Grabación de reuniones y compartir pantalla: permite grabar tanto el audio como el video de las conferencias de todas las personas que participan de la reunión de forma idéntica a como se visualiza en la conferencia real.
  - El archivo generado es un archivo en formato AVI o FLV.
  - Posibilidad de seleccionar el área de la pantalla a compartir en la conferencia.
  - Explorador para organizar las grabaciones realizadas con facilidad de *drag and drop*.
- Explorador de archivos: cada salón de conferencia posee un explorador de archivos de fácil uso el cual permite cargar y descargar de forma fácil archivos para que puedan ser visualizados por los demás participantes.

- Sistema para moderación: durante la conferencia aparece un menú para el usuario moderador de la conferencia que permite: agregar y denegar usuarios a la conferencia, admitir y denegar compartir pantallas, entre otros.



## **4. DISEÑO DE LA APLICACIÓN Y SOLUCIÓN QUE LA APLICACIÓN REALIZARÁ**

En los capítulos anteriores se ha identificado una necesidad en la sociedad, el contar con un medio alternativo de comunicación para recibir cátedras, sin tener que estar presencialmente en un salón de clases. Este capítulo muestra todos los componentes en que se divide el sistema, sirviendo como una guía mínima para el usuario final. También puede ser de utilidad para el programador, proporcionando la lógica del negocio del proyecto.

### **4.1. Resumen de la aplicación**

ECO es un proyecto integral que aporta tres programas para realizar las tutorías virtuales, desde el enfoque de gestión de cursos. Para atacar el enfoque de transmisión de cátedra es necesario hacer partícipe a una aplicación adicional, el servidor multimedia. El sistema se desarrolló y probó utilizando red5, por tanto, es la aplicación recomendada para la implementación. Considerando que el servidor de multimedia podría verse como parte del servidor Eco.aula, es recomendable que estos dos estén físicamente en el mismo servidor.

Se definen tres aplicaciones las cuales tienen un grupo objetivo de usuarios independiente: para Eco.Profeso: serán todos los profesores que imparten su cátedra. Eco.Estudiante: serán todos los estudiantes que deseen conectarse a un curso; y por último Eco.Aula: la cual está dirigida a los administradores del sistema y además a fungir como intermediario entre el profesor y el estudiante configurando la comunicación entre ellos.

Por tanto, en Eco.profesor y Eco.estudiante es necesario mantener la validación de conexión al servidor, de tal modo que ante alguna desconexión del servidor el usuario sea redireccionado fuera de la aplicación informando del inconveniente.

#### **4.2. Proceso básico del proyecto**

El proceso básico del proyecto consta de cinco principales subprocesos, los cuales son:

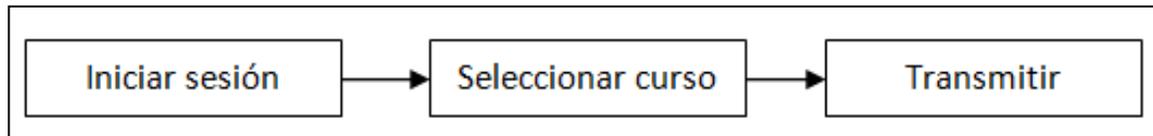
- Gestión de cursos
- Gestión de usuarios
- Inicio de sesión
- Selección de curso
- Transmisión o recepción de video

En los siguientes apartados se darán a conocer los elementos del sistema, con los cuales el usuario puede interactuar a fin de dar vida a los subprocesos anteriormente listados.

#### **4.3. Eco.Profesor**

Esta aplicación es la interfaz que el profesor utilizará para distribuir su cátedra a través del sistema. Esto significa que en la teoría de la comunicación, esta aplicación adopta el papel de emisor.

Figura 5. **Flujo de la aplicación**



Fuente: elaboración propia.

En las *interfaces* que requieren de una autenticación se mostrará un menú con ciertas utilidades para el usuario.

#### **4.3.1. Menú**

Permite crear una estructura ordenada de funcionalidades adicionales sin sobrecargar de información las ventanas.

- Cerrar sesión: termina la sesión activa del usuario, finalizando cualquier transmisión que pudiera estar en curso.
- Salir: cierra la sesión del usuario y cierra la aplicación.
- Acerca de: muestra un diálogo con la información de los desarrolladores, y el nombre y versión del sistema.

En los siguientes apartados se comparte una vista general de los elementos que componen Eco.profesor. Estos brindan la interfaz para el subproceso de transmisión.

### **4.3.2. Iniciar sesión**

Esta ventana se encarga de cubrir dos necesidades básicas que presenta el sistema. La primera, proveer al sistema de autenticación de usuario para configurar la aplicación con los cursos de un profesor en específico; y la segunda, mostrar el enlace que permite a un profesor reiniciar su contraseña, que ya sea por olvido o por hurto, debe ser modificada.

La autenticación se realiza por medio del modelo usuario/contraseña. Esto quiere decir que cada profesor cuenta con un identificador de usuario único en el sistema y una contraseña privada que le permite el acceso solamente a la persona con el conocimiento de este par de datos.

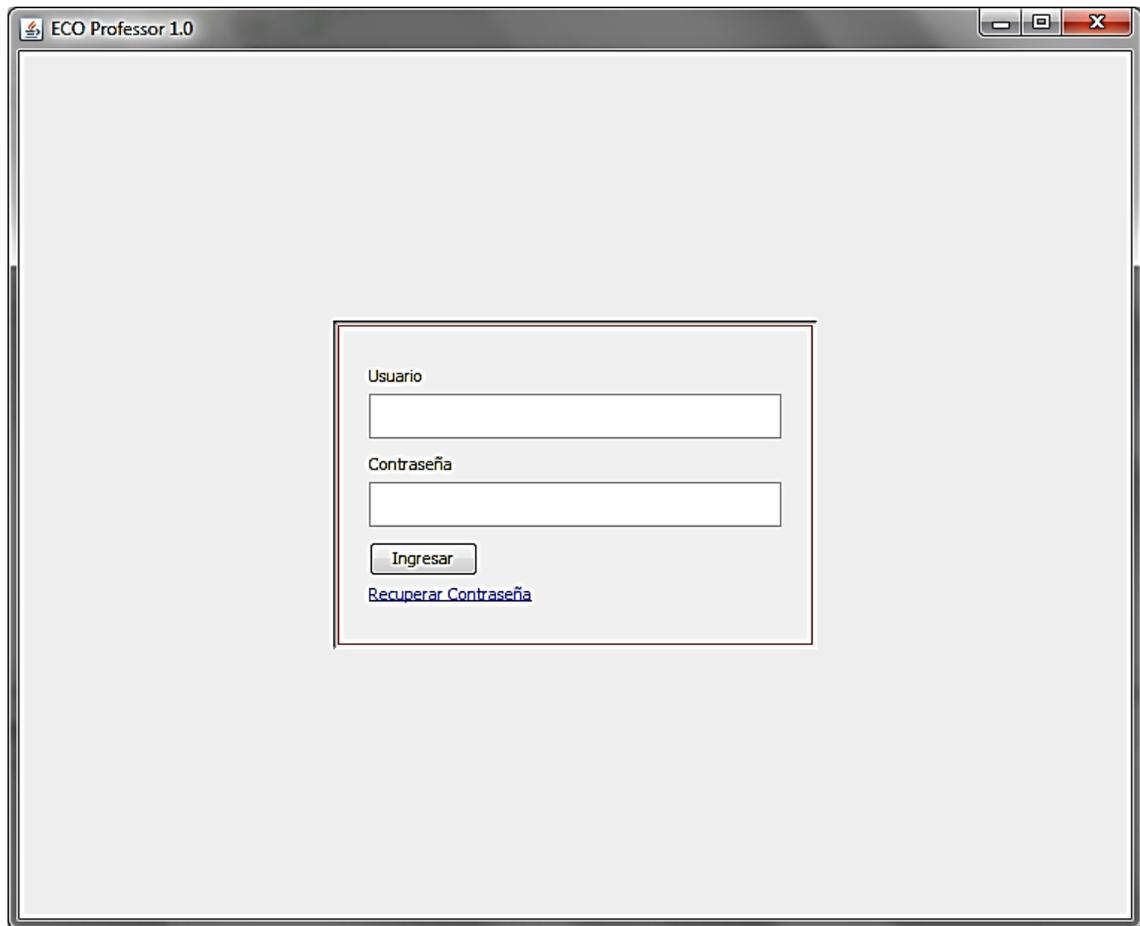
### **4.3.3. Validaciones y errores**

Las validaciones que se hacen para mejorar el desempeño de la aplicación, evitando realizar peticiones que fallarán, son: asegurar que el campo de usuario y contraseña no esten vacios.

Por otro lado, el mínimo de errores que se contemplan en este punto son:

- Error en la conexión al servidor: este error es lanzado cuando la dirección de Eco.aula configurada no responde ante las peticiones.
- Usuario no existe: el usuario utilizado para realizar la autenticación no se encuentra en la base de datos de usuarios.
- Contraseña incorrecta: la contraseña envía para la autenticación no corresponde a la contraseña almacenada en la base de datos.

Figura 6. Inicio de sesión



Fuente: elaboración propia, empleando Adobe Photoshop.

#### 4.3.4. Listado de cursos

Esta segunda ventana se considera la pantalla de bienvenida de la aplicación. Mostrando el listado de cursos a los que un determinado catedrático, identificado por la autenticación, está asignado.

Los cursos se muestran como lista hacia abajo, mostrando información general sobre el curso. Esto es: nombre del curso, cantidad de usuarios

suscritos, cantidad de usuarios conectados. Al pulsar clic sobre un curso, el programa abrirá la ventana de vídeo correspondiente al curso seleccionado.

Considerando que en algún momento un curso puede ser desactivada desde la aplicación administrativa, únicamente se debe mostrar aquellos cursos que aún permanecen activos.

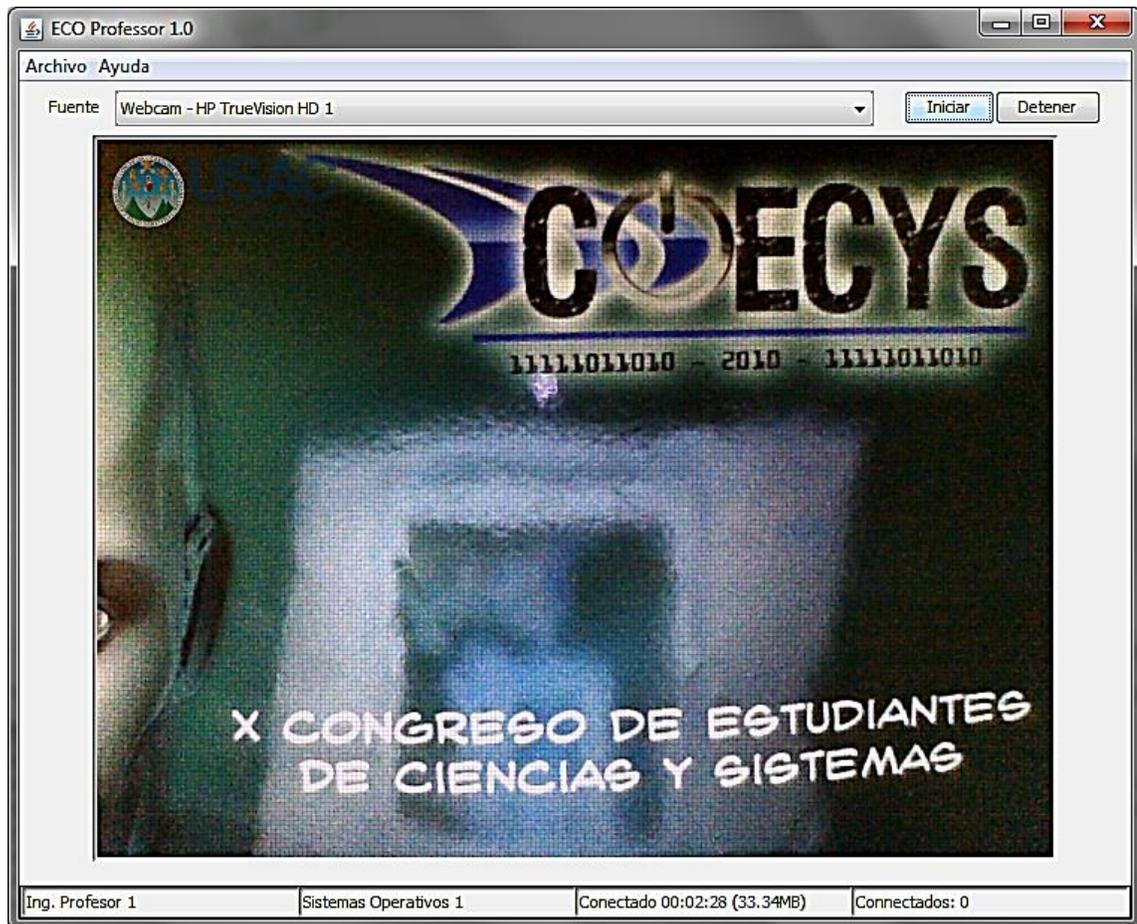
#### **4.3.5. Ventana de vídeo**

En esta última ventana es necesario que el profesor realice un mínimo de configuraciones antes de dar inicio a su cátedra. Es necesario que el profesor seleccione la fuente de video que será transmitida. Las opciones disponibles son: *Escritorio - Pantalla completo*, con la cual comparte la imagen de su escritorio; cámaras web, que agrega cada cámara al listado de fuentes.

Independientemente de la fuente de vídeo seleccionada, el audio que ingresa por el micrófono del ordenador siempre será transmitido. Mientras la transmisión se esté llevando a cabo, se actualizará cada cinco minutos la cantidad de estudiantes conectados.

Una vez que tenga seleccionada la fuente de información deberá hacer clic en el botón iniciar para comenzar la transmisión.

Figura 7. Visualización de video



Fuente: elaboración propia, a partir de la aplicación desarrollada.

#### 4.4. Eco.Aula

Esta aplicación será empleado por tres actores principales: administradores, los cuales trabajan directamente sobre la aplicación web. Esta rinda una interfaz para realizar tareas tales como creación de cursos, creación de usuarios, entre otros; Eco.profesor, accede mediante servicios web a los listados de cursos y al inicio de la transmisión. Por último Eco.estudiante, la

cual también utilizará los mismos servicios web para acceder a la información que requiere.

En su rol como administrador, todas las páginas que requieren una sesión activa se muestra un menú que permite el acceso a las demás páginas.

A continuación se muestran las pantallas con las que cuenta esta aplicación en su perspectiva administrativa.

#### **4.4.1. Iniciar sesión**

Esta interfaz brinda la misma funcionalidad de inicio de sesión del programa Eco.profesor. Además asegura que únicamente inicie sesión el usuario con perfil de administrador.

#### **4.4.2. Configurar cursos**

Se utiliza para la creación y actualización de ciclos académicos, cursos y secciones. Estos elementos son esenciales para la creación de asignaturas que funcionarán como canales de comunicación entre el profesor y el estudiante.

#### **4.4.3. Administrar cursos**

Muestra un listado de los cursos que se encuentran activos en el actual ciclo académico y además permite la edición de ciertas características correspondientes a los mismos. Además permite la creación de nuevas asignaturas, las cuales constan de un curso con una sección y correspondiente a un ciclo académico.

#### **4.4.4. Importar cursos**

Permite la importación de cursos de un ciclo académico anterior. Utilizando este método es posible utilizar el administrador de cursos para la creación inicial de las asignaturas y en los ciclos académicos futuros solamente ir realizando las modificaciones de asignación de catedrático que trae consigo el nuevo ciclo.

#### **4.4.5. Administrar usuarios**

Es el acceso a la gestión de usuarios. Los usuarios de las tres aplicaciones únicamente pueden ser creados desde la administración. Esto quiere decir que independientemente del perfil que el usuario maneje (profesor, estudiante o administrador), este deberá ser creado desde esta interfaz.

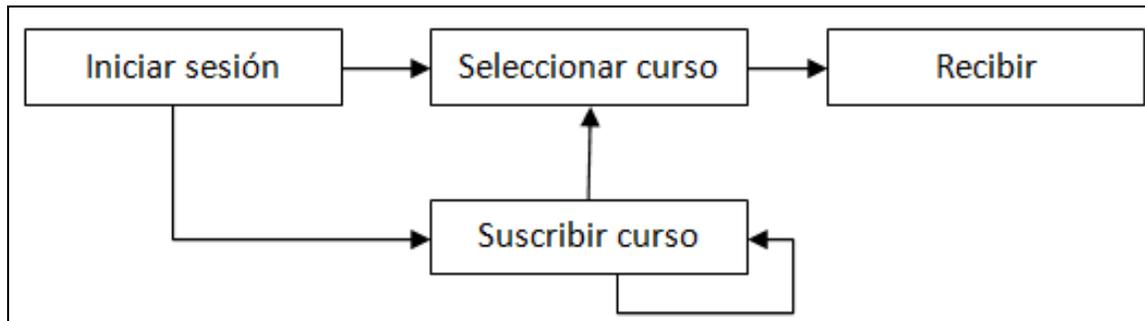
Esta interfaz permite las operaciones de creación y actualización. La eliminación se aborda desde el enfoque de desactivación de usuario.

#### **4.5. Eco.Estudiante**

Esta aplicación funge el papel de receptor en la teoría de comunicación. En esta aplicación acceden de forma exclusiva los estudiantes.

El flujo de la aplicación es como se muestra en la figura.

Figura 8. Diagrama de flujo de estudiante



Fuente: elaboración propia.

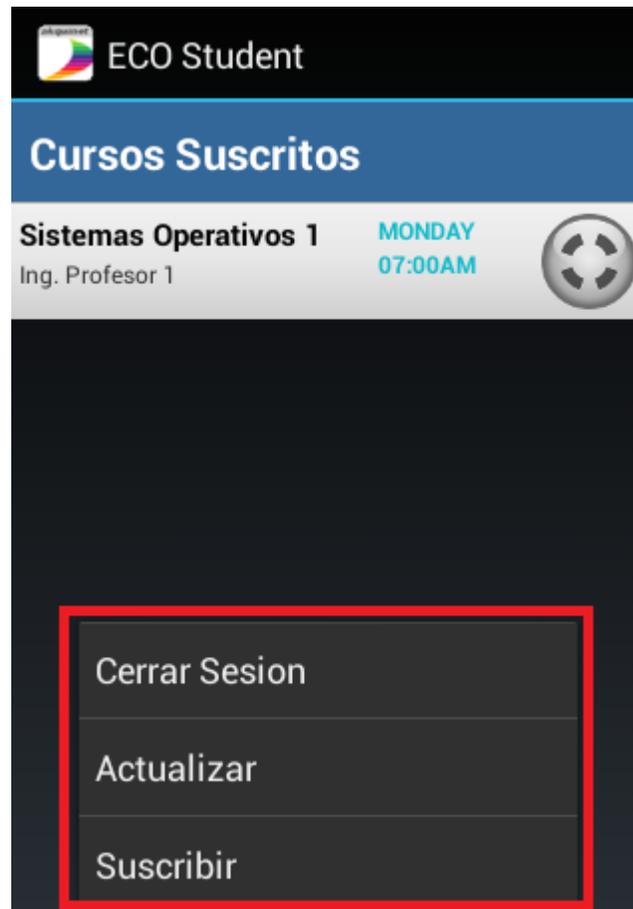
#### 4.5.1. Menú

En este programa se contemplan dos diferentes tipos de menú dependiendo de la interfaz en la que se encuentre el sistema. A continuación se listan las opciones que aparecen en los menús.

- Cerrar sesión: termina la sesión activa del usuario.
- Actualizar: refresca la información en pantalla, realizando una nueva consulta al servidor.
- Suscribir: ingresa a la ventana de suscripción a un curso activo.
- Preferencias: ingresa a la ventana para personalizar las preferencias de la aplicación.
- Buscar: activa el diálogo de búsqueda para suscripción de cursos.

En la figura se muestra de ejemplo el menú de la pantalla de principal, la cual es el listado de los cursos suscritos.

Figura 9. Listado de cursos en estudiante



Fuente: elaboración propia, empelando ECO Student.

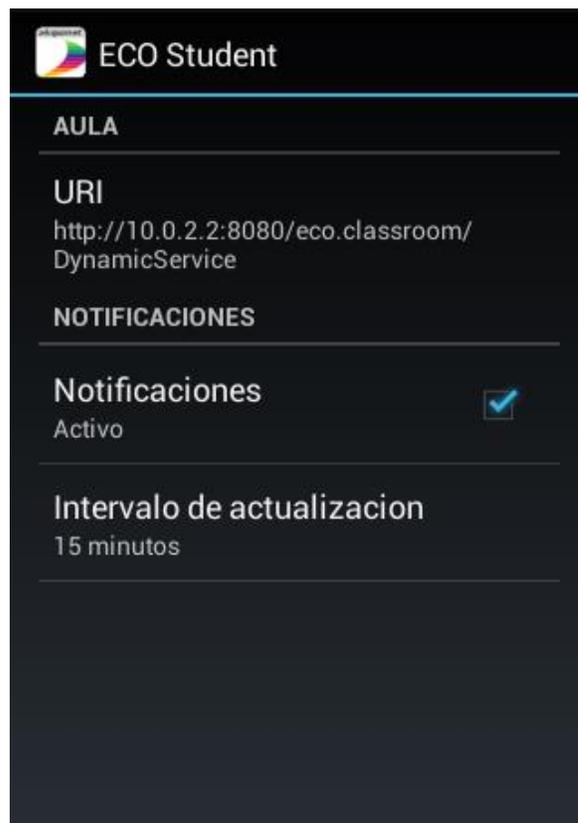
A continuación se muestran las *interfaces* que componen Eco.estudiante.

#### 4.5.2. Preferencias

En esta interfaz se le permite al usuario modificar parámetros de la aplicación. Los parámetros son:

- Aula
  - URI: esta es la dirección del servicio web del servidor aula al que se debe conectar.
- Notificaciones
  - Notificaciones: indica si el servicio de monitoreo de cursos activos debe permanecer en ejecución.
  - Intervalo de actualización: indica el tiempo entre peticiones que debe transcurrir al realizar las consultas de los cursos activos.

Figura 10. **Preferencias de estudiante**

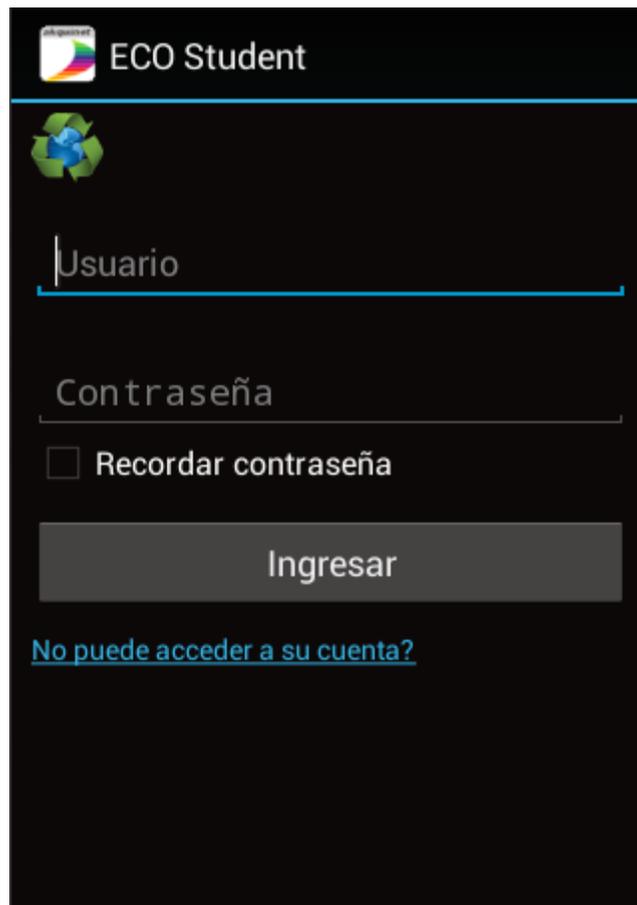


Fuente: elaboración propia, empleando ECO Student.

### 4.5.3. Iniciar sesión

Al igual que el inicio de sesión de Eco.profesor y Eco.aula, esta provee un método de autenticación y una referencia para el reinicio de la contraseña. Asegura que únicamente puedan iniciar sesión los usuarios con perfil de estudiante.

Figura 11. Inicio de sesión de estudiante



The image shows a login screen for 'ECO Student'. At the top left is a logo with a colorful square and the text 'ECO Student'. Below the logo is a recycling symbol. There are two input fields: 'Usuario' and 'Contraseña'. Below the password field is a checkbox labeled 'Recordar contraseña'. A large grey button labeled 'Ingresar' is positioned below the checkbox. At the bottom, there is a blue link that says 'No puede acceder a su cuenta?'.

Fuente: elaboración propia, empleando ECO Student.

#### 4.5.4. Cursos suscritos

Suscritos son todos aquellos cursos que el usuario seleccionó en la interfaz de suscribir cursos, independientemente de si realmente está llevando o no el curso. En este listado se muestra como información esencial el nombre del curso, día y hora próxima en la que se impartirá el curso y el estado de conexión del curso. El estado de conexión se muestra mediante un círculo en el lado derecho de cada curso, cuando se encuentra de color gris significa que está desconectado; mientras que si se muestra en color verde, el curso se está transmitiendo.

Al seleccionar un curso, automáticamente se comenzará la recepción y reproducción del vídeo, siempre y cuando esté conectado. En caso contrario se mostrará un mensaje indicando que el curso no se encuentra disponible.

Figura 12. Listado de cursos suscritos

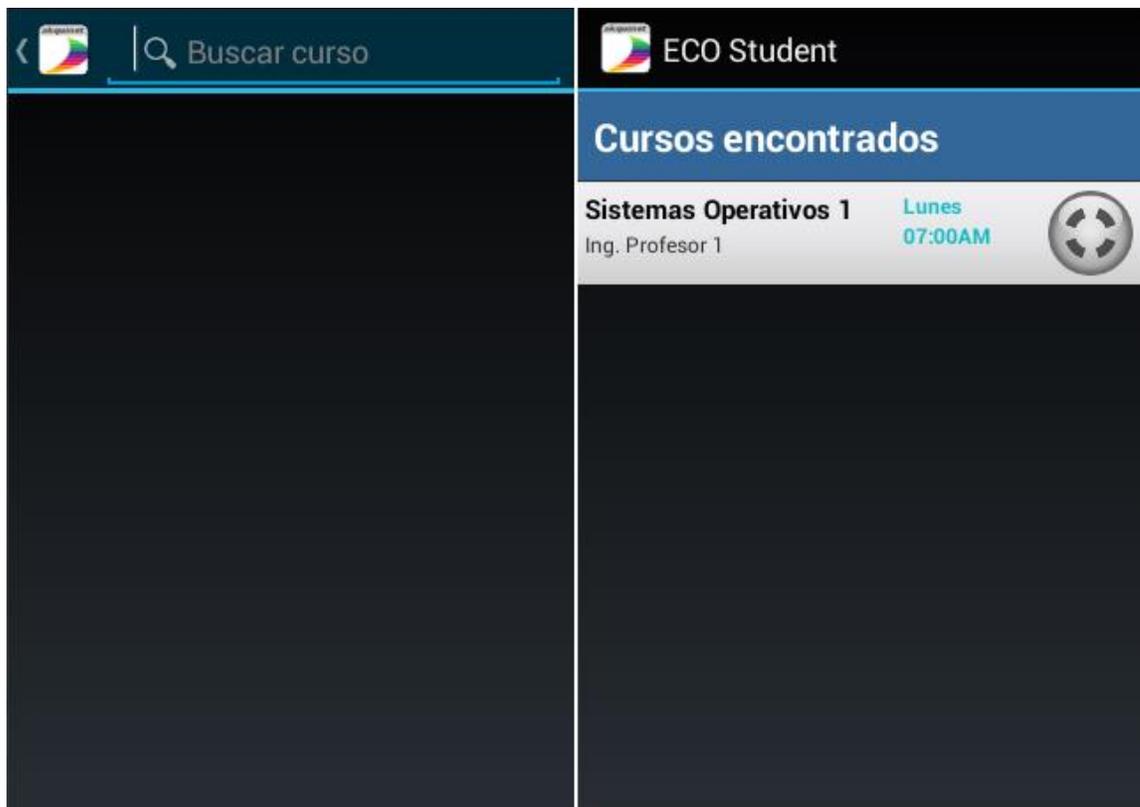


Fuente: elaboración propia, empleando ECO Student.

#### 4.6. Suscribir curso

Esta interfaz provee al usuario con un medio de suscripción a los cursos, facilitando la búsqueda de los mismos. Esto mediante el uso de un filtro por nombre del curso.

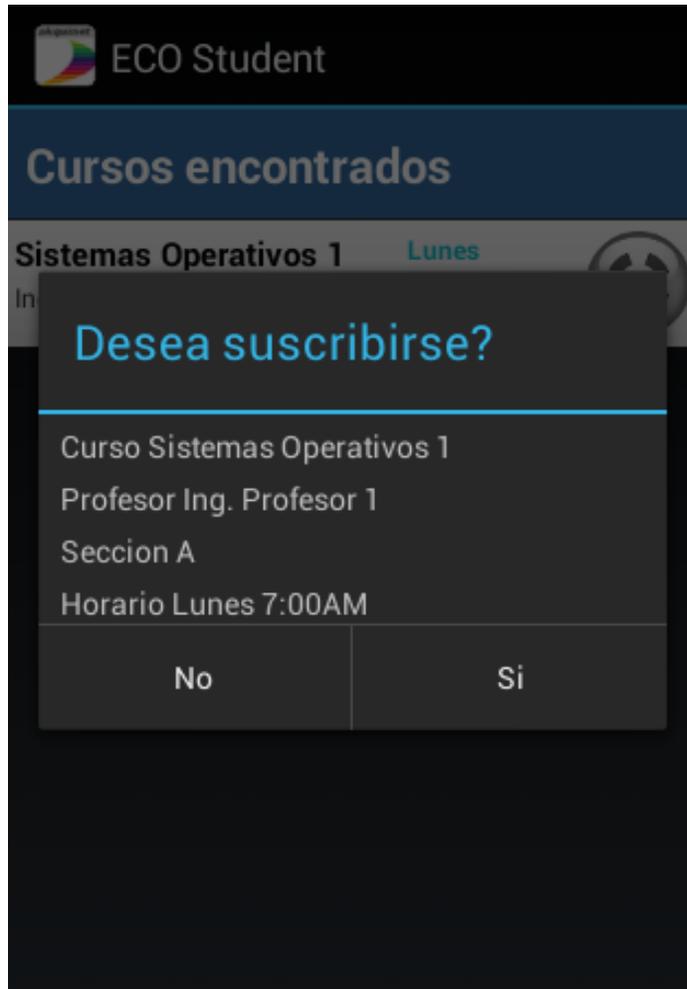
Figura 13. **Buscar cursos**



Fuente: elaboración propia, empleando ECO Student.

Al presionar en un cursos se muestra un mensaje de confirmación con el resumen de la información del curso como lo son nombre, profesor, sección y horario, tal y como se muestra en la figura.

Figura 14. **Confirmación de suscripción**



Fuente: elaboración propia, empleando ECO Student.

Si se presiona “No” el mensaje desaparece, permitiendo al usuario realizar nuevas búsquedas. Por otro lado, si se presiona “Sí”, aparece un mensaje indicando que el curso se ha suscrito exitosamente y en caso contrario muestra un mensaje indicando el error.

#### 4.7. Reproductor

En esta última interfaz es donde el estudiante puede recibir el vídeo transmitido por el profesor. El vídeo ocupa la pantalla completa y únicamente contiene los controles principales de pausar y continuar.

Figura 15. Visualización de cursos



Fuente: elaboración propia, empleando ECO Student.



## **5. DOCUMENTACIÓN Y TUTORIAL DE PROGRAMACIÓN DE LA APLICACIÓN**

En el capítulo anterior se una propuesta de diseño para las tres aplicaciones. En el presente capítulo se detallan todos los recursos necesarios y la descripción del proceso de desarrollo de la aplicación para que sirva como apoyo a todo persona que desee darle continuidad al proyecto. Sin embargo, no se explicará código fuente sino únicamente el funcionamiento.

### **5.1. Requerimientos para desarrollo y pruebas**

En los siguientes apartados se detallan en tres categorías los requerimientos, para el desarrollo del proyecto ECO. Se especifican herramientas, hardware y software que serán necesarias tanto en tiempo de desarrollo como en tiempo de implementación.

### **5.2. Herramientas para el desarrollo**

En este apartado se definen herramientas que, aunque algunas de ellas podrían no ser indispensables para el desarrollo, ciertamente facilitan y optimizan el tiempo de desarrollo. Estas no son necesarias en tiempo de ejecución de la aplicación.

#### **5.2.1. Eclipse**

Provee de un entorno de desarrollo integrado para variedad de lenguajes, entre los cuales se encuentra Java. Eclipse emplea módulos para ampliar su

funcionalidad enriqueciendo la plataforma, entre los cuales provee svn y maven. En general, eclipse es un editor de texto con resaltado de sintaxis.

Sin embargo, bajo ciertas circunstancias puede utilizarse el modo “lo que ves es lo que obtienes”, con lo cual eclipse permite generar interfaces gráficas mediante arrastrar y soltar objetos sobre el formulario. Específicamente esto será muy útil cuando se esté desarrollando Eco.estudiante en Android.

Para utilizar maven desde eclipse es necesario instalar el módulo *m2e*. Con este se podrá utilizar el artefacto *android-quickstart* para generar rápidamente un proyecto con gestión de paquetes mediante maven. Por otro lado para utilizar subversión se apoyará en el complemento subclipse.

Actualmente eclipse se encuentra en su versión de plataforma 4.2 nombrado Juno, lanzado el 27 de junio de 2012.

### **5.2.2. Netbeans**

Es un entorno de desarrollo integrado sobre el cual se puede programar en una amplia variedad de lenguajes, pero el que interesa en esta ocasión es Java. Para expandir su funcionalidad es posible el añadir nuevos módulos. En su última versión contiene de forma predeterminada módulos para el uso de maven y de subversionamiento. Además observando la popularidad de ciertas bibliotecas comerciales como *struts*, se han dado a la tarea de darles soporte.

Entre las características de la plataforma tenemos las siguientes:

- Administración de las interfaces de usuarios como menús y barras de herramientas.

- Administración de configuraciones de usuario.
- Administración del almacenamiento.
- Administración de ventanas.
- Gran variedad de asistentes pasa a paso.

### **5.2.3. Java**

Es un lenguaje de programación creado por la empresa Sun Microsystems en 1995. Este lenguaje está orientado al paradigma de programación orientado a objetos, como una característica peculiar se puede hacer referencia a la óptima forma de compilación realizada en *bytecode*. Este lenguaje permite el uso de programa punteros, como herramientas, juegos y aplicaciones de negocios.

### **5.2.4. Android SDK**

SDK son las siglas de Android Software Development Kit. Es un kit completo para desarrollo con el cual se puede crear aplicaciones que corran en el sistema operativo Android. Este kit de herramientas para desarrollo comprende un depurador de código, biblioteca y un simulador de dispositivo móvil para la prueba de aplicaciones basado en la tecnología de QEMU.

### **5.2.5. Hardware**

Todos los proyecto de software requieren un mínimo de hardware para realizar el desarrollo, pero además de eso, también requieren un mínimo de hardware para realizar las pruebas antes de la implementación.

### **5.2.6. Ordenador**

Es importante contar con un ordenador de escritorio o *laptop*, el cual tendrá dos papeles principales en el desarrollo del proyecto. El primero, como equipo sobre el cual se realizará el desarrollo como tal. Segundo, se utilizará como servidor y ordenador del catedrático al momento de realizar las pruebas de calidad.

Como mínimo es necesario un ordenador con procesador de doble núcleo trabajando a 2,1 GHz, con 2 GB de memoria RAM y 100 MB de espacio físico.

### **5.2.7. Dispositivo con Android**

Es recomendable contar con una amplia variedad, tanto en modelos de dispositivos como en versiones de sistemas operativos android que corren en los mismos. El desempeño de Eco.estudiante es directamente proporcional al CPU y GPU del dispositivo, por tanto, en dispositivos potentes la reproducción del vídeo se apreciará con mayor fluidez.

Entre las diferencias de dispositivos a considerar es necesario no solo diferentes teléfonos inteligentes sino que además *tablets*.

Los dispositivos deben tener un procesador ARMv6 o ARMv7.

### **5.2.8. Servidor**

Este hardware será indispensable al momento de realizar la implementación, ya que será el equipo donde se ejecute Eco.aula y además el servidor de multimedia que se seleccione. Los requerimientos son:

Tabla II. **Requerimientos de hardware para implementación**

Sistema operativo	Windows Server 2000 Windows Server 2003 Variantes de Linux MacOSX 10,4 o mejor
Hardware mínimo	Compatible con X86 3,2GHz o mejor 1GB de RAM disponible 100MB o 1GB Tarjeta ethernet 200MB de espacio en disco duro
Hardware recomendado	Quad Core 2GHz o mejor 4GB o más de RAM disponible 1GB Tarjeta Ethernet 200MB de espacio en disco (10k RPM)
Requerimientos de software	Java JRE 1,5/1,6/1,7

Fuente: elaboración propia.

### 5.3. Software

En esta sección se incluye la descripción y utilidad de componentes de software y en general aplicaciones utilizadas en el desarrollo o pruebas de la aplicación. El software presentado en este apartado será indispensable para el correcto funcionamiento de la aplicación al momento de realizar una implementación.

#### 5.3.1. Webcam-capture

El objetivo de este proyecto es proveer de una API simple para utilizar las cámaras integradas, USB o IP/red directamente desde una aplicación Java. Provee una interfaz básica para cámaras web y un conjunto de controladores

adicionales que pueden ser utilizados para reemplazar los controladores predeterminados y para reemplazar la funcionalidad.

Los requerimientos para este API son:

- Java 6 o posterior
- Cámara conectada al ordenador

Entre las características generales tenemos:

- *Standalone*: no requiere de software para funcionar.
- Independiente del sistema: Utilizable en la mayoría de plataformas para ordenadores (Windows, Linux, Dispositivos ARM, MacOS).
- Detector de movimiento: incluye una rutina para la detección de movimiento en una cámara web.
- Disponible para maven: está disponible en el repositorio central de Maven.
- Gráficas listas: provee de componentes de software para el despliegue en pantalla.

### **5.3.2. Xuggle-Xuggler**

Es una librería para Java de código abierto, que permite a los desarrolladores de software descomprimir, manipular y comprimir vídeos grabados previamente o incluso en tiempo real. Xuggler utiliza el potente manejador de multimedia *ffmpeg*, el cual es una plataforma para grabar, convertir, y transmitir audio y video, soportando una gran variedad de formatos.

Xuggler está compuesto por dos componentes principales, un conjunto de archivos java y un conjunto de librerías nativas compartidas. Por tanto, es necesario tener instalado el paquete *ffmpeg* antes de poder ejecutar un programa que utilice xuggler. *Ffmpeg* se encuentra instalado en todas las plataformas populares por lo que esto no representará ningún problema.

Como requerimiento es necesario estar trabajando con Java 1.5 o posterior.

### 5.3.3. Log4j

Es una biblioteca de código abierto para Java que permite a los desarrolladores de software seleccionar la salida y nivel de granularidad de los mensajes tales como información, advertencia, error fatal, entre otros. Estos mensajes se refieren a los lanzados en tiempo de ejecución y no en tiempo de compilación.

Los niveles de prioridad que maneja para los mensajes son:

- *Fatal*: almacena los errores críticos del sistema.
- *Error*: son errores que a pesar que afectan el sistema, este podrá continuar su ejecución.
- *Warn*: se utiliza para advertencias sobre eventos en los que se considera importante poner en aviso al usuario, pero que sin embargo, no afectan el funcionamiento del sistema.
- *Info*: mostrar mensajes con el estilo *verbose*, el cual consiste en informar al usuario de todas las acciones que realiza el programa.
- *Debug*: escribe mensajes de depuración.
- *Trace*: se utiliza para mensajes con un mayor detalle que *debug*.

Además de estos seis niveles, se consideran adicionalmente lo siguientes:

- *Off*: deshabilita todos los *logs*.
- *All*: es el nivel máximo de detalle, por lo que habilita todos los *logs*.

#### **5.3.4. Zodiac**

Esta biblioteca contiene rutinas para la implementación de los patrones *data access object*, *data transfer object* y *service-oriented architecture*. Permite trabajar con *cookies* para mantener la sesión entre cliente servidor.

Cuenta con componentes para trabajar tanto en *java enterprise edition* como en Android.

#### **5.4. Json-simple**

Es un kit de herramientas para Java que permite trabajar con codificación y decodificación JSON. Algunas de las características son:

- Compatibilidad con la especificación RFC4627
- No depende de librerías externas
- Compatible con Java 1,2 o posterior

#### **5.4.1. Vitamio**

Es una biblioteca que trabaja sobre la plataforma Android para un desarrollo versátil de multimedia y con soporte completo para decodificación de hardware y renderización GPU.

Vitatio permite visualizar vídeos en alta definición de 720p o 1080p en formatos MKV, FLV, MP4, MOV, TS, RMVB y algunos otros formatos comunes. Además permite una variedad de protocolos de comunicación como lo son RTSP, RTMP, MMS y HLS, incluyendo transmisión *onDemand* y en tiempo real. Se apoya en *ffmpeg* para analizar los medios de comunicación.

Además soporta características adicionales como subtítulos externos, múltiples pistas de audio, autoajuste de relación de la pantalla y detección automática de codificación.

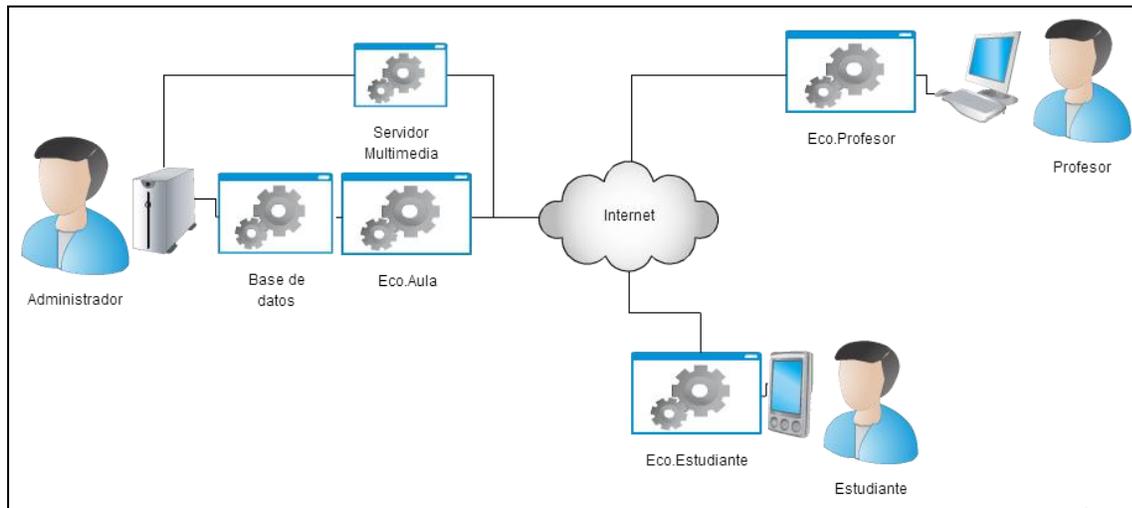
## **5.5. Arquitectura de eco**

El proyecto se divide en tres niveles:

- Profesor: nombrado Eco.Profesor
- Servidor: nombrado Eco.Aula
- Estudiante: nombrado Eco.Estudiante

Un nivel se refiere a una separación física; esto quiere decir que serán ejecutados en ordenadores separados. En la siguiente figura se muestra la forma de comunicación entre las aplicaciones.

Figura 16. Diagrama de arquitectura del software



Fuente: elaboración propia, empleando ECO Student.

## 5.6. Aplicación Eco.Profesor

Es una aplicación de escritorio creada con el lenguaje de programación orientada a objetos Java, creada con la herramienta maven que también proporciona rutinas para la gestión. Esta aplicación está diseñada con el patrón de arquitectura de tres capas modelo, vista y controlador (MVC).

### 5.6.1. Arquitectura de la aplicación

Este apartado explica el patrón de diseño empleado para proveer a la aplicación con la solución a requerimientos no funcionales inherentes a cualquier programa que se desarrolle. Esto es la mantenibilidad, flexibilidad, entre otros.

#### **5.6.1.1. Modelo**

En esta capa se encuentran todas las rutinas utilizadas para el acceder al servicio web proporcionado por Eco.aula. La mensajería entre estas dos aplicaciones es mediante el patrón de diseño *data transfer object* (DTO).

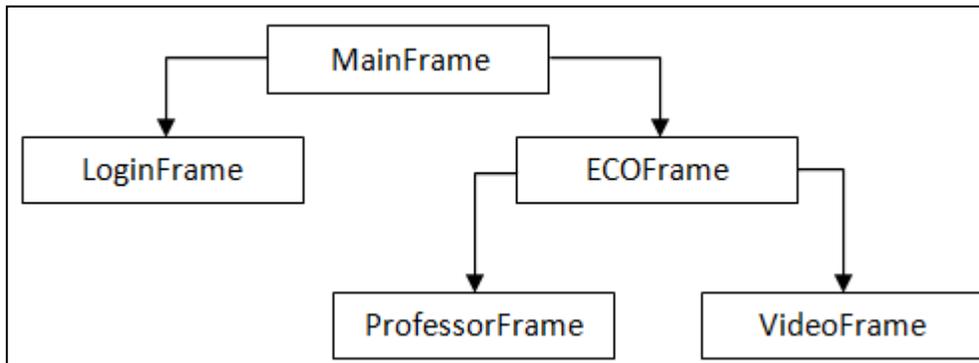
La conexión con el servicio web está implementado con el marco *com.zodiac*, Este permite realizar una conexión solamente con indicar la clase, constructor y método a utilizar en el servidor.

#### **5.6.1.2. Vista**

Se refiere a las interfaces que se utilizan para permitir la interacción del usuario con el sistema, utilizando objetos como cuadros de texto, botones, entre otros, y con ayuda de estilos, tipos de letras y colores para darle un diseño más atractivo.

A continuación se muestra en la figura un árbol de herencias utilizadas en las vistas.

Figura 17. **Jerarquía de vista Eco.profesor**



Fuente: elaboración propia.

### 5.6.1.3. **Controlador**

Es el intermediario entre el modelo y la vista, llevando las solicitudes desde la vista hacia el modelo y la información resultante desde el modelo hacia la vista. De esta forma se logra darle un flujo al programa, pues este representa la lógica del negocio.

En general existe un controlador por cada vista y los nombres corresponden al patrón: nombre de la vista reemplazando la palabra *frame* por *controller*. Además para cada controlador se registran escuchantes de las respuestas con el mismo nombre del controlador pero precedidos por una I.

Además de los controladores previamente descritos, existen otros. Siendo los principales descritos a continuación.

- *Encoder*: contiene las rutinas para la codificación del audio y video en *streams* para ser transmitidos al servidor multimedia.

- *RobotWideScreenDevice*: representa la ventana del escritorio, devolviendo mediante el método *getImage* la imagen correspondiente a la pantalla completa.
- *WebcamDevice*: representa una fuente de información desde un componente de *hardware*, específicamente una cámara web. Contiene el método *getImage* para obtener la imagen actual captada por la cámara.

## **5.7. Aplicación Eco.Aula**

Es una aplicación desarrollada en Java Enterprise Edition, empleando Zodiac para la implementación de patrones de diseño *Data Access Object (DAO)*, *Data Transfer Object (DTO)* y *Service Oriented Architecture (SOA)*. En general los patrones de diseño pretenden una programación limpia, ordenada y con componentes de software reutilizables. Además de ser una funcionalidad centralizada para las dos aplicaciones cliente, también resuelve el enfoque de administrador, pues provee una serie de páginas web a las cuales se les puede acceder desde el navegador y realizar las tareas administrativas.

### **5.7.1. Arquitectura de la aplicación**

El patrón DAO actúa como una interfaz común entre la aplicación y una fuente de información como la base de datos. Además permite, no solamente una base de datos, si no que la implementación de un mismo sistema sobre varios DBMS, logrando que el usuario final escoja la de su preferencia.

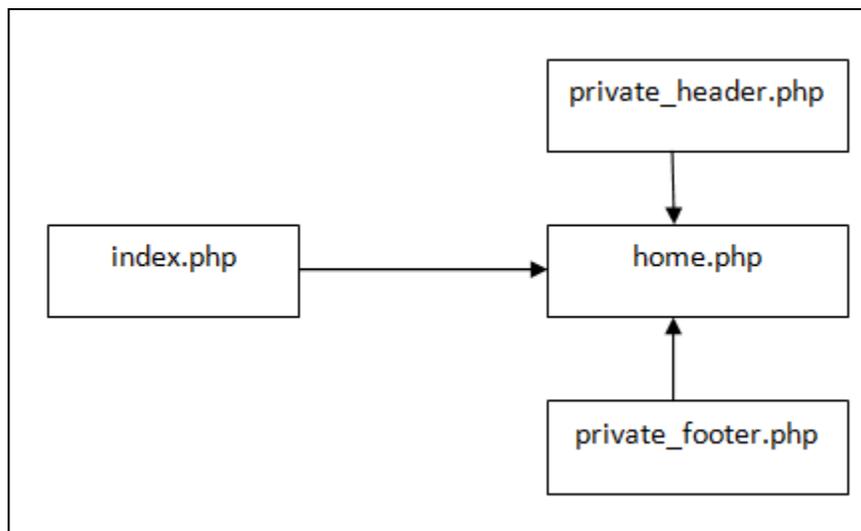
El patrón DTO se utiliza para transmisión de información entre diferentes sistemas y usualmente se utiliza en conjunto con el patrón DAO. Los DTO no tienen ningún comportamiento en especial sino que solamente almacenan información para ser serializada en bytes, xml, entre otros.

El patrón SOA permite la centralización de la lógica del negocio en un solo servidor, de tal forma que todos los subsistemas empleen conexiones hacia un punto central. Por tanto, al realizar un cambio en la lógica de negocio solamente será necesario modificar un lugar. Además provee mayor seguridad al no permitir el acceso directo a la base de datos.

### 5.7.1.1. Páginas web

En la figura se muestra un ejemplo del flujo de inicio de sesión y los componentes de una página protegida por una sesión.

Figura 18. Jerarquía de vista Eco.aula



Fuente: elaboración propia.

En la página index.php se encuentra la configuración de la página de inicio de sesión, desde la cual accede hacia páginas privadas, en el ejemplo llamada home.php. Esta página muestra el menú de acciones que se pueden tomar desde la página administrativa.

### 5.7.1.2. Servicio web

Mediante el servicio web se comparte toda la funcionalidad del servidor hacia las aplicaciones clientes, siente este servicio transparente al realizar las llamadas remotas a las funciones. Las clases con funciones que están disponibles para ser invocadas desde las aplicaciones clientes heredan de *BussinessLogic*. Sin embargo, existe la posibilidad de heredar desde *PrivateBussinessLogic*, la cual forzará a verificar que el usuario haya iniciado sesión para acceder a la función. Para realizar operaciones sobre la sesión, tal como la creación y destrucción es necesario heredar desde *SessionBussinessLogic*.

La lógica de negocio funciona como una primera capa al recibir todas las peticiones a ser procesadas. La segunda se llama capa de acceso a datos. Para la implementación de esta capa es necesario crear interfaces con todas las funciones que se utilizarán para obtener la información, independientemente si se utilizarán para acceder a base de datos, a otro servicio web o incluso a un archivo. Es necesario crear el archivo que implementa esta funcionalidad, la cual convencionalmente tiene el nombre *<nombre de la interfa><nombre del dbms>.java* por lo que es posible implementar el uso de múltiples bases de datos.

## 5.8. Aplicación Eco.Estudiante

Básicamente está desarrollada con la misma arquitectura que la aplicación *Eco.profesor*, con la diferencia que sus vistas, controladores y modelos están destinados a cubrir la participación del estudiante en el proceso.

El lenguaje de programación utilizado es Java. Sin embargo, se utiliza Android SDK para que su desarrollo sea dirigido a dispositivos portátiles que ejecuten *Android* como sistema operativo. La compatibilidad comprende desde la plataforma 8 hasta 17 de Android, que corresponden a *android 2.2.2* hasta *android 4.0*. Además como el reproductor se ejecuta utilizando *vitamio* es necesario contar con un procesador ARM v6 o ARM v7, el cual está presente en la mayoría de dispositivos de este tipo.

### **5.8.1. Arquitectura de la aplicación**

En este apartado se describe la arquitectura empleada en el desarrollo del programa cliente. Con ello, satisface una variedad de requerimientos no funcionales.

#### **5.8.1.1. Vista**

Debido a que está desarrollado en Android, los archivos de esta capa están distribuidos en dos directorios principales: las clases se encuentran en `org.usac.eco.student.model`. Sin embargo, la interfaz está definida en archivos xml dentro de la carpeta `res/layout`.

Las vistas en Android son llamadas Activity y por cada una existe un archivo xml que contiene la información de como se distribuyen los objetos por toda la pantalla. En `PlayerActivity.java` se encuentra el instanciador del reproductor de multimedia por RTMP mediante *vitamio*, enviándole la dirección del multimedia que se desea reproducir.

### **5.8.1.2. Controlador**

Debido a que la comunicación por internet debe ser en un nuevo hilo cuando trabajamos con Android, no es posible la implementación de la última capa llamada acceso a datos. Es por esto que el acceso a datos del servicio web está implementada en la capa de controlador.

Esta capa está compuesta por una serie de controladores que efectúan una operación especial cuando se realiza el llamado a las funciones del servidor Eco.aula, pero en su mayoría no existen tales operaciones si no que solamente se realiza la llamada remota.

### **5.8.1.3. Manifiesto de Android**

Es necesario asegurar ciertos permisos para el correcto funcionamiento de la aplicación. Estos permisos serán solicitados al usuario al momento de realizar la instalación del programa. Los permisos requeridos para Eco.estudiante son: *RECORD\_AUDIO*, *INTERNET* y *WRITE\_EXTERNAL\_STORAGE*. En su mayoría estos privilegios son utilizados por el componente *vitamio* para realizar la reproducción de los videos en tiempo real.

Es necesario registrar todas las interfaces llamadas *activity* colocando un nombre, orientación y un valor. Además es necesario registrar también todos los servicios, en este caso se registra *NotificationService*. Este servicio se encarga de revisar los cursos que están siendo impartidos y generar una notificación al sistema operativo.



## CONCLUSIONES

1. Se elaboró una guía explicando el proceso de desarrollo de la aplicación sobre el sistema operativo Android, con el cual todo estudiante con conocimientos sobre el lenguaje de programación Java podrá extender el alcance o crear un sistema propio.
2. Fomentar el desarrollo de nuevos profesionales a través de la implementación de nueva tecnología en el proceso de enseñanza de la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala.
3. Se determina que al proveer un nuevo canal de comunicación con características similares a la cara a cara, una mayor cantidad de estudiantes podrá asistir a clases mejorando sus conocimientos en el proceso.
4. Brindándole el sistema a la Facultad de Ingeniería, se le brinda la oportunidad de extender su presencia más allá de las locaciones físicas con las que cuenta actualmente, teniendo así un mayor impacto en la sociedad Guatemalteca.



## RECOMENDACIONES

1. Continuar con el proyecto, proveyendo nuevas funcionalidades y lograr así una plataforma integral para el estudiante, lo cual implica la modernización de la Escuela de ciencias y sistemas.
2. Integrar al sistema oficial de la Escuela de Ciencias y Sistemas para automatizar el proceso de apertura, administración y asignación de los cursos, tanto a los catedráticos como a los estudiantes.
3. Desarrollar el proyecto dirigido al estudiante en nuevas plataformas, como por ejemplo en una página web o un sistema iOS, para tener un mayor alcance.



## BIBLIOGRAFÍA

1. *Adobe Systems Incorporated Adobe's Real Time Messaging Protocol.* [en línea] <[http://www.adobe.com/content/dam/Adobe/en/devnet/rtmp/pdf/rtmp\\_specification\\_1.0.pdf](http://www.adobe.com/content/dam/Adobe/en/devnet/rtmp/pdf/rtmp_specification_1.0.pdf)>. [Consulta: mayo de 2015].
2. ALAN R. Dennis; Joseph S. Valacich *Rethinking Media Richness: Towards a Theory of Media Synchronicity.* [en línea] <<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.108.7118&rep=rep1&type=pdf>>. [Consulta: mayo de 2015].
3. Android Developer. *Sitio oficial para desarrolladores de Android.* [en línea]. <<http://developer.android.com/>>. [Consulta: mayo de 2015].
4. Internet Engineering Task Force. *Real Time Streaming Protocol.* [en línea]. <<https://tools.ietf.org/html/rfc2326>>. [Consulta: mayo de 2015].
5. Red5. *Sitio oficial de red5.* [en línea] <<http://www.red5.org/>>. [Consulta: mayo de 2015].
6. Xuggler. [www.benfarahmand.com](http://www.benfarahmand.com). [en línea]. <<http://www.benfarahmand.com/2012/11/tutorial-using-xuggler-in-processing.html>>. [Consulta: mayo de 2015].

7. Yixia Vitamio. [en línea]. <<https://github.com/yixia>>. [Consulta: mayo de 2015].