

**UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS**

**Facultad 7**



**Trabajo de Diploma para optar por el título de  
“Ingeniero en Ciencias Informáticas”.**

**Registro de Asistencia y evaluaciones para dispositivos móviles.**

**Autoras:**

Osleydi Quesada Pérez  
Zaida Cajigal La Rosa

**Tutor:** MSc. Reynel Fals de Pedro  
**Co-Tutor:** Ing. Yeinier Ferrás Cecilio

**La Habana, Cuba  
Junio de 2013  
“Año 55 de la Revolución”**

*Pensamiento:*



*“Los hombres son como los  
astros, que unos dan luz de sí  
y otros brillan con la que  
reciben”*

*José Martí.*

*Dedicatoria:*

*A mi mamá y a mi papá porque a ellos les debo todo lo que soy, por haberme apoyado siempre con mis estudios.*

*A mi hermana del alma, por haberme dado su apoyo incondicional en todo momento y siempre ha estado presente cuando lo necesito.*

*A mi novio por ser el sustento y alentarme en todo momento.*

*Osleydi.*

*Quiero dedicarle este trabajo en especial a mi mamá Arelis y a mi papá Fernando, porque me han dado todo y más de lo que ha estado a su alcance y siempre me han apoyado, guiándome y aconsejándome en todo momento, a ellos les debo todo lo que soy. A mi hermano Fernando, mi futuro esposo Gilberto y mis amigas que se me quieren mucho.*

*Zaida.*

## AGRADECIMIENTOS

---

*Agradecimientos:*

*Osleydi*

*A mi mamá, por haberme convertido en la persona que soy y estar presente siempre.*

*A mi papá, por darme tanto apoyo y estar pendiente en todo momento de mis estudios y mis resultados.*

*A mi hermana del alma Oslaida por su cariño y apoyo en todos los momentos.*

*A mi tía Basilia y a mi prima Berta por su cariño y estar siempre comprometidas conmigo y mis estudios.*

*A mi novio Andy por compartir conmigo cada momento y alentarme en los momentos difíciles.*

*A nuestro tuto y cotutor por guiarnos en el desarrollo de la tesis.*

*Al profesor Duany por su gran ayuda, por documentarnos y enseñarnos del tema.*

*Al profesor Yovannys por apoyarnos también y corregirnos nuestros errores.*

*A mi compañera Zaida por sobrellevar conmigo toda esta responsabilidad, y juntas obtener este maravilloso resultado.*

*A todos los profesores que de una manera u otra contribuyeron en la formación de mis capacidades.*

*A todas mis amistades que también aportaron en determinado momento a la elaboración de este proyecto.*

*A la Revolución y al comandante Raúl, por darme la posibilidad de estudiar en esta extraordinaria universidad.*

*Muchísimas gracias...*

*Agradecimientos:*

*Zaida*

*Quiero agradecer a mis padres Arelis y Fernando por guiarme hacia el camino correcto, brindarme amor, educación, confianza, dedicación y desear tanto mi bienestar. Gracias a ellos he podido alcanzar este sueño.*

*A mi hermano Fernando, por ser uno de los más grandes tesoros de mi corazón, lo adoro y espero que al día siga mis pasos.*

*A mi novio Gilberto por ser tan paciente conmigo y brindarme todo su amor y su cariño en los buenos y malos momentos.*

*A mis amigos incondicionales Leanni, Mileisis, Digmaris, Lidisbel, Annabellis y Yulia por estar presentes y ofrecerme su apoyo y comprensión.*

*Un agradecimiento especial a Yosvannys y Yoandris que sin su ayuda no hubiera podido realizar varias tareas de este trabajo.*

*A nuestro tutor Reynel y cotutor Yeinier por estar presente en la elaboración de nuestra tesis.*

*A mi compañera de tesis Osleydi, por confiar en mí y aceptarme tal y como soy.*

*A todos aquellos que me desean el bien y que de una forma u otra me ayudaron a ser mejor cada día.*

*A mi familia por apoyarme en todo momento y estar siempre pendiente de mí.*

*A mis compañeros de la UCI y a los de siempre, por compartir tantos momentos.*

*A la Revolución y al compañero Fidel Castro Ruz por hacer realidad esta universidad de excelencia.*

*Por último, le quiero agradecer también, a los obstáculos que han pasado por mi vida. Ellos son los que me han hecho fuerte, perseverante y finalmente, triunfadora.*

*Muchas gracias a todos...*

**Datos de Contacto**

**Tutor:** Reynel Fals de Pedro

**Correo:** [rfals@uci.cu](mailto:rfals@uci.cu)

**Título:** Máster en Ciencias.

**Área:** Facultad 7. Dpto. de Programación.

**Co-Tutor:** Yeinier Ferrás Cecilio

**Correo:** [yferras@uci.cu](mailto:yferras@uci.cu)

**Título:** Ing. en Ciencias Informáticas

**Área:** Facultad 7. Dpto. de Programación.

### **Resumen**

El avance en la telefonía móvil permite no solo las llamadas telefónicas, o mensajes de texto, sino que propicia al usuario una variedad de servicios para un mayor uso de la misma, lo que sitúa a la tecnología móvil en lugares cimeros en la sociedad. Las tecnologías móviles y su integración en el sector de la educación muestran beneficios potenciales de su uso.

La Universidad de las Ciencias Informáticas se encuentra inmersa en la tarea de fortalecer el proceso de seguimiento y control de los estudiantes. El retraso en la entrega de un registro impreso y el limitado número de evaluaciones que se pueden registrar de un estudiante, hace necesario buscar alternativas digitales. El sistema de gestión universitaria de la Universidad de las Ciencias Informáticas (UCI) resuelve de manera parcial este problema, pero en la facultad 7 se dificulta al acceso al Sistema Gestión Universitaria debido a la disponibilidad de una computadora que debe portar el profesor o estar disponible en el local de clases. Por este motivo se ve afectado el registro de las asistencias y evaluaciones frecuentes de los estudiantes.

Por las razones anteriormente descritas el objetivo de este trabajo es desarrollar un registro de asistencia y evaluaciones cualitativas y/o cuantitativas para dispositivos móviles con el Sistema Operativo Android en la Universidad de las Ciencias Informáticas. La aplicación está implementada sobre la plataforma Android, en el lenguaje Java, a través del Eclipse, además se tuvo en cuenta todos los pasos y métodos que nos ofrece el Proceso Unificado de Desarrollo del Software. Al software desarrollado fue evaluado mediante la prueba de Caja Negra; se le realizaron elaboraron los casos de prueba para la detección de la no conformidades.

Palabras Claves: Registro de Asistencia, Celulares, Telefonía Móvil, Dispositivos móviles, Dispositivos de cómputo, Servicio.

### **Abstract**

The advancement in mobile phone allows not only phone calls, or text messages, but one that offers the user a variety of services to more use of it, which puts mobile technology in top rankings in society. Mobile technologies and their integration into the education sector shows potential benefits of its use.

The University of Informatics Sciences (UIS) is immersed in the task of strengthening the monitoring and control process of the students. The delay in the delivery of a printed record and the limited number of evaluations of a student can register, digital alternatives is necessary. The university management system at the University of Informatics Sciences partially solves this problem, but in college 7 hinders access to University Management System due to the availability of a computer to carry with the teacher or available in the classroom. For this reason, is affected logging attendance and frequent assessment of students.

For the reasons described above the goal of this work is to develop a record of attendance and evaluations qualitative and / or quantitative for mobile devices with Android Operating System at the University of Informatics Sciences. La application is implemented on the Android platform in the Java language, through Eclipse, and took into account all the steps and methods offered by the Unified Process Software Development. The developed software was evaluated by testing Black Box; were performed test cases developed for the detection of non-compliance.



## ÍNDICE DE CONTENIDOS

<b>INTRODUCCIÓN .....</b>	<b>14</b>
<b>CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA .....</b>	<b>21</b>
1.1 APRENDIZAJE MÓVIL .....	21
1.2 SISTEMAS EXISTENTES RELACIONADOS CON LA INVESTIGACIÓN .....	22
1.3.1 ESCENARIO INTERNACIONAL .....	22
1.3 METODOLOGÍA, LENGUAJE Y HERRAMIENTAS DE DESARROLLO .....	25
1.3.1 LA PLATAFORMA ANDROID .....	25
1.3.1.1 ARQUITECTURA .....	25
1.3.2 METODOLOGÍA DE DESARROLLO .....	27
1.3.3 LENGUAJE DE MODELADO UML .....	28
1.3.4 LENGUAJE DE PROGRAMACIÓN JAVA .....	29
1.3.5 LIBRERÍA JUNIT-5.0.5 .....	29
1.3.6 VISUAL PARADIGM .....	30
1.3.7 HERRAMIENTAS DE DESARROLLO .....	30
<b>CONCLUSIONES DEL CAPÍTULO .....</b>	<b>31</b>
<b>CAPÍTULO II. CARACTERÍSTICAS DEL SISTEMA .....</b>	<b>33</b>
INTRODUCCIÓN .....	33
2.1 PROPUESTA DEL SISTEMA .....	33
2.2 MODELO DE DOMINIO .....	39
2.2.1 DESCRIPCIÓN DE LOS OBJETOS DEL DOMINIO .....	40
2.3 ESPECIFICACIONES DE REQUISITOS DEL SISTEMA .....	40
2.3.1 REQUERIMIENTOS FUNCIONALES .....	40
2.3.2 REQUERIMIENTOS NO FUNCIONALES .....	41
2.4 MODELO DE CASOS DE USO DEL SISTEMA .....	42
2.4.1 DEFINICIÓN DE ACTORES DEL SISTEMA .....	42
2.4.3 DIAGRAMA DE CASOS DE USO DEL SISTEMA .....	49
<b>CONCLUSIONES DEL CAPÍTULO .....</b>	<b>49</b>
<b>CAPÍTULO III: ANÁLISIS Y DISEÑO DEL SISTEMA .....</b>	<b>50</b>
INTRODUCCIÓN .....	50
3.1 MODELO DE ANÁLISIS .....	50
3.1.1 DIAGRAMA DE CLASES DEL ANÁLISIS .....	50
3.2 MODELO DE DISEÑO .....	53
3.2.1 DIAGRAMA DE CLASES DEL DISEÑO .....	53
3.3 DESCRIPCIÓN DE LA ARQUITECTURA .....	56
3.3.1 PATRÓN ARQUITECTÓNICO .....	56
3.3.2 PATRONES DE DISEÑO. PATRONES GRASP .....	59
3.4 MODELO DE DATOS .....	61
3.4.1 MODELO LÓGICO DE DATOS .....	62
3.4.2 MODELO FÍSICO DE DATOS .....	62
3.5 TRATAMIENTO DE EXCEPCIONES .....	63
3.6 SEGURIDAD .....	63
<b>CONCLUSIONES DEL CAPÍTULO .....</b>	<b>63</b>
<b>CAPÍTULO IV: IMPLEMENTACIÓN Y PRUEBA .....</b>	<b>64</b>

# ÍNDICE DE CONTENIDOS

---

INTRODUCCIÓN.....	64
5.1. MODELO DE IMPLEMENTACIÓN.....	64
5.1.1. DIAGRAMA DE COMPONENTES.....	64
5.1.2. DIAGRAMA DE DESPLIEGUE.....	64
5.2. PRUEBAS.....	65
6.2.1 PRUEBA DE CAJA NEGRA.....	65
6.2.1.1 CASOS DE PRUEBA.....	65
6.2.1.2 RESULTADOS DE LAS PRUEBAS DE CAJA NEGRA.....	68
<b>CONCLUSIONES DEL CAPÍTULO.....</b>	<b>69</b>
<b>CONCLUSIONES GENERALES.....</b>	<b>70</b>
<b>RECOMENDACIONES.....</b>	<b>71</b>
<b>REFERENCIAS BIBLIOGRÁFICAS.....</b>	<b>72</b>
<b>BIBLIOGRAFÍA.....</b>	<b>75</b>
<b>ANEXOS.....</b>	<b>82</b>
<b>GLOSARIO DE TÉRMINOS.....</b>	<b>83</b>

**ÍNDICE DE TABLAS**

TABLA 1 DEFINICIÓN DEL ACTOR. ....	42
TABLA 2 DESCRIPCIÓN DEL CUS_CARGAR_GRUPOS_DOCENTES .....	43
TABLA 3 DESCRIPCIÓN DEL CUS_GESTIONAR_ASISTENCIA.....	45
TABLA 4 DESCRIPCIÓN DEL CUS_GESTIONAR_EVALUACIÓN.....	46
TABLA 5 DESCRIPCIÓN DEL CUS_SELECCIONAR OPCIONES_DE_EVALUACIÓN.....	47
TABLA 6 DESCRIPCIÓN DEL CUS_GENERAR_REPORTE. ....	48
TABLA 7 DISEÑO DE CASO DE PRUEBA CU CARGAR GRUPOS DOCENTES.....	66
TABLA 8 DISEÑO DE CASO DE PRUEBA CU GENERAR REPORTE .....	67
TABLA 9 DISEÑO DE CASO DE PRUEBA CU GESTIONAR ASISTENCIA.....	68

## ÍNDICE DE FIGURAS

FIGURA 1 CUOTA EN EL MERCADO DE SO PARA DISPOSITIVOS MÓVILES 2012.....	15
FIGURA 2 ACTIVACIÓN DE DISPOSITIVOS ANDROID DIARIAMENTE A NIVEL MUNDIAL . .....	15
FIGURA 3 ARQUITECTURA DE ANDROID OS. ....	26
FIGURA 4 INTERFAZ SELECCIONAR GRUPO.....	34
FIGURA 5 INTERFAZ ADICIONAR GRUPO.....	34
FIGURA 6 INTERFAZ TOMAR ASISTENCIA DE ESTUDIANTES. ....	35
FIGURA 7 INTERFAZ MODIFICAR ESTADO DE ASISTENCIA. ....	36
FIGURA 8 INTERFAZ CONSULTAR PORCIENTO DE ASISTENCIA.....	36
FIGURA 9 INTERFAZ REGISTRAR EVALUACIONES (CUANTITATIVA). ....	37
FIGURA 10 INTERFAZ REGISTRAR EVALUACIONES (CUALITATIVA). ....	37
FIGURA 11 INTERFAZ MODIFICAR EVALUACIÓN. ....	38
FIGURA 12 INTERFAZ GENERAR REPORTE DEL REGISTRO. ....	38
FIGURA 13 INTERFAZ DEL REPORTE.....	39
FIGURA 15 MODELO DE DOMINIO. ....	40
FIGURA 16 DIAGRAMA DE CU DEL SISTEMA. ....	49
FIGURA 17 DIAGRAMA DE CLASES DEL ANÁLISIS CU GESTIONAR ASISTENCIA. ....	51
FIGURA 18 DIAGRAMA DE CLASES DEL ANÁLISIS CU GESTIONAR EVALUACIÓN.....	51
FIGURA 19 DIAGRAMA DE CLASES DEL ANÁLISIS CU CARGAR GRUPO. ....	52
FIGURA 20 DIAGRAMA DE CLASES DEL ANÁLISIS CU CONSULTAR ASISTENCIA. ....	52
FIGURA 21 DIAGRAMA DE CLASES DEL ANÁLISIS CUGENERAR REPORTE. ....	53
FIGURA 22 DIAGRAMA DE CLASES DEL DISEÑO DEL CU GESTIONAR EVALUACIÓN. ....	54
FIGURA 23 DIAGRAMA DE CLASES DEL DISEÑO DEL CU GESTIONAR ASISTENCIA. ....	54
FIGURA 24 DIAGRAMA DE CLASES DEL DISEÑO DEL CU CARGAR GRUPO DOCENTE.....	55
FIGURA 25 DIAGRAMA DE CLASES DEL DISEÑO DEL CU GENERAR REPORTE.....	55
FIGURA 26 PATRÓN MODELO-VISTA-CONTROLADOR. ....	57
FIGURA 27 CONCEPCIÓN MODELO-VISTA-CONTROLADOR DEL SISTEMA. ....	58
FIGURA 28 EJEMPLO PATRÓN CONTROLADOR.....	59
FIGURA 29 EJEMPLO PATRÓN EXPERTO.....	60
FIGURA 30 EJEMPLO PATRÓN CREADOR. ....	60
FIGURA 31 EJEMPLO PATRÓN BAJO ACOPLAMIENTO. ....	61
FIGURA 32 DIAGRAMA DE CLASES PERSISTENTES. ....	62
FIGURA 33 DIAGRAMA ENTIDAD-RELACIÓN. ....	62

## ***ÍNDICE DE FIGURAS***

---

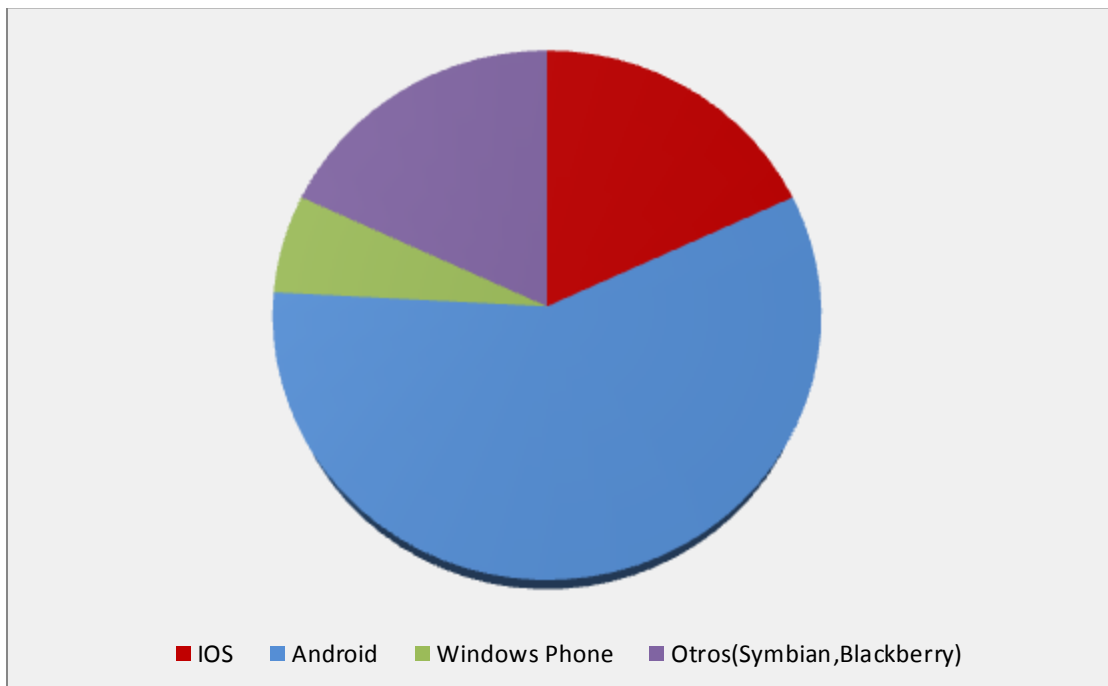
FIGURA 34 DIAGRAMA DE COMPONENTES DEL SISTEMA.....	64
FIGURA 35 DIAGRAMA DE DESPLIEGUE. ....	65
FIGURA 36 RESULTADOS DE LAS PRUEBAS DE CAJA NEGRA.....	69

## Introducción

La evolución constante de las Tecnologías de la Información y las Comunicaciones (TICs) (1) manifiesta un gran avance en diversas esferas de la sociedad tales como: la salud, la educación, las comunicaciones, el comercio, la industria entre otras. Entre dichas esferas se destaca el campo de comunicación, que se ha visto beneficiado de la utilización cada vez más incrementada de los dispositivos móviles.

Los dispositivos móviles (2) tienen mayor potencia de cómputo, el crecimiento del desarrollo de aplicaciones para estos aparatos representa un gran avance en los tiempos presentes y la mayoría tienen soporte para aplicaciones con lenguajes de programación web actuales como HTML5, CSS, JavaScript. También es posible utilizar una aplicación nativa para un dispositivo móvil, la cual está íntegramente programada en el entorno de desarrollo específico para determinado sistema operativo. Al mismo tiempo, entre los tipos más evolucionados se encuentran los teléfonos inteligentes y ordenadores personales en tableta; los cuales normalmente se equipan con un sistema operativo (3) para manejar, controlar y actualizar sus aplicaciones y datos.

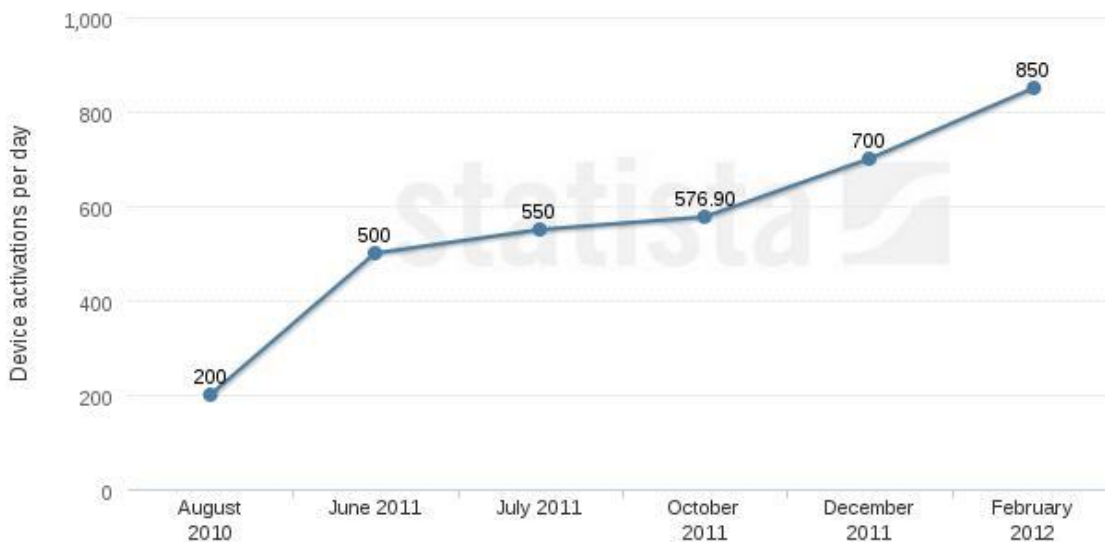
Existen varios sistemas operativos para dispositivos móviles, entre los que se destacan: GNU/Linux, Sistema Operativo BlackBerry (BlackBerry OS-BlackBerry Operating System), Windows Mobile, iPhone OS y Android. Dichos sistemas son más simples que los empleados en las computadoras, están más orientados a la conectividad inalámbrica, los formatos multimedia para móviles y las diferentes maneras de introducir información en ellos. Asimismo Android, plataforma creada por Google, se ha convertido en el sistema operativo móvil más utilizado, ampliamente difundido con una cuota de mercado cercana al 57% (4). En las siguientes gráficas se observa la estadística dicha cuota y el crecimiento de la utilización de Android sobre todos los restantes.



**Figura 1** Cuota en el mercado de sistemas operativos para dispositivos móviles 2012.

**Fuente:** Elaboración propia.

Daily Android OS activations in selected months between 2010 and 2012 (in 1,000)



**Figura 2** Activación de dispositivos Android diariamente a nivel mundial.

**Fuente:** Software Libre para Dispositivos Móviles. (5)

El sistema operativo Android inicialmente se utilizó en teléfonos, sin embargo, a partir de la versión 2.0 se extendió a ordenadores portátiles, tabletas, relojes de pulsera, auriculares y otros dispositivos de diversos modelos que han inundado el mercado, dentro de los más utilizados se encuentran: Sony Ericsson Xperia X10, HTC Legend, HTC Desire, HTC Wildfire, Motorola Milestone, NexusOne, Samsung Galaxy, Orange Boston, Huawei U8110 Selina / Ivy y LG GW620.

En Cuba la utilización de la tecnología móvil ha crecido considerablemente durante los últimos años (6). La integración de estas nuevas tecnologías a la educación va en ascenso, enriqueciendo los procesos de enseñanza-aprendizaje (PEA) (7) en universidades del país, lo que viabiliza el aprendizaje y control de estudiantes. El registro docente como técnica o procedimiento de evaluación del PEA refleja el perfil del alumno, guarda un historial con su comportamiento y permite valorar el rendimiento en el aula. Además, admite una evaluación objetiva y se obtienen informaciones valiosas acerca del rendimiento del estudiante; siendo está una forma de realizar cálculos estadísticos en el aula y del recinto educativo de manera que facilita la toma de decisiones.

El retraso en la entrega de un registro impreso y el limitado número de evaluaciones que se pueden registrar de un estudiante, hace necesario buscar alternativas digitales. Las universidades cubanas no están exentas de esta realidad y se han sumado, en algunos casos, a la creación de sistemas para la gestión académica que permitan brindar un servicio especializado que permita registrar dicha información de los estudiantes.

El Sistema de Gestión Académica de la Universidad de Ciencias Informáticas (UCI) persigue contribuir a mejorar los procesos relacionados con las carreras, la gestión del personal, el control y la planificación docente de manera que permite que la información gestionada, sea actualizada, fiable y accesible. Dicho sistema resuelve de manera parcial el registro de la asistencia y evaluaciones de los estudiantes. Hasta el presente no brinda la posibilidad de incluir las evaluaciones frecuentes en tiempo real, lo que trae consigo que el profesor no pueda registrar dichas evaluaciones en el propio turno de clases. También se ha hecho uso de aplicaciones como Excel o algún editor de texto por parte de los profesores para almacenar los datos de los estudiantes.

En la facultad 7 de la UCI se dificulta al acceso al Sistema Gestión Universitaria debido a disponibilidad de una computadora que debe portar el profesor o estar disponible en el local de clases. Por este motivo se ve afectado el registro de las asistencias y evaluaciones diarias de los estudiantes.



A partir de la problemática anteriormente expuesta se define como **problema científico**:

¿Cómo facilitar el proceso de gestión de la información relacionada con la asistencia y evaluación de los estudiantes en la Universidad de las Ciencias Informáticas?

El **objeto de estudio** se enmarca en el proceso de gestión de la información relacionada con la asistencia y evaluación de los estudiantes en el período docente, y el **campo de acción** en el proceso de gestión de la información relacionada con la asistencia y evaluación de los estudiantes en el período docente a partir de la telefonía móvil en la Universidad de las Ciencias Informáticas.

Se trazó como **objetivo general** de la investigación desarrollar una aplicación informática que controle la asistencia y evaluaciones cualitativas y/o cuantitativas para dispositivos móviles con el Sistema Operativo Android en la Universidad de las Ciencias Informáticas.

Para alcanzar dicho objetivo se planteó desarrollar las siguientes **tareas de investigación**:

1. Estudiar y analizar los sistemas informáticos que registran la asistencia y las evaluaciones de los estudiantes en el campo de la telefonía móvil.
2. Realizar un estudio de las tecnologías a utilizar.
3. Modelar el sistema, utilizando las herramientas y artefactos que nos brindan UML (Unified Modeling Language) y RUP (Rational Unified Process, o Proceso Unificado de Desarrollo de Software).
4. Desarrollar una herramienta para el registro de asistencia y evaluaciones para dispositivos móviles con sistema operativo Android.
5. Validar de la propuesta.

**Población:** La universidad posee un total de 1567 profesores distribuidos por las 7 facultades, de ellos 189 son docentes dirigentes y 1378 son docentes técnicos; lo que permite la atención de toda la matrícula, independientemente de la facultad en que se encuentre registrado cada profesor docente. La población sujeta a estudio está conformada por 1120 profesores con teléfonos celulares.

**Unidad:** La unidad de estudio utilizada en la investigación son los registros de asistencia y evaluación de los estudiantes en la universidad, los cuales dan a conocer cómo se comporta el proceso de gestión de la información de los estudiantes relacionada con la asistencia y evaluación, registrada por los profesores.

**Muestra:** Se escogió la facultad 7 con una cifra de 111 profesores en total. Se tomó una muestra representativa de 20 teléfonos celulares mediante un recorrido por departamentos, aulas y proyectos productivos de la facultad para ser objeto de la presente investigación, entre ellos se encuentran:

- Departamento Docente de Ciencias Sociales y Humanidades.(4)
- Departamento de Sistemas de Apoyo a la Salud. (proyecto productivo, 3)
- Departamento de Software Médico Imagenológico. (proyecto productivo, 3)
- Vicedecanato de Extensión y Residencia.(2)
- Departamento de Programación e Ingeniería.(4)
- Aulas.(4)

De esta muestra 5 dispositivos contaban con SO Android, lo que representa un 25% del total, estos se tomaron en función del proceso de asistencia y control de los estudiantes en la Universidad de las Ciencias Informáticas.

### **Técnica de muestreo.**

La técnica de muestreo utilizada en la investigación es la “no probabilística”; la cual “...no asegura la probabilidad que tiene cada elemento de la población de ser incluido en la muestra, por lo que no se logra la equiprobabilidad ni la representación de la población en la muestra. (8)

Con el objetivo de asegurar la mayor calidad posible de la información obtenida a partir de la muestra y su correspondencia con la población de origen, fue empleada como técnica de muestreo no probabilística el “muestreo intencional”, porque a través de ella le es posible al investigador, seleccionar los elementos que son representativos o con posibilidades de brindar mayor información dentro de la población.

### **Estrategia de investigación.**

Investigación descriptiva: Su principal objetivo es describir el fenómeno y reflejar lo esencial y más significativo del mismo, sin tener en cuenta las causas que lo originan, para lo que es necesario captar sus relaciones internas y regularidades, así como aquellos aspectos donde se revela lo general. (8)

### **Métodos de Investigación.**

Para llevar a cabo cada una de las tareas de la presente investigación, se utilizó un conjunto de métodos teóricos y dos métodos empíricos.

**Analítico-Sintético:** Este método permitirá conocer los procesos de gestión de la información de las actividades realizadas por los estudiantes de manera informatizada y los problemas que se desenvuelven en el contexto actual, a la vez proporcionarán las vías para resolverlos.

**Histórico-Lógico:** La presente investigación permite realizar un estudio acerca de las dificultades presentadas durante el registro de evaluaciones y asistencia para posibilitar, además, la corrección de los datos con el fin de obtener una solución óptima para el problema.

**Observación:** Se pudo obtener información de cómo se llevaba a cabo el proceso de seguimiento y control de los estudiantes de la universidad, de que actualmente las principales deficiencias están dadas por: el retraso en la entrega de un registro impreso y, el limitado número de evaluaciones que se pueden registrar de un estudiante y además el sistema de gestión universitaria existente no brinda la posibilidad de incluir las evaluaciones frecuentes del estudiante.

**Entrevista:** Se empleó como técnica para obtener datos, mediante el diálogo entre dos personas: El entrevistador "tesista" y el entrevistado "profesor"; se realiza con el fin de obtener información acerca de los dispositivos móviles así como tipo de dispositivo, sistema operativo que utiliza y versión.

El presente documento está estructurado por los capítulos que se relacionan a continuación:

**Capítulo 1.** Fundamentación Teórica para el Registro de Asistencia y Evaluaciones: Se definirán cada uno de los aspectos y conceptos relacionados con el sistema a implementar, así como el análisis de soluciones existentes, las herramientas y lenguajes de programación, para verificar que son aptas para la implementación de la aplicación. Además se puntualizará la metodología para el desarrollo de la misma.

**Capítulo 2.** Características del Sistema para el Registro de Asistencia y Evaluaciones: Se presentan las características generales del sistema, se procederá a realizar la captura de requisitos funcionales y no funcionales a los que se debe dar cumplimiento, se examina los temas referentes a las necesidades para obtener las bases y definir el objeto de automatización y la propuesta de solución que será desarrollada.

**Capítulo 3.** Análisis y Diseño del Sistema para el Registro de Asistencia y Evaluaciones: Se centra principalmente en el análisis y diseño del sistema, para obtener como principal resultado los diagramas de clases vinculados a estos flujos de trabajo. Se detalla además la arquitectura del sistema.

**Capítulo 4.** Implementación y Prueba para el Registro de Asistencia y Evaluaciones: Se hace referencia a las disciplinas de implementación y prueba. Se describe el modelo de implementación utilizado mediante los diagramas de componentes y de despliegue. Se describe el diseño de los casos de prueba y el resultado de las pruebas realizadas.

## **Capítulo 1. Fundamentación Teórica.**

### **1.1 Aprendizaje móvil.**

En el ámbito educativo, la inserción de dispositivos móviles ha permitido una evolución de los sistemas comunicativos, informativos y formativos. Específicamente en el campo del aprendizaje electrónico, se puede observar una evolución de carácter conceptual. Algunas experiencias (9) en este contexto, específicamente en instituciones universitarias, han sido documentadas y explicadas con el fin de desarrollar estrategias metodológicas desde el punto de vista del diseño como del desarrollo y evaluación de estrategias.

La elaboración de herramientas que apoyen al profesorado para la evaluación de alumnos en el aula no es una idea nueva, no obstante, la utilización de las nuevas tecnologías para la realización de estas actividades, es recomendable no solo por el beneficio que se pueda alcanzar en el área de conocimiento en que las personas se encuentren, sino porque abre nuevas posibilidades docentes e investigadoras gracias a la funcionalidad que estas herramientas incorporan. (10)

La generalización de uso y posesión de tecnologías móviles por parte de profesores y alumnos es un proceso consolidado en progreso, donde un sistema educativo con recursos limitados de tecnologías de la información y de la comunicación (TICs), debe intentar sacar el máximo partido de lo que los estudiantes traen a las aulas. (11)

Los dispositivos móviles facilitan la entrega de información específica en determinado contexto y de esta manera posibilitan el aprendizaje y la construcción del conocimiento (12). Estos aparatos en relación a la enseñanza, permiten oportunidades educativas únicas, la herramienta que viabiliza estas posibilidades es el aprendizaje móvil (m-learning–mobile learning).

El aprendizaje móvil consiste en el aprendizaje basado en la tecnología móvil de comunicación (13), y constituye el centro de múltiples investigaciones en diversas instituciones, algunos proyectos de investigación y desarrollo, entre otros relevantes que se pueden mencionar está, The MOBILearn Project (14). Su objetivo principal es la definición de modelos de soporte teóricos y validaciones empíricas para la efectiva enseñanza.

En consecuencia con los resultados alcanzados, se reflexiona acerca de las posibles tendencias del aprendizaje móvil partiendo desde la situación actual y se toma en cuenta la rápida evolución de la tecnología para conservarla inmersa en todos las esferas de la sociedad.

### 1.2 Sistemas existentes relacionados con la investigación.

El mejor resultado de toda investigación se enmarca en los estudios anteriores sobre el problema planteado. Es por esto que a continuación se presentan ejemplos de sistemas semejantes a lo que se propone en dicho trabajo.

#### 1.3.1 Escenario internacional.

Existen sistemas implementados que proponen el uso en las aulas de los dispositivos móviles para aprovechar las ventajas que estos brindan.

A nivel internacional se han desarrollado aplicaciones para dispositivos con el fin de mejorar la gestión académica, la investigación solo se enmarca en aquellas que ubican entre sus funcionalidades la evaluación y/o asistencia de alumnos. A continuación se describen dichos sistemas:

- **“Evaluación y seguimiento del aprendizaje en salas de clases utilizando dispositivos móviles” (2010), Chile (15).** Sistema desarrollado en el Departamento de electrónica, Universidad técnica Federico Santa María. El trabajo propone el uso del equipamiento multimedia de las aulas y los dispositivos móviles cada vez más comunes entre los estudiantes como alternativa.

El sistema anterior utiliza preguntas tipo test presentadas al curso a través de algún sistema de proyección, pizarra o similar; la comunicación inalámbrica de las respuestas por parte de cada estudiante a través de un equipo móvil, y el despliegue de la estadística por parte del profesor. La arquitectura Cliente/Servidor del sistema la componen un programa servidor en el computador del profesor, un sistema de almacenamiento de respuestas, y un programa cargado en el dispositivo móvil.

Se utilizó en dicho programa Java Micro Edition, Java Standard Edition en el servidor, y Bluetooth como protocolo de comunicación inalámbrico. Además de realizar evaluaciones de forma rápida, el sistema permite mantener un registro a largo plazo de éstas para efectos de seguimiento del aprendizaje individual. El prototipo construido muestra un buen desempeño y está preparado para iniciar evaluaciones pilotos con alumnos.

Esta plataforma posee la particularidad de que todos los estudiantes portan de un teléfono para acceder o enviar datos, lo cual no es propicio para la realización del

registro móvil de la presente investigación, debido a que se dificulta disponibilidad de un dispositivo móvil para cada estudiante en el local de clases.

- **Proyecto Gestión de alumnos para el profesorado mediante dispositivos móviles (2008), España (16).** Trata de una aplicación nombrada UXXI-Académico desarrollada en la Universidad de Alcalá y presentado en el III Concurso de Ideas y Proyectos para Personal de Administración y Servicios. Le ofrece al profesorado una herramienta de control online de sus clases y laboratorios, calificaciones, faltas de asistencia, presentación de prácticas y otros factores a tener en cuenta, además realizar una actualización automática y constante de los parámetros del alumnado en las bases de datos Universitarias.

Entre los beneficios del proyecto se encuentran:

- Eliminación del tiempo de transcripción al sistema informático de la información recogida en las clases.
- Eliminación de errores de transcripción de dicha información.
- Sincronización automática entre el sistema de información de la Universidad y el sistema portátil.
- Control del alumnado en las clases más seguro.
- Ahorro de gastos del alumno en cuanto a fotografías.

En el caso de la fotografía no sería necesario que el alumno entregase ninguna más, puesto que el profesorado la tendría en todo momento proveniente de la matrícula del alumno.

- Ahorro de gastos de papelería (fichas del alumno).

Este proyecto presenta similitudes al sistema que se pretende desarrollar, con respecto a ofrecerles a los usuarios una herramienta de control de las actividades docentes mencionadas inicialmente. Dicha solución difiere en que se aspira una aplicación que funcione de manera offline en el dispositivo móvil del profesor, para no presentar problemas de red o inestabilidad de los servicios.

- **Proyecto USMP Mobile (2011), Perú (17).** Utilización de dispositivos móviles para la gestión académica de alumnos y docentes de la Universidad de San Martín de Porres. Tiene la capacidad de hacer consultas a diversas noticias, eventos, acceso a la intranet académica, acceso a las aulas virtuales con conexión directa con el aplicativo, realizar y utilizar geolocalización para ubicar sus aulas, salones, facultades

y departamentos de la institución; y sobre todo, hacer consultas ya sea a sus cursos, modificar su perfil, entre otros. Además posee funcionalidades como:

- Perfil del estudiante/docente
- Listado de cursos
- Horarios
- Calificaciones
- Exámenes

El proyecto USMP Mobile busca mejorar la experiencia de los alumnos, docentes, administrativos con respecto a los servicios que brinda la universidad. Ante ello se busca crear nuevos modelos de servicios orientados a estos dispositivos ya sea smartphones o tablets.

Dicha aplicación se relaciona en cuanto a la funcionalidad referente a las calificaciones. Difiere en que funciona de manera online y emplea sistemas de geolocalización.

- **Integración de una aplicación móvil a una intranet. Caso: Toma de Asistencia Estudiantil (2005)** (18). Tesis para optar por el título de Ingeniero Informático desarrollada en la Universidad Católica del Perú.

Se desarrollaron dos sistemas, uno para un sistema Web y el otro para dispositivos móviles. Para dichos sistemas se utilizó JDeveloper 9.0 de Oracle para el primer caso y NetBeans 4.0 con el Mobility Pack para el segundo caso; adicionalmente para el último caso se hizo uso de tecnologías inalámbricas para la sincronización de información.

En ambas versiones de la aplicación se permite la verificación y modificación de la asistencia de los alumnos asignados al docente como también hacer comentarios individuales por sesión o alumno y calificar su participación. Adicionalmente, el docente puede crear sesiones en caso sean necesarias. Por medio de la aplicación Web, los alumnos podrán consultar su asistencia así como ver un listado de la asistencia de sus compañeros de clase, porcentajes de asistencia e inasistencia. Los docentes también tienen acceso a esta información con la diferencia que además podrá ver la asistencia particular de cada alumno.



Entre los beneficios del sistema esperados también en la presente investigación se encuentran la reducción del tiempo en la toma de la asistencia, disponibilidad de la información para ser consultada en cualquier momento y lugar, independencia de una PC y una mejor organización en el almacenamiento y procesamiento de estadísticas.

En contraste con esto existe la gran dependencia de la parte Web del sistema de lo que depende la disponibilidad de los servidores del Campus Virtual y del manejador de base de datos de la institución.

Después de analizar y estudiar algunos de los principales sistemas similares desarrollados por otros países del mundo, se pudo apreciar que las funcionalidades implementadas para la realización de dichos sistemas varían en dependencia del plan de estudio o sistema de evaluación del centro escolar. Algunos de dichos procedimientos se tomaron como referencia para apoyar la implementación de la herramienta donde el profesor es quien toma el control sobre el sistema para almacenar la información de los estudiantes.

En el contexto nacional no se ha encontrado ninguna aplicación informática para el registro de asistencias y/o evaluaciones de estudiantes que emplee la tecnología móvil.

### **1.3 Metodología, lenguaje y herramientas de desarrollo.**

#### **1.3.1 La plataforma Android.**

Android es un sistema operativo (OS) más extensamente considerado, como una plataforma emergente para dispositivos móviles que está obteniendo una gran aceptación. Su núcleo está basado en una versión modificada de Linux v2.6 (19) y su código fuente está disponible para la comunidad de desarrolladores para su libre adaptación. La empresa desarrolladora de esta plataforma es Google Inc., que con su proyecto Android Open Source Project (AOSP) (20) tiene como objetivo evolucionar este sistema proporcionando una plataforma de excelencia para el desarrollo de aplicaciones que mejore las experiencias de usuario de dispositivos móviles. Este proyecto está auspiciado por la Open Handset Alliance (21).

##### **1.3.1.1 Arquitectura.**

La plataforma Android está constituida por una serie de capas software (22) que cooperan entre sí para obtener una funcionalidad específica y dando lugar a lo que se denomina a software stack (20). Más específicamente, la plataforma Android está compuesta por un sistema operativo, software específico heterogéneo, denominado middleware (23) y, una serie de programas clave o esenciales.

El siguiente diagrama muestra componentes principales de la arquitectura Android:

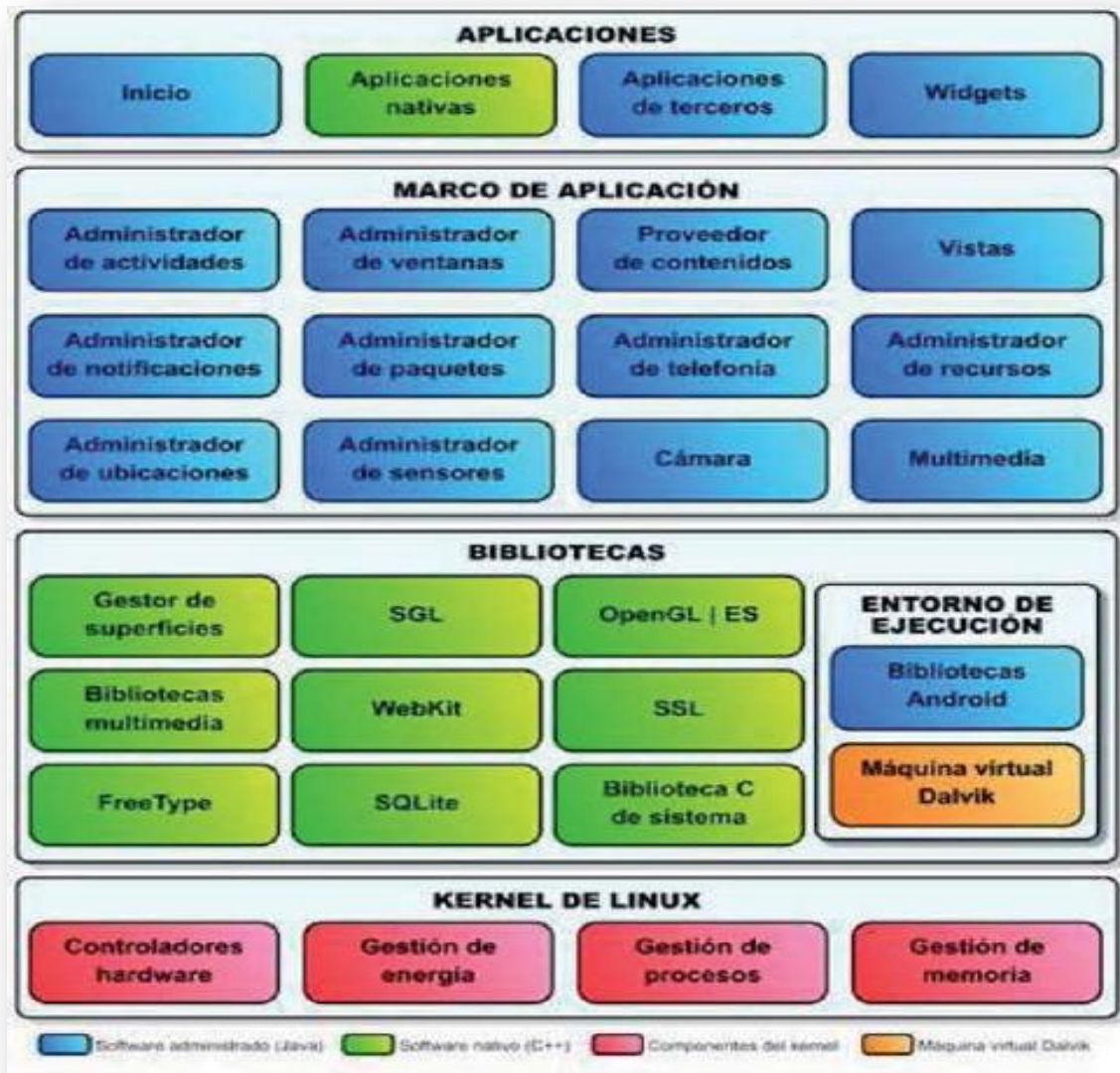


Figura 3 Arquitectura de Android OS.

Fuente: Informe Android (23).

Para realizar aplicaciones sobre la plataforma Android se instala el plugin ADT y el SDK en el IDE (Integrated Development Environment o Entorno de Desarrollo Integrado) de desarrollo. Un SDK de Android contiene las librerías, el emulador, documentación, ejemplos de código, tutoriales y una versión de la plataforma. Se usa el lenguaje de programación Java. (24)

Actualmente existen seis fabricantes: Dell, HTC, Kyocera, LG, Motorola, y Samsung, produciendo teléfonos inteligentes que usan el sistema operativo Android.

### **1.3.2 Metodología de desarrollo.**

En la actualidad no existe una metodología general para desarrollar cualquier proyecto de desarrollo de software, sino que estas tienen prácticas específicas que las hacen óptimas o no depender de las características del proyecto.

#### **XP**

La Programación Extrema es una metodología ligera de desarrollo de software que se basa en la simplicidad, la comunicación y la realimentación o reutilización del código desarrollado.

Las cuatro variables de esta metodología son:

1. Coste: Máquinas, especialistas y oficinas.
2. Tiempo: Total y de Entregas.
3. Calidad: Externa e Interna
4. Alcance: Intervención del cliente.

XP surgió como respuesta y posible solución a los problemas derivados del cambio en los requerimientos. Se plantea como una metodología a emplear en proyectos de riesgo. Con ella se aumenta la productividad. (25)

#### **RUP**

El Proceso Unificado Racional (Rational Unified Process en inglés, habitualmente resumido como RUP) es un proceso de desarrollo de software y junto con el Lenguaje Unificado de Modelado UML, constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos. (26)

RUP provee un enfoque disciplinado en la asignación de tareas y responsabilidades dentro de una organización de desarrollo. Su meta es asegurar la producción de software que satisfaga las necesidades de los usuarios finales, dentro de un calendario y presupuesto predecible. RUP divide el proceso en cuatro fases (Inicio, Elaboración, Construcción y Transición), dentro de las cuales se realizan varias iteraciones en número variable según el proyecto y en las que se hace un mayor o menor hincapié en las distintas actividades. Su ciclo de vida se caracteriza por estar dirigido por casos de uso, iterativo e incremental y centrado en la arquitectura.

### **Metodología seleccionada (RUP)**

Luego del estudio de las metodologías anteriores se decide utilizar RUP principalmente porque el proyecto a desarrollar será implementado orientado a objeto, es una de las metodologías de desarrollo de software más exitosas en la actualidad utilizada para un mejor análisis, implementación y documentación de los proyectos. La metodología consiste en dividir el proceso de desarrollo en ciclos, teniendo un producto final al culminar cada una de ellos, estos a la vez se dividen en fases que finalizan con un hito donde se debe tomar una decisión importante.

Aunque las aplicaciones Android no tienen control de su ciclo de vida y deben estar preparadas para su terminación en cualquier momento; debido a que cada aplicación se ejecuta en su propio proceso y el runtime de Android y gestiona el proceso de cada aplicación y por extensión de cada Actividad que contenga (27), se emplea una de las metodologías de desarrollo de software para guiar el desarrollo del mismo.

### **1.3.3 Lenguaje de modelado UML.**

Lenguaje de Modelado Unificado (Unified Modeling Language) es un lenguaje gráfico para especificar, construir, visualizar y documentar todas las disciplinas de un proyecto informático: desde el análisis con los casos de uso, hasta la implementación y configuración mediante los diagramas de despliegue. (28) UML es además un método formal de modelado, lo que aporta las como ventajas un mayor rigor en la especificación, permite realizar una verificación y validación del modelo realizado y se pueden automatizar determinados procesos y permite generar códigos a partir de los modelos y viceversa.

UML permite:

- Especificar cuáles son las características de un sistema antes de su construcción.
- Construir sistemas diseñados a partir de modelos especificados.
- Visualizar gráficamente un sistema de manera que otros puedan entenderlo.
- Documentar los elementos gráficos del sistema desarrollado para futuras revisiones.
- Verificar y validar el modelo realizado.
- Generar código a partir de los modelos y viceversa.

### 1.3.4 Lenguaje de programación Java.

Para programar para Android se pueden emplear diversos lenguajes de programación entre ellos Java, C#, C, .NET o Visual Basic. El lenguaje seleccionado en la presente investigación es Java debido a que es mucho más sencillo programar empleando el mismo, es el que ha usado Google por defecto y a la vez muy recomendado porque las aplicaciones desarrolladas podrán correr en otros sistemas operativos. Otro criterio de selección del lenguaje fue el estudio y empleo en la Práctica Profesional durante la carrera de Ingeniería Informática, por lo que se tiene dominio del mismo.

Es un lenguaje orientado a objetos, eso implica que su concepción es muy próxima a la forma de pensar humana. El lenguaje en sí mismo toma mucha de su sintaxis de C y C++ y es el único soportado por la inmensa mayoría de dispositivos móviles (29). Un SDK de Android emplea por defecto el lenguaje de programación Java para el desarrollo de aplicaciones. (24)

Java también posee otras características muy importantes como son:

- Aprovecha características de la mayoría de los lenguajes modernos evitando sus inconvenientes (en particular los del C++).
- Tiene una gran funcionalidad gracias a sus librerías.
- Seguro: La máquina virtual, al ejecutar el código java, realiza comprobaciones de seguridad.
- Multiplataforma: El mismo código java que funciona en un sistema operativo, funcionará en cualquier otro sistema operativo que tenga instalada la máquina virtual java.
- Indiferente a la arquitectura: Soporta aplicaciones que serán ejecutadas en los más variados entornos de red. Para ello, el compilador de Java genera bytecodes: un formato intermedio indiferente a la arquitectura, diseñado para transportar el código eficientemente a múltiples plataformas de hardware y software. (29)
- En este lenguaje se pueden programar aplicaciones independientes, como con cualquier otro lenguaje de propósito general y applets (pequeñas aplicaciones que se ejecutan en un documento HTML, siempre y cuando el navegador soporte Java). (30)

### 1.3.5 Librería iTest-5.0.5.

Es una librería que te permite generar ficheros pdf, admite editar y modificar las propiedades y las características de dichos archivos (31). Se emplea dicha librería para generar los diferentes reportes en el formato pdf desde el código java de la aplicación.

### **1.3.6 Visual Paradigm.**

Visual Paradigm para UML (VP-UML) es una herramienta Case multiplataforma (Windows/Linux/Mac OS X) que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y genera documentación. (28)

Esta herramienta está diseñada para una amplia gama de usuarios. VP-UML soporta los últimos estándares de la notación UML.

VP-UML 8.0 incluye mejoras como:

- Mayor velocidad en el proyecto abierto.
- Reducción de consumo de recursos para el proyecto.
- Buena integración con IDEs.
- Incluye versiones en español.
- Muy personalizable.
- Soporta múltiples lenguajes de programación.
- Aumenta la producción automática de código (Java), bases de datos y generación de informes.
- Interoperabilidad con modelos UML2 (metamodelos UML 2.x para plataforma Eclipse).
- Multiplataforma.

### **1.3.7 Herramientas de desarrollo.**

Para el desarrollo de aplicaciones en Android, la plataforma dispone de una serie de herramientas y recomendaciones para facilitar las implementaciones; Estas serán las utilizadas para el desarrollo de la aplicación. A continuación se comentan las diferentes herramientas, dispuestas en el orden que deberían instalarse en el computador de desarrollo (22):

**Java SE JDK:** Es el conjunto de recursos Java, o plataforma, que conforman la denominada Java Standard Edition, fundamental para el desarrollo de aplicaciones

Java. Este conjunto de recursos no es propio de la plataforma Android, pero debe ser instalada en el computador para poder desarrollar aplicaciones.

**Eclipse IDE:** Es el entorno de desarrollo recomendado por Android Developers para la implementación de aplicaciones (32). Existe una versión de Eclipse que ya viene preparada para facilitar el desarrollo de aplicaciones en lenguaje Java. Se recomienda la versión Eclipse 3.5 (Galileo) o superior, se empleará Eclipse 4.2 (Juno). También es posible obtener la versión clásica de este IDE a la cual se le debería añadir el módulo JDT para obtener un entorno de trabajo orientado a Java.

**Android SDK:** Es el conjunto de recursos fundamentales que dispone la plataforma Android para el desarrollo de aplicaciones Java. Posteriormente se debe configurar el IDE Eclipse para que localice el SDK de Android y disponga de sus recursos en el entorno de desarrollo. Entre los recursos de que dispone el SDK de Android se encuentra un emulador de dispositivos, para probar los desarrollos sin necesidad de realizar instalaciones sobre un dispositivo físico. (33)

**ADT Plugin para Eclipse (33):** Plugin que extiende las funcionalidades de Eclipse para proporcionar un entorno de desarrollo orientado a Android, dispone de toda una serie de utilidades como: componentes basados en el UI de Android, herramientas de debugger del SDK de Android, generación de aplicación en formato .apk.

**Crear un Android Virtual Device (33):** Es la configuración de dispositivo que será utilizada por el emulador de Android. Mediante AVD es posible definir los característicos hardware y software del dispositivo que se va a emular y para el cuál se quiere ejecutar la aplicación en desarrollo.

### **Conclusiones del Capítulo.**

El incremento en el uso de dispositivos móviles trae aparejada la necesidad de una mayor cantidad de aplicaciones móviles que cubran las necesidades de los usuarios en las diferentes esferas de la sociedad, en particular, la educación.

El análisis del aumento constante de utilización sistema operativo Android en los dispositivos móviles y la cuota en el mercado que ocupa el mismo, conllevaron a emplear dicha plataforma en función del desarrollo del presente registro docente.

## ***FUNDAMENTACIÓN TEÓRICA***

---

A través de la investigación de las soluciones existentes se adquirió el conocimiento necesario para el desarrollo de aplicaciones móviles y se investigó la plataforma de desarrollo Android de Google, se procedió a la implementación de la aplicación Control.

Las técnicas, herramientas y lenguajes consultados según la literatura permitieron seleccionar lo más adecuado en la presente investigación.

La selección de la metodología RUP posibilitará controlar, planificar, organizar y guiar el desarrollo de la plataforma.

El IDE Eclipse y el lenguaje de programación Java facilitan la implementación de la aplicación, constituyen herramientas empleadas por defecto en el desarrollo de aplicaciones móviles Android.



## **Capítulo II. Características del sistema.**

### **Introducción**

En el presente capítulo se permitirá conocer el contexto en el que se sitúa la solución propuesta. Se describirá el proceso que soportará el sistema. Además guiará el desarrollo del software hacia el sistema adecuado, mediante la captura y descripción de las condiciones o capacidades que el sistema debe cumplir y de las propiedades del mismo.

### **2.1 Propuesta del sistema.**

Se propone el desarrollo de un sistema para el proceso de gestión de la información relacionada con la asistencia y evaluación de los estudiantes para dispositivos móviles que utilicen el sistema operativo Android. Se realizó bajo la plataforma Android para dispositivos móviles con dicho sistema operativo.

El proceso comienza cuando el usuario accede a la aplicación de registro Control desde su dispositivo móvil. Se le muestra una interfaz con opciones que indican si desea iniciar el registro, configurar la aplicación o acudir a la ayuda del sistema. Luego el usuario escoge Configurar (debe ser esta opción para cargar los datos del grupo con el que se desea trabajar), se le muestra otra interfaz indicándole la manera de integrar su grupo de estudiantes al sistema. Una vez que el profesor haya hecho su elección, el sistema se encarga de localizar la dirección buscada por el usuario del fichero en formato texto y cargarlo en el sistema.

En la funcionalidad Iniciar se le muestra al usuario las diversas procedimientos que puede realizar en su grupo de estudiantes y seguidamente la interfaz correspondiente a la operación seleccionada, Registrar asistencia, Registrar evolución, Ver/Modificar asistencia, Ver/modificar evaluación o Generar reporte (pdf). Cada vez que el profesor registre información de los estudiantes, estos datos se almacenan de forma estructurada en la base de datos SQLite, de donde posteriormente se extrae dicha información para generar el reporte del corte hasta ese momento.

El resultado del sistema propuesto se evidencia a través de la descripción de las principales funcionalidades desarrolladas:

## CARACTERÍSTICAS DEL SISTEMA

**Interfaz RF1- Importar listado de los grupos de estudiantes:** En esta interfaz el profesor puede cargar los ficheros que tendrá los estudiantes asignados por grupo, con el objetivo de seleccionar el grupo y adicionarlo al sistema.



Figura 4 Interfaz Seleccionar grupo.



Figura 5 Interfaz Adicionar grupo.

**Interfaz RF2- Gestionar asistencia:** En este requisito se muestran las interfaces: Tomar asistencia de estudiantes y Modificar estado de asistencia, cada una cumple su propia función.

**Tomar asistencia de estudiantes:** En esta interfaz el profesor tiene la posibilidad de registrar la asistencia y especificar su estado, esta operación se podrá efectuar el mismo día cuantas veces sea necesario.



Figura 6 Interfaz Tomar asistencia de estudiantes.

**Modificar estado de asistencia:** En esta interfaz el profesor puede modificar el estado de la asistencia en caso de error.



Figura 7 Interfaz Modificar estado de asistencia.

**Interfaz RF3- Consultar porciento de asistencia:** Esta interfaz muestra las asistencias e inasistencia que ha tenido el estudiante, además del porciento de inasistencia de cada estudiante calculado de la siguiente manera (cantidad de inasistencias\*100/cantidad de horas clases).

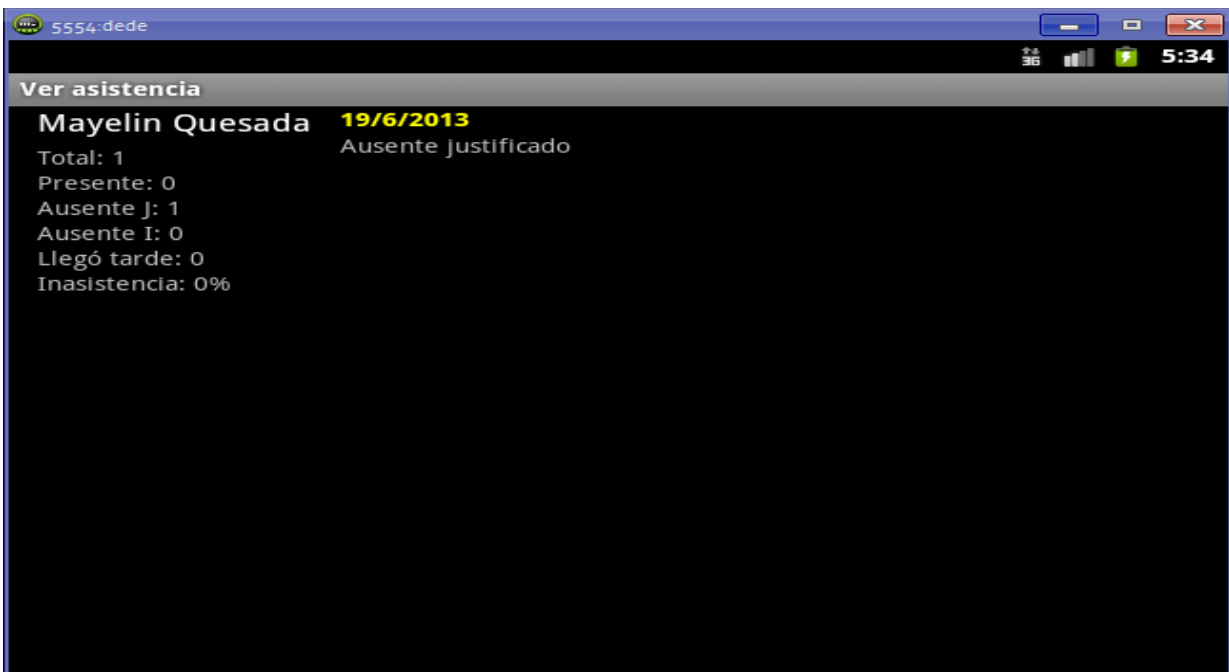


Figura 8 Interfaz Consultar porciento de asistencia.

## CARACTERÍSTICAS DEL SISTEMA

**Interfaz RF4- Gestionar evaluaciones:** En este requisito se muestran las interfaces: Seleccionar opciones de evaluación, Registrar evaluaciones y Modificar evaluaciones, en las cuales cada una desempeña su propia función.

**Seleccionar opciones de evaluación:** Esta interfaz brinda la posibilidad de seleccionar el tipo de evaluación; cuantitativa o cualitativa.

**Registrar evaluaciones:** El usuario registra la evaluación del estudiante según su tipo seleccionado.

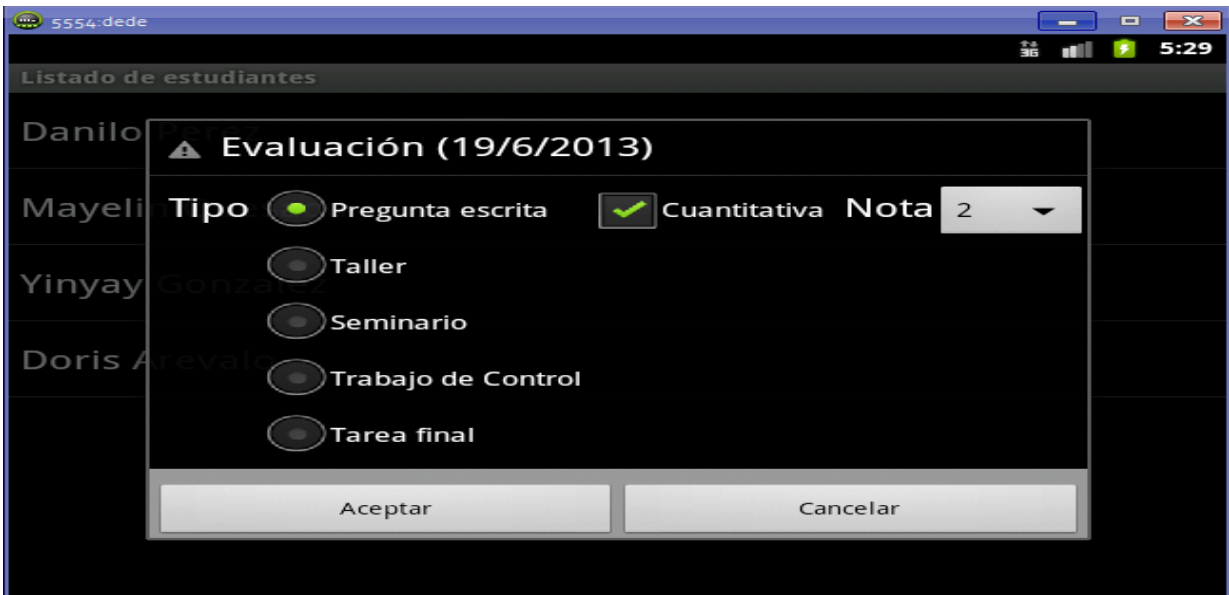


Figura 9 Interfaz Registrar evaluaciones (Cuantitativa).

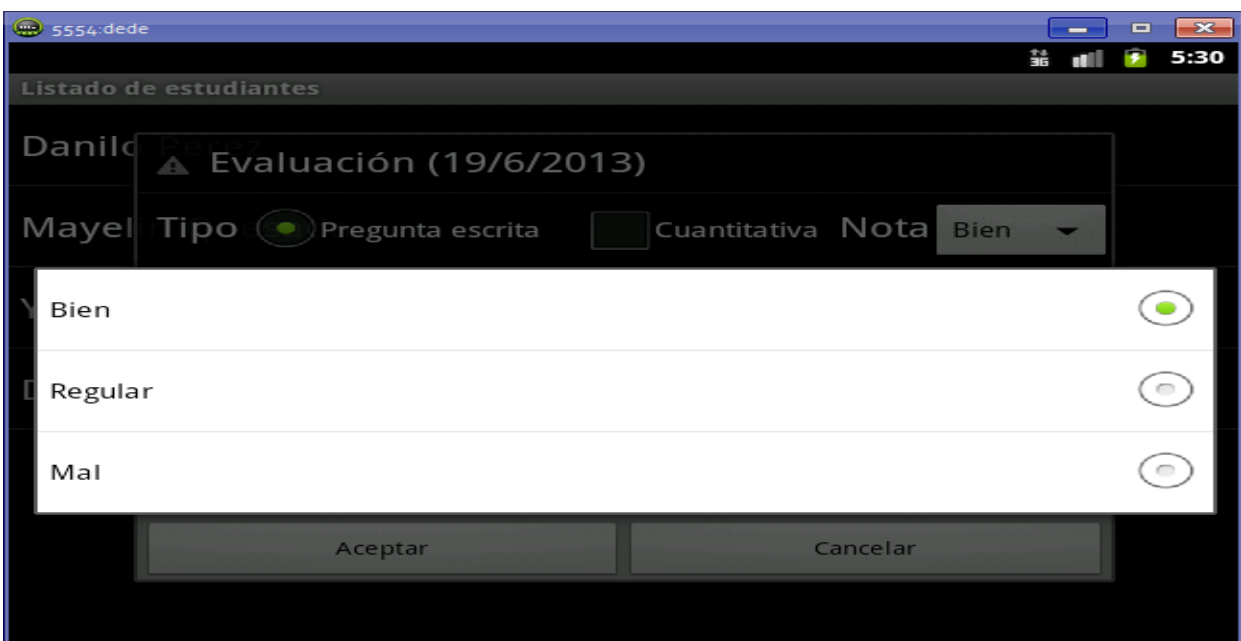


Figura 10 Interfaz Registrar evaluaciones (Cualitativa).

**Modificar evaluaciones:** El profesor puede modificar la nota obtenida por el estudiante.

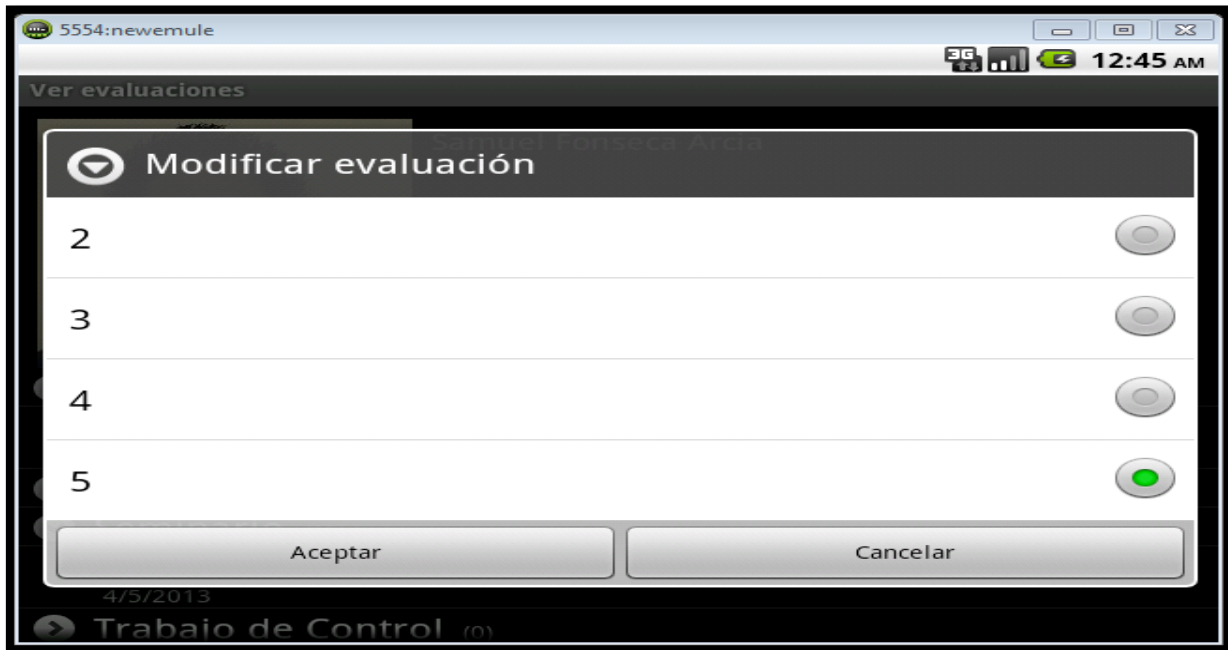


Figura 11 Interfaz Modificar evaluación.

**Interfaz RF5- Generar reporte del registro:** Permite generar reporte de asistencias y evaluaciones del grupo de estudiantes en formato pdf, y luego visualizar dicho reporte.

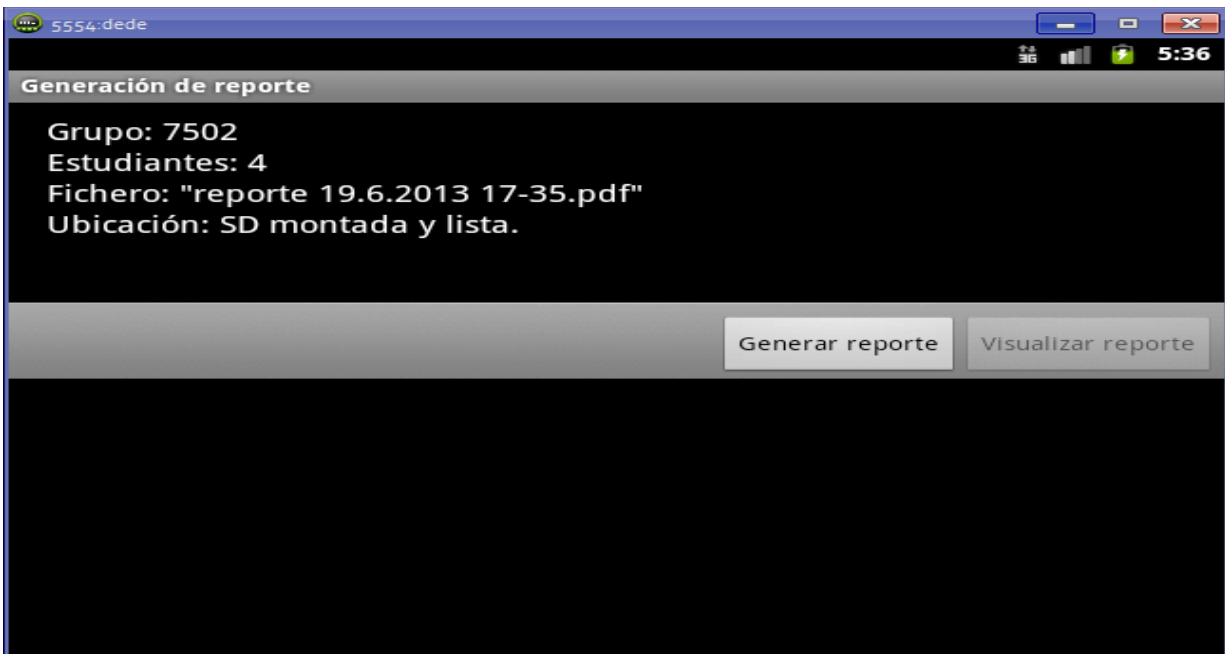
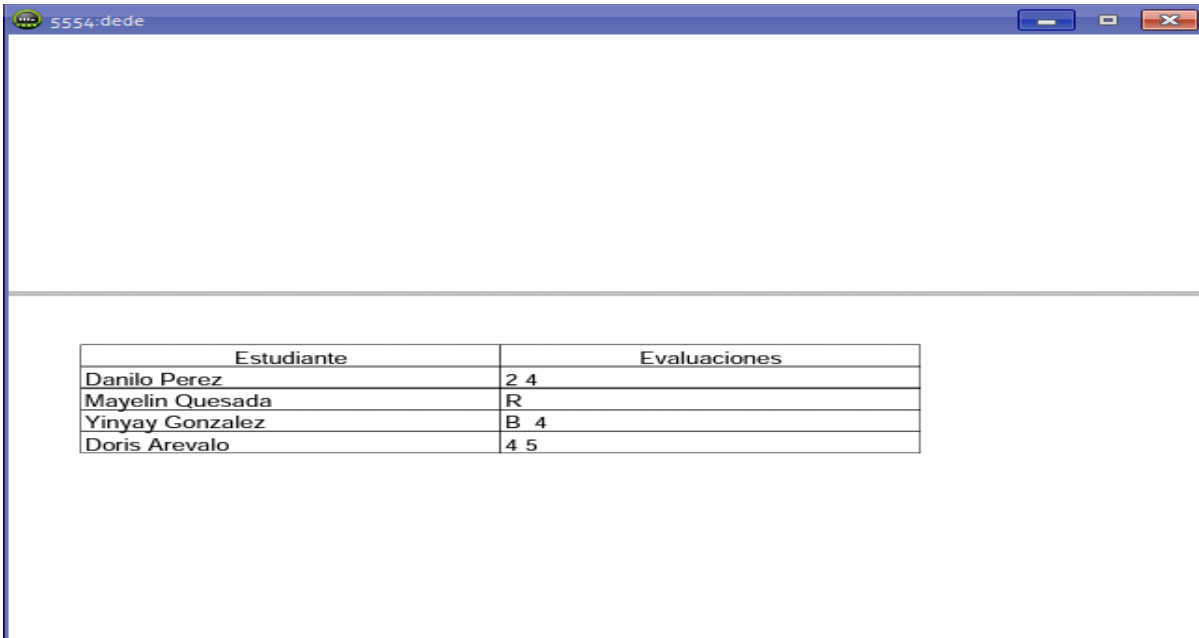


Figura 12 Interfaz Generar reporte del registro.



Estudiante	Evaluaciones
Danilo Perez	2 4
Mayelin Quesada	R
Yinyay Gonzalez	B 4
Doris Arevalo	4 5

Figura 13 Interfaz del reporte.

La plataforma Android además cuenta con las características que se mencionan a continuación.

- La estructura del el sistema operativo se compone de aplicaciones que se ejecutan en un framework Java de aplicaciones orientadas a objetos, todas ellas soportadas por la máquina virtual Dalvik con compilación en tiempo de ejecución.
- SQLite: base de datos para almacenamiento estructurado que se integra directamente con las aplicaciones. En Multimedia: soporte para medios con formatos comunes de audio, video e imágenes planas, soporte para pantalla multitáctil y dependiente del terminal.

## 2.2 Modelo de Dominio.

La aplicación que se desea desarrollar no se delimitan de forma clara los actores, ni los trabajadores, ni los procesos del negocio por lo cual se decide realizar un modelo de dominio el cual es un subconjunto del modelo de objeto del negocio, donde se representan los conceptos y eventos fundamentales que se expresan como clases con la cardinalidad que existe entre ellas.

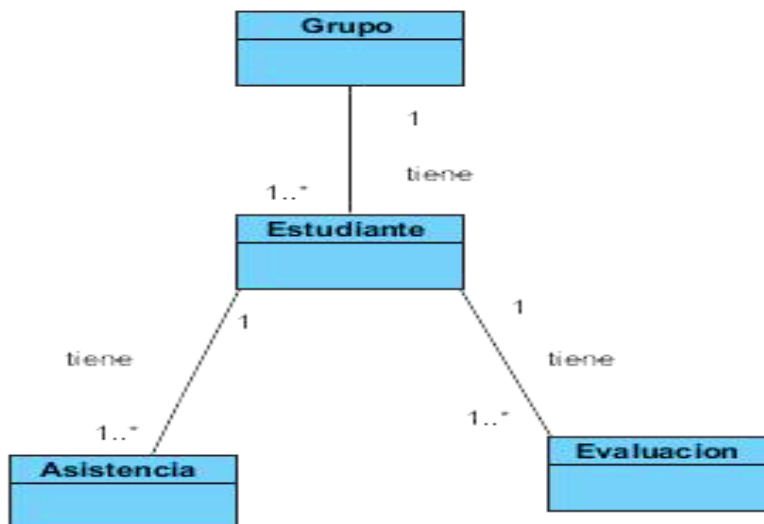


Figura 14 Modelo de dominio.

Fuente: Elaboración propia

### 2.2.1 Descripción de los objetos del dominio.

Grupo: Entidad que tiene el listado de grupos de ese profesor.

Estudiante: Entidad que tendrá los datos del estudiante.

Asistencia: Entidad que contendrá los diferentes tipos de asistencia del estudiante.

Evaluación: Entidad que especificará las diferentes evaluaciones del estudiante.

## 2.3 Especificaciones de requisitos del sistema.

Los requerimientos son propiedades o restricciones determinadas de forma precisa que deben satisfacerse. Los requerimientos o requisitos de un sistema describen los servicios que ha de ofrecer el sistema y las restricciones asociadas a su funcionamiento, se pueden clasificar en: funcionales y no funcionales. A continuación se especifican los requerimientos funcionales y no funcionales sujetos al desarrollo del sistema.

### 2.3.1 Requerimientos funcionales.

Los Requerimiento Funcionales expresan la naturaleza del funcionamiento del sistema (cómo interacciona el sistema con su entorno y cuáles van a ser su estado y funcionamiento). A continuación se enumeran los requerimientos funcionales identificados para el desarrollo del sistema.

1. Cargar datos de grupos docentes.
2. Gestionar asistencia.



- 2.1 Tomar asistencia de estudiantes.
- 2.2 Modificar estado de asistencia.
- 3. Consultar porciento de asistencia.
- 4. Gestionar evaluaciones.
  - 4.1. Seleccionar opciones de evaluación.
  - 4.2. Registrar evaluaciones.
  - 4.3. Modificar evaluaciones.
- 5. Generar reporte del registro.

### **2.3.2 Requerimientos no funcionales.**

Los requerimientos no funcionales van a ser las restricciones sobre el espacio de posibles soluciones. Los requerimientos no funcionales definidos son:

#### **Usabilidad 1.** Finalidad de la aplicación.

El registro para dispositivos móviles debe permitir a los profesores registrar la asistencia y las evaluaciones frecuentes de los estudiantes.

#### **Usabilidad 2.** Tipo de aplicación.

Una aplicación de Escritorio muy fácil de entender. La misma será desarrollada para que los profesores sean capaces de registrar los datos de cada estudiante.

#### **Confiabilidad 1.** Recuperación de la información.

Se podrá recuperar la información almacenada en la base de datos mediante la utilización de las copias de respaldo.

#### **Eficiencia 1.** Tiempo de respuesta.

El tiempo de respuesta debe coincidir con las funcionalidades ejecutadas siendo este el mínimo.

#### **Eficiencia 2.** Optimización de recursos.

## CARACTERÍSTICAS DEL SISTEMA

El sistema probará la disminución de los recursos críticos y el volumen de datos en las peticiones realizadas.

### **Soporte 1. Cantidad.**

La solución debe abarcar la mayor gama posible de dispositivos móviles con sistema Android OS.

### **Restricciones de diseño**

Se utilizará como lenguaje de programación Java, con la plataforma Android.

Estará implementado en la arquitectura Modelo-Vista-Controlador.

## **2.4 Modelo de Casos de Uso del Sistema.**

Los actores del sistema representan personas o sistemas automatizados que interactúan con el sistema. En el sistema propuesto se identificó los siguientes actores:

### **2.4.1 Definición de actores del sistema.**

Actor	Justificación
Profesor	Representa la persona encargada de gestionar los datos de los grupos para la asistencia y evaluación de los estudiantes.

Tabla 1 Definición del actor.

### **2.4.2 Definición de casos de uso del sistema.**

#### **CUS 1 Cargar datos de grupos docentes.**

Objetivo	Cargar datos de grupos docentes
Actores	Profesor
Resumen	El Caso de Uso (CU) inicia cuando el profesor selecciona el grupo docente para realizar las funcionalidades de registrar la asistencia y evaluaciones para determinado turno de clase.
Complejidad	Media
Prioridad	Crítico
Precondiciones	Exista el grupo y alumnos asignados al registro.

## CARACTERÍSTICAS DEL SISTEMA

Postcondiciones	El profesor tiene que tener los permisos para cargar el listado de dicho grupo.	
Flujo de eventos		
	Actor	Sistema
1.	El profesor crea los ficheros de los grupos que atiende y lo guarda en la SD del teléfono. Luego selecciona la opción Adicionar Grupo desde fichero.	2. El sistema verifica si están los datos del grupo cargados.
3.		2.1 El sistema muestra la opción Adicionar Grupo. [Ver Sección 1 "]
Sección 1: "Cargar Datos"		
Flujo básico		
	Actor	Sistema
1.	El profesor accede a la opción Adicionar Grupo desde fichero.	2. El sistema muestra un listado con todos los grupos que el atiende.
3.	El profesor selecciona el grupo.	4. El sistema muestra las funcionalidades a ejecutar y termina el CUS.

**Tabla 2** Descripción del CUS\_Cargar\_grupos\_docentes.

### CUS 2 Gestionar asistencia.

Objetivo	Gestionar la asistencia de uno o varios grupos docentes.	
Actores	Profesor	
Resumen	El Caso de Uso (CU) inicia cuando el profesor selecciona el grupo docente y la opción de registrar la asistencia de un grupo de estudiantes. Se toma la asistencia de un grupo de alumnos para determinado turno de clase.	
Complejidad	Media	
Prioridad	Crítico	
Precondiciones	El profesor debe tener claridad de los grupos que atiende para tomar, modificar o consultar el porcentaje de asistencia.	
Postcondiciones	Se toma la asistencia, se modifican los cambios o se consulta el porcentaje.	
Flujo de eventos		
	Actor	Sistema
1.	El profesor necesita tomar, modificar y consultar por ciento de asistencia.	El sistema verifica si están los

## **CARACTERÍSTICAS DEL SISTEMA**

		datos del grupo cargados.
3.		2.1 El sistema ejecuta alguna de las siguientes acciones: a) Si decide tomar la asistencia: ir a la sección 1 "Tomar asistencia" b) Si decide actualizar la asistencia de un estudiante: ir a la sección 2 "Modificar asistencia".
<b>Sección 1: "Tomar asistencia"</b>		
Flujo básico		
	Actor	Sistema
1.	El profesor accede a la opción Tomar asistencia.	2. El sistema muestra el listado de estudiante.
3.	El profesor selecciona el estudiante.	5. El sistema guarda el tipo de asistencia seleccionada y termina el CUS.
Flujos alternos		
Nº Evento		
	Actor	Sistema
1.		4.1 Se emite un mensaje para que marque el grupo y regresa a la acción 1 de esta sección. 4.2 Si el estudiante tiene su asistencia se muestra un mensaje informativo y regresa a la acción 1 de esta sección.
<b>Sección 2: "Modificar asistencia"</b>		
Flujo básico		
	Actor	Sistema
1.	El profesor accede a la opción Modificar asistencia.	2. El sistema muestra el listado de estudiantes.
3.	El profesor selecciona el estudiante a Modificar asistencia.	4. El sistema brinda la posibilidad de modificar la asistencia.
5.	El profesor realiza los cambios deseados.	6. Se verifica que los campos obligatorios estén llenos.

## ***CARACTERÍSTICAS DEL SISTEMA***

		6.1 Se actualiza la Información y finaliza el caso de uso.
Flujos alternos		
Nº Evento <2>		
	Actor	Sistema
		6.1. Se emite un mensaje para que llene los campos obligatorios y regresa a la acción 3 de esta sección.

**Tabla 3** Descripción del CUS\_Gestionar\_asistencia.

### **CUS 3 Gestionar evaluación.**

Objetivo	Gestionar la evaluación de uno o varios grupos.	
Actores	Profesor	
Resumen	Es aquí donde se seleccionan, registran, y se modifican las evaluaciones de los estudiantes.	
Complejidad	Media	
Prioridad	Crítico	
Precondiciones	Exista el grupo y alumnos asignados al registro.	
Postcondiciones	Se registra una nueva evaluación, se actualizan los datos o se selecciona la misma.	
Flujo de eventos		
	Actor	Sistema
1.	El profesor necesita seleccionar, registrar y modificar las evaluaciones de un estudiante.	El sistema verifica si están los datos del grupo cargados.
3.		3. El sistema ejecuta alguna de las siguientes acciones: a) Si decide registrarla evaluación, ir a la sección 2 "Registrar asistencia". b) Si decide modificar la evaluación de estudiante, ir a la sección 3 "Modificar evaluación".

## ***CARACTERÍSTICAS DEL SISTEMA***

Sección 1: "Registrar evaluación"		
Flujo básico		
	Actor	Sistema
1.	El profesor accede a la opción Registrar evaluación.	2. El sistema muestra el listado de estudiante.
3.	El profesor selecciona el estudiante.	4. El sistema guarda la evaluación seleccionada y termina el CUS.
Flujos alternos		
Nº Evento <1>		
	Actor	Sistema
1.		4.1 Se emite un mensaje para que marque el tipo de evaluación deseada y regresa a la acción 3.
Sección 2: "Modificar Evaluación"		
Flujo básico		
	Actor	Sistema
1.	El profesor accede a la opción Registrar evaluación.	2. El sistema muestra el listado de estudiantes.
3.	El profesor selecciona el estudiante a modificar evaluación.	4. El sistema brinda la posibilidad de modificar la evaluación.
5.	El profesor realiza los cambios deseados.	6. Se verifica que los campos obligatorios estén llenos. 6.1 Se actualiza la Información y finaliza el caso de uso.
Flujos alternos		
Nº Evento <2>		
	Actor	Sistema
		6.1. Se emite un mensaje para que llene los campos obligatorios y regresa a la acción 3 de esta sección.

**Tabla 4** Descripción del CUS\_Gestionar\_evaluación.

## ***CARACTERÍSTICAS DEL SISTEMA***

### **CUS 4 Seleccionar opciones de evaluación.**

Objetivo	Seleccionar opciones de evaluación.	
Actores	Profesor	
Resumen	El Caso de Uso (CU) inicia cuando el profesor selecciona el grupo para realizar la funcionalidad de registrar evaluaciones de los estudiantes para determinado turno de clase.	
Complejidad	Media	
Prioridad	Secundario	
Precondiciones	Exista el grupo y alumnos asignados al registro.	
Postcondiciones	El profesor tiene que tener los permisos para cargar el listado de dicho grupo.	
<b>Flujo de eventos</b>		
	<b>Actor</b>	<b>Sistema</b>
1.	El profesor selecciona la opción Cargar de grupos docentes.	2. El sistema verifica si están los datos del grupo cargados.
3.	El profesor selecciona la opción Registrar evaluación. a) Luego accede a la opción Seleccionar el tipo de asistencia.	4. El sistema ejecuta la siguiente acción: a) Si decide escoger una evaluación, ir a la sección 1 “Seleccionar opciones de evaluación”.
4.		
<b>Sección 1: “Seleccionar opciones de evaluación”</b>		
<b>Flujo básico</b>		
	<b>Actor</b>	<b>Sistema</b>
1.	El profesor accede a la opción Seleccionar opciones de evaluación.	2. El sistema muestra los tipos de evaluaciones a escoger.
6.	El profesor selecciona un tipo de evaluación.	4. El sistema muestra la funcionalidad a ejecutar y termina el CUS.

**Tabla 5** Descripción del CUS\_Seleccionar\_opciones\_de\_evaluación.

### **CUS 4 Generar reporte.**

<b>Objetivo</b>	Generar reporte en PDF de los estudiantes por grupo.
<b>Actores</b>	Profesor
<b>Resumen</b>	El Caso de Uso (CU) inicia cuando el profesor selecciona el grupo, luego realiza las funcionalidades de registrar evaluaciones y asistencia de los estudiantes para poder generar el reporte en

## ***CARACTERÍSTICAS DEL SISTEMA***

	PDF.	
<b>Complejidad</b>	Media	
<b>Prioridad</b>	Secundario	
<b>Precondiciones</b>	Exista el grupo y alumnos asignados al registro.	
<b>Postcondiciones</b>	Se muestra en la interfaz para enviar registro.	
<b>Flujo de eventos</b>		
	<b>Actor</b>	<b>Sistema</b>
1.	El profesor selecciona la opción Cargar de grupos docentes.	2. El sistema verifica si están los datos del grupo cargados.
		3. El sistema ejecuta la siguiente acción: a) Si decide generar reporte en PDF, ir a la sección 1 “Generar reporte.”
<b>Sección 1: “Generar Reporte”</b>		
<b>Flujo básico</b>		
	<b>Actor</b>	<b>Sistema</b>
1.	El profesor selecciona el grupo.	2. El sistema verifica si se encuentra el grupo cargado.
3.	El profesor después de registrar la asistencia y la evaluación de los estudiantes de ese grupo, selecciona la opción generar reporte.	4. El sistema crea el reporte en PDF de dicho grupo en el teléfono del profesor.
5.	El profesor si desea visualizar el PDF, selecciona la opción Visualizar reporte.	6. El sistema muestra la funcionalidad a ejecutar y termina el CUS.

**Tabla 6** Descripción del CUS\_Generar\_reporte.



### 2.4.3 Diagrama de Casos de Uso del Sistema.

El diagrama de casos de uso del sistema documenta el comportamiento del sistema desde el punto de vista del usuario donde interactúan los procesos y el actor.

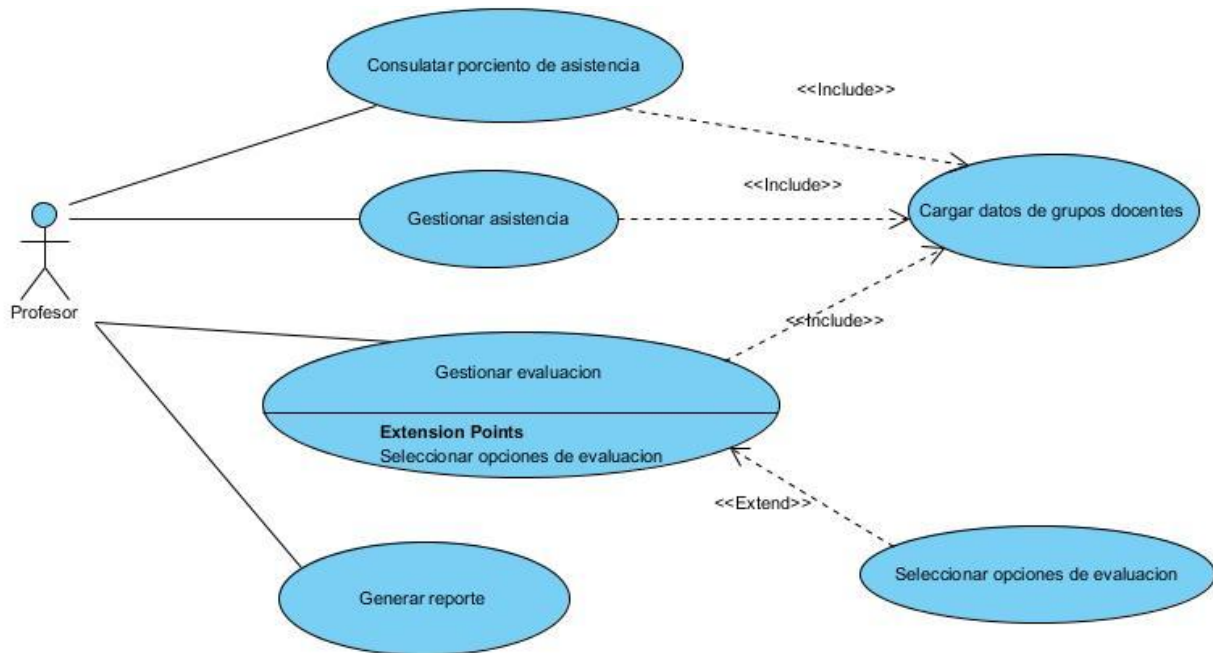


Figura 15 Diagrama de CU del Sistema.

Fuente: Elaboración propia.

### Conclusiones del Capítulo.

Este capítulo permitió crear el modelo del sistema que será implementado, utilizando los casos de usos para ese propósito.

Como resultado del trabajo realizado se decide priorizar el desarrollo de los casos de uso Gestionar evaluación, Gestionar asistencia, Cargar datos de grupos docentes por contar con funcionalidades críticas para el ciclo de vida del software.

Los casos de usos identificados dirigirán el trabajo a lo largo del resto de los flujos de trabajos propuestos por la metodología de desarrollo de software utilizada, por lo tanto serán los encargados de enlazar directamente los diferentes flujos de trabajo.

## Capítulo III: Análisis y Diseño del Sistema.

### Introducción.

En el presente capítulo se describen detalladamente los procesos que se llevan a cabo en el flujo de trabajo de Análisis y Diseño con el objetivo de convertir los requisitos funcionales en un esquema de cómo va a ser implementado en el sistema. Además se crean los principales artefactos: Modelo de Análisis que identifica las clases del análisis, se realizan los diagramas de clases donde se representan las relaciones existen entre ellas y los diagramas de colaboración para el análisis donde se modelan las interacciones entre los objetos de las diversas clases y después de haber cumplido con todas estas actividades y verificando los resultados de la misma se hace la entrada al Modelo de Diseño donde se procede a la confección del mismo.

### 3.1 Modelo de Análisis.

El modelo de análisis tiene las clases que se enmarcan en los requisitos funcionales, representan conceptos y relaciones del dominio. El objetivo principal de este modelo es obtener una comprensión profunda y una descripción detallada de los requisitos de software que ayuden a estructurar la vista interna del sistema. Para ello se definen las realizaciones de los casos de uso, donde se describe cómo se llevan a cabo cada uno de ellos, a través de las clases del análisis y de la interacción entre sus objetos.

El modelo es descrito en el lenguaje de los desarrolladores pero sin tener en cuenta el lenguaje de programación ni la plataforma, pues no está presente en sus propósitos precisar cómo se implementará la solución. Sirve como una primera aproximación al diseño constituyendo sus artefactos la entrada principal de este último.

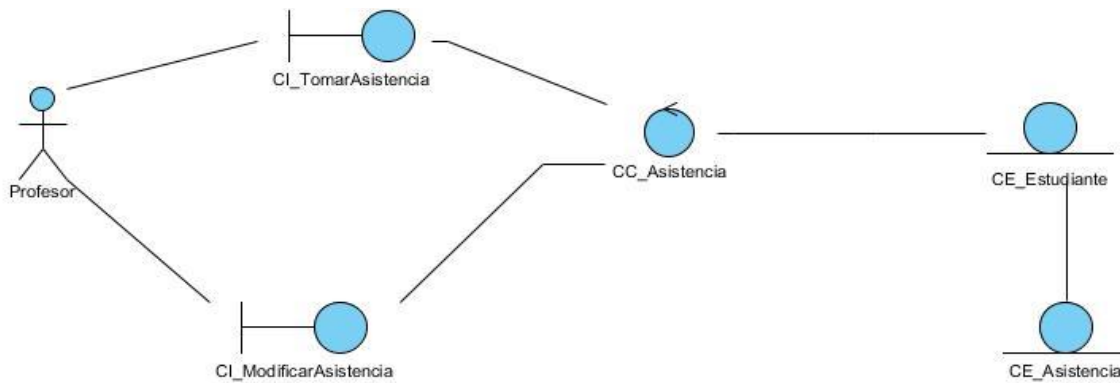
#### 3.1.1 Diagrama de Clases del Análisis.

En este epígrafe se especificarán las clases del análisis que facilitará la estructura de una vista interna del sistema. Además poseen atributos y entre ellas se establecen relaciones de asociación, agregación / composición, generalización / especialización y tipos asociativos.

La metodología RUP propone clasificar a las clases en:

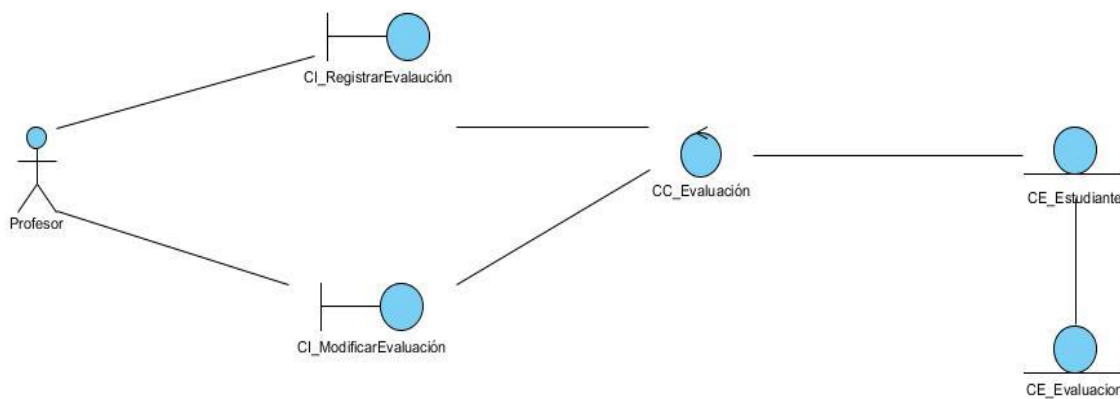
- Entidad: Modela información que posee larga vida y que es a menudo persistente.
- Interfaz: Modela la interacción entre el sistema y sus actores.
- Controladora: Coordina la realización de uno o unos pocos casos de uso coordinando las actividades de los objetos que implementan la funcionalidad del caso de uso.

A continuación se muestran los diagramas de clases del análisis para cada uno de los casos de uso del sistema:



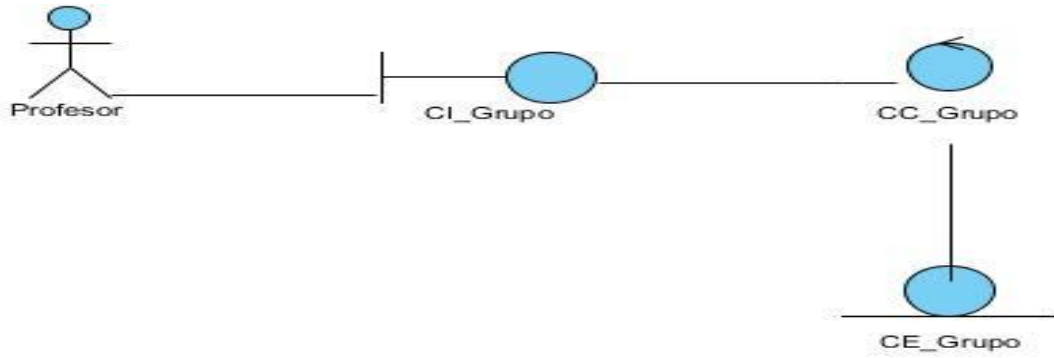
**Figura 16** Diagrama de clases del análisis CU Gestionar asistencia.

**Fuente:** Elaboración propia.



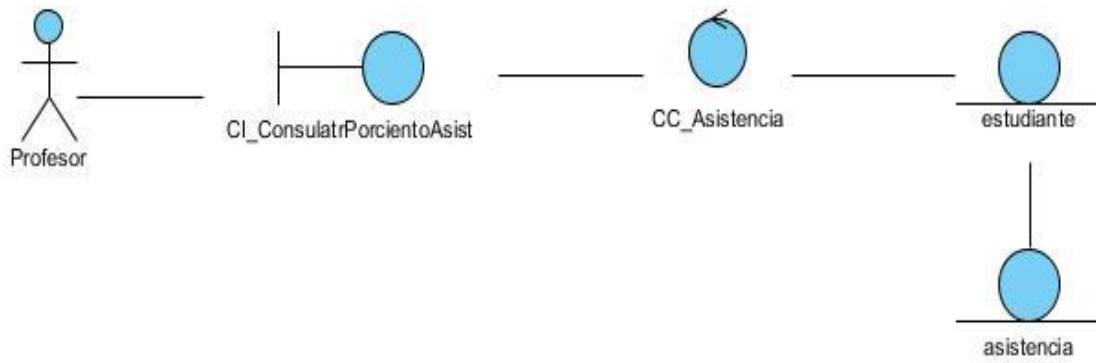
**Figura 17** Diagrama de clases del análisis CU Gestionar evaluación.

**Fuente:** Elaboración propia.



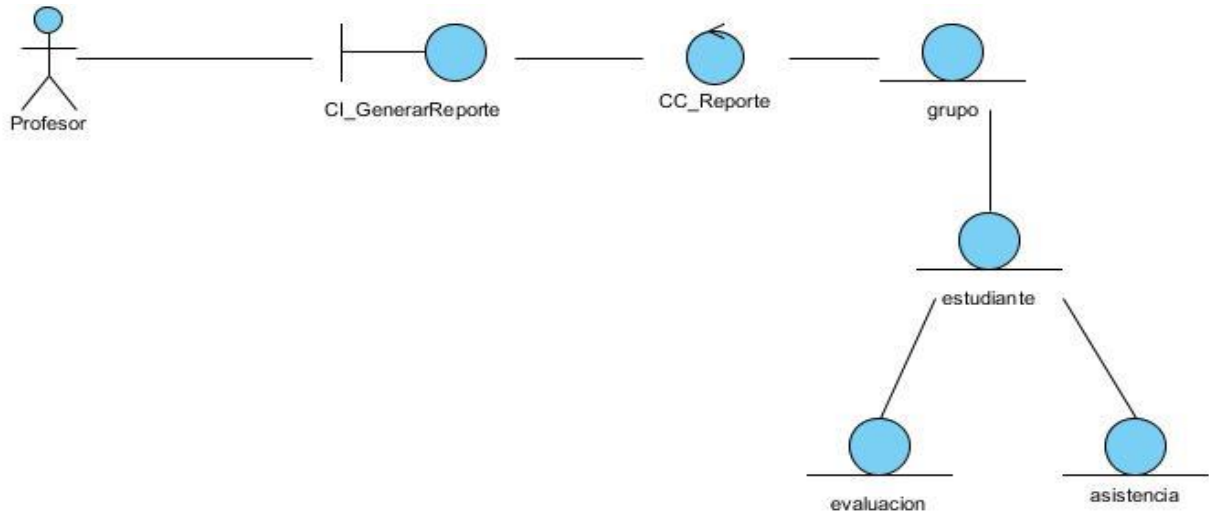
**Figura 18** Diagrama de clases del análisis CU Cargar grupo.

**Fuente:** Elaboración propia



**Figura 19** Diagrama de clases del análisis CU Consultar asistencia.

**Fuente:** Elaboración propia



**Figura 20** Diagrama de clases del análisis CU Generar reporte.

**Fuente:** Elaboración propia

## 3.2 Modelo de Diseño.

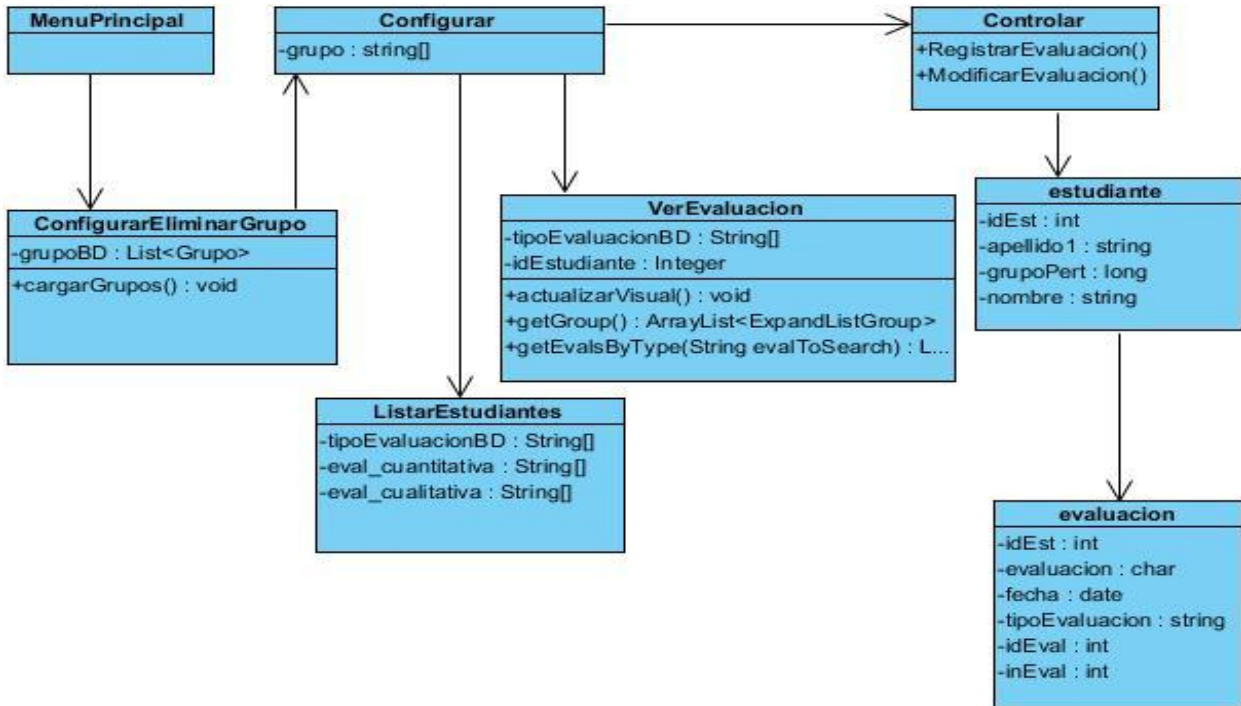
El modelo de diseño es una realización del diseño del sistema. Es un modelo físico y concreto. Se centra en cómo los requisitos funcionales y no funcionales tienen impacto en el sistema a desarrollar. Dentro de sus propósitos están: crear una entrada apropiada y un punto de partida para la implementación, descomponer los trabajos de implementación en partes más manejables que puedan ser llevadas a cabo por diferentes equipos de desarrollo. (34)

### 3.2.1 Diagrama de Clases del Diseño.

Un diagrama de Clases del Diseño es un tipo de diagrama estático que describe la estructura de un sistema mostrando sus clases, atributos y las relaciones entre ellos. Una clase es una descripción de objetos que comparten los mismos atributos, operaciones, métodos, relaciones y semántica. (34)

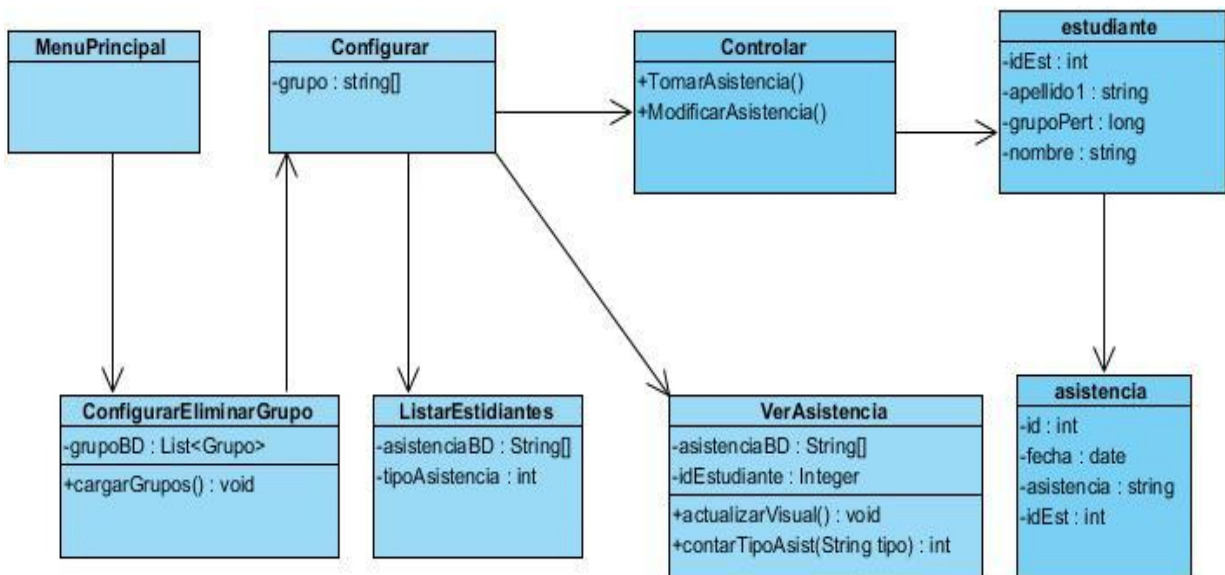
El diagrama general de las clases del diseño representa la interacción entre las clases de las diferentes capas que integran el sistema.

A continuación se muestran los diagramas de clases del diseño.



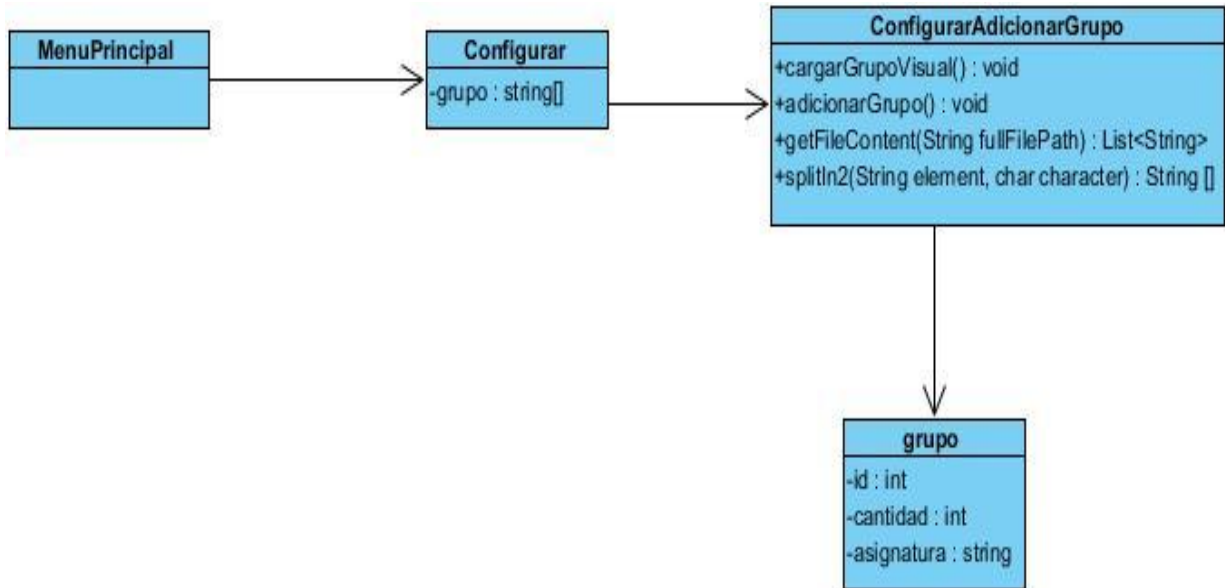
**Figura 21** Diagrama de clases del diseño del CU Gestionar evaluación.

Fuente: Elaboración propia.



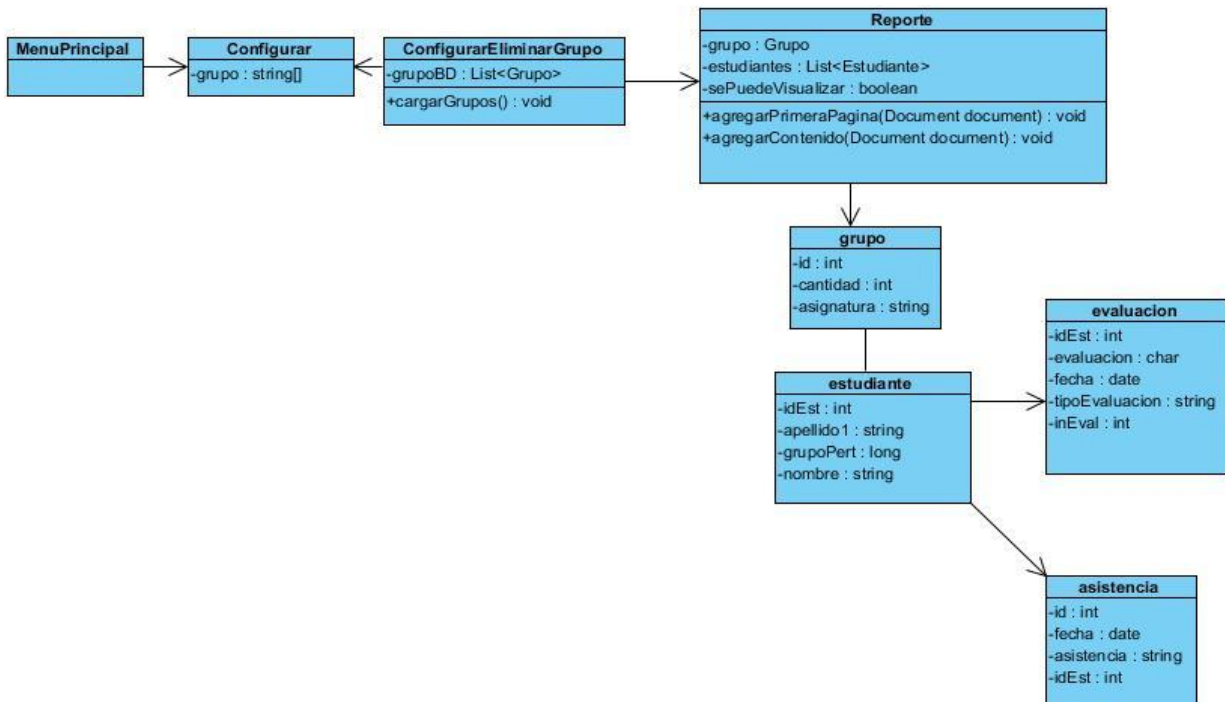
**Figura 22** Diagrama de clases del diseño del CU Gestionar asistencia.

Fuente: Elaboración propia



**Figura 23** Diagrama de clases del diseño del CU Cargar grupo docente.

Fuente: Elaboración propia



**Figura 24** Diagrama de clases del diseño del CU Generar reporte.

Fuente: Elaboración propia

### **3.3 Descripción de la Arquitectura.**

Una de las tareas más importantes en la construcción de cualquier sistema es la definición de su ambiente de desarrollo. La arquitectura de software define, de manera abstracta, los componentes que llevan a cabo alguna tarea de computación, sus interfaces y la comunicación entre ellos para la creación de un producto de software. Selecciona y diseña con base a objetivos prefijados para el sistema de información, que pueden ser de tipo funcional, de mantenibilidad, auditabilidad, flexibilidad e interacción con otros sistemas de información, teniendo en cuenta las limitaciones derivadas de las tecnologías disponibles para implementarlos.<sup>1</sup>

#### **3.3.1 Patrón Arquitectónico.**

Para el desarrollo de la aplicación Registro de Asistencia y Evaluaciones para dispositivos móviles sobre la plataforma Android se emplea patrón arquitectónico Modelo Vista Controlador (MVC) cuya principal bondad consiste en separar los datos de una aplicación, la interfaz de usuario y la lógica de negocios en tres componentes distintos que se relacionarán para al final tener como resultado la aplicación (35).

Se puede decir que en Android hay parte de MVC implementado de forma natural (36). El proceso de trabajo de elaboración de actividades con layouts, resources y capacidades de dispositivo "Controlling" ante los diferentes tipos de interfaces de dispositivos hardware. Lo único que su implementación nativa hace que se olvide parte del "controller" de la vista, además posee capacidades de gestión de eventos. Por ello parte del patrón estándar de las aplicaciones puede ser basado en un ejemplo más concreto de MVC, sin la parte de Controlling duro gestionado por el SO.

---

<sup>1</sup>IH-SW-DR-091 ALAS-HIS\_Documento de Arquitectura del Sistema



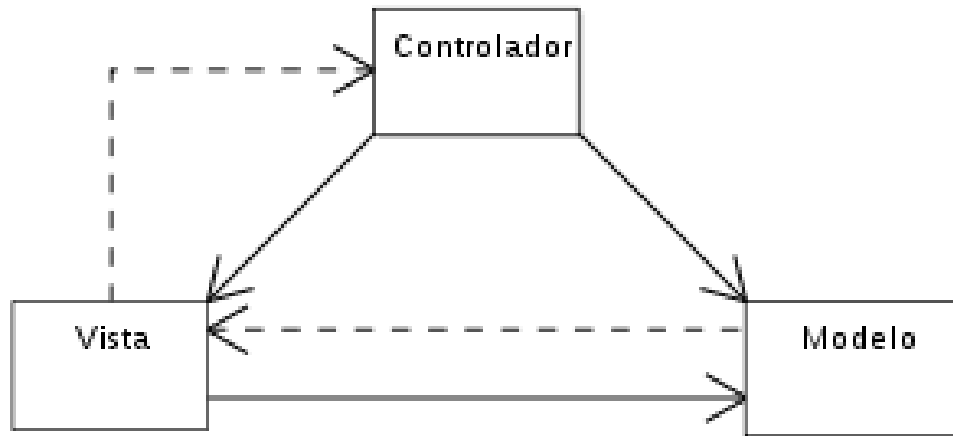


Figura 25 Patrón Modelo-Vista-Controlador.

Fuente: Elaboración propia

A continuación una explicación de qué se tratan los componentes de este modelo:

1. **Modelo.** El modelo se refiere a las representaciones que se construyen basadas en la información con la que operará en la aplicación. En Java, el modelo viene siendo análogo a los *beans* que tienen la particularidad de ser reutilizables y ayudan haciendo las aplicaciones escalables. En esta parte del modelo también juega la decisión de qué modelo para almacenar información se utilizará como base de datos, Web Services. El modelo que se elija depende obviamente de las necesidades de información de la aplicación.
2. **Vista.** La vista no es más que la interfaz con la que va a interactuar el usuario. En Android, las interfaces las se construyen en XML. Esta parte es realmente parecida a lo que se hace en el desarrollo web con los CSS. Se construye el esqueleto en XML que equivale al HTML de un sitio. Posteriormente, con ayuda de estilos, que también se escriben en XML, se puede empezar a darle formato de colores, posiciones, formato, etc. al esqueleto.
3. **Controlador.** Son todas esas clases que ayudarán a darle vida a esas interfaces que ya se construyen, permitirán desplegar y consumir información de/para el usuario. Estos controladores se programan en lenguaje Java y son el core de la aplicación.

A continuación se describen los elementos de cada basada en MVC:

1. Capa Vista - VIEW: Incluye las interfaces de usuario (Actividades, recursos, layouts), o aquellos artefactos de Android sin GUI (Servicios o Broadcast Receivers).
2. Presenter (Capa de negocio): Incluye clases de negocio, hacen de puente entre la vista y el modelo, incluyendo clases con métodos que alberguen la lógica de nuestra aplicación. Incluye también la librería iText. Siempre los artefactos de la capa Vista invocarán a las clases albergadas en esta capa, nunca accederán al modelo directamente.
3. Capa Modelo: Dos partes importantes:
  - Clases de Entidad: incluyen la definición lógica del modelo de datos con el que se trabaja. Estas clases serán manejadas por la capa de negocio.
  - Infraestructura Layer: Abstrae la complejidad tecnológica de la gestión del almacenamiento físico. Existen diversas formas en Android de manejar información, desde Content Providers, Ficheros, Bases de Datos, hasta WebServices; en este caso se empleará los ficheros la base de datos sQLite.

A continuación, se muestra como está compuesta dicha arquitectura en el sistema:

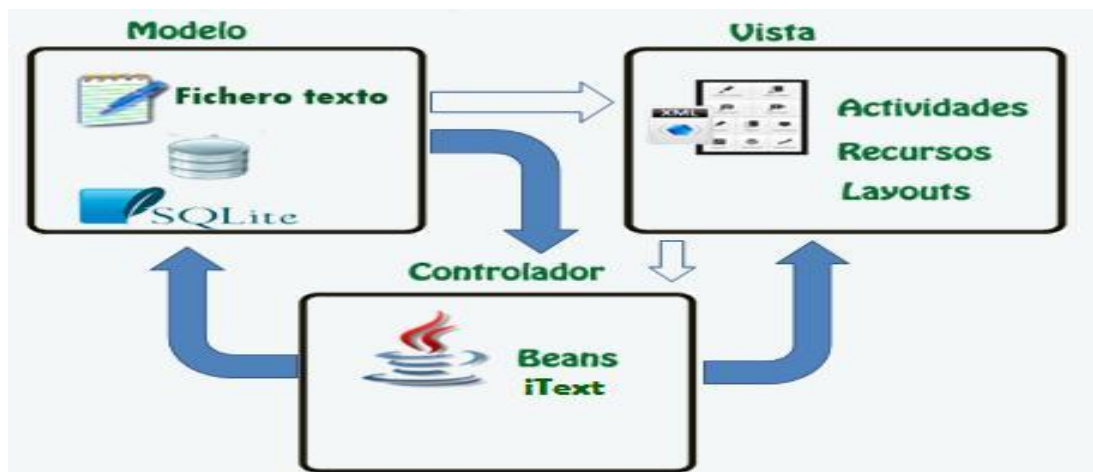


Figura 26 Concepción Modelo-Vista-Controlador del sistema.

Fuente: Elaboración propia

## 3.3.2 Patrones de Diseño. Patrones GRASP.

Los patrones GRASP describen los principios fundamentales de la asignación de responsabilidades a objetos, expresados en forma de patrones.

- **Controlador (Controller).**

El patrón controlador perteneciente al grupo de patrones GRASP se encarga de manejar los eventos del sistema que representan la visión general del sistema o de un caso de uso. En la Figura 28 se aprecia un ejemplo de este patrón, el cual propone el diseño de clases con la responsabilidad de controlar el flujo de eventos del sistema a clases específicas.

```
int tipoAsistencia = 0;

String[] eval_cuantitativa = new String[]{"2", "3", "4", "5"};
String[] eval_cualitativa = new String[]{"Bien", "Regular", "Mal"};

boolean estaVertical = true;

+ protected void onCreate(Bundle savedInstanceState) {}
- @Override
  protected void onResume() {
    super.onResume();
  }
+ protected void onItemClick(AdapterView l, View v, int position, long id) {}
```

Figura 27 Ejemplo patrón Controlador.

- **Experto.**

El patrón experto perteneciente al grupo de patrones GRASP plantea asignar una responsabilidad al experto en información, es decir, la clase que tiene los datos necesarios para cumplir con la responsabilidad. El problema que resuelve el patrón experto está referido al principio más básico mediante el cual las responsabilidades son asignadas en el diseño orientado a objetos. En la Figura 29 se puede apreciar el uso de este patrón con el que se pretende que los objetos realicen las acciones relacionadas con la información que poseen.

```

package com.example.control;

import java.util.List;

public class Controlar extends ListActivity
{
    String[] grupos = null;
    List<Grupo> objGrupos = null;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        AdministrarDBGrupo admDBGrupo = new AdministrarDBGrupo(getApplicationContext());
        admDBGrupo.abrirConexion();
        objGrupos = admDBGrupo.obtenerGrupos();
        admDBGrupo.cerrarConexion();

        grupos = new String[objGrupos.size()];
        for (int i = 0; i < grupos.length; i++)
        {
            String asignatura = "<Sin asignatura>";
            if (objGrupos.get(i).getAsignatura()!=null) {
                asignatura = objGrupos.get(i).getAsignatura();
            }
            grupos[i] = objGrupos.get(i).getNombre() + " - " + asignatura;
        }

        setListAdapter(new ArrayAdapter<String>(this, android.R.layout.simple_list_item_1, grupos));
    }
}

```

Figura 28 Ejemplo patrón Experto.

- **Patrón Creador.**

El patrón creador perteneciente al grupo de patrones GRASP se basa en asignarle a la clase la responsabilidad de crear una instancia de otra clase. Este patrón guía la asignación de responsabilidades relacionadas con la creación de objetos como se muestra en la Figura 30.

```

Ayuda.java  Informacion.java  ConfigurarAdicionarGrupo.java
}
private void adicionarGrupo()
{
    try
    {
        AdministrarDBGrupo administrarDBGrupo = new AdministrarDBGrupo(this);
        administrarDBGrupo.abrirConexion();
        long idGrupo = administrarDBGrupo.insertarGrupo(gFileContent.get(0), "<Sin asignar>", 0);
        administrarDBGrupo.cerrarConexion();

        AdministrarDBEstudiante administrarDBEstudiante = new AdministrarDBEstudiante(this);
        administrarDBEstudiante.abrirConexion();
        for (int i = 1; i < gFileContent.size(); i++) {
            String nombreEstudiante = gFileContent.get(i);
            String[] aux = splitIn2(nombreEstudiante, ' ');
            String nombre = aux[0];
            String apellido = aux[1];

            Estudiante e = new Estudiante(nombre, apellido, idGrupo);
            administrarDBEstudiante.insertarEstudiante(e);
        }
        administrarDBEstudiante.cerrarConexion();
    }
}

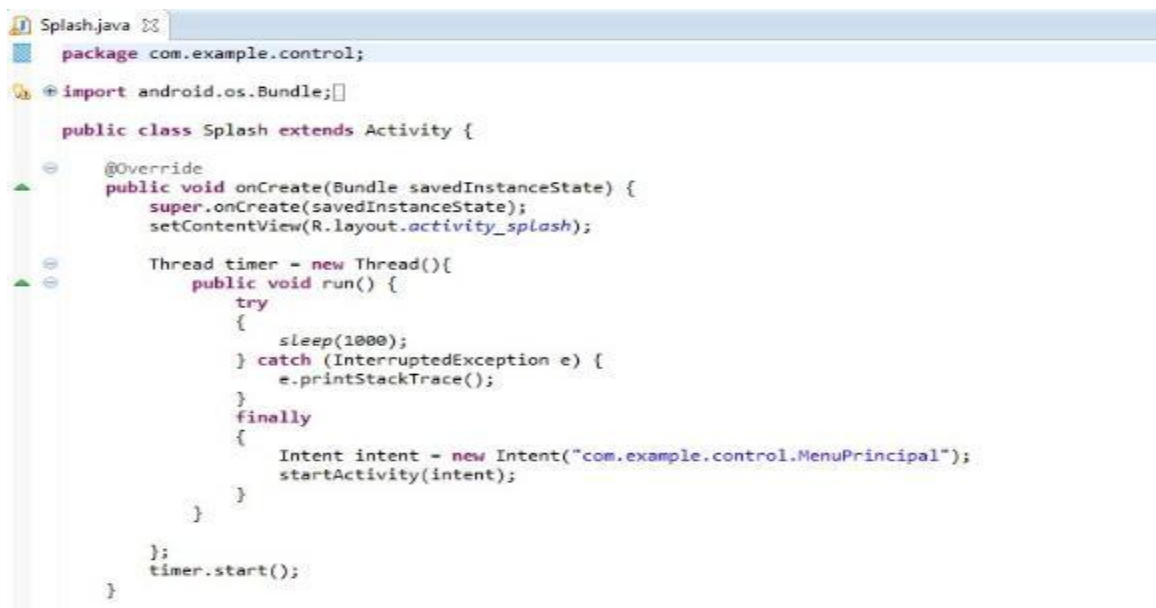
```

Figura 29 Ejemplo patrón Creador.

- **Bajo acoplamiento.**

El patrón bajo acoplamiento perteneciente al grupo de patrones GRASP consiste en asignar una responsabilidad para mantener bajo acoplamiento. El patrón propone el diseño de clases más independientes, lo que reduce el impacto del cambio y facilita la reutilización en otros sistemas.

En las aplicaciones Android esto se garantiza, una Activity es la unidad atómica de responsabilidad, una aplicación Android puede contener varias Activity, una de ellas marcada como principal. En la Figura 31 se puede apreciar un ejemplo.



```
 Splash.java
package com.example.control;

import android.os.Bundle;

public class Splash extends Activity {

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_splash);

        Thread timer = new Thread(){
            public void run() {
                try
                {
                    sleep(1000);
                } catch (InterruptedException e) {
                    e.printStackTrace();
                }
                finally
                {
                    Intent intent = new Intent("com.example.control.MenuPrincipal");
                    startActivity(intent);
                }
            }
        };
        timer.start();
    }
}
```

Figura 30 Ejemplo patrón Bajo Acoplamiento.

### 3.4 Modelo de Datos.

La importancia y utilidad de precisar una estructura que permita almacenar datos, reconocer el contenido, y recuperar la información hacen del modelo de datos una actividad esencial en el desarrollo del software. La estructura tiene que estar en correspondencia con las necesidades de las aplicaciones que la usarán.

El modelado de los datos responde a interrogantes importantes para el procesamiento de datos sobre las entidades que procesará el sistema, los atributos y las relaciones entre las mismas. Para responder estas interrogantes se realizan un conjunto de actividades que tienen como resultado primario un Modelo Lógico de Datos, a través de las clases persistentes y un Modelo Físico de Datos, mediante el Diagrama Entidad-Relación.

## 3.4.1 Modelo Lógico de Datos.

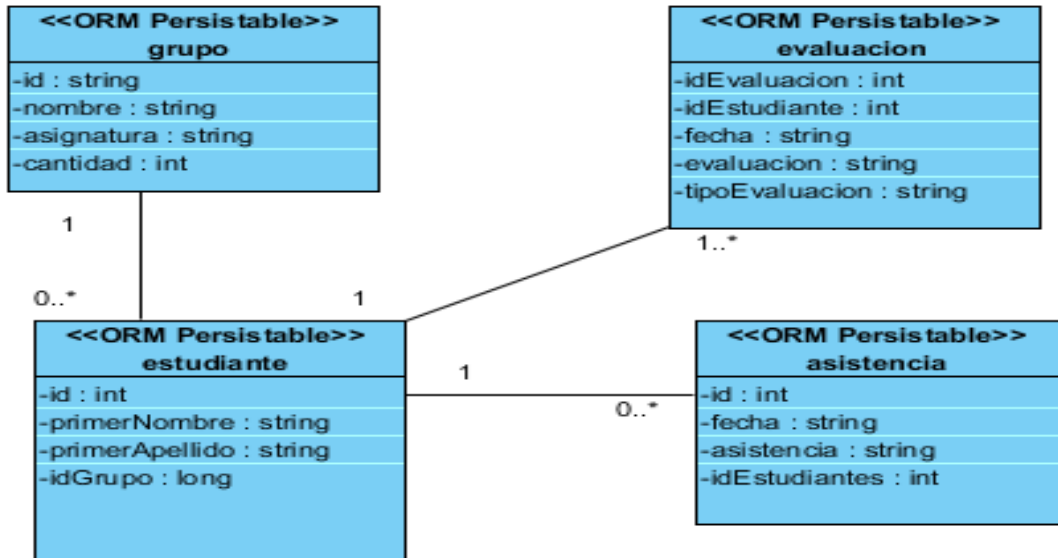


Figura 31 Diagrama de clases persistentes.

## 3.4.2 Modelo físico de datos.

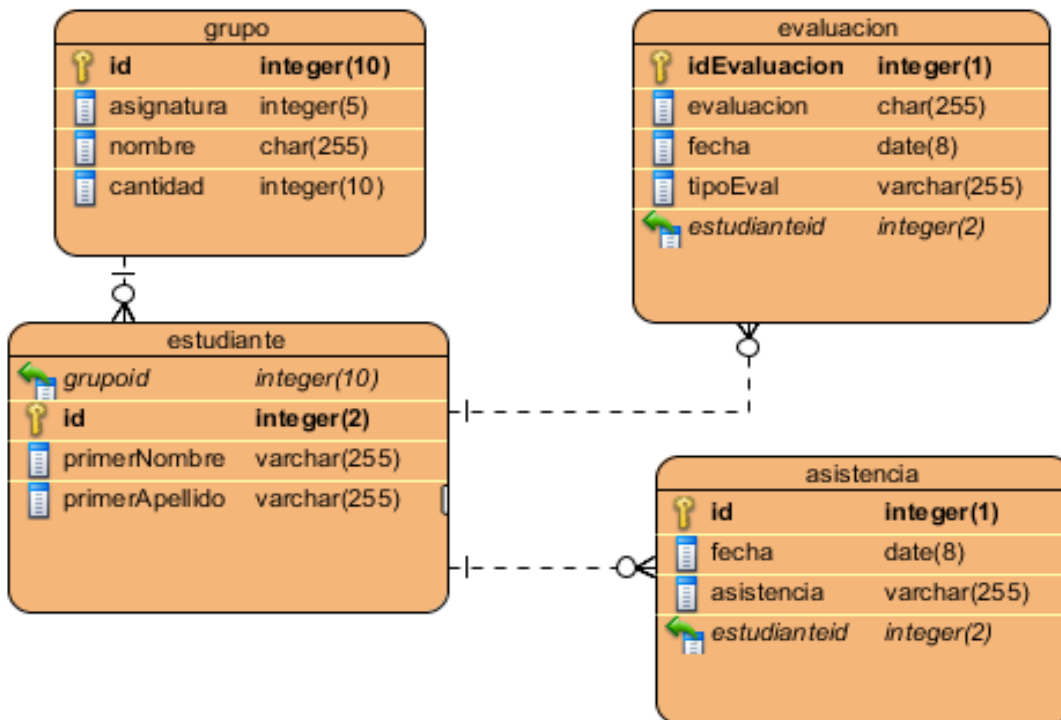


Figura 32 Diagrama entidad-relación.

### **3.5 Tratamiento de Excepciones.**

La función principal del tratamiento de excepciones es la determinación de los principales errores que pueden surgir durante el funcionamiento del sistema y proporcionarle su tratamiento. En el Registro de Asistencia y Evaluaciones el tratamiento de errores está enfocado principalmente a errores durante la interacción del usuario con el servicio solicitado, esto puede ocurrir cuando el proceso para registrar no sea el adecuado; es decir que el usuario pretenda registrar evaluaciones cuando primeramente debiera cargar los datos del grupo docente. Se trata en todo momento de minimizar la ocurrencia de errores de esta índole, aprovechando la posibilidad de enviarle un mensaje de alerta para que rectifique y así garantizar el buen funcionamiento de la aplicación.

### **3.6 Seguridad.**

La seguridad es un elemento indispensable con que deben contar los productos informáticos, es por ello que la plataforma no está ajena a ello y utiliza para la parte de seguridad los siguientes elementos:

- Gestor de excepciones: maneja los problemas ocurridos en tiempo de ejecución, no permite al usuario interactuar directamente con los errores del sistema.
- Permitir al usuario acceder solamente al contenido que él solicitó.

### **Conclusiones del Capítulo.**

El Análisis y Diseño contribuye a una arquitectura estable y sólida, un diseño robusto de la base de datos y a crear un plano del modelo de implementación.

La peculiaridad del diseño es modelar el sistema, encontrar su forma (incluida la arquitectura) para que soporte todos sus requisitos.

Una entrada esencial en el diseño es el resultado del análisis. El modelo de análisis proporciona una comprensión detallada de los requisitos e impone una estructura del sistema que debe conservarse lo más fielmente posible cuando se da forma al sistema.

## Capítulo IV: Implementación y Prueba.

### Introducción.

En este capítulo se hace referencia a las disciplinas de implementación y prueba. Se describe el modelo de implementación utilizado y se muestran los diagramas de componente y de despliegue. Se realiza el modelo de prueba a través de los casos de prueba aplicados en el desarrollo del sistema propuesto. Además, se expone la distribución del sistema en nodos y los protocolos de comunicación entre cada uno de ellos. Se explica la trazabilidad de las clases del diseño a componentes de implementación mediante los diagramas de componentes del modelo de implementación.

### 5.1. Modelo de Implementación.

El modelo de implementación representa la composición física de la implementación. En él está incluido el diagrama de despliegue, contiene además, los diagramas de componentes que muestran un conjunto de elementos del modelo tales como componentes, paquetes de implementación y sus relaciones.

#### 5.1.1. Diagrama de Componentes.

El diagrama de componentes es usado para estructurar el modelo de implementación y mostrar las relaciones entre los elementos de este, proporcionan las interfaces necesarias para la utilización de sus funcionalidades.

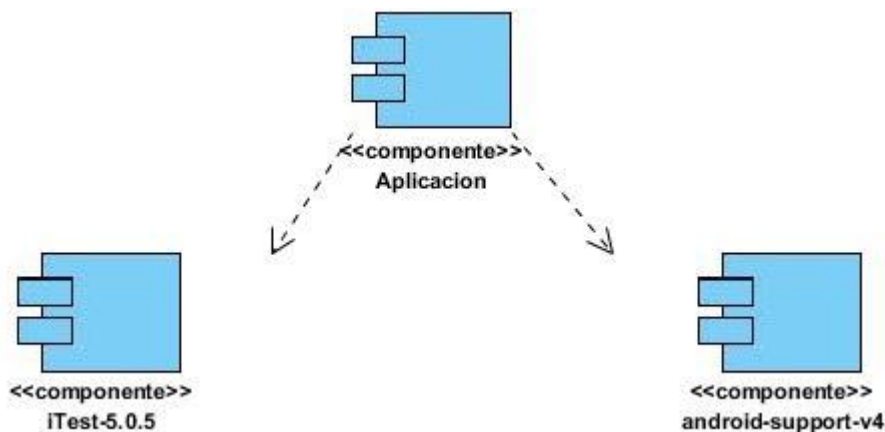


Figura 33 Diagrama de componentes del sistema.

#### 5.1.2. Diagrama de Despliegue.



En el diagrama de despliegue se indica la situación física de los componentes lógicos desarrollados, se sitúa el software en el hardware que lo contiene.



Figura 34 Diagrama de despliegue.

Fuente: Elaboración propia

## 5.2. Pruebas.

Culminada la implementación del registro móvil, se hace necesario realizar una serie de pruebas y validaciones para comprobar si el producto desarrollado cuenta con la calidad requerida. Con la variedad de tipos de pruebas que pueden ser aplicadas con el objetivo de obtener un producto con calidad, se seleccionó evaluar el desarrollo de la aplicación mediante el método de prueba de Caja Negra.

### 6.2.1 Prueba de Caja Negra

La prueba de caja negra se refiere a las pruebas que se llevan a cabo sobre la interfaz del software. O sea, los casos de prueba pretenden demostrar que las funciones del software son operativas, que la entrada se acepta de forma adecuada y que se produce un resultado correcto, así como que la integridad de la información externa se mantiene.

#### 6.2.1.1 Casos de Prueba

Los casos de prueba se emplean con el propósito de comprobar que todos los requisitos de una aplicación son revisados, y determinar que los mismos sean completamente satisfactorios. En el Registro de Asistencia para dispositivos móviles se diseñaron los casos de pruebas correspondientes a los casos de uso que se describen a continuación:

**CU Cargar grupos docentes.**

## IMPLEMENTACIÓN Y PRUEBA

Escenario	Descripción	Respuesta del sistema	Flujo central
1.1 Cuando se selecciona la opción Iniciar y muestra los grupos.	Cuando el profesor selecciona la opción Adicionar grupo desde la tarjeta SD.	El sistema muestra un listado con todos los grupos que el profesor atiende para que seleccione el que desea agregar al sistema.	El profesor accede a la opción de cargar los datos. El sistema carga los datos seleccionados y muestra las funcionalidades a ejecutar y termina el CUS.
1.2 Cuando se selecciona la opción Iniciar y no muestra los grupos.	Cuando el profesor no selecciona la opción Adicionar grupo desde el Servidor.	Muestra el mensaje de información "Seleccionar la opción de Cargar grupo".	El profesor accede a la opción Cargar datos. El sistema muestra un listado con todos los grupos que el atiende. El profesor no selecciona el grupo para agregarlo al sistema. El sistema muestra el mensaje de información "seleccionar la opción de Cargar Grupo".

**Tabla 7** Diseño de caso de prueba CU Cargar grupos docentes.

### CU Generar reporte.

Escenario	Descripción	Respuesta del sistema	Flujo
1.1 Generar Reporte datos cargados.	Se comienza la verificación cuando el profesor haya registrado la asistencia y las evaluaciones.	Se selecciona la opción Visualizar reporte y Generar reporte. Luego el sistema guarda la información.	El profesor selecciona el grupo. El sistema verifica si se encuentra el grupo cargado. El profesor después de registrar la asistencia y la evaluación de los estudiantes de ese grupo, selecciona la opción generar reporte. El sistema crea el reporte en formato PDF de dicho grupo en el teléfono del profesor. El profesor si desea visualizar el PDF, selecciona la opción visualizar reporte. El sistema muestra la funcionalidad a ejecutar y termina el CUS.
1.2 Generar Reporte sin los datos cargados.	Se realiza la verificación cuando no se haya registrado la asistencia y las evaluaciones.	El sistema muestra un mensaje de error especificando que no ha registrado la asistencia y las evaluaciones.	El profesor selecciona el grupo. El sistema verifica si se encuentra el grupo cargado. El profesor después de registrar la asistencia y la evaluación de los estudiantes de ese grupo, selecciona la opción generar reporte. El sistema crea el reporte en PDF de dicho grupo en el teléfono del profesor. El profesor si desea visualizar el PDF, selecciona la opción visualizar reporte. El sistema muestra un mensaje de error especificando no ha registrado la asistencia y las evaluaciones.

**Tabla 8** Diseño de caso de prueba CU Generar reporte

### CU Gestionar asistencia.

Escenario	Descripción	Respuesta del sistema	Flujo central
1.1 Registrar la asistencia a todos los estudiantes.	Se comienza al seleccionar un grupo y registrar la asistencia de los estudiante uno por uno.	Se muestra el listado de estudiantes.	El profesor accede a la opción tomar asistencia. El sistema muestra el listado de estudiante. El profesor selecciona el estudiante. El sistema guarda el tipo de asistencia seleccionada y termina el CUS.
1.2 Registrar la asistencia algunos estudiantes.	Se comienza al seleccionar un grupo y registrar la asistencia a algunos estudiantes.	Muestra un mensaje de información de error "Continuar registrando la asistencia".	El profesor accede a la opción tomar asistencia. El sistema muestra el listado de estudiante. El profesor selecciona el estudiante. El sistema muestra un mensaje de información de error "Continuar registrando la asistencia".

Tabla 9 Diseño de caso de prueba CU Gestionar asistencia.

#### 6.2.1.2 Resultados de las pruebas de caja negra.

En las pruebas de caja negras realizadas, de los 9 escenarios se identificaron 8 casos de pruebas. En la primera iteración se efectuaron 4 casos de pruebas detectándose 4 no conformidades ([Ver Anexo 1](#)) a las cuales se les dio solución. En la segunda iteración se detectaron 3 no conformidades de 3 de los casos de pruebas, y en la tercera y última iteración no se detectaron no conformidades del caso de prueba restante, las mismas fueron resueltas en su totalidad. En cada iteración de prueba se genera un resumen de todas las no conformidades existentes. En las dos iteraciones efectuadas se detectaron un total de 7 no conformidades, las cuales en su mayoría respondían a errores de bajo impacto en el correcto funcionamiento del sistema y todas tuvieron solución un tiempo máximo de 3 días, lo que indica que el Registro de Asistencia y Evaluaciones para dispositivos móviles desarrollada presenta buena calidad.

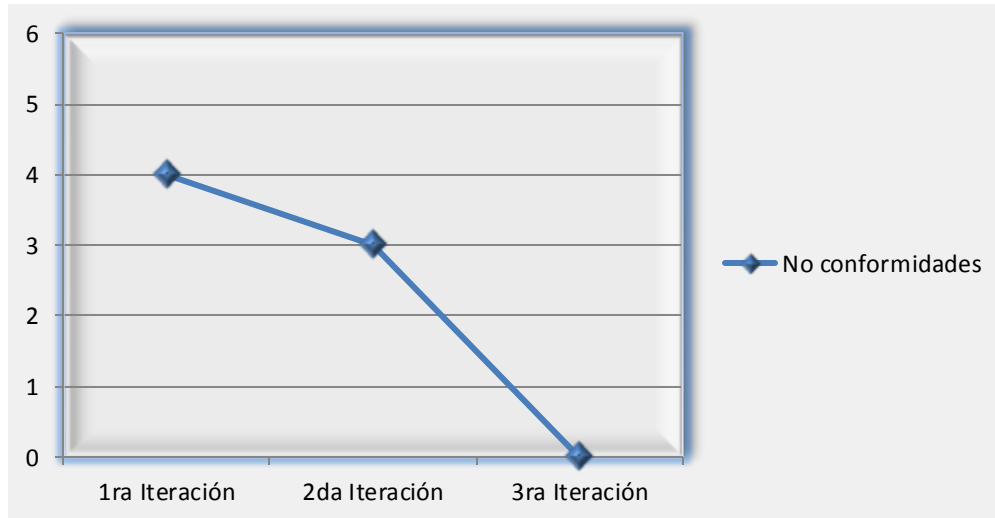


Figura 35 Resultados de las pruebas de caja negra.

### Conclusiones del Capítulo.

La realización de los diferentes diagramas permitió obtener una visión detallada del sistema. El diagrama de despliegue permitió una mejor visibilidad de la distribución de los nodos físicos que serán necesarios para el despliegue del sistema.

Se llevaron a cabo pruebas de caja negra mediante la cual se obtienen las no conformidades y los casos de pruebas correspondientes, brindaron la opción de comprobar la correcta ejecución de los requerimientos funcionales previamente definidos, en cada una de las iteraciones realizadas. Las pruebas realizadas al sistema tuvieron un resultado satisfactorio.

### **Conclusiones Generales.**

Con el desarrollo de este trabajo, se profundizó en el conocimiento de los métodos y herramientas para gestionar la asistencia y evaluaciones de estudiantes para teléfonos móviles y se aplicaron elementos de la metodología RUP, para documentar el software, con el cual se obtuvo la implementación de un registro de estudiantes con las funcionalidades previstas. El diseño y la implementación se rigieron por las características y particularidades de una aplicación en Android aplicando diferentes patrones de diseño y el patrón Modelo-Vista-Controlador.

Se puede concluir que se cumplió satisfactoriamente el objetivo trazado para este trabajo enfatizando en los siguientes puntos:

- Se realizó una investigación sobre los registros de asistencia y evaluación de estudiantes.
- El análisis de los principales sistemas que registran la asistencia y evaluaciones mediante dispositivos móviles, ha posibilitado conocer el funcionamiento de la aplicación Control ayudando en la definición de los requisitos del sistema.
- El análisis de diferentes herramientas, tecnologías, lenguajes y metodologías ha permitido definir cuáles utilizar en el desarrollo del sistema.
- La especificación de los requisitos que debe cumplir el sistema han posibilitado aplicar el patrón arquitectónico basado en componentes para describir la estructura general del sistema.
- El desarrollo de módulo del registro para dispositivos móviles ha permitido brindarle a los profesores una útil herramienta para el registro de la asistencia y evaluaciones de los estudiantes, así como de reporte del comportamiento del rendimiento de los mismos.
- La ejecución de las pruebas han posibilitado validar el correcto funcionamiento del sistema.

### **Recomendaciones**

Con el objetivo de efectuar un seguimiento y perfeccionamiento de esta investigación, se recomienda.

- Publicar el sistema propuesto en la comunidad de Android de la Universidad de las Ciencias Informáticas para que sea extensible al resto de los profesores.
- Analizar y considerar las sugerencias de los usuarios (profesores) sobre sistema.
- Crear futuras versiones del sistema para desplegarlo en otras universidades ayudando así a la digitalización del país.

### Referencias Bibliográficas.

1. **Stillo, Mónica.** *LOS DISCURSOS SOBRE LA INCLUSIÓN DIGITAL.* s.l. : COMMONS - Revista de Comunicación y Ciudadanía Digital. Vol. 1, Número 1, 2012.
2. **Alonso, Arturo Baz, Artime, Irene Ferreira y Rodríguez, María Álvarez.** *Dispositivos móviles.* Asturias, España : E.P.S.I.G : Ingeniería de Telecomunicación; Universidad de Oviedo, 2012.
3. **J. Ramírez, Israel.** *Los Sistemas Operativos.* Colombia : Universidad de los Andes, 2012.
4. Xataka móvil. *ESPACIO MASVOVIL.* [En línea] Weblogs SL. [Citado el: 20 de Marzo de 2013.] <http://www.xatakamovil.com>.
5. **Colina y Hector.** *Software Libre para Dispositivos Móviles.* Canaima : Venezolana de Telecomunicaciones-Vtelca, 2012.
6. **EcuRed.** Telefonía en Cuba. [En línea] EcuRed, 14 de Diciembre de 2010. [Citado el: 10 de Diciembre de 2012.] [http://www.ecured.cu/index.php/Telefon%C3%ADa\\_en\\_Cuba](http://www.ecured.cu/index.php/Telefon%C3%ADa_en_Cuba).
7. **Addine Fernández, Fátima y Recarey Fernández, Silvia.** *Didáctica, Teoría y Práctica.* Habana : Instituto Superior Pedagógico Enrique José Varona, 2002.
8. **UCI.** Tesis. *Diseño metodológico de la Investigación.* [En línea] Universidad de las ciencias informáticas(UCI). [Citado el: 23 de Noviembre de 2012.] [http://tesis.uci.cu/viewpage.php?page\\_id=21](http://tesis.uci.cu/viewpage.php?page_id=21).
9. **Ibáñez, y otros.** *Mobile Learning: Aprendiendo Historia con mi Teléfono, mi GPS y mi PDA.* s.l. : Universidad Autónoma de Madrid, 2008.
10. **Martínez, Carmen y J. Rueda, Antonio.** *Evaluación Continua de Alumnos Mediante el uso de Dispositivos Móviles Basados en iOS.* España : Departamento de Informática. Universidad de Jaén, 2013.
11. **Naismith, L., Lonsdale, P. y Vavoula y G.N. y Sharples, M.** Revisión de la Literatura en Tecnologías Móviles y el aprendizaje. [En línea] 2004. [Citado el: 18 de Febrero de 2013.] [www.futurelab.org.uk/research/lit\\_reviews.htm](http://www.futurelab.org.uk/research/lit_reviews.htm).. Informe11.
12. *Hacia una filosofía del m-learning.* **Nyiri, K.** Vaxjo, Suecia : Actas del Taller Internacional IEEE sobre tecnologías inalámbricas y móviles en Educación, 2002.
13. **Quinn, C.** mLearning: Mobile, Wireless, in your Pocket Learning. [En línea] [Citado el: 13 de Enero de 2013.] <http://www.linezine.com/2.1/features/cqmmwiyp.html>..
14. *Dispositivos Móviles.* **Hernández, Rocael y Morales, Miguel.** Guatemala : e-Learning en Galileo, 2013. Disponible en



[http://elearning.galileo.edu/revista/index.php?option=com\\_content&view=article&id=20&Itemid=115](http://elearning.galileo.edu/revista/index.php?option=com_content&view=article&id=20&Itemid=115).

15. **Mundaca Moraga, Bruno y J. González, Agustín.** *Evaluación y Seguimiento del Aprendizaje en Salas de Clases Utilizando Dispositivos Móviles*. Chile : Departamento de electrónica, Universidad técnica Federico Santa María, 2010.
16. **Arroyo, Miguel Vicente, Cáceres Tello, Jesús y Mencías Grande, Julio.** *Gestión de alumnos para el profesorado*. España : III CONCURSO DE IDEAS Y PROYECTOS PARA PERSONAL DE ADMINISTRACIÓN Y SERVICIOS, 2008.
17. **Franklin, Juan José, Flores, Juan José y Mori, Rosa.** *Utilización de dispositivos móviles para la gestión académica de alumnos y docentes de la Universidad de San Martín de Porres*. Lima : Universidad de San Martín de Porres, 2011.
18. **García, Daniel Adolfo.** *Integración de una Aplicación Móvil a la Intranet. Caso: Toma de Asistencia Estudiantil*. Lima : Universidad Católica del Perú, 2005.
19. Linux. [En línea] The Linux Foundation. [Citado el: 4 de Abril de 2013.] <http://www.linux.com/>.
20. **pcmag.com.** Software stack. *Enciclopedia*. [En línea] [Citado el: 10 de Diciembre de 2012.] [http://www.pcmag.com/encyclopedia\\_term/0,2542,t=software+stack&i=51702,00.asp](http://www.pcmag.com/encyclopedia_term/0,2542,t=software+stack&i=51702,00.asp).
21. **Open Handset Alliance Group.** Open Handset Alliance. [En línea] 2012. [Citado el: 4 de Abril de 2013.] <http://www.openhandsetalliance.com/>.
22. **Domínguez, José Antonio.** *Memoria: Desarrollo de aplicación móvil utilizando realidad aumentada*. Catalunya : Universidad Oberta de Catalunya, 2011. UOC-ETIG.
23. **Avila, Mejía.** *Android*. s.l. : UAM-I. Departamento de Ingeniería Eléctrica, 2012.
24. **Larramendi, Claudia Beatriz.** *Propuesta de arquitectura para desarrollo de aplicaciones compuestas para dispositivos móviles con Android*. La Habana : Universidad de las Ciencias Informáticas(UCI), 2012. TD\_05635\_12.
25. **Fernández Escribano, Gerardo.** *Introducción a Extreme Programming*. 2007.
26. **Jacobson, Ivar, Booch, Grady y Rambaugh, James.** *El Proceso Unificado de Desarrollo de Software*. Madrid, España : s.n., 2000.
27. **Fundación Universitaria de Las Palmas.** *Cursos de Orientación Profesional: Creación de aplicaciones móviles en Android*. Canarias : Servicio Canario de Empleo, 2007-2013.
28. UML Tool. *Visual Paradigm*. [En línea] Febrero de 2011. [Citado el: 20 de Enero de 2013.] <http://www.visual-paradigm.com/support/vpuml/releasenotes/642.jsp>.
29. **López, Ángel.** *Java la programación del futuro*. Uruguay : s.n., 1997.
30. **Flanagan, David.** *Java en Pocas Palabras*. 1999. 2da Edición.

## REFERENCIAS BIBLIOGRÁFICAS

---

31. **MediaProgramas SL.** ProgramasGratis.Net/itext 5.0.5. *iText*. [En línea] 2013. [Citado el: 4 de Junio de 2013.] <http://www.programas-gratis.net/b/itext-5.0.5>.
32. **Eclipse Foundation.** Eclipse. [En línea] Eclipse Foundation. [Citado el: 4 de Abril de 2013.] <http://www.eclipse.org/downloads/>.
33. Developers. *Get the Android SDK*. [En línea] [Citado el: 4 de Abril de 2013.] <http://developer.android.com/sdk/index.html>.
34. **Riesco y Daniel.** UML Diagrama de Clases y de Objetos. [En línea] [Citado el: 10 de Abril de 2013.] <http://sel.unsl.edu.ar/licenciatura/ingsoft2/UML-DiagramaClaseObjeto.pdf>.
35. **Androideity.** Androideity. *La Impotancia del MVC en Android*. [En línea] 2012. [Citado el: 30 de Abril de 2013.] <http://androideity.com/2012/05/10/la-importancia-del-mvc-en-android/>.
36. **Software, Tecnología por Cero-cool .** D Todo Un Poco. *Estructura de una aplicación en Android*. [En línea] 2011. [Citado el: 30 de Abril de 2013.] <http://www.dtodounpoco.net/1-estructura-de-una-aplicacion-en-android/>.

**Bibliografía**

1. **Stillo, Mónica.** *LOS DISCURSOS SOBRE LA INCLUSIÓN DIGITAL.* s.l. : COMMONS - Revista de Comunicación y Ciudadanía Digital. Vol. 1, Número 1, 2012.
2. **Alonso, Arturo Baz, Artime, Irene Ferreira y Rodríguez, María Álvarez.** *Dispositivos móviles.* Asturias, España : E.P.S.I.G : Ingeniería de Telecomunicación; Universidad de Oviedo, 2012.
3. **J. Ramírez, Israel.** *Los Sistemas Operativos.* Colombia : Universidad de los Andes, 2012.
4. Xataka móvil. *ESPACIO MASVOVIL.* [En línea] Weblogs SL. [Citado el: 20 de Marzo de 2013.] <http://www.xatakamovil.com>.
5. **Colina y Hector.** *Software Libre para Dispositivos Móviles.* Canaima : Venezolana de Telecomunicaciones-Vtelca, 2012.
6. **EcuRed.** Telefonía en Cuba. [En línea] EcuRed, 14 de Diciembre de 2010. [Citado el: 10 de Diciembre de 2012.] [http://www.ecured.cu/index.php/Telefon%C3%ADa\\_en\\_Cuba](http://www.ecured.cu/index.php/Telefon%C3%ADa_en_Cuba).
7. **Addine Fernández, Fátima y Recarey Fernández, Silvia.** *Didáctica, Teoría y Práctica.* Habana : Instituto Superior Pedagógico Enrique José Varona, 2002.
8. **UCI.** Tesis. *Diseño metodológico de la Investigación.* [En línea] Universidad de las ciencias informáticas(UCI). [Citado el: 23 de Noviembre de 2012.] [http://tesis.uci.cu/viewpage.php?page\\_id=21](http://tesis.uci.cu/viewpage.php?page_id=21).
9. **Ibáñez, y otros, y otros.** *Mobile Learning: Aprendiendo Historia con mi Teléfono, mi GPS y mi PDA.* s.l. : Universidad Autónoma de Madrid, 2008.
10. **Martínez, Carmen y J. Rueda, Antonio.** *Evaluación Continua de Alumnos Mediante el uso de Dispositivos Móviles Basados en iOS.* España : Departamento de Informática. Universidad de Jaén, 2013.
11. **Naismith, L., Lonsdale, P. y Vavoula y G.N. y Sharples, M.** Revisión de la Literatura en Tecnologías Móviles y el aprendizaje. [En línea] 2004. [Citado el: 18 de Febrero de 2013.] [www.futurelab.org.uk/research/lit\\_reviews.htm](http://www.futurelab.org.uk/research/lit_reviews.htm).. Informe11.
12. *Hacia una filosofía del m-learning.* **Nyiri, K.** Vaxjo, Suecia : Actas del Taller Internacional IEEE sobre tecnologías inalámbricas y móviles en Educación, 2002.
13. **Quinn, C.** mLearning: Mobile, Wireless, in your Pocket Learning. [En línea] [Citado el: 13 de Enero de 2013.] <http://www.linezine.com/2.1/features/cqmmwiyp.html>..
14. *Dispositivos Móviles.* **Hernández, Rocael y Morales, Miguel.** Guatemala : e-Learning en Galileo, 2013. Disponible en

[http://elearning.galileo.edu/revista/index.php?option=com\\_content&view=article&id=20&Itemid=115](http://elearning.galileo.edu/revista/index.php?option=com_content&view=article&id=20&Itemid=115).

15. **Mundaca Moraga, Bruno y J. González, Agustín.** *Evaluación y Seguimiento del Aprendizaje en Salas de Clases Utilizando Dispositivos Móviles*. Chile : Departamento de electrónica, Universidad técnica Federico Santa María, 2010.
16. **Arroyo, Miguel Vicente, Cáceres Tello, Jesús y Mencías Grande, Julio.** *Gestión de alumnos para el profesorado*. España : III CONCURSO DE IDEAS Y PROYECTOS PARA PERSONAL DE ADMINISTRACIÓN Y SERVICIOS, 2008.
17. **Franklin, Juan José, Flores, Juan José y Mori, Rosa.** *Utilización de dispositivos móviles para la gestión académica de alumnos y docentes de la Universidad de San Martín de Porres*. Lima : Universidad de San Martín de Porres, 2011.
18. **García, Daniel Adolfo.** *Integración de una Aplicación Móvil a la Intranet. Caso: Toma de Asistencia Estudiantil*. Lima : Universidad Católica del Perú, 2005.
19. Linux. [En línea] The Linux Foundation. [Citado el: 4 de Abril de 2013.] <http://www.linux.com/>.
20. **pcmag.com.** Software stack. *Enciclopedia*. [En línea] [Citado el: 10 de Diciembre de 2012.] [http://www.pcmag.com/encyclopedia\\_term/0,2542,t=software+stack&i=51702,00.asp](http://www.pcmag.com/encyclopedia_term/0,2542,t=software+stack&i=51702,00.asp).
21. **Open Handset Alliance Group.** Open Handset Alliance. [En línea] 2012. [Citado el: 4 de Abril de 2013.] <http://www.openhandsetalliance.com/>.
22. **Domínguez, José Antonio.** *Memoria: Desarrollo de aplicación móvil utilizando realidad aumentada*. Catalunya : Universidad Oberta de Catalunya, 2011. UOC-ETIG.
23. **Avila, Mejía.** *Android*. s.l. : UAM-I. Departamento de Ingeniería Eléctrica, 2012.
24. **Larramendi, Claudia Beatriz.** *Propuesta de arquitectura para desarrollo de aplicaciones compuestas para dispositivos móviles con Android*. La Habana : Universidad de las Ciencias Informáticas(UCI), 2012. TD\_05635\_12.
25. **Fernández Escribano, Gerardo.** *Introducción a Extreme Programming*. 2007.
26. **Jacobson, Ivar, Booch, Grady y Rambaugh, James.** *El Proceso Unificado de Desarrollo de Software*. Madrid, España : s.n., 2000.
27. **Fundación Universitaria de Las Palmas.** *Cursos de Orientación Profesional: Creación de aplicaciones móviles en Android*. Canarias : Servicio Canario de Empleo, 2007-2013.
28. UML Tool. *Visual Paradigm*. [En línea] Febrero de 2011. [Citado el: 20 de Enero de 2013.] <http://www.visual-paradigm.com/support/vpuml/releasenotes/642.jsp>.
29. **López, Ángel.** *Java la programación del futuro*. Uruguay : s.n., 1997.
30. **Flanagan, David.** *Java en Pocas Palabras*. 1999. 2da Edición.

31. **MediaProgramas SL.** ProgramasGratis.Net/itext 5.0.5. *iText*. [En línea] 2013. [Citado el: 4 de Junio de 2013.] <http://www.programas-gratis.net/b/itext-5.0.5>.
32. **Eclipse Foundation.** Eclipse. [En línea] Eclipse Foundation. [Citado el: 4 de Abril de 2013.] <http://www.eclipse.org/downloads/>.
33. Developers. *Get the Android SDK*. [En línea] [Citado el: 4 de Abril de 2013.] <http://developer.android.com/sdk/index.html>.
34. **Riesco y Daniel.** UML Diagrama de Clases y de Objetos. [En línea] [Citado el: 10 de Abril de 2013.] <http://sel.unsl.edu.ar/licenciatura/ingsoft2/UML-DiagramaClaseObjeto.pdf>.
35. **Androideity.** Androideity. *La Impotancia del MVC en Android*. [En línea] 2012. [Citado el: 30 de Abril de 2013.] <http://androideity.com/2012/05/10/la-importancia-del-mvc-en-android/>.
36. **Software, Tecnología por Cero-cool .** D Todo Un Poco. *Estructura de una aplicación en Android*. [En línea] 2011. [Citado el: 30 de Abril de 2013.] <http://www.dtodounpoco.net/1-estructura-de-una-aplicacion-en-android/>.
38. **Spector, J.M y Anderson, T.M.** Entornos para el aprendizaje permanente: la ergonomía, la arquitectura. 2002, Vol. 1.
39. **EcuRed.** La Telefonía en Cuba. [En línea] 14 de Diciembre de 2010. [Citado el: 24 de Noviembre de 2012.] [http://www.ecured.cu/index.php/Telefon%C3%ADa\\_en\\_Cuba](http://www.ecured.cu/index.php/Telefon%C3%ADa_en_Cuba).
40. **Camejo Valdivia, Yordanis.** *Las competencias, punto de partida para el rol de líder en el proyecto Akademos*. La Habana : Universidad de Las Ciencias Informáticas, 2007.
41. **Peláez, J.** *Arquitectura basada en capas*. s.l. : Algunas notas sobre Tecnologías Microsoft, 2009.
42. **Gracia, J.** Patrones de diseño . *Diseño de Software Orientado a Objetos*. . [En línea] [Citado el: 10 de Abril de 2013.] <http://www.ingenierossoftware.com/analisisydiseno/patrones-diseno.php>.
43. **Stufano y Cosimo.** *SOAP v1.1 Protocolo Simple de Acceso a Objetos*. 2001.
44. **CAPÍTULO 6: SOAP.** Disponible en <http://bibing.us.es/proyectos/.../TOMO+I%252F06+Capitulo+6+SOAP.pdf>.
45. **Arboleda y F, Hugo.** *Metodologías tradicionales. Modelos de Ciclo de Vida de Desarrollo de Software*. Valencia : s.n.
46. **Arañó, Eduardo, Marrero y Jessica.** *Interfaz para acceso al servidor de correos Zimbra*. La Habana : Universidad de las ciencias Informáticas(UCI), 2010.
47. **Espinosa, Dainelys, Llanes y R., Daniel.** *Plataforma de Gestión de Contenidos para Dispositivos Móviles. Versión 2.0*. La Habana : Universidad de las Ciencias Informáticas(UCI), 2010.

48. Kioskea. *Ciclo de Vida del Software*. [En línea] CCMBenchmark, 2013. [Citado el: 15 de Mayo de 2013.] <http://es.kioskea.net/contents/223-ciclo-de-vida-del-software>.
49. **Chacón, Fabio**. *Aproximación histórica a las tecnologías de la Educación a Distancia*. Caracas : Universidad Nacional Abierta, 2010.
50. Definición abc. *Tecnología*. [En línea] 2007. [Citado el: 4 de Junio de 2013.] <http://www.definicionabc.com/tecnologia/registro-2.php>.
51. DCSAPPMOVILES. *Android Tutorial - WebService*. [En línea] Tangient LLC. [Citado el: 4 de Junio de 2013.] <http://dcsappmobiles.wikispaces.com/Android+Tutorial+--+WebService>.
52. **Bianchi, Gloria**. *Desarrollo de Aplicaciones para dispositivos Móviles sobre la plataforma Android de Google*. San Juan Bosco:Argentina : Licenciatura en Informática, 2011.
53. **Báez, Manuel, Borrego, Álvaro y Cordero, Jorge**. *Introducción a Android*. Madrid : E.M.E. Editorial ©. ISBN: 978-84-96285-39-5.
54. **Chacón, Aguirre, y otros, y otros**. *DISEÑO DE UNA APLICACIÓN MÓVIL PARA LA CONSULTA ACADÉMICA DE LA FIIS-UTP*. Peru : Tesis, 2013.
55. **González Ramos, Jorge Aurelio y Pérez Gerónimo, Pedro Javier**. *Plataforma para la gestión, puesta en marcha y seguimiento de servicios de mensajería móvil basados en SMS*. La Habana : s.n., 2010.
56. **Hernández León, Rolando Alfredo y Coello González, Sayda**. *El proceso de investigación científica* . Ciudad de La Habana : Editorial Universitaria del Ministerio de Educación Superior, 2011. ISBN 978-959-16-1307-3.
57. **Alvarez de Zayas, Carlos**. *Metodología de la investigación científica* . Santiago de Cuba : s.n., 1995.
58. **Ferrer, Israel y Moreno, Luis**. android.es. *Buscador social de sitios*. [En línea] 26 de Octubre de 2011. [Citado el: 20 de Mayo de 2013.] <http://www.android.es/tag/sitios>.
59. **Silverio, Gonzalo**. programming notes::. Gonzalo Silverio. *Creacion de un PDF usando la libreria iText*. [En línea] 20 de Diciembre de 2010 . [Citado el: 19 de Mayo de 2013.] <http://gonzasilve.wordpress.com/2010/12/20/creacion-de-un-pdf-usando-la-libreria-itext/>.
60. Felinfo: Java, Linux, Virtualización. Open Source. *Generar un PDF en Java usando iText JAR* . [En línea] 28 de Julio de 2009. [Citado el: 19 de Mayo de 2013.] <http://felinfo.blogspot.com/2009/07/generar-un-pdf-en-java-usando-itext-jar.html>.
61. Instituto Nacional de Formación Docente. *Registro y Validez de Títulos*. [En línea] WordPress, 2011. [Citado el: 11 de Mayo de 2013.] <http://portales.educacion.gov.ar/infed/registro-y-validez-de-titulos/>.

62. **PandaID Soluciones, C.A.** ¿Qué es la Autenticación Fuerte? [En línea] 11 de Octubre de 2012. [Citado el: 2 de Diciembre de 2012.] <http://www.pandaid.com/que-es-la-autenticacion-fuerte/>.
63. **Rosa.** Seguridad Informática. [En línea] Wordpress, 28 de Enero de 2013. [Citado el: 20 de Enero de 2013.] <http://enboliviacom.wordpress.com/2013/01/28/seguridad-informatica/>.
64. **Microsoft.** Autenticación. [En línea] 2013. [Citado el: 29 de Enero de 2013.] <http://msdn.microsoft.com/es-es/library/syf5yeat.aspx>.
65. **Safelayer Labs.** Investigación y desarrollo: Seguridad, confianza y privacidad. *Contraseñas de un solo uso (One Time Passwords)*. [En línea] 2010. [Citado el: 13 de Diciembre de 2012.] <http://labs.safelayer.com/es/investigacion-y-desarrollo/areas/seguridad-confianza-y-privacidad/446-contrasenas-de-un-solo-uso>.
66. **Google Authenticator.** Apps Documentation & Support. [En línea] Google Team. [Citado el: 12 de Enero de 2013.] <http://support.google.com/a/bin/answer.py?hl=en&answer=1037451>.
67. **M'Raihi.** [En línea] Diciembre de 2005. [Citado el: 12 de Enero de 2013.] <http://tools.ietf.org/html/rfc4226>.
68. **M'Raihi, D.** Time-Based One-Time Password Algorithm. [En línea] Mayo de 2011. [Citado el: 13 de Diciembre de 2012.] <http://tools.ietf.org/html/rfc6238>.
69. **Seco, José Antonio González.** Microsoft .NET. [En línea] 29 de Septiembre de 2006. [Citado el: 12 de Enero de 2013.] <http://www.devjoker.com/contenidos/articulos/88/Introduccion-a-NET.aspx>.
70. **Kuhn, M.** OTPW. [En línea] 2003. <http://www.cl.cam.ac.uk/~mgk25/otpw.html>.
71. **Borghello, C.** Seguridad de la Información. *Seguridad de la Información*. [En línea] 2009. [Citado el: 2 de Diciembre de 2012.] <http://www.segu-info.com.ar/logica/identificacion>.
72. **Charles, J.** IEEEExplore Digital Library. [En línea] Mayo de 1999. [Citado el: 20 de Enero de 2013.] <http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=762786&url=http%3A%2F%2Fieeexplore.ieee.org%2Fielx5%2F2%2F16523%2F00762786>. ISSN : 0018-9162.
73. **Barra, Jonnathan De La.** [En línea] 18 de Septiembre de 2010. [Citado el: 16 de Enero de 2013.] <http://justindeveloper.wordpress.com/2010/09/18/microsoft-solutions-framework/>.
74. **Microsoft.** MSDN. *MSDN*. [En línea] 2013. [Citado el: 20 de nov de 2012.] [http://msdn.microsoft.com/es-es/library/dd566231\(v=vs.100\).aspx](http://msdn.microsoft.com/es-es/library/dd566231(v=vs.100).aspx).
75. —. MSDN. *MSDN*. [En línea] 2013. [Citado el: 20 de marzo de 2012.] [http://msdn.microsoft.com/es-es/library/w0x726c2\(v=vs.100\).aspx](http://msdn.microsoft.com/es-es/library/w0x726c2(v=vs.100).aspx).



76. —. MSDN. *MSDN*. [En línea] 2013. [Citado el: 21 de noviembre de 2012.] [http://msdn.microsoft.com/es-es/library/52f3sw5c\(v=vs.100\).aspx](http://msdn.microsoft.com/es-es/library/52f3sw5c(v=vs.100).aspx).
77. —. MSDN. *MSDN*. [En línea] 2013. [Citado el: 21 de noviembre de 2011.] [http://msdn.microsoft.com/es-es/library/kx37x362\(v=vs.100\).aspx](http://msdn.microsoft.com/es-es/library/kx37x362(v=vs.100).aspx).
78. **Sama, Condesa**. Androiddeity. *Arquitectura de android*. [En línea] 04 de junio de 2011. [Citado el: 21 de noviembre de 2012.] <http://androiddeity.com/2011/07/04/arquitectura-de-android/>.
79. —. Androiddeity. *Arquitectura de Android*. [En línea] 4 de Julio de 2011. [Citado el: 20 de Enero de 2013.] <http://androiddeity.com/2011/07/04/arquitectura-de-android/>.
80. **García, Joaquín**. Desarrollo de Software Orientado a Objetos. *Desarrollo de Software Orientado a Objetos*. [En línea] 2005. [Citado el: 12 de noviembre de 2012.] <http://www.ingenierossoftware.com/analisisydiseno/uml.php>.
81. **G., José David**. Tecnologías, software libres y otras inquietudes. *Tecnologías, software libres y otras inquietudes*. [En línea] 08 de febrero de 2010. [Citado el: 23 de enero de 2013.] <http://davidjguru.com/2010/02/08/la-arquitectura-de-tres-capas-introduccion/>.
82. **Ferrás, Claudia Beatriz Larramendi**. Propuesta de arquitectura para desarrollo de aplicaciones compuestas para dispositivos móviles con Android. *Propuesta de arquitectura para desarrollo de aplicaciones compuestas para dispositivos móviles con Android*. La Habana : s.n., 2012.
83. **Wesley, Adisson**. *Design Patterns. Elements of Reusable Object-Oriented*. 2003.
84. **Schackow, Stefan**. *ASP.NET 2.0 Security, Membership, and Role Management*. Wiley Publishing : Indianapolis, Indiana, 2006.
85. **Microsoft**. Microsoft. [En línea] Octubre de 2005. [Citado el: 14 de Febrero de 2013.] <http://msdn.microsoft.com/en-us/library/aa479030.aspx>.
86. **Lasater, Chris**. Codeproject. [En línea] 2007. [Citado el: 1 de Marzo de 2013.] <http://www.codeproject.com/KB/books/DesignPatterns.aspx>.
87. **Dofactory**. Dofactory. [En línea] 2001. [Citado el: 1 de Marzo de 2013.] <http://www.dofactory.com/Patterns/PatternCommand.aspx>.
88. **Microsoft**. Share Point. Tech Center. *Share Point. Tech Center*. [En línea] 2013. [Citado el: 16 de enero de 2013.] <http://technet.microsoft.com/en-us/sharepoint/bb267375.aspx>.
89. **Etcheverry, Lorena**. Pedeciba. [En línea] Marzo de 2010. [Citado el: 1 de Abril de 2013.] [www.pedeciba.edu.uy/bioinformatica/sibdyw/Clase\\_3.pdf](http://www.pedeciba.edu.uy/bioinformatica/sibdyw/Clase_3.pdf).
90. **Visual Paradigm International Ltd**. Free Download Manager. *Free Download Manager*. [En línea] Visual Paradigm International Ltd., 05 de marzo de 2007. [Citado el: 12 de nov de



2012.]

[http://www.freedownloadmanager.org/es/downloads/Paradigma\\_Visual\\_para\\_UML\\_%5Bcuenta\\_de\\_Plataforma\\_de\\_Java\\_14715\\_p/](http://www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_%5Bcuenta_de_Plataforma_de_Java_14715_p/).

91. **Oracle**. Java. Acerca de la tecnología Java. *Acerca de la tecnología Java*. [En línea]

Oracle. [Citado el: 05 de enero de 2013.]

[http://www.java.com/es/download/faq/whatis\\_java.xml](http://www.java.com/es/download/faq/whatis_java.xml).

92. **The PostgreSQL Global Development Group**. PostgreSQL . *PostgreSQL* . [En línea]

1996 - 2013. [Citado el: 03 de diciembre de 2012.] <http://www.postgresql.org/>.

93. **Preeman, Roger S.** Ingeniería de Software. Un enfoque práctico. 5ta Edición. [aut. libro]

Darrell Ince. *Ingeniería de Software. Un enfoque práctico. 5ta Edición*. 2002.

94. **Huanca, Daniel**. HerrorSoft. *HerrorSoft*. [En línea] 2009. [Citado el: 29 de enero de 2013.]

<http://herrorsoft.zxq.net/pruebacajanegra.html>.

95. **Vegas, Sira , Moreno, Ana María y Juristo, Natalia**. *Técnicas de Evaluación de Software*. 2006.

96. **Vivas, Alex**. KUMACHE.TICKS. Centro de Programación. *KUMACHE.TICKS. Centro de Programación*. [En línea] 26 de agosto de 2010. [Citado el: 15 de 11 de 2012.]

<https://sites.google.com/site/centrodeprogramacion/planeta-sql/postgresql>.

**ANEXOS****Anexos 1** Registro de no conformidades 1ra Iteración.

<b>No.</b>	<b>No Conformidad</b>
NC#1	El tiempo de espera para la solicitud del fichero texto agotado.
NC#2	El sistema no realiza ninguna acción cuando sin ser cargado los grupos y se desea registrar la asistencia.
NC#3	El sistema permite tomar la asistencia solo a algunos estudiantes de un mismo grupo docente.
NC#4	Una vez entrado el nombre de la asignatura con letra inicial mayúscula, cuando se inicia el registro de asistencia el sistema lo muestra con letra minúscula.

### **Glosario de Términos.**

#### **Android:**

Es una plataforma móvil, que incluye un sistema operativo móvil basado en Linux, junto con un conjunto de aplicaciones middleware y aplicaciones nativas. Está enfocado para ser utilizado en dispositivos móviles como teléfonos inteligentes, tabletas, Google TV y otros dispositivos. Es desarrollado por la Open Handset Alliance, la cual es liderada por Google. Este sistema por lo general maneja aplicaciones descargadas desde Google Play, aunque pueden ser instaladas por otros medios.

#### **Runtime:**

Android incluye un set de bibliotecas base que proporcionan la mayor parte de las funciones disponibles en las bibliotecas base del lenguaje Java. Cada aplicación Android corre su propio proceso, con su propia instancia de la máquina virtual Dalvik.

#### **XML:**

Es un lenguaje desarrollado por el W3 Consortium para permitir la descripción de información contenida en la red de redes a través de estándares y formatos comunes, de manera que tanto los usuarios de Internet como programas específicos (agentes) puedan buscar, comparar y compartir información en la red. El formato de XML es muy parecido al del HTML aunque no es una extensión ni un componente de éste. Sirve para que muchos programas interpreten bien cualquier tipo de dato y se comuniquen entre ellos sin intervención humana. Constituye el lenguaje base de los Servicios Web.

#### **Herramienta Case:**

Ingeniería de sistemas asistida por ordenador (Computer-Aided Systems Engineering - CASE) es la aplicación de tecnología informática a las actividades, las técnicas y las metodologías propias de desarrollo de sistemas. Su objetivo es automatizar o apoyar una o más fases del ciclo de vida del desarrollo de sistemas.

#### **Aplicación:**

Es el programa que el usuario activa para trabajar en el ordenador. Existen muchos programas de ordenador que pueden clasificarse como aplicación. Generalmente se les conoce como Software.

**Framework:**

Es una estructura de soporte definida en la cual otro proyecto de software puede ser organizado y desarrollado.