

**DESARROLLO DISEÑO E IMPLEMENTACIÓN DE UN *APP ANDROID* PARA
CONSULTA DE NOTAS Y HORARIOS DE CLASE EN LA COMUNIDAD
ESTUDIANTIL DE LA FUNDACIÓN CIDCA**

“MiCIDCA”

**RICHARD KEITH CÁRDENAS VESGA
JAAHSER JAIRO MÉNDEZ RODRÍGUEZ
JOSÉ ARMANDO REYES BRAVO**



**FUNDACIÓN UNIVERSITARIA LOS LIBERTADORES
FACULTAD DE INGENIERÍAS
INGENIERÍA DE SISTEMAS
BOGOTÁ D.C.
2015**

**DESARROLLO DISEÑO E IMPLEMENTACIÓN DE UN *APP ANDROID* PARA
CONSULTA DE NOTAS Y HORARIOS DE CLASE EN LA COMUNIDAD
ESTUDIANTIL DE LA FUNDACIÓN CIDCA**

“MiCIDCA”

**RICHARD KEITH CÁRDENAS VESGA
JAAHSER JAIRO MÉNDEZ RODRÍGUEZ
JOSÉ ARMANDO REYES BRAVO**

**TRABAJO DE GRADO PARA OPTAR AL TÍTULO DE INGENIERO DE
SISTEMAS**

**ASESOR
HELMAN MAURICIO ROJAS
COORDINADOR SISTEMAS Y TECNOLOGÍA**



**FUNDACIÓN UNIVERSITARIA LOS LIBERTADORES
FACULTAD DE INGENIERÍAS
INGENIERÍA DE SISTEMAS
BOGOTÁ D.C.**

2015

Nota de Aceptación

Presidente del Jurado

Firma del Jurado

Firma del Jurado

Bogotá, D.C. Mayo de 2015

DEDICATORIA

Con todo cariño y aprecio dedicamos esta investigación a Dios, a nuestros padres y familia, por su presencia y apoyo constante en este camino hacia la profesionalización.

AGRADECIMIENTOS

A Dios, por su acompañamiento durante estos días de guía en nuestro proyecto. A nuestras familias, por la constancia para animarnos, en los momentos difíciles que afrontamos, por su acompañamiento diario y afecto, son un pilar para cumplir esta nueva meta.

Al director de tesis, Helman Mauricio Rojas, por su asesoría y apoyo académico, al ingeniero Augusto Ángel, quien siempre tuvo la disposición para despejar las dudas. Al profesor Jorge Eliecer Vargas, por todos los conocimientos impartidos durante el pregrado.

A la Fundación Universitaria Los Libertadores y a CIDCA, y al cuerpo de docentes, siempre dispuestos a brindarnos sus experiencias y conocimientos para alcanzar esta meta tan añorada personal y profesionalmente.

Mutuo agradecimiento como compañeros en este sueño, porque a pesar de los debates, preocupaciones y sacrificios, estuvimos unidos con toda responsabilidad, ánimo y constancia para salvar cualquier inconveniente o reto hasta lograr que el producto de este trabajo, fuera la satisfacción de aportar a la comunidad estudiantil, una solución informática que pueda implementarse, para colocar a la Institución, a la vanguardia, frente a la exigencia tecnológica para el manejo de la información.

Por último gracias, a todas las personas que de una y otra manera, aportaron su granito de arena y estuvieron dispuestas a colaborarnos con información y saberes, para que este proyecto fuera culminado con éxito.

AUTORES

CONTENIDO

	Pág.
INTRODUCCIÓN	14
1. PROBLEMA.....	15
1.1. DEFINICIÓN DEL PROBLEMA	15
1.2. PLANTEAMIENTO DEL PROBLEMA.....	15
1.3. JUSTIFICACIÓN.....	16
1.4. DELIMITACIÓN	17
1.4.1. Espacial	17
1.4.2. Delimitación metodológica	18
1.4.3. Tecnológica.....	18
1.4.4. Cronológica.....	18
1.4.5. Financiera	19
1.4.6. Metodológica.....	19
1.5. ALCANCE	20
1.6. OBJETIVOS.....	21
1.6.1. Objetivo general.....	21
1.6.2. Objetivos específicos	21
1.7. ESTUDIO DE FACTIBILIDAD DEL PROYECTO.....	21
2. MARCOS DE REFERENCIA	25
2.1. ANTECEDENTES.....	25
2.2. MARCO INSTITUCIONAL	25
2.3. MARCO TEÓRICO	25
2.3.1. Ingeniería de <i>software</i>	25

2.3.2. <i>Android</i> SO	28
2.4. MARCO METODOLÓGICO	31
2.4.1. Metodología X.P.	31
2.4.2. Java	33
2.4.3. Eclipse	35
2.4.4. NetBeans	36
2.4.5. Web service	36
2.4.6. Oracle bases de datos	37
2.4.7. <i>Apache tomcat</i>	37
2.4.8. Lenguaje de Modelamiento Unificado (UML).	38
2.5. MARCO LEGAL	38
3. METODOLOGÍA	40
3.1. METODOLOGÍA <i>XP</i>	40
3.2. FASES METODOLÓGICAS PARA “ <i>MiCIDCA</i> ”	40
3.2.1. Fase I. Planificación y levantamiento de información	40
3.2.2. Fase II. Análisis y diseño de “ <i>MiCIDCA</i> ”	41
3.2.3. Fase III. Desarrollo de “ <i>MiCIDCA</i> ”.	41
3.2.4. Fase IV. Pruebas y ajustes de “ <i>MiCIDCA</i> ”	41
3.2.5. Fase V. Montaje, presentación y entrega final de “ <i>MiCIDCA</i> ”.....	41
3.2.6. Fase VI. Elaboración de manual de usuario para “ <i>MiCIDCA</i> ”	42
3.3. ANÁLISIS Y REQUERIMIENTOS	42
3.3.1. Requerimientos no funcionales. Contempla los siguientes:	42
3.3.2. Requerimientos funcionales.....	43
3.4. DIAGNÓSTICO PARA DESARROLLO DE “ <i>MiCIDCA</i> ” EJEMPLO	44
3.5. DISEÑO DE “ <i>MiCIDCA</i> ”	45
3.6. ANÁLISIS DEL SISTEMA	45
3.7. ANÁLISIS DE REQUERIMIENTOS	46
3.8. MATERIALES Y MÉTODOS.....	52

3.8.1 Materiales	52
3.8.2. Metodologías. Se aplicaron las siguientes:	53
3.9. DESARROLLO DEL PROYECTO	53
4. ANÁLISIS Y RESULTADOS	55
4.1. RESULTADOS DE LAS FASES METODOLÓGICAS.....	55
4.1.1. Fase I. Planificación y levantamiento de información	55
4.1.2. Fase II. Análisis y diseño de “MiCIDCA”	55
4.1.3. Fase III. Desarrollo de “MiCIDCA”	63
4.1.4. Fase IV. Pruebas y ajustes de “MiCIDCA”	63
4.1.5. Fase V. Montaje, presentación y entrega final de “MiCIDCA”.....	64
4.1.6. Fase VI. Elaboración de manuales de usuario para “MiCIDCA	64
5. CONCLUSIONES	65
6. RECOMENDACIONES	67
REFERENCIAS	68
ANEXOS.....	70

LISTA DE CUADROS

	Pág.
Cuadro 1. Cronograma de actividades <i>MICIDCA</i>	18
Cuadro 2 .Presupuesto-Estudio de factibilidad para	19
Cuadro 3. Requerimientos No funcionales: <i>software y Hardware</i>	42
Cuadro 4. Caso de uso Consulta notas estudiantes	60
Cuadro 5.Caso de uso-Consulta horario estudiantes	61
Cuadro 6. Caso de uso-Consulta horario docentes	62

LISTA DE FOTOS

	Pág.
Foto 1. Fundación CIDCA.....	17

LISTA DE FIGURAS

	Pág.
Figura 1. Proceso desarrollo de <i>software</i>	28
Figura 2. Estructura del sistema	31
Figura 3. Diagrama de Gantt <i>MiCIDCA</i>	54
Figura 4. Diagramas de Clases para “ <i>MiCIDCA</i> ”	56

LISTA DE ANEXOS

	Pág.
Anexo A. Manual de usuario con privilegios para “ <i>MiCIDCA</i> ”	71

RESUMEN

“MiCIDCA” es un *App Android* con acceso a la información requerida, a través de un *web service*, desarrollado para consultar en tiempo real, las notas y horarios de clases para estudiantes y horarios para docentes del CIDCA.

El presente trabajo de investigación para el desarrollo y logro de los objetivos establecidos se ha estructurado en su informe final de la siguiente manera:

Se describe la problemática del mecanismo actual utilizado en la consulta de notas y horarios de estudiantes y horarios de docentes de la Fundación CIDCA, identificando errores y necesidades del mismo. Se monta el prototipo del *App Android*, validando si responde como solución propuesta a la problemática encontrada. Se justifica la razón del desarrollo, diseño e implementación, del aplicativo y se establecen objetivos, tanto general y específicos del proyecto, cerrando el primer capítulo, con la presentación de limitaciones y el alcance de **“MiCIDCA”**.

Se construye el marco referencial y teórico-conceptual que sustenta las bases teóricas y se establecen las fases metodológicas de trabajo, así:

Fase I. Levantamiento de Información.

Fase II. Análisis y Diseño de **“MiCIDCA”**.

Fase III. Desarrollo de **“MiCIDCA”**.

Fase IV. Pruebas y Ajustes de **“MiCIDCA”**.

Fase V. Montaje y entrega final de **“MiCIDCA”**.

Fase VI. Elaboración de Manual para **“MiCIDCA”**.

PALABRAS CLAVES: CIDCA; notas, horarios, estudiantes, docentes, *App Android*, *web service*, *Java*, *consulta*

ABSTRACT

"MiCIDCA" is an App Android with access to the needed information, across a web service, developed to consult real time, the notes and schedules of classes for students and schedules for teachers of the CIDCA. The present work of investigation for the development and achievement of the established aims has been structured in his final report of the following way:

There are described the problematics of the current mechanism used in the consultation of notes and students' schedules and teachers' schedules of the Foundation CIDCA, identifying mistakes and needs of the same one. The prototype of the App Android is mounted, validating if it answers as solution proposed the opposing problematics. There justifies itself the reason of the development, design and implementation, of the Applicative one and they establish aims, so much general and specifics of the project, closing the first chapter, with the presentation of limitations and the scope of "MiCIDCA".

There is constructed the referential and theoretical - conceptual frame that sustains the theoretical bases and the methodological phases of work are established, this way:

Phase I. Raising Information.

Phase the II. Analysis and Design of "MiCIDCA".

Phase the III. Development of "MiCIDCA".

Phase the IV. Tests and Adjustments of "MiCIDCA".

Phase V. Assembly and final delivery of "MiCIDCA".

Phase the VI. Production of Manual for "MiCIDCA".

KEY WORDS: *CIDCA; notes, schedules, students, teachers, App Android, web service, Java, consultation*

INTRODUCCIÓN

El presente proyecto tiene como finalidad elaborar e implementar un *App* Android para la Fundación CIDCA como alternativa para la consulta de notas y horarios de los estudiantes y horarios de los docentes de la Institución.

El auge de los servicios que se proveen hoy en día en Internet, sumado al avance tecnológico, junto con la necesidad de acceso a la información en tiempo real y la tendencia hacia dispositivos más pequeños y rápidos incluyendo a los teléfonos inteligentes y las tabletas PC, son los factores determinantes del surgimiento de las mismas.

El desarrollo de una aplicación móvil para la Fundación Universitaria CIDCA representará un valor agregado a los servicios que la Institución brinda, porque como alumno se valoraran mucho los beneficios de estar a la vanguardia de las nuevas tecnologías, con el fin de estar informados de todo el andamiaje académico, si se puede acceder a la plataforma estudiantil, desde cualquier lugar, mediante los dispositivos móviles que prefiera o posea en el momento.

El aplicativo es móvil y para utilizar con *Android*, conformado por tres (3) módulos principales: consulta de horarios de estudiantes, de docentes y otro de consulta de notas de estudiantes, y funcionará al conectarse a la base de datos del sistema de información de la Fundación CIDCA, mediante un *web service*, desarrollado en *Java* e instalado en un servidor de aplicaciones de la Institución Educativa, mostrando en la pantalla del dispositivo desde donde se consulta, en forma inmediata la información que se requiere.

El sistema contará con una pantalla inicial de login, que validará la información contra la base de datos y da acceso, según el perfil establecido por la Fundación Universitaria CIDCA a los estudiantes o a los docentes para su consulta.

1. PROBLEMA

1.1. DEFINICIÓN DEL PROBLEMA

En la Fundación CIDCA, se ha identificado la dificultad para conocer en tiempo real de consulta, las notas y horarios por parte de los estudiantes y los horarios de los docentes, de manera que estos dos grupos de la comunidad estudiantil puedan llevar el promedio o resultado de los cortes de notas programados por la institución.

La problemática se evidencia cuando el estudiante o el docente requieren el estado de sus notas y horarios respectivamente y al no tener al alcance de la mano un PC con internet Explorer (IE versiones anteriores a 9), no pueden acceder al sistema de información que se maneja en la Fundación CIDCA, lo que puede causar inconvenientes, si por error humano en la alimentación de datos, no se registran correctamente las notas quedando expuestos a perder una materia o sencillamente en cuanto a horarios de clase, por desconocimiento, faltar a clase o no llegar a tiempo, por no saber cuándo se impartirá la cátedra.

Así mismo de alguna forma, se limita a los estudiantes y docentes perdiendo el disfrute y la facilidad que ofrecen los avances tecnológicos al no poder hacer uso de los mismo, en el manejo de la información, si se tiene en cuenta que en su gran mayoría, la comunidad estudiantil en CIDCA, utilizan dispositivos móviles para realizar consultas académicas.

1.2. PLANTEAMIENTO DEL PROBLEMA

De acuerdo a la anterior descripción, se ha fundamentado como pregunta orientadora de la investigación, la siguiente:

¿El diseño, desarrollo e implementación de un *App Android* incluido en el sistema de información de la Fundación CIDCA, permitirá que los estudiantes y los docentes, puedan realizar en tiempo real, las notas y horarios por parte de los estudiantes y los horarios por los docentes?

1.3. JUSTIFICACIÓN

Actualmente para los estudiantes es algo complicado poder consultar sus notas y horarios de clase en cualquier momento y lugar, ya que muchas veces cuando se necesita saber sobre estos dos temas, por lo general se hace desde las instalaciones de la Fundación. Así mismo para los docentes tener que estar consultando sus horarios de asignación de clases minutos antes de dar inicio a las mismas, o tener que tenerlas referenciadas de forma manual en alguna libreta, lo puede hacer fallar a la cátedra, perjudicándose a sí mismo y a los estudiantes.

Debido al uso masivo de los *Smartphone* y las *App Android*, es una combinación que se está desarrollando a gran velocidad, cada día es más común realizar transacciones, consultas y operaciones desde nuestros dispositivos móviles, la capacidad de estos ha evolucionado exponencialmente en los últimos 4 años, el servicio de internet cada día es mejor y a un costo más asequible, las aplicaciones para dispositivos móviles no paran de incursionar en los diferentes ámbitos de la sociedad y es esta una de las razones que con mayor incidencia hace que la Fundación requiera del diseño, desarrollo e implementación del *App*.

Aprovechando todas las facilidades que ofrece esta nueva tecnología, se plantea desarrollar e implementar un *App* con el que se dé solución en línea e inmediata a la problemática descrita.

Los estudiantes no tendrán la necesidad de buscar un equipo o PC que cuente con las características necesarias para poder hacer su consulta, sino que podrán

hacerlo las **24** horas del día desde su teléfono móvil, con un aplicativo liviano, confiable, amigable, funcional y fácil de usar.

Con el éxito del aplicativo este se presentara como el medio primario de consulta entre los estudiantes, dejando al actual sistema como secundario, tener estas dos (2) alternativas será bueno para la universidad ya que cuando se presente alguna eventualidad con alguna de ellas, la otra servirá como medio de contingencia para no interrumpir el servicio, esto ya que no están vinculadas como medio de acceso.

1.4. DELIMITACIÓN

1.4.1. Espacial. La implementación del aplicativo **MICIDCA**, se llevó a cabo, en las instalaciones de la CIDCA ubicada en la Carrera 13 No. 60-44 Barrio Chapinero, en la ciudad de Bogotá. **Ver foto 1.**

Foto 1. Fundación CIDCA



Fuente: Autores

1.4.2. Delimitación metodológica. Se ha escogido la metodología *XP*, porque permite brindar un producto de *software* que al iniciar e ir transcurriendo su diseño y desarrollo, se puede incluir requerimientos que el cliente necesite, antes de cerrar, cubriendo las necesidades específicas de una empresa o todo tipo de institución.

1.4.3. Tecnológica. El *App Android “MiCIDCA”*, ha sido diseñado con la limitante de prestar la función única para la consulta de notas y horarios de clase de los estudiantes y consulta de horarios de los docentes, sin extenderse a otras opciones académicas. Pero igual este prototipo de mecanismo de consulta puede servir para implementar otras consultas en el futuro.

1.4.4. Cronológica. Para el desarrollo de *App Android “MiCIDCA”*, se ha planteado en el **cuadro 1**, el Cronograma de Actividades a seguir por los desarrolladores del aplicativo así:

Cuadro 1. Cronograma de actividades MICIDCA

	Modo de tarea	Nombre de tarea	Duración	Comienzo	Fin	Predecesor	Nombres de los recursos
1		Planificación					Richard Keith Cárdenas, José
2		Determinar el alcance del proyecto	5 días	jue 15/01/15	mié 21/01/15		José Armando Reyes
3		Levantamiento de la información	3 días	jue 22/01/15	lun 26/01/15	2	José Armando Reyes
4		Antecedentes	3 días	mar 27/01/15	jue 29/01/15	3	José Armando Reyes
5		Elaboración del marco teórico	10 días	vie 30/01/15	jue 12/02/15	4	José Armando Reyes
6		Análisis de recursos	4 días	vie 30/01/15	mié 04/02/15		Jaahser Jairo Mendez
7		Análisis de costos y presupuesto	4 días	vie 13/02/15	mié 18/02/15	5	José Armando Reyes
8		Análisis					
9		Identificación de actores	5 días	dom 01/03/15	jue 05/03/15		José Armando Reyes
10		Definición de casos de uso	3 días	vie 06/03/15	mar 10/03/15	9	José Armando Reyes
11		Documentación de casos de uso	5 días	dom 22/03/15	jue 26/03/15		Jaahser Jairo Mendez
12		Identificación de necesidades	5 días	mié 11/03/15	mar 17/03/15	10	José Armando Reyes
13		Diagramas de secuencia actividad y modelo de análisis	5 días	vie 27/03/15	jue 02/04/15	11	Jaahser Jairo Mendez
14		Diseño					
15		Diseño de la estructura física	3 días	dom 01/02/15	mar 03/02/15		Richard Keith Cárdenas
16		Identificación de objetos	2 días	mié 04/02/15	jue 05/02/15	15	Richard Keith Cárdenas
17		Diseño de interfaces	2 días	mié 18/03/15	jue 19/03/15	12	José Armando Reyes
18		Ingreso de estudiantes y docentes	2 días	vie 20/03/15	lun 23/03/15	17	José Armando Reyes
19		Consulta de notas y horarios	2 días	mar 24/03/15	mié 25/03/15	18	José Armando Reyes
20		Formulario de notas estudiantes	2 días	jue 26/03/15	vie 27/03/15	19	José Armando Reyes
21		Formulario de horario estudiantes	2 días	lun 30/03/15	mar 31/03/15	20	José Armando Reyes
22		Formulario de horario docentes	2 días	mié 01/04/15	jue 02/04/15	21	José Armando Reyes
23		Implementación					
24		Generación de código Web Service	30 días	vie 06/02/15	jue 19/03/15	16	Richard Keith Cárdenas
25		Documentación del código	3 días	vie 15/05/15	mar 19/05/15	28	Jaahser Jairo Mendez
26		Generación de código App	30 días	vie 20/03/15	jue 30/04/15	24	Richard Keith Cárdenas
27		Documentación del código App	3 días	mié 20/05/15	vie 22/05/15	25	Jaahser Jairo Mendez
28		Pruebas	10 días	vie 01/05/15	jue 14/05/15	26	Richard Keith Cárdenas
29		Generación de manuales	5 días	lun 25/05/15	vie 29/05/15	27	Jaahser Jairo Mendez
30		Evaluación de la herramienta	5 días	vie 15/05/15	jue 21/05/15	28	José Armando Reyes
31		Corrección de errores	3 días	vie 15/05/15	mar 19/05/15	28	Richard Keith Cárdenas
32		Puesta en producción	2 días	lun 01/06/15	mar 02/06/15	29	Richard Keith Cárdenas
33		Entrega	2 días	mié 03/06/15	jue 04/06/15	32	José Armando Reyes, Jaahse

Fuente: Autores

1.4.5. Financiera. Para el desarrollo del App Android **“MiCIDCA”**, se ha estipulado el siguiente presupuesto. **Ver cuadro 2.**

Cuadro 2 .Presupuesto-Estudio de factibilidad para

<u>COSTOS</u>	Enero	Febrero	Marzo	Abril	Mayo	<u>Total</u>
Recurso Humano						
Desarrollador	\$ 93.750,00	\$ 93.750,00	\$ 93.750,00	\$ 93.750,00	\$ 187.500,00	\$ 562.500,00
Administrador	\$ 93.750,00	\$ 93.750,00	\$ 93.750,00	\$ 93.750,00	\$ 187.500,00	\$ 562.500,00
Documentador	\$ 93.750,00	\$ 93.750,00	\$ 93.750,00	\$ 93.750,00	\$ 187.500,00	\$ 562.500,00
Otros Costos						
Transportes	\$ 20.000,00	\$ 20.000,00	\$ 20.000,00	\$ 20.000,00	\$ 30.000,00	\$ 110.000,00
Acceso a GooglePlay				\$ 64.200,00		\$ 64.200,00
Papelería y Oficina	\$ 10.000,00	\$ 10.000,00	\$ 10.000,00	\$ 10.000,00	\$ 50.000,00	\$ 90.000,00
Infraestructura						
Servidor de Pruebas	\$ 50.000,00	\$ 50.000,00	\$ 50.000,00	\$ 50.000,00	\$ 50.000,00	\$ 250.000,00
PC para desarrollo	\$ 50.000,00	\$ 50.000,00	\$ 50.000,00	\$ 50.000,00	\$ 50.000,00	\$ 250.000,00
PC para documentación	\$ 50.000,00	\$ 50.000,00	\$ 50.000,00	\$ 50.000,00	\$ 50.000,00	\$ 250.000,00
PC para Administración	\$ 50.000,00	\$ 50.000,00	\$ 50.000,00	\$ 50.000,00	\$ 50.000,00	\$ 250.000,00
Servicio de Internet	\$ 40.000,00	\$ 40.000,00	\$ 40.000,00	\$ 40.000,00	\$ 40.000,00	\$ 200.000,00
<u>Total</u>	\$ 551.250,00	\$ 551.250,00	\$ 551.250,00	\$ 615.450,00	\$ 882.500,00	\$ 3.151.700,00

Fuente: Elaboración propia

1.4.6. Metodológica. El diseño y desarrollo del App Android **“MiCIDCA”** en su realización comprendió seis (6) fases metodológicas denominadas de la siguiente manera:

Fase I. Planificación y levantamiento de información

Fase II. Análisis y diseño de **“MiCIDCA”**

Fase III. Desarrollo de **“MiCIDCA”**

Fase IV. Pruebas y Ajustes de **“MiCIDCA”**

Fase V. Montaje y presentación final de “**MiCIDCA**”

Fase VI. Elaboración de Manual para de “**MiCIDCA**”.

1.5. ALCANCE

- Únicamente consulta de horario de estudiantes
- Únicamente consulta de notas de estudiantes
- Únicamente consulta de horarios de docentes
- Desarrollo para aplicativos *Android* en versiones superiores a *Ice Cream Sandwich*
- Nivel de acceso a consulta del sistema de información de Fundación CIDCA.
- Documentación de instalación de *App Android* en cualquiera versión.
- Documentación funcional dependiendo del alcance de actividades del *software*.
- Terminado el proyecto será entregado código de aplicativo y del *web service*.
- Implementado el *software*, su administración será responsabilidad de la Fundación CIDCA.
- Después de Implementado el *software*, se dará un mes de soporte sobre:
 - ✓ Errores masivos con él instalador *.jar* almacenado en *Google Play*.
 - ✓ Errores masivos de conexión en los que la causa sea el *web service*.
 - ✓ Errores masivos de la visualización del aplicativo.
 - ✓ La aplicación es gratis, para la Fundación CIDCA y para la comunidad estudiantil.

1.6. OBJETIVOS

1.6.1. Objetivo general

Diseñar, desarrollar e implementar un *App Android* para consulta de notas y horarios para los estudiantes y horarios para los docentes de la Fundación CIDCA.

1.6.2. Objetivos específicos

- Diseñar y desarrollar un aplicativo con todos los estándares de calidad conocidos (interfaz amigable, fácil de usar, ligera y funcional).
- Optimizar la generación de consultas a la base de datos de la Fundación CIDCA, con rapidez y confiabilidad a través del aplicativo.
- Divulgar y promover el uso del aplicativo como única herramienta de consulta de notas y horarios de estudiantes y horarios de docentes la Fundación CIDCA.
- Brindar a estudiantes y docentes la herramienta de fácil acceso, con la que en cualquier momento, lugar y a través de diferentes dispositivos móviles pueda consultar notas y horarios, garantizando una alta disponibilidad.

1.7. ESTUDIO DE FACTIBILIDAD DEL PROYECTO

1.7.2. Estudio factibilidad administrativo-Recurso humano. El proyecto cuenta con 3 Tecnólogos con experiencia en diferentes áreas, un Líder de contratos de Tecnología, un Director de Proyectos de Seguridad y Servicio de Soporte Técnico, por lo que los costos se asociarán al valor promedio de los salarios actuales de los empleados, ya que su costo se traduce en horas basadas en salario actual y cumplen las siguientes funciones dentro del desarrollo del *App*:

- **Jasher Jairo Méndez:** documentación y diseño.
- **José Armando Reyes Bravo:** administración y documentación del proyecto.
- **Richard Keith Cárdenas:** diseño, modelado, desarrollo e implementación de *software*.

Se cuenta con la asesoría del Director de Proyecto con dedicación de cuatro (4) horas semanales, para afinamiento de detalles y ajuste de empalme al entorno de producción.

Se calculó el salario del recurso humano, tomando como base el valor-hora-hombre (**\$9.375.00** pesos mcte) y se promedió, según el salario actual de los integrantes del equipo de trabajo que corresponde a **\$1.500.000**, para los tres (3) y que al ser multiplicado por la cantidad de horas invertidas en el proyecto, arroja un total de **\$281.250.00** pesos mcte al mes.

1.7.3. Estudio factibilidad financiero. En el proyecto de desarrollo e implementación de *App Android* para consulta de notas y horarios de estudiantes y horarios de docentes de la Fundación del CIDCA, inició el día 15 de Enero de 2015, ha conllevado una serie de inversiones de considerables de recursos económicos, sobre los que se debe realizar un estudio de factibilidad juicioso, para controlar costos actuales y futuros, a fin de que el sostenimiento del ciclo de vida del proyecto, corresponda a la solución esperada y no sea base de otra problemática para la institución.

1.7.4. Estudio de factibilidad técnica-Recursos de infraestructura. Los recursos de infraestructura con los que se cuenta son los siguientes:

- Cuatro (4) computadores
- Tres (3) equipos portátiles
- Un (1) equipo de escritorio (**PC**) que funcionará como servidor de pruebas

- El *software* requerido, el licenciado y el libre, que se requiere en el desarrollo e implementación del proyecto, se tomarán de una conexión a *internet* de 3 megas banda ancha y de una *ip* pública para realizar las conexiones de prueba requeridas.

Para el momento de la implementación, la Fundación CIDCA, deberá contratar, un servidor de aplicaciones para la instalación del *web service* necesario para enlazar el *App Android* con la base de datos y poder realizar la consulta.

El costo de los equipos se realizó con base al costo mínimo de alquiler en el mercado por mes, que corresponde a **\$50.000.00** pesos mcte, por mes, teniendo en cuenta que son propios y han sido puestos a disposición del proyecto. En cuanto al el costo de consumo de *Internet*, se trabaja por valor de las horas que corresponde a **\$40.000.00** pesos mcte, por mes.

Otros costos: A parte del desarrollo que se lleva a cabo entre semana de manera individual, el equipo de trabajo se reunirá los días sábados para realizar el acoplamiento del trabajo y la sincronización del mismo, es por eso que se tienen contemplados unos gastos de transporte, tres (3) veces por semana incluyendo el día sábado, correspondiente a valor por transporte de **\$20.000.00** pesos mcte, por mes.

La adquisición del derecho a suscripción y uso del *App a Store Google Play*, para subir la *App Android*, facilitando el acceso a la comunidad estudiantil de la Fundación CIDCA; corresponde a un valor de **U\$25.USD**.

Los gastos de papelería y otros, están alrededor de los **\$10.000.00** pesos mcte, por mes y para los primeros meses y en el mes final de **\$50.000.00** pesos mcte, por mes. Los mismos se sumaran por los cinco (5) meses de duración del

proyecto. El consolidado suma un total de **\$3.151.700,00** pesos detallada en el **cuadro 2.**, correspondiente a la delimitación financiera.

2. MARCOS DE REFERENCIA

2.1. ANTECEDENTES

En la actualidad existen varias aplicaciones similares al *App* propuesto en este proyecto de grado, para el manejo de la información académica que relaciona a los estudiantes y docentes en cuanto a notas y horarios, con su respectiva institución académica. Son elaboradas bajo el requerimiento de la institución y cuentan con soporte de los creadores durante todo el ciclo de vida del mismo. Sin embargo en la Fundación CIDCA, el acceso a dicha información solo se puede realizar si se tiene un *PC* a la mano, limitando al estudiante y al docente a que la consulta no sea inmediata.

2.2. MARCO INSTITUCIONAL

Nombre de la Institución: Fundación CIDCA

Ciudad: Bogotá D.C.

Área: Educación Superior

2.3. MARCO TEÓRICO

2.3.1. Ingeniería de *software*. (Sommerville, 2005, pág. 24), refiere:

Es la que contempla todos los aspectos del desarrollo y de la evolución de sistemas complejos, donde el *software* desempeña un papel principal. Por lo tanto, la ingeniería de sistemas comprende el desarrollo de *hardware*, políticas y procesos de diseño y distribución de sistemas, así como la ingeniería del *software*.

Los ingenieros de sistemas están involucrados en la especificación del sistema, en la definición de su arquitectura y en la integración de las diferentes partes para crear el sistema final.

La ingeniería de *software* es una disciplina de la ingeniería de sistemas que comprende todos los aspectos de la producción de *software*, desde

las etapas iniciales de la especificación del sistema, hasta el mantenimiento de éste después de que se utiliza.

Dentro de este concepto hay dos apartados importantes: el primero afirma que es una disciplina de la ingeniería, es decir que los ingenieros aplican teorías, métodos y herramientas donde sean convenientes, pero las utilizan de forma selectiva y siempre tratando de descubrir soluciones a los problemas, y segundo que todos los aspectos de producción de *software*, la ingeniería comprende disciplinas técnicas, para construir el producto, de gestión, para planificar, monitorear y controlar proyectos y de Soporte, para medir y asegurar la calidad del producto.

Muchas personas asocian el término *Software* con los programas de computadora, sin embargo, se deben tener en cuenta otros aspectos tales como los documentos asociados y la configuración de datos necesarios para que los programas operen de forma correcta. En este contexto, se puede afirmar que existen dos tipos de *software*.

Productos genéricos: Son sistemas aislados producidos por una organización de desarrollo y que se venden al mercado abierto a cualquier cliente. La especificación es controlada por la organización que desarrolla.

Productos personalizados (o hechos a medida): Son sistemas requeridos por un cliente en particular. La especificación de los productos personalizados, por lo general, es desarrollada y controlada por la organización que compra el *software*.

En cuanto al proceso de producción *software*, se puede decir que es un conjunto de actividades y resultados asociados que producen un producto de *software*. Para lo cual existen cuatro actividades fundamentales que son comunes para todos los procesos y son:

Especificación de *software*: los clientes e ingenieros definen el *software* a producir y las restricciones sobre su operación.

Desarrollo de *software*: el *software* se diseña y programa.

Validación de *Software*: el *software* se valida para asegurar que es lo que el cliente realmente quiere.

Evolución del *Software*: el *software* se modifica para adaptarlo a los cambios requeridos por el cliente y el mercado.

Además estas actividades genéricas pueden organizarse de diferentes formas y describirse en diferentes niveles de detalle para diferentes tipos de *software*. Sin embargo, el uso de un proceso inadecuado del *software* puede reducir la calidad o la utilidad del producto de *software* que se va a desarrollar y/o incrementar los costos de desarrollo.

En cuanto al desarrollo del *software* (Audisistem, 2012) refiere

Desarrollo de *software*. Desarrollar un *software* significa construirlo simplemente mediante su descripción. Está es una muy buena razón para considerar la actividad de desarrollo de *software* como una ingeniería. En un nivel más general, la relación existente entre un *software* y su entorno es clara ya que el *software* es introducido en el mundo de modo de provocar ciertos efectos en el mismo.

Aquellas partes del mundo que afectarán al *software* y que serán afectadas por él será el Dominio de Aplicación, es allí donde los usuarios o clientes observarán si el desarrollo del *software* ha cumplido su propósito.

Una de las mayores deficiencias en la práctica de construcción de *software* es la poca atención que se presta a la discusión del problema, en general los desarrolladores se centran en la solución dejando el problema inexplorado, el problema a resolver debe ser deducido a partir de su solución.

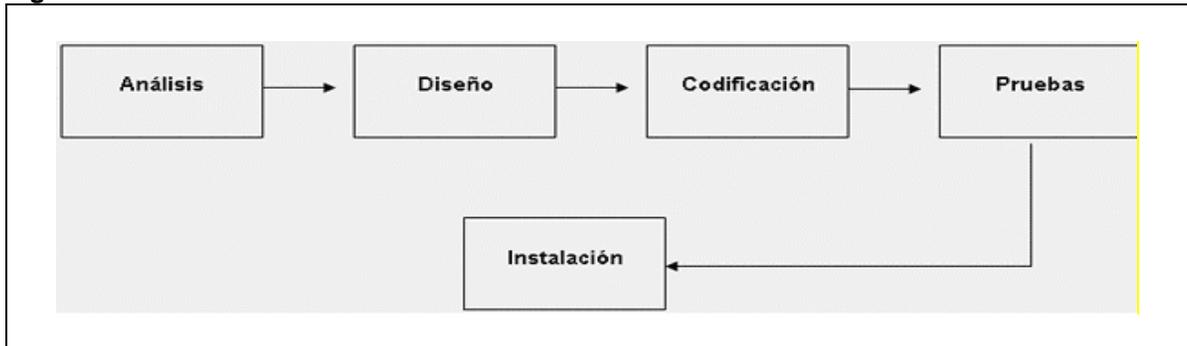
Esta aproximación orientada a la solución puede funcionar en campos donde todos los problemas son bien conocidos, clasificados e investigados, donde la innovación se ve en la detección de nuevas soluciones a viejos problemas, pero el desarrollo de *software* no es un campo con tales características, la versatilidad de las computadoras y su rápida evolución hace que exista un repertorio de problemas en constante cambio y cuya solución *software* sea de enorme importancia.

Cuando se va desarrollar un *software* intervienen muchas personas como lo es el cliente que es el que tiene el problema en su empresa y desea que sea solucionado, para esto existe el Analista de Sistema que es el encargado de hacerle llegar todos los requerimientos y necesidades que tiene el cliente a los programadores que son las personas encargadas de realizar lo que es la codificación y diseño del sistema para después probarlo y lo instalan al cliente.

Es así como intervienen varias personas ya que una sola persona no podría determinar todo lo necesario lo más seguro que le haga falta algún requerimiento o alguna parte del nuevo sistema y entre más estén involucradas mejor para cubrir con todos los requerimientos del sistema.

Ver figura 1.

Figura 1. Proceso desarrollo de *software*



Fuente: <http://www.monografias.com/> 1

2.3.2. **Android SO.** (Pedrozo, 2012), expresa que:

De un tiempo a esta parte los usuarios de móviles han ido requiriendo más opciones para sus terminales y han ido exprimiendo las que estos ofrecían. Las compañías fabricantes de teléfonos ofrecen día a día mejores terminales, llegando al punto de que estos son capaces de realizar muchas funciones destinadas, antiguamente, exclusivamente para *PC's*. *Google* no ha desaprovechado la oportunidad de realizar junto con su gran equipo de desarrolladores un sistema operativo moderno y a la altura de las circunstancias.

Se trata de un SO móvil basado en *Linux* orientado a teléfonos y tabletas.

Historia. Fue desarrollado por *Android inc.*, empresa comprada por *Google* en 2005. Inicialmente el sistema se desarrolló exclusivamente por *Google* pero más tarde pasó a formar parte de la *Open Handset Alliance*, la cual está liderada por *Google*. Una de las ventajas que contiene el SO es que permite que personas ajenas a *Google* desarrollen aplicaciones, como es mi propio caso, así consiguen que las aplicaciones basadas en el sistema sean mucho mayores que las de los sistemas cerrados, y esto repercute en la venta de terminales con *Android SO*. El primer teléfono que salió al mercado con este dispositivo fue el *HTC G1* o *Dream* lanzado originalmente en *EEUU* a finales de 2008, venía con la versión *Android 1.0* instalada y rápidamente se convirtió en un éxito de ventas, actualmente la versión 2.3 poco tiene que ver ya con el anticuado sistema 1.0.

Características. Posee las siguientes:

- Diseño de dispositivo, *Android* permite la utilización de casi cualquier resolución móvil disponible hasta día de hoy lo que hace el sistema adaptable desde pantallas de 2,5" hasta tabletas de 10" Además permite el uso de librerías de gráficos en 2D y 3D.

- Almacenamiento, se realiza mediante *SQLite* como ya he mencionado anteriormente.
- Conectividad Soporte para estas redes de conectividad inalámbrica: *GSM/EDGE, IDEN, CDMA, EV-DO, UMTS, Bluetooth, Wi-Fi, LTE, y WiMAX.*
- Mensajería, tanto las formas tradicionales de mensajería en Colombia como son *SMS* y *MMS* como los correos electrónicos con sistema *push* son bien recibidos en *Android*.
- Navegador web, basado en el motor *WebKit*, usado por ejemplo en *Chrome* o *Safari* y mediante el motor de *JavaScript V8* el navegador nativo de *Android* es uno de los mejores a nivel de *SO* móviles.
- Soporte de Java, realmente *Android* no dispone de una máquina virtual de java como tal, debido a que los dispositivos móviles son de escasos recursos así que se apoya en la máquina virtual de *Dalvik* para ejecutar el código *Java*.
- Soporte multimedia, *Android* soporta la mayoría de los formatos
- Multimedia más relevantes de hoy en día, en concreto: *WebM, H.263, H.264 (en 3GP o MP4), MPEG-4 SP, AMR, AMR-WB (en un contenedor 3GP), AAC, HE-AAC (en contenedores MP4 o 3GP), MP3, MIDI, Ogg Vorbis, WAV, JPEG, PNG, GIF, BMP.61*
- Soporte para *streaming*, El soporte para *streaming* de descarga en *HTML* se hace de forma nativa, pero el soporte para adobe *flash* se hace mediante una aplicación que proporciona *Adobe*, aunque lo cierto es que son los dispositivos actuales los que está empezando a ejecutar aplicaciones o visualizaciones en *Adobe Flash*, los terminales con una año de antigüedad no lo soportan.
- Soporte para *hardware* adicional, el sistema soporta una infinidad de *software* externo al funcionamiento tradicional de un dispositivo que sirve para hacer llamadas, entre otros encontramos: soporte para cámaras de fotos, de vídeo, pantallas táctiles, *GPS*, acelerómetros, magnetómetros, giroscopios, aceleración 2d y 3d, sensores de proximidad y de presión, termómetro.
- Entorno de desarrollo, como ya he explicado anteriormente tiene un gran entorno de desarrollo basado en *Eclipse* y en el emulador suministrado por *Google*.
- *Market*, Dispone de un mercado de aplicaciones (*Android Market*) permitiendo de este modo descargar e instalar aplicaciones sin

necesidad de un *PC*. Actualmente este mercado sobrepasa las 100.000 aplicaciones y sigue aumentando rápidamente.

- Multi-táctil, se incluyen las funciones multitáctiles de forma nativa para todos los dispositivos *Android* que dispongan de una pantalla con esta capacidad.
- Bluetooth, la versión 2.2 ya da soporte para *A2DP* y *AVRCP*, envío de archivos (*OPP*), exploración del directorio telefónico, marcado por voz y envío de contactos entre teléfonos.
- Video llamada, a partir de la *versión 2.3* lo hace de manera nativa, los dispositivos que ofrecen una cámara frontal como el *Samsung Galaxy S* y esta versión del sistema operativo ya pueden realizarlas sobre *IP*.
- Multitarea, para todas las aplicaciones y desde su primera versión.
- *Tethering*, permite al teléfono ser usado como punto de acceso *WIFI*, está disponible a partir de la versión 2.2

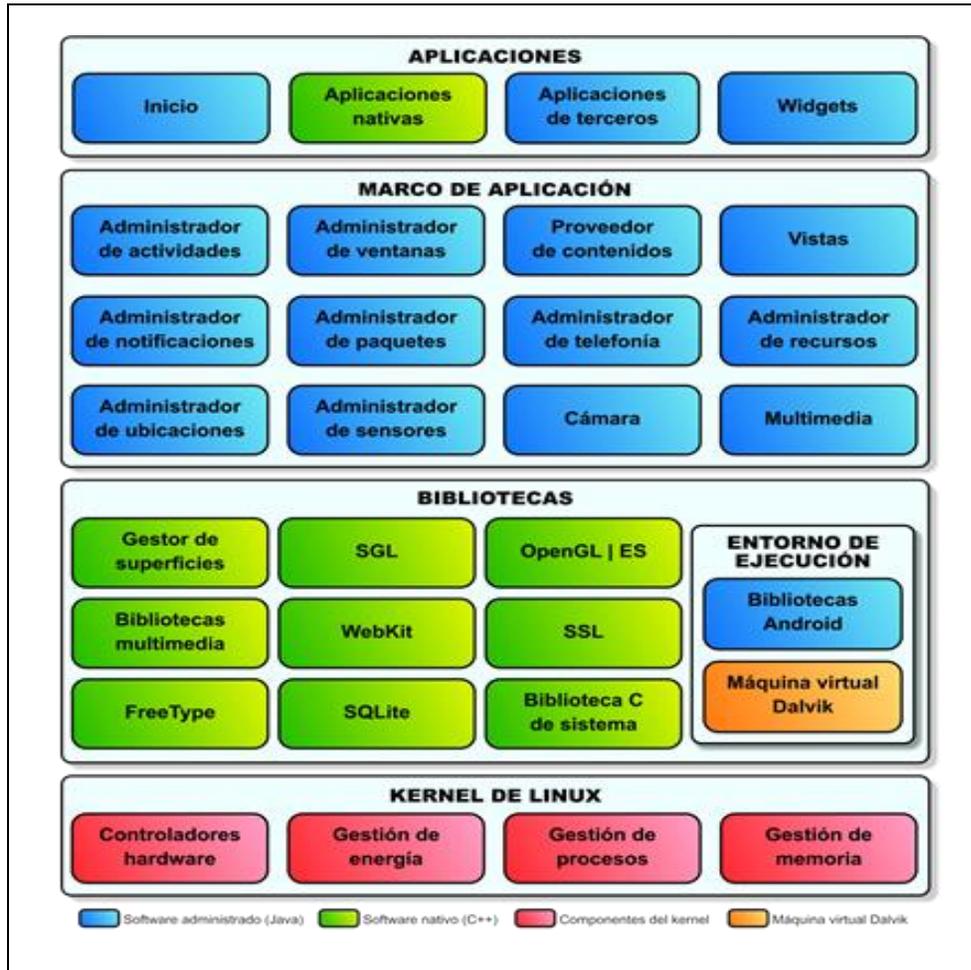
Arquitectura del sistema. Donde se encuentran todas aplicaciones instaladas en el sistema, algunas de las aplicaciones base son: un cliente de correo electrónico, programa de *SMS*, calendario, mapas, navegador, contactos. Como ya he mencionado están escritas completamente en *Java*.

Marco de trabajo de aplicaciones, o *Framework* de aplicaciones o armazón de aplicaciones, cualquier desarrollador puede acceder a ellos para la realización de sus aplicaciones, está diseñado para simplificar la reutilización de los componentes del sistema.

Bibliotecas: se incluyen una gran variedad de bibliotecas de *C/C++* usadas por ciertas partes sistema. Los desarrolladores pueden hacer uso de ellas, las más importantes son: *System C library*, bibliotecas de medios, bibliotecas de gráficos, *3D*, *SQLite* entre otras.

Runtime de Android. Este incluye una recopilación de bibliotecas base que proporcionan casi la totalidad de las funciones disponibles en las bibliotecas de *Java*. Cada una de las aplicaciones que se estén ejecutando utiliza su propio proceso con una llamada a la máquina virtual *Dalvik*. El diseño de ésta máquina permite que se ejecuten concurrentemente varias de ellas sin que haya ningún problema en el dispositivo. **Ver figura 2.**

Figura 2. Estructura del sistema



Fuente: Ángel. J. Vico 1 Arquitectura de *Android*

2.4. MARCO METODOLÓGICO

En la metodología de desarrollo y estándares utilizados en el mismo se proporcionan las guías para poder conocer todo el camino a recorrer desde antes de empezar la implementación, con lo cual se asegura la calidad del producto final, así como también el cumplimiento en la entrega del mismo en un tiempo estipulado.

2.4.1. Metodología X.P. La programación extrema (*X.P.*), es una metodología moderna con un alto nivel de aceptación (tiene alrededor de diez (10) años en la

industria del *software*). La filosofía de *X.P.*, está basada en satisfacer completamente las necesidades del cliente, por eso lo integra como una parte esencial del equipo de desarrollo de programación.

La metodología *X.P.*, fue inicialmente creada para el desarrollo de aplicaciones dónde el cliente no sabe muy bien lo que quiere, lo que provoca un cambio constante en los requisitos que debe cumplir el desarrollo de una aplicación. Siendo las características más relevantes: fácil adaptación a las necesidades del cliente y a la reevaluación constante en periodos de tiempo cortos.

La metodología *X.P.*, está diseñada para el desarrollo de aplicaciones que requieran de un grupo de programadores pequeño, dónde la comunicación sea más factible que en grupos de desarrollo grandes. La comunicación es un punto importante y debe realizarse entre los programadores, los jefes de proyecto y los clientes.

Según (*Wells*, 2001) en el capítulo V.¹ refiere en cuanto a las etapas de la metodología anterior, lo siguiente.

Fases de desarrollo

- **Planificación:** el cliente que para el caso en estudio del presente trabajo de grado es Institución, realizó un resumen de los problemas evidenciados que justifican la urgente necesidad de implementar un sistema de información, siendo relevante la demora que ocasiona, el no poder saber en tiempo real, las notas y horarios de clase, lo que genera incertidumbre de los estudiantes (antiguos y nuevos), manifestándose en pérdida de tiempo significativo para los estudiantes.

Viene entonces la planificación a través de un cronograma de actividades realizado por los autores y representante de la institución, acordando en el mismo, fechas, tiempos, fases de trabajo, entrega de

¹ Ciclo de vida de un proyecto. Metodología *X.P.*

avances, todo proyectado hacia el logro de los objetivos establecidos con la institución para solucionar la problemática.

- **Diseño:** Se creó una *App* con las operaciones básicas (conexión a base de datos, consultas de horarios y notas).

Con ayuda de otras áreas de la institución se tomaron decisiones de diseño y aspectos de la *App*.

El código fue diseñado, limpio, fácil de entender y con facilidad de modificar.

- **Desarrollo:** Se presentaron avances tanto de código como de interfaz gráfica para que el cliente tomara decisiones del cuerpo y de la forma en que interactuaría el *App* con las personas que la iban a utilizar.

Se programó en parejas a medida que todo iba quedando claro se programaba y se realizaba diseño para luego unificar.

Se realizaron pruebas en las cuales se realizaban simulaciones de consulta teniendo en cuenta los perfiles

2.4.2. Java. (Telocapa, 2013), refiere que:

Es un lenguaje de programación de ordenadores, diseñado como una mejora de C++, y desarrollado por *Sun Microsystems* (compañía actualmente absorbida por *Oracle*). Hay varias hipótesis sobre su origen, aunque la más difundida dice que se creó para ser utilizado en la programación de pequeños dispositivos, como aparatos electrodomésticos (desde microondas hasta televisores interactivos). Se pretendía crear un lenguaje con algunas de las características básicas de C++, pero que necesitara menos recursos y que fuera menos propenso a errores de programación.

De ahí evolucionó (hay quien dice que porque el proyecto inicial no acabó de funcionar) hasta convertirse en un lenguaje muy aplicable a Internet y programación de sistemas distribuidos en general.

Pero su campo de aplicación no es exclusivamente Internet: una de las grandes ventajas de *Java* es que se procura que sea totalmente independiente del *hardware*: existe una "máquina virtual *Java*" para varios tipos de ordenadores.

Un programa en *Java* podrá funcionar en cualquier ordenador para el que exista dicha "máquina virtual *Java*" (hoy en día es el caso de los ordenadores equipados con los sistemas operativos *Windows*, *Mac OS*

X, *Linux*, y algún otro; incluso muchos teléfonos móviles actuales son capaces de usar programas creados en *Java*), y aún hay más: el sistema operativo *Android* para teléfonos móviles propone usar *Java* como lenguaje estándar para crear aplicaciones. Como inconveniente, la existencia de ese paso intermedio hace que los programas *Java* no sean tan rápidos como puede ser un programa realizado en *C*, *C++* o *Pascal* y optimizado para una cierta máquina en concreto.

- ¿Por qué usar *Java*? Puede interesarnos si queremos crear programas que se vayan a manejar a través de un *interfaz web* (sea en Internet o en una Intranet de una organización), programas distribuidos en general, o programas que tengan que funcionar en distintos sistemas sin ningún cambio (programas "portables"), o programas para un *Smartphone Android*, entre otros casos.
- ¿Cuándo no usar *Java*? Como debe existir un paso intermedio (la "máquina virtual") para usar un programa en *Java*, no podremos usar *Java* si queremos desarrollar programas para un sistema concreto, para el que no exista esa máquina virtual. Y si necesitamos que la velocidad sea la máxima posible, quizá no sea admisible lo (poco) que ralentiza ese paso intermedio.
- Qué más aporta *Java*? Tiene varias características que pueden sonar interesantes a quien ya es programador, y que ya irá conociendo poco a poco quien no lo sea. La sintaxis del lenguaje es muy parecida a la de *C++* (y a la de *C*).
- Al igual que *C++*, es un lenguaje orientado a objetos, con las ventajas que eso puede suponer a la hora de diseñar y mantener programas de gran tamaño.
- *Java* permite crear programas multitarea.
- Permite excepciones, como alternativa más sencilla para manejar errores, como ficheros inexistentes o situaciones inesperadas.
- Es más difícil cometer errores de programación que en *C* y *C++* (no existen los punteros).
- Se pueden crear entornos "basados en ventanas", gráficos, acceder a bases de datos, etc.
- Instalación de la máquina virtual *Java*. Este *software* va a permitir ejecutar código *Java* en el equipo. A la máquina virtual *Java* también se la conoce como entorno de ejecución *Java*, *Java Runtime Environment (JRE)* o *Java Virtual Machine (JVM)*.

2.4.3. Eclipse. (Retamar, 2012), el entorno de desarrollo Eclipse se puede definir de esta manera:

Incluyendo sus *plugins*, está desarrollado por completo en el lenguaje Java. Un problema habitual en herramientas Java (como *NetBeans*) es que son demasiado “pesadas”. Es decir, necesitan una máquina muy potente para poder ejecutarse de forma satisfactoria. En gran medida, estas necesidades vienen determinadas por el uso del *API Swing* para su interfaz gráfico.

Swing es una librería de *widgets* portable a cualquier plataforma que disponga de una máquina virtual *Java* pero a costa de no aprovechar las capacidades nativas del sistema donde se ejecuta, lo cual supone una ejecución sensiblemente más lenta que la de las aplicaciones nativas.

SWT es una librería de *widgets* equivalente a *Swing* en la cual, se aprovechan los *widgets* nativos del sistema sobre el que se ejecuta. El hecho de aprovechar los *widgets* activos, permite que la ejecución de interfaces de usuario sea mucho más rápida y fluida que si se utilizase *Swing* y, además, siempre dispone del “*Look and Feel*” del sistema, sin necesidad de “emularlo”.

Eclipse se define como “*An IDE for everything and nothing in particular*” (un *IDE* para todo y para nada en particular). Eclipse es, en el fondo, únicamente un armazón (*workbench*) sobre el que se pueden montar herramientas de desarrollo para cualquier lenguaje, mediante la implementación de los *plugins* adecuados. La arquitectura de *plugins* de Eclipse permite, además de integrar diversos lenguajes sobre un mismo *IDE*, introducir otras aplicaciones accesorias que pueden resultar útiles durante el proceso de desarrollo como: herramientas *UML*, editores visuales de interfaces, ayuda en línea para librerías, etc.

Desde *Eclipse IDE for Java EE Developers* el programador podrá ir diseñando y trabajando con cada módulo de forma independiente. Para desarrollar cada uno de los aspectos del diseño dispone de un completísimo *pack* de funciones, utilidades y herramientas que le facilitarán todo el trabajo.

Además, *Eclipse IDE for Java EE Developers* viene integrado con su propio depurador (debugging), así como con una interfaz donde chequear y realizar tests una vez compilado el proyecto en *Java*.

Algunos de los elementos con los que viene equipado *Eclipse IDE for Java EE Developers* son: editor de texto, corrector y resaltador de sintaxis de los comandos, compilador para hacer la compilación en tiempo real, herramientas para la creación de tus propias clases, etc.

2.4.4. NetBeans. (Eslava, 2013, pág. 77), lo define como:

Es un IDE muy completo para programar en varios lenguajes, aunque se especializa principalmente en *Java*. *NetBeans IDE* es un entorno de desarrollo, una herramienta para que los programadores puedan escribir, compilar, depurar y ejecutar programas. Está escrito en *Java* - pero puede servir para cualquier otro lenguaje de programación. Existe además un número importante de módulos para extender el *NetBeans IDE*. *NetBeans IDE* es un producto libre y gratuito sin restricciones de uso.

También está disponible *NetBeans Platform*; una base modular y extensible usada como estructura de integración para crear grandes aplicaciones de escritorio. Empresas independientes asociadas, especializadas en desarrollo de *software*, proporcionan extensiones adicionales que se integran fácilmente en la plataforma y que pueden también utilizarse para desarrollar sus propias herramientas y soluciones.

Ambos productos son de código abierto y gratuito para uso tanto comercial como no comercial. El código fuente está disponible para su reutilización de acuerdo con la *Common Development and Distribución Licencia (CDDL) v1.0 and the GNU General Public License (GPL) v2*.

2.4.5. Web service. (Eslava, 2013, pág. 89), *refiere que un servicio web* (en inglés, Web service o web service), es:

Una tecnología que utiliza un conjunto de protocolos y estándares que sirven para intercambiar datos entre aplicaciones. Distintas aplicaciones de *software* desarrolladas en lenguajes de programación diferentes, y ejecutadas sobre cualquier plataforma, pueden utilizar los servicios web para intercambiar datos en redes de ordenadores como Internet. La interoperabilidad se consigue mediante la adopción de estándares abiertos.

Las organizaciones *OASIS* y *W3C* son los comités responsables de la arquitectura y reglamentación de los servicios *web*. Para mejorar la interoperabilidad entre distintas implementaciones de servicios Web se ha creado el organismo *WS-I*, encargado de desarrollar diversos perfiles para definir de manera más exhaustiva estos estándares. Es una máquina que atiende las peticiones de los clientes web y les envía los recursos solicitados.

2.4.6. Oracle bases de datos. *Apache* (Fundation *Apache*, 1999), documenta esta base de datos de la siguiente manera:

Oracle la Primera Base de Datos Diseñada para *Grid Computing*, es un sistema de gestión de base de datos relacional fabricado por Oracle Corporation. Oracle es básicamente un herramienta cliente/servidor para la gestión de base de datos la gran potencia que tiene y su elevado precio hace que solo se vea en empresas muy grandes y multinacionales, por norma general.

Oracle Corporation: es una de las mayores compañías de *software* del mundo sus productos van desde bases de datos (*Oracle*) hasta sistemas de gestión. Cuenta además, con herramientas propias de desarrollo para realizar potentes aplicaciones, como *Oracle Designer*.

Características:

- Oracle *Database* ha sido diseñada para que las organizaciones puedan controlar y gestionar grandes volúmenes de contenidos no estructurados en un único repositorio con el objetivo de reducir los costes y los riesgos asociados a la pérdida de información.
- Una *BD Oracle* tiene una estructura física y una estructura lógica:
- La estructura física se corresponde a los ficheros del sistema operativo.
- La estructura lógica está formada por los tablespace y los objetos de un esquema de BD

2.4.7. Apache tomcat. (Fundation *Apache*, 1999), refiere de *Apache tomcat* (*Jakarta Tomcat* o simplemente *Tomcat*) lo siguiente:

Funciona como un contenedor de servlets desarrollado bajo el proyecto *Jakarta en la Apache Software. Foundation*. *Tomcat* implementa las especificaciones de los *servlets* y de *JavaServer Pages (JSP)* de *Oracle Corporation* (aunque creado por *Sun Microsystems*).

Tomcat es desarrollado y actualizado por miembros de la *Apache Software Foundation* y voluntarios independientes. Los usuarios disponen de libre acceso a su código fuente y a su forma binaria en los términos establecidos en la *Apache Software License*. Las primeras distribuciones de *Tomcat* fueron las versiones 3.0.x. Las versiones más recientes son las 8.x, que implementan las especificaciones de *Servlet 3.0* y de *JSP 2.2*. A partir de la versión 4.0, *Jakarta Tomcat* utiliza el contenedor de *servlets Catalina*.

Dado que *Tomcat* fue escrito en *Java*, funciona en cualquier sistema operativo que disponga de la máquina virtual *Java*.

Tomcat empezó siendo una implementación de la especificación de los servlets comenzada por *James Duncan Davidson*, que trabajaba como arquitecto de *software* en *Sun Microsystems* y que posteriormente ayudó a hacer al proyecto de código abierto y en su donación a la *Apache Software Foundation*.

2.4.8. Lenguaje de Modelamiento Unificado (UML). El Lenguaje de Modelamiento Unificado (*UML-Unified Modeling Language*) es un lenguaje gráfico para visualizar, especificar y documentar cada una de las partes que comprende el desarrollo de *software*. *UML* entrega una forma de modelar cosas conceptuales como lo son procesos de negocio y funciones de sistema, además de cosas concretas como lo son escribir clases en un lenguaje determinado, esquemas de base de datos y componentes de *software* reusables.

Modelo de Clases, un diagrama de clases sirve para visualizar las relaciones entre las clases que involucran el sistema, las cuales pueden ser asociativas, de herencia, de uso y de contenimiento.

2.5. MARCO LEGAL

El (Ministerio de Educación, 2014.), define el *software* como:

Un conjunto de herramientas creadas por una persona o un grupo con el fin de resolver necesidades puntuales. Estos desarrollos pueden ir desde una aplicación sencilla compuesta por un número pequeño de instrucciones hasta la conformación de un complejo sistema de información.

Gracias al *software*, el *Hardware*, es decir, la máquina, realiza sus tareas. Así como los libros y las obras de teatro tienen derechos de autor, el *software*, por ser una creación generada desde la inteligencia individual, debe tener un permiso explícito para su uso y aprovechamiento.

Por lo tanto, si el *software* no tiene un permiso que autorice su uso libre, requiere de una licencia, normalmente representada por un documento que dice frases como "se autoriza el uso del *software* xx a nombre de una empresa o persona natural XXX". Un ejemplo de este tipo de *software* son los sistemas operativos que corren en nuestros computadores, las herramientas ofimáticas, etc.

Colombia ha adoptado un rol protagónico en la defensa de los derechos de autor y la propiedad intelectual, desarrollando un conjunto de normas que regulan, protegen y penalizan a aquellas personas que violen estos derechos.

La Ley 44 de 1993 especifica penas entre dos y cinco años de cárcel, así como el pago de indemnizaciones por daños y perjuicios a quienes comentan el delito de piratería de *software*. Se considera delito el uso o reproducción de un programa de computador de manera diferente a como está estipulado en la licencia. Los programas que no tengan licencia son ilegales y es necesaria una licencia por cada copia instalada en los computadores.

A partir del mes de julio de 2001, y gracias a la reforma hecha al Código de procedimiento penal, quien sea encontrado usando, distribuyendo o copiando *software* sin licencia tendrá que pagar con cárcel hasta por un período de 5 años.

Sin embargo, uno de los logros más importantes de la legislación colombiana en materia de protección de derechos de autor fue la Ley 603 de 2000, en la cual todas las empresas deben reportar en sus Informes Anuales de Gestión el cumplimiento de las normas de propiedad intelectual y derechos de autor. La Dirección de Impuestos y Aduanas Nacionales (DIAN) quedó encargada de supervisar el cumplimiento de estas leyes, mientras que las Superintendencias quedaron responsables de vigilar y controlar a estas empresas.

3. METODOLOGÍA

3.1. METODOLOGÍA XP

La metodología *XP* está diseñada para el desarrollo de aplicaciones que requieran de un grupo de programadores pequeño, donde la comunicación sea más factible que en grupos de desarrollo grandes. La comunicación es un punto importante y debe realizarse entre los programadores, los jefes de proyecto y los clientes.

Igualmente la metodología *XP*, fue inicialmente creada para el desarrollo de aplicaciones donde el cliente partiendo de un requerimiento inicial, sobre la marcha va manifestando otros requerimientos a suplir, lo que provoca un cambio constante en los mismos que debe cumplir el desarrollo de una aplicación. Siendo las características más relevantes: fácil adaptación a las necesidades del cliente y a la reevaluación constante en periodos de tiempo cortos.

Por lo anterior se aplicó en el diseño y desarrollo de este proyecto este tipo de metodología para el desarrollo de *software*.

3.2. FASES METODOLÓGICAS PARA “*MiCIDCA*”

La aplicación de la metodología *XP*, desglosa en su diseño y desarrollo, seis (6) fases metodológicas de trabajo así:

3.2.1. Fase I. Planificación y levantamiento de información. En esta fase se planificó la ruta de trabajo y se escogieron las herramientas metodológicas tales como:

- a. **“Check list”**: se fueron enumerando aspectos sobre la problemática a solucionar y los requisitos que debería presentar el aplicativo para responder a las necesidades de consulta de notas y horarios por los estudiantes y de horarios por los docentes.
- b. **Entrevistas informales**: se realizaron para determinar con precisión los tres (3) aspectos mencionados anteriormente.
- c. **Bases teóricas**: se construyó un marco de referencia conformado por los antecedentes y los marcos teórico-conceptual, institucional, metodológico y legal con los que se sustentó la temática del presente trabajo de grado.

3.2.2. Fase II. Análisis y diseño de “MiCIDCA”. Con el resultado del levantamiento de información de la fase anterior, se determina el modelado de la aplicación informática propuesta, donde los componentes se desglosan en: diagramas, módulos, casos de uso que conformaron el primer diseño del aplicativo a presentar a la institución para la respectiva aprobación.

3.2.3. Fase III. Desarrollo de “MiCIDCA”. En el desarrollo del aplicativo propuesto, se realizaron los siguientes pasos:

- a. Se escogió lenguaje para la programación.
- b. Se buscó un modo de conexión a la Base de Datos
- c. Se escogió un sistema operativo para dispositivos móviles

3.2.4. Fase IV. Pruebas y ajustes de “MiCIDCA”. Se monta el prototipo del *App*, para someterlo a pruebas que fueron diseñadas para corregir cualquier error al ejecutar el aplicativo.

3.2.5. Fase V. Montaje, presentación y entrega final de “MiCIDCA”. Se procede a montar el *App* para el desarrollo del aplicativo y se presenta a los directivos la Fundación CIDCA.

3.2.6. Fase VI. Elaboración de manual de usuario para “MiCIDCA”. Se elaboró el manual de usuario para manejo del aplicativo, en procura de la correcta operación de “*MiCIDCA*”.

3.3. ANÁLISIS Y REQUERIMIENTOS

Son los requerimientos mínimos para el desarrollo e implementación del aplicativo y se dividen en no funcionales y funcionales.

3.3.1. Requerimientos no funcionales. Contempla los siguientes:

a. Software y Hardware. Ver Cuadro 5.

Cuadro 3. Requerimientos No funcionales: software y Hardware

Tipo	Dispositivos	Especificaciones
<i>Hardware</i>	Procesador	Core5/Core7
	Disco Duro	500Gb/1Tb
	Memoria RAM	4G
<i>Software</i>	Sistema Operativo	Windows 7-8/Android
	Lenguaje de Programación	Java

Fuente: Elaboración propia

b. Apariencia: el *software* deberá tener una interfaz de usuario con ambientes amigables ya que la facilidad con la que el usuario lo manipule mejorara los tiempos.

c. Operatividad. El *software* será desarrollado para *Android* con el lenguaje de programación *Java*

Los requerimientos de rapidez, fiabilidad y portabilidad los aportarán los equipos desde los que consulten los estudiantes y docentes, siempre y cuando hayan descargado la aplicación “*MiCIDCA*” de *Store Google Play*

d. Mantenimiento: las actualizaciones de versiones será realizado por el mismo equipo de desarrollo.

3.3.2. Requerimientos funcionales. Las actividades a realizar por este *App* se harán a petición de un usuario o docente (rol Estudiante, rol Docente) y tendrá que tener un comportamiento específico de acuerdo a su rol, para que el usuario correspondiente pueda realizar los procesos sin ningún problema.

a. Requerimientos funcionales de desempeño. El *App* debe estar en capacidad de dar respuesta al acceso de todos los usuarios con tiempo de respuesta aceptable y uniforme, en períodos de alta, media y baja demanda de uso del aplicativo.

El *App* debe estar en capacidad de dar respuesta al acceso de todos los usuarios con tiempo de respuesta aceptable y uniforme, en períodos de alta, media y baja demanda de uso del mismo.

b. Requerimientos funcionales de disponibilidad. Debe estar disponible todo el tiempo para que en cualquier momento los usuarios puedan acceder a él sin complicaciones.

c. Requerimientos funcionales de escalabilidad. El *App* debe estar en capacidad de permitir en el futuro el desarrollo de nuevas funcionalidades, modificar o eliminar funcionalidades después de su construcción y puesta en marcha inicial.

d. Requerimientos funcionales de facilidad de uso e ingreso de información. La aplicación informática debe ser de fácil uso, con una interfaz que permita al usuario poder indagar y navegar por todo el *App* con facilidad.

La aplicación debe presentar mensajes de error que permitan al usuario identificar el tipo de error y comunicarlo a los desarrolladores.

e. Requerimientos funcionales de flexibilidad. La aplicación debe ser diseñada y construida con la mayor flexibilidad.

f. Requerimientos funcionales de mantenimiento. El *App* debe tener su respectivo manual de usuario.

El *App* debe estar en capacidad de permitir en el futuro su fácil mantenimiento con respecto a los posibles errores que se puedan presentar durante la operación del sistema.

3.4. DIAGNÓSTICO PARA DESARROLLO DE “MiCIDCA” EJEMPLO

Es un *App* que se desarrolló según los requerimientos de la Fundación CIDCA.

a. Riesgos técnicos. El principal riesgo de *App* es que pierda conectividad por falta de disponibilidad de internet.

b. Riesgos del negocio. El *App* está siendo desarrollado a la medida de los requerimientos de la Fundación CIDCA y perfeccionada solo para la institución, así que en el momento de que sea requerido por otra institución, se tendrá que cambiar algunos aspectos, ajustando según necesidades y requerimientos.

✓ **Riesgos conocidos.** Falta de disponibilidad en las fechas de entrega.

✓ **Riesgos Técnicos ajenos a la aplicación.** Fallas de *en la conectividad o en los Smartphone.*

3.5. DISEÑO DE “MiCIDCA”

Para el modelaje del nuevo diseño se establecieron las siguientes condiciones enumeradas a continuación:

- Script
- Diagrama de clases.
- Diagramas de secuencia.
- Diagramas de actividades.
- Casos de uso.

3.6. ANÁLISIS DEL SISTEMA

Son los requerimientos los que muestran los elementos y las funciones que son necesarios para el desarrollo de un proyecto de *software*. Una de las metas de la ingeniería enfocada a los requerimientos es entregar una especificación de los requerimientos de *software* de manera correcta y por supuesto completa.

Aplicando los conocimientos de desarrollo adquirido durante la carrera, se busca realizar el desarrollo de un *App Android* que facilite la consulta de notas y horarios de la comunidad estudiantil CIDCA. Gracias a estos conocimientos se logró plantear una solución al inconveniente que se presentaba al requerir consultar las notas en curso.

Se realizó entre un grupo de estudiantes una entrevista rápida, de que tan bueno sería tener una aplicación en el teléfono para dicho propósito, y como sería la forma en que les gustaría visualizarla y la forma de acceder a la información.

3.7. ANÁLISIS DE REQUERIMIENTOS

El análisis de requisitos es una de las tareas más importantes en el ciclo de vida del desarrollo de *software*, puesto que en ella se determinan los “planos” de la nueva aplicación. En cualquier proyecto *software* los requisitos son las necesidades del producto que se debe desarrollar. Por ello, en la fase de análisis de requisitos se deben identificar claramente estas necesidades y documentarlas. Como resultado de esta fase se debe producir un documento de especificación de requisitos en el que se describa lo que el futuro sistema debe hacer. Por tanto, no se trata simplemente de una actividad de análisis, sino también de síntesis.

Introducción. En esta sección se proporcionará una introducción a todo el documento de Especificación de Requisitos del *App* (ERS). Consta de varias subsecciones: propósito, ámbito del sistema, definiciones, referencias y visión general del documento.

Propósito. El propósito es definir cuáles son los requerimientos que se deben tener en cuenta para el desarrollo, e implementación del *Software* para la consulta de horarios y notas de la comunidad estudiantil. Estos requerimientos permitirán dar una solución óptima y completa al cliente para suplir dichas necesidades de registro y manejo de la misma y poder crear nuevas estrategias.

Ámbito. *App Android* para la consulta de horarios y notas por parte de los estudiantes y de horarios por los docentes Fundación CIDCA para que la aplicación recibió el nombre de “**MICIDCA**”.

• **¿Qué hará el sistema?** El *App* móvil se desarrollarla inicialmente para la consulta de notas y horarios de los estudiantes y horarios de docentes. desarrollado en Programación Orientada a Objetos (POO) con un lenguaje *Java Eclipse*, se ejecutará en una plataforma *Android* en cualquiera de sus versiones.

Conectado el *App* se comunicará mediante internet con un *web service*, que realiza la petición y posterior conexión con la base de datos de la Fundación CIDCA, luego realizará la consulta y mostrará en pantalla los datos en tiempo real de notas y horarios.

- **Objetivos, beneficios y metas.** Los beneficios que ofrece el *App*, son los que incluyen las características que de la información se requiere como: disponibilidad, al tener acceso cuando se requiere en forma inmediata, esto teniendo en cuenta que el **99%** de la comunidad estudiantil cuenta con un *Smartphone*.

Al tener en tiempo real la información, los estudiantes y docentes, cuentan con una herramienta muy útil, ya que desde el comienzo de cada semestre podrán consultar horarios, nombre de asignatura, ubicación del aula y donde recibirán la catedra o será impartida. En el transcurso del semestre podrán consultar notas en línea y estar atentos a posibles errores humanos en la carga de información de las asignaturas, y de una forma más oportuna poder realizar el respectivo ajuste. La principal meta del aplicativo, es lograr la aceptación aproximadamente del 95% de la comunidad estudiantil de la fundación CIDCA.

Definiciones, acrónimos y abreviaturas

- **ERS:** Especificación de Requisitos de *Software*.
- **App:** Aplicación o *software*.
- **POO:** Programación Orientada a Objetos.
- **JAVA:** Lenguaje de Programación de *Software*.
- **ECLIPSE:** Entorno de desarrollo **JAVA** para aplicaciones móviles.
- **ANDROID:** Sistema operativo para móviles de *google*.
- **SMARTPHONE:** Teléfono inteligente.

Referencias Libro *Java*. Administración de una página Infografía ANSI/IEEE Std. 830-1984 Guía del IEEE para la Especificación de Requerimientos *Software*.

Descripción general del *App Android MiCIDCA*

Perspectiva del *software*. El *App* debe generar un nivel de satisfacción muy alto dentro de la comunidad estudiantil de CIDCA, que se manifestara gracias a la rapidez y disponibilidad con la que logren tener acceso a su información.

Funciones del *software*

- **Autenticación:** El *App* deberá contar con un enlace para autenticación de estudiantes y docentes que se conectara con la base de datos de CIDCA para su verificación y posterior acceso.
- **Consulta de horarios docentes:** El *App* se conecta con la base de datos de CIDCA y teniendo en cuenta la autenticación previa dará el acceso para visualizar los horarios asignados, el aula y el nombre de la asignatura a impartir.
- **Consulta de notas estudiantes:** El *App* se conecta con la base de datos de CIDCA y teniendo en cuenta la autenticación previa dará el acceso para visualizar las notas subidas al sistema corte a corte.
- **Consulta de horarios estudiantes:** El *App* se conecta con la base de datos de CIDCA y teniendo en cuenta la autenticación previa dará el acceso para visualizar los horarios asignados, el aula y el nombre de la asignatura a cursar por los estudiantes.

Características de los usuarios. Este producto está dirigido a los estudiantes y docentes de la institución, que en su gran mayoría ya están familiarizados con las

nuevas tecnologías y aplicaciones, adaptándose al cambio y a las facilidades para el manejo del *software* desde teléfonos inteligentes y que como usuarios de dispositivos móviles, sienten la necesidad de estar siempre informados acerca de los servicios que provee la aplicación.

Restricciones generales. Se determinaron las siguientes:

- Acceso limitado al entorno de producción.
- Acceso específico a tablas en las bases de datos.
- Desarrollo limitado a sistemas operativos *Android*
- Desarrollo limitado a *Java*

Suposiciones y dependencias. Se contemplan las siguientes.

- Dependencia de que el sistema operativo *Android* continúe con su estructura básica, y no afecte futuras instalaciones.
- Posibles actualizaciones del *App* por cambio de versión de *Android*.
- Modificación del servidor de aplicaciones de la Fundación CIDCA que puedan afectar la configuración del *web service*.
- Cambio de estructura de la base de datos de la Fundación CIDCA por requerimientos internos.
- Cambios del sistema de autenticación de la comunidad estudiantil por requerimientos internos.

Requisitos futuros. Probablemente y dependiendo del éxito de la aplicación, se generara una demanda de servicios complementarios por parte de los estudiantes y docentes, que estará enfocada, no solo hacia la consulta sino a la modificación de la información en el módulo de docentes y estudiantes del sistema de información de la universidad.

- **Docentes:** poder subir las notas de los estudiantes desde el aplicativo móvil, hacer inserción, modificación y eliminación de datos.
- **Estudiantes:** poder realizar la consulta global de su plan de estudios, realizar la inscripción de sus materias a cursar en el semestre, consultar y/o modificar su información general.

Requisitos específicos. Contempla lo siguiente:

- El *App* deberá permitir el fácil y seguro acceso a la consulta de notas y horarios.
- El sistema debe realizar autenticación de usuarios con la base de datos de docentes y estudiantes de CIDCA.
- El *App* debe realizar la conexión a la base de datos en CIDCA por medio del *Web service* desarrollado en *Java*.
- El *App* debe ser compatible y realizar conexión a una Base de datos en Oracle, actual BD que contiene el sistema de información del CIDCA.
- El *App* debe contar con medidas de seguridad, para este caso encriptación en MD5.
- El *App* debe contar con un sistema de desconexión por inactividad tras 30 minutos de la misma.
- El *App* debe desarrollarse exclusivamente para *Android*
- Debe iniciar con una pantalla de logueo que contenga los campos de usuario y contraseña.
- Después del logueo debe mostrar dos iconos para la consulta de horarios y notas.
- Debe contar en todas las secciones del *software* con un icono para realizar desconexión y salida del aplicativo.
- Debe contar con un icono de regreso al *frame* anterior, así mismo poder regresar con el botón de retroceso del móvil.

- Debe contar con una reorganización del horario para que se imprima en pantalla de una forma más cómoda para la comunidad estudiantil que la actual.
- El Icono de inicio debe contar con el logo de la Fundación CIDCA mas el nombre del *App*.
- Los colores utilizados pueden ser de libre elección por el programador.
- Entorno grafico agradable, amigable y fácil de usar.

Interfaces externas. Las condiciones son:

- Aplicaciones instaladas en el servidor de aplicaciones, que puedan entrar en conflicto con el *web service*.
- Aplicaciones instaladas en los teléfonos móviles que puedan entrar en conflicto con la instalación de "**MICIDCA**".

Beneficios de la implementación. Contempla los siguientes:

- **Operacionales.** Podemos indicar que la operación es el día a día de la institución en que la comunidad estudiantil debe realizar la consulta de sus notas y horarios por lo que este es el beneficio más importante de todos, ya que se está dando una alternativa de solución a este, se puede decir que si anteriormente se debe tener acceso a un pc para poder realizar la consulta y ahora vamos a poder hacerlo desde nuestros móviles que siempre tenemos encendidos y disponibles vamos a tener un beneficio significativo de tiempo y disponibilidad, la mejora en este beneficio es de alrededor de un **70%**.
- **De gestión.** Para la Fundación CIDCA es muy importante contar con nuevas tecnologías para acceso a la información, ya que esto le da un plus importante dentro de la comunidad estudiantil, el cual demuestra el interés de la universidad por ofrecer unas mejoras importantes para los estudiantes y docentes.

- **Estratégicos.** Para la Fundación CIDCA es muy importante contar con esta nueva tecnología, ya que es tendencia mundial para acceso a la información, así mismo poder empezar a explotar mucho más este acceso. La institución puede continuar con el desarrollo de la plataforma para ofrecer a sus estudiantes y docentes una herramienta mucho más completa para incrementar el número de servicios que se pueden acceder desde ella.
- **De infraestructura.** Uno de los beneficios más importantes es lograr dar por terminada la dependencia de las computadoras para realizar nuestras consultas y poder tener dos **(2)** alternativas de la misma, así mismo se complementarían para ser contingencia una de la otra y poder ofrecer a la comunidad estudiantil una mayor disponibilidad a la información.
- **De it.** Obviamente la tecnología juega un papel muy importante en este beneficio ya que se está posicionando a la institución en el presente de esta tecnología, haciendo que esta sea más competitiva, esto siempre y cuando se logre llevar el desarrollo de la plataforma a unos estándares mucho más altos.

3.8. MATERIALES Y MÉTODOS

3.8.1 Materiales. A continuación se hace una relación de las herramientas requeridas para el proceso de desarrollo del proyecto, tanto en *software* como en *Hardware*:

Se adquirieron tres computadores portátiles con las siguientes características para diseñar, modelar y compilar la aplicación, dos equipos con un procesador *Core i7-4510u de 2.6HGz*, con tecnología *VT-X*, *4GB* en memoria *RAM* y *1 TB* en disco duro.

El otro equipo posee un procesador *Core i5* de quinta generación con tecnología *VT-X*, *4GB* en memoria *RAM*, *1 TB* en disco duro.

El requerimiento de la tecnología *VT-X* en los procesadores, es indispensable para el proceso de virtualización del OS *Android* y poder correr las respectivas pruebas de funcionamiento.

Se está utilizando el *IDE Eclipse Luna* para compilar la aplicación teniendo en cuenta que este tienen mayor compatibilidad con el entorno *Android*, también se utiliza *Oracle DB 9.2.8* el cual se encuentra instalado en la Fundación CIDCA y para la cual se está llevando el desarrollo, se está trabajando con *Office 2010* y *2013* para el proceso de documentación del proyecto.

3.8.2. Metodologías. Se aplicaron las siguientes:

- Presentación de trabajos escritos con normas *ICONTEC*, para el anteproyecto.
- Programación *Extrema XP*, para el desarrollo del *web service* y el *App* móvil.
- Especificación de requisitos según el estándar *IEEE 830*.

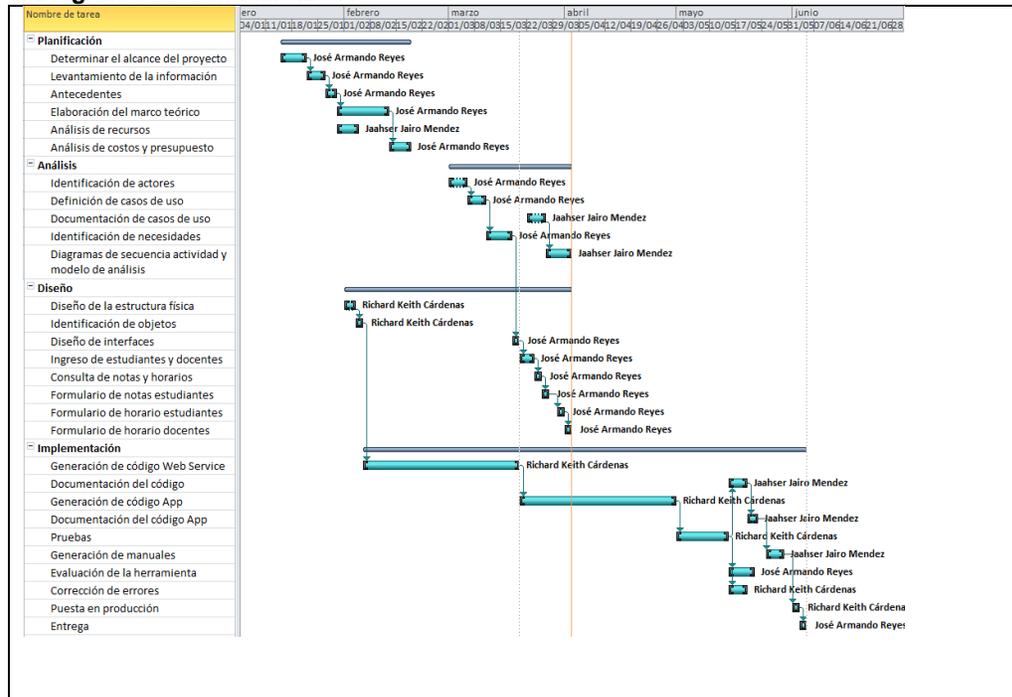
3.9. DESARROLLO DEL PROYECTO

- Se ha presentado un gran entusiasmo por parte de comunidad estudiantil de la Fundación CIDCA, por que el proyecto pueda llevarse a cabo.
- Se presenta más dificultad en el desarrollo del *web service* que en la *App*
- El tiempo para dedicar al proyecto se presenta como el inconveniente
- **Análisis del desarrollo del proyecto.** En la socialización de la puesta en marcha del *App* ha tenido gran aceptación en la comunidad estudiantil ya la tendencia del uso de dispositivos móviles en su mayoría celulares con *Android*, para todos resultaría muy cómodo, una vez instalado el aplicativo, la consulta

de notas y horarios en tiempo real, máxime el inconformismo que ha generado el manejo del actual sistema.

- **El desarrollo del web service:** el ajuste de la conexión con la base de datos, una acción que es común entre los programadores, se convirtió en un acción que debía llevarse al mismo tiempo, combinar o incluir los parámetros ya establecidos por la Fundación CIDCA y la configuración de su sistema de información actual, pero en el levantamiento de las bases teóricas, se hayo la documentación para diseñar y desarrollar el *software* requerido por la institución.
- Una de las limitantes de mayor incidencia fue el tiempo para terminar el *software*, por las múltiples actividades que los involucrados poseen, en roles diferentes en a este proyecto. **Ver figura 3.**

Figura 3. Diagrama de Gantt *MiCIDCA*



Fuente: Elaboración propia

4. ANÁLISIS Y RESULTADOS

4.1. RESULTADOS DE LAS FASES METODOLÓGICAS

4.1.1. Fase I. Planificación y levantamiento de información. Los resultados obtenidos fueron los siguientes:

a. *Check list*: En cuanto al *App* se enumeraron errores y necesidades tales como:

- Dificultad para realizar las consultas de notas y horarios en línea.
- Falta de portabilidad en la consultas.

b. Entrevistas: Para alcanzar la mayor certeza en la información recopilada se realizaron entrevistas no estructuradas con el fin de sondear e identificar la problemática para la que se crearía la *App* propuesta en este trabajo de grado.

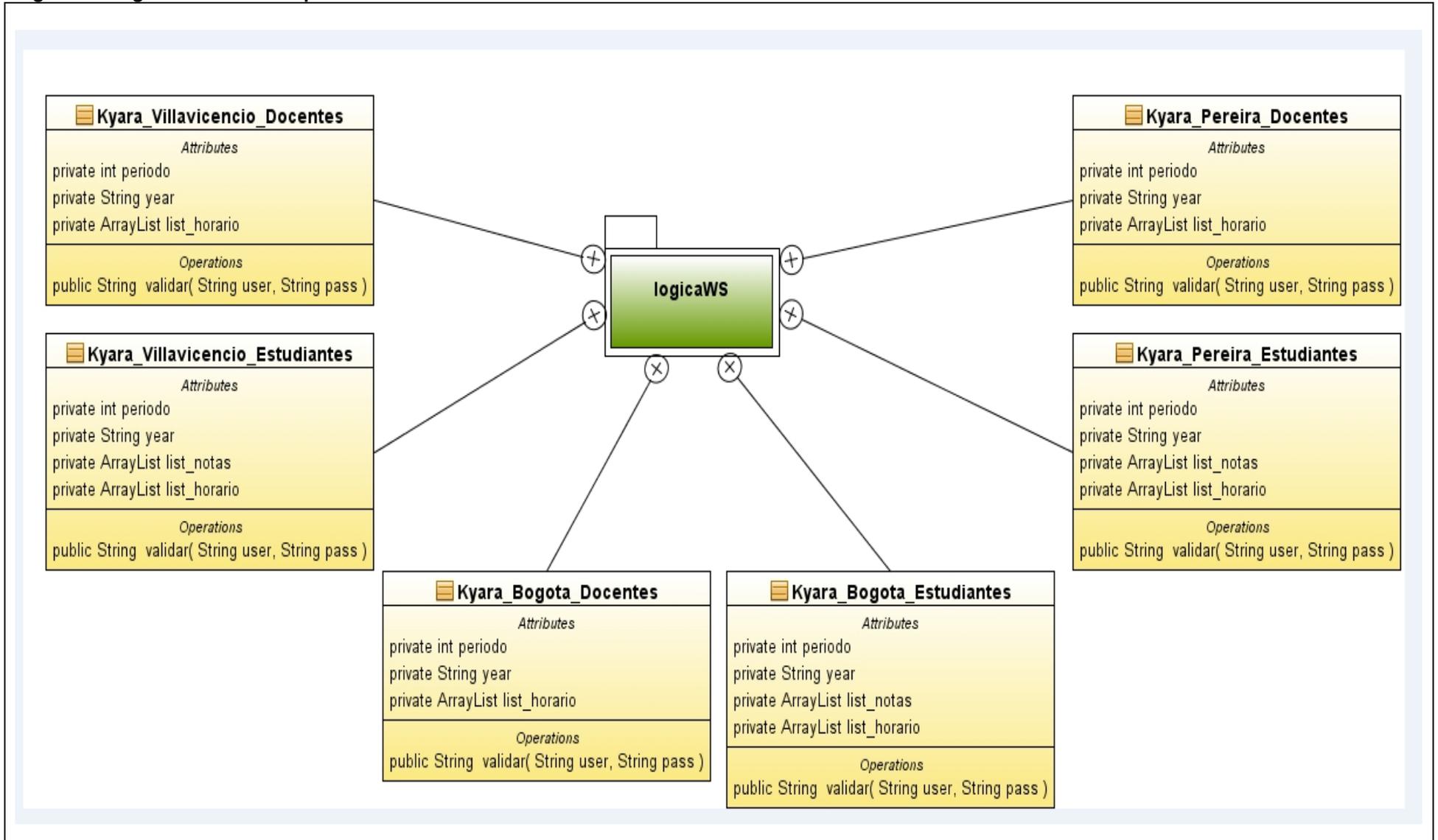
c. Consulta y selección de fuentes bibliográficas y de internet: se recurrió a la búsqueda de todos los textos y consulta en la *web* de todo lo referente a la programación, desarrollo y diseño del tipo de *App* que se decidió implementar con los directivos de la institución, una vez se identificaron los problemas a solucionar. Con estas se construyó el marco teórico-conceptual, un glosario y la lista de siglas que se manejaron para realizar el presente trabajo de grado.

4.1.2. Fase II. Análisis y diseño de “MiCIDCA”. Desarrollada esta etapa los resultados arrojados fueron los siguientes:

El diseño de la *App* debe contener para responder a la problemática de la Fundación CIDCA como mínimo el diagrama de clases (**Ver figura 4**) y los casos de uso que se describen en los **cuadros 4, 5 y 6**.

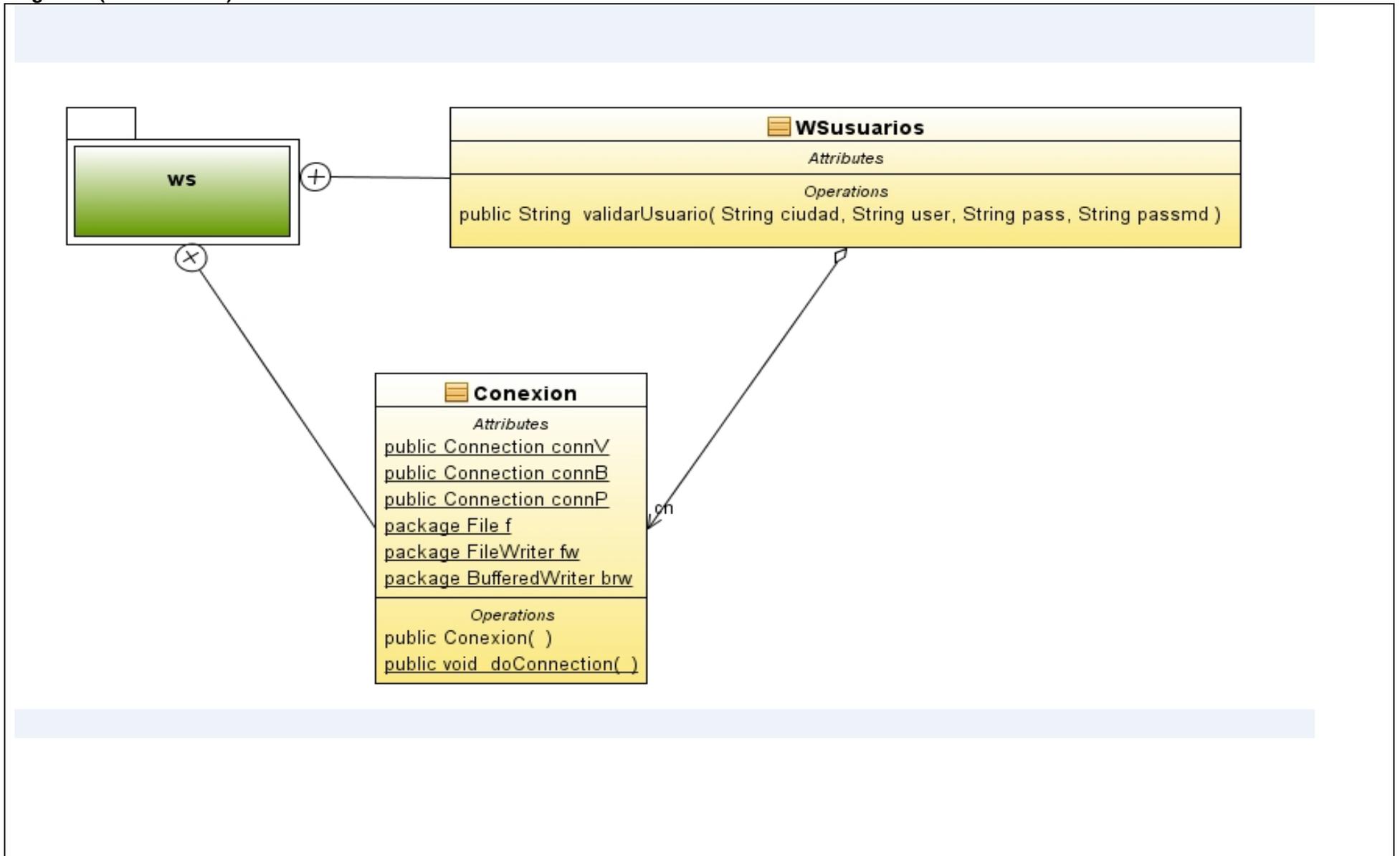
- Diagramas de Clases. Ver figura 4.

Figura 4. Diagramas de Clases para “MiCIDCA”



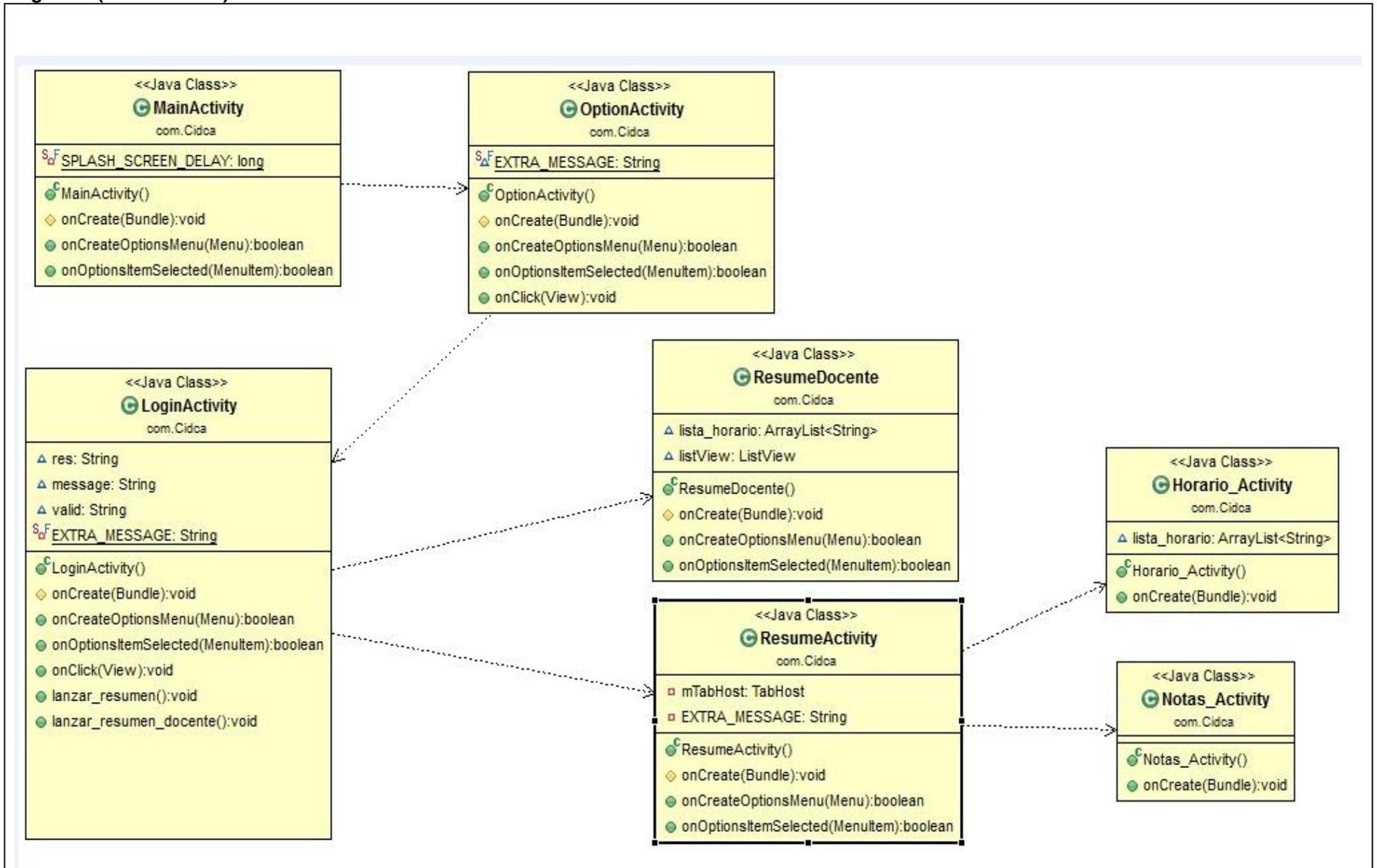
Fuente: Elaboración propia

Figura 4. (Continuación)



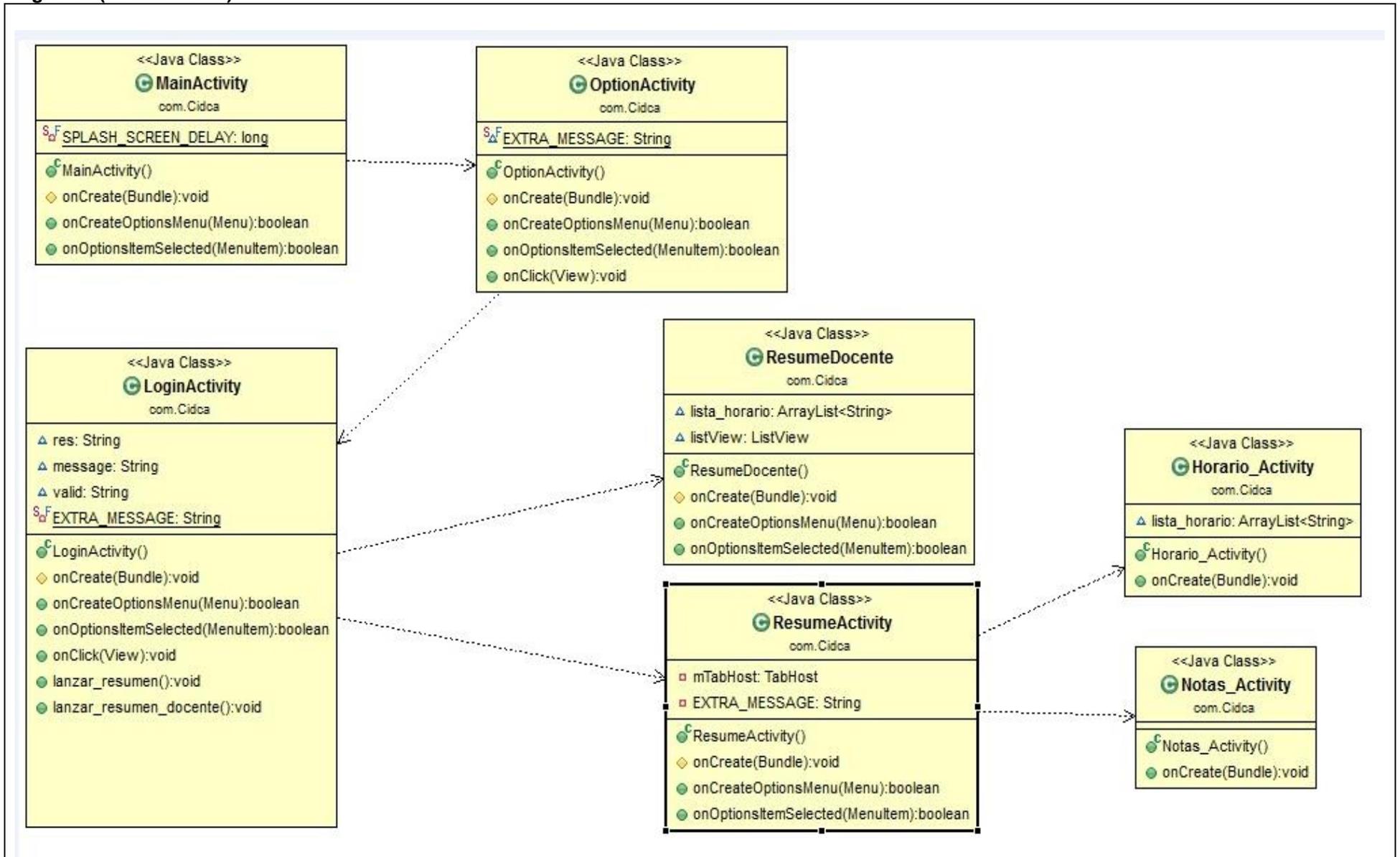
Fuente: Elaboración propia

Figura 4. (Continuación)



Fuente: Elaboración propia

Figura 4. (Continuación)



Fuente: Elaboración propia

Cuadro 4. Caso de uso Consulta notas estudiantes

ESPECIFICACIÓN DEL CASO DE USO: Consulta de notas estudiantes			
ID:	CSMC-01		
NOMBRE:	Consulta de notas estudiantes		
<pre> graph TD Estudiante((Estudiante)) U1(Selecdonar Sede) U2(Selecdonar Estudiante) U3(Ingresar Código) U4(Ingresar Clave) U5(Consultar Notas) U1 --> U2 U2 --> U3 U3 --> U4 U4 --> U5 Estudiante --> U1 Estudiante --> U2 Estudiante --> U3 Estudiante --> U4 Estudiante --> U5 </pre>			
DESCRIPCIÓN:	Se presenta cuando un estudiante quiere realizar la consulta de sus notas publicadas por la Fundación CIDCA desde su teléfono móvil.		
AUTOR:	José Armando Reyes, Richard Cárdenas, Jaahser Méndez		
FECHA DE CREACIÓN:	15/05/2015	FECHA ULTIMA MODIFICACIÓN:	27/05/2015
ACTORES:	Estudiante		
PRECONDICIONES:	* Para poder realizar la consulta de sus notas, el estudiante debe estar activo en el sistema.		
POS CONDICIONES:	Ingreso de sesión realizado con éxito Tomar correctamente la información de las notas del periodo en curso.		
FLUJO NORMAL DE EVENTOS			
<ul style="list-style-type: none"> • Validar: <ol style="list-style-type: none"> 1. El actor debe seleccionar la Sede a la cual pertenece 2. El actor debe seleccionar el tipo de usuario 3. El actor debe diligenciar el campo de código registrando el asignado 4. El actor debe diligenciar el password relacionado con el código 5. El sistema validara el estado del actor 6. El sistema presentara una respuesta positiva o negativa según el estado del actor 7. El Actor tendrá acceso a la visualización de sus notas del periodo en curso 			
FLUJOS ALTERNOS			
<ul style="list-style-type: none"> • Cancelar Ingreso <ol style="list-style-type: none"> 1. El actor decide cancelar el ingreso a la consulta de notas 2. El sistema cancela el acceso del actor por usuario inactivo 3. Se cancela el acceso del actor por falla de comunicaciones 			
EXCEPCIONES			
<ul style="list-style-type: none"> • Datos Faltantes o errados: <ol style="list-style-type: none"> 1. Se muestra un mensaje por el sistema indicando usuario no existe si es el caso 2. Se muestra un mensaje por el sistema indicando que la clave esta errada 			
REFERENCIA:			
ANOTACIONES:	El actor podrá realizar la consulta de su horario y notas, sin necesidad de iniciar todo el proceso de ingreso		

Fuente: elaboración propia

Cuadro 5. Caso de uso-Consulta horario estudiantes

ESPECIFICACIÓN DEL CASO DE USO: Consulta de horario estudiantes			
ID:	CSMC-02		
NOMBRE:	Consulta de horario estudiantes		
<pre> graph TD Estudiante((Estudiante)) --- Sede([Seleccionar Sede]) Estudiante --- Estudiante_UseCase([Seleccionar Estudiante]) Estudiante ---Codigo([Ingresar Código]) Estudiante --- Clave([Ingresar Clave]) Estudiante --- Horario([Consultar Horario]) Sede --> Estudiante_UseCase Estudiante_UseCase --> Codigo Codigo --> Clave Clave --> Horario </pre>			
DESCRIPCIÓN:	Se presenta cuando un estudiante quiere realizar la consulta de su horario de clase publicado por la Fundación CIDCA desde su teléfono móvil.		
AUTOR:	José Armando Reyes, Richard Cárdenas, Jasser Méndez		
FECHA DE CREACIÓN:	15/05/2015	FECHA ULTIMA MODIFICACIÓN:	27/05/2015
ACTORES:	Estudiante		
PRECONDICIONES:	* Para poder realizar la consulta de su horario, el estudiante debe estar activo en el sistema y tener materias inscritas		
POS CONDICIONES:	Ingreso de sesión realizado con éxito Tomar correctamente la información del horario de clase del periodo en curso.		
FLUJO NORMAL DE EVENTOS			
<ul style="list-style-type: none"> • Validar: <ol style="list-style-type: none"> 1. El actor debe seleccionar la Sede a la cual pertenece 2. El actor debe seleccionar el tipo de usuario 3. El actor debe diligenciar el campo de código registrando el asignado 4. El actor debe diligenciar el password relacionado con el código 5. El sistema validara el estado del actor 6. El sistema presentara una respuesta positiva o negativa según el estado del actor 7. El Actor tendrá acceso a la visualización de su horario del periodo en curso 			
FLUJOS ALTERNOS			
<ul style="list-style-type: none"> • Cancelar Ingreso <ol style="list-style-type: none"> 1. El actor decide cancelar el ingreso a la consulta de horario 2. El sistema cancela el acceso del actor por usuario inactivo 3. Se cancela el acceso del actor por falla de comunicaciones 			
EXCEPCIONES			
<ul style="list-style-type: none"> • Datos Faltantes o errados: <ol style="list-style-type: none"> 1. Se muestra un mensaje por el sistema indicando usuario no existe si es el caso 2. Se muestra un mensaje por el sistema indicando que la clave esta errada 			
REFERENCIA:			
ANOTACIONES:	El actor podrá realizar la consulta de su horario y notas, sin necesidad de iniciar todo el proceso de ingreso		

Fuente: Elaboración propia

Cuadro 6. Caso de uso-Consulta horario docentes

ESPECIFICACIÓN DEL CASO DE USO: Consulta de horario docente			
ID:	CSMC-03		
NOMBRE:	Consulta de horario docentes		
<pre> graph TD Docente((Docente)) --- U1(Seleccionar Sede) U1 --> U2(Seleccionar Docente) U2 --> U3(Ingresar Código) U3 --> U4(Ingresar Clave) U4 --> U5(Consultar Horario) </pre>			
DESCRIPCIÓN:	Se presenta cuando un docente quiere realizar la consulta de su horario de asignación de clases publicado por la Fundación CIDCA desde su teléfono móvil.		
AUTOR:	José Armando Reyes, Richard Cárdenas, Jasser Méndez		
FECHA DE CREACIÓN:	15/05/2015	FECHA ULTIMA MODIFICACIÓN:	27/05/2015
ACTORES:	Docente		
PRECONDICIONES:	*Para poder realizar la consulta de su horario asignado, el docente debe estar activo en el sistema y tener clases asignadas		
POS CONDICIONES:	Ingreso de sesión realizado con éxito Tomar correctamente la información del horario de clase asignadas del periodo en curso.		
FLUJO NORMAL DE EVENTOS			
<ul style="list-style-type: none"> • Validar: <ol style="list-style-type: none"> 1. El actor debe seleccionar la Sede a la cual pertenece 2. El actor debe seleccionar el tipo de usuario 3. El actor debe diligenciar el campo de código registrando el asignado 4. El actor debe diligenciar el password relacionado con el código 5. El sistema validara el estado del actor 6. El sistema presentara una respuesta positiva o negativa según el estado del actor 7. El Actor tendrá acceso a la visualización de su horario asignado del periodo en curso 			
FLUJOS ALTERNOS			
<ul style="list-style-type: none"> • Cancelar Ingreso <ol style="list-style-type: none"> 1. El actor decide cancelar el ingreso a la consulta de horarios asignados 2. El sistema cancela el acceso del actor por usuario inactivo 3. Se cancela el acceso del actor por falla de comunicaciones 			
EXCEPCIONES			
<ul style="list-style-type: none"> • Datos Faltantes o errados: <ol style="list-style-type: none"> 1. Se muestra un mensaje por el sistema indicando usuario no existe si es el caso 2. Se muestra un mensaje por el sistema indicando que la clave esta errada 			
REFERENCIA:			
ANOTACIONES:	El actor tendrá acceso a la información de hora en la que debe impartir la clase, además del salón asignado.		

Fuente: elaboración propia

4.1.3. Fase III. Desarrollo de “MiCIDCA”. En el desarrollo del *App*, se realizaron los siguientes pasos:

- d. El lenguaje para la programación fue *Java*.
- e. El modo de conexión del *App* a la Base de Datos, fue mediante el *web service*
- f. El sistema operativo en los dispositivos móviles seleccionado fue *Android*.

4.1.4. Fase IV. Pruebas y ajustes de “MiCIDCA”. Finalizada la etapa se obtuvieron los siguientes resultados:

Se realizaron pruebas y se presentan a continuación así:

- **Revisión del código de la aplicación:** se realizaron pruebas de escritorio y una revisión sistemática del código fuente para la detección de posibles fallos y permitir corregir la *App* gradualmente hasta el producto final compilando varias veces para depurar el código.
- **Pruebas unitarias:** se realizaron pruebas por separado tanto al *web service* como al *App* para verificar su funcionalidad desde *Eclipse*.
- **Pruebas de integración:** se montó un servidor de pruebas con la misma configuración y estructura de la base de datos que tiene la Fundación CIDCA, para poder probar la funcionalidad de la *App* conectado al *web service* y a la base de datos para probar conectividad, disponibilidad, y detectar posibles fallos en el ambiente antes de salir a producción la *App*.
- **Prueba Alfa:** se montó una prueba Alfa en el *Store de Google Play* para realizar la verificación y corrección de posibles fallos antes de publicar la *App*, haciendo uso de la herramienta de comprobación de código en la cuenta para programadores que ofrece el mismo store.

- **Prueba Beta:** se realizó una publicación Beta en el *store* de *Google Play* para realizar pruebas y verificar posible inestabilidad en la *App* para su posterior corrección.

4.1.5. Fase V. Montaje, presentación y entrega final de “MiCIDCA”. Se procede a montar el *App* en el *Store Google Play* se presenta a los directivos de la Fundación CIDCA.

4.1.6. Fase VI. Elaboración de manuales de usuario para “MiCIDCA”. Los manuales de usuario-cliente, para el correcto manejo de *App*.

Manejo Básico de la aplicación

Revisión básica del código de programación

Posibles fallas del *App*,

5. CONCLUSIONES

- Para el desarrollo de la aplicación se han logrado encontrar una gran variedad de tutoriales que apoyan la creación de la misma con todas las especificaciones propuestas inicialmente, esta gran cantidad de material disponible ha facilitado el desarrollo de la misma.
- A pesar que no se ha terminado el proyecto con su implementación se ha generado una gran expectativa por parte de la comunidad que conocen el proyecto, el cual esperan que sea todo un éxito y poder usar el aplicativo cuanto antes.
- Con los conocimientos adquiridos en el transcurso de la carrera de ingeniería de sistemas, se es suficiente para lograr llevar a cabo un desarrollo de *software* de calidad.
- En la actualidad es bastante sencillo poder desarrollar aplicaciones para móviles gracias a la tendencia de uso de estos dispositivos, a la demanda que se presenta dentro de la comunidad actual, y sumado a esto la gran cantidad de herramientas fáciles de usar y documentación para este fin.
- Al estar desarrollando la aplicación se desearía poder contar con más disponibilidad y recursos para hacerla más completa de acuerdo a las necesidades de los estudiantes y docentes, como poder inscribir materias o subir notas

Dando rigor de ingeniería y a través de la metodología extrema (*X.P*), se logró crear una herramienta brindando un mejor funcionamiento a la consulta de notas y horarios por los estudiantes y horarios por los docentes.

Aplicando los conocimientos adquiridos en pregrado se realizó el análisis respectivo del proceso operativo y los diferentes procedimientos generados en la institución y se obtuvo el modelado y arquitectura adecuados para desarrollar el *App* que aporte la solución práctica a la problemática evidenciada en la Fundación CIDCA.

Se logró montar un *App Android*, que dé solución a los requerimientos de consulta de notas y horarios que tiene la Fundación CIDCA.

Finalmente se cumplió el objetivo de desarrollar una *App Android* "**MiCIDCA**", para consulta de notas y horarios por los estudiantes y horarios por los docentes de la Fundación CIDCA.

6. RECOMENDACIONES

Se recomienda culminar el trabajo de desarrollo del aplicativo para hacerlo más funcional de acuerdo con las necesidades de la comunidad estudiantil.

Se recomienda poder incluir en un futuro dentro de la *APP* una función para generar notificaciones e información de forma masiva por parte de la institución para la comunidad.

La fundación CIDCA debe adoptar como suyo el aplicativo, e incluirlo dentro de su inventario con el fin de dar sostenimiento al mismo, y poder presupuestar mantenimientos y actualizaciones necesarias para su correcto funcionamiento.

Socializar la aplicación para que los estudiantes nuevos o que aún no tengan el aplicativo y sean incentivados a usarlo.

REFERENCIAS

Audisistem. (26 de 08 de 2012). Construcción del sistema de información. Recuperado el 15 de 03 de 2015, de <http://audisitem.webnode.com.co/news/construccion-del-sistema-de-informacion/>

Eslava, V. (2013). El nuevo PHP conceptos avanzados. 89. España: Bobok Publishing S.L. Recuperado el 15 de 03 de 2015, de <http://www.amazon.es/El-nuevo-PHP-Conceptos-avanzados/dp/8468644331>

Fundation Apache. (1999). Software Apache Tomcat. Estados Unidos. Recuperado el 15 de 03 de 2015, de Disponible en Internet: <http://tomcat.apache.org/>

Ministerio de Educación. (2014.). Qué tanto sabe sobre licenciamiento? Bogotá. Recuperado el 10 de 03 de 2015, de <http://www.colombiaaprende.edu.co/html/docentes/1596/article-73576.html>

Pedrozo, G. (2012). Sistemas Operativos en Dispositivos. Argentina. Recuperado el 10 de 03 de 2015, de http://exa.unne.edu.ar/informatica/SO/Sistemas_Operativos_en_Dispositivos_Moviles.pdf

Retamar, Á. (2012). Mi primera hora con eclipse. Asturias, España. Recuperado el 10 de 03 de 2015, de <http://telecable.es/personales/retamar/articulos/eclipse.pdf>

Sommerville, I. (2005). Ingeniería de software. Madrid, España. Recuperado el 10 de 03 de 2014, de http://zeus.inf.ucv.cl/~bcrawford/AULA_ICI441/Ingenieria%20del%20Software%207ma.%20Ed.%20-%20lan%20Sommerville.pdf

Telocapa, M. (2013). Lenguaje de programación en Java. 1. México. Recuperado el 15 de 03 de 2015, de <http://www.academica.mx/blogs/lenguaje-programacion-en-java.html>

Wells, D. (2001). Ciclo de vida de un proyecto xp. Recuperado el 15 de 03 de 2015, de <http://oness.sourceforge.net/proyecto/html/ch05s02.html>

ANEXOS

Anexo A. Manual de usuario con privilegios para “MiCIDCA”

MANUAL BÁSICO
APP ANDROI “MiCIDCA”

Consulta de Notas y Horarios
Estudiantes
Consulta de Horarios Docentes

Fundación CIDCA

Autores

RICHARD KEITH CÁRDENAS VESGA
JAAHSER JAIRO MÉNDEZ RODRÍGUEZ
JOSÉ ARMANDO REYES BRAVO

2015

CONTENIDO

	Pág.
1. Descargar App.....	74
2. Ingreso App a Institución	745
3. Activity de Bienvenida	76
4. Activity permite seleccionar la sede y el usuario.....	77
5. Activity muestra el login del usuario y sede seleccionados.....	78
6. El login exitoso los datos del Estudiante	80
7. Activity muestra el login para el docente.....	81

LISTA DE ILUSTRACIONES

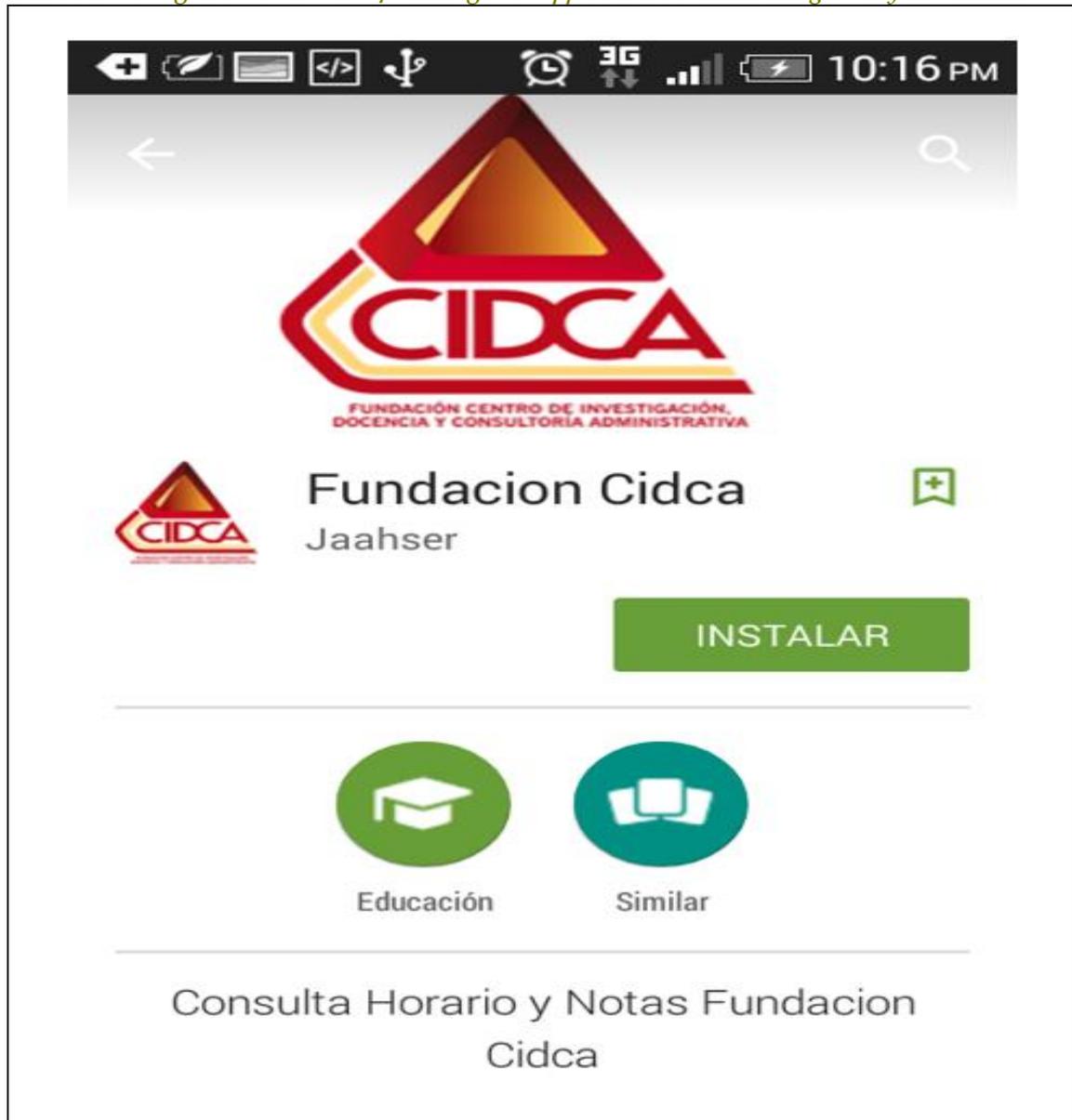
	Pág.
Ilustración 1. Ingreso “MiCIDCA”/Descargar el <i>App</i> desde del <i>Store Google Play</i> de Android.....	74
Ilustración 2. Ingreso a <i>App</i>	755
Ilustración 3. <i>App</i> muestra el <i>Activity</i> de bienvenida.....	766
Ilustración 4. <i>Activity</i> permite seleccionar la sede y el usuario.....	777
Ilustración 5. <i>Activity</i> muestra el <i>login</i> del usuario y sede seleccionados.....	78
Ilustración 6. El <i>login</i> exitoso muestra los datos del Estudiante	80
Ilustración 7. <i>Activity</i> muestra el <i>login</i> para el docente	81
Ilustración 8. El <i>login</i> exitoso muestra los datos del Docente.....	82

1. Descargar App

El usuario debe Descargar el App desde del Store Google Play de Android

Encontrará el pantallazo: Ver. Ilustración 1.

Ilustración 1. Ingreso "MiCIDCA"/Descargar el App desde del Store Google Play de Android



Fuente: Autores

2. Ingreso App a Institución

Una vez instalado el App se ubica en el dispositivo con el logo de la Institución

Encontrará el pantallazo: Ver. Ilustración 2.

Ilustración 2. Ingreso a App



Fuente: Autores

3. *Activity de Bienvenida*

Una vez iniciado el *App* mostrará el *Activity* de bienvenida

Encontrará el pantallazo: Ver. Ilustración 3.

Ilustración 3. *App* muestra el *Activity* de bienvenida



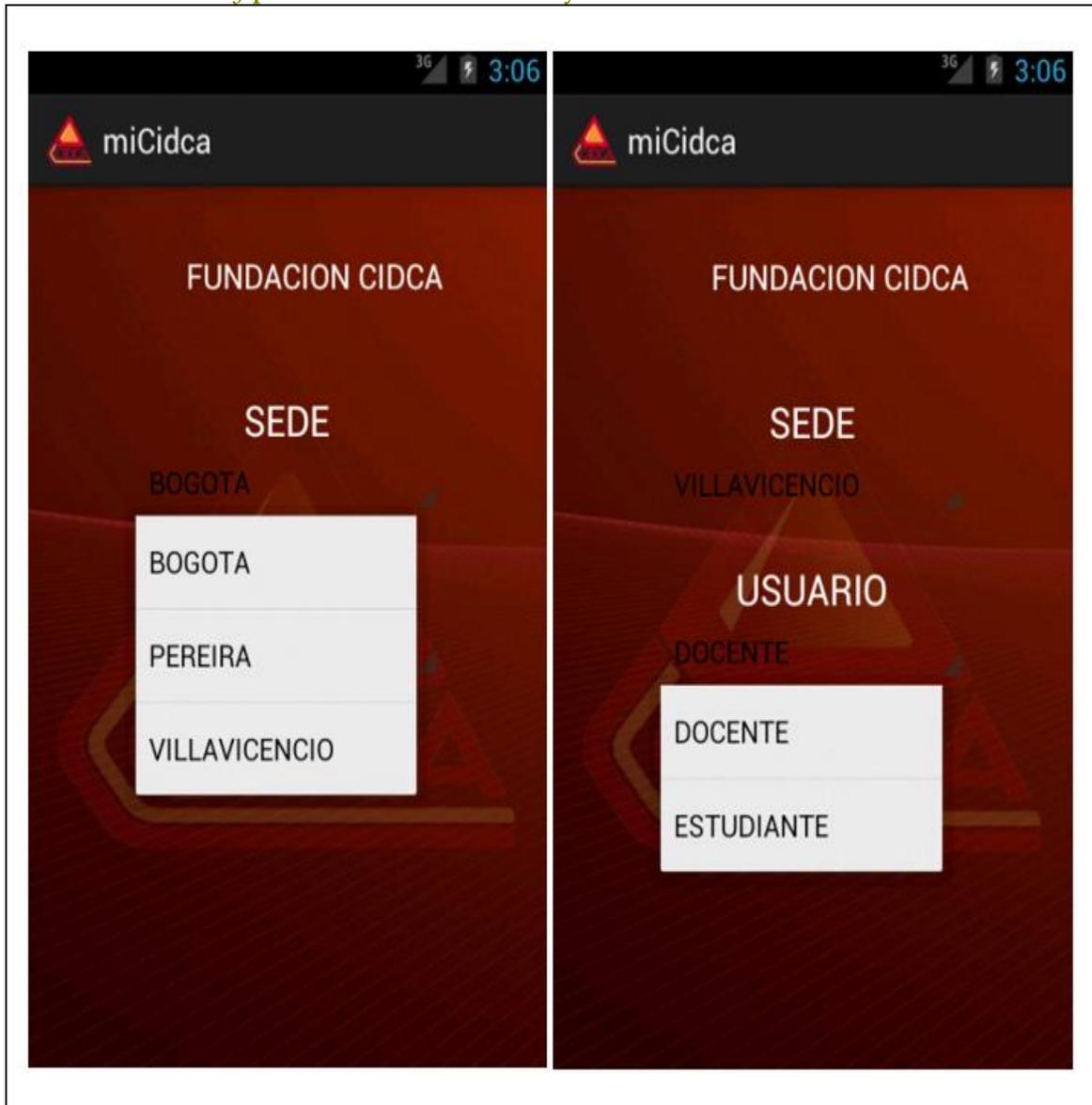
Fuente: Autores

4. *Activity* permite seleccionar la sede y el usuario

El siguiente *Activity* permite seleccionar la sede y el usuario

Encontrará el pantallazo: Ver. Ilustración 4.

Ilustración 4. *Activity* permite seleccionar la sede y el usuario



Fuente: Autores

5. *Activity* muestra el login del usuario y sede seleccionados

Este *Activity* muestra el login del usuario y sede seleccionados

Encontrará el pantallazo: Ver. Ilustración 5.

Ilustración 5. *Activity* muestra el login del usuario y sede seleccionados



Fuente: Autores

6. El *login* exitoso los datos del Estudiante

Si el *login* es exitoso mostrará los datos del Estudiante

Encontrará el pantallazo: Ver. Ilustración 6.

Ilustración 6. El *login* exitoso muestra los datos del Estudiante



The screenshot shows a mobile application interface with a dark red background. At the top, there is a status bar with '3G', a battery icon, and the time '3:07'. Below the status bar is a header with a logo and the word 'RESUMEN'. The main content area displays the student's name 'DIANA ALEJANDRA AGUIRRE SUAREZ', ID '1121936925', and profession 'TECNICO PROFESIONAL EN INDUSTRIAL (VILLAVICENCIO)'. Below this is a table with two sections: 'NOTAS' and 'HORARIO'. The 'NOTAS' section is highlighted with a blue bar. The table lists courses and their corresponding grades in columns C1, C2, C3, and DEF. At the bottom, there is a dark grey button with the text 'BIENVENIDO'.

NOTAS		HORARIO			
MATERIA		C1	C2	C3	DEF
DESARROLLO DEL PENSAMIENTO II	50	-	-	-	15
ESTADISTICA PROBABILISTICA	-	-	-	-	-
INVESTIGACION DE OPERACIONES I	-	-	-	-	-
MATEMATICAS III	-	-	-	-	-
INGLES I	-	-	-	-	-
SISTEMAS DE PRODUCCION	-	-	-	-	-
CONTROL TOTAL DE CALIDAD	-	-	-	-	-

Fuente: Autores

Ilustración 6. (Continuación)

The image is a screenshot of a mobile application interface. At the top, there is a status bar with '3G', a battery icon, and the time '3:08'. Below this is a dark header with a logo on the left and the word 'RESUMEN' in white. The main content area has a dark red background with white text. It displays the student's name 'DIANA ALEJANDRA AGUIRRE SUAREZ', ID '1121936925', and profession 'TECNICO PROFESIONAL EN INDUSTRIAL (VILLAVICENCIO)'. Below this, there are two tabs: 'NOTAS' and 'HORARIO'. The 'HORARIO' tab is selected, indicated by a blue underline. The schedule lists five classes with their times, locations, subjects, and teachers.

RESUMEN

**DIANA ALEJANDRA
AGUIRRE SUAREZ
1121936925
TECNICO PROFESIONAL EN INDUSTRIAL
(VILLAVICENCIO)**

NOTAS **HORARIO**

Lunes 14:45 Aula105
CONTROL TOTAL DE CALIDAD
FABIAN ANDRES CABRERA GUTIERREZ

Lunes 17:0 Aula105
MATEMATICAS III
NELSON JOSE GONZALEZ GAONA

Martes 14:45 Aula105
INGLES I
MARIA MARLEN GUTIERREZ BELTRAN

Martes 16:15 Aula105
SISTEMAS DE PRODUCCION
FLAMINIO ROLANDO CASTRO PEREZ

Miercoles 14:0 Aula105
DESARROLLO DEL PENSAMIENTO II

Fuente: Autores

7. Activity muestra el login para el docente

Este Activity muestra el login para el docente

Encontrará el pantallazo: Ver. Ilustración 7.

Ilustración 1. Activity muestra el login para el docente



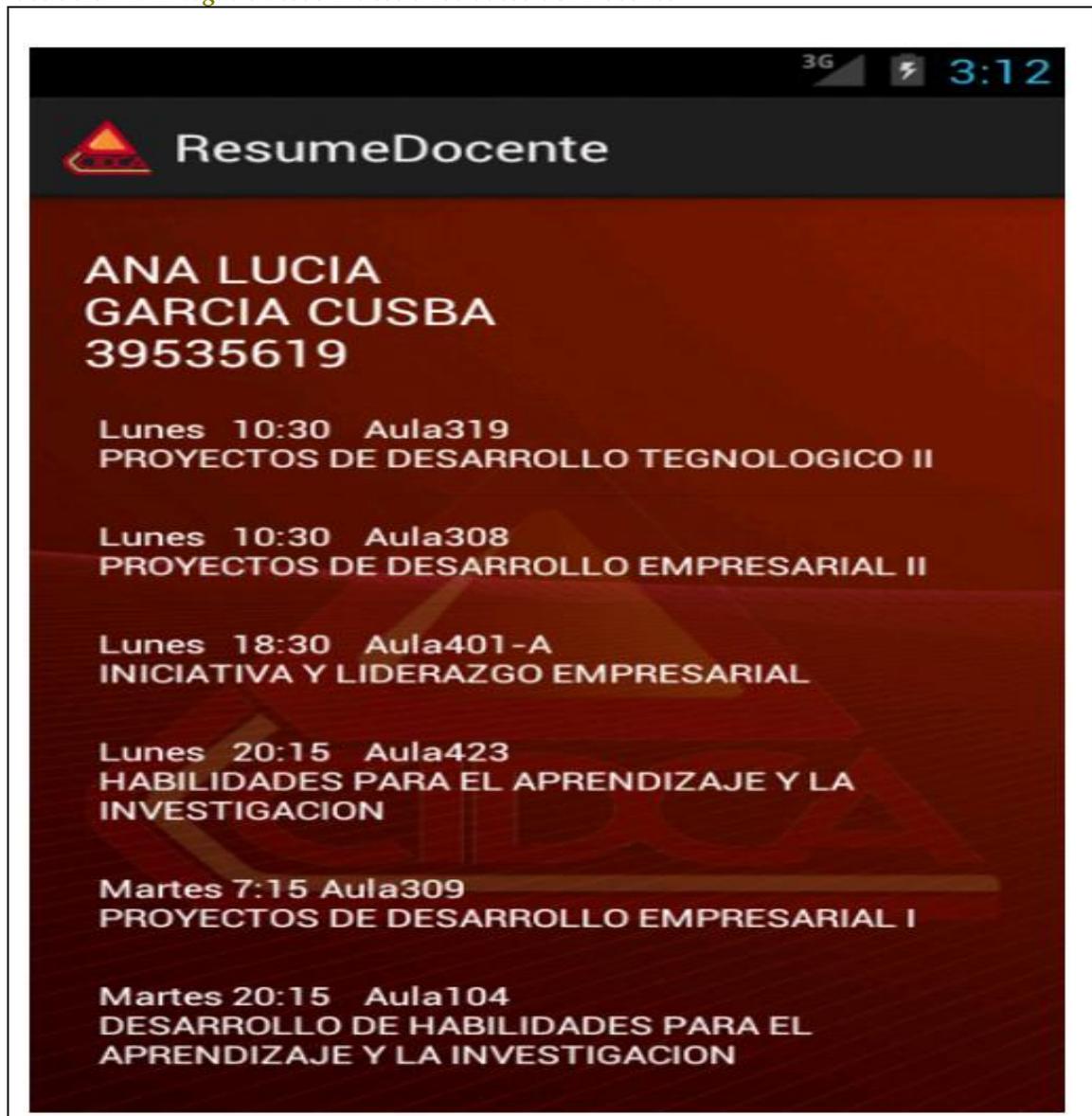
Fuente: Autores

8. El *login* exitoso mostrará los datos del Docente

Si el *login* es exitoso mostrará los datos del Docente

Encontrará el pantallazo: Ver. Ilustración 8.

Ilustración 2. El *login* exitoso muestra los datos del Docente



Fuente: Autores

