



Pontificia Universidad
Católica del Ecuador

ESCUELA DE INGENIERÍA EN SISTEMAS

Tema:

DESARROLLO DE LA AGENDA ACADÉMICA MÓVIL PARA LOS ESTUDIANTES DE LA PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR SEDE AMBATO BASADO EN EL SISTEMA OPERATIVO ANDROID.

Proyecto de Investigación previo a la obtención del título de Ingeniero de Sistemas y Computación.

Línea de Investigación:

Sistemas de información y/o nuevas tecnologías de la información y comunicación y sus aplicaciones.

Autor:

JOSÉ ALEXIS CHACÓN ESTRELLA

Director:

SANTIAGO ALEJANDRO ACURIO MALDONADO

Ambato - Ecuador

Noviembre 2018

PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR
SEDE AMBATO
HOJA DE APROBACIÓN

Tema:

DESARROLLO DE LA AGENDA ACADÉMICA MÓVIL PARA LOS
ESTUDIANTES DE LA PONTIFICIA UNIVERSIDAD CATÓLICA DEL
ECUADOR SEDE AMBATO BASADO EN EL SISTEMA OPERATIVO
ANDROID.

Línea de investigación:

Sistemas de Información y/o Nuevas Tecnologías de la Información y
Comunicación y sus aplicaciones

Autor:

JOSÉ ALEXIS CHACÓN ESTRELLA



Santiago Alejandro Acurio Maldonado, Ing. Mg.

CALIFICADOR

f. 

Teresa Milena Freire Aillón, Ing. Mg.

CALIFICADORA

f. 


José Marcelo Balseca Manzano, Ing. Mg.

CALIFICADOR

f. 

Mónica Patricia Mena Moreno, Ing. Mg.

**DIRECTORA DE LA ESCUELA DE INGENIERÍA
EN SISTEMAS**

f. 

Hugo Rogelio Altamirano Villarroel, Dr.

SECRETARIO GENERAL PUCESA

f. 



Ambato - Ecuador

Noviembre 2018

DECLARACIÓN Y AUTORIZACIÓN

Yo: **JOSÉ ALEXIS CHACÓN ESTRELLA**, con **CC. 055001442-7**, autor del trabajo de graduación intitulado: “DESARROLLO DE LA AGENDA ACADÉMICA MÓVIL PARA LOS ESTUDIANTES DE LA PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR SEDE AMBATO BASADO EN EL SISTEMA OPERATIVO ANDROID”, previa a la obtención del título profesional de **Ingeniero en Sistemas**, en la escuela de **Sistemas**.

1.- Declaro tener pleno conocimiento de la obligación que tiene la Pontificia Universidad Católica del Ecuador, de conformidad con el artículo 144 de la Ley Orgánica de Educación Superior, de entregar a la SENESCYT en formato digital una copia del referido trabajo de graduación para que sea integrado al Sistema Nacional de Información de la Educación Superior del Ecuador para su difusión pública respetando los derechos de autor.

2.- Autorizo a la Pontificia Universidad Católica del Ecuador a difundir a través de sitio web de la Biblioteca de la PUCE Ambato, el referido trabajo de graduación, respetando las políticas de propiedad intelectual de Universidad

Ambato, noviembre 2018



JOSÉ ALEXIS CHACÓN ESTRELLA

CC. 055001442-7



BIBLIOTECA

DEDICATORIA

Sin duda alguna el presente trabajo va dedicado a toda mi familia en especial a mis padres José Chacón y Marcia Estrella quienes durante toda mi formación personal y profesional han sido mi soporte para seguir adelante, sin dejarme decaer en ningún momento; demostrando su amor, corrigiendo mis faltas y celebrando mis triunfos. Son mi ejemplo a seguir, no saben lo muy agradecido que estoy con Dios por tenerlos, los quiero mucho papis.

A la memoria de mi abuelita Marina Montenegro “Mami Mari”, mi segunda madre, a quién la llevo siempre en mi corazón, que desde pequeño me cuidó con un amor inigualable y me enseñó muchas cosas sobre la vida a través de sus consejos y enseñanzas, este logro va para ti.

Mis tíos, Byron Estrella y Cumandá Segovia, mi abuelito Augusto Estrella por estar siempre presentes, acompañándome y brindándome su apoyo para alcanzar la culminación de este proyecto.

Mi querida hermana, Kelly junto a mis adoradas primas, Gaby, María José, Nayely y la pequeña Monserrath quienes son parte fundamental en mi vida y estuvieron conmigo durante toda esta etapa.

¡Gracias amada familia!

José Alexis Chacón Estrella

AGRADECIMIENTO

Agradezco a Dios por haberme dado la vida y la salud para poder culminar con este proyecto, a toda mi familia por estar siempre presentes. A mis compañeros de curso con los cuales compartimos experiencias inigualables, además, agradezco a todos los docentes de la Escuela de Ingeniería en Sistemas quienes impartieron sus conocimientos, consejos, palabras de aliento durante mi formación universitaria en especial agradezco al Ing. Santiago Acuario quien me proporcionó toda su ayuda posible para llevar a cabo este proyecto además de ser un docente excepcional a lo largo de mi carrera universitaria, es un gran amigo, mil gracias Ing. Santi.

José Alexis Chacón Estrella

RESUMEN

El presente proyecto de investigación tiene como objetivo el desarrollo de la agenda académica móvil para los estudiantes de la Pontificia Universidad Católica del Ecuador Sede Ambato para dispositivos Android, para lo cual se fundamenta teórica y metodológica la solución al problema; se realiza un análisis estructural y funcional de otras aplicaciones que incursionan en el ámbito educativo; y se integra la aplicación con la plataforma educativa Moodle. La metodología escogida para el desarrollo es Mobile-D. Con sus fases: definición de los requerimientos funcionales; preparación del ambiente de trabajo del que se obtiene la codificación; las pruebas de funcionamiento, para finalmente obtener el producto terminado: "Mi Agenda". La agenda académica móvil constituye un instrumento de comunicación entre el docente y el estudiante, pues, por estar conectada a la plataforma educativa Moodle, brinda un inicio de sesión con el mismo usuario y contraseña del estudiante registrado. Cuenta con un sistema de notificaciones automáticas que permite al estudiante recordar las tareas que son activadas por el docente, además el estudiante puede visualizar detalladamente las tareas designadas de cada materia y registrar anotaciones con imágenes.

Palabras clave: agenda Educativa, integración Moodle, desarrollo móvil

ABSTRACT

The aim of this study is to develop a mobile academic agenda for the students at the Pontifical Catholic University of Ecuador in Ambato for Android devices. To this end, the solution to the problem is theoretically and methodologically substantiated. A structural and functional analysis of other accessible applications in education is carried out and the application is integrated into the Moodle platform. The chosen methodology for this development is Mobile-D which makes it possible to define functional requirements, prepare the work environment from which the coding is obtained and execute test runs to result in a final product named My Agenda. The mobile academic agenda is a communication tool between professor and student. Since it is connected to the Moodle platform, it uses the same username and password of the enrolled student to log in. It has an automatic notification system that reminds the student about all the professor-activated assignments. The student can also visualize extensively the designated assignments of each subject and register entries with images.

Key words: educational agenda, Moodle integration, mobile development

TABLA DE CONTENIDO

DECLARACIÓN Y AUTORIZACIÓN.....	iii
DEDICATORIA	iv
AGRADECIMIENTO	v
RESUMEN.....	vi
ABSTRACT	vii
INTRODUCCIÓN.....	1
CAPÍTULO I FUNDAMENTOS TEÓRICOS.....	2
1.1. Antecedentes	2
1.2. Problema	3
1.2.1. Descripción del problema.....	3
1.2.2. Preguntas básicas.....	3
1.3. Justificación.....	4
1.4. Objetivos	4
1.4.1. General	4
1.4.2. Específicos.....	4
1.5. Fundamentos teóricos.....	5
1.5.1. Agenda.....	5
1.5.2. Agenda académica.....	8
1.5.3. Agendas móviles	8
1.5.4. Agendas móviles académicas.....	9
1.5.5. PUCE Ambato.....	11
1.5.6. Estudiantes	13
1.5.7. Estudiantes universitarios	13
1.5.8. Aplicaciones estudiantiles universitarias en la PUCE Ambato	14
1.5.9. Sistemas Operativos	18
1.5.10. Sistemas operativos móviles.....	21
1.5.11. Aplicaciones móviles.....	23
1.5.12. Características de las aplicaciones móviles Android.....	24
1.5.13. Entorno de desarrollo de apps móviles	25
1.5.14. Ionic Framework.....	27
1.5.15. HTML5	29
1.5.16. Css3.....	30
1.5.17. JavaScript	31

1.5.18. Node.js	32
1.5.19. AngularJS.....	32
1.5.20. Apache Cordova	34
1.5.21. Base de datos	35
1.5.22. Sistema de gestión de bases de datos (SGBD)	36
CAPÍTULO II METODOLOGÍA	38
2.1. Metodología de investigación	38
2.1.1. Investigación de campo.....	38
2.1.2. Método analítico.....	38
2.1.3. Técnicas de recolección de información	38
2.1.4. Instrumentos	38
2.1.5. Población	38
2.2. Metodología de desarrollo	40
2.2.1. Definición	40
2.3. Metodología Mobile-D	40
2.4. Fases de la Metodología	40
2.4.1. Exploración	41
2.4.2. Inicialización.....	42
2.4.3. Producción	42
2.4.4. Estabilización	43
2.4.5. Pruebas.....	43
CAPÍTULO III DESARROLLO.....	44
3.1. Fase de Exploración.....	44
3.1.1. Definición de partes interesadas	44
3.1.2. Encuesta	44
3.1.3. Análisis de las encuestas.....	44
3.1.4. Definición del alcance	55
3.1.5. Establecimiento del proyecto	55
3.2. Fase de inicialización	60
3.2.1. Preparación de ambiente	60
3.2.2. Entrenamiento.....	65
3.3. Fase de producción	70
3.3.1. Arquitectura del aplicativo	71
3.3.2. Diagrama entidad-relación de Moodle.....	72

3.3.3. Definición del login y creación de funciones principales.....	77
3.3.4. Menú de navegación.....	83
3.3.5. Lista de materias que cursa el estudiante.....	83
3.3.6. Lista de tareas.....	88
3.3.7. Anotaciones.....	92
3.3.8. Notificaciones <i>push</i>	100
3.4. Fase de Estabilización.....	106
3.5. Fase de pruebas.....	112
3.5.1. Emulador de aplicaciones Ionic.....	112
3.5.2. Generación del apk.....	113
3.5.3. Pruebas de caja negra.....	114
CAPÍTULO IV ANÁLISIS DE RESULTADOS.....	121
4.1. Resultado de la prueba a los estudiantes.....	121
4.2. Resultado de la prueba al docente.....	125
CAPÍTULO V CONCLUSIONES Y RECOMENDACIONES.....	126
4.3. Conclusiones.....	126
4.4. Recomendaciones.....	128
BIBLIOGRAFÍA.....	129
ANEXOS.....	133

TABLA DE GRÁFICOS

Figuras

Figura 1.1: Moodle PUCE Ambato.....	15
Figura 1.2: Sistema de Gestión Universitaria – ACADEMICS	16
Figura 1.3: Menú del Academics.....	17
Figura 1.4: Ubicación del Sistema Operativo	19
Figura 1.5: Funcionamiento de la arquitectura MVC	33
Figura 1.6: Contenedor Apache Cordova.....	35
Figura 1.7: Sistema de gestión de base de datos	36
Figura 2.1: Fases de la Metodología Mobile-D.....	41
Figura 3.1: Información del entorno de trabajo utilizado.....	59
Figura 3.2: Interfaz de Notepad++	60
Figura 3.3: Versión de Node.js para los diferentes sistemas operativos	61
Figura 3.4: Asistente de instalación Node.js	62
Figura 3.5: Selección de carpeta donde se instala Cordova	63
Figura 3.6: Instalación Cordova	63
Figura 3.7: Instalación de Ionic	64
Figura 3.8: Página de descarga Android Studio.....	64
Figura 3.9: Generación de iconos y SplashScreen	66
Figura 3.10: Ícono de la aplicación móvil	67
Figura 3.11: Pantalla inicial (SplashScreen).....	67
Figura 3.12: Estructura del proyecto Ionic.....	68
Figura 3.13: Arquitectura de una aplicación en Ionic.....	71
Figura 3.14: Modelo entidad-relación de Moodle. Parte 1	73
Figura 3.15: Modelo entidad-relación de Moodle. Parte 2.....	74
Figura 3.16: Modelo entidad-relación de Moodle. Parte 3.....	74
Figura 3.17: Modelo entidad-relación de Moodle. Parte 4.....	75
Figura 3.18: Modelo entidad-relación de Moodle. Parte 5.....	76
Figura 3.19: Inicio de sesión de la aplicación móvil.....	78
Figura 3.20: Código de configuración inicial de la aplicación	79
Figura 3.21: Código de la pantalla inicio de sesión	80
Figura 3.22: Código de controlador que permite realizar el login en la app.....	81
Figura 3.23: Código PHP para la validación del usuario	82
Figura 3.24: Pantalla de menú principal.....	83
Figura 3.25: Pantalla lista de materias del estudiante	84
Figura 3.26: Código de la pantalla lista de materias.....	85
Figura 3.27: Código del controlador de la pantalla lista de materias	86
Figura 3.28: Código PHP de acceso a la base de datos para la lista de materias...87	
Figura 3.29: Pantalla lista de tareas.....	88
Figura 3.30: Código del controlador de la pantalla lista de tareas	89
Figura 3.31: Código PHP de acceso a la base de datos para la lista de tareas	91
Figura 3.32: Pantalla de anotaciones.....	92
Figura 3.33: Código del controlador para las anotaciones	96
Figura 3.34: Código PHP para guardar las notas.....	98
Figura 3.35: Código PHP para visualizar las notas	99
Figura 3.36: Código PHP para borrar las notas	100
Figura 3.37: Importación de servicio OneSignal.....	102

Figura 3.38: Código para utilizar las notificaciones	103
Figura 3.39: Código PHP de las notificaciones	104
Figura 3.40: Notificaciones	105
Figura 3.41: Vista del emulador para realizar las pruebas	113
Figura 3.42: Carpeta donde se genera el apk	114

Tablas

Tabla 1.1: Características de los frameworks	26
Tabla 3.1. Estructura de un proyecto en Ionic.....	70
Tabla 3.2: Tablas y campos usados en la aplicación móvil	77
Tabla 3.3: Test de mejora inicio de sesión	107
Tabla 3.4: Test de mejora materias	108
Tabla 3.5: Test de mejora tareas	110
Tabla 3.6: Test de mejora notas	111
Tabla 3.7: Test de mejora notificaciones.....	112
Tabla 3.8: Formato de prueba para los estudiantes	116
Tabla 3.9: Formato de prueba para el docente. Parte 1	117
Tabla 3.10: Formato de prueba para el docente. Parte 2	118
Tabla 3.11: Formato de prueba para el docente. Parte 3	118
Tabla 3.12: Formato de prueba para el docente. Parte 4	120
Tabla 3.13: Formato de prueba para el docente. Parte 5	120
Tabla 4.1: Resultado de prueba al docente.....	125

Gráficos

Gráfico 3.1: Frecuencia que se utiliza el teléfono móvil.....	45
Gráfico 3.2: Sistema operativo más utilizado	46
Gráfico 3.3: Frecuencia con que acceden a la Plataforma Educativa.....	47
Gráfico 3.4: Dispositivo que usan para acceder a la Plataforma Educativa.....	48
Gráfico 3.5: Frecuencia del uso de agendas móviles	49
Gráfico 3.6: Medio para recordar tareas pendientes	50
Gráfico 3.7: Beneficio de la aplicación móvil	51
Gráfico 3.8: Aspecto funcional	52
Gráfico 3.9: Posibles funciones de la aplicación móvil	53
Gráfico 3.10: Posibles factores en la aplicación móvil.....	54
Gráfico 4.1: Resultado ámbito estructura de la aplicación.....	121
Gráfico 4.2: Resultado ámbito operación de la aplicación.....	122
Gráfico 4.3: Resultado ámbito apariencia	123
Gráfico 4.4: Resultado ámbito contenido	124

INTRODUCCIÓN

El proyecto de investigación surge en la Pontificia Universidad Católica del Ecuador (PUCE) Sede Ambato como respuesta al objetivo general: desarrollar la agenda académica móvil para los estudiantes de la PUCE Ambato basada en el sistema operativo Android. La aplicación móvil mediará la comunicación entre el docente y el estudiante, puesto que permite visualizar la lista de materias académicas con sus respectivas tareas y, entre otras funciones, cuenta con un sistema de notificaciones automáticas para recordar al estudiante su tarea cuando es activada por el docente.

El trabajo se documenta por capítulos, los cuales brevemente se describen:

Capítulo I.- Se detallan los antecedentes, problema, justificación, objetivos; adicionalmente se presenta la fundamentación teórica.

Capítulo II.- Se da a conocer la metodología que se aplica en el proyecto, así como también una descripción de lo que se hace en cada una de sus fases.

Capítulo III.- Se desarrolla el proyecto sobre la base de la metodología, se puntualiza la codificación de las funciones de cada pantalla y se muestran los resultados obtenidos de la aplicación. Posteriormente se hace constar las pruebas de mejora por cada módulo. Se genera el instalador (apk) y los formatos de la evaluación para ser aplicada a los estudiantes y docente.

Capítulo IV.- Se evidencian los resultados obtenidos de las pruebas con su respectiva tabulación e interpretación de datos.

En la parte final se encuentran las conclusiones y recomendaciones del proyecto, seguido de la bibliografía y los anexos.

CAPÍTULO I

FUNDAMENTOS TEÓRICOS

1.1. Antecedentes

Dentro del ámbito educativo, la agenda académica es un medio en el cual el estudiante registra actividades a cumplirse en posteriores tiempos, convirtiéndose en un recurso informativo y recordatorio que facilite su desempeño (Aragón, 2010).

Durante los últimos años, las aplicaciones han tenido una alta acogida en los dispositivos móviles, según se calculaba a finales del año 2014 el número de éstos superaba la cifra de la población mundial (Cisco, 2015). Las posibilidades que los actuales móviles poseen, permiten la fácil accesibilidad de la información desde cualquier lugar (TicBeat, 2011). Así como los ordenadores cuentan con sus propios programas, los móviles cuentan con aplicaciones, las cuales están direccionadas tanto al entretenimiento como a mejorar la productividad personal, especialmente en el campo académico; estas aplicaciones están desarrolladas para determinados sistemas operativos como: Android, IOS, Windows Phone.

En este proyecto se desarrollará la agenda académica móvil, la cual utiliza la base de datos de la plataforma Moodle para acceder y obtener información de los cursos del estudiante; con esto, la aplicación permitirá visualizar los deberes asignados de cada materia; la misma que cuenta con un sistema de notificaciones automático, de tal manera que el estudiante se mantenga informado y adicionalmente tenga la posibilidad de guardar recordatorios como notas y fotos.

1.2. Problema

1.2.1. Descripción del problema

Actualmente los estudiantes de la PUCE Ambato para recordar tareas que deben cumplir en determinado tiempo, acuden a la plataforma Moodle, en ocasiones utilizan un cuaderno para tomar notas o registrar sus deberes.

Los estudiantes, no cuentan con un medio de fácil acceso al sistema académico, que a corto o largo plazo tienen las siguientes consecuencias:

- Desinformación acerca de sus calificaciones, presentación de tareas y horario de sus respectivas clases.
- Incomodidad de los estudiantes para acceder al sistema.
- Efectos negativos en su rendimiento académico.

1.2.2. Preguntas básicas

¿Por qué se origina?

La inexistencia de una agenda académica desarrollada como aplicación móvil.

¿Cuándo se origina?

Se origina cuando el estudiante, en el proceso educativo, tiene su información académica dispersa en: apuntes, plataforma Moodle, fotografías, hojas sueltas, etc.

¿Dónde se detecta?

Mediante observación directa se ha detectado que los estudiantes de la PUCE Ambato, no logran cumplir sus tareas en los tiempos y plazos indicados debido

a la falta de comunicación directa y pertinente, que genera dificultades en su desempeño académico.

1.3. Justificación

El presente proyecto de investigación, tiene como propósito el desarrollo de una agenda académica para el beneficio de los estudiantes de la PUCE Ambato, ya que les permite mantenerse informados con las tareas que deben cumplir en un tiempo previsto; esto facilita recordar y gerenciar mejor el tiempo para presentar sus tareas en los plazos indicados; además de ahorrar el tiempo de consulta mediante la aplicación, el estudiante tiene la posibilidad de crear anotaciones, de esta manera puede tener información más detallada sobre la actividad.

Cabe mencionar que la aplicación es para dispositivos móviles, la cual se desarrolla para el sistema operativo Android. A sabiendas de que en la actualidad la mayor parte de estudiantes, cuentan con un dispositivo de este tipo y lo utilizan en cualquier momento; es por ello que resulta adecuado realizar el presente proyecto en plataforma móvil.

1.4. Objetivos

1.4.1. General

Desarrollar la agenda académica móvil para los estudiantes de la PUCE Ambato basada en el sistema operativo Android.

1.4.2. Específicos

- Fundamentar teórica y metodológicamente sobre software que permita el desarrollo de la aplicación en el sistema operativo Android.

- Sintetizar las diferentes aplicaciones sobre agendas estudiantiles disponibles en el mercado.
- Integrar la plataforma estudiantil Moodle con la aplicación para poder acceder a la información de los estudiantes.
- Realizar pruebas de la aplicación en diferentes dispositivos móviles Android, de los estudiantes.

1.5. Fundamentos teóricos

1.5.1. Agenda

1.5.1.1. Definición

Según la (Real Academia Española, 2017) la palabra agenda proviene del latín “lo que ha de hacerse”. Es un cuaderno, dispositivo electrónico o libro, donde las personas apuntan actividades para poder recordarlas, las cuales pueden ser: asuntos, compromisos, quehaceres previamente ordenados, que deben cumplirse en determinados períodos de tiempo. Además, se menciona que la agenda es un conjunto de temas que deben ser tratados en una junta.

Por otra parte, la agenda también es considerada como una manera de mantener controlada la empresa; con su ayuda se puede organizar y gestionar diferentes procesos en tiempos establecidos, así como también conocer las actividades que debe realizar cada persona o grupo en la empresa (Hammer, 2002).

1.5.1.2. Tipos de agenda

El autor (Madiedo, 2014) concluye que existen tres tipos de agendas, cada una con características de uso diferentes que deben ser utilizadas dependiendo cuál se adapte mejor a la necesidad y comodidad personal:

Agenda manual

Es la agenda tradicional en formato papel, dentro de la misma existe una hoja para cada día del año en donde se puede anotar los asuntos que se desean recordar. Este tipo de agendas son utilizadas por los estudiantes durante todo el año escolar.

Al día de hoy, existen agendas electrónicas que cada vez van ganando interés; sin embargo, las agendas manuales aún siguen siendo utilizadas.

La ventaja de dichas agendas es que permiten acceder de forma rápida y fácil al contenido apuntado, pero a la vez pueden ser vulnerables ya que no cuenta con sistemas de seguridad que impidan el acceso de información confidencial, también no cuenta con recordatorios de eventos o alarmas.

Agenda electrónica

También conocidas como PDA (Personal Digital Assistant), son agendas que en un espacio reducido almacenan mucha información. En la actualidad tienen funciones de un ordenador personal como: creación de documentos, acceso a internet, reproducción de archivos multimedia, correo electrónico, calendario, entre otras.

Las principales características de este tipo de agendas son:

- Registra una amplia cantidad de contactos.

- Como es electrónica, conserva su fácil portabilidad y acceso con relación a la agenda tradicional.
- Cuenta con un sistema de alarma que permite recordar: fechas, eventos o tareas pendientes.
- Por su versatilidad es posible modificar cualquier actividad programada.
- Cuenta con claves de acceso para mediatizar la información a personas no autorizadas.

Una de las principales desventajas de esta agenda, es la pérdida de información por algún desperfecto, si no existe una respectiva copia de seguridad.

Ordenador

Con la ayuda de este tipo de agendas fácilmente se puede organizar las tareas a corto, medio y largo plazo, pues cuenta con un visor de actividades donde se las ve en conjunto, estructuradas por día, semana o mes; también cuenta con recordatorios de las mismas.

Las aplicaciones informáticas con mayor renombre son: Microsoft Outlook y el calendario de Google.

1.5.2. Agenda académica

1.5.2.1. ¿Qué es una agenda académica?

La agenda académica es un instrumento informativo que permite vincular todo lo referente a la evolución del alumno en sus estudios, hacia sus padres (Albaina & Aranda, 2002).

Por otra parte (Gomez, Uruñuela, & Rafael, 2008) señalan que la agenda académica a más de ser un instrumento organizacional del estudiante para la coordinación, comunicación y relación con la comunidad educativa, es donde se planifica día a día las actividades didácticas que realizan los estudiantes en la institución, con el fin de mantener informadas a las familias de los estudiantes.

1.5.2.2. ¿Para qué sirve la agenda académica?

La agenda académica permite al estudiante tomar apuntes sobre las tareas o actividades que debe cumplir para un determinado tiempo; con esto, el estudiante puede organizar las tareas, cómo y cuándo se llevarán a cabo. También dicha agenda ayuda al educador, pues no tiene que mandar hojas informativas sobre lo que debe cumplir el estudiante, cabe recalcar que las agendas sirven también para mantener informada a la familia del estudiante, una herramienta de comunicación intermediaria en lo que debe cumplir e implicarse (Guitart, 2002).

1.5.3. Agendas móviles

Los autores (Octavian, Nguyen, Tam, & Wang, 2009); (Hewlett-Packard Company, 2016) concuerdan que las PDA del inglés Personal Digital Assistant, son pequeños dispositivos portátiles diseñados como un

organizador, los cuales poseen funcionalidades similares a las que realiza un computador. Cuenta con un procesador, memoria RAM, espacio de almacenamiento, funciones de red integradas, su pantalla puede ser sensible al tacto o en teclado, o ambos.

Inicialmente estos dispositivos presentaban características básicas de una agenda electrónica como: calendario, calculadora, lista de contactos, bloc de notas y recordatorios. Con el pasar del tiempo la tecnología ha ido creciendo a gran escala, ampliando de manera sustancial las funcionalidades de las PDA sin afectar su portabilidad, es así que en la actualidad constan de una alta variedad de aplicaciones de procesamiento de texto, acceso al correo electrónico, hojas de cálculo, corrección ortográfica, diccionarios, juegos, hasta la posibilidad de crear documentos, navegar por internet o la reproducción de música y video.

La llegada de los teléfonos inteligentes al mercado, supuso un gran incremento de nuevos competidores a este sector, facilitando el desarrollo de diversos sistemas operativos para las PDA como: IOS, Android, Symbian OS, MeeGo. Sin embargo, existen dos de ellos que fueron los pioneros en el desarrollo y crecimiento de las PDA; PalmOS que fue comercializado por la empresa Palm Inc. y Pocket PC comercializado por Microsoft.

Cabe mencionar que hoy en día la mayoría de las PDA son teléfonos inteligentes ya que poco a poco han sido reemplazados por estos.

1.5.4. Agendas móviles académicas

Para (García, 2004) las funciones que posee una PDA han permitido el aumento de la productividad, puesto que ayudan a realizar múltiples

operaciones dentro de diferentes campos, como el educativo. Es así que son utilizadas tanto por profesores como estudiantes, por ejemplo, en Educación Física los profesores deben realizar anotaciones de las actividades del alumnado, posteriormente procesar la información teniendo en cuenta que imparten clase a una multitud de estudiantes, con la incorporación de la PDA en este campo, pueden hacerlo simultáneamente, lo que resulta un ahorro de tiempo y trabajo.

El uso de registros convencionales como libretas y cuadernos, por más personalizados que estén para el uso del profesor, presentan muchas limitaciones por el simple hecho de hacer cuantiosas operaciones y esto a largo plazo complica el proceso haciéndolo lento, tedioso y con probabilidad de error.

Ahora ya es posible con la ayuda de un computador que cabe en la palma de la mano, anotar todo lo que suceda en clase al instante usando aplicaciones ofimáticas; las que además de servir para evaluar la clase, como por ejemplo cronometrar y guardar los tiempos parciales y finales de los alumnos en una hoja de Excel, pueden utilizar un procesador de texto en Word para describir la clase impartida y/o tenerla como recordatorio.

Al respecto (Herrera, 2003) señala lo importante que están siendo estos dispositivos en las matemáticas, pues en la actualidad están en desarrollo programas de álgebra, donde no se trata de resolver el ejercicio automáticamente solo introduciendo valores, sino que el estudiante indique los pasos que se deben aplicar para solucionarlo, es quien dirige el proceso.

La finalidad es incrementar la habilidad mental para calcular y estimar, con el uso de la tecnología en la enseñanza.

El autor (Roschelle, 2003) concluye que el impacto que ha tenido la tecnología de las PDA en la educación es notorio, cada vez su uso es obligatorio en el aula. En un estudio realizado por el investigador a los profesores, afirma que ofrecen herramientas de instrucción que tienen un gran valor en el aprendizaje de los estudiantes.

Además de la interconectividad y al ser un dispositivo móvil, permite el acceso de aplicaciones para el aprendizaje desde cualquier lugar y hora. Los estudiantes adquieren mayor conocimiento e incremento de su desempeño académico, con el uso de estos dispositivos portátiles.

Por otra parte, el costo de las PDA para la mayor parte de los estudiantes e instituciones educativas es inaccesible.

1.5.5. PUCE Ambato

La información acerca la universidad fue obtenida de los sitios web (Pontificia Universidad Católica del Ecuador, 2017); (Formacion.ec, 2017) donde se describe lo siguiente:

1.5.5.1. Historia

El 13 de Enero de 1982 empezaron las gestiones de la Cámara de Comercio de Ambato y de Monseñor Vicente Cisneros, Obispo de Ambato, para conseguir la creación de una sede la PUCE en Ambato. Así, el 15 de julio de 1986 se emitió la resolución para que funcione el Programa de Extensión de Estudios de la PUCE en la ciudad de Ambato.

Al poco tiempo se crearon los programas de Tecnología Médica e inglés, con gran acogida de los estudiantes.

El 10 de febrero de 1989, el H. Consejo Superior de la PUCE expidió la estructura administrativa para el funcionamiento del Programa Académico de la PUCE en la ciudad de Ambato. Como consecuencia, el Consejo Superior aprobó la creación de los Programas de Optometría e Informática.

Con todos estos antecedentes, el Consejo Nacional de Universidades y Escuelas Politécnicas, en su sesión de 29 de julio de 1993, aprobó la creación y funcionamiento de la Extensión Universitaria.

El 16 de marzo de 1994, la extensión se transforma en Sede y por lo tanto se realiza la suscripción del convenio entre la PUCE y la Diócesis de Ambato a fin de que funcione la Sede en ésta ciudad.

Misión:

La PUCESA, es una comunidad académica y de servicio que forma parte del Sistema Nacional de la PUCE, se inspira en los principios cristianos, propicia el diálogo de las diversas disciplinas del saber con la fe, promueve la generación y desarrollo del conocimiento científico y tecnológico, contribuye a la formación humanista, profesional, cultural y al bienestar de sus integrantes, para satisfacer las necesidades de la colectividad.

Visión:

La PUCESA se consolidará como la institución de educación superior que mantiene su liderazgo con una visión integradora en la formación humanista, profesional y cultural, basada en la búsqueda de la verdad científica y tecnológica y el fortalecimiento de su identidad cristiana, para contribuir al desarrollo social y al buen vivir.

1.5.5.2. Carreras de la PUCE Ambato

Actualmente la Pontificia Universidad Católica del Ecuador oferta los siguientes programas de pregrado:

- Carrera de Administración
- Carrera de Diseño Industrial
- Carrera de Jurisprudencia
- Carrera de Psicología
- Carrera de Sistemas

Así como también cuenta con programas de postgrados tales como:

- Maestría en Innovación en Educación

1.5.6. Estudiantes

De acuerdo (Campelo & Viel, 2009) ser estudiante es una construcción compleja; no se crea de un día al otro, sino que en el transcurso de toda la vida escolar va adquiriendo rasgos que lo forman. Esto quiere decir que el estudiante, conforme supera los diferentes niveles educativos, desarrolla habilidades, capacidades, destrezas y conocimientos.

Cabe mencionar que es diferente un estudiante de primaria, secundaria o universitario; en efecto, cada vez el nivel de estudio va incrementando para los jóvenes, los cuales tienen que afrontar nuevos desafíos para permanecer en las instituciones educativas y posteriormente llegar a ser profesionales.

1.5.7. Estudiantes universitarios

Para (Guzmán, 2013) el estudiante universitario es un individuo que ha madurado el perfil personal progresivamente durante su vida estudiantil, a tal punto de cambiar la actitud, proyectándose un futuro profesional y metas que alcanzar.

Los elementos clave que el estudiante tiene en cuenta para que resulte mejor su vida universitaria son:

- Tiempo: necesita más dedicación una carrera universitaria, ya que no va ser lo mismo que los estudios escolares, será un cambio muy considerable.

- Organización personal: ordena actividades tanto cotidianas como académicas según su prioridad.
- Hábitos y técnicas de estudio: como el estudiante se enfrenta a una nueva realidad, tiene que adoptar nuevas costumbres de estudio para que no decaiga su rendimiento.

El estudio universitario exige responsabilidad y esfuerzo personal por quien desea superarse en la vida, pero el estudiante no está solo en estos cambios. La universidad aporta con varias maneras como: tutores que asesoran y orientan académicamente con la finalidad de que el estudiante no tenga problemas de rendimiento, además de potenciar las herramientas intelectuales, las competencias y valores con los que el universitario servirá profesionalmente a la sociedad.

1.5.8. Aplicaciones estudiantiles universitarias en la PUCE Ambato

La universidad, en el que hacer educativo, cuenta con dos plataformas que facilitan la interacción didáctica y la gestión académica, las cuales son utilizadas por los docentes y estudiantes frecuentemente.

El Moodle PUCE Ambato o anteriormente conocido como EVAE, está basado en Moodle; según (Moodle, 2017) es una plataforma educativa virtual de enseñanza y aprendizaje, la cual permite a los educadores crear cursos en línea y agregar contenido fácilmente, además de contar con una interfaz sencilla e intuitiva que despierta interés en los estudiantes.



Figura I.1: Moodle PUCE Ambato

Fuente de: (Pontificia Universidad Católica del Ecuador Ambato, 2018)

Para (Peña, 2013) esta plataforma al ser de código abierto e impulsada por una comunidad global ofrece constantes actualizaciones, y en cada una de ellas nuevas características de las que podemos resaltar la siguientes:

- Dispone de un menú de actividades que el educador puede incluir dentro de los cursos como: foros, tareas, glosarios, talleres, lecciones. Esto permite que el aprendizaje sea interactivo, es decir: entre el alumno y la plataforma, ya que ofrece retroalimentación al estudiante como cuestionarios o documentos para reforzar el conocimiento, además de la creación de foros útiles para la interacción entre los estudiantes.
- Con el objetivo de estimular sensorialmente el aprendizaje en los estudiantes, Moodle permite a los educadores subir contenidos multimedia: videos, imágenes, audios, etc.

- En cuanto a personalización: Moodle es flexible, por lo que brinda al educador diseñar su curso como guste, es decir, cambiar de apariencia, añadir actividades y ordenarlas, así mismo habilitar o deshabilitar cada una de ellas a los participantes durante un período de tiempo, establecer periodos predeterminados de publicación de contenidos. Todo esto es de gran ayuda al educador ya que puede adecuar la plataforma según el nivel, objetivos, intereses de los participantes.
- Moodle promueve el autoaprendizaje, el estudiante es quien elige el momento, modo que usará la plataforma para realizar las actividades propuestas, reforzar conocimientos aprendidos en la clase, repasar contenidos, entre otras.

Por otra parte, se encuentra funcionando un Sistema de Gestión Académica conocido como Academics, el cual es accesible para el personal administrativo, docente y estudiantil de la Institución.



Figura I.2: Sistema de Gestión Universitaria – ACADemics

Fuente de: (Pontificia Universidad Católica del Ecuador Ambato, 2018)

Este sistema cuenta con varias funciones para los usuarios, las cuales están organizadas en un menú:

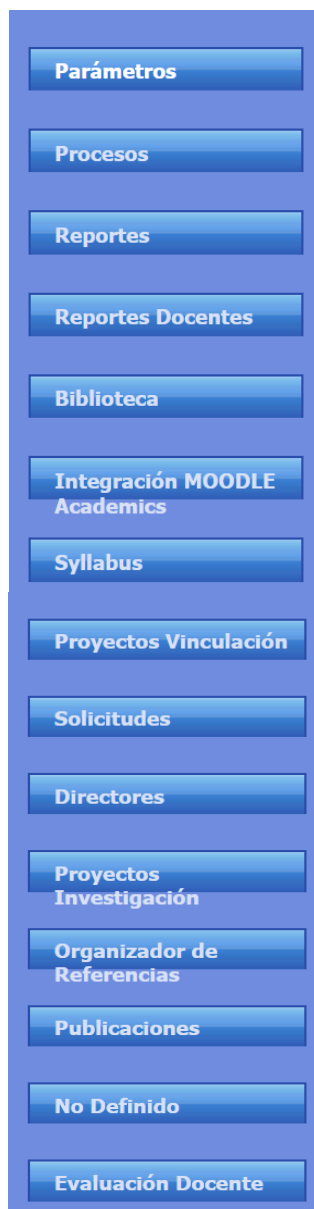


Figura I.3: Menú del Academics

Fuente de: (Pontificia Universidad Católica del Ecuador Ambato, 2018)

- A través del Academics los estudiantes aprueban sus hojas de aranceles, que, en sí, muestran el valor a cancelar por semestre. En cuanto a la parte administrativa facilita el proceso de matriculación, ya que el sistema se encarga de organizar los planes de estudio de

acuerdo a cada estudiante y envía las hojas de aranceles a sus respectivas escuelas para que sean aprobadas, por lo que ayuda a optimizar dicho proceso y evita que la matriculación sea presencial.

- Permite al estudiante conocer el contenido que se trató durante toda la semana a través del seguimiento del Syllabus, en donde se indica el tema a tratar, la semana correspondiente, las actividades a realizar, las estrategias de enseñanza y el resultado del aprendizaje, además de calificar el grado de cumplimiento de los mismos.
- Facilita el proceso de heteroevaluación, en el cual los estudiantes contestan preguntas objetivas para evaluar a los docentes.
- Cuenta con una lista de solicitudes predeterminadas que los estudiantes pueden enviar.
- Permite que los estudiantes consulten las calificaciones obtenidas durante el semestre. Esto es posible gracias a la integración que existe entre el Moodle y el Academics, en donde el educador a través de la opción Integración Moodle-Academics realiza el proceso de transferencia de calificaciones de las diversas tareas generadas en el sistema Moodle.

1.5.9. Sistemas Operativos

Los autores (Flynn & McIver, 2010); (Candela, 2007), concuerdan que un sistema operativo es un programa intermediario que gestiona todo el hardware y software del computador. El principal objetivo de un sistema operativo, es dar un entorno donde el usuario pueda ejecutar de manera fácil y eficiente

programas. Por lo que, si un usuario da una orden, el sistema operativo debe asegurarse que se ejecute, en caso contrario, si sucediera algún error al realizar una operación, debe indicar al usuario que la orden no fue ejecutada y el respectivo error.

Según (Tanenbaum, 2009) los componentes principales que interactúan con el sistema operativo, como se muestra en la figura 4 son: el hardware que consiste en el teclado, mouse, monitor, impresora, discos, tarjetas, circuitos integrados, en sí, los objetos físicos. En la parte superior del hardware se encuentra el software. Existen dos modos de operación del computador: el modo kernel o privilegiado con el cual se ejecuta el sistema operativo, con este modo se tiene acceso total y se puede ejecutar cualquier instrucción, utilizado por usuarios avanzados a través de la consola. Por otra parte, el modo usuario solo permite un limitado conjunto de instrucciones y es el software, con interfaz gráfica, que usa la mayor parte de usuarios.

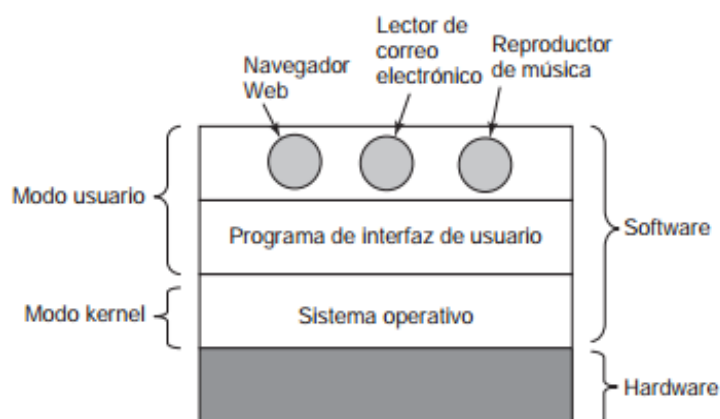


Figura I.4: Ubicación del Sistema Operativo

Fuente de: (Tanenbaum, 2009)

Además, existen varios sistemas operativos que se adaptan según las necesidades de los usuarios, entre los más conocidos están:

- Windows: Utilizado por la mayor parte de usuarios, brinda varias versiones, fue creado por la empresa Microsoft y es de licencia de pago.
- Mac OS: Propiedad de la empresa Apple de uso exclusivo para sus propios productos.
- Linux: Sistema operativo de código abierto, por lo que es gratis y puede ser modificado a conveniencia, a más de ser seguro.
- Unix: Usado en las supercomputadoras de grandes empresas, posee una seguridad alta.

Para (Fernández, 2015) los sistemas operativos se pueden clasificar:

- **Multiusuario:**

Sistemas operativos que permiten que dos o más usuarios puedan utilizar simultáneamente los programas.

- **Multitarea:**

Estos sistemas operativos permiten que varios programas se ejecuten a la vez.

- **Multiprocesador:**

Son aquellos sistemas operativos que permiten abrir el mismo programa, pero en diferentes CPUs.

1.5.10. Sistemas operativos móviles

Al igual que los sistemas operativos para computadoras, el de los móviles facilita la interacción entre el hardware, software y el usuario al ejecutar trabajos con un recurso tecnológico como el smartphone; por ende (Malave & Beauperthuy, 2011) consideran al sistema operativo para móviles, como el programa principal que es capaz de gestionar todo los recursos que posee el dispositivo, de manera que facilite al usuario o programador las herramientas o interfaces necesarias para llevar a cabo una acción.

De acuerdo (Baz, Ferreira, Álvarez, & Garcia, 2013) existen varios sistemas operativos móviles en el mercado, pero cada uno cuenta con distintas características y formas de interactuar con el usuario. Entre los más importantes se encuentran: IOS de la compañía Apple, Android creado por Google, Windows Phone por Microsoft y otros que en el transcurso del tiempo dejaron de existir.

Android

El sistema operativo Android es uno de los más populares a nivel mundial, fue creado por Google. Está basado en el núcleo de Linux, diseñado especialmente para teléfonos inteligentes; sin embargo, fue tanto su crecimiento que adaptaron el sistema a otros dispositivos como tablets, televisores, PCs, relojes, entre otros. Cuenta con una tienda donde se puede adquirir aplicaciones, películas y libros que son de pago o gratis (Castillo, 2017).

A diferencia de la competencia, Android es de código abierto, por lo que cualquier persona con conocimientos en programación puede modificar el

código del sistema operativo o crear nuevas aplicaciones, ya que además es una plataforma de software que usa el popular lenguaje JAVA (Báez & Borrego, 2010).

Al igual que los móviles evolucionan, Android también lo hace, es así que tiene una gran variedad de versiones con el objetivo de ir mejorando al sistema, comenzó desde la versión 1.6 llamada Donut hasta la actual 7.1.2 conocida como Nougat (Google, 2017).

iOS

Conocido anteriormente con el nombre de iPhone OS, desarrollado por Apple Inc. Al igual que Android al inicio fue creado para el iPhone, posteriormente fue adaptado para los demás dispositivos de la compañía.

Lo que destaca de este sistema operativo, es su simplicidad y fluidez con respecto a los otros, además de contar con una tienda que ofrece aplicaciones con interfaz elegantes las cuales se pueden adquirir gratis o mediante pago. Todo esto hace que la mayoría de personas usen estos dispositivos a pesar de no contar con características de las más potentes, su principal fuerte está en la optimización entre hardware y software haciéndolo casi perfecto (Apple Inc, 2016).

Windows Phone

Es un sistema relativamente nuevo que ha ido creciendo a grandes pasos, lanzado al mercado en 2014 por Microsoft, basado en el núcleo Windows Embedded CE 6.

Con la unión que tuvo con el fabricante de celulares Nokia dio a conocer sus primeros Smartphones bajo el nombre Luminia, lo cual ha llevado a conocerse mundialmente pero aún no tiene tanta acogida como Android o iOS (Microsoft, 2016).

1.5.11. Aplicaciones móviles

Según (Cuello & Vittone, 2013) las aplicaciones, conocidas como apps, ya venían instaladas desde años atrás en los sistemas operativos de Nokia o BlackBerry, pero carecían de posibilidades por las características que tenían los móviles de esa época en comparación a los actuales teléfonos.

También argumentan que una aplicación móvil es un software que se ejecuta en un sistema operativo móvil; para entender este concepto hacen referencia a que: así como existen programas para ordenadores, existen las aplicaciones que son para los móviles.

Según (Mobile Marketing Association Spain, 2011) al principio las aplicaciones solo estaban enfocadas a la productividad personal como: alarmas, calendarios, clientes de correo y calculadoras. Hoy en día existe una gran cantidad de aplicaciones para determinados usos, a tal punto de que se puede encontrar las siguientes categorías: multimedia, productividad, entretenimiento, utilidades, comunicaciones, entre otras.

Cada día se desarrollan aplicaciones que ofrecen funcionalidades nuevas y atractivas para los usuarios, ya que en la actualidad existe un nivel considerable del uso de terminales móviles.

1.5.12. Características de las aplicaciones móviles Android

Existen muchos sistemas operativos para móviles, pero, cada uno cuenta con características que los diferencian de los demás, uno de ellos es Android (Gironés, 2013)

- Android es una plataforma de desarrollo libre, basado en Linux por lo que no se debe pagar licencia para poder usar el sistema.
- Es adaptable para cualquier hardware, no fue diseñado solo para teléfonos o tabletas. En la actualidad podemos encontrar una variedad de dispositivos que llevan en su interior el sistema operativo Android, por esta razón exige un mayor esfuerzo para el programador ya que las aplicaciones que desarrolle deben adaptarse al hardware y software de los dispositivos.
- Las aplicaciones Android son desarrolladas en Java, esto es una gran ventaja porque permiten ser ejecutadas en cualquier tipo de CPU.
- Cuenta con una gran cantidad de servicios incorporados como: reconocimiento de voz, multimedia, navegador, base de datos, localización GPS, etc.
- Con respecto a la seguridad, cada aplicación cuenta con una serie de permisos que limitan su participación.
- Android posee una Máquina Virtual Dalvik que fue implementada por Google de la máquina virtual de JAVA, optimizada para dispositivos móviles de baja potencia y poca memoria.

1.5.13. Entorno de desarrollo de apps móviles

Un entorno de desarrollo también llamado framework son estructuras de software que constan de módulos personalizables ya desarrollados, en otras palabras, son códigos comunes que realizan funciones genéricas los cuales podemos conjuntamente acoplarlos para desarrollar una aplicación (Galindo, 2008).

Según (Sánchez, 2006) al usar un framework se quiere que el desarrollador se centre en el verdadero problema que desea solucionar y no desperdicie su tiempo en programar módulos ya existentes, por ende, los principales objetivos de un framework son: la reutilización de código, la optimización de tiempo y la facilidad en el desarrollo de una aplicación.

Existen diferentes frameworks que se ajustan dependiendo al proyecto a realizar, por ejemplo: orientados al desarrollo de aplicaciones web, otros al desarrollo de aplicaciones multiplataforma, así como también para un sistema operativo, etc.

1.5.13.1. Características de los entornos de desarrollo

De acuerdo a (Gutiérrez, 2010) cada framework cuenta con sus propias particularidades; sin embargo, en todos podemos encontrar las siguientes características:

Control y autenticación de acceso	Tiene mecanismos que impiden el acceso a determinadas páginas según el tipo de usuario, además de poseer maneras para autenticarlo mediante login.
-----------------------------------	--

Abstracción de URLs y sesiones	El framework se encarga directamente de manejar las sesiones y URLs.
Controladores	El framework tienen controladores para gestionar eventos, como la inserción de datos a través de formularios o el acceso a diferentes páginas, dichos controladores son adaptables según la necesidad del proyecto.
Acceso a datos	Integran herramientas o interfaces necesarias para archivos XML, txt, BBDD, etc.

Tabla I.1: Características de los frameworks

Fuente de: (Gutiérrez, 2010)

1.5.13.2. Ventajas de los entornos de desarrollo

De acuerdo a (acensTechnologies, 2014) las ventajas de utilizar un framework son:

- Optimización del tiempo

Proporcionan códigos funcionales, por lo cual el desarrollador no tiene que preocuparse en mantener ni reescribir el código, lo que permite reducir el tiempo de desarrollo.

- Reduce riesgos de desarrollo

Los framework proveen mayor seguridad al desarrollar aplicaciones ya que tienen modelos de programación fiables y probados.

- Proporciona una arquitectura consistente entre aplicaciones

Al utilizar un framework las aplicaciones generadas mantienen la misma estructura, lo que permite que sean fácil de proporcionar mantenimiento o soporte.

1.5.13.3. Desventajas de los entornos de desarrollo

Según (acensTechnologies, 2014) afirma que existen las siguientes desventajas al utilizar un framework:

- Falta de flexibilidad

Los componentes que se utilicen, creen o modifiquen, deben limitarse a la arquitectura del framework.

- Dificultad de aprendizaje

Cada framework se aplica de diferente manera por lo que es necesario conocer la documentación para el uso del mismo.

- Creatividad limitada

Puede darse el caso de que el desarrollador no consiga modificar un módulo del framework ya que éste no permita ser reescrito

1.5.14. Ionic Framework

Según (Ionic Framework, 2017) es un framework de código abierto, gratuito, que permite desarrollar aplicaciones híbridas multiplataforma basadas en HTML5, CSS y JavaScript.

Junto a AngularJS y Apache Cordova asegura que las aplicaciones sean escalables y rápidas. Además, cuenta con componentes, hojas de estilo y su propia estructura de aplicación que facilita el desarrollo.

Ionic al estar construido sobre la base de Cordova, puede integrar sus plugins que utilizan funciones nativas como: acceder a fotos, cámara, localización, entre otras.

A continuación, se describe las principales características de Ionic Framework:

- **AngularJS:** Ionic utiliza para el desarrollo de aplicaciones la arquitectura modelo-vista-controlador (MVC) esto permite al desarrollador crear aplicaciones fáciles y de manera organizada.
- **Cordova Plugins:** A través de sus API permite el acceso de funciones nativas, tales como: cámara, sensores, datos, etc.
- **Componentes CSS:** Aportan con los elementos necesarios para la aplicación, además que sus estilos predeterminados pueden ser adaptados a un propio diseño.
- **Componentes JavaScript:** Mediante funciones de JavaScript cubren los elementos dinámicos que no pueden hacerse con HTML5 y CSS.
- **Ionic CLI:** Son los comandos necesarios para iniciar, construir y ejecutar la aplicación.

En conclusión, al utilizar Ionic framework, se puede generar, con el mismo código, aplicaciones para las distintas plataformas móviles; además dispone de una alta cantidad de componentes que permiten, sin tanto esfuerzo, tener una aplicación con apariencia profesional.

1.5.15. HTML5

De acuerdo a (Lancker, 2012) desde los inicios de internet para entregar en la red contenido de una forma amigable y organizada, se creó una versión simple de HTML que permitía generar una estructura sencilla de páginas web, el mismo que los navegadores interpretaban y generaban contenido con diseños y funcionalidades especiales. Como todo lenguaje evoluciona, nació HTML5, una versión mejorada de HTML, la cual combina tecnologías como CSS y JavaScript que aportan nuevas funcionalidades y atributos dando mejores funcionalidades a los aplicativos y portales web.

Características principales:

- Mínimo uso de recursos, evita código redundante.
- El contenido se distribuye y se formatea con HTML5.
- Manejo de contenido multimedia en los sitios web.
- Ya no se depende de módulos o plugins en los portales web.
- Se puede usar en múltiples dispositivos.
- HTML5 permite el uso del funcionamiento OFFLINE.
- Funcionalidad de arrastrar y soltar elementos.
- Uso de geolocalización.
- Nuevas etiquetas para utilizar más funcionalidades para los portales web.

1.5.16. Ccss3

Las siglas CSS significan Cascading Style Sheets que en español se las conoce como hojas de estilo en cascadas, son aquellas que dan vida a los diseños e interactividad en un portal o aplicativo web; nace con el objetivo de separar la presentación y contenido, es decir, todo lo referente al diseño, colores, tipo de letra, formas etc., se lo realiza en un archivo CSS independiente a los archivos HTML relacionados con el contenido de la página (Collel, 2016).

Según (Aubry, 2007) CSS3 es un CSS de nivel 3 con una arquitectura que se divide en varios módulos los cuales permiten acceder a nuevas capacidades y mejoras, las mismas que no afectan a la compatibilidad con antiguas versiones, dispone de más de 20 módulos independientes los unos de los otros disponibles para su uso.

Las principales características del CSS3 son:

- Bordes. - Pueden tener mejores formatos con más colores y formas, lo que permite dar el efecto redondeado a imágenes, botones, etc.
- Fondos. - Mejor precisión a la hora de usar fondos en múltiples capas y redimensionamiento más controlado.
- Color. - Opacidad, gradientes, valores de color HSL
- Texto. - Sombras, Desbordamiento, ajuste de línea.
- Transformaciones. - Escalar, sesgar, mover, rotar en 2D o 3D.
- Transiciones. - Transición sencillas de estilos.

- Cajas. - Sombras, cajas redimensionales, overflow separado en vertical u horizontal, compensación entre entorno y borde, modelos para especificar altura y anchura.
- Contenido. - Los estilos pueden añadir contenido a los elementos.
- Opacidad. - Los elementos pueden ser transparentes.
- Fuentes web. - Capacidad de añadir fuentes más variadas y en tiempo real.

1.5.17. JavaScript

JavaScript es un lenguaje de desarrollo que se ejecuta en el lado del cliente, utilizado principalmente para crear páginas web dinámicas, es decir, páginas que integran efectos, animaciones al pulsar un botón, ventanas de alerta, etc. En conjunto con HTML se puede lograr muchos efectos de alta calidad y funcionalidad, lo que ayuda a los usuarios a interactuar eficazmente en las páginas web (Prescott, 2017).

Según (Mozilla, 2018) afirma que el lenguaje JavaScript es muy flexible ya que sobre el núcleo de este lenguaje los desarrolladores han creado una gran cantidad de herramientas, añadiendo nuevas características y funcionalidades con un esfuerzo mínimo, como:

- Creación de APIs dentro de los navegadores para crear contenido HTML dinámico y establecer estilos CSS.
- API de terceros, permite a los desarrolladores incrustar funcionalidades de proveedores de contenido como Twitter o Facebook.

- Rápido desarrollo de aplicaciones y sitios, ya que se pueden aplicar librerías y marcos de trabajo de terceros al HTML.

1.5.18. Node.js

Es un lenguaje de programación orientado a objetos, creado por Ryan Dahl quien utilizó el motor de JavaScript V8 de Google Chrome para desarrollarlo, pero a diferencia del lenguaje JavaScript, este se ejecuta en el lado del servidor (Campusano, Fredes, & Badilla, 2014).

Node.js posee como arquitectura un modelo E/S sin bloqueo, lo que permite al programador desarrollar aplicaciones de red escalables, es decir, puede aceptar miles de conexiones simultáneas con un uso intensivo de datos en tiempo real, de manera eficiente (Abernethy, 2011).

De acuerdo a (Rodríguez, 2017) Node.js cuenta con una gran cantidad de módulos nativos que vienen instalados por defecto, ofrece la posibilidad de incluir módulos externos que se los puede implementar mediante el instalador de paquetes npm. Estos módulos dan acceso a varias librerías para obtener funcionalidades y mejores resultados al desarrollar aplicativos.

Para usar Ionic Framework, es necesario contar con Node.js, ya que se emplea como base o servidor de aplicación.

1.5.19. AngularJS

Según (Rivero, 2016) es un framework estructural de código abierto desarrollado por Google, está basado en el lenguaje de programación JavaScript, enfocado principalmente para crear páginas web de una sola página.

A diferencia de otros frameworks, este se encarga de administrar la parte lógica de una aplicación, mediante el uso de la arquitectura MVC, la cual se encarga de separar la aplicación en diferentes capas:

- Modelo. - Son los datos mostrados en la vista y con los que el usuario interactúa.
- Vista. Es lo que el usuario ve, es decir la interfaz.
- Controlador. - Interactividad entre las diferentes capas según las peticiones que realiza el usuario.

Para comprender de mejor manera se muestra el funcionamiento de esta arquitectura:

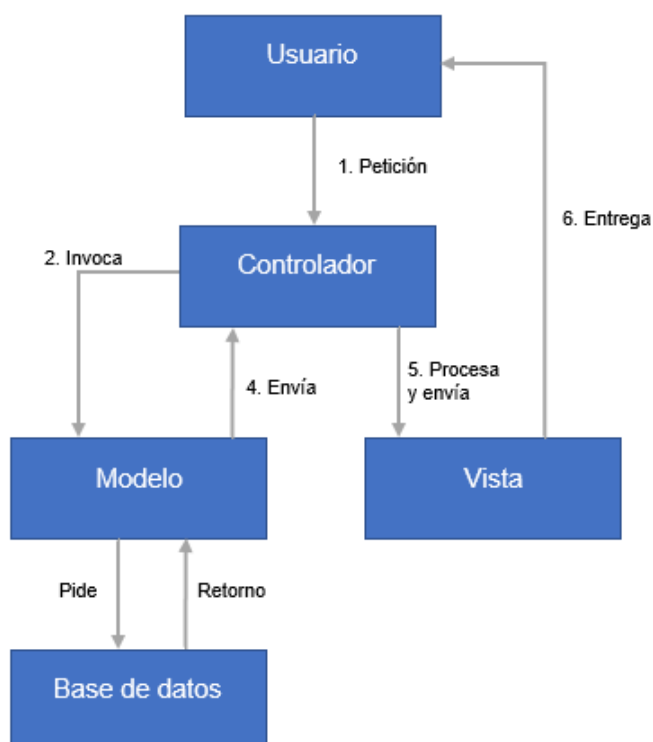


Figura I.5: Funcionamiento de la arquitectura MVC

Elaborado por: José Chacón

De acuerdo a (Solis, 2015) las principales características que lo convierten en un excelente framework son:

- Debido a su gran popularidad, existe gran cantidad de contenido en la web que sirve de guía a los desarrolladores.
- Permite la personalización total de la aplicación hasta el más mínimo detalle, ya que no utiliza componentes gráficos.
- El framework tiene un peso estimado de 105 kb, además de estar completamente optimizado para reducir el consumo de recursos del sistema.
- Puede ser utilizado con otros frameworks como: Bootstrap, JQuery o PhoneGap, entre otros.
- Al ser un framework permite reducir el tiempo de desarrollo, sin perder calidad en la aplicación.

1.5.20. Apache Cordova

Es un framework de desarrollo móvil de código abierto que mediante el uso de tecnologías como CSS3, HTML5 y JavaScript permite el desarrollo multiplataforma, lo que evita usar un lenguaje de programación nativo para cada Sistema Operativo móvil. Con este framework se puede usar varias APIs nativas de cada plataforma para acceder a los recursos y funcionalidades del dispositivo móvil: datos, sensores, estado de red, entre otros. Además de dar mayores funcionalidades a HTML y JavaScript, las aplicaciones desarrolladas son híbridas, esto quiere decir, que no son puramente nativas ni basadas en web (Apache Cordova , 2015).

Para (Ionic, 2016) el uso de Apache Cordova en Ionic es de suma importancia, ya que este último al ser un framework HTML5, para que pueda ejecutarse y utilice funciones como una aplicación nativa, requiere de un contenedor nativo; con Apache Cordova esto es posible, ya que provee librerías nativas y trabaja bajo la misma API JavaScript, independientemente del sistema operativo para el cual se esté desarrollando, como se muestra a continuación:

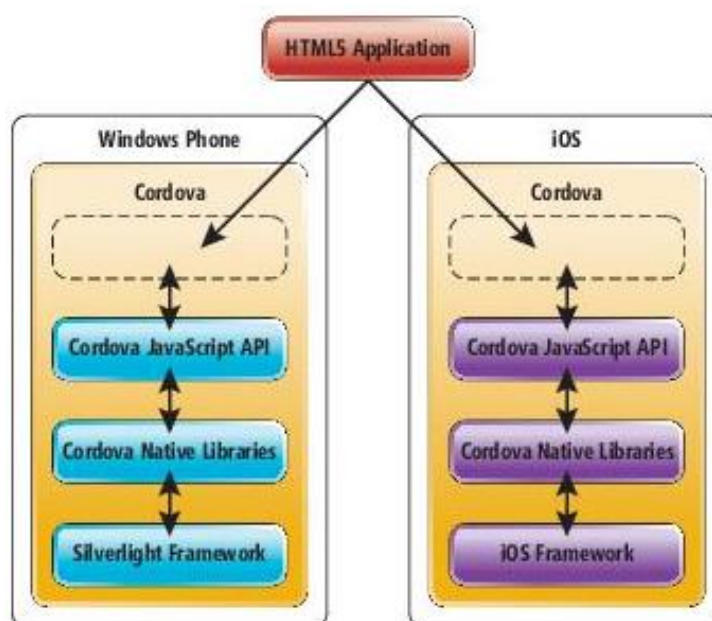


Figura I.6: Contenedor Apache Cordova

Fuente de: (Eberhardt, 2012)

1.5.21. Base de datos

Según (Gómez, 2013) define a una base de datos como un conjunto de datos interrelacionados, los cuales están almacenados de manera estructurada o agrupada y que pueden ser accesibles por distintos usuarios o aplicaciones simultáneamente.

Por otra parte (Marqués, 2009) afirma que antes, los datos se los utilizaba en ficheros desconectados dentro de los programas, esto causaba problemas ya

que existía duplicidad. Con una base de datos, se evita este problema porque todos los datos están integrados en juegos de ficheros. De esta manera los datos pueden ser compartidos a toda la empresa y no solo para un departamento en específico.

1.5.22. Sistema de gestión de bases de datos (SGBD)

Para (Marqués, 2009) es un programa que permite crear, definir, mantener una base de datos, además de proporcionar a los usuarios un control de la misma, es decir que pueden insertar, actualizar y eliminar datos.

Según (Date, 2001) un sistema de gestión de base de datos consta de cuatro componentes: software, hardware, datos, usuarios. Mediante la siguiente figura, se indica la interacción de dichos componentes con el sistema gestor:

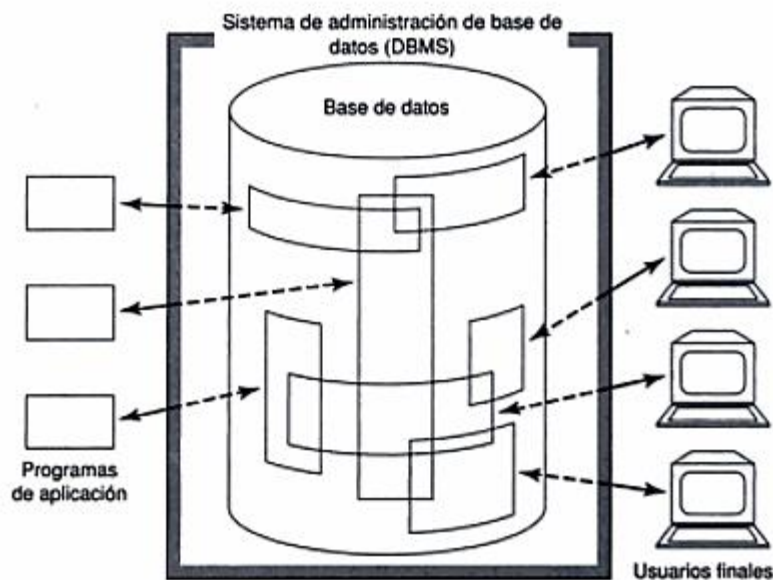


Figura I.7: Sistema de gestión de base de datos

Fuente de: (Date, 2001)

Las principales ventajas que ofrece un SGBD de acuerdo a (Paré, Santillán, & Costa, 2005) son las siguientes:

- Con un SGBD los usuarios pueden realizar consultas complejas de cualquier tipo.
- El administrador tiene la posibilidad de proporcionar reglas de integridad, frente a operaciones no permitidas.
- Permite a varios usuarios a la vez acceder a la base de datos.
- En cuanto a seguridad los SGDB permiten mantener la confidencialidad de los datos, mediante el establecimiento de claves en los diferentes niveles, que son: el nivel de toda la base de datos, el nivel de entidad y el de atributo.

CAPÍTULO II

METODOLOGÍA

2.1. Metodología de investigación

El presente proyecto utiliza varios métodos de investigación, los cuales se describen a continuación:

2.1.1. Investigación de campo

Para recolectar datos e información es necesario tener un contacto directo sobre las personas involucradas en el estudio, las cuales son estudiantes y personal administrativo de la PUCE Ambato.

2.1.2. Método analítico

Antes de desarrollar el sistema se necesita analizar los datos suministrados que utiliza la plataforma, así como también sus respectivos procesos.

2.1.3. Técnicas de recolección de información

Se aplica una encuesta destinada a los estudiantes de la PUCE Ambato ya que la aplicación va dirigida hacia ellos, de esta manera también se recolectó información necesaria para los requerimientos de la aplicación.

2.1.4. Instrumentos

La encuesta contiene preguntas objetivas que facilitaron el análisis de resultados. Ver Anexo 1.

2.1.5. Población

La encuesta se realizó a los estudiantes de la PUCE Ambato, para ello fue necesario conocer el número total de estudiantes: 977 en el período semestral

septiembre 2017 a enero 2018; el mismo fue obtenida a través de la Secretaria General de la Institución.

2.1.5.1. Muestra

Con la población de 977 estudiantes, se calcula la muestra por medio de la siguiente fórmula:

$$n = \frac{N * Z^2 * p * (1 - p)}{(N - 1) * e^2 + Z^2 * p * (1 - p)}$$

Donde:

n= Tamaño de la muestra

N= Tamaño de la población

p= Probabilidad de éxito

q= Probabilidad de fracaso

Z= Nivel de confianza

e= Margen de error

Con un nivel de confianza del 95%, margen de error del 5%, tanto la probabilidad de éxito y de fracaso del 50%, se obtiene como resultado una muestra de 276, esto significa el número total de encuestas que se debe aplicar.

2.2. Metodología de desarrollo

2.2.1. Definición

Según (Gutierrez, 2011) describe que la metodología es un conjunto de pasos que se debe seguir ordenadamente y de forma rigurosa para cumplir un objetivo, a partir de requisitos previos.

Existen varias metodologías que se ajustan a lo que se desea realizar, por lo tanto, la metodología que se utilizó para el desarrollo de la aplicación es Mobile-D.

2.3. Metodología Mobile-D

Según (Durán, 2013) Mobile-D es una metodología de desarrollo ágil, la cual está basada en: Extreme Programming, Crystal Methodologies y Rational Unified Process para la implementación, escalabilidad y cobertura del ciclo de vida respectivamente.

Además de ser utilizada para el desarrollo de aplicaciones móviles, también se puede usar en aplicaciones de: simulación, financieras, logísticas y de seguridad.

Esta metodología permite ciclos de desarrollo relativamente cortos y rápidos para equipos pequeños de desarrolladores, por lo que el producto final debe ser totalmente funcional en un tiempo estimado de diez semanas (Ramírez, 2009).

2.4. Fases de la Metodología

De acuerdo con el autor (Abrahamsson, Ihme, Kolehmainen, Kyllönen, & Salo, 2010), Mobile-D está compuesta por cinco fases: exploración, inicialización,

producción, estabilización y pruebas del sistema, cada una de ellas cuenta con determinadas etapas como se puede observar en la siguiente figura:



Figura II.1: Fases de la Metodología Mobile-D

Fuente de: (Agile Software Development Methodologies At Vtt Electronics, 2008)

2.4.1. Exploración

Esta fase consiste en establecer los objetivos del proyecto, así como también definir la meta a la que se debe alcanzar, definir los recursos necesarios para que no haya ningún problema ni retraso durante la realización del proyecto (Abrahamsson, Ihme, Kolehmainen, Kyllönen, & Salo, 2010).

Específicamente se realiza la recolección de requerimientos mediante la comunicación con el cliente, para ello, se utilizan técnicas de recolección de información usando herramientas de investigación científica como la entrevista o encuesta.

Para esta fase se realizó una reunión con la persona que administra la plataforma en el departamento de Desarrollo de Software de la PUCE Ambato, con el fin de definir la posible arquitectura de la aplicación, el entorno y las herramientas necesarias para el desarrollo; se aplicó una encuesta

dirigida a los estudiantes de la institución para analizar las necesidades con respecto al aplicativo. Ver página 44.

2.4.2. Inicialización

En esta fase es necesario que los desarrolladores se aseguren de contar con todas las herramientas a utilizar, también es recomendable que se familiaricen con las mismas (Abrahamsson, Ihme, Kolehmainen, Kyllönen, & Salo, 2010). Para el efecto, se preparan las herramientas que sustenten el desarrollo del aplicativo, con las que se realizan varios entrenamientos sobre la base de documentación existente en la web, con la finalidad de conocer su instalación, estructura, uso, entre otras. Ver página 60

2.4.3. Producción

Esta fase consiste en implementar las funcionalidades requeridas en la aplicación, utiliza ciclos iterativos e incrementales, que están conformados por: la planificación, el trabajo y la liberación, los mismos que cuentan con un día para su realización (Abrahamsson, Ihme, Kolehmainen, Kyllönen, & Salo, 2010). A saber:

- Día de planificación donde se realiza test de aceptación.
- Día de trabajo en donde se documenta los procesos.
- Día de liberación donde se realiza una lista de deficiencias.

En esta fase se ejecuta las actividades de desarrollo, la aplicación se trabaja por módulos, validando por separado cada funcionalidad, realizando pruebas unitarias antes de la integración. Ver página 70.

2.4.4. Estabilización

Esta es la fase más importante del proyecto, ya que se reúnen los equipos de desarrollo con la finalidad de integrar todos los módulos existentes para formar un producto final. Los desarrolladores realizan similares tareas a las que se hizo en la fase de producción: planificación, trabajo, documentación y fecha de entrega (Abrahamsson, Ihme, Kolehmainen, Kyllönen, & Salo, 2010).

Como se menciona anteriormente sobre esta fase, con respecto al presente proyecto, se realiza la integración de los respectivos módulos desarrollados para obtener el producto final, se verifica su correcta integridad y funcionalidad, a la vez se enlistan los errores encontrados para ser corregidos en la siguiente fase. Ver página 106.

2.4.5. Pruebas

Esta es la última fase de la metodología en donde se comprueba el correcto funcionamiento del sistema, de la misma manera se busca defectos y se los corrige si los hubiese. Se realiza la planificación, trabajo y liberación del sistema completo y funcional (Abrahamsson, Ihme, Kolehmainen, Kyllönen, & Salo, 2010).

Se estructuran pruebas que verifiquen en conjunto la correcta ejecución y desempeño de la aplicación, con el objetivo de encontrar alguna falencia y corregirla. Ver página 112.

CAPÍTULO III

DESARROLLO

3.1. Fase de Exploración

3.1.1. Definición de partes interesadas

Los estudiantes de la PUCE Ambato constituyen la parte interesada; por el hecho de que la agenda académica móvil va dirigida a este grupo humano, los requerimientos fueron obtenidos de los mencionados beneficiarios sobre la base de sus necesidades detectadas en la encuesta. Ver página 53.

3.1.2. Encuesta

La encuesta fue dirigida a los estudiantes de la PUCE Ambato con la finalidad de justificar el desarrollo y conocer las posibles funcionalidades de la aplicación móvil. Ver Anexo 1

3.1.3. Análisis de las encuestas

Al aplicar la encuesta fueron obtenidos los siguientes resultados:

Pregunta 1

¿Con qué frecuencia usa el teléfono móvil?

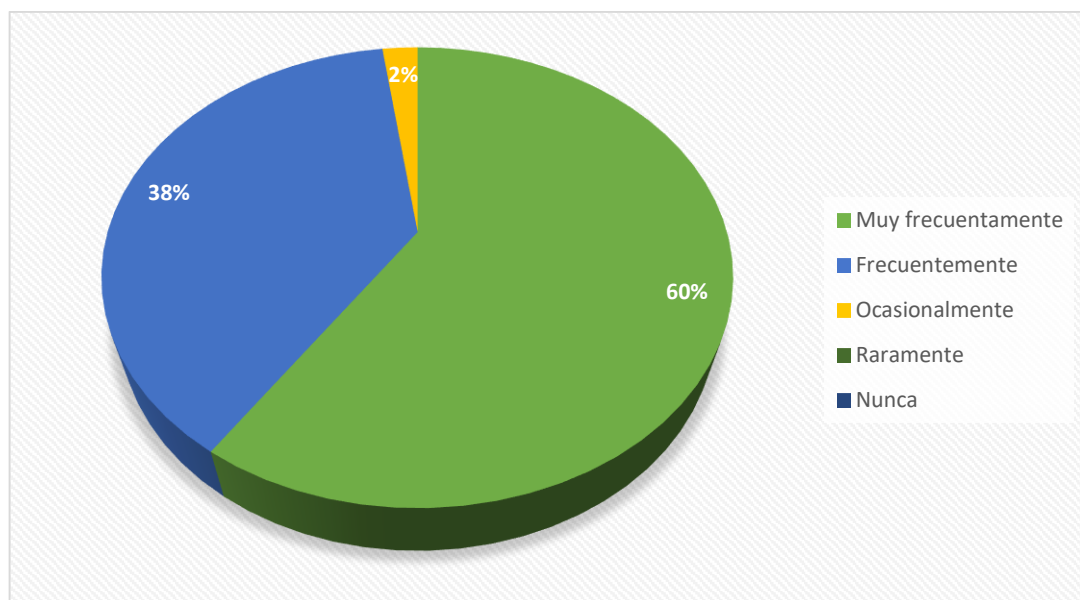


Gráfico III.1: Frecuencia que se utiliza el teléfono móvil

Elaborado por: José Chacón

Interpretación

Un alto índice de resultados demuestra que los estudiantes universitarios están en constante uso de los teléfonos móviles en su diario vivir, lo que evidencia que es un instrumento de comunicación importante y permanente en su cotidianidad social y familiar. Esto constituye una garantía de funcionalidad del aplicativo.

Pregunta 2

¿Qué sistema operativo utiliza su dispositivo móvil?

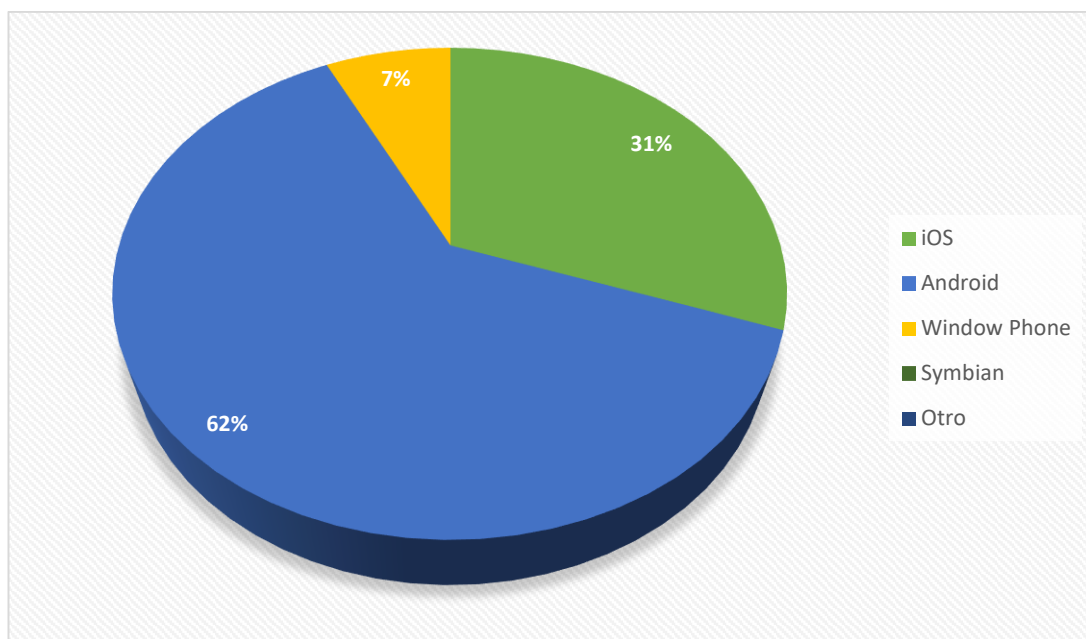


Gráfico III.2: Sistema operativo más utilizado

Elaborado por: José Chacón

Interpretación

Según las encuestas realizadas la mayoría de estudiantes cuenta con un dispositivo Android, lo cual justifica el desarrollo de la aplicación en este sistema, cabe destacar que existe un considerable porcentaje de estudiantes que posee IOS por lo que, para futuros proyectos se puede tomar en cuenta.

Pregunta 3

¿Con qué frecuencia accede a la Plataforma Educativa (MOODLE PUCE AMBATO)?

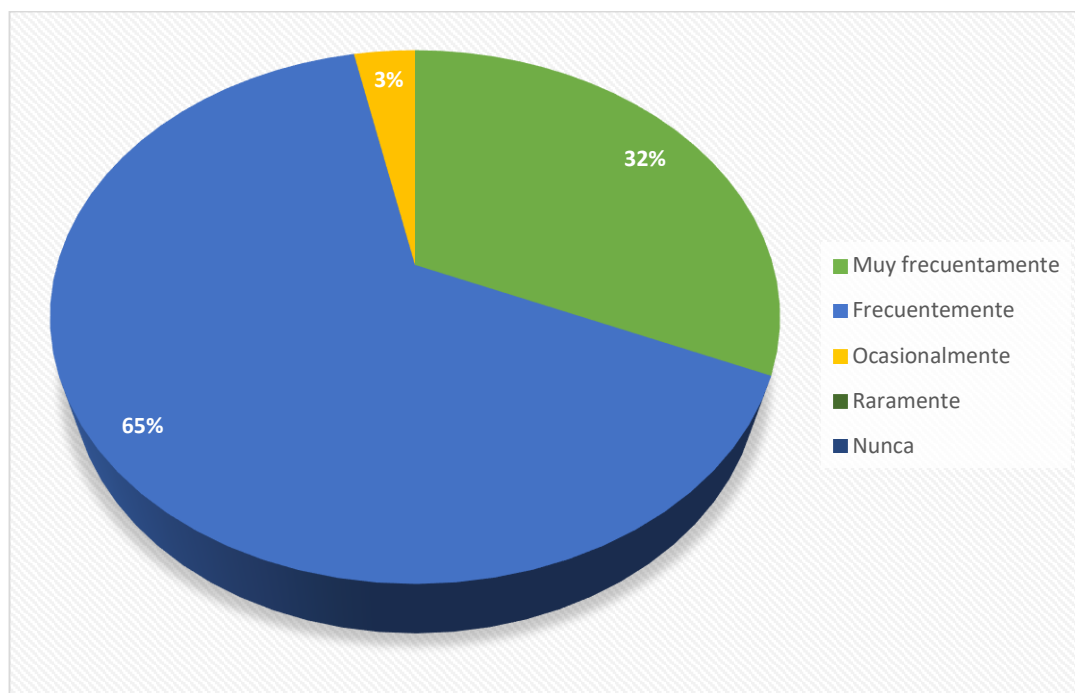


Gráfico III.3: Frecuencia con que acceden a la Plataforma Educativa

Elaborado por: José Chacón

Interpretación

De acuerdo a las encuestas, existe un alto nivel de estudiantes que acceden a la Plataforma Educativa frecuentemente, de esto se deduce la valía de su uso en el desarrollo académico, ya que este entorno virtual de enseñanza y aprendizaje mejora la interacción en el proceso educativo y se constituye un canal efectivo ofertado por la universidad.

Pregunta 4

¿Qué dispositivo usa para ingresar a la Plataforma Educativa (MOODLE PUCE AMBATO)?

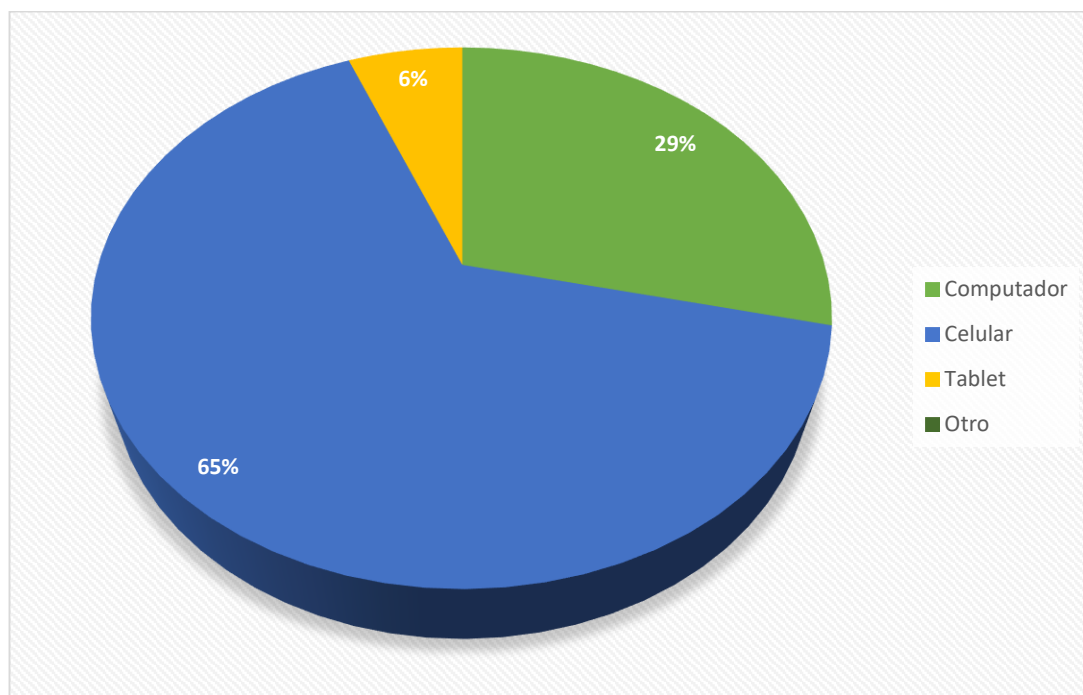


Gráfico III.4: Dispositivo que usan para acceder a la Plataforma Educativa

Elaborado por: José Chacón

Interpretación

La mayor parte de estudiantes ingresa a la Plataforma Educativa Moodle a través de un celular, por lo que claramente se puede determinar que este dispositivo reúne condiciones favorables que lo ubican en uno de los de mayor acogida; siendo una razón por la cual se desarrollará la aplicación para dispositivos móviles.

Pregunta 5

¿Utiliza alguna aplicación móvil para recordar, organizar o guardar información educativa?

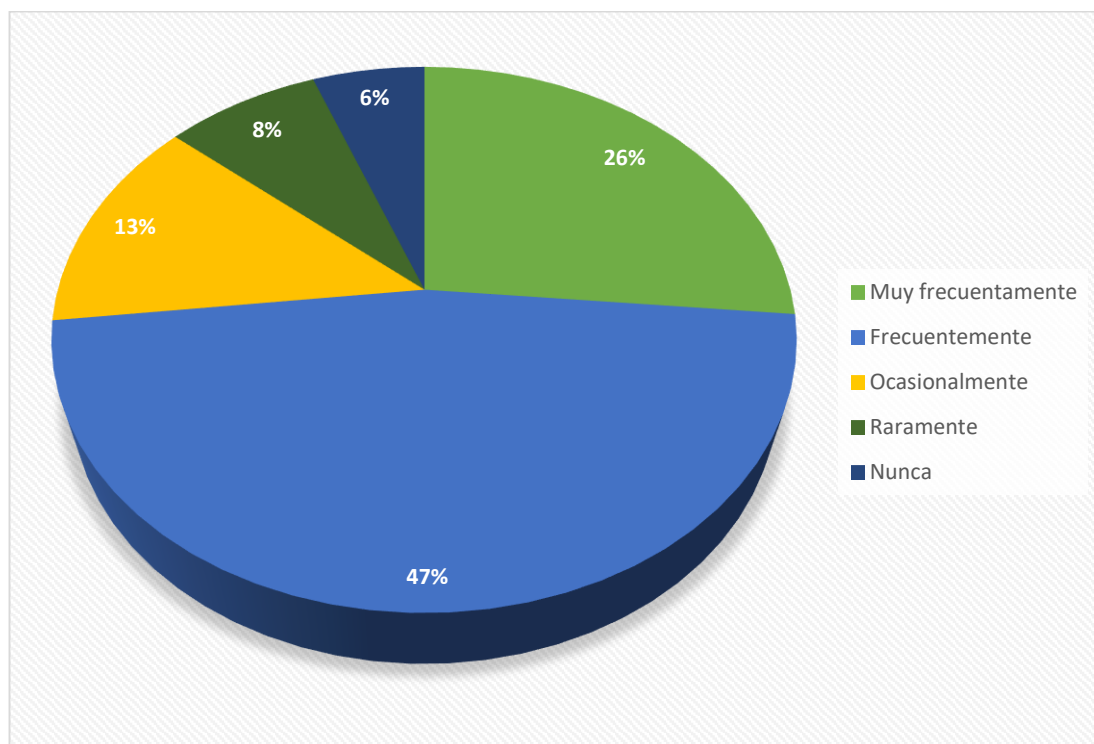


Gráfico III.5: Frecuencia del uso de agendas móviles

Elaborado por: José Chacón

Interpretación

De acuerdo a los resultados obtenidos en la encuesta, existe un alto nivel de estudiantes que usan aplicaciones de agendas educativas en sus dispositivos móviles, esto demuestra la necesidad de las mismas dentro del ámbito académico, hasta convertirse quizá en una herramienta de uso masivo e imprescindible a corto plazo.

Pregunta 6

¿A través de qué medio le gustaría que le recuerden las tareas pendientes de la Plataforma Educativa (MOODLE PUCE AMBATO)?

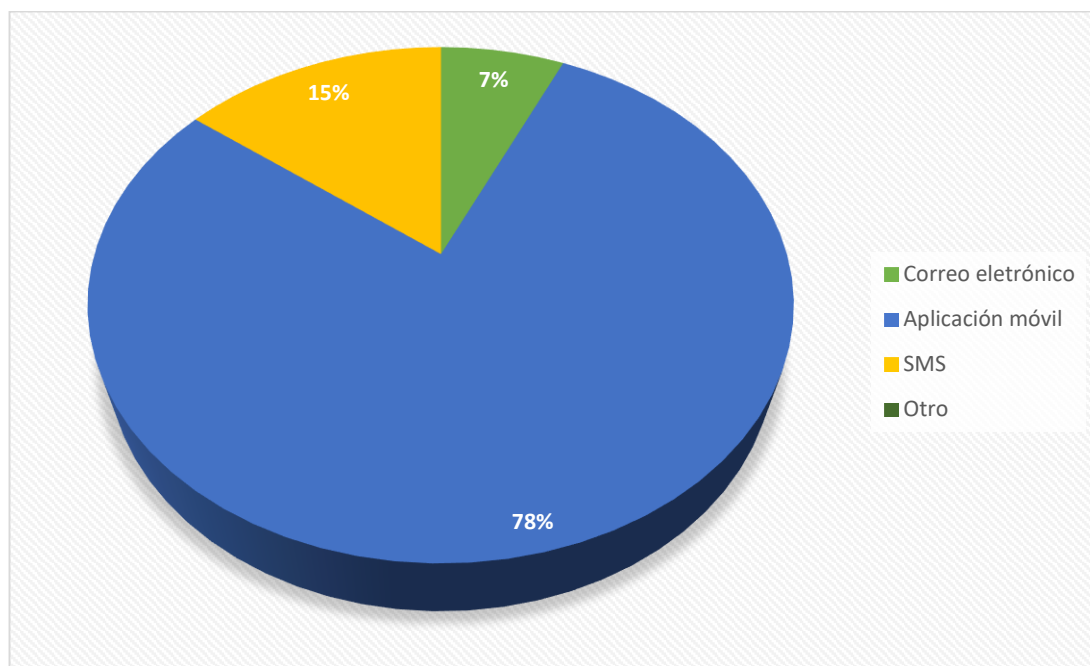


Gráfico III.6: Medio para recordar tareas pendientes

Elaborado por: José Chacón

Interpretación

Los estudiantes, al usar con mayor frecuencia los dispositivos móviles, como se sustentan en la pregunta 1, por obvias razones, prefieren que sea una aplicación móvil la encargada para realizar recordatorios de tareas pendientes.

Pregunta 7

¿Cree que al usar una aplicación móvil de notificaciones de tareas se mejorará en el cumplimiento de las mismas?

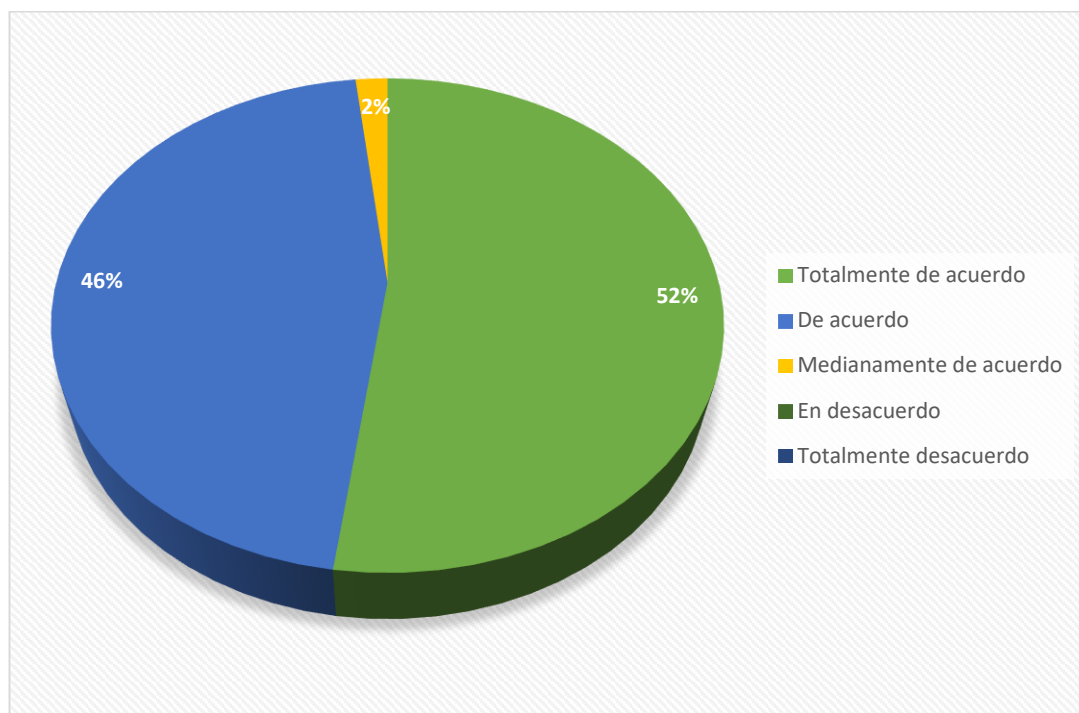


Gráfico III.7: Beneficio de la aplicación móvil

Elaborado por: José Chacón

Interpretación

Según las personas encuestadas, un gran porcentaje están totalmente de acuerdo y de acuerdo en que la aplicación móvil ayudará al cumplimiento de las tareas, de esto se concluye que el proyecto será de gran beneficio para los estudiantes quienes por falta de conocimiento sobre las tareas no pueden cumplirlas a tiempo, siendo la solución el desarrollo de una aplicación móvil de notificaciones.

Pregunta 8

¿Le gustaría que la aplicación móvil de notificaciones de tareas muestre un resumen de las tareas pendientes?

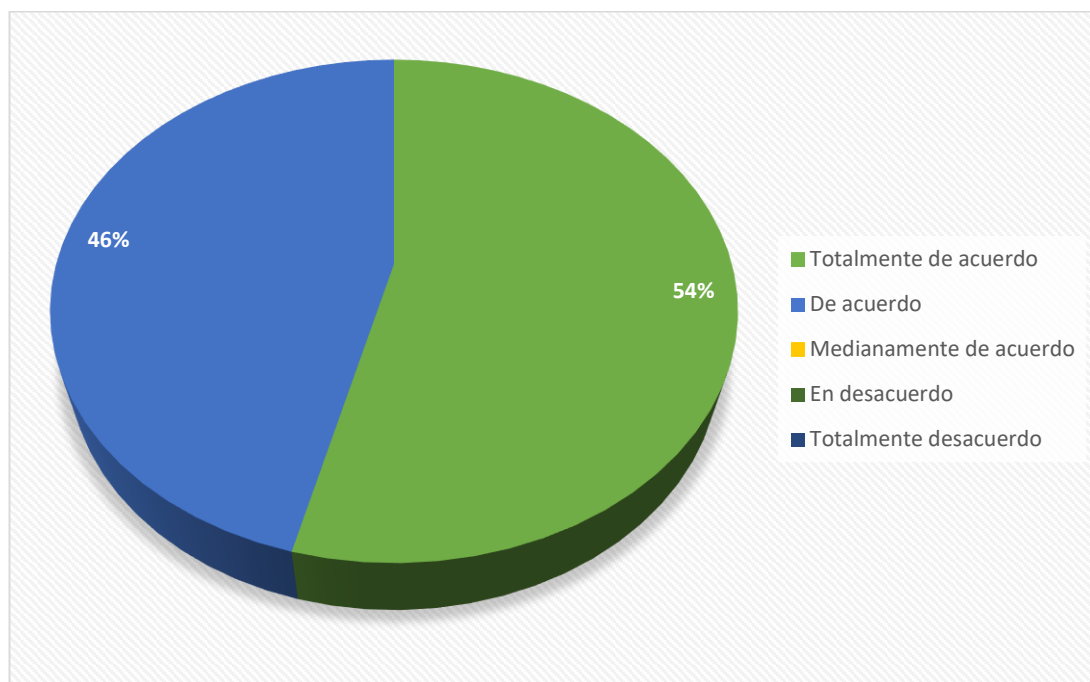


Gráfico III.8: Aspecto funcional

Elaborado por: José Chacón

Interpretación

En una mayor tendencia los encuestados muestran interés sobre la visualización de un resumen de las tareas pendientes que genere una ampliada comprensión la naturaleza de la tarea y más implicaciones que ella entraña, por lo que debe ser considerada una función que realice la aplicación.

Pregunta 9

¿Qué funciones le gustaría que tenga la aplicación móvil de notificaciones de tareas?

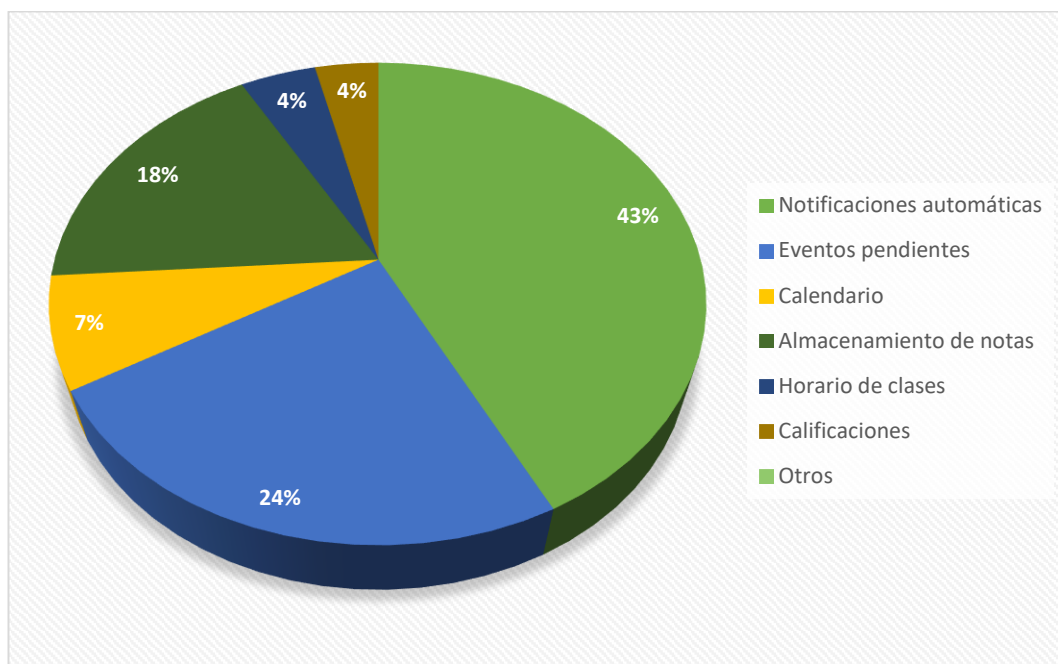


Gráfico III.9: Posibles funciones de la aplicación móvil

Elaborado por: José Chacón

Interpretación

De acuerdo a los resultados obtenidos tres son las funciones que les gustaría que tenga la aplicación móvil y que jerárquicamente se ubican con una mayor acogida, ellas son: las notificaciones automáticas, eventos pendientes y el almacenamiento de notas; con esto se tiene una mejor visión sobre los requerimientos que se debe tomar en cuenta para la aplicación.

Pregunta 10

¿Qué factores te gustaría que tenga la aplicación móvil de notificaciones de tareas?

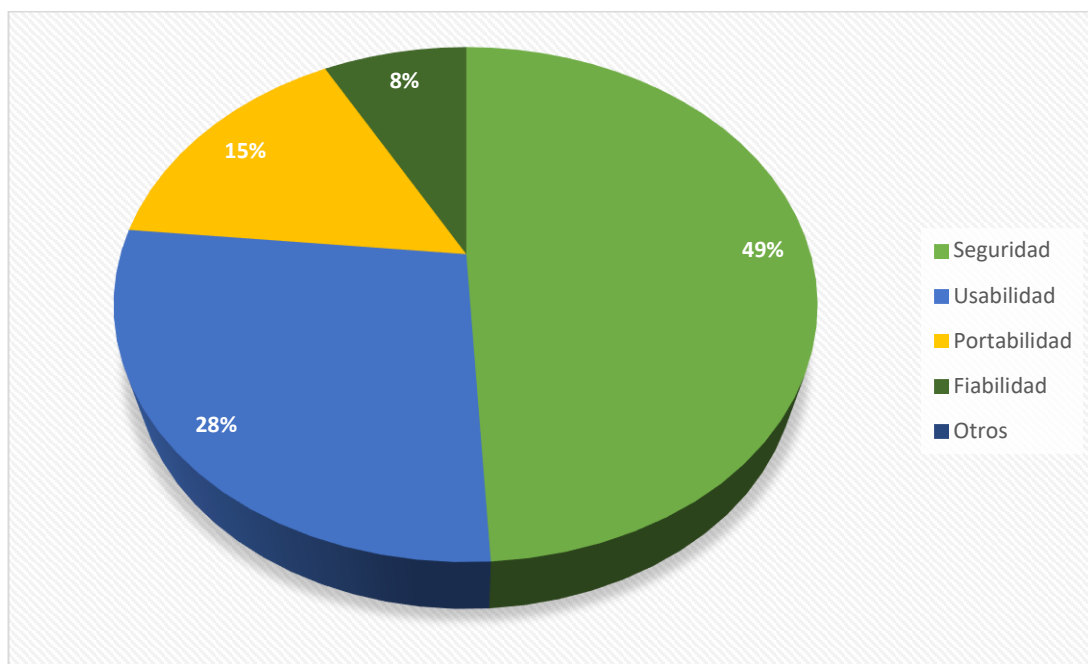


Gráfico III.10: Posibles factores en la aplicación móvil

Elaborado por: José Chacón

Interpretación

Según las encuestas realizadas, entre los factores que les gustaría que tenga la aplicación móvil, la mayoría opta por la seguridad y la usabilidad, estos factores son de mucha importancia para asegurar la calidad del producto final por lo que serán tomados en cuenta de acuerdo al resultado obtenido, sin dejar a un lado los factores con menor porcentaje.

3.1.4. Definición del alcance

Después de llevar a cabo una encuesta a los estudiantes de la PUCE Ambato se hizo un análisis de requerimientos de los interesados, del cual se pudo definir con más precisión las funcionalidades que contará la aplicación:

- La aplicación será desarrollada para dispositivos Android.
- Para el ingreso a la aplicación el usuario contará con un inicio de sesión.
- En la pantalla principal se mostrará el listado de las materias donde el estudiante tenga tareas pendientes.
- Al pulsar sobre una materia se desplegará todas las tareas con información detallada de la misma.
- El usuario tendrá la posibilidad de crear notas y guardarlas dentro de la aplicación.
- La aplicación contará con un sistema de notificaciones automáticas, el cual cumplirá con la función de recordar alguna tarea que tenga el estudiante.

3.1.5. Establecimiento del proyecto

3.1.5.1. Selección del entorno

Para comenzar con el desarrollo del proyecto es primordial contar con las herramientas adecuadas, así como también la adquisición de un dispositivo Android con el fin de realizar las respectivas pruebas de la aplicación.

3.1.5.2. Asignación del personal

Tanto el diseño como el desarrollo de la aplicación son realizados por el autor del presente proyecto.

3.1.5.3. Adquisición de herramientas

De acuerdo con la metodología se adquieren las herramientas que se utilizan para el desarrollo de la aplicación móvil, las mismas que se detallan a continuación:

3.1.5.3.1. Hosting

Se utiliza un alojamiento web en GoDaddy, destinado para el web service y la instalación de la plataforma Moodle, además de un dominio para su respectivo acceso.

El hosting cuenta con las siguientes especificaciones:

- Sistema operativo Linux.
- 100 gb de espacio del disco.
- Ancho de banda sin medición
- 50 usuarios FTP.
- 10 bases de datos, el phpMyAdmin admite bases de datos compatibles con MySQL 5.0 o superior, MariaDB 5.0 o superior.
- 5 gb de almacenamiento de correo electrónico
- Certificado SSL

3.1.5.3.2. Moodle

Con el objetivo de emular el ambiente de enseñanza y aprendizaje que posee la PUCE Ambato, se instala la plataforma educativa Moodle; cabe señalar que no se utiliza el Moodle de la Universidad por fines de seguridad. Esta instalación es necesaria, ya que la aplicación está conectada con su base de datos, para visualizar información especificada y notificar al estudiante, conforme el educador publique a través de la plataforma; además, se emplea para realizar las pruebas respectivas de la aplicación.

A continuación, se detallan los requisitos de instalación, relacionado a la versión de Moodle 3.3.2¹ usada para el presente proyecto:

- Versión mínima de PHP 5.4.4 o superior
- Soporta los siguientes servidores de base de datos: PostgreSQL 9.1, MySQL 5.5.31, MariaDB 5.5.31, Microsoft SQL Server 2008, Oracle Database 10.2. Estas son las versiones mínimas soportadas, los desarrolladores recomiendan usar las más recientes y estables.
- Navegadores con la versión mínima que soporta: Google Chrome 30.0, Mozilla Firefox 25.0, Apple Safari 6, Microsoft Internet Explorer 9.
- Servidor con Apache o IIS

3.1.5.3.3. Ionic Framework

Es el entorno de trabajo donde se codifica la aplicación, se utiliza la versión 3 de Ionic la cual tiene las siguientes características:

¹ Para información más detalla de los requisitos visite:
https://docs.moodle.org/all/es/Notas_de_Moodle_3.0

- Para el desarrollo de aplicaciones híbridas móviles utiliza tecnología web, es decir, HTML 5, CSS y JavaScript.
- Como sistema nativo de gestión de base de datos soporta SQLite, sin embargo, se puede usar APIs o web service para conectar con otros gestores como MySQL, que fue el caso del presente proyecto.
- Los sistemas operativos soportados para construir aplicaciones, son Android y IOS desde la versión 4.4+ y 8+, respectivamente.
- Se puede desarrollar aplicaciones bajo Ionic en cualquier sistema operativo como: Windows, Linux, Mac OS X, siendo este último necesario para desarrollar e implementar aplicaciones para IOS.
- Previo a la instalación de este Framework, es necesario adquirir Node.js y Apache Cordova en sus últimas versiones estables.
- Para compilar aplicaciones en Android, se debe instalar Java JDK y el SDK de dicha plataforma. Es recomendable descargar Android Studio ya que incluye el SDK, o a su vez adquirir solo el SDK Manager para instalar o mantener actualizado los componentes específicos para el funcionamiento del framework.

Al momento del desarrollo, los paquetes instalados para Ionic cuentan con las siguientes versiones:

```
global packages:

@ionic/cli-utils : 1.1.2
Cordova CLI      : 7.1.0
Ionic CLI        : 3.1.2

local packages:

@ionic/cli-plugin-cordova : 1.2.1
@ionic/cli-plugin-ionic1  : 1.1.2
Cordova Platforms         : android 6.0.0
Ionic Framework           : ionic1 1.3.3

System:

Node       : v6.10.3
OS         : Windows 10
Xcode      : not installed
ios-deploy : not installed
ios-sim    : not installed
```

Figura III.1: Información del entorno de trabajo utilizado

Elaborado por: José Chacón

3.1.5.3.4. Notepad++

Es una herramienta parecida al bloc de notas que permite editar texto sin formato de forma sencilla. Sin embargo, proporciona opciones más avanzadas, que son útiles para usuarios con alta experiencia en programación. Cabe mencionar que esta herramienta es de código abierto, sin costo alguno, la misma soporta varios lenguajes de programación.

Con la ayuda de Notepad++ se codifica los archivos HTML, JavaScript, CSS y PHP para la aplicación.

A continuación, se muestra la interfaz de dicha herramienta:

```

1  <?php
2
3  /*
4   siempre tener en cuenta "config.inc.php"
5  */
6  require("config.inc.php");
7
8  //if posted data is not empty
9  if (!empty($_POST)) {
10     //preguntamos si el usuario y la contraseña esta vacia
11     //sino muere
12     if (empty($_POST['username']) || empty($_POST['password'])) {
13
14         // creamos el JSON
15         $response["success"] = 0;
16         $response["message"] = "Por favor entre el usuario y el password";
17
18         die(json_encode($response));
19     }
20
21     //si no hemos muerto (die), nos fijamos si exist en la base de datos
22     $query = " SELECT 1 FROM users WHERE username = :user";
23
24     //actualizamos el :user
25     $query_params = array(
26         ':user' => $_POST['username']
27     );
28
29     //ejecutamos la consulta
30     try {
31         // estas son las dos consultas que se van a hacer en la base de datos

```

Figura III.2: Interfaz de Notepad++

Elaborado por: José Chacón

3.2. Fase de inicialización

3.2.1. Preparación de ambiente

Antes de iniciar la programación del aplicativo, se debe instalar el framework² de desarrollo por lo cual se realiza los siguientes pasos:

Instalación de Node.js

Node.js es usado para instalar todas las dependencias necesarias de un proyecto en Ionic, a través de un manejador de paquetes conocido como npm. Además, permite al desarrollador visualizar la aplicación en el navegador, ya que ejecuta un servicio de desarrollo de servidor.

² Para más detalle sobre la instalación de Ionic dirígase:
<https://ionicframework.com/docs/v1/guide/installation.html>,
<https://code.tutsplus.com/es/tutorials/ionic-from-scratch-getting-started-with-ionic--cms-29862>

Para la instalación, es necesario dirigirse a la página web <https://nodejs.org/es/download/> en donde se encuentra para descarga el instalador, cabe mencionar que existen dos versiones para los diferentes sistemas operativos, Ionic recomienda utilizar las versiones LTS ya que son más estables:

Descargas

Versión actual: 8.9.3 (includes npm 5.5.1)

Descargue el código fuente de Node.js o un instalador pre-compilado para su plataforma, y comience a desarrollar hoy.




	LTS Recomendado para la mayoría		Actual Últimas características	
				
	Windows Installer		Macintosh Installer	
	node-v8.9.3-x64.msi		node-v8.9.3.pkg	
				
			Source Code	
			node-v8.9.3.tar.gz	
Windows Installer (.msi)	32-bit		64-bit	
Windows Binary (.zip)	32-bit		64-bit	
macOS Installer (.pkg)			64-bit	
macOS Binaries (.tar.gz)			64-bit	
Linux Binaries (x86/x64)	32-bit		64-bit	
Linux Binaries (ARM)	ARMv6	ARMv7	ARMv8	
Source Code			node-v8.9.3.tar.gz	

Figura III.3: Versión de Node.js para los diferentes sistemas operativos

Fuente de: (nodeJS, 2017)

Se ejecuta el instalador descargado para luego proceder con la instalación.

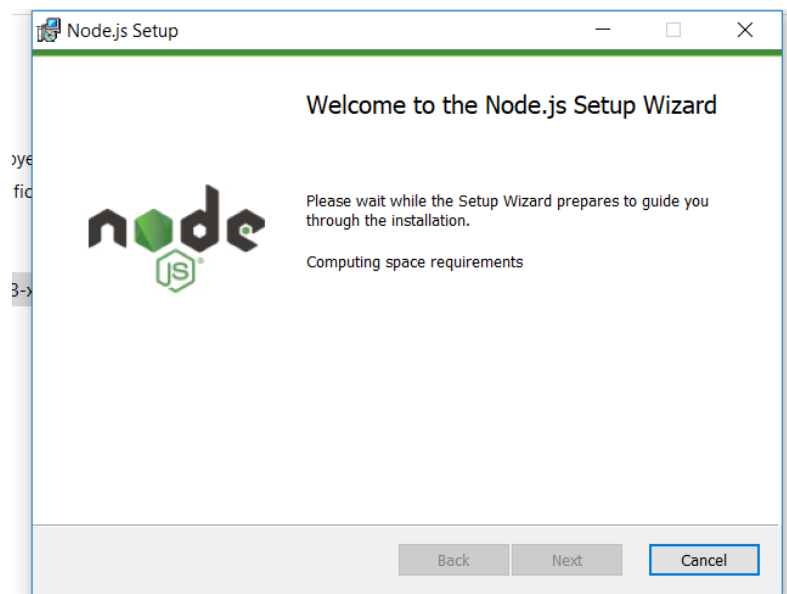


Figura III.4: Asistente de instalación Node.js

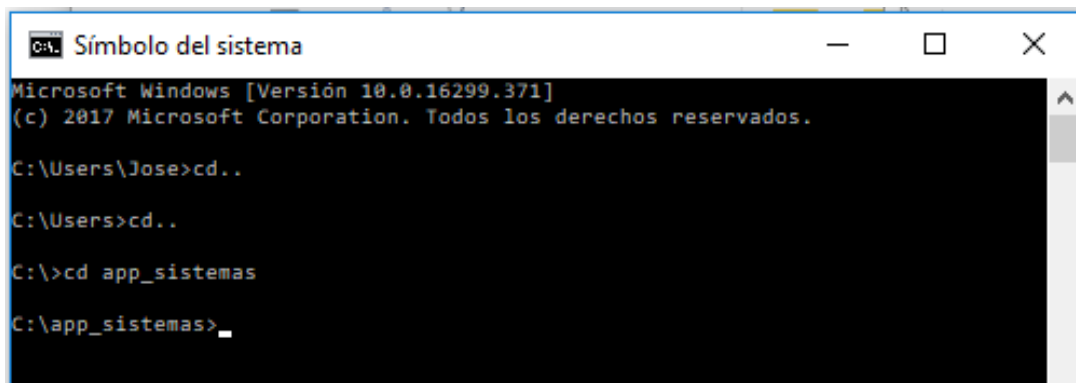
Elaborado por: José Chacón

Al finalizar se obtiene instalado Node.js y los comandos npm necesarios para continuar con el siguiente paso.

Instalación de Cordova

Cordova permite acceder a los componentes nativos del dispositivo a través del uso de plugins, además de usarse para encapsular aplicaciones, es decir, compilar todo el proyecto de Ionic, basado en tecnología web, en un archivo ejecutable para ser instalado en el dispositivo móvil.

Para la instalación Cordova se utiliza el Administrador de Paquetes Node (npm) y la Interfaz de Línea de Comando (CLI) que en el caso de Windows es la terminal o cmd.



```

c:\ Símbolo del sistema
Microsoft Windows [Versión 10.0.16299.371]
(c) 2017 Microsoft Corporation. Todos los derechos reservados.

C:\Users\Jose>cd..
C:\Users>cd..
C:\>cd app_sistemas
C:\app_sistemas>_

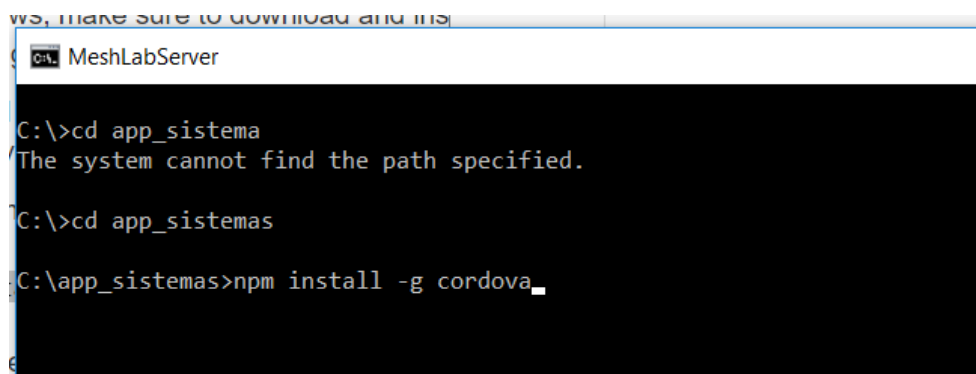
```

Figura III.5: Selección de carpeta donde se instala Cordova

Elaborado por: José Chacón

Como se puede apreciar en la figura 13, este procedimiento se realiza para seleccionar el directorio y crear una carpeta, en la cual se guarda todos los archivos de la aplicación, posteriormente, se ingresa en la misma y se ejecuta el siguiente comando para instalar Cordova:

- npm install -g cordova



```

c:\ MeshLabServer

C:\>cd app_sistema
The system cannot find the path specified.

C:\>cd app_sistemas
C:\app_sistemas>npm install -g cordova_

```

Figura III.6: Instalación Cordova

Elaborado por: José Chacón

Después se procede a instalar Ionic Framework, para esto se utiliza el siguiente comando:

- npm install -g ionic

```
MeshLabServer
C:\>cd app_sistema
The system cannot find the path specified.
C:\>cd app_sistemas
C:\app_sistemas>npm install -g ionic
```

Figura III.7: Instalación de Ionic

Elaborado por: José Chacón

Culmina la instalación del framework; por último, para poder desarrollar aplicaciones en Android es necesario instalar el kit de desarrollo, para lo cual se descarga e instala Android Studio que incluye el SDK, el mismo que se lo obtiene mediante el siguiente URL:

<https://developer.android.com/studio/index.html?hl=es-419>

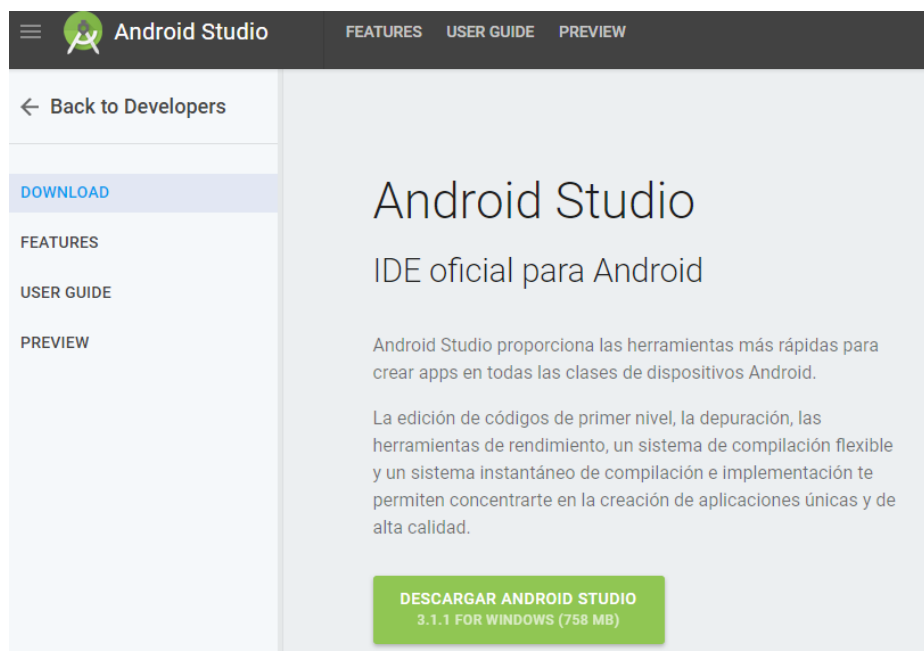


Figura III.8: Página de descarga Android Studio

Fuente de: (Android Developers, 2017)

Una vez realizado todos los pasos anteriores, se tiene listo el ambiente para iniciar el desarrollo de la aplicación.

3.2.2. Entrenamiento

Antes de empezar el desarrollo del aplicativo, es necesario familiarizarse con las herramientas, para ello se investiga documentación existente en la página oficial del framework (Ionic, 2016) con la finalidad de entender mejor los conceptos básicos, su utilización, estructura, entre otras.

A demás de contar con la documentación oficial, los siguientes enlaces son de mucha ayuda para incrementar el conocimiento sobre esta herramienta:

- <http://ccoenraets.github.io/ionic-tutorial/>
- <http://www.w3ii.com/es/ionic/default.html>
- <https://ajgallego.gitbooks.io/ionic/>
- <https://guiadev.com/manual-ionic-framework/>
- <https://demo.mobiscroll.com/angularjs/forms/textfields#>

A continuación, se detallan los comandos más comunes que se usan en Ionic y los que se aplican para el desarrollo del aplicativo:

Cabe mencionar que se debe ingresar en la consola de Windows y ubicarse en la carpeta donde se encuentra el proyecto para ejecutar estos comandos.

Para crear un nuevo proyecto

Ionic posee varias líneas de comandos que permiten fácilmente la creación de proyectos, entre ellas se encuentra una instrucción para generar plantillas en blanco:

- \$ ionic start myApp blank

Por otra parte, Ionic permite generar proyectos con contenido, a través de los siguientes comandos:

- \$ ionic start myApp tabs
- \$ ionic start myApp sidemenu

Generación de íconos y pantalla inicial del aplicativo

Para generar los íconos y la pantalla inicial de la aplicación en todos los tamaños se utiliza:

- ionic resources



















<input type="checkbox"/> Name	Date modified	Type
 drawable-land-hdpi	12/17/2017 9:45 PM	File folder
 drawable-land-ldpi	12/17/2017 9:45 PM	File folder
 drawable-land-mdpi	12/17/2017 9:45 PM	File folder
 drawable-land-xhdpi	12/17/2017 9:45 PM	File folder
 drawable-land-xxhdpi	12/17/2017 9:45 PM	File folder
 drawable-land-xxxhdpi	12/17/2017 9:45 PM	File folder
 drawable-port-hdpi	12/17/2017 9:45 PM	File folder
 drawable-port-ldpi	12/17/2017 9:45 PM	File folder
 drawable-port-mdpi	12/17/2017 9:45 PM	File folder
 drawable-port-xhdpi	12/17/2017 9:45 PM	File folder
 drawable-port-xxhdpi	12/17/2017 9:45 PM	File folder
 drawable-port-xxxhdpi	12/17/2017 9:45 PM	File folder
 mipmap-hdpi	12/17/2017 9:45 PM	File folder
 mipmap-ldpi	12/17/2017 9:45 PM	File folder
 mipmap-mdpi	12/17/2017 9:45 PM	File folder
 mipmap-xhdpi	12/17/2017 9:45 PM	File folder
 mipmap-xxhdpi	12/17/2017 9:45 PM	File folder
 mipmap-xxxhdpi	12/17/2017 9:45 PM	File folder

Figura III.9: Generación de iconos y SplashScreen

Elaborado por: José Chacón

Al ejecutar dicha línea, se obtiene el ícono que identifica a la aplicación, la cual se muestra en el móvil al instalarlo, además, genera una pantalla inicial conocida como splashscreen que aparece al arrancar la aplicación.



Figura III.10: Ícono de la aplicación móvil

Elaborado por: José Chacón



Figura III.11: Pantalla inicial (SplashScreen)

Elaborado por: José Chacón

Permitir acceso a la red

A través de este comando se configura la aplicación para que puede acceder a URLs externas:

- \$ ionic cordova plugin add cordova-lugin-inappbrowser

Permitir exportar en varias plataformas

Estos comandos sirven para agregar las plataformas de compilación dependiendo del sistema operativo móvil al que va dirigida la aplicación:

- \$ ionic cordova platform add nombre-de-la-plataforma

Estructura básica del proyecto

Al ser un framework, Ionic aporta un entorno de trabajo organizado, pues está estructurado en la jerarquía de archivos del sistema operativo de la siguiente manera:

Nombre	Fecha de modifica...	Tipo
.sourcemaps	3/5/2018 16:25	Carpeta de archivos
node_modules	3/5/2018 14:43	Carpeta de archivos
platforms	3/5/2018 14:36	Carpeta de archivos
plugins	3/5/2018 14:44	Carpeta de archivos
resources	3/5/2018 14:44	Carpeta de archivos
src	3/5/2018 12:24	Carpeta de archivos
www	3/5/2018 11:13	Carpeta de archivos
.editorconfig	1/5/2018 12:49	Archivo EDITORC...
.gitignore	1/5/2018 12:49	Documento de tex...
config.xml	3/5/2018 14:44	Documento XML
ionic.config.json	3/5/2018 17:28	JSON File
package.json	3/5/2018 14:44	JSON File
tsconfig.json	1/5/2018 12:49	JSON File
tslint.json	1/5/2018 12:49	JSON File

Figura III.12: Estructura del proyecto Ionic

Elaborado por: José Chacón

En la siguiente tabla se detalla la funcionalidad que cada una de las carpetas de la estructura del framework aporta para el desarrollo de una aplicación en Ionic:

Carpeta	Funcionalidad
hooks/	En esta carpeta se guarda o se alojan los scripts en los que se tiene funciones que controlan los eventos.
node_modules/	Se crea automáticamente al instalar dependencias npm, contiene los paquetes necesarios para el funcionamiento del framework.
platforms/	Contiene los proyectos de las plataformas para los cuales se va a compilar.
plugins/	Aloja los plugins o módulos instalados que se utilizan en la aplicación.
resources/	Contiene el ícono y la pantalla de inicio (SplashScreen) del aplicativo en todos los tamaños.
src/	Es la carpeta principal en donde se codifica el aplicativo, en ella se encuentran las páginas que contendrán la aplicación con sus respectivos archivos HTML, SCSS, TS para definir la plantilla, los estilos y el comportamiento de las páginas, respectivamente.
www/	Esta carpeta se genera automáticamente y guarda la versión del código cada vez que se realiza cambios.

.editorconfig y .gitignore	Archivos relacionados a la configuración del editor de código y de git.
config.xml	En este archivo se indica los permisos especiales de la aplicación, por ejemplo, para ser desplegada en una plataforma de distribución.
ionic.config.json	Contiene configuraciones propias de Ionic sobre el proyecto.
package.json	Carpeta en donde se encuentra las dependencias y paquetes de nodeJS.
tsconfig.json	Archivo de configuración global de TypeScript.
tslint.json	Archivo que ayuda a los editores de código a prevenir errores de TypeScript.

Tabla III.1. Estructura de un proyecto en Ionic.

Elaborado por: José Chacón

Es recomendable usar un editor texto con el fin de facilitar el acceso y codificación a las diferentes carpetas y archivos del proyecto

3.3. Fase de producción

En esta fase se realiza varias tareas, tales como: determinar la arquitectura y el modelo de datos de la aplicación, además de documentar el avance que se obtiene durante el proceso desarrollo.

Cabe mencionar que las tareas como programación simultánea no se pueden realizar en este proyecto, ya que es una sola persona la que desarrolla el mismo.

3.3.1. Arquitectura del aplicativo

Como se menciona en capítulos anteriores, Ionic Framework trabaja de la mano con Angular, por ende, utiliza el patrón de arquitectura Modelo-Vista-Controlador, el mismo se encarga de separar la interfaz de la parte lógica de una aplicación.

A continuación, se indica la arquitectura con la que se trabaja todo el desarrollo:

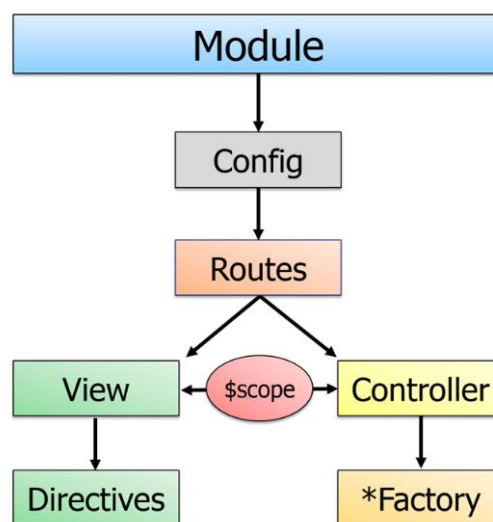


Figura III.13: Arquitectura de una aplicación en Ionic

Fuente de: (Gallego, 2017)

Como se puede apreciar en la figura 21, existen otros componentes además de las vistas y controladores que actúan dentro de una aplicación en Ionic, como son: factorías, directivas, las rutas y la configuración.

Para entender mejor sobre estos componentes se describe las tareas que cumplen en la aplicación, según (Gallego, 2017):

- El modelo son los datos que maneja el aplicativo, este componente se encarga de enviar o recibir información según las peticiones del controlador.
- Los controladores sirven de enlace entre la vista y el modelo, se encargan de obtener los datos de los servicios y los envían en variables `$scope` para ser mostradas en las vistas.
- La vista es la interfaz de usuario que permite visualizar los datos procesados, según lo que proporcione el modelo.
- Las rutas y la configuración sirven para asociar los controladores con las respectivas vistas de la aplicación.
- Las directivas son funciones o variables dinámicas que permiten crear componentes visuales personalizados, a través código HTML encapsulado que puede ser reutilizable.

3.3.2. Diagrama entidad-relación de Moodle

Como el aplicativo que se desarrolla en el presente proyecto necesita mostrar información específica de los estudiantes, es de suma importancia conocer la estructura de la base de datos de Moodle: tablas, campos, relaciones, claves principales y foráneas que posee, para ello, se toma como referencia el siguiente diagrama entidad-relación:

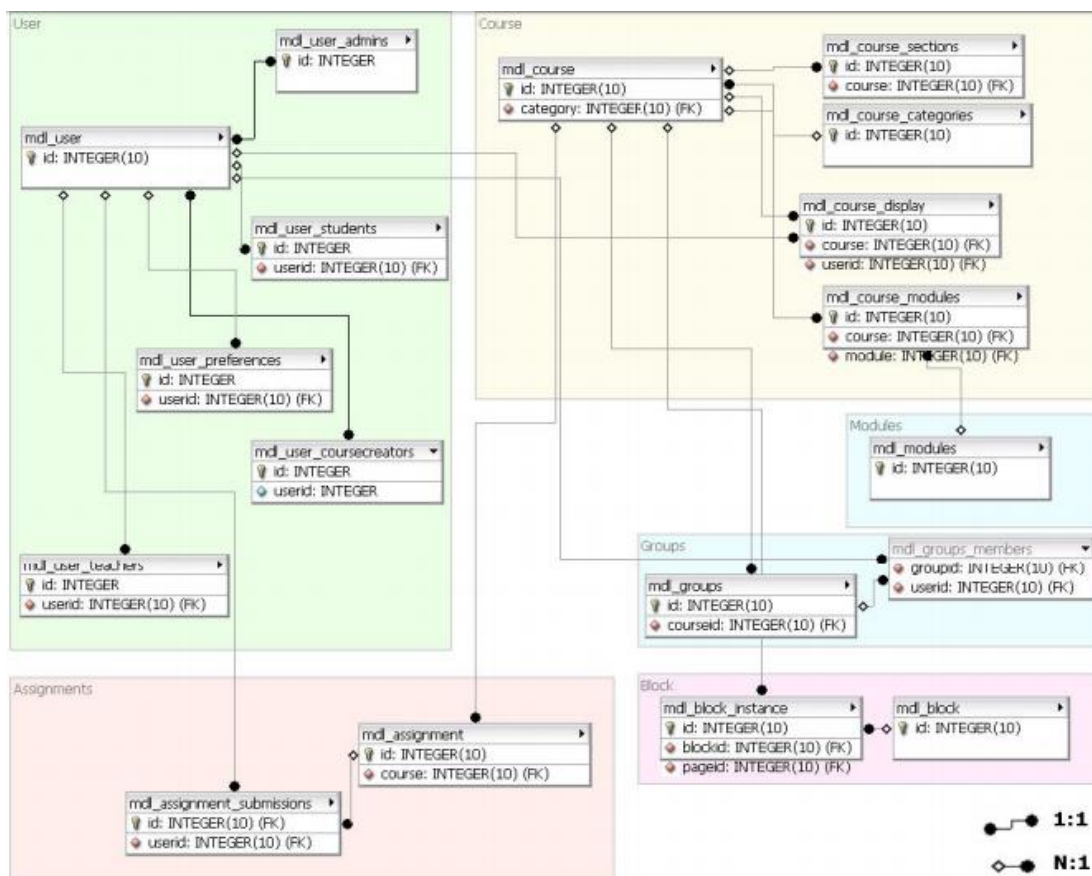
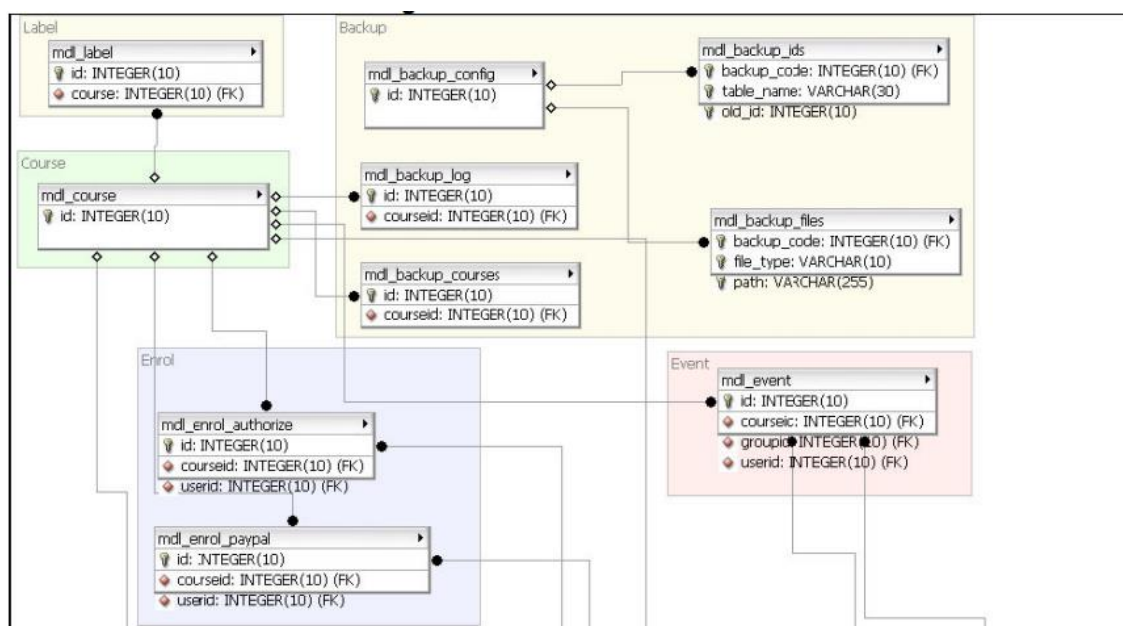


Figura III.14: Modelo entidad-relación de Moodle. Parte 1

Fuente de: (Duvan & Eduard, 2007)



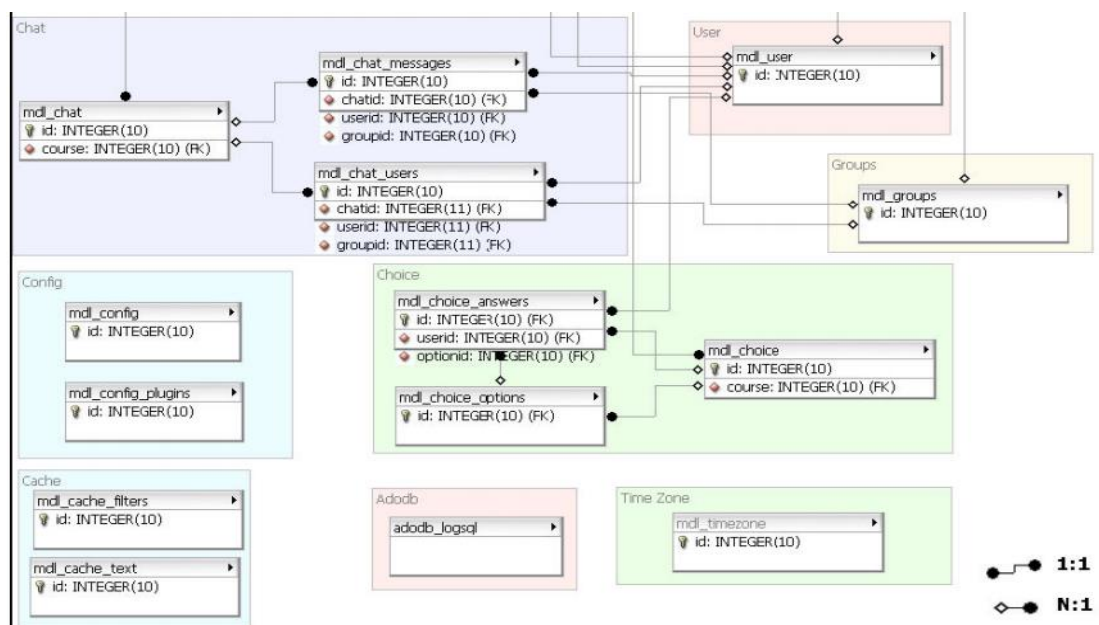


Figura III.15: Modelo entidad-relación de Moodle. Parte 2

Fuente de: (Duvan & Eduard, 2007)

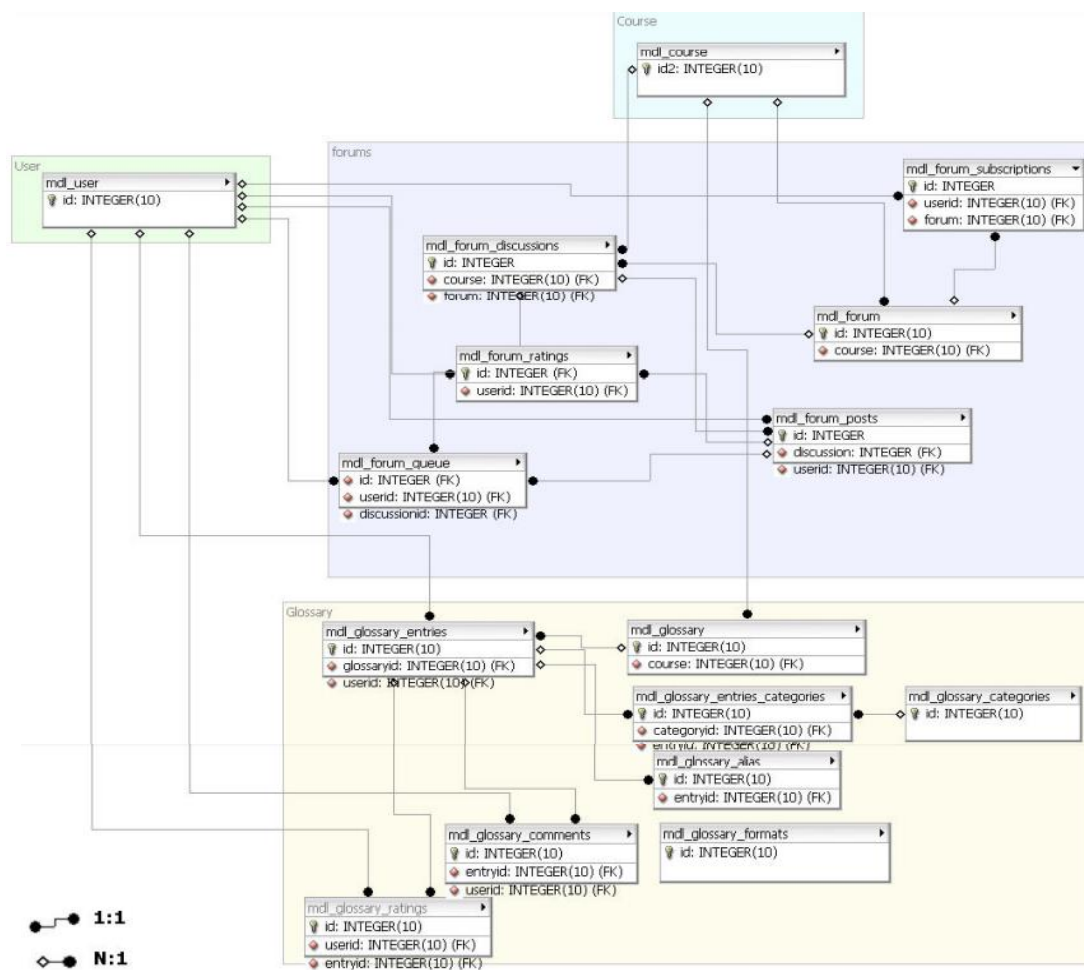


Figura III.16: Modelo entidad-relación de Moodle. Parte 3

Fuente de: (Duvan & Eduard, 2007)

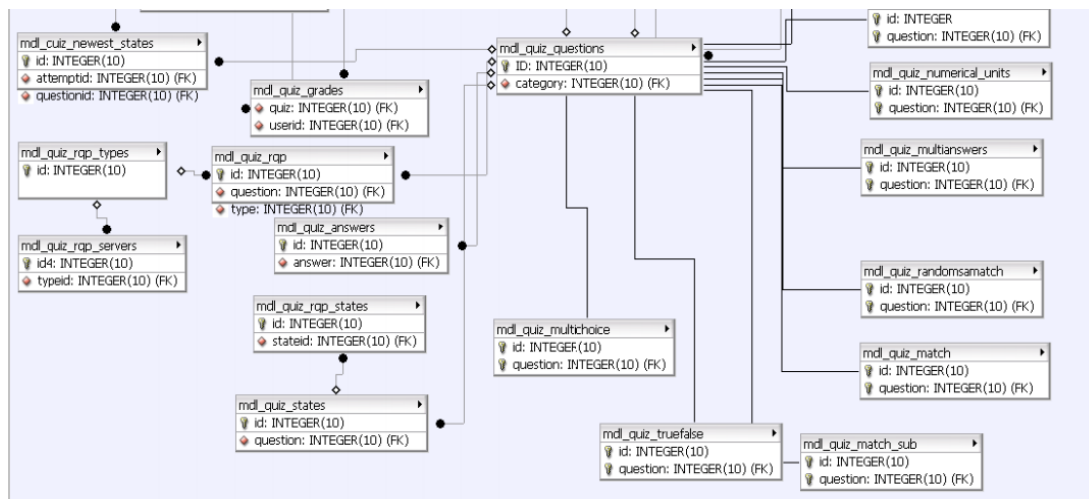


Figura III.18: Modelo entidad-relación de Moodle. Parte 5

Fuente de: (Duvan & Eduard, 2007)

Una vez realizado el análisis de la base de datos predeterminada de Moodle, se selecciona las tablas y campos necesarios, estos se describen a continuación:

Tablas	Campos
mood_user	id, username, password, firstname, lastname, phone2
mood_user_enrolments	enrolid, userid
mood_enrol	id, courseid
mood_course	id, fullname, shortname, visible
mood_course_modules	course, module, instance
mood_modules	id, name
mood_grade_items	iteminstance, courseid, itemmodule
mood_assignment	id, course, name, timeavailable, assignmenttype
mood_assign	id, course, name, due date
mood_assign_user_mapping	userid, assignment
mood_assign_grades	id, assignment, userid,

mood_assign_user_flags	assignment, userid,
sys_nota	nota_id, id, nota_nota, nota_grafico, nota_fecha
eduk_dispositivos	dispo_id, dispo_celular, dispo_pk

Tabla III.2: Tablas y campos usados en la aplicación móvil

Elaborado por: José Chacón

Cabe mencionar que la aplicación solo tiene permisos de lectura sobre la base de datos de Moodle, aquella es conectada con un web service encargado de obtener los datos respectivos de los estudiantes.

Con el propósito de no modificar las tablas que vienen por defecto, para poder almacenar las notas y fotos que ingresan los usuarios, se crea una tabla para manipular dicha información.

3.3.3. Definición del login y creación de funciones principales

Los colores utilizados en el login (inicio de sesión) son representativos de la PUCE Ambato, en la parte central superior de la pantalla se encuentra el logo institucional.

Para el inicio de sesión se utiliza dos *editext* con la finalidad de que el estudiante, ingrese el usuario y contraseña respectivamente, en la parte inferior se encuentra un button para acceder a la aplicación.

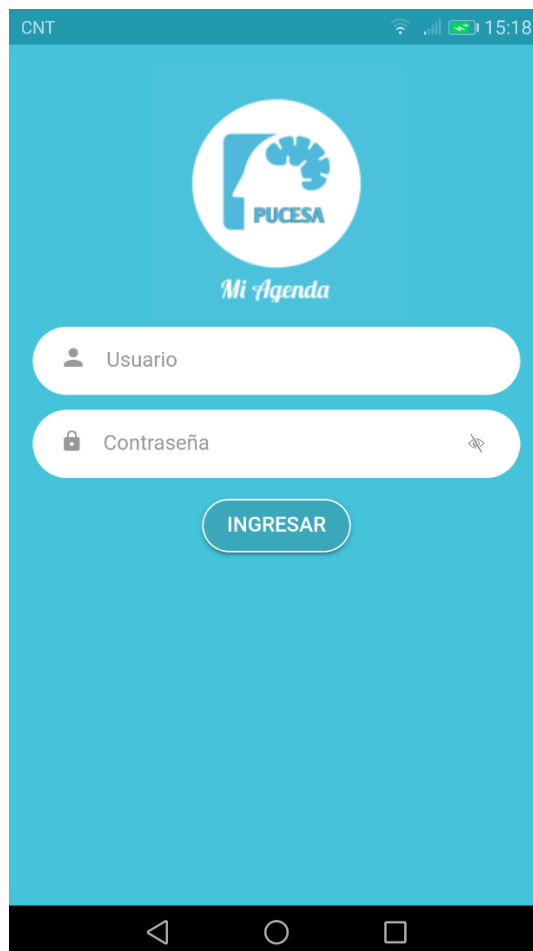


Figura III.19: Inicio de sesión de la aplicación móvil

Elaborado por: José Chacón

Para la pantalla de inicio se crea las siguientes funciones en el código:

```
import { Component, ViewChild } from '@angular/core';
import { Nav, Platform } from 'ionic-angular';
import { StatusBar } from '@ionic-native/status-bar';
import { SplashScreen } from '@ionic-native/splash-screen';
import { HomePage } from '../pages/home/home';
import { ListPage } from '../pages/list/list';
import { LoginPage } from '../pages/login/login';
import { AplicacionPage } from '../pages/aplicacion/aplicacion';
import { ListatareaPage } from '../pages/listatarea/listatarea';
import { AnotacionesPage } from '../pages/anotaciones/anotaciones';
import { AuthProvider } from '../providers/auth/auth';

@Component({
  templateUrl: 'app.html'
})
export class MyApp {
  @ViewChild(Nav) nav: Nav;
  rootPage: any = LoginPage;
```

```

pages: Array<{title: string, component: any}>;
pages_apl: Array<{title: string, component: any}>;
pages_out: Array<{title: string, component: any}>;
pages_ltareas: Array<{title: string, component: any}>;
pages_anotaciones: Array<{title: string, component: any}>;

constructor(public platform: Platform, public statusBar: StatusBar,
  public splashScreen: SplashScreen, public _AUTH: AuthProvider) {

  this.initializeApp();
  // used for an example of ngFor and navigation
  this.pages = [
    { title: 'Materias', component: AplicacionPage },
    { title: 'List', component: ListPage },
    { title: 'Logout', component: LoginPage }
  ];

  this.pages_apl = [
    { title: 'Materias', component: AplicacionPage }
  ];

  this.pages_out = [
    { title: 'Salir', component: LoginPage }
  ];

  this.pages_ltareas = [
    { title: 'Ultimas tareas', component: ListatareaPage }
  ];

  this.pages_anotaciones = [
    { title: 'Anotaciones', component: AnotacionesPage }
  ];
}

initializeApp() {
  this.platform.ready().then(() => {
    // Okay, so the platform is ready and our plugins are available.
    // Here you can do any higher level native things you might need.
    this.statusBar.styleDefault();
    this.splashScreen.hide();
    var notificationOpenedCallback = function(jsonData) {
      console.log('notificationOpenedCallback: ' + JSON.stringify(jsonData));
    };
    window["plugins"].OneSignal
      .startInit("9d8c60e0-9d26-4abf-8ce5-38c7a6ef7ff1", "904393618615")
      .handleNotificationOpened(notificationOpenedCallback)
      .endInit();

    window['plugins'].OneSignal.getIds(id=>{
      localStorage.setItem("playersID", id.userId);
    });
  });
}
openPage(page:any):void {
  this.nav.setRoot(page.component);
}
}

```

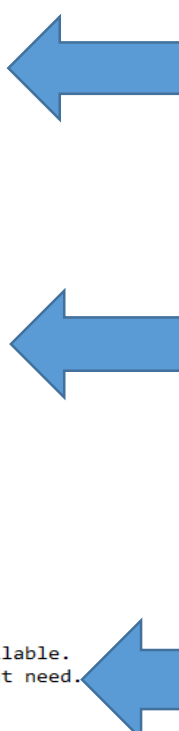


Figura III.20: Código de configuración inicial de la aplicación

Elaborado por: José Chacón

Dentro de la dirección `src/app` del proyecto, se encuentra el archivo `app.components.ts`, el cual permite: la inicialización del aplicativo (1), la declaración de las rutas con las respectivas pantallas (2) y la importación de plugins extras (3).

```

<ion-content padding class="color_fondo" >
  <div align="center">
    <ion-img width="120" height="120"
      src="assets/imgs/loginverfin1.png"></ion-img>
    </div>

  <form>
    <ion-list>
      <ion-item>
        <ion-label floating>Usuario</ion-label>
        <ion-input type="text" [(ngModel)]="loginInfo.txtUsuario"
          name="txtUsuario" required ></ion-input>
      </ion-item>
      <ion-item>
        <ion-label floating>Contraseña</ion-label>
        <ion-input type="password" [(ngModel)]="loginInfo.txtContrasena"
          name="txtContrasena" required ></ion-input>
      </ion-item>
      <ion-item>
        <div align="center">
          <button ion-button large (click)="loginUser();" >
            Ingresar al sistema </button>
          </div>
        </ion-item>
    </ion-list>
  </form>
</ion-content>

```

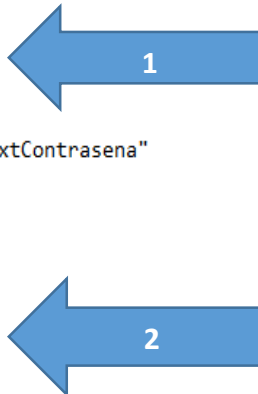


Figura III.21: Código de la pantalla inicio de sesión

Elaborado por: José Chacón

1.- Como es una vista, se utiliza código HTML para la presentación de la información y clases CSS para el diseño, en cuanto a la captura de datos se usa la directiva estructural ngModel, que permite enlazar los datos definidos en el controlador con los campos del formulario.

2.- Para el botón se utilizan directivas y propiedades establecidas por Ionic, además, contiene el evento clic que llama a la clase loginUser definida en el controlador.

```

import { Component } from '@angular/core';
import { IonicPage, NavController, NavParams } from 'ionic-angular';
import { HomePage } from '../home/home';
import { AplicacionPage } from '../aplicacion/aplicacion';
import { AlertController } from 'ionic-angular';
import { HttpClient } from '@angular/common/http';

@IonicPage()
@Component({
  selector: 'page-login',
  templateUrl: 'login.html',
})
export class LoginPage {
  loginInfo: any;
  ip_valor: any;
  constructor(public navCtrl: NavController, public navParams: NavParams,
    private alertCtrl: AlertController, public http: HttpClient) {
    this.loginInfo = {};
    this.ip_valor="http://www.agendaeis.com/edukar/ws_xp/";
  }

  ionViewDidLoad() {
    console.log('ionViewDidLoad LoginPage');
  }

  loginUser()
  {
    if(this.loginInfo.txtUsuario==undefined)
    {
      // Alertas original alert('Ingrese el usuario');
      let alert3 = this.alertCtrl.create({
        title: '¡Error de acceso!',
        subTitle: 'Ingrese el usuario',
        buttons: ['Aceptar']
      });
      alert3.present();
      //
      return false;
    }
    var data = {
      username: this.loginInfo.txtUsuario,
      password: this.loginInfo.txtContrasena,
      kp:localStorage['playersID']
    }
    var link = this.ip_valor+"login.php";
    this.http.post(link, data).subscribe(data => {
      console.log(data["token"]);
      localStorage.setItem("token", data["token"]);
      localStorage.setItem("nombres", data["nombre"]);
      localStorage.setItem("idus", data["idus"]);

      if(data["idus"]>0)
      {
        this.navCtrl.setRoot(AplicacionPage);
      }
      else
      {
        let alert2 = this.alertCtrl.create({
          title: '¡Error de acceso!',
          subTitle: 'Usuario o clave incorrecta',
          buttons: ['Aceptar']
        });
        alert2.present();
      }
    }, error => {
      alert("Dispositivo sin internet internet o usuario ya existe");
      console.error(error);
    });
  }
}

```

Figura III.22: Código de controlador que permite realizar el login en la app

Elaborado por: José Chacón

Este código permite la conexión al web service para validar el ingreso del alumno:

```
<?php
if (isset($_SERVER['HTTP_ORIGIN'])) {
    header("Access-Control-Allow-Origin: {$_SERVER['HTTP_ORIGIN']}");
    header('Access-Control-Allow-Credentials: true');
    header('Access-Control-Max-Age: 86400'); // cache for 1 day
}
// Access-Control headers are received during OPTIONS requests
if ($_SERVER['REQUEST_METHOD'] == 'OPTIONS') {

    if (isset($_SERVER['HTTP_ACCESS_CONTROL_REQUEST_METHOD']))
        header("Access-Control-Allow-Methods: GET, POST, OPTIONS");

    if (isset($_SERVER['HTTP_ACCESS_CONTROL_REQUEST_HEADERS']))
        header("Access-Control-Allow-Headers: {$_SERVER['HTTP_ACCESS_CONTROL_REQUEST_HEADERS']}");
    exit(0);
}

include("config.php");
$data = json_decode(file_get_contents("php://input"));
$password = $data->password;
$username = $data->username;
$kp=trim($data->kp);
$nombre='';
$id_usuario='';
$phone2='';
$verifica=0;
$userInfo = $db->query("SELECT * FROM mood_user WHERE username='".$username."'");
$userInfo = $userInfo->fetchAll();

foreach ($userInfo as $row) {
    $nombre=$row["firstname"];
    $id_usuario=$row["id"];
    $phone2=$row["phone2"];
    $verifica=password_verify($password, $row["password"]);
}
$token;
if ($verifica){
    $token = $id_usuario;
    $entrega=array(
        "token"=>$token,
        "nombre"=>str_replace(' ','',$nombre),
        "idus"=>$id_usuario
    );
    //-----
    $dispo_id=0;
    $usercel = $db->query("SELECT * FROM eduk_dispositivos WHERE dispo_celular=
        '".trim($phone2)."'");
    $usercel = $usercel->fetchAll();
    foreach ($usercel as $rowpx) {
        $dispo_id=$rowpx["dispo_id"];
    }
    if($dispo_id)
    {
        $uokin= $db->query("update eduk_dispositivos set dispo_celular=
            '".trim($phone2)."',dispo_pk='".$kp."' where dispo_id='".$dispo_id."'");
    }
    else
    {
        $uokin= $db->query("insert into eduk_dispositivos (dispo_celular,dispo_pk)
            values ('".trim($phone2)."', '".$kp."'");
    }
    //-----
    echo json_encode($entrega);
} else {
    $entrega=array(
        "token"=>0,
        "nombre"=>'',
        "idus"=>0
    );
    echo json_encode($entrega);
}
?>
```

Figura III.23: Código PHP para la validación del usuario

Elaborado por: José Chacón

3.3.4. Menú de navegación

En esta vista se tiene el menú desplegable que permite navegar entre las opciones que contiene la aplicación.

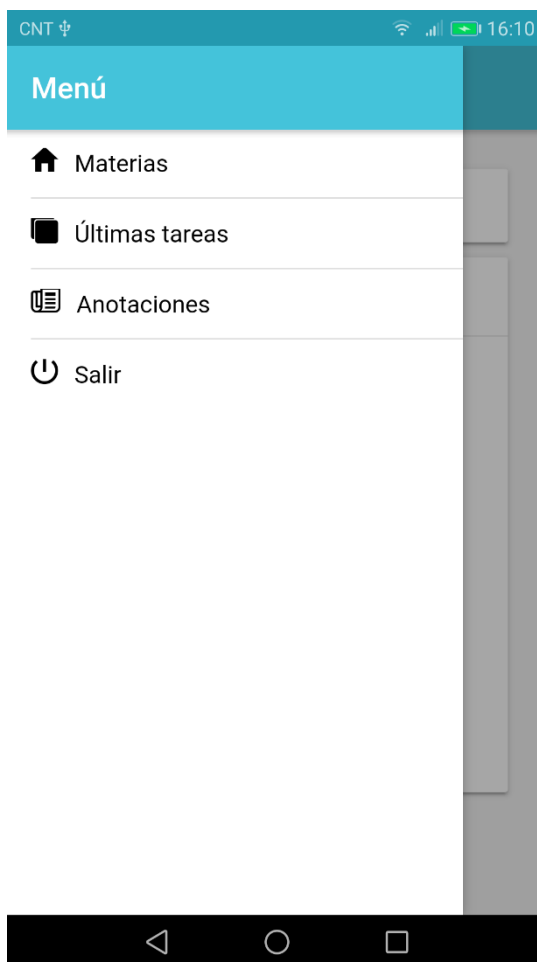


Figura III.24: Pantalla de menú principal

Elaborado por: José Chacón

3.3.5. Lista de materias que cursa el estudiante

Esta pantalla contiene el nombre del estudiante en la parte superior, seguidamente de la lista de las materias que cursa, con el número de tareas designadas.

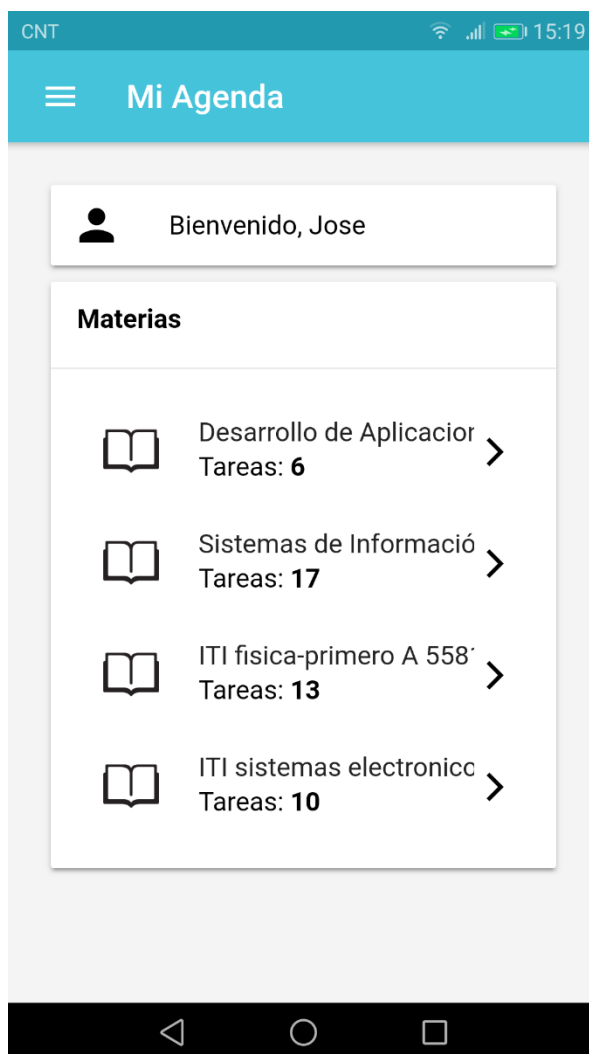


Figura III.25: Pantalla lista de materias del estudiante

Elaborado por: José Chacón

Para obtener el listado de las materias que se encuentra en la base de datos se usa el acceso al web service mediante los siguientes códigos.

En la carpeta pages/aplicación.html se tiene la vista de esta sección:

```

<ion-header class="color_fondo" >
  <ion-navbar >
    <button ion-button menuToggle>
      <ion-icon name="menu"></ion-icon>
    </button>
    <ion-title>Edukar</ion-title>
  </ion-navbar>
</ion-header>

<ion-content padding>|
  <ion-card>
    <ion-item>
      <ion-icon name="person" item-start></ion-icon>
      {{nusuario}}
    </ion-item>
  </ion-card>

  <ion-card>
    <ion-item>
      <div class="item item-text-wrap">
        <label class="item titulos_barra"
          style="font-weight:bold">Materias</label>
      </div>
    </ion-item><hr />
    <ion-list inset>
      <ion-item *ngFor="let lista_mn of listamateriadata"
        (click)="ver_tarea(lista_mn);" >
        <ion-icon name="document" item-start></ion-icon>
        <h2>{{lista_mn.shortname}} </h2>
        Tareas <b>{{lista_mn.totaltarea}}</b>
      </ion-item>
    </ion-list>
  </ion-card>
</ion-content>

```



Figura III.26: Código de la pantalla lista de materias

Elaborado por: José Chacón

1.- Despliega el nombre del alumno mediante el uso de la variable nusuario que se encuentra definida en el fichero ts que controla esta pantalla, en este caso es el archivo page/aplicación.ts

2.- La directiva estructural de angular *ngFor permite hacer una lista dinámica de los elementos, esto sirve para desplegar el nombre de las materias que toma el alumno a través de la variable lista_mn.shortname y el número de tareas con la variable lista_mn.totaltareas, definidas en el fichero nombrado anteriormente.

```

import { Component } from '@angular/core';
import { IonicPage, NavController, NavParams } from 'ionic-angular';
import { HttpClient } from '@angular/common/http';
import { ListatareaPage } from '../listatarea/listatarea';

@IonicPage()
@Component({
  selector: 'page-aplicacion',
  templateUrl: 'aplicacion.html',
})
export class AplicacionPage {
  nusuario: any
  ip_valor: any;
  lista_mn: any;
  listamateriadata: any;
  id:any;
  constructor(public navCtrl: NavController,
    public NavParams: NavParams,public http: HttpClient) {

    this.nusuario=localStorage['nombreus'];
    this.ip_valor="http://www.agendaeis.com/edukar/ws_xp/";
  }
  ionViewDidLoad() {
    console.log('ionViewDidLoad AplicacionPage');
    this.listar_materias();
  }

  listar_materias()
  {
    var datall = {
      datousuario: localStorage['idus']
    }
    var link = this.ip_valor+"listar_materia.php";
    this.http.post(link, datall).subscribe(data => {
      //console.log(data["opcion"]);
      this.listamateriadata=data;
    }, error => {
      alert("Aun no tiene materias asignadas");
    });
  }
  ver_tarea(lista_mn)
  {
    //alert(lista_mn.id);
    this.navCtrl.push(ListatareaPage,{
      idmateriax:lista_mn.id
    });
  }
}

```

Figura III.27: Código del controlador de la pantalla lista de materias

Elaborado por: José Chacón

El código de la figura 31, indica la conexión al web service, con el objetivo de poder utilizar el archivo listar_materia.php que se encuentran en el servidor y de esta manera acceder a la base de datos.

```

<?php
if (isset($_SERVER['HTTP_ORIGIN'])) {
    header("Access-Control-Allow-Origin: {$_SERVER['HTTP_ORIGIN']}");
    header('Access-Control-Allow-Credentials: true');
    header('Access-Control-Max-Age: 86400');    // cache for 1 day
}

// Access-Control headers are received during OPTIONS requests
if ($_SERVER['REQUEST_METHOD'] == 'OPTIONS') {

    if (isset($_SERVER['HTTP_ACCESS_CONTROL_REQUEST_METHOD']))
        header("Access-Control-Allow-Methods: GET, POST, OPTIONS");

    if (isset($_SERVER['HTTP_ACCESS_CONTROL_REQUEST_HEADERS']))
        header("Access-Control-Allow-Headers: {$_SERVER['HTTP_ACCESS_CONTROL_REQUEST_HEADERS']}");

    exit(0);
}
include("config.php");
$data = json_decode(file_get_contents("php://input"));
$bandera=0;
$datousuario = $data->datousuario;
$lista_manos="select mood_course.id,shortname,
count(mood_course.id) as totaltarea from mood_course
inner join mood_course_display on mood_course_display.course=mood_course.id
inner join mood_user on mood_user.id=mood_course_display.userid
inner join mood_assignment on mood_assignment.course=mood_course.id
where mood_user.id=".trim($datousuario)." and visible=1 group by mood_course.id";

$lista_manos="SELECT DISTINCT c.id,fullname as shortname,count(c.id) as totaltarea
FROM mood_user_enrolments ue, mood_enrol e, mood_course c
WHERE ue.enrolid = e.id
AND c.id = e.courseid
AND c.visible = 1
AND ue.userid=".trim($datousuario)."
group by c.id ";

$file = fopen("archivo.txt", "w");
fwrite($file, $lista_manos . PHP_EOL);
fclose($file);
$req_data = $db->query($lista_manos);
$req_data = $req_data->fetchAll();

foreach ($req_data as $row) {
    // $bandera++;
    // $concatena_cat.= $row["catag_nombre"].",";
    $bandera=1;
}

if($bandera==1)
{
    echo json_encode($req_data);
} else {
    echo "ERROR";
}
?>

```

Figura III.28: Código PHP de acceso a la base de datos para la lista de materias

Elaborado por: José Chacón

Una vez conectado al web service se utiliza el código de la figura anterior, el mismo que accede y extrae información de la base de datos con el uso de consultas, para mostrar en la pantalla lista de materias.

3.3.6. Lista de tareas

En esta pantalla se despliegan las tareas de cada materia que tiene el estudiante, cada una de ella consta con el nombre de la actividad, la fecha máxima de entrega y la descripción detallada de la misma:

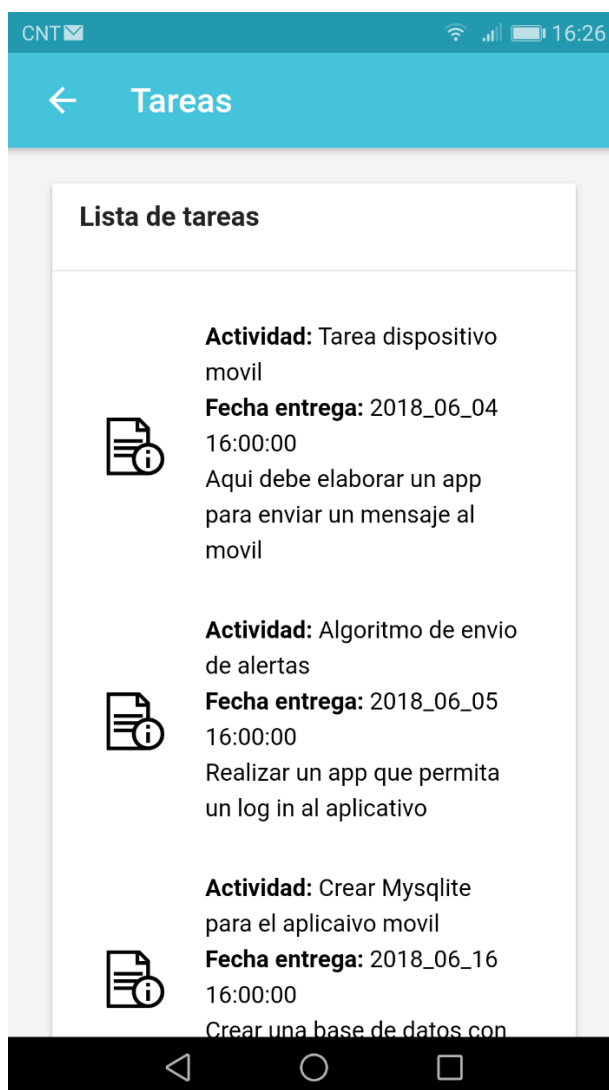


Figura III.29: Pantalla lista de tareas

Elaborado por: José Chacón

```

import { Component } from '@angular/core';
import { IonicPage, NavController, NavParams } from 'ionic-angular';
import { HttpClient } from '@angular/common/http';
@IonicPage()
@Component({
  selector: 'page-listatarea',
  templateUrl: 'listatarea.html',
})
export class ListatareaPage {
  idmateriax:any;
  ip_valor:any;
  listatareadata:any;
  stringdata:any;
  constructor(public navCtrl: NavController, public NavParams: NavParams,
  public http: HttpClient) {
    this.idmateriax=navParams.get('idmateriax');
    this.ip_valor="http://www.agendaeis.com/edukar/ws_xp/";
  }
  ionViewDidLoad() {
    console.log('ionViewDidLoad ListatareaPage');
    this.listar_tarea();
  }

  listar_tarea()
  {
    var data1 = {
      id_usuario: localStorage['idus'],
      idmateriap: this.idmateriax
    }
    var link = this.ip_valor+"listar_tarea.php";
    this.http.post(link, data1).subscribe(data => {
      this.listatareadata=data;
    }, error => {
      alert("Aun no tiene tareas disponibles");
    });
  }
}

```

Figura III.30: Código del controlador de la pantalla lista de tareas

Elaborado por: José Chacón

Con este código se establece la conexión al web service, en donde se encuentra el archivo llamado listar_tarea.php para poder usarlo y a través de él, acceder a la información.

```

<?php
if (isset($_SERVER['HTTP_ORIGIN'])) {
    header("Access-Control-Allow-Origin: {$_SERVER['HTTP_ORIGIN']}");
    header('Access-Control-Allow-Credentials: true');
    header('Access-Control-Max-Age: 86400'); // cache for 1 day
}
// Access-Control headers are received during OPTIONS requests

if ($_SERVER['REQUEST_METHOD'] == 'OPTIONS') {

    if (isset($_SERVER['HTTP_ACCESS_CONTROL_REQUEST_METHOD']))
        header("Access-Control-Allow-Methods: GET, POST, OPTIONS");

    if (isset($_SERVER['HTTP_ACCESS_CONTROL_REQUEST_HEADERS']))
        header("Access-Control-Allow-Headers:{$_SERVER['HTTP_ACCESS_CONTROL_REQUEST_HEADERS']}");

    exit(0);
}
include("config.php");
$data = json_decode(file_get_contents("php://input"));
$bandera=0;
$datousuario = $data->id_usuario;
$idmateriap = $data->idmateriap;

if($idmateriap)
{
    $lista_manos="select mood_course.id,shortname,mood_assignment.name,
from_unixtime(duedate,'%Y_%m_%d %H:%i:%s') as fecha,
REPLACE(REPLACE(intro,'<p>',''),'</p>','') as intro from mood_course
inner join mood_course_display on mood_course_display.course=mood_course.id
inner join mood_user on mood_user.id=mood_course_display.userid
inner join mood_assignment on mood_assignment.course=mood_course.id
where mood_user.id=".trim($datousuario)." and visible=1 and
mood_course.id=".$idmateriap." order by timedue desc";

    $lista_manos="SELECT a.course as id,'' as shortname,a.name,
from_unixtime(duedate,'%Y_%m_%d %H:%i:%s') as fecha,
REPLACE(REPLACE(intro,'<p>',''),'</p>','') as intro
FROM mood_assign a
LEFT JOIN mood_assign_grades g ON g.assignment = a.id
LEFT JOIN mood_assign_user_flags uf ON uf.assignment = a.id AND uf.userid = g.userid
LEFT JOIN mood_course_modules cm ON cm.course = a.course AND cm.instance = a.id
LEFT JOIN mood_modules md ON md.id = cm.module AND md.name = 'assign'
LEFT JOIN mood_grade_items gri ON gri.iteminstance = a.id AND gri.courseid = a.course
AND gri.itemmodule = md.name
LEFT JOIN mood_assign_user_mapping um ON g.id = um.userid AND um.assignment = a.id
where a.course=".$idmateriap." and deletioninprogress=0
ORDER BY duedate desc ";
}
else
{
    $lista_materia='';
    $lista_materia="SELECT DISTINCT c.id
FROM mood_user_enrolments ue, mood_enrol e, mood_course c
WHERE ue.enrolid = e.id
AND c.id = e.courseid
AND c.visible = 1
AND ue.userid=".trim($datousuario)."
group by c.id ";
}

```

```

$lista_manos="select mood_course.id,shortname,mood_assignment.name,
from_unixtime(timedue,'%Y_%m_%d %H:%i:%s') as fecha,
REPLACE(REPLACE(intro,'<p>',''),'</p>','') as intro from mood_course
inner join mood_course_display on mood_course_display.course=mood_course.id
inner join mood_user on mood_user.id=mood_course_display.userid
inner join mood_assignment on mood_assignment.course=mood_course.id
where mood_user.id=".trim($datousuario)." and visible=1 order by timedue desc";

$lista_manos="SELECT a.course as id,'' as shortname,a.name,
from_unixtime(duedate,'%Y_%m_%d %H:%i:%s') as fecha,
REPLACE(REPLACE(intro,'<p>',''),'</p>','') as intro
FROM mood_assign a
LEFT JOIN mood_assign_grades g ON g.assignment = a.id
LEFT JOIN mood_assign_user_flags uf ON uf.assignment = a.id AND uf.userid = g.userid
LEFT JOIN mood_course_modules cm ON cm.course = a.course AND cm.instance = a.id
LEFT JOIN mood_modules md ON md.id = cm.module AND md.name = 'assign'
LEFT JOIN mood_grade_items gri ON gri.iteminstance = a.id
AND gri.courseid = a.course AND gri.itemmodule = md.name
LEFT JOIN mood_assign_user_mapping um ON g.id = um.userid AND um.assignment = a.id
where a.course in (".$lista_materia.") and deletioninprogress=0
ORDER BY duedate desc ";

}

$file = fopen("ltarea.txt", "w");
fwrite($file, $lista_materia . PHP_EOL);
fclose($file);

$req_data = $db->query($lista_manos);
$req_data = $req_data->fetchAll();

foreach ($req_data as $row) {
    // $bandera++;
    // $concatena_cat.= $row["catag_nombre"].",";
    $bandera=1;
}

if($bandera==1)
{
    echo json_encode($req_data);
} else {
    echo "ERROR";
}

?>

```

Figura III.31: Código PHP de acceso a la base de datos para la lista de tareas

Elaborado por: José Chacón

Mediante el archivo PHP que se encuentra dentro del web service, se accede a la base de datos en el servidor y a través de consultas se obtiene información específica de las tareas, para ser mostrada en la aplicación.

3.3.7. Anotaciones

En esta pantalla el estudiante puede registrar anotaciones y fotos, cuenta con una caja de texto para escribir, un botón para seleccionar, otro para cargar la foto y el último para guardar las anotaciones; las cuales después de ser guardadas se muestran en la parte inferior de la pantalla y tienen la posibilidad de ser editadas o eliminadas:

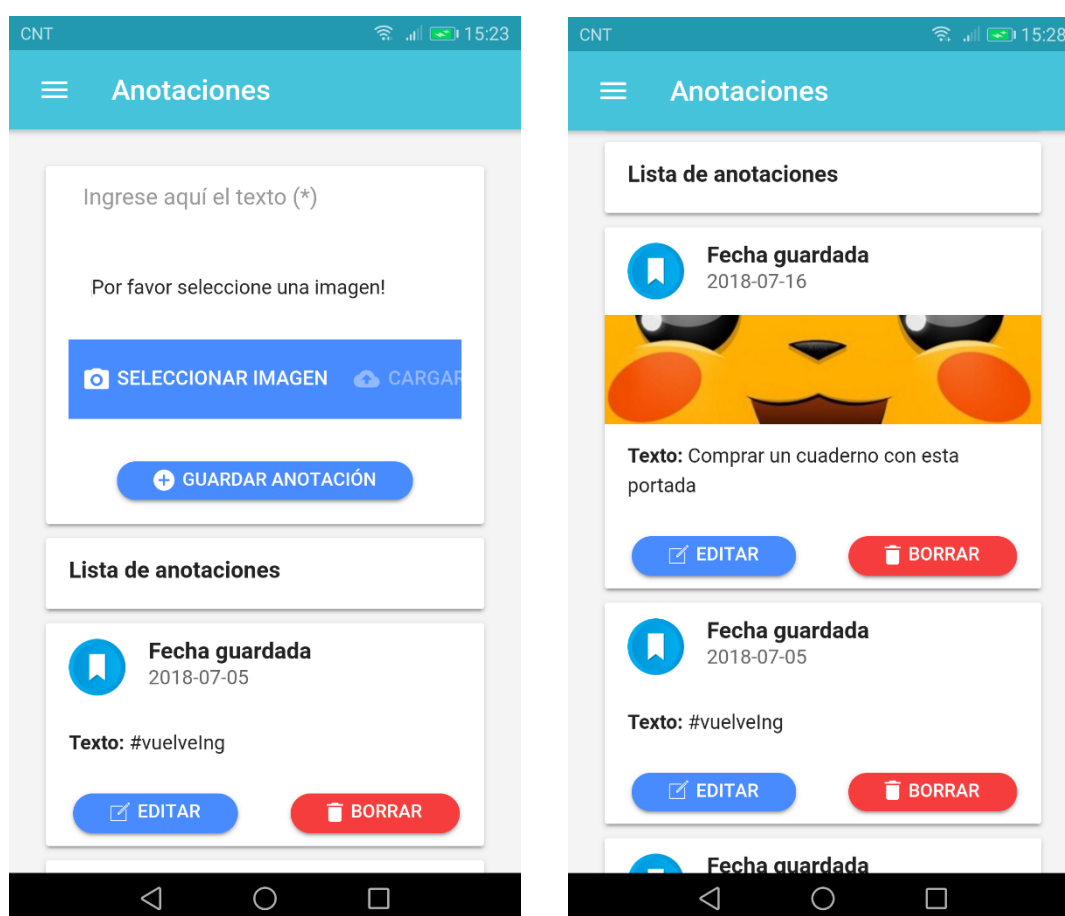


Figura III.32: Pantalla de anotaciones

Elaborado por: José Chacón

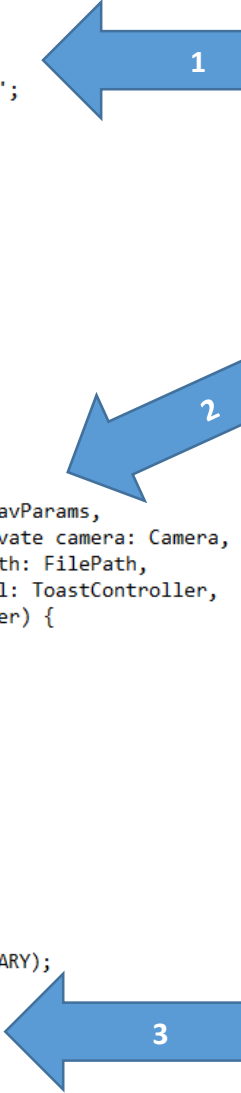
El código para implementar las anotaciones en el aplicativo es el siguiente:

```

import { Component } from '@angular/core';
import { IonicPage, NavController, NavParams, ActionSheetController, ToastController,
Platform,LoadingController, Loading } from 'ionic-angular';
import { AlertController } from 'ionic-angular';
import { HttpClient } from '@angular/common/http';
import { File } from '@ionic-native/file';
import { Transfer, TransferObject } from '@ionic-native/transfer';
import { FilePath } from '@ionic-native/file-path';
import { Camera } from '@ionic-native/camera';
declare var cordova: any;
@IonicPage()
@Component({
  selector: 'page- anotaciones',
  templateUrl: 'anotaciones.html',
})
export class AnotacionesPage {
  apunte: any;
  ip_valor: any;
  listatarnotas:any;
  lastImage: string = null;
  loading: Loading;
  nombre_imagen:any;
  constructor(public navCtrl: NavController, public NavParams: NavParams,
private alertCtrl: AlertController,public http: HttpClient,private camera: Camera,
private transfer: Transfer, private file: File, private filePath: FilePath,
public actionSheetCtrl: ActionSheetController, public toastCtrl: ToastController,
public platform: Platform, public loadingCtrl: LoadingController) {
    this.apunte = {};
    this.ip_valor="http://www.agendaeis.com/edukar/ws_xp/";
  }

  public presentActionSheet() {
    let actionSheet = this.actionSheetCtrl.create({
      title: 'Select Image Source',
      buttons: [
        {
          text: 'Load from Library',
          handler: () => {
            this.takePicture(this.camera.PictureSourceType.PHOTOLIBRARY);
          }
        },
        {
          text: 'Use Camera',
          handler: () => {
            this.takePicture(this.camera.PictureSourceType.CAMERA);
          }
        },
        {
          text: 'Cancel',
          role: 'cancel'
        }
      ]
    });
    actionSheet.present();
  }
  public takePicture(sourceType) {
    var options = {
      quality: 100,
      sourceType: sourceType,
      saveToPhotoAlbum: false,
      correctOrientation: true
    };
  }
}

```



```

this.camera.getPicture(options).then((imagePath) => {
  if (this.platform.is('android') &&
    sourceType === this.camera.PictureSourceType.PHOTOLIBRARY) {
    this.filePath.resolveNativePath(imagePath)
      .then(filePath => {
        let correctPath = filePath.substr(0, filePath.lastIndexOf('/') + 1);
        let currentName = imagePath.substring(imagePath.lastIndexOf('/') + 1,
          imagePath.lastIndexOf('?'));
        this.copyFileToLocalDir(correctPath, currentName, this.createFileName());
      });
    } else {
    var currentName = imagePath.substr(imagePath.lastIndexOf('/') + 1);
    var correctPath = imagePath.substr(0, imagePath.lastIndexOf('/') + 1);
    this.copyFileToLocalDir(correctPath, currentName, this.createFileName());
    }
  }, (err) => {
    this.presentToast('Error while selecting image.');
```

← 4

```

  });
}
private createFileName() {
  var d = new Date(),
  n = d.getTime(),
  newFileName = n + ".jpg";
  this.nombre_imagen= n + ".jpg";
  return newFileName;
}
private copyFileToLocalDir(namePath, currentName, newFileName) {
  this.file.copyFile(namePath, currentName, cordova.file.dataDirectory,
  | newFileName).then(success => {
    this.lastImage = newFileName;
  }, error => {
    this.presentToast('Error while storing file.');
```

← 5

← 6

```

  });
}

private presentToast(text) {
  let toast = this.toastCtrl.create({
    message: text,
    duration: 3000,
    position: 'top'
  });
  toast.present();
}

public pathForImage(img) {
  if (img === null) {
    return '';
  } else {
    return cordova.file.dataDirectory + img;
  }
}

ionViewDidLoad() {
  console.log('ionViewDidLoad AnotacionesPage');
  this.lista_annotaciones();
}

public uploadImage() {
  var url = "http://www.agendaeis.com/edukar/ws_xp/upload.php";
  var targetPath = this.pathForImage(this.lastImage);
  var filename = this.lastImage;
  var options = {
    fileKey: "file",
    fileName: filename,
    chunkedMode: false,
    mimeType: "multipart/form-data",
    params : {'fileName': filename}
  };
  const fileTransfer: TransferObject = this.transfer.create();
  this.loading = this.loadingCtrl.create({
    content: 'Uploading...',
```

← 7

← 8

← 9

```

  });
};

```

```

this.loading.present();
fileTransfer.upload(targetPath, url, options).then(data => {
  this.loading.dismissAll()
  this.presentToast('Image succesful uploaded.');
```

```

}, err => {
  this.loading.dismissAll()
  this.presentToast('Error while uploading file.');
```

```

});
}
guarda_nota()
{
  if(this.apunte.nota_nota==undefined)
  {
    let alert3 = this.alertCtrl.create({
      title: '¡Error de acceso!',
      subTitle: 'Por favor ingrese el texto',
      buttons: ['Aceptar']
    });
    alert3.present();
    //
    return false;
  }
  var data = {
    id_usuario: localStorage['idus'],
    nota_usuario:this.apunte.nota_nota,
    nota_id:this.apunte.nota_id,
    imagen_g:this.nombre_imagen
  }
  var link = this.ip_valor+"guardar_nota.php";
  this.http.post(link, data).subscribe(data => {
    let alert2 = this.alertCtrl.create({
      title: 'OK',
      subTitle: 'Registro guardado',
      buttons: ['Aceptar']
    });
    alert2.present();
    this.apunte.nota_id=''
    this.apunte.nota_nota='';
    this.lista_annotaciones();
  }, error => {
    alert("Dispositivo sin internet internet o usuario ya existe");
    console.error(error);
  });
}
lista_annotaciones()
{
  var datall = {
    id_usuario: localStorage['idus'],
    nota_idp:this.apunte.nota_id
  }
  var link = this.ip_valor+"listar_notas.php";
  this.http.post(link, datall).subscribe(data => {
    this.listatarnotas=data;
  }, error => {
    alert("Aun no tiene notas");
  });
}
}

```

10

11

```

editar_nota(lista_n)
{
  console.log(lista_n);
  var datall = {
    nota_idp: lista_n.nota_id
  }
  var link = this.ip_valor+"listar_vernota.php";
  this.http.post(link, datall).subscribe(data => {
    this.apunte.nota_nota=data["mensaje"];
    this.apunte.nota_id=data["nota_id"];
    this.lista_annotaciones();
  }, error => {
    alert("Aun no tiene materias asignadas");
  });
}
borrar_nota(lista_n)
{
  if(confirm("Esta seguro en borrar el registro?"))
  {
    var datall = {
      nota_idp: lista_n.nota_id
    }
    var link = this.ip_valor+"listar_borrarnota.php";
    this.http.post(link, datall).subscribe(data => {
      this.apunte.nota_nota='';
      this.apunte.nota_id='';
      alert(data["mensaje"]);
      this.lista_annotaciones();
    }, error => {
      alert("Aun no tiene materias asignadas");
    });
  }
}
}
}
}

```

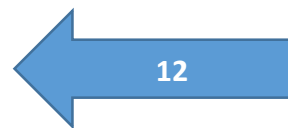


Figura III.33: Código del controlador para las anotaciones

Elaborado por: José Chacón

- 1.- A más de importar los servicios y complementos predeterminados, se añaden los servicios para poder utilizar la cámara del dispositivo, cargar fotos desde la galería, poder subir archivos, entre otros.
- 2.- Todo lo importado anteriormente se declara en el constructor, seguido se encuentra el acceso al web service.
- 3.- Crea el cuadro de diálogo de opciones: cargar fotos, usar la cámara o cancelar la acción.
- 4.- Sirve para obtener los datos de la imagen, a través del uso de una librería de Android.
- 5.- Crea un nuevo nombre para guardar la imagen.

- 6.- En caso de tomar una foto, se copia la imagen en una carpeta local.
- 7.- El ToastCtrl sirve para mostrar una notificación sobre una operación o para mostrar un mensaje del sistema.
- 8.- En este código se realiza lo siguiente: guarda en una variable el URL donde está el archivo PHP para subir la imagen, guarda el archivo a subir y el nombre en otras variables.
- 9.- Se usa el complemento de Ionic conocido como FileTransfer el cual permite subir la imagen enviando las anteriores variables como parámetros.
- 10.- Esta función envía un mensaje de alerta si no ingresa el texto, en caso contrario llama al archivo guardar_nota.php para ingresar la anotación en la base de datos.
- 11.- Permite visualizar todas las anotaciones guardadas a través del archivo listar_notas.php, encargado de hacer las consultas en la base de datos.
- 12.- Sirve para editar las anotaciones, se envía por el método post el id de la anotación a editar, junto con el nombre del archivo listar_vernota.php, el cual se encarga de realizar la consulta y mostrar los datos en la aplicación para editarla y finalmente guardarla.
- 13.- Este código permite eliminar una anotación, para ello se necesita el id de la nota con el nombre del archivo listar_borrarnota.php para enviarlo por el método post y eliminarla.

Estos son los archivos PHP, ubicados en el web service para hacer las consultas respectivas de las anotaciones en la base de datos:

```
<?php
if (isset($_SERVER['HTTP_ORIGIN'])) {
    header("Access-Control-Allow-Origin: {$_SERVER['HTTP_ORIGIN']}");
    header('Access-Control-Allow-Credentials: true');
    header('Access-Control-Max-Age: 86400'); // cache for 1 day
}
if ($_SERVER['REQUEST_METHOD'] == 'OPTIONS') {
    if (isset($_SERVER['HTTP_ACCESS_CONTROL_REQUEST_METHOD']))
        header("Access-Control-Allow-Methods: GET, POST, OPTIONS");
    if (isset($_SERVER['HTTP_ACCESS_CONTROL_REQUEST_HEADERS']))
        header("Access-Control-Allow-Headers: {$_SERVER['HTTP_ACCESS_CONTROL_REQUEST_HEADERS']}");
    exit(0);
}
include("config.php");
$data = json_decode(file_get_contents("php://input"));
$bandera=0;
$datousuario = str_replace("'",'', $data->id_usuario);
$nota_usuario = $data->nota_usuario;
$nota_id = $data->nota_id;
$fecha_hoy=date("Y-m-d");
$nota_grafico=$data->imagen_g;
$existe_nota='';
$file = fopen("archivo.txt", "w");
fwrite($file, $nota_id. PHP_EOL);
fclose($file);
if(trim($nota_id)>0)
{
    $lista_nota="select * from sys_nota where nota_id='".trim($nota_id);
    $existe_nota=0;
    $req_data = $db->query($lista_nota);
    $req_data = $req_data->fetchAll();
    foreach ($req_data as $row) {
        $existe_nota=$row["nota_id"];
    }
}

if($existe_nota)
{
    $actualiza="update sys_nota set nota_nota='". $nota_usuario."',
    nota_fecha='". $fecha_hoy."', nota_grafico='". $nota_grafico.'" where nota_id=". $existe_nota;
    $ok_data = $db->query($actualiza);
}
else
{
    $inserta_nota="insert into sys_nota (id,nota_nota,nota_fecha,nota_grafico)
    values ('". $datousuario."', '". $nota_usuario."', '". $fecha_hoy."', '". $nota_grafico.'")";
    $ok_data = $db->query($inserta_nota);
}
$file = fopen("archivo.txt", "a");
fwrite($file, $inserta_nota. PHP_EOL);
fclose($file);
$bandera=1;
$resultado_tr=array(
    "opcion">"1",
    "mensaje">"Registro guardado"
);
if($bandera==1)
{
    echo json_encode($resultado_tr);
} else {
    echo "ERROR";
}
?>
```

Figura III.34: Código PHP para guardar las notas

Elaborado por: José Chacón

```

<?php
if (isset($_SERVER['HTTP_ORIGIN'])) {
    header("Access-Control-Allow-Origin: {$_SERVER['HTTP_ORIGIN']}");
    header('Access-Control-Allow-Credentials: true');
    header('Access-Control-Max-Age: 86400');    // cache for 1 day
}
if ($_SERVER['REQUEST_METHOD'] == 'OPTIONS') {
    if (isset($_SERVER['HTTP_ACCESS_CONTROL_REQUEST_METHOD']))
        header("Access-Control-Allow-Methods: GET, POST, OPTIONS");
    if (isset($_SERVER['HTTP_ACCESS_CONTROL_REQUEST_HEADERS']))
        header("Access-Control-Allow-Headers:
            {$_SERVER['HTTP_ACCESS_CONTROL_REQUEST_HEADERS']}");
    exit(0);
}
include("config.php");
$data = json_decode(file_get_contents("php://input"));
$bandera=0;
$datousuario = str_replace("'",'', $data->id_usuario);
$nota_id = str_replace("'",'', $data->nota_idp);
$existe_nota='';
$lista_nota="select * from sys_nota where nota_id='".trim($nota_id);
$existe_nota='';
$req_data = $db->query($lista_nota);
$req_data = $req_data->fetchAll();
foreach ($req_data as $row) {
    $existe_nota=$row["nota_nota"];
    $existe_id=$row["nota_id"];
}
$bandera=1;
$req_data=array(
    "opcion"=>"1",
    "mensaje"=>$existe_nota,
    "nota_id"=>$existe_id
);
if($bandera==1)
{
    echo json_encode($req_data);
} else {
    echo "ERROR";
}
?>

```

Figura III.35: Código PHP para visualizar las notas

Elaborado por: José Chacón

```

<?php
if (isset($_SERVER['HTTP_ORIGIN'])) {
    header("Access-Control-Allow-Origin: {$_SERVER['HTTP_ORIGIN']}");
    header('Access-Control-Allow-Credentials: true');
    header('Access-Control-Max-Age: 86400');    // cache for 1 day
}
if ($_SERVER['REQUEST_METHOD'] == 'OPTIONS') {
    if (isset($_SERVER['HTTP_ACCESS_CONTROL_REQUEST_METHOD']))
        header("Access-Control-Allow-Methods: GET, POST, OPTIONS");
    if (isset($_SERVER['HTTP_ACCESS_CONTROL_REQUEST_HEADERS']))
        header("Access-Control-Allow-Headers:
            {$_SERVER['HTTP_ACCESS_CONTROL_REQUEST_HEADERS']}");
    exit(0);
}
include("config.php");
$data = json_decode(file_get_contents("php://input"));
$bandera=0;
$datousuario = str_replace("'",'', $data->id_usuario);
$nota_id = str_replace("'",'', $data->nota_idp);
$existe_nota='';
$lista_nota="delete from sys_nota where nota_id=".trim($nota_id);
$existe_nota='';
$req_data = $db->query($lista_nota);
$bandera=1;
$req_data=array(
    "opcion"=>"2",
    "mensaje"=>"Borrado con exito",
    "nota_id"=>""
);
if($bandera==1)
{
    echo json_encode($req_data);
} else {
    echo "ERROR";
}
?>

```

Figura III.36: Código PHP para borrar las notas

Elaborado por: José Chacón

3.3.8. Notificaciones *push*

Ionic Framework para incorporar notificaciones automáticas en las aplicaciones, utiliza el complemento conocido como OneSignal, el cual con pocas líneas de código permite implementarlas fácilmente.

A través de su página web, Ionic³ brinda la documentación necesaria para poder añadir este complemento dentro de un proyecto, por lo que solo hay que seguir los pasos correctamente para poderlo usar. También, la propia desarrolladora de OneSignal⁴ aporta con documentos sobre la instalación y configuración de su complemento.

Siguiendo la documentación se procede a aplicar en el proyecto:

```
import { BrowserModule } from '@angular/platform-browser';
import { ErrorHandler, NgModule } from '@angular/core';
import { IonicApp, IonicErrorHandler, IonicModule } from 'ionic-angular';
import { MyApp } from './app.component';
import { HomePage } from '../pages/home/home';
import { ListPage } from '../pages/list/list';
import { LoginPage } from '../pages/login/login';
import { AplicacionPage } from '../pages/aplicacion/aplicacion';
import { ListatareaPage } from '../pages/listatarea/listatarea';
import { AnotacionesPage } from '../pages/ anotaciones/ anotaciones';
import { StatusBar } from '@ionic-native/status-bar';
import { SplashScreen } from '@ionic-native/splash-screen';
import { AuthProvider } from '../providers/auth/auth';
import { HttpClientModule } from '@angular/common/http';
import { File } from '@ionic-native/file';
import { Transfer } from '@ionic-native/transfer';
import { FilePath } from '@ionic-native/file-path';
import { Camera } from '@ionic-native/camera';
import { OneSignal } from '@ionic-native/onesignal';
```



```
@NgModule({
  declarations: [
    MyApp,
    HomePage,
    ListPage,
    LoginPage,
    AplicacionPage,
    ListatareaPage,
    AnotacionesPage
  ],
```

³ <https://ionicframework.com/docs/native/onesignal>

⁴ <https://documentation.onesignal.com/docs/ionic-sdk-setup>

```

imports: [
  BrowserModule,
  HttpClientModule,
  IonicModule.forRoot(MyApp),
],
bootstrap: [IonicApp],
entryComponents: [
  MyApp,
  HomePage,
  ListPage,
  LoginPage,
  AplicacionPage,
  ListatareaPage,
  AnotacionesPage
],
providers: [
  StatusBar,
  SplashScreen,
  File,
  Transfer,
  Camera,
  OneSignal,
  FilePath,
  {provide: ErrorHandler, useClass: IonicErrorHandler},
  AuthProvider
]
})
export class AppModule {}

```



Figura III.37: Importación de servicio OneSignal

Elaborado por: José Chacón

En el archivo `src/app/app.module.ts` se importa los servicios de OneSignal (1) y se declara en el `providers` (2) para poder utilizarlo, esto se hace para cada servicio que se desea añadir.

```

import { Component, ViewChild } from '@angular/core';
import { Nav, Platform } from 'ionic-angular';
import { StatusBar } from '@ionic-native/status-bar';
import { SplashScreen } from '@ionic-native/splash-screen';
import { HomePage } from '../pages/home/home';
import { ListPage } from '../pages/list/list';
import { LoginPage } from '../pages/login/login';
import { AplicacionPage } from '../pages/aplicacion/aplicacion';
import { ListatareaPage } from '../pages/listatarea/listatarea';
import { AnotacionesPage } from '../pages/ anotaciones/ anotaciones';
import { AuthProvider } from '../providers/auth/auth';

@Component({
  templateUrl: 'app.html'
})
export class MyApp {
  @ViewChild(Nav) nav: Nav;
  rootPage: any = LoginPage;

  pages: Array<{title: string, component: any}>;
  pages_apl: Array<{title: string, component: any}>;
  pages_out: Array<{title: string, component: any}>;
  pages_ltareas: Array<{title: string, component: any}>;
  pages_anotaciones: Array<{title: string, component: any}>;

```

```

constructor(public platform: Platform, public statusBar: StatusBar,
public splashScreen: SplashScreen,public _AUTH: AuthProvider,
private oneSignal: OneSignal, private alertCtrl: AlertController) {
  this.initializeApp();
  // used for an example of ngFor and navigation
  this.pages = [
    { title: 'Materias', component: AplicacionPage },
    { title: 'List', component: ListPage },
    { title: 'Logout', component: LoginPage }
  ];
  this.pages_apl = [
    { title: 'Materias', component: AplicacionPage }
  ];
  this.pages_out = [
    { title: 'Salir', component: LoginPage }
  ];
  this.pages_ltareas = [
    { title: 'Ultimas tareas', component: ListatareaPage }
  ];
  this.pages_annotaciones = [
    { title: 'Anotaciones', component: AnotacionesPage }
  ];
}
initializeApp() {
  this.platform.ready().then(() => {
    // Okay, so the platform is ready and our plugins are available.
    // Here you can do any higher level native things you might need.
    this.statusBar.styleDefault();
    this.splashScreen.hide();
    var notificationOpenedCallback = function(jsonData) {
      console.log('notificationOpenedCallback: ' + JSON.stringify(jsonData));
    };
    window["plugins"].OneSignal
      .startInit("9d8c60e0-9d26-4abf-8ce5-38c7a6ef7ff1", "904393618615")
      .handleNotificationOpened(notificationOpenedCallback)
      .endInit();
    window['plugins'].OneSignal.getIds(id=>{
      localStorage.setItem("playersID",id.userId);
    });
  });
}
openPage(page:any):void {
  this.nav.setRoot(page.component);
}
}

```




Figura III.38: Código para utilizar las notificaciones

Elaborado por: José Chacón

Dentro del archivo `src/app/app.component.ts` se inyecta como dependencia `OneSignal` y `AlertController` (1) para poder mostrar las notificaciones.

En la parte inferior se encuentra la función en donde está el código para cargar el complemento cuando se inicialice la aplicación (2), además de contener el enlace entre la aplicación y `OneSignal` para enviar las notificaciones a los

dispositivos (3); para ello, se necesita el Google Server Api Key y Google Number. La manera de obtener estas llaves, es añadiendo un nuevo proyecto en Firebase, posteriormente se elige el proyecto creado previamente en Google Cloud Platform, después el sistema, de manera automática, genera las llaves únicas para poder usarlas con nuestro proyecto.

En el web service se encuentra el siguiente código:

```
<?php
if (isset($_SERVER['HTTP_ORIGIN'])) {
    header("Access-Control-Allow-Origin: {$_SERVER['HTTP_ORIGIN']}");
    header('Access-Control-Allow-Credentials: true');
    header('Access-Control-Max-Age: 86400');    // cache for 1 day
}
// Access-Control headers are received during OPTIONS requests
if ($_SERVER['REQUEST_METHOD'] == 'OPTIONS') {
    if (isset($_SERVER['HTTP_ACCESS_CONTROL_REQUEST_METHOD']))
        header("Access-Control-Allow-Methods: GET, POST, OPTIONS");
    if (isset($_SERVER['HTTP_ACCESS_CONTROL_REQUEST_HEADERS']))
        header("Access-Control-Allow-Headers:{$_SERVER['HTTP_ACCESS_CONTROL_REQUEST_HEADERS']}");
    exit(0);
}

date_default_timezone_set('America/Guayaquil');
include("config.php");
$data = json_decode(file_get_contents("php://input"));
$bandera=0;
$idpedido = str_replace("'",'', $data->idpedido);
$pedi_fecha=date("Y-m-d h:i:s");
$id=date("Ymdhis");
$gasec_idkey='7e193add-23f0-463f-afef-2f81c00e5211';

$resultado_tr=array(
    "opcion">"1",
    "mensaje">"Tareas pendientes..."
);
include("notificacion_test.php");
$mensaje="Tiene tareas pendientes...movil";
$api='9d8c60e0-9d26-4abf-8ce5-38c7a6ef7ff1';
$keyrest='YWQzYjllZjYtMDFjNS000DEwLThiZWVetYzk3ZjA5ZDg0M2M2';
$iduser=$gasec_idkey;
$response= sendMessageAll($mensaje,$api,$keyrest,$iduser);
$return["allresponses"]=$response;
$return=json_encode($return);
$bandera=1;
if($bandera==1)
{
    echo json_encode($resultado_tr);
} else {
    echo "ERROR";
}
}
|>
```

Figura III.39: Código PHP de las notificaciones

Elaborado por: José Chacón

Con este código se accede a la base de datos para saber las tareas que son activadas, una vez identificado el estado de la misma, se envía el mensaje de notificación, que puede contener cualquier texto, también es necesario el Google Server Api Key y el iduser para enviar las notificaciones a dispositivos específicos.

Una vez configurado el complemento OneSignal en el proyecto se obtiene el siguiente resultado:

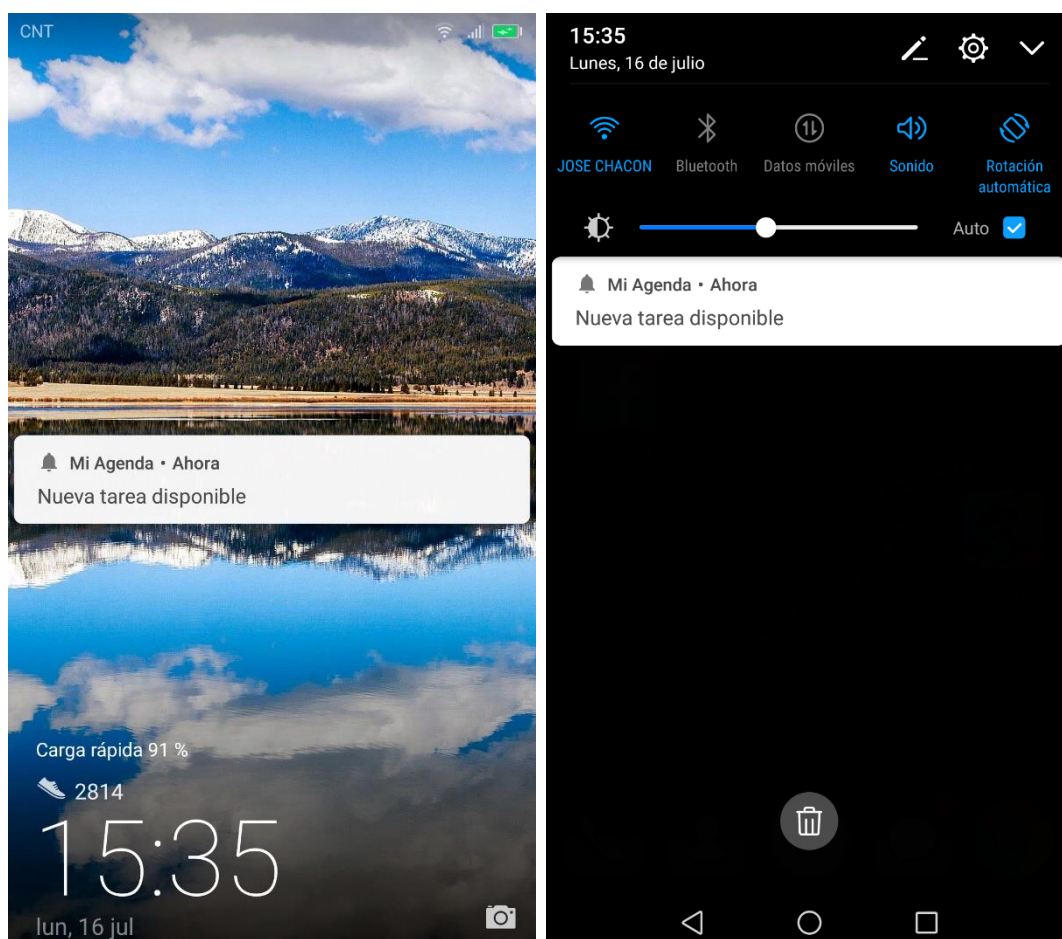


Figura III.40: Notificaciones

Elaborado por: José Chacón

3.4. Fase de Estabilización

En esta etapa se realizan reuniones en las que se va desarrollando y mejorando el aplicativo, para este fin se usa la siguiente plantilla donde se toma nota de cada reunión y poder aplicar los cambios si era necesario:

Módulo: Inicio de sesión

Nombre:	RF-01 Vista Login				
Descripción:	Ingreso del estudiante a la aplicación.				
Actores:	Usuario externo: Estudiantes de la institución educativa.				
Pre-Condiciones:	Estar registrado en la plataforma educativa Moodle.				
Flujo	Usuario externo: Ingresa al sistema, para esto se obtiene los datos de la tabla con los siguientes campos: Tabla: mood_user				
	Nro.	Nombre Campo	Tipo de dato	Tipo Registro	Obligatorio
	1	username	Alfanumérico	Ingresa el usuario	SI
	2	password	Alfanumérico (máximo 10 caracteres)	Ingresa el usuario	SI
Flujo Secundario	N / A				
Post-Condición	Si los datos son correctos el usuario ingresa a la aplicación				
Criterio de Aceptación	N / A				

Caso de Uso	<pre> graph LR Alumno((Alumno)) -.-> LOGIN((LOGIN)) LOGIN -.-> MIAGENDA((MI AGENDA)) </pre>
Observaciones/Acciones:	Indique mensaje de error, si el estudiante ingresa de manera incorrecta el usuario o contraseña

Tabla III.3: Test de mejora inicio de sesión

Elaborado por: José Chacón

Módulo: Materias

Nombre:	RF-02 Vista Lista de Materias				
Descripción:	Una vez dentro de la aplicación el estudiante puede ver las materias que recibe.				
Actores:	Usuario externo: Estudiante de la institución educativa.				
Pre-Condiciones:	Estar registrado en la plataforma educativa Moodle y matriculado en el semestre en curso.				
Flujo	Usuario externo: El estudiante puede ver la lista de materias tomadas en el semestre en curso: Tabla: mood_course				
	Nro.	Nombre Campo	Tipo Dato	Tipo Registro	Obligatorio
	1	mood_cours e.id	Numérico	Generado	SI
2	shortname	Alfanumérico	Ingresa el usuario Moodle	SI	

	3	visible	Numérico	Ingresar el usuario Moodle	SI
Flujo Secundario	N / A				
Post-Condición	El estudiante puede ver las tareas asignadas a la materia.				
Criterio de Aceptación	N / A				
Caso de Uso	<pre> graph TD Alumno((Alumno)) -.-> MI_Agenda([MI AGENDA]) MI_Agenda <-.-> Lista_Materias([LISTA DE MATERIAS]) </pre>				
Observaciones/ Acciones:	Ubicar un contador de tareas en la parte inferior de cada materia.				

Tabla III.4: Test de mejora materias

Elaborado por: José Chacón

Módulo: Tareas

Nombre:	RF-03 Vista Lista de Tareas
Descripción:	Una vez dentro de la aplicación el estudiante puede ver sus tareas pendientes.
Actores:	Usuario externo: Estudiante de la institución educativa.
Pre-Condiciones:	Estar registrado en la plataforma educativa Moodle y matriculado en el semestre en curso.

Flujo	Usuario externo: El estudiante puede ver la lista de tareas pendientes por materia: Tabla: mood_assignment				
	Nro.	Nombre Campo	Tipo Dato	Tipo Registro	Obligatorio
	1	name	Alfanumérico	Ingresa el usuario Moodle	SI
	2	from_unixtime	Unix Time	Ingresa el usuario Moodle	SI
3	intro	Alfanumérico	Ingresa el usuario Moodle	SI	
Flujo Secundario	N / A				
Post-Condición	El estudiante puede ver las tareas asignadas de cada materia.				
Criterio de Aceptación	N / A				
Caso de Uso	<pre> graph TD Alumno((Alumno)) -.-> MI_AGENDA([MI AGENDA]) MI_AGENDA -.-> LISTA_DE_MATERIAS([LISTA DE MATERIAS]) LISTA_DE_MATERIAS -.-> TAREAS([TAREAS]) </pre>				

Observaciones/acciones:	Incluir en cada tarea la fecha máxima de presentación e íconos de advertencia
--------------------------------	---

Tabla III.5: Test de mejora tareas

Elaborado por: José Chacón

Módulo: Anotaciones

Nombre:	RF-04 Vista registro de anotaciones				
Descripción:	Una vez dentro de la aplicación el estudiante puede registrar sus anotaciones para recordar algo pendiente.				
Actores:	Usuario externo: El estudiante de la institución educativa.				
Pre-Condiciones:	Estar registrado en la plataforma educativa Moodle y matriculado en el semestre en curso.				
Flujo	Usuario externo: El estudiante puede ver la lista de anotaciones y registrar nuevas anotaciones: Tabla: sys_nota				
	Nro.	Nombre Campo	Tipo Dato	Tipo Registro	Obligatorio
	1	nota_id	Numérico	Generado	SI
	2	nota_nota	Alfanumérico	Ingresa el estudiante	SI
	3	nota_grafico	Alfanumérico	Ingresa el estudiante	SI
	4	nota_fecha	Fecha	Ingresa el estudiante	SI
Flujo Secundario	N / A				
Post-Condición	El estudiante puede ver las anotaciones registradas y registrar nuevas anotaciones.				

Criterio de Aceptación	N / A
Caso de Uso	<pre> graph TD ALUMNO((ALUMNO)) -.-> MI_AGENDA([MI AGENDA]) MI_AGENDA -.-> NOTAS([NOTAS]) NOTAS -.-> NUEVA_NOTA([NUEVA NOTA]) NOTAS -.-> EDITAR_NOTA([EDITAR NOTA]) NOTAS -.-> ELIMINAR_NOTA([ELIMINAR NOTA]) NOTAS -.-> LISTAR([LISTAR]) </pre> <p>The diagram shows an actor 'ALUMNO' connected to a use case 'MI AGENDA'. 'MI AGENDA' is connected to 'NOTAS'. 'NOTAS' is connected to four other use cases: 'NUEVA NOTA', 'EDITAR NOTA', 'ELIMINAR NOTA', and 'LISTAR'.</p>
Observaciones/acciones:	Incluir la posibilidad de tomar y guardar fotos.

Tabla III.6: Test de mejora notas

Elaborado por: José Chacón

Módulo: Notificaciones

Nombre:	RF-05 Notificaciones				
Descripción:	Una vez dentro de la aplicación el estudiante puede recibir notificaciones de las nuevas tareas.				
Actores:	Usuario externo: El estudiante de la institución educativa.				
Pre-Condiciones:	Estar registrado en la plataforma educativa Moodle y matriculado en el semestre en curso. Haber iniciado sesión desde dispositivo, mínimo una vez.				
Flujo	Usuario externo: El estudiante puede recibir notificaciones de tareas nuevas, para ello se crea una tabla para registrar el usuario con el id del dispositivo para enviar notificaciones a determinados estudiantes: Tabla: eduk_dispositivos				
	Nro.	Nombre Campo	Tipo Dato	Tipo Registro	Obligatorio

	1	dispo_id	Integer	Generado	SI
	2	dispo_celular	Alfanumérico	Ingresa el estudiante	SI
	3	dispo_pk	Alfanumérico	Generado	SI
Flujo Secundario	N / A				
Post-Condición	El estudiante puede recibir notificaciones.				
Criterio de Aceptación	N / A				
Caso de Uso	<pre> graph TD ALUMNO((ALUMNO)) ONE_SIGNAL((ONE SIGNAL)) MI_AGENDA((MI AGENDA)) TAREAS((TAREAS)) ONE_SIGNAL -.-> NOTIFICACION ALUMNO MI_AGENDA -.-> TAREAS TAREAS -.-> ONE_SIGNAL </pre>				
Observaciones/acciones:	N/A				

Tabla III.7: Test de mejora notificaciones

Elaborado por: José Chacón

3.5. Fase de pruebas

3.5.1. Emulador de aplicaciones Ionic

Para verificar el diseño y realizar las pruebas de funcionamiento de la aplicación durante el desarrollo, se utiliza un emulador. Para esto, Ionic Framework tiene el siguiente comando para activarlo:

- ionic serve

Esta línea de código permite compilar el aplicativo y desplegarlo en un navegador, el resultado es el siguiente:

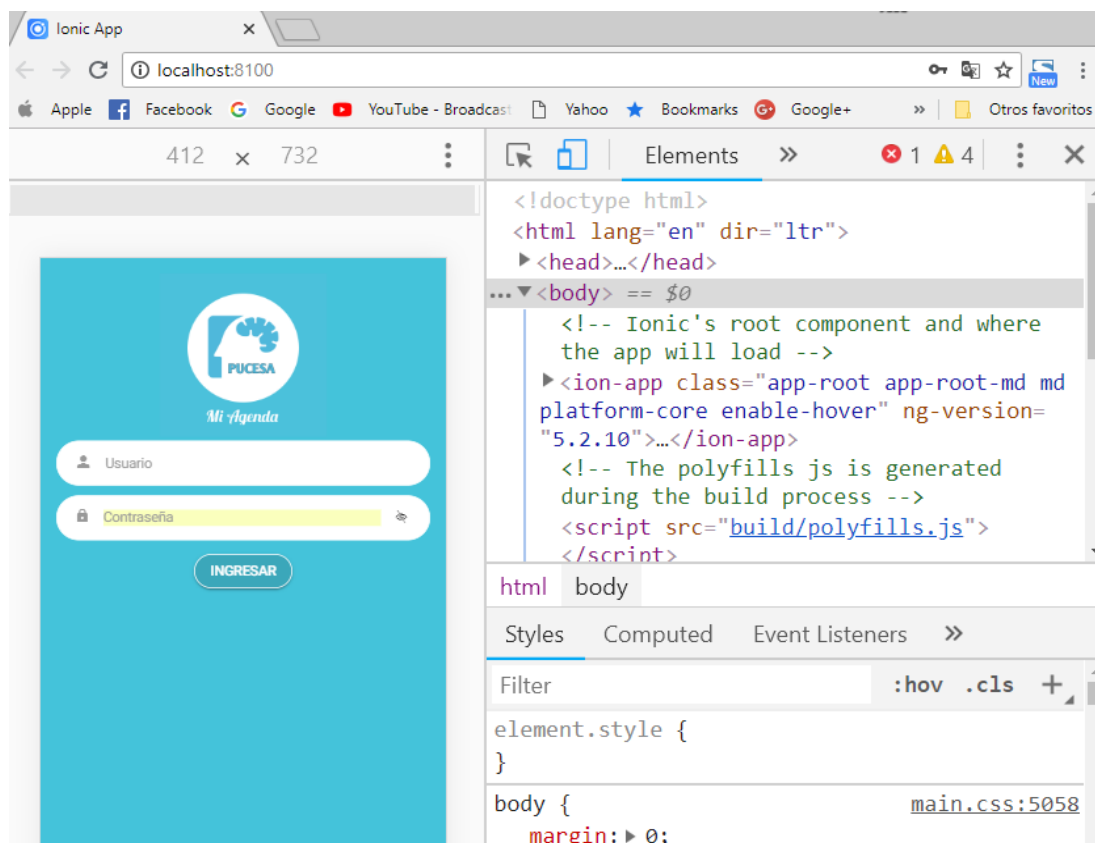


Figura III.41: Vista del emulador para realizar las pruebas

Elaborado por: José Chacón

También es posible enviar la compilación directo al dispositivo móvil con el siguiente comando:

- ionic run android –device

3.5.2. Generación del apk

Una vez elaboradas las pruebas de manera local a través del emulador, se usa el siguiente comando para generar el apk del aplicativo:

- ionic build android

Al ejecutarse, genera un archivo en la siguiente ruta .../platforms/android/build/outputs/apk un android-debug.apk, que es el apk para instalar de manera manual la aplicación en un teléfono con sistema operativo Android; el mismo que se puede descargar desde Dropbox⁵; luego se procede a realizar las pruebas respectivas a los estudiantes:

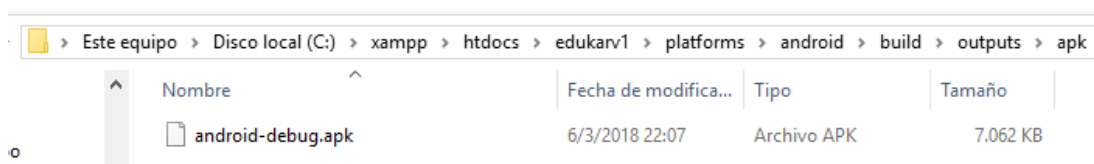


Figura III.42: Carpeta donde se genera el apk

Elaborado por: José Chacón

3.5.3. Pruebas de caja negra

Según (Pressman, 2005) a diferencia de las pruebas de caja blanca que se encargan de verificar el comportamiento interno y la estructura lógica de un programa, las pruebas de caja negra se centran en los requisitos funcionales del software y se realizan en la interfaz gráfica del mismo.

Las pruebas de caja negra permiten encontrar errores como:

- Falta de funciones.
- Errores en la interfaz.
- Errores de acceso a base de datos externas.
- Errores de comportamiento.
- Errores de inicialización y de terminación.

⁵ Enlace de descarga apk: <https://www.dropbox.com/s/tvns4zbdsoqwtpa/app-debug.apk?dl=0>

Una vez terminado el desarrollo de la aplicación móvil, se lleva a cabo la prueba de caja negra, con la finalidad de comprobar el correcto funcionamiento o encontrar deficiencias, por parte de los usuarios, al usar el aplicativo y ser corregidas si fuese el caso.

Para ello, se aplica dos tipos de encuesta: una dirigida a los estudiantes con lenguaje común y otra dirigida a un docente de la institución con conocimientos informáticos.

A continuación, se muestra el formato de las pruebas aplicadas con los respectivos aspectos que fueron evaluados:

Ámbito: estructura de la aplicación

La aplicación móvil "Mi Agenda", mantiene todas las páginas estructuradas	De acuerdo <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> en desacuerdo
La aplicación móvil "Mi Agenda", mantiene elementos internos (ej.: áreas de texto, imágenes, menú, botones) que siguen un formato a lo largo del aplicativo	De acuerdo <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> en desacuerdo
La distribución de los elementos estructurales (ej. scroll, áreas de contenido, botones, menú) es adecuada.	De acuerdo <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> en desacuerdo
La cantidad de los elementos estructurales utilizados es apropiada	De acuerdo <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> en desacuerdo
La distribución de los elementos estructurales mantiene consistencia a lo largo de la aplicación móvil "Mi Agenda"	De acuerdo <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> en desacuerdo

Ámbito: operación de la aplicación

La aplicación móvil "Mi agenda" cuenta con un acceso seguro al sistema	De acuerdo <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> en desacuerdo
Es fácil desplazarse entre las páginas en la aplicación móvil "Mi Agenda".	De acuerdo <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> en desacuerdo
La aplicación móvil "Mi agenda" funciona correctamente (no se perciben errores)	De acuerdo <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> en desacuerdo
La ejecución de tareas en la aplicación móvil "Mi Agenda" (ej.: navegar por las páginas, hacer clic en botones, seleccionar opciones, entre	De acuerdo <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> en desacuerdo

otros) mantiene un formato a lo largo de la aplicación	
La aplicación móvil “Mi Agenda” ejecuta operaciones CRUD (crear, visualizar, actualizar y borrar) en las notas, sin inconvenientes	De acuerdo <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> en desacuerdo
Se identifican fácilmente los iconos, listas, resultados, zonas activas y el tipo de acción que se desea ejecutar	De acuerdo <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> en desacuerdo
La velocidad de operaciones dentro de la aplicación “Mi Agenda” es aceptable	De acuerdo <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> en desacuerdo

Ámbito: apariencia

Los tipos de fuentes que se utilizan en la aplicación móvil “Mi Agenda”, permite diferenciar el contenido que se presenta (ej.: Títulos de actividades, fechas de entrega, descripción de tareas, entre otros)	De acuerdo <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> en desacuerdo
Los títulos en cada página de la aplicación móvil “Mi Agenda” facilitan la navegación	De acuerdo <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> en desacuerdo
Los colores utilizados en la aplicación móvil “Mi Agenda” son apropiados	De acuerdo <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> en desacuerdo
La aplicación móvil “Mi Agenda” mantiene un mismo formato respecto a los colores que mejoran el entendimiento del usuario	De acuerdo <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> en desacuerdo

Ámbito: contenido

La fiabilidad mostrada en la aplicación móvil “Mi Agenda”, refleja la información de la plataforma educativa Moodle	De acuerdo <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> en desacuerdo
La aplicación móvil “Mi Agenda” cumple con el objetivo de notificar tareas pendientes cuando son activadas en la plataforma educativa	De acuerdo <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> en desacuerdo
El contenido de las notificaciones que llegan al dispositivo móvil es claro y preciso.	De acuerdo <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> en desacuerdo
La aplicación móvil “Mi Agenda” despliega correctamente la lista de materias donde el alumno está matriculado	De acuerdo <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> en desacuerdo
Al seleccionar una materia en la aplicación “Mi Agenda” se puede visualizar correctamente la lista de tareas con su respectiva actividad, fecha de entrega y descripción	De acuerdo <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> en desacuerdo
En la aplicación “Mi Agenda” se visualiza correctamente el contenido de las anotaciones guardadas	De acuerdo <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> en desacuerdo

Tabla III.8: Formato de prueba para los estudiantes

Modificado de: (Acurio, 2018)

Prueba: Inicio de sesión

Caso de prueba: Inicio de sesión		Identificador caso de prueba: CP01	
		Fecha:	
Descripción: Verificar el ingreso del estudiante a la aplicación móvil			
Precondiciones: Se debe ingresar correctamente el usuario y contraseña con las cuales el estudiante está registrado en la plataforma educativa Moodle			
Pasos y condiciones de ejecución:			
Condición	Entrada	Datos	Resultado esperado
Ingreso de un usuario y contraseña registrado en el Moodle	Usuario Contraseña	adminusuario 098131596_Jos	Ingreso a la aplicación móvil
Ingreso de un usuario y contraseña no registrado en el Moodle o incorrecto	Usuario Contraseña	Cualquiera Cualquiera	Muestra una alerta indicando que el usuario y/o contraseña es incorrecta
Usuario y/o contraseña en blanco	Usuario Contraseña	En blanco En blanco	
Resultados del caso de prueba:			
Muy satisfactorio <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> Insatisfactorio			
Observación:			
Responsable ejecución:			
Firma:			

Tabla III.9: Formato de prueba para el docente. Parte 1

Elaborado por: José Chacón

Prueba: Lista de materias

Caso de prueba: Lista de materias		Identificador caso de prueba: CP02	
		Fecha:	
Descripción: Comprobar el despliegue de la lista de materias en las que está matriculado el estudiante			
Precondiciones: El estudiante debe iniciar correctamente la sesión y estar matriculado en alguna materia			
Paso y condiciones ejecución:			
Condición	Entrada	Datos	Resultado esperado
Correcto inicio de sesión	Usuario Contraseña	adminusuario 098131596_Jos	Visualización de las materias que toma el estudiante
Resultados del caso de prueba:			
Muy satisfactorio <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> Insatisfactorio			
Observación:			

Responsable ejecución:
Firma:

Tabla III.10: Formato de prueba para el docente. Parte 2

Elaborado por: José Chacón

Prueba: Lista de tareas

Caso de prueba: Lista de tareas		Identificador caso de prueba: CP03	
		Fecha:	
Descripción: Comprobar el correcto despliegue de la lista de tareas del estudiante con su respectiva información (actividad, fecha de entrega y descripción).			
Precondiciones: El estudiante debe iniciar correctamente la sesión y estar matriculado en alguna materia			
Pasos y condiciones de ejecución:			
Condición	Entrada	Datos	Resultado esperado
Estar matriculado en una materia y activada la tarea	Seleccionar la materia	No aplica	Visualización de las tareas activadas con su respectiva información
	Seleccionar en el menú la opción últimas tareas	No aplica	Visualización de todas las tareas activadas de las materias con su respectiva información
Activar una tarea desde la plataforma educativa Moodle	Creación de una tarea por el docente	Información de la tarea	En la aplicación verificar la visualización de la tarea deslizando con el dedo hacia abajo para actualizar la página
Resultados del caso de prueba:			
Muy satisfactorio <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> Insatisfactorio			
Observación:			
Responsable ejecución:			
Firma:			

Tabla III.11: Formato de prueba para el docente. Parte 3

Elaborado por: José Chacón

Prueba: Anotaciones

Caso de prueba: Anotaciones		Identificador caso de prueba: CP04	
		Fecha:	
Descripción: Verificar las operaciones CRUD (crear, visualizar, actualizar y borrar) que se puede hacer en las anotaciones			
Precondiciones: El estudiante debe iniciar correctamente la sesión			
Pasos y condiciones de ejecución:			
Condición	Entrada	Datos	Resultado esperado
Correcto inicio de sesión	Ingresar texto Seleccionar y cargar imagen	Cualquiera Cualquiera	Despliegue de una alerta indicando que la anotación fue guardada y visualización en la parte inferior de la anotación registrada
	Ingresar solo texto	Cualquiera	
	Ingresar solo foto y cargar imagen	Cualquiera	Despliegue de una alerta indicando que se debe ingresar texto
Elegir una anotación registrada	Pulsar el botón editar, cambiar el texto o imagen	Cualquiera	Despliegue de una alerta indicando que la anotación fue guardada y visualización en la parte inferior de la anotación modificada
Tener anotaciones registrada	Pulsar el botón borrar y confirmar la eliminación	No aplica	Despliegue de una alerta indicando que la anotación fue borrada y verificación que ya no exista la misma
Tener anotaciones registradas	Seleccionar en el menú la opción anotaciones	No aplica	Visualización de las notas registradas
Dejar los campos en blanco	Pulsar el botón guardar	No aplica	Despliegue de una alerta indicando que se debe ingresar texto
Resultados: Muy satisfactorio <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> Insatisfactorio			

Observación:
Responsable ejecución:
Firma:

Tabla III.12: Formato de prueba para el docente. Parte 4

Elaborado por: José Chacón

Prueba: Notificaciones

Caso de prueba: Notificaciones	Identificador caso de prueba: CP05		
	Fecha: 17/7/2018		
Descripción: Recibir notificaciones cuando las tareas son activadas en la plataforma educativa Moodle			
Precondiciones: Estar registrado en la plataforma educativa Moodle y matriculado en una materia Debe iniciar sesión desde el dispositivo al menos una vez			
Pasos y condiciones de ejecución:			
Condición	Entrada	Datos	Resultado esperado
Iniciar sesión correctamente	El docente debe activar una tarea desde la plataforma educativa Moodle	Información de la tarea	Despliegue de una alerta indicando que existe una nueva tarea disponible
Salir de aplicación y bloquear el dispositivo		Información de la tarea	Notificación indicando que existe una nueva tarea disponible
Iniciar sesión correctamente	El docente puede actualizar una tarea desde la plataforma educativa Moodle	Información de la tarea	Despliegue de una alerta indicando que una tarea fue actualizada
Salir de aplicación y bloquear el dispositivo		Información de la tarea	Notificación indicando que una tarea fue actualizada
Resultados del caso de prueba:			
Muy satisfactorio <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> Insatisfactorio			
Observación:			
Responsable ejecución:			
Firma:			

Tabla III.13: Formato de prueba para el docente. Parte 5

Elaborado por: José Chacón

CAPÍTULO IV

ANÁLISIS DE RESULTADOS

4.1. Resultado de la prueba a los estudiantes

Antes de aplicar la prueba, se selecciona del universo de estudiantes a un grupo de ellos, mediante la técnica de muestreo selectivo; ya que permite escoger aquellas unidades que aportan con información esencial y son de fácil acceso.

La prueba es aplicada a 11 estudiantes pertenecientes al primer y segundo semestre de la Escuela de Ingeniería en Sistemas de la PUCE Ambato, en el mes de julio del 2018. Con esto se busca evaluar las diferentes características de la aplicación móvil desarrollada, agrupadas en los siguientes ámbitos: estructura, operación, apariencia y contenidos.

A continuación, se muestra los resultados obtenidos de la prueba:

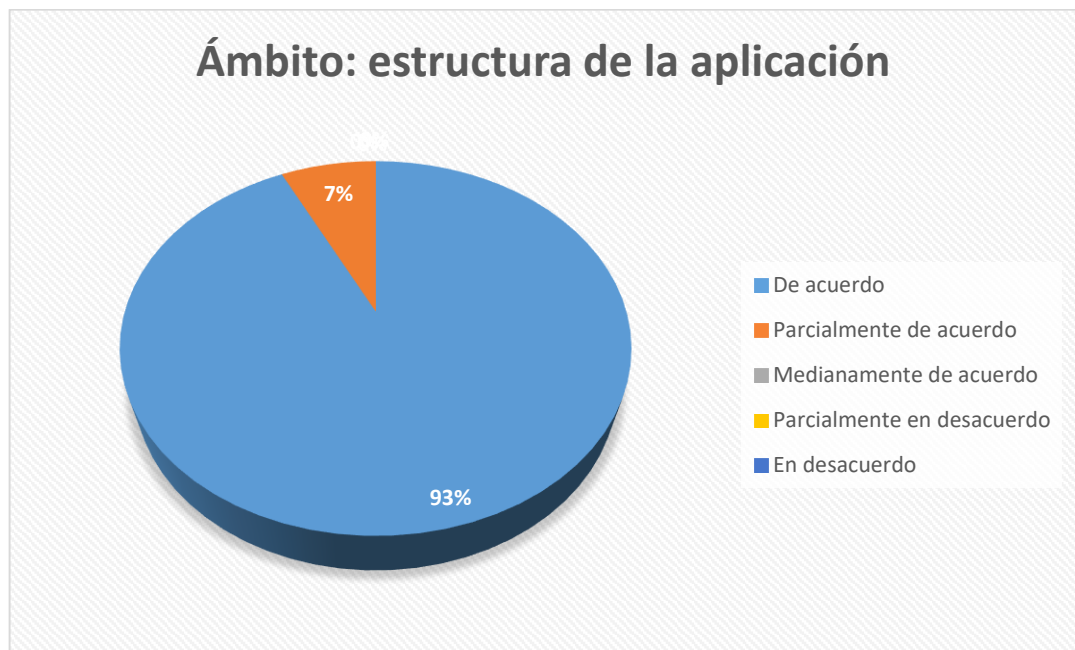


Gráfico IV.1: Resultado ámbito estructura de la aplicación

Elaborado por: José Chacón

Interpretación

La mayoría de estudiantes coinciden en que las páginas son estructuradas, los elementos internos siguen el mismo formato en toda la aplicación, la cantidad de los mismos es adecuada y su distribución es coherente.

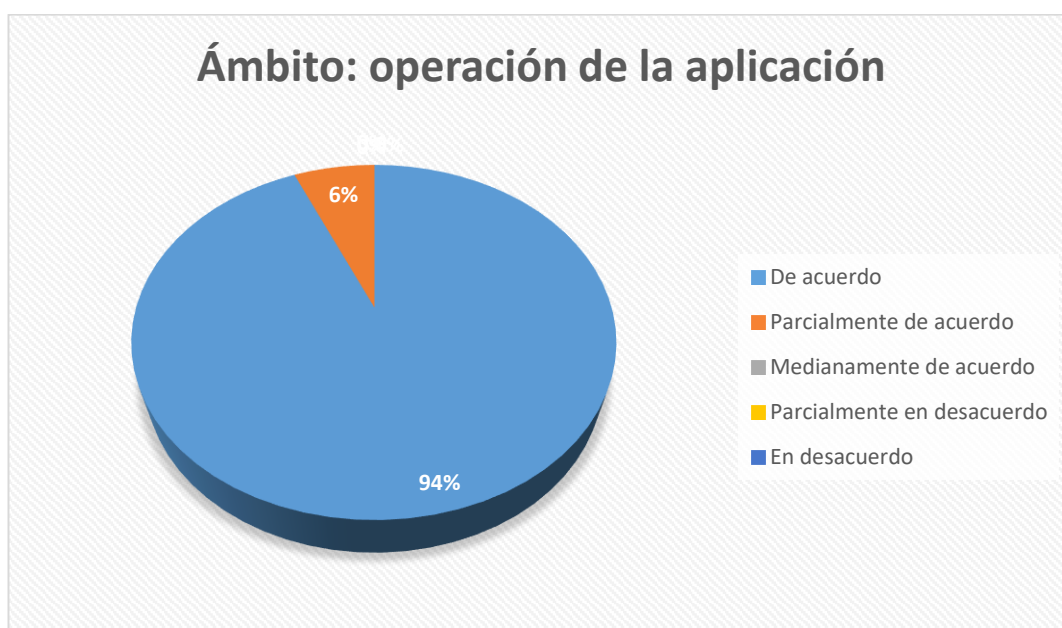


Gráfico IV.2: Resultado ámbito operación de la aplicación

Elaborado por: José Chacón

Interpretación

Los encuestados en su mayoría, concuerdan que el aplicativo cuenta con un acceso seguro al sistema, pueden desplazarse fácilmente entre las páginas, la aplicación funciona correctamente, también la ejecución de la tareas mantiene un solo formato, además las operaciones CRUD no presentan inconvenientes, el estudiante identifica con facilidad el tipo de acción que puede realizar al identificar los íconos, y la velocidad de las operaciones es aceptable.

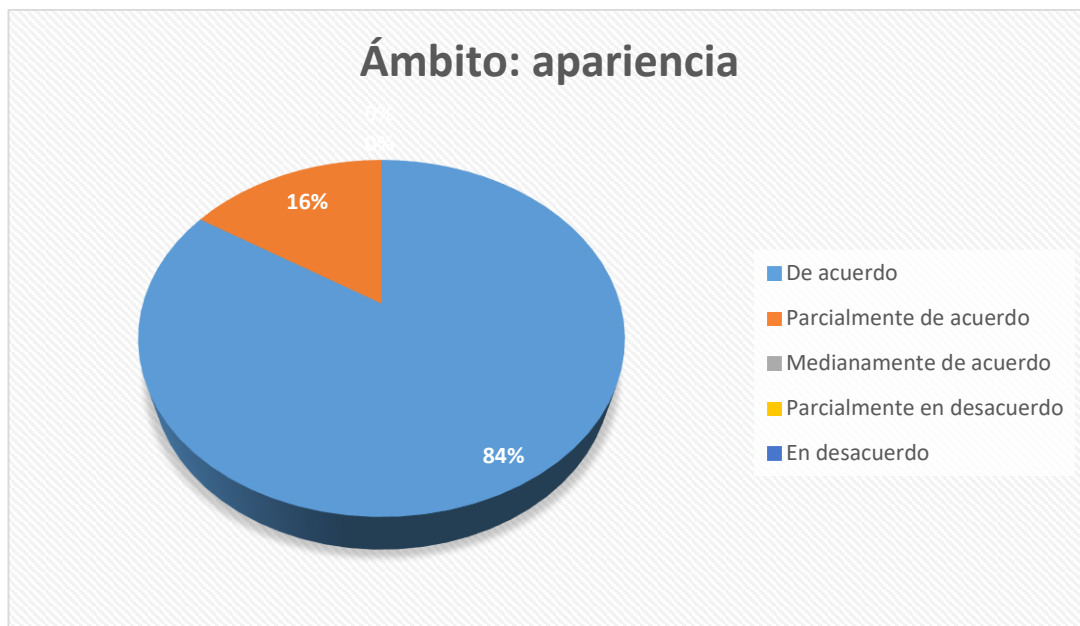


Gráfico IV.3: Resultado ámbito apariencia

Elaborado por: José Chacón

Interpretación

La mayor parte de estudiantes encuestados, encuentran varios tipos de fuentes que les permiten diferenciar el contenido que se presenta, los títulos en cada página facilitan la navegación, los colores son apropiados y mantienen un formato que mejoran el entendimiento.

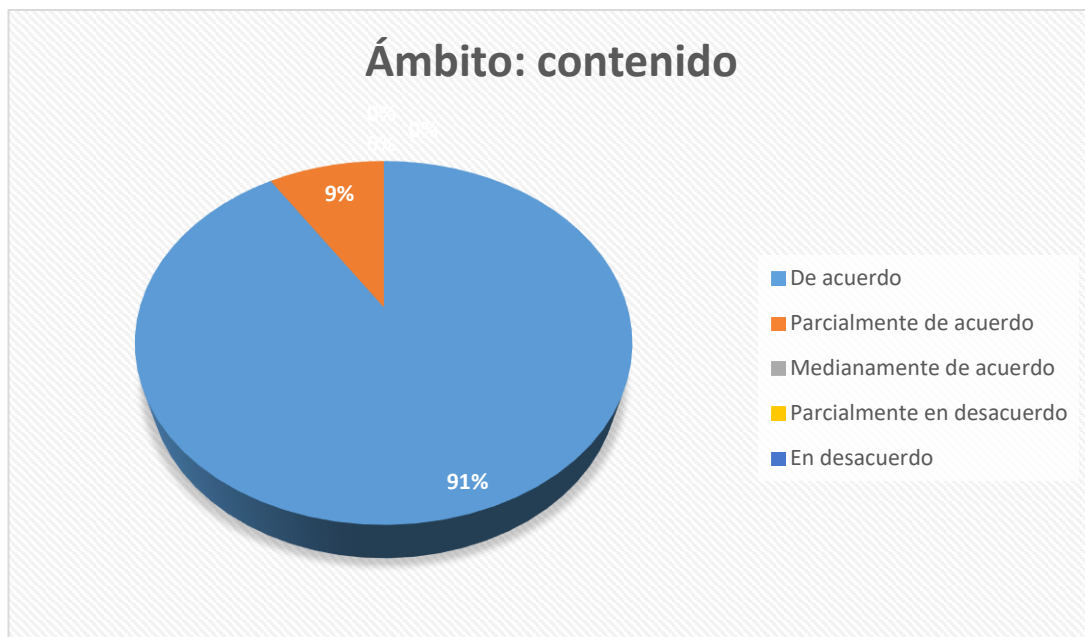


Gráfico IV.4: Resultado ámbito contenido

Elaborado por: José Chacón

Interpretación

Un alto porcentaje de los encuestados que realizan las pruebas, concuerdan que el aplicativo refleja la misma información de la plataforma educativa Moodle, cumple con el objetivo de notificar la tareas pendientes cuando son activadas, además de que el contenido de las notificaciones es claro y preciso, también la aplicación despliega correctamente la lista de materias donde el estudiante está matriculado, así como también puede visualizar la lista de tareas con su respectiva información y ver el contenido de las anotaciones guardadas correctamente.

4.2. Resultado de la prueba al docente

Con el objetivo de conocer la perspectiva de un profesional en el campo informático acerca del funcionamiento del aplicativo, la Mg. Ing. Mónica Mena Directora de la Escuela de Ingeniería en Sistema de la PUCE Ambato tiene la gentileza de someter a prueba el mismo; actividad que se ejecuta el día 17 de junio del 2018.

El instrumento de evaluación está estructurado por casos de prueba; a continuación se muestran los resultados obtenidos:

Caso de prueba	Objetivo	Resultado de la prueba
Inicio de sesión	Verificar el ingreso del estudiante a la aplicación móvil	Muy satisfactorio
Lista de materias	Comprobar el despliegue de la lista de materias en las que está matriculado el estudiante	Muy satisfactorio
Lista de tareas	Comprobar el correcto despliegue de la lista de tareas del estudiante con su respectiva información (actividad, fecha de entrega y descripción).	Muy satisfactorio
Anotaciones	Verificar las operaciones CRUD (crear, visualizar, actualizar y borrar) que se puede hacer en las anotaciones	Muy satisfactorio
Notificaciones	Recibir notificaciones cuando las tareas son activadas en la plataforma educativa Moodle	Muy satisfactorio

Tabla IV.1: Resultado de prueba al docente

Elaborado por: José Chacón

Como se evidencia, todos los casos de prueba, arrojan un resultado de muy satisfactorio; razón suficiente para determinar la correcta funcionalidad del aplicativo. Para mayor detalle sobre la evaluación ejecutada ver el Anexo 3.

Además, la encuestada resalta la importancia del presente proyecto el cual constituye el inicio para futuros desarrollos en beneficio de la educación universitaria.

CAPÍTULO V

CONCLUSIONES Y RECOMENDACIONES

4.3. Conclusiones

- Se cumplió con el desarrollo de la agenda académica móvil para los estudiantes de la institución basada en el sistema operativo Android, la cual cumple con la función de enviar notificaciones de las tareas que son activadas en la plataforma educativa Moodle, además el estudiante puede visualizar la lista de las materias con las respectivas tareas en donde está matriculado, así como también la posibilidad de guardar anotaciones con texto e imagen en la aplicación.
- El uso de Ionic Framework facilitó el desarrollo del aplicativo, pues ofrece un entorno de trabajo organizado al utilizar el patrón de arquitectura MVC, además para la codificación emplea tecnologías web: HTML, JavaScript, CSS que son muy conocidas y aplicadas en la actualidad, el framework proporciona una gran cantidad de documentación sobre sus componentes de interfaz, funciones nativas, y un API, que facilitaron su incorporación en la aplicación.
- Se realizó una exploración de las agendas estudiantiles existentes en el mercado como: My Study Life, Agenda Escolar; este análisis permitió conocer las funcionalidades que poseen, con la finalidad de fundamentar y desarrollar una agenda que se ajuste a las condiciones de la plataforma educativa. Del resultado del análisis se escogieron las siguientes funcionalidades: registro de anotaciones, notificaciones,

seguridad de acceso, listado de asignaturas. Además, se referencia la estructura y distribución de los elementos de las pantallas.

- Para obtener la información de los estudiantes que se muestra dentro de la aplicación móvil se utilizó un web service para realizar la conexión y acceder a la base de datos del Moodle. Cabe mencionar que se utilizó una versión similar al Moodle con el que cuenta la PUCE Ambato (Moodle 3).
- La metodología Mobile-D, demostró ser adecuada para el presente proyecto, puesto que al seguir de manera eficaz los procedimientos establecidas por dicha metodología, y contar con ciclos de desarrollo rápido, se obtuvo un prototipo funcional del aplicativo para ser puesto a prueba en la fase de estabilización antes de dar a conocer a los estudiantes, esto ayudó a corregir errores y realizar mejoras para la consecución de los requerimientos de los interesados
- Las pruebas de la aplicación, contempladas en los ámbitos: estructura, operación, apariencia y contenido, efectuadas a los estudiantes, arrojaron resultados óptimos; demostrando así la factibilidad de uso de este aplicativo en la actividad escolar.
- Mediante el complemento OneSignal, se logró implementar las notificaciones push en la aplicación móvil, con ello, fue posible informar a los estudiantes al momento que son activadas las tareas en la plataforma educativa Moodle.

4.4. Recomendaciones

- A partir de este proyecto incursionar más en la investigación e incorporación de nuevas funcionalidades en la aplicación móvil que beneficien la comunicación interactiva entre profesores y estudiantes en su quehacer académico, así como también ampliar el soporte para los otros Sistemas Operativos como IOS o Windows Phone.
- Se recomienda implementar la aplicación móvil en la PUCE Ambato con la finalidad de que se convierta en una herramienta tecnológica de apoyo que fortalezca la dinámica educativa.
- Para el correcto funcionamiento de la aplicación se recomienda utilizar un dispositivo móvil con una versión de Android superior a 4.0
- Al existir una variedad de framework para desarrollo móvil, se sugiere seleccionar el que más se ajuste con las necesidades del proyecto y que el desarrollador cuente con habilidades básicas para utilizarlo.
- Se recomienda realizar un estudio profundo sobre las posibles vulnerabilidades de seguridad que podría existir en el acceso del aplicativo y la conexión por web service hacia el servidor de Moodle.
- Cada vez que se realice una actualización de Moodle se recomienda un análisis de la estructura de la base de datos sobre todo en las tablas: mood_user, mood_course, mood_ assign, mood_user_enrolments, mood_assignment, mood_assign_user_mapping.

BIBLIOGRAFÍA

- Abernethy, M. (14 de Junio de 2011). *IBM developerWorks*. Obtenido de <https://www.ibm.com/developerworks/ssa/opensource/library/os-nodejs/>
- Abrahamsson, P., Ihme, T., Kolehmainen, K., Kyllönen, P., & Salo, O. (2010). *Mobile-D for Mobile Software: How to Use Agile Approaches for the Efficient Development of Mobile Applications*. Finlania: VTT Technical Research.
- acensTechnologies. (2014). *acens*. Obtenido de <https://www.acens.com/wp-content/images/2014/03/frameworks-white-paper-acens-.pdf>
- Acurio, S. (2018). *Repositorio PUCESA*. Obtenido de <http://repositorio.pucesa.edu.ec/bitstream/123456789/2426/1/76704.pdf>
- Agile Software Development Methodologies At Vtt Electronics. (14 de Junio de 2008). *Agile Software Development Methodologies*. Obtenido de <http://agile.vtt.fi/mobiled.html>
- Albaina, I., & Aranda, J. (2002). *La Educación y el proceso autonómico: textos legales y jurisprudenciales. Volumen XVI*. Madrid: Ministerio de Educación.
- Android Developers. (2017). *Android Developers*. Obtenido de <https://developer.android.com/?hl=es-419>
- Apache Cordova . (2015). *Apache Cordova* . Obtenido de <https://cordova.apache.org/docs/en/latest/guide/overview/index.html>
- Apple Inc. (2016). *Apple IOS*. Obtenido de <https://www.apple.com/es/ios/what-is/>
- Aragón, V. (Octubre de 2010). La observación en el ámbito educativo. Obtenido de https://archivos.csif.es/archivos/andalucia/ensenanza/revistas/csicsif/revista/pdf/Numero_35/VIRGINIA_ARAGON_2.pdf
- Aubry, C. (2007). *HTML5 y CSS3 Revolucione el diseño de sus sistios web*. Barcelona: Ediciones ENI. Obtenido de

<https://books.google.com.ec/books?id=Cijd-Hw-aK4C&printsec=frontcover&hl=es#v=onepage&q&f=false>

Báez, M., & Borrego, Á. (2010). *Univerisad Computense Madrid*. Obtenido de <http://www.it-docs.net/ddata/18.pdf>

Baz, A., Ferreira, I., Álvarez, M., & Garcia, R. (2013). *Universidad de Oviedo*. Obtenido de http://isa.uniovi.es/docencia/SIGC/pdf/telefonía_movil.pdf

Campelo, A., & Viel, P. (2009). Obtenido de https://bibliotecasabiertas2.files.wordpress.com/2013/11/ser_estudiant_e__viel_campelo-3111.pdf

Campusano, C., Fredes, N., & Badilla, C. (28 de Julio de 2014). *Universidad Técnica Federico Santa María*. Obtenido de http://profesores.elo.utfsm.cl/~agv/elo322/1s14/projects/reports/G15/Informe_Latex.pdf

Candela, S. (2007). *Fundamentos de sistemas operativos: teoría y ejercicios resueltos*. Madrid : Paraninfo.

Castillo. (17 de Abril de 2017). Obtenido de <https://media.readthedocs.org/pdf/androidos/latest/androidos.pdf>

Cisco. (03 de Febrero de 2015). Cisco. Obtenido de http://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/white_paper_c11-520862.html

Collel, J. (2016). *Fundació per a la Universitat Oberta de Catalunya*. Obtenido de [https://www.exabyteinformatica.com/uoc/Informatica/Tecnologias_y_herramientas_para_el_desarrollo_web/Tecnologias_y_herramientas_para_el_desarrollo_web_\(Modulo_1\).pdf](https://www.exabyteinformatica.com/uoc/Informatica/Tecnologias_y_herramientas_para_el_desarrollo_web/Tecnologias_y_herramientas_para_el_desarrollo_web_(Modulo_1).pdf)

Cuello, J., & Vittone, J. (2013). *Diseñando apps para móviles*. Duque Catalina.

Date, J. (2001). *Introducción a los sistemas de bases de datos*. Pearson Educación.

Durán, F. (Julio de 2013). *Escuela Politécnica Nacional*. Recuperado el 30 de Abril de 2016, de <http://bibdigital.epn.edu.ec/bitstream/15000/6546/1/CD-5012.pdf>

Duvan, M., & Eduard, J. (2007). *Universidad de San Buenaventura Sede Bogotá*. Obtenido de <http://biblioteca.usbbog.edu.co:8080/Biblioteca/BDigital/40640.pdf>

Eberhardt, C. (Mayo de 2012). *Microsoft Developer Network*. Obtenido de <https://msdn.microsoft.com/es-es/magazine/hh975345.aspx>

Fernández, J. (2015). *Universidad de Murcia*. Obtenido de <http://dis.um.es/~jfernand/docencia/si/tiposso.pdf>

- Flynn, I., & McIver, A. (2010). *Sistemas Operativos*. Cengage Learning Editores.
- Formacion.ec. (2017). *Formacion.ec*. Obtenido de <http://www.formacion.ec/oferta/1430/?nivel=pregrado>
- Galindo, J. (14 de Enero de 2008). *Universitat Oberta de Catalunya*. Obtenido de <http://openaccess.uoc.edu/webapps/o2/bitstream/10609/876/1/00765ffc.pdf>
- Gallego, A. (Diciembre de 2017). *GitBook*. Obtenido de <https://ajgallego.gitbooks.io/ionic/content/arquitectura.html>
- García, O. (Agosto de 2004). *efdeportes.com*. Obtenido de <http://www.efdeportes.com/efd75/pda.htm>
- Gironés, J. (2013). *El Gran Libro de Android*. Barcelona: Marcombo.
- Gomez, F., Uruñuela, P., & Rafael, A. (2008). *Convivencia. Premios 2007*. España: Ministerio de Educación.
- Gómez, M. d. (Febrero de 2013). *Universidad Autónoma Metropolitana*. Obtenido de http://www.cua.uam.mx/pdfs/conoce/libroselec/Notas_del_curso_Bases_de_Datos.pdf
- Google. (2017). *Android*. Obtenido de https://www.android.com/intl/es_es/history/#/marshmallow
- Guitart, R. (2002). *Las actitudes en el centro escolar: Reflexiones y propuestas*. Barcelona: Grao.
- Gutierrez, D. (Julio de 2011). *Univerisidad de los Andes*. Obtenido de http://www.codecompiling.net/files/slides/IS_clase_13_metodos_y_procesos.pdf
- Gutiérrez, J. (2010). *Universidad de Sevilla*. Obtenido de http://www.lsi.us.es/~javierj/investigacion_ficheros/Framework.pdf
- Guzmán, N. (2013). *¿Qué significa ser estudiante universitario? Representaciones de los alumnos de Ciencia de la Educación de la UNT*. Buenos Aires. Obtenido de <https://www.aacademica.org/000-054/451.pdf>
- Hammer, M. (2002). *La agenda: lo que toda empresa debe hacer para dominar la década*. España: Grupo Planeta (GBS).
- Herrera, T. (01 de Noviembre de 2003). *eduteka*. Obtenido de <http://eduteka.icesi.edu.co/articulos/Handhelds1>
- Hewlett-Packard Company. (2016). *HP*. Obtenido de <http://www.hp.com/hpinfo/newsroom/press/pdabrochure.pdf>

- Ionic. (2016). *Ionic Framework*. Obtenido de <https://ionicframework.com/docs/v1/guide/preface.html>
- Ionic Framework. (2017). *Ionic Framework*. Obtenido de <https://ionicframework.com/docs/intro/concepts/>
- Lancker, L. (2012). *HTML5 Los fundamentos del lenguaje*. Barcelona: Ediciones ENI.
- Madiedo, L. (2014). *Gestión del tiempo, recursos e instalaciones*. Málaga: IC Editorial.
- Malave, K., & Beauperthuy, J. (2011). "Android" el sistema operativo de Google para dispositivos móviles. *Negotium*, 96.
- Marqués, M. (Enero de 2009). *UNIVERSITAT JAUME I DE CASTELLÓ*. Obtenido de http://www3.uji.es/~mmarques/apuntes_bbdd/apuntes.pdf
- Microsoft. (2016). *Microsoft*. Obtenido de <https://www.microsoft.com/es-mx/windows/phones>
- Mobile Marketing Association Spain. (05 de Noviembre de 2011). *MMA SPAIN*. Obtenido de <https://mmaspain.com/wp-content/uploads/2015/09/Libro-Blanco-Apps.pdf>
- Moodle. (2017). *Moodle*. Obtenido de https://docs.moodle.org/all/es/Acerca_de_Moodle
- Mozilla. (03 de Febrero de 2018). *MDN web doc*. Obtenido de https://developer.mozilla.org/es/docs/Learn/Getting_started_with_the_web/JavaScript_basics
- nodeJS. (2017). *nodeJS*. Obtenido de <https://nodejs.org/en/download/>
- Octavian, L., Nguyen, H., Tam, B., & Wang, B. (2009). *UC San Diego*. Obtenido de <https://cseweb.ucsd.edu/classes/sp00/cse221/reports/lucngu-tam-wan.pdf>
- Paré, R., Santillán, L., & Costa, D. (2005). *Universidad Oberta de Catalunya*. Obtenido de <http://www.uoc.edu/masters/oficiales/img/913.pdf>
- Peña, M. (2013). *Centro Virtual Cervantes*. Obtenido de https://cvc.cervantes.es/ensenanza/biblioteca_ele/asele/pdf/24/24_913.pdf
- Pontificia Universidad Católica del Ecuador. (2017). *PUCESA*. Recuperado el 15 de 01 de 2016, de <https://pucesa.edu.ec/mision-vision/>
- Pontificia Universidad Católica del Ecuador Ambato. (2018). *Academics*. Obtenido de <http://app.pucesa.edu.ec:9000/academics/>
- Pontificia Universidad Católica del Ecuador Ambato. (2018). *Moodle PUCESA*. Obtenido de <http://moodle.pucesa.edu.ec/login/index.php>

- Prescott, P. (2017). *La programación JavaScript*. Babelcube Inc.
- Pressman, R. (2005). *Ingeniería del software. Un enfoque práctico*. México: McGraw-Hill Interamericana.
- Ramírez, R. (2009). *Universidad Oberta de Catalunya*. Recuperado el 30 de Abril de 2016, de [https://www.exabyteinformatica.com/uoc/Informatica/Tecnologia_y_de_sarrollo_en_dispositivos_moviles/Tecnologia_y_desarrollo_en_dispositivos_moviles_\(Modulo_4\).pdf](https://www.exabyteinformatica.com/uoc/Informatica/Tecnologia_y_de_sarrollo_en_dispositivos_moviles/Tecnologia_y_desarrollo_en_dispositivos_moviles_(Modulo_4).pdf)
- Real Academia Española. (2017). *Diccionario de la lengua española*. Obtenido de <http://dle.rae.es/?id=14mRqHq>
- Rivero, M. (2016). *AngularJs Paso a Paso*. Leanpub.
- Rodríguez, F. (2017). *Universidad de Sevilla*. Obtenido de https://idus.us.es/xmlui/bitstream/handle/11441/66716/TFG_Gabriel%20Rodr%C3%ADguez%20Flores.pdf?sequence=1
- Roschelle, J. (2003). Unlocking the learning value of wireless mobile devices. *Journal of Computer Assisted Learning*.
- Sánchez, M. (12 de Mayo de 2006). *Universidad de la Américas Puebla*. Obtenido de http://catarina.udlap.mx/u_dl_a/tales/documentos/lis/sanchez_r_ma/capitulo2.pdf
- Solis, C. (2015). *Manual del Guerrero: ANGULARJS*. Obtenido de http://manualdelguerrero.com/angularjs/adelanto/manual_angularjs_preview.pdf
- Tanenbaum, A. (2009). *Sistemas Operativos Modernos*. Amsterdam. Obtenido de <https://norbertomn.files.wordpress.com/2013/09/sistemas-operativos-modernos.pdf>
- TicBeat. (Mayo de 2011). *innovanube*. Recuperado el 19 de Octubre de 2015, de <http://www.ticbeat.com/innovacion/desarrollo-aplicaciones-moviles/>

ANEXOS

Anexo 1

Pontificia Universidad Católica del Ecuador Sede Ambato

Escuela de Ingeniería en Sistemas

Encuesta

La siguiente encuesta está dirigida a los estudiantes de la PUCESA, como parte del desarrollo del proyecto de investigación: **“Desarrollo de la Agenda Académica móvil para los estudiantes de la Pontificia Universidad Católica del Ecuador Sede Ambato basada en el Sistema Operativo Android”**.

Toda la información recogida de esta encuesta será utilizada confidencialmente.

Objetivo: Recoger información sobre la funcionalidad que tendría una aplicación móvil de notificaciones en los estudiantes de la Pontificia Universidad Católica del Ecuador Sede Ambato.

Instrucción: Marque con una X la respuesta que usted considere correcta.

1. ¿Con qué frecuencia usa el teléfono móvil?

<input type="checkbox"/> Muy frecuentemente	<input type="checkbox"/> Frecuentemente	<input type="checkbox"/> Ocasionalmente	<input type="checkbox"/> Raramente	<input type="checkbox"/> Nunca
---	---	---	------------------------------------	--------------------------------

2. ¿Qué sistema operativo utiliza su dispositivo móvil?

<input type="checkbox"/>	iOS
<input type="checkbox"/>	Android
<input type="checkbox"/>	Windows Phone
<input type="checkbox"/>	Symbian
<input type="checkbox"/>	Otro ¿Cuál?:

3. ¿Con qué frecuencia accede a la Plataforma Educativa (EVAE-MOODLE) de la PUCESA?

<input type="checkbox"/> Muy frecuentemente	<input type="checkbox"/> Frecuentemente	<input type="checkbox"/> Ocasionalmente	<input type="checkbox"/> Raramente	<input type="checkbox"/> Nunca
---	---	---	------------------------------------	--------------------------------

4. ¿Qué dispositivo usa para ingresar a la Plataforma Educativa (EVAE-MOODLE) de la PUCESA?

<input type="checkbox"/>	Computador
<input type="checkbox"/>	Celular
<input type="checkbox"/>	Tablet
<input type="checkbox"/>	Otro ¿Cuál?:

5. ¿Utiliza alguna aplicación móvil para recordar, organizar o guardar información educativa?

<input type="checkbox"/> Muy frecuentemente	<input type="checkbox"/> Frecuentemente	<input type="checkbox"/> Ocasionalmente	<input type="checkbox"/> Raramente	<input type="checkbox"/> Nunca
---	---	---	------------------------------------	--------------------------------

6. ¿A través de que medio le gustaría que le recuerden las tareas pendientes de la Plataforma Educativa (EVAE-MOODLE) de la PUCESA?

	Correo electrónico
	Aplicación móvil
	SMS
	Otro ¿Cuál?:

7. ¿Cree que al usar una aplicación móvil de notificaciones de tareas se mejorará en el cumplimiento de las mismas?

<input type="checkbox"/> Totalmente de acuerdo	<input type="checkbox"/> De acuerdo	<input type="checkbox"/> Medianamente de acuerdo	<input type="checkbox"/> En desacuerdo	<input type="checkbox"/> Totalmente desacuerdo
--	-------------------------------------	--	--	--

8. ¿Le gustaría que la aplicación móvil de notificaciones de tareas muestre un resumen de las tareas pendientes?

<input type="checkbox"/> Totalmente de acuerdo	<input type="checkbox"/> De acuerdo	<input type="checkbox"/> Medianamente de acuerdo	<input type="checkbox"/> En desacuerdo	<input type="checkbox"/> Totalmente desacuerdo
--	-------------------------------------	--	--	--

9. ¿Qué funciones le gustaría que tenga la aplicación móvil de notificaciones de tareas?

	Notificaciones automáticas
	Eventos pendientes
	Calendario
	Almacenamiento de notas
	Horario de clases
	Calificaciones
	Otro ¿Cuál?:

10. ¿Qué factores te gustaría que tenga la aplicación móvil de notificaciones de tareas?

	Seguridad
	Usabilidad
	Portabilidad
	Fiabilidad
	Otro ¿Cuál?:

Anexo 2

Pontificia Universidad Católica del Ecuador Sede Ambato

Escuela de Ingeniería en Sistemas

El siguiente cuestionario va dirigido a los estudiantes de la PUCESA, para la evaluación preliminar de la aplicación móvil, perteneciente al trabajo de graduación titulado:

“Desarrollo de la Agenda Académica móvil para los estudiantes de la Pontificia Universidad Católica del Ecuador Sede Ambato basada en el Sistema Operativo Android”.

Fecha:

Por favor, marque el casillero que represente su opinión (grado de acuerdo) sobre las características de la aplicación móvil.

ÁMBITO: ESTRUCTURA DE LA APLICACIÓN

La aplicación móvil “Mi Agenda”, mantiene todas las páginas estructuradas	De acuerdo <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> en desacuerdo
La aplicación móvil “Mi Agenda”, mantiene elementos internos (ej.: áreas de texto, imágenes, menú, botones) que siguen un formato a lo largo del aplicativo	De acuerdo <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> en desacuerdo
La distribución de los elementos estructurales (ej. scroll, áreas de contenido, botones, menú) es adecuada.	De acuerdo <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> en desacuerdo
La cantidad de los elementos estructurales utilizados es apropiada	De acuerdo <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> en desacuerdo
La distribución de los elementos estructurales mantiene consistencia a lo largo de la aplicación móvil “Mi Agenda”	De acuerdo <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> en desacuerdo

ÁMBITO: OPERACIÓN DE LA APLICACIÓN

La aplicación móvil “Mi agenda” cuenta con un acceso seguro al sistema	De acuerdo <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> en desacuerdo
Es fácil desplazarse entre las páginas en la aplicación móvil “Mi Agenda”.	De acuerdo <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> en desacuerdo
La aplicación móvil “Mi agenda” funciona correctamente (no se perciben errores)	De acuerdo <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> en desacuerdo
La ejecución de tareas en la aplicación móvil “Mi Agenda” (ej.: navegar por las páginas, hacer clic en botones, seleccionar opciones, entre otros) mantiene un formato a lo largo de la aplicación	De acuerdo <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> en desacuerdo
La aplicación móvil “Mi Agenda” ejecuta operaciones CRUD (crear, visualizar, actualizar y borrar) en las notas, sin inconvenientes	De acuerdo <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> en desacuerdo
Se identifican fácilmente los iconos, listas, resultados, zonas activas y el tipo de acción que se desea ejecutar	De acuerdo <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> en desacuerdo
La velocidad de operaciones dentro de la aplicación “Mi Agenda” es aceptable	De acuerdo <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> en desacuerdo




ÁMBITO: APARIENCIA

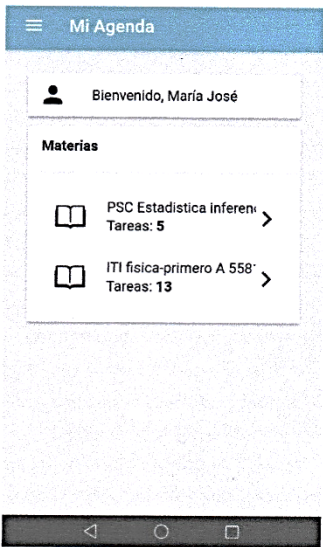

Los tipos de fuentes que se utilizan en la aplicación móvil “Mi Agenda”, permite diferenciar el contenido que se presenta (ej.: Títulos de actividades, fechas de entrega, descripción de tareas, entre otros)	De acuerdo <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> en desacuerdo
Los títulos en cada página de la aplicación móvil “Mi Agenda” facilitan la navegación	De acuerdo <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> en desacuerdo
Los colores utilizados en la aplicación móvil “Mi Agenda” son apropiados	De acuerdo <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> en desacuerdo
La aplicación móvil “Mi Agenda” mantiene un mismo formato respecto a los colores que mejoran el entendimiento del usuario	De acuerdo <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> en desacuerdo

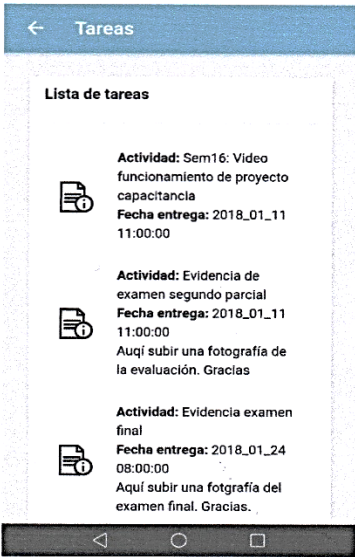

ÁMBITO: CONTENIDO

La fiabilidad mostrada en la aplicación móvil “Mi Agenda”, refleja la información de la plataforma educativa Moodle	De acuerdo <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> en desacuerdo
La aplicación móvil “Mi Agenda” cumple con el objetivo de notificar tareas pendientes cuando son activadas en la plataforma educativa	De acuerdo <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> en desacuerdo
El contenido de las notificaciones que llegan al dispositivo móvil es claro y preciso.	De acuerdo <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> en desacuerdo
La aplicación móvil “Mi Agenda” despliega correctamente la lista de materias donde el alumno está matriculado	De acuerdo <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> en desacuerdo
Al seleccionar una materia en la aplicación “Mi Agenda” se puede visualizar correctamente la lista de tareas con su respectiva actividad, fecha de entrega y descripción	De acuerdo <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> en desacuerdo
En la aplicación “Mi Agenda” se visualiza correctamente el contenido de las anotaciones guardadas	De acuerdo <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> en desacuerdo

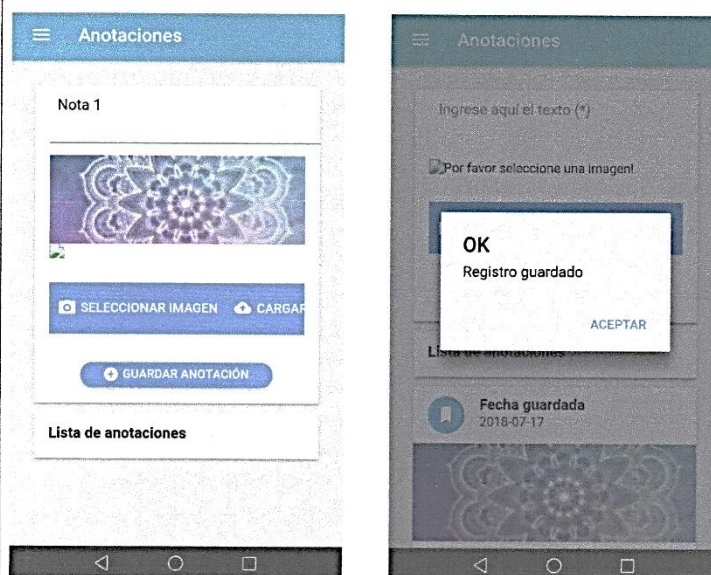
Anexo 3

Caso de prueba: Inicio de sesión		Identificador caso de prueba: CP01	
		Fecha: 17/7/2018	
Descripción: Verificar el ingreso del estudiante a la aplicación móvil			
Precondiciones: Se debe ingresar correctamente el usuario y contraseña con las cuales el estudiante está registrado en la plataforma educativa Moodle			
Pasos y condiciones de ejecución:			
Condición	Entrada	Datos	Resultado esperado
Ingreso de un usuario y contraseña registrado en el Moodle	Usuario Contraseña	adminusuario 098131596_Jos	Ingreso a la aplicación móvil
Ingreso de un usuario y contraseña no registrado en el Moodle o incorrecto	Usuario Contraseña	Cualquiera Cualquiera	Muestra una alerta indicando que el usuario y/o contraseña es incorrecta
Usuario y/o contraseña en blanco	Usuario Contraseña	En blanco En blanco	
Resultados del caso de prueba:			
			
Correcto ingreso del usuario y contraseña		Ingreso incorrecto o campos en blanco en usuario y/o contraseña	
Muy satisfactorio <input checked="" type="checkbox"/>		Insatisfactorio <input type="checkbox"/>	
Observación:			
Responsable ejecución: Mg. Ing. Mónica Mena			
Firma:			
			

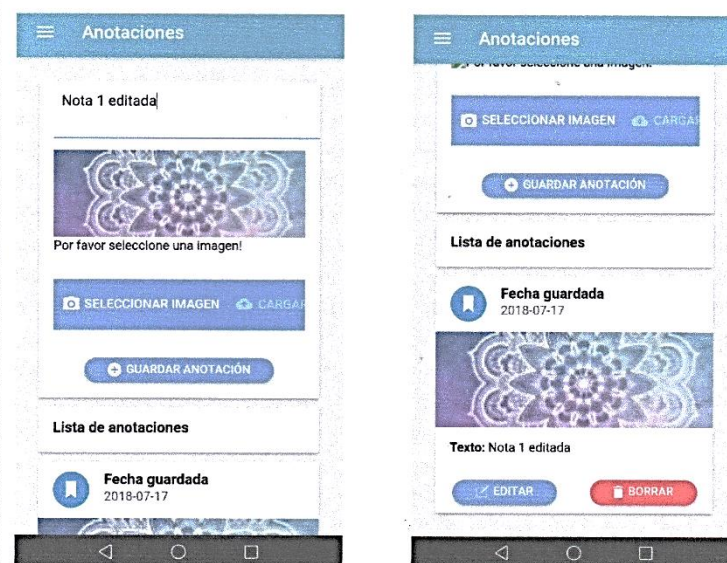
Caso de prueba: Lista de materias		Identificador caso de prueba: CP02	
		Fecha: 17/7/2018	
Descripción: Comprobar el despliegue de la lista de materias en las que está matriculado el estudiante			
Precondiciones: El estudiante debe iniciar correctamente la sesión y estar matriculado en alguna materia			
Paso y condiciones ejecución:			
Condición	Entrada	Datos	Resultado esperado
Correcto inicio de sesión	Usuario Contraseña	adminusuario 098131596_Jos	Visualización de las materias que toma el estudiante
Resultados del caso de prueba:			
			
<p>Una vez ingresado correctamente el usuario y contraseña se muestra la lista de materias que toma el estudiante</p> <p>Muy satisfactorio <input checked="" type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> Insatisfactorio</p>			
Observación:			
Responsable ejecución: Mg. Ing. Mónica Mena			
Firma:			
			

Caso de prueba: Lista de tareas		Identificador caso de prueba: CP03	
		Fecha: 17/7/2018	
Descripción: Comprobar el correcto despliegue de la lista de tareas del estudiante con su respectiva información (actividad, fecha de entrega y descripción).			
Precondiciones: El estudiante debe iniciar correctamente la sesión y estar matriculado en alguna materia			
Pasos y condiciones de ejecución:			
Condición	Entrada	Datos	Resultado esperado
Estar matriculado en una materia y activada la tarea	Seleccionar la materia	No aplica	Visualización de las tareas activadas con su respectiva información
	Seleccionar en el menú la opción últimas tareas	No aplica	Visualización de todas las tareas activadas de las materias con su respectiva información
Activar una tarea desde la plataforma educativa Moodle	Creación de una tarea por el docente	Información de la tarea	En la aplicación verificar la visualización de la tarea deslizando con el dedo hacia abajo para actualizar la página
Resultados del caso de prueba:			
 <p>Al pulsar sobre una materia se despliega la lista de tareas correspondientes a la misma</p>			
Muy satisfactorio <input checked="" type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> Insatisfactorio			
Observación:			
Responsable ejecución: Mg. Ing. Mónica Mena			
Firma: 			

Caso de prueba: Anotaciones		Identificador caso de prueba: CP04	
		Fecha: 17/7/2018	
Descripción: Verificar las operaciones CRUD (crear, visualizar, actualizar y borrar) que se puede hacer en las anotaciones			
Precondiciones: El estudiante debe iniciar correctamente la sesión			
Pasos y condiciones de ejecución:			
Condición	Entrada	Datos	Resultado esperado
Correcto inicio de sesión	Ingresar texto	Cualquiera	Despliegue de una alerta indicando que la anotación fue guardada y visualización en la parte inferior de la anotación registrada
	Seleccionar y cargar imagen	Cualquiera	
	Ingresar solo texto	Cualquiera	Despliegue de una alerta indicando que se debe ingresar texto
Elegir una anotación registrada	Pulsar el botón editar, cambiar el texto o imagen	Cualquiera	Despliegue de una alerta indicando que la anotación fue guardada y visualización en la parte inferior de la anotación modificada
Tener anotaciones registrada	Pulsar el botón borrar y confirmar la eliminación	No aplica	Despliegue de una alerta indicando que la anotación fue borrada y verificación que ya no exista la misma
Tener anotaciones registradas	Seleccionar en el menú la opción anotaciones	No aplica	Visualización de las notas registradas
Dejar los campos en blanco	Pulsar el botón guardar	No aplica	Despliegue de una alerta indicando que se debe ingresar texto

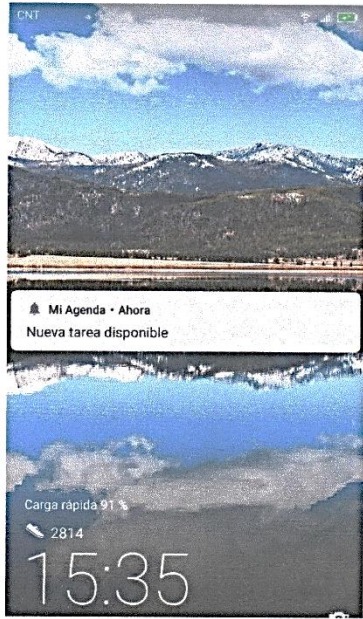
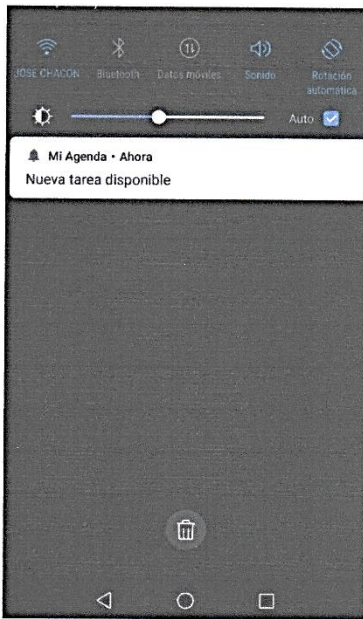
Resultados del caso de prueba:

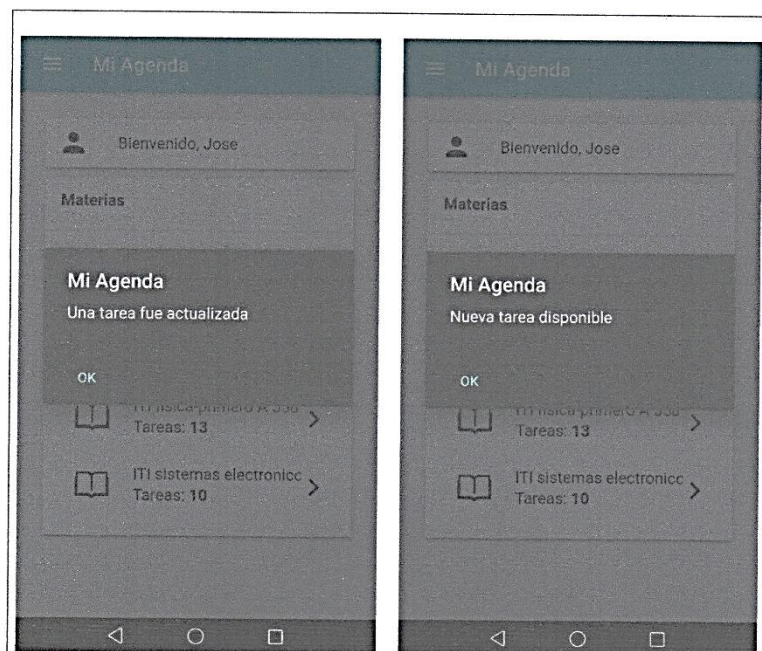
Registro de una nueva anotación y mensaje de guardado correctamente



Edición de una anotación existente

<p>Mensaje de confirmación antes de borrar la anotación</p>	<p>Campos en blanco, despliega un mensaje de error</p>
<p>Muy satisfactorio <input checked="" type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> Insatisfactorio</p>	
<p>Observación:</p>	
<p>Responsable ejecución: Mg. Ing. Mónica Mena</p>	
<p>Firma: </p>	

Caso de prueba: Notificaciones		Identificador caso de prueba: CP05	
		Fecha: 17/7/2018	
Descripción: Recibir notificaciones cuando las tareas son activadas en la plataforma educativa Moodle			
Precondiciones: Estar registrado en la plataforma educativa Moodle y matriculado en una materia Debe iniciar sesión desde el dispositivo al menos una vez			
Pasos y condiciones de ejecución:			
Condición	Entrada	Datos	Resultado esperado
Iniciar sesión correctamente	El docente debe activar una tarea desde la plataforma educativa Moodle	Información de la tarea	Despliegue de una alerta indicando que existe una nueva tarea disponible
Salir de aplicación y bloquear el dispositivo		Información de la tarea	Notificación indicando que existe una nueva tarea disponible
Iniciar sesión correctamente	El docente puede actualizar una tarea desde la plataforma educativa Moodle	Información de la tarea	Despliegue de una alerta indicando que una tarea fue actualizada
Salir de aplicación y bloquear el dispositivo		Información de la tarea	Notificación indicando que una tarea fue actualizada
Resultados del caso de prueba:			
			
<p>Notificación cuando el móvil esta bloqueado y se activa o actualiza una tarea</p>			



Notificación cuando el estudiante esta dentro del aplicativo y se activa o actualiza una tarea

Muy satisfactorio Insatisfactorio

Observación:

Responsable ejecución: Mg. Ing. Mónica Mena

Firma: