



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

**DISEÑO DE LA APLICACIÓN PARA NOTIFICACIÓN DE ACTIVIDADES  
ACADÉMICAS A TRAVÉS DE LA MENSAJERÍA INSTANTÁNEA EN LA FACULTAD  
DE ARQUITECTURA, UNIVERSIDAD DE SAN CARLOS DE GUATEMALA**

**Brayan Armando Yaquián González**  
Asesorado por el Ing. Ricardo Mazariegos

Guatemala, febrero de 2018

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA



FACULTAD DE INGENIERÍA

**DISEÑO DE LA APLICACIÓN PARA NOTIFICACIÓN DE ACTIVIDADES  
ACADÉMICAS A TRAVÉS DE LA MENSAJERÍA INSTANTÁNEA EN LA FACULTAD  
DE ARQUITECTURA, UNIVERSIDAD DE SAN CARLOS DE GUATEMALA**

TRABAJO DE GRADUACIÓN

PRESENTADO A LA JUNTA DIRECTIVA DE LA  
FACULTAD DE INGENIERÍA  
POR

**BRAYAN ARMANDO YAQUIÁN GONZÁLEZ**  
ASESORADO POR EL ING. RICARDO MAZARIEGOS

AL CONFERÍRSELE EL TÍTULO DE

**INGENIERO EN CIENCIAS Y SISTEMAS**

GUATEMALA, FEBRERO DE 2018

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA  
FACULTAD DE INGENIERÍA



**NÓMINA DE JUNTA DIRECTIVA**

DECANO	Ing. Pedro Antonio Aguilar Polanco
VOCAL I	Ing. Ángel Roberto Sic García
VOCAL II	Ing. Pablo Christian de León Rodríguez
VOCAL III	Ing. José Milton de León Bran
VOCAL IV	Br. Oscar Humberto Galicia
VOCAL V	Br. Carlos Enrique Gómez Donis
SECRETARIO	Inga. Lesbia Magalí Herrera López

**TRIBUNAL QUE PRACTICÓ EL EXAMEN GENERAL PRIVADO**

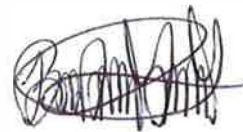
DECANO	Ing. Pedro Antonio Aguilar Polanco
EXAMINADOR	Ing. Edgar Estuardo Santos Sutuj
EXAMINADOR	Ing. Oscar Alejandro Paz Campos
EXAMINADOR	Ing. Everest Darwin Medinilla Rodríguez
SECRETARIO	Ing. Pablo Christian de León Rodríguez a.i.

## **HONORABLE TRIBUNAL EXAMINADOR**

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

### **DISEÑO DE LA APLICACIÓN PARA NOTIFICACIÓN DE ACTIVIDADES ACADÉMICAS A TRAVÉS DE LA MENSAJERÍA INSTANTÁNEA EN LA FACULTAD DE ARQUITECTURA, UNIVERSIDAD DE SAN CARLOS DE GUATEMALA**

Tema que me fuera asignado por la Dirección de la Escuela de Ingeniería en Ciencias y Sistemas con fecha 7 de agosto de 2016.



**Brayan Armando Yaquián González**



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

Guatemala, 18 de Octubre de 2017

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 del estudiante **BRAYAN ARMANDO YAQUIAN GONZALEZ** con carné **201213578** y CUI **2146 76951 0601**, titulado **"DISEÑO DE LA APLICACIÓN PARA NOTIFICACIÓN DE ACTIVIDADES ACADÉMICAS A TRAVÉS DE LA MENSAJERÍA INSTANTÁNEA EN LA FACULTAD DE ARQUITECTURA, UNIVERSIDAD DE SAN CARLOS DE GUATEMALA"** 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



Guatemala 11 de septiembre de 2017.

**Ing. Marlon Pérez Turk**  
Director de la Escuela de Ciencias y Sistemas  
Facultad de Ingeniería USAC.

Respetable Ing. Pérez:

Por medio de la presente hago de su conocimiento que el estudiante **Brayan Armando Yaquian Gonzalez** que se identifica con CUI No. 2146769510601 y código estudiantil No. 201213578 , de la Carrera de Ingeniería en Ciencias y Sistemas, Facultad de Ingeniería, USAC. Culminó satisfactoriamente el contenido de su trabajo de graduación titulado "DISEÑO DE LA APLICACIÓN PARA NOTIFICACIÓN DE ACTIVIDADES ACADÉMICAS A TRAVÉS DE LA MENSAJERÍA INSTANTÁNEA EN LA FACULTAD DE ARQUITECTURA, UNIVERSIDAD DE SAN CARLOS DE GUATEMALA", por lo tanto, considero que se puede proseguir con la siguiente fase del proceso de graduación.

Sin otro particular, me suscribo,

Att.

F: 

**Ing. Ricardo Israel Mazarlegos Castillo, M.B.A.**  
Asesor de tesis

**Ricardo Israel Mazarlegos Castillo**  
Ingeniero en Ciencias y Sistemas  
Colegiado No.: 12451

E  
S  
C  
U  
E  
L  
A  
  
D  
E  
  
I  
N  
G  
E  
N  
I  
E  
R  
I  
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.: 24767644

*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 **"DISEÑO DE LA APLICACIÓN PARA NOTIFICACIÓN DE ACTIVIDADES ACADÉMICAS A TRAVÉS DE LA MENSAJERÍA INSTANTÁNEA EN LA FACULTAD DE ARQUITECTURA, UNIVERSIDAD DE SAN CARLOS DE GUATEMALA"**, realizado por el estudiante **BRAYAN ARMANDO YAQUIÁN GONZÁLEZ**, aprueba el presente trabajo y solicita la autorización del mismo.*

"ID Y ENSEÑAD A TODOS"

  
Ing. Marilyn Antonio Pérez Turck

Director

*Escuela de Ingeniería en Ciencias y Sistemas*



Guatemala, 25 de enero de 2018



Ref.DTG.D.033.2018

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: **DISEÑO DE LA APLICACIÓN PARA NOTIFICACIÓN DE ACTIVIDADES ACADÉMICAS A TRAVÉS DE LA MENSAJERÍA INSTANTÁNEA EN LA FACULTAD DE ARQUITECTURA, UNIVERSIDAD DE SAN CARLOS DE GUATEMALA**, presentado por el estudiante universitario: **Brayan Armando Yaquián González**, y después de haber culminado las revisiones previas bajo la responsabilidad de las instancias correspondientes se autoriza la impresión del mismo.

IMPRÍMASE.

  
Ing. Pedro Antonio Aguilar Polanco  
Decano

Guatemala, enero de 2018



/cc





## **ACTO QUE DEDICO A:**

<b>Dios</b>	Por haber puesto en mi camino a las personas indicadas para llegar hasta aquí.
<b>Mi mamá</b>	Aura Elizabeth González López, porque sin ella nada de esto hubiese sido posible y por ser todo para mí.
<b>Mi abuela</b>	Irma Elena López por haberme enseñado el valor de la vida.
<b>Mi novia</b>	Ana Lucía Corado Sarceño por haber estado junto a mí en todo momento apoyándome y alentándome a seguir adelante.
<b>Mis tíos</b>	Por sus sabios consejos y por guiarme en el sendero de la vida.
<b>Mis primos</b>	Por ser tan buenos conmigo y quererme como a un hermano, especialmente a Luisa Navas y Luz González.



## **AGRADECIMIENTOS A:**

<b>Universidad de San Carlos de Guatemala</b>	Por ser mi casa de estudios que me enseñó a perseverar para lograr mis objetivos.
<b>Facultad de Ingeniería</b>	Por ser una fuente de sabiduría, por llenarme de lecciones que me enseñaron a ser perseverante y valiente.
<b>Ricardo Mazariegos</b>	Por impulsarme hacia la excelencia y por haberme brindado su ayuda incondicional.
<b>Roberto Leal</b>	Por confiar en mí para llevar a cabo este proyecto y brindarme todo su apoyo.
<b>Luz González</b>	Por haber crecido junto a mí y ser como una verdadera hermana.
<b>Francisco González</b>	Por preocuparse por mí, aconsejarme, apoyarme y, además, ser un ejemplo para mí.
<b>Mis amigos de la Facultad</b>	Manuel, Fernando, Ricky, por compartir muchos buenos momentos conmigo.
<b>Mis amigos de la infancia</b>	Iván Osorio, Iván Morán y Oscar, por ser como mis hermanos.



## ÍNDICE GENERAL

ÍNDICE DE ILUSTRACIONES.....	V
LISTA DE SÍMBOLOS .....	VII
GLOSARIO .....	IX
RESUMEN.....	XI
OBJETIVOS.....	XIII
INTRODUCCIÓN .....	XV
1. DESCRIPCIÓN DEL PROBLEMA.....	1
1.1. Definición del problema .....	1
1.1.1. Comunicación .....	2
1.1.2. Tergiversar la información .....	4
1.1.3. Retardo en la entrega de información.....	6
1.2. ¿Cómo un software de mensajería puede ayudar a un estudiante? .....	8
1.2.1. Filtros de información .....	12
2. DEFINICIÓN DEL PROYECTO.....	13
2.1. Descripción de la aplicación .....	13
2.2. Objetivos de la aplicación .....	14
2.3. Alcance del prototipo .....	15
2.4. Roles de la aplicación.....	17
3. MODELO CONCEPTUAL DE LOS DATOS .....	19
3.1. Método por utilizar .....	21

3.2.	Identificación de los componentes del modelo conceptual de datos .....	23
3.2.1.	Diccionario de entidades .....	23
3.3.	Relaciones entre entidades .....	26
3.3.1.	Descripción de cada relación .....	27
3.3.2.	Modelo relacional de la aplicación.....	29
4.	DISEÑO DEL SISTEMA.....	31
4.1.	Arquitectura del sistema.....	31
4.2.	Diseño físico de los datos .....	32
4.3.	Prototipos de pantallas.....	34
4.4.	Mecanismos de control y aceptación .....	41
5.	IMPLEMENTACIÓN DEL SISTEMA .....	43
5.1.	Criterio de selección de las herramientas .....	43
5.2.	Servidor y aplicación móvil.....	47
5.2.1.	Versión uno del servidor, PHP .....	49
5.2.2.	Versión dos del servidor, Socket IO y Node JS.....	51
5.2.3.	Versión tres del servidor, Firebase.....	55
5.2.3.1.	Firebase .....	55
5.3.	Gestión de la información y envío de mensajes .....	58
5.4.	Notificaciones al usuario .....	60
5.4.1.	Barra de estado.....	60
5.4.2.	Ícono de número de mensajes y Ticker.....	62
5.4.3.	Notificaciones en tiempo real dentro de una sala de chat.....	64
5.5.	Publicación de la aplicación móvil .....	65
6.	EVALUACIÓN DEL SISTEMA .....	69

6.1.	Requerimientos no funcionales .....	70
6.1.1.	Rendimiento.....	70
6.1.2.	Seguridad .....	73
6.1.3.	Ergonomía .....	74
6.2.	Detección y corrección de errores del sistema .....	76
6.2.1.	Alto consumo de datos móviles .....	77
6.2.2.	Alto consumo de batería .....	79
6.2.3.	Baja estabilidad en el envío de notificaciones .....	82
6.2.3.1.	<i>One Signal</i> .....	84
	CONCLUSIONES .....	85
	RECOMENDACIONES .....	87
	BIBLIOGRAFÍA .....	89
	ANEXOS.....	91





# ÍNDICE DE ILUSTRACIONES

## FIGURAS

1.	Esbozo general de comunicación .....	3
2.	Ejemplo 2 de tergiversar la información .....	6
3.	Notificaciones en Android.....	10
4.	Arquitectura cliente-servidor DBMS .....	20
5.	Símbolo para relacionar entidades.....	26
6.	Modelo relacional de la aplicación .....	29
7.	Arquitectura del sistema.....	32
8.	Pantalla de Log In .....	34
9.	Pantalla de registro de usuarios.....	35
10.	Pantalla de bienvenida.....	36
11.	Lista de cursos .....	37
12.	Pantalla de mensajes para maestros .....	38
13.	Pantalla de asignación de cursos para alumnos .....	39
14.	Pantalla de mensajes para alumnos .....	40
15.	Gráfica de uso de diferentes sistemas operativos para celular en FARUSAC .....	45
16.	Procedimiento superficial para entrar a una página web.....	49
17.	Consumo de MIA versión 2 en una semana de uso.....	53
18.	Diagrama general de <i>Firestore Cloud Messaging</i> .....	56
19.	Gráfica de uso de diferentes sistemas operativos para celular en FARUSAC .....	59
20.	Ejemplo de documento en formato JSON .....	60
21.	Notificación de la aplicación MIA en la barra de estado.....	62

22.	Ícono de nuevos mensajes dentro de la aplicación.....	63
23.	Ícono de nuevos mensajes fuera de la aplicación.....	64
24.	Nuevos mensajes en una sala de chat de un curso.....	65
25.	Consola de <i>Play Store</i> .....	66
26.	Gestión de versiones en Google <i>Play Console</i> .....	68
27.	Aplicación móvil MIA para descarga en <i>Play Store</i> .....	68
28.	Prueba de estrés al servidor .....	71
29.	Rendimiento del servidor recibiendo trescientas peticiones por segundo .....	72
30.	Descubrimiento de características .....	76
31.	Consumo excesivo de datos de MIA en una semana de uso .....	79
32.	Notificación sobre el uso excesivo de batería de MIA en Android. ....	81
33.	Aplicaciones protegidas .....	83
34.	Consola de <i>One Signal</i> . ....	84

## TABLAS

I.	Ejemplo de tergiversar información.....	5
II.	Diferencia de accesibilidad entre mensajería instantánea y plataforma de manejo de cursos.....	8
III.	Ejemplo de diccionario de entidades .....	21
IV.	Diccionario de entidades.....	23
V.	Identificación de relaciones del modelo relacional.....	27
VI.	Extracto de tabla de MySQL .....	33
VII.	Atributos de calidad y métrica de aceptación.....	41
VIII.	Comparativa de uso de Smartphone .....	43
IX.	Nivel de interés en que la aplicación se lleve a cabo en términos del sistema operativo que usa en su celular el estudiante.....	46

## LISTA DE SÍMBOLOS

<b>Símbolo</b>	<b>Significado</b>
<b>kb</b>	Kilobyte
<b>mb</b>	Megabyte



## GLOSARIO

<b>API</b>	Es la interfaz de programación de aplicaciones. Es un conjunto de funciones, subrutinas y procedimientos que ofrece cierta biblioteca para ser utilizada por otro software.
<b>DBMS</b>	Es un programa que sirve para almacenar y posteriormente acceder a los datos de forma rápida y estructurada.
<b>FARUSAC</b>	Es el acrónimo para hacer referencia a la Facultad de Arquitectura de la Universidad de San Carlos de Guatemala.
<b>FCM</b>	Acrónimo de <i>Firestore Cloud Messaging</i> .
<b>IP</b>	Está conformada por cuatro octetos de bites y sirve para identificar a una interfaz de red (computadora, tablet, celular y otros).
<b>JSON</b>	Es el acrónimo de <i>JavaScript Object Notation</i> , el cual sirve para el envío y recepción de datos.
<b>Push</b>	Es un método de comunicación que consiste en que el servidor de una aplicación envía un mensaje al cliente de la misma.

<b>Smartphone</b>	Es un término que se utiliza para hacer referencia a los teléfonos inteligentes, los cuales son semejantes a una minicomputadora.
<b>Spam</b>	Hace referencia a mensajes que no son de interés para el receptor y hacen que se pierda la verdadera información que se necesita transmitir.
<b>Tap</b>	Se dice que un usuario hace un tap cuando toca la pantalla de su teléfono de tipo “touch”, es análogo a un clic en la computadora.
<b>Ticker</b>	Es un mensaje que se muestra durante algunos segundos cuando se recibe una notificación en el teléfono.
<b>Token</b>	Es una cadena de caracteres que brinda autorización a un usuario para poder usar un servicio computarizado.
<b>URL</b>	Significa <i>Uniform Resource Locator</i> , el cual sirve para identificar una dirección web, por ejemplo <a href="http://www.google.com">www.google.com</a>
<b>USAC</b>	Es el acrónimo para hacer referencia a la Universidad de San Carlos de Guatemala.

## RESUMEN

El principal objetivo de este trabajo es mejorar las vías de comunicación entre los profesores y alumnos, ya que muchas veces la falta de comunicación se ve reflejada en el mal desempeño del alumno. La aplicación será de mensajería instantánea, ya que para un estudiante es más factible que las actividades escolares le lleguen al teléfono en forma de notificación, que tener que acceder a una página o algún blog.

Los profesores tendrán la capacidad de enviar mensajes en las salas de chat que le sean asignadas. Cada sala de chat corresponde a un curso con su respectiva sección. Los alumnos que se registren en la aplicación tendrán acceso a asignarse dentro de cualquier sala de chat para recibir mensajes directos y rápidos en cuanto a temas de interés.

No se debe perder el sentido de la aplicación ya que es un medio de entrega de información académica puntual y específica, por lo tanto, no sería ideal usarla para saludar o mandar chistes por medio de ella, ya que existen otro tipo de redes sociales para ello.

En el sentido técnico este trabajo puede servir como una guía para indagar sobre la mejor forma de hacer un sistema de notificaciones para teléfono celular inteligente con sistema operativo Android, debido a que se debe tener cuidado con manejar aspectos como la batería del teléfono, los datos del usuario, la facilidad para utilizar la aplicación, la rapidez de los mensajes, etcétera.





## OBJETIVOS

### General

Diseñar un software sencillo e innovador enfocado en mejorar la comunicación entre alumnos y docentes, así como también publicar noticias por parte de la dirección general de la institución a todos los alumnos y/o docentes, a través de la mensajería instantánea para teléfonos celulares con sistema operativo Android.

### Específicos

1. Diseñar un software con notificaciones instantáneas de mensajes en el celular Android del alumno para mejorar la comunicación entre los alumnos y profesores.
2. Determinar los aspectos que implica la realización de un sistema de notificaciones *push* en dispositivos con sistema operativo Android.
3. Centralizar toda la información de modo que esta sea manipulable desde el servidor para que al momento de hacer cambios no se tenga que hacer modificaciones en la aplicación.
4. Diseñar un flujo de datos para mejorar la comunicación entre docentes y estudiantes.
5. Determinar la mejor arquitectura del lado del servidor para trabajar aplicaciones en tiempo real.



## INTRODUCCIÓN

Las redes sociales han venido a revolucionar el mundo, han ayudado a varios grupos sociales a organizarse para crear eventos con asistencia masiva, mantienen informadas a las personas sobre eventos cercanos que están próximos a acontecer, cumpleaños de personas y un sinfín de aplicaciones más.

Una de las principales aplicaciones de las redes sociales es mantener comunicado al mundo, no importando la hora, el día o qué tan alejada esté una persona de otra, solo con tener un teléfono inteligente y acceso a internet esta podrá enviar mensajes instantáneos siendo estos recibidos y notificados en cuestión de segundos.

Para nadie es un secreto que el mundo ha cambiado gracias a la tecnología, es por eso que aprovechando estas herramientas de software las universidades se han tomado la tarea de actualizarse e ir implantando sistemas tecnológicos para el beneficio del estudiante, entre los más comunes se pueden encontrar el control de notas a través de plataformas web, las páginas web informativas, *fanpages* en redes sociales, correo electrónico, entre otros.

Sin embargo, un problema que se mantiene es la comunicación de mensajes de forma directa con el estudiante. Es decir, el estudiante debe estar pendiente de su bandeja de entrada de correos, tiene que ingresar a alguna página web o, en el peor de los casos, ingresar a una *fanpage* para observar algún tipo de información por parte de sus profesores o del órgano de difusión de información de su facultad.

Es allí donde toma un papel importante la mensajería instantánea, ya que esta se caracteriza por el envío de mensajes de forma directa con notificaciones instantáneas. La plataforma que se implementará cuenta con este tipo de tecnología y será aplicado para la comunicación directa de mensajes entre el alumno y el profesor o unidad de difusión de mensajes.

# 1. DESCRIPCIÓN DEL PROBLEMA

En este capítulo se definirá cuál es el problema que se quiere abordar, el segmento social en el cual se plantea el problema y cuáles son las actuales desventajas que trae dicho problema. También se presenta la hipotética solución que se le dará.

## 1.1. Definición del problema

El problema que se describirá en esta sección es la comunicación, aspecto fundamental para el proceso de aprendizaje. “La mayor parte de los seres humanos intenta entrar en contacto con otros semejantes. Lamentablemente con frecuencia surgen dificultades (...) antes de descubrir que en el origen de los problemas existe un problema de comunicación, ya existen muchos errores imputados a uno u otro de los interlocutores”<sup>1</sup>. Todo lo mencionado anteriormente no escapa a la relación que existe entre un maestro y un alumno, por ejemplo, darse cuenta –tarde– de que el profesor asignó una tarea repercute directamente sobre la calidad en el desarrollo de la misma.

El segmento social sobre el cual van planteados los problemas son los estudiantes y profesores de la Facultad de Arquitectura de la Universidad de San Carlos de Guatemala.

---

<sup>1</sup> LANGEVIN HOGUE, Lise. *La comunicación: un arte que se aprende*. p. 13.

En los siguientes apartados se explicarán algunos de los problemas más significativos desde el punto de vista del autor, en cuanto a los problemas de comunicación y en la próxima sección se dará a conocer la hipotética solución a los problemas descritos.

### **1.1.1. Comunicación**

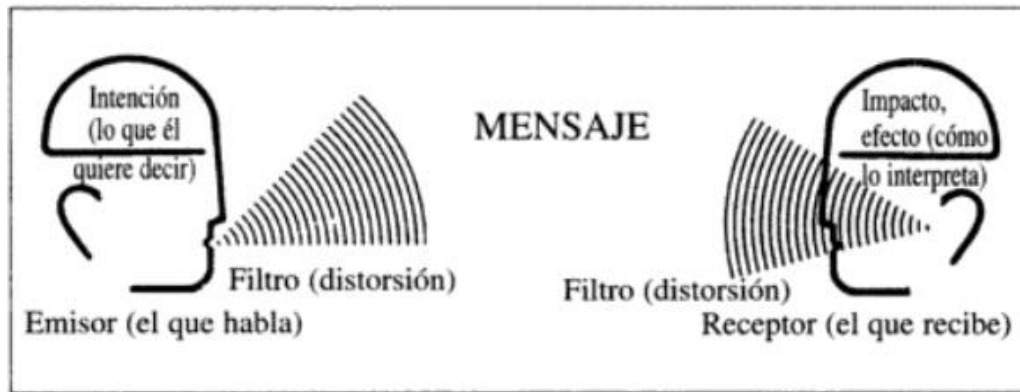
Antes de empezar a definir los posibles problemas que se puedan originar debido a la comunicación, es preciso definir ¿Qué es comunicación? Para poder aplicarlo al segmento social que se estudia.

Lise Langevin menciona que: “en un dialogo que se establece entre un emisor (el que habla) y un receptor (el que recibe el mensaje) el objetivo que persigue el emisor es transmitir un mensaje que sea bien entendido por el receptor. Si no fuera así, ¿de qué le serviría el esfuerzo que hace para comunicarse?”<sup>2</sup>

---

<sup>2</sup> Ibíd.

Figura 1. **Esbozo general de comunicación**



Fuente: LANGEVIN HOGUE, Lise, *La comunicación un arte que se aprende*. España: Salterrae, 2000. 18 p.

En la figura 1 se observa el proceso para una comunicación verbal, pero ¿qué desventajas trae transmitir información por medio de la voz? Cuando se transmite información por medio de la voz el mensaje tiende a ser entendido de varias maneras dependiendo del receptor, ya que no todos piensan igual. Además de esto, también la información tiende a ser olvidada por las personas. Según algunos estudios, las personas tienen mayor capacidad de abstraer información visual que auditiva.

Hay muchas formas de comunicación ya sea mímica, verbal o escrita. En este documento se hará énfasis sobre la comunicación por medio de la escritura, este es un tipo de comunicación que está de moda hoy día debido a los avances tecnológicos en la sociedad, las redes sociales y las herramientas de mensajería instantánea que han revolucionado la forma de vivir de las personas. Todo el mundo está compartiendo pensamientos, haciendo



recordatorios o simplemente saludando a través de estas herramientas. Por ese motivo se ha decidido orientar estos servicios a la comunidad estudiantil, específicamente al segmento social descrito anteriormente.

### **1.1.2. Tergiversar la información**

En la sección 1.1.1 se dio una definición para la comunicación así como también se abordaron ciertos problemas que podían surgir de ella. En esta sección se empezarán a dar detalles sobre estos, uno de los más comunes y perjudiciales para un alumno es tergiversar la información. Según Julián Pérez Porto y Ana Gardey del portal de definiciones “definición de” tergiversar la información es: “acción que se refiere a modificar el sentido de un discurso o de un suceso para provocar una interpretación equivocada”<sup>3</sup>.

En este documento al hablar de discurso se va a referir a sentencias emitidas por un profesor para dar a conocer alguna información. La principal causa de tergiversar información es la ambigüedad en la sentencia. Por ejemplo, un profesor quiere dejar una tarea a sus alumnos, el profesor anuncia la tarea a las 9 a.m. (véase la tabla I).

---

<sup>3</sup> PÉREZ PORTO, Julián y GARDEY, Ana. *Definición.de*. <https://definicion.de/tergiversar/>. [En línea] [consultado en el: 28 de julio de 2017.]

Tabla I. **Ejemplo de tergiversar información**

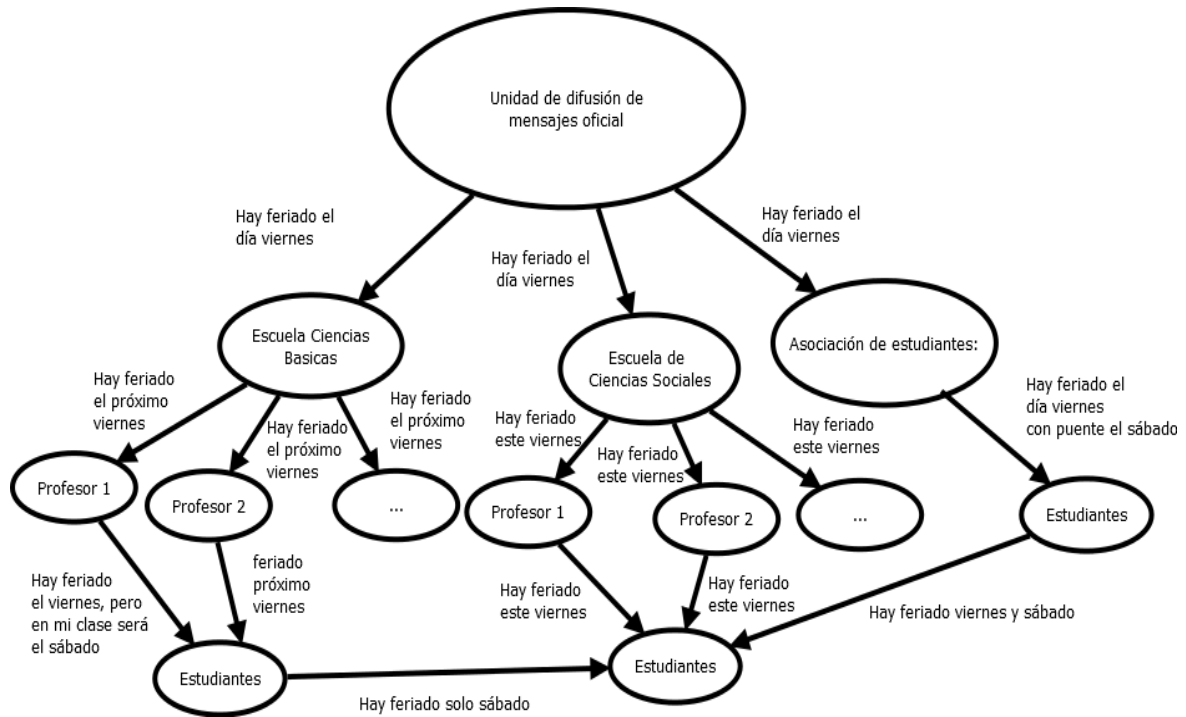
<b>Lo que el profesor dice</b>	<b>Lo que Juan entiende</b>	<b>Lo que Pedro entiende</b>
Mándenme la tarea para antes de las doce.	La tarea es para las 12 p.m. del día.	La tarea es para las 12 a.m. del día siguiente.

Fuente: elaboración propia.

Esta es una manera de tergiversar la información, porque tanto Juan como Pedro entendieron de diferente manera el mensaje del profesor y estos dirán a sus amigos diferente información, los amigos replicarán la información de maneras distintas y así sucesivamente. El origen de haber tergiversado la información, en la sentencia del profesor, fue la ambigüedad que existe, al decir las doce no se está refiriendo a un día en específico ni tampoco si se está refiriendo a la media noche o al medio día.

Otra forma de tergiversar la información es la replicación del mensaje a través de varias personas, algo así como el famoso juego de 1990 “el teléfono descompuesto” en donde hay que emitir un mensaje a través de varias personas, entre más personas se encuentran jugándolo más probabilidades hay de que se tergiverse el mensaje, enfocándolo en el segmento social seleccionado, esto puede verse reflejado en la figura 2.

Figura 2. **Ejemplo 2 de tergiversar la información**



Fuente: elaboración propia a través del programa DIA

Como se observa en el árbol de la imagen la información fue tergiversada debido al fenómeno del “teléfono descompuesto” esto pasa por las diversas fuentes de información y no tener un solo órgano encargado de emitir mensajes oficiales respecto de las actividades estudiantiles.

### 1.1.3. Retardo en la entrega de información

El retardo en la entrega de información es otro gran problema cuando se habla de comunicación. Por retardo en la entrega de información se debe

entender toda aquella información que es de importancia para el receptor y que llega a las manos de forma tardía, es decir, el receptor tiene que actuar de forma reactiva ante la situación que se presenta debido a la falta de información a tiempo.

Hay mucha diferencia en la calidad de los trabajos cuando se habla de una persona proactiva en comparación de una persona reactiva. Persona proactiva es toda aquella que trabaja para que las cosas sucedan, mientras que la reactiva es aquella que trabaja en base a las cosas que le suceden. En muchas ocasiones el origen de la reactividad se origina por el retardo en la entrega de la información.

La pregunta es ¿Por qué existe retardo en la entrega de información? Es necesario aclarar que "entrega de la información" no precisamente se refiere a que el emisor entrega tarde la información, sino también el canal de comunicación que puede retrasar la entrega del mensaje. Por ejemplo, cuando una persona ve un video en la plataforma de internet Youtube y su ancho de banda es bajo, el vídeo tiende a tardar en cargarse para su visualización. Este problema es del canal de comunicación.

Otro factor importante en cuanto al canal de comunicación es el nivel de accesibilidad que este posee. En el contexto del problema abordado en este documento, se presenta la tabla II con las diferencias entre un canal de comunicación de mensajería instantánea y una plataforma de manejo de cursos convencional.

Tabla II. **Diferencia de accesibilidad entre mensajería instantánea y plataforma de manejo de cursos**

<b>Mensajería instantánea</b>	<b>Plataforma de manejo de cursos convencional</b>
<ol style="list-style-type: none"> <li>1. Dar clic a la aplicación</li> <li>2. Buscar grupo</li> <li>3. Ver información</li> </ol>	<ol style="list-style-type: none"> <li>1. Ingresar dirección URL</li> <li>2. Identificarse en el sistema</li> <li>3. Ir a la pestaña de cursos</li> <li>4. Buscar el curso del que se quiere ver información</li> <li>5. Ir a la pestaña de anuncios</li> <li>6. Ver información</li> </ol>

Fuente: elaboración propia con tablas de Microsoft Word 2010

Una herramienta de mensajería instantánea posee más accesibilidad que una plataforma de cursos convencional, como se puede ver en la tabla II. Sin embargo, esto no significa que la mensajería instantánea sea mejor para el manejo de cursos, la plataforma posee más características y es más completa debido a que tiene opciones como cargar material, llevar un mejor orden en el cronograma.

## 1.2. **¿Cómo un software de mensajería puede ayudar a un estudiante?**

En esta sección es oportuno volver a plantear ¿Qué es lo que busca el software que se diseñará en este documento?, su objetivo primordial, es difundir mensajes y mejorar la comunicación entre el profesor y el alumno,

además de emitir mensajes generales para todos los alumnos. Las preguntas que surgen a partir de este enunciado son: ¿Qué atributos se deben buscar para que el canal tenga un alto nivel de accesibilidad?, ¿Es necesario tener atributos como sección de documentos, foros y otros, cuando mi objetivo es difundir información? y ¿Cómo puedo evitar tergiversar la información y evitar el retardo en la información? Las respuestas a estas interrogantes se darán en las secciones 1.2.1 y 1.2.2.

### **1.2.1. Mensajería instantánea**

En este apartado se explicará por qué la mensajería instantánea puede ser una solución fiable para las preguntas planteadas anteriormente, pero antes se debe definir qué es mensajería instantánea. En el libro *Comercio electrónico* se menciona que: “mensajería instantánea es una forma de comunicación en tiempo real, entre dos o más personas, basada en texto”<sup>4</sup>. De la definición anterior se puede agregar que las notificaciones de los mensajes instantáneos le dan un gran alcance a la información, ya que para el usuario basta con dar un tap en la notificación para acceder a ella (véase figura 3).

---

<sup>4</sup> CASTAÑO, Juan José y JURADO, Susana. . *Comercio electrónico*.2016. 78p.

Figura 3. **Notificaciones en Android**



Fuente: LÓPEZ, Alejandro, *Status bar notification*. [En línea] 2016. [Citado el: 12 de junio de 2016] <http://rootear.com/android/recupera-notificaciones-android>.

Ahora que ya se definió qué es mensajería instantánea, se contestarán las preguntas planteadas en la sección 1.2.

- Para tener un nivel alto de acceso a la información es necesario difundir mensajes que lleguen rápidamente y en tiempo real al usuario a través de notificaciones en el teléfono celular, para tener acceso de forma instantánea a cualquier mensaje. La mensajería instantánea posee todos estos elementos, además es una herramienta que las personas están acostumbradas a ver a cada instante.
- Se deben desechar algunos atributos que no son parte de la difusión de mensajes, por ejemplo, un foro sirve para temas de discusión de cualquier tema. En este caso no interesa discutir temas, tampoco se quiere compartir documentos, el objetivo es difundir mensajes. Las ventajas de desechar estos elementos son importantes, ya que se obtiene una aplicación más liviana y se optimiza el espacio del teléfono, además se puede tener un acceso directo a los mensajes sin tener que pasar por menús y opciones que pueden resultar tediosas.
- Para manejar la información se deben definir estrictas reglas de seguridad para los emisores de mensajes y de ese modo evitar que la información se vea tergiversada, por ejemplo, que solo el profesor de cierto curso pueda enviar información a los alumnos asignados a su curso en la aplicación y que el único que pueda enviar mensajes generales sea el órgano competente de anunciar feriados, actividades oficiales, etcétera.
- El retardo en la información no es problema para la mensajería instantánea debido a los mensajes en “tiempo real” que esta posee.



### 1.2.2. Filtros de información

El filtro de información es una de las partes fundamentales del software, siendo este el encargado de presentarle al alumno exclusivamente la información que a él le compete de forma automática. El estudiante únicamente deberá asignarse en la aplicación los cursos que está llevando, para que esta filtre la información de los cursos asignados. A continuación se definen las reglas para el filtro de información.

- El estudiante puede asignarse la cantidad de cursos que desee.
- El estudiante recibirá notificaciones únicamente de los cursos a los que esté asignado o mensajes generales que envíe el servidor.
- El servidor es el único que puede enviar mensajes a todos los alumnos registrados en la aplicación.
- El profesor podrá enviar mensajes únicamente a los estudiantes que estén asignados a su curso, en la aplicación.
- El alumno no podrá enviar mensajes para evitar el *spam*.

## 2. DEFINICIÓN DEL PROYECTO

En este capítulo se definirá un concepto formal de una posible solución al problema planteado en el capítulo anterior. Se dará a conocer la herramienta y los elementos que la componen, también se definirán los objetivos, los alcances y los tipos de usuario identificados.

### 2.1. Descripción de la aplicación

La herramienta que se implementará tiene las características de un software de mensajería instantánea, ya que como se explicó en la sección 1.2.1 la mensajería instantánea se ajusta a las características de la solución por ser una vía de información rápida (tiempo real), confiable y eficaz.

- **Aplicación móvil orientada a un chat**

Una de las principales intenciones del software planteado es explotar recursos como las aplicaciones móviles, las cuales se están volviendo parte de la vida diaria, y darles un enfoque productivo, que sirva para aumentar el rendimiento del estudiante y que, a la vez, le sirvan para mantener una relación más cercana y divertida con sus maestros. Contará con mensajes ambientados a un chat de mensajería móvil, además de tener información de primera mano, sin necesidad de tener que ingresar a una URL, sino más bien, únicamente revisar la notificación en su teléfono celular y acceder de forma instantánea a la información en tiempo real.

La aplicación será ambientada a un chat, pero es importante decir que la comunicación será unilateral. El único emisor de mensajes será el profesor, esto se hace con el afán de evitar que los estudiantes puedan salirse del contexto de la información que el profesor quiere transmitir y que los alumnos que vean tarde el mensaje encuentren saturada su bandeja de entrada de mensajes emitidos por sus compañeros. Este es un fenómeno común en los grupos sociales que se manejan en la mensajería instantánea.

- **Usabilidad de la aplicación**

Debe ser ergonómica, es decir, el alumno debe adaptarse rápidamente a ella, sin tantas opciones y sin tantas pantallas que puedan llegar a confundir. Se debe mantener el objetivo, el cual es mantener informada a la población.

## **2.2. Objetivos de la aplicación**

La aplicación por diseñar debe responder a los objetivos de una aplicación de mensajería instantánea, aunque según las políticas que se han definido hasta el momento esta presenta algunas variantes que enmarcan una significativa diferencia entre una y otra. Para empezar se definirán los objetivos que tienen en común ambas aplicaciones:

- Comunicar el mensaje del emisor a uno o más receptores.
- Enviar mensajes de la manera más simple posible.
- Comunicar el mensaje del emisor en tiempo real.
- Crear un mensaje legible para los receptores con diferentes opciones para estructurar la información.
- Notificar mensajes en la aplicación, con el sistema de notificaciones de Android (Figura 3).

Ahora se definirán las características que serán específicas para la aplicación por diseñar.

- Crear un mensaje legible para los receptores con diferentes opciones de color para el texto.
- Crear un mensaje legible para los receptores con diferentes opciones de tipografía para el texto.
- Crear políticas internas de envío de mensajes respecto de los roles de profesor y alumno.
- Crear una lista de cursos de FARUSAC a los cuales el alumno se puede asignar dentro de la aplicación
- Manipular la información de los cursos y profesores de forma centralizada, de modo que los cambios hechos en el servidor se vean reflejados automáticamente en la aplicación.

### **2.3. Alcance del prototipo**

- Usuarios de la aplicación

El alcance respecto de los usuarios que tendrá la aplicación serán los profesores y alumnos de la Facultad de Arquitectura de la Universidad de San Carlos de Guatemala. No obstante, la aplicación podrá ser descargada por cualquier persona que pueda acceder al centro de descargas de Android, Play Store.

- Cursos de la aplicación

Se contará con una base de datos con los cursos de dicha Facultad, en cada curso la aplicación se limitará a la siguiente información para formar su salón de chat:

- Código del curso
- Nombre de curso
- Sección de curso
- Catedrático que imparte el curso

Es importante notar que la información que se solicita para cada curso, es puramente para la identificación del mismo, sin embargo, la aplicación no es informativa en cuanto al salón donde se imparte el curso, el horario en que se imparte, etcétera. Se limitará en función del envío de mensajes a través de códigos únicos para identificar cada salón.

- Opciones de la aplicación:
  - Formato HTML para mensajes
  - Sesiones de usuario basada en roles
  - Asignarse cursos para los alumnos
  - Envío de mensajes únicamente de profesores

- Mensajes de la aplicación

Todos los mensajes que emita un profesor deben pasar por el servidor, además este también podrá enviar mensajes a los alumnos registrados en la aplicación de manera general.

- Mensajes sin conexión

.Si un alumno no tiene conexión a internet los mensajes estarán en espera hasta que el alumno tenga acceso a internet.

## **2.4. Roles de la aplicación**

- Modo súper usuario

- 

Este será el rol de la persona que pueda manipular a nivel general la mensajería, es decir, es el único que puede enviar mensajes a todos los usuarios de la aplicación, incluyendo a maestros y alumnos. Los datos que se requieren para el manejo del súper usuario son:

- Contraseña de súper usuario
- Código único para el súper usuario

- Maestros de la aplicación

El rol maestro podrá enviar mensajes únicamente a los alumnos que estén asignados dentro de su curso en la aplicación. Los datos que se requieren para el manejo de maestros son:

- Nombre del maestro
- Contraseña
- Código único para cada maestro
- Lista de cursos que imparte

- Alumnos de la aplicación

El rol alumno no podrá enviar mensajes a ningún usuario, su función es únicamente recibir mensajes ya sea del súper usuario o de sus respectivos maestros. Los datos que se requieren para el manejo de alumnos son:

- Nombre del alumno
- Carné
- Contraseña
- Cursos que recibe
- 

Es importante notar que la información que se requiere para el rol alumno y el rol maestro es mínima, no se piden datos como la edad, el teléfono, etcétera. La aplicación se limita a enviar mensajes y ser una vía de comunicación rápida y eficiente a través de un código de identificación único para cada persona.

### **3. MODELO CONCEPTUAL DE LOS DATOS**

En este capítulo se tratará un tema muy importante para el funcionamiento de la aplicación, ya que se detallará qué datos se tienen que guardar, dónde serán guardados y de qué forma serán almacenados.

Es importante mencionar los tipos de almacenamiento más usados en una computadora, el primero es el almacenamiento primario, es decir, la memoria que se usa para que un programa se ejecute debido a la rapidez para la transferencia de datos. Este tipo de almacenamiento es temporal, ya que la información se pierde cuando el equipo se apaga. En el disco duro, también llamado memoria secundaria, el tipo de almacenamiento es no volátil y es más lento que el almacenamiento primario, sin embargo, perdura a lo largo del tiempo.

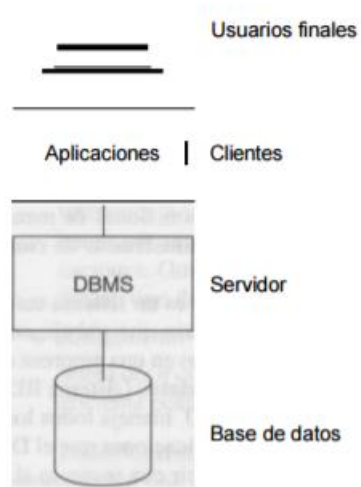
En el diseño de la aplicación planteada se usará como memoria secundaria al disco duro de la computadora, ya que este es el más usado para guardar datos que perduren a lo largo del tiempo. Se deben identificar los diferentes elementos de los que se requiere guardar información, a estos se les nombrará como “entidades”. Además, estas entidades están conformadas por una serie de atributos y relaciones con otras entidades.

A nivel físico la información estará manejada por medio de un DBMS a través del paradigma cliente-servidor, es decir, el DBMS será el servidor que gestione los datos y los almacene, mientras que el cliente es el usuario que solicita los datos al servidor. En el nivel físico una entidad se transforma en una



tabla y sus atributos en las columnas de las tablas, cada registro en la tabla se conoce como tupla.

Figura 4. **Arquitectura cliente-servidor DBMS**



Fuente: ADDISON, Wesley, *Introducción a los sistemas de bases de datos*. 7a ed. México: PEARSON EDUCACIÓN S.A., 2000. 48

Tabla III. **Ejemplo de diccionario de entidades**

ENTIDAD	DESCRIPCIÓN	ATRIBUTOS
<b>Alumno</b>	El conjunto de los estudiantes de una universidad determinada	<b>id alumno</b> <b>carnet</b> <b>apellidos</b> <b>nombres</b> <b>nivel</b>
<b>Asignatura</b>	El conjunto de las asignaturas que se dictan en la universidad	<b>id asignatura</b> <b>código</b> <b>asignatura</b> <b>créditos</b>
<b>Programa</b>	El conjunto de los programas académicos que se dictan en la universidad	<b>id programa</b> <b>programa</b>
<b>Periodo</b>	El conjunto de los periodos académicos que han transcurrido en la universidad	<b>id programa</b> <b>periodo</b>

Fuente: ADDISON, Wesley, *Introducción a los sistemas de bases de datos*. 7a ed. México: PEARSON EDUCACIÓN S.A., 2000. 48 p.

### 3.1. Método por utilizar

Hay muchas formas de representar los datos en el disco duro de la computadora, pero el método más usado para conceptualizar los datos es el modelo entidad-relación este corresponde al tipo relacional de base de datos, en el libro *Base de datos relacionales* se afirma que el modelo entidad-relación es: “la percepción de un mundo real que consiste en un conjunto de objetos básicos llamados entidades y de unas relaciones entre estos objetos. Se utiliza para esquematizar la estructura lógica general de lo que será la base de datos.”<sup>5</sup>

<sup>5</sup> OSORIO RIVERA, Fray León. *Bases de datos relacionales: Teoría y práctica*. p. 25.

Una de las principales ventajas de utilizar este método es la integridad que proporciona a los datos, es decir, se deben cumplir ciertas restricciones para modificar, eliminar o agregar datos, estas restricciones el DBMS debe hacerlas respetar a toda costa para mantener la consistencia de los datos y evitar que estos se corrompan. La integridad de la información se guarda a través de llaves. Estas se describirán a continuación (la base de las definiciones fueron tomadas del libro *Base de datos relacionales*).

- Superllave

Es un conjunto de uno o más atributos, los cuales, juntos, permiten identificar en forma única a una entidad dentro del conjunto de entidades.

- Llave candidato

Son todas las posibles combinaciones de atributos que podrían ser llave primaria.

- Llave primaria

Es la llave candidato que el diseñador de la base de datos elige para identificar una entidad dentro de un conjunto de entidades.

- Llave foránea

Son las llaves que se forman a partir de una relación, estas se construyen tomando los atributos que constituyen las llaves primarias de los conjuntos de entidades que definieron el conjunto de relaciones.

### 3.2. Identificación de los componentes del modelo conceptual de datos

Una vez que ya se ha descrito la metodología que se va a seguir para el manejo de los datos en el disco duro por medio de un DBMS, es oportuno empezar a identificar cuáles son las entidades que participarán en el modelo entidad-relación a través de un diccionario de entidades.

#### 3.2.1. Diccionario de entidades

En este diccionario se detallarán los nombres de las entidades, los atributos de cada entidad y una pequeña descripción de la entidad, tal y como se vio en la tabla III.

Tabla IV. Diccionario de entidades

Entidad	Descripción	Atributos
Maestro	Representará al rol de maestros (sección 2.4) de FARUSAC.	-Código de maestro -Nombre -Contraseña -Tipo (súper usuario o maestro normal)
Alumno	Representará al rol de alumnos (sección 2.4) de FARUSAC.	-Carné -Nombre -Contraseña

Continuación de la tabla IV

Curso	Es la representación de los cursos de FARUSAC que sean almacenados en la base de datos.	-Código de curso -Sección -Nombre
Asignación maestro	Representa los cursos que un maestro impartirá en determinado semestre.	-Código del maestro -Código del curso -Sección del curso -Fecha de asignación
Asignación alumno	Representará todas las asignaciones de cursos que hagan los alumnos. Es importante guardarlos como master detalle, para llevar un registro histórico de cada asignación.	-Código del alumno que se va a asignar -Código del curso que se asignará -Sección del curso que se le asignará -Fecha de la asignación
Periodo	Indica en que año y semestre fue asignado un curso a un maestro o alumno.	-Semestre -Año

Continuación de la tabla IV

Mensaje	Representa la entidad encargada de guardar cada mensaje que pase por el servidor	-Un identificador para el mensaje -El carné del alumno que recibió el mensaje -El código del maestro que emitió el mensaje -El contenido del mensaje - Si un mensaje ya fue visto
---------	--	---

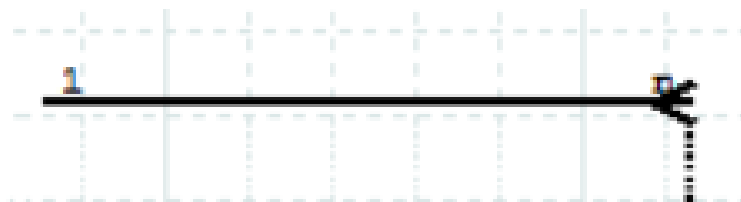
Fuente: elaboración propia con tablas de Microsoft Word 2010.

### 3.3. Relaciones entre entidades

Ahora que ya están definidas las entidades que participarán en el modelo entidad-relación, es necesario determinar las relaciones que tiene cada una de ellas entre sí. Estas serán representadas conceptualmente a través de diagramas UML.

Se puede empezar definiendo que para que exista una relación deben existir dos entidades que tengan alguna conexión entre sí en la lógica de la aplicación. También debe existir una cardinalidad en la relación.

Figura 5. **Símbolo para relacionar entidades**



Fuente: elaboración propia con el programa DIA.

En la figura 5 se puede observar la representación de una relación en UML, del lado izquierdo expresa cardinalidad de uno, mientras que el lado derecho expresa cardinalidad de “n” o muchos.

También existen otros tipos de relaciones, como las de uno a uno y las de muchos a muchos, estas dos últimas no están bien vistas cuando se diseñan bases de datos, debido a que tiene anomalías de programación y conceptuales.

### 3.3.1. Descripción de cada relación

En esta sección se van a describir todas las relaciones identificadas para las entidades de la sección 3.2.1

Tabla V. Identificación de relaciones del modelo relacional

Entidad con cardinalidad 1	Entidad con cardinalidad n	Descripción de la relación
Maestro	Asignación para maestro	Un maestro puede asignarse como catedrático de muchos curso
Alumno	Asignación para alumno	Un alumno puede asignarse como estudiante de muchos cursos.
Curso	Asignación para maestro	Un curso puede tener a muchos maestros asignados dependiendo de la sección y del semestre que esté cursando.
Curso	Asignación para alumno	Un curso puede tener a muchos alumnos asignados.



Continuación de la tabla V

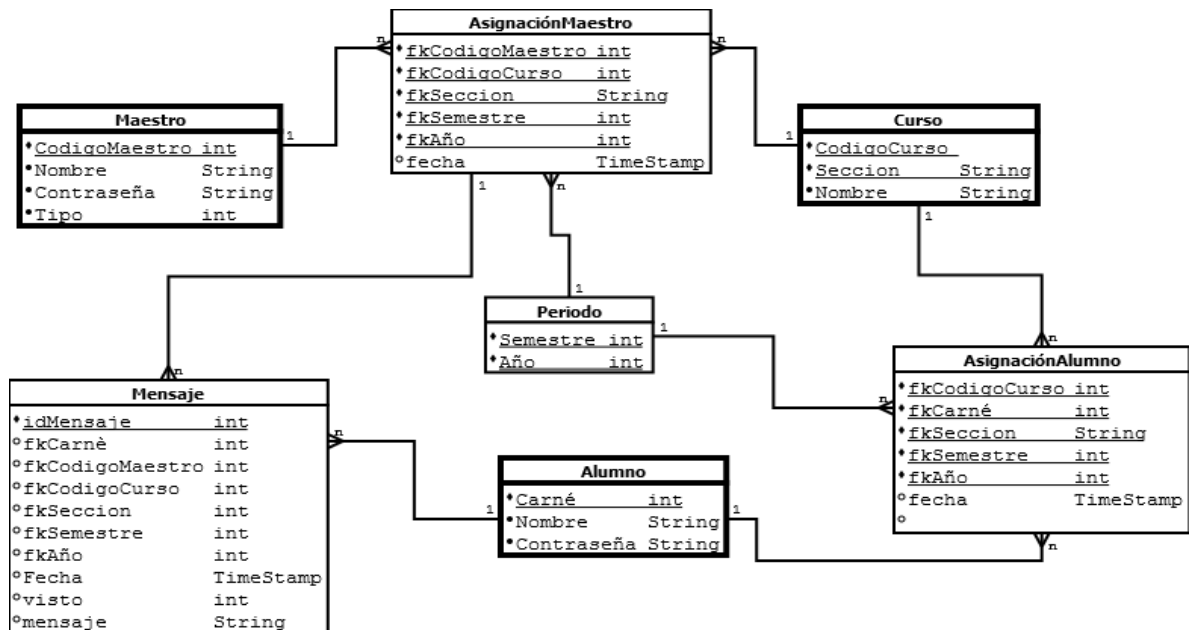
Periodo	Asignación para maestro	En un periodo universitario pueden haber muchas delegaciones de maestros a catedráticos de algún curso..
Periodo	Asignación para alumno	En un periodo universitario puede haber muchas asignaciones de alumnos a cursos.
Asignación para el Maestro	Mensaje	Por cada asignación de curso que se le haga el maestro puede emitir muchos mensajes a alumnos que estén asignados a dicho curso.
Alumno	Mensaje	Un alumno puede recibir muchos mensajes

Fuente: elaboración propia con tablas de Microsoft Word 2010.

### 3.3.2. Modelo relacional de la aplicación

En la figura 6 se puede observar el resultado al juntar las entidades de la sección 3.2.1 con las relaciones de la sección 3.3.1. Este sería el resultado final del modelo conceptual para el manejo de los datos en el DBMS.

Figura 6. Modelo relacional de la aplicación



Fuente: elaboración propia con el programa DIA.



## 4. DISEÑO DEL SISTEMA

En este capítulo se creará todo el bosquejo del sistema, empezando por la arquitectura del sistema, tomando como base el paradigma cliente-servidor siendo esta una de las arquitecturas más usadas en la creación de software. Luego se describirán las especificaciones del ambiente donde correrá la aplicación móvil y el servidor, se bosquejarán las pantallas de la aplicación buscando la ergonomía y la utilización de las mismas, por último, se definirán los mecanismos de control y aceptación que serán evaluados en el capítulo 6 después de haber implementado el sistema.

### 4.1. Arquitectura del sistema

La arquitectura del sistema está asociada a dos elementos principales:

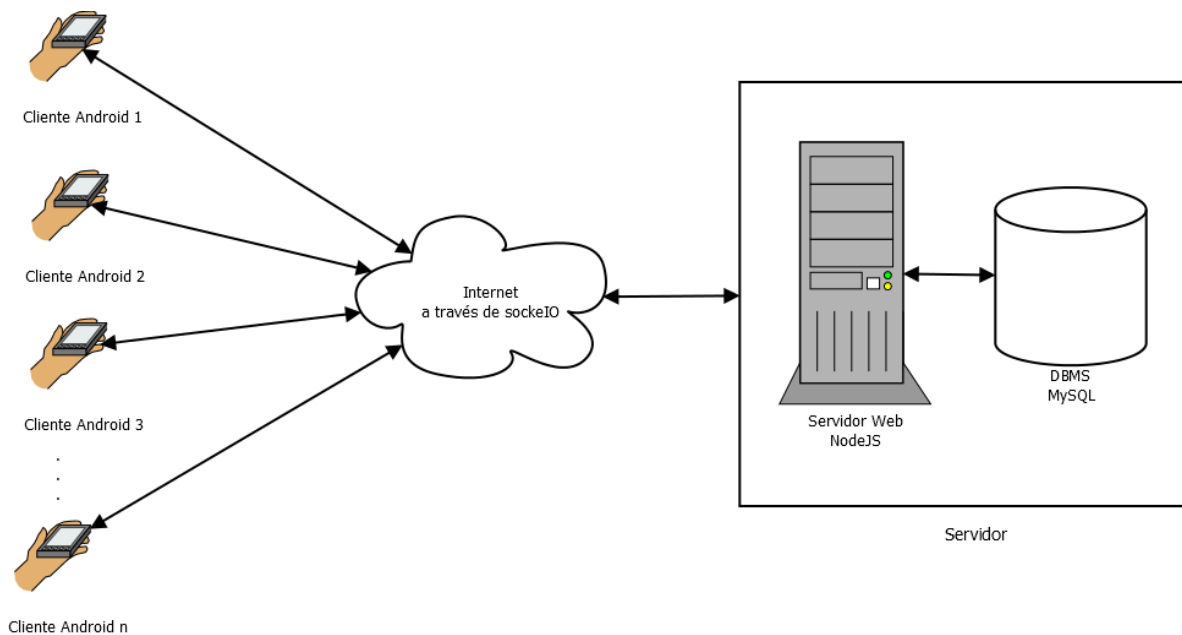
- Servidor
- Cliente

Este modelo de arquitectura es muy usado en proyectos de software y la idea principal es que todos los usuarios conocidos como clientes hagan peticiones a un ente llamado servidor, el cual se encarga de administrar y centralizar la información.

La relación de cardinalidad que hay entre el cliente y el servidor es de uno a muchos, es decir, un servidor recibe peticiones y administra a muchos clientes a través de internet. En este documento el rol de cliente lo reciben dispositivos móviles con sistema operativo Android, mientras que el rol de servidor será para

una computadora con un servidor web en NodeJS y un gestor de datos en MySQL para el modelo de datos del capítulo 3. La descripción de las herramientas mencionadas anteriormente se detallará en la sección 5.1.

Figura 7. **Arquitectura del sistema**



Fuente: elaboración propia con el programa DIA.

## 4.2. **Diseño físico de los datos**

Ahora que ya se ha definido una arquitectura formal para el sistema y también un modelo lógico para los datos, es momento de definir la transformación que este sufrirá para pasar de la abstracción conceptual a un objeto físico dentro del disco duro de la computadora. Las entidades del modelo conceptual se transformarán en tablas dentro del DBMS y cada atributo de la entidad se transformará en las columnas de la tabla.

La tabla se va llenando con registros ingresados por el servidor web a través de peticiones que lleguen de los diferentes clientes, cada fila llena dentro de la tabla se conoce con el nombre de “tupla”.

Como ya se mencionó, el DBMS será MySQL con el motor de base de datos InnoDB, ya que está especializado para modelos estrictamente relacionales y de ese modo asegurar la integridad de los datos para evitar la inconsistencia de los datos y la duplicidad.

Tabla VI. **Extracto de tabla de MySQL**

The diagram shows a table with three columns: 'id', 'nombre', and 'edad'. The table is enclosed in a black box with dashed lines. A bracket labeled 'Atributos' points to the column headers. A bracket labeled 'Tuplas' points to the rows of data.

id	nombre	edad
1	Jose	45
2	Alcacer	55
7	George	21

Fuente: captura de pantalla de MySQL Command.

### 4.3. Prototipos de pantallas

En esta sección se definen los prototipos para las pantallas que tendrá la aplicación móvil, los cuales fueron definidas junto con la Dirección de Planificación de FARUSAC. Estas pantallas están divididas en tres partes las de uso general, las específicas de los maestros y las específicas de los alumnos.

Figura 8. Pantalla de Log In



Fuente: elaboración propia con la plataforma MockFlow.

La pantalla de Log In servirá para la autenticación de alumnos, maestros y súper usuario, se debe especificar el rol en el campo “Tipo de usuario”. Además, el campo “Registro” es el código de identificación personal y “Contraseña” es la palabra secreta especificada cuando un usuario se registra; los datos serán validados en el servidor y si los datos son correctos el usuario podrá entrar a la aplicación.

Figura 9. **Pantalla de registro de usuarios**



The image shows a smartphone screen with a registration form. The title bar at the top is blue and contains the word "Registro". The form fields are as follows:

- Registro o carné**: A text input field.
- Nombre**: A text input field.
- Contraseña**: A text input field.
- Repita contraseña**: A text input field.
- Rol**: A dropdown menu with "Maestro" selected.

At the bottom of the form is a yellow button labeled "Registrarse".

Fuente: elaboración propia con la plataforma MockFlow.



La pantalla para registro de usuarios con roles de maestro o alumno será la misma, a excepción del súper usuario. Este no contará con un registro directo en la aplicación sino será directamente con el departamento de informática de FARUSAC.

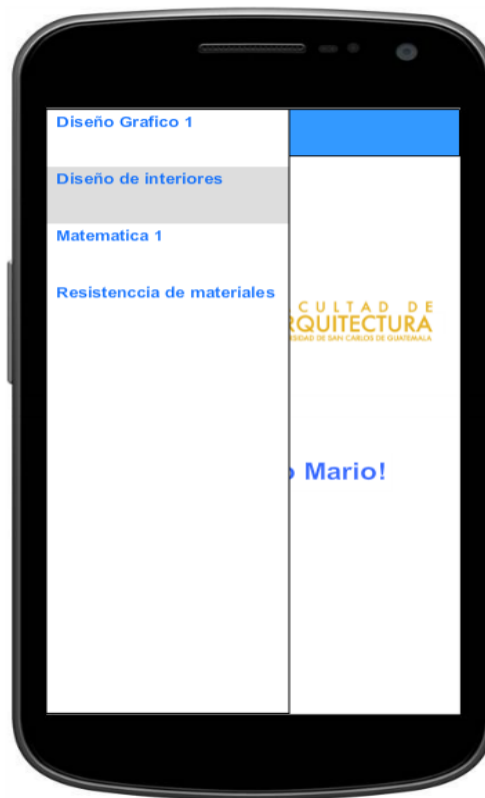
Figura 10. **Pantalla de bienvenida**



Fuente: elaboración propia con la plataforma MockFlow.

Independientemente si se trata de un alumno, maestro o súper usuario la aplicación mostrará un mensaje de bienvenida con el nombre del usuario y el logo de FARUSAC.

Figura 11. **Lista de cursos**



Fuente: elaboración propia con la plataforma MockFlow.

Se pretende tener el menor número de menús posibles y posicionados estratégicamente. En la barra de lista de cursos un alumno podrá verificar a qué cursos está asignado (asignación dentro de la aplicación) y con un tap podrá entrar a ver los mensajes de dicho curso, mientras que un maestro podrá ver sus cursos asignados y al entrar podrá ver los mensajes que ya envió o enviar

un nuevo mensaje. Por último, el súper usuario también contará con la barra de cursos, pero además también podrá publicar noticias en la pantalla principal de la aplicación, las noticias pueden ir dirigidas a: “alumnos”, “maestros” o “todos”.

Figura 12. **Pantalla de mensajes para maestros**



Fuente: elaboración propia con la plataforma MockFlow.

Cuando un maestro presiona un botón sobre la barra de cursos aparecerá la pantalla para enviar mensajes a dicho curso, en esta se pueden ver los mensajes que se han enviado con anterioridad y también aparece un campo de texto para ingresar un nuevo mensaje. Además, se tendrá la opción de emitir un texto por vía de la voz.

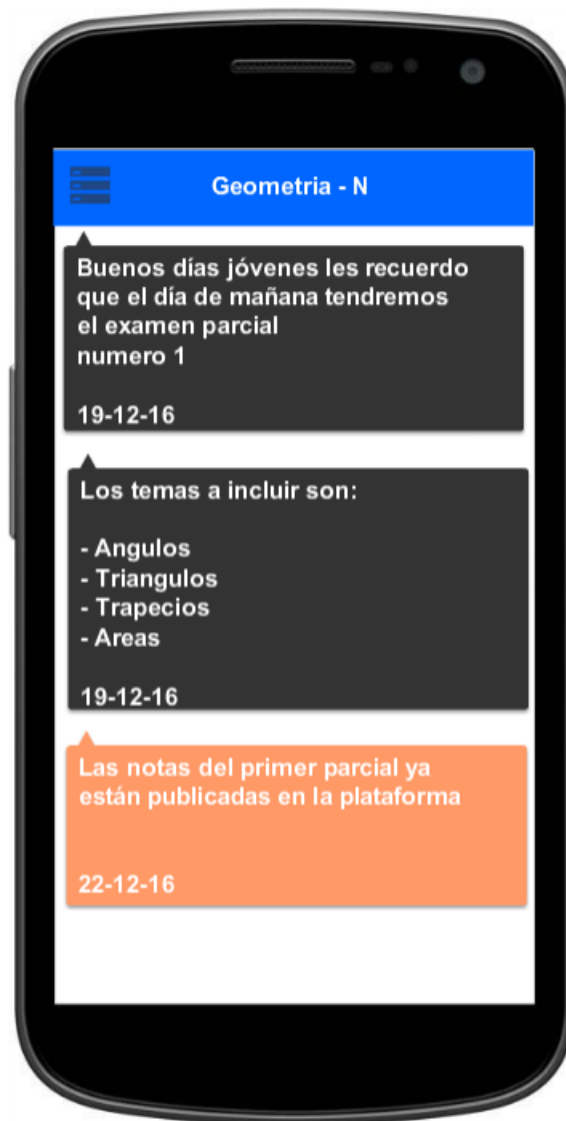
Figura 13. **Pantalla de asignación de cursos para alumnos**



Fuente: elaboración propia con la plataforma MockFlow.

Para facilitar la búsqueda de cursos para la asignación del alumno, la pantalla de asignación de cursos contará con una barra de búsqueda en la parte superior de la pantalla, los cursos serán filtrados en base a los parámetros de búsqueda.

Figura 14. **Pantalla de mensajes para alumnos**



Fuente: elaboración propia con la plataforma MockFlow.

Los mensajes no leídos por el alumno tendrán diferente color al que tienen los mensajes que ya fueron leídos.

#### 4.4. Mecanismos de control y aceptación

La evaluación del proyecto después de su implementación se hará a través de los mecanismos de control y aceptación establecidos en esta sección, justo antes de la fase de implementación del sistema. En términos generales, los atributos de calidad que se evaluarán, son aquellos vitales para el funcionamiento de un software de mensajería instantánea, esto implica que se debe garantizar la comodidad de todos los usuarios del sistema, así como la seguridad de los datos que el usuario le proporcione a la aplicación.

En la tabla VII se puede apreciar el atributo de calidad por evaluar, así como la métrica para considerar aceptable el atributo.

Tabla VII. **Atributos de calidad y métrica de aceptación**

<b>Atributo de calidad</b>	<b>Métrica</b>
Velocidad del mensaje	Siempre que el receptor tenga una conexión activa a internet, el mensaje no debe demorar más de 5 segundos en ser entregado.
Consumo de internet al entrar a la aplicación	La primera pantalla de la aplicación son las noticias, se extraerán únicamente 10 al inicio. Se pretende manejar un rango de 60 kb a 120 kb cada vez que un usuario ingresa.

Continuación de la tabla VII

Consumo de internet para ver los mensajes de un curso	Esta métrica depende de la cantidad de caracteres que tenga cada mensaje de dicho curso. Se extraerán únicamente 10 mensajes al inicio se obtendrán más a medida que se vayan requiriendo por parte del usuario. Se pretende manejar un rango de 50 kb a 100 kb cada vez que un usuario ingresa.
Consumo de internet sin actividad de la aplicación	La aplicación no debe de consumir datos mientras esté cerrada, al menos que se reciba una notificación <i>push</i> o sea necesario actualizarla.
Concurrencia aceptada por el servidor.	El servidor debe responder con un rendimiento adecuado (no óptimo) hasta con 4 000 usuarios.
Navegación entre pantallas	Al cambiar de una pantalla a otra la aplicación no debe demorar más de 2 segundos con una conexión estable a internet.

Fuente: elaboración propia con tablas de Microsoft Word 2010.

## 5. IMPLEMENTACIÓN DEL SISTEMA

### 5.1. Criterio de selección de las herramientas

La herramienta seleccionada para la realización de este software es Android. La llegada de los teléfonos inteligentes ha revolucionado la forma de vivir de las personas, en especial Android, ya que es el sistema operativo líder en el mercado de dispositivos móviles. Según el proveedor de inteligencia de datos tecnológicos IDC (*International Data Corporation*), en 2015 este sistema contaba con el 87,6 % de la población de smartphome activos, tal y como se puede apreciar en la tabla VIII.

Tabla VIII. **Comparativa de uso de Smartphone**

Period	Android	iOS	Windows Phone	Others
2015Q3	84.3%	13.4%	1.8%	0.5%
2015Q4	79.6%	18.6%	1.2%	0.5%
2016Q1	83.4%	15.4%	0.8%	0.4%
2016Q2	87.6%	11.7%	0.4%	0.3%

Fuente: IDC, *Worldwide Smartphone OS Market Share*,  
<http://www.idc.com/prodserv/smartphone-os-market-share.jsp>. Consulta: 03 de octubre del 2016.

Una de las principales características de Android es la constante liberación de aplicaciones, ya que permite publicar aplicaciones de manera sencilla a cualquier usuario, esto provoca un gran catálogo de aplicaciones de todo tipo y para diferentes gustos, según el portal oficial de desarrolladores, Android

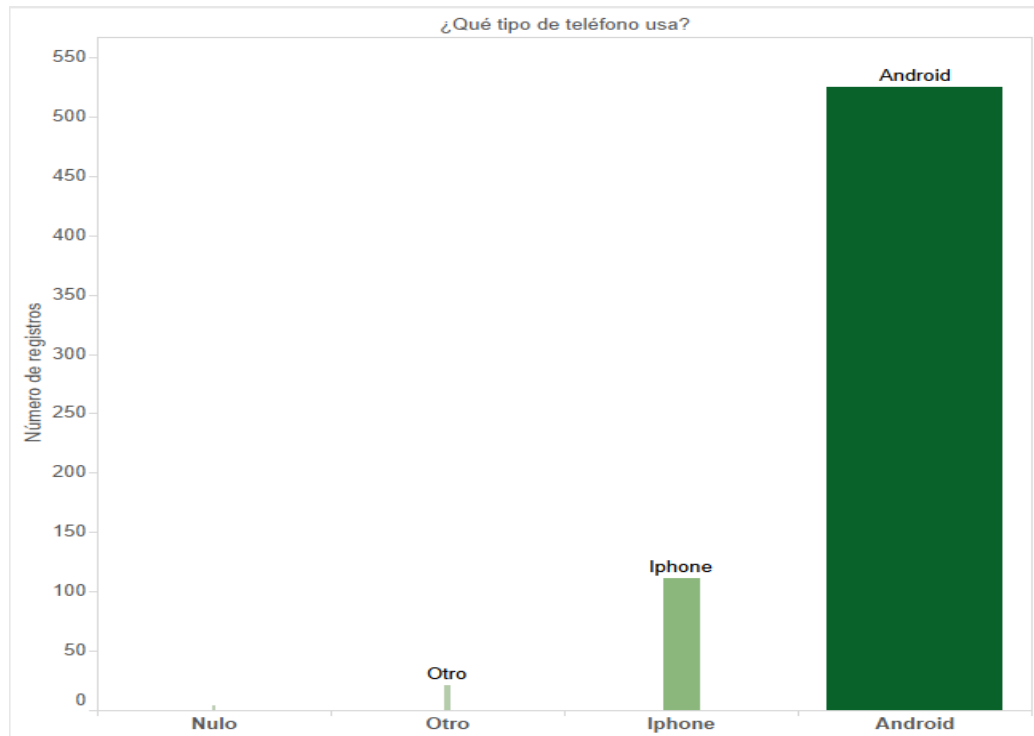


Develop: “*Android users download more than 1,5 billion apps and games from Google Play each month*”, los usuarios de Android descargan aplicaciones de forma masiva y muchas de ellas son gratis.

Para corroborar que los resultados de la encuesta de IDC fueran aplicables al segmento de usuarios que usarán la aplicación, es decir, los estudiantes de FARUSAC; se hizo una encuesta (Anexo 1) sobre una muestra de 661 estudiantes para averiguar cuál es el porcentaje que utilizan el sistema operativo Android. Los resultados fueron los siguientes:

- El 79 % de las personas utiliza Android
- El 17 % de las personas utiliza iOS
- El 3 % de las personas utiliza otro sistema operativo
- El 1 % de las personas no contestó a la pregunta

Figura 15. **Gráfica de uso de diferentes sistemas operativos para celular en FARUSAC**



Fuente: elaboración propia con el programa Tableau.

También se evaluó el nivel de interés de los estudiantes encuestados en cuanto a que la aplicación se lleve a cabo, los resultados se muestran en la tabla IX. Se separaron en términos del sistema operativo que usa el estudiante. Los niveles de aceptación se midieron en términos de cuatro respuestas siendo estas: “Me encantaría”, esta medida no solo expresa interés por la aplicación sino que también se expresa gusto por la misma. “Sería interesante”, en esta medida expresa interés por la aplicación. Estas dos se consideran aceptables.

“Me da igual” es la que se designó para no mostrar interés por la aplicación y “No me gustaría” es la que se selecciona para expresar disgusto

por la aplicación, estos dos últimos niveles se consideran como no aceptables para que la aplicación se realice.

**Tabla IX. Nivel de interés en que la aplicación se lleve a cabo en términos del sistema operativo que usa en su celular el estudiante**

¿Qué tipo de teléfono usa?	¿Te gustaría una a...	Número de registros
Otro	Nulo	1
Iphone	Nulo	1
Android	Nulo	8
Nulo	Nulo	2
Otro	Me da igual	3
Iphone	Me da igual	4
Android	Me da igual	10
Otro	Me encataría	5
Iphone	Me encataría	48
Android	Me encataría	208
Nulo	Me encataría	1
Iphone	No me gustaría	3
Android	No me gustaría	15
Otro	Sería interesante	12
Iphone	Sería interesante	55
Android	Sería interesante	284
Nulo	Sería interesante	1

Fuente: elaboración propia con el programa Tableau.

Al 41 % de los estudiantes que usan Android le encantaría la idea de tener la aplicación y al 56 % parecería interesante la idea, es decir se tiene un 97 % de aceptación sobre el segmento de usuarios objetivo.

- **Diversidad de dispositivos en la plataforma Android**

Uno de los principales retos cuando se hacen aplicaciones para la plataforma Android es la diversidad de dispositivos, ya que estos tienen muchas variantes dependiendo de muchos factores tales como el fabricante del

dispositivo, el modelo del dispositivo, la resolución de la pantalla del dispositivo y sobre todo el API de Google que posee el dispositivo.

Google ha lanzado 25 APIs para la plataforma Android hasta el año 2017. En Guatemala las APIs más usadas están entre el API 19 al 24, y las marcas de teléfono más usadas son Huawei, Samsung, LG y Sony, es por eso que las pruebas de la aplicación que se desarrollaron en el capítulo 6 de este documento van enfocadas en dispositivos con las características mencionadas anteriormente.

Cuando se trabaja con dispositivos móviles es indispensable el manejo de la batería y el consumo de datos del dispositivo, además de la rapidez de la aplicación. Estos aspectos deben tomarse en cuenta aunque el usuario nunca pueda tener una percepción directa de ellos ya que si estos existen pueden causar malestar en los usuarios y el desuso de la aplicación.

## **5.2. Servidor y aplicación móvil**

Cuando se habla de mensajería instantánea, el primer aspecto por tomar en cuenta en el servidor es “el tiempo real en la aplicación”. Esta es una tarea sumamente difícil debido a que la conexión debe mantenerse abierta siempre para que los mensajes puedan ser notificados en cualquier momento del día. Pero... ¿por qué se necesita una conexión abierta siempre?, ¿qué implica mantener una conexión abierta en un dispositivo móvil?

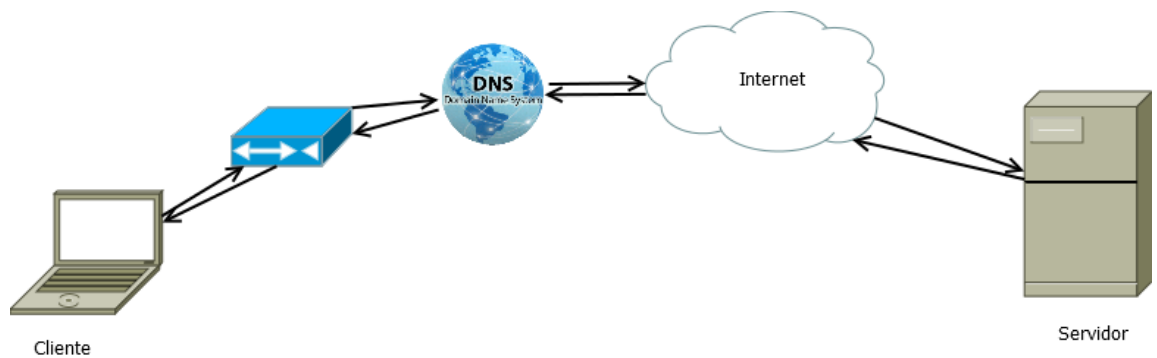
Para responder a estas preguntas fueron necesarias tres versiones del servidor y adaptar la aplicación a estos cambios, justamente es allí donde radica la información más valiosa de este documento. Es importante mencionar

que cada nueva versión de la aplicación corrige no solo a su antecesora sino a todas las anteriores.

### 5.2.1. Versión uno del servidor, PHP

Para la primera versión de “MIA”, se intentó utilizar un el lenguaje de programación PHP en el servidor, a través de una comunicación con protocolo Http. Este protocolo es el que utilizan las páginas web para enviar y recibir información y de esa forma interactuar con el usuario. A simple vista esto no podría acarrear ningún problema. Entonces ¿cuál es la desventaja de hacer un programa de mensajería instantánea siguiendo el modelo de interacción de una página web? Antes de contestar la pregunta anterior, se debe comprender qué se hace en realidad cuando se visita una página web, por lo menos, de una forma muy superficial para comprender el concepto (véase figura 16)

Figura 16. Procedimiento superficial para entrar a una página web



Fuente: elaboración propia con el programa DIA.

La figura 16 muestra el proceso para entrar a una página web de manera muy superficial, es decir, muchos de los procesos reales fueron omitidos en el diagrama para darle simplicidad a la explicación. Primero, suponiendo que un usuario desea entrar al dominio “http://www.google.com”, la petición sale de su computadora a través del *router* con el que se conecta al internet. La petición está estructurada a través del protocolo Http en donde viaja información

importante para que esta se lleve a cabo, por ejemplo, la dirección IP de la computadora que está haciendo la petición.

Cuando la petición sale del *router* sale hasta llegar al DNS (*Domain Name Service*) el cuál se encarga de identificar a qué dirección IP corresponde el nombre de dominio al cual el usuario está intentando acceder. Cuando la dirección IP del dominio es obtenida, la petición viaja por el internet hasta llegar al servidor que contiene la página que se está solicitando. El procesador despacha la solicitud y envía de regreso la información hacia la computadora que lo solicitó a través de su dirección IP que va en el encabezado de la petición Http.

En conclusión, el punto más importante que se debe rescatar de la explicación anterior es que el cliente sí puede iniciar una comunicación con el servidor, pero el servidor no puede iniciar una comunicación con el cliente. Esto se debe a que el servidor necesita la dirección IP del cliente para comunicarse con él y esto solo lo puede lograr a través de una petición por parte del cliente.

- **Notificaciones Push**

Ahora que ya está definido todo lo anterior, se puede retomar el concepto de mensajería instantánea, el objetivo de estas es que la información llegue de forma rápida al usuario y la forma más rápida de hacer llegar información en un dispositivo móvil es a través de las notificaciones push. Esta se define como una notificación que no necesita esperar a ser pedida por el usuario, sino que al contrario, la notificación llega automáticamente al dispositivo sin ninguna espera, de esa forma se puede hacer mucho más eficiente el flujo de la información urgente y de primera mano. Hay varias formas de hacer llegar la

notificación al usuario (sección 5.4), pero el principal objetivo es llamar la atención de este para que pueda obtener la información lo más pronto posible.

Hay un gran problema entre la forma de trabajar de PHP y la de las notificaciones push, un servidor con PHP necesita una petición del usuario para que este pueda interactuar con el cliente, pero una notificación push debe llegar al usuario automáticamente.

Sin duda ese fue un gran problema para la primera versión de la aplicación, sin embargo, se obtuvo una solución, ¿qué tal si los dispositivos móviles hacían peticiones cada 10 segundos al servidor para verificar si tenían alguna actualización? Sin duda, parecía una buena solución porque el usuario no tenía que interactuar con el dispositivo para obtener su nueva información y de esa forma el servidor en cada nueva petición podía obtener la dirección IP del dispositivo y en caso de tener nueva información este podía enviársela.

Sin duda la primera versión no fue la mejor solución para el problema, varios errores fueron detectados (véase sección 6.2.1) y mejorados en la versión 2.

### **5.2.2. Versión dos del servidor, Socket IO y Node JS.**

La versión 2 del servidor tuvo como objetivo solventar los problemas de consumo de datos de la aplicación, se hizo una investigación sobre alguna herramienta que tuviera un modelo de trabajo para tareas “en tiempo real” ya que como se explicó en la sección pasada el modelo de PHP no fue el mejor para trabajar las notificaciones push.



La tecnología que más se adecuaba a la tarea fue Socket IO, este es un *framework* que provee Node JS y sirve precisamente para hacer aplicaciones en “tiempo real”. Además, esta es una tecnología orientada a eventos lo cual es beneficioso para la aplicación ya que al detectar el evento de envío de un mensaje se puede activar una función para notificar a todos los receptores.

El lenguaje de programación que utiliza Node JS es JavaScript, además node es asíncrono, es decir, tiene varios hilos que se ejecutan paralelamente, esto es muy beneficioso ya que permite servidores escalables.

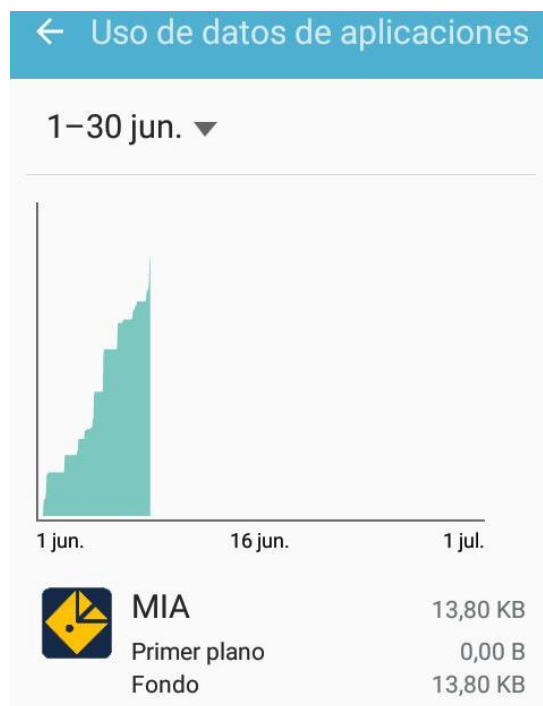
Socket IO sería el *framework* que se usaría para darle a la aplicación el tiempo real que necesita, la gran ventaja de esta tecnología es que el servidor no necesita una petición para poder enviarle datos al cliente. Esto definitivamente fue de gran ayuda ya que evita hacer peticiones en un intervalo de tiempo para capturar datos y aun mejor no hay que esperar ese intervalo para actualizar la información. La mayor virtud de Socket IO es que las notificaciones son instantáneas ya que su intervalo de actualización tiende a cero, debido a que su latencia es menor a un segundo, claro, esto depende de la conexión a internet y de otros aspectos. Socket IO también provee su *framework* para implementar un cliente en la plataforma Android, por lo que la tecnología se adapta perfectamente a la solución.

Para lograr que el servidor envíe información al dispositivo sin ninguna petición, el servidor debe registrar la dirección con la que se hizo la conexión y el dispositivo debe permanecer conectado al servidor permanentemente para que este pueda usar la dirección y enviarle información en cualquier momento.

Luego de que la versión 2 del servidor fuera implementada y adecuada a la plataforma Android se obtuvieron resultados muy favorables en cuando al

consumo de datos reduciendo estos hasta 13mb en cinco días aproximadamente, mientras que con la versión 1 se estimó un consumo 864mb en el mismo lapso, mejorando la tasa de consumo en un 66,4 % aproximadamente.

Figura 17. **Consumo de datos de MIA en una semana de uso**



Fuente: captura de pantalla en un dispositivo con sistema operativo Android.

- *Heartbeat*

Se le conoce como *heartbeat* a las señales que el dispositivo envía constantemente al servidor para mantener una conexión activa en todo momento, por lo tanto, si un cliente no está activo el servidor no podrá enviarle ninguna información, en este caso no podrá enviarle notificaciones. Para lograr una conexión activa permanente el dispositivo necesita enviar señales al servidor indicándole que este se encuentra activo y el servidor le responde que la señal está recibida. Lo ventajoso de esto es que la información que se envía en cada *heartbeat* es mínima ahorrando considerablemente los datos que se consumen. Cuando el dispositivo se encuentra sin internet por alguna razón o con una mala señal de internet, el intervalo del *heartbeat* se intensifica hasta lograr la reconexión nuevamente.

En conclusión, la tecnología elegida fue muy favorable ya que cuando esta versión de la aplicación fue publicada en fase beta para ser probada por la comunidad de “testers”, los resultados fueron muy favorables. También los tiempos de llegada de las notificaciones fueron muy cortos, las notificaciones fueron inestables y el gasto de batería subió considerablemente (véase sección 6.2.1).

### **5.2.3. Versión tres del servidor, *Firestore***

En esta versión del servidor se trabajaron los problemas abordados en la sección 6.2.2 y se le dio una solución al consumo excesivo de batería. Estos cambios en la arquitectura del servidor implican cambios sobre la implementación de la aplicación móvil ya que también fue necesario rediseñarla de modo que fuera compatible con esta versión del servidor.

#### **5.2.3.1. *Firestore***

*Firestore* es una tecnología proporcionada por Google, el concepto de esta es muy amplio, ya que cuenta con muchos servicios que ayudan al programador a hacer sus aplicaciones de manera óptima y sencilla. Algunos de sus servicios se cobran dependiendo del nivel de consumo en kb que estos tengan, algunas otras de la cantidad de veces que se usen, y algunos otros son gratis.

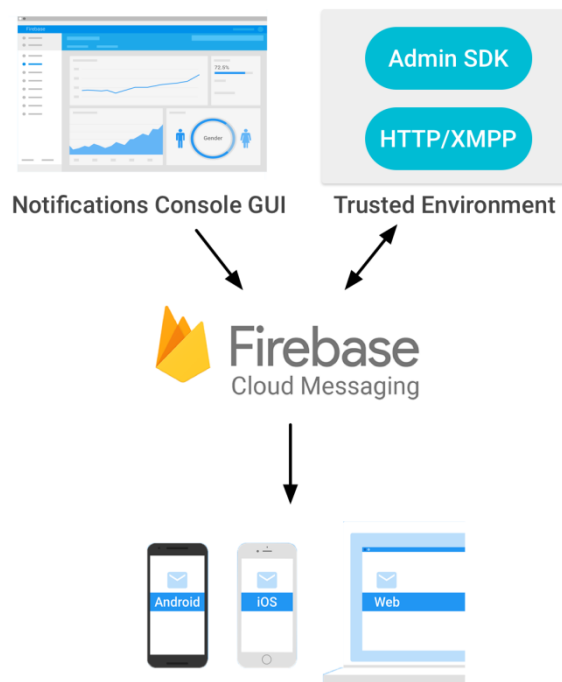
Los servicios que provee *Firestore* son los siguientes: *Analytics, Cloud Messaging, Cloud Functions, Authentication, Realtime Database, Storage, Hosting, Test Lab* y *Crash Reporting*. Cada servicio tiene una funcionalidad en específico con un conjunto de herramientas para cumplir con distintas funcionalidades, el uso de algún servicio no implica que se tengan que utilizar los demás servicios.

Después de una investigación sobre qué tecnología se puede utilizar para evitar el consumo excesivo de datos o de batería se llegó a la conclusión de que *Firestore Cloud Messaging*, uno de los servicios que provee *Firestore*, está especializado en la entrega de notificaciones push de manera óptima y eficaz.

- *Firebase Cloud Messaging*

Es uno de los servicios que provee *Firebase*, este se especializa en el envío de notificaciones push a dispositivos móviles, computadoras y tabletas, además también es multiplataforma ya que iOS, Android y plataformas web son compatibles con él.

Figura 18. Diagrama general de *Firebase Cloud Messaging*



Fuente: página web oficial de *Firebase*.

Al analizar la arquitectura de *Firebase* que se muestra en la figura 18, en la parte superior se puede observar de lado izquierdo el GUI. A este se puede acceder desde la consola de *Firebase* en su plataforma web; en este se pueden

observar algunos datos que registra FCM, por ejemplo, cuántos clientes le dan clic a las notificaciones, etcétera.

Siguiendo con la parte superior, de lado derecho se puede observar el API del servicio, es decir, todas las funcionalidades de FCM pueden ser llamadas a través de este API. En la llamada se pueden personalizar diferentes aspectos de la notificación push, por ejemplo, qué sonido tiene que ser reproducido al enviar una notificación, qué imagen, qué título y qué cuerpo debe llevar la notificación.

Por supuesto, hay muchas otras opciones que se pueden enviar a FCM cuando se desea enviar una notificación, pero los esenciales son los descritos anteriormente. Este API es consumido a través de un protocolo Http. A diferencia de la versión 1 del servidor este protocolo no servirá para actualizar información, sino únicamente para indicarle a FCM que se desea enviar una notificación.

Por último, en la parte inferior del diagrama se pueden observar los dispositivos destino, a los cuales les será entregada la notificación. Un SDK es introducido en la aplicación en la que se desea recibir mensajes, se crea un servicio en segundo plano para recibir los mensajes y para luego representarlos en la pantalla.

FCM genera un token único por cada dispositivo en el que se instala la aplicación móvil, estos van a su servidor para ser identificados y poder enviarles directamente las notificaciones push.

En otras palabras, la arquitectura que se definió en la sección 4.1 se verá afectada debido a que ahora se cuenta con un sistema intermedio (*middleware*) entre los dispositivos clientes y el servidor.

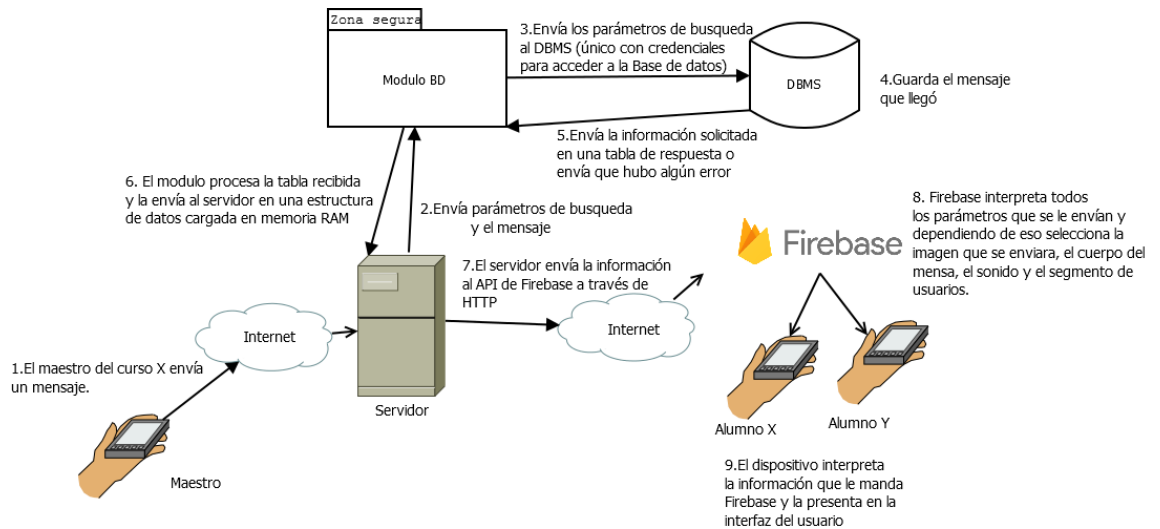
Luego de que la nueva arquitectura se implementara ,tanto en el servidor como en la aplicación móvil, esta se pasó a fase beta para hacer pruebas. Efectivamente, ningún dispositivo notificó alto consumo de batería, tampoco se notificó sobre alto consumo de datos. Esto se logró debido a que con FCM la aplicación únicamente está activa cuando el usuario está navegando dentro de ella o cuando FCM la despierta para mostrar alguna notificación push. Aunque el sistema mejoró mucho conforme fue evolucionando esta última versión presentó únicamente inestabilidad en la recepción de mensajes en algunos dispositivos móviles. (Véase sección 6.2.3)

### **5.3. Gestión de la información y envío de mensajes**

La información que se maneja en el servidor es completamente relacional para conservar la integridad de la información en alumnos, cursos, mensajes y estudiantes (capítulo 3), para gestionar esta información esta debe ser extraída del DBMS y manipulada por el servidor para ser procesada según sean las necesidades de este.

Además, esta extracción de datos debe hacerse de forma segura (sección 6.2) a través de un módulo que sea consumido por el servidor, siendo el módulo el que tenga contacto directo con el DBMS, el servidor en Node JS debe recibir y enviar mensajes a los usuarios. Por ejemplo, en la figura 19 se puede ver la gestión de información en el caso en el que un maestro envía mensajes a sus alumnos de determinado curso.

Figura 19. **Gráfica de uso de diferentes sistemas operativos para celular en FARUSAC**



Fuente: elaboración propia con el programa DIA.

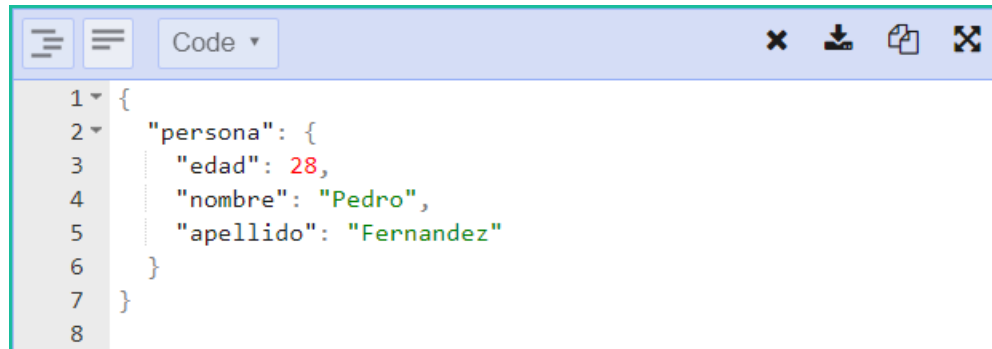
Algo muy importante de mencionar es que tanto el servidor como los clientes “no hablan el mismo idioma”, es decir, están hechos en distintos lenguajes de programación, por lo tanto, es necesaria una estandarización en los mensajes para que estos puedan interpretarlos y saber qué es lo que el otro le quiere solicitar.

JSON es un formato para el intercambio de datos entre diferentes dispositivos. La principal característica de JSON es que el formato de los mensajes es con los atributos “clave-valor”, es decir, a cada elemento que se desee enviar se le asigna un nombre (clave) y un valor, cuando se desea obtener algún valor se hace por medio de su clave.

Hay muchos tipos de valor entre los más usados están las cadenas, los enteros, objetos de tipo JSON, entre otros.



Figura 20. **Ejemplo de documento en formato JSON**



```
1 {  
2   "persona": {  
3     "edad": 28,  
4     "nombre": "Pedro",  
5     "apellido": "Fernandez"  
6   }  
7 }  
8
```

Fuente: elaboración propia con la aplicación JSON Formatter.

En el ejemplo anterior se puede observar un documento en formato JSON para una persona, su valor es otro objeto JSON en el cual se encuentran los elementos: edad que tiene como valor un entero, mientras que nombre y apellido tienen como valor cadenas.

#### **5.4. Notificaciones al usuario**

Las notificaciones de nuevos mensajes juegan un papel muy importante cuando se habla de mensajería instantánea, ya que el objetivo de estas es captar la atención del usuario en el instante que llega un mensaje para que este pueda leerlo lo más pronto posible. Hay varias formas de notificar a los usuarios que tienen un nuevo mensaje. Regularmente estas van acompañadas de un sonido y una vibración distintiva para que el usuario sepa que tiene una nueva notificación de determinada aplicación, si el dispositivo móvil cuenta con un led, las notificaciones también pueden activarlo.

#### **5.4.1. Barra de estado**

La forma más utilizada para reportar las notificaciones es la “barra de estado” (figura 3) ya que es la manera más rápida de acceso a la información. Esta barra siempre se encuentra contenida en la parte superior de la pantalla de los dispositivos, al deslizar el dedo hacia abajo la barra de estado se despliega y aparecen las notificaciones y todas las aplicaciones. Regularmente el contenido de estas está dado por un título, un ícono que las caracteriza y un texto más pequeño que el título con el contenido del mensaje que se le quiere enviar al usuario.

En la aplicación implementada en este documento se utiliza como título “Mensajería FARUSAC”, el contenido del mensaje son los últimos cinco mensajes que el usuario recibió identificados, por el nombre y la sección del curso en el que se recibió el mensaje y el contenido del mensaje, por último ,el ícono de FARUSAC.

Figura 21. **Notificación de la aplicación MIA en la barra de estado**



Fuente: captura de pantalla en un dispositivo con sistema operativo Android.

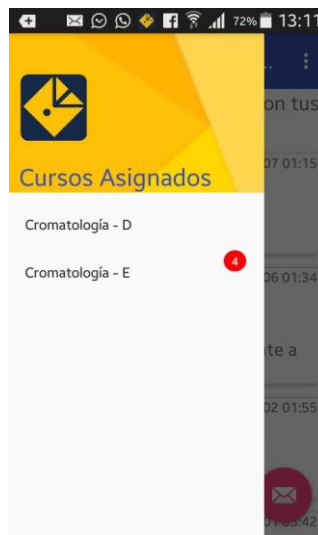
#### 5.4.2. **Ícono de número de mensajes y “ticker”**

El “ticker” es un mensaje que aparece en la barra de estado cuando este está en modo contenido, el texto debe expresar información importante para captar la atención del usuario y acceder inmediatamente a ella.

En “mensajería FARUSAC” la información que expresará el “ticker” es el nombre, sección del curso que mandó el último mensaje, así como también el mensaje.

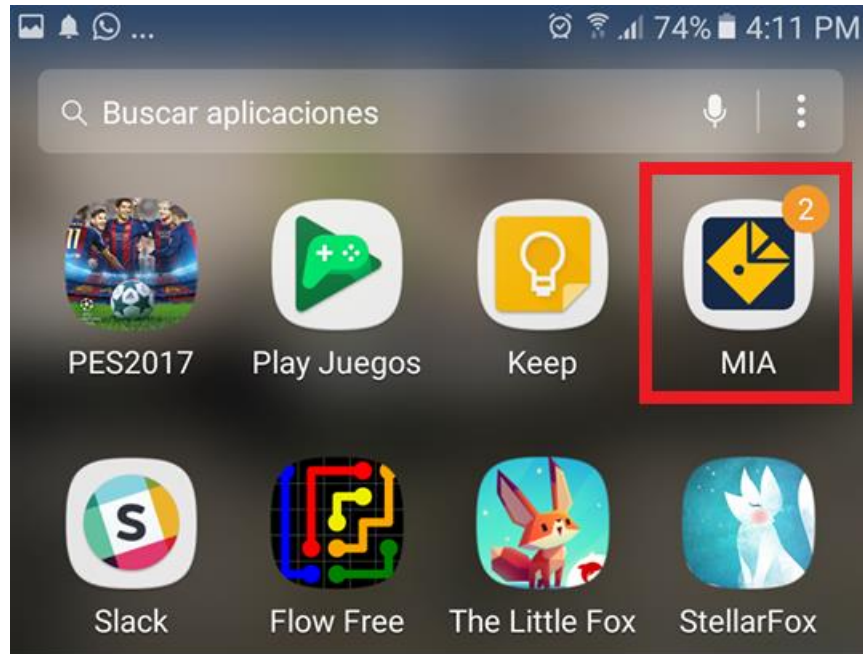
El “ticker” y las notificaciones push en la barra de estado son notificaciones para cuando el usuario se encuentre fuera de la aplicación, cuando el usuario está usando la aplicación la manera de notificar es diferente. Se observa un ícono color rojo en el cuál aparece la cantidad de mensajes de cada sala en la barra de cursos (sección 5.5.1). Si casualmente un mensaje llega mientras la barra está activa este se refresca en tiempo real para notificarle al usuario de un nuevo mensaje.

Figura 22. **Ícono de nuevos mensajes dentro de la aplicación**



Fuente: captura de pantalla en un dispositivo con sistema operativo Android.

Figura 23. **Ícono de nuevos mensajes fuera de la aplicación**



Fuente: captura de pantalla en un dispositivo con sistema operativo Android.

#### **5.4.3. Notificaciones en tiempo real dentro de una sala de chat**

El último tipo de notificación que se contempló, es la visualización de un mensaje cuando un alumno está dentro de una sala de chat de algún curso. Se debe poder notar la diferencia entre el nuevo mensaje y los mensajes que ya fueron leídos, tal y como se ilustra en la figura 24.

Figura 24. **Nuevos mensajes en una sala de chat de un curso**



Fuente: captura de pantalla de la aplicación MIA en un dispositivo con Android.

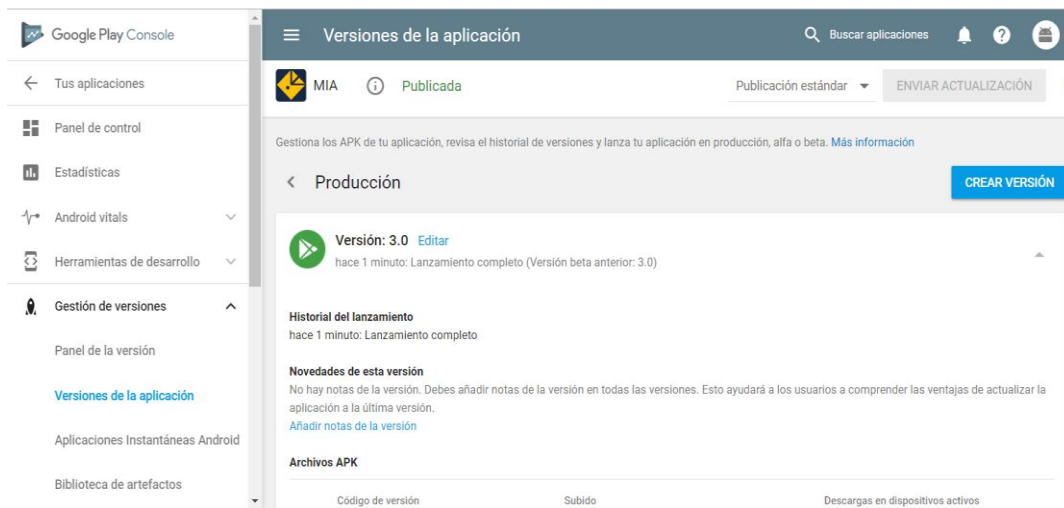
## 5.5. **Publicación de la aplicación móvil**

Como se mencionó en la sección 5.1 Play Store se llama la tienda oficial de Google y sirve para divulgar aplicaciones Android, estas pueden ser gratuitas o de paga. Luego de que la aplicación superara la fase de pruebas, se consideró como estable y lista para liberarla a producción en Play Store. La aplicación fue clasificación para adolescentes y de tipo gratuito, además solo

estará disponible en la República de Guatemala. Esta medida se tomó ya que se supone que solo alumnos de FARUSAC pueden descargarla, por lo tanto, no tendría sentido publicarla en otros países.

En Play Store además de publicar las aplicaciones también se tiene acceso a una consola donde se pueden supervisar los comentarios que esta ha tenido, la valoración que le dan los usuarios, los errores que se han producido y en qué dispositivo sucedió, esto se conoce como Android Vitals. En la figura 25 se puede observar una captura de pantalla de la consola de Play Store.

Figura 25. **Consola de *Play Store***



Fuente: captura de pantalla de la versión web de *Google Play Console*.

- Además en la consola también se encuentran otras funcionalidades, en la pestaña de “Gestión de versiones” se pueden administrar la aplicación en las siguientes fases:

- Alpha: esta versión es conocida únicamente por el equipo de desarrollo, se pueden probar algunas funcionalidades, no obstante, no todas las funciones tienen que estar terminadas forzosamente.
- Beta: esta fase inicia cuando se termina la implementación de la aplicación y el equipo cree que podría salir a producción, pero todavía no está probada por usuarios sino solo por desarrolladores. La aplicación se publica únicamente para personas seleccionadas para que puedan probar las diferentes funcionalidades, estos son conocidos como beta testers.
- Producción: es la versión final de la aplicación, esta puede ser vista por cualquier persona que entre a la Google Play Store, teniendo en cuenta las políticas de restricción del país.

Figura 26. **Gestión de versiones en Google Play Console**



Fuente: captura de pantalla de la versión web de *Google Play Console*.



Google Play también posee estadísticas muy interesantes de las aplicaciones, por ejemplo, se pueden observar gráficos con información desinstalaciones, instalaciones, errores, uso por país cuando las aplicaciones son liberadas para distintos países, entre otros.

Hay otras funcionalidades más específicas a las que se puede acceder, por ejemplo, se pueden administrar productos digitales para comprar dentro de la aplicación y suscripciones con paga mensual, estas no son aplicables para el tipo de aplicación del que se habla en este documento. La consola únicamente puede ser accedida por administradores de aplicaciones, no por usuarios normales que únicamente entran a descargar.

Figura 27. **Aplicación móvil MIA para descarga en *Play Store***



Fuente: captura de pantalla de *Play Store* en su versión web.

## 6. EVALUACIÓN DEL SISTEMA

En el último capítulo se evaluarán algunos de los requerimientos no funcionales que se consideran indispensables para “Mensajería FARUSAC”. En el libro *Ingeniería del software* se lee: “los requerimientos no funcionales, como su nombre sugiere, son aquellos que no se refieren directamente a las funciones específicas que proporciona el sistema, sino a las propiedades emergentes de este como la fiabilidad, el tiempo de respuesta o la capacidad de almacenamiento”<sup>6</sup>. Por lo tanto, los requerimientos no afectan directamente las funcionalidades de la aplicación, pero sí las afectan indirectamente, ya que si no se manejan de forma adecuada estos podrían repercutir en largo, mediano o corto plazo al software.

Idealmente todos los requerimientos no funcionales deben ser tomados en cuenta para la implementación de un software, pero estos pueden variar dependiendo de qué sea lo que se esté haciendo. Por ejemplo, cuando se tiene un software con un número fijo de usuarios se puede obviar la escalabilidad del software, siendo este uno de los requerimientos no funcionales que podrían tomarse en cuenta.

---

<sup>6</sup> SOMMERVILLE, Ian. *Ingeniería del software*. Madrid : Pearson Educación S.A., 2005. 111p.

## **6.1. Requerimientos no funcionales**

En la implementación de “Mensajería FARUSAC” se tomaron en cuenta tres requerimientos no funcionales los cuales son de vital importancia para este software y que podrían repercutir en algún tiempo, estos son el rendimiento, la seguridad y la ergonomía de la aplicación.

### **6.1.1. Rendimiento**

El rendimiento del servidor es un requerimiento muy importante a tomar en cuenta, este es inversamente proporcional al número de usuarios que se conecten al servidor, es decir, entre más clientes con dispositivo Android se conecten al servidor, menor va a ser su rendimiento y procesará más lentamente las solicitudes que se le envíen. El segmento de usuarios para esta aplicación no tiene una varianza muy grande, por lo tanto, se puede tener una media de aproximadamente dos mil usuarios. El número va a variar dependiendo del número de estudiantes que ingresen a la aplicación en determinado año y semestre, pero su variación no va a ser en gran magnitud. En la figura 28 se puede observar una prueba de carga que se le hizo al servidor con concurrencia de trescientos usuarios conectados al mismo tiempo, cada uno haciendo una petición al servidor por segundo.

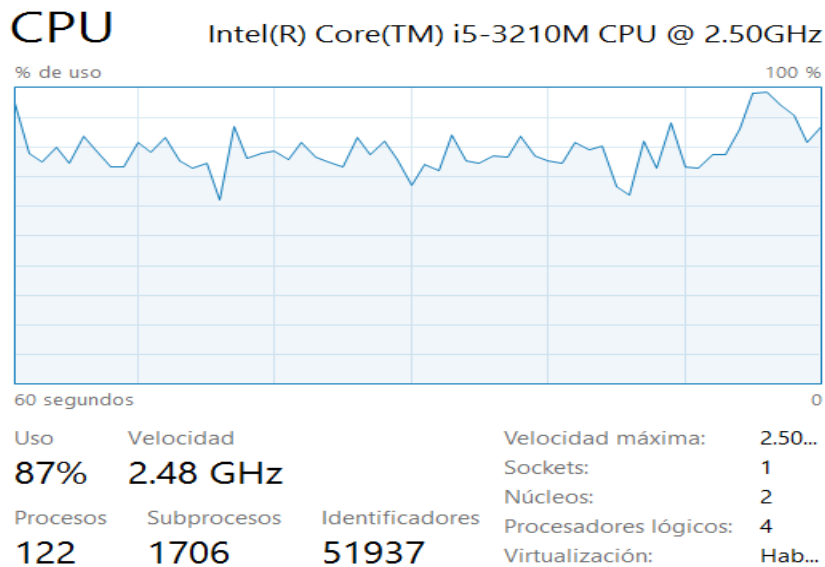
Figura 28. Prueba de estrés al servidor

```
loadtest -c 300 http://localhost:8081
mean latency: 287.5 ms
[Thu Jan 12 2017 01:40:58 GMT-0600 (Hora estándar, América Central)] INFO Requests: 159856, requests per second: 1134, m
ean latency: 264.7 ms
[Thu Jan 12 2017 01:41:03 GMT-0600 (Hora estándar, América Central)] INFO Requests: 165778, requests per second: 1176, m
ean latency: 254.9 ms
[Thu Jan 12 2017 01:41:08 GMT-0600 (Hora estándar, América Central)] INFO Requests: 170929, requests per second: 1038, m
ean latency: 287.4 ms
[Thu Jan 12 2017 01:41:13 GMT-0600 (Hora estándar, América Central)] INFO Requests: 176490, requests per second: 1115, m
ean latency: 270.7 ms
[Thu Jan 12 2017 01:41:18 GMT-0600 (Hora estándar, América Central)] INFO Requests: 181070, requests per second: 915, me
an latency: 323.1 ms
[Thu Jan 12 2017 01:41:23 GMT-0600 (Hora estándar, América Central)] INFO Requests: 185798, requests per second: 941, me
an latency: 321.9 ms
[Thu Jan 12 2017 01:41:28 GMT-0600 (Hora estándar, América Central)] INFO Requests: 189886, requests per second: 820, me
an latency: 367.3 ms
[Thu Jan 12 2017 01:41:33 GMT-0600 (Hora estándar, América Central)] INFO Requests: 195224, requests per second: 1070, m
ean latency: 290.2 ms
[Thu Jan 12 2017 01:41:38 GMT-0600 (Hora estándar, América Central)] INFO Requests: 200687, requests per second: 1093, m
ean latency: 275.3 ms
[Thu Jan 12 2017 01:41:43 GMT-0600 (Hora estándar, América Central)] INFO Requests: 204959, requests per second: 856, me
an latency: 348.2 ms
[Thu Jan 12 2017 01:41:48 GMT-0600 (Hora estándar, América Central)] INFO Requests: 210277, requests per second: 1041, m
ean latency: 289.1 ms
[Thu Jan 12 2017 01:41:53 GMT-0600 (Hora estándar, América Central)] INFO Requests: 215814, requests per second: 1126, m
ean latency: 262.1 ms
[Thu Jan 12 2017 01:41:58 GMT-0600 (Hora estándar, América Central)] INFO Requests: 220268, requests per second: 880, me
an latency: 335.7 ms
[Thu Jan 12 2017 01:42:03 GMT-0600 (Hora estándar, América Central)] INFO Requests: 224367, requests per second: 823, me
an latency: 376.5 ms
```

Fuente: captura de pantalla de la consola de comandos de Windows, usando el programa LoadTest para Node JS.

Para trescientos usuarios haciendo una petición por segundos se obtuvo una latencia de aproximadamente 0,3 segundos por solicitud, el cual es un tiempo muy bueno para responder, ya que es muy difícil que trescientos usuarios hagan una petición por segundo en el servidor. Si la población total es de dos mil personas, se estima que en un día de gran concurrencia, el mayor de usuarios conectados al mismo tiempo sea de cincuenta usuarios. En la figura 29 se puede observar el rendimiento del servidor tras recibir trescientas peticiones concurrentes por segundo. Las especificaciones del servidor se detallan en la imagen.

Figura 29. **Rendimiento del servidor recibiendo trescientas peticiones por segundo**



Fuente: captura de pantalla del administrador de tareas de Windows

El procesador llegó a ocuparse en un 87 % lo cual es una utilidad bastante grande, por lo mismo se empezaron a implementar técnicas para compartir la carga de tareas entre el servidor, la base de datos y los dispositivos móviles.

Se hizo uso de memoria estática en los dispositivos móviles para manejar información temporal y reutilizarla, con esto se logró reducir el número de peticiones al servidor y disminuir la carga.

Para evitar la sobrecarga de memoria RAM del servidor, se utilizaron disparadores en la base de datos, de modo que si hay operaciones dependientes en la base de datos sea ella misma quien se ocupe de estas tareas y el servidor no sea quien tenga que administrarlas y dar instrucciones innecesarias, debido a que cada vez que el servidor interactúa con la base de

datos este tiene que abrir una conexión y luego cerrarla, por lo tanto, es preferible que el DBMS se encargue de sus propios asuntos.

En el dispositivo móvil se manejaron las listas RecyclerView de Android, las cuales se caracterizan por reciclar las celdas que ya fueron vistas en la memoria cache del dispositivo, de modo que si es necesario volver a mostrarlas estén disponibles al instante, tomando en cuenta que la memoria caché es la más rápida para procesamiento y manejo de información en tiempo de ejecución.

### **6.1.2. Seguridad**

Cuando se habla de seguridad rápidamente se viene a la cabeza la base de datos, ya que es donde se concentra toda nuestra información importante, tales como la información de los usuarios el historial de mensajes. Por lo tanto, esta debe tener un acceso sumamente restringido y se deben implementar técnicas para asegurar que solo las personas autorizadas puedan entrar y hacer uso de la información.

Primero, la información de acceso a la base de datos está en un área restringida para que nadie pueda tener acceso a esos datos, ni siquiera entrando al servidor, ya que las credenciales para entrar a la base de datos se encuentran en un módulo aislado del servidor y, por lo tanto, alguien que haga una petición no tiene contacto directo con la base de datos.

Otro gran problema es la inyección de código SQL dentro de los mensajes, de esta forma poder hacer sub consultas sobre otros datos a los que el intruso no puede acceder. Para darle solución a este problema fue necesario el uso de consultas por parámetro, el cual tiene por objetivo aislar la consulta de

los parámetros de búsqueda, para que al momento de hacer la consulta los parámetros son clasificados uno por uno para verificar que no contengan inyecciones SQL.

### **6.1.3. Ergonomía**

Este requerimiento no funcional pertenece a la aplicación móvil, es decir, está del lado del cliente y se define como la facilidad que un usuario tiene para adaptarse a la aplicación móvil. La facilidad para usar aplicaciones solía ser una tarea bastante difícil en a comienzos de este siglo, muchas personas no entraban a las redes sociales porque decían que estas eran difíciles de usar y que no entendían su significado. Lo mismo pasó con Dropbox, las personas no usaban el almacenamiento en la nube, porque ni siquiera entendían el concepto.

La ergonomía es un requerimiento no funcional que muchas personas no toman en cuenta cuando implementan un software, pero es muy importante ya que si el usuario considera que no puede usarlo lo dejará de usar, el aprendizaje debe ser intuitivo y el usuario debe adaptarse lo más pronto posible a la aplicación.

“Mensajería FARUSAC” se implementó de manera que se ocuparan el menor número de pantallas, menús, botones que fueran posibles y enviar al usuario a la información de interés con el flujo más corto que se pudo encontrar.

La información se clasificó por sectores. Por ejemplo, toda la información de los cursos se puede encontrar en la barra de cursos (figura 22), las notificaciones son ya conocidas en el mundo de los Smartphone, ya que cualquiera que posee un teléfono inteligente seguramente ha usado el sistema

de notificaciones (figura 3) y, por último, la sección de noticias generales, las cuales aparecen al inicio de la aplicación. Por lo tanto, los usuarios para acceder a ellas únicamente deben presionar el botón de entrar.

La información personal se encuentra en la parte superior derecha de la aplicación, en esta se puede administrar la cuenta que está activa en el dispositivo, se cuenta con asignación de cursos. Dentro de esta pestaña aparecen todos los cursos que están cargados en el servidor, el usuario puede seleccionar los cursos de interés para recibir notificaciones sobre este. La última opción sirve para cerrar la sesión actual en el dispositivo, en caso de que ya no se quieran recibir notificaciones.

Adicional a lo anteriormente mencionado se implementó el uso de descubrimiento de características la primera vez que un usuario entra a la aplicación, así se trata de llamar la atención del usuario para mostrar las características que se consideran más importantes para la aplicación a través de animaciones llamativas y con de una breve descripción (véase figura 30).



Figura 30. **Descubrimiento de características**



Fuente: captura de pantalla de la aplicación MIA.

Con la menor cantidad posible de pantallas se pretende que el usuario pueda encontrar información deseada de forma sencilla, rápida y con pocos tap.

## **6.2. Detección y corrección de errores del sistema**

Cuando se habla de aplicaciones móviles la probabilidad de que se encuentren errores aumenta considerablemente ya que a diferencia de una aplicación para escritorio o una página web, el usuario puede percibir con más facilidad los aspectos no funcionales de la aplicación. Para un usuario es muy importante el gasto de la batería durante el día, si una aplicación gasta demasiada batería el usuario seguramente se dará cuenta rápidamente.

Otro de los aspectos más importantes cuando se habla de dispositivos móviles es el consumo de datos de un plan, ya que un usuario regular tiene

limitado el número de kb que puede gastar determinado lapso. Actualmente para un usuario es muy importante permanecer conectado todos los días debido a que las redes sociales, el correo, etcétera, se han convertido en una necesidad, por lo tanto, si los datos son malgastados por alguna aplicación, el usuario sin duda la desinstalará del dispositivo.

Todos estos aspectos serán abordados en este último capítulo. La información fue tomada de una muestra de usuarios que voluntariamente colaboraron con el testeo de la aplicación en su fase beta antes de ser publicada en la Play Store.

### **6.2.1. Alto consumo de datos móviles**

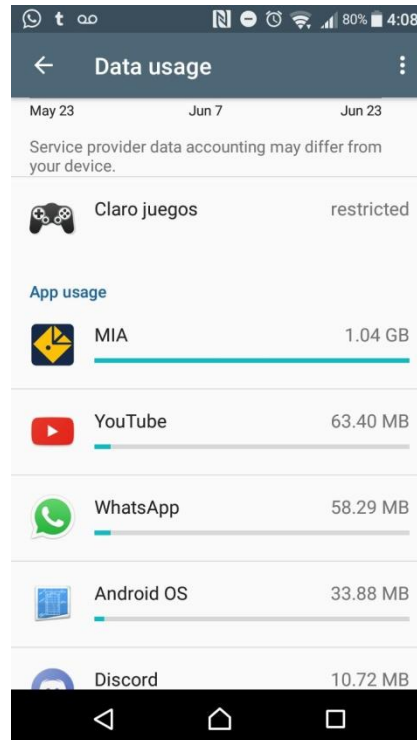
Este problema fue detectado en la versión uno de la aplicación, como se mencionó en el capítulo 5.2.1 el modelo de comunicación fue a través de peticiones hacia el servidor en un intervalo de tiempo de 10 segundos durante todo el tiempo mientras estuviera instalada la aplicación en el dispositivo.

“Tiempo real” no precisamente es que el mensaje llegará en el mismo instante en que se envió, debido a que todo sistema tiene un tiempo de latencia en el viaje de la información a través de la red, incluso hay tiempos de latencia de forma local en un dispositivo. Por ejemplo, el tiempo en el que viaja la información desde el disco duro al CPU es más grande en comparación con el tiempo de latencia que tarda en llegar la información de la memoria RAM al CPU. Sin embargo, tener que esperar 10 segundos para refrescar la información ¿en realidad es tiempo real? se podría pensar en bajar el intervalo de tiempo para refrescar la información, pero esto implica que se van a hacer más peticiones por minuto.

A continuación se hará un cálculo aproximado del gasto de datos durante una semana con un intervalo de tiempo de 5 segundos, si cada petición contiene toda la información de los encabezados más la información que se quiere enviar, se puede tener un aproximado de 10 kb por petición. A partir de la información anterior se puede decir que cada minuto se hacen 12 peticiones, por lo tanto, cada día se hacen 17 280 peticiones cada una con un peso de 10 kb, en un día un usuario estaría gastando 172 800 kb lo que corresponde a un aproximado de 173 mb. En cinco días serían gastados 864 mb aproximadamente, tomando en consideración que hay planes de datos móviles de 100 mb para gastar en 30 días, la solución queda totalmente anulada, esto podría pasar desapercibido en una página web, pero un dispositivo móvil esto queda en evidencia rápidamente.

Entre más se baje el intervalo de tiempo, más rápido se refrescará la información para el usuario, pero también será más consumo de sus datos, por lo tanto se tuvo que crear otra versión de la aplicación que contemplara este caso (véase sección 5.2.2). En la figura 31 se puede apreciar el consumo excesivo de datos de "MIA" en uno de los dispositivos donde se probó en fase beta.

Figura 31. **Consumo excesivo de datos de MIA en una semana de uso**



Fuente: captura de pantalla del administrador de datos en un dispositivo con Android.

### 6.2.2. **Alto consumo de batería**

La versión 2 del servidor fue implementada con la tecnología de Socket IO. Como consecuencia, se tuvo que implementar la misma tecnología del lado del dispositivo para que estos obtuvieran la compatibilidad necesaria. En la versión de Socket IO para dispositivos Android se pretende emular el ambiente orientado a eventos que brinda Node JS. Esto implica mantener un *heartbeat* (véase sección 5.2.2) del lado del dispositivo, y conexión activa. Posiblemente esto no tenga ninguna repercusión en una computadora debido a que están conectadas a una corriente de energía eléctrica y en el caso de las laptop poseen un batería con más capacidad que un dispositivo móvil.

Los tiempos de conexión activa son muy importantes cuando se desea utilizar Socket IO y esto no se tomó en cuenta cuando se eligió la tecnología. Por ejemplo, cuando un usuario desea entrar a una página para acceder a juegos donde hay varios participantes y tenga que jugar en tiempo real, el tiempo de conexión varía en qué tanto tiempo esté jugando el usuario, pero seguramente no será veinticuatro horas, los siete días de la semana. Por otro lado, cuando se trabajan notificaciones push éstas deben tener una conexión activa en todo momento, debido a que un maestro puede enviarle un mensaje a sus alumnos, cualquier día a cualquier hora y para que el servidor pueda notificar dicho mensaje en tiempo real, el dispositivo tiene que permanecer conectado siempre reduciendo el rendimiento en la batería notablemente.

Cuando se liberó la versión 2 en fase beta esta fue probada por los usuarios tester, lo descrito anteriormente tuvo sus consecuencias a tal punto que los dispositivos empezaron a reportar consumo excesivo de batería por parte de la aplicación llegando a consumir 150 % más que la aplicación que está por en segundo lugar de consumo, como se puede observar en la figura 32.

Figura 32. **Notificación sobre el uso excesivo de batería de MIA en Android**



Fuente: captura de pantalla en un dispositivo con sistema operativo Android.

Según lo explicado anteriormente, Socket IO trabaja muy bien cuando se van a desarrollar aplicaciones que son en tiempo real, pero hay que tomar muy en cuenta el tiempo que se necesita mantener activa la conexión y en qué dispositivo se va a implementar. Para un sistema de notificaciones push no es lo más indicado, debido a que la conexión debe permanecer activa siempre, esto ocasiona un número excesivo de *heartbeats* diarios y, en consecuencia, un consumo excesivo de batería, siendo esta muy valiosa en un dispositivo móvil.

Además de los problemas de batería, la aplicación también mostró serios problemas de estabilidad al momento de recibir las notificaciones debido a que mantener una conexión estable con los proveedores de internet móvil de

Guatemala en cada momento del día es una tarea casi imposible. Esto ocasiona que la conexión con el servidor se pierda y, por lo tanto, reconexión pueda durar hasta horas y, en consecuencia, las notificaciones no llegan al dispositivo instantáneamente.

### **6.2.3. Baja estabilidad en el envío de notificaciones**

La estabilidad de las notificaciones es una de las tareas más difíciles cuando se habla de mensajería instantánea, especialmente en la plataforma Android, debido a que depende directamente del dispositivo en el que esté instalada la aplicación, ya que dependiendo del fabricante, estos tienen diferentes tipos de gestionar la batería del dispositivo. Muchas veces en la gestión de la batería los fabricantes deciden bloquear las aplicaciones que no estén abiertas o deciden bloquear los procesos en segundo plano cuando el dispositivo está bloqueado. Esto perjudica directamente a la recepción de las notificaciones ya que estas, como se explicó en la sección 5.2, necesitan mantener una conexión abierta siempre.

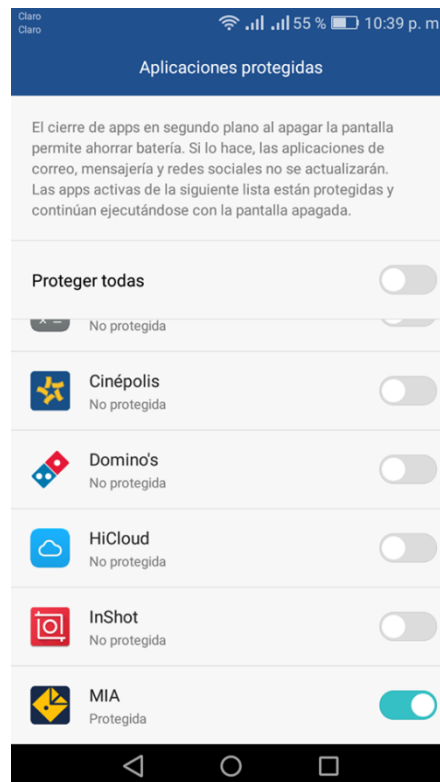
Otro aspecto importante por tomar en cuenta son las dependencias de los servicios con los que cuenta el dispositivo, por ejemplo, para el correcto funcionamiento de Firebase es necesario que el dispositivo cuente con una total disponibilidad de Google Play Services, el cual es un conjunto de servicios que Google proporciona para que sus herramientas funcionen correctamente y debido a que esta es una herramienta de Google, si no se cuenta con disponibilidad total de Google Play Service las notificaciones no llegarán.

Por último, se puede mencionar el caso particular de los dispositivos Huawei, estos bloquean las aplicaciones que usan servicios en segundo plano para ahorrar batería lo cual perjudica la llegada de notificaciones, sin embargo,

ellos proporciona la opción de “Aplicaciones Protegidas”, con esta opción es posible indicarle al dispositivo que la aplicación seleccionada como protegida deberá seguir con los procesos en segundo plano aunque esta se encuentre cerrada o el celular se encuentre bloqueado.

En pocas palabras la estabilidad de las notificaciones está ligada a los factores que implica la diversidad de dispositivos y sus diferentes fabricantes, APIs, etcétera.

Figura 33. **Aplicaciones protegidas**



Fuente: captura de pantalla en un dispositivo Huawei con sistema operativo Android.

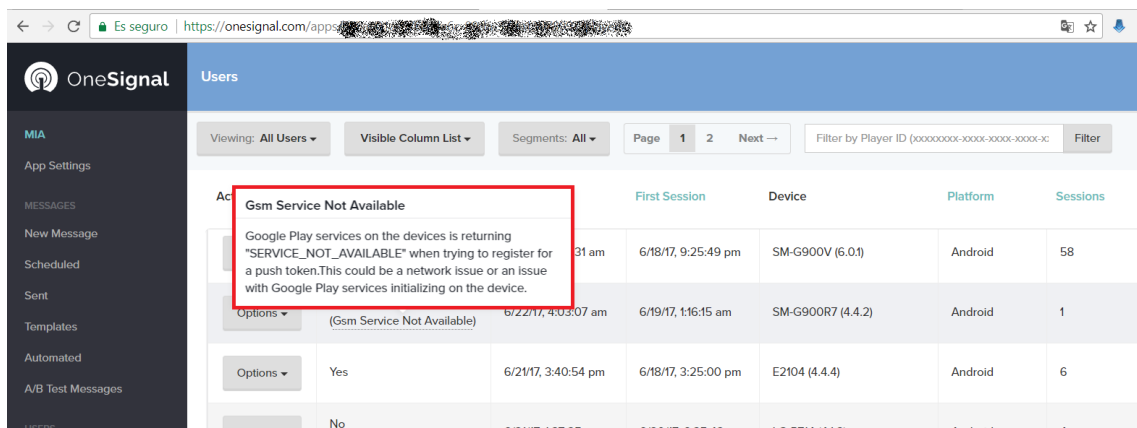


### 6.2.3.1. One Signal

*One Signal* solucionó la inestabilidad en la llegada de notificaciones push a los dispositivos, ya que esta tecnología funciona como un administrador de las notificaciones push que se envían con FCM. Su consola es mucho más completa, por cada mensaje que se envía se generan reportes por cada dispositivo y en caso de fallar la notificación se hacen reintentos hasta que el dispositivo destino permita la entrada de la notificación.

En caso de que un dispositivo definitivamente no permita la entrada de las notificaciones *One Signal* notifica en su consola la razón por la que no puede enviarle las notificaciones push a dicho dispositivo. Algunos ejemplos de razones por las que ya no pueden llegar las notificaciones de manera definitiva pueden ser porque el usuario ha desinstalado la aplicación, porque el usuario tiene bloqueada alguna dependencia de FCM, tal y como se puede observar en la figura 34.

Figura 34. Consola de *One Signal*



The screenshot shows the OneSignal Users console interface. A table lists user sessions with columns for 'Ac', 'First Session', 'Device', 'Platform', and 'Sessions'. One row is highlighted with a red box, indicating an error: 'Gsm Service Not Available'. The error message states: 'Google Play services on the devices is returning "SERVICE\_NOT\_AVAILABLE" when trying to register for a push token. This could be a network issue or an issue with Google Play services initializing on the device.'

Ac	First Session	Device	Platform	Sessions
Gsm Service Not Available Google Play services on the devices is returning "SERVICE_NOT_AVAILABLE" when trying to register for a push token. This could be a network issue or an issue with Google Play services initializing on the device.	6/18/17, 9:25:49 pm	SM-G900V (6.0.1)	Android	58
(Gsm Service Not Available)	6/19/17, 1:16:15 am	SM-G900R7 (4.4.2)	Android	1
Yes	6/21/17, 3:40:54 pm	E2104 (4.4.4)	Android	6
No	6/21/17, 4:37:25 pm	LG-P741 (4.1.2)	Android	4

Fuente: captura de pantalla de la consola web de *One Signal*.

## CONCLUSIONES

1. La implementación de mensajería instantánea reduce significativamente los tiempos en que la información llega al receptor, esto mejora el flujo de comunicación entre maestros y alumnos.
2. La implementación de un sistema de notificaciones "push" para dispositivos móviles con sistema operativo Android, es muy complicada debido a la diversidad de dispositivos y a que cada uno trabaja de forma diferente dependiendo de su fabricante, API de Android o resoluciones de pantalla, por ejemplo.
3. Los requerimientos no funcionales, tales como el consumo de batería y de datos móviles es muy importante cuando se trabajan aplicaciones para dispositivos móviles, ya que impacta directamente en la decisión de un usuario en mantener o desinstalar la aplicación.
4. Es muy importante que toda la información esté centralizada en el servidor debido a que si se desea cambiar, el cambio debe ser transparente para las aplicaciones, por ejemplo, agregar o quitar cursos.
5. El mejor flujo para brindar información de manera clara y directa es que los maestros tengan una vía de comunicación unilateral, directa y de primera mano con el alumno, de esa manera se puede evitar que se tergiverse la información y evitar los mensajes que no son deseados. También se debe poder enviar información masiva a todos los usuarios de la aplicación o a grupos grandes como todos los alumnos o todos los

docentes. Estos mensajes deben ser enviados por una entidad encargada de dar avisos como suspensión de clases o actividades generales.

## RECOMENDACIONES

1. Reducir el tiempo de conexión con el servidor cuando un usuario está dentro de la aplicación, ya que el sistema actual mantiene una conexión abierta desde que el usuario abre la aplicación hasta que él sale de ella. Sin embargo, esta característica ya no es necesaria debido a que se introdujo FCM dentro de la aplicación, así que lo correcto es mantener abierta la conexión únicamente cuando se hace la llamada para pedir información al servidor y cuando este responda, cerrar la conexión inmediatamente.
2. Implementar un sistema donde pueda haber transferencia de archivos multimedia tales como imágenes, archivos de texto o sonidos, ya que aportaría mucho a la aplicación. Se puede, además, explorar FCM porque permite enviar también este tipo de información.
3. Desarrollar la aplicación para otras plataformas móviles complementarias a Android que están actualmente en el mercado tales como iOS de Apple y Windows Phone de Microsoft.
4. La arquitectura del servidor puede ser escalable para que el rendimiento del servidor no se vea afectado a medida que crezcan los usuarios cuando la aplicación se desarrolle para los demás sistemas operativos.
5. La capacitación para los usuarios de la aplicación es importante para indicar los fundamentos por los cuales la aplicación fue creada y hacer

ver que el concepto de esta es para entregar información académica importante de forma rápida y eficaz, no para saludar, enviar chistes o enviar información que se puede hacer en una mensajería instantánea convencional.

## BIBLIOGRAFÍA

1. ADDISON, Wesley, *Introducción a los sistemas de bases de datos*. 7ª ed. México: PEARSON EDUCACIÓN S.A., 2000. 48 p.
2. CASTAÑO DÍEZ, Juan José. *Comercio electrónico*. España: EDITEX, 2016. 178 p.
3. IDC, *Worldwide Smartphone OS Market Share*, [en línea]. <http://www.idc.com/prodserv/smartphone-os-market-share.jsp>. [Consulta: 3 de octubre de 2016].
4. LANGEVIN HOGUE, Lise. *La comunicación: un arte que se aprende*. España: Sal Terrae, 2000. 13 p.
5. LÓPEZ, Alejandro, *Status bar notification*. [en línea]. <http://rootear.com/android/recupera-notificaciones-android>. [Consulta: 12 de junio de 2016].
6. OSORIO RIVERA, Fray León. *Bases de datos relacionales teoría y práctica*. Colombia: Instituto Tecnológico Metropolitano, 2008. 25 p.
7. PÉREZ PORTO Julián, GARDEY Ana. Definición.de. [en línea]. <https://definicion.de/tergiversar/>. [consulta: 28 de julio de 2017.]

8. SOMMERVILLE, Ian. *Ingeniería del software*. 7ª ed. Madrid: PEARSON EDUCACIÓN S.A., 2005. 111 p.

## ANEXOS

### Anexo 1. Encuesta para alumnos de FARUSAC

Encuesta App Mensajería instantánea  
FARUSAC 2016 para alumnos

¿Qué tipo de teléfono usa?

Android

Iphone

Otro

¿Te gustaría una aplicación de teléfono celular similar a Whatsapp, donde recibas información de tus docentes de forma instantánea y recibas notificaciones en tu teléfono para recordatorios de tareas, eventos, etc.?

Choose

Me encantaría

Sería interesante

Me da igual

No me gustaría

Google Forms

Fuente: elaboración propia con formularios de Google.



