

CLASSAPP: UN PROTOTIPO DE APLICACIÓN MÓVIL PARA LA GESTIÓN DE CLASES

GUSTAVO ADOLFO GARZÓN VILLAMIZAR

UNIVERSIDAD INDUSTRIAL DE SANTANDER  
FACULTAD DE INGENIERÍAS FÍSICO-MECÁNICAS  
ESCUELA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA  
BUCARAMANGA

2016

CLASSAPP: UN PROTOTIPO DE APLICACIÓN MÓVIL PARA LA GESTIÓN DE CLASES

GUSTAVO ADOLFO GARZÓN VILLAMIZAR

Trabajo de Grado para optar al título de  
Ingeniero de Sistemas

Director

SERGIO FERNANDO CASTILLO CASTELBLANCO

Ingeniero de Sistemas, Ph.D

UNIVERSIDAD INDUSTRIAL DE SANTANDER  
FACULTAD DE INGENIERÍAS FÍSICO-MECÁNICAS  
ESCUELA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA  
BUCARAMANGA

2016

## DEDICATORIA

A mis padres y hermanos, por estar a mi lado en los momentos más críticos, por ser mi fuerza cuando el cuerpo me ha fallado y por tolerar todos mis defectos.

A Daniel Fernando Santos Diaz, por enseñarme cuando debo esforzarme, cuando debo brindar por un motivo y cuando debo dejar de aferrarme al pasado. Y por todos los buenos consejos en todos estos años.

A Mario Andrés Ordóñez Serrano, por “el mundo de las ideas”, por mostrarme que la realidad nunca es absoluta y por diversificar mi criterio político.

A los doctores Mario Bueno Duran, Jesús Insuasty Enriquez y Oscar Abuchaibe Campos, por salvar mi vida, por todo su apoyo y por regalarme unos años más de vida sin secuelas ni contratiempos. A los tecnólogos de radioterapia del Centro de Cáncer y Enfermedades Hematológicas y al personal clínico de FOSUNAB, por su amabilidad, su paciencia y por llevar a cabo mis tratamientos de manera exitosa.

A los compañeros de carrera que me mostraron su lado más solidario durante mi enfermedad, por sacar tiempo de sus compromisos personales para darme ánimos justo cuando más lo necesitaba.

Finalmente, dedico este trabajo de grado al doctor Alberto Montoya Puyana y a los directivos de la Universidad Autónoma de Bucaramanga. Su oportuna gestión salvó mi vida y le devolvió la tranquilidad a mi familia.

## **AGRADECIMIENTOS**

Al profesor Sergio Fernando Castillo Castelblanco, por darme la oportunidad de desarrollar este proyecto bajo su tutoría, por su inspiración a la hora de explorar el desarrollo de aplicaciones móviles, por su inmensa paciencia y por su comprensión ante las dificultades de mi enfermedad.

Al profesor Manuel Guillermo Flórez Becerra, por sus aportes clave para la usabilidad y rapidez de la aplicación.

Al profesor Raúl Ramos Pollán, por iniciarme en el mundo del Machine Learning y la inspiración para retomar esa línea académica en el futuro.

A María Cecilia Flórez Bautista, por su valioso apoyo en los procesos administrativos durante mi estadía en la Universidad Industrial de Santander, y por su paciencia con mis problemas de salud.

A Martha Isabel Perez Estupiñan, por sus aportes y su asesoría técnica, los cuales agilizaron la redacción de este trabajo de grado.

## CONTENIDO

	pág.
INTRODUCCIÓN.....	15
1. PLANTEAMIENTO Y JUSTIFICACIÓN DEL PROBLEMA .....	16
2. OBJETIVOS.....	17
2.1. OBJETIVO GENERAL.....	17
2.1.1. OBJETIVOS ESPECÍFICOS .....	17
3. MARCO DE REFERENCIA.....	18
3.1. MARCO TEÓRICO .....	18
3.1.1. Aplicaciones móviles (APPS). .....	18
3.1.2. Gestión de cursos. ....	19
3.1.3. Aplicaciones móviles para la gestión de cursos. ....	19
3.2. ESTADO DEL ARTE.....	19
4. HERRAMIENTAS .....	28
4.1. ANDROID .....	28

4.2. ANDROID STUDIO (IDE).....	29
4.3. ANDROID SOFTWARE DEVELOPMENT KIT (SDK).....	30
4.4. SQLITE.....	31
4.5. JAVA EXCEL API (JXL).....	32
4.6. MPANDROIDCHART LIBRARY.....	33
4.7. CLASE SIMPLEFILEDIALOG .....	34
5. ESPECIFICACIÓN DE REQUISITOS .....	35
5.1. APROXIMACIÓN PRELIMINAR (HEURÍSTICOS DE BRUEGGE Y DUTOIT) .....	35
5.2. IDENTIFICACIÓN DE CASOS DE USO .....	35
5.3. DOCUMENTO DE ESPECIFICACIÓN DE REQUISITOS SIMPLIFICADO .....	57
5.4. PROTOTIPOS DE INTERFAZ DE USUARIO .....	62
6. DISEÑO DE LA APLICACIÓN .....	66
6.1. PATRÓN DE DISEÑO: MODELO-VISTA-CONTROLADOR (MVC).....	66
6.2. MODELO ENTIDAD-RELACIÓN .....	67
6.3. DIAGRAMA DE CLASES.....	68
6.4. ESTRUCTURA DE PAQUETES (ANDROID STUDIO) .....	70

6.5. DIAGRAMA DE NAVEGACIÓN .....	73
7. METODOLOGÍA .....	74
7.1. PROTOTIPADO EVOLUTIVO (MODELO DE PROTOTIPOS) .....	74
7.2. FASES DE PROTOTIPADO .....	75
8. DESCRIPCIÓN DEL SISTEMA.....	80
8.1. REQUERIMIENTOS DE HARDWARE .....	80
8.2. DEPENDENCIAS.....	80
8.3. DESCRIPCIÓN DEL SOFTWARE .....	81
8.3.1. Roles y usuarios. . .....	81
8.3.2. Funcionalidades. . .....	82
9. MANIFIESTO DE PERMISOS DEL SISTEMA OPERATIVO ANDROID .....	91
10. PRUEBAS.....	101
10.1. PRUEBAS DE DESEMPEÑO GENERAL .....	101
10.1.1. Procesamiento. . .....	101
10.1.2. Cierres inesperados (Crashes).....	101
10.1.3. Versiones de Android. . .....	102

10.1.4. Hardware (Cámara y acelerómetro).....	102
10.2. PRUEBAS DE DESEMPEÑO EN SQLITE.....	102
10.3. PRUEBAS DE USABILIDAD (INTERFAZ GRÁFICA).....	104
10.4. PRUEBAS DE FUNCIONALIDAD.....	106
10.5. EVALUACIÓN DEL CUMPLIMIENTO DE OBJETIVOS.....	106
11. CONCLUSIONES.....	108
12. RECOMENDACIONES.....	110
CITAS.....	113
BIBLIOGRAFIA.....	120
ANEXOS.....	127

## LISTA DE FIGURAS

pág.

<b>Figura 1. Capas en la arquitectura de Android. ....</b>	<b>28</b>
<b>Figura 2. Patrón MVC en Android Studio. ....</b>	<b>66</b>
<b>Figura 3. Modelo Entidad-Relación, ClassApp. ....</b>	<b>68</b>
<b>Figura 4. Diagrama de Clases de ClassApp. ....</b>	<b>69</b>
<b>Figura 5. Estructura de paquetes en Android Studio. ....</b>	<b>70</b>
<b>Figura 6. Diagrama de navegación, ClassApp. ....</b>	<b>73</b>
<b>Figura 7. Resultados iniciales de los métodos de ejecución en SQLite .....</b>	<b>103</b>
<b>Figura 8. Eliminación de los picos de procesamiento .....</b>	<b>103</b>

## LISTA DE ANEXOS

	pág.
<b>ANEXO A. HEURÍSTICOS DE BRUEGGE Y DUTOIT .....</b>	<b>127</b>
<b>ANEXO B. FORMATO ESPECIAL DE GRUPOS.....</b>	<b>136</b>

## RESUMEN

**TÍTULO:** CLASSAPP: UN PROTOTIPO DE APLICACIÓN MÓVIL PARA LA GESTIÓN DE CLASES\*

**AUTOR:**

GARZÓN VILLAMIZAR, GUSTAVO ADOLFO\*\*

**PALABRAS CLAVE:**

Llamado a lista, Asistencia, Gestión de clases, Gestión en el aula, Calificar estudiantes, Puntaje, Gestión escolar, Aplicaciones para el docente, Android, Java, Prototipado evolutivo, Desarrollo de aplicaciones móviles.

**DESCRIPCIÓN:**

Los paradigmas informáticos han venido cambiando desde mediados de los años 80. Los computadores de escritorio perdieron terreno, dando paso a todo tipo de dispositivos de uso personal. Esto ocurrió gracias a la miniaturización de los componentes electrónicos y a la creciente necesidad de portabilidad en software y documentos. El paradigma más reciente es el de los dispositivos móviles o smartphones; estos dispositivos cuentan con un poder de procesamiento suficiente como para tener su propio sistema operativo, llevar a cabo operaciones complejas y ejecutar módulos de software.

Esto favoreció el crecimiento de la industria de las apps (aplicaciones móviles) la cual es dominada (alrededor del mundo) por los dispositivos que ejecutan el sistema operativo Android. Adicionalmente, se ha observado una migración de profesionales en los roles de desarrollo y diseño tradicionales y una enorme adopción de esta nueva tendencia. ClassApp es una de estas aplicaciones, y pretende, en esencia, facilitar las labores diarias de un profesor en sus sesiones docentes. ClassApp agiliza los procesos académicos brindando una interfaz amable y provee un almacenamiento seguro de la información dentro del dispositivo.

La aplicación es gratuita y se encuentra alojada en Google Play. Su código fuente, manual de usuario y términos de licencia, se encuentran alojados en un sitio web y se espera que la aplicación evolucione con la ayuda de futuros estudiantes, docentes y demás entidades interesadas en estos procesos académicos del día a día.

---

\* Trabajo de grado

\*\* Facultad de ingenierías Físico-Mecánicas. Escuela de Ingeniería de Sistemas. Director: Sergio Fernando Castillo Castelblanco.

## SUMMARY

**TITLE:** CLASSAPP: A MOBILE APPLICATION PROTOTYPE FOR CLASS MANAGEMENT\*

**AUTHORS:**

GARZÓN VILLAMIZAR, GUSTAVO ADOLFO\*\*

**KEYWORDS:**

Roll call, Attendance, Class management, Classroom management, Student grading, Score, School control, Teacher apps, Android, Java, Evolutionary Prototyping, Mobile application development.

**DESCRIPTION:**

Software paradigms have been changing since mid-80s. Desktops lost ground, making way to all sorts of personal use devices. This happened thanks to electronic components miniaturization and the growing needs for software and documents portability. The latest paradigm is related to mobile devices and smartphones. Those devices have enough processing power to carry out complex operations, have an operating system of their own and execute software modules.

This processing power allowed the mobile apps industry growth, currently worldwide dominated by devices Android O.S devices. Additionally, there has been a developer's migration from the traditional development and design roles, as well a huge adoption of this mobile application development trend. ClassApp is one of those applications, which essentially seeks to ease the everyday work of teachers in their teaching sessions. ClassApp speeds up academic processes by providing a friendly interface and secure information storage into the device.

ClassApp is free and hosted on Google Play. Its source code, user manual and licensing terms are hosted on a website; It is expected that the application evolves thanks to future students and teachers contribution and any other stakeholders involved in everyday academic processes.

---

\* Degree Work

\*\* School of Physics and Mechanical Engineering. School of Systems Engineering. Director: Sergio Fernando Castillo Castelblanco.

## INTRODUCCIÓN

La sociedad actual ha venido cambiando de paradigmas informáticos desde mediados de los años 80. Se pasó del computador de escritorio a todo tipo de dispositivos personales gracias a la miniaturización de las piezas electrónicas y a la necesidad -cada vez más creciente- de portabilidad en documentos, software y actividades laborales. El paradigma más reciente es el de los dispositivos móviles o smartphones; estos dispositivos poseen suficiente poder de procesamiento como para tener su propio sistema operativo, llevar a cabo operaciones complejas y ejecutar módulos de software. Esto favoreció el crecimiento de la industria de las apps -la cual se encuentra considerablemente dominada por los dispositivos Android en Colombia [1] (entre un 70% y 90% en diferentes categorías de contenido)- y la correspondiente migración de profesionales en los roles de desarrollo y diseño que se manejaban tradicionalmente.

ClassApp es una de estas aplicaciones, y pretende, en esencia, facilitar las labores diarias de un profesor en sus sesiones docentes. Es frecuente que se tome asistencia en un salón de clase con una lista impresa o en un documento digital en Microsoft Excel. La calificación de actividades también está sujeta a esta filosofía, y en ocasiones pueden presentarse errores durante estos dos procesos académicos. ClassApp agiliza los procesos brindando una interfaz amable y menos propensa a errores, combinando estas cualidades con los beneficios de cualquier aplicación móvil y el almacenamiento seguro de la información dentro del dispositivo.

La aplicación, además, es gratuita y se encuentra alojada en Google Play (la tienda de aplicaciones para el sistema operativo Android), ya que el público objetivo en Colombia tan solo cuenta con un 8% de dispositivos iOS y menos de un 4% que ejecutan Windows Phone [2]. Su código fuente, manual de usuario y términos de licencia, se encuentran alojados en un sitio web y se espera que la aplicación evolucione con la ayuda de futuros estudiantes, docentes y demás entidades interesadas en estos procesos académicos del día a día.

## 1. PLANTEAMIENTO Y JUSTIFICACIÓN DEL PROBLEMA

El sistema operativo Android es uno de los más usados actualmente y cuenta con un gran número de apps disponibles para descargar. Para finales de 2014 se estima que el número de dispositivos que utilizan Android supere los 1100 millones, lo cual corresponderá a un 44% del mercado total de dispositivos en el mundo. Comparado con los más de 870 millones registrados para finales de 2013 (un 38% del mercado total de dispositivos en ese entonces), esta diferencia de 230 millones de dispositivos en tan solo un año hace evidente la importancia de Android en un mercado tan dinámico como el de los dispositivos móviles [3].

En los últimos años se ha venido presentando una mayor necesidad de adoptar el uso de aplicaciones móviles, alcanzando todos los sectores de la sociedad moderna [4].

Uno de ellos es el sector educativo, en el cual intervienen actividades administrativas y tareas docentes [5]. Estas últimas, ejecutadas a diario, consisten en tomar asistencia, llevar registro de calificaciones, registro de participación de los estudiantes, entre otras. Además requieren cierta independencia para almacenar sus registros diarios y llevar a cabo sus tareas de una forma más cómoda.

Este proyecto pretende dar solución a estas situaciones mediante la utilización de dispositivos móviles, para los cuales se desarrollará un prototipo que aproveche todo el potencial de tablets y smartphones. Además de los elementos de desarrollo de software, pruebas y requerimientos, para el desarrollo de este prototipo se hará énfasis en el modelo de permisos asociados al Manifiesto de Permisos de Android<sup>1</sup>. Para la elaboración de este prototipo se definieron los objetivos que se presentan a continuación.

---

<sup>1</sup> De este modelo, una de sus características está relacionada con el manifiesto de permisos, el cual define a qué partes del dispositivo tendrá acceso una aplicación [6]. Sin embargo, muchos programadores elaboran sus aplicaciones sin tener en cuenta el modelo de seguridad de Android, principalmente debido a que la aplicación puede correr sin necesidad de implementar dicho modelo.

## **2. OBJETIVOS**

### **2.1. OBJETIVO GENERAL**

Desarrollar un prototipo software móvil para dispositivos Android que facilite a docentes universitarios la gestión de información de estudiantes, gestión de temas de la asignatura y generalidades de la sesión docente, aprovechando las ventajas de los dispositivos móviles.

#### **2.1.1. OBJETIVOS ESPECÍFICOS**

1) Elaborar el DER Documento de Especificación de Requisitos simplificado de la aplicación.

2) Diseñar una aplicación que incluya las siguientes funcionalidades:

- Importar toda la información inicial para la aplicación (periodo, asignaturas, grupos, horarios y estudiantes) desde una base de datos remota,
- Registro de información adicional sobre los estudiantes (notas y/o recordatorios),
- Comunicación con los estudiantes (envío de correos electrónicos),
- Control de asistencia,
- Comentarios y recordatorios de las sesiones docentes,
- Exportar documentos de Excel/CSV con la información de asistencia de un grupo o un estudiante.

3) Caracterizar los permisos asociados al Manifiesto de Permisos del sistema operativo Android, indicando la importancia de cada uno de sus ítems y sus implicaciones técnicas.

### 3. MARCO DE REFERENCIA

#### 3.1. MARCO TEÓRICO

**3.1.1. Aplicaciones móviles (APPS).** En general, una de las funciones de los sistemas operativos es soportar y ejecutar aplicaciones (apps), facilitando la comunicación con las características de más bajo nivel en el dispositivo donde están instaladas. Por supuesto, esta misma filosofía aplica tanto para Android como para cualquiera de los demás sistemas operativos para dispositivos móviles (iOS, Windows Phone, BlackBerry OS, Firefox OS, Ubuntu Touch OS) [7].

El objetivo de las aplicaciones es facilitar la ejecución de una actividad, dando una serie de herramientas de interfaz de usuario que permitan llevarla a cabo. Inicialmente las aplicaciones se encontraban en el ámbito exclusivo de los computadores personales, pero poco a poco fueron migrando a otros dispositivos que tuvieran su propio sistema operativo instalado.

Con la llegada de los dispositivos móviles y smartphones (dimensiones reducidas y el mismo rendimiento de un computador personal en la palma de la mano), se produjo un cambio en la tendencia de las aplicaciones, convirtiéndolas en instrumentos que entregan contenido de forma pasiva y que no necesariamente permiten desarrollar tareas complejas [8].

Android dispone de un centro de descargas llamado Google Play, en el cual es posible tener acceso a aplicaciones gratuitas y propietarias [9]. Estas aplicaciones se encuentran organizadas por categorías y su publicación está sujeta a los criterios de aprobación de Google [10]. Publicar una aplicación en Google Play requiere que se realice un registro previamente, el cual tiene un costo de 25 dólares y se paga una sola vez [11].

**3.1.2. Gestión de cursos.** La actividad docente implica la ejecución de algunas actividades diarias, ya sea por tradición, por facilitar la organización de las sesiones docentes o simplemente porque el plantel educativo exige la realización de las mismas. A la hora de gestionar un curso (classroom management) las actividades más frecuentes incluyen: tomar asistencia (roll call), crear trabajos o evaluaciones, asignar calificaciones, asociar estudiantes por grupos o asignaturas, generar reportes de asistencia o calificaciones, comunicación con los estudiantes fuera de la sesión docente, entre otras. Teniendo identificadas estas actividades es posible plantear alternativas que faciliten su ejecución, desde las más convencionales (listas impresas) hasta los enfoques más recientes asociados a tecnologías de la información y la comunicación (software especializado). Cabe destacar que la ejecución de estos protocolos de clase también depende del nivel educativo en que se encuentren los estudiantes e incluso de la cantidad de estudiantes por grupo [12]. Es frecuente que la educación primaria y secundaria otorguen una mayor importancia a la toma de asistencia, comparado con la educación superior. En general, los primeros años en el sistema educativo son más estrictos en términos de la aplicación de estas actividades.

**3.1.3. Aplicaciones móviles para la gestión de cursos.** Al identificar las actividades claves en la gestión de cursos, el siguiente paso consiste en buscar ayudas tecnológicas que faciliten los procesos de las sesiones docentes [13]. Inicialmente se trabajó con aplicaciones de escritorio, las cuales evolucionaron para alojarse en internet y convertirse en aplicaciones web [14]. El paradigma informático más reciente es el de las aplicaciones móviles, y por ende, también surgieron apps para dispositivos Android, iOS y Windows Phone, que apoyaran las necesidades de las sesiones docentes. Estas necesidades incluyen: toma de asistencia, creación y calificación de actividades, creación de grupos de estudiantes y asignaturas, generar reportes y mantener una comunicación extraclase con los estudiantes por correo electrónico [15] u otras formas de mensajería instantánea.

## **3.2. ESTADO DEL ARTE**

En el mercado existe software para control de asistencia, gestión de notas y agenda de clases, tanto en el ámbito de las aplicaciones web [16] como en el ámbito de las apps para

dispositivos móviles en la categoría “education” [17]. Este proyecto se centra en las apps para dispositivos móviles, haciendo énfasis en las apps para Android. Las aplicaciones para dispositivos móviles se encuentran en las respectivas tiendas online de cada sistema operativo, siendo las más conocidas: Windows Phone Store, Google Play (Android), App Store (iOS) y BlackBerry World. Adicionalmente, es posible encontrar apps (en sus respectivos formatos según su sistema operativo) en otros sitios web, para descarga manual.

Para seleccionar las aplicaciones relevantes se utilizó el siguiente criterio: la búsqueda se realizará utilizando palabras clave que estén relacionadas con los objetivos de este proyecto (control de asistencia “attendance”, clases “class”, gestión de clases “class management” y “school control”, y aplicaciones para docentes “teacher apps”), Google Play mostrará una serie de apps para descargar, ya sea gratuitas o de pago. De ellas se tomarán aquellas con un puntaje mayor a 4 estrellas, que pertenezcan a la categoría “Education” y que se encuentren en los lugares más significativos (36 primeros puestos).

Se obtienen entonces los siguientes resultados:

<b>Criterios</b> <b>Palabra clave: “attendance”</b> <b>Categoría: “Education”</b> <b>Calificación: “4 stars+”</b>	
<b>App:</b> Attendance Register (students)  <b>Developer:</b> Rudra Nirvan	<ul style="list-style-type: none"> <li>- Calendario</li> <li>- Toma de asistencia (múltiples opciones)</li> <li>- Posibilidad de añadir comentarios cada clase</li> <li>- Estadísticas de asistencia</li> <li>- Edición de nombres de asignaturas</li> <li>- Ordenar visualmente las asignaturas</li> <li>- Acceso con contraseña</li> </ul>
<b>App:</b> Attendance Manager  <b>Developer:</b> Yash Ketkar	<ul style="list-style-type: none"> <li>- Toma de asistencia</li> <li>- Estadísticas de asistencia</li> <li>- Límite de fallas (porcentaje)</li> <li>- Posibilidad de importar y exportar backups</li> </ul>

<p><b>App:</b> Attendance Taker <b>Developer:</b> Ferid Cafer</p>	<ul style="list-style-type: none"> <li>- Toma de asistencia</li> <li>- Estadísticas de asistencia</li> <li>- Historial de asistencias pasadas</li> <li>- Gráficas estadísticas semanales</li> <li>- Exporta asistencia (Microsoft Excel)</li> </ul>
<p><b>App:</b> m Attendance <b>Developer:</b> Shiksha Apps</p>	<ul style="list-style-type: none"> <li>- Toma de asistencia</li> <li>- Estadísticas de asistencia</li> <li>- Importar grupos (Microsoft Excel)</li> <li>- Exporta asistencia (Microsoft Excel)</li> </ul>
<p><b>App:</b> K12 Attendance <b>Developer:</b> K12 Inc</p>	<ul style="list-style-type: none"> <li>- Toma de asistencia (por horas)</li> <li>- Límite de fallas (porcentaje y alertas)</li> <li>- Muestra la asistencia de los últimos 10 días</li> <li>- Acceso con contraseña</li> </ul>
<p><b>App:</b> Safety Attendance - Evaluation <b>Developer:</b> ForSafetySolutions</p>	<ul style="list-style-type: none"> <li>- Toma de asistencia (personalizada)</li> <li>- Posibilidad de integración con códigos QR</li> <li>- Importar grupos (CSV, Contact List)</li> <li>- Permite tomar o asignar fotografías</li> <li>- Posibilidad de añadir premios y distinciones</li> <li>- Distribución en el aula (Seating chart)</li> <li>- Exporta asistencia (CSV)</li> <li>- Módulo para calificar actividades (cualitativo)</li> <li>- Envío de reportes por correo electrónico</li> <li>- Posibilidad de importar y exportar backups</li> <li>- Selección aleatoria de estudiantes</li> <li>- Soporte multi idioma</li> </ul>
<p><b>App:</b> AMIZONER (for Amity Students) <b>Developer:</b> metalwihen</p>	<ul style="list-style-type: none"> <li>- Toma de asistencia</li> <li>- Estadísticas de asistencia</li> <li>- Límite de fallas (código de colores)</li> <li>- Posibilidad de ingresar un límite de fallas (%)</li> <li>- Horario de clases</li> <li>- Sincronización con servidor central</li> <li>- Acceso con contraseña</li> </ul>

<p><b>App:</b> Jadawel <b>Developer:</b> Madar Software</p>	<ul style="list-style-type: none"> <li>- Toma de asistencia</li> <li>- Alertas al alcanzar límite de fallas</li> <li>- Horario de clases</li> <li>- Multi rol (docente o estudiante)</li> <li>- Alerta si se aproxima una sesión docente</li> <li>- Soporte multi idioma</li> </ul>
<p><b>App:</b> Teacher's Gradebook - Additio <b>Developer:</b> Didactic Labs, S.L.</p>	<ul style="list-style-type: none"> <li>- Calendario</li> <li>- Toma de asistencia (múltiples opciones)</li> <li>- Horario de clases</li> <li>- Permite asignar fotografías</li> <li>- Envío de correos electrónicos a los padres</li> <li>- Posibilidad de importar y exportar backups</li> <li>- Importar grupos (CSV)</li> <li>- Exporta asistencia (CSV, RTF)</li> <li>- Exporta calificaciones (CSV, RTF)</li> <li>- Selección aleatoria de estudiantes</li> <li>- Acceso con contraseña</li> </ul>
<p><b>App:</b> Teacher Gradebook FREE <b>Developer:</b> Fran Meneu</p>	<ul style="list-style-type: none"> <li>- Toma de asistencia (múltiples opciones)</li> <li>- Horario de clases</li> <li>- Permite asignar fotografías</li> <li>- Módulo para calificar actividades (cualitativo, cuantitativo)</li> <li>- Posibilidad de importar y exportar backups</li> <li>- Importar grupos (CSV)</li> <li>- Exporta asistencia (CSV, PDF)</li> <li>- Exporta calificaciones (CSV, PDF)</li> <li>- Selección aleatoria de estudiantes</li> <li>- Acceso con contraseña</li> </ul>
<p><b>App:</b> TeacherKit - Class manager <b>Developer:</b> ITWorx Inc.</p>	<ul style="list-style-type: none"> <li>- Toma de asistencia (personalizada)</li> <li>- Módulo para calificar disciplina</li> <li>- Importar grupos</li> <li>- Permite tomar o asignar fotografías</li> </ul>

	<ul style="list-style-type: none"> <li>- Comunicación con los padres</li> <li>- Soporte multi idioma</li> </ul>
<b>App:</b> Pocket Attendance <b>Developer:</b> RnL Developers	<ul style="list-style-type: none"> <li>- Calendario</li> <li>- Toma de asistencia (múltiples opciones)</li> <li>- Horario de clases</li> <li>- Alerta si se aproxima una sesión docente</li> <li>- Estadísticas de asistencia</li> <li>- Posibilidad de crear notificaciones</li> <li>- Diseñada para un rol de estudiante</li> </ul>

<p><b>Criterios</b></p> <p><b>Palabra clave:</b> “class”</p> <p><b>Categoría:</b> “Education”</p> <p><b>Calificación:</b> “4 stars+”</p>
<p>(Las aplicaciones que aparecen en los resultados son principalmente agendas para estudiantes y herramientas para aprender idiomas. En su mayoría permiten crear horarios de clases y notificaciones).</p>

<p><b>Criterios</b></p> <p><b>Palabra clave:</b> “class management”</p> <p><b>Categoría:</b> “Education”</p> <p><b>Calificación:</b> “4 stars+”</p>	
<b>App:</b> Google Classroom <b>Developer:</b> Google Inc.	<ul style="list-style-type: none"> <li>- Módulo para calificar actividades (Google Drive y Docs)</li> <li>- Crear grupos (Google Drive y Docs)</li> <li>- Comunicación con estudiantes (Google+)</li> <li>- Gratuito para instituciones educativas</li> <li>- Exportar documentos de clase (Google Drive y Docs)</li> <li>- Backups automáticos (Google Drive y Docs)</li> </ul>

<b>App:</b> Classroom Management Student <b>Developer:</b> Mythware	- Función “levantar la mano” - Mensajería instantánea con el docente - Envío y recepción de archivos
--	--

<b>Criterios</b> <b>Palabra clave:</b> “school control” <b>Categoría:</b> “Education” <b>Calificación:</b> “4 stars+”	
<b>App:</b> LiveSchool Teacher <b>Developer:</b> LiveSchool Inc.	(Rol de docente – primaria/secundaria) - Sistema de puntos por buen comportamiento - Comentarios centralizados (varios docentes) - Reportes semanales para los padres - Se emula una "cuenta de banco" con los puntos ganados por los estudiantes
<b>App:</b> LiveSchool Student <b>Developer:</b> LiveSchool Inc.	(Rol de estudiante - primaria/secundaria) - Consulta del sistema de puntos (que emula una "cuenta de banco") en tiempo real - Ver las puntuaciones acumuladas - Ver los comentarios de los docentes - Ver las recompensas por puntuaciones adquiridas
<b>App:</b> Grade Tracker Pro (Free!) <b>Developer:</b> LetsLearn - NSouthProductions	(Rol de estudiante) - Posibilidad de llevar control de notas - Promedio por asignatura, por semestre y por categorías - Calendario/Agenda - Horario de clases - Escala de notas personalizada (cualitativa o cuantitativa) - Cálculo de la siguiente nota para aprobar la asignatura - Posibilidad de importar y exportar backups

	<ul style="list-style-type: none"> <li>- Posibilidad de acumular promedios semestrales</li> </ul>
<p><b>App:</b> School Timetable Deluxe <b>Developer:</b> Tobias Schürg</p>	<p>(La aplicación se encuentra en idioma alemán)</p> <ul style="list-style-type: none"> <li>- Horario de clases</li> <li>- Posibilidad de añadir el app como Widget</li> </ul>
<p><b>App:</b> Quick TimeTable <b>Developer:</b> IDAL MEDIA</p>	<p>(Rol de estudiante)</p> <ul style="list-style-type: none"> <li>- Horario de clases</li> <li>- Posibilidad de añadir el app como Widget</li> <li>- Varios tipos de representación del horario</li> <li>- Posibilidad de importar y exportar backups</li> <li>- Soporte multi idioma</li> </ul>
<p><b>App:</b> Timetable <b>Developer:</b> Gabriel Ittner</p>	<p>(Rol de estudiante)</p> <ul style="list-style-type: none"> <li>- Sincronización con servidor central</li> <li>- Posibilidad de almacenar notas y notificaciones</li> <li>- Tipos de representación del horario (lista o cuadrícula)</li> <li>- Búsquedas dentro del horario de clases</li> <li>- Posibilidad de añadir un Widget de sesiones y tareas</li> <li>- Alerta si se aproxima una sesión docente</li> <li>- Posibilidad de silenciar el teléfono desde la aplicación</li> </ul>
<p><b>App:</b> School Assistant + <b>Developer:</b> Gil Castro</p>	<p>(Rol de estudiante)</p> <ul style="list-style-type: none"> <li>- Posibilidad de llevar control de notas</li> <li>- Alerta de exámenes próximos</li> <li>- Integración con Google Drive y Google Calendar</li> <li>- Soporte multiusuario</li> <li>- Posibilidad de añadir el app como Widget</li> <li>- Alerta si se aproxima una sesión docente</li> <li>- Posibilidad de silenciar el teléfono desde la aplicación</li> <li>- Acceso con contraseña</li> </ul>

<p><b>App:</b> Edmodo <b>Developer:</b> Edmodo, Inc</p>	<ul style="list-style-type: none"> <li>- Orientado a ciclos de primaria y secundaria en Estados Unidos y Canadá (K-12)</li> <li>- Su enfoque emula una red social</li> <li>- Permite comunicarse fuera de la clase</li> <li>- Permite crear actividades y encuestas</li> <li>- Reportes de notas individuales y colectivas</li> <li>- Agrega archivos a una carpeta personal</li> </ul>
---	---

<p><b>Criterios</b></p> <p><b>Palabra clave:</b> “teacher apps”</p> <p><b>Categoría:</b> “Education”</p> <p><b>Calificación:</b> “4 stars+”</p>	
<p><b>App:</b> Teacher <b>Developer:</b> Pixelsight</p>	<ul style="list-style-type: none"> <li>- Permite almacenar información de estudiantes (datos personales, teléfono, E-mail, nombre del docente, fotografía, notificaciones).</li> <li>- Permite enviar correos electrónicos a los estudiantes o a otros docentes.</li> <li>- Permite tomar o asignar fotografías</li> <li>- Permite guardar fotografías de eventos o actividades</li> <li>- Búsqueda de estudiantes por nombre o apellido</li> <li>- Creación de actividades y exámenes</li> <li>- Permite filtrar estudiantes por género, grupo, asignatura, puntaje, código u orden de lista.</li> </ul>
<p><b>App:</b> Remind <b>Developer:</b> Remind101</p>	<ul style="list-style-type: none"> <li>- Envío de mensajes a grupos, asignaturas o estudiantes individuales (se mantiene la privacidad de los números telefónicos)</li> <li>- Chat con capacidad para 10 personas</li> <li>- Posibilidad de postear contenidos sobre las sesiones docentes</li> </ul>

	<ul style="list-style-type: none"><li>- Posibilidad de compartir archivos entre docentes y estudiantes</li></ul>
<b>App:</b> HomeWork <b>Developer:</b> Klwinkel.com	<ul style="list-style-type: none"><li>- Horario de clase</li><li>- Notificaciones de tareas y exámenes</li><li>- Permite almacenar máximo 20 asignaturas por día (horario limitado)</li><li>- Permite realizar cambios temporales al horario de clase</li><li>- Permite personalizar el layout de inicio</li></ul>

## 4. HERRAMIENTAS

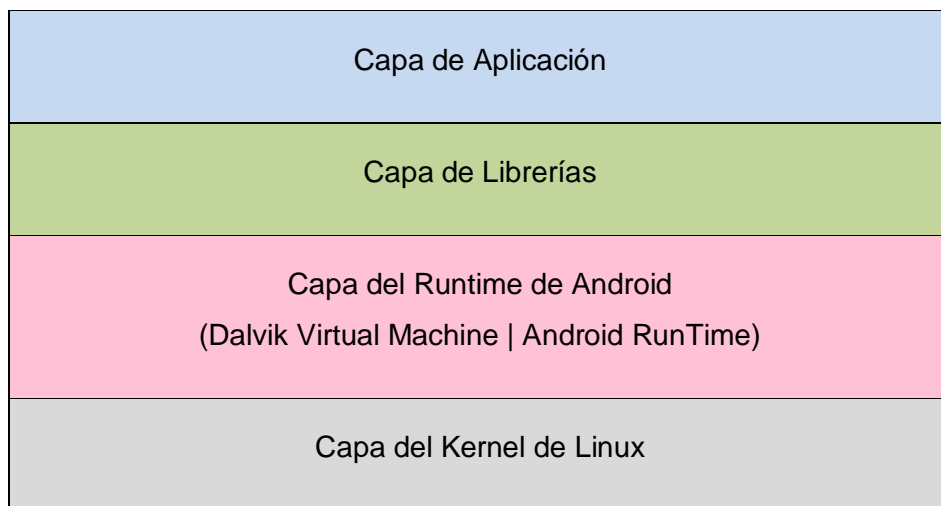
### 4.1. ANDROID

Android es un sistema operativo móvil basado en el kernel de Linux. Actualmente se encuentra en un 44% del total de dispositivos móviles alrededor del mundo, y las proyecciones indican que a finales de 2015 existirán más de 2600 millones de dispositivos con este sistema operativo (es decir, más del 47% del mercado total de dispositivos) [3].

Fue creado por Android Inc en 2003, inicialmente pensado para pequeños dispositivos como cámaras fotográficas digitales. En 2005 Android fue comprado por Google, y se empezó a hacer énfasis en la importancia de las preferencias del usuario, su ubicación geográfica y la integración con los primeros smartphones. Oficialmente, entró al mercado de los dispositivos móviles en octubre de 2008 [18].

La arquitectura de Android sigue un patrón de pila (stack de software) con una serie de capas internas especializadas. Esto permite que las funciones de una capa utilicen elementos de las capas inferiores fácilmente [19], como se indica en la Figura 1.

Figura 1. Capas en la arquitectura de Android.



Se observan cuatro capas principales [20]:

#### a) Capa de Aplicación

Esta capa proporciona algunas interfaces de programación, tales como: proveer contenido, gestión de recursos, gestión de notificaciones.

#### b) Capa de Librerías

Las librerías de Android incluyen librerías enlazadas de C y C++, conectadas hacia arriba con la capa de aplicación, y hacia abajo con la capa del kernel de Linux. Entre otras, se dispone de librerías asociadas al lenguaje C, una librería de multimedia, una librería de red, una librería de base de datos, una librería de visualización 3D y una librería de manejo de fuentes.

#### c) Capa del Runtime de Android

Esta capa incluye la librería de comunicación con el kernel y la máquina virtual Dalvik (ahora llamada Android Runtime - ART), la cual se usa para ejecutar los programas de Java, cada uno procesándose como un sistema embebido. Sus programas ocupan menos espacio en memoria y es posible ejecutar varios programas al mismo tiempo.

#### d) Capa del Kernel de Linux

Android utiliza un kernel de Linux, para manejar los servicios básicos del sistema (gestión de memoria, gestión de procesos, pilas, seguridad, y todo lo relacionado con drivers). Adicionalmente, el kernel actúa como una capa de comunicación entre aplicaciones y hardware.

## **4.2. ANDROID STUDIO (IDE)**

Android Studio es un entorno integrado de desarrollo (IDE) promocionado por Google Inc. desde mayo de 2013 [21], y actualmente distribuido como versión estable. Se encuentra disponible para sistemas operativos Windows, Linux y Mac OS X. Google anunció que es el IDE oficial para desarrollo de aplicaciones nativas Android [22] y en la actualidad se debate sobre una posible migración masiva desde la plataforma Eclipse, lo cual ha generado polémica entre los desarrolladores [23].

Este IDE se basa en la plataforma IntelliJ IDEA, la cual inició como un proyecto de la empresa JetBrains [24] para desarrolladores de aplicaciones en Java y fue evolucionando desde la incorporación de módulos ADT (Android Developer Tools) hasta una integración parcial con los requerimientos para trabajo con Android [25]. Google Inc. adoptó este proyecto, mejoró la integración con las librerías internas de Android y liberó su propia distribución, denominándola “Android Studio” [26].

Android Studio cuenta con algunas ventajas respecto a otros IDE [27]:

- Tiene ayudas de código (Code Completion) más intuitivas que Eclipse.
- Permite ver en tiempo real los cambios que se hagan al archivo XML que codifica la interfaz gráfica de las aplicaciones, sin tener que cambiar de ventana.
- Se integra perfectamente con Gradle dando un mejor control sobre las configuraciones de compilación, firma digital de archivos apk (signing configurations) y versiones para diferentes dispositivos (application flavors).
- Las funciones de refactoring (renombrar variables, métodos y clases) son mucho más confiables que las de Eclipse.
- Puede generar múltiples archivos apk mucho más rápido que Eclipse y además permite configurar buenas prácticas de desarrollo (strings, colores y valores en archivos XML) para asegurar la calidad del código que se elabora.
- Agregar librerías (archivos .jar) es más sencillo y con menos posibilidades de fallas, comparado con el mismo proceso en otros IDE. Incluso cuando se tienen librerías que dependen de otras librerías preexistentes.

#### **4.3. ANDROID SOFTWARE DEVELOPMENT KIT (SDK)**

En ocasiones el flujo de trabajo y las condiciones de desarrollo de aplicaciones implican que se utilice un IDE diferente a Android Studio. Los motivos de este fenómeno pueden ser tan sencillos como la disposición previa de un IDE diferente, o que el equipo de desarrollo ya estaba capacitado para el manejo de otras herramientas y se quiere evitar costos extras de entrenamiento.

Particularmente, cuando se está desarrollando aplicaciones para dispositivos Android es necesario utilizar, en conjunto con el IDE adoptado, el Android SDK. Esta herramienta permite ejecutar comandos de consola, compilar mediante instrucciones puntuales, correr scripts que automaticen los procesos de ensamble, compatibilidad con Gradle, entre otras funciones que normalmente se ejecutan en segundo plano y son transparentes al desarrollador [28].

Las aplicaciones más utilizadas del SDK incluyen, entre otras, al Android SDK Manager, AVD Manager, Mobile Device Emulator y el Dalvik Debug Monitor Server (DDMS).

El SDK Manager lista los componentes de compilación y ejecución disponibles, las API de las diferentes versiones para realizar pruebas de compatibilidad y permite actualizar componentes específicos del entorno tan pronto estén disponibles las notificaciones para cambio de versión.

AVD Manager es el gestor de máquinas virtuales del SDK (Android Virtual Devices), el cual dispone de una interfaz gráfica que permite crear y configurar las VMs que se correrán con el Mobile Device Emulator.

Mobile Device Emulator (o simplemente “emulador”) permite ejecutar las máquinas virtuales que se configuraron con el AVD Manager, sin necesidad de conectar físicamente un dispositivo. Esto facilita las pruebas preliminares de funcionalidad durante la etapa de diseño y desarrollo.

Por otra parte, el Dalvik Debug Monitor Server (DDMS) permite el mapeo de puertos (port-forwarding services), capturas de pantalla en el dispositivo (screenshots), información sobre hilos (thread) y pilas (heap) en el dispositivo, logs o informes de errores (logcat), procesos generales de la arquitectura por capas de Android, y muchas otras funcionalidades [29].

#### **4.4. SQLITE**

SQLite es un sistema de gestión de bases de datos relacional que utiliza una gran parte del lenguaje estándar SQL-92 [30]. Está programado en lenguaje C y se distribuye como una librería que se adapta a varios tipos de sistemas con desempeño reducido [31].

Se caracteriza por un tamaño compacto, rapidez a la hora de ejecutar las operaciones que se le indiquen, fácil integración en entornos con bajas prestaciones de memoria y procesamiento, acoplamiento directo con la aplicación que lo utiliza (contrario al modelo

cliente-servidor clásico) y una estrategia de bloqueo del archivo en donde se almacena cada base de datos (lo cual acelera aún más las transacciones).

Android Studio incorpora a SQLite dentro de su estructura principal [32], de fábrica, siendo entonces innecesario descargar el instalador tradicional o el conector JDBC como librería .jar [33] (como ocurría con Eclipse y otros IDE).

SQLite cuenta con una alta tolerancia a fallos debido a las pruebas exhaustivas que se le realizan antes de liberar una versión: exploración de casos de uso por medio de árboles de decisión (llegando a abarcar un 100% de las posibles situaciones), desbordamientos de memoria, valores límite para cada elemento de la base de datos [34], fallos por cortes de suministro de energía eléctrica en el dispositivo, pruebas de regresión, tecnología Atomic Commit [35], entre otras [36].

Es posible realizar ajustes para incrementar el desempeño de SQLite en dispositivos en donde la gestión de memoria es crítica, por ejemplo: revisando la manera en que se procesan las operaciones con cadenas de texto con la librería Bionic C (los query son cadenas de texto), la velocidad de lectura y escritura según el dispositivo de almacenamiento, el sistema de archivos (EXT3 o EXT4 para Android) e incluso se plantea la posibilidad de utilizar la GPU de los dispositivos para liberar carga en el procesador [37].

#### **4.5. JAVA EXCEL API (JXL)**

Java Excel API es una librería que permite crear, leer y modificar archivos de Microsoft Excel en tiempo de ejecución. Está desarrollada con el lenguaje Java, por tanto, se puede ejecutar en cualquier dispositivo que utilice la máquina virtual de Java [38].

La madurez de este proyecto es visible, ya que tiene una buena documentación y ha venido perfeccionándose desde 2003. Es posible utilizar JXL en entornos web, en servlets de JSP y por supuesto, en el desarrollo de aplicaciones nativas para Android. Cualquier entorno basado en Java puede utilizar esta librería [39].

Se distribuye como un archivo .jar, pero también existen versiones con el código fuente empaquetado como archivo .zip, y una variante para trabajar con C Sharp.

JXL tiene la capacidad de leer archivos Microsoft Excel (.xls) 95, 97, 2000 y 2003. Puede generar archivos de la versión 97 en adelante, leer y escribir fórmulas dentro de las celdas, dar formato a textos, a números y a fechas. Es posible también colorear el fondo, bordes y patrones a las celdas [39]. Permite insertar imágenes, copiar gráficos estadísticos (charts) y otras funciones adicionales.

Se eligió JXL debido a que las demás librerías para trabajo con archivos .xls exigen una configuración más compleja, ocupan un mayor espacio en la aplicación y requieren la instalación de software adicional. Un ejemplo de esta problemática es Apache POI [40], el cual posee una completa cobertura para toda la suite de Microsoft Office, pero su dificultad para la implementación impidió su uso en este proyecto.

#### **4.6. MPANDROIDCHART LIBRARY**

En ciertas ocasiones resulta más eficiente observar la información de una aplicación como gráficos estadísticos. MP Android Chart nos brinda esa posibilidad utilizando pocas líneas de código, ya que al ser una librería, contiene métodos que generan los gráficos automáticamente [41].

MP Android Chart permite también interactuar con los gráficos generados (girar, reposicionar, animar y seleccionar áreas), sin afectar las medidas de desempeño en fps para Android. Esta librería fue elaborada con el lenguaje Java, por lo tanto, se distribuye en un paquete ".JAR" y su uso dará mejores resultados si se mantiene en entornos de trabajo basados en Java (JFC, JVM, Android nativo, JSP, etc).

La dinámica de trabajo es la siguiente: la librería requiere una serie de valores numéricos para dimensionar las gráficas a generar, una serie de labels o rótulos para añadir información a la gráfica, indicaciones para posicionar una guía de categorías y seleccionar un tipo de gráfico.

Se dispone de gráficos redondos (PieChart), gráficos de barras (BarChart simples o agrupados), gráficos lineales (simples, por puntos o polinomios grado 3), gráficos de dispersión, Candle Stick Charts (para información financiera), Bubble Charts y Radar Charts (spider web chart).

#### **4.7. CLASE SIMPLEFILEDIALOG**

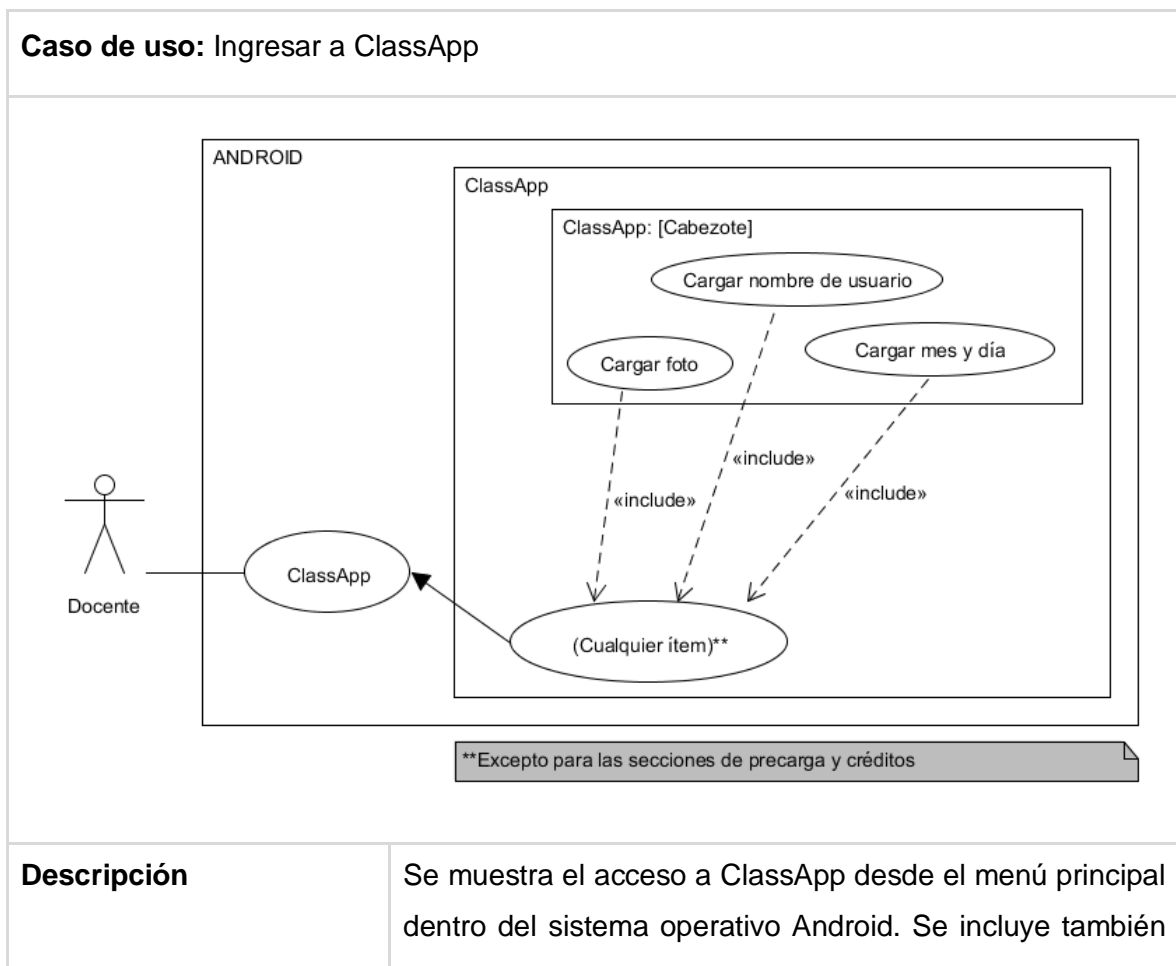
Es una implementación que utiliza llamados a las clases Dialog, AlertDialog y DialogInterface, las cuales logran mostrar un explorador de archivos con: título, texto descriptivo, zona de contenido y botones de confirmación positiva o negativa. Fue elaborada inicialmente por Gregory Shpitalnik [42] y tuvo una gran acogida en la comunidad de desarrolladores de Android, debido a que no se cuenta con una clase que emule un explorador de archivos con opciones estándar (abrir, guardar, crear carpeta). Los desarrolladores han personalizado esta implementación [43] de manera informal o dando crédito al autor bajo una licencia CPOL (Code Project Open License) [44].

## 5. ESPECIFICACIÓN DE REQUISITOS

### 5.1. APROXIMACIÓN PRELIMINAR (HEURÍSTICOS DE BRUEGGE Y DUTOIT)

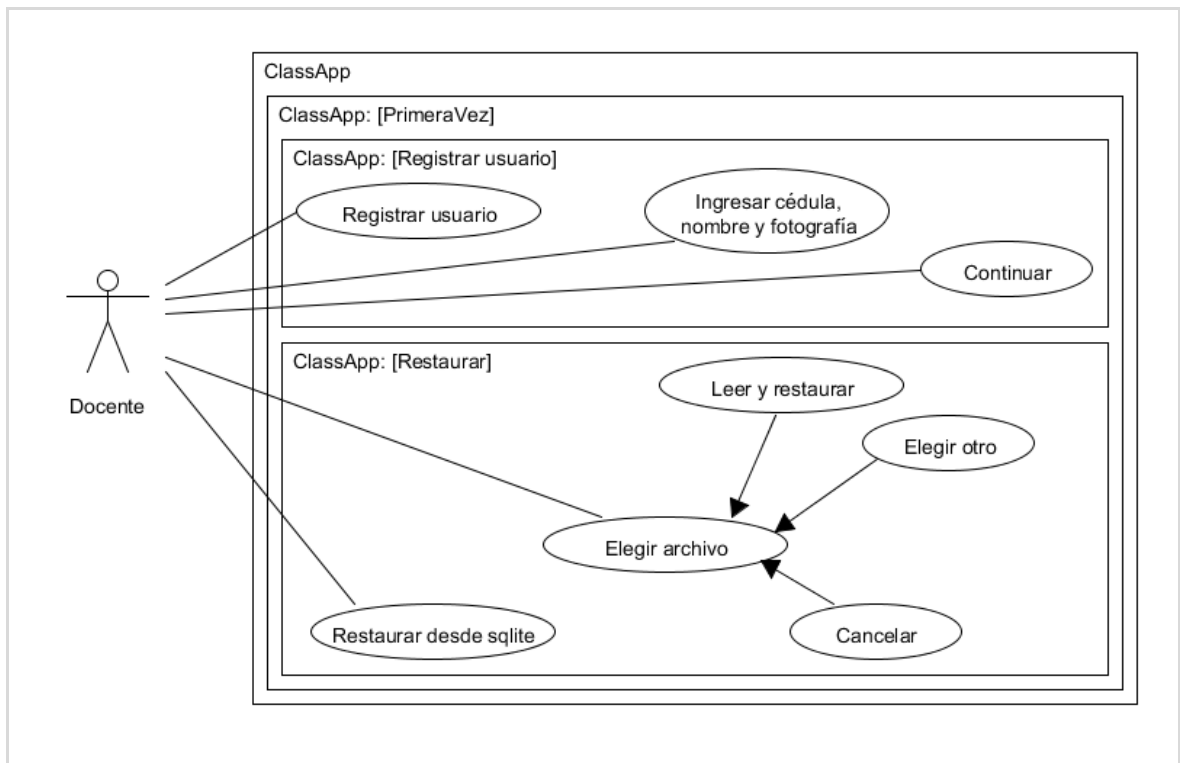
Antes de redactar el documento de especificación de requisitos, es necesario tener claro quiénes son los actores, casos de uso, escenarios excepcionales, posibles relaciones entre casos de uso (en este caso no aplican), objetos del análisis y requerimientos no funcionales. Bruegge y Dutoit [45] proponen una serie de heurísticos (preguntas) que facilitan la obtención de estos elementos (puede verse en el Anexo A).

### 5.2. IDENTIFICACIÓN DE CASOS DE USO



	un esquema que describe qué funciones especiales son llamadas dentro de todas las pantallas de la aplicación (excepto la precarga y créditos), ya que son mostradas de manera uniforme (fotografía, nombre de usuario, mes y día del mes). Se especifican las opciones que pueden ocurrir si existe (o no) un usuario registrado.
<b>Actores participantes</b>	Docente
<b>Flujo de eventos</b>	Encontrándose en el menú de aplicaciones de Android: 1. El usuario ejecuta la aplicación (ícono ClassApp) 2. ClassApp muestra una precarga de bienvenida 3. ¿Es la primera vez que se ingresa a ClassApp?: Sí: Se muestran dos opciones para registrar un usuario NO: ClassApp muestra el menú principal (Portada)
<b>Condiciones de entrada</b>	Que se encuentre instalada la aplicación en el dispositivo.
<b>Condiciones de salida</b>	El docente sale de la aplicación presionando el botón de [inicio] del dispositivo Android.

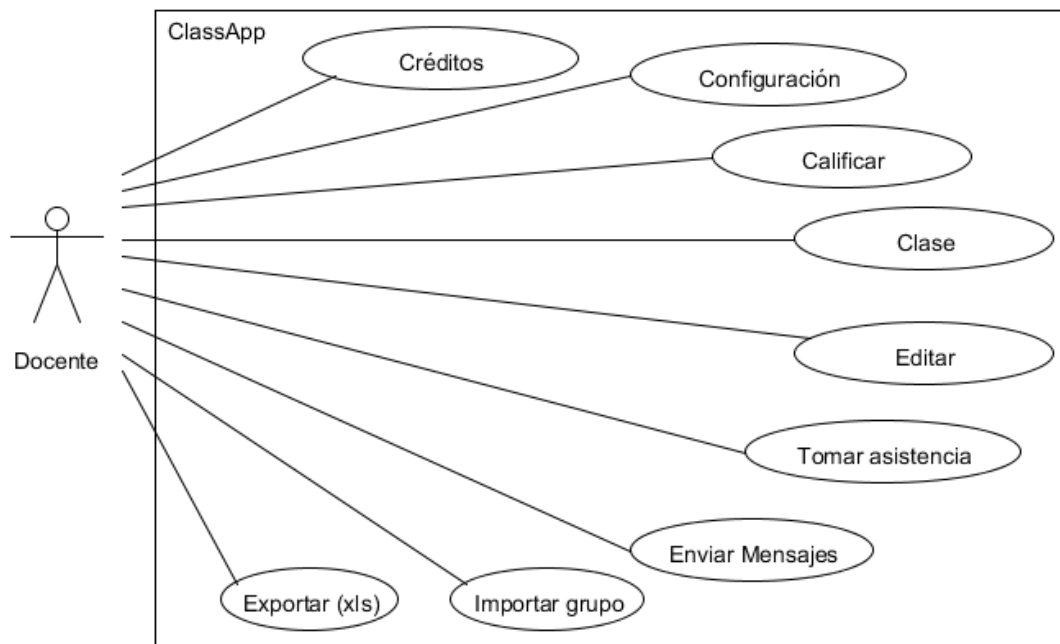
**Caso de uso:** Registrar usuario (primera vez)



<b>Descripción</b>	Se muestra el primer acceso a ClassApp desde el momento de su instalación. ClassApp requiere que se configure un usuario para poder utilizar todas sus funciones.
<b>Actores participantes</b>	Docente
<b>Flujo de eventos</b>	<p>Encontrándose en el menú de aplicaciones de Android:</p> <ol style="list-style-type: none"> <li>1. El usuario ejecuta la aplicación (ícono ClassApp)</li> <li>2. ClassApp muestra una precarga de bienvenida</li> <li>3. ClassApp muestra la pantalla “primera vez”</li> </ol> <p>3.1. El usuario presiona el botón “Registrarse como usuario único”</p> <p>3.1.1. El usuario deberá ingresar cédula y nombre. La fotografía es opcional.</p> <p>3.1.2. Se crea el usuario y el usuario presiona el botón “Continuar”.</p>

	<p>3.2. El usuario presiona el botón “Restaurar desde archivo SQLite”</p> <p>3.2.1. El usuario presiona el botón “Elegir archivo”</p> <p>3.2.2. El usuario busca el archivo sqlite</p> <p>3.2.3. El usuario presiona el botón “Leer y restaurar”</p> <p>3.2.4. Se pregunta al usuario si también desea restaurar fotografías</p> <p>4. ClassApp muestra el menú principal (Portada)</p>
<b>Condiciones de entrada</b>	Que se encuentre instalada la aplicación en el dispositivo y que todavía NO exista un usuario registrado.
<b>Condiciones de salida</b>	El docente sale de la aplicación presionando el botón de [inicio] del dispositivo Android.

### Caso de uso: Portada de ClassApp (Inicial)



#### Descripción

Tan pronto el usuario ha ingresado a ClassApp, este caso de uso describe las opciones que se le presentan por medio de un menú de íconos (Portada).

#### Actores participantes

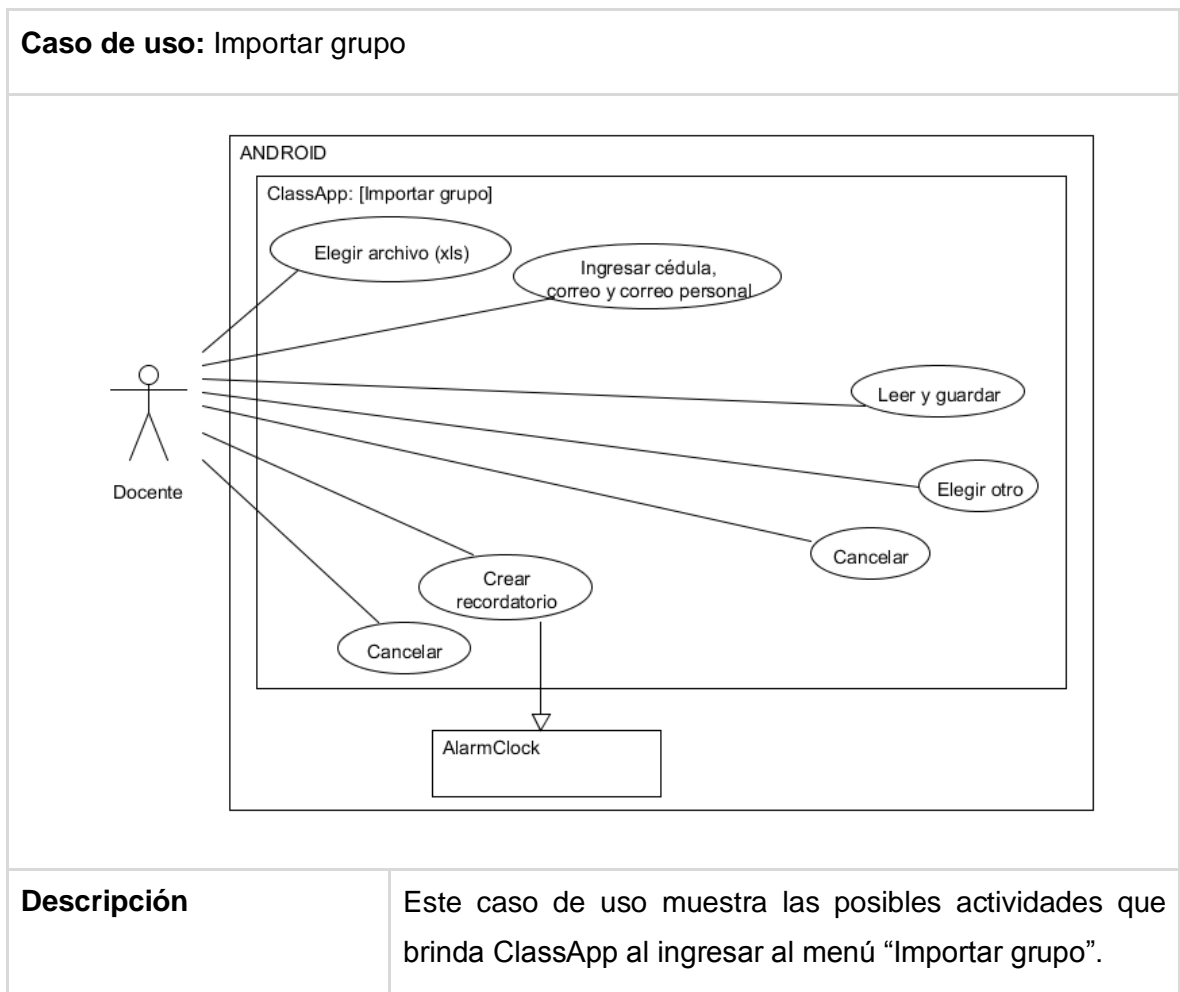
Docente

#### Flujo de eventos

Encontrándose en ClassApp:

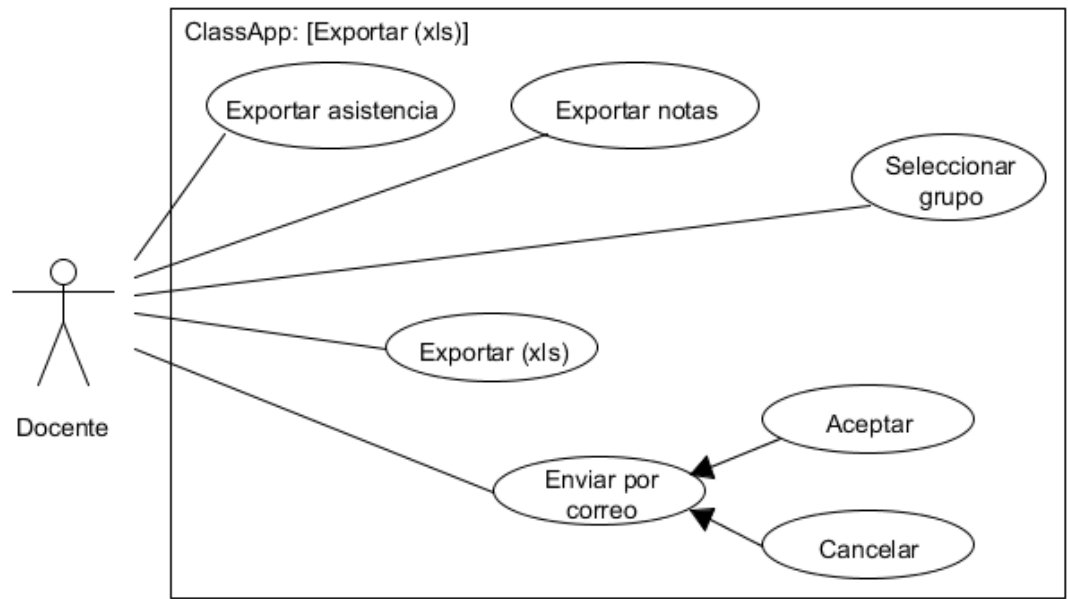
1. ClassApp muestra el menú principal (Portada)
- 1.1. El usuario presiona el botón "Tomar asistencia"
- 1.2. El usuario presiona el botón "Importar grupo"
- 1.3. El usuario presiona el botón "Exportar"
- 1.4. El usuario presiona el botón "Clase"
- 1.5. El usuario presiona el botón "Calificar"
- 1.6. El usuario presiona el botón "Editar"
- 1.7. El usuario presiona el botón "Enviar mensajes"

	<p>1.8. El usuario presiona el botón “Configuración”</p> <p>1.9. El usuario presiona el botón “Créditos”</p> <p>2. Se muestra la pantalla que corresponda a la opción elegida</p>
<b>Condiciones de entrada</b>	Que se encuentre instalada la aplicación en el dispositivo y que exista un usuario previamente.
<b>Condiciones de salida</b>	El docente sale de la aplicación presionando el botón de [inicio] del dispositivo Android.



<b>Actores participantes</b>	Docente
<b>Flujo de eventos</b>	<ol style="list-style-type: none"> <li>1. El usuario ejecuta la aplicación</li> <li>2. ClassApp muestra el menú de opciones iniciales</li> <li>3. El usuario presiona el botón “Importar grupo”</li> <li>4. El usuario presiona el botón “Elegir archivo”</li> <li>5. El usuario busca el archivo (xls) en donde está el listado de estudiantes para el grupo a crear</li> <li>6. El usuario deberá ingresar su cédula y correo institucional. El correo personal es opcional. <ol style="list-style-type: none"> <li>6.1. El usuario presiona el botón “Leer y guardar” <ol style="list-style-type: none"> <li>6.1.1. Los estudiantes, el grupo, la asignatura, el nombre del docente, el periodo y la demás información, se almacenan.</li> <li>6.1.2. ClassApp pregunta al usuario si desea crear un recordatorio (a manera de horario de clases) para este grupo, o si desea finalizar en ese punto.</li> <li>6.1.3. En caso afirmativo, el usuario ingresa la(s) hora(s) de inicio de la clase, y es redirigido a la aplicación AlarmClock de Android. Allí, es posible seleccionar los días en que se debe repetir la(s) alarma(s).</li> </ol> </li> <li>6.2. El usuario presiona el botón “Elegir otro” (regresa al numeral 3)</li> <li>6.3. El usuario presiona el botón “Cancelar” (es enviado a la portada)</li> </ol> </li> </ol>
<b>Condiciones de entrada</b>	Que se encuentre instalada la aplicación en el dispositivo y que exista un usuario previamente.
<b>Condiciones de salida</b>	El docente sale de la aplicación presionando el botón de [inicio] del dispositivo Android.

### Caso de uso: Exportar (xls)



<b>Descripción</b>	Este caso de uso muestra las posibles actividades que brinda ClassApp al ingresar al menú “Exportar (xls)”.
<b>Actores participantes</b>	Docente
<b>Flujo de eventos</b>	<ol style="list-style-type: none"><li>1. El usuario ejecuta la aplicación</li><li>2. ClassApp muestra el menú de opciones iniciales</li><li>3. El usuario presiona el botón “Exportar (xls)”</li><li>4. ClassApp muestra las opciones de exportación<ol style="list-style-type: none"><li>4.1. El usuario presiona el botón “Exportar asistencia”</li><li>4.2. El usuario presiona el botón “Exportar notas”</li></ol></li><li>5. ClassApp pide que se seleccione un grupo del listado</li><li>6. El usuario presiona el botón “Exportar (xls)”</li><li>7. Se genera el archivo en la carpeta Downloads</li><li>8. Se pregunta al usuario si desea enviar este archivo por</li></ol>

	correo automáticamente (requiere tener un cliente de correo instalado)
<b>Condiciones de entrada</b>	Que se encuentre instalada la aplicación en el dispositivo y que exista un usuario previamente. También se requiere que exista al menos un grupo previamente importado.
<b>Condiciones de salida</b>	El docente sale de la aplicación presionando el botón de [inicio] del dispositivo Android.

<b>Caso de uso: Tomar asistencia</b>	
<pre> classDiagram     actor Docente     class ClassApp {         class TomarAsistencia {             Tomar asistencia Grupo [X]             Aceptar             Cancelar         }         TomarAsistencia         Seleccionar grupo         Volver al listado         Tomar foto         Dialogo fin         Dialogo guardar         Asignar asistencia     }     Docente --&gt; TomarAsistencia     Docente --&gt; Seleccionar grupo     Docente --&gt; Volver al listado     Docente --&gt; Tomar foto     Docente --&gt; Dialogo fin     Docente --&gt; Dialogo guardar     Docente --&gt; Asignar asistencia     TomarAsistencia --&gt; Seleccionar grupo     Asignar asistencia --&gt; Presente     Asignar asistencia --&gt; Ausente     Asignar asistencia ..&gt; Cargar foto : «include»     Asignar asistencia ..&gt; UltimaAsistencia : «include»     Asignar asistencia ..&gt; AsistenciaPromedio : «include»     Asignar asistencia ..&gt; DesempenoEstudiante : «include»     </pre>	
<b>Descripción</b>	Este caso de uso muestra las posibles actividades que brinda ClassApp al ingresar al menú “Tomar asistencia” y al recibir una alerta de “Tomar asistencia” (opcional).
<b>Actores participantes</b>	Docente

## Flujo de eventos

1. El usuario ejecuta la aplicación
2. ClassApp muestra el menú de opciones iniciales
3. El usuario presiona el botón “Tomar asistencia”
4. ClassApp pide que se seleccione un grupo del listado
5. Tan pronto se selecciona un grupo, aparece una pantalla con el primer estudiante. Esta pantalla, a su vez, realiza llamados a funciones para cargar “última asistencia”, “asistencia promedio”, “desempeño del estudiante”, la fotografía del estudiante (si existe), su nombre y su código. Se ofrece la opción de tomar una fotografía y de regresar al ListView de estudiantes. Aparecen también los botones “presente” y “ausente” para asignar la asistencia de ese estudiante.
  - 5.1. El usuario presiona el botón “presente”
    - 5.1.1. Se almacena “presente” para ese estudiante y se regresa al numeral 5 con el siguiente estudiante (según el orden de lista).
  - 5.2. El usuario presiona el botón “ausente”
    - 5.2.1. Se almacena “ausente” para ese estudiante y se regresa al numeral 5 con el siguiente estudiante (según el orden de lista).
  - 5.3. El usuario presiona el botón “Tomar foto”
    - 5.3.1. Android abre la interfaz de la cámara del dispositivo
    - 5.3.2. El usuario toma la fotografía del estudiante
    - 5.3.3. La fotografía es almacenada automáticamente
    - 5.3.4. Se regresa al numeral 5, sin cambiar de estudiante.
  - 5.4. El usuario presiona el botón “Volver al listado”
    - 5.4.1. ClassApp regresa al ListView de estudiantes

	<p>6. Al llegar al último estudiante, existen dos opciones:</p> <p>6.1. Se llegó allí porque manualmente se seleccionó el último estudiante.</p> <p>6.1.1. ClassApp muestra un diálogo en donde se indica al usuario que, si bien llegó al final de la lista, es posible que existan estudiantes sin asignar asistencia.</p> <p>6.1.2. El usuario tiene dos opciones:</p> <p>6.1.2.1. El usuario presiona el botón “Revisar” (y es enviado al ListView de estudiantes).</p> <p>6.1.2.2. El usuario presiona el botón “Cancelar” (y es enviado a la portada. No se almacena la asistencia).</p> <p>6.2. Se llegó allí porque se recorrió el listado completamente.</p> <p>6.2.1. ClassApp muestra un diálogo en donde se indica al usuario que ya se asignó asistencia a todos los estudiantes.</p> <p>6.2.2. El usuario tiene dos opciones:</p> <p>6.2.2.1. El usuario presiona el botón “Guardar la asistencia” (la asistencia es almacenada y se actualizan las estadísticas de cada estudiante automáticamente).</p> <p>6.2.2.2. El usuario presiona el botón “Cancelar” (No se almacena la asistencia).</p> <p>6.2.3. Para ambos casos, el usuario es redireccionado a la portada y se muestra un mensaje de estado.</p>
<p><b>Condiciones de entrada</b></p>	<p>Que se encuentre instalada la aplicación en el dispositivo y que exista un usuario previamente. También se requiere que exista al menos un grupo previamente importado.</p>

<b>Condiciones de salida</b>	El docente sale de la aplicación presionando el botón de [inicio] del dispositivo Android.
------------------------------	--

<b>Caso de uso: Enviar mensajes</b>	
<b>Descripción</b>	Este caso de uso muestra las posibles actividades que brinda ClassApp al ingresar al menú “Enviar mensajes”.
<b>Actores participantes</b>	Docente
<b>Flujo de eventos</b>	<ol style="list-style-type: none"> <li>1. El usuario ejecuta la aplicación</li> <li>2. ClassApp muestra el menú de opciones iniciales</li> <li>3. El usuario presiona el botón “Enviar mensajes”</li> <li>4. Aparece el menú de enviar mensajes</li> </ol> <ol style="list-style-type: none"> <li>4.1. El usuario presiona el botón “Enviar correo</li> </ol>

	<p>electrónico”</p> <p>4.1.1. Aparece el menú de selección de destinatarios</p> <p>4.1.1.1. El usuario presiona el botón “Ver grupos”</p> <p>4.1.1.1.1. El usuario selecciona uno de los grupos</p> <p>4.1.1.2. El usuario presiona el botón “Ver asignaturas”</p> <p>4.1.1.2.1. El usuario selecciona una de las asignaturas</p> <p>4.1.1.3. El usuario presiona el botón “Buscar estudiante”</p> <p>4.1.1.3.1. El usuario ingresa nombre/apellido/código de algún estudiante y presiona el botón “Buscar”</p> <p>4.1.1.3.2. El usuario selecciona uno de los estudiantes</p> <p>4.1.1.4. El usuario presiona el botón “Personalizado”</p> <p>4.1.1.4.1. El usuario ingresa el correo electrónico del destinatario y presiona el botón “Guardar y continuar”</p> <p>4.2. El usuario presiona el botón “Enviar mensaje de WhatsApp”</p> <p>5. Aparece el menú de selección de plantilla del mensaje y se da la opción de redactar un mensaje desde cero</p> <p>5.1. El usuario presiona el botón “Redactar mensaje desde cero”</p> <p>5.2. El usuario selecciona cualquiera de las 4 plantillas</p> <p>6. El usuario presiona el botón “¡Enviar mensaje!”</p> <p>7. El mensaje y los destinatarios son entregados a la aplicación respectiva (cliente de correo   WhatsApp)</p> <p>8. Al regresar de la aplicación que gestiona el envío del mensaje, ClassApp muestra un mensaje de confirmación.</p> <p>9. El usuario presiona “Finalizar y continuar” y es redireccionado a la portada.</p>
<b>Condiciones de entrada</b>	Que se encuentre instalada la aplicación en el dispositivo y

	que exista un usuario previamente. También se requiere que exista al menos un grupo previamente importado.
<b>Condiciones de salida</b>	El docente sale de la aplicación presionando el botón de [inicio] del dispositivo Android.

<b>Caso de uso: Calificar</b>	
<b>Descripción</b>	Este caso de uso muestra las posibles actividades que brinda ClassApp al ingresar al menú “Calificar”.
<b>Actores participantes</b>	Docente
<b>Flujo de eventos</b>	<ol style="list-style-type: none"> <li>1. El usuario ejecuta la aplicación</li> <li>2. ClassApp muestra el menú de opciones iniciales</li> <li>3. El usuario presiona el botón “Calificar”</li> <li>4. ClassApp pide que se seleccione un grupo del listado</li> </ol>

5. ClassApp pide que se seleccione un trabajo del listado  
6. Tan pronto se selecciona un trabajo, aparece una pantalla con el primer estudiante. Esta pantalla, a su vez, realiza llamados a funciones para cargar el porcentaje del trabajo, nombre del trabajo, fotografía del estudiante (si existe), su nombre y su código. Se ofrece la opción de regresar al ListView de estudiantes. Aparecen también los botones de incremento y decremento para la cifra entera y decimal de la calificación. Igualmente, aparece el botón “Calificar”, para asignar la nota.

6.1. El usuario presiona el botón “+” de la cifra entera

6.1.1. La cifra entera se incrementa en una unidad

6.2. El usuario presiona el botón “-” de la cifra entera

6.2.1. La cifra entera disminuye en una unidad

6.3. El usuario presiona el botón “+” de la cifra decimal

6.3.1. La cifra decimal se incrementa en una unidad

6.4. El usuario presiona el botón “-” de la cifra decimal

6.4.1. La cifra decimal disminuye en una unidad

6.5. El usuario presiona el botón “Volver al listado”

6.5.1. ClassApp regresa al ListView de estudiantes

6.6. El usuario presiona el botón “Calificar”

6.6.1. Se almacena la calificación que se asignó para ese estudiante, en ese trabajo, y se regresa al numeral 6 con el siguiente estudiante (según el orden de lista).

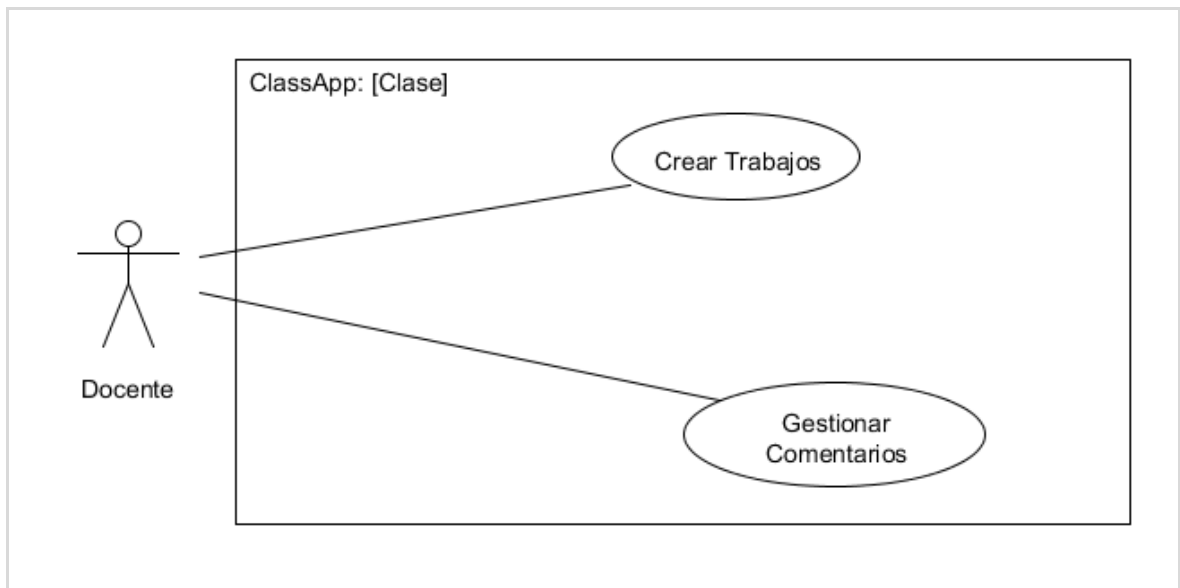
7. Al llegar al último estudiante, existen dos opciones:

7.1. Se llegó allí porque manualmente se seleccionó el último estudiante.

7.1.1. ClassApp muestra un diálogo en donde se indica al usuario que, si bien llegó al final de la lista, es posible que

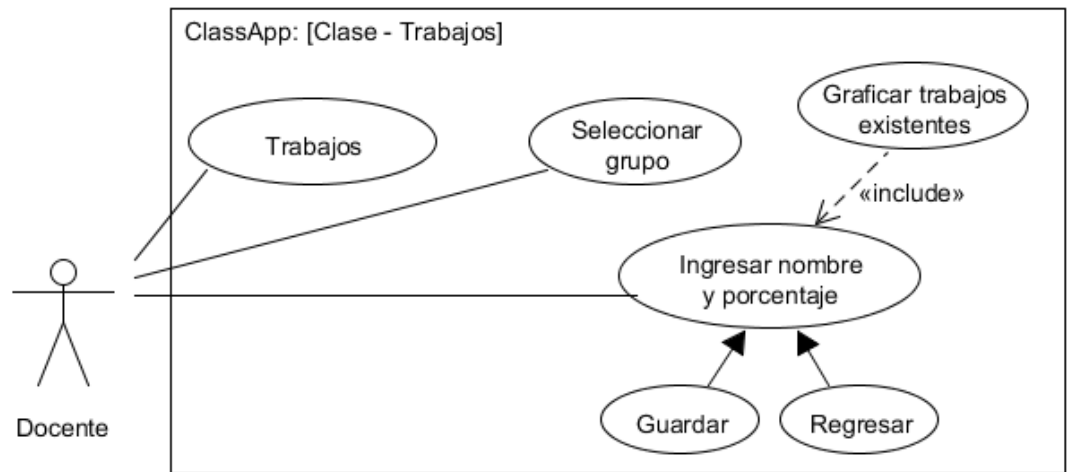
	<p>existan estudiantes sin asignar una calificación.</p> <p>7.1.2. El usuario tiene dos opciones:</p> <p>7.1.2.1. El usuario presiona el botón “Revisar” (y es enviado al ListView de estudiantes).</p> <p>7.1.2.2. El usuario presiona el botón “Cancelar” (y es enviado a la portada. No se almacenan las calificaciones).</p> <p>7.2. Se llegó allí porque se recorrió el listado completamente.</p> <p>7.2.1. ClassApp muestra un diálogo en donde se indica al usuario que ya se calificaron todos los estudiantes.</p> <p>7.2.2. El usuario tiene dos opciones:</p> <p>7.2.2.1. El usuario presiona el botón “Guardar notas” (las calificaciones son almacenadas y se actualizan las estadísticas de cada estudiante automáticamente).</p> <p>7.2.2.2. El usuario presiona el botón “Cancelar” (No se almacenan las calificaciones).</p> <p>7.2.3. Para ambos casos, el usuario es redireccionado a la portada y se muestra un mensaje de estado.</p>
<b>Condiciones de entrada</b>	Que se encuentre instalada la aplicación en el dispositivo y que exista un usuario previamente. También se requiere que exista al menos un grupo previamente importado.
<b>Condiciones de salida</b>	El docente sale de la aplicación presionando el botón de [inicio] del dispositivo Android.

**Caso de uso:** Clase



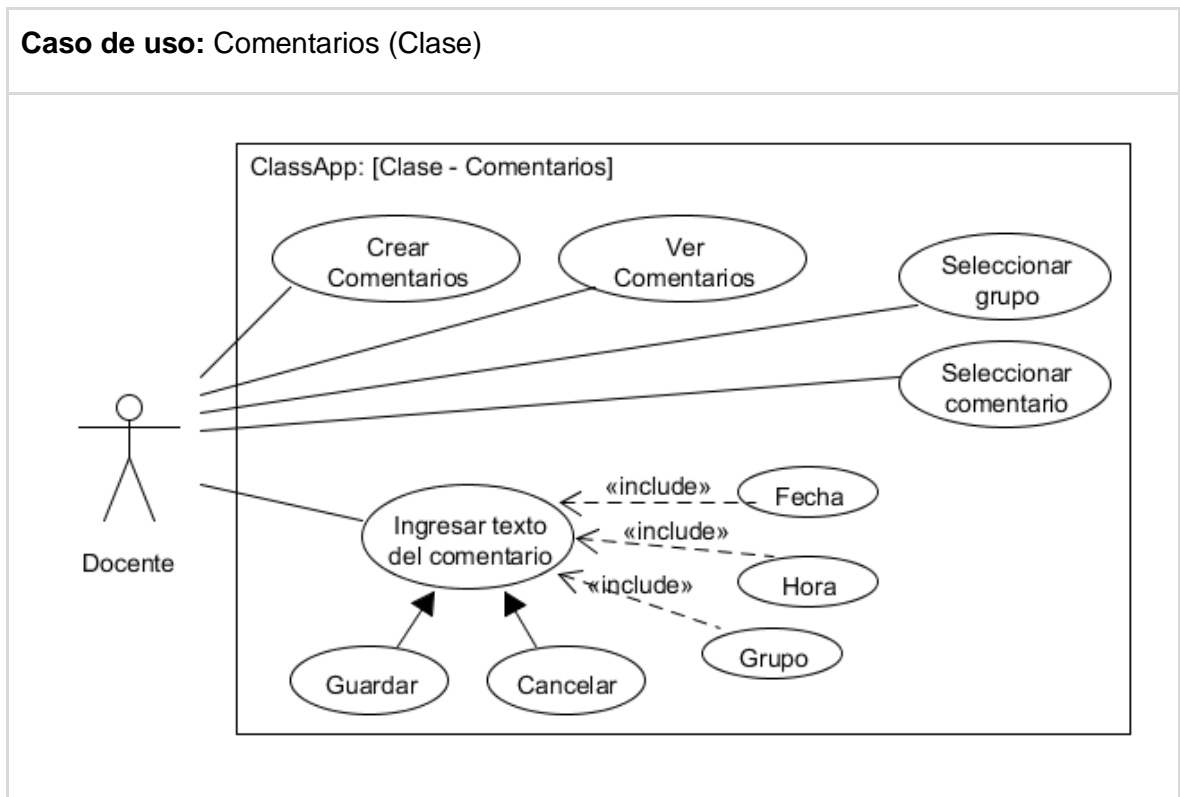
<b>Descripción</b>	Este caso de uso muestra las posibles actividades que brinda ClassApp al ingresar al menú “Clase”.
<b>Actores participantes</b>	Docente
<b>Flujo de eventos</b>	<ol style="list-style-type: none"> <li>1. El usuario ejecuta la aplicación</li> <li>2. ClassApp muestra el menú de opciones iniciales</li> <li>3. El usuario presiona el botón “Clase”</li> <li>4. ClassApp muestra las opciones disponibles (Crear trabajos   Gestionar comentarios)</li> </ol>
<b>Condiciones de entrada</b>	Que se encuentre instalada la aplicación en el dispositivo y que exista un usuario previamente. También se requiere que exista al menos un grupo previamente importado.
<b>Condiciones de salida</b>	El docente sale de la aplicación presionando el botón de [inicio] del dispositivo Android.

**Caso de uso: Trabajos (Clase)**



<b>Descripción</b>	Este caso de uso muestra las posibles actividades que brinda ClassApp al ingresar al menú “Trabajos (Clase)”.
<b>Actores participantes</b>	Docente
<b>Flujo de eventos</b>	<ol style="list-style-type: none"> <li>1. El usuario ejecuta la aplicación</li> <li>2. ClassApp muestra el menú de opciones iniciales</li> <li>3. El usuario presiona el botón “Clase”</li> <li>4. El usuario presiona el botón “Trabajos”</li> <li>5. ClassApp pide que se seleccione un grupo del listado</li> <li>6. El usuario deberá ingresar el nombre del trabajo y su porcentaje correspondiente.</li> <li>7. El usuario tiene dos opciones:             <ol style="list-style-type: none"> <li>7.1. El usuario presiona el botón “Guardar”                 <ol style="list-style-type: none"> <li>7.1.1. El trabajo es creado y almacenado</li> <li>7.1.2. Se actualiza automáticamente el gráfico circular (PieChart) de la parte inferior de la pantalla</li> <li>7.1.3. Se regresa al numeral 6, hasta que se alcance un</li> </ol> </li> </ol> </li> </ol>

	<p>100% en la suma de todos los trabajos, o hasta que el usuario presione el botón “Regresar”.</p> <p>7.2. El usuario presiona el botón “Regresar”</p> <p>7.2.1. El usuario es redirigido a la portada</p>
<b>Condiciones de entrada</b>	Que se encuentre instalada la aplicación en el dispositivo y que exista un usuario previamente. También se requiere que exista al menos un grupo previamente importado.
<b>Condiciones de salida</b>	El docente sale de la aplicación presionando el botón de [inicio] del dispositivo Android.



<b>Descripción</b>	Este caso de uso muestra las posibles actividades que brinda ClassApp al ingresar al menú “Comentarios (Clase)”.
<b>Actores participantes</b>	Docente
<b>Flujo de eventos</b>	<ol style="list-style-type: none"> <li>1. El usuario ejecuta la aplicación</li> <li>2. ClassApp muestra el menú de opciones iniciales</li> <li>3. El usuario presiona el botón “Clase”</li> <li>4. El usuario presiona el botón “Gestionar comentarios”</li> <li>5. El usuario tiene dos opciones: <ol style="list-style-type: none"> <li>5.1. El usuario presiona el botón “Crear comentario” <ol style="list-style-type: none"> <li>5.1.1. ClassApp pide que se seleccione un grupo del listado</li> <li>5.1.2. El usuario deberá ingresar el texto del comentario, y verificar que la fecha, hora y grupo que se muestran en pantalla sean correctos.</li> <li>5.1.3. El usuario tiene dos opciones: <ol style="list-style-type: none"> <li>5.1.3.1. El usuario presiona el botón “Guardar comentario”</li> <li>5.1.3.2. El usuario presiona el botón “Cancelar” (no se guarda el comentario) <ol style="list-style-type: none"> <li>5.1.3.2.1. El usuario es redirigido a la portada</li> </ol> </li> </ol> </li> <li>5.2. El usuario presiona el botón “Ver comentarios” <ol style="list-style-type: none"> <li>5.2.1. ClassApp pide que se seleccione un grupo del listado</li> <li>5.2.2. ClassApp pide que se seleccione un comentario del listado</li> <li>5.2.3. El comentario seleccionado es mostrado en su totalidad al usuario</li> </ol> </li> </ol> </li> </ol> </li></ol>
<b>Condiciones de entrada</b>	Que se encuentre instalada la aplicación en el dispositivo y

	que exista un usuario previamente. También se requiere que exista al menos un grupo previamente importado.
<b>Condiciones de salida</b>	El docente sale de la aplicación presionando el botón de [inicio] del dispositivo Android.

<b>Caso de uso: Editar</b>	
<pre> classDiagram     actor Docente     class ClassApp["ClassApp: [Editar]"]     ClassApp --&gt; UC1("Editar Comentario")     ClassApp --&gt; UC2("Editar correo personal")     ClassApp --&gt; UC3("Editar Trabajo")     ClassApp --&gt; UC4("Editar Calificaciones")     ClassApp --&gt; UC5("Editar Asistencia")   </pre>	
<b>Descripción</b>	Este caso de uso muestra las posibles actividades que brinda ClassApp al ingresar al menú “Editar”.
<b>Actores participantes</b>	Docente
<b>Flujo de eventos</b>	<ol style="list-style-type: none"> <li>1. El usuario ejecuta la aplicación</li> <li>2. ClassApp muestra el menú de opciones iniciales</li> <li>3. El usuario presiona el botón “Editar”</li> <li>4. El usuario tiene seis opciones: <ol style="list-style-type: none"> <li>4.1. El usuario presiona el botón “Editar comentario”</li> </ol> </li> </ol>

	<p>4.2. El usuario presiona el botón “Editar correo personal”</p> <p>4.3. El usuario presiona el botón “Editar trabajo”</p> <p>4.4. El usuario presiona el botón “Editar calificaciones”</p> <p>4.5. El usuario presiona el botón “Editar asistencia”</p> <p>5. ClassApp muestra el contenido que existía previamente</p> <p>6. El usuario deberá ingresar/editar los nuevos contenidos</p> <p>7. El usuario tiene dos opciones:</p> <p>7.1. El usuario presiona el botón “Guardar cambios”</p> <p>7.1.1. El contenido que se editó es almacenado</p> <p>7.1.2. Se actualizan las estadísticas internamente (en los casos que aplique)</p> <p>7.2. El usuario presiona el botón “Cancelar” (no se guardan los cambios)</p> <p>8. El usuario es redirigido a la portada</p>
<p><b>Condiciones de entrada</b></p>	<p>Que se encuentre instalada la aplicación en el dispositivo y que exista un usuario previamente. También se requiere que exista al menos un grupo previamente importado.</p>
<p><b>Condiciones de salida</b></p>	<p>El docente sale de la aplicación presionando el botón de [inicio] del dispositivo Android.</p>

### 5.3. DOCUMENTO DE ESPECIFICACIÓN DE REQUISITOS SIMPLIFICADO

Núm.	Rol involucrado	Objetivo (Característica)	Funcionalidad
1	Docente	"Gestionar mis clases con poco esfuerzo"	La solución que se desarrolle deberá tener un carácter portable (ejecutarse en dispositivos móviles), fácil de usar y con tiempos de respuesta cortos. Se da a entender que se requiere una solución de procesamiento por lotes, especialmente para el almacenamiento de datos de estudiantes.
2	Docente	"Crear estudiantes"	La aplicación deberá brindar la posibilidad de almacenar datos de estudiantes, y tener ciertas estadísticas sobre cada uno.
3	Docente	"Crear grupos"	Los grupos están constituidos por estudiantes, lo cual implica que estos dos últimos requisitos están asociados. Se debe dar la posibilidad al usuario de ingresar estudiantes y que automáticamente sean relacionados con un grupo.
4	Docente	"Crear asignaturas"	Una asignatura puede tener varios grupos asociados, lo cual implica otra relación a nivel de objetos. El sistema debe brindar la posibilidad de crear una asignatura, ya sea manualmente o por un proceso que acceda al código y nombre de la misma.

5	Docente	“Crear periodo académico”	En cada periodo académico se dictan varias asignaturas y se crean varios grupos. La aplicación debe garantizar una manera para la creación de un periodo académico. De esta manera, se puede mantener la integridad referencial en caso de eliminar entidades. El sistema debe brindar la posibilidad de crear un periodo, ya sea manualmente o por un proceso que acceda a esa información de manera indirecta.
6	Docente	“Identificar estudiantes”	Una forma rápida de identificar a un estudiante es por medio de su fotografía. Funcionalmente, esto implica acceder a los medios de adquisición de imágenes del sistema (API de la cámara) y al sistema de archivos (para almacenar la fotografía). Teniendo el registro fotográfico del estudiante, es posible identificarlo cómodamente a la hora de tomar asistencia y asignar calificaciones.
7	Docente	“Tomar lista rápidamente y poder corregir errores”	La aplicación debe brindar los medios para tomar asistencia de una manera ágil, que reduzca la posibilidad de fallos (identificación del estudiante, interfaz amigable, posibilidad de regresar y editar) y que requiera de la menor cantidad de pasos posible.

8	Docente	“Crear trabajos para cada grupo y poder asignarles un porcentaje”	Se requiere que el sistema garantice los medios para crear una actividad, asociarla a un grupo, identificarla por medio de una descripción y darle un porcentaje de la nota definitiva.
9	Docente	“Calificar trabajos y poder corregir errores”	La aplicación debe brindar los medios para calificar actividades de una manera ágil, que reduzca la posibilidad de fallos (identificación del estudiante, interfaz amigable, posibilidad de regresar y editar) y que requiera de la menor cantidad de pasos posible.
10	Docente	“Necesito tener un informe de asistencia del grupo en Excel”	Contando con la información de asistencia previamente almacenada, es posible generar un archivo de Microsoft Excel (.xls) con la información del grupo que se requiera. El formato deberá asociar fechas de clase y estudiantes, destacando las sesiones en donde el estudiante fue reportado como “ausente”.
11	Docente	“Necesito tener un informe de las notas del grupo en Excel”	Contando con las calificaciones previamente almacenadas, es posible generar un archivo de Microsoft Excel (.xls) con la información del grupo que se requiera. El formato deberá asociar actividades (trabajos o exámenes) y estudiantes, mostrando las notas de cada estudiante del grupo.

			Se asume que es deseable reportar una nota definitiva automáticamente, ya que se cuenta con los porcentajes de cada actividad.
12	Docente	<p>“A veces se necesita enviar un mensaje urgente a los estudiantes, fuera de las horas de clase, y que le llegue a todos automáticamente.</p> <p>Si la aplicación va a ser móvil, sería práctico poder enviar mensajes por WhatsApp”</p>	<p>Este requerimiento implica dos requisitos: que se disponga del correo electrónico de cada estudiante (preferiblemente el correo personal) y del número celular (con previo permiso del estudiante) para poder contactarlo vía WhatsApp.</p> <p>Otro elemento a destacar es la necesidad de enviar mensajes a un grupo entero. Incluso a una asignatura entera (varios grupos), y se asume que el docente no desea digitar todas las direcciones manualmente.</p> <p>El envío de mensajes por WhatsApp es una opción muy práctica, pero en ese caso, se requiere que los grupos o asignaturas estén registrados previamente como “chat de grupo” dentro de WhatsApp. Esto obedece a que WhatsApp no dispone de un API que permita crear “chat de grupo” desde una aplicación externa.</p>
13	Docente	<p>(Recordatorios)</p> <p>“Quiero que me salga una alarma que me recuerde a qué hora tengo clase”</p>	El usuario desea programar un mecanismo de notificación para saber en qué momento tiene una sesión docente. Android dispone de una serie de clases (Calendar, Alarm,

			AlarmClock) que facilitan esta tarea. Su interfaz predefinida implementa un selector de repetición (días de la semana), un selector de hora (TimePicker) y un Message (texto) complementario.
14	Docente	(Comentarios) “Quiero poder guardar notas sobre los temas que se vieron en la clase de hoy”	Este requerimiento implica la asociación de una cadena de texto a una fecha determinada y a un grupo de clase. Implica también que deberá implementarse un campo de texto suficientemente amplio y que la base de datos tenga un tipo de estructura adecuada para almacenar textos de cierta longitud.
15	Docente	(Editar) “¿Se puede cambiar algún dato después?, sería bueno poder corregir errores más adelante”	Esta funcionalidad se contempla para algunos elementos del contenido ingresado a ClassApp. Algunos datos corresponden a llaves primarias y deben ser conservados para no entrar en conflicto con la integridad referencial en SQLite. La funcionalidad es limitada y debe advertírsele al usuario.
16	Docente	(Backup y restauración) “No quiero que se me pierda la información”	El usuario desea que exista una funcionalidad que genere un archivo en donde se almacene toda la información de la base de datos. Se asume que también desea almacenar las fotografías de los estudiantes que tenga matriculados hasta ese momento.

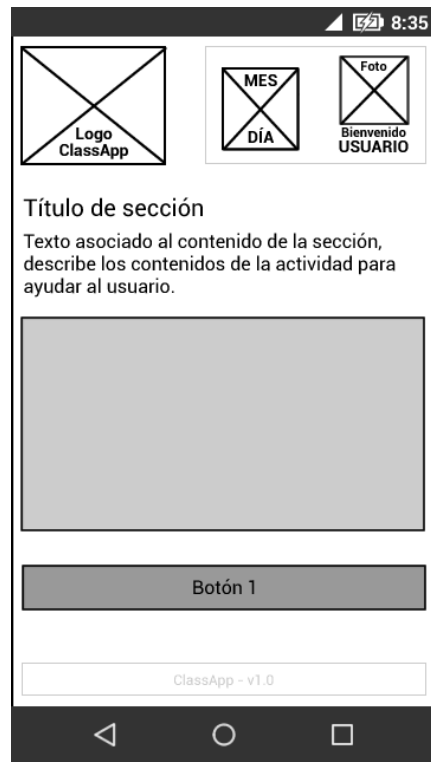
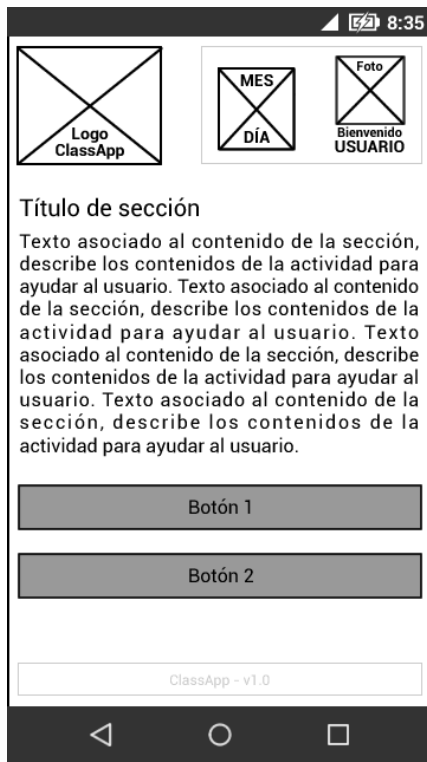
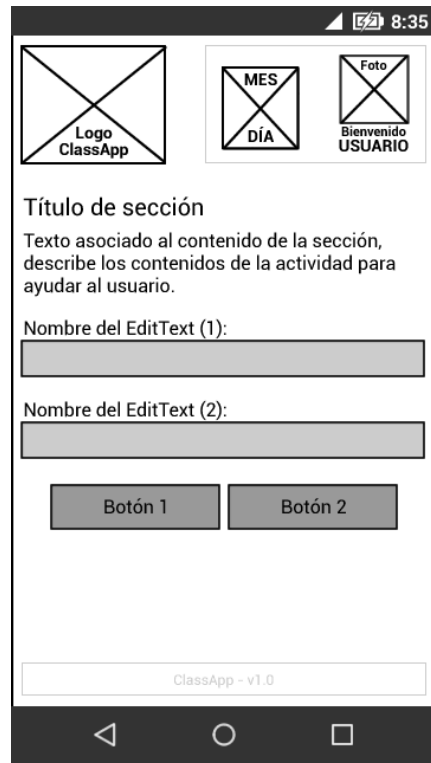
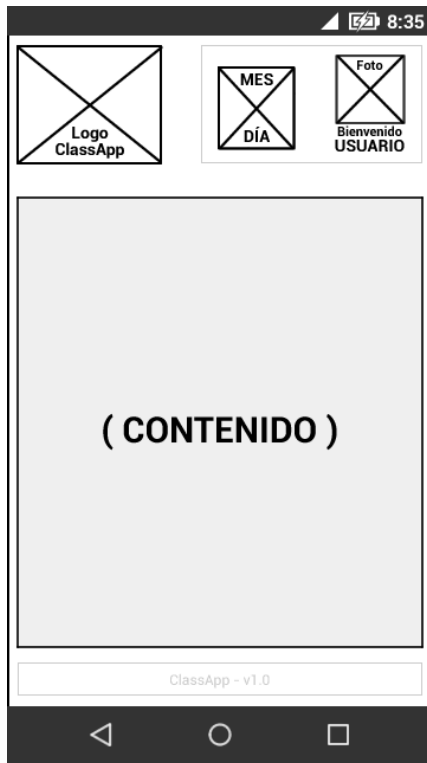
			Igualmente, se observa que el docente muestra su intención de recuperar los datos luego de que la aplicación sufra un daño o se invoque el reset de fábrica.
17	Docente	(Reset de fábrica) “¿Cómo hago para dejar el programa como cuando lo descargué?”	Existen algunas situaciones en donde los usuarios desean volver al estado inicial de la aplicación. Algunas circunstancias durante la fase de desarrollo también requieren que exista ese comando. Se utiliza entonces una serie de instrucciones SQL-92 para eliminar las tablas y forzar a la creación de una base de datos sin registros.

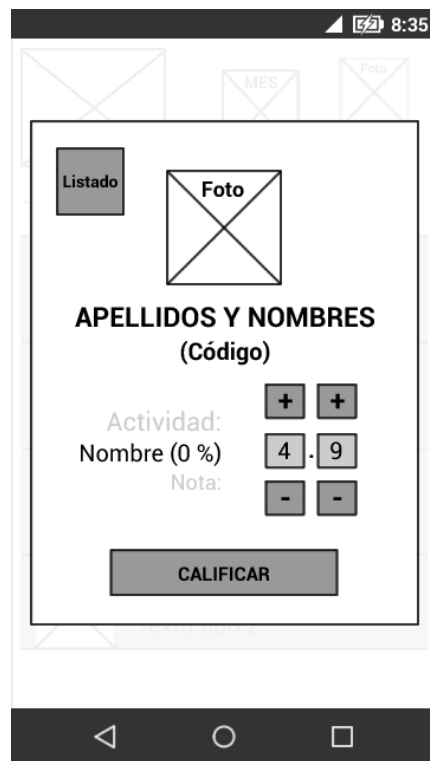
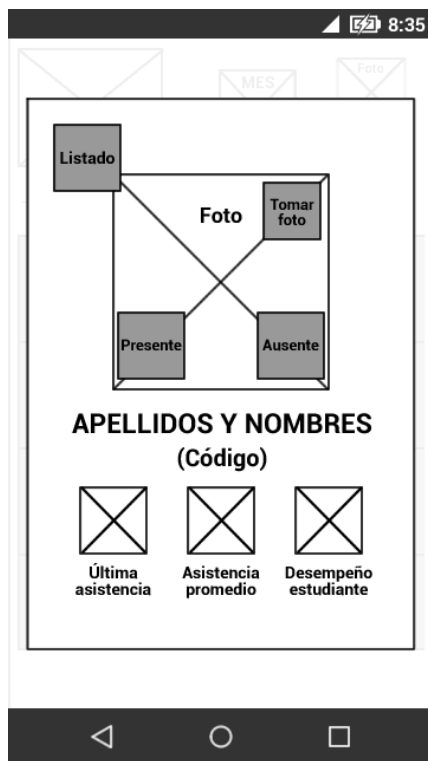
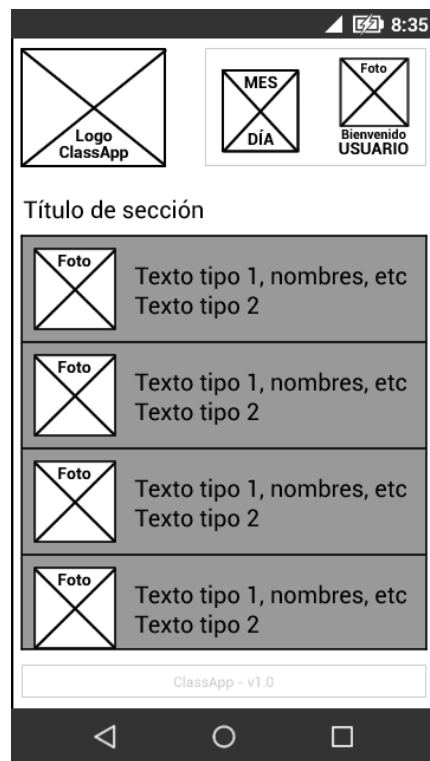
#### 5.4. PROTOTIPOS DE INTERFAZ DE USUARIO

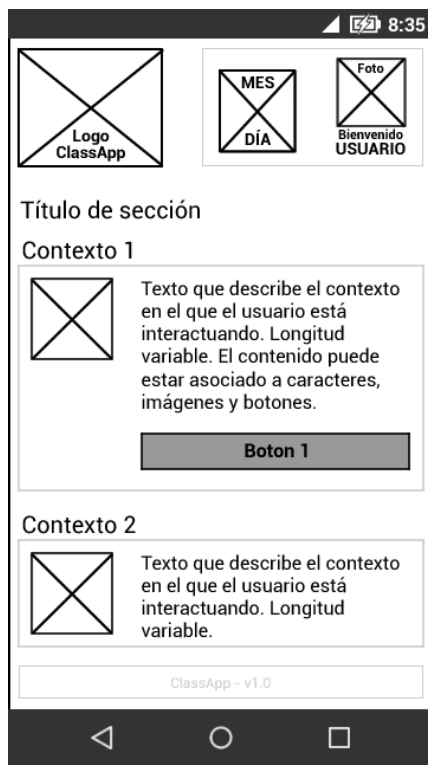
Ya se dispone de suficientes parámetros para ir dando forma a la interfaz gráfica de la aplicación. Se sabe que el usuario es único, que su rol corresponde a la docencia, y se conocen además las funcionalidades que deberá tener ClassApp.

La manera tradicional de abordar esta fase implica elegir una herramienta de prototipado rápido en el ámbito de UX (gráfico). Inicialmente se probó la herramienta Pidoco [47], la cual ofrece un periodo de prueba tras el cual se desactiva la cuenta de usuario y se pierden los wireframe (bocetos) elaborados. Finalmente, ante la poca complejidad de un wireframe, se decide utilizar un editor vectorial de imágenes. Se elige la herramienta Macromedia Fireworks 8 [48] y se procede a elaborar las interfaces gráficas.

Se observa que existen patrones de interfaz de usuario que se repiten varias veces dentro de la aplicación:







## 6. DISEÑO DE LA APLICACIÓN

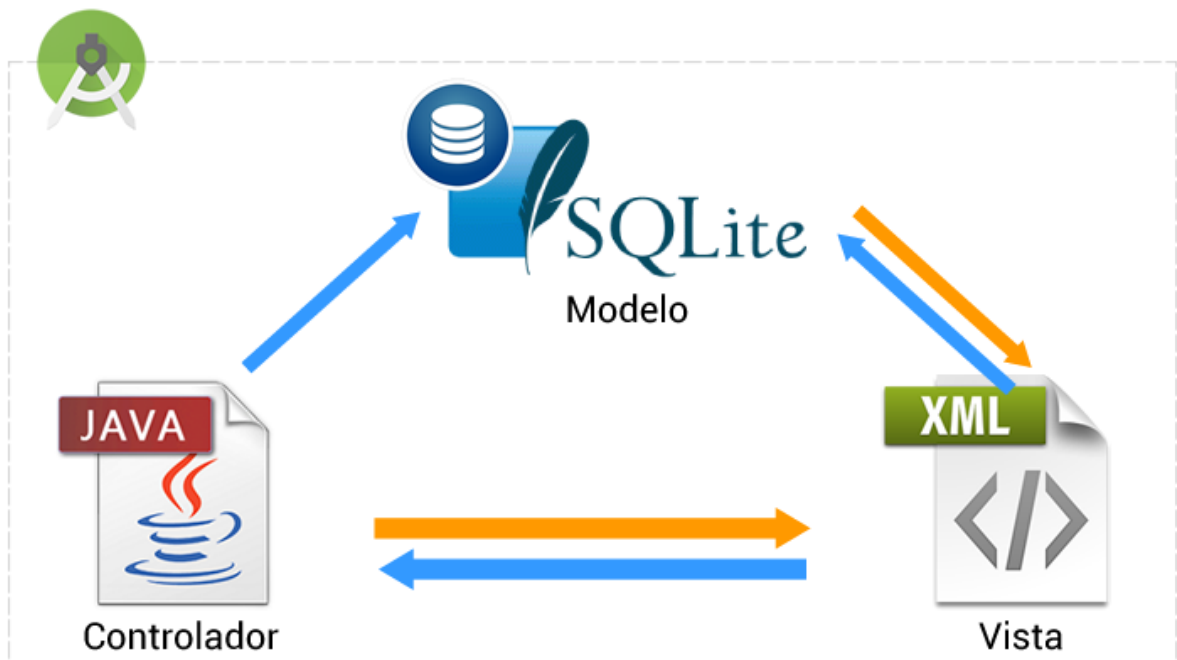
### 6.1. PATRÓN DE DISEÑO: MODELO-VISTA-CONTROLADOR (MVC)

El patrón MVC es una estrategia útil a la hora de garantizar que la aplicación desarrollada sea fácil de mantener, escalable y que sus componentes internos no entren en conflicto.

Su filosofía consiste en separar el ámbito gráfico (interfaz), la representación de los datos (generalmente asociada a la estructura de la base de datos) y las clases de Java en donde se declaran los métodos que controlan la aplicación.

En el ámbito de los desarrollos para dispositivos móviles, Android Studio se caracteriza por hacer un uso intensivo de este patrón de diseño, separando de manera explícita la interfaz (layouts XML), el modelo de negocio (estructura de la base de datos) y los objetos de control (clases de Java).

Figura 2. Patrón MVC en Android Studio.



Para el caso de iOS, al trabajar con XCode también se experimenta esta misma filosofía. La diferencia esencial radica en el lenguaje utilizado para elaborar las clases (Objective-C). Las interfaces se almacenan con sintaxis de formato XML, pero su extensión es “.storyboard” o “.xib”. El motor de base de datos que se utilice dependerá del criterio del desarrollador.

## **6.2. MODELO ENTIDAD-RELACIÓN**

Este modelo muestra con mayor claridad los elementos y las relaciones que intervienen en la lógica de negocio de la aplicación. Las situaciones en donde se presentan relaciones N:M generan tablas intermedias que existirán en la declaración SQLite.

Las entidades compuestas que se generan son: “EstudianteHasTrabajos” y “EstadísticasEstudiantes”, cada una con llaves foráneas apuntando a las clases de donde se extraen sus datos.

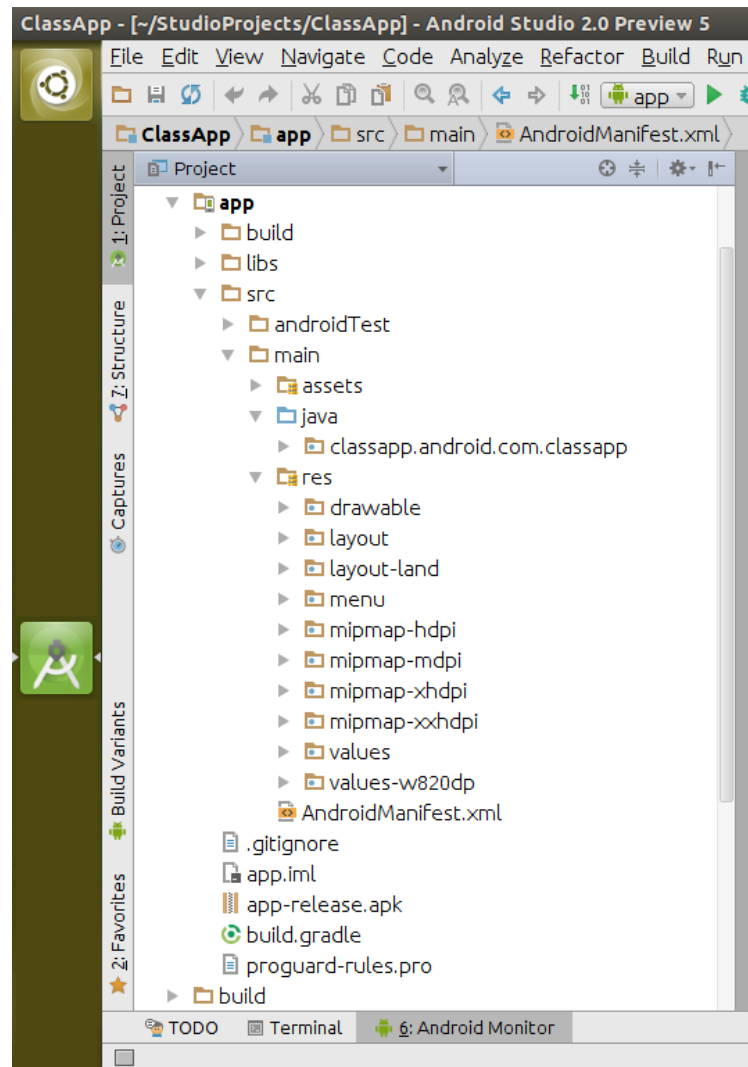




## 6.4. ESTRUCTURA DE PAQUETES (ANDROID STUDIO)

Android Studio organiza los archivos de cada proyecto en diferentes estructuras predefinidas dentro del IDE. Considerando el enfoque de ClassApp y el tipo de archivos que se deben manejar, se decide utilizar la estructura “Project”, la cual distribuye los archivos de la siguiente manera:

Figura 5. Estructura de paquetes en Android Studio.



A continuación se explican las características de las principales carpetas:

<b>Libs</b>	Almacena las librerías externas (.jar) que se deseen usar en la aplicación. Deben ser declaradas también dentro del archivo “build.gradle”.
<b>Assets</b>	<p>Puede contener elementos como fuentes (tipos de letra), imágenes y contenido multimedia. Su desventaja radica en que (salvo las fuentes) los archivos son empaquetados de manera que no pueden ser llamados durante el tiempo de ejecución.</p> <p>Una manera de evadir esta restricción consiste en copiar alguno de los archivos a una carpeta temporal de la aplicación. De esta manera, se puede leer y visualizar en cualquier layout del app.</p>
<b>Java</b>	Es la carpeta en donde se almacenan las clases (controlador del modelo MVC) de la aplicación. Las clases corresponden a archivos “.java” y siguen las reglas tradicionales de programación orientada a objetos.
<b>Drawable</b>	Es ideal para almacenar íconos, imágenes y elementos multimedia que se vayan a utilizar en la aplicación. Esta carpeta no posee restricciones (como la carpeta “assets”) y puede almacenar ilustraciones vectoriales (drawables XML). Hace parte del concepto “vista” del modelo MVC.
<b>Layout</b>	La “vista” del modelo MVC tiene su punto fuerte en la carpeta “Layout”. Aquí se almacenan todas las interfaces gráficas con orientación “portrait” que componen la aplicación como archivos XML. Estos archivos son llamados por cada clase de Java que extienda a la clase Activity o similares.
<b>Layout-land</b>	Es una versión de la carpeta “Layout” dedicada exclusivamente a almacenar archivos XML para interfaces con orientación “landscape” que requieran de una diagramación

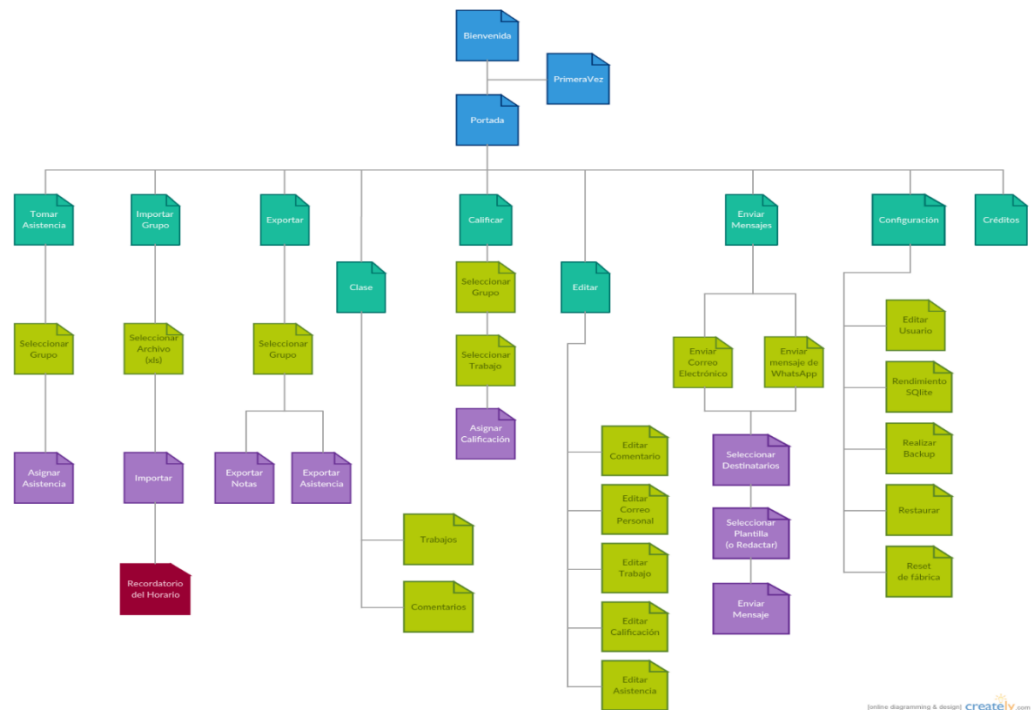
	especial. Estos archivos son llamados automáticamente tan pronto la orientación del dispositivo cambia.
<b>Mipmap-(* )</b>	Las carpetas “mipmap” almacenan los íconos de la aplicación que se muestran en el menú de Android. Vienen asociadas a varias densidades de pixeles y cargan automáticamente los íconos según las características de la pantalla en cada dispositivo.
<b>Values</b>	<p>Contiene archivos XML asociados también al concepto de “vista” del patrón MVC. Entre ellos se encuentran:</p> <p>“<b>string.xml</b>”: define algunos nombres a los cuales se asocian cadenas de texto dentro de la aplicación. Esto facilita el mantenimiento y la corrección de errores, especialmente en cadenas que se repiten en toda la aplicación.</p> <p>“<b>colors.xml</b>”: define qué nombre se le dará a cierto color que existe en formato hexadecimal, lo cual facilita su utilización en los layout y las clases.</p> <p>“<b>style.xml</b>”: define los diferentes parámetros de apariencia que puede tener un objeto gráfico en la aplicación. Es el equivalente a los estilos CSS en el ámbito de desarrollo web.</p> <p>“<b>dimen.xml</b>”: define ciertas cifras asociadas a tamaños, indicando a qué unidades de medida corresponden (px, dp, sp).</p>
<b>AndroidManifest.xml</b>	Es un archivo XML en donde se declara: qué clases de Java están asociadas a un layout, en qué clase inicia la aplicación, si la aplicación puede recibir notificaciones directas a una actividad (layout), permisos especiales (se puede ver en la sección “9. Manifiesto De Permisos Del Sistema Operativo

	Android”), nombres asociados a cada clase, entre otros elementos.
<b>Build.gradle</b>	Este archivo contiene información como: versión de la aplicación, alias de la versión, sdk mínimo y máximo que se soporta, librerías declaradas para el uso en la aplicación, y algunas directivas de compilación.

## 6.5. DIAGRAMA DE NAVEGACIÓN

En esta fase del proyecto ya es posible definir el mapa o diagrama de navegación de la aplicación. Como en todo programa, debe existir una interfaz inicial desde la cual se pueda ingresar a cualquier sección y funcionalidad de ClassApp. Los elementos que no tienen ítems internos, son aquellos en donde no existe posibilidad de navegación. Se codifican los elementos por colores para indicar una jerarquía de navegación y facilitar la comprensión del diagrama. Este gráfico fue elaborado con la herramienta Creately [49].

Figura 6. Diagrama de navegación, ClassApp.



## 7. METODOLOGÍA

### 7.1. PROTOTIPADO EVOLUTIVO (MODELO DE PROTOTIPOS)

Dentro de la filosofía del modelo en espiral, se define un desarrollo de prototipos (por fases), al que se denomina “prototipado evolutivo”, donde se desarrolla el concepto principal del software a medida que se avanza. El prototipado se inicia desarrollando los aspectos más importantes del sistema, los cuales se presentan al cliente y según la retroalimentación que se reciba, se realizan cambios y se continúa con el desarrollo del prototipo. Todos los modelos de desarrollo implican la existencia de un inicio y de una finalización de actividades. En este caso, el modelo finaliza cuando los desarrolladores y el cliente acuerdan que el prototipo satisface las necesidades y se considera “bueno”.

En esas condiciones, se mejoran los detalles que sean necesarios y se entrega este nuevo prototipo como producto final; dependiendo el tipo de empresa, en muchos casos ocurre que el prototipo sea “operativo”, es decir, que pueda ser utilizado por el cliente en condiciones reales [50].

Para aquellos entornos en donde los requerimientos van cambiando con rapidez, es recomendable utilizar esta metodología. Igualmente, se recomienda su utilización cuando el usuario no logra describir con claridad los requerimientos. Incluso cuando los desarrolladores no tienen claridad sobre aspectos como la arquitectura del sistema o los algoritmos que se van a utilizar, también es recomendable utilizar esta metodología [51].

En general, el prototipado evolutivo muestra signos visibles de progreso en el desarrollo, lo cual es favorable en entornos donde existe una gran presión en los tiempos de entrega del software. Además es menos estricto que otros métodos [52], ya que si el cliente considera que el software se encuentra incompleto, en las siguientes fases se irán realizando las correcciones necesarias hasta obtener el prototipo final.

El prototipado evolutivo, sin embargo, presenta algunas desventajas. Debido a la incertidumbre que maneja el prototipado, no es posible conocer desde el inicio cuánto tomará crear un producto aceptable. Esto genera una tendencia a elaborar software sin

prestar suficiente atención a los requerimientos y detalles del sistema que se está elaborando. En resumen, el prototipado evolutivo es ideal para proyectos cuyos requerimientos, estructura, algoritmos y arquitectura son propensos a cambiar durante su ciclo de vida.

Cada fase del prototipado evolutivo contiene las siguientes etapas [53]:

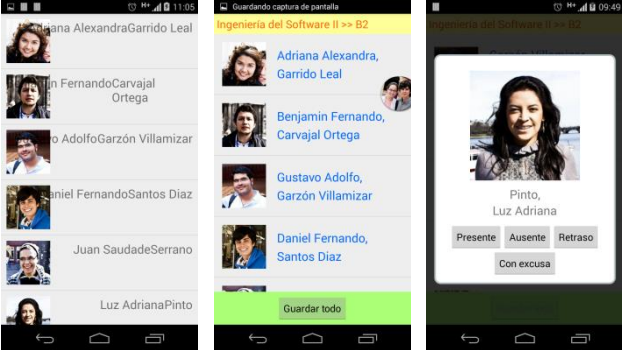
- 1) Planeación y análisis
- 2) Diseño rápido
- 3) Construcción de un prototipo
- 4) Evaluación del prototipo.

## 7.2. FASES DE PROTOTIPADO

Para el desarrollo de ClassApp se plantea un prototipado de tres fases, en las cuales se revisa la lógica de la aplicación, la interfaz gráfica, los elementos de control, seguridad de la aplicación y algunas pruebas sencillas de las funcionalidades que se van concretando.

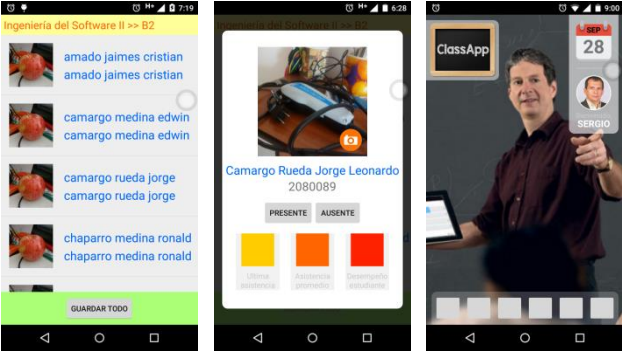
### FASE 1: Primera entrega

<b>1) Planeación y análisis</b>	Inicialmente se realizaron reuniones periódicas para identificar el problema, definir los requisitos iniciales y revisar qué tan viable era elaborar un cronograma aproximado de trabajo. En ese momento, el autor del proyecto se encontraba aprendiendo a trabajar con el lenguaje Java y revisando la arquitectura por capas de Android. Esto se vio reflejado en una demora inicial y múltiples inconvenientes en la elaboración del primer prototipo.
---------------------------------	---

<p><b>2) Diseño rápido</b></p>	<p>Partiendo de los requisitos iniciales fue posible proponer un diseño inicial de clases y entidades. Algunos factores (como discutir el crecimiento modular de la aplicación) fueron aplazados mientras el autor del proyecto ganaba una mejor comprensión del lenguaje y sus alcances.</p> <p>Inicialmente el módulo de asistencia fue llenado con datos estáticos de estudiantes desde un array. Elegir un motor de bases de datos y la adaptación al modelo de datos de SQLite, fueron dos motivos que atrasaron el llenado dinámico del componente ListView.</p>
<p><b>3) Construcción de un prototipo</b></p>	<p>El primer prototipo consistió en una serie de Layouts rudimentarios sin ningún tipo de diseño gráfico. Los elementos funcionales iniciales se centraron en hacer funcionar algunos botones y demostrar la implementación del componente ListView para el módulo de toma de asistencia.</p> 
<p><b>4) Evaluación del prototipo</b></p>	<p>El prototipo inicial se revisó en términos de usabilidad, cumplimiento de requisitos hasta ese momento y velocidad de ejecución.</p>

## FASE 2: Segunda entrega

<b>1) Planeación y análisis</b>	La segunda fase tuvo un proceso mucho más ágil debido a que se logró un dominio suficiente del lenguaje Java y de las características de Android Studio. Los requerimientos fueron aprobados de manera definitiva y se definió qué tipo de librerías se necesitaban para leer o generar archivos de Excel, crear gráficos estadísticos y abrir un explorador visual de archivos.
<b>2) Diseño rápido</b>	Se lograron cargar datos dinámicamente desde SQLite y se procedió a crear la estructura total de la base de datos. Se logró acceder a la cámara del dispositivo, manipular fotografías y guardar archivos en ciertas carpetas. Se logró también cargar los ImageView dinámicamente, y se conocieron mejor las características de los diferentes layout (Linear, Frame, Relative).
<b>3) Construcción de un prototipo</b>	Para esta fase se consideró aplicar los primeros avances de diseño gráfico, especialmente a la portada de la aplicación. Se propuso utilizar en la portada una serie de imágenes, pero se afectó el rendimiento de la aplicación. Finalmente se decidió que la portada tuviera grupos de botones tradicionales, el logotipo de ClassApp un calendario y la fotografía del docente.

	
<p><b>4) Evaluación del prototipo</b></p>	<p>Los esfuerzos se centraron en evaluar aspectos funcionales, revisar que se generaran los archivos exportados, que se almacenaran las imágenes correctamente, que los gráficos estadísticos funcionaran y que los valores estadísticos y notas fueran coherentes.</p>

### FASE 3: Entrega final

<p><b>1) Planeación y análisis</b></p>	<p>La aplicación se llevó a Google Play y se creó el manual de usuario en un blog.</p>
<p><b>2) Diseño rápido</b></p>	<p>Se añadieron funcionalidades de evaluación de desempeño para SQLite, se eliminaron los objetos sobrantes de las pruebas anteriores y se trabajó en reducir el tamaño físico de la aplicación.</p>
<p><b>3) Construcción de un prototipo</b></p>	<p>Se rediseñaron los layout de toma de asistencia y calificaciones para mejorar la rapidez, usabilidad y organización. Se crearon los íconos definitivos para la aplicación y se eliminaron los layout sobrantes de las pruebas iniciales.</p>

<p><b>4) Evaluación del prototipo</b></p>	<p>Se revisó el ámbito funcional y visual por igual, se realizaron pruebas de carga en dos dispositivos reales y múltiples máquinas virtuales en modo Tablet y smartphone. Se utilizaron también diferentes API Level para verificar posibles excepciones.</p>

## 8. DESCRIPCIÓN DEL SISTEMA

### 8.1. REQUERIMIENTOS DE HARDWARE

ClassApp se planteó desde el principio como una aplicación orientada a dispositivos móviles, lo cual indica que las prestaciones de las máquinas en donde se ejecutará son reducidas. Explorando sus funcionalidades, ClassApp requiere que el dispositivo cuente con: una cámara principal (no se requiere una resolución en especial) y un medio de almacenamiento extraíble (sdcard o similar). También es necesario el hardware para mantener una conexión a internet, ya sea por redes de datos 3G o por WiFi.

De manera experimental se observó que es suficiente con que los dispositivos cuenten con al menos 256 MB de memoria RAM, y al menos 1 GHz de capacidad de procesamiento en CPU. Las prestaciones en GPU son complementarias.

### 8.2. DEPENDENCIAS

En esencia, ClassApp requiere que el dispositivo móvil ejecute una versión de Android igual o superior a 2.3 (API Level 10 o superior). Si bien es una restricción radical, ClassApp se podrá ejecutar en un 99.7% de los dispositivos Android del mundo.

Versión	Nombre	API	Distribución (%)	
2.2	Froyo	8	0.2 %	
2.3.3 - 2.3.7	Gingerbread	10	3.0 %	99.7 %
4.0.3 - 4.0.4	Ice Cream Sandwich	15	2.7 %	
4.1.x	Jelly Bean	16	9.0 %	

4.2.x		17	12.2 %	
4.3		18	3.5 %	
4.4	KitKat	19	36.1 %	
5.0	Lollipop	21	16.9 %	
5.1		22	15.7 %	
6.0	Marshmallow	23	0.7 %	

Información recolectada por siete días, hasta el 4 de enero de 2016.

Tomado de [54]

Una de las razones para tomar como requisito mínimo el API Level 10, es la compatibilidad con clases de SQLite que son indispensables para el funcionamiento de ClassApp. Especialmente para los métodos de pruebas de rendimiento transaccional.

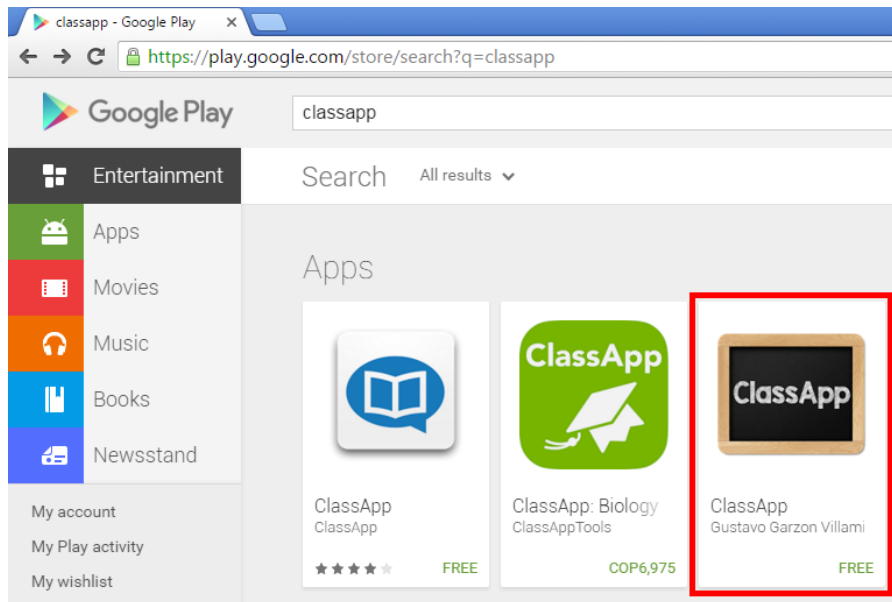
### 8.3. DESCRIPCIÓN DEL SOFTWARE

**8.3.1. Roles y usuarios.** En ClassApp se realiza un planteamiento que la define como aplicación de usuario único. La justificación radica en dos razones: el tipo de actividades que se ejecutan sobre la aplicación (docencia y gestión de clases) y la naturaleza personal/privada de los dispositivos móviles.

La información que se maneja en ClassApp no compromete la privacidad del docente, son simplemente datos y cifras de desempeño estudiantil.

Los dispositivos móviles se han convertido en un accesorio privado, tal como una billetera o un documento personal de identificación. Por ese motivo, no es frecuente que las personas presten su smartphone o tablet a terceros y, por ende, no es crítico que se implemente un proceso de autenticación por usuario y contraseña.

Para descargar la aplicación se debe ingresar a la tienda Google Play ( <https://play.google.com> ), y en el campo de texto superior se ingresa como palabra clave “classapp”. Finalmente, se selecciona la opción que se muestra en la imagen:



**8.3.2. Funcionalidades.** Existen nueve íconos en la portada, los cuales agrupan las funcionalidades de ClassApp.

En total, y complementando la información de los capítulos anteriores, se tienen:

Funcionalidad	Descripción
<b>Portada</b>	<p>Al iniciar ClassApp se muestran nueve íconos que permiten acceder a todas sus funcionalidades:</p> <div data-bbox="753 1686 1456 1810" style="border: 1px solid black; padding: 5px; text-align: center;"> </div>

	<table border="1" data-bbox="753 239 1455 363"> <tr> <td>“Tomar asistencia”</td> <td>“Importar grupo”</td> <td>“Exportar”</td> <td>“Clase”</td> </tr> </table> <table border="1" data-bbox="753 415 1455 539"> <tr> <td></td> <td></td> <td></td> <td></td> </tr> </table> <table border="1" data-bbox="753 539 1455 663"> <tr> <td>“Calificar trabajos”</td> <td>“Editar”</td> <td>“Enviar mensajes”</td> <td>“Configurar”</td> </tr> </table> <table border="1" data-bbox="753 716 928 915"> <tr> <td></td> </tr> <tr> <td>“Créditos”</td> </tr> </table> <p data-bbox="753 968 1455 1150">La portada y la toma de asistencia son dos layouts experimentales de este prototipo. En ellos se calcula dinámicamente el tamaño de los íconos para cada tamaño de pantalla.</p>	“Tomar asistencia”	“Importar grupo”	“Exportar”	“Clase”					“Calificar trabajos”	“Editar”	“Enviar mensajes”	“Configurar”		“Créditos”
“Tomar asistencia”	“Importar grupo”	“Exportar”	“Clase”												
															
“Calificar trabajos”	“Editar”	“Enviar mensajes”	“Configurar”												
															
“Créditos”															
<p data-bbox="318 1486 591 1520"><b>1) Tomar asistencia</b></p>	<table border="1" data-bbox="753 1251 1455 1520"> <thead> <tr> <th colspan="4">Íconos</th> </tr> </thead> <tbody> <tr> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td>“Presente”</td> <td>“Ausente”</td> <td>“Fotografía”</td> <td>“Ver listado”</td> </tr> </tbody> </table> <p data-bbox="753 1577 1455 1808">Esta funcionalidad permite asignar valores de asistencia a cada estudiante. Los posibles valores iniciales son: “presente” y “ausente”, para una toma de asistencia ágil. El docente puede editarlos posteriormente con valores adicionales: “retraso” y</p>	Íconos								“Presente”	“Ausente”	“Fotografía”	“Ver listado”		
Íconos															
															
“Presente”	“Ausente”	“Fotografía”	“Ver listado”												

	<p>“con excusa”. Estos valores se codifican con enteros 1, 2, 3 y 4 al ser almacenados en la base de datos.</p> <p>Al ingresar, se pide al usuario que seleccione un grupo. Luego se abre por omisión una pantalla en donde se muestra al primer estudiante de la lista y se encuentran botones para “presente”, “ausente”, “fotografía” y “ver listado”.</p> <p>Al dar click a “presente” o “ausente”, se carga el siguiente estudiante según el orden de lista.</p> <p>Si se desea cambiar un valor de asistencia, se puede dar click al botón “ver listado” y buscar el estudiante.</p> <p>Al dar click al botón “fotografía” se envía un intent a la cámara del dispositivo, se adquiere la fotografía y se muestra una interfaz para que el docente recorte la foto con el rostro del estudiante en el centro.</p> <p>Al final, se indica al docente que ha finalizado la toma de lista y que debe guardarla para continuar.</p>
<p><b>2) Importar grupo</b></p>	<p>Permite crear grupos de estudiantes desde un archivo de Microsoft Excel (.xls). Este archivo deberá tener un nombre de archivo determinado y una cierta estructura interna (puede verse en el Anexo B). Además podrá contener un número ilimitado de estudiantes.</p> <p>Al ingresar, se solicita al usuario buscar y elegir el archivo, y se apunta por omisión a la carpeta “Downloads” del dispositivo. Se solicita también la cédula, correo institucional y correo personal para poder continuar con el proceso.</p> <p>Se oprime el botón “leer y guardar”, los estudiantes son cargados y el grupo es creado automáticamente (junto con otros datos que se extraen del archivo). Se ofrece la opción de crear un recordatorio para las sesiones</p>

	<p>docentes con ese grupo, o finalizar el proceso hasta ese punto.</p> <p>* En este momento, cuando existe al menos un grupo, es posible utilizar las demás funciones de ClassApp.</p>
<p><b>3) Recordatorios</b></p>	<p>Esta funcionalidad permite activar alarmas de Android tan pronto se aproxime una sesión docente.</p> <p>Al ser redirigido desde la funcionalidad “importar grupo”, al docente se le ofrecen dos opciones: la asignatura se dicta “en horas diferentes y días diferentes” ó “a la misma hora en días diferentes”.</p> <p>Posteriormente se le solicita que ingrese la hora en que inicia la clase y al presionar el botón de confirmación, se redirige a la aplicación AlarmClock de Android para seleccionar los días en que se repite cada sesión. La misma secuencia se sigue para la primera opción, hasta que el usuario presione el botón “Finalizar, ¡eso era todo!”.</p>
<p><b>4) Exportar (.xls)</b></p>	<p>Permite generar archivos de Microsoft Excel (.xls) con información de asistencia o notas de cada grupo.</p> <p>Al ingresar, se pregunta al usuario si desea generar un archivo de asistencia o notas. Luego se le pide que seleccione un grupo, luego se debe confirmar la orden de “exportar” y se le muestra al usuario el nombre de archivo y la ubicación (carpeta “Downloads”). Se da la opción de enviar el archivo generado por correo electrónico. En dado caso, se envía un intent que lista los clientes de correo del dispositivo, y se regresa automáticamente a ClassApp tan pronto se haya enviado el mensaje.</p>

<p><b>5) Crear trabajos</b></p>	<p>Esta funcionalidad permite crear trabajos, exámenes o actividades en general, asignándoles un porcentaje y validando que, máximo, se llegue a un 100% como total.</p> <p>Al ingresar, se pide que se seleccione un grupo, y luego se solicita una descripción y un porcentaje para el trabajo. Si se presiona el botón “Guardar”, la parte inferior de la pantalla se actualiza y va mostrando un gráfico circular con cada trabajo nuevo. Tan pronto se completa un 100%, ClassApp no permite que se ingresen más trabajos para ese grupo.</p>
<p><b>6) Calificar trabajos</b></p>	<p>Permite asignar una nota cuantitativa (de 0.0 a 5.0) a los trabajos asociados a los estudiantes de un grupo.</p> <p>Al ingresar, se pide que se seleccione un grupo y luego que se seleccione uno de los trabajos. Se muestra entonces una pantalla con la misma estructura de la toma de asistencia, iniciando con el primer estudiante en orden de lista, y ofreciendo una interfaz para asignación de notas:</p> <div data-bbox="915 1276 1295 1486" data-label="Image"> </div> <p>Al final, se indica al docente que se ha terminado de calificar estudiantes y que debe guardar los cambios para continuar.</p>
<p><b>7) Crear comentarios</b></p>	<p>Esta funcionalidad permite crear un comentario que se asocia a una fecha, hora y grupo en particular. Los</p>

	<p>comentarios de ClassApp pueden utilizarse como agenda personal o incluso como temario o tabla de contenido para cada grupo.</p> <p>Al ingresar, se solicita al usuario que seleccione un grupo para asignarle el comentario. Luego se le pide que verifique los metadatos del comentario (grupo, fecha y hora) y que ingrese el texto del comentario. Se oprime el botón “guardar comentario”, el comentario es almacenado y se le da la opción al docente de revisar los comentarios que existen, o de regresar a la portada.</p>
<b>8) Ver comentarios</b>	<p>Esta funcionalidad permite visualizar un listado de comentarios según el grupo que se seleccione.</p> <p>Al ingresar, se solicita al usuario que seleccione un grupo, se listan los comentarios asociados a ese grupo y el docente puede visualizarlos haciendo click sobre cada uno. Al hacerlo, se mostrará una pantalla con algunos metadatos (fecha y hora) y el texto del comentario.</p>
<b>9) Editar</b>	<p>Permite cambiar los valores de algunos datos ingresados por el docente. Es posible editar: comentarios, campos de correo personal, trabajos, calificaciones y tomas de asistencia.</p> <p>Al ingresar, se presenta al usuario una serie de botones para editar cada opción. Estas realizarán redirecciones a las interfaces que permiten cambiar el contenido de esos datos.</p>
<b>10) Enviar correo electrónico</b>	<p>Esta funcionalidad permite la comunicación con los estudiantes en momentos diferentes a los de la sesión docente, particularmente utilizando mensajes de correo electrónico. Para una mayor efectividad de este</p>

	<p>método, se recomienda obtener los correos electrónicos personales de cada estudiante y así garantizar que los estudiantes sean notificados ante cualquier aviso.</p> <p>Al ingresar, se le solicita al docente que seleccione los destinatarios (un grupo, una asignatura, un estudiante o un destinatario personalizado). Luego, se da la opción de seleccionar una plantilla para el mensaje o de redactar un mensaje desde cero. ClassApp muestra cualquiera de las opciones que se hayan seleccionado y brinda la oportunidad de seguir modificando el mensaje. Se presiona “¡enviar mensaje!” y aparece un diálogo que permite elegir el cliente de correo que se desea utilizar.</p> <p>Después de enviar el mensaje, se redirige al docente automáticamente a ClassApp y se le notifica que el mensaje fue puesto en manos del cliente de correo.</p> <p><small>*En estos casos, la disponibilidad de una red WiFi/3G o incluso los fallos del cliente de correo, no son responsabilidad de ClassApp.</small></p>
<p><b>11) Enviar mensajes de WhatsApp</b></p>	<p>Permite la comunicación con los estudiantes en momentos diferentes a los de la sesión docente, particularmente utilizando mensajes de WhatsApp.</p> <p>Al ingresar, se le solicita al docente que seleccione una plantilla para el mensaje o que redacte un mensaje desde cero. ClassApp muestra cualquiera de las opciones que se hayan seleccionado y brinda la oportunidad de seguir modificando el mensaje. Se presiona “¡enviar mensaje!” y el docente es redirigido a WhatsApp. Estando allí, se seleccionarán los miembros del grupo (para mayor agilidad, se</p>

	<p>recomienda que exista previamente un chat de grupo con los integrantes) y se oprime el botón de envío.</p> <p>Después de enviar el mensaje, el docente deberá utilizar el botón “atrás” de Android para regresar a ClassApp. Al regresar, se le notificará que el mensaje fue puesto en manos de la aplicación WhatsApp.</p> <p>*En estos casos, la disponibilidad de una red WiFi/3G o que WhatsApp no se encuentre instalado, no son responsabilidad de ClassApp.</p>
<p><b>12) Rendimiento SQLite</b></p>	<p>Permite evaluar el desempeño del motor de base de datos SQLite utilizando tres métodos de inserción de registros:</p> <ul style="list-style-type: none"> <li>- “CompileStatement”</li> <li>- “RawSQL”</li> <li>- “Transaction”.</li> </ul> <p>Al ingresar, se ofrecen tres botones para ejecutar cada uno de estos métodos, y se da la opción de generar un archivo de Microsoft Excel (.xls) con los resultados de las tres ejecuciones.</p> <p>Los resultados de cada método se muestran con la información experimental (N, N<sub>95%</sub>, tiempo promedio, tiempo promedio real y cantidad de registros) y gráficos estadísticos de: tiempos empleados (previamente ordenados), eliminación de picos de procesamiento (aceptación del 95% de los datos), y un histograma de frecuencias para validar el tiempo promedio.</p>
<p><b>13) Backups SQLite</b></p>	<p>Esta funcionalidad ayuda a generar un archivo “.sqlite” que contiene todos los registros y la estructura actual de la base de datos de ClassApp.</p>

	<p>Al ingresar, se pide al usuario que confirme si desea generar un “backup (copia de respaldo)”. Al confirmar, se indica la ubicación y nombre del archivo “.sqlite” y se ofrece la opción de regresar a la portada.</p>
<b>14) Restaurar desde backup</b>	<p>Permite tomar el archivo “.sqlite” generado por la funcionalidad de backups, leer sus contenidos y crear una base de datos sqlite de manera manual, con los mismos contenidos del backup.</p> <p>Al ingresar, se solicita buscar el archivo “.sqlite” del backup, se confirma la orden de restaurar y ClassApp se reinicia automáticamente con los datos recuperados.</p>
<b>15) Reset de fábrica</b>	<p>Esta funcionalidad permite eliminar completamente el contenido de la base de datos SQLite, para los casos en los que el docente lo crea conveniente o para pruebas futuras de los estudiantes que deseen mejorar este proyecto.</p> <p>Al ingresar, se muestra un solo botón y una advertencia de pérdida total de datos.</p>
<b>16) Créditos</b>	<p>En esta sección se indica que ClassApp es producto de un trabajo de grado para optar al título de Ingeniero de Sistemas, se muestra la versión oficial, se nombra a los autores, el director de proyecto, a la Universidad Industrial de Santander, ciudad y año de elaboración.</p>

## 9. MANIFIESTO DE PERMISOS DEL SISTEMA OPERATIVO ANDROID

Todo proyecto de desarrollo en Android contiene, además de los archivos esenciales del patrón modelo (base de datos), vista (layouts) y controlador (clases de Java), un archivo particular llamado "AndroidManifest.xml". Está ubicado en la carpeta raíz de la aplicación y la información que contiene es esencial para que los desarrollos puedan ser ejecutados por Android. Entre otros propósitos, el manifiesto de permisos sirve para:

- Nombrar el paquete de Java asociado a la aplicación para diferenciarla de otras apps y en general, para poder identificarla.
- Describir los componentes (actividades, services, broadcast receivers y content providers) que definen a la aplicación. También se nombran las clases y funcionalidades que implementan cada componente. Todo eso permite que Android conozca qué elementos corresponden a un componente y bajo qué condiciones se pueden ejecutar.
- Define qué procesos están asociados a los componentes de la aplicación.
- Declara qué permisos debe tener la aplicación para poder utilizar elementos de hardware (normalmente se encuentran protegidos dentro del API) o para comunicarse con otras aplicaciones de Android. También define qué permisos deben tener las demás aplicaciones para poder interactuar con los módulos desarrollados.
- Define la versión mínima del API de Android que requiere la aplicación.
- Muestra las librerías de Android que sean necesarias para ejecutar la aplicación.

Como se indica en los objetivos del proyecto, se hará énfasis en caracterizar los permisos asociados al manifiesto, indicando la importancia de cada uno y sus implicaciones para el desarrollador.

Los permisos se declaran con la siguiente sintaxis:

```

<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="classapp.android.com.classapp">

    <uses-permission android:name="android.permission.INTERNET" />
    <uses-permission android:name="android.permission.STORAGE" />
    <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
    <uses-permission android:name="android.permission.CAMERA" />

    <application...>

```

Los permisos utilizados en ClassApp se indican en la tabla siguiente con doble asterisco (\*\*).

Se puede observar que los permisos se identifican mediante constantes tipo String y además no tienen un orden secuencial. Existen alrededor de 127 permisos, pero las aplicaciones frecuentemente usan menos de 10 de ellos. A continuación se caracterizan las constantes más relevantes para las actividades de desarrollo:

android.permission	Características
ACCESS_CHECKIN_PROPERTIES	Permite el acceso de lectura y escritura a la tabla "properties" en la base de datos de Check-In. La idea principal es poder editar valores almacenados por aplicaciones georeferenciadas.
ACCESS_COARSE_LOCATION	Permite ubicar el dispositivo de manera aproximada mediante Cell-ID o un router WiFi al que se esté conectado. Consume pocos recursos comparado con el uso de GPS.
ACCESS_FINE_LOCATION	Permite ubicar el dispositivo con una alta precisión, generalmente utilizando el servicio de GPS del operador de telefonía móvil. Consume una cantidad considerable de recursos de cómputo.

ACCESS_LOCATION_EXTRA_COMMANDS	Permite interactuar con el proveedor de servicios de ubicación, especialmente enviando mensajes complementarios al GPS.
ACCESS_NETWORK_STATE	Permite tener acceso a estadísticas sobre el uso de red (WiFi, 3G, 4G). Se usa frecuentemente con el permiso "INTERNET" aunque no existen dependencias entre los dos.
ACCESS_WIFI_STATE	Permite obtener información sobre las redes WiFi cercanas y algunas opciones de conectividad. Ofrece ciertas características que pueden ser útiles para los desarrolladores.
ACCOUNT_MANAGER	Permite que la aplicación utilice los métodos de "AccountAuthenticators" para automatizar el acceso a apps que requieran logins o credenciales especiales.
ADD_VOICEMAIL	Permite la gestión de la funcionalidad "correo de voz" (voicemail).
BATTERY_STATS	Permite acceder a las estadísticas de la batería del dispositivo, es útil cuando se desea planificar el consumo eléctrico u ofrecer consejos al usuario para mejorar el desempeño del Smartphone.
BIND_INPUT_METHOD	Permite la gestión de interacciones con el "InputMethodService", especialmente en aplicaciones que manejan teclados en pantalla en varios idiomas.
BLUETOOTH	Permite que la aplicación pueda utilizar el hardware para conexiones Bluetooth y llevar a cabo acciones como: solicitar una conexión, aceptar una conexión y transferir archivos.

BLUETOOTH_ADMIN	<p>Permite iniciar funciones de exploración de red Bluetooth para descubrir dispositivos y acceder a la configuración de Bluetooth.</p> <p>Nota: debe usarse al mismo tiempo con el permiso "BLUETOOTH".</p>
BROADCAST_PACKAGE_REMOVED	<p>Permite que la aplicación pueda recibir mensajes tan pronto se desinstala un paquete asociado a la aplicación (u otra aplicación diferente). Suele utilizarse para eliminar archivos generados por la aplicación desarrollada.</p>
BROADCAST_SMS	<p>Permite recibir notificaciones tan pronto se recibe un mensaje SMS. Este permiso está reservado para aplicaciones internas de Android (system key signed).</p>
BROADCAST_STICKY	<p>Permite capturar mensajes de notificación de Android (Broadcasts) incluso cuando las actividades de la aplicación se encuentran en pausa o saturadas.</p> <p>Se utiliza en apps que entran en pausa bajo ciertas circunstancias, y no se desea que se pierdan mensajes como "batería baja", "modo de ahorro de energía", "pantalla apagada", entre otros.</p>
CALL_PHONE	<p>Permite iniciar el proceso de llamada telefónica sin tener que marcar el número con la interfaz gráfica (Dialer user interface) ni confirmar la llamada. Debe ser utilizado con precaución ya que tiene el potencial de realizar llamadas a destinos internacionales e incurrir en costos indeseados para el usuario.</p>
CALL_PRIVILEGED	<p>Permite iniciar el proceso de llamada telefónica, incluso a números de emergencia, de manera automática. Debe usarse con precaución ya que tiene el potencial de realizar llamadas a destinos internacionales e incurrir en costos indeseados para el usuario.</p>

CAMERA**	Permite acceder a todas las funciones de las dos cámaras del dispositivo. Se considera peligroso porque puede activar la cámara sin la autorización explícita del usuario, y enviar información que podría violar la privacidad del propietario del dispositivo.
CHANGE_COMPONENT_ENABLED_STATE	Permite que la aplicación cambie el estado de un componente (incluso de otra aplicación). Este permiso está reservado para aplicaciones internas de Android (system key signed).
CHANGE_CONFIGURATION	Permite modificar algunos elementos de la configuración actual de Android, sin embargo, recientemente se realizaron cambios en el perfil de seguridad y se considera "protection level: system". Este permiso está reservado para aplicaciones internas de Android (system key signed).
CHANGE_NETWORK_STATE	Permite realizar cambios en la conectividad de las redes del dispositivo. Suele utilizarse para ahorrar algunos taps (clicks) al usuario a la hora de gestionar redes WiFi y 3G.
CHANGE_WIFI_STATE	Permite realizar cambios en la conectividad de las redes WiFi. Suele utilizarse para ahorrar algunos taps (clicks) al usuario a la hora de gestionar redes WiFi.
CLEAR_APP_CACHE	Permite que la aplicación pueda limpiar el caché (archivos temporales) de otras aplicaciones. Este permiso está reservado para aplicaciones internas de Android (system key signed).
CONTROL_LOCATION_UPDATES	Permite habilitar o deshabilitar las notificaciones de cambios en la ubicación geográfica. Este permiso está reservado para aplicaciones internas de Android (system key signed).

DELETE_CACHE_FILES	Permite que la aplicación pueda eliminar completamente el caché (archivos temporales) de otras aplicaciones. Este permiso está reservado para aplicaciones internas de Android (system key signed).
DELETE_PACKAGES	Permite desinstalar paquetes (aplicaciones) de Android directamente. Este permiso está reservado para aplicaciones internas de Android (system key signed).
DISABLE_KEYGUARD	Permite deshabilitar la funcionalidad de bloqueo de pantalla (slider, código o trazado) para que el usuario no tenga que ingresar alguno de estos mecanismos de desbloqueo.
EXPAND_STATUS_BAR	Permite manipular la barra de estado de Android (barra en donde aparece la hora, estado de batería y WiFi, señal, etc). Las posibles opciones son: expandir o colapsar.
FLASHLIGHT	Permite acceder y manipular las funcionalidades de la linterna del dispositivo.
GET_PACKAGE_SIZE	Permite conocer el espacio físico que ocupa cualquier aplicación.
INSTALL_PACKAGES	Permite instalar paquetes (aplicaciones) de Android directamente. Este permiso está reservado para aplicaciones internas de Android (system key signed).
INTERNET**	Permite tener acceso a internet y/o manipular ciertos sockets. Es uno de los permisos más utilizados debido a que otorga un gran potencial de conectividad.
KILL_BACKGROUND_PROCESSES	Permite forzar el cierre de aplicaciones que se encuentren ejecutándose en segundo plano. Se usa cuando se desea cerrar procesos con el método "killBackgroundProcesses()".

LOCATION_HARDWARE	Permite utilizar el hardware de geolocalización, así como sus métodos asociados. Es utilizado por aplicaciones georeferenciadas y cualquier módulo que desee acceder a la ubicación actual del dispositivo.
MANAGE_DOCUMENTS	Permite acceder a las funcionalidades de gestión de archivos para aplicaciones tipo “document picker” (explorador de archivos).
MODIFY_AUDIO_SETTINGS	Permite configurar las opciones de audio del dispositivo, volumen global y parcial, gestión de ringtones, volumen de los audífonos, entre otras.
MODIFY_PHONE_STATE	Permite acceder al estado de funciones de telefonía, pero no permite realizar llamadas. Este permiso está reservado para aplicaciones internas de Android (system key signed).
MOUNT_FORMAT_FILESYSTEMS	Permite utilizar las funciones de formateo en dispositivos de almacenamiento extraíble (sdcard). Este permiso está reservado para aplicaciones internas de Android (system key signed).
MOUNT_UNMOUNT_FILESYSTEMS	Permite montar y desmontar dispositivos de almacenamiento extraíble. Este permiso está reservado para aplicaciones internas de Android (system key signed).
READ_CALENDAR	Permite que la aplicación pueda tener acceso al calendario del usuario. Se considera que debe usarse con precaución por posibles violaciones a la privacidad.
READ_CONTACTS	Permite que la aplicación pueda acceder a la lista de contactos del dispositivo. Se considera que debe usarse con precaución por posibles violaciones a la privacidad.

READ_EXTERNAL_STORAGE**	Permite que la aplicación tenga acceso de lectura al dispositivo de almacenamiento extraíble. Requiere que se utilice el “API Level 16” para funcionar.
READ_FRAME_BUFFER	Permite crear capturas de pantalla (screenshots) y tener acceso a los datos “frame buffer data”. Este permiso está reservado para aplicaciones internas de Android (system key signed).
READ_LOGS	Permite que la aplicación pueda tener acceso de lectura a logs de bajo nivel (logs del sistema). Este permiso está reservado para aplicaciones internas de Android (system key signed).
READ_SMS	Permite que la aplicación pueda leer mensajes de texto SMS.
REBOOT	Permite que la aplicación pueda dar la orden de reiniciar el dispositivo. Este permiso está reservado para aplicaciones internas de Android (system key signed).
RECEIVE_BOOT_COMPLETED	Permite que la aplicación pueda recibir el mensaje de broadcast generado al finalizar el proceso de arranque del sistema operativo Android. Tiene algunos efectos negativos en el rendimiento, ya que es un listener que consume recursos significativamente en la aplicación.
RECEIVE_SMS	Permite que la aplicación reciba notificaciones y tenga acceso a cada nuevo mensaje de texto SMS.
RECORD_AUDIO	Permite que la aplicación pueda grabar audio y tener acceso a las funciones de apoyo del micrófono.
SEND_SMS	Permite que la aplicación envíe mensajes de texto SMS. Debe manejarse con cuidado ya que el usuario puede incurrir en costos indeseados.

SET_ALARM**	Permite que la aplicación envíe un intent al AlarmClock de Android, para configurar una alarma nueva o existente.
SET_TIME	Permite que la aplicación pueda editar la hora y fecha del sistema. Este permiso está reservado para aplicaciones internas de Android (system key signed).
SET_TIME_ZONE	Permite que la aplicación pueda cambiar la zona horaria. No tiene restricciones conocidas.
SET_WALLPAPER	Permite que la aplicación pueda cambiar el papel tapiz (Wallpaper) del dispositivo.
STATUS_BAR	Permite manipular la barra de estado de Android en su totalidad. Es posible incluso deshabilitarla para que no sea visible. Este permiso está reservado para aplicaciones internas de Android (system key signed).
UPDATE_DEVICE_STATS	Permite que la aplicación envíe una petición para actualizar las estadísticas del dispositivo. Este permiso está reservado para aplicaciones internas de Android (system key signed).
VIBRATE	Permite acceder a las funcionalidades de vibración en el dispositivo. Debe usarse con cuidado porque provoca que la batería del dispositivo se agote rápidamente.
WAKE_LOCK	Permite que la aplicación ordene al "PowerManager" no apagar la pantalla ni disminuir la actividad eléctrica del procesador. Debe usarse con cuidado porque provoca que la batería del dispositivo se agote rápidamente.
WRITE_CALENDAR	Permite que la aplicación pueda añadir nuevos eventos al calendario del usuario.
WRITE_CALL_LOG	Permite que la aplicación tenga permisos de escritura sobre los datos de cada contacto. Curiosamente, no se otorgan permisos de lectura.

WRITE_CONTACTS	Permite que la aplicación tenga permisos de lectura y escritura sobre los datos de cada contacto.
WRITE_EXTERNAL_STORAGE**	Permite que la aplicación tenga acceso de escritura al dispositivo de almacenamiento extraíble. Si se usa este permiso, no es necesario declarar "READ_EXTERNAL_STORAGE".

## 10. PRUEBAS

Se realizaron una serie de tests dentro de ClassApp para determinar en qué medida se cumplieron los objetivos del proyecto, para evaluar bajo qué circunstancias la aplicación se queda sin memoria y para saber qué parámetros pueden poner en peligro la estabilidad de la aplicación.

### 10.1. PRUEBAS DE DESEMPEÑO GENERAL

**10.1.1. Procesamiento.** El procesamiento de datos sufrió picos constantes: al momento de presentarse cierres inesperados y cuando se trabajó con los experimentos de inserts de SQLite. Los mayores bloqueos de pantalla ocurrieron cuando se ejecutaron los métodos “compileStatement” y “RawSQL”. En segundo lugar, se presentaron bloqueos (skipped frames) al momento de importar grupos con 26 estudiantes y 5 campos, es decir, con poca información. No se le dio demasiada importancia porque el bloqueo dura aproximadamente 800 ms en un dispositivo gama media, y 1200 ms en dispositivos gama baja.

**10.1.2. Cierres inesperados (Crashes).** La mayor cantidad de cierres inesperados se produjo en la etapa inicial de aprendizaje del lenguaje Java y de la arquitectura en Android. Un error frecuente era asignar de manera errónea las funciones “onClick” de los botones, acceder “resources” que no se habían declarado, realizar “intents” hacia clases que todavía no existían o incluso crear clases y layouts sin declararlos en el manifiesto de Android.

Otros crashes ocurrieron al ejecutar ClassApp en dispositivos de gama baja, cuando se intentaba utilizar la cámara con instrucciones de Java y/o al recortar fotografías recién adquiridas. Al parecer son errores asociados al API Level de cada dispositivo, pero no se logró una correlación efectiva de esta hipótesis.

**10.1.3. Versiones de Android.** Para estas pruebas se utilizaron tres dispositivos físicos (Android 2.3, 5.0) y múltiples máquinas virtuales (tablet y smartphone), cada una con diferente API Level. Los cambios más evidentes corresponden al renderizado incorrecto de fuentes en la versión 2.3. Otros cambios se hacen evidentes en la densidad de píxeles en pantalla, que es gestionada de forma distinta con cada API Level. Por supuesto, las aplicaciones que gestionan el uso de la cámara sufren cambios en cada versión de Android, y no muestran sus componentes de recorte de fotografías de la misma manera.

**10.1.4. Hardware (Cámara y acelerómetro).** En dispositivos de gama baja se observó que realizar un llamado al API de la cámara, ocurren fallos con mucha frecuencia. La prueba que se realizó consistió en acceder al módulo de toma de asistencia para tomar una fotografía simple. Sería deseable establecer si existe una correlación adicional entre la versión de Android de estos dispositivos y la ocurrencia del problema (estos dispositivos normalmente ejecutan Android 2.3 o versiones anteriores).

Respecto al acelerómetro, se observa que las clases son cargadas de nuevo cuando se cambia la orientación. Esto representa un problema en aquellos casos en donde se realizan procesos críticos durante el evento onCreate y no se desea que sean ejecutados dos o más veces seguidas. Este incidente es independiente de la versión de Android o de las prestaciones del dispositivo en que se realiza la prueba.

## **10.2. PRUEBAS DE DESEMPEÑO EN SQLITE**

Se tomó la decisión de trabajar con SQLite debido a que el conector de Java para MySQL recarga la máquina virtual del dispositivo. Con esto, el siguiente paso es evaluar el rendimiento de este motor de bases de datos aplicándole una carga considerable de operaciones SQL-92. El método tradicional para probar la carga es iterativo: se realizan operaciones de alto consumo de procesamiento (sentencias insert) en el orden de 100, 200 y 300 registros, cada uno con 6 campos. Se utilizaron cadenas de texto reales para darle más confiabilidad a la prueba.

Existen tres métodos de ejecución en SQLite: "transaction", "RawSQL" y "compileStatement". Se realizaron las mediciones correspondientes, con los siguientes resultados:

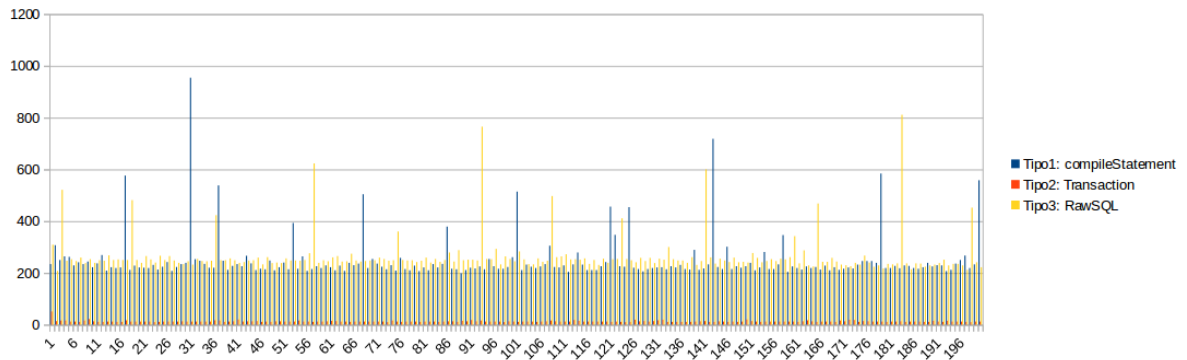


Figura 7. Resultados iniciales de los métodos de ejecución en SQLite

Los picos de procesamiento que se observan fueron eliminados mediante la adopción del 95% inicial de los datos (previamente ordenados):

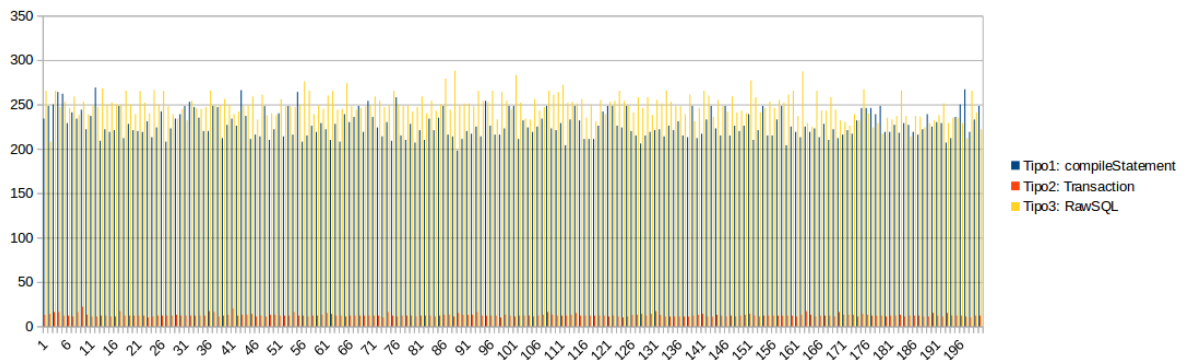


Figura 8. Eliminación de los picos de procesamiento

Estos experimentos fueron incluidos en un módulo de ClassApp, en la sección de "Configuración". El siguiente es un ejemplo de una ejecución de los tres métodos:

N = 300 N <sub>95%</sub> = 285 26 registros (6 campos)		
<b>Transaction</b>	<b>CompileStatement</b>	<b>RawSQL</b>
11.16 ns	218.94 ns	217 ns

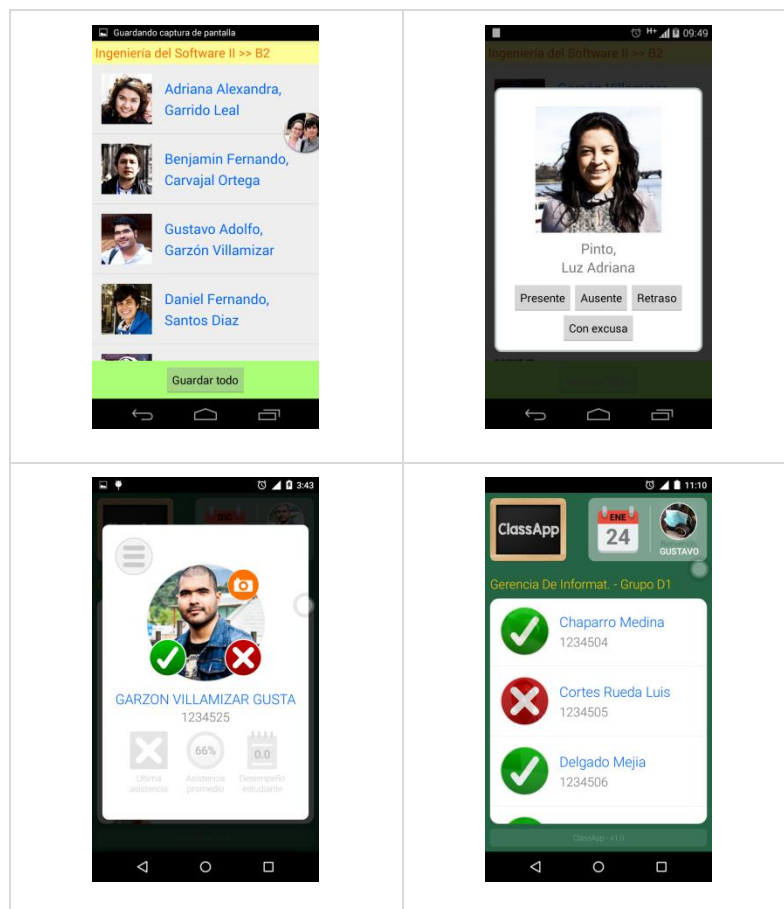
Se observa que en términos de tiempo de ejecución, las diferencias favorecieron al método "transaction". Globalmente, este método superó a los demás en un orden de entre 16 y 19 veces. Se le da importancia a estas cifras debido a que un dispositivo móvil no dispone de grandes prestaciones, y debe desarrollarse cada aplicación siempre pensando en dejar la mayor cantidad de memoria disponible para la máquina virtual y los demás módulos de Android.

### 10.3. PRUEBAS DE USABILIDAD (INTERFAZ GRÁFICA)

La primera fase de prototipado de ClassApp no puso mucho empeño en el ámbito gráfico, justamente porque apenas se estaba conociendo el lenguaje de desarrollo, la arquitectura y el IDE. Los layouts que se crearon inicialmente eran funcionales, pero su valor estético no era el mejor. Además, la interfaz se planeó para ayudar a resolver una tarea determinada (por ejemplo, tomar asistencia), pero no se contempló que en un dispositivo móvil la cantidad de clicks (taps) debería ser la menor posible.

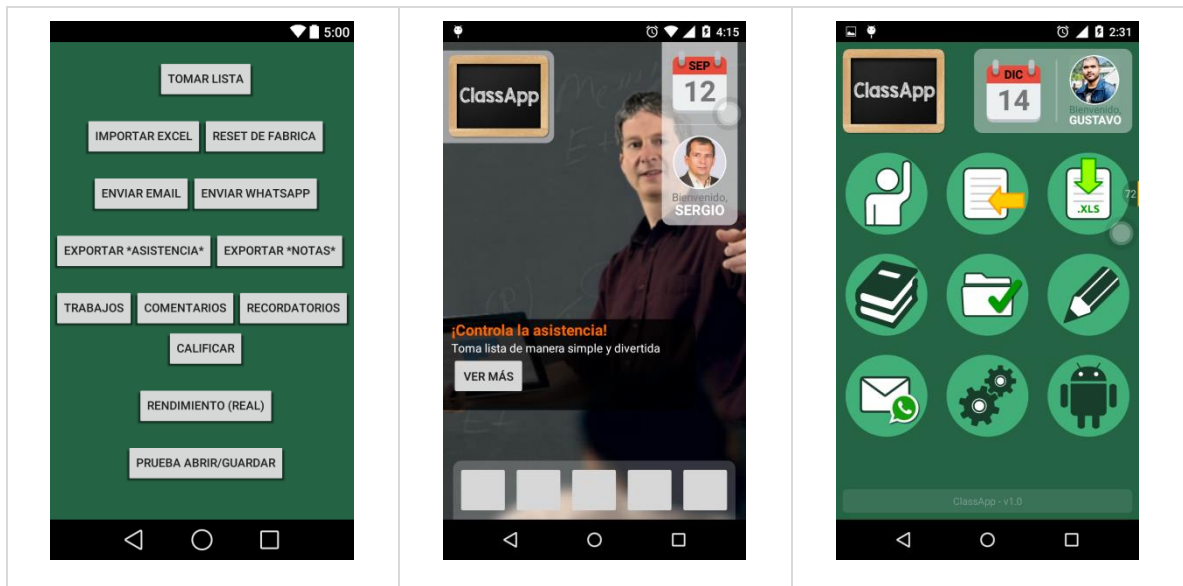
Siguiendo estos criterios, se replantearon las interfaces gráficas de ClassApp para la segunda fase, personalizando los estilos de los componentes y planteando nuevas formas de interacción con los ListView.

Un ejemplo concreto es el siguiente: la interfaz gráfica para la toma de asistencia consistía en un ListView que se mostraba al ingresar a la actividad. Al dar click a cualquiera de los estudiantes, se abría una ventana con opciones para tomar asistencia. Para repetir el proceso con el siguiente estudiante en orden de lista, había que regresar al ListView y seleccionarlo manualmente. La mejora que se aplicó consiste en ahorrarse esos clicks adicionales mediante la carga automática de estudiantes. La actividad ahora inicia con la ventana de opciones abierta con el primer estudiante. Al asignar el valor de asistencia, automáticamente se muestra el siguiente en orden de lista. Adicionalmente, se activa un ícono en el ListView que indica el valor de asistencia que se le dio a cada estudiante, mostrándose como una imagen con transparencia del 90% sobre cada foto.



Otro ejemplo que vale la pena destacar, es la portada de la aplicación. Inicialmente, era un menú con botones planos, sin personalizar, netamente funcionales. Se planteó un diseño que utilizara imágenes en pantalla completa para representar cada elemento de menú.

Estas imágenes irían cambiando de manera animada al dar click a una barra de botones en la parte inferior, pero la idea se abandonó luego de observar graves problemas de desempeño durante las animaciones. Adicionalmente, la gestión de imágenes para diferentes tipos de pantalla es un trabajo al que debe dedicársele mucho tiempo. Se decidió entonces optar por una alternativa más clásica: un menú de íconos, un fondo de color plano y una estructura constante (cabezote, contenido y pie de página) que se mostrará en todos los layout de la aplicación (salvo en la sección de créditos).



#### 10.4. PRUEBAS DE FUNCIONALIDAD

Tras una revisión final de la aplicación, en la cual se utilizaron todas sus funcionalidades, se determina que las características que se definieron en el Documento de Especificación de Requisitos simplificado (Pág. 44) fueron implementadas en su totalidad.

#### 10.5. EVALUACIÓN DEL CUMPLIMIENTO DE OBJETIVOS

Finalmente, se presenta un resumen que contrasta los objetivos del proyecto con el producto entregado, determinando el porcentaje de avances alcanzados y la ubicación exacta de cada ítem en el presente documento.

<b>Objetivo</b>	<b>Cumplimiento</b>	<b>Ubicación</b>
1) Elaborar el DER Documento de Especificación de Requisitos simplificado de la aplicación.	100%	Pág. 44
2) Diseñar una aplicación que incluya las siguientes funcionalidades: <ul style="list-style-type: none"> <li>- Importar toda la información inicial para la aplicación (periodo, asignaturas, grupos, horarios y estudiantes) desde una base de datos remota,</li> <li>- Registro de información adicional sobre los estudiantes (notas y/o recordatorios),</li> <li>- Comunicación con los estudiantes (envío de correos electrónicos),</li> <li>- Control de asistencia,</li> <li>- Comentarios y recordatorios de las sesiones docentes,</li> <li>- Exportar documentos de Excel/CSV con la información de asistencia de un grupo o un estudiante.</li> </ul>	100%	Pág. 51 Pág. 98
3) Caracterizar los permisos asociados al Manifiesto de Permisos del sistema operativo Android, indicando la importancia de cada uno de sus ítems y sus implicaciones técnicas.	100%	Pág. 70

## 11. CONCLUSIONES

- Se desarrolló un prototipo software móvil para dispositivos Android que facilitó la gestión de información de estudiantes, la toma de asistencia y otras generalidades de la sesión docente.
- La utilización de prototipos rápidos ayudó a visualizar de manera ágil qué soluciones constituían la mejor alternativa de diseño, sin necesidad de entrar en detalles con un proceso de desarrollo de layouts XML.
- El módulo de toma de asistencia reduce la cantidad de tiempo que el docente tiene que dedicar a este proceso, mediante una interfaz intuitiva y la posibilidad de exportar la información de manera instantánea. El proceso de calificación de actividades, igualmente, tiene el potencial de agilizar las actividades en cada sesión docente.
- Las aplicaciones móviles requieren metodologías de desarrollo ágiles, de naturaleza iterativa, que muestren resultados en poco tiempo y que sean suficientemente flexibles ante los cambios en los requerimientos.
- Desarrollar interfaces que agilicen los procesos requiere un estudio muy detallado. Es deseable contar con la ayuda de los profesionales que normalmente desempeñan estos procesos, para conocer qué dificultades se les presentan y qué partes del proceso pueden ser automatizadas.
- Si bien los dispositivos móviles son prácticos en términos de portabilidad, todavía no logran igualar el poder de cómputo de un computador de escritorio, lo cual es necesario para ejecutar aplicaciones más robustas (como MySQL) y para no permitir que la máquina virtual consuma la gran mayoría de la memoria RAM.
- Existe un gran potencial para mejorar el desempeño de SQLite utilizando el poder de cómputo de la GPU en cada dispositivo. Esto se ha venido experimentando utilizando el framework RenderScript de Android, el cual se incluye desde el API Level 8. Esto podría

generar beneficios para cualquier iniciativa que requiera de cómputo intensivo con dispositivos móviles. Requiere el manejo de directivas y funciones del estándar C99 (lenguaje C en 1999).

- La fragmentación de versiones de Android y los diferentes tamaños de pantalla de los dispositivos constituyen un problema difícil de abordar. Especialmente cuando se trabaja con interfaces personalizadas e íconos de un tamaño uniforme, como es el caso de ClassApp. Para abordar este problema se calculó la proporción de cada elemento de la interfaz gráfica respecto al ancho de la pantalla del dispositivo, lo cual garantiza buenos resultados en diferentes densidades de píxeles. Esta estrategia se utilizó en la portada (para calcular el tamaño de los íconos) y en la interfaz de toma de asistencia.

- La carga de procesamiento que impone la máquina virtual Dalvik, provoca picos en las cantidades de tiempo empleadas por otros procesos de Android. Esto entorpece las mediciones y provoca que se tenga que recurrir a estrategias de eliminación de medidas erróneas dentro de un conjunto de datos, lo cual ocurrió con las pruebas de SQLite. Una forma de probar esta hipótesis, es ejecutando ClassApp en una máquina virtual de Android Studio. Al observar los resultados de las pruebas de desempeño SQLite, es visible que no existen picos de procesamiento, ya que el poder de cómputo emulado se destina 100% a las aplicaciones. Android RunTime (ART) es la máquina virtual que se elaboró después de Dalvik, y se dice que será optimizada en versiones posteriores para obtener un mejor rendimiento general en dispositivos móviles.

- ClassApp fue publicada en la tienda Google Play Store, buscando facilitar el acceso e instalación, y obtener estadísticas de crashes, versiones de dispositivos, reportes de usuarios y otros datos que ayuden a mejorar la aplicación.

## 12. RECOMENDACIONES

- Debe tenerse en cuenta que cuando cambia la orientación del dispositivo (uso del acelerómetro) Android vuelve a cargar la clase tipo Activity que se está ejecutando en ese momento. Se recomienda bloquear el acelerómetro temporalmente con una instrucción de Java, lo cual impide que se pierdan datos por volver a cargar todos los procesos de la clase actual.
- El API de interacción con el hardware tiene limitaciones según la versión de Android en que se ejecute ClassApp. Pueden existir métodos obsoletos al pasar de una versión a otra, o simplemente podrían no existir métodos para la situación que se esté manejando. Es un fenómeno difícil de abordar, debido a que el desarrollador tiene que buscar una forma alterna de acceder al hardware.
- En versiones iguales o anteriores a Android 2.3 (Gingerbread) se observa que al utilizar por medio de Java a la cámara del dispositivo, tienden a presentarse cierres inesperados o crashes. Este fenómeno afecta la funcionalidad del módulo de toma de fotografías de ClassApp, y obliga al usuario a volver a ingresar a la aplicación para repetir la toma. Se recomienda revisar si es posible un acceso optimizado al hardware de la cámara utilizando procesos de más bajo nivel, o trabajando con el lenguaje C.
- Android aún no ha desarrollado un API para manipular los métodos AlarmClock en su totalidad [55], además que los fabricantes de dispositivos tienden a reemplazar la aplicación "Alarmas" con versiones diferentes en cada referencia. El problema radica en que esta clase y su integración con las notificaciones (notification builder) son clave para mejorar la funcionalidad de recordatorios en ClassApp. Se recomienda revisar la referencia de la clase AlarmClock con frecuencia, ya que varios desarrolladores han solicitado extender las funcionalidades del API.
- Los archivos que se almacenen (en tiempo de desarrollo) en la carpeta "assets" (salvo los tipos de letra o Typeface) tienen restricciones de lectura. Es recomendable copiar estos

archivos (en tiempo de ejecución) a una carpeta temporal de la aplicación para que puedan ser leídos desde esa nueva ubicación.

- La carpeta "drawable" tiene restricciones para los nombres de los archivos. Estos no pueden contener: caracteres especiales, espacios ni caracteres en mayúscula. Adicionalmente, no es posible crear subdirectorios en esta carpeta, debido al mecanismo de traducción de recursos (R.java). Se sugiere consultar las implicaciones de editar la sección "sourceSets" del archivo "build.gradle" como solución alternativa a esta situación.

- Debido a las directivas de almacenamiento en dispositivos móviles, siempre que se genere un archivo en la memoria sdcard, debe notificarse al MediaScannerHelper. Esta notificación avisa al sistema de archivos que se ha creado un elemento nuevo y que debe ser indexado de inmediato. Es deseable realizar este procedimiento ya que, de lo contrario, el archivo solo será visible después de reiniciar el dispositivo móvil.

- Se sugiere que todos los permisos de utilización de hardware y algunos módulos de Android, sean declarados en el archivo "AndroidManifest.xml" en la sección "uses-permission", de la siguiente forma:

```
<uses-permission android:name="android.permission.STORAGE" />
```

- Para publicar una aplicación en Google Play Store es necesario generar un APK firmado digitalmente. Adicionalmente, es necesario ir incrementando la versión de la aplicación cada vez que se publiquen cambios definitivos en el archivo APK. Esto se realiza editando el archivo "build.gradle" en la sección "defaultConfig", ítem "versionCode".

- Al añadir librerías (".jar") a la aplicación, existe un método que utiliza la interfaz gráfica de Android Studio y un método manual. El método manual consiste en: copiar los archivos ".jar" en la carpeta "libs" del proyecto, y luego añadir cada uno editando el archivo "build.gradle" en la sección "dependencies" de la siguiente manera:

```
dependencies {  
    compile fileTree(include: ['*.jar'], dir: 'libs')  
    compile files('libs/jxl.jar')  
    compile files('libs/mpandroidchartlibrary-2-1-3.jar')
```

}

- Se sugiere consultar el protocolo que se sigue para que una publicación en Google Play sea considerada como “actualización”. Si una de las actualizaciones implica un cambio en la estructura de la base de datos o un cambio crítico, es posible que exista una pérdida de datos que sin duda afectará al usuario final.

- Para trabajar con la propiedad “android:alpha” en un layout XML, se debe saber de antemano que no se mostrará correctamente en dispositivos con versiones de Android inferiores al API Level 11. Una solución que se implementó en este proyecto fue el uso de imágenes PNG con transparencia (32 bit), las cuales están soportadas en un rango más amplio de versiones.

- Teniendo en cuenta que las aplicaciones móviles están en auge, y que pedagógicamente el patrón MVC constituye una buena estrategia de enseñanza para elaborar aplicaciones con interfaz gráfica (Java + XML + SQLite), se sugiere a la Escuela de Ingeniería de Sistemas e Informática de crear una asignatura electiva orientada al desarrollo de este tipo de aplicaciones, con la ventaja de poder utilizar tecnologías gratuitas (Android, Android Studio y JDK) y no incurrir en costos de licenciamiento.

## CITAS

- [1] CASTRO ALEX. COMSCORE, INC. "Futuro digital Colombia 2015". 22 p. [Internet]. Disponible en: <  
[https://www.comscore.com/lat/layout/set/popup/content/download/32509/1831411/version/1/file/2015\\_Colombia\\_Digital\\_Future\\_in\\_Focus.pdf](https://www.comscore.com/lat/layout/set/popup/content/download/32509/1831411/version/1/file/2015_Colombia_Digital_Future_in_Focus.pdf) >. [Citado en 2016-01-10].
- [2] CASTRO ALEX. COMSCORE, INC. "Futuro digital Colombia 2015". 25 p. [Internet]. Disponible en: <  
[https://www.comscore.com/lat/layout/set/popup/content/download/32509/1831411/version/1/file/2015\\_Colombia\\_Digital\\_Future\\_in\\_Focus.pdf](https://www.comscore.com/lat/layout/set/popup/content/download/32509/1831411/version/1/file/2015_Colombia_Digital_Future_in_Focus.pdf) >. [Citado en 2016-01-10].
- [3] GARTNER, INC. "Gartner Says Worldwide Traditional PC, Tablet, Ultramobile and Mobile Phone Shipments On Pace to Grow 7.6 Percent in 2014". [Internet]. Disponible en: <  
<http://www.gartner.com/newsroom/id/2645115> >. [Citado en 2014-10-22].
- [4] OBSERVATORIO DE APPS. "Prospectiva de Adopción de Dispositivos Móviles". [Internet]. Disponible en: <  
[http://www.observatorioapps.com/es/ObservatorioApps/Prospectiva\\_de\\_Adopcion\\_de\\_Dispositivos\\_Moviles](http://www.observatorioapps.com/es/ObservatorioApps/Prospectiva_de_Adopcion_de_Dispositivos_Moviles) >. [Citado en 2015-12-26].
- [5] PIEDADE, M.B.; SANTOS, M.Y. "Student Relationship Management: Concept, practice and technological support". Engineering Management Conference, 2008. IEMC Europe 2008. IEEE International. [Internet]. Disponible en: <  
<http://ieeexplore.ieee.org/xpl/articleDetails.jsp?reload=true&tp=&arnumber=4618026> >. [Citado en 2014-10-25].
- [6] ANDROID, INC. "Security". [Internet]. Disponible en: <  
<https://source.android.com/security/> >. [Citado en 2016-01-12].

- [7] CHANDAN. "Introduction to Application Development for Portable Devices-1". [Internet]. Disponible en: < <http://thechandan.wordpress.com/2013/09/12/introduction-to-application-development-for-portable-devices-1> >. [Citado en 2014-10-26].
- [8] GAHRAN, AMY. "What's a mobile app?". [Internet]. Disponible en: < <http://www.contentious.com/2011/03/02/whats-a-mobile-app> >. [Citado en 2014-10-27].
- [9] GOOGLE, INC. "Google Play". [Internet]. Disponible en: < <https://play.google.com> >. [Citado en 2014-10-27].
- [10] ANDROID, INC. DEVELOPER CONSOLE HELP. "Upload & distribute apps". [Internet]. Disponible en: < <https://support.google.com/googleplay/android-developer/answer/113469?hl=en> >. [Citado en 2014-10-27].
- [11] ANDROID, INC. DEVELOPERS. "Get Started with Publishing". [Internet]. Disponible en: < <http://developer.android.com/distribute/googleplay/start.html> >. [Citado en 2014-10-27].
- [12] Chiu, C., Young, L., & Lo, C. (2015). The Impact of Elementary School Management Regarding the Smaller Class Quality Education. 2015 IEEE International Conference on Computational Intelligence & Communication Technology.
- [13] Raichel, N. (2013). Computerized class management software: Teachers' views and assessment. *Journal of OrgChange Mgmt Journal of Organizational Change Management*, 26(5), 830-852.
- [14] Zeng, B., Feng, S., & Zhang, J. (2010). Web-Based Class Management System Using E-commerce Technology. 2010 Third International Symposium on Electronic Commerce and Security.
- [15] Bongani T., M., & Johnson O., D. (2012). Enhancing Online University Class Management System with Instant Email Feedback Alert. *Lecture Notes In Engineering And Computer Science*, (1), 425.

[16] MY ATTENDANCE TRACKER (MyAT). "Simple, free attendance tracking software". [Internet]. Disponible en: < <https://www.myattendancetracker.com/free-attendance-tracking-software> >. [Citado en 2014-10-28].

[17] GOOGLE, INC. "Education - Android Apps on Google Play". [Internet]. Disponible en: < <https://play.google.com/store/apps/category/EDUCATION> >. [Citado en 2015-11-12].

[18] THE VERGE. "Android: A visual history". [Internet]. Disponible en: < <http://www.theverge.com/2011/12/7/2585779/android-history> >. [Citado en 2014-10-25].

[19] TICAY RIVAS, JAIME ROBERTO. "La arquitectura de Android". [Internet]. Disponible en: < <http://catedratelefonica.ulpgc.es/blog/arucas/2012/10/18/la-arquitectura-de-android> >. [Citado en 2014-10-25].

[20] BRADY, PATRICK. GOOGLE, INC. "Android Anatomy and Physiology". [Internet]. Disponible en: < <https://sites.google.com/site/io/anatomy--physiology-of-an-android> >. [Citado en 2014-10-26].

[21] WIKIPEDIA, THE FREE ENCYCLOPEDIA. "Android Studio". [Internet]. Disponible en: < [https://en.wikipedia.org/wiki/Android\\_Studio](https://en.wikipedia.org/wiki/Android_Studio) >. [Citado en 2015-08-11].

[22] ANDROID, INC. DEVELOPERS. "Android Studio Overview". [Internet]. Disponible en: < <http://developer.android.com/tools/studio/index.html> >. [Citado en 2015-09-03].

[23] ST. JOHN, REX. "Android Studio vs. Eclipse: What You Need To Know". [Internet]. Disponible en: < <https://www.airpair.com/android/android-studio-vs-eclipse> >. [Citado en 2015-07-23].

[24] WIKIPEDIA, THE FREE ENCYCLOPEDIA. "JetBrains". [Internet]. Disponible en: < <https://en.wikipedia.org/wiki/JetBrains> >. [Citado en 2015-08-14].

- [25] INTELLIJ IDEA. “Features”. [Internet]. Disponible en: < <https://www.jetbrains.com/idea/features/android.html> >. [Citado en 2015-09-27].
- [26] ANDROID DEVELOPERS BLOG. “Android Studio: An IDE built for Android”. [Internet]. Disponible en: < <http://android-developers.blogspot.in/2013/05/android-studio-ide-built-for-android.html> >. [Citado en 2015-07-01].
- [27] QUORA. “What should I use for Android development, Android Studio or Eclipse?”. [Internet]. Disponible en: < <https://www.quora.com/Should-I-use-Android-Studio-or-Eclipse-for-my-first-Android-app> >. [Citado en 2015-07-02].
- [28] ANDROID, INC. DEVELOPERS. “Tools Help”. [Internet]. Disponible en: < <http://developer.android.com/tools/help/index.html> >. [Citado en 2015-07-16].
- [29] ANDROID, INC. DEVELOPERS. “Using DDMS”. [Internet]. Disponible en: < <http://developer.android.com/tools/debugging/ddms.html> >. [Citado en 2015-07-16].
- [30] PHP: HYPERTEXT PREPROCESSOR. “Introduction to SQLite”. [Internet]. Disponible en: < [http://talks.php.net/show/sqlite\\_jan/1](http://talks.php.net/show/sqlite_jan/1) >. [Citado en 2015-06-22].
- [31] SQLITE. “About SQLite”. [Internet]. Disponible en: < <https://www.sqlite.org/about.html> >. [Citado en 2015-06-22].
- [32] ANDROID, INC. DEVELOPERS. “android.database.sqlite”. [Internet]. Disponible en: < <http://developer.android.com/intl/es/reference/android/database/sqlite/package-summary.html> >. [Citado en 2015-06-22].
- [33] SAITO, TARO L. BITBUCKET, ATLISSIAN. “SQLite JDBC Driver”. [Internet]. Disponible en: < <https://bitbucket.org/xerial/sqlite-jdbc> >. [Citado en 2015-06-22].
- [34] SQLITE. “Limits In SQLite”. [Internet]. Disponible en: < <http://www.sqlite.org/limits.html> >. [Citado en 2015-06-22].

[35] SQLITE. "Atomic Commit In SQLite". [Internet]. Disponible en: < <http://www.sqlite.org/atomiccommit.html> >. [Citado en 2015-06-22].

[36] SQLITE. "How SQLite Is Tested". [Internet]. Disponible en: < <http://www.sqlite.org/testing.html> >. [Citado en 2015-06-22].

[37] SZETO, EDMUND. LINARO. "Optimizing SQLite database in Android". [Internet]. Disponible en: < <http://www.linaro.org/blog/android-blog/optimizing-sqlite-database-android> >. [Citado en 2015-06-22].

[38] KHAN, ANDY. JEXCELAPI. "Java Excel API - A Java API to read, write, and modify Excel spreadsheets". [Internet]. Disponible en: < <http://jexcelapi.sourceforge.net> >. [Citado en 2015-06-22].

[39] SOURCEFORGE. "A Java API to read, write, and modify Excel spreadsheets". [Internet]. Disponible en: < <http://sourceforge.net/projects/jexcelapi/files/> >. [Citado en 2015-06-22].

[40] THE APACHE POI PROJECT. "Apache POI - the Java API for Microsoft Documents". [Internet]. Disponible en: < <http://poi.apache.org> >. [Citado en 2015-06-22].

[41] JAHODA, PHILIPP. "MPAndroidChart". [Internet]. Disponible en: < <https://github.com/PhilJay/MPAndroidChart/releases> >. [Citado en 2015-06-22].

[42] SHPITALNIK, GREGORY. CODEPROJECT. "Android: Ready to use simple directory chooser dialog with new folders creation". [Internet]. Disponible en: < <http://www.codeproject.com/Articles/547636/Android-Ready-to-use-simple-directory-chooser-dial> >. [Citado en 2015-06-23].

[43] SCORCH WORKS. "Simple File Dialog For Android Applications". [Internet]. Disponible en: < <http://www.scorchworks.com/Blog/simple-file-dialog-for-android-applications/> >. [Citado en 2015-06-23].

[44] CODEPROJECT. "CPOL: Code Project Open License". [Internet]. Disponible en: < <http://www.codeproject.com/info/cpol10.aspx> >. [Citado en 2015-06-23].

[45] Bruegge, B. & Dutoit, A. (2010). Object-oriented software engineering : using UML, patterns, and Java. Boston: Prentice Hall. p 119.

[46] Grady, R. (1992). Practical software metrics for project management and process improvement. Englewood Cliffs, NJ: Prentice Hall. p 32.

[47] PIDOCO GMBH. "Pidoco - a GUI prototyping tool". [Internet]. Disponible en: < <https://pidoco.com> >. [Citado en 2015-06-23].

[48] WIKIPEDIA, THE FREE ENCYCLOPEDIA. "Adobe Fireworks". [Internet]. Disponible en: < [https://en.wikipedia.org/wiki/Adobe\\_Fireworks](https://en.wikipedia.org/wiki/Adobe_Fireworks) >. [Citado en 2015-06-23].

[49] CREATELY. "Creately - Online Diagram Software to draw Flowcharts, UML & more". [Internet]. Disponible en: < <https://creately.com> >. [Citado en 2015-06-23].

[50] CARRERA, MARCELO. "Proyectos Tecnológicos Modelo de Desarrollo Prototipado". [Internet]. Disponible en: < [http://www.imaginar.org/iicd/tus\\_archivos/LLL/docs/5\\_laboratorios.pdf](http://www.imaginar.org/iicd/tus_archivos/LLL/docs/5_laboratorios.pdf) >. [Citado en 2014-10-29].

[51] GARZÁS PARRA, JAVIER. ESCUELA SUPERIOR DE INGENIERÍA INFORMÁTICA. UNIVERSIDAD REY JUAN CARLOS. "Sistemas de información - Tema 4 - Bases del desarrollo software". [Internet]. Disponible en: < [http://kybele.escet.urjc.es/documentos/SI/%5BSI-2006-07%5DT4\\_Bases\\_del\\_Desarrollo\\_Software.pdf](http://kybele.escet.urjc.es/documentos/SI/%5BSI-2006-07%5DT4_Bases_del_Desarrollo_Software.pdf) >. [Citado en 2014-10-29].

[52] Hekmatpour, Sharam. "Experience with evolutionary prototyping in a large software project". ACM SIGSOFT Software Engineering Notes Homepage archive. Volume 12 Issue 1, Jan. 1987. Pages 38-41. [Internet]. Disponible en: <

<http://dl.acm.org/citation.cfm?id=24574.24577&coll=DL&dl=ACM> >. [Citado en 2014-11-24].

[53] AGUILAR CASTILLO, GILDARDO. UNIVERSIDAD VERACRUZANA. “Ciclo de vida de desarrollo de sistemas”. [Internet]. Disponible en: < <http://www.geocities.ws/gildardoaguilar/adp4.pdf> >. [Citado en 2014-11-25].

[54] ANDROID, INC. DEVELOPERS. “Dashboards”. [Internet]. Disponible en: < <http://developer.android.com/about/dashboards/index.html> >. [Citado en 2016-01-03].

[55] ANDROID, INC. DEVELOPERS. “AlarmClock”. [Internet]. Disponible en: < <http://developer.android.com/reference/android/provider/AlarmClock.html> >. [Citado en 2016-01-06].

## BIBLIOGRAFIA

AGUILAR CASTILLO, Gildardo. UNIVERSIDAD VERACRUZANA. "Ciclo de vida de desarrollo de sistemas". [Internet]. Disponible en: < <http://www.geocities.ws/gildardoaguilar/adp4.pdf> >. [Citado en 2014-11-25].

ANDROID, INC. "Security". [Internet]. Disponible en: < <https://source.android.com/security/> >. [Citado en 2016-01-12].

ANDROID, INC. DEVELOPER CONSOLE HELP. "Upload & distribute apps". [Internet]. Disponible en: < <https://support.google.com/googleplay/android-developer/answer/113469?hl=en> >. [Citado en 2014-10-27].

ANDROID, INC. DEVELOPERS. "AlarmClock". [Internet]. Disponible en: < <http://developer.android.com/reference/android/provider/AlarmClock.html> >. [Citado en 2016-01-06].

ANDROID, INC. DEVELOPERS. "android.database.sqlite". [Internet]. Disponible en: < <http://developer.android.com/intl/es/reference/android/database/sqlite/package-summary.html> >. [Citado en 2015-06-22].

ANDROID DEVELOPERS BLOG. Android Studio: An IDE built for Android. [Internet]. Disponible en: < <http://android-developers.blogspot.in/2013/05/android-studio-ide-built-for-android.html> >. [Citado en 2015-07-01].

ANDROID, INC. DEVELOPERS. "Android Studio Overview". [Internet]. Disponible en: < <http://developer.android.com/tools/studio/index.html> >. [Citado en 2015-09-03].

ANDROID, INC. DEVELOPERS. "Dashboards". [Internet]. Disponible en: < <http://developer.android.com/about/dashboards/index.html> >. [Citado en 2016-01-03].

ANDROID, INC. DEVELOPERS. "Get Started with Publishing". [Internet]. Disponible en: < <http://developer.android.com/distribute/googleplay/start.html> >. [Citado en 2014-10-27].

ANDROID, INC. DEVELOPERS. "Tools Help". [Internet]. Disponible en: < <http://developer.android.com/tools/help/index.html> >. [Citado en 2015-07-16].

ANDROID, INC. DEVELOPERS. "Using DDMS". [Internet]. Disponible en: < <http://developer.android.com/tools/debugging/ddms.html> >. [Citado en 2015-07-16].

THE APACHE POI PROJECT. "Apache POI - the Java API for Microsoft Documents". [Internet]. Disponible en: < <http://poi.apache.org> >. [Citado en 2015-06-22].

BONGANI T., M., & JOHNSON O., D. Enhancing Online University Class Management System with Instant Email Feedback Alert. Lecture Notes In Engineering and Computer Science, 2012) (1), 425.

BRADY, PATRICK. GOOGLE, INC. "Android Anatomy and Physiology". [Internet]. Disponible en: < <https://sites.google.com/site/io/anatomy--physiology-of-an-android> >. [Citado en 2014-10-26].

BRUEGGE, B. & DUTOIT, A. Object-oriented software engineering : using UML, patterns, and Java. Boston: Prentice Hall. 2010. p 119.

CARRERA, Marcelo. "Proyectos Tecnológicos Modelo de Desarrollo Prototipado". [Internet]. Disponible en: < [http://www.imaginar.org/iicd/tus\\_archivos/LLL/docs/5\\_laboratorios.pdf](http://www.imaginar.org/iicd/tus_archivos/LLL/docs/5_laboratorios.pdf) >. [Citado en 2014-10-29].

CODEPROJECT. "CPOL: Code Project Open License". [Internet]. Disponible en: < <http://www.codeproject.com/info/cpol10.aspx> >. [Citado en 2015-06-23].

COMSCORE. Futuro digital Colombia 2015. [en línea] [Reston, Virginia, Estados Unidos] Comscore, 2015. [citado 3 Dic. 2015] Disponible en Internet: <http://www.comscore.com/lat/Prensa-y-Eventos/Blog/Futuro-Digital-America-Latina-2015>

CHANDAN. "Introduction to Application Development for Portable Devices-1". [Internet]. Disponible en: < <http://thechandan.wordpress.com/2013/09/12/introduction-to-application-development-for-portable-devices-1> >. [Citado en 2014-10-26].

CHIU, C., YOUNG, L., & LO, C. (2015). The Impact of Elementary School Management Regarding the Smaller Class Quality Education. 2015 IEEE International Conference on Computational Intelligence & Communication Technology.assessment. Journal of OrgChange Mgmt Journal of Organizational Change Management, 26(5), 830-852.

CREATELY. "Creately - Online Diagram Software to draw Flowcharts, UML & more". [Internet]. Disponible en: < <https://creately.com> >. [Citado en 2015-06-23].

GAHRAN, AMY. "What's a mobile app?". [Internet]. Disponible en: < <http://www.contentious.com/2011/03/02/whats-a-mobile-app> >. [Citado en 2014-10-27].

GARTNER, INC. "Gartner Says Worldwide Traditional PC, Tablet, Ultramobile and Mobile Phone Shipments On Pace to Grow 7.6 Percent in 2014". [Internet]. Disponible en: < <http://www.gartner.com/newsroom/id/2645115> >. [Citado en 2014-10-22].

GARZÁS PARRA, JAVIER. ESCUELA SUPERIOR DE INGENIERÍA INFORMÁTICA. UNIVERSIDAD REY JUAN CARLOS. "Sistemas de información - Tema 4 - Bases del desarrollo software". [Internet]. Disponible en: < [http://kybele.escet.urjc.es/documentos/SI/%5BSI-2006-07%5DT4\\_Bases\\_del\\_Desarrollo\\_Software.pdf](http://kybele.escet.urjc.es/documentos/SI/%5BSI-2006-07%5DT4_Bases_del_Desarrollo_Software.pdf) >. [Citado en 2014-10-29].

GOOGLE, INC. Google Play. [Internet]. Disponible en: < <https://play.google.com> >. [Citado en 2014-10-27].

GOOGLE, INC. "Education - Android Apps on Google Play". [Internet]. Disponible en: < <https://play.google.com/store/apps/category/EDUCATION> >. [Citado en 2015-11-12].

GRADY, R. Practical software metrics for project management and process improvement. Englewood Cliffs, NJ: Prentice Hall. 1992. p 32.

HEKMATPOUR, Sharam. "Experience with evolutionary prototyping in a large software project". ACM SIGSOFT Software Engineering Notes Homepage archive. Volume 12 Issue 1, Jan. 1987. Pages 38-41. [Internet]. Disponible en: < <http://dl.acm.org/citation.cfm?id=24574.24577&coll=DL&dl=ACM> >. [Citado en 2014-11-24].

IEEE International. [Internet]. Disponible en: < <http://ieeexplore.ieee.org/xpl/articleDetails.jsp?reload=true&tp=&arnumber=4618026> >. [Citado en 2014-10-25].

INTELLIJ IDEA. "Features". [Internet]. Disponible en: < <https://www.jetbrains.com/idea/features/android.html> >. [Citado en 2015-09-27].

JAHODA, PHILIPP. "MPAndroidChart". [Internet]. Disponible en: < <https://github.com/PhilJay/MPAndroidChart/releases> >. [Citado en 2015-06-22].

KHAN, ANDY. JEXCELAPI. "Java Excel API - A Java API to read, write, and modify Excel spreadsheets". [Internet]. Disponible en: < <http://jexcelapi.sourceforge.net> >. [Citado en 2015-06-22].

MY ATTENDANCE TRACKER (MyAT). "Simple, free attendance tracking software". [Internet]. Disponible en: < <https://www.myattendancetracker.com/free-attendance-tracking-software> >. [Citado en 2014-10-28].

OBSERVATORIO DE APPS. "Prospectiva de Adopción de Dispositivos Móviles". [en línea]. [Citado en 2015-12-26]. Disponible en Internet:

[http://www.observatorioapps.com/es/ObservatorioApps/Prospectiva\\_de\\_Adopcion\\_de\\_Dispositivos\\_Moviles](http://www.observatorioapps.com/es/ObservatorioApps/Prospectiva_de_Adopcion_de_Dispositivos_Moviles) >.

PHP: HYPERTEXT PREPROCESSOR. "Introduction to SQLite". [Internet]. Disponible en: < [http://talks.php.net/show/sqlite\\_jan/1](http://talks.php.net/show/sqlite_jan/1) >. [Citado en 2015-06-22].

PIDOCO GMBH. "Pidoco - a GUI prototyping tool". [Internet]. Disponible en: < <https://pidoco.com> >. [Citado en 2015-06-23].

PIEADADE, M.B.; SANTOS, M.Y. "Student Relationship Management: Concept, practice and technological support. Engineering Management Conference, 2008. IEMC Europe 2008.

QUORA. "What should I use for Android development, Android Studio or Eclipse?". [Internet]. Disponible en: < <https://www.quora.com/Should-I-use-Android-Studio-or-Eclipse-for-my-first-Android-app> >. [Citado en 2015-07-02].

SAITO, TARO L. BITBUCKET, ATLISSIAN. "SQLite JDBC Driver". [Internet]. Disponible en: < <https://bitbucket.org/xerial/sqlite-jdbc> >. [Citado en 2015-06-22].

SHPIITALNIK, GREGORY. CODEPROJECT. "Android: Ready to use simple directory chooser dialog with new folders creation". [Internet]. Disponible en: <

< <http://www.scorchworks.com/Blog/simple-file-dialog-for-android-applications/> >. [Citado en 2015-06-23].

SOURCEFORGE. "A Java API to read, write, and modify Excel spreadsheets". [Internet]. Disponible en: < <http://sourceforge.net/projects/jexcelapi/files/> >. [Citado en 2015-06-22].

SQLITE. "About SQLite". [Internet]. Disponible en: < <https://www.sqlite.org/about.html> >. [Citado en 2015-06-22].

SQLITE. “Atomic Commit In SQLite”. [Internet]. Disponible en: < <http://www.sqlite.org/atomiccommit.html> >. [Citado en 2015-06-22].

SQLITE. “How SQLite Is Tested”. [Internet]. Disponible en: < <http://www.sqlite.org/testing.html> >. [Citado en 2015-06-22].

SQLITE. “Limits In SQLite”. [Internet]. Disponible en: < <http://www.sqlite.org/limits.html> >. [Citado en 2015-06-22].

ST. JOHN, REX. “Android Studio vs. Eclipse: What You Need To Know”. [Internet]. Disponible en: < <https://www.airpair.com/android/android-studio-vs-eclipse> >. [Citado en 2015-07-23].

SZETO, EDMUND. LINARO. “Optimizing SQLite database in Android”. [Internet]. Disponible en: < <http://www.linaro.org/blog/android-blog/optimizing-sqlite-database-android> >. [Citado en 2015-06-22].

TICAY RIVAS, JAIME ROBERTO. “La arquitectura de Android”. [Internet]. Disponible en: < <http://catedratelefonica.ulpgc.es/blog/arucas/2012/10/18/la-arquitectura-de-android> >. [Citado en 2014-10-25].

THE VERGE. “Android: A visual history”. [Internet]. Disponible en: < <http://www.theverge.com/2011/12/7/2585779/android-history> >. [Citado en 2014-10-25].

WIKIPEDIA, THE FREE ENCYCLOPEDIA. “Adobe Fireworks”. [Internet]. Disponible en: < [https://en.wikipedia.org/wiki/Adobe\\_Fireworks](https://en.wikipedia.org/wiki/Adobe_Fireworks) >. [Citado en 2015-06-23].

WIKIPEDIA, THE FREE ENCYCLOPEDIA. “Android Studio”. [Internet]. Disponible en: < [https://en.wikipedia.org/wiki/Android\\_Studio](https://en.wikipedia.org/wiki/Android_Studio) >. [Citado en 2015-08-11].

WIKIPEDIA, THE FREE ENCYCLOPEDIA. “JetBrains”. [Internet]. Disponible en: < <https://en.wikipedia.org/wiki/JetBrains> >. [Citado en 2015-08-14].

ZENG, B., FENG, S., & ZHANG, J. (2010). Web-Based Class Management System Using E-commerce Technology. 2010 Third International Symposium on Electronic Commerce and Security.

## ANEXOS

### ANEXO A. HEURÍSTICOS DE BRUEGGE Y DUTOIT

#### a) Identificación de actores

##### **¿Cuáles grupos de usuarios son apoyados por el sistema para realizar su trabajo?**

El docente es el único usuario de la aplicación, no se contemplan grupos de usuarios. Esto lleva a que ClassApp sea una aplicación móvil que no requiere el concepto de usuario y contraseña. No se necesita realizar una validación en un dispositivo que es de carácter privado (Smartphone).

##### **¿Cuáles grupos de usuarios realizan funciones secundarias, como el mantenimiento y la administración?**

ClassApp trabaja con un solo usuario y las labores de mantenimiento se realizaron durante las etapas de desarrollo. Durante el tiempo de uso, no existe un rol que se encargue del mantenimiento ni la administración.

##### **¿Interactuará el sistema con algún sistema de HW o SW externo?**

No. ClassApp solo interactúa con su base de datos SQLite, y con las capas de la arquitectura Android (por medio de clases de Java). Toda la interacción ocurre dentro del ámbito del dispositivo.

#### b) Identificación de escenarios

##### **¿Cuáles son las tareas que el actor desea que realice el sistema?**

El actor (docente) debe tener la posibilidad de:

- Crear nuevos grupos (importar) sin tener que ingresar manualmente a cada estudiante,
- Tomar asistencia a cada grupo,
- Enviar correos grupales o individuales,
- Enviar mensajes vía WhatsApp,
- Exportar la asistencia de un grupo (Microsoft Excel),

- Exportar las notas de un grupo (Microsoft Excel),
- Crear trabajos para cada grupo con su respectivo porcentaje,
- Calificar trabajos de cada grupo,
- Crear recordatorios asociados a cada grupo,
- Crear comentarios asociados a cada grupo,
- Realizar un reset de fábrica,
- Generar backups,
- Editar la mayor cantidad de elementos que sea posible,
- Restaurar desde un backup.

### **¿Qué información consulta el actor?**

El docente podrá consultar quiénes son los integrantes de cada grupo, tener cierta información estadística de los estudiantes (promedios, asistencia, desempeño), revisar asistencia grupal, revisar las notas grupales, revisar los comentarios existentes, revisar el listado de trabajos y comparar los tiempos de ejecución de sentencias SQLite por tres métodos diferentes (perfil de desarrollador).

### **¿Quién crea esos datos?**

Los datos son ingresados en su totalidad por el docente. Es recomendable que se importe como mínimo un grupo para que ClassApp pueda funcionar.

### **¿Se pueden modificar o eliminar?, ¿Quién lo hace?**

Sí, en la sección “Edición” es posible editar y eliminar la mayoría de datos que se hayan ingresado al sistema. Existen algunas restricciones debido a que ciertos datos son utilizados como índices en la base de datos SQLite y se debe mantener la integridad referencial en la medida de lo posible. El encargado de estas tareas es el docente.

### **¿Qué cambios externos necesita informar el actor al sistema?, ¿Con qué frecuencia?, ¿Cuándo?**

El docente debe proporcionarle al sistema un listado de estudiantes cada vez que desee crear un grupo. El formato que maneja ClassApp contiene: periodo académico, nombre del docente, grupo, nombre de la asignatura y su código, nombre del estudiante, código del estudiante, correo institucional del estudiante y código de la carrera del estudiante.

Igualmente, se actualiza el periodo actual cada vez que se importa un grupo. Estas situaciones generalmente se presentan una sola vez al iniciar el semestre.

**¿Cuáles eventos necesita el actor que le informe el sistema?, ¿Con cuánta latencia?**

Es deseable que el sistema le informe al docente si los datos se almacenaron, editaron o eliminaron correctamente. Igualmente, se informará al actor sobre las implicaciones de eliminar un elemento, para que tenga la posibilidad de cancelar el comando. También es necesario informar al actor sobre los resultados de la importación y exportación de archivos. Al no tratarse de operaciones complejas, se espera que el tiempo de respuesta entre la ejecución del comando y la notificación sea rápido (no deberá superar 2 segundos).

**c) Identificación de requerimientos no funcionales**

Cuando se desarrolla un software surgen también una serie de aspectos que, si bien no están relacionados con el comportamiento funcional del sistema, son importantes al momento de la interacción usuario-software. Estos aspectos son visibles para el usuario y tienen una relación directa con la usabilidad. El modelo FURPS+ [45] propone que los requerimientos se agrupen en las categorías: usabilidad, confiabilidad, rendimiento y soportabilidad. Está basado en el modelo FURPS propuesto originalmente por Grady y Caswell [46].

Categoría	Pregunta guía
<p><b>Usabilidad</b></p>	<p><b>¿Cuál es el nivel de experticia del usuario?</b>            ClassApp está orientado a docentes universitarios y las actividades a las cuales brinda asistencia son cotidianas para ellos. No requiere un conocimiento avanzado de informática. Requiere, por otra parte, experiencia con el manejo de pantallas táctiles, ya que se ejecuta en un smartphone.</p> <p><b>¿Cuáles estándares de interfaz son familiares al usuario?</b></p>

	<p>La aplicación contiene elementos tradicionales de interfaz, tales como input de texto, botones convencionales, textos descriptivos, entre otros. Los nuevos elementos para el usuario están asociados a la interfaz gráfica de Android, tales como el componente ListView, el ActionBar y el calendario.</p> <p>Es posible que existan algunos conflictos debido a que el usuario podría confundir imágenes convencionales con botones activos, pero el usuario tiende a acostumbrarse rápidamente a la simbología usada.</p> <p><b>¿Qué documentación se le debería proveer al usuario?</b></p> <p>Es necesario indicarle al usuario cómo navegar por las funciones de ClassApp, qué acciones realiza cada una, qué formatos de archivo se reciben para importar (y qué campos deben tener), qué formatos de archivo se generan al exportar (y con qué características internas) y un breve ejemplo de creación de elementos (grupos, asignaturas, estudiantes, periodos, docentes, etc).</p>
<p><b>Confiabilidad</b></p>	<p><b>¿Qué tan confiable, disponible y robusto debería ser el sistema?</b></p> <p>Las aplicaciones para Android poseen un alto grado de confiabilidad al ser desarrolladas en cualquiera de sus lenguajes nativos (Java, C, C++), como es el caso de ClassApp.</p> <p>No se requieren cálculos de alta exigencia en el procesamiento, entonces no es necesario que la aplicación sea robusta. En general la aplicación debe estar disponible según la demanda del usuario, la cual no debería ser mayor a dos horas por sesión docente.</p> <p><b>¿Se puede reiniciar el sistema en caso de una falla?</b></p> <p>ClassApp no se ejecuta de forma continua como otros tipos de</p>

	<p>software, por ende, el usuario se limita a ejecutar la aplicación y a cerrarla cuando ha terminado sus actividades.</p> <p>En este caso, el reinicio se refiere a que la aplicación se ejecuta de nuevo luego de sufrir un fallo. En dado caso, no se perderán datos, ya que ClassApp trabaja con transacciones SQLite en instantes concretos durante la ejecución.</p> <p><b>¿Qué cantidad de datos puede perder el sistema?</b></p> <p>Los datos que intervienen en los procesos de ClassApp se almacenan en una base de datos SQLite y todo se maneja con transacciones. La posibilidad de pérdida de datos es muy baja, debido a que los datos se almacenan solamente cuando el usuario da la orden explícita para ello.</p> <p>Para futuras versiones sería deseable realizar pruebas en donde se provoque una falla intencional justo al momento de ejecutar el comando RawQuery de las transacciones. De esa forma, este tipo de incidentes estaría en el ámbito SQLite.</p> <p><b>¿Cómo debería el sistema manejar las excepciones?</b></p> <p>Dentro de ClassApp los datos que generan excepciones están gestionados mediante bloques try-catch de Java. Las excepciones más frecuentes están asociadas a SQLite, IO (creación de archivos) y JXL (al abrir o cerrar archivos). Se aplicó una cantidad considerable de tiempo de desarrollo a capturar estas situaciones, entre otros motivos, porque el lenguaje Java lo exige.</p> <p><b>¿Existen requerimientos de seguridad del sistema?</b></p> <p>ClassApp se ejecuta para un solo usuario por dispositivo, por lo tanto no requiere de un inicio de sesión autenticado.</p> <p><b>¿Existen requerimientos de protección del sistema?</b></p>
--	---

	<p>No, ya que la información que interviene en las actividades no se considera confidencial, no hay requerimientos de disponibilidad permanente y los fallos por concurrencia son nulos.</p>
<p><b>Rendimiento</b></p>	<p><b>¿Cómo debe responder el sistema?</b></p> <p>Las tareas críticas están asociadas a la velocidad de almacenamiento en la base de datos. Es en ese contexto en donde se requiere una velocidad que el usuario pueda percibir.</p> <p>Las actividades que registran mayor latencia son**:</p> <ul style="list-style-type: none"> <li>- Importar grupo (procesar xls, crear objetos y almacenar datos)</li> <li>- Tomar asistencia (al momento de almacenar cada ítem)</li> <li>- Calificar (al momento de almacenar notas)</li> </ul> <p>**Prueba realizada con MotoG (primera generación)</p> <p>CPU: Quad-core 1.2 GHz Cortex-A7</p> <p>GPU: Adreno 305</p> <p>Tiempo empleado: entre 1 y 2 segundos.</p> <p><b>¿Qué cantidad de tareas de usuario se vuelven en algún momento críticas?</b></p> <p>Solo las tareas de almacenamiento (creación y edición).</p> <p><b>¿Cuántos usuarios concurrentemente se debe soportar?</b></p> <p>ClassApp se ejecuta de manera local en el dispositivo móvil, soportando un usuario. No existen problemas de concurrencia.</p> <p><b>¿Qué tan grande es el almacenamiento de datos típicos de sistemas comparables?</b></p> <p>El modelo de datos de ClassApp es relativamente simple, y los productos similares en el mercado ofrecen pocas funcionalidades adicionales. Por ello, se asume que el almacenamiento es pequeño (ver sección “2.2. Estado del arte”).</p>

	<p><b>¿Cuál es la peor latencia aceptada por los usuarios?</b></p> <p>Un tiempo de almacenamiento superior a 2 segundos puede incomodar al usuario promedio.</p>
<p><b>Soportabilidad, mantenimiento y portabilidad</b></p>	<p><b>¿Qué extensiones previstas se tienen para el sistema?</b></p> <p>ClassApp es desarrollada por el autor del proyecto y por tanto, las posibles adiciones se plantean en la sección de recomendaciones.</p> <p><b>¿Quién realiza el mantenimiento del sistema?</b></p> <p>Se realizará el mantenimiento al sistema por parte del autor del proyecto, en los momentos del proceso de desarrollo que se consideren más convenientes.</p> <p><b>¿Existen planes para portar el sistema a diferentes ambientes de software o hardware?</b></p> <p>No. El proyecto solamente abarca dispositivos móviles basados en Android, Smartphones y Tablets.</p>
<p><b>Implementación</b></p>	<p><b>¿Existen limitaciones en la plataforma hardware?</b></p> <p>Android está diseñado para aprovechar la arquitectura de procesadores ARM y ejecutar aplicaciones con el menor consumo posible de CPU. ClassApp no implementa actividades de alto consumo de CPU, por lo tanto, el hardware de procesamiento no implica restricciones. Por otro lado, las dimensiones de la pantalla del dispositivo son una limitante para la presentación gráfica de ClassApp, influyendo en la diagramación y el aprovechamiento del espacio visual.</p> <p>No se recomienda el uso de ClassApp en dispositivos Android Wear, o en general, dispositivos con tamaño reducido de pantalla.</p>

	<p><b>¿Hay limitaciones impuestas por el equipo de pruebas?</b></p> <p>La única condición en ClassApp es, que como mínimo, exista un grupo creado previamente. Esta medida permite que las funciones principales se ejecuten sin contratiempos.</p>
<b>Interfaz</b>	<p><b>¿Debe el sistema interactuar con algún sistema existente?</b></p> <p>No. Todos los procesos y actividades se llevan a cabo dentro de la misma aplicación, excepto la sección “Enviar mensajes”, la cual hace llamados a WhatsApp y al cliente de correo que tenga instalado el dispositivo. Las interacciones con la máquina virtual son controladas automáticamente por Android.</p> <p><b>¿Cómo son los datos exportados/importados al sistema?</b></p> <p>Los datos de importación y exportación deberán encontrarse en formato Microsoft Excel (.xls). Es un formato frecuente en ambientes académicos y es óptimo para estructuras de datos donde el elemento base sea un registro con campos.</p>
<b>Operación</b>	<p><b>¿Quién maneja la ejecución del sistema?</b></p> <p>ClassApp es una aplicación que ejecuta el docente. No se requiere un perfil específico para ello.</p>
<b>Empaquetamiento</b>	<p><b>¿Quién instala el sistema?</b></p> <p>El docente es quien instala la aplicación. ClassApp se encuentra en Google Play Store utilizando como palabra clave “classapp”. Google Play Store requiere que el usuario tenga una cuenta de Google o un correo en Gmail para poder acceder y descargar la aplicación.</p> <p>Si el usuario dispone del archivo APK, puede realizar la instalación sin tener que ir a Google Play Store.</p> <p><b>¿Cuántas instalaciones se tienen previstas?</b></p> <p>No definido. ClassApp se ejecuta de manera local, lo cual no</p>

	<p>interfiere con la cantidad de instalaciones.</p> <p><b>¿Existen restricciones de tiempo para la instalación?</b></p> <p>Ninguna en particular. Se empleará el tiempo que le toma a Android instalar una aplicación.</p>
<p><b>Legal</b></p>	<p><b>¿Cómo debe ser licenciado el sistema?</b></p> <p>Este proyecto se liberará de forma gratuita, tanto el APK como el código fuente. El público es libre de realizar modificaciones y mejoras dando crédito a los autores.</p> <p><b>¿Existen fallas de responsabilidad asociadas con fallas del sistema?</b></p> <p>No. Los elementos que puede controlar el usuario no tienen comunicación directa con las capas de bajo nivel de Android. Para producir una falla en el sistema se requiere tener acceso como superusuario, es decir, que el dispositivo se encuentre “rooteado”. Además, se requiere un medio que permita enviar comandos maliciosos al sistema, tal como una consola.</p> <p><b>¿Hay alguna regalía por derechos de licencia que incurra por el uso de algoritmos específicos?</b></p> <p>Ninguna. En ClassApp no se utilizan técnicas ni algoritmos especiales (propietarios / con patente). Se utilizan algunas librerías a las cuales se les otorga su crédito como lo solicitan los autores originales (Java Excel API, MPAndroidChart y la clase SimpleFileDialog.java).</p>

## ANEXO B. FORMATO ESPECIAL DE GRUPOS

Durante las etapas de desarrollo de ClassApp se consideraron varias opciones para acceder a los datos de los grupos de estudiantes. Lo que se tenía claro, es que la opción elegida debería permitir:

- El almacenamiento de grupos de estudiantes por lotes (automáticamente),
- Que fuera posible editar la información del grupo antes de ser importada y,
- Que la solución elegida fuera lo más flexible posible en términos de informática.

Se exploraron bases de datos remotas MySQL, orígenes de datos ODBC, WebServices e incluso datos en formato JSON. Estos métodos quedaban enlazados de forma rígida a la estructura de la base de datos que se estuviera utilizando, y no brindaban la suficiente flexibilidad que se requería para que ClassApp fuera empleada en cualquier entidad educativa.

Finalmente, se decidió realizar la implementación utilizando documentos de Microsoft Excel (.xls) almacenados localmente. Los documentos cuentan con información de estudiantes, asignatura, grupo, docente y periodo académico. Estos datos aparecen en dos ámbitos: dentro de las celdas del documento y dentro del nombre del archivo.

A la plantilla con la cual se elabora este documento se le denominó “Formato especial de grupos”. Se suministra el siguiente enlace para su descarga:

<https://drive.google.com/open?id=0B-C5De7aqYogQXJzcWVNckU0Yk0>

TEXT0\_CODIGOMATERIA\_CODIGOGRUPO\_ANIO\_SEMESTRE.xls - LibreOffice Calc

ASIGNATURA	(NOMBRE DE LA MATERIA)	PROFESOR	(NOMBRE DEL PROFESOR)	
ORD.	CODIGO	NOMBRE ESTUDIANTE	PROGRAMAACAD.	CORREO ELECTRONICO
1	(codigo estudiante)	(apellidos y nombres del estudiante)	(codigo carrera)	(correo del estudiante)
2	(codigo estudiante)	(apellidos y nombres del estudiante)	(codigo carrera)	(correo del estudiante)
3	(codigo estudiante)	(apellidos y nombres del estudiante)	(codigo carrera)	(correo del estudiante)
4	(codigo estudiante)	(apellidos y nombres del estudiante)	(codigo carrera)	(correo del estudiante)
...	...	...	...	...
(n-3)	(codigo estudiante)	(apellidos y nombres del estudiante)	(codigo carrera)	(correo del estudiante)
(n-2)	(codigo estudiante)	(apellidos y nombres del estudiante)	(codigo carrera)	(correo del estudiante)
(n-1)	(codigo estudiante)	(apellidos y nombres del estudiante)	(codigo carrera)	(correo del estudiante)
(n)	(codigo estudiante)	(apellidos y nombres del estudiante)	(codigo carrera)	(correo del estudiante)

### Formato especial de grupos - ClassApp

Para el caso particular de la Universidad Industrial de Santander, es posible utilizar directamente el formato de Excel que genera el sistema de la Dirección de Admisiones y Registro Académico. Este formato contiene la información necesaria y no requiere de ninguna modificación.

lista\_estudiantes\_21870\_D1\_2015\_1 [Read-Only] [Compatibility Mode] - Microsoft Excel (Product Activation Failed)

ASIGNATURA	GERENCIA DE INFORMATICA I	GRUPO D1			
ORD.	CODIGO	NOMBRE ESTUDIANTE	PROGRAMA ACAD.	CORREO ELECTRONICO	OBSERVACIONES
1	1234501	AMADO JAIMES CRISTIAN LEONARDO	11	nombre.apellido@correo.uis.edu.co	
2	1234502	CAMARGO MEDINA EDWIN EDUARDO	11	nombre.apellido@correo.uis.edu.co	
3	1234503	CAMARGO RUEDA JORGE LEONARDO	11	nombre.apellido@correo.uis.edu.co	
4	1234504	CHAPARRO MEDINA RONALD AUGUSTO	11	nombre.apellido@correo.uis.edu.co	
5	1234505	CORTES RUEDA LUIS ANTONIO	11	nombre.apellido@correo.uis.edu.co	
6	1234506	DELGADO MEJIA SANTIAGO	11	nombre.apellido@correo.uis.edu.co	
7	1234507	ESPARZA USME CYNTHIA CAROLINA	11	nombre.apellido@correo.uis.edu.co	
8	1234508	FERRIN SARMIENTO DIEGO ARMANDO	11	nombre.apellido@correo.uis.edu.co	
9	1234509	GALVEZ CASANOVA JEFERSSON ANDRES	11	nombre.apellido@correo.uis.edu.co	
10	1234510	GARZON ROBLES ANYI KATHERINE	11	nombre.apellido@correo.uis.edu.co	
11	1234511	GARZON VARGAS MARIA JULIANA	11	nombre.apellido@correo.uis.edu.co	
12	1234512	GOMEZ PARRA PEDRO ANDRES	11	nombre.apellido@correo.uis.edu.co	
13	1234513	GONZALEZ CORONADO JULIAN ANDRES	11	nombre.apellido@correo.uis.edu.co	
14	1234514	GUERRERO CORZO JULIAN ALBERTO	11	nombre.apellido@correo.uis.edu.co	

## Formato: Universidad Industrial de Santander

Luego de descargar el formato especial de grupos, se procede a personalizar su contenido y nombre de archivo, de la siguiente manera:

### **a) Datos dentro del documento (filas y columnas)**

Los datos dentro del documento (ubicación y contenido de filas y columnas) deben contener información con el siguiente estándar:

B5: Nombre de la materia/asignatura.

B6: Nombre del profesor.

A9: Orden del estudiante en la lista – incremental: 1,2,3, ... ,n.

B9: Código del estudiante

C9: Apellidos y nombres del estudiante – En esta columna lo importante es que los nombres y apellidos estén en la MISMA CELDA. No importa el orden “apellidos+nombres” o “nombres+apellidos”.

D9: Código de la carrera – Para el caso de los colegios, puede usarse cualquier combinación de caracteres que sea constante. Ej.: “Instituto Caldas”.

E9: Correo electrónico del estudiante.

### **b) Datos en el nombre del archivo**

El nombre de archivo del formato también debe contener información con el siguiente estándar:

TEXTO\_CODIGOMATERIA\_CODIGOGRUPO\_ANIO\_SEMESTRE.xls

**¡ATENCIÓN!**: el nombre de archivo NO PUEDE contener ESPACIOS. Por otra parte, el caracter guión bajo “\_” debe mantenerse, ya que es utilizado por ClassApp para segmentar la cadena de texto y obtener cada ítem de información. También debe conservarse el orden de cada ítem, de lo contrario la información se almacenará en ClassApp de manera errónea y no podrá utilizar la aplicación.

“TEXTO”: es una cadena libre. Puede contener cualquier combinación de caracteres. Generalmente se le asigna la palabra ‘listado’.

“CODIGOMATERIA”: las instituciones educativas acostumbran asignar códigos a las asignaturas. Este texto puede contener cualquier combinación de números y letras que represente un código identificador.

“CODIGOGRUPO”: al igual que las asignaturas, los grupos también suelen ser identificados con algún tipo de código. Este texto puede contener cualquier combinación de números y letras, aunque generalmente consiste de dos o tres caracteres.

“ANIO” (AÑO): corresponde al año actual en formato AAAA (Ej.: “2016”).

“SEMESTRE”: es un valor entero correspondiente a “1” o “2”. Si se desea usar en instituciones educativas de primaria o secundaria, basta con usar el caracter “1”.

Ejemplos de nombres de archivo válidos:

- “listado\_12345\_A1\_2016\_1.xls”
- “colegioABC\_678\_B2\_2015\_1.xls”
- “universidad\_hk900\_C88\_2015\_2.xls”

Ejemplos de nombres de archivo INVÁLIDOS:

- “listado 12345 A1 2016 1.xls”
- “678\_1255\_B2\_colegioABC 1.xls”

– “C88 universidad hk900\_\_20152.xls”

Al cumplirse los anteriores requisitos, el formato está listo para ser almacenado manualmente en la carpeta “/Downloads” del dispositivo móvil, a donde apuntará ClassApp en el momento en que se ingrese al módulo “Importar grupo”.