



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

**ORGANIZADOR ACADÉMICO APLICANDO NUEVAS TECNOLOGÍAS
MÓVILES, CON PARADIGMA MULTIPLATAFORMA Y REDES SOCIALES**

Luis Fernando Leal Hernández

Marvin Francisco Castillo Barrera

Asesorados por el Ing. Edgar Estuardo Santos Sutuj

Guatemala, marzo de 2015

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA



FACULTAD DE INGENIERÍA

**ORGANIZADOR ACADÉMICO APLICANDO NUEVAS TECNOLOGÍAS
MÓVILES, CON PARADIGMA MULTIPLATAFORMA Y REDES SOCIALES**

TRABAJO DE GRADUACIÓN

PRESENTADO A LA JUNTA DIRECTIVA DE LA
FACULTAD DE INGENIERÍA
POR

LUIS FERNANDO LEAL HERNÁNDEZ
MARVIN FRANCISCO CASTILLO BARRERA
ASESORADOS POR EL ING. EDGAR ESTUARDO SANTOS SUTUJ

AL CONFERÍRSELES EL TÍTULO DE

INGENIEROS EN CIENCIAS Y SISTEMAS

GUATEMALA, MARZO DE 2015

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA
FACULTAD DE INGENIERÍA



NÓMINA DE JUNTA DIRECTIVA

DECANO	Ing. Murphy Olympo Paiz Recinos
VOCAL I	Ing. Angel Roberto Sic García
VOCAL II	Ing. Pablo Christian de León Rodríguez
VOCAL III	Inga. Elvia Miriam Ruballos Samayoa
VOCAL IV	Br. Narda Lucía Pacay Barrientos
VOCAL V	Br. Walter Rafael Véliz Muñoz
SECRETARIO	Ing. Hugo Humberto Rivera Pérez

TRIBUNAL QUE PRACTICÓ EL EXAMEN GENERAL PRIVADO

DECANO	Ing. Murphy Olympo Paiz Recinos
EXAMINADOR	Ing. Edgar Josué González Constanza
EXAMINADOR	Ing. Cesar Augusto Fernández Cáceres
EXAMINADOR	Ing. Roberto Estuardo Ruiz Cruz
SECRETARIO	Ing. Hugo Humberto Rivera Pérez

Luis Fernando Leal Hernández

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA
FACULTAD DE INGENIERÍA



NÓMINA DE JUNTA DIRECTIVA

DECANO	Ing. Murphy Olympo Paiz Recinos
VOCAL I	Ing. Angel Roberto Sic García
VOCAL II	Ing. Pablo Christian de León Rodríguez
VOCAL III	Inga. Elvia Miriam Ruballos Samayoa
VOCAL IV	Br. Narda Lucía Pacay Barrientos
VOCAL V	Br. Walter Rafael Véliz Muñoz
SECRETARIO	Ing. Hugo Humberto Rivera Pérez

TRIBUNAL QUE PRACTICÓ EL EXAMEN GENERAL PRIVADO

DECANO	Ing. Murphy Olympo Paiz Recinos
EXAMINADOR	Ing. Oscar Alejandro Paz Campos
EXAMINADOR	Ing. Pedro Pablo Hernández
EXAMINADOR	Ing. Marlon Francisco Orellana López
SECRETARIO	Ing. Hugo Humberto Rivera Pérez

Marvin Francisco Castillo Barrera

HONORABLE TRIBUNAL EXAMINADOR

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

ORGANIZADOR ACADÉMICO APLICANDO NUEVAS TECNOLOGÍAS MÓVILES, CON PARADIGMA MULTIPLATAFORMA Y REDES SOCIALES

Tema que nos fuera asignado por la Dirección de la Escuela de Ingeniería en Ciencias y Sistemas, con fecha febrero de 2013.

Marvin Francisco Castillo Barrera

Luis Fernando Leal Hernández

Guatemala, 05 de noviembre de 2014.

Ingeniero
Marlon Antonio Pérez Türk
Director
Escuela de Ciencias y Sistemas
Facultad de Ingeniería
Universidad de San Carlos de Guatemala

Ingeniero Pérez Türk:

Me complace saludarle, haciendo referencia al trabajo de graduación titulado "ORGANIZADOR ACADÉMICO APLICANDO NUEVAS TECNOLOGÍAS MÓVILES, CON PARADIGMA MULTIPLATAFORMA Y REDES SOCIALES", desarrollado por el estudiante Marvin Francisco Castillo Barrera con número de carné 200615067 y el estudiante Luis Fernando Leal Hernández con número de carné 200715367, que como asesor apruebo el contenido del mismo.

Para su conocimiento y efectos, sin otro particular, me suscribo.



Atentamente,
Ing. Edgar Santos
No. Colegiado: 5266
Asesor

Edgar Santos
INGENIERO EN CIENCIAS Y SISTEMAS
Colegiado 5266



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

Guatemala, 11 de Febrero del 2015

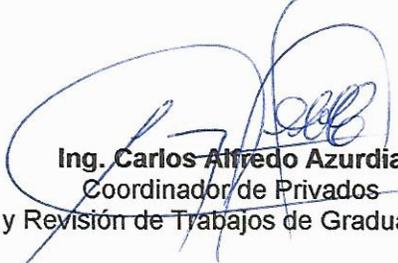
Ingeniero
Marlon Antonio Pérez Türk
Director de la Escuela de Ingeniería
En Ciencias y Sistemas

Respetable Ingeniero Pérez:

Por este medio hago de su conocimiento que he revisado el trabajo de graduación de los estudiantes **MARVIN FRANCISCO CASTILLO BARRERA** con carné **2006-15067**, y **LUIS FERNANDO LEAL HERNÁNDEZ** con carné **2007-15367**, titulado: **“ORGANIZADOR ACADÉMICO APLICANDO NUEVAS TECNOLOGÍAS MÓVILES, CON PARADIGMA MULTIPLATAFORMA Y REDES SOCIALES”**, y a mi criterio el mismo cumple con los objetivos propuestos para su desarrollo, según el protocolo.

Al agradecer su atención a la presente, aprovecho la oportunidad para suscribirme,

Atentamente,


Ing. Carlos Alfredo Azurdia
Coordinador de Privados
y Revisión de Trabajos de Graduación



UNIVERSIDAD DE SAN CARLOS
DE GUATEMALA



FACULTAD DE INGENIERÍA
ESCUELA DE CIENCIAS Y SISTEMAS
TEL: 24767644

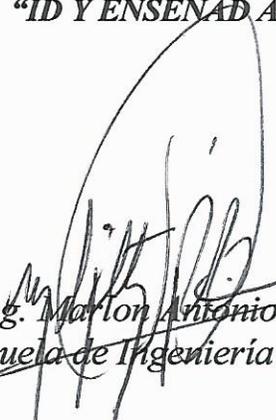
E
S
C
U
E
L
A

D El Director de la Escuela de Ingeniería en Ciencias y
E Sistemas de la Facultad de Ingeniería de la Universidad de
C San Carlos de Guatemala, luego de conocer el dictamen del
I asesor con el visto bueno del revisor y del Licenciado en
E Letras, del trabajo de graduación **“ORGANIZADOR
N ACADÉMICO APLICANDO NUEVAS TECNOLOGÍAS
C MÓVILES, CON PARADIGMA MULTIPLATAFORMA Y
I REDES SOCIALES”**, realizado por los estudiantes LUIS
A FERNANDO LEAL HERNÁNDEZ y MARVIN FRANCISCO
S CASTILLO BARRERA, aprueba el presente trabajo y solicita
la autorización del mismo.

Y

“ID Y ENSEÑAD A TODOS”

S
I
S
T
E
M
A
S


Ing. Marlon Antonio Pérez Türk
Director, Escuela de Ingeniería en Ciencias y Sistemas



Guatemala, 10 de marzo 2015



El Decano de la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala, luego de conocer la aprobación por parte del Director de la Escuela de Ingeniería en Ciencias y Sistemas, al Trabajo de Graduación titulado: **ORGANIZADOR ACADÉMICO APLICANDO NUEVAS TECNOLOGÍAS MÓVILES, CON PARADIGMA MULTIPLATAFORMA Y REDES SOCIALES**, presentado por los estudiantes universitarios: **Luis Fernando Leal Hernández y Marvin Francisco Castillo Barrera**, y después de haber culminado las revisiones previas bajo la responsabilidad de las instancias correspondientes, autoriza la impresión del mismo.

IMPRÍMASE:

Ing. Murphy Olympo Paiz Recinos
Decano

Guatemala, 16 de marzo de 2015

/gdech



ACTO QUE DEDICO A:

Mi madre

Aracely Hernández. Por su amor y apoyo en todos los aspectos de mi vida. Por creer en mí y sacarme adelante con mucho esfuerzo.

La memoria de mi padre

Luis Felipe Leal. Por enseñarme a luchar y no rendirme, y por brindarme su amor incondicional.

Fausto Argueta

Por brindarme su cariño, ejemplo y apoyo sin el cual esto no sería posible.

Mi familia

Por ofrecerme su cariño y siempre estar ahí.

Luis Fernando Leal Hernández

ACTO QUE DEDICO A:

- Mi padre** José Castillo, quien siempre me ha apoyado sin importar las circunstancias.
- Mi madre** María Esperanza Barrera, quien me motivó a nunca darme por vencido.
- Mis hermanos** Juan Carlos y Elsa Castillo, quienes siempre han estado de mi lado.
- Mi familia y amigos** Porque sin su apoyo, nada de esto hubiese sido posible.

Marvin Francisco Castillo Barrera

AGRADECIMIENTOS A:

**Universidad de
San Carlos de
Guatemala**

Por ser mi casa de estudio, mi segunda casa.

Facultad de Ingeniería

Por brindarme el conocimiento y gratos recuerdos de tan bella y gratificante profesión.

**Mis amigos de la
Facultad**

Por compartir tantas experiencias, metas, esfuerzos, sacrificios, desvelos y triunfos a lo largo de la carrera.

**Mi compañero en este
trabajo de graduación**

Marvin Castillo, por las arduas horas invertidas y la dedicación, lo cual hoy nos lleva a concluir este proyecto.

Mi madre

Por brindarme la oportunidad de convertirme en un profesional, por su esfuerzo y sacrificio incondicional.

Luis Fernando Leal Hernández

AGRADECIMIENTOS A:

**Universidad de
San Carlos de
Guatemala**

Al brindarme la oportunidad de crecer personal y académicamente.

Facultad de Ingeniería

Por darme los conocimientos y amistades que me hacen hoy en día.

**Mis amigos de la
Facultad**

Porque sin su apoyo, estos años hubiesen sido más pesados y vacíos.

**Mi compañero en
este trabajo de
graduación**

Luis Leal, porque sin su amistad, perseverancia y constancia este proyecto no fuera posible.

Mis padres

Porque ellos son quienes me han permitido llegar donde estoy.

Marvin Francisco Castillo Barrera

ÍNDICE GENERAL

ÍNDICE DE ILUSTRACIONES.....	VII
GLOSARIO	XI
RESUMEN.....	XV
OBJETIVOS.....	XVII
INTRODUCCIÓN	XIX
1. DESARROLLO MÓVIL MULTIPLATAFORMA.....	1
1.1. Antecedentes.....	1
1.2. Actualidad del desarrollo de aplicaciones móviles.....	4
1.2.1. Situación actual en Guatemala	9
1.3. Importancia del desarrollo móvil multiplataforma.....	12
1.3.1. Desarrollo móvil multiplataforma en Guatemala	13
1.4. Aplicaciones nativas <i>versus</i> aplicaciones web	13
1.4.1. Aplicación móvil nativa	14
1.4.2. Aplicación móvil web	14
1.4.3. Aplicación móvil híbrida	15
1.4.4. Conocimiento de la diferencia entre enfoques en Guatemala	15
1.5. Herramientas de desarrollo móvil multiplataforma y <i>BenchMarking</i>	16
1.5.1. Comparativa de conocimiento en existencia de herramientas multiplataforma	17
1.5.2. Comparativa de uso de herramientas multiplataforma	19

1.5.3.	Comparativa de plataformas destino más comunes.....	20
1.5.4.	Comparativa de enfoque de aplicaciones destino ...	22
1.5.5.	Comparativa de aplicaciones preinstaladas accesibles.....	24
1.5.6.	Comparativa de calidad y rendimiento de aplicaciones creadas con herramientas multiplataforma.....	25
1.5.7.	Selección y recomendación.....	26
2.	ESTUDIO DE LA TECNOLOGÍA Y SU IMPACTO EN GUATEMALA.....	27
2.1.	Identificación de la teoría que soporta la investigación	27
2.2.	Teoría de investigación y relación con la tecnología escogida.....	31
3.	IDENTIFICACIÓN DEL PROBLEMA Y SOLUCIÓN QUE LA APLICACIÓN REALIZARÁ	33
3.1.	Antecedentes	33
3.2.	Mercado objetivo.....	34
3.3.	<i>Benchmark</i> de la aplicación	37
3.3.1.	iStudiez Pro.....	38
3.3.2.	Class Buddy.....	40
4.	DISEÑO DE LA APLICACIÓN BAJO LA NECESIDAD IDENTIFICADA.....	43
4.1.	Análisis a través de casos de uso	43
4.2.	Diseño y modelado de la arquitectura de la solución	45
4.2.1.	Arquitectura: vista estática y funcional	46
4.2.2.	Arquitectura: vista dinámica	47

4.3.	Diseño del modelo de datos	48
4.3.1.	Modelo de datos en servidor.....	49
4.3.2.	Modelo de datos locales	51
4.4.	Modelado UML adicional	52
4.4.1.	Diagrama de clases	53
4.4.1.1.	Capa de datos	53
4.4.1.2.	Capa de lógica de negocio	53
4.4.1.3.	Capa de presentación al usuario	54
4.4.2.	Diagrama de secuencias	56
4.5.	Prototipo	57
4.5.1.	Pantalla inicial.....	57
4.5.2.	Registro de nuevo usuario.....	58
4.5.3.	Pantalla principal	58
4.5.4.	Actividades de hoy.....	60
4.5.5.	Calendario	61
4.5.6.	Tareas	62
4.6.	Validaciones	64
4.7.	Diseño intuitivo y usabilidad	65
5.	DOCUMENTACIÓN Y TUTORIAL DE PROGRAMACIÓN DE LA APLICACIÓN.....	67
5.1.	Requisitos.....	68
5.1.1.	Herramientas	68
5.1.1.1.	Repositorio de control de versiones dentro de Google Code	68
5.1.1.2.	Cliente Subversion.....	69
5.1.1.3.	Plugin FTP	70
5.1.1.4.	Dispositivos de prueba	71
5.1.1.5.	Codename One Build Server	72

	5.1.1.6.	Lector de códigos QR.....	74
5.1.2.		Hardware.....	74
	5.1.2.1.	Requisitos de instalación y ejecución de la aplicación(clientes)	75
		5.1.2.1.1.	Requisitos para iOS 75
		5.1.2.1.2.	Requisitos para Android..... 75
	5.1.2.2.	Requisitos para desarrollo e implementación (servidor).	77
5.1.3.		Software	78
	5.1.3.1.	Requisitos de instalación y ejecución de la aplicación(clientes)	78
	5.1.3.2.	Requisitos para desarrollo e implementación.	79
		5.1.3.2.1.	Software del servidor 79
		5.1.3.2.2.	Software de desarrollo .. 80
5.2.		Tutorial de desarrollo y referencias	84
	5.2.1.	Implementación de modelo de datos en servidor	84
	5.2.2.	Poblado de catálogos.....	86
	5.2.3.	Desarrollo de procedimientos almacenados.....	88
	5.2.4.	Desarrollo de servicios web.....	90
		5.2.4.1.	Conexión a la base de datos 91
		5.2.4.2.	Servicios web de manipulación de datos 91
	5.2.5.	Desarrollo de objetos persistentes	95
	5.2.6.	Desarrollo de capa de presentación y lógica de la aplicación.....	97
		5.2.6.1.	Capa de presentación 98
		5.2.6.2.	Desarrollo de lógica de la aplicación .. 102

5.2.7.	Desarrollo de notificaciones <i>Push</i>	104
5.2.8.	Desarrollo de interacción con Twitter.....	107
5.2.9.	Pruebas e instalación en dispositivos reales	109
5.2.9.1.	Pruebas preliminares.....	110
5.2.9.2.	Instalación y pruebas en dispositivos reales.....	111
5.3.	Consideraciones de implementación	113
CONCLUSIONES		115
RECOMENDACIONES		117
BIBLIOGRAFÍA.....		119
APÉNDICES		121

ÍNDICE DE ILUSTRACIONES

FIGURAS

1.	Venta anual de dispositivos,1990-2012.....	4
2.	Evolución del mercado de plataformas móviles (2009-2013).....	5
3.	Personal de IT con conocimiento en desarrollo móvil	9
4.	Experiencia según plataforma.....	10
5.	Motivo de involucramiento en desarrollo móvil.....	10
6.	Experiencia entre los estudiantes según año de ingreso y semestre...	11
7.	Estadística de desarrollo móvil multiplataforma en Guatemala.....	13
8.	Conocimiento de diferencia entre aplicación web y nativa	15
9.	Conocimiento de herramientas multiplataforma entre los desarrolladores	18
10.	Conocimiento de herramientas multiplataforma entre los desarrolladores guatemaltecos	18
11.	Porcentaje de uso de herramientas multiplataforma	19
12.	Porcentaje de uso de herramientas multiplataforma en Guatemala.....	20
13.	Plataforma destino al usar herramientas multiplataforma	20
14.	Soporte de plataformas por parte de herramientas multiplataforma	21
15.	Enfoque de la industria de herramientas multiplataforma	22
16.	Complejidad de las herramientas: tiempo de aprendizaje y tiempo promedio de desarrollo de aplicaciones.....	23
17.	Número de aplicaciones preinstaladas accesibles desde herramientas multiplataforma.....	24
18.	Calificación de calidad de aplicaciones multiplataforma.....	25

19.	Calificación de rendimiento por categoría de herramientas multiplataforma	26
20.	Modelo de aceptación de la tecnología.....	28
21.	Teoría de la dominación de la tecnología	29
22.	Telefonía móvil en Guatemala y su crecimiento	34
23.	Población estudiantil universitaria en Guatemala	36
24.	Pantalla inicial de iStudiez Pro.....	38
25.	Pantalla principal de Class Buddy.....	41
26.	Diagrama de casos de uso	44
27.	Diagrama de arquitectura	48
28.	Diseño de base de datos servidor.....	50
29.	Diseño de modelo de datos local.....	52
30.	Diagrama de clases	55
31.	Diagrama de secuencias para Compartir Item.....	56
32.	Pantalla inicial de la aplicación	57
33.	Pantalla principal de la aplicación.....	59
34.	Formulario del calendario de la aplicación.....	61
35.	Formulario de tareas.....	62
36.	Agregar actividades o tareas a un curso.....	64
37.	Configuración de subversión en Netbeans	70
38.	Codename One Build Server	72
39.	Generación de binario nativo con Build Server.....	73
40.	Configuración de registro en IDE	73
41.	Codename One sobre Netbeans	82
42.	Instalación de <i>plugins</i> Codename One y PHP	83
43.	Ejemplo de código SQL de creación de la base de datos en el servidor	86
44.	Ejemplo de código SQL de poblado de tablas catálogo.....	87

45.	Ejemplo de procedimiento almacenado de consulta a través de cláusula Select	89
46.	Ejemplo de procedimiento almacenado de inserción a través de cláusula Insert	90
47.	Conexión a través del archivo “db_config.php”	91
48.	Ejemplo de acceso a un parámetro desde servicio web y asignación del valor del parámetro a una variable	92
49.	Ejemplo de servicio web de consulta de información	93
50.	Consumo de servicios web desde el código de la aplicación	94
51.	Ejemplo de objeto persistente	96
52.	Registro de objetos persistentes y ejemplo de manipulación de los mismos	97
53.	Sección de objetos de la aplicación: en esta sección se listan todos los formularios pertenecientes a la aplicación	99
54.	Componentes disponibles: librería de componentes gráficos disponibles para utilización.	100
55.	Propiedades del objeto seleccionado: despliega y permite editar las propiedades del objeto seleccionado.	100
56.	Árbol de componentes: muestra de manera jerárquica la estructura de los objetos pertenecientes al formulario seleccionado.	101
57.	Visualización gráfica: vista previa del resultado final del formulario editado.	101
58.	Ejemplo de lógica de aplicación y eventos de interfaz gráfica a través de la clase StateMachine.	103
59.	Descripción de las notificaciones	104
60.	Llave de servidor para notificaciones <i>push</i>	105
61.	Clase para enviar notificaciones	106
62.	Clase “enviarTweetUsuario”	108

63.	Implementación de publicación de Tweet y envío de mensaje directo a usuarios suscritos.	109
64.	Pruebas en emulador de dispositivos	110
65.	Creación de archivo para instalación en Android.....	112
66.	Opciones de instalación dentro del servidor de Codename One	112

TABLAS

I.	Requisitos por plataforma	79
----	---------------------------------	----

GLOSARIO

Android	<p>Sistema operativo basado en el kernel Linux y que fue desarrollado por Android Inc., empresa que sería apoyada y posteriormente adquirida por Google. Este sistema operativo ha sido adoptado en múltiples dispositivos tales como televisores y relojes inteligentes pero su mayor enfoque está en los dispositivos móviles, mercado que domina en la actualidad.</p>
API	<p>Siglas en inglés para Application Programming Interface consiste en un set de bloques de programación predefinidos tales como protocolos, rutinas y clases que brindan a un programador bloques de construcción para realizar una tarea en particular, el programador se encarga de integrar y dar una lógica de interacción a estos componentes sin preocuparse su definición y construcción interna.</p>
Aplicación móvil	<p>Programa computacional destinado a ejecutarse en un dispositivo móvil tal como un Smartphone o una tablet y que corre sobre un sistema operativo móvil y es administrado por este. Por lo general se encuentran a través de plataformas de distribución tales como el App Store de Apple o el Market Place de Google. Durante el proceso de desarrollo se tiene</p>

especial consideración en la optimización de recursos, debido a la limitante que impone el que un dispositivo móvil goce de menos recursos que una computadora personal.

BlackBerry OS

Sistema operativo desarrollado y destinado para ejecutarse sobre dispositivos BlackBerry. Desarrollado por la empresa RIM, este sistema operativo ha sido desarrollado pensando en su uso en el área profesional y de negocios.

FTP

Protocolo (File Transfer Protocol) de red estandarizado y diseñado para la transferencia de archivos desde una computadora a otra a través de una red.

IDE

Herramienta de software utilizada por programadores para facilitar el desarrollo de aplicaciones. Consiste en un entorno integrado (IDE significa Integrated Development Environment) que reúne todas las herramientas utilizadas durante el desarrollo de software tales como: editor de código fuente, compilador y herramientas de automatización de compilación, sugerencias de autocompletado de código, *debugger* y otras herramientas que de otro modo deberían ser accedidas de manera externa e independientes entre sí.

IOS	Sistema operativo móvil creado específicamente para dispositivos móviles fabricados por Apple Inc. tales como iPhone, iPad, iPod Touch. Inicialmente conocido como iPhone Operating System en la actualidad es utilizado en múltiples dispositivos móviles Apple y no solamente en teléfonos.
Kernel (núcleo)	Pieza de software de un sistema operativo, que sirve como intermediario entre las aplicaciones y el hardware o algunos recursos adicionales.
SDK	Siglas en inglés para Software Development Kit, como su nombre lo indica es un paquete de herramientas que permiten a un programador desarrollar aplicaciones de software para alguna plataforma en específico.
Sistema Operativo Móvil	Sistema operativo destinado a controlar y ejecutarse sobre un dispositivo móvil tal como un Smartphone o una tablet. Por lo general son mucho más sencillos y ligeros que los sistemas operativos de computadoras personales convencionales.
Smartphone	Teléfono móvil que brinda la funcionalidad de un teléfono común (realizar llamadas, mensajes de texto, entre otros) pero que ofrece mayor poder de cómputo y características avanzadas adicionales tanto de software como de hardware.

Tablet

Computadora móvil cuyo tamaño es mayor al de un Smartphone pero menor al de las computadoras portátiles comunes, tienen la característica de tener la pantalla, batería, circuitería y entrada (teclado y mouse por medio de pantalla táctil) integrados en una única unidad lo cual brinda alta portabilidad.

Web Service

Método de comunicación e intercambio de información entre dispositivos heterogéneos a través de una red (comúnmente internet) a través de protocolos estandarizados.

Windows Phone

Sistema operativo móvil enfocado al mercado de consumo y no al mercado empresarial como su sucesor Windows Mobile. Este ha sido desarrollado por Microsoft, está basado en el núcleo (kernel) Windows Embedded e integra múltiples servicios comunes en sistemas operativos Windows para PC tales como Skype, Windows Media Player y Outlook.

RESUMEN

La mayoría de desarrolladores de aplicaciones móviles se enfocan y especializan en una única plataforma, debido a las diferencias entre las plataformas existentes, así como las técnicas y herramientas de desarrollo de cada una. En el presente documento se plantea una alternativa a estas dificultades y por medio de la práctica se ofrece a la población estudiantil una herramienta tecnológica que sirva para apoyar sus actividades académicas y compartición de recursos estudiantiles.

El primer capítulo, describe la parte investigativa del desarrollo móvil multiplataforma, se describen los antecedentes, la actualidad de la materia y su tendencia a futuros. Se presenta la importancia creciente del desarrollo móvil utilizando enfoque multiplataforma (a diferencia del enfoque específico), posteriormente se describen lo que son las web-Apps y las aplicaciones nativas, para finalizar con el *benchmarking* o comparativas entre los enfoques, técnicas y herramientas presentadas en el capítulo; así como un análisis y sondeo entre la comunidad de desarrolladores de Guatemala, para finalmente presentar una propuesta de herramienta de desarrollo para móviles.

En los siguientes capítulos el enfoque del trabajo se torna en realizar la construcción de una aplicación móvil, utilizando las herramientas y resultado del capítulo inicial, de modo que la aplicación construida sea un activo de referencia y guía para estudiantes y profesionales de tecnología y que a la vez, la herramienta generada sea de beneficio para toda la población estudiantil, por medio del ofrecimiento de facilidades para la organización de actividades y tareas académicas y la compartición de recursos entre la red estudiantil.

OBJETIVOS

General

Proporcionar un estudio detallado acerca del desarrollo móvil con un enfoque multiplataforma nativo, las opciones y herramientas disponibles, comparativas de las mismas y a la vez ejemplificar la investigación por medio de la construcción de una aplicación móvil estudiantil.

Específicos

1. Brindar una comparativa entre el desarrollo móvil multiplataforma web y nativo así como las diferencias, ventajas y desventajas de cada enfoque, y su situación en el mercado y comunidad de desarrollo móvil.
2. Ofrecer a los estudiantes de Ingeniería en Ciencias y Sistemas y público en general, una alternativa viable al desarrollo de software móvil nativo destinado a múltiples plataformas.
3. Brindar una aplicación móvil multiplataforma funcional cuyo código y técnicas de construcción estén disponible para referencia, estudio y ejemplo para futuras generaciones de profesionales de la informática.
4. Construir una aplicación móvil que ejemplifique la parte investigativa del trabajo y que a la vez sea de beneficio para la población estudiantil, por medio de: organización de actividades y entregables académicos y compartición de recursos educativos.

INTRODUCCIÓN

En la actualidad el mercado de nuevas tecnologías móviles crece aceleradamente y prevalece como una tendencia en auge, dentro del área de tecnologías de la información. Este crecimiento se ha visto acompañado a su vez por el incremento de la presencia en redes sociales, tanto de personas individuales como de entidades comerciales, gubernamentales y educativas.

La falta de estandarización y diferencias entre las múltiples plataformas de tecnologías móviles existentes dificultan el proceso de creación y desarrollo de proyectos y productos orientados a las nuevas tecnologías ya mencionadas, ya que estas diferencias provocan que se haga necesario duplicar o multiplicar esfuerzos, para alcanzar audiencias de las múltiples plataformas existentes.

Es importante que los profesionales egresados de Ingeniería en Ciencias y Sistemas de la USAC (y otras casas de estudios superiores), se encuentren a la vanguardia del desarrollo de tecnologías móviles. Por esta razón el actual trabajo de graduación tiene como objetivo principal introducir al desarrollo de nuevas tecnologías móviles, utilizando un enfoque “code once run everywhere” (generando aplicaciones nativas), por medio del cual los desarrolladores pueden superar el gran obstáculo que representan las diferencias e incompatibilidad entre plataformas móviles.

A su vez la investigación se ejemplificará, por medio de la implementación de una aplicación de control de actividades académicas que brinde a la población estudiantil, una herramienta capaz de organizar y compartir sus contenidos, horarios, entregas, exámenes, documentos, imágenes, notas de

texto y voz y obtener alertas y recordatorios por medio del calendario y redes sociales.

1. DESARROLLO MÓVIL MULTIPLATAFORMA

El desarrollo de aplicaciones y tecnologías móviles se encuentra en la actualidad en auge con alto grado de crecimiento, esto debido a varios factores, por ejemplo la creciente tendencia y necesidad de la sociedad a realizar operaciones comerciales, educacionales y recreativas de manera rápida, eficiente, en cualquier lugar y en todo momento, todo esto sin necesidad de acceder a una computadora. Además de este ejemplo también se puede mencionar al aumento de la presencia en redes sociales, tanto con fines recreativos, como económicos, educativos e informativos.

Una de las grandes limitantes y posiblemente el mayor problema para los desarrolladores de aplicaciones móviles, es la diferencia e incompatibilidad entre las diferentes plataformas móviles presentes en el mercado (Android, iOS, RIM, Windows Mobile y otros), lo cual dificulta y multiplica el tiempo y esfuerzo necesario para llevar una aplicación móvil a todas las plataformas móviles mencionadas y a los usuarios de estas.

En el primer capítulo de este documento, se trata el tema del desarrollo móvil multiplataforma, un enfoque que busca erradicar el problema mencionado.

1.1. Antecedentes

El 3 de abril del 2013 se celebraron 40 años del lanzamiento del primer teléfono móvil, hecho acreditado a Motorola, el campo de la telefonía móvil ha evolucionado enormemente desde entonces.

A finales del siglo XX los teléfonos celulares empezaron a incluir aplicaciones móviles sencillas tales como editores de ring tones polifónicos, calculadoras y relojes para distintos horarios.

El inicio del nuevo milenio trajo consigo una rápida evolución en el área de las aplicaciones móviles, esto debido a que la incursión de sistemas operativos para móviles (iOS, Android, RIM) abrió la puerta al desarrollo de software por parte de terceros y no solamente el fabricante del dispositivo. Gracias a esto la diversidad de actividades que un usuario puede realizar en un dispositivo móvil se vio enriquecida y ya no estaba limitada a realizar llamadas y enviar mensajes de texto.

Las primeras aplicaciones móviles reconocidas fueron lanzadas para dispositivos ejecutando el sistema operativo EPOC, principalmente PDAs en la década de los 90s, inicialmente para dispositivos de 16 bits. Posteriormente los dispositivos evolucionaron a los 32 bits y 2 megabytes de memoria de acceso aleatorio (RAM), lo cual permitió ejecutar aplicaciones más poderosas y la instalación de aplicaciones adicionales por medio de paquetes de software o descargas.

El sistema operativo EPOC fue desarrollado en el lenguaje OPL (Open Programming Language), lo cual permitía al usuario con suficientes conocimientos técnicos, crear sus propias aplicaciones.

Entre mediados y finales de los 90 "Palm OS" emergía como fuerte rival de EPOC logrando un enorme éxito comercial, gracias a sus precios más económicos, pantalla táctil y una gran variedad de aplicaciones y la posibilidad de desarrollar nuevas por medio de C/C++.

Posteriormente surgió Java ME (Java Micro Edition) que nació como una plataforma Java diseñada para dispositivos embebidos y de recursos limitados y se volvió el rey indiscutible en el desarrollo de aplicaciones móviles.

Las implementaciones iniciales de Java ME se dieron para los sistemas operativos móviles Symbian y Bada y otros móviles, ejecutando una máquina virtual Java (JVM). La presencia de Bada en el mercado de tecnología móvil ha ido en receso hasta llegar a presentar una porción no significativa de la población (menos del 1 por ciento), Symbian mantuvo un alto porcentaje de presencia en el mercado hasta el 2011, año en el que tuvo un 25.98 por ciento de presencia pero desde entonces ha disminuido hasta un 1,11 por ciento al finalizar 2012.

El desarrollo de aplicaciones para Symbian no se limitaba a Java ME, la forma más habitual de desarrollarlas era por medio de C++ y QT de Nokia pero podía realizarse también a través de Python, Flash Light, Ruby, entre otros.

La reducción en la presencia de Symbian en el mercado se ha visto marcada gracias al creciente mercado de plataformas como Android e IOS, lo cual sitúa en la situación actual del mercado de aplicaciones móviles, tema de la siguiente sección. Se pueden resumir los antecedentes más importantes de las plataformas y desarrollo móvil en el siguiente listado:

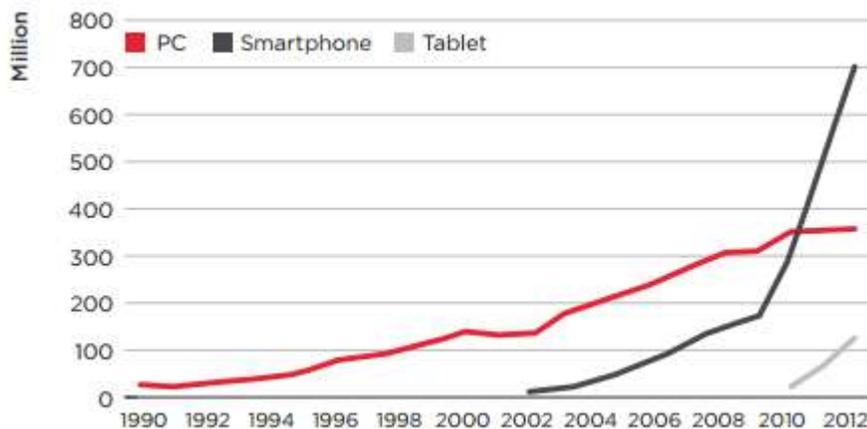
- EPOC, desarrollo en OPL
- Palm OS, desarrollo en C++
- Symbian y Bada, desarrollo en Java ME, C/C++,QT y otros
- Java ME, utilizado para otras plataformas móviles menores

1.2. Actualidad del desarrollo de aplicaciones móviles

El lanzamiento del primer dispositivo Android y el primer iPhone (en 2008 y 2007 respectivamente), marcó un antes y un después en el mercado de teléfonos inteligentes y desarrollo de aplicaciones móviles.

Los beneficios ofrecidos por los dispositivos móviles con alto poder de cómputo, han provocado que la tendencia tecnológica actual haya evolucionado a una era dominada por la movilidad y el consumo de tecnología computacional portátil de propósitos generales y diversos, esto ha marcado un incremento significativo en la venta de tabletas y teléfonos inteligentes en los últimos años. Este comportamiento puede ser analizado en la siguiente figura.

Figura 1. **Venta anual de dispositivos, 1990-2012**

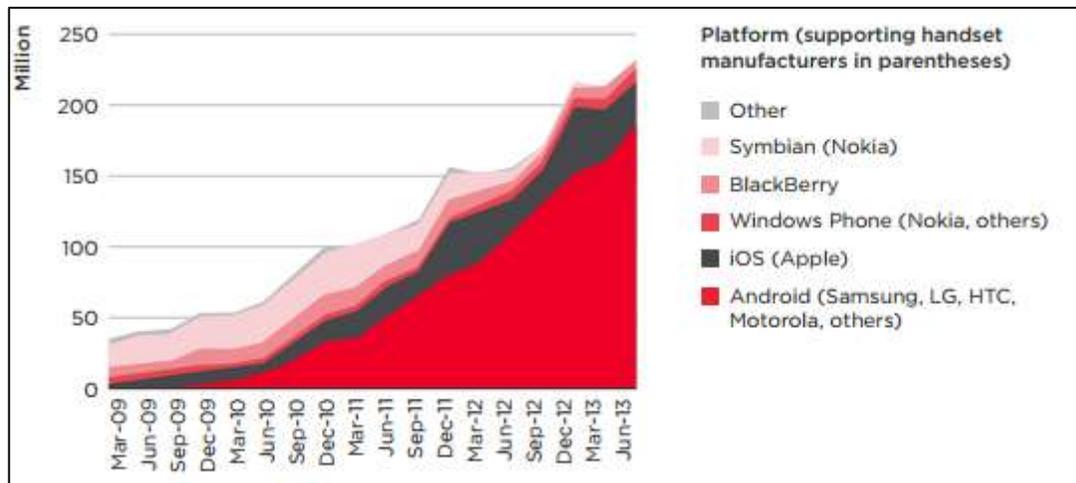


Fuente: GSMA Intelligence (Platform Wars). p. 56.

A partir de este punto, la presencia de otras plataformas como Symbian y Black Berry ha sufrido un declive significativo al grado de llegar a hablarse de un duopolio entre Android y iOS, plataformas que comparten más del 90 por ciento del mercado en la actualidad, siendo Android la de mayor presencia de

las dos. Con un índice bastante menor se encuentran Black Berry y Windows Phone compartiendo un 6 por ciento del mercado entre ambas. Este comportamiento puede ser analizado en la siguiente gráfica, en la cual es posible observar el comportamiento del mercado en los últimos 5 años (2009-2013).

Figura 2. **Evolución del mercado de plataformas móviles (2009-2013)**



Fuente: GSMA Intelligence (Platform Wars). p. 78.

El incremento en el comercio de dispositivos móviles inteligentes trajo consigo la creación de mercados de aplicaciones en los cuales los usuarios tienen acceso a una amplia variedad de aplicaciones, con fines diversos. Así mismo esto ha dado a desarrolladores independientes, estudios pequeños y también grandes, la oportunidad de crear productos que puedan llegar a un conjunto gigantesco de posibles usuarios y clientes.

Esta nueva tendencia tecnológica, ha provocado una creciente demanda de personal técnico especializado en la materia de desarrollo de tecnologías

móviles y cada día es más común que las empresas busquen servicios de desarrollo de software móvil a la medida y personalizado.

En la actualidad la construcción de aplicaciones se realiza mayormente por medio de herramientas y kits de desarrollo oficiales, cuyo objetivo es crear software para una única plataforma destino. A continuación se describen las herramientas de desarrollo actuales, para cada una de las 4 plataformas con presencia significativa en el presente mercado.

- Android: el método oficial y más común de desarrollo de aplicaciones Android consiste en el uso del lenguaje Java y el SDK Android, pero existen también otras herramientas disponibles, mayoritariamente de terceros que por lo general están enfocadas a la creación de aplicaciones sencillas y de baja complejidad, para personas con poco conocimiento en programación, algunos ejemplo son:
 - Basic4Android
 - AndroWish
 - App Inventor for Android

El kit de desarrollo oficial incluye todas las herramientas necesarias tales como *debugger*, librerías y APIs; código fuente y tutoriales, documentación y un emulador utilizado para realizar pruebas de las aplicaciones desarrolladas. Aunque es posible desarrollar aplicaciones Android utilizando solamente un editor de texto, una línea de comandos y el SDK Android, el desarrollo comúnmente se realiza utilizando un IDE. El IDE oficial es Eclipse usando el *plugin* “ADT” (Android Development Tools), pero es posible utilizar otros IDEs tales como Netbeans a través de un plugin dedicado o bien IntelliJ IDEA. En mayo de 2013 Google

anunció el lanzamiento del IDE Android Studio, dedicado específicamente al desarrollo para Android y basado en software del actual IntelliJ IDEA. El sitio web oficial con toda la información y tutoriales para desarrollo Android es: <http://developer.android.com/index.html>. El kit de desarrollo de Android puede ser instalado en sistemas operativos Windows, Linux y Mac OSX.

- IOS: puesto que Apple impone un ecosistema bastante cerrado en todos sus productos, el desarrollo de aplicaciones para iOS es realizado con el kit de desarrollo oficial de Apple, utilizando el IDE Xcode, el cual en la actualidad solo puede ser ejecutado sobre computadoras Apple, lo cual limita el desarrollo a empresas y desarrolladores independientes que cuentan con hardware y software de Apple. La programación se realiza utilizando el lenguaje Objective-C y el paradigma Modelo-Vista-Controlador, pero también algunos elementos pueden ser escritos en C/C++. El sitio web oficial para desarrolladores IOS es: <https://developer.apple.com>.
- Windows Phone: el desarrollo de Windows Phone es llevado a cabo utilizando la conocida herramienta de Microsoft Visual Studio, a través de la instalación de un SDK específico para la programación destinada a la plataforma móvil, esto permite aprovechar el conocimiento y experiencia en el desarrollo de aplicaciones Microsoft, para acortar la curva de aprendizaje de la programación móvil. El sitio web oficial para el desarrollo móvil de Windows Phone es: <http://dev.windows.com/es-es>.
- BlackBerry: el desarrollo de aplicaciones para BB cuenta en la actualidad con una demanda menor a la existente algunos años atrás, esto en su mayor parte por el dominio de Android y iOS en el mercado. El presente

mercado de aplicaciones BlackBerry se encuentra dominado por aplicaciones ejecutivas y empresariales. A diferencia de iOS, el ecosistema de desarrollo para BlackBerry es bastante abierto y diverso, similar al ecosistema para Android, pero cabe remarcar que a pesar de que Android cuenta con diversas opciones de desarrollo, la mayoría son alternativas de terceros, lo cual marca una diferencia con BlackBerry ya que la compañía ofrece a los desarrolladores diferentes alternativas soportadas oficialmente, algunas de estas alternativas son:

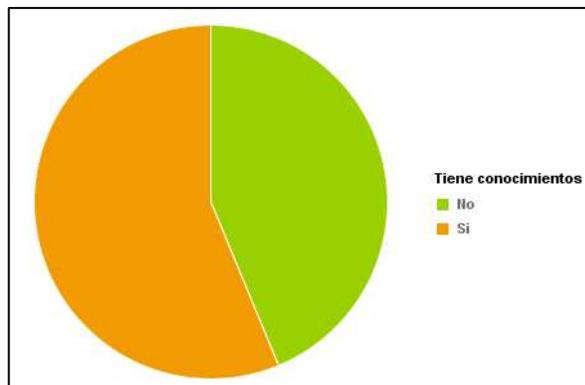
- Kit de desarrollo nativo: el kit de desarrollo nativo, es el método tradicional para desarrollo de aplicaciones BlackBerry, inicialmente llevado a cabo a través del lenguaje Java y herramientas oficiales de BlackBerry tal como un IDE dedicado o *plugins* para IDEs como Netbeans.
- HTML 5: BlackBerry cuenta con soporte oficial para el desarrollo de aplicaciones a través de HTML5 y tecnologías web, lo cual permite la creación de software móvil apoyándose de tecnología como Apache Cordova y WebWorks, que permite escribir aplicaciones web destinadas a ejecutarse en dispositivos móviles.
- Android: las herramientas BlackBerry para desarrollo en Android, permiten empaquetar aplicaciones Android existentes para ser ejecutadas en dispositivo BlackBerry, esto es llevado a cabo a través *plugins* existentes para Android Studio, Eclipse y herramientas en línea de comandos.

El sitio web oficial para desarrolladores BlackBerry es el siguiente:
<http://developer.blackberry.com>.

1.2.1. Situación actual en Guatemala

Actualmente en Guatemala el crecimiento en la demanda de tecnología móvil y la presencia de teléfonos inteligentes y tabletas son factores que incrementan la necesidad de personal con conocimiento en el desarrollo de aplicaciones, lo cual ha llevado a entidades educativas tales como universidades y a profesionales de IT a incursionar en la materia. La encuesta dirigida a estudiantes y profesionales de IT, demuestra que un 56,25 por ciento de los encuestados tiene conocimientos de desarrollo para dispositivos móviles, se espera que debido a la demanda este número crezca en los próximos meses.

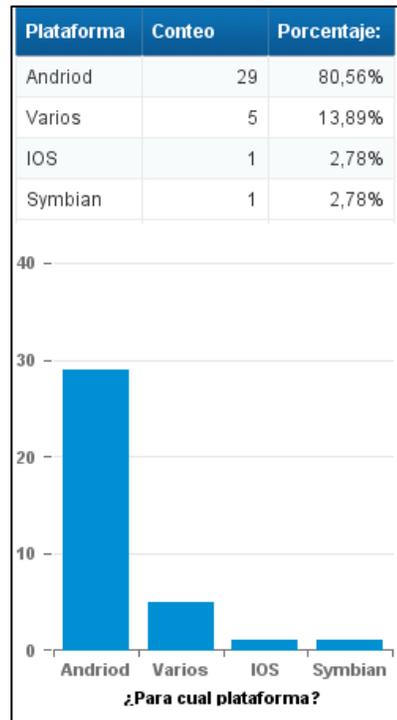
Figura 3. Personal de IT con conocimiento en desarrollo móvil



Fuente: elaboración propia.

Del grupo de encuestados con conocimiento en desarrollo móvil, la mayoría tiene experiencia en desarrollo exclusivamente para la plataforma Android (80,56 %), un pequeño porcentaje ha desarrollado para iOS o Symbian (2,78 %) y un 13 por ciento del grupo, ha desarrollado para múltiples plataformas (13,89 %).

Figura 4. **Experiencia según plataforma**



Fuente: elaboración propia a través de encuesta.

Es posible observar que la mayoría se ha involucrado en el ámbito del desarrollo móvil a través de algún proyecto universitario.

Figura 5. **Motivo de involucramiento en desarrollo móvil**

Motivo	Porcentaje: %
Proyecto Universitario	72,22
Autoaprendizaje	19,44
Trabajo	8,33

Fuente: elaboración propia.

El mayor porcentaje de personas encuestadas con conocimiento en la materia, se encuentra entre los estudiantes ingresados en 2007 y 2008 que cuentan en la actualidad con pensum cerrado, seguido de estudiantes ingresados en distintos años pero que cursan actualmente cursos de los últimos semestres.

Figura 6. **Experiencia entre los estudiantes según año de ingreso y semestre**

Año ingreso	Semestre actual	Porcentaje: %
2007	Pensum cerrado- Posgrado	16,67
2008	Pensum cerrado- Posgrado	11,11
2007	8	8,33
2010	10	8,33
2006	10	5,56
2006	Pénsum cerrado- Posgrado	5,56
2007	10	5,56
2008	8	5,56
2009	Pénsum cerrado- Posgrado	5,56
1998	10	2,78
2006	7	2,78
2009	8	2,78
2010	5	2,78
2010	7	2,78
2010	8	2,78
2010	9	2,78
2012	10	2,78
2013	3	2,78
2013	4	2,78

Fuente: elaboración propia a través de encuesta.

1.3. Importancia del desarrollo móvil multiplataforma

Es evidente que en el área de tecnologías de información, el desarrollo de aplicaciones móviles es en la actualidad un mercado de creciente demanda y de importancia relevante. Los desarrolladores de software, tanto individuales como empresas deben estar a la vanguardia tecnológica que permita satisfacer las necesidades de software y tecnologías móviles.

Lamentablemente el panorama actual en la materia dificulta el despliegue de las tecnologías desarrolladas a los usuarios de las distintas plataformas móviles existentes, debido a que el proceso de desarrollo para las plataformas existentes es muy diferente entre estas; es evidente que existen diferencias a nivel de arquitectura, herramientas, técnicas, paradigmas y lenguajes de programación entre cada. Esto provoca que el desarrollo móvil se vea segmentado y que actualmente sea poco común encontrar desarrolladores con un conocimiento profundo en las múltiples plataformas, lo cual obliga a las empresas a contratar múltiples programadores según el conjunto de plataformas que desean alcanzar, lo cual multiplica el tiempo, costo y esfuerzo necesario.

Es por esto que se demanda una forma de reducir el tiempo, costo y esfuerzo del desarrollo móvil, utilizando técnicas y herramientas que permitan desarrollar aplicaciones una única vez, pero que puedan ser desplegadas con pocas modificaciones en las múltiples plataformas existentes, permitiendo a las empresas y desarrolladores alcanzar múltiples audiencias de una manera más rápida y eficiente. A la vez que se minimiza la curva de aprendizaje y la necesidad de contar con múltiples desarrolladores de software móvil, ya que se requiere el entrenamiento y dominio de una única herramienta de desarrollo.

1.3.1. Desarrollo móvil multiplataforma en Guatemala

Actualmente, en Guatemala el personal involucrado en el desarrollo de aplicaciones móviles utiliza en su mayoría las herramientas oficiales de desarrollo ya que no sabe de la existencia, o no tiene conocimiento en el uso de herramientas multiplataforma. Menos de un 14 por ciento de los encuestados conoce alguna herramienta móvil multiplataforma, siendo Phonegap la herramienta más conocida.

Figura 7. Estadística de desarrollo móvil multiplataforma en Guatemala

Metodo de desarrollo	Porcentaje: %
Oficial	86,11
PhoneGap	8,33
Cordova	5,56

Fuente: elaboración propia.

1.4. Aplicaciones nativas *versus* aplicaciones web

Existen 2 tipos de aplicaciones móviles con enfoques y objetivos de ejecución muy diferentes entre sí, cada uno de estos tipos tiene ventajas y desventajas sobre el otro, por lo cual no es posible indicar que un tipo sea mejor que el otro. Estos tipos de aplicación tienen influencia y consideraciones a tomar en cuenta cuando se considera el desarrollo de una aplicación, y con la evidente necesidad actual del desarrollo móvil multiplataforma, se hace necesario conocer la forma de abordar estos tipos de aplicación para las herramientas multiplataforma disponibles. Es importante mencionar que en la

actualidad ha surgido un tercer tipo de aplicación, que en realidad no es un tipo nuevo, sino un híbrido de los dos tipos básicos.

1.4.1. Aplicación móvil nativa

Aplicación móvil diseñada para ser ejecutada de forma específica en una plataforma móvil. Se desarrollan con herramientas (kits de desarrollo, entornos integrados y lenguajes de programación) y técnicas específicas para cada plataforma, lo cual implica una mayor inversión de esfuerzo y tiempo (tanto de aprendizaje como de construcción), si se desea desplegar en múltiples plataformas. La ventaja de las aplicaciones nativas radica en el aprovechamiento y optimización al máximo, de los recursos del dispositivo y características nativas del mismo.

1.4.2. Aplicación móvil web

Conocidas en el área de desarrollo móvil como web-apps, estas son aplicaciones móviles creadas con tecnología de desarrollo web (HTML, CSS, JavaScript) y destinadas a ejecutarse dentro del dispositivo móvil, por medio de un navegador o tecnología intérprete web. Utiliza como capa intermedia un intérprete (navegador web) por lo cual su rendimiento es menor al de una aplicación móvil nativa y no aprovecha del todo los recursos del dispositivo. La ventaja de estas aplicaciones es que su desarrollo es más rápido que el de una aplicación nativa, ya que se basa en tecnología conocida de la industria (HTML, CSS, JavaScript) y permite el despliegue en múltiples plataformas, con una menor inversión de tiempo y esfuerzo.

1.4.3. Aplicación móvil híbrida

Tal como su nombre lo indica, este tipo de aplicaciones combinan características de los 2 tipos ya mencionados. Se desarrollan con tecnología web tal como HTML y JavaScript, pero brindan APIs para acceder a recursos y características nativas del dispositivo tales como la cámara. Esta combinación hace que el despliegue multiplataforma sea más rápido que en un enfoque nativo (aunque más lento que un desarrollo web) y brinda un mayor rendimiento que un desarrollo web (aunque menor rendimiento que un desarrollo totalmente nativo). Estas aplicaciones se colocan en el medio del umbral para los dos enfoques mencionados.

1.4.4. Conocimiento de la diferencia entre enfoques en Guatemala

La encuesta realizada demuestra que actualmente la mayoría de desarrolladores conoce la diferencia entre una aplicación nativa y una aplicación web pero a pesar de esto, más de la mitad sigue sin conocer alguna herramienta utilizada para realizar desarrollo multiplataforma.

Figura 8. **Conocimiento de diferencia entre aplicación web y nativa**

¿Conoce la diferencia?	Porcentaje: %
Si	75,00
No	25,00
Suma:	100,00

Fuente: elaboración propia.

1.5. Herramientas de desarrollo móvil multiplataforma y *BenchMarking*

Actualmente, la alta demanda de software destinado a ejecutarse en dispositivos móviles y la falta de recursos humanos, tiempo y fondos para poder desarrollar aplicaciones nativas para cada plataforma existente, han provocado que se cree un ambiente propicio para la penetración de herramientas, que permiten la creación de aplicaciones destinadas a publicarse en múltiples plataformas, pero lo más común es el uso de estas herramientas para publicar en Android e iOS únicamente.

Existen en la actualidad múltiples herramientas disponibles, creadas por diferentes fabricantes, cada una con fortalezas y debilidades en comparación con las otras. Lamentablemente en Guatemala aún existe una comunidad pequeña de desarrolladores de aplicaciones y de esta comunidad, el conocimiento de herramientas con enfoque multiplataforma es muy limitado.

A nivel mundial, solo el 20 por ciento de la comunidad de desarrolladores utiliza actualmente una herramienta de desarrollo multiplataforma y tienen un alto grado de satisfacción con la misma. Un 63 por ciento de usuarios de herramientas multiplataforma, crea más de la mitad de sus aplicaciones en ellas.

A pesar del alto grado de satisfacción en el uso de herramientas multiplataforma, solo el 5 por ciento de las aplicaciones disponibles en los mercados de aplicaciones han sido desarrolladas con una herramienta multiplataforma, lo cual demuestra el poco conocimiento en las mismas, a pesar de sus diferentes ventajas y el alto grado de satisfacción de sus usuarios.

El rendimiento de las aplicaciones es la debilidad principal de las herramientas multiplataforma. El 50 por ciento de sus usuarios califica el rendimiento de las aplicaciones creadas con herramientas multiplataforma como considerablemente menor, esto debido principalmente a que la mayoría de herramientas multiplataforma genera aplicaciones móviles web, destinadas a ejecutarse sobre un intérprete web que cumple el rol de capa intermedia de ejecución, desperdiciando así los recursos de hardware de los dispositivos móviles.

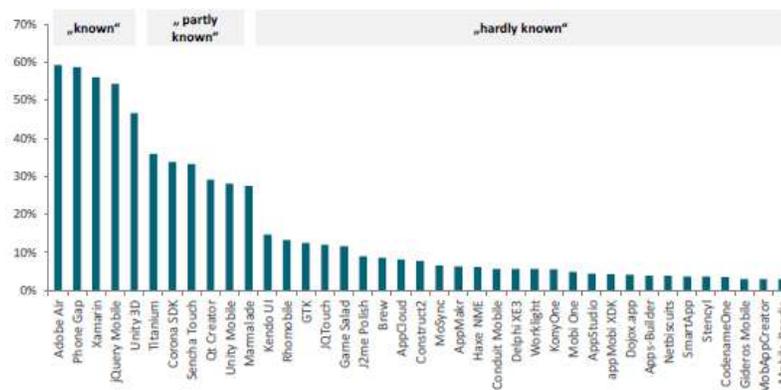
Existen en la actualidad alternativas de desarrollo móvil multiplataforma que generan código nativo, las cuales brindan las ventajas del desarrollo multiplataforma web como reducción de tiempo, esfuerzo y costo pero, obteniendo un rendimiento mayor al de una aplicación web, debido a la generación de código nativo para cada plataforma, lo cual hace que estas herramientas sean posiblemente las mejores opciones al obtener las ventajas de cada enfoque. El 90 por ciento de usuarios de herramientas multiplataforma con generación de código nativo, califican de satisfactoria su experiencia con las mismas, e indican que el rendimiento obtenido es bastante alto y comparable con el producido por herramientas nativas oficiales; el otro 10 por ciento prefiere sacrificar el rendimiento y aprovechamiento de recursos, sobre un diseño visual más detallado y versátil como el obtenido con herramientas de desarrollo web.

1.5.1. Comparativa de conocimiento en existencia de herramientas multiplataforma

En la actualidad, varias empresas han lanzado herramientas de desarrollo multiplataforma, pero el conocimiento de las mismas por parte de la comunidad de desarrolladores sigue siendo muy bajo, actualmente solo 11 herramientas son conocidas por más del 20 por ciento de desarrolladores. En Guatemala,

menos de la mitad de los desarrolladores conoce la existencia de alguna herramienta multiplataforma y un muy bajo porcentaje de estos conoce una herramienta multiplataforma que genera aplicaciones nativas.

Figura 9. **Conocimiento de herramientas multiplataforma entre los desarrolladores**



Fuente: research2guidance, Cross Platform Tool Benchmarking 2011.

Figura 10. **Conocimiento de herramientas multiplataforma entre los desarrolladores guatemaltecos**

¿Conoce alguna herramienta?	Porcentaje: %
No	55,56
Si, genera web-apps	37,04
Si, genera código nativo	7,41
Summa:	100,00

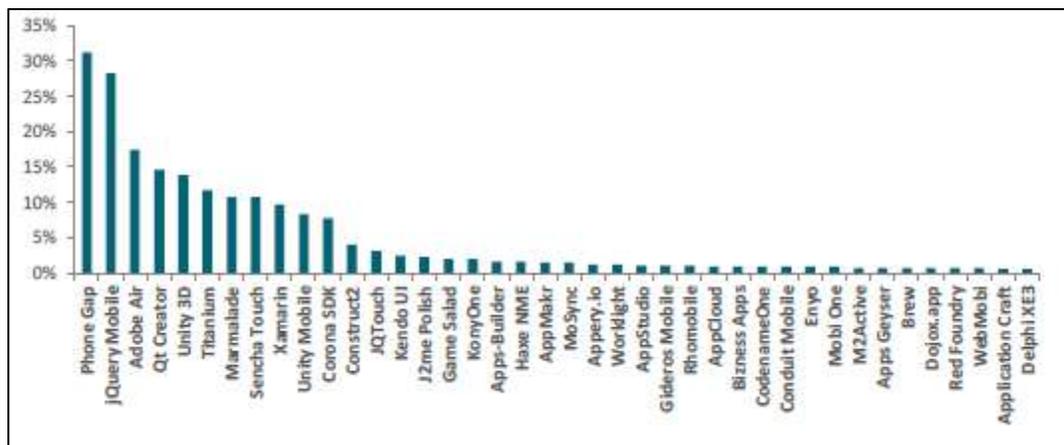
Fuente: elaboración propia.

1.5.2. Comparativa de uso de herramientas multiplataforma

Además del indicador de porcentaje de conocimiento de las herramientas por parte de los desarrolladores, también es importante analizar el porcentaje de uso de las herramientas ofrecidas. Es un hecho establecido que la herramienta de desarrollo multiplataforma más utilizada es Phone Gap, mas esto no significa que sea la mejor.

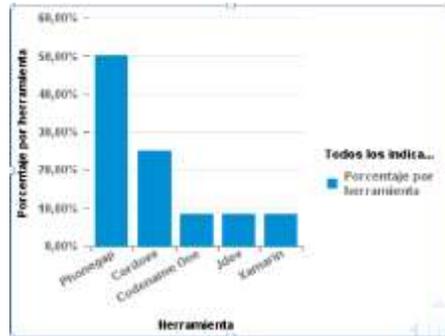
Según la encuesta realizada es posible observar que actualmente en Guatemala, Phonegap es la herramienta más utilizada seguida por Cordova.

Figura 11. **Porcentaje de uso de herramientas multiplataforma**



Fuente: research2guidance, Cross Platform Tool Benchmarking 2013.

Figura 12. **Porcentaje de uso de herramientas multiplataforma en Guatemala**

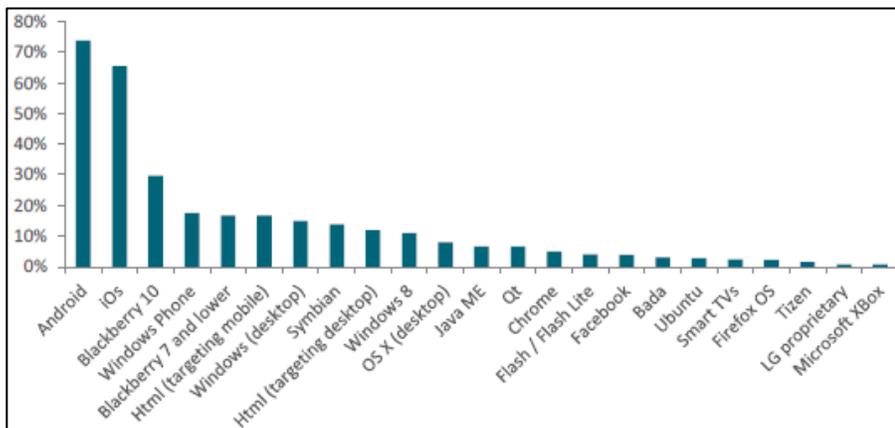


Fuente: elaboración propia.

1.5.3. Comparativa de plataformas destino más comunes

Aun cuando la mayoría de herramientas disponibles en la actualidad, tienen como enfoque principal la creación de aplicaciones móviles, algunas de estas son capaces de generar software destinado a ambientes web o incluso de escritorio, pero a pesar de esto el mayor porcentaje de uso se da para dos plataformas: Android e iOS.

Figura 13. **Plataforma destino al usar herramientas multiplataforma**



Fuente: research2guidance, Cross Platform Tool Benchmarking 2013.

Es posible intuir que debido a que Android e iOS son las plataformas móviles más utilizadas en la actualidad, estas son o serán soportadas por la mayoría de herramientas multiplataforma en un futuro no muy lejano. El tercer lugar corresponde a HTML (y aplicaciones web), ya que un 44 por ciento de herramientas multiplataforma tienen como enfoque la generación de aplicaciones que se apegan a los estándares de la W3C. Windows Phone y Black Berry tienen actualmente una presencia bastante menor a Android e iOS, pero un tercio de herramientas tienen soporte para estas plataformas.

Figura 14. **Soporte de plataformas por parte de herramientas multiplataforma**



Fuente: research2guidance, Cross Platform Tool Benchmarking 2013.

1.5.4. Comparativa de enfoque de aplicaciones destino

En la categorización de las aplicaciones desarrolladas con herramientas multiplataforma, un 35 por ciento de estas son juegos, 22 por ciento aplicaciones utilitarias, 21 por ciento son aplicaciones de negocios, 20 por ciento aplicaciones educativas y 19 por ciento aplicaciones de entretenimiento.

Figura 15. Enfoque de la industria de herramientas multiplataforma

	General	Utilities	Games	Retail	Hotels/ gastronomy	News/ Media	Sports	Medical/ health	IT	Financial	Education	Automotives	Heavy industries	Aerospace/ defense	Government	Enterprise apps
Web app toolkits																
Appery.io																
App Studio																
DaVinci Suite																
Genery																
Kindo UI																
Magnite																
Weaver App																
CP IDEs																
AIR																
Codename One																
Corona SDK																
Embarcadero RAD Studio																
Egels																
Genexus																
12ME patch																
LiveCode																
Marmalade																
MoSync																
PhoneGap																
Smartface App Studio																
TotalCross																
Unity																
V-Play																
WebMobi																
Xamarin																
CP IDEs (Enterprise)																
AppPresence																
AppCondit																
Append																
Giterra																
Service2Media M2Active																
App factories																
AppCaly																
AppMachine																
Combit Mobile																
iGenApps																
Inshive Factory																
MobileAppCreator																
MobileCart																
Mobile Roadie																
Page																
Red Foundry																
SaaSmob SmartApp																
ShoutEm																
Spri Specifics																
ViziApps																

Fuente: research2guidance, Cross Platform Tool Benchmarking 2013.

La complejidad de las herramientas multiplataforma es medida por el tiempo estimado para aprender a usar la herramienta y el tiempo promedio en desarrollar una aplicación general promedio. Obviamente el tiempo necesario para desarrollar una aplicación depende de muchos factores, tales como los requerimientos, alcance y tamaño de la aplicación. El tiempo reportado en la investigación consiste en el tiempo estimado por los fabricantes de herramientas multiplataforma para crear una aplicación sencilla.

Figura 16. **Complejidad de las herramientas: tiempo de aprendizaje y tiempo promedio de desarrollo de aplicaciones**

	Time-to-learn			Average app development time	
	Days	Weeks	Months	Days	Weeks
Web app toolkits					
Appery.io	█			█	
App Studio	█			█	
Blazor Suite		█		█	
Genies					█
Wanda UI	█			█	
Magmito				█	
Waver Apps	█			█	
CP IDEs					
Collabora One	█			█	
Corona SDK					
Embarcadero RAD Studio XE5					
Flora		█			
Genetix			█		
ONE pollin					█
LiveCode		█		█	
Monstade		█			█
Neclys				█	
Smartface App Studio	█			█	
TotalCross			█		
Unity			█		
V-Play		█			
WebMatrix	█			█	
Xamarin		█		█	
CP IDEs (Enterprise)					
AnyResense		█		█	
AppConkit				█	
Apprental	█			█	
Genies		█		█	
Genies /Media M2Active					█
App factories					
Appfusa		█		█	
AppMachine	█			█	
Conduit Mobile	█			█	
iGenApps	█			█	
Jacobi Factory	█			█	
MobileCrawler	█			█	
MobileCart	█			█	
Mobile Roadie	█			█	
Phapp	█			█	
Real Foundry	█			█	
Seasmeb SmartApp	█			█	
Shardit	█			█	
Spot Specific	█			█	
WorkApps	█			█	

Fuente: research2guidance, Cross Platform Tool Benchmarking 2013.

1.5.5. Comparativa de aplicaciones preinstaladas accesibles

Muchas veces, es necesario que una aplicación desarrollada tenga acceso a aplicaciones preinstaladas dentro de las plataformas móviles, tales como: calendario, contactos, sistema de archivos, librería de imágenes, mapas, mensajes de texto, teléfono y mapas. Es necesario tener en cuenta cuales de estas aplicaciones integradas tiene acceso cada herramienta multiplataforma, para saber cuál se adapta más a ciertas necesidades. Algunas herramientas no mencionadas en la siguiente gráfica, tales como Phone Gap, presentan dificultades para acceder a las aplicaciones preinstaladas debido a su naturaleza web.

Figura 17. **Número de aplicaciones preinstaladas accesibles desde herramientas multiplataforma**

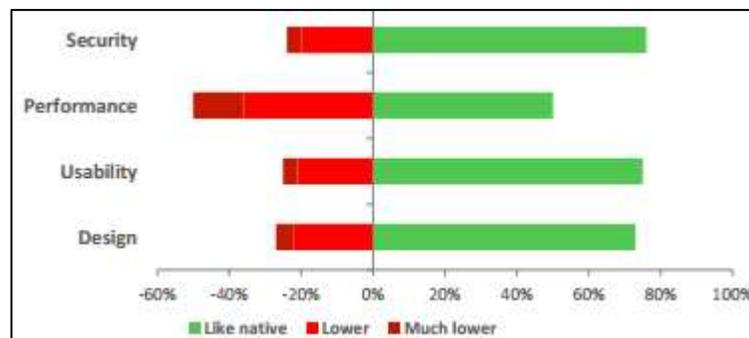


Fuente: research2guidance, Cross Platform Tool Benchmarking 2013.

1.5.6. Comparativa de calidad y rendimiento de aplicaciones creadas con herramientas multiplataforma

Para el actual estudio, la calidad de una aplicación está determinada por su rendimiento, usabilidad, diseño visual y seguridad. La comparativa está realizada contra la calidad obtenida por aplicaciones desarrolladas utilizando métodos oficiales y herramientas nativas. Es posible notar que el rendimiento de las aplicaciones obtenidas, es la debilidad número 1 de la mayoría de herramientas multiplataforma, esto debido a que la mayor parte de estas tienen un enfoque web, el cual utiliza una capa intermedia para la ejecución del software.

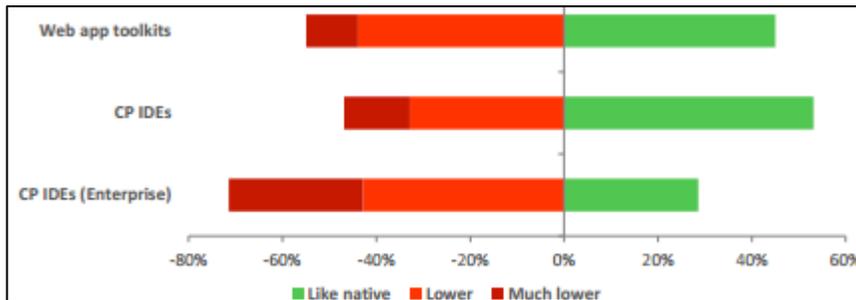
Figura 18. Calificación de calidad de aplicaciones multiplataforma



Fuente: research2guidance, Cross Platform Tool Benchmarking 2013.

Asimismo es posible analizar el rendimiento obtenido con técnicas multiplataforma, según la categorización de la herramienta utilizada. Es notable que el rendimiento de kits de desarrollo web se encuentra por debajo del rendimiento obtenido con IDEs multiplataforma, como Codename One o Marmalede.

Figura 19. **Calificación de rendimiento por categoría de herramientas multiplataforma**



Fuente: research2guidance, Cross Platform Tool Benchmarking 2013.

1.5.7. Selección y recomendación

La alternativa seleccionada y propuesta, con la cual se desarrollará la aplicación de ejemplo y referencia para futuros profesionales, consiste en el Kit Codename One. Esta herramienta utiliza el lenguaje Java para el desarrollo de aplicaciones, el código java es convertido en Bytecodes java, que posteriormente son traducidos a código nativo para cada plataforma destino. Brinda las ventajas del desarrollo multiplataforma con enfoque *Code once, run everywhere* tales como: ahorro de tiempo, esfuerzo y costos, a la vez que brinda aplicaciones con rendimiento comparable (o superior en el caso de iOS) al obtenido con herramientas oficiales y nativas de cada plataforma. La versión inicial de Codename One es gratuita y provee todas las librerías necesarias para desarrollo de aplicaciones promedio y de complejidad baja/normal. El proyecto actualmente cuenta con soporte de muchas organizaciones, debido a las ventajas y facilidades que ofrece y existe abundante documentación y soporte por parte de sus creadores. El sitio oficial web es: <http://www.codenameone.com/>

2. ESTUDIO DE LA TECNOLOGÍA Y SU IMPACTO EN GUATEMALA

2.1. Identificación de la Teoría que soporta la investigación

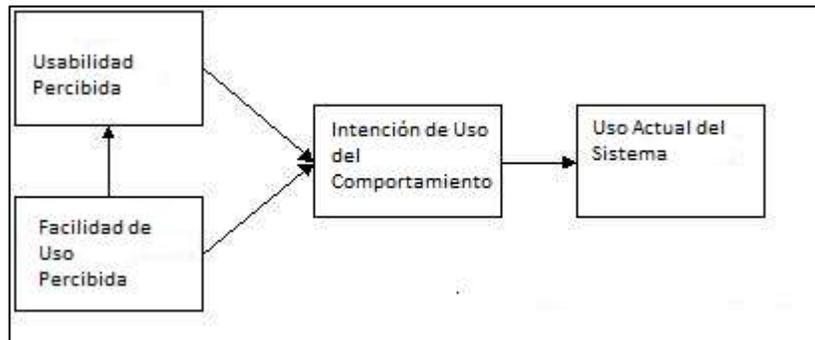
Como en cualquier tipo de investigación, en el campo de conocimientos de Sistemas de la Información existen diferentes teorías que se pueden aplicar en un grado menor o mayor. Así, al definir una teoría, se utiliza su marco de trabajo, que ayuda a comprender hacia quién está enfocado el desarrollo de la investigación (sus beneficiarios), a quienes afectará en su desarrollo, cuál será el enfoque usado para el desarrollo y qué comportamientos se deben conocer con anterioridad para definir requisitos a cumplir.

Existen diferentes teorías, algunas cambian dependiendo del autor de la publicación. Para la elaboración de este documento, se tomaron como referencia las publicadas por la Brigham Young University, en su sitio de internet sobre Teorías para Sistemas de Información, la cual es una colección pública, construida con base en el conocimiento de sus usuarios.

En el caso de la investigación y desarrollo del proyecto que es la razón de este documento, existen dos teorías que son las más cercanas a su fin:

- El modelo de aceptación de la tecnología, que es el enfoque principal de la investigación.
- La teoría de la dominación de la tecnología, que sirve como soporte secundario de la investigación.

Figura 20. **Modelo de aceptación de la tecnología**



Fuente Davis; Fred D. (1989) y Davis, Fred D., et al. (1989).

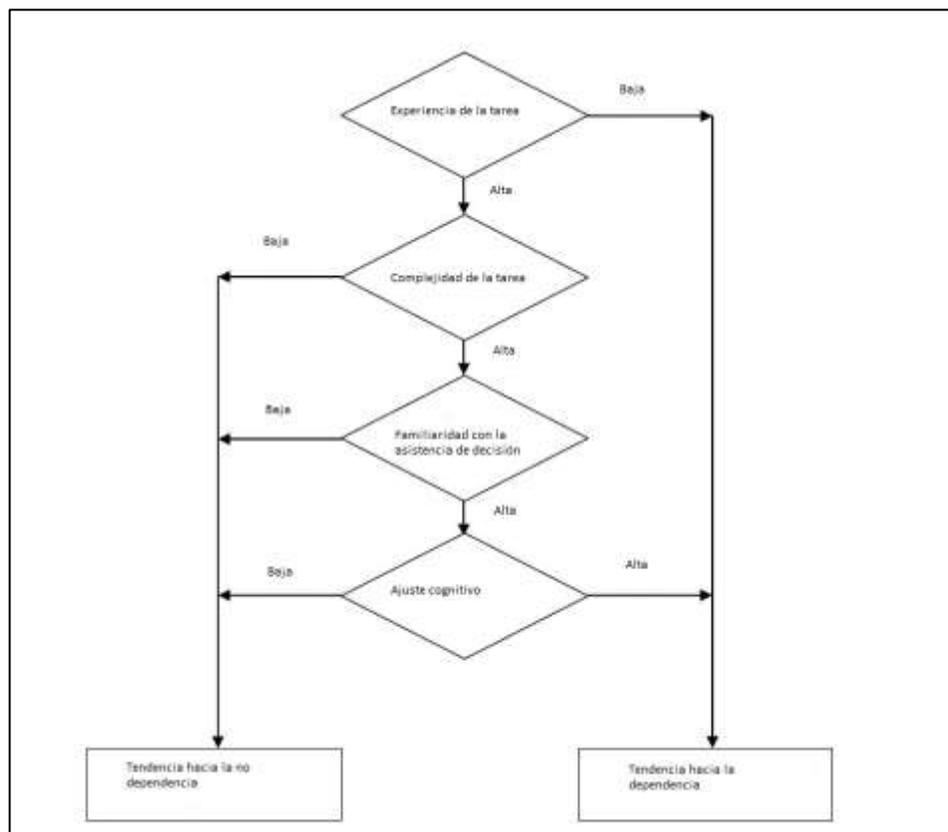
El modelo de aceptación de la tecnología, también conocido por siglas en inglés TAM, es una adaptación de la Teoría de Acción Razonada (TRA), en el campo de la Investigación de Sistemas. TAM postula que la utilidad y facilidad de uso percibidas, determinan la intención de un individuo de usar un sistema, teniendo la intención de usarlo como un mediador con el sistema actual. La utilidad percibida también se considera directamente afectada por la facilidad de uso percibida. Los investigadores han simplificado TAM al remover la construcción de actitud encontrada en TRA, de la especificación actual. En resumen, se debe buscar la facilidad de uso y utilidad, para que el sistema que se desarrolla, sea usado para interactuar con otro sistema.

Anteriormente se describió TAM como una adaptación de la Teoría de Acción Razonada (TRA), la cual se puede definir que postula que comportamiento individual es conducido por intenciones de comportamiento, las cuales son una función de una actitud del individuo hacia comportamientos y normas subjetivas que rodean el comportamiento. En otras palabras TRA modela la investigación de comportamientos que se ven afectados por el propio individuo y las normas que rodean dicho comportamiento, los sentimientos

negativos o positivos que tenga hacia este. Al referirse a normas subjetivas, se refiere a que si el individuo percibe que si realiza un comportamiento, personas importantes a el, lo aprobaran o no. TAM sustituye muchas de las medidas de actitud, por la facilidad de uso y utilidad, dos términos íntimamente relacionados con las medidas de aceptación tecnológica.

TRA busca que tanto la facilidad de uso percibida, la cual depende de la usabilidad percibida, afecten la intención de comportamiento de uso, y a su vez el uso actual del sistema. Este flujo se puede apreciar en la siguiente ilustración.

Figura 21. Teoría de la dominación de la tecnología



Fuente: Davis, Fred D. (1989) y Davis, Fred D., et al. (1989).

La teoría de la dominación de la tecnología (TTD, en inglés), que no afecta fuertemente la investigación, como TAM, pero también es necesario definir. Según la wiki de Investigación de Sistemas de Brigham, TTD tiene dos factores dependientes:

- Dependencia, la medida en la cual un individuo aplica a una asistencia de decisión inteligente e integra las recomendaciones de estas en su juicio.
- Dominio, el estado en la toma de decisiones, en el que una asistencia de decisión inteligente toma el control principal en un proceso de toma de decisión, y no el usuario en sí.

La teoría de la dominación de la tecnología (TTD) también tiene cuatro factores independientes:

- Experiencia en la tarea, el nivel de experiencia que tiene quien hace la decisión, al respecto de completar una tarea.
- Complejidad de la tarea, la medida en que las habilidades cognitivas, de quien hace la decisión, son retadas a completar cierta tarea.
- Familiaridad con la asistencia de decisión, la medida en que, de nuevo, quien hace la decisión, está cómodo con la asistencia de decisión basada en experiencias anteriores o entrenamiento especial.
- Ajuste cognitivo, la medida en que procesos cognitivos son utilizados para la asistencia de decisión para completar la tarea.

Al tener ambos factores, se puede decir que quien hace una decisión puede apoyarse en una asistencia de decisión inteligente bajo estas condiciones:

- Quien hace la decisión tiene poca experiencia en la tarea (factores independientes).
- Los factores dependientes son muy altos.

2.2. Teoría de investigación y relación con la tecnología escogida

Es necesario definir de manera general la tecnología que ha sido escogida para solución que se desea, la cual involucra el desarrollo de software, en este caso de una aplicación móvil.

Debido a que el modelo de aceptación de la tecnología es la principal teoría detrás de este proyecto, la utilidad y la facilidad de uso son factores muy importantes para lograr esta aceptación, por lo tanto, el uso de tecnologías de desarrollo móvil es la que mejor se ajusta a estas necesidades. Considerando la penetración actual de los dispositivos inteligentes se aumenta la disposición de las personas a usar el sistema, el hecho de que tenga las suficientes impresiones positivas como para usarlo.

En cuanto la teoría de la dominación de la tecnología, se trata que exista la dependencia, la necesidad del individuo a utilizar de la aplicación y que esta sirva para la toma de decisiones asistida, el dominio de esta. Ambos puntos, se pueden lograr mediante el uso de dispositivos móviles inteligentes, ya que estos son los aparatos con capacidad de procesamiento adecuada, que son más accesibles y con mayor posibilidad de ser portados.

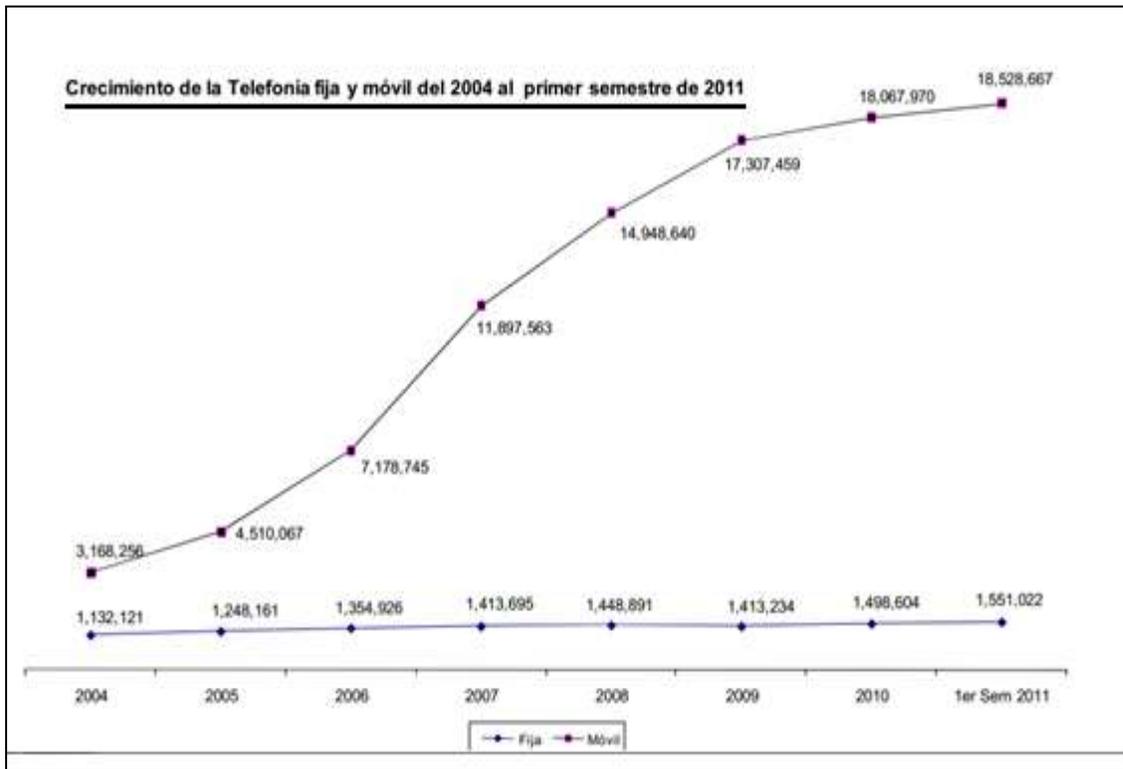
3. IDENTIFICACIÓN DEL PROBLEMA Y SOLUCIÓN QUE LA APLICACIÓN REALIZARÁ

3.1. Antecedentes

El hablar de los antecedentes, que hacen referencia a esta área de desarrollo en dispositivos móviles, es dirigirse a las necesidades que existen o que han existido, las razones de su ser y como se han resuelto, o por lo menos como se han tratado de resolver. También son la base por la que esta aplicación trata de crear tecnología que apoye a los estudiantes, sin importar cual sea el centro académico al cual pertenezcan, como les beneficia no solo el producto final, sino que el desarrollo completo.

En cada periodo estudiantil, el estudiantado de la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala, se da a la tarea de consultar los horarios de cursos publicados y de esta forma crear su horario de semestre. Muchas veces este horario solo se escribe en un papel, lo cual da lugar a situaciones no deseables como pérdida de la información. Similar a esto, la agenda de tareas y actividades del estudiante muchas veces es un documento en papel y esto da lugar a olvidar fechas de entregas o exámenes. Es evidente que se necesita una manera más eficiente de organizar las actividades estudiantiles de la población universitaria.

Figura 22. **Telefonía móvil en Guatemala y su crecimiento**



Fuente: Superintendencia de Telecomunicaciones, 2011.

3.2. Mercado objetivo

El mercado objetivo de la aplicación abarca a la población estudiantil de Guatemala e incluso de fuera del país, que tienen acceso a un dispositivo móvil inteligente, sea un celular o una computadora en formato *tablet*, usando los sistemas operativos Android e iOS principalmente. Este mercado incluye, pero no está limitado, a quienes asisten a una centros de educación superior, los estudiantes universitarios, sin importar si asisten a una institución pública o privada.

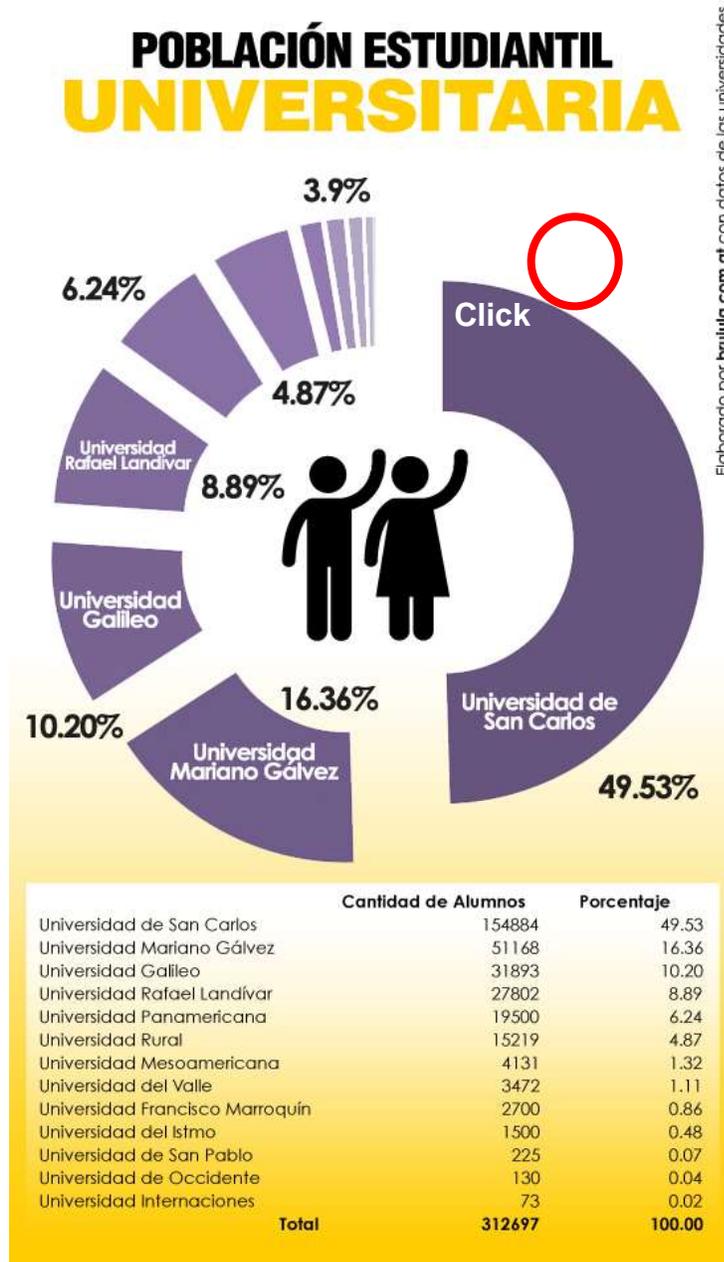
En el 2011, en Guatemala existían actualmente 312 697 estudiantes universitarios, una cantidad que va en aumento. De esta población universitaria, la mayor parte está compuesta por la Universidad de San Carlos de Guatemala, con 153 112 estudiantes, seguida por la Universidad Mariano Gálvez con 60 096.

Un aspecto importante del mercado, es que Guatemala es uno de los países con mayor consumo de servicios y tecnologías de la información relacionadas a dispositivos móviles inteligentes. Actualmente, el setenta por ciento de la población tiene acceso a las telecomunicaciones por medio de dispositivos celulares, y el mercado de los teléfonos inteligentes sigue creciente, con la dominación del sistema operativo de Google, Android y el de Apple, iOS, seguidos por BlackBerry OS.

A pesar de que a nivel general del país y de la región centroamericana, los niveles de aceptación de los dispositivos móviles inteligentes es aún menor que en países más desarrollados, como Estados Unidos, entre la población universitaria, la gran mayoría son usuarios de estos. De nuevo, los sistemas operativos más populares son Android y iOS, que son las plataformas de desarrollo preferidas. La plataforma está dirigida a los estudiantes que a su vez quieren aprovechar las capacidades de sus dispositivos, con un organizador cuyo desarrollo está enfocado en los universitarios, o quienes necesiten un horario dinámico, permitiéndoles organizar mejor su tiempo y las tareas que se les han sido asignadas.

La mayor parte de usuarios de dispositivos móviles inteligentes utilizan un plan de datos que les permite tener acceso al internet usando la red celular, o se conectan por medio de la red inalámbrica existente dentro de los campus universitarios, por lo que el uso de la red no es ningún problema.

Figura 23. Población estudiantil universitaria en Guatemala



Fuente: Brujula.com.gt con información de elPeriódico, “Los 312,697 universitarios ignotos” publicado el 03 de abril de 2011.

3.3. **Benchmark de la aplicación**

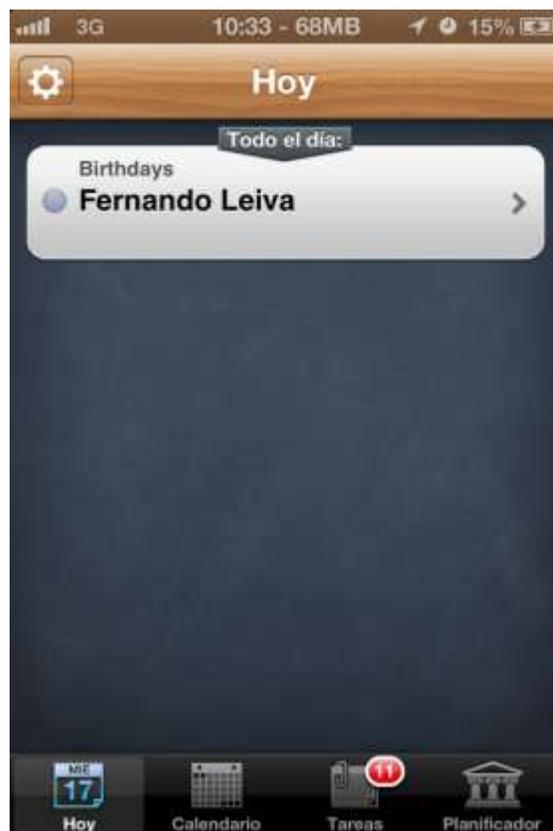
Actualmente existen soluciones que brindan funciones similares a las que brinda este proyecto, cada una con su respectivo conjunto de ventajas y desventajas cuando se les compara entre ellas. Aunque enfocadas a un segmento del mercado muy similar, existen diferencias que aumentan las oportunidades de éxito, y también fortalezas que pueden aprovecharse. Cabe destacar que existen más aplicaciones que pueden ser consideradas como puntos de referencia, pero solamente se analiza una aplicación por plataforma principal de desarrollo y no se toma en cuenta las plataformas candidatas, aunque en el caso de Windows Phone, no es posible hacer la comparación debido a la poca expansión que ha tenido dicha tienda.

Las aplicaciones con las cuales se hará el análisis, el *benchmarking*, son iStudiez Pro, disponible para la plataforma iOS y Class Buddy: Student Planner, para Android, siendo ambas las mejor calificadas y con mayor cantidad de descargas en sus respectivas plataformas, aunque en el caso de la plataforma Android, la cantidad de aplicaciones y usuarios de dichas aplicaciones es mas baja que en iOS. A continuación se presenta un análisis en comparación con estas aplicaciones, dando a conocer tanto las fortalezas como debilidades que tienen y que se deben superar.

3.3.1. iStudiez Pro

Esta aplicación está disponible en la tienda de iTunes, para todos los dispositivos con iOS 4 en adelante. Está enfocada hacia estudiantes universitarios, de diversificado o de secundaria, o a los padres de estos, para llevar un control completo de las actividades que realizan en sus respectivos establecimientos educativos. Permite la sincronización con el servicio de iCloud de Apple, para poder utilizar el servicio con los mismos datos entre diferentes aplicaciones. Entre sus características mas destacables se pueden mencionar:

Figura 24. **Pantalla inicial de iStudiez Pro**



Fuente: elaboración propia, captura de pantalla de iStudiez Pro.

- Sumario Inteligente, que muestra todos los eventos que se tienen para el día actual.
- Revisión de asignaciones, la cual permite ver las tareas u otras actividades ordenadas por curso, fecha y prioridad.
- Integración con iCalc, se sincroniza con el calendario local del dispositivo.
- Planificador, que permite administrar la calendarización de los cursos.
- Notificaciones, las cuales alertan al usuario en caso de una actividad cercana.
- Notas de los cursos que se han asignado a lo largo del uso de la aplicación.

También se debe destacar que existe una versión para tabletas, que es compatible con cualquier iPad o iPad mini y también permite la sincronización por medio de la nube, pero con una interfaz adecuada para estos dispositivos. El acercamiento de las notificaciones es de tipo local, por lo tanto no necesita de una conexión constante de internet para poderlas realizar.

Sin embargo, existen algunas debilidades que tiene sobre el proyecto objetivo y con las demás aplicaciones de la competencia, siendo la principal el hecho que no es multiplataforma, estando limitada a la tienda de iTunes y dispositivos iOS sin planes de desarrollar una versión para los otros sistemas operativos móviles, al menos por el momento. Otra desventaja sería la necesidad de ingresar datos de manera manual, ya que no existe un servidor del cual los usuarios puedan descargar los datos necesarios debido al enfoque genérico de la aplicación, que no está enfocada a ninguna institución académica; es una debilidad en la facilidad de uso. Cabe agregar también la falta de interacción con redes sociales, y por lo tanto, de interacción con el resto de usuarios de la aplicación, y también por el lado económico, la aplicación no

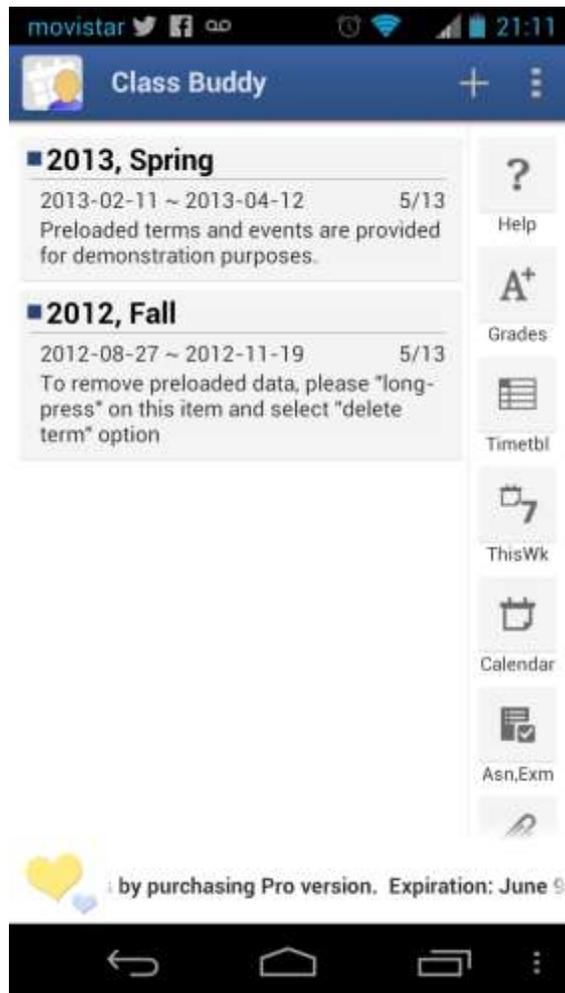
es gratuita, y al momento de la realización de este documento tiene un valor de aproximadamente tres dólares estadounidenses.

3.3.2. Class Buddy

A diferencia de iStudiez, esta aplicación existe en dos versiones: Pro (profesional), versión completa de pago y Lite (ligera), la cual no cuenta con algunas características de la versión profesional, pero que es gratuita y tiene publicidad inserta en la aplicación. Existe también una tercera versión, aunque es solamente un demo por tiempo limitado de la edición profesional.

Cuenta con la mayoría de características que ofrece iStudiez Pro, como los son, organización por semestre, cursos, actividades como juntas con otros estudiantes, notas de todos los cursos, recordatorios en tiempo real y control de gastos, algo que no ofrece la aplicación de iOS. A pesar de que no permite sincronización con la nube, permite exportar la calendarización de la aplicación a un archivo separado por comas (para luego enviarlo o guardarlo dentro de la memoria externa), a la aplicación de documentos en la nube de Google y sincronización automática con el calendario de Google, un *widget* que se puede agregar a la pantalla de inicio de Android, entre otras. En cuanto a seguridad ofrece protección con contraseña y la opción de hacer copias de seguridad (respaldos). La versión gratuita no tiene las opciones de exportar los datos de la aplicación o de copias de seguridad.

Figura 25. Pantalla principal de Class Buddy



Fuente: elaboración propia, captura de pantalla de Class Buddy.

En cuanto a sus debilidades en comparación con el proyecto, comparte varias con la aplicación de iOS, siendo importante señalar el hecho que no es multiplataforma, estando limitada a dispositivos con Android, y sin planes de desarrollar una versión para los otros sistemas operativos móviles, al igual que iStudiez. También cae en la necesidad de ingresar datos de manera manual, ya que no existe un servidor del cual los usuarios puedan descargar los datos

necesarios, debido al enfoque genérico de la aplicación. Destaca en la aplicación también la falta de interacción con redes sociales, y por lo tanto, de interacción con el resto de usuarios de la aplicación. También por el lado económico, la aplicación en su versión profesional, tiene un valor de dos dólares estadounidenses, pero a pesar de ser más barata, no compite en cuanto a características y diseño con iStudiez pro.

4. DISEÑO DE LA APLICACIÓN BAJO LA NECESIDAD IDENTIFICADA

Luego de identificar la necesidad y requerimientos tecnológicos para beneficio de la población objetivo (comunidad estudiantil), y de haber realizado un análisis para una propuesta que solvete los puntos identificados, es necesario realizar el diseño de una solución tecnológica que cubra las carencias identificadas y que, a su vez, ejemplifique e implemente el segmento investigativo del actual trabajo de graduación, es decir el desarrollo de móviles utilizando un enfoque multiplataforma y obteniendo aplicaciones nativas. En el actual capítulo se presenta el diseño y modelado en el cual está basada la aplicación desarrollada, en la mayor parte del capítulo se ha utilizado la notación UML para realizar el modelado por ser la notación más aceptada y utilizada en el medio.

4.1. Análisis a través de casos de uso

El análisis a través de casos de uso permite explorar los requerimientos y necesidades de un sistema desde el punto de vista del usuario del mismo. Esta herramienta facilita el análisis del funcionamiento y operaciones de un sistema desde la perspectiva de los usuarios de este, sin especificar como se implementan dichas operaciones. Por medio de esta notación es posible definir qué hace un sistema sin entrar a detalles de cómo lo hace. A continuación se presenta el modelo de casos de uso para la aplicación móvil estudiantil.

Debido a que la notación de casos de uso está orientada a los usuarios de un sistema, que no necesariamente son personas del ámbito del desarrollo de

software, el diagrama de casos de uso y el nombre de los objetos es autodescriptivo y no necesita explicación a profundidad.

Figura 26. **Diagrama de casos de uso**



Fuente: elaboración propia.

4.2. Diseño y modelado de la arquitectura de la solución

La arquitectura de un sistema consiste en el diseño y abstracción a más alto nivel, en donde se identifican los componentes (vista estática), la responsabilidad de cada componente (vista funcional) y cómo se comportan e interactúan los componentes a lo largo del tiempo, para lograr los objetivos del sistema (vista dinámica).

La arquitectura utilizada en el desarrollo del proyecto es bastante sencilla, y está basada en el conocido y aceptado modelo “cliente-servidor”. La capa cliente consiste en los móviles ejecutando la aplicación, y en el administrador que realiza las tareas involucradas a través de una computadora, ejecutando un navegador web o herramientas cliente administrativas.

La capa de servidor consiste en los servicios web brindados por un servidor web PHP accedido a través de internet. Estos servicios web a su vez cumplen el papel de intermediarios entre los datos almacenados en un servidor corriendo, un DBMS MySQL, y los clientes del sistema previamente mencionados. Asimismo los servicios web permiten a la aplicación comunicarse con la red social Twitter, a través del conjunto de librerías públicas que Twitter pone a disposición de los desarrolladores. La comunicación entre todos los componentes del sistema se realiza a través del estándar JSON, que brinda un lenguaje común para facilitar el intercambio de información entre los diversos elementos.

A continuación se identifican todos los elementos que componen el sistema y su papel o función dentro del mismo:

4.2.1. Arquitectura: vista estática y funcional

- **Móviles:** los móviles ejecutando la aplicación son los clientes principales de la arquitectura y consisten en el conjunto de dispositivos móviles pertenecientes a la población estudiantil.
- **Administrador:** el administrador o administradores del sistema tiene a su cargo tareas como: ajustes y tuneo de los elementos involucrados, actualización de la información ofrecida a los estudiantes, verificación de problemas y solución de los mismos. El acceso al sistema por parte del administrador se realiza a través de computadoras, ejecutando herramientas administrativas tales como PHPMyAdmin a través de internet.
- **Servidor de base de datos:** elemento importante de la arquitectura y perteneciente a la capa de servidores. Este servidor ejecuta un DBMS MySQL en el cual se almacena la información de la aplicación tal como: ciclos académicos, horarios, salones, profesores, alumnos usuarios y su asignación a cátedras. Esta información se encuentra almacenada en una base de datos relacional y todos los accesos (DML) se realizan a través de procedimientos almacenados.
- **Servidor Web:** componente de la capa de servidores que sirve como intermediario entre los móviles clientes y los datos en el servidor de base de datos, así como también la red social twitter (por medio de la API correspondiente). Los servicios web ofrecidos por este servidor fueron desarrollados en PHP y un intérprete del mismo lenguaje realiza su ejecución.

- API Twitter: conjunto de herramientas y librerías que permiten realizar la interacción entre la aplicación y la red social Twitter, con el objetivo de publicar noticias y enviar mensajes con contenido educativo a los usuarios a través de su cuenta en la red social.

4.2.2. Arquitectura: vista dinámica

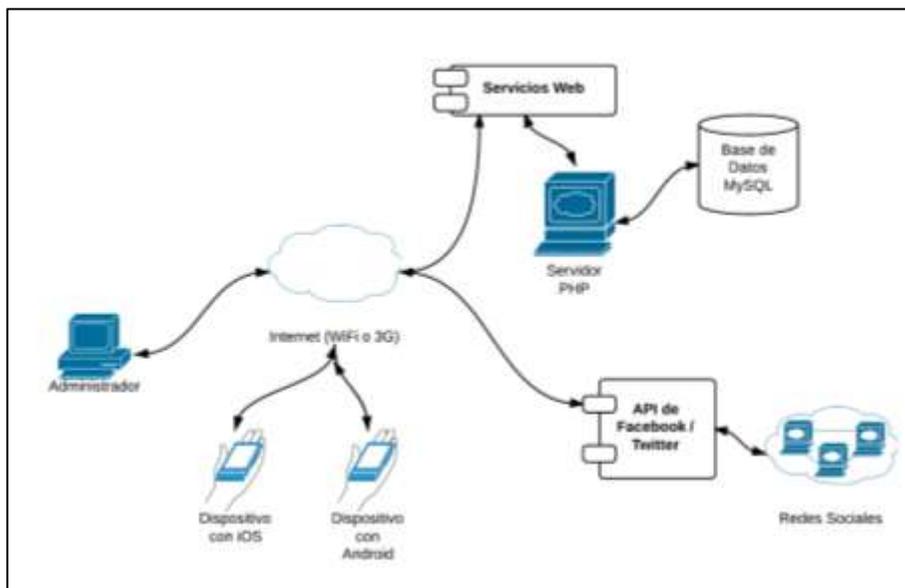
En esta sección se describen las interacciones entre los distintos componentes del sistema a través del tiempo.

Los dispositivos móviles ejecutando la aplicación se conectan a internet y realizan peticiones a través de web services ofrecidos por el servidor PHP. Estas peticiones a su vez son procesadas por el servidor PHP y dependiendo de la naturaleza de la petición, puede requerir la consulta o actualización de información en la base de datos relacional dentro del servidor de base de datos, o bien la publicación de información en la red social Twitter a través de las librerías conocidas como “API Twitter”. El servidor PHP luego envía una respuesta a los dispositivos móviles conteniendo la información solicitada o un status, indicando si se realizó o no la actualización de información de manera correcta. Las peticiones que los dispositivos realizan son:

- Registrarse en el sistema
- Consulta de ciclos académicos disponibles
- Consulta de horarios de cursos para un ciclo académico dado
- Asignación de un curso a un horario dado
- Notificar nueva actividad
- Consultar actividades nuevas
- Agregar anotaciones
- Consultar anotaciones

Cabe mencionar que la aplicación almacenará localmente en los dispositivos información personalizada para un acceso más rápido y eficiente.

Figura 27. Diagrama de arquitectura



Fuente: elaboración propia, con software FluidUI.

4.3. Diseño del modelo de datos

En el ámbito del modelado de datos de la aplicación, es importante analizar que se hace necesario realizar dos tipos de modelado. El modelado local que abarca los datos que localmente se almacenan en cada teléfono móvil, lo cual implica que son pertenecientes a cada usuario y el modelado de la base de datos en el servidor de datos. Este modelado incluye los datos que se almacenan en el servidor para consulta y consumo de todos los usuarios de la aplicación, ejemplos de estos datos son: ciclos disponibles, horarios para asignaturas de dicho ciclo, salones pertenecientes a cada curso, entre otros.

4.3.1. Modelo de datos en servidor

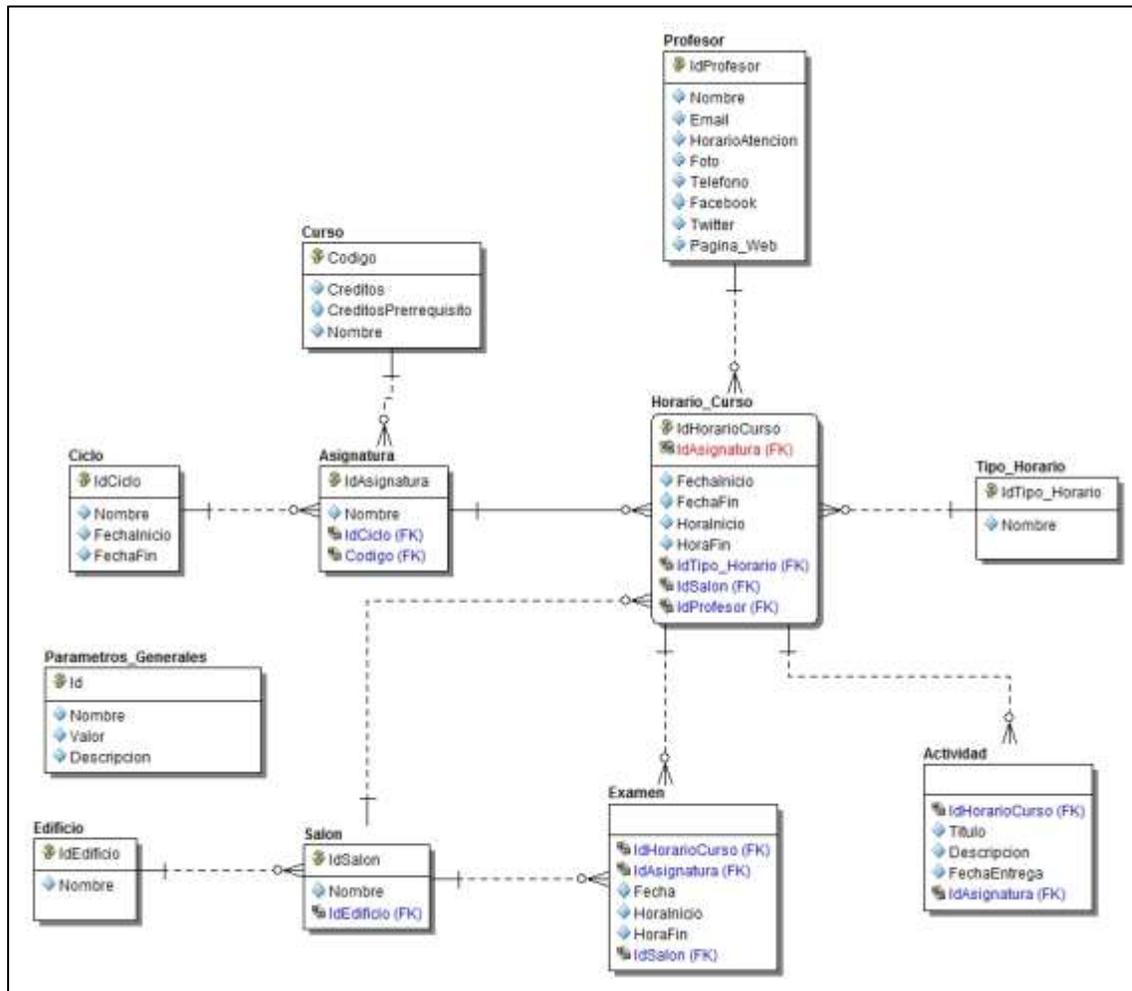
La base de datos en el servidor, es una base de datos relacional, la cual se ha diseñado utilizando el modelo entidad-relación y ha sido implementada usando el lenguaje SQL dentro de un gestor de base de datos MySQL.

Esta base de datos fue diseñada para almacenar toda la información referente a cursos, horarios, profesores, salones, secciones, cursos por carrera, facultades, prerrequisitos, entre otros. Se ha utilizado una nomenclatura de nombres y modelado que permite identificar el rol de cada tabla en el modelo, así como sus relaciones con el resto de tablas sin necesitar un detalle y explicación a fondo, la única excepción es la tabla "Detalle_ciclo" la cual es utilizada para romper relaciones de muchos a muchos entre otras tablas como "Ciclo" y "Curso". Esto debido a que en cada ciclo académico, un mismo curso puede aparecer en varios horarios (secciones), así como también un curso se imparte en varios ciclos. Además que en cada ciclo un curso puede tener: cátedra formal, laboratorio, práctica, seminario. El modelo entidad-relación de la base de datos del servidor se presenta a continuación:

4.3.2. Modelo de datos locales

El modelo de datos, para la información que la aplicación almacena localmente en cada dispositivo, también ha sido diseñado utilizando la nomenclatura entidad-relación, pero este modelo no será implementado utilizando una base de datos relacional con SQL. La implementación de este modelo de datos se realizó utilizando objetos dentro del almacenamiento persistente dentro de cada dispositivo, utilizando códigos identificadores de referencia para otros objetos persistentes cuando se realiza una relación entre entidades. La información almacenada en cada dispositivo es personalizada y depende del uso que se le dé a la aplicación, ejemplos de la información contenida en los datos locales son: cursos asignados por el usuario, actividades pendientes del usuario, fechas de exámenes y su ubicación. La siguiente figura ilustra el modelo de datos locales:

Figura 29. Diseño de modelo de datos local



Fuente: elaboración propia.

4.4. Modelado UML adicional

Para comprender de mejor manera la solución planteada por medio de la aplicación, sin entrar a detalle de su implementación se presentan 2 modelos adicionales que amplían el diseño del software, estos modelos son: diagrama de clases y diagrama de secuencias (para el caso de uso agregar actividad).

4.4.1. Diagrama de clases

En el diagrama de clases, se modelan las abstracciones de objetos a utilizar para el desarrollo de la aplicación. El diseño de la aplicación utiliza una arquitectura de tres capas (datos, lógica de negocio, capa de presentación) en la cual cada capa se construye sobre otra capa subyacente. A continuación se describe cada capa y se concluye con el modelo final representado por medio de un diagrama en UML.

4.4.1.1. Capa de datos

La capa de datos está dividida en dos áreas, una para cada uno de los modelos de datos presentados en la sección de modelado de datos de este capítulo. El modelo de datos en el servidor, es abstraído en el diagrama de clases a través de la clase `WebServiceRequest`, que brinda operaciones para realizar peticiones a la base de datos y leer la respuesta obtenida al ejecutar la petición. El modelo de datos local se abstrae a través de la clase `PObject` (objeto de almacenamiento persistente) y sus clases hijas correspondientes, así como la clase `PersistentStorage` que ofrece funciones para el manejo de objetos persistentes.

4.4.1.2. Capa de lógica de negocio

La capa de lógica de negocio funciona como un intermediario entre lo que el usuario ve y los datos, aplicando flujos según sea necesario en la lógica de la aplicación, es esta capa la que manipula los datos según las necesidades definidas en el software, en el modelo de clases. Esta capa se representa por medio de la clase `BusinessLayer`. Cabe mencionar que la implementación de

esta clase se realizó a través de la clase StateMachine, definida como controlador de interfaces gráficas por default dentro de “Codename One”.

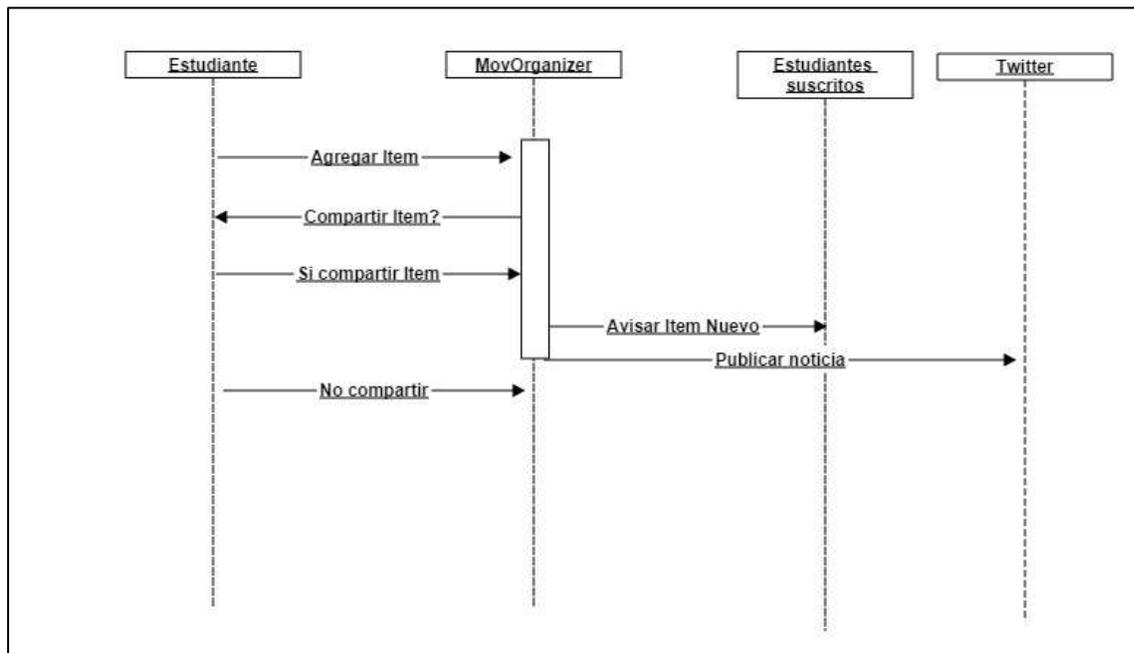
4.4.1.3. Capa de presentación al usuario

La capa de presentación al usuario, representa todos los objetos que el usuario ve y por medio de los cuales interactúa con el software, esta capa realiza solicitudes de procesamiento a la capa de lógica de negocio, la cual a su vez accede y manipula la información de la capa de datos. Para esta capa el modelado se ha realizado por medio de la clase UserInterface es implementada por Codename One a través del objeto theme.res.

4.4.2. Diagrama de secuencias

Dentro de UML, el diagrama de secuencias es una herramienta que permite diseñar, modelar y visualizar de forma interacciones complejas entre un sistema y los actores (usuarios) que lo utilizarán. La aplicación móvil perteneciente a este proyecto, no contiene interacciones complejas con diferentes usuarios, ya que está diseñada para ser utilizada dentro de un móvil personal y sus operaciones no contienen complejidad que amerite ser modelada. La única excepción es el caso de uso Compartir Ítem (y sus casos de uso específicos), ya que implica la posible interacción con múltiples usuarios a través de internet, razón por la cual es útil analizar este caso de uso por medio de un diagrama de secuencias.

Figura 31. Diagrama de secuencias para Compartir Ítem



Fuente: elaboración propia.

4.5. Prototipo

Como parte de un modelo de desarrollo evolutivo, es necesaria la creación de prototipos para el desarrollo de la aplicación, que sirven como base para las siguientes fases de este, y que dan una idea simple de como será la interfaz gráfica que interactuará con el usuario.

4.5.1. Pantalla Inicial

En esta pantalla es donde los usuarios existentes se registran dentro de la aplicación, después de verificar los datos. Si el usuario desea crear una cuenta nueva cuando utiliza por primera vez la aplicación, se puede registrar un usuario nuevo en el servidor. Solamente es necesario registrarse una vez, quedando los datos almacenados de manera permanente dentro del almacenamiento local del dispositivo móvil.

Figura 32. **Pantalla inicial de la aplicación**



Fuente: elaboración propia, con software de FluidUI.

4.5.2. Registro de nuevo usuario

Cuando el usuario decide registrarse por primera vez en la base de datos de estudiantes que usan la aplicación, se muestra un formulario en donde deben llenar los datos básicos para poder ser aceptados dentro del sistema.

Es necesario que proporcione una contraseña que sea de por lo menos seis caracteres de largo, un correo electrónico válido y de manera opcional, proporcionar perfiles de las redes sociales de Facebook y Twitter y una descripción opcional del registrado.

No es posible tener más de un usuario registrado bajo el mismo correo electrónico y solamente es posible registrar a un usuario si la aplicación no hubiera sido utilizada anteriormente, y se hubiera autorizado un registro válido en esta.

4.5.3. Pantalla principal

Es necesario que la aplicación tenga una pantalla inicial para los usuarios que ya se encuentra registrados, la cual muestra la información más relevante de las actividades que se han programado para el presente día.

Figura 33. **Pantalla principal de la aplicación**



Fuente: elaboración propia, con software de FluidUI.

Cuando el usuario ha sido registrado exitosamente en la aplicación, es decir ha ingresado una cuenta de correo válida y su respectiva contraseña, se le muestran las actividades principales organizadas en forma de pestañas en la pantalla principal de la aplicación. Esta interfaz es completamente uniforme entre los dispositivos, sin importar el sistema operativo móvil utilizado por usuario. Esta cuenta con los siguientes:

- Actividades de hoy
- Calendario
- Tareas
- Planificador
- Acerca de nosotros

Cada una de estas tareas es accesible por su respectiva pestaña en la parte inferior de la aplicación. La aplicación trata de buscar la forma más sencilla e intuitiva de lograr cumplir con su misión, así que se decidió utilizar este formato debido a su gran aceptación en la actualidad del desarrollo de aplicaciones móviles, ya que permite navegar entre diferentes funciones de la aplicación de manera delicada y eficiente.

Al finalizar de hacer cualquier tipo de operación, esta pantalla inicial será a donde todos los formularios convergerán, siendo en sí, el eje central del desarrollo de la aplicación.

4.5.4. Actividades de hoy

En esta parte de la aplicación se presentan las actividades, ya sean tareas, cursos u otro tipo de asignaciones, que están próximas a la fecha y hora actual del sistema. De esta manera el estudiante puede tener acceso inmediato a los sucesos que se encuentran más cercanos y que por lo tanto necesitan su atención inmediata.

Este formulario está compuesto por una lista dinámica, la cual con base en la fecha actual y a las actividades en el calendario, se llena, mostrando los eventos más relevantes, ya sea por su urgencia o por su nivel de relevancia. Se pueden seleccionar las actividades y dirigirse a formularios que ofrecen mayor información e incluso editarlos si se desea un cambio a último minuto.

4.5.5. Calendario

Como elemento básico de una aplicación que es un organizador, es necesario el incluir un calendario, para la fácil administración de los eventos del estudiante.

Figura 34. **Formulario del calendario de la aplicación**



Fuente: elaboración propia, con software de FluidUI.

Al igual que el formulario inicial de la pantalla principal, su función es tener una vista global de las actividades a desarrollarse, pero a diferencia de esta, se pueden mostrar las actividades en cualquier día o mes. Su función principal es la de facilitar la búsqueda de actividades pendientes y de controlar los días que vienen.

Al seleccionar un día es posible ver y editar las actividades que se encuentra asignadas en este, por medio de los formularios asignados para dichas funciones. Este calendario se encuentra completamente sincronizado con la nube, mediante el servidor privado que proporciona la aplicación, permitiendo tener un respaldo seguro de las actividades de manera remota.

4.5.6. Tareas

Debe existir también un elemento dentro de la interfaz gráfica para la administración de las tareas que el estudiante agrega a la aplicación y que también le permita organizarlas.

Figura 35. **Formulario de tareas**



Fuente: elaboración propia, con software de FluidUI.

En el formulario de tareas permite la presentación de las actividades que se han ido agregando a los cursos del actual ciclo académico. Acá se permite la edición de dichas actividades, borrarlas o agregar nuevas a los cursos que han sido previamente agregados. Las actividades no solo se limitan a tareas, sino que a sesiones, salidas de campo, laboratorios extracurriculares, entre otros.

Cuando el usuario desea agregar nuevas actividades, solamente debe presionar el botón en la parte superior derecha, que le permite entrar al formulario para ingresar los datos de la nueva actividad. Dentro de dicho formulario el usuario debe especificar como mínimo, el nombre de la actividad, la descripción, la fecha o rango de fecha en que se realizará, el curso al cual pertenece y si desea tener una notificación y cuando desea que el dispositivo inteligente le de la alarma.

Es posible también compartir la actividad mediante el uso de redes sociales, esto se puede hacer posible durante la creación de la tarea o después, seleccionándola directamente de la lista dinámica. Lo que se comparte es una dirección de internet en la cual se encuentra una página que genera contenido dinámico con la información de la actividad, a manera que sea accesible para quienes los desean compartir.

Figura 36. **Agregar actividades o tareas a un curso**



The screenshot shows a mobile application interface titled "Agregar Tarea" (Add Task). At the top left, there is a blue button labeled "Atrás" (Back). Below the title, there are two input fields: "Título" (Title) and "Descripción" (Description). Below these fields, there are four rows of data, each with a label on the left and a value on the right, followed by a right-pointing arrow: "Fecha" (Date) with the value "13/04/13", "Curso" (Course) with the value "Bases de Datos 2", "Prioridad" (Priority) with the value "Media", and "Notificación" (Notification) with the value "1 Hora Antes". At the bottom of the screen, there are two blue social media sharing icons: a Twitter icon and a Facebook icon.

Fuente: elaboración propia, con software de FluidUI.

4.6. Validaciones

- Crear un usuario en la nube.
- Ingresar a la aplicación con un usuario previamente creado en la nube.
- Agregar ciclos académicos a la aplicación.
- Sincronizar los ciclos del usuario con la nube.
- Agregar cursos a cada ciclo académico, dentro de la aplicación.
- Sincronizar los cursos del usuario con la nube.
- Agregar actividades a cada curso, dentro de la aplicación.
- Sincronizar las actividades del usuario con la nube.
- Agregar notas de voz y fotografías a ciertas actividades, de manera local.
- Compartir actividades en forma de páginas en redes sociales.
- Notificar al usuario de actividades cercanas a cierta fecha.

- Sincronizar actividades existentes cuando se registra en un nuevo dispositivo.

4.7. Diseño intuitivo y usabilidad

- Métricas objetivas
 - Tiempo requerido para introducir datos
 - Errores al teclear datos
 - Interacciones durante la instalación de la aplicación
 - Interacciones durante la desinstalación de la aplicación
 - Tiempo necesario y errores al aprender a usar la aplicación
- Métricas subjetivas
 - Satisfacción con el proceso de instalación
 - Satisfacción con la optimización del tamaño de pantalla
 - Satisfacción con la interfaz
 - Seguridad durante la conducción
 - Satisfacción con la pantalla táctil

5. DOCUMENTACIÓN Y TUTORIAL DE PROGRAMACIÓN DE LA APLICACIÓN

En los primeros capítulos del presente trabajo, este ha tenido un enfoque bastante teórico e investigativo en el cual se ha dado a conocer el conocimiento, los fundamentos conceptuales y un análisis del panorama actual en el desarrollo de aplicaciones móviles. Se ha hecho ver que actualmente las diferencias entre las distintas plataformas móviles y sus respectivas herramientas de desarrollo dificultan y multiplican el esfuerzo necesario (y por ende el costo), para llevar una aplicación móvil a los usuarios de las distintas plataformas móviles. Por lo cual se llevó a cabo la investigación y análisis cuyo resultado fue presentar una alternativa al desarrollo móvil multiplataforma, del cual la comunidad de desarrolladores pudiera obtener los mejores beneficios.

Desde un inicio el objetivo de la investigación se inclinó por encontrar una herramienta que brindara las facilidades de generar aplicaciones para múltiples plataformas, pero a la vez pudiera ser usada para crear software nativo destinado a ejecutarse en cada una de estas. La propuesta consiste en la herramienta de desarrollo de software Codename One.

En el presente capítulo se implementa y ejemplifica la creación de una aplicación multiplataforma nativa y se brinda la arquitectura, requisitos, herramientas y un tutorial de programación que pueda servir como referencia o guía para futuros desarrollos de software móvil. La aplicación consiste en un organizador estudiantil móvil con funcionalidad de: registro y control de horarios, actividades y entregas, recordatorios por medio de alertas y redes sociales, así como compartición de recursos como notas de voz, de texto e imágenes.

5.1. Requisitos

En la actual sección se detallan los recursos necesarios, tanto a nivel de hardware como de software, que fueron necesarios para el desarrollo, la implementación y el correcto funcionamiento de la aplicación objetivo de este proyecto. Se incluyen las herramientas primarias e indispensables, tales como: servidor de base datos, servidor de aplicaciones web y SDK; así como también las herramientas secundarias y no esenciales, como lo son: dispositivos móviles de prueba, plugin FTP, repositorio de control de versiones y lector de códigos QR para la instalación en dispositivos de prueba.

5.1.1. Herramientas

Para el desarrollo de la aplicación se utilizan diferentes herramientas, siendo las básicas para cualquier desarrollo de software, como por ejemplo un repositorio, un entorno integrado de desarrollo, entornos de desarrollo, por nombrar algunos de los utilizados en la presente aplicación.

5.1.1.1. Repositorio de control de versiones dentro de Google Code

Google Code es un sitio enfocado a desarrolladores de proyectos relacionados a Google o proyectos open-source, cuenta con múltiples herramientas y servicios tales como APIs, documentación, foros de discusión, blogs, hospedaje de proyectos y material de referencia para las APIs de Google tales como: Google Maps, Youtube, entre otras.

El servicio utilizado durante el proyecto fue el servicio de hospedaje de proyectos de Google Code, este servicio provee un ambiente de desarrollo colaborativo multiusuario, en el cual se puede implementar un control de versiones eficiente a través de internet.

5.1.1.2. Cliente Subversion

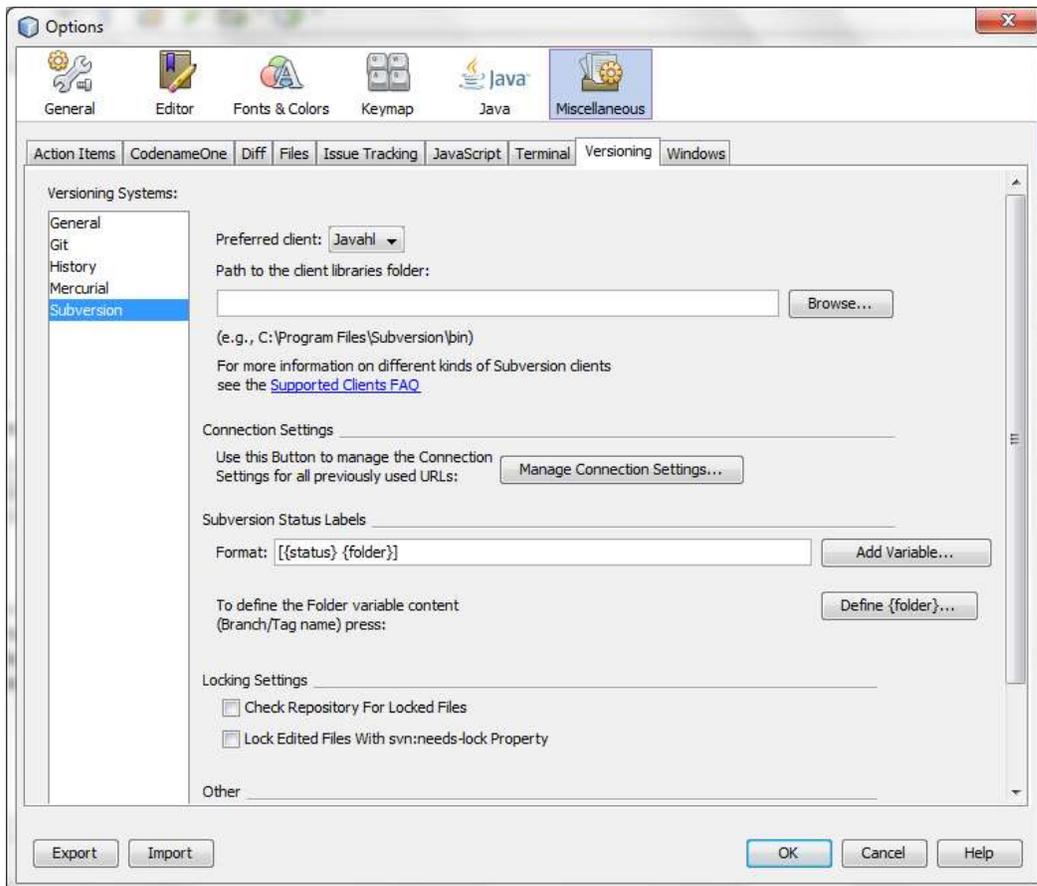
El servicio de hospedaje de proyectos de Google Code, fue utilizado para el manejo de un repositorio central que permitiera montar un ambiente de desarrollo multiusuario con control de versiones, pero se hace necesario contar con una herramienta de versionamiento que utilice este repositorio como insumo para el control. Se usó la herramienta de versionamiento Subversion para este fin, ya que esta cuenta con soporte para Netbeans y se integran fácil y eficientemente.

La interacción entre el repositorio central y Netbeans se realiza a través del cliente Subversion, por medio de envío de comandos de versionamiento desde Netbeans a través del cliente Subversion, el cual a su vez realiza las acciones necesarias sobre el repositorio.

Los pasos necesarios para poder utilizar el control de versiones son los siguientes:

- Instalar el cliente Subversion en cada máquina de desarrollo.
- Configurar en Netbeans la ubicación del ejecutable de Subversion a través de la opción Path to client libraries folder (Tools->Options-Miscellaneous-Versioning->Subversion).
- Crear un nuevo repositorio o bien configurar uno existente dentro de Netbeans, a través del menú Subversion dentro de la opción Team.

Figura 37. Configuración de Subversión en Netbeans



Fuente: elaboración propia, captura de pantalla de Netbeans 7.2.

5.1.1.3. Plugin FTP

La arquitectura de la aplicación incluye una base de datos remota, la cual es accedida por medio de *web services* desarrollados en PHP. Estos *web services* deben ser desplegados para ser ejecutados en un servidor web remoto.

El desarrollo y *testing* de estos *web services* se realiza localmente en el ambiente de desarrollo del programador, para luego ser desplegado en el servidor remoto. Esta tarea de despliegue debe ser realizada por medio del protocolo FTP. Aunque existen herramientas aisladas de FTP se hizo uso de un plugin de Netbeans para facilitar la integración de tareas del cliente FTP dentro del IDE, lo cual facilita la administración y ejecución de tareas de despliegue de los *web services*.

5.1.1.4. Dispositivos de prueba

A pesar de que se cuenta con un emulador de dispositivos móviles dentro del software utilizado para el desarrollo es posible realizar pruebas preliminares en este, se hicieron, es recomendado llevar a cabo las pruebas dentro de dispositivos reales, para analizar cómo se comportaría la aplicación al ser ejecutada en diferentes destinos, tomando en cuenta la diferencia de recursos de hardware, software, tamaño de pantalla y otros aspectos entre los múltiples dispositivos móviles del mercado. Durante el desarrollo del proyecto, los siguientes dispositivos fueron utilizados para llevar a cabo las pruebas:

- Iphone 4S, iOS 6.1.3
- HTC Evo 4g, Android 4.0 (Ice Cream Sandwich)
- Alcatel One Touch 990 (Frozen Yogurt)
- LG Optimus G,Android 4.3 (Jelly Bean)

En Windows Phone y Black Berry las pruebas se realizaron dentro de un emulador por no disponer de los móviles correspondientes.

5.1.1.5. Codename One Build Server

Herramienta brindada por Codename One dentro de su servicio, cuya finalidad es convertir el código de programación en aplicaciones nativas para los dispositivos correspondientes. Es un servicio en la nube que toma como entrada el código intermedio (java bytecode), generado por el ambiente de desarrollo del programador (Netbeans en este caso) y lo traduce a un binario nativo, el cual puede ser descargado o instalado directamente por medio de un código QR.

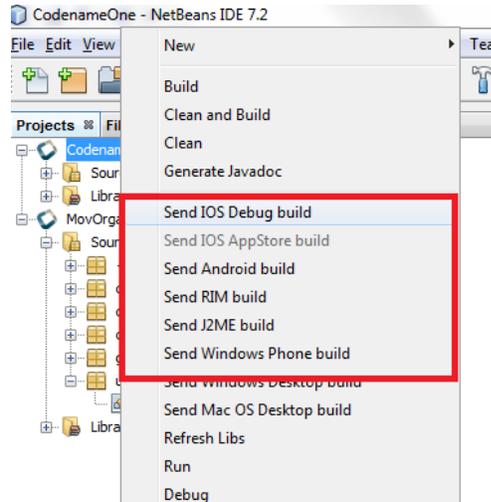
Figura 38. Codename One Build Server



Fuente: elaboración, con programa de Adobe Illustrator CS6.

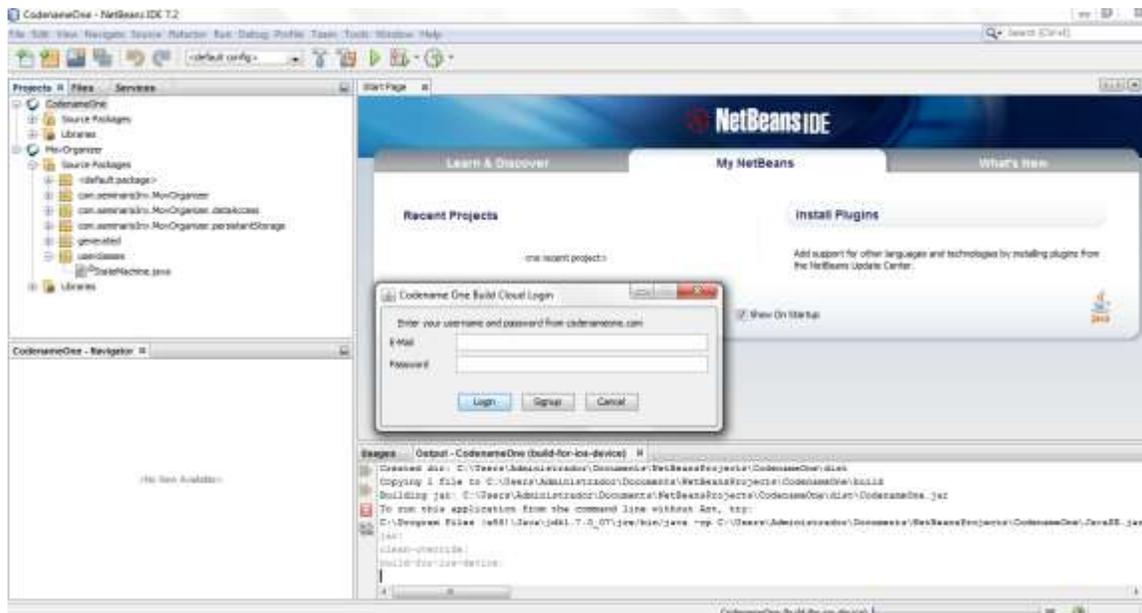
Para hacer uso del Build Server, es necesario estar registrado en Codename One y haber realizado previamente la configuración de este registro dentro del IDE. La configuración se realiza una única vez y se realiza al dar clic derecho sobre el proyecto e intentar generar un binario nativo para cualquier plataforma por primera vez.

Figura 39. **Generación de binario nativo con Build Server**



Fuente: elaboración propia, captura de pantalla de Netbeans 7.2.

Figura 40. **Configuración de registro en IDE**



Fuente: elaboración propia, captura de pantalla de Netbeans 7.2.

El uso del Build Server y el registro a Codename One no es esencial para desarrollar una aplicación, ya que esta puede ser construida en Netbeans y probada dentro del emulador, el uso del Build Server es recomendado para generar binarios nativos que puedan ser probados en dispositivos reales.

5.1.1.6. Lector de códigos QR

Debido a que el Build Server de Codename One dispone de la funcionalidad de generar un código QR, para descargar e instalar la aplicación dentro de un dispositivo, a lo largo de la duración del proyecto se hizo uso de lectores de códigos QR para cada dispositivo de pruebas, con el fin de instalar de manera rápida las diferentes versiones de la aplicación que eran objetivo de verificación y validación. Es posible instalar lectores de códigos QR dentro de un dispositivo, accediendo al mercado de aplicaciones del mismo, la instalación de esta herramienta se encuentra fuera del alcance de este trabajo.

5.1.2. Hardware

En la actual sección dedicada a los requisitos de hardware, se hace necesario realizar una ramificación a dos vertientes: requisitos de hardware para ejecutar la aplicación (requisitos de los dispositivos móviles clientes) y requisitos de hardware para el desarrollo e implementación de la herramienta (requisitos de servidor).

5.1.2.1. Requisitos de instalación y ejecución de la aplicación (clientes)

El hardware necesario para la instalación y utilización de la aplicación en cada dispositivo inteligente, es muy dependiente de la plataforma en la que se planea realizar el uso del sistema.

5.1.2.1.1. Requisitos para IOS

Si la plataforma destino es iOS, existe un gran hermetismo en sus productos y políticas, los dispositivos que utilizan este sistema operativo son todos fabricados por Apple y no existen variantes en el hardware, ni opción a personalización de los mismos, por esta razón los requisitos de hardware no se definen como un conjunto de recursos como tal, sino como modelos de dispositivos. Se define como mínimo que el dispositivo destino sea un iPhone 3GS, siendo recomendado un iPhone 4 o iPad de primera generación, y considerándose óptimo, el iPhone 4S o el iPad 2. Es importante recalcar que independientemente de cual sea el dispositivo, se debe contar con 30 megabytes de almacenamiento.

5.1.2.1.2. Requisitos para Android

En el caso de Android (y otras) plataformas, el ecosistema y mercado es bastante más abierto y diverso que iOS, por lo cual existe una amplia gama de fabricantes y dispositivos, cada uno con diferentes prestaciones y recursos, dándose también el caso de poder personalizar el dispositivo, según las necesidades o gustos de cada cliente. Esto hace que la definición de requisitos de hardware se haga con base en recursos y no con base en dispositivos como

el caso de IOS, a continuación se detallan los requisitos de hardware en términos de recursos computacionales:

- Requisitos de procesador: es necesario al menos un procesador ARMv6 de 500Mhz de un núcleo, aunque es recomendable un procesador ARM Cortex A8 o Snapdragon de 1Ghz. Cada vez los dispositivos tienen características de procesamiento más poderosas, con procesadores de doble y cuádruple núcleo y arquitecturas como ARM A15, por lo tanto las necesidades de la aplicación son fácilmente satisfechas con los estándares actuales.
- Requisitos de memoria: en cuanto a memoria de acceso aleatorio, también conocida como RAM o memoria volátil, es necesario el tener al menos 512 megabytes de memoria en el dispositivo, para uso óptimo y fluido se recomienda 1 gigabyte de este recurso.
- Requisitos de almacenamiento: referente al almacenamiento, es necesario tener 30 megabytes disponibles en el dispositivo para asegurar el rendimiento óptimo. La aplicación cuenta con una base de datos local, la cual está proyectada a crecer un promedio de 1 megabyte por semestre estudiantil (dato obtenido según las pruebas realizadas en los dispositivos usados durante el desarrollo). Al hacer un estimado de uso de 10 semestres de período universitario, se estima un crecimiento de 10 megabytes, lo cual sumado a los 20 megabytes de instalación brinda el total de 30 megabytes, planteados como requisito de almacenamiento.
- Conexión a internet: es necesario contar con el hardware de red encendido para un buen porcentaje de las opciones de la aplicación, tales como: consultar horarios y salones por semestre, compartir eventos

y recursos entre usuarios. De no tener este recurso, la aplicación puede seguir siendo utilizada si ya se cuenta con horarios consultados y asignados desde internet y no se tendrá acceso a notificaciones automáticas entre usuarios.

5.1.2.2. Requisitos para desarrollo e implementación (servidor)

Según la arquitectura de la solución, es necesario contar con un servidor de base datos y un servidor de aplicaciones web que servirá como capa de acceso a los datos a través de internet por medio de *web services*.

Del análisis realizado según los datos obtenidos por medio de encuestas a estudiantes universitarios a nivel nacional (encuestas digitales dirigidas a estudiantes de diversas universidades), fue posible identificar que el uso y acceso a los servidores mencionados no implica un problema para los estándares actuales en este tipo de recursos, ya que los accesos al servidor se realizarán mayormente en período de inicio de semestre, por esta razón mínimas se detallan a continuación.

Es recomendable contar con un procesador Xeon de doble núcleo, con una frecuencia de 2Ghz, 4GB de RAM y 100GB de disco duro, siendo recomendado el tener un enlace dedicado de 2 mega baudios por segundo. Es necesario contar con un sistema de almacenamiento redundante, ya sea con copias de seguridad de la BD y/o discos en espejo; un sistema de alimentación estable, utilizando un regulador de voltaje, una fuente de alimentación de emergencia y una IP pública para poder comunicarse con el servidor a través de internet.

Durante el desarrollo, construcción y pruebas de la aplicación se trabajó con un servidor brindado por el asesor de este trabajo de graduación, el cual cumplía con las necesidades expresadas.

5.1.3. Software

Al igual que los requisitos de hardware, los requisitos de software también deben ser clasificados en dos diferentes categorías: los requisitos de software para instalación y ejecución de la aplicación (software dentro de los móviles clientes), y los requisitos de software para la construcción e implementación de la misma (software utilizado por un desarrollador y software necesario en el servidor).

5.1.3.1. Requisitos de instalación y ejecución de la aplicación (clientes)

Los dispositivos cliente en los cuales se instalará y ejecutará la aplicación deben cumplir con ciertos requisitos de software para el uso de la misma, estos requisitos son mínimos y fácilmente pueden ser satisfechos con la configuración de software de los móviles actuales. Los requisitos recomendados aseguran contar con compatibilidad para futuras actualizaciones y tener acceso a futuras características exclusivas de versiones más recientes del sistema operativo.

A continuación se presenta una tabla que resume estos requisitos:

Tabla I. **Requisitos por plataforma**

Plataforma	Requisitos mínimos sistema operativo	Requisitos recomendados sistema operativo	Requisitos adicionales
Android	Android 2.3.3	Android 4.0 o superior	Acceso a internet
IOS	IOS 4.3	IOS 6.2	Acceso a internet
BlackBerry OS	BB OS 6	BB OS 7.1	Acceso a internet
Windows Mobile	Windows Mobile 6.5	Windows Phone 7.8	Acceso a internet

Fuente: elaboración propia.

5.1.3.2. Requisitos para desarrollo e implementación

Para el desarrollo e implementación de la aplicación, se cuenta con dos capas o categorías en las cuales se agrupa el software requisito, el software del servidor y el software de desarrollo. Estas herramientas de software se detallan a continuación:

5.1.3.2.1. Software del servidor

Es indispensable que el servidor cuente como mínimo con PHP 5, para brindar servicios web utilizados para el acceso a datos remotamente desde los dispositivos a través de internet. Estos datos a su vez son gestionados por medio de un sistema de administración de bases de datos (DBMS) MySQL 5.0, el cual deberá estar instalado y configurado para alojar la base de datos y brindar el cliente de administración PHPMyAdmin. Es indispensable que el servidor cuente con el servicio FTP activado y disponible para poder realizar el *deploy* de los *web services*, necesarios desde el ambiente de desarrollo.

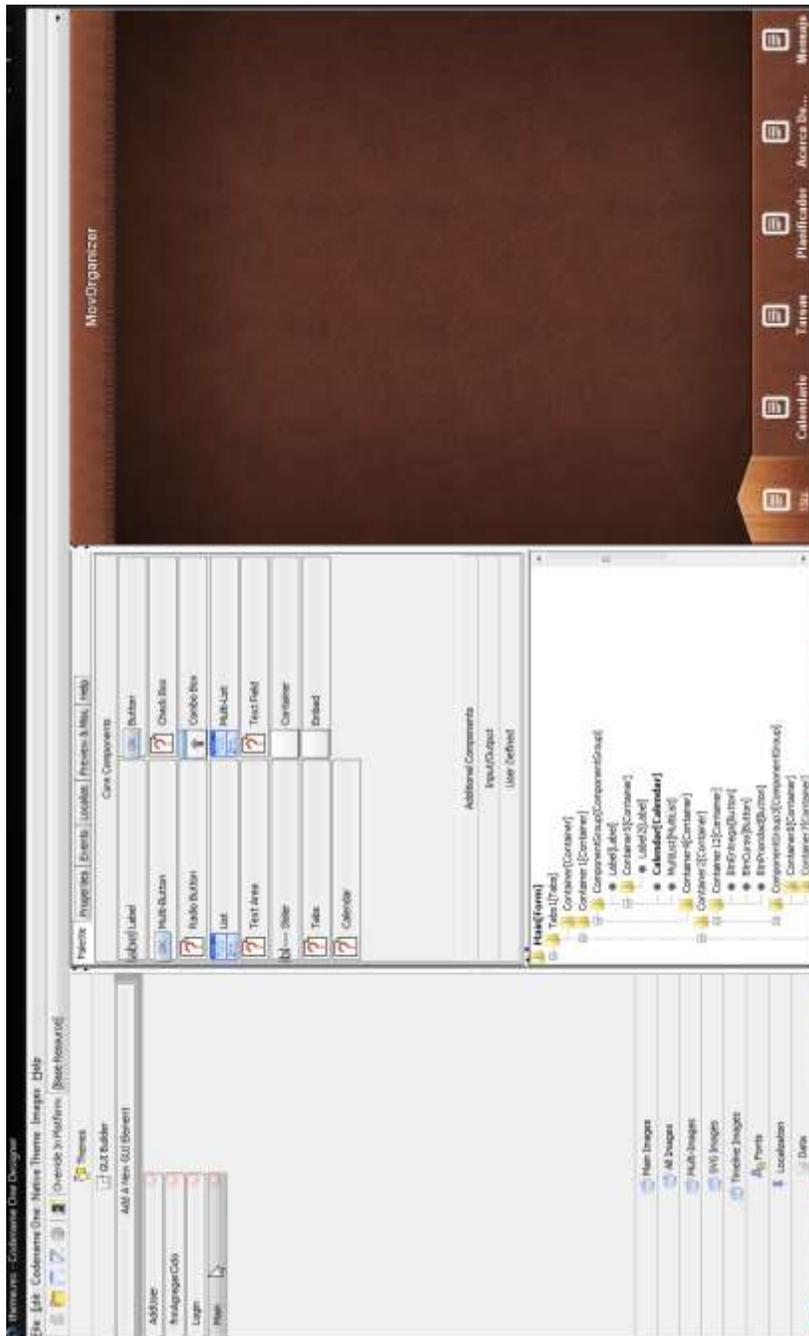
5.1.3.2.2. Software de desarrollo

- Java Development Kit (JDK): herramientas destinadas al desarrollo de aplicaciones utilizando el lenguaje orientado a objetos Java y a generar software que se ejecute sobre la máquina virtual Java. El proceso de desarrollo utilizado a lo largo del actual proyecto difiere del proceso común con el lenguaje java, ya que a diferencia del flujo convencional de desarrollo para esta plataforma, el resultado obtenido luego de que el JDK procesa el código fuente de una aplicación no es ejecutado directamente, en vez de esto se usa como una representación intermedia de la aplicación, la cual posteriormente es procesada y traducida por Codename One (herramienta descrita en las próximas secciones) en código nativo de cada una de las plataformas destino. Es necesario el uso del kit en su versión 1.7 (o superior) para el correcto funcionamiento de las herramientas de desarrollo. El uso de versiones anteriores a esta, puede provocar comportamientos inesperados y otro tipo de problemas.
- Netbeans IDE: ambiente integrado de desarrollo que brinda múltiples características para facilitar la programación de aplicaciones. Se integra de manera excepcional con el lenguaje java y sus herramientas de desarrollo así como con PHP, herramienta utilizada para construir los servicios web utilizados por la aplicación. Se utilizó la versión 7.3 del software.
- Codename One: conjunto de herramientas creadas para realizar desarrollo de aplicaciones móviles con el objetivo de construir aplicaciones en un único lenguaje (java) y desplegar la aplicación resultante en múltiples plataformas destino, la herramienta permite la creación de aplicaciones nativas para cada una de estas plataformas.

Existen diferentes planes de pago y licenciamiento para el uso del kit, utilizando, para el desarrollo del proyecto, dos planes: gratuito y básico, el cual tiene un costo de nueve dólares estadounidenses. Adicionalmente se ofrece también planes con un mayor costo, que brindan opciones, tales como: tener acceso al material de capacitación y características más avanzadas, como notificaciones remotas o acceso al almacenamiento en los servidores propios del desarrollador. La instalación del paquete es bastante sencilla y consiste en ejecutar la opción *plugins* dentro de Netbeans (Tools->*plugins*) y luego colocar Codename One en el buscador de *plugins* y seleccionar Install.

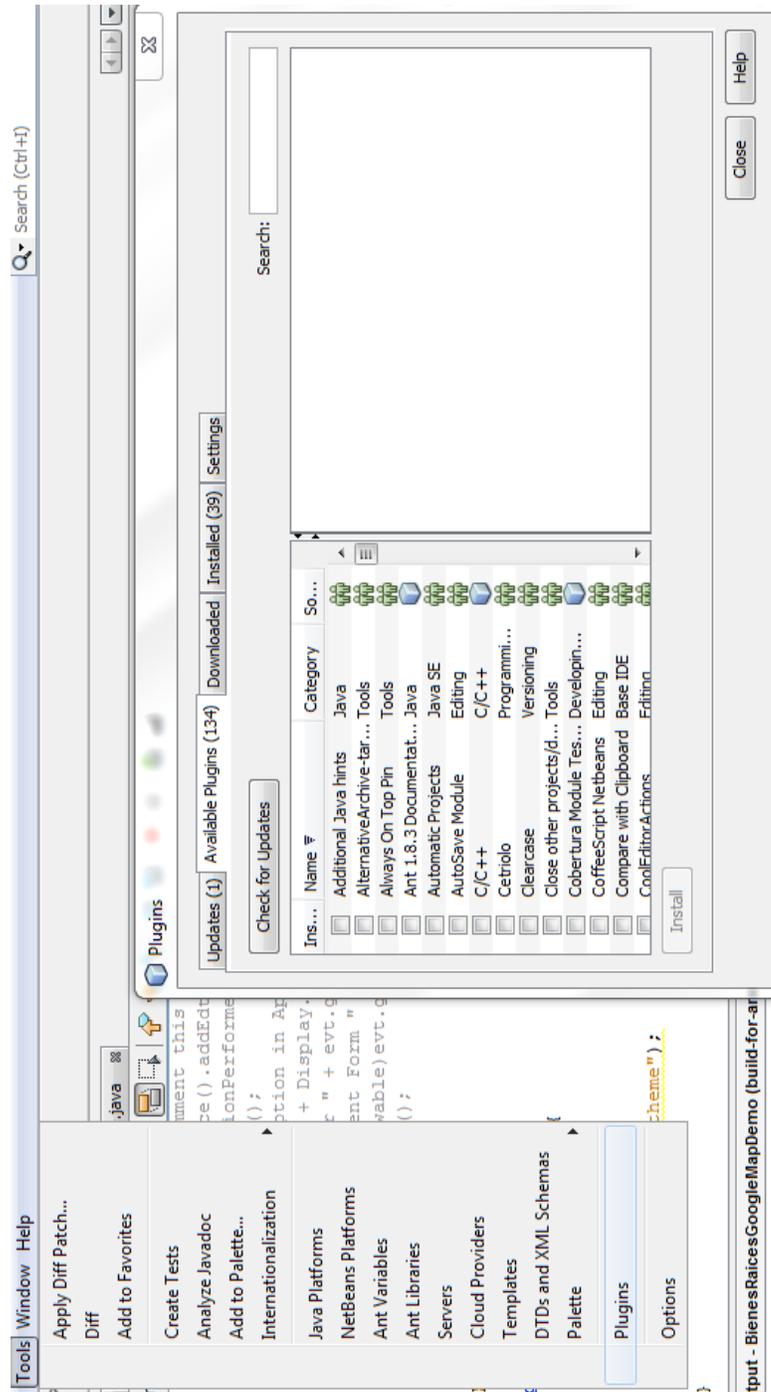
- Plugin PHP: conjunto de herramientas instaladas como *plugin* dentro de Netbeans y utilizadas para desarrollar software PHP, tal como aplicaciones web o *web services*. Los *web services* deben ser desplegados en el servidor de aplicaciones, luego de que han sido probados por medio del protocolo FTP, razón por la cual se instaló un plugin FTP en Netbeans para facilitar este trabajo. Tanto el *plugin* PHP y el *plugin* FTP se instalan utilizando el mismo procedimiento que el *plugin* Codename One descrito.

Figura 41. Codename One sobre Netbeans



Fuente: elaboración propia, captura de pantalla de Netbeans 7.2.

Figura 42. Instalación de plugins Codename One y PHP



Fuente: elaboración propia, captura de pantalla de Netbeans 7.2.

5.2. Tutorial de desarrollo y referencias

El desarrollo de la aplicación fue realizado utilizando una arquitectura de tres capas, utilizando únicamente procedimientos almacenados para la manipulación de datos en el servidor y servicios web, para la ejecución de los mismos. A nivel local fueron utilizados objetos persistentes como abstracción del almacenamiento de datos local. Luego de desarrollada la capa de datos, fue desarrollada la capa de lógica de la aplicación y posteriormente la capa de presentación y las notificaciones PUSH e interacción con Twitter. A continuación se describe el proceso de desarrollo de la aplicación.

5.2.1. Implementación de modelo de datos en servidor

Tal como se expone en la sección 5.3.1. fue diseñado un modelo relacional para solventar las necesidades identificadas para la aplicación. Este modelo tiene como objetivo el almacenar la información compartida entre todos los usuarios de la aplicación. Esta información puede ser consultada por cualquier dispositivo y luego ser utilizada como base para el almacenamiento de información local personalizada, la información compartida en el servidor se encuentra almacenada en las siguientes tablas principales:

- Facultad
- Escuela_unidad
- Carrera
- Ciclo
- Docente
- Edificio
- Salón
- Curso

- Detalle_Ciclo(horario de un curso en determinado ciclo)
- Curso por carrera(relación entre carrera y cursos)

El modelo relacional de la sección 5.3.1. fue diseñado utilizando la notación entidad-relación en el software E/R Studio, de la empresa Embarcadero en su versión de prueba. Esta herramienta de diseño de bases de datos, brinda la funcionalidad de generación del código SQL, correspondiente al modelo diseñado y permite la selección de diferentes sistemas de bases de datos destino. Para el desarrollo de la aplicación del actual proyecto el modelo presentado en la sección 5.3.1. fue implementado en un gestor de base de datos MySQL, por lo cual se utilizó la funcionalidad de generación de código SQL de E/R Studio para crear el DDL de la base de datos correspondiente. El código completo de creación de la base de datos en el servidor se encuentra en el *script* “2013_4_25_1_CreateDataBaseCompleta1.sql” entregado a la Escuela de Ciencias y Sistemas de la Facultad de Ingeniería USAC para referencia y ejemplo a futuros estudiantes.

Para el desarrollo completo de la aplicación se ha utilizado una notación estandarizada para el nombrado de todos los *scripts* desarrollados, incluyendo código DDL y DML. Esta notación estandarizada permite identificar el propósito de cada *script*, así como su orden o secuencia lógica de ejecución en caso de requerirse trasladar la base de datos completa a un nuevo servidor. Este estándar sigue la nomenclatura “año_mes_día_correlativo_proposito.sql”, ejemplos: “2013_4_25_1_CreateDataBaseCompleta1.sql”, “2013_4_25_2_InsertIntoFacultad.sql”.

Como ejemplo del código SQL DDL en el *script* de creación del modelo de datos “2013_4_25_1_CreateDataBaseCompleta1.sql”, se puede observar la creación de las tablas Actividad y Carrera.

Figura 43. **Ejemplo de código SQL de creación de la base de datos en el servidor**

```
CREATE TABLE Actividad(  
    Id                INT                NOT NULL,  
    Detalle_Ciclo     INT                NOT NULL,  
    Titulo             VARCHAR(50)       NOT NULL,  
    Tipo_Actividad    INT,  
    Descripcion       VARCHAR(50)       NOT NULL,  
    FechaEntrega      DATE,  
    Prioridad         INT,  
    Comentario_Descripcion VARCHAR(50),  
    PRIMARY KEY (Id)  
)ENGINE=MYISAM  
;  
  
--  
-- TABLE: Carrera  
--  
CREATE TABLE Carrera(  
    Id                INT                AUTO_INCREMENT,  
    Nombre             VARCHAR(50)       NOT NULL,  
    Facultad           INT                NOT NULL,  
    Nivel_Carrera     INT,  
    Escuela_Unidad    INT,  
   Codigo             VARCHAR(50),  
    Comentario_Descripcion VARCHAR(50),  
    PRIMARY KEY (Id)  
)ENGINE=MYISAM  
;
```

Fuente: elaboración propia, captura de pantalla de Notepad++.

5.2.2. Poblado de catálogos

Para los catálogos tales como:

- Tipo_ciclo: primer semestre, segundo semestre, vacaciones junio, entre otros.
- Facultad
- Nivel_carrera: licenciatura, técnico, posgrado, entre otros.
- Tipo_clase: seminario, cátedra, laboratorio, practica, entre otros.

- Carrera
- Escuela_unidad
- Salon
- Curso

Fue recopilada información oficial de la Universidad de San Carlos de Guatemala, que posteriormente fue alimentada a la base de datos de la aplicación, por medio del lenguaje SQL con sentencias DML INSERT. El poblado de cada tabla catálogo fue realizado con un *script* para cada una, estos *scripts* tienen la nomenclatura y se encuentran en el código entregado a la Facultad de Ingeniería USAC: “año_mes_dia_correlativo_InsertIntoNombre_tabla.sql”. Ejemplos: 2013_4_25_2_InsertIntoFacultad.sql, 2013_4_25_7_InsertIntoRol.sql.

Figura 44. Ejemplo de código SQL de poblado de tablas catálogo

```

INSERT INTO Facultad (Nombre) VALUES ('Facultad de Agronomía');
INSERT INTO Facultad (Nombre) VALUES ('Facultad de Arquitectura');
INSERT INTO Facultad (Nombre) VALUES ('Escuela Superior de Arte');
INSERT INTO Facultad (Nombre) VALUES ('Escuela de Ciencias Políticas');
INSERT INTO Facultad (Nombre) VALUES ('Centro de Estudios del Mar y Acuicultura');
INSERT INTO Facultad (Nombre) VALUES ('Escuela de Ciencias de la Comunicación');
INSERT INTO Facultad (Nombre) VALUES ('Centro Universitario de Peten');
INSERT INTO Facultad (Nombre) VALUES ('Centro Universitario de Imbab');
INSERT INTO Facultad (Nombre) VALUES ('Centro Universitario de Occidente');
INSERT INTO Facultad (Nombre) VALUES ('Centro Universitario del Norte');
INSERT INTO Facultad (Nombre) VALUES ('Centro Universitario de Occidente');
INSERT INTO Facultad (Nombre) VALUES ('Centro Universitario del Noroccidente');
INSERT INTO Facultad (Nombre) VALUES ('Centro Universitario de Santa Rosa');
INSERT INTO Facultad (Nombre) VALUES ('Centro Universitario del Sur');
INSERT INTO Facultad (Nombre) VALUES ('Centro Universitario del Suroccidente');
INSERT INTO Facultad (Nombre) VALUES ('Centro Universitario del Sureste');
INSERT INTO Facultad (Nombre) VALUES ('Centro Universitario del San Marcos');
INSERT INTO Facultad (Nombre) VALUES ('Ciencias Jurídicas y Sociales');
INSERT INTO Facultad (Nombre) VALUES ('Ciencias Económicas');
INSERT INTO Facultad (Nombre) VALUES ('Escuela de Formación de Profesores de Educación Media');
INSERT INTO Facultad (Nombre) VALUES ('Escuela de Historia');
INSERT INTO Facultad (Nombre) VALUES ('Facultad de Humanidades');
INSERT INTO Facultad (Nombre) VALUES ('Facultad de Ingeniería');
INSERT INTO Facultad (Nombre) VALUES ('Instituto Tecnológico Maya de Estudios Superiores');
INSERT INTO Facultad (Nombre) VALUES ('Instituto Tecnológico Universitario Guatemala Sur');
INSERT INTO Facultad (Nombre) VALUES ('Centro Regional Universitario de Jutiapa');
INSERT INTO Facultad (Nombre) VALUES ('Escuela de Ciencias Lingüísticas');
INSERT INTO Facultad (Nombre) VALUES ('Facultad de Ciencias Médicas');
INSERT INTO Facultad (Nombre) VALUES ('Facultad de Odontología');
INSERT INTO Facultad (Nombre) VALUES ('Escuela de Ciencias Psicológicas');
INSERT INTO Facultad (Nombre) VALUES ('Facultad de Ciencias Químicas y Farmacia');
INSERT INTO Facultad (Nombre) VALUES ('Escuela de Trabajo Social');
INSERT INTO Facultad (Nombre) VALUES ('Veterinaria');
INSERT INTO Facultad (Nombre) VALUES ('Otro');

```

Fuente: elaboración propia, captura de pantalla de Notepad++.

5.2.3. Desarrollo de procedimientos almacenados

La arquitectura diseñada para la programación de la aplicación utiliza como estándar para el acceso a datos el uso de procedimientos almacenados, lo cual brinda múltiples ventajas de rendimiento, seguridad y atomicidad a la vez que facilita el mantenimiento y escalabilidad del software. Por esta razón dentro del código de la aplicación (escrito en lenguaje Java) y el código PHP para realizar los servicios web, no es posible encontrar consultas a base de datos y código SQL, solamente llamadas y paso de parámetros a procedimientos almacenados del gestor de base de datos MySQL.

El código de los procedimientos almacenados utilizados se encuentra almacenado en *scripts* con extensión .sql, los cuales se encuentran en el código entregado a la Facultad de Ingeniería junto a este trabajo de graduación.

Para el desarrollo de los procedimientos almacenados, se ha utilizado una notación estandarizada para el nombrado y archivo de los mismos. Esta nomenclatura es la siguiente:

año_mes_día_correlativo_TablaOperaciónParametrosCommand.sql.

El año, mes y día hacen referencia a la fecha de creación del procedimiento almacenado, el correlativo indica el orden en que se crearon los archivos para cierta fecha, el segmento Tabla hace referencia al nombre de la tabla en la base de datos, la Operación depende de la operación SQL a realizar (select, update, insert, delete), los parámetros indican los nombres de los campos de base de datos, pasados como parámetros al procedimiento almacenado y la palabra Command es constante. Algunos ejemplos de *scripts* de creación de procedimientos almacenados son:

- 2013_5_5_2_UsuarioInsertCommand.sql (inserción en la tabla usuario).
- 2013_5_5_3_UsuarioSelectPorUserCommand.sql (consultar usuario por user name).
- 2013_5_5_4_usuarioSelectPorPersonaCommand.sql (consultar usuario por campo persona).

Puesto que la mayoría de tablas, almacenan información que solo está destinada a ser leída por la aplicación, la mayoría de procedimientos almacenados consisten en procedimientos de consulta a través de una cláusula de SQL Select, en esta cláusula, comúnmente se utiliza de parámetro el identificador o llave primaria, para seleccionar un único elemento definido, o bien el valor “-1”, ya que se utilizó como estándar el valor de parámetro “-1” para la selección de todos los elementos de la tabla. Todos los procedimientos de consulta a través de la cláusula Select contienen exactamente la misma estructura y sintaxis, variando únicamente en el nombre de la tabla consultada, y los parámetros utilizados. A continuación se ejemplifica un procedimiento de consulta.

Figura 45. **Ejemplo de procedimiento almacenado de consulta a través de cláusula Select**

```
CREATE PROCEDURE facultadSelectPorIdCommand (PID INT)
BEGIN
    SELECT * FROM Facultad
    WHERE Facultad.Id = PID OR PID = '-1';
END//
```

Fuente: elaboración propia, captura de pantalla de Notepad++.

Además de los múltiples procedimientos almacenados de consulta de información, también existen procedimientos destinados a insertar información en la base de datos, ejemplos de estos procedimientos son, el procedimiento de registro de usuarios (realiza una inserción en la tabla usuario) y procedimiento de compartición de anotaciones (realiza una inserción en la tabla anotación). Estos procedimientos realizan la inserción de información a través de la cláusula Insert de SQL, tomando como valores para la inserción, parámetros del procedimiento almacenado correspondiente. A continuación se ejemplifica un procedimiento almacenado de inserción de información:

Figura 46. **Ejemplo de procedimiento almacenado de inserción a través de cláusula Insert**

```

DROP PROCEDURE IF EXISTS anotacionInsertCommand//
CREATE PROCEDURE anotacionInsertCommand(PdetalleCiclo INT,Pnombre VARCHAR(50),Ptipo_Anotacion INT,PidNota VARCHAR(100),
Puser VARCHAR(50), PCodigoCurso INT)
BEGIN
INSERT INTO Anotacion (Detalle_Ciclo,Nombre,Tipo_Anotacion,IdNota,User,FechaCreacion,Codigo_Curso)
VALUES
(PdetalleCiclo,Pnombre,IF(Ptipo_Anotacion=0,NULL,Ptipo_Anotacion),PidNota,IF(ASCII(Puser)=0,NULL,Puser),Now(),PCodigoCurso)
;
END//

```

Fuente: elaboración propia, captura de pantalla de Notepad++.

5.2.4. Desarrollo de servicios web

Luego de haber creado el modelo de datos y los procedimientos almacenados que acceden a la información en la base de datos, es necesario crear una interfaz que provea acceso a la información a través de internet, debido a que los usuarios potenciales de la aplicación pueden utilizarla en cualquier parte. Esta tarea fue realizada a través de servicios web desarrollados en el lenguaje PHP.

Los servicios web desarrollados utilizan el *driver* MySQLi para llevar a cabo la conexión y ejecución de instrucciones a la base de datos MySQL.

5.2.4.1. Conexión a la base de datos

La conexión a la base de datos, fue centralizada en el archivo “db_config.php”. En este archivo se crea una conexión a través de la función “mysqli_connect”, la cual devuelve una referencia a la conexión que puede ser utilizada en los múltiples servicios web desarrollados, de esta forma si es necesario realizar un ajuste a la conexión (como migración de servidor por ejemplo), solo es necesario realizar el cambio en un único punto dentro del archivo “db_config.php”.

Figura 47. Conexión a través del archivo “db_config.php”

```
// Create connection
$conexion=mysqli_connect($db_server,$db_user,$db_password,$db_name);
```

Fuente: elaboración propia, captura de pantalla de Netbeans 7.2.

5.2.4.2. Servicios web de manipulación de datos

La arquitectura diseñada para el desarrollo del software, emplea un servicio web con su respectivo archivo .php para cada procedimiento almacenado desarrollado. El estándar utilizado, consiste en utilizar en el archivo .php del servicio web el mismo nombre de su procedimiento almacenado existente, por ejemplo para el procedimiento almacenado de consulta de facultades por medio de Id, se desarrolló el procedimiento almacenado

“facultadSelectPorIdCommand” y este procedimiento almacenado corresponde al servicio web facultadSelectPorIdCommand.php.

El paso de parámetros a los procedimientos almacenados se realiza a través del método GET para los procedimientos de consulta de información, y con el método POST para los procedimientos de inserción de información.

Los parámetros son capturados en la interfaz de usuario de la aplicación y pasados a los servicios web para que estos los utilicen en la ejecución de los procedimientos almacenados, el acceso desde el servicio web a un parámetro enviado desde la interfaz de usuario, se realiza a través de los arreglos predefinidos de PHP “\$_GET []” y “\$_POST []”, los cuales son accedidos utilizando el nombre del parámetro como índice.

Figura 48. **Ejemplo de acceso a un parámetro desde servicio web y asignación del valor del parámetro a una variable**

```
$idFacultad = $_GET['idFacultad'];
```

Fuente: elaboración propia, captura de pantalla de Netbeans 7.2.

La ejecución de los procedimientos almacenados desde los servicios web, se realiza por medio de la función query() utilizando como referencia el objeto de conexión global, la función query() recibe como parámetro una instrucción a ejecutar en la base de datos, por lo cual según la arquitectura utilizada el parámetro de la función query() siempre es una ejecución de un procedimiento almacenado. Esta función devuelve un resultado, el cual por estándar se captura en la variable \$result.

El objeto \$result contiene varios registros, según la cantidad de tuplas devueltas por el procedimiento almacenado, por lo cual se recorre el set de datos a través de la función “fetch_assoc()” y cada tupla es asignada a la variable \$row , la cual se descompone en los múltiples campos contenidos por la tupla para formar un objeto de tipo stdClass, el cual es convertido a la notación JSON a través de la función json_encode().

Figura 49. **Ejemplo de servicio web de consulta de información**

```
$idFacultad = $_GET['idFacultad'];

$query = "CALL cicloSelectPorFacultad('$idFacultad')";

$result = $con->query($query);

if($result)
{
    $json = "{\n\"Ciclos\":[\n";

    while($row=$result->fetch_assoc())
    {
        $objCiclo = new stdClass();

        $objCiclo->Id = $row['Id'];
        $objCiclo->Nombre = $row['Nombre'];
        $objCiclo->FechaInicio = $row['FechaInicio'];
        $objCiclo->FechaFin = $row['FechaFin'];
        $objCiclo->Tipo_Ciclo = $row['Tipo_Ciclo'];
        $objCiclo->Facultad = $row['Facultad'];

        $json = $json. json_encode($objCiclo). ",\n";
    }

    $json = $json . "]\n}";
    echo $json;
}
```

Fuente: elaboración propia, captura de pantalla de Netbeans 7.2.

La integración de los servicios web desarrollados en PHP con el código de la aplicación (escrito en lenguaje Java), es llevada a cabo a través de la clase `WebServiceRequest` del paquete `dataAcces`, esta clase extiende a la clase `ConnectionRequest` de las librerías de Codename One.

Esta clase contiene métodos para realizar el consumo de servicios web y obtener la información devuelta por los mismos, en forma de una cadena en formato JSON.

A continuación se ejemplifica el consumo del *web service* presentado en la sección anterior (`cicloSelectPorFacultad.php`), desde el código de la aplicación móvil a través de la clase `WebServiceRequest`. Es importante notar que el resultado del consumo del *web service* devuelve una estructura de datos tipo `Vector`, la cual en este caso en particular almacena todos los ciclos académicos para la facultad utilizada como parámetro.

Figura 50. **Consumo de servicios web desde el código de la aplicación**

```
private Vector obtenerCiclos(int idFacultad)
{
    String strJSON = "";
    WebServiceRequest request = new WebServiceRequest("http://www.ingenieria.com/usuario/2015/usuario/generar_cicloSelectPorFacultad.php", webSocketManager);
    JSONParser parser = new JSONParser();
    InputStream input;
    HashMap tabla = null;
    Vector resultado = null;

    strJSON = request.obtenerJSONGet("idFacultad",String.valueOf(idFacultad));
    input = new ByteArrayInputStream(strJSON.getBytes());

    try
    {
        tabla = parser.parse(new InputStreamReader(input));
        resultado = (Vector)tabla.get("ciclos");
    }
    catch (java.io.IOException ex)
    {
        System.out.println(ex);
    }
    return resultado;
}
```

Fuente: elaboración propia, captura de pantalla de Netbeans 7.2.

5.2.5. Desarrollo de objetos persistentes

La capa de datos de la aplicación se encuentra dividida en dos subcapas, una capa de datos en un servidor de base de datos (tal como se expone en la sección de diseño), la cual almacena información destinada a todos los usuarios de la aplicación y que es accedida, según se expuso en las secciones previas y una capa de datos local a cada dispositivo, en la cual cada usuario almacena información personalizada tal como: que cursos se ha asignado, a que facultad pertenecen los horarios correspondientes a su asignación y calendario de entregas.

Para la capa de datos locales, se hizo uso de almacenamiento persistente en el dispositivo, a través de la clase Storage y la interfaz Externalizable de Codename One. Todo el manejo de objetos persistentes es realizado por clases en el paquete `PersistentStorage`.

La clase `PersistentStorage` proporciona rutinas para realizar tareas tales como:

- Escribir objetos persistentes en el almacenamiento
- Borrar todos los objetos persistentes
- Listar todos los objetos persistentes
- Leer objetos persistentes
- Verificar la existencia de un objeto persistente

Para cada tipo de objeto identificado en el modelo de datos local presentado en la sección de diseño, se ha creado una clase dentro del paquete `PersistentStorage`, cada una de estas clases hereda la clase `PObject` que a su vez implementa la interfaz `Externalizable`. Estos objetos son utilizados como

parámetro en las funciones de la clase PersistentStorage para realizar las operaciones correspondientes.

Figura 51. **Ejemplo de objeto persistente**

```
/**
 *
 * @author Wicho
 */
public class PSFacultad extends PSObject {

    public PSFacultad() {
        super();
    }

    public PSFacultad(String nombre) {
        this.setNombre(nombre);
    }

    public PSFacultad(int id) {
        super(id);
    }

    public PSFacultad(int id,String nombre) {
        super(nombre, id);
    }

}
```

Fuente: elaboración propia, captura de pantalla de Netbeans 7.2.

Es importante mencionar que cada tipo (clase) de objeto persistente, debe ser registrado en la clase PersistentStorage para su uso, esto se realiza dentro de la función estática registerObjects () de la clase PersistentStorage.

Figura 52. Registro de objetos persistentes y ejemplo de manipulación de los mismos

```
private static Storage instancia;  
/**  
 * Metodo que registra cada objeto que queremos hacer persistente para almacenamiento  
 */  
public static void registerObjects()  
{  
    Util.register("PSObject", com.seminarioInv.MovOrganizer.persistentStorage.PSObject.class);  
    Util.register("PSCiclo", com.seminarioInv.MovOrganizer.persistentStorage.PSCiclo.class);  
    Util.register("PSUsuario", com.seminarioInv.MovOrganizer.persistentStorage.PSUsuario.class);  
    Util.register("PSFacultad", com.seminarioInv.MovOrganizer.persistentStorage.PSFacultad.class);  
    Util.register("PSDetalleCiclo", com.seminarioInv.MovOrganizer.persistentStorage.PSDetalleCiclo.class);  
    Util.register("PSActividad", com.seminarioInv.MovOrganizer.persistentStorage.PSActividad.class);  
}  
  
public static void escribirObjeto(String nombre, Object objeto)  
{  
    instancia.writeObject(nombre, objeto);  
    Storage.getInstance().clearCache();  
}  
  
public static void borrarTodos()  
{  
    instancia.clearStorage();  
    instancia.clearCache();  
}  
  
public static String[] listarTodos()  
{  
    return instancia.listEntries();  
}
```

Fuente: elaboración propia, captura de pantalla de Netbeans 7.2.

5.2.6. Desarrollo de capa de presentación y lógica de la aplicación

Es necesario diferenciar las arquitecturas a las cuales corresponde el desarrollo de la aplicación, las cuales deben implementarse en las capas de presentación y lógicas.

5.2.6.1. Capa de presentación

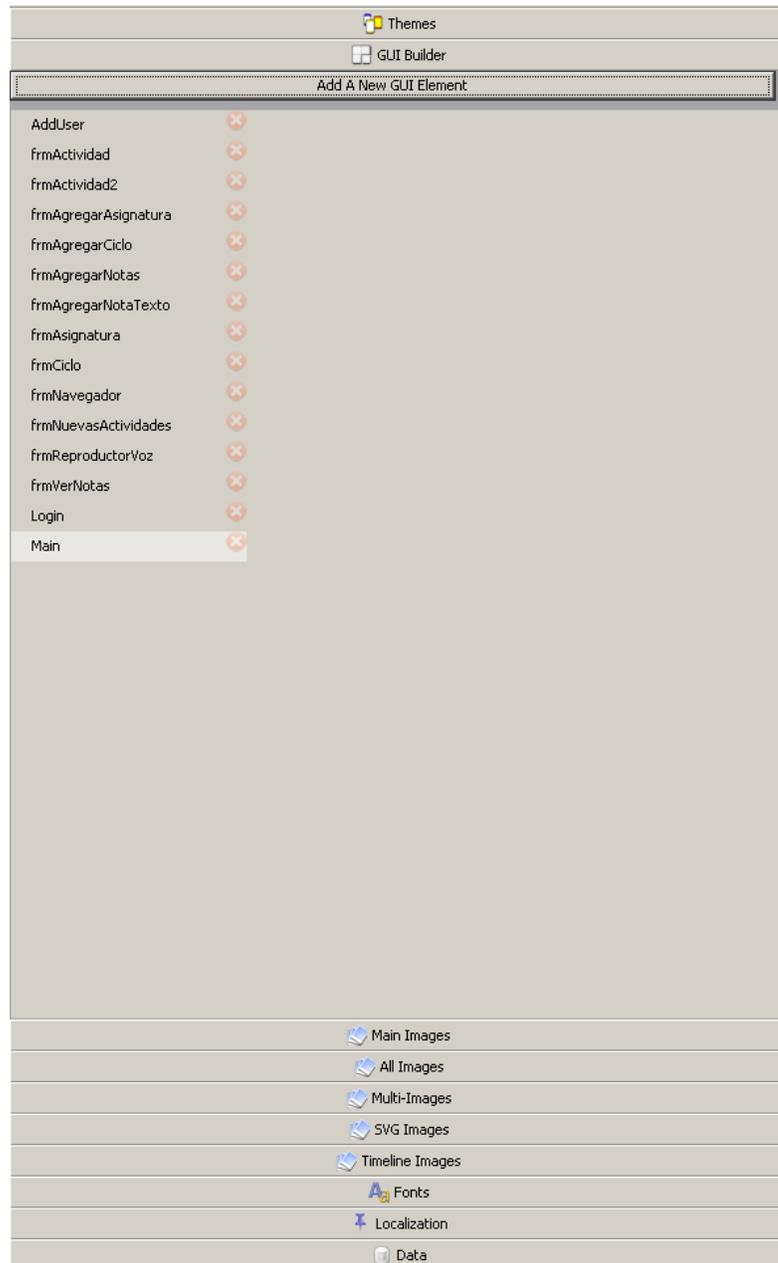
La capa de presentación corresponde a la interfaz de usuario de la aplicación, la arquitectura de 3 capas desarrollada está basada en el patrón Modelo-Vista-Controlador en el cual la interfaz de usuario corresponde a la vista, el modelo corresponde a los datos ya descritos y el controlador corresponde a la lógica de aplicación que será descrita en la próxima sección.

La capa de presentación o vista, es desarrollada de manera centralizada y gráficamente a través del objeto `theme.res` de paquete por defecto en el código del programa.

Dentro de `theme.res` es posible agregar formularios y editar los mismos a través de componentes tales como calendarios, etiquetas y campos de texto que son colocados gráficamente o bien a través de formatos visuales predefinidos.

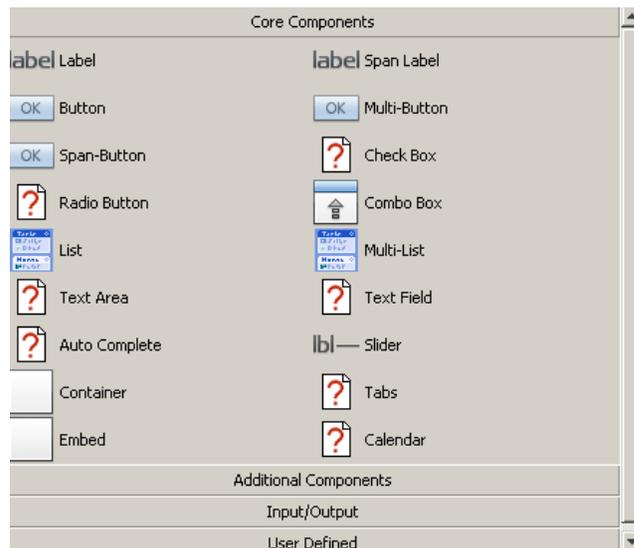
El editor de interfaces gráficas se compone de 5 secciones mostradas en las siguientes imágenes:

Figura 53. **Sección de objetos de la aplicación: en esta sección se listan todos los formularios pertenecientes a la aplicación**



Fuente: elaboración propia, captura de pantalla de Netbeans 7.2.

Figura 54. **Componentes disponibles: librería de componentes gráficos disponibles para utilización**



Fuente: elaboración propia, captura de pantalla de Netbeans 7.2.

Figura 55. **Propiedades del objeto seleccionado: despliega y permite editar las propiedades del objeto seleccionado**

Properties	
Name	Value
Name	ComponentGroup6
LabelForComponent	[null]
NextFocusUp	[null]
NextFocusDown	[null]
NextFocusLeft	[null]
NextFocusRight	[null]
UIID	ComponentGroup
Focusable	<input type="checkbox"/>
Enabled	<input checked="" type="checkbox"/>
ScrollVisible	<input checked="" type="checkbox"/>

Fuente: elaboración propia, captura de pantalla de Netbeans 7.2.

5.2.6.2. Desarrollo de lógica de la aplicación

La capa de lógica de aplicación abarca la manipulación y uso de la información capturada en la interfaz de usuario, aplicando la lógica definida para la aplicación. En la arquitectura de 3 capas definidas, la capa de lógica de la aplicación es el análogo al controlador en el patrón modelo-vista-controlador.

En esta capa se aplican las reglas, validaciones y flujos lógicos de los procesos diseñados para la aplicación.

La implementación es llevada a cabo a través de la clase `StateMachine` predefinida en Codename One y desarrollada en el lenguaje Java.

En esta clase se tiene acceso a los objetos definidos en la capa de presentación (o vista según MVC), a través de referencias obtenidas por medio de llamadas a funciones cuyo nombre inicia con la palabra *find* y es seguido del nombre del objeto en la interfaz de usuario, del cual se desea obtener una referencia. Ejemplo: para obtener una referencia al objeto `cmbCiclosActuales`, se utiliza la función `findCmbCiclosActuales()`.

Toda la definición de eventos de los objetos de la interfaz gráfica es encontrada en la clase `StateMachine`.

Figura 58. Ejemplo de lógica de aplicación y eventos de interfaz gráfica a través de la clase StateMachine

```

private Vector obtenerCicloPorResultado(Lex idPorCiclo)
{
    return this.obtenerCiclo(idPorCiclo, "http://www.lopezodp.com/uaaw/2013/revistas/uev/riscuba/lectPost/resultad.php");
}

private Vector obtenerCiclo(Lex idPorCiclo, String urlServicio)
{
    String stCiclo = "";
    WebServiceRequest request = new WebServiceRequest(urlServicio, new WebClientManager());
    JSONObject parser = new JSONObject();
    InputStream input;
    BufferedReader tabla = null;
    Vector resultado = null;

    stCiclo = request.obtenerJSONSet("idPorCiclo", String.valueOf(idPorCiclo));
    input = new ByteArrayInputStream(request.getJSON().getBytes());

    try
    {
        tabla = parser.parse(new InputStreamReader(input));
        resultado = (Vector)tabla.get("ciclos");

        catch (java.io.IOException ex)
        {
            System.out.println(ex);
        }

        return resultado;
    }

@Override
protected void onFzAgregarCiclo_OneCicloActualizacion(Component c, ActionEvent event) {
    stCicloActualizacion = (stCiclo) this.findCicloActualizar().getSelectedItems();
    stCicloActualizacion = null;

    this.findTxtAreaNombreCiclo().setText("Nombre del ciclo: " + cicloSeleccionado.toString());
    this.findTxtAreaInicioCiclo().setText("Inicio: " + this.obtenerFechaCicloSeleccionado().getFechaInicio());
    this.findTxtAreaFinCiclo().setText("Fin: " + this.obtenerFechaCicloSeleccionado().getFechaFin());
    this.findTxtAreaDuracionCiclo().setText("Duración del ciclo: " + this.obtenerFechaCicloSeleccionado().getFechaFin());
}
}

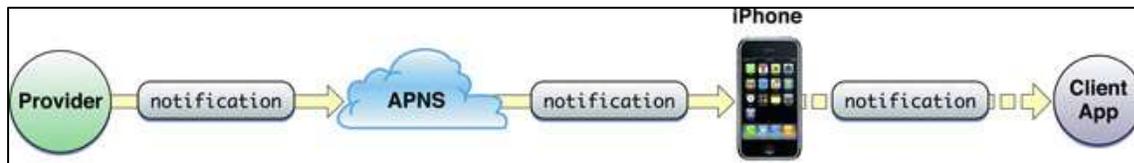
```

Fuente: elaboración propia, captura de pantalla de Netbeans 7.2.

5.2.7. Desarrollo de Notificaciones *Push*

En la actualidad, la mayoría de dispositivos inteligentes y sus aplicaciones, reciben notificaciones de manera casi instantánea y sin intervención directa del usuario, facilitando así el alertar al usuario de diferentes eventos.

Figura 59. Descripción de las Notificaciones



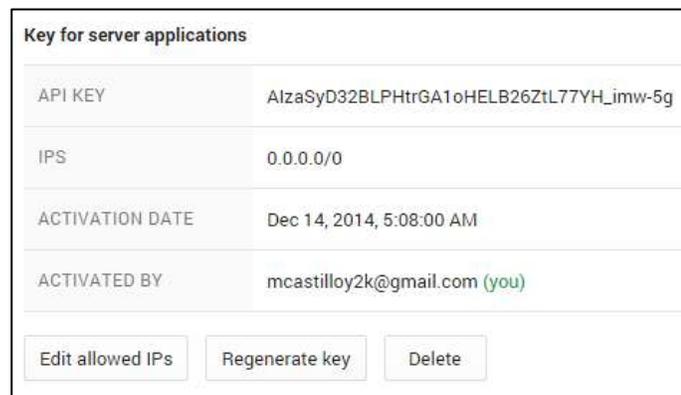
Fuente: Apple.

Las notificaciones *push* son aquellas que son enviadas al dispositivo directamente, en un modelo llamado publicador y suscriptor, en el cual se suscribe a canales de comunicación con un servidor y este le envía información nueva por medio de este. Antes de que se desarrollara por completo este modelo, las notificaciones (como correo electrónico) eran recibidas por medio de una revisión constante del servidor, cada cierto tiempo. Los BlackBerry fueron los primeros dispositivos que permitían notificaciones *push*.

Actualmente las notificaciones *push* están disponibles en todos los dispositivos móviles modernos: iOS, Android, BlackBerry, Windows, entre otros. Durante el desarrollo de la aplicación se tomó en cuenta como plataformas principales iOS y Android, debido a la infraestructura y la documentación existente para la utilización de dichas plataformas y sus notificaciones *push*.

En el caso de las notificaciones *push* con Apple y los dispositivos iOS es necesario el contar con un certificado P12, el cual es similar al que se usa para firmar las aplicaciones. Este certificado debe estar disponible en una ubicación accesible, como por ejemplo, un servicio de almacenamiento de archivos en línea. Google por su parte, utiliza una clave de aplicación y una llave de identificación, las cuales se obtienen al registrarse en la plataforma de desarrollo de Google en la nube. Es necesario también el habilitar el servicio de Mensajería de Aplicaciones de Android en la plataforma de Google.

Figura 60. **Llave de servidor para notificaciones *push***



Key for server applications	
API KEY	AlzaSyD32BLPHtrGA1oHELB26ZtL77YH_imw-5g
IPS	0.0.0.0/0
ACTIVATION DATE	Dec 14, 2014, 5:08:00 AM
ACTIVATED BY	mcastilloy2k@gmail.com (you)

Fuente: Elaboración propia, captura de pantalla de la consola de Google.

En Codename One existen una serie de pasos para que un dispositivo reciba notificaciones *push*:

- Registro local, para que la aplicación se registre y pueda solicitar notificaciones.
- Existe una bandera que indica si existe o no soporte para las notificaciones *push*.

5.2.8. Desarrollo de interacción con Twitter

La interacción con Twitter permite hacer uso de esta red social, con el objetivo de notificar a la comunidad estudiantil acerca de noticias relacionadas a sus cursos asignados.

Esto a través de la creación de Tweets en una cuenta dedicada y mensajes privados a los usuarios asignados a un curso, cada vez que se presenta nuevo contenido del mismo.

Para la programación de esta funcionalidad, fue utilizada la API oficial de Twitter para PHP por medio de la clase "tmhOAuth", que brinda operaciones predefinidas para interactuar con Twitter desde PHP.

La implementación es llevada a cabo a través de la clase "enviarTweetUsuario" que hace uso de las operaciones de la clase "tmhOAuth" para ofrecer las operaciones "enviarTweet" y "enviarMensajeDirecto".

Figura 62. Clase “enviarTweetUsuario”

```
<?php
include 'tmhOAuth/tmhOAuth.php';

class enviarTweetUsuario {
    public $tmhOAuth;

    public function __construct()
    {
        $this->tmhOAuth = new tmhOAuth(array(
            'consumer_key' => 'be3OzSCQ9ia58bkvoUPIg',
            'consumer_secret' => '03JJ13iZsDV8DnnREBHxSr3L2gfs4FDGuvygsNvDtUs',
            'user_token' => '1508545099-fxkzjtvK5I7NJJHqZxG9JKGITFwg2SW721s7oNM',
            'user_secret' => 'UGu28nS8Oo6qeck1EXIE2ZCCQizr1LVMmc5w6V1CQ',
        ));
    }

    public function enviarTweet($message) {
        $response = $this->tmhOAuth->request('POST', $this->tmhOAuth->url('1.1/statuses/update'), array(
            'status' => $message
        ));
        return $response;
    }

    public function enviarMensajeDirecto($user,$message) {
        $response = $this->tmhOAuth->request('POST', $this->tmhOAuth->url('1.1/statuses/update'), array(
            'status' => 'd '.$user.' '.$message
        ));
        return $response;
    }
}
?>
```

Fuente: elaboración propia, captura de pantalla de Netbeans 7.2.

La clase “enviarTweetUsuario” es utilizada dentro del servicio web encargado de agregar una nueva actividad a la base de datos del servidor. Este servicio web es disparado cada vez que un usuario agrega una nueva actividad académica desde su dispositivo.

El servicio web registra la nueva actividad y posteriormente obtiene un listado de todos los usuarios suscritos al curso al cual pertenece la actividad (a través del procedimiento almacenado “AsignacionSelectCommand”), posteriormente hace uso de la clase “enviarTweetUsuario” para publicar un

Tweet con la descripción de la actividad y envía un mensaje directo de Twitter a cada usuario en la lista de asignación.

Figura 63. **Implementación de publicación de Tweet y envío de mensaje directo a usuarios suscritos**

```
$query ="CALL $asignacionSelectCommand($detalleCiclo:");
$asignados = $con->query($query);
$twitterObj = new enviarTweetUsuario();
$message ="El usuario $User agrego la actividad $titulo";
$twitterObj->enviarTweet($message);

while($row=$asignados->fetch_assoc())
{
    $userTwitter = "";
    if(charAt($row['Twitter'], 0)=='@')
        $userTwitter= substr($row['Twitter'],1);

    //omitir al usuario creador de la actividad

    $resultado = $twitterObj->enviarMensajeDirecto($userTwitter, $message);

    if(intval($resultado) == 403)
    {
        $twitterObj->enviarTweet("@$userTwitter @iguena para ver info nueva.");
    }
}
}
```

Fuente: elaboración propia, captura de pantalla de Netbeans 7.2.

5.2.9. Pruebas e instalación en dispositivos reales

Es necesario realizar pruebas en ambientes simulados y reales, para garantizar la estabilidad de la aplicación cuando esta sea llevada a producción. Para este fin se utilizan tanto emuladores como dispositivos físicos.

5.2.9.1. Pruebas preliminares

Las pruebas preliminares son desarrolladas dentro del emulador de dispositivos de Codename One, en el cual es posible seleccionar múltiples dispositivos destino para realizar la prueba.

Para realizar pruebas en el emulador solo es necesario seleccionar la opción “run” o “ejecutar” en Netbeans.

Figura 64. Pruebas en emulador de dispositivos



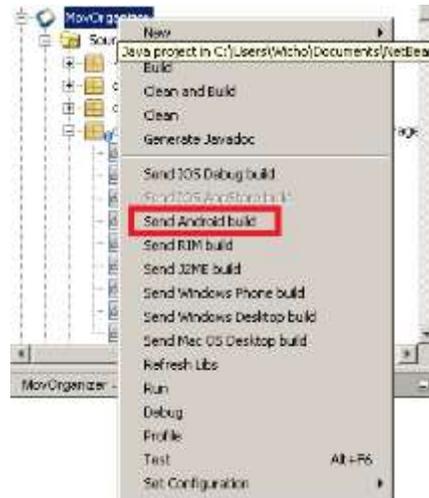
Fuente: elaboración propia, captura de pantalla de Netbeans 7.2.

5.2.9.2. Instalación y pruebas en dispositivos reales

La instalación se realiza a través de tres posibles procesos, estos involucran el servidor Codename One. Es necesario realizar los siguientes pasos:

- Hacer click derecho sobre el proyecto de Netbeans y seleccionar la opción Send Build correspondiente a la plataforma destino deseado.
- Cuando el servidor Codename One termina de construir la aplicación destino, se debe ingresar al sitio de Codename One en la sección Build Server: www.codenameone.com/build-server.html.
- En la sección Build Status se muestran las últimas construcciones, seleccionando una, es posible realizar una de estas tres operaciones, las cuales representan las formas de instalar y probar la aplicación en un dispositivo real.
 - Descargar el archivo de la aplicación (APK para Android, Ipa para IOS por ejemplo) para posteriormente instalarlo en el dispositivo manualmente.
 - Utilizar un código QR y un lector de estos para instalar la aplicación.
 - Enviar por correo un enlace para instalar la aplicación.

Figura 65. Creación de archivo para instalación en Android



Fuente: elaboración propia, captura de pantalla de Netbeans 7.2.

Figura 66. Opciones de instalación dentro del servidor de Codename One



Fuente: elaboración propia, captura de pantalla de la consola de Codename One.

5.3. Consideraciones de implementación

Existen ciertas consideraciones a tener en cuenta, para la implementación de la aplicación o bien desarrollo de mejoras e incremento de funcionalidad a la misma, estas consideraciones se detallan a continuación:

- La notación utilizada para el intercambio y manejo de información entre la aplicación y el servidor de base de datos es JSON (javascript object notation), por medio de web services desarrollados en PHP, por lo cual el direccionamiento a un nuevo servidor de base de datos, o el uso de otros web services (por ejemplo: web services de consumo de información oficial de la Facultad de Ingeniería) debería realizarse manteniendo esta notación y cumpliendo con la definición de datos actuales.
- La manipulación de datos y el SQL correspondiente (DML) ha sido desarrollado en su totalidad, dentro de procedimientos almacenados del gestor de base de datos MySQL, con el objetivo de no colocar código SQL dentro de la aplicación o los web services PHP, evitando así posibles problemas de seguridad y accesos malintencionados a los servidores. Esto también facilita el mantenimiento ya que si se identifica un problema o malfuncionamiento dentro de algún aspecto de la aplicación, solo es necesario corregir el procedimiento almacenado correspondiente una única vez, y no realizar varias modificaciones en donde se utilice el *query* implicado. Si se agrega funcionalidad a la aplicación se debe mantener este estándar, por medio de manejar todo el acceso a datos dentro de la misma base de datos y únicamente utilizar llamadas al procedimiento correspondiente, dentro de la capa de web services.

- La funcionalidad Push notifications utilizada, es un servicio de pago dentro de Codename One, por lo cual para utilizar esta opción es necesario contratar un plan profesional del servicio o bien utilizar código nativo para cada plataforma, por medio de las Interfaces nativas de Codename One, que permiten escribir código nativo de cada plataforma cuando es completamente necesario.
- Debido a las diferencias entre el manejo y funcionalidad de calendarios de cada plataforma, Codename One no brinda herramientas para manipular los eventos y calendarios de cada plataforma, por lo cual el desarrollo de esta funcionalidad fue desarrollado y deberá ser mantenido e incrementado por medio de interfaces nativas que permitan desarrollar código nativo para cada plataforma.

CONCLUSIONES

1. El esfuerzo requerido para desplegar aplicaciones móviles usando las herramientas oficiales, se multiplica proporcionalmente a las plataformas en las que se desea realizar el despliegue.
2. Existen herramientas de desarrollo multiplataforma, siendo estas en su mayoría generadoras de web-apps.
3. Aunque PhoneGap, es la herramienta de desarrollo móvil multiplataforma más popular y utilizada, es la segunda con el peor rendimiento y desaprovechamiento de recursos en las web-apps generadas.
4. Es mucho más común y sencillo utilizar herramientas de desarrollo multiplataforma que generan web-apps, pero el rendimiento de las aplicaciones obtenidas es mucho menor que una nativa y su desaprovechamiento de recursos y características es mucho más elevado.
5. La mayoría de estudiantes y profesionales de ciencias y sistemas y tecnologías de la información, cuentan con experiencia en desarrollo móvil obtenido por medio de algún curso universitario, pero la mayor parte de ellos solo han utilizado las herramientas oficiales orientada a una única plataforma.

RECOMENDACIONES

1. Mantener e impulsar la incursión de prácticas y proyectos de desarrollo móvil, dentro de los laboratorios de los cursos de Ingeniería en Ciencias y Sistemas de la USAC, debido al auge actual del mercado.
2. Utilizar la herramienta Codename One para realizar el desarrollo de aplicaciones nativas destinadas a múltiples plataformas.
3. Suscribirse al canal de noticias de Codename One y mantener constante consulta a su sitio, ya que la herramienta crece y es mejorada día a día.
4. Utilizar la aplicación desarrollada durante este proyecto como punto de partida y referencia para futuros desarrollos móviles multiplataforma.
5. Continuar el desarrollo y mejora de la aplicación desarrollada durante el proyecto, realizando actividades tales como: incursión de otras redes sociales además de Twitter, consumir *web services* oficiales de la Facultad de Ingeniería para tener una versión oficial de horarios semestrales.
6. Expandir el alcance de la aplicación desarrollada para que incluya otras facultades diferentes de la Facultad de Ingeniería.

BIBLIOGRAFÍA

1. ALMOG, Shai. *Codename One Developers Guide version 1.0.1*. Tel Aviv. Israel: Codename One, 2013 p. 589.
2. Brigham Young University. *Theories used in IS Research*. [en línea] <http://istheory.byu.edu/wiki/Main_Page>, Utah, USA. [Consulta: 15 de enero de 2014].
3. Codename One. *How do I?* [en línea] Tel Aviv, Israel. <<http://www.codenameone.com/how-do-i.html>> [Consulta: 25 de enero de 2014].
4. _____. *How Do I - Use Push Notification Send Server Push Messages?* [en línea] Tel Aviv, Israel. <<http://www.codenameone.com/how-do-i---use-push-notification-send-server-push-messages.html>> [Consulta: 25 de enero de 2014].
5. Guatemala Superintendencia de Telecomunicaciones. *Crecimiento de telefonía móvil y fija*, [en línea] <http://www.sit.gob.gt/files/8913/9636/9391/crecimiento_de_la_telefonia_fija_y_movil_2do_2013.pdf> [Consulta: 30 de octubre de 2014].
6. Judge Simon. *Insights into developing for mobile*. [en línea] <<http://www.mobilephonedev.com>> [Consulta: 23 de junio de 2014].

7. Research2guidance. *Cross Platform Tool Benchmarking*, [en línea]
⟨<http://research2guidance.com/cross-platform-tool-benchmarking-2013/>⟩. [Consulta: 15 de enero de 2014].

APÉNDICES

1. ENCUESTA DE CONOCIMIENTO Y EXPERIENCIA EN DESARROLLO MOVIL POR PARTE DE ESTUDIANTES DE CIENCIAS Y SISTEMAS.

1. ¿En qué año entraste a la universidad?
2. ¿Qué semestre cursas actualmente?
3. ¿Has desarrollado alguna aplicación móvil o tienes conocimientos en la materia?
4. ¿Para cual plataforma?
5. ¿Lo hiciste utilizando el SDK y herramientas oficiales u otros?
6. ¿Cuál fue el motivo?
7. Según tu respuesta anterior ¿En qué curso, semestre, proyecto o tutorial te basaste?
8. ¿Conoces la diferencia entre una mobile web-app y una mobile app(nativa)?
9. ¿Conoces un sdk o herramienta para desarrollar aplicaciones móviles multiplataforma?
10. Si respondiste "Si", ¿cuál es su nombre?

2. RESULTADOS TABULADOS ENCUESTA 1

Ida por el programa a...	¿Ha descubierto alguna aplicación móvil o internet...?	¿Usa un ordenador?	¿Usa un móvil con internet?	¿Usa un ordenador?	¿Usa un móvil con internet?	¿Usa un ordenador?	¿Usa un móvil con internet?
200	Programa de...
201	Programa de...
202	Programa de...
203	Programa de...
204	Programa de...
205	Programa de...
206	Programa de...
207	Programa de...
208	Programa de...
209	Programa de...
210	Programa de...
211	Programa de...
212	Programa de...
213	Programa de...
214	Programa de...
215	Programa de...
216	Programa de...
217	Programa de...
218	Programa de...
219	Programa de...
220	Programa de...
221	Programa de...
222	Programa de...
223	Programa de...
224	Programa de...
225	Programa de...
226	Programa de...
227	Programa de...
228	Programa de...
229	Programa de...
230	Programa de...
231	Programa de...
232	Programa de...
233	Programa de...
234	Programa de...
235	Programa de...
236	Programa de...
237	Programa de...
238	Programa de...
239	Programa de...
240	Programa de...
241	Programa de...
242	Programa de...
243	Programa de...
244	Programa de...
245	Programa de...
246	Programa de...
247	Programa de...
248	Programa de...
249	Programa de...
250	Programa de...
251	Programa de...
252	Programa de...
253	Programa de...
254	Programa de...
255	Programa de...
256	Programa de...
257	Programa de...
258	Programa de...
259	Programa de...
260	Programa de...
261	Programa de...
262	Programa de...
263	Programa de...
264	Programa de...
265	Programa de...
266	Programa de...
267	Programa de...
268	Programa de...
269	Programa de...
270	Programa de...
271	Programa de...
272	Programa de...
273	Programa de...
274	Programa de...
275	Programa de...
276	Programa de...
277	Programa de...
278	Programa de...
279	Programa de...
280	Programa de...
281	Programa de...
282	Programa de...
283	Programa de...
284	Programa de...
285	Programa de...
286	Programa de...
287	Programa de...
288	Programa de...
289	Programa de...
290	Programa de...
291	Programa de...
292	Programa de...
293	Programa de...
294	Programa de...
295	Programa de...
296	Programa de...
297	Programa de...
298	Programa de...
299	Programa de...
300	Programa de...

3. ENCUESTA DE USO Y ACEPTACIÓN DE TECNOLOGÍA MÓVIL CON FINES ACADÉMICOS Y PRESENCIA DE SMARTPHONES

1. ¿En qué institución educativa estudia?
2. ¿Cuál es su facultad/carrera o área de estudio?
3. ¿Cuenta con un Smartphone?(Si/no)
4. ¿Qué sistema operativo tiene su Smartphone?(Si tiene más de uno, seleccione todos)
5. ¿Conoce y tiene acceso al market de aplicaciones de su Smartphone?
6. ¿Utiliza frecuentemente las aplicaciones de su Smartphone ajenas a llamadas, mensajería y redes sociales?
7. ¿Utiliza actualmente alguna aplicación con fines académicos/educativos?
8. ¿Qué actividades realiza?
9. ¿Conoce algún método de compartición de recursos educativos tales como: notas de texto, notas de voz, imágenes y fotografías, documentos?
 - Si respondió si: ¿Cuenta con notificaciones inmediatas y recordatorios?
 - Si respondió no: ¿le gustaría compartir recursos, notificando y descargando inmediatamente en su teléfono cuando hay nuevo contenido?
10. ¿Le agradaría y daría uso a una aplicación móvil que ayude a organizar sus horarios, actividades, entregas, recordatorios en su calendario (y redes sociales) y que permita compartir contenido entre compañeros de clase?

4. RESULTADOS TABULADOS DE LA ENCUESTA 2

Institucion	Facultad/Carrera	Ocurrencias
Universidad Panamericana	Ciencias Economicas	1
Universidad Panamericana	Ingenieria	1
Universidad del Itsmo	Ciencias Economicas	1
Universidad Galileo	Ingenieria	1
Universidad Galileo	Ciencias de la Comunicacion	1
Universidad Mariano Galvez	Ciencias de la Administracion	6
Universidad Mariano Galvez	Administracion de empresas	5
Universidad Mariano Galvez	Auditoria	1
Universidad Mariano Galvez	Diseño Grafico	1
Universidad Mariano Galvez	Ingenieria	9
Universidad Mariano Galvez	Maestria en administracion de negocios	4
Universidad Mariano Galvez	Mercadotecnia	3
Universidad Rafael Landivar	Derecho	1
Universidad de San Carlos	Area Docente	1
Universidad de San Carlos	Arquitectura	1
Universidad de San Carlos	Ciencia Politica Sociologia	2
Universidad de San Carlos	Ciencias Juridicas y Sociales	1
Universidad de San Carlos	Ciencias Economicas	1
Universidad de San Carlos	Ciencias de la Comunicacion	1
Universidad de San Carlos	Facultad de Humanidades	1
Universidad de San Carlos	Ingenieria	24

Institucion	Facultad/Carrera	Plataforma	Ocurrencias	AccessMarket	SuccessMarket	UsaAppAcademica	NetUsaAppAcademica	DotUsaApp	NetBariote
Universidad Panamericana	Ciencias Economicas	Android	1	1	0	1	0	1	0
Universidad Panamericana	Ingenieria	iOS	1	1	0	0	1	1	0
Universidad del Itsmo	Ciencias Economicas	Android	1	1	0	0	1	1	0
Universidad Galileo	Ingenieria	Android	1	1	0	1	0	1	0
Universidad Galileo	Ciencias de la Comunicacion	Android	1	1	0	0	1	1	0
Universidad Mariano Galvez	Ciencias de la Administracion	Android	0	0	1	1	0	0	0
Universidad Mariano Galvez	Administracion de empresas	Android	4	4	0	2	3	4	0
Universidad Mariano Galvez	Administracion de empresas	iOS	1	1	0	0	1	1	0
Universidad Mariano Galvez	Auditoria	Android	1	1	0	1	0	1	0
Universidad Mariano Galvez	Diseño Grafico	Android	1	1	0	0	1	1	0
Universidad Mariano Galvez	Ingenieria	Android	0	0	1	2	0	0	0
Universidad Mariano Galvez	Ingenieria	Windows Phone	1	1	0	0	1	1	0
Universidad Mariano Galvez	Maestria en administracion de negocios	Android	1	1	0	1	2	2	0
Universidad Mariano Galvez	Maestria en administracion de negocios	iOS	1	1	0	1	0	1	0
Universidad Mariano Galvez	Mercadotecnia	Android	1	1	0	0	1	1	0
Universidad Mariano Galvez	Mercadotecnia	iOS	1	1	0	0	1	1	0
Universidad Rafael Landivar	Derecho	Android	1	0	1	0	1	1	0
Universidad de San Carlos	Area Docente	Android	1	1	0	0	1	1	0
Universidad de San Carlos	Arquitectura	Android	1	1	0	0	1	1	0
Universidad de San Carlos	Ciencia Politica Sociologia	Android	1	1	0	1	0	1	0
Universidad de San Carlos	Ciencia Politica Sociologia	iOS	1	1	0	1	0	1	0
Universidad de San Carlos	Ciencias Juridicas y Sociales	Android	1	1	0	1	0	1	0
Universidad de San Carlos	Ciencias Economicas	Android	1	1	0	1	0	1	0
Universidad de San Carlos	Ciencias de la Comunicacion	BlackBerry	1	0	1	0	1	1	0
Universidad de San Carlos	Facultad de Humanidades	Android	1	1	0	0	1	1	0
Universidad de San Carlos	Ingenieria	Android	15	14	1	3	12	14	1
Universidad de San Carlos	Ingenieria	iOS	4	4	0	1	0	4	0
Universidad de San Carlos	Ingenieria	Windows Phone	1	1	0	0	1	1	0

ANEXOS

1. Modelado de requerimientos a través de casos de uso:
<http://users.dcc.uchile.cl/~psalinas/uml/casosuso.html>
2. Modelo entidad relación: <http://www.genbetadev.com/bases-de-datos/fundamento-de-las-bases-de-datos-modelo-entidad-relacion>
3. Modelado de estructura estática a través de diagrama de clases:
<http://users.dcc.uchile.cl/~psalinas/uml/modelo.html>
4. Arquitectura cliente servidor:
http://catarina.udlap.mx/u_dl_a/tales/documentos/lis/marquez_a_bm/capitulo5.pdf
5. Formato de intercambio de datos “JSON”: <http://www.json.org/>
6. Lenguaje PHP: <http://www.w3schools.com/php/>
7. Lenguaje SQL: <http://www.w3schools.com/sql/>
8. Lenguaje Java: <https://docs.oracle.com/javase/tutorial/java/>
9. Patron “Modelo-Vista-Controlador”:
<http://www.lab.inf.uc3m.es/~a0080802/RAI/mvc.html>

