

UNIVERSIDAD CATÓLICA DE LA SANTÍSIMA CONCEPCIÓN  
Facultad de Ingeniería  
Ingeniería Civil Informática



**UNIVERSIDAD CATOLICA  
DE LA SANTISIMA CONCEPCION**

## **Implementación de una teclera virtual sobre una plataforma Android para la UCSC.**

**Alex Heisinger Vivanco**

Informe de Proyecto de Título para optar al Título de  
Ingeniero Civil Informático

**Profesor Guía**  
Pedro Gómez M.

Concepción, Octubre 2015

# Resumen

Las tecleras son un sistema de respuesta inalámbrica, inmediata del estudiante de las cuales se distinguen dos modalidades, existen por una parte las tecleras físicas, que se desarrollan a partir de un dispositivo o control remoto exclusivamente destinado a responder preguntas que son proyectadas en una pantalla para un público en el contexto de una sala de clases. Y por otra parte, están las tecleras digitales, que son aquellas que pueden funcionar sobre un sistema operativo para dispositivos móviles. Este trabajo de investigación se avocará al desarrollo de estas últimas, entendidas en el sentido que ya se ha mencionado.

Las cualidades que presentan en común estos sistemas es que funcionan sobre la base de un profesor que propone una pregunta y sus alternativas a un conjunto de estudiantes, los que envían sus respuestas al moderador (profesor), para ser analizadas y entregar estadísticas con los resultados.

La finalidad de esta investigación es ofrecer una nueva alternativa al proyecto de tecleras físicas disponibles en el mercado actual. Este proyecto de título se enmarca en la realización de una aplicación Android con fines educacionales, que se sustenta en base a una arquitectura cliente-servidor y una conexión a una red WiFi, con el propósito de que alumnos puedan contestar preguntas realizadas por un profesor, para generar estadísticas de opiniones sobre algún tema específico, mejorar el proceso de enseñanza aprendizaje a través de la toma de decisiones conjunta en lo que respecta una asignatura.

Las ventajas del desarrollo de esta aplicación son principalmente el bajo costo, dado que la instalación es gratuita. A su vez, promueve la mejora en el rendimiento, por el hecho de incentivar la tecnología en la sala de clases y aumentar la atención de los alumnos, también sirve como herramienta para que los profesores puedan evaluar el aprendizaje de los alumnos. Por último destaca que es un sistema portable en términos prácticos porque se encuentra en dispositivos móviles con sistema operativo Android.

# Abstract

The tecleras are a system of immediate wireless student response, which two types are distinguished, there are on the one hand the physical tecleras, which develop from a device or remote control only intended to answer questions that are projected on a screen for an audience in the context of a classroom. On the other hand, there are digital tecleras, which are those that can run on an operating system for mobile devices. This investigation work will develop these two types, mentioned above.

The qualities that these systems have in common is that they work on a teacher's basis who proposes a question and alternatives to a group of students, who send their answers to the moderator (teacher), to be analysis and submit stadistics with the results.

The purpose of this research is to offer a new alternative to the draft physical tecleras available on the market today. This title project is focus on the realization of an Android application for educational purposes, which is sustain based on a client-server architecture and a connection to a WiFi network, with the purpose that students can answer questions from a teacher, to generate statistics of views on a specific topic, improve the teaching-learning process through joint decision-making on matters concerning a subject.

The advantages of the development of this application is mainly the low cost, because the installation is free. Also, it promotes the improvement in the performance, by the fact it encourages the technology in the classroom and increase the student's attention. Moreover serves as a tool for teachers to assess the student's learning. Finally it stresses that it is a portable system in practical terms because it is on mobile devices with Android operating system.

# Agradecimientos

Gracias a DIOS por guiarme en este camino y permitirme alcanzar este logro tan importante en mi vida, que quiero dedicar a mi familia, amistades, profesores y todas aquellas personas que en algún momento se han hecho partícipes de este proceso, brindándome el apoyo y ánimo necesarios para motivarme a cumplir mis metas y sueños.

Especialmente a mi madre Norma Vivanco León y a mi abuela Norma León Hernández, que son el pilar fundamental en mi vida y con su ejemplo me han enseñado las cosas más valiosas: el amor, el respeto, la humildad, la responsabilidad, la solidaridad, el esfuerzo y la perseverancia... valores que me han llevado a ser quien soy hoy en día.

También quiero mencionar a los padres de mi compañero Gustavo Inzunza que durante los años de carrera me acogieron en su casa, facilitando nuestro estudio. Además a Maritza González quien me acompañó durante más de tres años brindándome su apoyo y compañía en los peores momentos. Por último quiero agradecer a Francisca Valdebenito, que me ayudó con la ortografía y redacción de este informe y a Bastian Contreras por diseñar los logos de la aplicación.

**Alex Heisinger Vivanco**

# Índice general

<b>1. Introducción</b>	<b>1</b>
1.1. Presentación del Problema . . . . .	1
1.2. Justificación del Problema . . . . .	2
1.3. Delimitación del Problema . . . . .	3
1.4. Objetivo General . . . . .	3
1.5. Objetivos Específicos . . . . .	3
1.6. Metodología . . . . .	4
<b>2. Marco Teórico y Estado del Arte</b>	<b>6</b>
2.1. Marco Teórico . . . . .	6
2.1.1. Normas sobre Software e Ingeniería de Software . . . . .	7
2.1.1.1. Norma ISO 25000 . . . . .	7
2.1.1.2. Norma IEEE 830 . . . . .	7
2.1.2. Tecleras . . . . .	8
2.1.3. Tecleras Físicas . . . . .	10
2.1.4. Tecleras Digitales. . . . .	12
2.1.4.1. Computación Móvil. . . . .	13
2.1.4.2. Android . . . . .	13
2.1.4.3. Cliente-Servidor . . . . .	14
2.1.5. Conclusión . . . . .	15
2.2. Estado del Arte. . . . .	16
2.2.1. RESP . . . . .	16
2.2.2. ARS intranet . . . . .	16
2.2.3. ClickerSchool Clicker Virtual . . . . .	17
2.2.4. OpenARS . . . . .	19

<b>3. Especificación de Requisitos</b>	<b>21</b>
3.1. Introducción . . . . .	21
3.1.1. Propósito . . . . .	21
3.1.2. Ámbito del Sistema . . . . .	21
3.1.3. Vista General del Documento . . . . .	22
3.1.4. Referencias . . . . .	22
3.2. Descripción General . . . . .	22
3.2.1. Perspectiva del Producto . . . . .	22
3.2.2. Funciones del Producto . . . . .	22
3.2.3. Características de Usuarios . . . . .	23
3.2.4. Restricciones . . . . .	23
3.2.5. Suposiciones y Dependencias . . . . .	24
3.2.6. Requisitos comunes de las interfaces . . . . .	24
3.2.6.1. Interfaces de usuario . . . . .	24
3.2.6.2. Interfaces de hardware . . . . .	24
3.2.6.3. Interfaces de comunicación . . . . .	26
3.3. Requisitos Específicos . . . . .	26
3.3.1. Requisitos funcionales . . . . .	27
3.3.1.1. Conectar Profesor. . . . .	27
3.3.1.2. Conectar Alumno. . . . .	28
3.3.1.3. Mostar Datos de Conexión. . . . .	29
3.3.1.4. Eliminar Pregunta. . . . .	29
3.3.1.5. Crear Pregunta. . . . .	30
3.3.1.6. Enviar Pregunta. . . . .	31
3.3.1.7. Recibir Pregunta. . . . .	32
3.3.1.8. Enviar Respuesta. . . . .	33
3.3.1.9. Recibir Respuesta. . . . .	33
3.3.1.10. Enviar datos por mail. . . . .	34
3.3.1.11. Desconectar. . . . .	35
3.3.1.12. Generar gráfico con respuestas. . . . .	36
3.3.2. Requisitos no funcionales . . . . .	36
3.3.2.1. Seguridad . . . . .	36
3.3.2.2. Rendimiento . . . . .	37

3.3.2.3.	Fiabilidad . . . . .	37
3.3.2.4.	Flexibilidad . . . . .	38
3.3.2.5.	Portabilidad . . . . .	38
3.3.2.6.	Restricciones Técnicas . . . . .	38
3.3.2.7.	Restricciones de Negocio . . . . .	38
3.3.2.8.	Otros requisitos no Funcionales . . . . .	38
<b>4.</b>	<b>Diseño</b>	<b>39</b>
4.1.	Diagrama de Actividad. . . . .	40
4.2.	Diagrama Casos de Uso. . . . .	41
4.3.	Diagrama de Componentes. . . . .	42
4.4.	Bases de datos. . . . .	43
4.4.1.	Diccionario de Datos. . . . .	44
4.5.	Interfaces de Usuario. . . . .	44
<b>5.</b>	<b>Implementación</b>	<b>55</b>
5.1.	Funcionalidades . . . . .	55
5.2.	Implementación de las Funcionalidades . . . . .	56
5.2.1.	Conectar Profesor . . . . .	56
5.2.2.	Mostrar Datos de Conexión . . . . .	58
5.2.3.	Crear Pregunta . . . . .	60
5.2.4.	Eliminar Pregunta . . . . .	63
5.2.5.	Enviar Datos por Mail . . . . .	65
5.2.6.	Generar Gráfico con Respuestas . . . . .	67
5.2.7.	Enviar Pregunta . . . . .	69
5.2.8.	Recibir Respuesta . . . . .	70
5.2.9.	Conectar Alumno . . . . .	71
5.2.10.	Recibir Pregunta . . . . .	74
5.2.11.	Enviar Respuesta . . . . .	75
5.2.12.	Desconectar . . . . .	77
<b>6.</b>	<b>Pruebas</b>	<b>78</b>
6.1.	Resumen del Plan de Pruebas. . . . .	78
6.1.1.	Descripción . . . . .	78

6.1.2.	Propósito . . . . .	78
6.1.3.	Objetivos del plan de pruebas . . . . .	79
6.1.4.	Referencias . . . . .	79
6.2.	Plan de Pruebas. . . . .	79
6.2.1.	Pruebas de Requerimientos . . . . .	79
6.2.1.1.	Descripción . . . . .	79
6.2.1.2.	Criterio para la evaluación. . . . .	79
6.2.1.3.	Condiciones de ejecución . . . . .	80
6.2.1.4.	Requerimientos de Ambiente de Pruebas . . . . .	80
6.2.1.5.	Banco de Prueba . . . . .	80
6.2.1.6.	Resultados de las Pruebas . . . . .	81
6.2.2.	Pruebas de Usabilidad . . . . .	82
6.2.2.1.	Descripción . . . . .	82
6.2.2.2.	Criterio para la evaluación. . . . .	82
6.2.2.3.	Condiciones de ejecución . . . . .	83
6.2.2.4.	Requerimientos de Ambiente de Pruebas . . . . .	83
6.2.2.5.	Banco de Prueba . . . . .	83
6.2.2.6.	Resultado de Pruebas . . . . .	85
6.2.3.	Test de Saturación con 20 Dispositivos . . . . .	90
6.2.3.1.	Descripción . . . . .	90
6.2.3.2.	Criterio para la evaluación. . . . .	90
6.2.3.3.	Condiciones de ejecución . . . . .	90
6.2.3.4.	Requerimientos de Ambiente de Pruebas . . . . .	90
6.2.3.5.	Resultado de las pruebas . . . . .	90
<b>7.</b>	<b>Conclusiones</b>	<b>91</b>
	<b>Bibliografía</b>	<b>95</b>
	<b>Apéndice A: Glosario</b>	<b>98</b>
	<b>Apéndice B: Guía de Usuario</b>	<b>99</b>

# Índice de figuras

2.1. Control Remoto del Profesor. . . . .	11
2.2. Control Remoto del Alumno. . . . .	11
2.3. Receptor de señal. . . . .	11
2.4. Convertidor USB. . . . .	12
2.5. Diagrama entorno cliente-servidor. . . . .	15
2.6. Capturas de Pantalla de la aplicación RESP . . . . .	17
2.7. Capturas de Pantalla de la aplicación ARS intranet . . . . .	18
2.8. Capturas de Pantalla de la aplicación ClickerSchool Clicker Virtual . . . . .	19
2.9. Capturas de Pantalla de la aplicación OpenARS . . . . .	20
3.1. Interfaz tentativa de selección de usuario. . . . .	25
3.2. Interfaz tentativa del profesor. . . . .	25
3.3. Interfaz tentativa del alumno. . . . .	26
4.1. Diagrama de Actividad. . . . .	40
4.2. Caso de Uso Profesor. . . . .	41
4.3. Caso de Uso Alumno. . . . .	41
4.4. Diagrama de Componentes . . . . .	43
4.5. Wireframe Selección de Perfil . . . . .	45
4.6. Selección de Perfil. . . . .	45
4.7. Wireframe Primera Vista Perfil Profesor . . . . .	46
4.8. Primera Vista Perfil Profesor. . . . .	46
4.9. Wireframe Crear Pregunta . . . . .	47
4.10. Crear Pregunta. . . . .	47
4.11. Wireframe Enviar Pregunta. . . . .	48
4.12. Enviar Pregunta. . . . .	48

4.13. Wireframe Finalizar Pregunta. . . . .	49
4.14. Finalizar Pregunta. . . . .	49
4.15. Wireframe Ver Gráfico de Torta. . . . .	50
4.16. Ver Gráfico de Torta. . . . .	50
4.17. Wireframe Enviar Datos por Mail. . . . .	51
4.18. Enviar Datos por Mail. . . . .	51
4.19. Wireframe Primera Vista Perfil Alumno. . . . .	52
4.20. Primera Vista Perfil Alumno. . . . .	52
4.21. Wireframe Esperar Pregunta. . . . .	53
4.22. Primera Esperar Pregunta. . . . .	53
4.23. Wireframe Recibir Pregunta. . . . .	54
4.24. Recibir Pregunta. . . . .	54
1. Selección de Perfil. . . . .	99
2. Ver tutorial. . . . .	100
3. Establecer Conexión. . . . .	100
4. Crear Pregunta. . . . .	101
5. Enviar y Eliminar pregunta . . . . .	102
6. Finalizar Pregunta. . . . .	102
7. Esperar y Recibir Pregunta. . . . .	103
8. Enviar Estadísticas por Email. . . . .	104
9. Ver Gráfico con Estadísticas. . . . .	104

# Índice de tablas

2.1. Grupo de Control Primer Semestre . . . . .	9
2.2. Grupo Experimental Segundo Semestre . . . . .	10
3.1. Especificación Conectar Profesor. . . . .	27
3.2. Especificación Conectar Alumno. . . . .	28
3.3. Especificación Mostar Datos de Conexión. . . . .	29
3.4. Especificación Eliminar Pregunta. . . . .	29
3.5. Especificación Crear Pregunta. . . . .	30
3.6. Especificación Enviar Pregunta. . . . .	31
3.7. Especificación Recibir Pregunta. . . . .	32
3.8. Especificación Enviar Respuesta. . . . .	33
3.9. Especificación Recibir Respuesta. . . . .	33
3.10. Especificación Enviar datos por mail. . . . .	35
3.11. Especificación Desconectar. . . . .	35
3.12. Especificación Generar Gráfico con Respuestas. . . . .	36
4.1. Diccionario de Datos. . . . .	44
6.1. Prueba de requerimientos de la aplicación parte 1. . . . .	80
6.2. Prueba de requerimientos de la aplicación parte 2. . . . .	81
6.3. Resultado de las prueba de requerimientos de la aplicación parte 1. . . . .	81
6.4. Resultado de las prueba de requerimientos de la aplicación parte 2. . . . .	82
6.5. Prueba de usabilidad de la aplicación parte 1. . . . .	84
6.6. Prueba de usabilidad de la aplicación parte 2. . . . .	85
6.7. Resultados con promomedios prueba de usabilidad de la aplicación modo profesor parte 1. . . . .	86

6.8. Resultado con promedios prueba de usabilidad de la aplicación modo profesor parte 2. . . . .	87
6.9. Resultado con promedios prueba de usabilidad de la aplicación modo alumno parte 1. . . . .	88
6.10. Resultado con promedios prueba de usabilidad de la aplicación modo alumno parte 2. . . . .	89

# Capítulo 1

## Introducción

### 1.1. Presentación del Problema

En el mundo globalizado, la inclusión de nuevas formas de tecnología como métodos complementarios a la forma de enseñanza constituye una opción viable para optimizar los resultados académicos de instituciones de educación en todos sus niveles. Desde un tiempo, se ha buscado incorporar estas tecnologías en la sala de clases, puesto que se ha comprobado empíricamente que los medios audiovisuales han contribuido a mejorar el desempeño de los educandos, desarrollando las habilidades relacionadas a la concentración, interés y atención que prestan durante el transcurso de las clases (Vega, 2013).

Siguiendo en la línea de estas herramientas se encuentran las tecleras. Se trata de un sistema que funciona en base a 5 elementos (Foretech, 2013), los cuales son: control del profesor, control del alumno, la red, espacio para proyectar las preguntas y el software que procese los datos. El sistema de las tecleras ha sido probado en universidades y colegios y su implementación ha sido altamente motivante para profesores y alumnos en el hecho de continuar integrando en sus clases este interactivo sistema (Ulloa, 2013). Este trabajo pretende implementar un sistema de tecleras de forma digital, a través de un programa en Android, el cual constará de dos perfiles: uno para el profesor y uno para los alumnos; éste último deberá conectarse al primero a través de su dirección IP y recibir las preguntas del profesor, el cual, por su parte, crea las preguntas, las envía y procesa los datos recibidos de las respuestas.

Esta aplicación intenta dar una forma alternativa a las tecleras que ya son usadas en los

distintos ámbitos de la educación, para aumentar y desarrollar la atención y cooperación de los alumnos en sus cátedras. A su vez, intenta reducir el uso de artículos físicos, teniendo en consideración todos los procesos materiales e instrumentales que requiere la implementación de una teclera física.

## **1.2. Justificación del Problema**

El desarrollo de una aplicación móvil compatible con los requerimientos tecnológicos y altos estándares de la sociedad actual es un desafío que debe configurarse en atención a las necesidades de los usuarios y a la funcionalidad específica buscada; así también, debe estar en armonía con otros factores de la mayor relevancia como es en este caso, la factibilidad de su aplicación en el ámbito académico y también otros aspectos materiales y económicos.

Continuando en este último orden de ideas, se encuentra la importancia del desarrollo de la aplicación, pues significa un ahorro de dinero para la Universidad Católica de la Santísima Concepción, ya que ofrece una nueva alternativa de cambio de tecleras físicas al formato virtual, lo cual minimiza el gasto de la compra de artículos que se requiere para su implementación y a su vez, permite que esté al alcance de la mayoría de usuarios que poseen teléfonos inteligentes con sistema operativo Android, aprovechando la generalización y masificación en el acceso y uso de celulares con dicho sistema operativo.

Es relevante indicar que la aplicación otorga anonimato de opinión a sus usuarios. Lo cual cobra especial importancia a la hora de coordinar ciertas actividades de la clase o de programar de evaluaciones, aspectos que se caracterizan por generar discrepancias de opinión entre los asistentes. En estos casos surge la utilidad práctica de esta herramienta, pues otorga confidencialidad respecto de la opción escogida, a la vez que significa un ahorro de tiempo para todos quienes se ven involucrados en el proceso de coordinación o reprogramación señalados. También, permite a profesores evaluar periódica y sistemáticamente el aprendizaje de los alumnos.

Sumado a lo anterior, se tiene que es una aplicación que se caracterizará por ser portable para el usuario. En este sentido, cabe mencionar que pocas aplicaciones cuentan con dichas características, y que ninguna se encuentra disponible de forma gratuita y en español (Véase Sección 2.2).

### **1.3. Delimitación del Problema**

En este apartado se presentan las delimitaciones de la teclera virtual a desarrollar.

- El ámbito de funcionamiento está limitado a la Facultad de Ingeniería de la UCSC, en donde se implementará la teclera.
- Esta aplicación se desarrollará en plataforma Android (Java).
- En cuanto a las estadísticas que mostrará la aplicación, solo se visualizará la cantidad y porcentaje de alumnos que respondieron a cada una de las alternativas a través de un gráfico de torta.
- Se consideró implementar sólo cuatro alternativas a cada pregunta, con el objeto de conservar el diseño de la aplicación, de modo tal que pueda visualizarse adecuándose a los márgenes de la pantalla de cualquier dispositivo, especialmente en atención a los equipos con pantalla de menor tamaño, en los cuales podría sufrir alteraciones de esquemas.

### **1.4. Objetivo General**

Desarrollar una clickera virtual sobre una plataforma Android para la Facultad de Ingeniería de la UCSC.

### **1.5. Objetivos Específicos**

Para lograr el objetivo general en este proyecto se debe cumplir con lo siguiente:

1. Analizar las funcionalidades del sistema.
2. Diseñar y modelar la aplicación.
3. Construir la aplicación Android.
4. Testear la aplicación construida.

## 1.6. Metodología

Continuando la línea de los objetivos específicos e inmediatamente a continuación, es necesario tratar y desarrollar la serie de etapas que se requiere para llevar a cabo los mencionados objetivos, es decir, la forma o modo en que se desenvolverá la investigación desarrollada. A continuación se expone la sistematización o metodología del trabajo realizado:

- Metodología general para el desarrollo de software.
  - Actividad 1: Se seguirá el método de desarrollo de software incremental, dado que el análisis de los requerimientos es estático, es decir, no puede ser objeto de modificaciones posteriores, y la especificación, diseño y construcción, dinámicos, ya que va evolucionando paulatinamente conforme avanza el desarrollo del proyecto por ende se harán entregables que irán evolucionando.
  - Actividad 2: Seguir los estándares de calidad según la ISO 25000.
- Objetivo 1: Analizar las funcionalidades del sistema.
  - Actividad 1: Estudiar el funcionamiento de las tecleras físicas.
    - Sub-actividad 1: Entrevista con profesores.
    - Sub-actividad 2: Buscar en Google scholar.
    - Sub-actividad 3: Bases de Datos SiBUCSC.
    - Sub-actividad 4: Observar, conocer y aprender de aplicaciones similares.
  - Actividad 2: Estudiar y conocer las herramientas de desarrollo a utilizar :
    - Entornos de desarrollos (Eclipse o Android Estudio).
    - Emular Android (SDK Manager, Andy, Youwave, Android para PC).
    - Testing (test de estrés).
  - Actividad 3: Seguir el estándar de la IEEE 830 para Especificación de Requisitos.
  - Actividad 4: Documentar los requerimientos.

- Objetivo 2: Diseñar y modelar la aplicación.
  - Actividad 1: Seleccionar herramientas de UML.
  - Actividad 2: Utilizar las herramientas de UML seleccionadas en la actividad 1 del objetivo 2.
- Objetivo 3: Construir la aplicación Android.
  - Actividad 1: Aprender, recordar y aplicar conocimientos de programación en JAVA.
  - Actividad 2: Codificar la aplicación por medio del entorno de desarrollo escogido en la actividad 2 del objetivo 1.
  - Actividad 3: Creaciones de versiones de prueba.
- Objetivo 4: Testear el software construido.
  - Actividad 1: Crear pruebas.
  - Actividad 2: Aplicar pruebas seleccionadas en la actividad anterior.
  - Actividad 3: Corregir Errores.

# Capítulo 2

## Marco Teórico y Estado del Arte

### 2.1. Marco Teórico

Las tecleras digitales, conocidas también como clickers o sistema inalámbrico de respuesta del estudiante, es un dispositivo individual en el que los estudiantes, por medio de un control remoto, pueden responder preguntas planteadas por un profesor, obteniendo de manera inmediata los resultados en gráficos y datos estadísticos, los cuales pueden ser mostrados en una pantalla para el conocimiento del público. Pero ¿es necesaria la implementación de esta tecnología? Esta pregunta se podría responder simplemente diciendo que sí, dado que las tecleras digitales facilitan la comunicación entre el profesor y los alumnos o entre el expositor y su audiencia, pero no basta con solo la interacción, si no que también el rendimiento del alumno (Fernández, 2012). Este sistema agiliza las actividades pedagógicas y ofrece múltiples aplicaciones en otras áreas.

Para la realización de esta aplicación se hará uso de tecnologías recientes, tales como el ambiente Android y sus respectivas herramientas proporcionadas por Google. Para lograr la comunicación entre dispositivos, se utilizará una arquitectura cliente-servidor, ambas herramientas tecnológicas se conocerán más a fondo en este estudio para facilitar el entendimiento y construcción de la aplicación.

## **2.1.1. Normas sobre Software e Ingeniería de Software**

### **2.1.1.1. Norma ISO 25000**

La familia de las normas ISO/IEC tiene por objeto la creación de un marco de trabajo común para evaluar la calidad del producto software. La familia ISO/IEC 25000, con la que se ha trabajado en este proyecto, surge como consecuencia de la evolución de sus antecesoras la ISO/IEC 9126 y la ISO/IEC 14598, dedicadas a describir las particularidades de su modelo de calidad de software y el proceso de evaluación de los productos respectivamente. Esta norma guía según una serie de estándares el desarrollo de productos de software a través de la especificación de requisitos y la evaluación de características de calidad. La estructura de la familia ISO/IEC 25000 está compuesta por cinco divisiones (ISO, 2015):

- ISO/IEC 2500n: División para la gestión de calidad.
- ISO/IEC 2501n: División para el modelo de calidad.
- ISO/IEC 2502n: División de medición de calidad.
- ISO/IEC 2503n: División de requisitos de calidad.
- ISO/IEC 2504n: División de evaluación de calidad.

### **2.1.1.2. Norma IEEE 830**

Es un formato de especificación de requisitos de software según una versión estándar y cuya última actualización es la IEEE 830, de octubre de 2008. En conformidad a esta norma todo buen documento de requisitos deberá incluir de una u otra forma la información presentada en este estándar, aunque no es obligatorio seguir estrictamente el formato dado en el estándar 830. Este formato establece como se organizará un documento de requisitos según el estándar IEEE 830. La estructura u organización en conformidad a esta norma se basa en los siguientes puntos (Mendez, 2008):

- Introducción
- Descripción General
- Requisitos Específicos

- Apéndices

### **2.1.2. Tecleras**

Al centrarse en un ámbito pedagógico, las tecleras en general poseen los siguientes objetivos (ColegioMayor, 2014):

- Obtener evidencia inmediata acerca del nivel de aprendizaje de los alumnos.
- Identificar en el transcurso de la clase los contenidos que deben ser profundizados, como también aquellos que han sido correctamente asimilados por los alumnos.
- Concentrar los esfuerzos pedagógicos en las necesidades y carencias de los alumnos.
- Fomentar el trabajo colaborativo de los alumnos, como también su capacidad de razonamiento y argumentación.

Al implementar esta tecnología en cualquier grupo de aprendizaje que posea interacción con un profesor y al cumplir estos objetivos, se podría afectar positivamente el rendimiento de los alumnos, tal como pasaremos a explicar a continuación. También, es posible que el profesor sepa en que áreas reforzar los contenidos y qué materias fueron asimiladas de manera más efectiva por los alumnos.

Para reforzar y complementar la idea anterior, se tiene un estudio realizado por la Universidad de Santiago de Chile en la carrera de Ingeniería de Ejecución Mecánica (Vega, 2013). En este trabajo, una docente implementó el uso de tecleras a un curso determinado, el cual fue evaluado durante el curso de un año; el primer semestre, con el sistema tradicional de enseñanza, y el segundo, complementándolo con el uso de esta tecnología. Los resultados que arrojó esta investigación fueron los siguientes: el grupo de alumnos que cursó la asignatura durante la primera parte del año, sin el uso de tecleras, obtuvo un 64,29 % de aprobación, mientras que el grupo de los estudiantes que cursaron esta misma asignatura el segundo semestre, utilizando las tecleras, obtuvieron el 84,45 % de aprobación. En conclusión, existió una diferencia mayor al 20 % en el rendimiento con respecto a los alumnos que utilizaron tecleras y los que no lo hicieron

en el curso del semestre respectivo. Ciertamente fueron los alumnos del segundo semestre los que mostraron un mayor índice de aprobación de la asignatura en cuestión, lo cual se considera como indiciario respecto a la eficiencia de las tecleras, tal como se ha afirmado anteriormente. En las Tablas 2.1 y 2.2 se muestra la comparación de notas finales de una cierta asignatura entre el grupo control (primer semestre) y el grupo experimental (segundo semestre), con y sin la prueba optativa que reemplaza una nota.

Ya se analizó la dimensión positiva en relación a este estudio, y como conclusión se establece que se observa un progreso y un apoyo a los estudiantes, ya que el uso de la tecnología los hace prestar más atención a los contenidos expuestos por el profesor (lo que tiene como consecuencia la obtención de mejores resultados académicos, que posteriormente inciden en la aprobación de la asignatura). Sin embargo esta tecnología también puede presentar algunas desventajas tales como (Cilleros, 2014):

- Costo en tiempo en capacitación de los profesores.
- Costo en tiempo en preparación de la clase.
- Costo en tiempo en la confección de las preguntas.
- Costo en tiempo de entrega y recepción del material a los estudiantes.
- Ansiedad de los estudiantes por la tecnología por sobre el contenido mostrado por el profesor en clases.

Notas Promedio	N alumnos s/p	% alumnos s/p	N alumnos c/p	% alumnos c/p
1,0-2,9	0	0	0	0
3,0-3,9	5	35,75	4	28,57
4,0-4,9	8	57,14	9	64,69
5,0-5,9	1	7,15	1	7,14
6,0-7,0	0	0	0	0

Tabla 2.1: Grupo de Control Primer Semestre

En la Tabla 2.1 se muestra el grupo de control del primer semestre, en el cual 64,29 % de los alumnos obtuvieron nota sobre 4,0. Luego, se muestra el resultado obtenido después de una prueba de recuperación que reemplaza a una nota de la asignatura. En esta oportunidad, el porcentaje de aprobación aumentó al 71,83 %. Cabe destacar que el primer semestre este grupo no utilizó teclera.

Notas Promedio	N alumnos s/p	% alumnos s/p	N alumnos c/p	% alumnos c/p
1,0-2,9	1	22,22	1	22,22
3,0-3,9	6	13,33	2	4,44
4,0-4,9	36	80	40	88,89
5,0-5,9	2	4,45	20	4,46
6,0-7,0	0	0	0	0

Tabla 2.2: Grupo Experimental Segundo Semestre

En la Tabla 2.2 se muestran los resultados del grupo experimental del segundo semestre, en el cual el curso obtuvo un 84,45 % sin prueba de recuperación y un 93,35 % con prueba de recuperación, respectivamente. Es importante destacar el aumento en el porcentaje de aprobación del curso debido al uso de las tecleras.

### 2.1.3. Tecleras Físicas

Las tecleras físicas son las que se conocen hasta el momento y las que se han implementado en colegios o universidades con más recursos y con intención de aumentar la interacción estudiante-profesor. Éstas son conocidas como Tecleras RF FORETECH MM (Foretech, 2013). Este sistema incluye el equipamiento hardware y el software necesario para operar (los cuales se definen más adelante). Aunque además es necesario un computador de escritorio o notebook y un proyector para visualizar las preguntas emitidas por el profesor. Los componentes de las Tecleras RF FORETECH MM se describen a continuación:

1. **Control remoto profesor** (Véase Figura 2.1) : Permite detener o cerrar, iniciar actividades, cambiar modo de actividad, volver a la diapositiva anterior, iniciar y pausar tiempo de respuesta, ver los resultados en tabla o gráfico, avanzar a la diapositiva siguiente y quitar barras superior e inferior durante la actividad, además de tener incorporado un puntero láser.
2. **Control remoto estudiante** (Véase Figura 2.2): Permite responder con opciones de alternativas de la A a la F.
3. **Un receptor de señal** (Véase Figura 2.3): Dispositivo inalámbrico que permite emitir la señal a los controles tanto de profesor como alumnos y quien comparte la información con el computador.



Figura 2.1: Control Remoto del Profesor.  
Malleducativo (2011)



Figura 2.2: Control Remoto del Alumno.  
Factory (2012)



Figura 2.3: Receptor de señal.  
(Factory, 2011)

4. **Cable para conexión con el computador con adaptador RJ45-USB incluido**(Véase Figura 2.4): Permite conectar el receptor de señales con el computador o notebook para el envío de paquetes de información para ser procesados en el computador y generar las estadísticas.
5. **Disco de instalación (solo para tecleras que requieran procesamiento de datos en un computador)**: Disco que contiene el software para la ejecución del programa en el computador. Para que la instalación sea exitosa y el software funcione correctamente el computador utilizado debe contar con los siguientes



Figura 2.4: Convertidor USB.  
Papiro (2013)

requisitos como mínimo:

- 64 MB de memoria RAM.
- Sistema operativo: Windows Me/2000, XP, Vista ó 7.
- Office 2000/2003/2007/2010
- 100 MB de espacio libre en disco duro.
- 1 puerto USB libre.

Anteriormente se habló de un receptor de señal para los controles tanto del profesor como del alumno; este receptor responde a dos tecnologías de recepción de señal (Cilleros, 2014), las cuales son:

- **Tecnología Infrarroja.** Un estudiante responde apuntando la teclera hacia el emisor de la señal de infrarrojos, el cual se encuentra conectado al computador mediante un cable USB y de esta forma recibe los datos.
- **Tecnología Radio Frecuencia.** El estudiante debe digitar en su teclera la respuesta seleccionada; los datos son recepcionados por una antena de radiofrecuencia conectada al computador que mediante el software asociado registra los datos. Esta última es la más utilizada debido a la comodidad de no tener que estar apuntando al receptor para que pueda recibir la señal.

#### 2.1.4. Tecleras Digitales.

En esta sección se tratará de los aspectos de las tecleras digitales, los cuales los diferencian de las físicas.

#### **2.1.4.1. Computación Móvil.**

Dado el gran auge que ha experimentado en este último tiempo la tecnología portátil o móvil y junto a la llegada de los smartphome, es que surge la necesidad de acercarse a este contexto. Lo anterior puede vislumbrarse con claridad en la realidad chilena, debido a que existen más aparatos de la telefonía móvil activos que habitantes (de Telecomunicaciones, 2013). Este proyecto estará centrado en tecleras digitales a partir de dispositivos móviles, los cuales cumplirán la función de los componentes mencionados anteriormente de tecleras físicas, pero ¿qué es la computación móvil? Es una tecnología que habilita el acceso a recursos digitales en cualquier momento y desde cualquier lugar Dávalos (2012), la cual se soporta en tres pilares básicos:

1. Las redes inalámbricas.
2. La portabilidad (dispositivos móviles).
3. La movilidad (usuarios).

La computación móvil está cambiando la sociedad actual de una forma tan significativa como lo ha hecho Internet. Y a pesar que esta revolución sólo está empezando, los nuevos terminales ofrecen unas capacidades similares a las de un computador, lo que permite que puedan ser utilizados para leer nuestro correo, chatear o simplemente navegar por Internet. Pero a diferencia los computadores, un teléfono móvil siempre está en el bolsillo del usuario. Esto permite un nuevo sin fin de aplicaciones mucho más cercanas al usuario como por ejemplo la que se quiere implementar en este proyecto. Todo esto sustentable en el surgimiento de sistemas operativos para smartphome destacando entre varios el sistema operativo Android por sobre iPhone, Windows Phone o BlackBerry.

#### **2.1.4.2. Android**

En materia de sistemas operativos para dispositivos móviles y en cuanto a la ejecución de aplicaciones existen varias plataformas, pero ¿por qué se prefiere Android? La respuesta a esta pregunta es que Android posee cualidades específicas que lo hacen diferente a las otras plataformas (Calleja, 2013), las cuales se pasarán a conocer y detallar. Primero, se debe mencionar que es una plataforma que basa su lenguaje en

Linux, el cual es de adquisición libre y en el que todos pueden desarrollar sin incurrir en costos. Es por esto que se dice que Android es un lenguaje de código abierto; es portable debido a que:

- Las aplicaciones son desarrolladas en lenguaje JAVA, con lo que se asegura la visualización de futuras aplicaciones por el uso de máquinas virtuales que emplea el lenguaje mencionado.
- Contempla una gran cantidad de servicios incorporados al sistema operativo y fáciles de utilizar, por ejemplo, localización por GPS, reconocimiento de voz, bases de datos SQL, entre otros.
- Optimización para baja potencia y poca memoria, ya que Android utiliza la Máquina Virtual Dalvik, Creada por Google de la máquina virtual de Java optimizada para dispositivos móviles.

#### 2.1.4.3. Cliente-Servidor

El entorno cliente-servidor es la arquitectura que cumplirá la base de comunicación entre el profesor y los alumnos. Esta tecnología trata de que varios equipos llamados clientes, se comunican por medio de un servidor, el que contesta a los requerimientos de los clientes (Alegsa, 2010). Por lo general los clientes inician la sesión de comunicación, mientras que los servidores esperan la llegada de solicitudes. En el caso particular de este proyecto, es el profesor quien tendrá la aplicación servidor y los alumnos serán los clientes. Por lo general, los clientes y los servidores se comunican entre sí a través de una red, pero también pueden alojar ambos en un mismo hardware.

Las ventajas y desventajas de la arquitectura cliente-servidor se mencionan a continuación (Commons, 2014).

1. **Ventajas:** Como se observa en la Figura 2.5 el servidor es el centro de la conexión en la red, por tanto, la información compartida se puede transferir a todos los nodos sin importar la dirección. La cantidad de puntos de entrada que permite el acceso a los datos no es importante, es por esto que se dice que la seguridad es mejorada. Dado que los clientes no forman parte indispensable en la ejecución, se pueden conectar o eliminar sin alterar, interrumpir o modificar la red, por esto se menciona que es una arquitectura escalable.

2. **Desventajas:** Por la gran complejidad técnica del servidor se mantienen costos muy elevados en la implementación de esta arquitectura, pues es necesario invertir en una red, en el servidor y cada cliente debe tener su dispositivo propio. Si se observa nuevamente la Figura 2.5 se nota que la conexión depende exclusivamente del servidor, por tanto si ésta falla, se pierde la conexión de todos clientes que se encuentran conectados con él en ese momento; cabe mencionar que el servidor es tolerable a fallos, por ende, pocas veces deja de funcionar por completo.

En la Figura 2.5 se observa como cinco computadores están conectados a un servidor único, el cual recibe y distribuye la información dependiendo de cuales fueron las peticiones realizadas por los clientes.

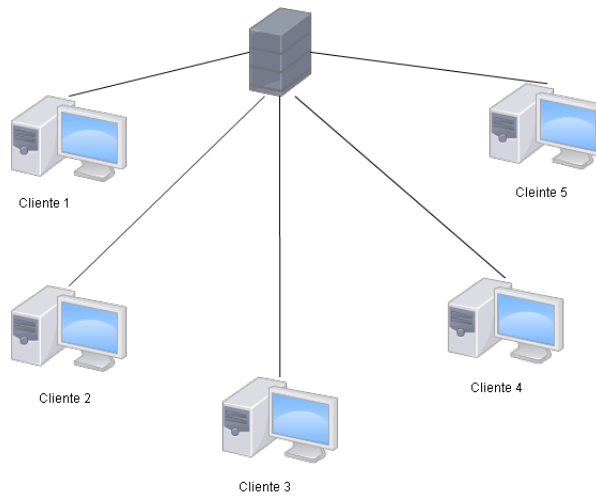


Figura 2.5: Diagrama entorno cliente-servidor.

### 2.1.5. Conclusión

En conclusión, las tecleras digitales funcionan de igual forma que las físicas, aunque mostrando ciertas ventajas en cuanto al tema de la portabilidad. Si se hace una comparación en base a los componentes de una teclera física, los controles tanto del profesor como de los alumnos vienen dados por los smartpone. El cable de conversión a UBS no es necesario ya que es un sistema inalámbrico, no se requiere el CD de instalación debido a que los cálculos son hechos por el dispositivo móvil del profesor

a la hora de recibir los datos del alumno. Y por último, en cuanto al receptor de señal, es tarea del mismo smartphone emitir la señal o conectarse a una red, pues tiene incorporado un sistema de red WiFi especialmente destinado para estos efectos.

## **2.2. Estado del Arte.**

En esta sección se ofrece una breve descripción de los métodos de tecleras digitales que se encuentran disponibles en el mercado de aplicaciones para Android. Estos datos se encuentran actualizados a la fecha Junio de 2014 (Page, 2014).

### **2.2.1. RESP**

Esta aplicación está enfocada al área de las empresas, y también al área educacional. En el primero de estos ámbitos sirve como apoyo en las exposiciones, ya que permite comprender de mejor forma el contenido que se trata de explicar. Esto, porque permite interactuar de forma inmediata con el moderador y del mismo modo, saber, a través de comentarios instantáneos si los oyentes han podido entender lo que el expositor intenta enseñar. Este sistema consta de dos modos: el monitor y los auditores. En la parte del monitor, el usuario administra las preguntas que hará para posteriormente enviarlas a los auditores. En este mismo modo, existe un código llamado sala virtual, a través del cual se debe dar a conocer a los auditores para que se conecten con el monitor, y respondan las preguntas que éste envía. Esta aplicación fue probada en el transcurso de la investigación y se verificó que la conexión era inestable, de modo tal que este software no cumplía el objetivo prometido. Esta aplicación es gratuita, se encuentra en español, pero ya no está disponible en play store. En la Figura 2.6 se muestran algunas capturas de pantalla de la aplicación RESP.

### **2.2.2. ARS intranet**

Esta aplicación es solo una parte del sistema completo, ya que requiere de dos elementos más para funcionar íntegramente. En este sistema se visualizan las letras a,b,c,d, que son las alternativas a elegir para formular una respuesta a la pregunta proyectada en

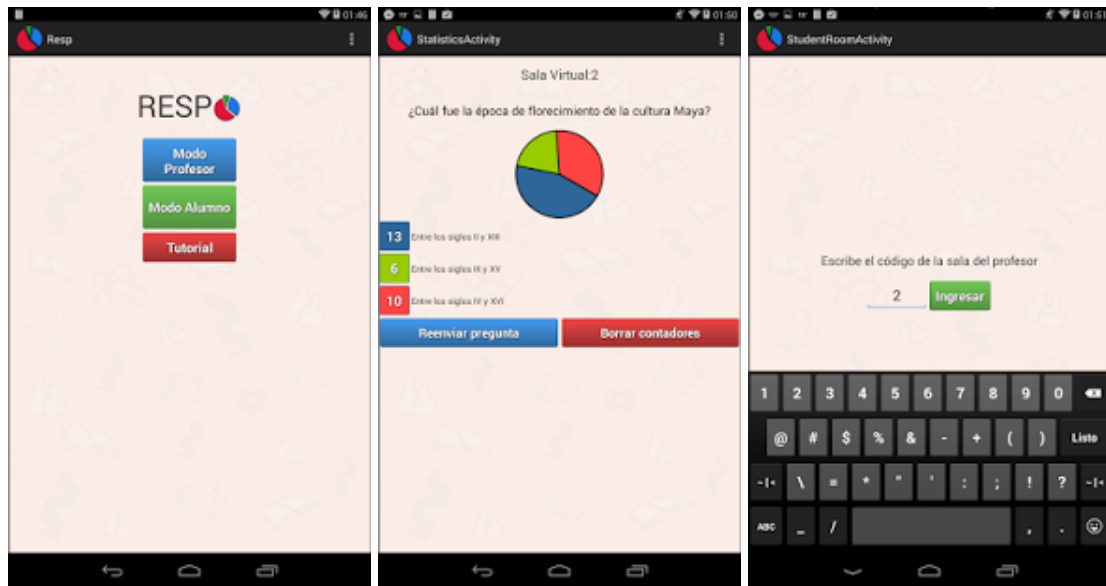


Figura 2.6: Capturas de Pantalla de la aplicación RESP (apk4ios, 2013)

un Power Point. Este último es uno de los elementos que conforman el sistema completo. Un servidor web, llamado Speaker, el cual guarda todas las respuestas en una base de datos, luego los procesa y más tarde genera estadísticas con ellos. Este sistema está exclusivamente orientado al ámbito empresarial. En síntesis, se necesita, en primer lugar, tener la aplicación, un computador y un proyector digital para proyectar las preguntas en power point y también acceso al servidor Speaker. Los requisitos indispensables para el funcionamiento de la aplicación, en el caso del cliente, son: un teléfono smartphone, conexión a internet y sistema operativo Android del 1.6 al 4.0 o iOS, ya que también esta aplicación está disponible para iPhone y iPad. En el caso del servidor, se necesita Apache + PHP, y para la base de datos MySQL, y además, que el computador soporte power point. Esta aplicación es gratuita, se encuentra en la play store y está en inglés. En la Figura 2.7 se muestran algunas capturas de pantalla de la aplicación ARS intranet.

### 2.2.3. ClickerSchool Clicker Virtual

Es un sistema similar a los explicados, pues cumple las funciones de cliente y servidor. Tiene por objeto que sus usuarios puedan realizar votaciones a través de la página web en la cual se encuentra disponible, de modo que puedan obtener sus resultados de

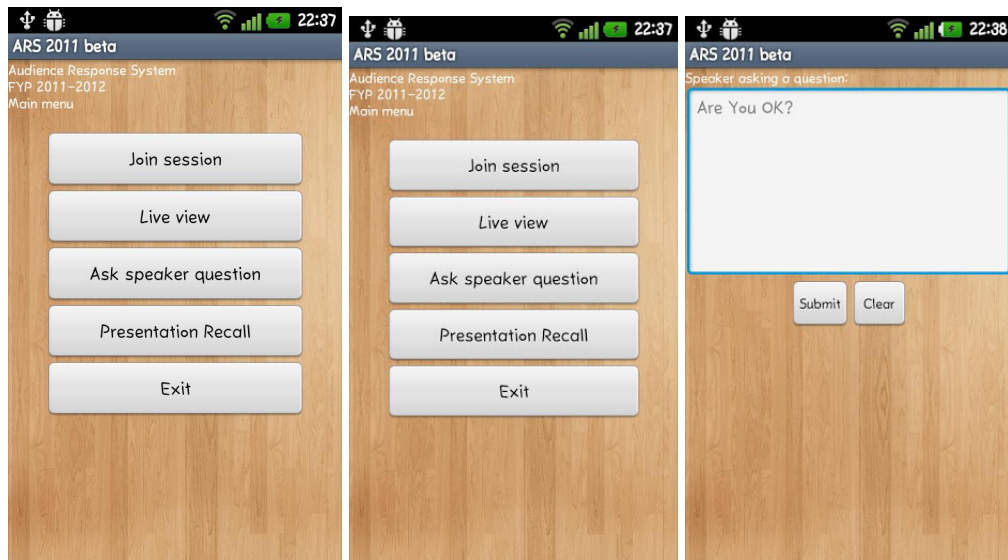


Figura 2.7: Capturas de Pantalla de la aplicación ARS intranet (Brothersoft, 2012)

forma inmediata, sin ningún proceso de por medio (como sería el conteo de votos), pues toda la información se obtiene directamente de la aplicación. Otro uso que se le puede atribuir a esta aplicación es para realizar evaluaciones de selección múltiple a alumnos en torno a un determinado tópico y luego asignarle automáticamente un puntaje en conformidad a su respuesta. Entre los aspectos a destacar de esta aplicación, se encuentra que emite informes que se pueden enviar vía mail, un chat fluido entre cliente y servidor y que este sistema funciona bajo cualquier dispositivo que tenga conexión a Internet, y no necesariamente una aplicación Android. Sólo requiere de suscripción, la cual tiene un costo de USD \$0.99. En conclusión, esta aplicación no constituye un aporte innovador, puesto que es solo una adaptación de la página web [www.clickerschool.com](http://www.clickerschool.com) a smartphone, de modo que ésta se pueda utilizar en cualquier momento y lugar y sea más accesible al público. Es necesario dejar en claro que no es una aplicación nueva. Esta aplicación se encuentra disponible en play store, está en inglés y es de pago. En la Figura 2.8 se muestran algunas capturas de pantalla de la aplicación ClickerSchool Clicker Virtual.



Figura 2.8: Capturas de Pantalla de la aplicación ClickerSchool Clicker Virtual (PlayStore, 2014a)

#### 2.2.4. OpenARS

Esta aplicación Android es la parte cliente del sistema, en el cual se responden las preguntas enviadas por el moderador, el cual es un servidor web alojado en la dirección [www.openars.dk](http://www.openars.dk). Es un sistema separado en dos modos al igual que RESP ya descrito en la Sección 2.2.1, solo que estos modos están separados uno en un smartphone y el otro en un servidor web. Esta aplicación se encuentra en play store, es gratis y está en inglés. En la Figura 2.9 se muestran algunas capturas de pantalla de la aplicación OpenARS.

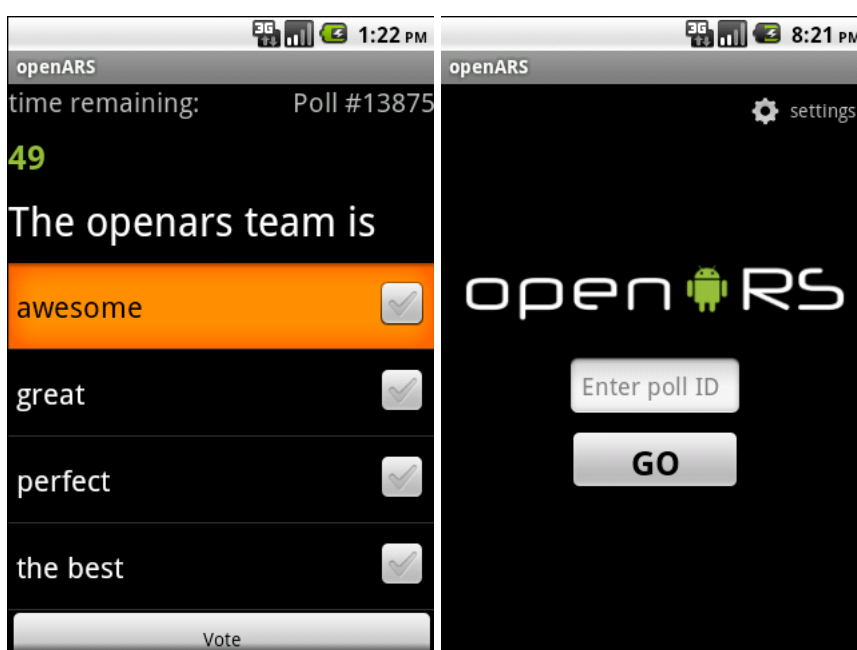


Figura 2.9: Capturas de Pantalla de la aplicación OpenARS (PlayStore, 2014b)

# Capítulo 3

## Especificación de Requisitos

### 3.1. Introducción

Con este software se pretende entregar un apoyo a docentes a la hora de desempeñar sus métodos de enseñanza, incentivando a los alumnos con la ayuda de la tecnología a fin de que presten la debida atención a lo que el profesor intenta presentar en su clase. Además permite hacer evaluaciones en tiempo real a los alumnos sobre un tópico determinado, el docente recibirá la respuesta de todos los alumnos plasmada en un gráfico, cabe destacar que con esto el profesor sabrá inmediatamente si los contenidos presentados en la clase se percibieron de buena forma.

#### 3.1.1. Propósito

Describir lo que hará el sistema, como operará el mismo y explicar hacia donde está enfocado, documentando adecuadamente las funciones que tendrá para cuando se requiera una posterior mantención, actualización o modificación del sistema.

#### 3.1.2. Ámbito del Sistema

El software teclera virtual será una aplicación para Android, orientada al ámbito educacional, sin embargo su ámbito de ejecución no se agota aquí; es posible darle usos alternativos, que dependerán de la sola voluntad de los clientes, tales como evaluaciones en tiempo real o juegos de trivia. Esta aplicación contempla un completo registro de respuestas para eventuales y futuras comparaciones en el desempeño entre 2 o más

cursos a través del envío de la información al mail del usuario, quien podrá recuperar los testimonios cuando estime conveniente.

El sistema tendrá dos registros distintos de usuarios, *usuario perfil profesor o servidor* y *usuario perfil alumno o cliente*.

### **3.1.3. Vista General del Documento**

Este documento tiene como objetivo describir las funcionalidades necesarias para el desarrollo de una aplicación móvil que permita estimular la reacción y atención de un grupo de alumnos, frente a los contenidos propuestos por un profesor o bien, inducir en el uso de la tecnología en la sala de clases.

### **3.1.4. Referencias**

Esta especificación de requisitos se ha adecuado al formato del estándar IEEE 830 del año 2008.

## **3.2. Descripción General**

### **3.2.1. Perspectiva del Producto**

La aplicación teclera virtual sobre una plataforma Android para la UCSC no tendrá interacción con otras aplicaciones externas en lo relativo al software, ya que, bajo él subyace una interacción con el motor de bases de datos para aplicaciones Android SQLite.

### **3.2.2. Funciones del Producto**

Las siguientes funciones representan las funcionalidades que tendrá la aplicación a desarrollarse.

- A) Conectar Profesor.
- B) Conectar Alumno.
- C) Mostrar datos de conexión.

- D) Crear preguntas.
- E) Eliminar preguntas.
- F) Enviar preguntas.
- G) Recibir pregunta.
- H) Enviar respuesta.
- I) Recibir respuestas.
- J) Enviar datos por mail.
- K) Generar gráfico con respuestas.
- L) Desconectar.

### 3.2.3. Características de Usuarios

- *Perfil profesor o servidor*: Es al cual deben conectarse los alumnos, además es el que envía las preguntas y el que visualiza el gráfico con las respuestas.
- *Perfil alumno o cliente*: Es quien debe conectarse con la información requerida y responder las preguntas propuestas por el profesor.

### 3.2.4. Restricciones

- Limitaciones de Hardware: La aplicación debe ser capaz de funcionar en un dispositivo móvil con:
  - Procesador de 800 Mhz o superior.
  - Memoria RAM de 300 MB o superior.
  - Espacio disco de 500 MB o superior.
- Lenguaje de programación:
  - Deberá ser desarrollada en JAVA para Android.
- El usuario no usará contraseña para autenticarse.

- El tiempo de respuesta tiene que ser menor a 2 segundos.
- La aplicación deberá tener acceso a red WiFi.
- Para que la visualización de la aplicación sea estándar y pueda verse de forma que no se desvirtúe la imagen, no se podrá visualizar de forma horizontal.

### **3.2.5. Suposiciones y Dependencias**

Los usuarios del software deberán utilizar la aplicación por medio un dispositivo móvil con sistema operativo Android.

Para las futuras versiones o actualizaciones del software se requerirá la descarga completa del mismo, la cual reemplazará a la version previa que se tenía instalada.

Es estrictamente necesario que ambos perfiles esten conectados a la misma red Wifi para que se establezca la comunicación.

### **3.2.6. Requisitos comunes de las interfaces**

#### **3.2.6.1. Interfaces de usuario**

A continuación se presentan las interfaces tentativas de las pantallas del software, las cuales están sujetas a eventuales cambios en transcurso del desarrollo del proyecto, como también a la posibilidad de que sea necesaria la creación de algunas diferentes a las que se muestran a continuación.

En la Figura 3.1 se muestra la pantalla principal de la aplicación, en la cual es posible optar para entrar con el perfil profesor o bien, como alumno; en la Figura 3.2 es posible ver la información de conexión que debe ser proporcionada a los alumnos y un botón destinado a crear las preguntas que serán enviadas a los alumnos conectados. Y por último, en la Figura 3.3 se puede observar que se requieren los datos de conexión proporcionados por el profesor y más abajo, la pregunta enviada por éste.

#### **3.2.6.2. Interfaces de hardware**

Para un correcto uso de las funcionalidades ofrecidas por el software es necesario disponer de un teléfono inteligente con pantalla touch y sistema operativo Android, el

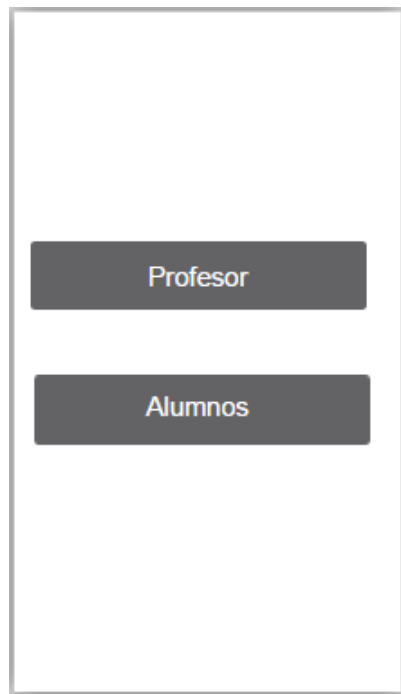


Figura 3.1: Interfaz tentativa de selección de usuario.



Figura 3.2: Interfaz tentativa del profesor.



The image shows a mobile application interface for a student. It features two text input fields at the top, labeled 'Nombre' and 'Dirección IP'. Below these is a question prompt '¿ Pregunta ?'. Underneath the question are four dark grey buttons stacked vertically, labeled 'Alternativa A', 'Alternativa B', 'Alternativa C', and 'Alternativa D'.

Figura 3.3: Interfaz tentativa del alumno.

cual debe tener acceso a WiFi de modo tal que pueda conectarse a una red en común con los distintos perfiles de usuarios.

### 3.2.6.3. Interfaces de comunicación

Los protocolos de comunicación para la interacción entre profesor y alumno, serán los utilizados por la arquitectura Cliente-Servidor(Véase Sección 2.4.4) y para la comunicación con la bases de datos se utilizará SQLite, herramienta proporcionada por los compiladores de Android que facilita la conexión y llamadas a dicha base de datos.

## 3.3. Requisitos Específicos

En este apartado se presentan los requisitos funcionales y no funcionales de la aplicación, de forma que se describa lo que la aplicación quiere hacer y a su vez reconocer las limitaciones del sistema.

### 3.3.1. Requisitos funcionales

A continuación en las Tablas 3.1, 3.2, 3.3, 3.5, 3.4, 3.6, 3.7, 3.8, 3.9, 3.10, 3.12 y 3.11 se muestran los requisitos funcionales del sistema.

#### 3.3.1.1. Conectar Profesor.

Nombre	A) Conectar Profesor
<b>Entradas:</b>	cliente, puerto.
Procesos	<p>A.1 cliente es un string que aceptará caracteres entre <math>[a, z]</math>, <math>[A, Z]</math>, el carácter de “espacio” y numeros entre <math>[0, 9]</math> con un máximo de 30 caracteres, cliente es el nombre que utiliza los alumnos para establecer la conexión con el servidor.</p> <p>A.2 Para puerto se utilizarán números naturales del 0 al 9999, y representarán la interfaz a través de la cual se podrán enviar y recibir los diferentes tipos de mensajes, este número es asignado dentro del código de la aplicación y es utilizado por los clientes para localizar al servidor, además es donde el servidor espera a los clientes.</p> <p>A.3 Para que el profesor logre la conexión es necesario que permanezca escuchando continuamente a los alumnos que intenten conectarse a él, para ello se crea un ciclo infinito en donde interactúan indefinidamente. Cabe destacar que dichas interacciones ocurren en un puerto diferente al especificado en A.2, lo cual permite que se conecten varios alumnos al profesor.</p> <p>A.4 Para que Conectar Profesor sea exitosa serán obligatorios los campos: cliente y puerto.</p> <p>A.5 Una conexión exitosa consiste en cumplir las especificaciones desde A.1 al A.3, de ser así el profesor se conectará y se mantendrá en espera de los alumnos que quieran conectarse con él.</p>
Salida 1	cliente, puerto

Tabla 3.1: Especificación Conectar Profesor.

**3.3.1.2. Conectar Alumno.**

Nombre	B) Conectar Alumno
<b>Entradas:</b>	ip, nombre_alumno, puerto.
Procesos	<p>B.1 Para ip se aceptarán números entre [0,9] y el caracter “.”, en donde se debe considerar un formato tal que, existan cuatro campos con un números decimales hasta el 255 separados por un puntos(por ejemplo 192.168.1.3).</p> <p>B.2 Es estrictamente necesario que el profesor proporcione los datos de conexión, dado que se debe ingresar la misma dirección IP que éste posee.</p> <p>B.3 Para nombre_alumno se utilizarán los requisitos ya definidos en A.1.</p> <p>B.4 Para puerto se utilizarán los requisitos ya definidos en A.2.</p> <p>B.5 Para que Conectar Alumnos sea exitoso es estrictamente necesario que el profesor esté conectado según lo descrito en la Tabla 3.1.</p> <p>B.6 Una conexión exitosa consiste en cumplir las especificaciones desde B.1 al B.5, de ser así el profesor recibirá el mensaje "nombre_alumno conectad@ con ip y puerto ". Consecutivamente el profesor enviará un mensaje al alumno que estableció la conexión que dirá "Bienvenido nombre_alumno".</p>
Salida 1	mensajes en B.6

Tabla 3.2: Especificación Conectar Alumno.

**3.3.1.3. Mostar Datos de Conexión.**

Nombre	C) Mostar Datos de Conexión
<b>Entradas:</b>	ip_profesor, puerto_profesor.
Procesos	<p>C.1 Para ip_profesor se utilizarán los requisitos ya definidos en B.1.</p> <p>C.2 Para puerto_profesor se utilizarán los requisitos ya definidos en A.2.</p> <p>C.3 Para que mostrar los datos de conexión sea exitosa se deben cumplir las especificaciones desde C.1 al C.2. Si se cumple lo anterior se desplegará la dirección IP del profesor y el puerto del profesor en la parte superior izquierda a través de un texto.</p>
Salida 1	ip_profesor, puerto_profesor.

Tabla 3.3: Especificación Mostar Datos de Conexión.

**3.3.1.4. Eliminar Pregunta.**

Nombre	E) Eliminar Pregunta
<b>Entradas:</b>	pregunta.
Procesos	<p>E.1 Para pregunta se utilizarán los requisitos ya definidos en D.1.</p> <p>E.2 Para que eliminar pregunta sea exitoso se debe confirmar la eliminación presionando en aceptar en el dialogo desplegado al realizar un click prolongado en la pregunta a eliminar en este caso se desplegará el mensaje: "Pregunta eliminada con éxito", en caso contrario no se eliminara la pregunta.</p> <p>E.3 Una vez eliminada la pregunta exitosamente se actualizará la lista con las preguntas a mostrar para ser usadas.</p>
Salida 1	mensaje en E.2.
Salida 2	E.3.

Tabla 3.4: Especificación Eliminar Pregunta.

**3.3.1.5. Crear Pregunta.**

Nombre	D) Crear Pregunta
<b>Entradas:</b>	pregunta, alternativa_a, alternativa_b, alternativa_c, alternativa_d.
Procesos	<p>D.1 Para pregunta sólo se aceptarán caracteres entre <math>[a, z]</math>, <math>[A, Z]</math> y el carácter de “espacio”, además de los números del <math>[0, 9]</math> con un máximo de 30 caracteres.</p> <p>D.2 Para alternativa_a sólo se aceptarán caracteres entre <math>[a, z]</math>, <math>[A, Z]</math> y el carácter de “espacio”, además de los números del <math>[0, 9]</math> con un máximo de 30 caracteres.</p> <p>D.3 Para alternativa_b sólo se aceptarán caracteres entre <math>[a, z]</math>, <math>[A, Z]</math> y el carácter de “espacio”, además de los números del <math>[0, 9]</math> con un máximo de 30 caracteres.</p> <p>D.4 Para alternativa_c sólo se aceptarán caracteres entre <math>[a, z]</math>, <math>[A, Z]</math> y el carácter de “espacio”, además de los números del <math>[0, 9]</math> con un máximo de 30 caracteres.</p> <p>D.5 Para alternativa_d sólo se aceptarán caracteres entre <math>[a, z]</math>, <math>[A, Z]</math> y el carácter de “espacio”, además de los números del <math>[0, 9]</math> con un máximo de 30 caracteres.</p> <p>D.6 Para que la opción crear pregunta sea exitosa, el campo preguntado debe ser obligatoriamente ingresado; en caso contrario, se desplegará por pantalla el mensaje: “Ingresar Pregunta”.</p> <p>D.7 Para realizar la operación exitosamente y cumplir con el registro es necesario completar a lo menos dos alternativas; si no cumple con estos requerimientos se desplegará el mensaje: “Debe ingresar a lo menos dos alternativa”.</p> <p>D.8 Si se cumplen con éxito los requisitos especificados en D.1 hasta D.7 se desplegará el mensaje “Pregunta Creada con Éxito”.</p> <p>D.9 Una vez creada la pregunta exitosamente se actualizará la lista con las preguntas a mostrar para ser usadas.</p>
Salida 1	mensajes en D.6, D.7 y D.8 .
Salida 2	pregunta, alternativa_a, alternativa_b, alternativa_c, alternativa_d.
Salida 3	D.9.

Tabla 3.5: Especificación Crear Pregunta.

**3.3.1.6. Enviar Pregunta.**

Nombre	F) Enviar Pregunta
<b>Entradas:</b>	pregunta_a_enviar, lista_cliente.
Procesos	<p>F.1 pregunta_a_enviar es un string que consiste en la concatenación de las variables pregunta ya especificado en D.1 , alternativa_a especificado en D.2, alternativa_b especificado en D.3, alternativa_c especificado en D.4 y alternativa_d especificado en D.5, separados por un salto de línea.</p> <p>F.2 lista_cliente es un arreglo de objetos, en este caso, de los alumnos junto a su puerto. Se necesita este parametro para que el profesor logre realizar el envío a todos los alumnos conectados con él.</p> <p>F.3 Para lograr el envío de la pregunta, se debe confirmar presionando el botón aceptar en el dialogo desplegado al realizar un click en la pregunta a enviar, en este caso se desplegará el mensaje: “Pregunta enviada a lista_cliente”, en caso contrario no se enviará la pregunta.</p> <p>F.4 Para que Enviar Pregunta sea exitoso se deben cumplir los requisitos F.1 al F.3.</p>
Salida 1	Mensaje en F.3.
Salida 2	pregunta_a_enviar.

Tabla 3.6: Especificación Enviar Pregunta.

**3.3.1.7. Recibir Pregunta.**

Nombre	G) Recibir Pregunta
<b>Entradas:</b>	pregunta_a_enviar.
Procesos	<p>G.1 Para pregunta_a_enviar se utilizarán los requerimientos especificados en F.1.</p> <p>G.2 Para recibir el mensaje enviado, el profesor debe confirmar el envío presionando en el botón enviar al hacer un click corto en la pregunta seleccionada.</p> <p>G.3 Una vez recibido el mensaje se separará pregunta_a_enviar en 5 nuevas variables una para pregunta, y los otros cuatro para las alternativas alternativa_a, alternativa_c, alternativa_c, alternativa_d, cada separación se establece según los saltos de línea que existan entre ellos.</p> <p>G.4 Para que recibir mensaje sea exitoso se deben cumplir los requisitos de G.1 al G.3, luego de concluir estos pasos la pregunta es mostrada a los alumnos a través de un Textview y las cuatro alternativas en cuatro botones diferentes, todo esto para que posteriormente se puedan enviar las respuestas.</p>
Salida 1	pregunta, alternativa_a, alternativa_c, alternativa_c, alternativa_d.
Salida 2	G.4.

Tabla 3.7: Especificación Recibir Pregunta.

**3.3.1.8. Enviar Respuesta.**

Nombre	H) Enviar Respuesta
<b>Entradas:</b>	respuesta.
Procesos	<p>H.1 Para respuesta se utilizará un string el cual será "a", "b", "c" ó "d", lo que implicará la respuesta a enviar por el alumno. El valor de respuesta vendrá dado según el botón de la alternativa que presione el alumno.</p> <p>H.2 Para enviar la respuesta al profesor, los alumnos deben presionar en la alternativa y así de esta manera enviar la letra correspondiente a la alternativa seleccionada.</p> <p>H.3 Para que enviar respuesta sea exitoso se deben cumplir los requisitos H.1 y H.2, próximo a realizar el envío, se ocultará la pregunta y los botones con las alternativas para que los alumnos respondan sólo una vez.</p>
Salida 1	respuesta.

Tabla 3.8: Especificación Enviar Respuesta.

**3.3.1.9. Recibir Respuesta.**

Nombre	I) Recibir Respuesta
<b>Entradas:</b>	respuesta.
Procesos	<p>I.1 Para respuesta se utilizará el requisito ya especificados en H.1.</p> <p>I.2 Al ser recibida la respuesta se realizará una clasificación dependiendo cual sea la respuesta y se contarán, obteniendo la cantidad de respuestas que fueron 'a', 'b', 'c' ó 'd', además de la cantidad total de respuestas recibidas.</p> <p>I.3 Si se cumplen con éxito los requisitos del I.1 al I.2, se desplegará un mensaje que dirá: "cliente a respondido alternativa respuesta".</p>
Salida 1	suma_total, suma_a, suma_b, suma_c, suma_d.
Salida 2	Mensaje en I.3.

Tabla 3.9: Especificación Recibir Respuesta.

**3.3.1.10. Enviar datos por mail.**

Nombre	J) Enviar datos por mail
<b>Entradas:</b>	suma_total, suma_a, suma_b, suma_c, suma_d, pregunta, alternativa_a, alternativa_b, alternativa_c, alternativa_d, mensaje, asunto.
Procesos	<p>J.1 Para suma_total se utilizará un número entero que corresponde al total de alumnos que enviaron sus respuestas, suma_total viene dado por la suma de suma_a, suma_b, suma_c y suma_d .</p> <p>J.2 Para suma_a se utilizará un número entero que corresponde al total de alumnos que respondieron la alternativa “a”, cada vez que un alumno envía como respuesta la alternativa “a” suma_a aumenta en 1 su valor.</p> <p>J.3 Para suma_b se utilizará un número entero que corresponde al total de alumnos que respondieron la alternativa “b”, cada vez que un alumno envía como respuesta la alternativa “b” suma_b aumenta en 1 su valor.</p> <p>J.4 Para suma_c se utilizará un número entero que corresponde al total de alumnos que respondieron la alternativa “c”, cada vez que un alumno envía como respuesta la alternativa “c” suma_c aumenta en 1 su valor.</p> <p>J.5 Para suma_d se utilizará un número entero que corresponde al total de alumnos que respondieron la alternativa “d”, cada vez que un alumno envía como respuesta la alternativa “a” suma_a aumenta en 1 su valor.</p> <p>J.6 Para pregunta se utilizarán los requisitos ya especificados en D.1.</p> <p>J.7 Para alternativa_a se utilizarán los requisitos definidos en D.2.</p> <p>J.8 Para alternativa_d se utilizarán los requisitos definidos en D.3.</p> <p>J.9 Para alternativa_c se utilizarán los requisitos definidos en D.4.</p> <p>J.10 Para alternativa_d se utilizarán los requisitos definidos en D.5.</p>

Nombre	J) Enviar datos por mail
Procesos	<p>J.11 Para la variable mensaje, se utilizará un string que contendrá la información a enviar por email, específicamente la información que abarcará el mensaje será: “Han respondido suma_total de alumnos a la pregunta → ¿pregunta ?, dónde suma_a alumnos respondieron alternativa_a, suma_b alumnos respondieron alternativa_b, suma_c alumnos respondieron alternativa_c, suma_d alumnos respondieron alternativa_d”.</p> <p>J.12 Para asunto se usará un string que contendrá el mensaje: “Información App Clickers Respuestas Alumnos”.</p> <p>J.13 Para que el envío sea exitoso se deben cumplir los requisitos J.1 al J.12, en caso que no existan datos, es decir, no ha realizado ninguna respuesta, se desplegará el mensaje: “No existe información para enviar!!!”.</p>
Salida 1	suma_total, suma_a, suma_b, suma_c, suma_d, pregunta, alternativa_a, alternativa_b, alternativa_c, alternativa_d, mensaje, asunto.
Salida 2	Mensaje en J.13.

Tabla 3.10: Especificación Enviar datos por mail.

### 3.3.1.11. Desconectar.

Nombre	K) Desconectar
<b>Entradas:</b>	desconectar, nombre_usuario.
Procesos	<p>K.1 Para desconectar se utilizará un booleano el cual al iniciar la conexión tendrá el valor false.</p> <p>K.2 Al presionar en desconectar la variable cambia a su valor a true.</p> <p>K.3 Durante el ciclo de conexión su condición será la variable desconectar, por lo tanto este será infinito hasta que no se presione el botón y la variable cambie su valor.</p> <p>K.4 Para que desconectar sea exitoso se deben cumplir los requisitos de K.1 a K.3, en dicho caso se mostrará el mensaje: “nombre_usuario Eliminado”.</p>
Salida 1	desconectar.
Salida 2	Mensaje en K.4.

Tabla 3.11: Especificación Desconectar.

**3.3.1.12. Generar gráfico con respuestas.**

Nombre	L) Generar gráfico con respuestas
<b>Entradas:</b>	suma_a, suma_b, suma_c, suma_d.
Procesos	<p>L.1 Para suma_a se utilizarán los requisitos definidos en J.2.</p> <p>L.2 Para suma_b se utilizarán los requisitos definidos en J.3.</p> <p>L.3 Para suma_c se utilizarán los requisitos definidos en J.4.</p> <p>L.4 Para suma_d se utilizarán los requisitos definidos en J.5.</p> <p>L.5 Para generar el gráfico con las respuestas de los alumnos se utilizará un JavaScript proporcionado por Google llamado <i>Pie Chart Visualization</i>, un servicio gratuito para aplicaciones móviles, éste se muestra en la aplicación a través de un WebView al que se le pasa los parámetros suma_a, suma_b, suma_c, suma_d por medio de la función en Android <i>JavascriptInterface</i>, con estos datos se genera el gráfico de torta con las estadísticas de las respuestas.</p> <p>L.6 Para que Generar gráfico con respuestas sea exitoso se deben cumplir los requisitos L.1 al L.5 y se mostrará el gráfico con la suma de las respuestas enviadas por los alumnos.</p>
Salida 1	Gráfico en L.6 .

Tabla 3.12: Especificación Generar Gráfico con Respuestas.

**3.3.2. Requisitos no funcionales****3.3.2.1. Seguridad**

Los requisitos no funcionales de seguridad se muestran a continuación:

- Se deben seguir las recomendaciones pertinentes (Tomás., 2014) para no permitir ingeniería inversa en la generación del .apk de la aplicación, para evitar copias del código o reutilización del mismo.
- Se deben seguir los protocolos pertinentes de firma digital proporcionada por el IDE de desarrollo y no omitir estos al crear el ejecutable, para no sufrir alteraciones de código por terceros, recordar las contraseñas utilizadas y el seudónimo de desarrollador que se crea al momento de generar el ejecutable, dicha acción sirve

además para subir la aplicación a Play Store para ser descargada por cualquier usuario de smartphone con sistema operativo Android.

- El usuario de la aplicación en modo profesor debe proporcionar la información de la dirección IP a los alumnos que quiere que se conecten con él, y sólo a los que quiere, para evitar conexiones no deseadas.

### **3.3.2.2. Rendimiento**

A continuación se presentan los requisitos no funcionales de rendimiento:

- Se requiere un smartphone con sistema operativo Android para que pueda funcionar la aplicación.
- Se requiere que el sistema operativo sea como mínimo la versión 2.3 Gingerbread.
- Se requiere que el disco duro del smartphone tenga un espacio libre de a lo menos 3 MB.
- se requiere resolución de pantalla de 240x320 pixeles o superior.

### **3.3.2.3. Fiabilidad**

A continuación se describen los principales factores que se deben considerar para garantizar la fiabilidad del sistema a desarrollar y por ende, reducir al máximo la presencia de fallos futuros en el sistema:

- Se realizará configuraciones distintas de permisos dependiendo del tipo de usuario que interactúe. El tipo de usuario se indentificará al inicio de la aplicación con la decisión de ser alumno o profesor.
- El sistema debe mantener madurez suficiente, es decir, debe utilizar componentes y herramientas en cuanto al diseño, construcción y pruebas reconocidas que tengan mas de 3 años en el mercado y que tengan soporte por parte del fabricante.

#### **3.3.2.4. Flexibilidad**

En la configuración de los parámetros de instalación no es necesario que se requiera la modificación del código fuente de la instalación. El sistema debe proporcionar total independencia de la topología de red utilizada, es decir, el sistema debe ser capaz de funcionar en esquemas múltiples de comunicación, tanto para equipos conectados vía red WiFi.

#### **3.3.2.5. Portabilidad**

El sistema está diseñado para ser utilizado en cualquier versión de Android superior a las 2.3. La aplicación puede funcionar de igual forma en teléfonos o en tablet de 7,9 o 11 pulgadas. Para que la visualización sea prácticamente estándar se suspendió el giro de la pantalla horizontal para evitar degeneración de los componentes de la aplicación.

#### **3.3.2.6. Restricciones Técnicas**

En este apartado se presentan las restricciones técnicas del sistema:

- El software deberá ser compatible con las distintas versiones de Android desde la versión 2.3 a la 5.0.
- El sistema deberá utilizar como sistema de gestión de bases de datos SQLite, para almacenar los datos de las preguntas creadas por el profesor.
- el software será desarrollado bajo las exigencias y restricciones del lenguaje Android (Java).

#### **3.3.2.7. Restricciones de Negocio**

El software será completamente gratuito para todos los usuarios de la aplicación que se encuentren registrados en GooglePlay y tengan acceso a Play Store, para realizar la descarga del ejecutable.

#### **3.3.2.8. Otros requisitos no Funcionales**

La aplicación Teclera Virtual será desarrollada en su totalidad en idioma español.

# Capítulo 4

## Diseño

En este apartado se presenta de forma gráfica como es el funcionamiento de la aplicación. Se comienza mostrando el diagrama de actividad, seguido de los casos de uso, para continuar con un diagrama de componentes y consecutivamente mostrar lo que se puede representar de la base de datos utilizada, dado que ésta no se usa como tal. Este caso se explicará con mas detalle en la Sección 4.4. Por último, se muestran los wireframe del diseño de la aplicación con sus respectivas interfaces de usuario terminadas.

## 4.1. Diagrama de Actividad.

En la Figura 4.1 se presenta el diagrama de actividad para entender de forma simplificada lo que ocurre durante los diferentes procesos que se ejecutan en el sistema.

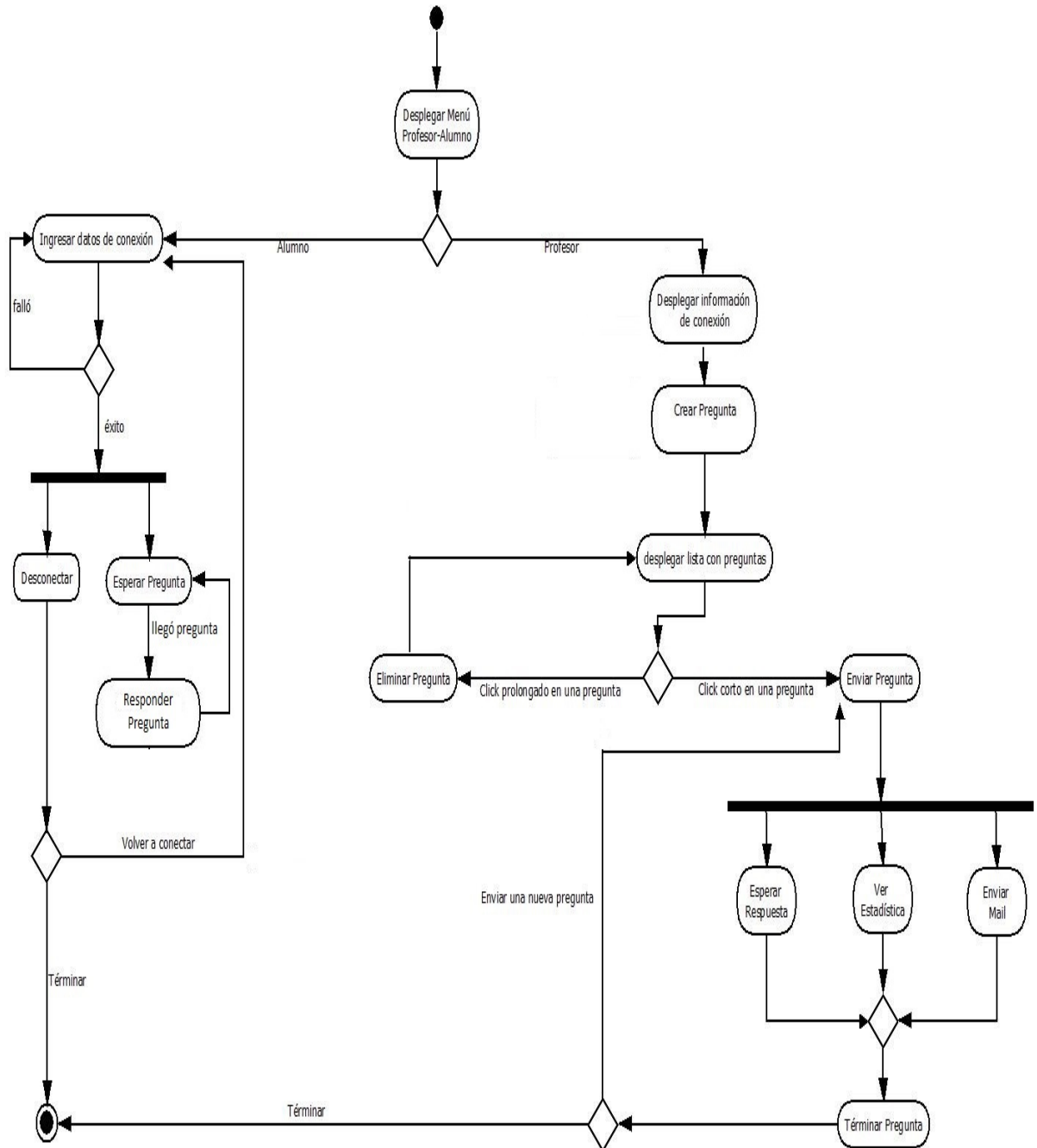


Figura 4.1: Diagrama de Actividad.

## 4.2. Diagrama Casos de Uso.

En la Figura 4.2 y 4.3 se muestran los casos de uso con cada actividad que realizan por separado cada actor de la aplicación, ya sea este profesor o alumno, además es usado para demostrar como se quiere que interactúe cada usuario con la aplicación y describir cómo se usa el sistema

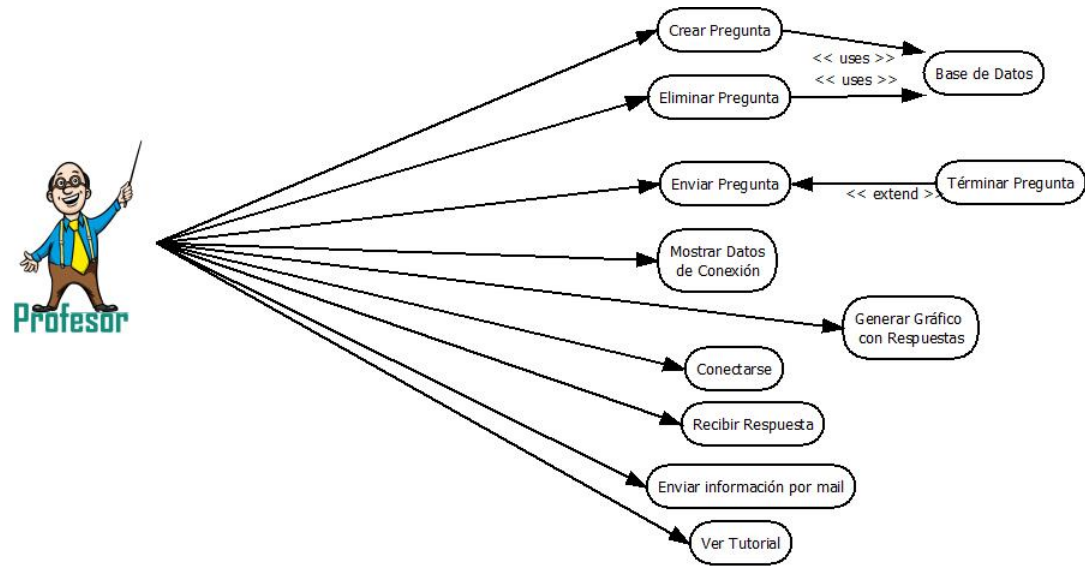


Figura 4.2: Caso de Uso Profesor.

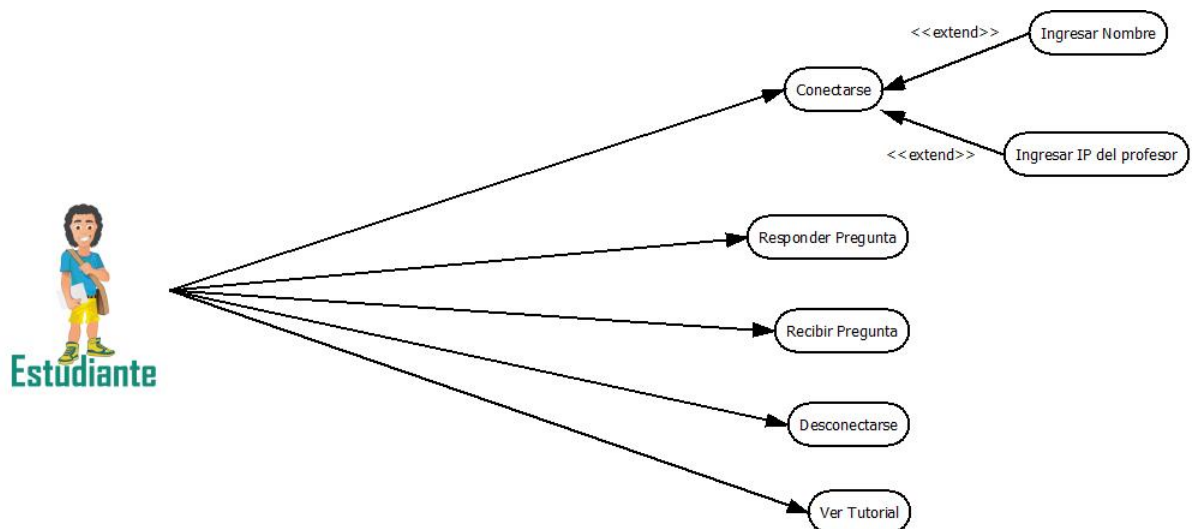


Figura 4.3: Caso de Uso Alumno.

### 4.3. Diagrama de Componentes.

En la siguiente Figura 4.4 se muestra el diagrama de componentes para entender como se relacionan los componentes físicos del sistema; en este diagrama se muestran todos los componentes utilizados, los cuales son simples paquetes representativos de los tipos de elementos utilizados en el software. El componente `clickers.apk` es el ejecutable de instalación de la aplicación, además, es una compresión de todos los demás componentes del sistema. Cada componente `.java` depende de un archivo `.xml` para su visualización en el `smatphone`. El componente `inicio.java` depende de `activity_inicio.xml`, estos archivos son los responsables de ejecutar la pantalla de inicio, para luego dar paso a `principal.java`, que depende por su parte de `activity_principal.xml`, que se encarga de mostrar el menú de selección de perfil. A su vez, `principal.java` depende de dos componentes, `MainActivity.java`, para el profesor, y `MainActivity_server.java`, para el alumno. los cuales dependen de otros archivos, que son: `activity_main_activity_server.xml` es el `.xml` principal; `DbHelper.JAVA`, que hereda de `Db_gestionar.java` y es utilizado para la creación de la base de datos; `dialogo_eliminar.xml`, es el dialogo para suprimir preguntas, el cual fue diseñado de forma externa al `.xml` principal; `nueva_pregunta.xml`, que es un dialogo para crear una nueva pregunta, creado de manera externa al `.xml` principal y `activity_grafico_torta.java`, que depende de `grafico_torta.xml` que tiene por objeto visualizar el gráfico con las respuestas de los alumnos, para `MainActivity_server.java`. Y para `MainActivity.java`, `activity_main_activity.xml`.

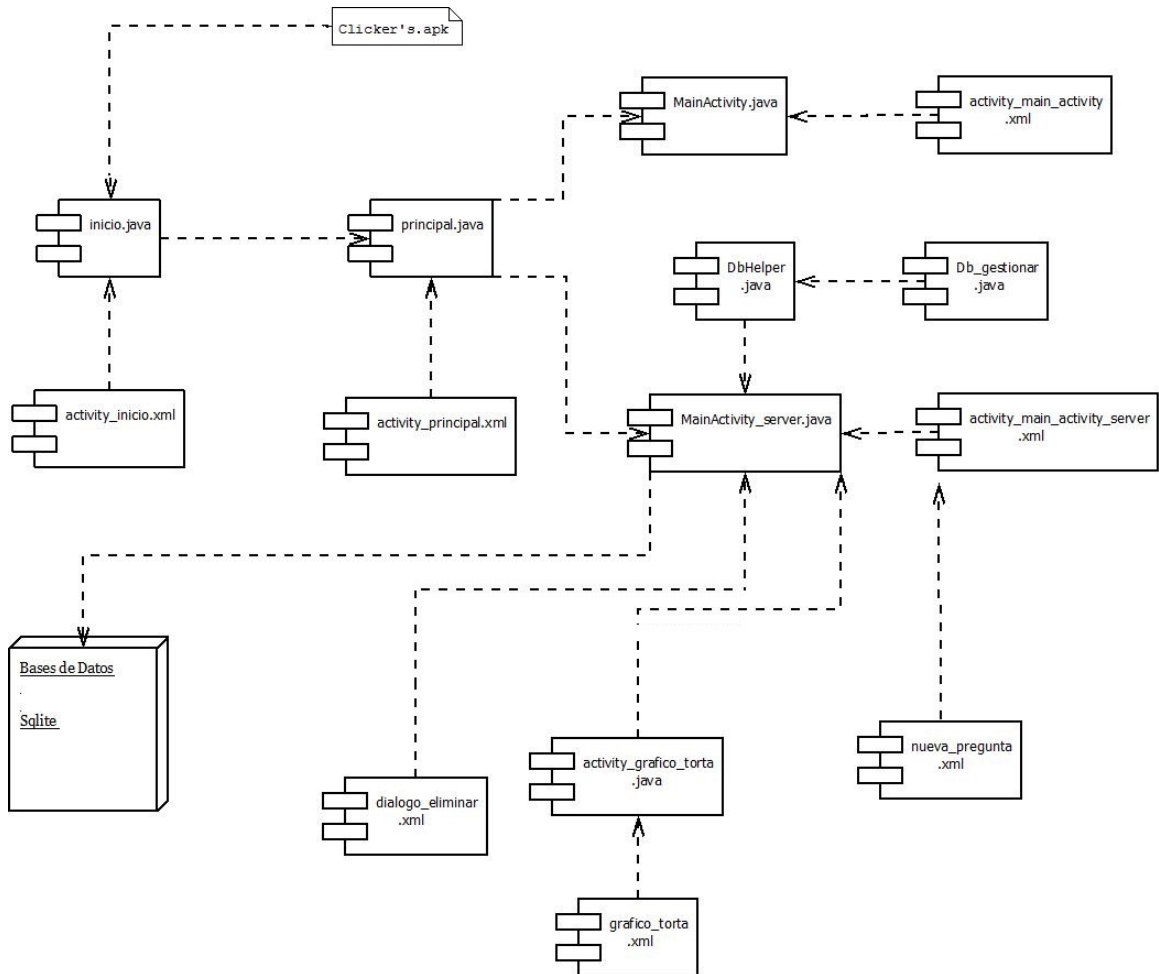


Figura 4.4: Diagrama de Componentes .

## 4.4. Bases de datos.

En este proyecto se habla del uso de SQLite para la implementación de la base de datos, sin embargo, ésta no es usada como tal, puesto que se crea sólo una tabla, destinada a almacenar las preguntas creadas por el profesor. Es decir, se utiliza con la finalidad de que dichas preguntas queden almacenadas en el dispositivo aún cuando se cierre la aplicación y se vuelva a abrir con posterioridad. De lo anterior se puede deducir que no se puede modelar la base de datos, dado que no existe una relación entre tablas, si no que más bien, existe una sola de ellas, la cual se utiliza con el mismo objetivo que los archivos de texto, pero con la diferencia de que se crea una base de datos que almacena únicamente una tabla.

#### 4.4.1. Diccionario de Datos.

En esta sección se muestra por medio de la Tabla 4.1 el diccionario de datos de la tabla “Preguntas” y no se presenta ni el modelo entidad-relación ni el modelo relacional, por el motivo mencionado anteriormente en la Sección 4.4.

ATRIBUTO	CLAVE	TIPO DATO	VALOR NULO	DOMINIO
codigo_pregunta	PK	Integer	NOT NULL	
pregunta		Text	NOT NULL	
alternativa_a		Text	NULL	
alternativa_b		Text	NULL	
alternativa_c		Text	NULL	
alternativa_d		Text	NULL	

Tabla 4.1: Diccionario de Datos.

#### 4.5. Interfaces de Usuario.

Seguidamente, se presentan las interfaces de usuario diseñadas para la aplicación móvil “Clickers”, primero, un *WireFrame* que es un esquema de página o plano de pantalla, una guía visual que representa el esqueleto o estructura visual de un Software. Luego, aquella que muestra la pantalla a utilizar en la aplicación .

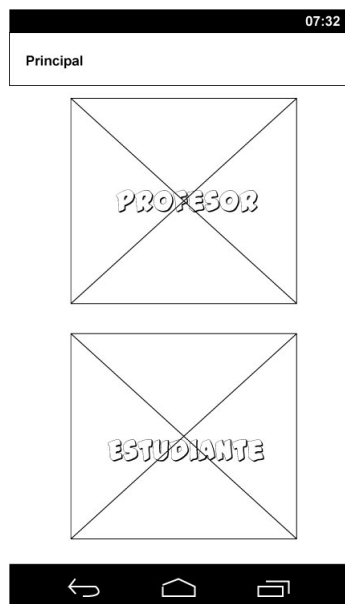


Figura 4.5: Wireframe Selección de Perfil .

En la Figura 4.5 se puede visualizar la pantalla correspondiente a la selección de perfil, donde el usuario de la aplicación decide si la utilizará en modo de Profesor o en modo de Alumno. La pantalla a utilizar se muestra en la Figura 4.6.



Figura 4.6: Selección de Perfil.



Figura 4.7: Wireframe Primera Vista Perfil Profesor .

En la Figura 4.7 se puede observar la pantalla correspondiente al momento de seleccionar el modo Profesor, en la parte superior se muestran los datos de conexión y también un botón para crear preguntas. La pantalla a utilizar se muestra en la Figura 4.8.

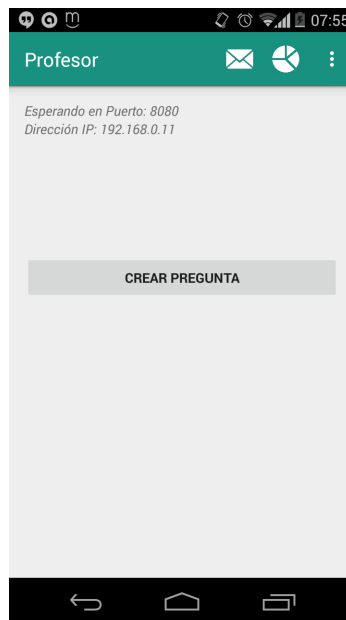


Figura 4.8: Primera Vista Perfil Profesor.

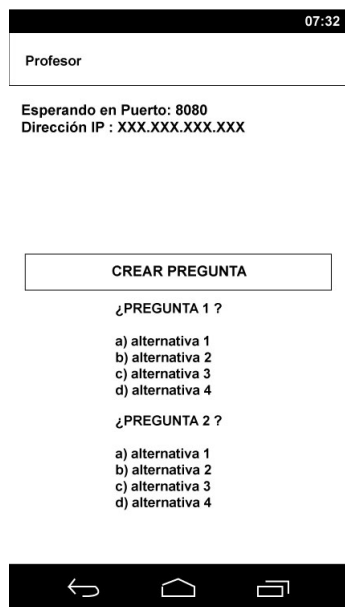


Figura 4.9: Wireframe Crear Pregunta .

En la Figura 4.9 se puede visualizar la pantalla correspondiente al momento de estar creadas las diferentes preguntas, se muestran en una lista y se pueden utilizar varias veces. La pantalla a utilizar se muestra en la Figura 4.10.

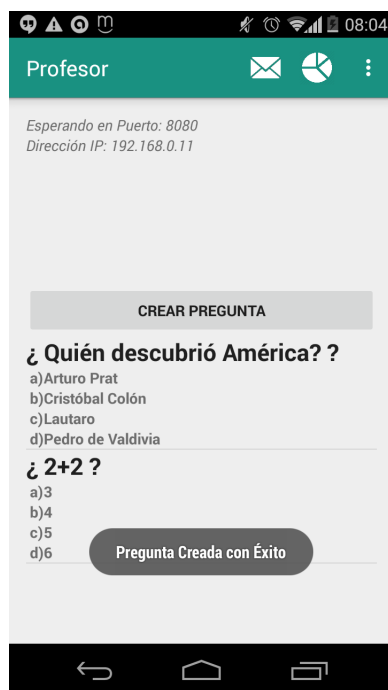


Figura 4.10: Crear Pregunta.

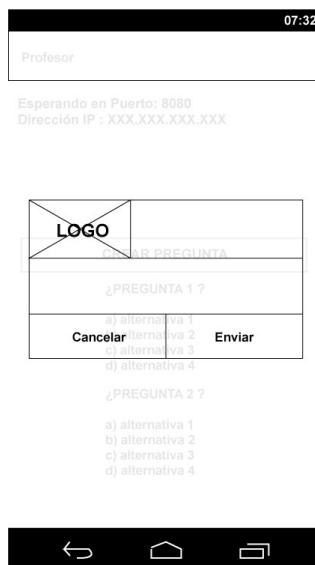


Figura 4.11: Wireframe Enviar Pregunta.

En la Figura 4.11 se puede ver la pantalla correspondiente al momento de hacer click en una pregunta en la cual aparece un dialogo con la confirmación de envío de una pregunta a los alumnos o bien retractarse al presionar cancelar. La pantalla a utilizar se muestra en la Figura 4.12.

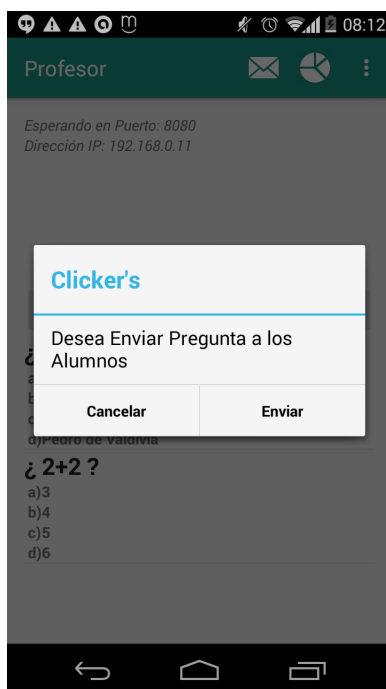


Figura 4.12: Enviar Pregunta.



Figura 4.13: Wireframe Finalizar Pregunta.

En la Figura 4.13 se puede visualizar la pantalla correspondiente al momento de enviar la pregunta, inmediatamente se muestra un botón para finalizar la pregunta e inicializar los datos de respuesta, al ser presionado este botón se puede enviar una nueva pregunta a los alumnos. La pantalla a utilizar se muestra en la Figura 4.14.

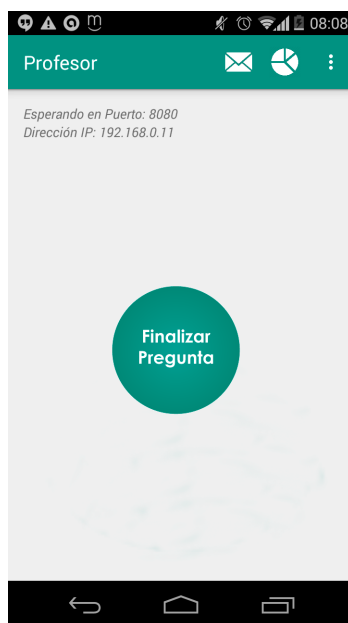


Figura 4.14: Finalizar Pregunta.

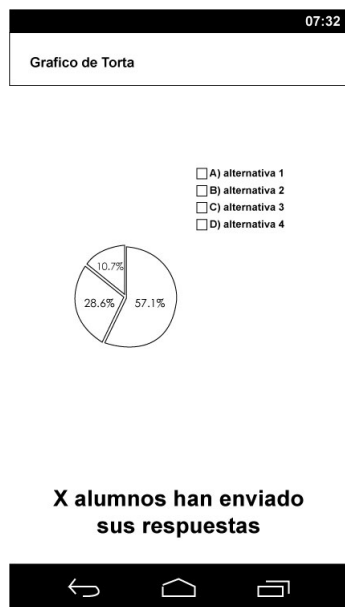


Figura 4.15: Wireframe Ver Gráfico de Torta.

En la Figura 4.15 se puede visualizar la pantalla correspondiente al momento presionar el icono de gráfico, siempre y cuando exista información se mostrará en gráfico de torta con los datos de las respuestas. La pantalla a utilizar se muestra en la Figura 4.16.

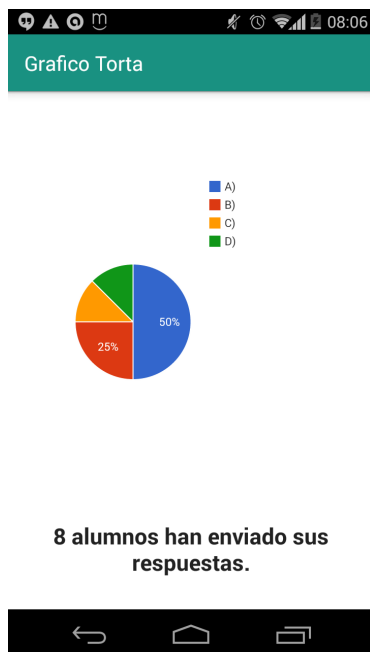


Figura 4.16: Ver Gráfico de Torta.

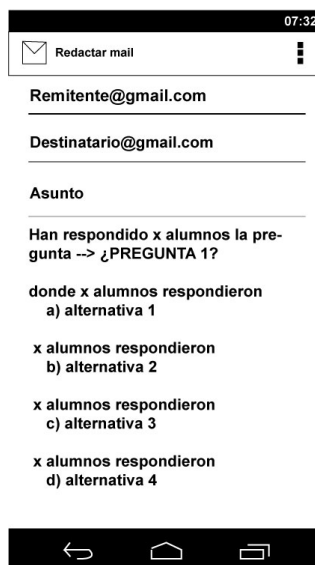


Figura 4.17: Wireframe Enviar Datos por Mail.

En la Figura 4.17 se puede advertir la pantalla correspondiente al momento presionar el icono de buzón, siempre y cuando exista información, se mostrará el mail con la información de la pregunta y las alternativas enviadas, como también la cantidad de alumnos que respondieron a cada alternativa. La pantalla a utilizar se muestra en la Figura 4.18.

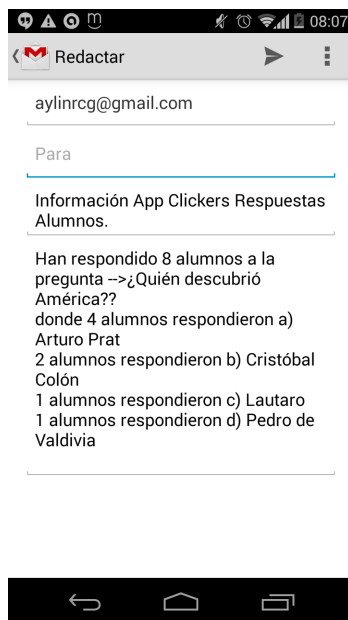


Figura 4.18: Enviar Datos por Mail.



Figura 4.19: Wireframe Primera Vista Perfil Alumno.

En la Figura 4.19 se puede contemplar la pantalla correspondiente al momento de seleccionar el modo Alumno. Aquí el cliente debe ingresar la información de la IP proporcionada por el profesor y asignarse un nombre de usuario. La pantalla a utilizar se muestra en la Figura 4.20.



Figura 4.20: Primera Vista Perfil Alumno.



**Esperando Pregunta .....**



Figura 4.21: Wireframe Esperar Pregunta.

En la Figura 4.21 se puede notar la pantalla correspondiente al momento de establecer la conexión y en el cual el alumno espera las preguntas a enviar del profesor. También se observa en la parte superior el botón desconectar, que es utilizado para romper la conexión entre alumno y profesor. La pantalla a utilizar se muestra en la Figura 4.22.



Figura 4.22: Primera Esperar Pregunta.

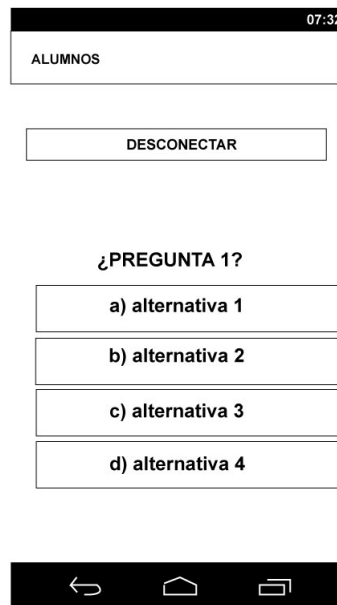


Figura 4.23: Wireframe Recibir Pregunta.

En la Figura 4.23 se puede visualizar la pantalla correspondiente al momento de recibir la pregunta enviada por el profesor y es aquí donde el alumno envía su respuesta. La pantalla a utilizar se muestra en la Figura 4.24.

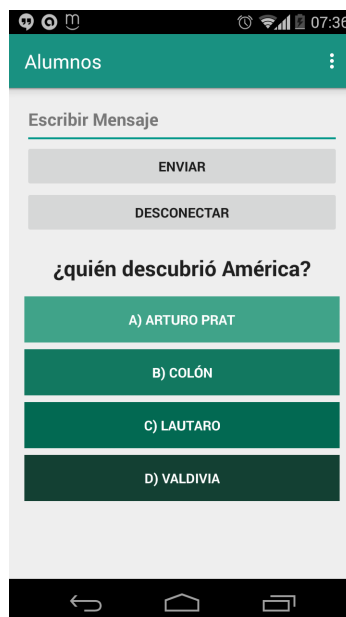


Figura 4.24: Recibir Pregunta.

# Capítulo 5

## Implementación

En este apartado se presenta la manera en que fueron implementadas las funcionalidades de la aplicación por parte del modo Profesor y del modo Alumno.

### 5.1. Funcionalidades

Para la implementación de cada una de las funcionalidades de la aplicación es necesario tener instalada previamente la última versión de JAVA JDK (Java Development Kit) y el SDK (Standard Development Kit) el cual contiene todas las herramientas necesarias para poder crear las aplicaciones y, posteriormente, instalar el IDE para desarrollar. Existen dos IDE de desarrollo: Eclipse con un plugin para la programación Android y Android Studio, esta última proporcionada por Google para el desarrollo de aplicaciones móviles en dicho sistema operativo. En este proyecto la herramienta seleccionada para desarrollar el software fue Android Studio 1.1.0 debido a sus constantes actualizaciones con mejoras notables de optimización a la hora de compilar y crear los ejecutables, además de su forma de usarse intuitiva, fácil de aprender y comprender; cabe destacar que a la vez se utilizará SQLite para el almacenamiento de información en la base de datos, como también un plugin de Mozilla Firefox llamado SQLite Manager para manipular o simplemente observar las bases de datos creadas en el teléfono. Es importante mencionar que en este capítulo no se explica como utilizar la aplicación, si no que los componentes de código de programación; para ello, se adjunta como anexo una guía de usuario o tutorial para tener completo conocimiento de como debe usarse la aplicación.

## 5.2. Implementación de las Funcionalidades

En el presente apartado se muestra explícitamente el código de las funcionalidades de la aplicación.

### 5.2.1. Conectar Profesor

Para la funcionalidad conectar profesor, en primer lugar se inicializan dos variables, una de la clase *ChatClient* y un string que servirá para guardar el mensaje a enviar. Inmediatamente después, se crea el constructor de la clase con los objetos que serán utilizados, posteriormente se crean los objetos de las clases *DataInputStream* y *DataOutputStream*, para escuchar y enviar mensajes respectivamente. Luego, se crea el socket obteniendo el nombre de usuario de los alumnos guardado en *connectClient.name* y la dirección IP, por último, se constuyen todas las validaciones con mensajes de error y de información para mantener constantemente informado al usuario por si hay algún inconveniente, por si se desconecta algún cliente o hubo un fallo inesperado. Cabe destacar que esta funcionalidad está implementada en la parte del profesor.

```

1 private class ConectarProfesor extends Thread {
2
3     Socket socket;
4     ChatClient connectClient;
5     String msgToSend = "";
6
7     CrearSocket(ChatClient client, Int puerto){
8         connectClient = client;
9         this.socket= socket;
10        client.socket = socket;
11        client.crearSocket = this;
12    }
13
14    @Override
15    public void run() {
16        try{
17            DataInputStream dataInputStream = null;

```

```
18     DataOutputStream dataOutputStream = null;  
19     dataInputStream = new DataInputStream(socket.  
20         getInputStream());  
21     dataOutputStream = new DataOutputStream(socket.  
22         getOutputStream());  
23     String cliente = dataInputStream.readUTF();  
24  
25     connectClient.name = cliente;  
26     socket = new Socket(cliente, ip);  
27  
28     } catch (IOException e) {  
29         e.printStackTrace();  
30     } finally {  
31         if (dataInputStream != null) {  
32             try {  
33                 dataInputStream.close();  
34             } catch (IOException e) {  
35                 // TODO Auto-generated catch block  
36                 e.printStackTrace();  
37             }  
38         }  
39  
40         if (dataOutputStream != null) {  
41             try {  
42                 dataOutputStream.close();  
43             } catch (IOException e) {  
44                 // TODO Auto-generated catch block  
45                 e.printStackTrace();  
46             }  
47         }  
48  
49         userList.remove(connectClient);  
50         MainActivity_server.this.runOnUiThread(new  
51             Runnable() {
```

```

50         @Override
51         public void run() {
52             Toast.makeText
53             (MainActivity_server.this,
54             connectClient.name + " Eliminado.", Toast.
55                 LENGTH_LONG).show();
56
57             msgLog += "-- " + connectClient.name + "
58                 Eliminado\n";
59             MainActivity_server.this.runOnUiThread(new
60                 Runnable() {
61
62                 @Override
63                 public void run() {
64                     chatMsg.setText(msgLog);
65                 }
66             });
67         }
68     }
69 }

```

### 5.2.2. Mostrar Datos de Conexión

Para mostrar los datos de conexión, que son dos; puerto y dirección IP, primero es necesario obtener ambos. Al puerto se le asigna por defecto el valor 8080, y el de la dirección IP se obtiene a través de la clase *getIpAddress()*. Una vez que se tienen estos dos valores, se muestran los datos obtenidos en la parte superior del modo profesor por medio de un *TextView*.

```

1
2     static final int SocketServerPORT = 8080;
3     TextView infoIp, infoPort;
4
5     serverSocket = new ServerSocket(SocketServerPORT);
6         MainActivity_server.this.runOnUiThread(new
7             Runnable() {

```

```
7
8         @Override
9         public void run() {
10             infoPort.setText("Esperando en Puerto:
11                 "
12                 + serverSocket.getLocalPort());
13         }
14     });
15
16     infoIp = (TextView) findViewById(R.id.infoip);
17     infoPort = (TextView) findViewById(R.id.infoport);
18
19     infoIp.setText(getIpAddress()); //obteniendo la IP

```

```
1 private String getIpAddress() {
2     String ip = "";
3     try {
4         Enumeration<NetworkInterface> enumNetworkInterfaces
5             = NetworkInterface
6                 .getNetworkInterfaces();
7         while (enumNetworkInterfaces.hasMoreElements()) {
8             NetworkInterface networkInterface =
9                 enumNetworkInterfaces
10                    .nextElement();
11             Enumeration<InetAddress> enumInetAddress =
12                 networkInterface
13                    .getInetAddresses();
14             while (enumInetAddress.hasMoreElements()) {
15                 InetAddress inetAddress = enumInetAddress.
16                    nextElement();
17
18                 if (inetAddress.isSiteLocalAddress()) {
19                     ip += "Dirección IP: "
```

```

16         + InetAddress.getHostAddress()
17         + "\n";
18     }
19 }
20
21 }
22
23 } catch (SocketException e) {
24     // TODO Auto-generated catch block
25     e.printStackTrace();
26     ip += "Algo ocurri mal! " + e.toString() + "\n";
27 }
28
29 return ip;
30 }

```

### 5.2.3. Crear Pregunta

Esta funcionalidad comienza cuando el profesor presiona el botón crear pregunta, una vez presionado éste, se crea un diálogo con cinco *Labels* en los cuales se deben ingresar la pregunta y las cuatro alternativas respectivamente. Realizado el ingreso, se debe presionar el botón guardar para que estos datos sean almacenados en la tabla Preguntas. Los datos se capturan por medio del comando `.getText().toString()`; y se guardan en la base de datos a través de `gestionar.insertar(pregunta.trim(),a.trim(),b.trim(),c.trim(),d.trim())`; Por último, se refresca la lista con las preguntas para que se logre ver inmediatamente la nueva pregunta creada.

```

1 public void apretar_btn_crear(final View view) {
2
3     LayoutInflater inflater = LayoutInflater.from(
4         view.getContext());
5     View alertapregunta = inflater.inflate(R.layout.
6         nueva_pregunta, null);

```

```
5
6     AlertDialog.Builder alertDialog = new AlertDialog.
7         Builder(view.getContext());
8     alertDialog.setView(alertapregunta);
9     alertDialog.setTitle("Nueva Pregunta...");
10    final TextView txtpregunta = (TextView) alertapregunta.
11        findViewById(R.id.pregunta);
12    final TextView txtalternativaa = (TextView)
13        alertapregunta.findViewById(R.id.alternativa_a);
14    final TextView txtalternativab = (TextView)
15        alertapregunta.findViewById(R.id.alternativa_b);
16    final TextView txtalternativac = (TextView)
17        alertapregunta.findViewById(R.id.alternativa_c);
18    final TextView txtalternativad = (TextView)
19        alertapregunta.findViewById(R.id.alternativa_d);
20
21    alertDialog
22        .setCancelable(false)
23        .setPositiveButton("Guardar", new
24            DialogInterface.OnClickListener() {
25                public void onClick(DialogInterface dialog,
26                    int id) {
27
28                    String pregunta = txtpregunta.getText()
29                        .toString();
30                    String a = txtalternativaa.getText().
31                        toString();
32                    String b = txtalternativab.getText().
33                        toString();
34                    String c = txtalternativac.getText().
35                        toString();
36                    String d = txtalternativad.getText().
37                        toString();
```

```
27
28
29
30     if (pregunta.equals("")) {
31         Toast.makeText(MainActivity_server.
32             this, "Ingresar Pregunta",
33             Toast.LENGTH_LONG).show();
34         return;
35     }
36     if (a.equals("") && b.equals("") || a.
37         equals("") && c.equals(""))
38         || a.equals("") && c.equals("") || a.
39         equals("") && d.equals("")
40         || b.equals("") && c.equals("") || b.
41         equals("") && d.equals("")
42         || c.equals("") && c.equals("")) {
43     Toast.makeText(MainActivity_server.this, "Debe ingresar a lo
44         menos dos alternativa",
45         Toast.LENGTH_LONG).show();
46     return;
47 }
48
49     gestionar.insertar(pregunta.trim(), a.
50         trim(), b.trim(), c.trim(), d.trim())
51     ;
52 //lista = (ListView) findViewById(R.id.
53     LVPregunta);
54 //Refrescar Lista
55 cursor = gestionar.cargarCursorPregunta
56     ();
57 String [] from = new String[]{
58     gestionar.TABLE_Pregunta, gestionar.
59     TABLE_A1, gestionar.TABLE_A2,
60     gestionar.TABLE_A3, gestionar.
61     TABLE_A4 };

```

```

49         int [] to = new int[]{R.id.pregunta,R.
50             id.a,R.id.b,R.id.c,R.id.d};
51         adapter = new SimpleCursorAdapter(view.
52             getContext(), R.layout.para_listview
53             ,cursor,from,to,0);
54         lista.setAdapter(adapter);
55
56         Toast.makeText(MainActivity_server.this
57             , "Pregunta Creada con xito ",
58             Toast.LENGTH_LONG).show();
59         //BDD
60     }
61 })
62 .setNegativeButton("Cancelar", new
63     DialogInterface.OnClickListener() {
64         public void onClick(DialogInterface dialog,
65             int id) {
66             dialog.cancel();
67         }
68     });
69 AlertDialog alertD = alertDialog.create();
70 alertD.show();
71 }

```

#### 5.2.4. Eliminar Pregunta

Para esta funcionalidad se crea un diálogo llamado *alerta\_eliminar*. Este diálogo surge al momento de hacer un click prolongado en la pantalla que indique una pregunta, en ese diálogo se muestran dos botones, uno para confirmar la eliminación y otro para cancelarla. Al momento de confirmar la eliminación se guarda la pregunta seleccionada a través del comando `((TextView(view.findViewById(R.id.pregunta))).getText().toString());` en la variable *string\_pregunta* y posteriormente se elimina la pregunta junto a sus alternativas por medio de `gestionar.eliminar(string_pregunta);`

```
1 //Al hacer click largo en una pregunta
2     lista.setOnItemLongClickListener(new AdapterView.
3         OnItemLongClickListener() {
4             @Override
5             public boolean onItemLongClick(AdapterView<?>
6                 parent, final View view, final int position,
7                 long id) {
8                 LayoutInflater inflater = LayoutInflater.
9                     from(view.getContext());
10                View alerta_eliminar = inflater.inflate(R
11                    .layout.dialogo_eliminar, null);
12                AlertDialog.Builder ventana_eliminar = new
13                    AlertDialog.Builder(view.getContext());
14
15                ventana_eliminar.setTitle("Clicker's");
16                ventana_eliminar.setPositiveButton("Eliminar",
17                    new DialogInterface.OnClickListener() {
18                        @Override
19                        public void onClick(DialogInterface dialog,
20                            int which) {
21                            //eliminar pregunta seleccionada
22
23                            String string_pregunta = ((TextView) (
24                                view.findViewById(R.id.pregunta))).
25                                getText().toString();
26                            gestionar.eliminar(string_pregunta);
27                            Toast.makeText(MainActivity_server.this
28                                , "Pregunta eliminada con xito !!!"
29                                ,
30                                    Toast.LENGTH_LONG).show();
31                            //Refrescar Lista
32                            cursor = gestionar.cargarCursorPregunta
33                                ();
34                            String [] from = new String[] {
```

```

22         gestionar.TABLE_Pregunta,gestionar.
23         TABLE_A1,gestionar.TABLE_A2,
24         gestionar.TABLE_A3,gestionar.
25         TABLE_A4 );
26     int [] to = new int[]{R.id.pregunta,R.
27         id.a,R.id.b,R.id.c,R.id.d};
28     adapter = new SimpleCursorAdapter(view.
29         getContext(), R.layout.para_listview
30         , cursor, from, to, 0);
31     lista.setAdapter(adapter);
32     }
33     }).setNegativeButton("Cancelar", new
34     DialogInterface.OnClickListener() {
35         @Override
36         public void onClick(DialogInterface dialog,
37             int which) {
38             //cancelar envio
39             dialog.cancel();
40         }
41     });
42     ventana_eliminar.show();
43
44     return false;
45 }
46 });

```

### 5.2.5. Enviar Datos por Mail

Al momento de presionar el menú con forma de buzón se da inicio a esta funcionalidad. La cual, a través de un intent, redirecciona a la redacción de un mail, que en esta función llevará por defecto el asunto y el mensaje. Cabe mencionar que el destinatario al que se le enviará la información aparece por defecto como el usuario registrado en el smartphone. Todo esto siempre y cuando la variable *str\_pregunta* sea distinta de vacío,

lo cual significa que el profesor ya envió una pregunta.

```

1  if (id == R.id.compartir_mail) {
2      if (!str_pregunta.equals("")) {
3          Intent emailIntent = new Intent(Intent.
4              ACTION_SEND);
5          emailIntent.setData(Uri.parse("mailto:"));
6          asunto = "Informaci n App Clickers Respuestas
7              Alumnos.";
8          int suma_respuestas =a+b+c+d;
9          mensaje ="Han respondido "+suma_respuestas+"
10             alumnos a la pregunta -->"+str_pregunta+"\n"
11             +
12             "donde "+a+" alumnos respondieron "+str_a+"\n"+
13             +b+" alumnos respondieron "+str_b+"\n"+
14             +c+" alumnos respondieron "+str_c+"\n"+
15             +d+" alumnos respondieron "+str_d+"\n";
16          emailIntent.putExtra(Intent.EXTRA_SUBJECT,
17              asunto);
18          emailIntent.putExtra(Intent.EXTRA_TEXT, mensaje
19              );
20          emailIntent.setType("message/rfc822");
21          startActivity(Intent.createChooser(emailIntent,
22              "Email "));
23      }else{
24          Toast.makeText(MainActivity_server.this, "No
25             existe informaci n para enviar!!!",
26              Toast.LENGTH_LONG).show();
27      }
28  }
29
30  return super.onOptionsItemSelected(item);
31
32  }

```

### 5.2.6. Generar Gráfico con Respuestas

Para generar el gráfico de torta, se obtienen desde la main la cantidad de respuestas que arrojaron las alternativas a,b,c ó d, respectivamente. Al momento de presionar el menú en forma de gráfico, se envían estos valores a la clase *Grafico\_Torta*, para ser utilizados posteriormente en la formación del mismo. Siempre y cuando la variable *suma\_respuestas* sea distinta de cero, donde *suma\_respuestas* es la cantidad total de respuestas.

```

1  if (id == R.id.grafico_torta) {
2  int suma_respuestas = a+b+c+d;
3      if (suma_respuestas!= 0) {
4          Intent grafico = new Intent(this, Grafico_Torta
5              .class);
6          grafico.putExtra("NUM1", a);
7          grafico.putExtra("NUM2", b);
8          grafico.putExtra("NUM3", c);
9          grafico.putExtra("NUM4", d);
10         startActivity(grafico);
11     }else{
12         Toast.makeText(MainActivity_server.this, "No
13             existe informacín para ver el gr fico!!!"
14             ,
15             Toast.LENGTH_LONG).show();
16     }
17 }
18 }

```

Una vez enviado los datos a la clase *Grafico\_Torta* se crea un *WebView* en el cual se mostrará el gráfico. Esta clase sólo recibe los datos, ya que los manda a un *html* que es interpretado por un servidor de Google, dado que esta es una herramienta para desarrolladores facilitada por google para la creación de gráficos de torta (Developers, 2015). En breves palabras, esta clase muestra en el smartphone una página web con los datos que se le entregó con anterioridad y el gráfico creado.

```

1  public class Grafico_Torta extends ActionBarActivity {

```

```
2
3     WebView webView;
4     TextView cantidad_respuestas;
5     int num1, num2, num3, num4;
6     int respuestas=0;
7
8     @Override
9     protected void onCreate(Bundle savedInstanceState) {
10         super.onCreate(savedInstanceState);
11         setContentView(R.layout.activity_grafico__torta);
12
13         Intent grafico = getIntent();
14         num1 = grafico.getIntExtra("NUM1", 0);
15         num2 = grafico.getIntExtra("NUM2", 0);
16         num3 = grafico.getIntExtra("NUM3", 0);
17         num4 = grafico.getIntExtra("NUM4", 0);
18
19
20         webView = (WebView)findViewById(R.id.web);
21         cantidad_respuestas = (TextView) findViewById(R.id.
22             cantidad_respuestas);
23         webView.addJavascriptInterface(new WebAppInterface(), "
24             Android");
25         webView.getSettings().setJavaScriptEnabled(true);
26         webView.getSettings().setUseWideViewPort(true);
27         webView.setVerticalScrollBarEnabled(false);
28         webView.setHorizontalScrollBarEnabled(false);
29         webView.loadUrl("file:///android_asset/chart.html");
30
31         respuestas = num1 + num2 + num3 + num4 + num5;
32         cantidad_respuestas.setText(""+respuestas+" alumnos han
33             enviado sus respuestas.");
34     }
```

```
34 public class WebAppInterface {
35
36     @JavascriptInterface
37     public int getNum1() {
38         return num1;
39     }
40
41     @JavascriptInterface
42     public int getNum2() {
43         return num2;
44     }
45
46     @JavascriptInterface
47     public int getNum3() {
48         return num3;
49     }
50
51     @JavascriptInterface
52     public int getNum4() {
53         return num4;
54     }
55
56     @JavascriptInterface
57     public int getNum5() {
58         return num5;
59     }
60 }
61
62 }
```

### 5.2.7. Enviar Pregunta

La funcionalidad Enviar Pregunta está creada en la parte del profesor, el mensaje a enviar se recibe como parámetro de entrada, en este caso la pregunta, y un *List* que contiene a los clientes que se conectaron. Se crea un ciclo *for* con el tamaño del *List* y

se envía por medio de la función *chatsendmsg* de la *chatThread* a todos los clientes conectados, en consecuencia, se observa un mensaje con el nombre de todos los usuarios a los que fue enviada la pregunta.

```

1 public void EnviarPregunta(String msg, List userList){
2
3     for(int i=0; i<userList.size(); i++){
4         userList.get(i).chatThread.sendmsg(msg);
5         msgLog += "- Enviado a " + userList.get(i).name + "
6             \n";
7     }

```

### 5.2.8. Recibir Respuesta

Recibir respuesta es utilizado en el modo profesor, el código se enmarca dentro de un ciclo *while* infinito dado que su condición es true. Luego, se crea la condición de que si *dataInputStream* tiene información y guarda lo que contiene en la variable *newMsg*, posteriormente a la variable *msgLog* se le asigna la información con el nombre del usuario que respondió cierta pregunta y se muestra con un texto en la aplicación. Consecutivamente se pregunta si la respuesta recibida fue la alternativa 'a', 'b', 'c' ó 'd' y se cuentan el total de respuestas con el total de alumnos que respondieron a cada respuesta. Finalizando se pregunta si existe algún mensaje a enviar por parte del profesor al alumno, si es así, se envía y se vacía el mensaje, para en un futuro enviar uno nuevo.

```

1 while (true) {
2     if (dataInputStream.available() > 0) {
3         String newMsg = dataInputStream.readUTF();
4         msgLog += n + " respondi alternativa " +
5             newMsg;
6         MainActivity_server.this.runOnUiThread(new
7             Runnable() {
8                 @Override
9                 public void run() {

```

```

8         chatMsg.setText(msgLog);
9     }
10    });
11    if (newMsg.equals("a\n")) suma_a++;
12    if (newMsg.equals("b\n")) suma_b++;
13    if (newMsg.equals("c\n")) suma_c++;
14    if (newMsg.equals("d\n")) suma_d++;
15        Int suma_total = suma_a+suma_b+
16                suma_c+suma_d;
17    }
18    if (!msgToSend.equals("")) {
19        dataOutputStream.writeUTF(msgToSend);
20        dataOutputStream.flush();
21        msgToSend = "";
    } }

```

### 5.2.9. Conectar Alumno

Conectar alumno es una función implementada en el modo alumno que recibe como parámetros el nombre del usuario, la dirección IP, el puerto y el socket. Primero, se crea el constructor y los objetos para escuchar y recibir mensajes. Luego, se crea un objeto socket que se obtuvo de la primera funcionalidad de este apartado, se continúa estableciendo la conexión con el puerto 8080 predefinido y con el nombre de usuario y dirección IP obtenidos del *editText* en que se ingresaron. Inmediatamente después, se envía al profesor el nombre del usuario que se conecta para guardarlo en la *List*. Se finaliza con las validaciones pertinentes a un programa cliente-servidor, verificación de errores en la conexión, detección de caídas y validaciones. Si todo lo anterior ocurre bien, se ocultan los *editText* y se pasa al modo de esperar pregunta.

```

1 private class ConectarAlumno extends Thread {
2
3     String name;
4     String dstAddress;
5     int dstPort;

```

```
6      String msgToSend = "";
7      boolean goOut = false;
8      editTextUserName = (EditText) findViewById(R.id.
          username);
9      ConectarAlumno(String name, String address, int port,
          CrearSocket socket) {
10         this.name = name;
11         dstAddress = address;
12         dstPort = port;
13     }
14     @Override
15     public void run() {
16         name =
17         Socket socket = null;
18         DataOutputStream dataOutputStream = null;
19         DataInputStream dataInputStream = null;
20         try {
21             socket = new CrearSocket(dstAddress, dstPort);
22             conexion = EstablecerConexion(socket, name,
                address, port)
23             dataOutputStream = new DataOutputStream(socket.
                getOutputStream());
24             dataInputStream = new DataInputStream(socket.
                getInputStream());
25             dataOutputStream.writeUTF(name);
26             dataOutputStream.flush();
27             msgLog = dataInputStream.readUTF();
28             MainActivity.this.runOnUiThread(new Runnable()
                {
29                 @Override
30                 public void run() {
31                     Toast.makeText
32                     (MainActivity.this, msgLog , Toast.
                        LENGTH_LONG).show();
33                 }
            }
```

```
34     });
35 } catch (UnknownHostException e) {
36     e.printStackTrace();
37     final String eString = e.toString();
38     MainActivity.this.runOnUiThread(new Runnable()
39         {
40         @Override
41         public void run() {
42             Toast.makeText
43             (MainActivity.this, eString, Toast.
44                 LENGTH_LONG).show();
45         }
46     });
47 } catch (IOException e) {
48     e.printStackTrace();
49     final String eString = e.toString();
50     MainActivity.this.runOnUiThread(new Runnable()
51         {
52         @Override
53         public void run() {
54             Toast.makeText
55             (MainActivity.this, eString, Toast.
56                 LENGTH_LONG).show();
57         }
58     });
59 } finally {
60     if (socket != null) {
61         try {
62             socket.close();
63         } catch (IOException e) {
64             // TODO Auto-generated catch block
65             e.printStackTrace();
66         }
67     }
68     if (dataOutputStream != null) {
```

```

65         try {
66             dataOutputStream.close();
67         } catch (IOException e) {
68             // TODO Auto-generated catch block
69             e.printStackTrace();
70         }}
71     if (dataInputStream != null) {
72         try {
73             dataInputStream.close();
74         } catch (IOException e) {
75             // TODO Auto-generated catch block
76             e.printStackTrace();
77         }}
78     MainActivity.this.runOnUiThread(new Runnable()
79     {
80         @Override
81         public void run() {
82             loginPanel.setVisibility(View.VISIBLE);
83             chatPanel.setVisibility(View.GONE);
84             layout_pregunta.setVisibility(View.GONE
85             );
86         } });
87     }}

```

### 5.2.10. Recibir Pregunta

En recibir pregunta el alumno termina con éxito el proceso de conexión. Es un ciclo *while* que depende de la variable desconectar que en un comienzo tiene el valor de false. En principio, recibe por medio de *dataInputStream* el string enviado por el profesor. Vale la pena mencionar que el profesor no envía preguntas ni alternativas por separado, sino que todo en uno. Luego de recibir el string lo guarda en la variable *new\_pregunta* y por medio de la función *split* se separa en más strings por salto de línea, obteniendo por separado lo que será la pregunta y las distintas alternativas. Se termina vaciando la variable *new\_pregunta* para que no ocurran problemas con posterioridad.

```
1
2 while (!desconectar) {
3     if (dataInputStream.available() > 0) {
4         new_pregunta= dataInputStream.readUTF()
5         ;
6         final String lines[] = new_pregunta.
7             split("\\n");
8
9         MainActivity.this.runOnUiThread(new
10             Runnable() {
11                 @Override
12                 public void run() {
13                     layout_pregunta.setVisibility(View.
14                         VISIBLE);
15                     text_pregunta.setText(lines[0]);
16                     btn_a.setText(lines[1]);
17                     btn_b.setText(lines[2]);
18                     btn_c.setText(lines[3]);
19                     btn_d.setText(lines[4]);
20                 }
21             });
22
23         new_pregunta= ""; }
24 }
```

### 5.2.11. Enviar Respuesta

Enviar respuesta es una funcionalidad del sistema encargada del alumno, trata de lo que constantemente se ha hablado en este capítulo; sobre los objetos que escuchan y envían mensajes, en este caso se usa *dataOutputStream* para enviar mensajes. Se crea una función general para el enviar las respuestas y se particulariza preguntando en un principio cual fue la respuesta enviada 'a', 'b', 'c', ó 'd' y, dependiendo cual de estas sea, se envía el caracter correspondiente. Seguidamente se muestra un mensaje de éxito

de envío para el usuario y se finaliza ocultando los botones y la pregunta, de modo tal que no sea contestada más de una vez por un mismo alumno.

```
1 private void sendMsg(String msg) {
2     msgToSend = msg;
3     dataOutputStream.writeUTF(msgToSend);
4     dataOutputStream.flush();
5 }
6 private void EnviarRespuesta(respuesta) {
7     if(resuesta == 'a') {
8         EstablecerConexion.sendMsg("a" + "\n");
9         Toast.makeText(MainActivity.this,
10            "Respuesta enviada con xito !!!",
11            Toast.LENGTH_LONG).show();
12         layout_pregunta.setVisibility(View.GONE);
13     }
14     if(resuesta == 'b') {
15         EstablecerConexion.sendMsg("b" + "\n");
16         Toast.makeText(MainActivity.this,
17            "Respuesta enviada con xito !!!",
18            Toast.LENGTH_LONG).show();
19         layout_pregunta.setVisibility(View.GONE);
20     }
21     if(resuesta == 'c') {
22         EstablecerConexion.sendMsg("c" + "\n");
23         Toast.makeText(MainActivity.this,
24            "Respuesta enviada con xito !!!",
25            Toast.LENGTH_LONG).show();
26         layout_pregunta.setVisibility(View.GONE);
27     }
28     if(resuesta == 'd') {
29         EstablecerConexion.sendMsg("d" + "\n");
30         Toast.makeText(MainActivity.this,
31            "Respuesta enviada con xito !!!",
32            Toast.LENGTH_LONG).show();
```

```
33     layout_pregunta.setVisibility(View.GONE);
34     }
35     if(!msgToSend.equals("")){
36
37         msgToSend = "";
38     }
39 }
```

### 5.2.12. Desconectar

Para la funcionalidad desconectar, simplemente se le cambia su valor original de *false* a *true*, el cual produce que termine el ciclo infinito *while* y por ende, la conexión del alumno.

```
1 private void disconnect() {
2     desconectar = true;
3 }
```

# Capítulo 6

## Pruebas

En este Capítulo se presenta el plan de pruebas que validan el óptimo funcionamiento de la aplicación desarrollado en este proyecto.

### **6.1. Resumen del Plan de Pruebas.**

#### **6.1.1. Descripción**

El plan de pruebas describe los pasos que se deben seguir para verificar que el sistema construido satisfaga los requerimientos. En este caso, la aplicación será probada en cuanto a usabilidad, pruebas de requerimientos y un test de strees para considerar las limitaciones de la aplicación y de esta forma determinar en que momento ocurriría una falla en el sistema.

#### **6.1.2. Propósito**

El propósito principal de realizar una prueba al sistema es que éste sea mejorado. En consecuencia, en la eventualidad de que no apruebe dicho examen satisfactoriamente, corresponde efectuar las modificaciones pertinentes y se pone a prueba nuevamente el sistema.

### **6.1.3. Objetivos del plan de pruebas**

El objetivo del plan de pruebas es reunir la información necesaria para planificar y controlar el esfuerzo de pruebas al sistema. Así también, busca identificar los elementos o datos de prueba e identificar eficiencia en usabilidad en el software.

### **6.1.4. Referencias**

Para realizar las pruebas se debe considerar toda la documentación necesaria de la aplicación, ésta última se encuentra en este mismo proyecto (Véase Capítulo 3 y Capítulo 4). Además, se debe contar con una plataforma de pruebas preparada a la que se le conoce como banco de pruebas.

## **6.2. Plan de Pruebas.**

A continuación se muestra el estudio con las pruebas de requerimientos, usabilidad y test de stress realizadas a la aplicación móvil teclera virtual sobre una plataforma Android para la UCSC.

### **6.2.1. Pruebas de Requerimientos**

#### **6.2.1.1. Descripción**

El objetivo de las pruebas de requerimientos es buscar diferencias entre los requerimientos y el software en funcionamiento, además de encontrar posibles defectos en el sistema que no signifiquen una deficiencia general del sistema, si no solo de la parte especificada o el requisito especificado. En esta inspección se busca evaluar la consistencia, completitud y ambigüedad de los requisitos funcionales individualmente considerados, estos requisitos deben ser analizados por todos los usuarios involucrados con el sistema, en este caso en particular, por el profesor y el alumno.

#### **6.2.1.2. Criterio para la evaluación.**

En esta fase de la prueba se verifica drásticamente si un requisito cumple o no cumple con lo esperado; no existen evaluaciones intermedias ni nada por el estilo, con

lo cual, si el requisito se cumple se muestra una C y en el caso contrario, de no cumplir el requisito se denota con NC.

**6.2.1.3. Condiciones de ejecución**

La ejecución del software se hizo en condiciones de un smartphone, es decir, un teléfono móvil con varias aplicaciones instaladas y ejecutandose en segundo plano. El modelo del dispositivo móvil de pruebas fue un Samsung Galaxy Tab 4 con versión de Android: 4.1.2.

**6.2.1.4. Requerimientos de Ambiente de Pruebas**

El ambiente de pruebas debe ser lo más parecido a lo que será el uso real de la aplicación, esto debido a que se trata de un software con muchos usuarios conectados a la vez y se desea conocer el real desempeño que tendrá.

**6.2.1.5. Banco de Prueba**

A continuación se presenta el cuadro de prueba que se realizó para validar los requerimientos documentados en el Capítulo 3, dicho cuadro se obtuvo del capítulo 7 de Metodología de Gestión de Requerimientos (Sites, 2010). En la Tabla 6.1 se muestran los requisitos del A al F y en la Tabla 6.2 los requisitos G al L.

Prueba de Requerimientos						
	RA	RB	RC	RD	RE	RF
El requisito se encuentra debidamente documentado						
El requisito no tiene errores de sintaxis y morfológicos						
El requisito cumple con las expectativas del cliente						
El requisito no tiene palabras ambiguas						
El requisito tiene documentado todo lo solicitado						
Se realizó el proceso de reunión y se tiene la documentación correcta						

Tabla 6.1: Prueba de requerimientos de la aplicación parte 1.

<b>Prueba de Requerimientos</b>						
	RG	RH	RI	RJ	RK	RL
El requisito se encuentra debidamente documentado						
El requisito no tiene errores de sintaxis y morfológicos						
El requisito cumple con las expectativas del cliente						
El requisito no tiene palabras ambiguas						
El requisito tiene documentado todo lo solicitado						
Se realizó el proceso de reunión y se tiene la documentación correcta						

Tabla 6.2: Prueba de requerimientos de la aplicación parte 2.

#### 6.2.1.6. Resultados de las Pruebas

A continuación se muestra en las Tablas 6.3 y 6.4 los resultados de las pruebas de requerimientos de los requisitos funcionales de la aplicación.

<b>Prueba de Requerimientos</b>						
	RA	RB	RC	RD	RE	RF
El requisito se encuentra debidamente documentado	C	C	C	C	C	C
El requisito no tiene errores de sintaxis y morfológicos	C	C	C	C	C	C
El requisito cumple con las expectativas del cliente	C	C	C	C	C	C
El requisito no tiene palabras ambiguas	C	C	C	C	C	C
El requisito tiene documentado todo lo solicitado	C	C	C	C	C	C
Se realizó el proceso de reunión y se tiene la documentación correcta	C	C	C	C	C	C

Tabla 6.3: Resultado de las prueba de requerimientos de la aplicación parte 1.

<b>Prueba de Requerimientos</b>						
	RG	RH	RI	RJ	RK	RL
El requisito se encuentra debidamente documentado	C	C	C	C	C	C
El requisito no tiene errores de sintaxis y morfológicos	C	C	C	C	C	C
El requisito cumple con las expectativas del cliente	C	C	C	C	C	C
El requisito no tiene palabras ambiguas	C	C	C	C	C	C
El requisito tiene documentado todo lo solicitado	C	C	C	C	C	C
Se realizó el proceso de reunión y se tiene la documentación correcta	C	C	C	C	C	C

Tabla 6.4: Resultado de las prueba de requerimientos de la aplicación parte 2.

En consecuencia a las pruebas realizadas para cada requerimiento funcional del sistema, se pudo comprobar el correcto funcionamiento de la aplicación móvil. No se encontraron diferencias entre los requerimientos y el software en funcionamiento, además de no percibir defectos en el sistema. Pro otro lado no se encontraron inconsistencias, incompletitudes ni ambigüedades.

## **6.2.2. Pruebas de Usabilidad**

### **6.2.2.1. Descripción**

Las pruebas de usabilidad nos permiten asegurar la calidad de la experiencia de usuario. Consiste en invitar público objetivo a probar el software y proporcionar comentarios, en este caso cuantitativos, y valiosos para mejorar la satisfacción de los usuarios de la aplicación.

### **6.2.2.2. Criterio para la evaluación.**

Los ítems de las distintas pruebas serán evaluadas del 1 al 5. Siendo 5 la evaluación más exitosa. Un ítem evaluado con menos de 3 se considera como no aprobado y, por tanto, se debe mejorar hasta que se vuelva a evaluar con éxito.

### **6.2.2.3. Condiciones de ejecución**

Para realizar esta evaluación, se consideró una muestra de 20 usuarios, considerando un promedio de las respuestas obtenidas expresadas en notas del 1 al 5, como ya se explicó en el punto anterior. En el caso de obtener como resultado un número decimal, se debe aproximar al más alto. Debido a que dependiendo del perfil que se adopte, la usabilidad es completamente diferente, entonces se realizará por separado para el modo profesor y para el modo alumno. Para esta prueba, 5 usuarios mantuvieron el perfil de profesor y 15 el de alumno.

### **6.2.2.4. Requerimientos de Ambiente de Pruebas**

El ambiente de pruebas se realizó por separado y bajo una conexión estable a una red WiFi. Todos los clientes utilizaron teléfonos smartphone, en los cuales el de menor versión fue Android 2.3 Gingerbread y la mayor fue Android 5.0 Lollipop.

### **6.2.2.5. Banco de Prueba**

Para la realización de las pruebas de usabilidad se consideró un test desarrollado por la Universidad Politécnica de Catalunya para la evaluación de las aplicaciones móviles orientadas a la educación (Gabaldón, 2004). En las Tablas 6.5 y 6.6 se muestra el cuestionario que se le realizó a los 20 usuarios que usaron la aplicación.

<b>Prueba de Usabilidad</b>					
	1	2	3	4	5
<b>1) Estructura de la aplicación</b>					
1.1) La organización de los elementos estructurales de la aplicación (barras de desplazamiento, zona de contenidos, botones, etc.) es buena.					
1.2) La cantidad de elementos estructurales es excesiva.					
1.3) La distribución de los elementos estructurales se mantiene constante a lo largo de la aplicación					
<b>2) Operación de la aplicación</b>					
2.1) El recorrido que se hace por el contenido de la aplicación es fácil.					
2.2) La relación mutua entre el usuario y la aplicación es buena.					
2.3) Las acciones que solicita la aplicación son fáciles de ejecutar.					
2.4) Se identifican fácilmente las figuras, las tablas, los hipertextos y el tipo de acción que se debe ejecutar.					
2.5) La velocidad del funcionamiento de la aplicación, considerando el tipo de tarea que se exige, es buena.					
2.6) Hay demasiados errores durante la ejecución de la aplicación.					
2.7) La ejecución de tareas (navegar en la aplicación, hacer click en botones, seleccionar opciones, etc.) sigue un estándar a lo largo de la aplicación.					
<b>3) Información al Usuario</b>					
3.1) Las dudas del usuario se resuelven fácilmente					
3.2) La aplicación mantiene al usuario informado sobre las tareas en ejecución.					
3.3) Los datos que busca el usuario son fáciles de encontrar.					

Tabla 6.5: Prueba de usabilidad de la aplicación parte 1.

<b>Prueba de Usabilidad</b>					
	1	2	3	4	5
<b>4) Apariencia</b>					
4.1) La presentación del contenido (tipo y tamaño de fuente, uso del color, dispositivo de los elementos según su significado) es buena.					
<b>5) Intuición</b>					
5.1) Los procedimientos de navegación por la aplicación o ejecución de tareas asignadas se aprenden de forma prácticamente inmediata.					
<b>6) Contenido</b>					
6.1) La distribución del contenido de la aplicación (imágenes, textos, etc.) es buena.					
6.2) La información que se presenta en la aplicación es demasiado extensa.					
6.3) No hay errores en la información que se presenta en la aplicación.					
6.4) La información que se presenta en la aplicación es fácil de entender y memorizar.					
<b>7) Experiencia del Usuario</b>					
7.1) Opinión general sobre la aplicación.					

Tabla 6.6: Prueba de usabilidad de la aplicación parte 2.

#### 6.2.2.6. Resultado de Pruebas

En las Tablas 6.7 y 6.8 se muestran los resultados obtenidos al realizar el cuestionario a los usuarios que probaron la aplicación en modo profesor.

<b>Prueba de Usabilidad</b>					
	1	2	3	4	5
<b>1) Estructura de la aplicación</b>					
1.1) La organización de los elementos estructurales de la aplicación (barras de desplazamiento, zona de contenidos, botones, etc.) es buena.				X	
1.2) La cantidad de elementos estructurales es excesiva.					X
1.3) La distribución de los elementos estructurales se mantiene constante a lo largo de la aplicación				X	
<b>2) Operación de la aplicación</b>					
2.1) El recorrido que se hace por el contenido de la aplicación es fácil.				X	
2.2) La relación mutua entre el usuario y la aplicación es buena.					X
2.3) Las acciones que solicita la aplicación son fáciles de ejecutar.					X
2.4) Se identifican fácilmente las figuras, las tablas, los hipertextos y el tipo de acción que se debe ejecutar.					X
2.5) La velocidad del funcionamiento de la aplicación, considerando el tipo de tarea que se exige, es buena.					
2.6) Hay demasiados errores durante la ejecución de la aplicación.					X
2.7) La ejecución de tareas (navegar en la aplicación, hacer click en botones, seleccionar opciones, etc.) sigue un estándar a lo largo de la aplicación.					X
<b>3) Información al Usuario</b>					
3.1) Las dudas del usuario se resuelven fácilmente-			X		
3.2) La aplicación mantiene al usuario informado sobre las tareas en ejecución.					X
3.3) Los datos que busca el usuario son fáciles de encontrar.					X

Tabla 6.7: Resultados con promedios prueba de usabilidad de la aplicación modo profesor parte 1.

<b>Prueba de Usabilidad</b>					
	1	2	3	4	5
<b>4) Apariencia</b>					
4.1) La presentación del contenido (tipo y tamaño de fuente, uso del color, dispositivo de los elementos según su significado) es buena.				X	
<b>5) Intuición</b>					
5.1) Los procedimientos de navegación por la aplicación o ejecución de tareas asignadas se aprenden de forma prácticamente inmediata.				X	
<b>6) Contenido</b>					
6.1) La distribución del contenido de la aplicación (imágenes, textos, etc.) es buena.					X
6.2) La información que se presenta en la aplicación es demasiado extensa.					X
6.3) No hay errores en la información que se presenta en la aplicación.					X
6.4) La información que se presenta en la aplicación es fácil de entender y memorizar.					X
<b>7) Experiencia del Usuario</b>					
7.1) Opinión general sobre la aplicación.				X	

Tabla 6.8: Resultado con promedios prueba de usabilidad de la aplicación modo profesor parte 2.

En las Tablas 6.9 y 6.10 se muestran los resultados obtenidos al realizar el cuestionario a los usuarios que usaron la aplicación en modo alumno.

<b>Prueba de Usabilidad</b>					
	1	2	3	4	5
<b>1) Estructura de la aplicación</b>					
1.1) La organización de los elementos estructurales de la aplicación (barras de desplazamiento, zona de contenidos, botones, etc.) es buena.					X
1.2) La cantidad de elementos estructurales es excesiva.					X
1.3) La distribución de los elementos estructurales se mantiene constante a lo largo de la aplicación			X		
<b>2) Operación de la aplicación</b>					
2.1) El recorrido que se hace por el contenido de la aplicación es fácil.			X		
2.2) La relación mutua entre el usuario y la aplicación es buena.				X	
2.3) Las acciones que solicita la aplicación son fáciles de ejecutar.					X
2.4) Se identifican fácilmente las figuras, las tablas, los hipertextos y el tipo de acción que se debe ejecutar.					X
2.5) La velocidad del funcionamiento de la aplicación, considerando el tipo de tarea que se exige, es buena.					X
2.6) Hay demasiados errores durante la ejecución de la aplicación.					X
2.7) La ejecución de tareas (navegar en la aplicación, hacer click en botones, seleccionar opciones, etc.) sigue un estándar a lo largo de la aplicación.					X
<b>3) Información al Usuario</b>					
3.1) Las dudas del usuario se resuelven fácilmente.			X		
3.2) La aplicación mantiene al usuario informado sobre las tareas en ejecución.					X
3.3) Los datos que busca el usuario son fáciles de encontrar.			X		

Tabla 6.9: Resultado con promedios prueba de usabilidad de la aplicación modo alumno parte 1.

<b>Prueba de Usabilidad</b>					
	1	2	3	4	5
<b>4) Apariencia</b>					
4.1) La presentación del contenido (tipo y tamaño de fuente, uso del color, dispositivo de los elementos según su significado) es buena.				X	
<b>5) Intuición</b>					
5.1) Los procedimientos de navegación por la aplicación o ejecución de tareas asignadas se aprenden de forma prácticamente inmediata.			X		
<b>6) Contenido</b>					
6.1) La distribución del contenido de la aplicación (imágenes, textos, etc.) es buena.					X
6.2) La información que se presenta en la aplicación es demasiado extensa.					X
6.3) No hay errores en la información que se presenta en la aplicación.					X
6.4) La información que se presenta en la aplicación es fácil de entender y memorizar.					X
<b>7) Experiencia del Usuario</b>					
7.1) Opinión general sobre la aplicación.				X	

Tabla 6.10: Resultado con promedios prueba de usabilidad de la aplicación modo alumno parte 2.

En base a las pruebas de usabilidad realizadas para el modo profesor y el modo alumno se logró asegurar la calidad de la experiencia de usuario, dado que ningún ítem fue evaluado con nota inferior a tres. Los comentarios realizados por los usuarios invitados a probar la aplicación fueron positivos, se mostraron contentos y asombrados por la simplicidad de uso que tenía el sistema. Las recomendaciones de los usuarios fueron específicamente el cambiar algunos objetos de posición, para que se logren apreciar de mejor forma.

### **6.2.3. Test de Saturación con 20 Dispositivos**

#### **6.2.3.1. Descripción**

El objetivo del test de saturación es poner a prueba la robustez y la confiabilidad del software sometiéndolo a condiciones de uso extremas. En éste se incluyen una cantidad importante de envíos de información para intentar colapsar al sistema y así ver de que forma reacciona.

#### **6.2.3.2. Criterio para la evaluación.**

Para que el test sea exitoso todos los dispositivos que se conectarán al servidor deben ser aceptados optimamente, sin que la conexión de uno o más clientes se vea afectada o falle.

#### **6.2.3.3. Condiciones de ejecución**

El test se realizará a partir de 20 dispositivos con sistema operativo Android, conectados a una misma red WiFi y en el cual 19 de éstos harán la función de alumnos y sólo uno de profesor.

#### **6.2.3.4. Requerimientos de Ambiente de Pruebas**

El ambiente de pruebas es bajo una conexión estable de internet, todos los clientes utilizaron telefonos smartphone en los cuales el de menor versión fue Android 2.3 Gingerbread y la mayor fue Android 5.0 Lollipop, tal como en los requerimientos de ambiente de pruebas de usabilidad.

#### **6.2.3.5. Resultado de las pruebas**

Dadas todas las condiciones mencionadas, el resultado del test de saturación fue exitoso, ya que al conectarse 19 alumnos como clientes a un profesor como servidor, no ocurrió ninguna anomalía en la ejecución del programa y éste funcionó eficientemente de acuerdo a lo requerido por los diferentes procesos que se debían ejecutar en ciertos momentos, incluyendo el envío de información mutua.

# Capítulo 7

## Conclusiones

La tecnología, las ciencias informática y la educación son ámbitos en constante desarrollo y a través de los cuales se busca lograr una cierta armonía, que permita la inclusión de interdependencia de ambas disciplinas. Todo, con una finalidad que se traduce en el punto central de esta investigación; que una esté al servicio de la otra, y que en definitiva, esto reporte un beneficio a la comunidad.

En este orden de ideas, y como una manifestación del principio recién expuesto es que se escogió el tema a investigar. La implementación de una teclera virtual sobre una plataforma Android para la UCSC es un proyecto que busca incorporar una forma alternativa de aprendizaje en los alumnos utilizando un sistema cómodo, económico, práctico, amigable y altamente eficiente, como ha quedado demostrado.

Se expondrán a continuación, sinteticamente las conclusiones que arrojó esta investigación.

1. En en cuanto al análisis de las funcionalidades del sistema es posible consignar, en primer término que a partir de la investigación, observación de otras aplicaciones similares y entrevistas a docentes que las tecleras físicas son un sistema realmente factible como método de enseñanza en el aula, gracias a el aporte de la tecnología en la sala de clases, el aumento del interes de los alumnos y la mejora en el rendimiento en los casos señalados en la investigación.

Luego, en lo relativo a la selección de herramientas para desarrollar la aplicación, de las que existen a disposición se eligió Android Studio sobre Eclipse principalmente porque es una herramienta proporcionada por Google, actualizaciones constantes del sistema que permiten optimizaciones a nivel de procesos de com-

pilación y creación de ejecutables, además es más amigable al usuario dado que es intuitivo y fácil de aprender a usar. A su vez se descarto Eclipse básicamente, porque necesita un plugging adicional para poder usarlo y esto se traduce en un mayor tiempo de instalación lo que, en definitiva hace mas engorroso el proceso. En consecuencia, Eclipse es un software de uso general para la programación de muchos lenguajes y Andoid Studio se especifica en el desarrollo de aplicaciones Android.

A continuación, y siguiendo dentro de este mismo lineamiento, cabe destacar que para documentar las funcionalidades del sistema se siguió el estándar de la IEEE 830 para asegurar la calidad del contenido expuesto. Finalmente el hecho de haber documentado los requerimientos otorga dos claves esenciales dentro de esta investigación; primero sirve de guía de procedimiento a realizar en la construcción del software y, segundo ayuda a exponer con claridad los pasos a seguir en la contrucción de la aplicación.

2. En lo relativo al diseño de la aplicación, se logró mostrar gráficamente el funcionamiento de la misma a través de distintos diagramas con diferentes interpretaciones, empezando por un diagrama de actividad que sirve para entender de forma simplificada lo que ocurre durante los diferentes procesos que se ejecutan en el sistema, siguiendo de casos de uso para cada actividad que realizan por separado cada actor de la aplicación, ya sea este profesor o alumno, además es usado para demostrar como se quiere que interactue cada actor con la aplicación y describir como se usa el sistema, hasta llegar a un diagrama de componentes para entender como se relacionan los componentes físicos del sistema, en este diagrama se muestran todos los componentes utilizados los cuales son simples paquetes representativos de los tipos de elementos utilizados en el software, para terminar se muestran los wireframe del diseño de la aplicación con sus respectivas interfaces de usuario terminadas.
3. En el ámbito de la construcción de la aplicación y, luego de haber obtenido los conocimientos necesarios para la creación de una aplicación en Android, como enlazar los archivos .java con los .xml, se realizó la aplicación a partir de la base de comunicación cliente-servidor, relacionando aspectos básicos de la gráfica, como incorporar los *layouts* y los botones adecuados para que los actores

de la aplicación se conecten de forma exitosa, y de esta forma vincularlo con el código de la arquitectura. En consecuencia, se logro la aplicación de la manera adecuada transformando lo ya conseguido en lo que se quería implementar. Se creó la base de datos para almacenar las preguntas a enviar por el profesor, se estableció el gráfico con las respuestas de los alumnos, se implementó el envío de los datos por mail y por último, el texto con la pregunta y los botones con las alternativas que recibe el alumnos al momento de que el profesor envía su pregunta. Acto seguido, se perfecciono la parte estructural principal mencionada agregando mensajes de información al usuario. En conclusión agrupanto los aspectos mencionados se obtuvo una aplicación estable en conexión, fluida y amigable para el usuario.

4. El testeó de la aplicación implicó el analisis y desarrollo de tres ítems que se mencionan a continuación:

**Test de Requerimientos:** Se consiguió un test exitoso en cuanto a los requerimientos funcionales de la aplicación, dado que no se encontraron ambigüedades, inconsistencias ni incompletitud, esto se demostró con el respectivo test mostrado en la Sección 6.2.1.6.

**Test de Usabilidad:** Este ítem se refiere a como se maneja el usuario con la aplicación. En este caso se testeó la usabilidad con un grupo de 20 personas separadamente, los cuales evaluaron los diversos puntos que se incluyeron en el cuestionario de la Sección 6.2.2.6. Como resultado este arrojó que cada punto superó el promedio de 3 puntos establecido base para considerarlo exitoso, por lo tanto la aplicación realizada cumple con los requisitos básicos de usabilidad necesarios.

**Test de Stress:** Este test fue aplicado a 20 personas que se encontraban en un mismo recinto, conectados a una misma red WiFi, con teléfonos Android y utilizando la aplicación a testear. En este caso la aplicación no sufrió desconexiones, ni fallos de ninguna índole, permitiendo que 19 usuarios conectados como alumnos a un profesor utilizaron exitosamente la aplicación, lo cual significa que se logra una aplicación eficiente.

Para finalizar resta establecer que el proyecto de investigación constituye realmente un aporte al ámbito de ingeniería informática, pues además de obtener una aplicación

bien lograda en todos sus aspectos técnicos contribuye a mejorar el rendimiento de los alumnos en los cuales se pretende implementar.

# Bibliografía

- Alegsa, L. (2010). Definición de cliente/servidor (computación). Accedido 08-01-2015 a <http://www.alegsa.com.ar/Dic/cliente>
- apk4ios (2013). Capturas de pantalla de la aplicación resp. Accedido 30-07-2014 a [http://apk4ios.com/APK\\_Resp\\_iOS.html](http://apk4ios.com/APK_Resp_iOS.html).
- Brothersoft (2012). Capturas de pantalla de la aplicación ars intranet. Accedido 13-04-2014 a [http://android.brothersoft.com/ars\\_\\_\\_intranet-376624.html](http://android.brothersoft.com/ars___intranet-376624.html).
- Calleja, J. (2013). Ventajas de android para el desarrollo de aplicaciones móviles. Accedido 15-10-2014 a <http://esdroids.com/ventajas-de-android-para-el-desarrollo-de-aplicaciones-moviles/>.
- Cilleros, F. (2014). Tecleras digitales (clickers). Accedido 03-11-2014 a <http://es.slideshare.net/FranciscoCillerosJar/tecleras-digitales-clickers>.
- ColegioMayor (2014). Tecleras digitales de evaluación. Accedido 05-12-2014 a <http://www.colegiomayor.cl/tobalaba/web/index.php/quienes-somos/informatica-educativa/tecleras-de-evaluacion>.
- Commons, C. (2014). Entorno cliente/servidor. Accedido 08-01-2015 a <http://es.ccm.net/contents/148-entorno-cliente-servidor>.
- de Telecomunicaciones, S. (2013). Informe sectorial:telecomunicaciones en chilel. Accedido 13-10-2015 a [http://www.subtel.gob.cl/images/stories/apoyo\\_articulos/-notas\\_prensa/analisis\\_sectorial\\_dic2012\\_20130315](http://www.subtel.gob.cl/images/stories/apoyo_articulos/-notas_prensa/analisis_sectorial_dic2012_20130315).
- Developers, G. (2015). Visualization: Pie chart. Accedido 28-03-2015 a <https://google-developers.appspot.com/chart/interactive/docs/gallery/piechart>.

- Dávalos, A. G. (2012). Introducción a la computación móvil. Accedido 05-10-2014 a [http://augusta.uao.edu.co/moodle/file.php/2896/Introduccion\\_Computacion\\_Movil\\_2012.pdf](http://augusta.uao.edu.co/moodle/file.php/2896/Introduccion_Computacion_Movil_2012.pdf).
- Factory, G. (2011). Receptor de señal. Accedido 01-04-2014 a [http://www.rmm.cl/usuarios/mruiz1/doc/Manual %20Botoneras %20Enjoy](http://www.rmm.cl/usuarios/mruiz1/doc/Manual%20Botoneras%20Enjoy)
- Factory, G. (2012). Control remoto alumno para tecleras físicas. Accedido 01-04-2014 a <http://educinf.cl/project/tecleras-digitales-rf/>.
- Fernández, R. B. (2012). Implementación de novedosa metodología aumenta la participación de estudiantes en el aula. Accedido 09-11-2014 a [http://medicina.med.uchile.cl/noticias /127-implementacion-de-novedosa-metodologia-aumenta-participacion-de-estudiantes-en-el-aula.html](http://medicina.med.uchile.cl/noticias/127-implementacion-de-novedosa-metodologia-aumenta-participacion-de-estudiantes-en-el-aula.html).
- Foretech, M. (2013). Tecleras digitales rf. Accedido 11-05-2014 a [http://www.maleducativo.cl/index.php?route=product/productproduct\\_id=85](http://www.maleducativo.cl/index.php?route=product/productproduct_id=85).
- Gabaldón, D. R. (2004). Cuestionario sobre la evaluación de la usabilidad de aplicaciones usadas en educación. Accedido 20-04-2015 a [http://www.tesisenred.net/bitstream/handle/10803/6542/12ApendiceA.pdf? sequence=11](http://www.tesisenred.net/bitstream/handle/10803/6542/12ApendiceA.pdf?sequence=11).
- ISO (2015). La familia de normas iso/iec 25000. Accedido 10-10-2015 a <http://iso25000.com/index.php/normas-iso-25000>.
- Maleducativo (2011). Control remoto profesor para tecleras físicas. Accedido 01-04-2014 a [http://www.maleducativo.cl/index.php? route=product/productproduct\\_id=85](http://www.maleducativo.cl/index.php?route=product/productproduct_id=85).
- Mendez, G. (2008). Especificación de requisitos según el estándar de ieee 830. Accedido 11-10-2015 a <https://www.fdi.ucm.es/profesor/gmendez/docs/is0809/iee830.pdf>.
- Page, L. (2014). Apps nuevas más y las mejores actualizaciones. Accedido 15-07-2014 a <https://play.google.com/store/apps?hl=es>.
- Papiro (2013). Convertidor usb para tecleras físicas. Accedido 01-04-2014 a [http://www.papiro.cl/Manual %20Tecleras.pdf](http://www.papiro.cl/Manual%20Tecleras.pdf).

- PlayStore (2014a). Capturas de pantalla de la aplicación clickerschool clicker virtual. Accedido 10-04-2015 a <https://play.google.com/store/apps/details?id=com.eduware.ClickerSchoolRemote>.
- PlayStore (2014b). Capturas de pantalla de la aplicación openars. Accedido 07-04-2015 a <https://play.google.com/store/apps/details?id=com.eduware.ClickerSchoolRemote>.
- Sites, G. (2010). Metodología gestión de requerimientos. Accedido 11-10-2015 a <https://sites.google.com/site/metodologiareq/capitulo-vii>.
- Tomás., J. (2014). *El gran libro de Android avanzado*. alfaomega. páginas 366–389.
- Ulloa, C. (2013). Nuevo sistema de enseñanza interactiva “tecleras digitales”. Accedido 01-04-2015 a <http://crea.ulagos.cl/el-centro-de-recursos-tiene-a-disposicion-de-los-docentes-nuevo-sistema-de-ensenanza-interactiva-tecleras-digitales/>.
- Vega, A. A. S. (2013). Mejora del proceso de enseñanza y aprendizaje en la carrera de ingeniería de ejecución mecánica. diseño micro-curricular basado en resultados de aprendizaje. Accedido 15-05-2014 a [http://www.scielo.cl/scielo.php?pid=S0718-50062013000400002script=sci\\_arttext](http://www.scielo.cl/scielo.php?pid=S0718-50062013000400002script=sci_arttext).

## Apéndice A: Glosario

<b>Término</b>	<b>Definición</b>
Google	empresa especializada en productos y servicios relacionados con Internet, software, dispositivos electrónicos y otras tecnologías.
Software	Término genérico que se aplica a los componentes no físicos de un sistema informático (RAE).
Sistema Operativo	Conjunto de órdenes y programas que controlan los procesos básicos de una computadora y permiten el funcionamiento de otros programas (RAE).
Linux	Sistema Operativo de libre distribución creado por UNIX para computadores de escritorio.
Java	Lenguaje de programación basado en objetos.
SQL	Lenguaje declarativo de acceso a bases de datos.
Smartphone	Teléfono celular con pantalla táctil, que permite al usuario conectarse a internet, gestionar cuentas de correo electrónico e instalar otras aplicaciones y recursos a modo de pequeño computador.
Servidor	Ordenador o máquina informática que está al “servicio” de otras máquinas, ordenadores o personas llamadas clientes y que le suministran a estos, todo tipo de información.
String	En informática es traducido como cadena de caracteres, o array de caracteres. Es un método sencillo de guardar en memoria un conjunto de caracteres.
Robustez de Software	Sistema informático que puede ejecutar diversos procesos de manera simultánea sin generar fallos o bloquearse
Wireframe	Es un esquema de página o plano de pantalla, es una guía visual que representa el esqueleto o estructura visual de un Software.
Plugin	Es aquella aplicación que, en un programa informático, añade una funcionalidad adicional o una nueva característica al software.

## Apéndice B: Guía de Usuario

El presente apéndice muestra un manual de usuario de la aplicación implementada, a fin de entender su funcionamiento e informar al usuario sobre el manejo del software.

Se presenta en la Figura 1 la pantalla de selección de modo, en ésta se logra apreciar dos imagenes, al presionar en ellas se direccionan al perfil señalado.



Figura 1: Selección de Perfil.

Se puede observar un tutorial similar a esta guía para el usuario dentro de aplicación, tanto en el perfil profesor como en el perfil alumno, presionando en la barra de menú y luego seleccionando tutorial, tal y como se observa en la Figura 2.

Inmediatamente después de la selección de perfil, se debe establecer la conexión entre profesor y alumno, el profesor comparte su dirección IP al alumno para que se conecte

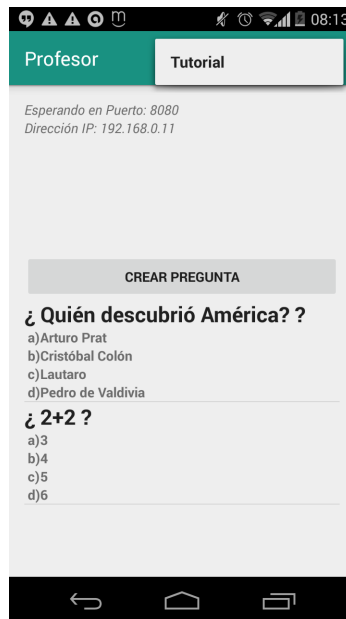


Figura 2: Ver tutorial.

con él y éste se asiga un nombre de usuario. Al lado izquierdo de la Figura 3 se aprecia la pantalla del alumno, el lugar donde debe indicar su nombre y donde debe ir la dirección IP. Y a la derecha, la pantalla del profesor con la información de conexión desplegada.

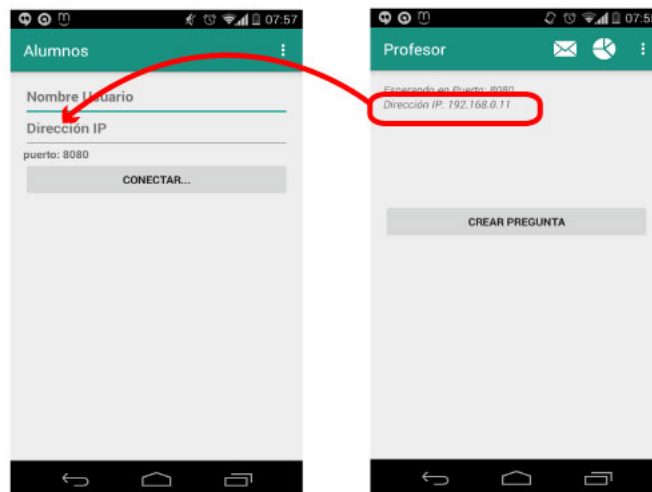


Figura 3: Establecer Conexión.

Una vez establecida la conexión en el modo profesor se puede observar el botón crear pregunta, el cual sirve para guardar las preguntas creadas en la base de datos de manera tal que en el futuro puedan ser enviadas a los alumnos. Para estos efectos, es necesario ingresar la pregunta y a lo menos dos alternativas. A la izquierda de la Figura 4 se observa el mensaje de error al ingresar sólo una alternativa y a la derecha se aprecia una pregunta bien ingresada con sus cuatro alternativas. Siguiendo con el modo

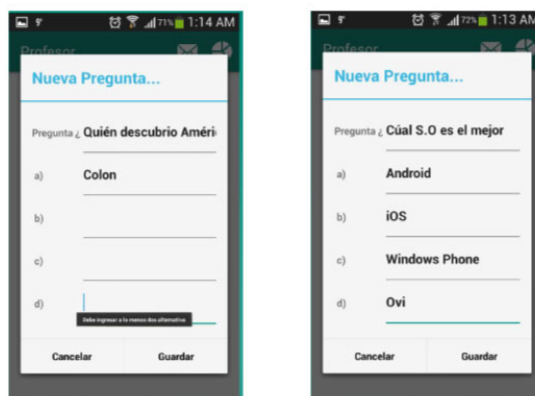


Figura 4: Crear Pregunta.

profesor, en la Figura 5 al costado derecho, es posible apreciar la forma de eliminar una pregunta, que se logra con un click prolongado sobre la misma, y a la izquierda, se puede observar la confirmación del envío de una pregunta a los alumnos, que se consigue al realizar un click sobre una pregunta.

Luego de enviar una pregunta, el profesor espera las respuestas del alumno. Inmediatamente después de presionar el botón enviar, aparece un botón de finalizar la pregunta, mientras no se presione éste, se mantienen las estadísticas de las respuestas enviadas por los alumnos. Al presionar el botón finalizar pregunta, las estadísticas vuelven a cero y se puede realizar el envío de una nueva pregunta. En la Figura 6 se muestra la pantalla con el botón para realizar la finalización de una pregunta.

Por el lado de el modo alumno, al momento de establecer la conexión, éste queda en modo de espera hasta que el profesor envíe su pregunta, tal como se aprecia a la izquierda de la Figura 7. Mientras que a la derecha se muestra la pregunta recibida por el alumno. Las alternativas son botones, los alumnos deben presionar el botón de la alternativa que crean sea la correcta, y así enviar la respuesta al profesor. Al contestar

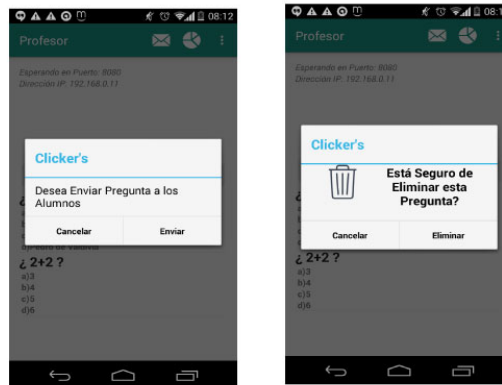


Figura 5: Enviar y Eliminar pregunta

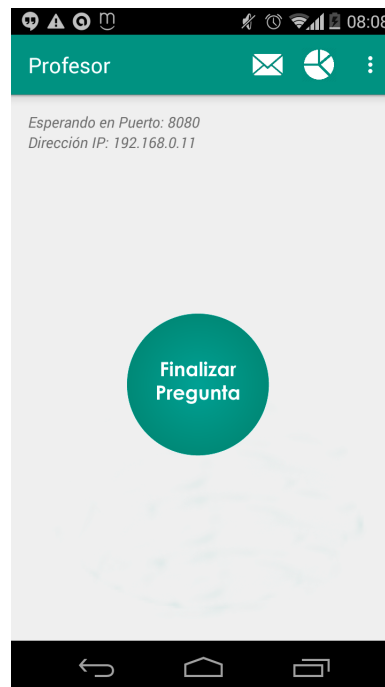


Figura 6: Finalizar Pregunta.

desaparece inmediatamente la pregunta, evitando que un alumno conteste dos veces a la misma pregunta y vuelve al modo de esperar pregunta.

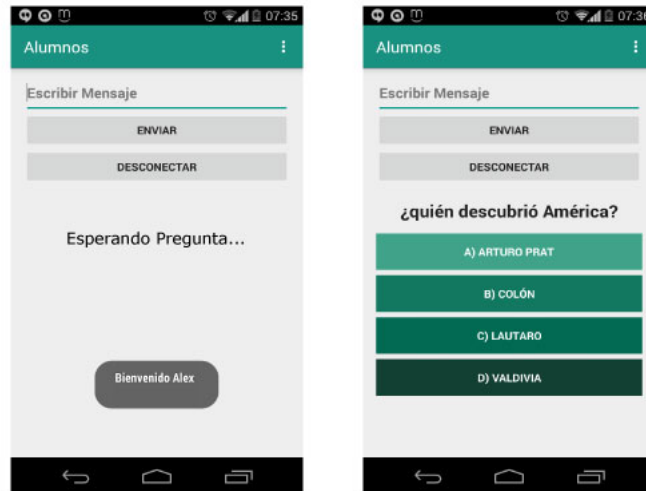


Figura 7: Esperar y Recibir Pregunta.

Sin haber terminado la pregunta, es decir, siguiendo en el modo profesor y luego de haber recibido las respuestas de los alumnos, se observa en la barra superior de menu dos íconos, al presionar el ícono con forma de buzón se envía la información obtenida con la cantidad total de alumnos que enviaron sus respuestas y la cantidad particular de alumnos que respondieron a una alternativa en específico, tal como se observa en la Figura 8. Y si se elige presionar el ícono con forma de gráfico, se puede obtener un gráfico de torta con los porcentajes de respuestas a una alternativa, como se ve en la Figura 9.

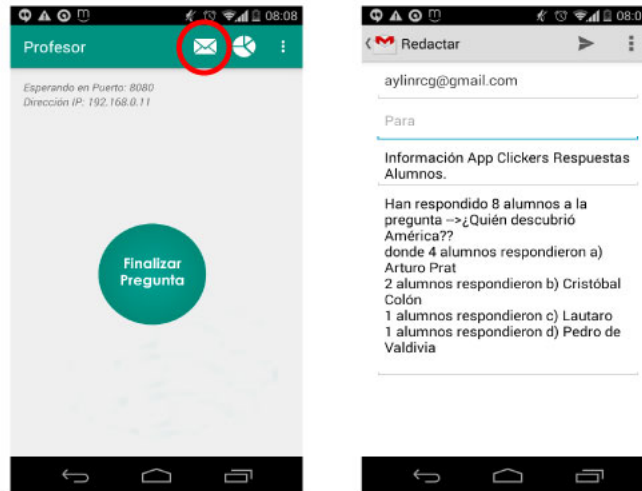


Figura 8: Enviar Estadísticas por Email.

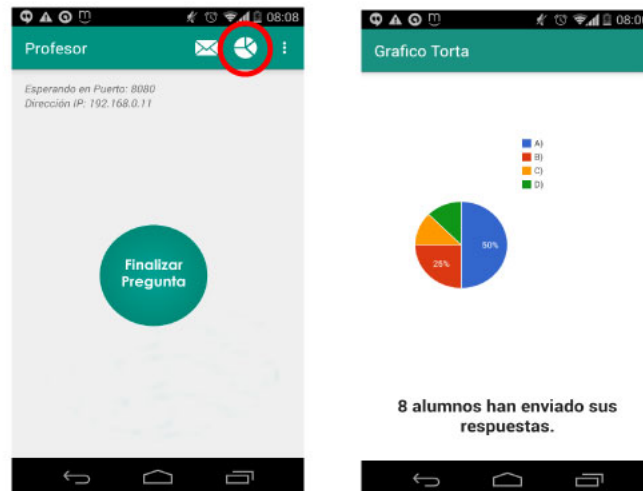


Figura 9: Ver Gráfico con Estadísticas.