

UNIVERSIDAD NACIONAL TECNOLÓGICA DE LIMA SUR
FACULTAD DE INGENIERÍA DE SISTEMAS Y ADMINISTRACIÓN
DE EMPRESAS
CARRERA PROFESIONAL DE INGENIERÍA DE SISTEMAS



**“PROTOTIPO MÓVIL DE NOTIFICACIONES ACADÉMICAS PARA LA
UNIVERSIDAD NACIONAL TECNOLÓGICA DE LIMA SUR”**

TRABAJO DE SUFICIENCIA PROFESIONAL
Para optar el Título Profesional de

INGENIERO DE SISTEMAS

PRESENTADO POR EL BACHILLER

ESPINOZA MEJIA, ERNESTO VIDAL

Villa El Salvador
2015

DEDICATORIA

Dedico este trabajo a Dios, a mi padre A. Vidal, mi madre Aguida, mis hermanas Geraldine y Zulema y mi hermano Adolfo quienes son las personas a las que más quiero.

AGRADECIMIENTO

Agradezco a la UNTELS el haberme dado
la oportunidad estudiar Ingeniería de Sistemas.

Agradezco a los asesores y profesores que
compartieron sus conocimientos y experiencias para que
éste trabajo sea una realidad.

Y gracias especiales a todos los amigos
y amigas que compartieron su tiempo
conmigo.

ÍNDICE

LISTADO DE FIGURAS.....	vi
LISTADO DE TABLAS.....	viii
RESUMEN	ix
ABSTRACT	x
INTRODUCCIÓN	xi
I. CAPITULO 1: PLANTEAMIENTO DEL PROBLEMA	13
1.1. Descripción de la realidad problemática	13
1.2. Justificación del proyecto	14
1.3. Delimitación del proyecto	14
1.4. Formulación del problema	15
1.5. Objetivos	15
1.5.1. Objetivo general.....	15
1.5.2. Objetivos específicos	15
II. CAPITULO 2: MARCO TEÓRICO	16
2.1. Antecedentes de la Investigación.....	16
2.1.1. Antecedentes internacionales.....	16
2.1.2. Antecedentes nacionales.....	18
2.2. Bases Teóricas	19
2.2.1. SCRUM	19
2.2.2. Android.....	23
2.2.3. Google Cloud Messaging.....	27
2.3. Marco conceptual	30
III. CAPITULO 3: DESCRIPCIÓN DEL PROTOTIPO	34
3.1. Análisis del prototipo.....	34
3.1.1. Requerimientos del sistema.....	34
3.1.2. Sprint backlog	36
3.2. Construcción del prototipo.....	37
3.2.1. Realización del Sprint 1	37
3.2.2. Realización del Sprint 2	43
3.2.3. Realización del Sprint 3	46
3.2.4. Realización del Sprint 4	48
3.3. Revisión y consolidación de resultados	49
3.3.1. Análisis de encuesta	49
3.3.2. Revisión del prototipo	56
CONCLUSIONES	70

RECOMENDACIONES	71
BIBLIOGRAFÍA	72
ANEXOS	75

LISTADO DE FIGURAS

Figura 1 Proceso SCRUM	20
Figura 2 Arquitectura Android.....	24
Figura 3 Arquitectura de GCM.....	28
Figura 4 Mockup login.....	37
Figura 5 Mockup menú principal.....	38
Figura 6 Mockup información personal.....	38
Figura 7 Mockup notas del ciclo actual.....	39
Figura 8 Mockup historial académico	39
Figura 9 Mockup descarga de historial académico	40
Figura 10 Mockup historial de notificaciones.....	40
Figura 11 Mockup recepción de notificaciones y actualización de icono de aplicación	41
Figura 12 Modelo de base de datos del sistema.....	42
Figura 13 Interfaz web docente	43
Figura 14 Google developers console.....	44
Figura 15 Base de datos SQLite.....	45
Figura 16 Flujo Prototipo, GCM y Servidor web.....	47
Figura 17 Valoración del SIGU según utilidad.....	52
Figura 18 Problema encontrado vs Estudiantes.....	53
Figura 19 Porcentaje de estudiantes que usan Smartphone	53
Figura 20 Smartphones según sistema operativo	54
Figura 21 Estudiantes que utilizarían la aplicación	55
Figura 22 Funcionalidades requeridas por estudiantes.....	56
Figura 23 Resultados de la primera entrada del caso de prueba 1	58

Figura 24 Resultados de la segunda y tercera entrada del caso de prueba 1.....	59
Figura 25 Información personal del usuario	60
Figura 26 Notas del ciclo actual.....	61
Figura 27 Historial académico	62
Figura 28 Lista de mensajes.....	63
Figura 29 Interfaz web de registro de notas	64
Figura 30 Interfaz web de redacción de mensajes.....	64
Figura 31 Cola de notificaciones GCM en el dispositivo	65
Figura 32 Descarga del historial académico del prototipo	66
Figura 33 PDF generado del historial	67
Figura 34 Detalle de mensaje de un curso	68
Figura 35 Forzado de sincronización.....	69

LISTADO DE TABLAS

Tabla 1 Product Backlog del sistema	35
Tabla 2 Especificación de los Sprint Backlog	36
Tabla 3 Entradas caso de prueba 1	57
Tabla 4 Acciones Caso de prueba 2	59
Tabla 5 Acciones Caso de prueba 3	63
Tabla 6 Acciones Caso de prueba 4	66

RESUMEN

Aprovechando el avance de las tecnologías móviles y su posibilidad de aplicación en el ámbito académico, en el presente trabajo se busca desarrollar una aplicación para visualizar la información académica de los estudiantes de pregrado de la Universidad Nacional Tecnológica de Lima Sur. Para este propósito se eligió Scrum como metodología de trabajo, Android como sistema operativo y Google Cloud Messaging para el envío y recepción de notificaciones, las cuales avisarán oportunamente la actualización de la información. Como resultado final se obtuvo un prototipo de aplicación que muestra información referente a: Historial académico, información personal y calificaciones del periodo en curso.

Tras finalizar el trabajo, se pudo demostrar que las tecnologías móviles poseen gran potencial de aplicación, más allá de su uso para fines comerciales (sean redes sociales, juegos, publicidad, etc.) se pueden crear aplicaciones con fines netamente académicos.

PALABRAS CLAVE: Aplicación móvil, Android, Google Cloud Messaging, Información académica, SCRUM.

ABSTRACT

Taking advantage of the advancement of mobile technologies and their applicability for academic purposes, the present thesis aims to develop an application to display the academic information of undergraduates of the “Universidad Nacional Tecnológica de Lima Sur”. For this purpose was chosen Scrum methodology, Android as operating system and Google Cloud Messaging for sending and receiving notifications, that notify opportunely update the information. As a final result a prototype application that displays information about academic history, personal information and qualifications of the current period was obtained.

After completing the thesis, it was proved that mobile technologies have great potential for application beyond its use for commercial purposes (for example: social networks, games, advertising, etc.) can create applications with purely academic purposes.

KEY WORDS: Mobile application, Android, Academic information, Google Cloud Messaging, SCRUM.

INTRODUCCIÓN

Desde el inicio de su existencia la Universidad Nacional Tecnológica de Lima Sur (UNTELS) ha utilizado sistemas de información para agilizar sus procesos y dar servicio a la comunidad estudiantil, entre estos se encuentra el Sistema Informativo de Gestión Universitaria (SIGU), el cual es utilizado por profesores para administrar las calificaciones de los estudiantes inscritos en sus cursos y por los estudiantes para informarse sobre todo de su historial académico y las notas del ciclo actual. El sistema en cuestión opera en entorno web y si bien es cierto que cumple su función, ha demostrado tener deficiencias respecto a la disponibilidad y seguridad.

Es por ello que mediante este trabajo se plantea el desarrollo de un prototipo de aplicación móvil para smartphone, aprovechando el gran uso que los estudiantes hacen de estos dispositivos. El prototipo se encargará de administrar los mensajes que recibe del servidor, con lo cual los estudiantes ya no tendrán que acceder constantemente al sistema web si no que será la aplicación quien les informe sobre sus calificaciones u otros en el momento oportuno.

El trabajo se ha desarrollado en 3 capítulos, en el capítulo 1 se plantea la realidad problemática y se enuncian los objetivos del trabajo, en el capítulo 2 se desarrolla el marco teórico y conceptual, y en el capítulo 3 se describe el desarrollo del prototipo.

CAPITULO 1: PLANTEAMIENTO DEL PROBLEMA

1.1. Descripción de la realidad problemática

En la UNTELS, mediante el SIGU se permite a los estudiantes informarse sobre sus calificaciones del ciclo actual, revisar su historial académico y su información personal, resultando un sistema de mucha utilidad. En los últimos años, este sistema quedaba inactivo en fechas regularmente posteriores a exámenes parciales, lo cual se debe al aumento del estudiantado de la universidad y con ello incrementó la cantidad de usuarios que desean acceder casi simultáneamente, siendo así el sistema puede quedar fuera de servicio durante horas o días enteros.

Por otra parte, la programación de prácticas no se encuentra establecida en el syllabus, por lo que el docente durante una clase informa a sus estudiantes la fecha de realización de la práctica, de presentarse alguna eventualidad que impida que la prueba se lleve a cabo no existe un medio formal que informe a los estudiantes de la postergación o cancelación de la prueba.

Frente a esta situación se plantea la siguiente pregunta ¿Es posible solucionar este problema mediante la realización del prototipo de notificaciones?

1.2. Justificación del proyecto

Con el desarrollo de esta investigación la UNTELS contará con un prototipo de aplicación que complemente las funciones del SIGU, a su vez los estudiantes dispondrán de un modo novedoso y más interactivo de información sobre su progreso académico.

Al hacer uso de las tecnologías móviles no solo se desea mostrar las bondades que estas poseen, si no que la universidad se convenza de las soluciones prácticas que estas permiten.

El prototipo también servirá para que futuros estudiantes se animen a desarrollar una aplicación que logre abarcar más procesos de la gestión académica utilizando tecnología móvil.

1.3. Delimitación del proyecto

El prototipo a desarrollar trabajará con los datos que un estudiante de pregrado de la carrera de Ingeniería de Sistemas puede generar a lo largo de los periodos académicos a los que haya asistido.

Tanto para la aplicación cliente y la aplicación web, se implementará Google Cloud Messaging, el que permitirá el envío y almacenamiento de las notificaciones.

El prototipo estará destinado a smartphone Android, quedando tablets y equipos de otros sistemas operativos excluidos. Así mismo mostrará reportes, como son las notas del ciclo actual e historial

académico y permitirá la descarga de este último, como tal el prototipo permite visualizar información mas no modificarla.

Al ser un prototipo con fines demostrativos, la aplicación trabajará con un servidor que contendrá una base de datos que simule a la utilizada por el SIGU, por ende es limitada y con datos ficticios.

1.4. Formulación del problema

¿El prototipo móvil de notificaciones permitirá que los estudiantes de la UNTELS se enteren oportunamente sobre su información académica?

1.5. Objetivos

1.5.1. Objetivo general

Desarrollar el prototipo móvil para brindar información académica usando tecnología Android y metodología SCRUM.

1.5.2. Objetivos específicos

- Determinar los componentes de la tecnología Android de los cuales el prototipo se servirá.
- Realizar la implementación de GCM tanto en el aplicativo web y móvil.
- Realizar una evaluación del SIGU y su complementación con el prototipo.

CAPITULO 2: MARCO TEÓRICO

2.1. Antecedentes de la Investigación

2.1.1. Antecedentes internacionales

La tesina “Propuesta de una aplicación (Web) móvil en la Facultad de Contaduría, Administración y Sistemas” (Torres Huesca, 2011) comenta la problemática relacionada con la página web de la Universidad Veracruzana, la cual contenía información muy útil para el público de la universidad pero que no estaba optimizada para ser accedida desde dispositivos móviles. Siendo así, la autora del trabajo propone el desarrollo de una aplicación para el sistema operativo Windows Phone 7 que utilizará el navegador móvil, ésta mostrará los eventos próximos a realizarse en su universidad y otras funciones accesibles a través de usuario y contraseña como son la revisión de horarios, notas del periodo actual, etc.

Asimismo la tesis “Desarrollo de una Aplicación Móvil: Caso Universiada 2012” (González Mérida, 2012) aprovechando el evento

deportivo Universiada, donde distintas universidades de México compiten en diversas disciplinas, aborda el proceso de realización de un aplicativo móvil para el sistema operativo iOS cuya finalidad es mantener informados a sus usuarios sobre resultados de competencias, ubicación de sedes, cronogramas y conteo de medallas; todo ello desplegado de forma intuitiva y con información en tiempo real. El trabajo concluye afirmando la utilidad que poseen las plataformas móviles, lo fácil que resulta trabajar con estas y con una aplicación que fue valorada por el público al que estuvo dirigido.

Desde España, el trabajo de fin de grado “Aplicación Android para la empresa Travelling-Service” (Gómez Matesanz, 2014) relata el análisis, diseño, implementación y pruebas de una aplicación móvil para el sistema operativo Android; ésta aplicación trabajó en conjunto con un servidor web y se encargó del contacto de los usuarios con la empresa así como la gestión y búsqueda de servicios que un viaje implica, tomando como punto de partida el hecho de que Travelling-Service tenía gran demanda a partir de medios telefónicos y que no estaba utilizando las bondades de la tecnología móvil como si lo hacían empresas similares.

Por otra parte El trabajo “Gamification: Encuestas de satisfacción de los servicios de alimentación javerianos” (Perez Garcia, 2013), contempla el desarrollo de un prototipo de

aplicación móvil. El problema planteado en este trabajo trata sobre la poca participación que hacían los estudiantes sobre las encuestas de satisfacción de los servicios de alimentación de su universidad, ante esta situación el autor plantea y desarrolla una aplicación móvil Android usando técnicas de gamificación, con lo que buscó motivar a los estudiantes a realizar las encuestas. El producto final de este trabajo fue sometido a prueba por muchos estudiantes, a los que en mayoría les pareció un muy buen aplicativo.

2.1.2. Antecedentes nacionales

La tesis “Diseño de una aplicación móvil para la consulta académica de la FIIS-UTP” (Aguirre Chacón & Sinche Ricra, 2013) expone la necesidad de crear un aplicativo móvil, aprovechando el avance de los teléfonos inteligentes, que se encargue de las consultas académicas más demandadas del portal web de la Universidad Tecnológica del Perú, al cual según explican los autores se accede mayoritariamente a través de un ordenador y que genera situaciones incómodas, como por ejemplo en la averiguación de notas después de un periodo prácticas.

Por otro lado hay universidades que han logrado desarrollar aplicaciones móviles relacionadas a la gestión universitaria entre otros servicios como es el caso de la Universidad de Lima y su

aplicación ULIMA APP (Universidad de Lima, 2014) que está disponible para los sistemas operativos Android y iOS, disponibles en Google Play y App Store respectivamente, la cual se encarga de mostrar noticias, conectar con el aula virtual en la que estudiantes pueden revisar sus calificaciones, mostrar el menú de la cafetería, informar sobre números telefónicos de atención, realizar pagos de mensualidades, ver horarios, etc. Esta aplicación fue desarrollada por un equipo de 7 personas pertenecientes al IT LAB de la carrera de Ingeniería de Sistemas de la universidad.

A su vez la Universidad Peruana de Ciencias Aplicadas (UPC) posee una aplicación similar llamada UPC Móvil (Universidad Peruana de Ciencias Aplicadas, 2014), aunque más completa ya que permite hacer reservaciones (computadoras y cubículos) y dar seguimiento a tramites; esta también se encuentra disponible para los sistemas operativos Android y iOS.

2.2. Bases Teóricas

2.2.1. SCRUM

Scrum es una metodología ágil utilizada para gestionar el desarrollo de proyectos; no es una técnica que describa como realizar un producto, más bien es un marco de trabajo en que pueden emplearse distintos procesos y técnicas. Scrum nos enseña la eficacia relativa entre las prácticas de gestión de

producto y prácticas de desarrollo de manera que podemos mejorar (Schwaber & Sutherland, 2013).

Scrum está compuesto por el Equipo de trabajo, los roles que desempeñan, los eventos, los artefactos realizados y las reglas asociadas. Cada componente tiene un fin específico que debe cumplirse para el éxito de Scrum (Caudevilla, 2010).

En la figura 1 se presenta el proceso de SCRUM y las relaciones entre sus componentes.

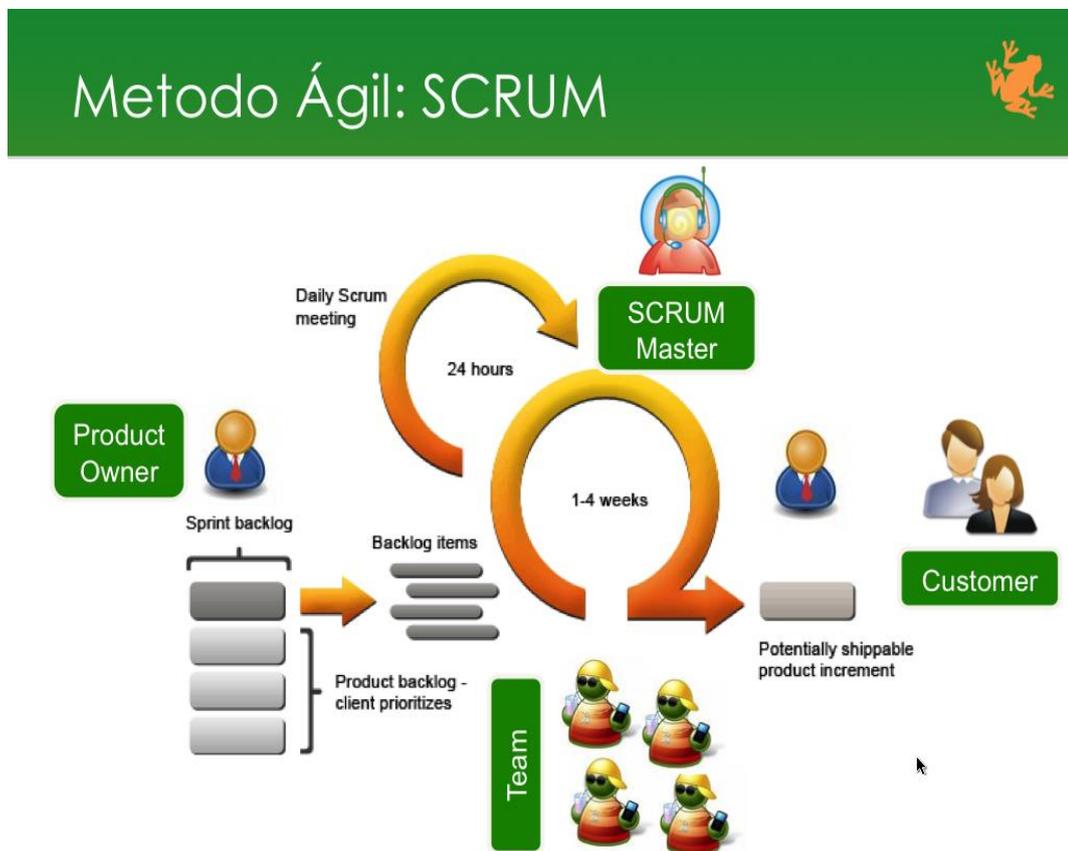


Figura 1 Proceso SCRUM (Caudevilla, Metodología Ágil: SCRUM, 2010)

Roles de SCRUM

- **Product owner**

Representante del cliente, escribe las historias de usuario (requerimientos de sistema) y las agrega al Product Backlog. Se asegura de que el equipo de trabajo cumpla con los requerimientos.

- **Scrum master**

Facilita las labores del equipo de trabajo de tal forma que puedan lograr los objetivos de cada Sprint. Asegura de que los roles y reglas se cumplan.

- **Equipo de desarrollo**

Convierte los requerimientos del Product Backlog en funcionalidades del software. Responsable de entregar el producto. Debe ser auto gestionado y organizado, y sus integrantes poseen habilidades variadas.

Artefactos del SCRUM

- **Sprint**

Es una iteración en el marco de trabajo de Scrum, en esta el equipo debe desarrollar las tareas establecidas en el Backlog a menos que el Scrum Master detecte un inconveniente (por ejemplo: cambios en las circunstancias del negocio, interferencias, la

tecnología utilizada no funciona, etc.) que haga inviable la continuación del sprint. Su duración máxima es de 30 días.

- **Product backlog**

Documento que lista los requerimientos del sistema a desarrollar. Es de carácter dinámico pues se mantiene y agregan más requisitos durante el ciclo de vida del proyecto.

- **Sprint backlog**

Grupo de requerimientos que son desarrollados en un determinado sprint, se estima una duración de trabajo de 4 a 16 horas para cada requerimiento.

Reuniones

- **Daily SCRUM**

Se realiza todos los días de un sprint y se comenta sobre el estado del proyecto. Tiene una duración de 15 minutos y son invitados tanto el equipo de desarrollo como los responsables. La reunión debe ocurrir a la misma hora y ubicación todos los días. Durante esta cada miembro del equipo contesta a tres preguntas: ¿Qué has hecho desde ayer?, ¿Qué es lo que harás hasta la reunión de mañana?, ¿Has tenido algún problema que te haya impedido alcanzar tu objetivo?

- **Sprint planning meeting**

Reunión de planificación del Sprint se lleva a cabo cada 15 o 30 días, en esta se selecciona el trabajo a realizar; con el equipo se prepara el Sprint Backlog, se identifica y comunica cuanto del trabajo se podrá realizar dentro del sprint actual. Dura ocho horas como máximo.

- **Sprint review meeting**

Reunión de Revisión del Sprint, se realizan las siguientes actividades: Revisar el trabajo completado y no completado, y presentar el trabajo completado a los interesados. Puede durar hasta 4 horas.

- **Sprint retrospective**

Retrospectiva del Sprint, es una reunión donde cada integrante del equipo comenta sus impresiones sobre el sprint recién terminado, el fin de la retrospectiva es realizar la mejora continua del proceso. Dura como máximo 4 horas.

2.2.2. Android

Android es un sistema operativo basado en Linux y creado inicialmente para dispositivos móviles, usa Java como lenguaje de programación para sus aplicaciones, las cuales son compiladas en byte-code para ser ejecutado en una máquina virtual llamada Dalvik (aunque en versiones más recientes ha sido reemplazada

por ART), que está diseñada para usar óptimamente la memoria y hardware en entornos móviles (Open Handset Alliance, 2015). Es de código abierto y evoluciona constantemente, permitiendo a desarrolladores acceder a las diversas funcionalidades que el hardware ofrece (Google Developers, 2015). Está conformado por alrededor de 12 millones de líneas de código habiéndose utilizado los lenguajes de programación C, C++, Java y el lenguaje de etiquetas XML (Ávila Mejía, 2012).

Arquitectura

Android está provisto por una pila de software compuesto de muchas capas (Cancela Garcia & Ostos Lobo, 2015) que pueden verse en la figura 2 y serán descritas a continuación.

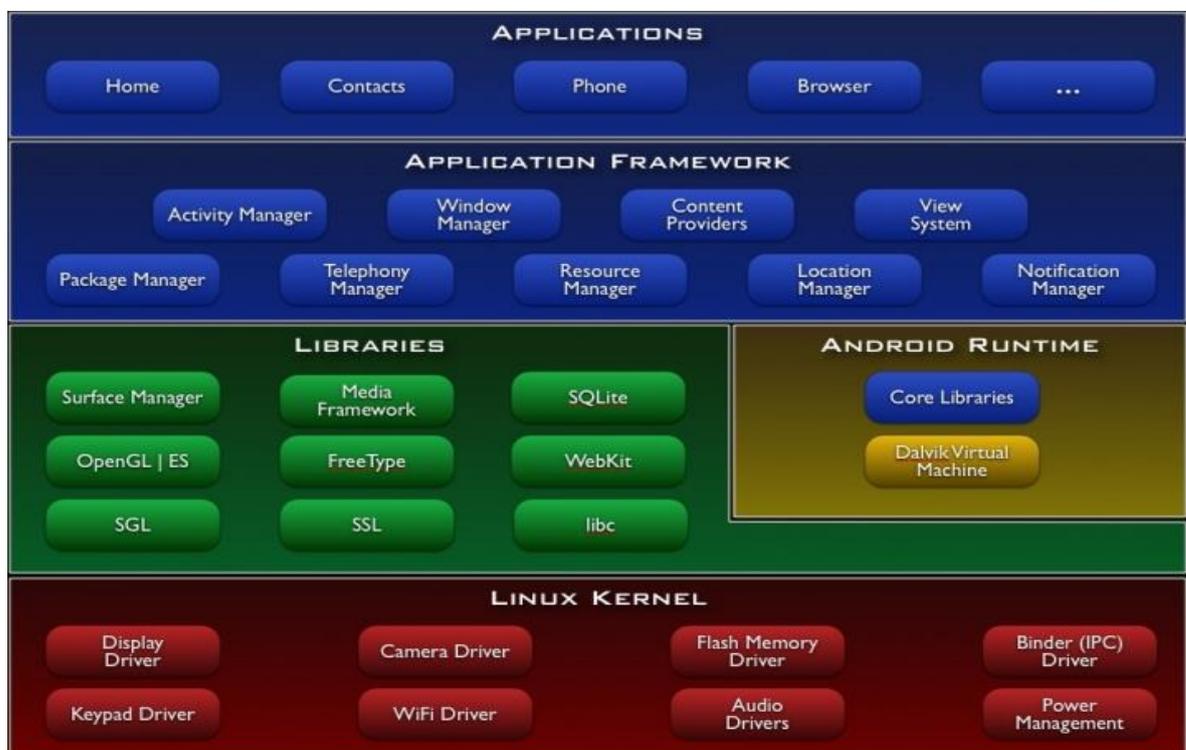


Figura 2 Arquitectura Android. (Android Developers, 2015)

Aplicaciones

Se ubican las aplicaciones que vienen preinstaladas en el sistema operativo y las que agregue el usuario del dispositivo, estas se sirven de las APIs y librerías de las capas anteriores.

Framework de aplicaciones

Conformado por el conjunto de herramientas de desarrollo usados por cualquier aplicación. Entre las principales se tiene:

- **Activity manager:** gestiona el ciclo de vida de las aplicaciones.
- **Window manager:** administra las ventanas de las aplicaciones y usa la librería Surface manager.
- **Telephony manager:** da acceso a las funcionalidades de telefonía proveídas por el hardware del dispositivo.
- **Content provider:** permite que las aplicaciones compartan datos entre ellas.
- **View System:** proporciona vistas que son incluidas en las interfaces de usuario, pueden ser listas, botones, text-boxes, etc.
- **Location manager:** en base al GPS o una red disponible determina el posicionamiento geográfico del dispositivo.
- **Notification manager:** otorga servicios para informar al usuario que una aplicación requiere su atención, permite emitir un sonido, activar el vibrador, encender las luces, etc.

- **Package manager:** permite obtener información de los paquetes instalados y gestiona la instalación de nuevos paquetes.
- **Resource manager:** proporciona acceso a los recursos de una aplicación que están incluidos en su código fuente, pueden ser cadenas de texto localizadas, imágenes, etc.

Librerías

Esta capa está compuesta por múltiples librerías escritas en C y C++ que son usadas por el Framework de aplicaciones, algunas de estas son:

- **Surface manager:** compone las imágenes que se muestran en pantalla realizando operaciones sobre mapas de bits.
- **Scalable Graphics Library (SGL):** motor gráfico 2D de Android.
- **OpenGL ES:** librería encargada de los gráficos 3D, utiliza aceleración por hardware si el dispositivo lo permite.
- **Web kit:** motor web usado por el navegador, puede ser incluido en otras aplicaciones.
- **Librería libc:** incluye las cabeceras y funciones según la librería estándar de C.
- **SSL:** permite acceder a internet de forma segura utilizando criptografía.

Runtime de Android

Conjunto de librerías base que proporcionan funcionalidades a las librerías Java. Cada aplicación en Android se ejecuta en una instancia de la máquina virtual Dalvik.

Kernel Linux

Funciona como capa de abstracción entre el hardware del dispositivo y la pila de software, también es usado para la gestión de procesos, memoria, drivers, etc.

2.2.3. Google Cloud Messaging

Google Cloud Messaging (GCM) (Google Developers, 2015) es un servicio gratuito que permite a los desarrolladores enviar mensajes “downstream” (desde servidores a aplicaciones cliente) y mensajes “upstream” (desde las aplicaciones cliente a servidores). El mensaje enviado podría indicar a la aplicación cliente que hay nuevos datos que se deben recoger del servidor, por lo cual tendría que sincronizarse con éste; un mensaje puede contener hasta 4 KB de datos.

GCM se encarga de poner en cola todos los mensajes que recibe y de entregarlos al destino indicado.

Arquitectura

Una implementación de GCM incluye un servidor de conexión proporcionado por Google, un servidor de aplicaciones de terceros que interactúa con el servidor de conexión, y una aplicación cliente.

En la figura 3 se aprecia la interacción entre un dispositivo móvil y una aplicación de terceros a través de GCM.

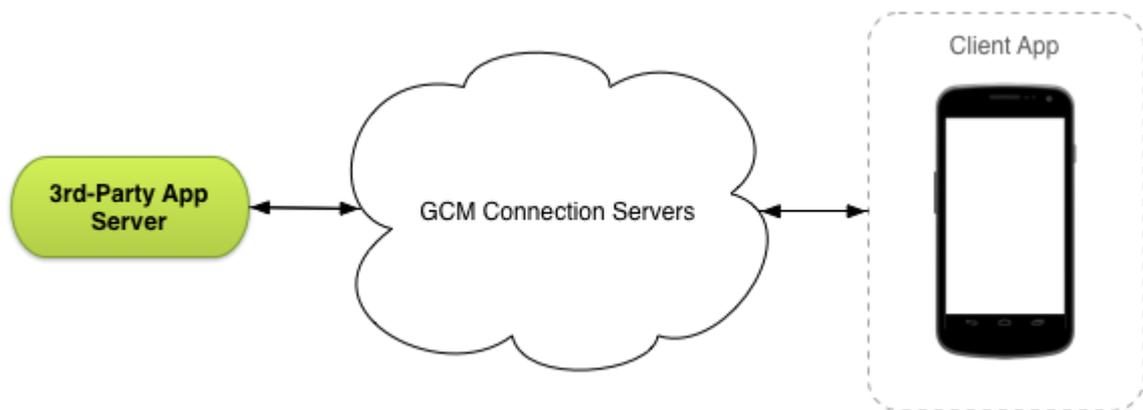


Figura 3 Arquitectura de GCM (Google Developers, 2015)

La interacción entre estos componentes se relaciona de la siguiente manera:

- Los servidores de conexión de GCM (GCM Connection Servers) proporcionados por Google toman los mensajes del servidor de aplicaciones de terceros (3rd- Party App Server) y envían estos mensajes a una aplicación cliente (Client App). Actualmente Google ofrece servidores de conexión para HTTP y XMPP.

- El servidor de aplicaciones envía mensajes a un servidor de conexión GCM el cual encola y almacena el mensaje, y luego lo envía a la aplicación cliente.
- Para que la aplicación cliente pueda recibir mensajes de GCM, esta debe registrarse en GCM y obtener un ID de registro. Si está utilizando el servidor de conexión XMPP, la aplicación cliente puede enviar mensajes “upstream” de vuelta al servidor de aplicaciones.

Otros aspectos a tomar en cuenta:

- Cada mensaje puede permanecer en cola durante 4 semanas como máximo.
- Un mensaje puede ser enviado a la vez a 1000 dispositivos distintos.
- GCM puede mantener una cola de 100 mensajes por dispositivo. Si se sobrepasa esta cantidad todos los mensajes anteriores se borran.
- La cantidad de mensajes enviados por minuto hacia un solo dispositivo no puede ser muy alta, de lo contrario GCM asumirá los mensajes como spam y no los recibirá.

2.3. Marco conceptual

Smartphone

Un Smartphone (Alegsa, 2015) es un teléfono móvil que incorpora características de una computadora personal. Los smartphones suelen permitir al usuario instalar nuevas aplicaciones, aumentando así sus funcionalidades. Esto es posible porque ejecutan un sistema operativo potente de fondo. Pueden incluir un mini teclado, una pantalla táctil, un lápiz óptico, etc. Los smartphone tienden a incluir acceso a internet, servicios de e-mail, cámara integrada, navegador web, procesador de textos, etc.

SQLITE

SQLite (SQLite Org, 2015) es una librería escrita en C que implementa un motor de base de datos transaccional. Es de código abierto y libre para cualquier uso sea comercial o privado. En contraposición a otras bases de datos, SQLite no trabaja como un proceso separado de la aplicación principal y toda la información relacionada a la base de datos es almacenada en un solo archivo cuyo formato permite su uso entre sistemas operativos de 32 o 64 bits.

Cualquier programa que utilice la funcionalidad de SQLite lo hará a través de llamadas simples a subrutinas y funciones, con ello se reduce la latencia en el acceso a la base de datos debido a que las llamadas a funciones son más eficientes que la comunicación entre procesos

(Universidad de El Salvador Facultad de Ingeniería y Arquitectura Escuela de Ingeniería de Sistemas Informáticos, 2014).

Prototipo

Desde el punto de vista de la informática (Alegsa, 2010), un prototipo de un sistema informático, es una muestra más simplificada de un sistema. Un prototipo provee resultados rápidos y nos da un vistazo a grandes rasgos de cómo será el sistema; este puede aceptarse y convertirse en un sistema completo, o descartarse al no cumplirse las expectativas previstas. Los prototipos suelen ser presentados a los clientes quienes se encargan de validar los resultados.

APK

Android Application Package (APK) es un formato y extensión de archivo que identifican a las aplicaciones Android. Los archivos de este tipo utilizan el formato de compresión ZIP, en su interior se encuentran los recursos que la aplicación requiere (imágenes, ficheros de texto simple, ficheros XML, archivos de configuración, etc.) y un archivo de extensión "dex" el cual es un ejecutable de la máquina virtual Dalvik (Blazek, 2011).

Dalvik

Es la máquina virtual de Android, su función principal es interpretar los archivos "dex" y ejecutar el byte-code de éstos (Rodríguez, 2011). Está diseñado para trabajar con recursos de hardware limitados (memoria, CPU, almacenamiento, etc.) y múltiples instancias de ésta

pueden ejecutarse a la vez sin mayor impacto en el rendimiento del dispositivo. Dalvik está basado en registros (un registro es una memoria que almacena operaciones o valores muy utilizados por el CPU), lo que significa que los operandos para las instrucciones están dirigidos en la misma instrucción permitiendo que éstas se ejecuten rápidamente (Sinnathamby, 2012).

Cliente / Servidor web

El cliente es un programa que requiere o solicita servicios, mientras que un servidor es un programa que brinda servicios (base de datos, imágenes, archivos de texto, correo, etc.). Desde un punto de vista lógico, el cliente y servidor se encuentran separados y la comunicación entre ellos se realiza a través de una infraestructura de red.

En entornos web, el cliente realiza las peticiones de un servicio hacia el servidor el cual responde transfiriendo documentos hipertexto, siendo necesario el uso del protocolo HTTP (Hypertext Transfer Protocol) para que la comunicación se realice (Departamento de Informática de la Universidad de Valladolid, 2006)).

MySQL

Es un SGBD (Sistema Gestor de Bases de Datos) de código abierto, multiplataforma, fácil de usar, de buena performance y confiabilidad, utilizado por gran cantidad de sistemas cliente-servidor (aplicaciones web mayoritariamente) (MySQL, 2015). Esta desarrollado en C/C++ y entre sus características resaltantes se encuentran la posibilidad de

crear y configurar usuarios, permitiendo asignarles distintos permisos o privilegios; facilidad para la importación y exportación de datos, contar con un conjunto de datos diverso, una API disponible para otros lenguajes de programación (Java, PHP, entre otros), etc. (Casillas Santillán, Gibert Ginesta, & Pérez Mora, 2008).

CAPITULO 3: DESCRIPCIÓN DEL PROTOTIPO

3.1. Análisis del prototipo

3.1.1. Requerimientos del sistema

Siguiendo los lineamientos de la metodología SCRUM, para el inicio del desarrollo del prototipo es necesario recabar las historias de usuario (revisar anexo “Historias de Usuario”), a partir de ellas se establecen los requerimientos y con estos últimos se construirá el “Product Backlog”. Cabe resaltar que dicho documento crecerá a lo largo de las iteraciones siempre que se soliciten nuevos requerimientos.

Ya que el prototipo móvil requiere indispensablemente de un servidor web con el cual sincronizarse, parte del desarrollo del producto contempla funcionalidades realizables a partir del servidor mencionado, el cual soportará el lenguaje de programación PHP y sistema de gestión de bases de datos MySQL. En la tabla 1 se muestra el Product Backlog del sistema.

Código	Requerimiento	Estimación (horas)	Prioridad
REQ-101	Desarrollo de mockups	4	1
REQ-102	Modelado de la base de datos	4	1
REQ-103	Creación de la base de datos	4	1
REQ-104	Desarrollo de scripts PHP para el registro de notas	24	1
REQ-105	Maquetado de la interfaz web	4	3
REQ-106	Conectar con GCM desde los scripts PHP	4	2
REQ-107	Enviar mensajes a GCM después de una actualización de notas	2	2
REQ-108	Modelar y crear la base de datos SQLite para la aplicación móvil	2	1
REQ-109	Desarrollar las clases java que administren el login	3	1
REQ-110	Sincronizar con el servidor automáticamente después del login	8	1
REQ-111	Desarrollar la interfaz gráfica para el login del estudiante	2	3
REQ-112	Desarrollar las clases java que se permitan ver las notas del ciclo actual.	5	1
REQ-113	Desarrollar la interfaz gráfica para las notas del ciclo actual	4	3
REQ-114	Desarrollar las clases java que se encarguen del historial académico	5	1
REQ-115	Desarrollar la interfaz gráfica para el historial académico	4	3
REQ-116	Implementar la descarga del historial en formato PDF	6	3
REQ-117	Generar una notificación local por cada notificación de GCM recibida	2	2
REQ-118	Habilitar la redacción y envío de mensajes del docente desde la interfaz web	3	2
REQ-119	Enviar notificación a GCM después de redactar un mensaje	1	2
REQ-120	Almacenar las mensajes entrantes en la BD SQLite	3	2
REQ-121	Programar las clases java para mostrar los mensajes	2	2
REQ-122	Diseñar la interfaz gráfica para el historial de mensajes	2	3
REQ-123	Desarrollar las clases para visualizar la información personal	1	3
REQ-124	Diseñar la interfaz de información personal	1	3
REQ-125	Mostrar un contador de notificaciones en el ícono de la aplicación	1	3
REQ-126	Agregar la opción de sincronización forzada	3	2

Tabla 1 Product Backlog del sistema (Elaboración propia)

3.1.2. Sprint backlog

En base al Product Backlog se crea por cada iteración del proyecto un Sprint Backlog, en éste se describen el grupo de tareas a realizar por el equipo de desarrollo para cumplir con un determinado grupo de requerimientos. En la tabla 2 se presentan los Sprints Backlog de las 4 iteraciones que forman parte del desarrollo del producto.

Iteración	Código	Requerimiento
Sprint 1	REQ-101	Desarrollo de mockups
	REQ-102	Modelado de la base de datos
	REQ-103	Creación de la base de datos
	REQ-104	Desarrollo de scripts PHP para el registro de notas
	REQ-105	Maquetado de la interfaz web
Sprint 2	REQ-108	Modelar y crear la base de datos SQLite para la aplicación móvil
	REQ-106	Conectar con GCM desde los scripts PHP
	REQ-110	Sincronizar con el servidor automáticamente después del login
	REQ-109	Desarrollar las clases java que administren el login
	REQ-107	Enviar mensajes a GCM después de una actualización de notas
REQ-111	Desarrollar la interfaz gráfica para el login del estudiante	
Sprint 3	REQ-112	Desarrollar las clases java que se permitan ver las notas del ciclo actual.
	REQ-114	Desarrollar las clases java que se encarguen del record académico
	REQ-113	Desarrollar la interfaz gráfica para las notas del ciclo actual
	REQ-115	Desarrollar la interfaz gráfica para el record académico
	REQ-116	Implementar la descarga del historial en formato PDF
REQ-117	Generar una notificación local por cada notificación de GCM recibida	
Sprint 4	REQ-118	Habilitar la redacción y envío de mensajes del docente desde la interfaz web
	REQ-119	Enviar notificación a GCM después de redactar un mensaje
	REQ-120	Almacenar las mensajes entrantes en la BD SQLite
	REQ-121	Programar las clases java para mostrar los mensajes
	REQ-122	Diseñar la interfaz gráfica para el historial de mensajes
	REQ-123	Desarrollar las clases para visualizar la información personal
	REQ-124	Diseñar la interfaz de información personal
	REQ-125	Mostrar un contador de notificaciones en el ícono de la aplicación
REQ-126	Agregar la opción de sincronización forzada	

Tabla 2 Especificación de los Sprint Backlog (Elaboración propia)

3.2. Construcción del prototipo

3.2.1. Realización del Sprint 1

Como primer paso en la construcción del prototipo se procede a crear los mockup que corresponden a las figuras 4, 5, 6, 7, 8, 9, 10 y 11, los que mostrarán a un alto nivel las interfaces con las que interactuará el usuario.



Figura 4 Mockup login (Elaboración propia)



Figura 5 Mockup menú principal (Elaboración propia)



Figura 6 Mockup información personal (Elaboración propia)

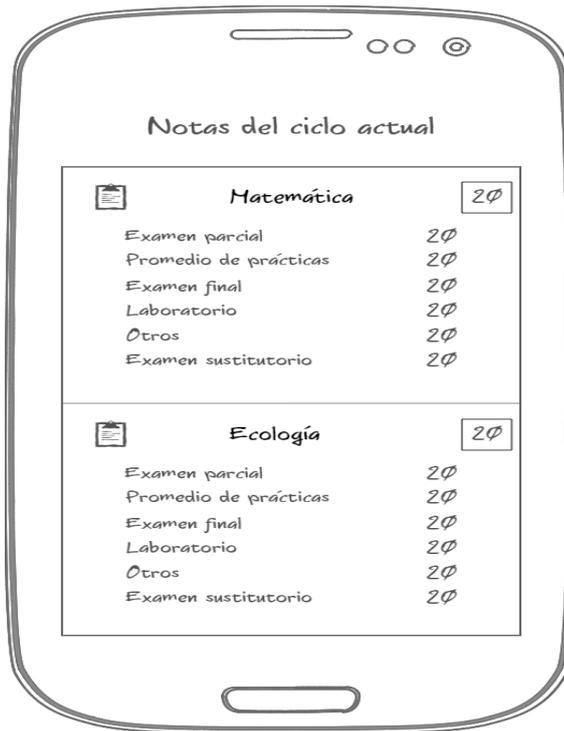


Figura 7 Mockup notas del ciclo actual (Elaboración propia)

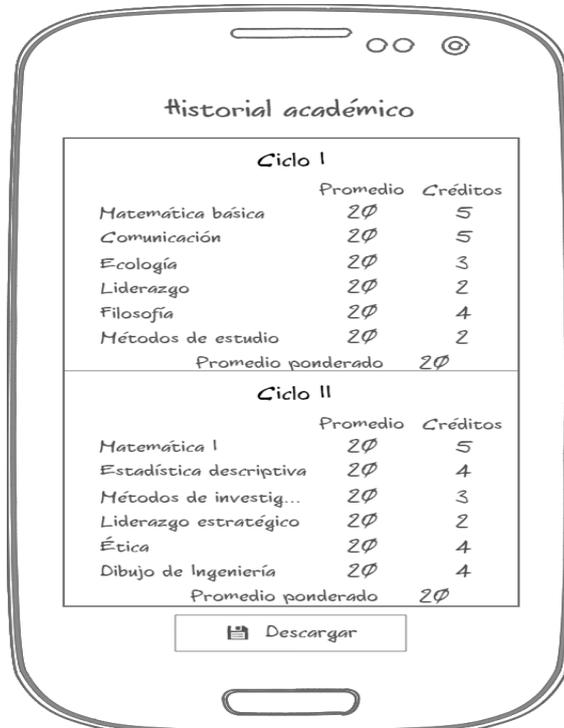


Figura 8 Mockup historial académico (Elaboración propia)

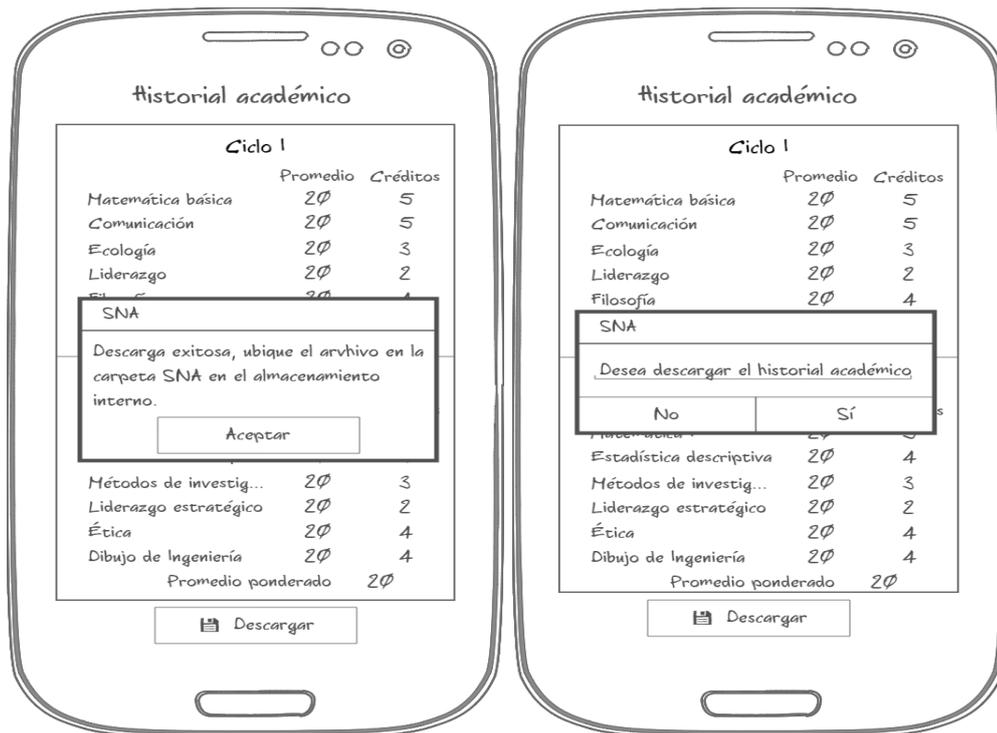


Figura 9 Mockup descarga de historial académico (Elaboración propia)

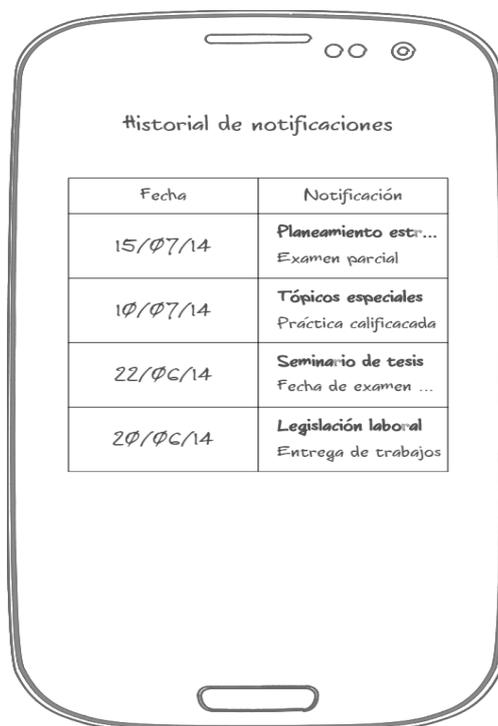


Figura 10 Mockup historial de notificaciones (Elaboración propia)



Figura 11 Mockup recepción de notificaciones y actualización de icono de aplicación (Elaboración propia)

Para el desarrollo del prototipo es necesario contar con una fuente de datos, como lo indica el requerimiento REQ-102, que permita tanto al usuario de la aplicación visualizar sus registros académicos y al docente registrar notas de sus estudiantes. En la figura 12 se presenta el modelo de base de datos MySQL con el que trabajará el servidor web, para realizar este requerimiento fue necesario el uso del software MySQL Workbench (MySQL, 2015).

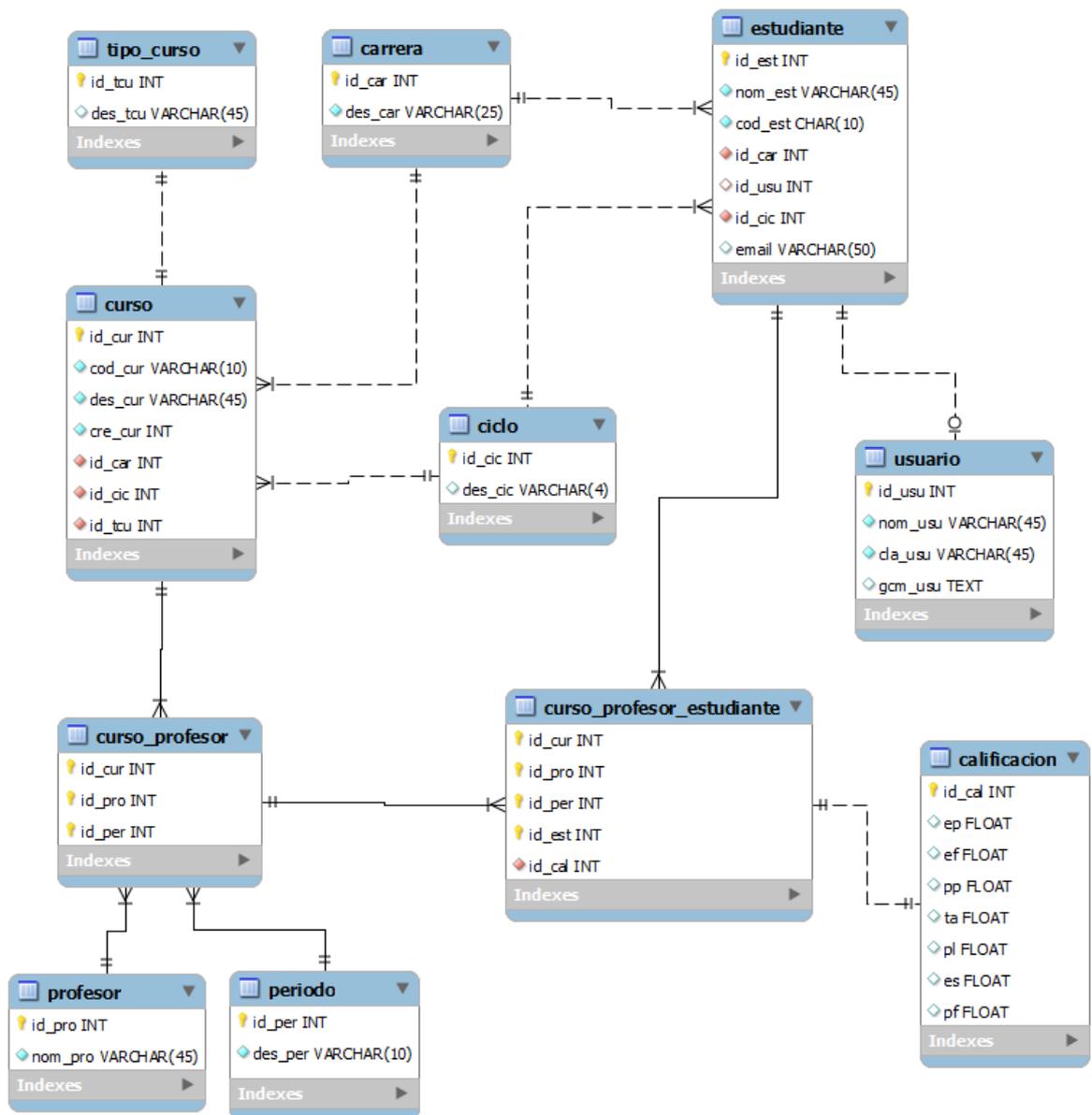


Figura 12 Modelo de base de datos del sistema (Elaboración propia)

A partir del modelo de base de datos se genera y ejecuta el script SQL necesario que permita su creación en el SGBD MySQL (REQ-103).

Disponiendo de la base de datos se procede a crear los scripts PHP que le permitirán al docente registrar las notas de sus estudiantes en la base de datos (REQ-104).

El maquetado de la interfaz web se realizó utilizando HTML y CSS (REQ-105). En la figura 13 se muestra la interfaz web del docente donde registrará notas y redactará mensajes.

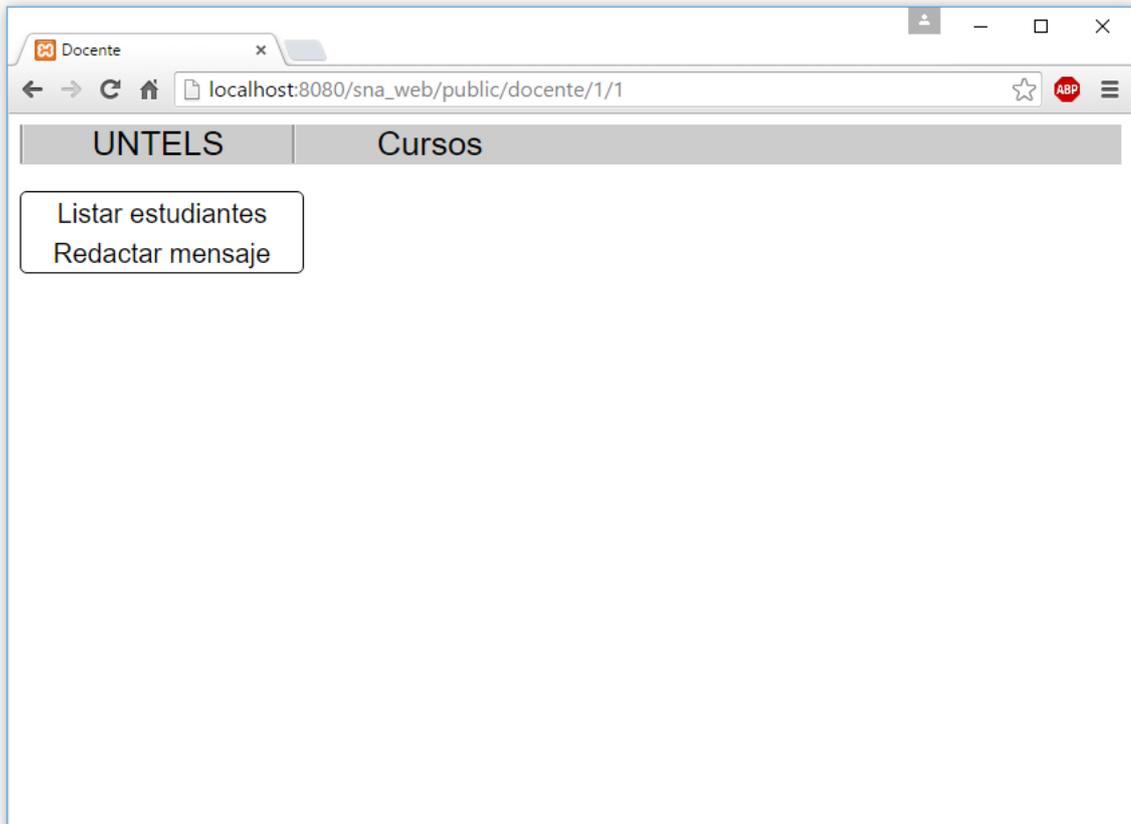


Figura 13 Interfaz web docente (Elaboración propia)

3.2.2. Realización del Sprint 2

Después de habilitar el registro de notas es necesario conectar la aplicación alojada en el servidor web con GCM. Para esto fue necesario registrar la aplicación en la consola de Google tal como muestra la figura 14, con ello se obtuvo una clave de aplicación que permitió enviar notificaciones a GCM desde el servidor.

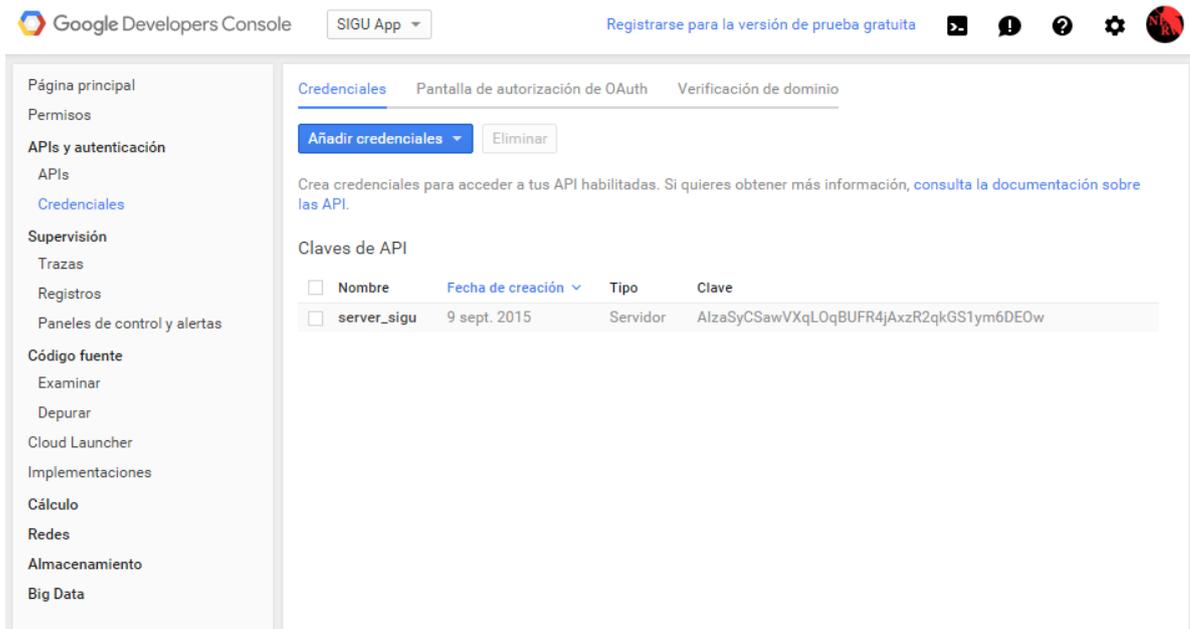


Figura 14 Google developers console. (Google Developers, 2015)

Luego de registrar la aplicación se modifican los scripts PHP encargados de registrar las notas para que se envíe una notificación a GCM cada vez que se actualicen los registros de un estudiante en particular (REQ-106, REQ-107).

Con el servidor web funcionando correctamente, el primer paso para construir el prototipo es crear una base de datos local en SQLite (REQ-108) la que almacenará los registros del estudiante que haga uso de éste. En la figura 15 se muestra el modelo de base de datos SQLite; para cumplir con este requerimiento se hizo uso del software SQLite Studio (SQLite Studio, 2015).

Para la programación del prototipo se utilizó el entorno de desarrollo Android Studio (Android Developers, 2015) .

Como indican los requerimientos REQ-109 y REQ-111, el prototipo solo permitirá el acceso a aquellos estudiantes que ingresen su usuario y contraseña correctos, para tal motivo se crean las clases que hagan las peticiones al servidor web y que verifiquen si el estudiante ingresó correctamente los datos solicitados. Habiendo habilitado el login, el prototipo se sincroniza con el servidor web y extrae todo el historial académico del estudiante para después almacenarlo localmente en la base de datos SQLite (REQ-110).

3.2.3. Realización del Sprint 3

Se procede a implementar las clases java para mostrar las notas del ciclo actual al estudiante (REQ-112 y REQ-113).

Siguiendo el proceso, se desarrollan los requerimientos REQ-114 y REQ-115 que permitirán al estudiante visualizar su historial académico.

Hasta este punto el prototipo ya es funcional, sin embargo, la base de datos SQLite se ha mantenido sin cambio alguno después de la sincronización inicial, por ello es necesario desarrollar las clases que administren las notificaciones enviadas por GCM para mantener la base de datos SQLite actualizada (REQ-117), siendo así se implementa el servicio Android de GCM en el prototipo, el cual actuará como un proceso en segundo plano para cumplir 2 funciones: Registrar el smartphome en los

servidores de GCM (retornándonos un identificador único que se almacenará en la base de datos MySQL) y estar a la espera de mensajes entrantes enviados por GCM; también se generan notificaciones locales, que emitirán un sonido o harán que el dispositivo vibre, para avisar al estudiante sobre la información entrante.

Con los datos actualizados es posible generar la descarga del historial académico en formato PDF (REQ-116), para ello fue necesario usar la librería iText (SourceForge, 2015) la cual puede ser usada gratuitamente bajo la licencia AGPL.

Terminado este sprint el flujo de trabajo del prototipo con GCM y el servidor web se puede apreciar en la figura 16.

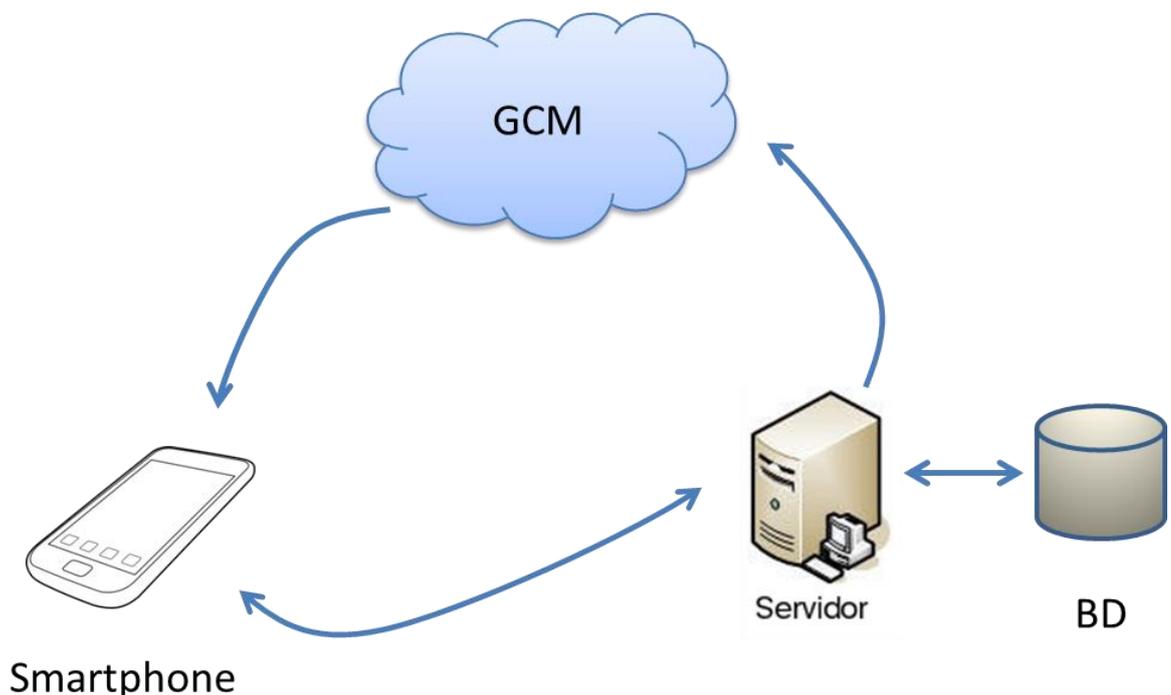


Figura 16 Flujo Prototipo, GCM y Servidor web. (Elaboración propia)

- El servidor contiene la aplicación web PHP y la base de datos MySQL.

- Haciendo uso de la aplicación web se actualizan los datos del estudiante en la BD MySQL y se envía un mensaje (conteniendo los datos actualizados y el identificador del dispositivo de destino) a GCM.

- GCM recibe el mensaje, lo encola y envía al dispositivo destino.

- El Smartphone recibe el mensaje de GCM y almacena los datos en la BD SQLite.

- Cuando el estudiante hace login o fuerza la sincronización desde el prototipo móvil el servidor retorna todos sus datos.

3.2.4. Realización del Sprint 4

En este cuarto sprint se desarrolló la última funcionalidad importante, que es permitir que un docente redacte mensajes para sus estudiantes tal como está determinado en los requerimientos REQ-118, REQ-119, REQ-120, REQ-121 y REQ-122. Por este motivo se habilitó un formulario en la interfaz web donde se redactará el mensaje cuyo envío se realiza a través de GCM y como tal, llegará a modo de notificación a los smartphones de los estudiantes; todo mensaje recibido es almacenado en la base de datos SQLite.

También se desarrolló una interfaz en el prototipo que permite al usuario revisar mensajes anteriores.

Tanto las clases y la interfaz que se encargan de mostrar la información personal (REQ-123, REQ-124) fueron desarrolladas

en esta iteración pues su complejidad y prioridad no eran tan relevantes.

Como detalle visual que apoye al estudiante a determinar si hubo actualizaciones en sus datos, se agregó un contador sobre el icono de la aplicación el cual mostrará la cantidad de notificaciones sin revisar.

Por último se concluyó el desarrollo del prototipo agregando la opción de sincronización forzada con el servidor, esto para mantener actualizada la base de datos aún si GCM dejase de funcionar.

3.3.Revisión y consolidación de resultados

3.3.1. Análisis de encuesta

Para obtener la opinión de los estudiantes respecto al SIGU y determinar su interés sobre una aplicación para smartphones que desempeñe una función similar, pero que aproveche las bondades de la tecnología móvil, se desarrolló una encuesta online (accesible a partir de este link: <http://goo.gl/forms/c8EO8kgI9A>) el día miércoles 14 de octubre, la cual estuvo constituida de las siguientes preguntas:

¿A qué carrera perteneces?

- Administración de Empresas
- Ing. Ambiental
- Ing. Electrónica y Telecomunicaciones
- Mecánica y Eléctrica

- Ing. De Sistemas

¿Qué tal útil considera al SIGU?

- Muy útil.
- Útil.
- Poco útil.

¿Mientras usaba el SIGU, éste presentó algún problema?

- Sí.
- No.

¿Con cuál(es) de los siguientes problemas se encontró?

- El sistema de dejó de funcionar.
- El sistema demoraba mucho en dar respuesta
- No se sabe cuando un profesor sube las notas
- Otros ...

¿Utiliza Smartphone?

- Sí
- No

¿De qué sistema operativo?

- Android
- iOS
- Windows Phone
- Otros ...

¿Utilizaría una aplicación que se encargue de su información académica?

- Sí.
- No.

¿Cuál(es) de las siguientes funcionalidades le gustaría que tuviera dicha aplicación?

- Ver historial académico.
- Ver las notas del ciclo actual.
- Alertar cuando un profesor suba una nota.
- Alertar la fecha de una práctica.
- Ver historial de alertas.
- Otros...

De esta encuesta se pudo recuperar las respuestas de 38 estudiantes de la UNTELS. A continuación se muestra el análisis de los datos más resaltantes de la encuesta haciendo uso de la estadística.

El primer objetivo de la encuesta era conocer que tan aceptado es el SIGU, para ello el criterio elegido fue la utilidad que tiene este sistema para cada estudiante. Como se aprecia en la figura 17, 84% de los encuestados (32 personas) consideran que el sistema es útil, en otras palabras, que el sistema cumple con el fin para el que fue desarrollado (brindar información académica, entre otros) y con las exigencias de los estudiantes.

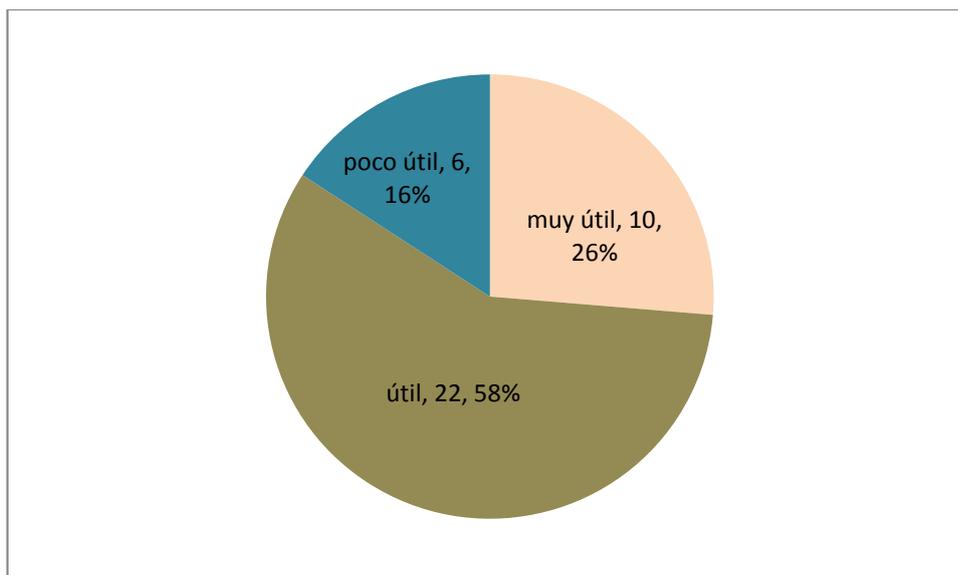


Figura 17 Valoración del SIGU según utilidad. (Elaboración propia)

El siguiente aspecto relevante de la encuesta fue determinar qué problemas detectaron los estudiantes mientras interactuaban con el sistema. La figura 18 muestra la cantidad de veces que un problema se presentó entre distintos estudiantes, siendo el problema “No se sabe cuándo un profesor sube las notas” el más recurrente (21 personas lo consideran), cabe resaltar que este problema no está relacionado a ninguna falla de programación o de hardware, sino a la inexistencia de un puente de comunicación que informe al estudiante el momento oportuno para acceder al sistema y ver sus registros actualizados.

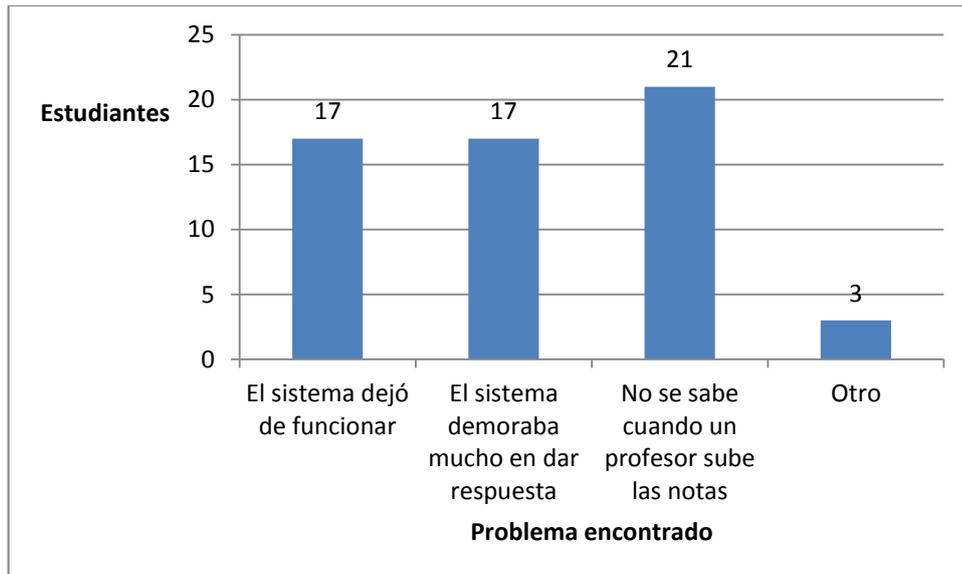


Figura 18 Problema encontrado vs Estudiantes. (Elaboración propia)

Las preguntas siguientes están más relacionadas al propósito de este trabajo, por ello era necesario conocer la penetración de las tecnologías móviles entre los encuestados.

La primera pregunta busca segmentar al conjunto de estudiantes tomando como criterio si usan o no un smartphone; tal como muestra la figura 19, el 82% (31 personas) utilizan éste dispositivo siendo un porcentaje alto para la cantidad total de encuestados.

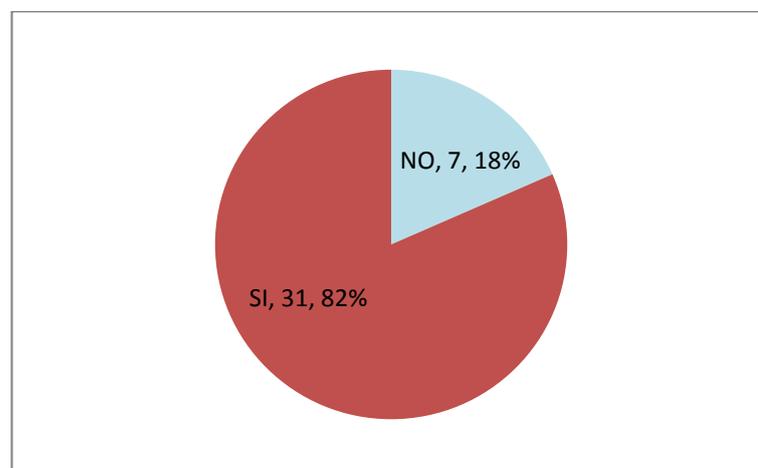


Figura 19 Porcentaje de estudiantes que usan Smartphone (Elaboración propia)

A partir de las 31 personas que indicaron poseer un smartphone se realiza otra segmentación, tomando como criterio el sistema operativo del dispositivo.

La figura 20 indica que un 77% (24 personas) tienen Android como sistema operativo lo cual es un resultado muy conveniente para los fines del presente trabajo. Sin embargo no se deben despreciar las proporciones de estudiantes que usan otros sistemas operativos pues la cantidad real que representan podría ser relevante para generar trabajos similares al presente.

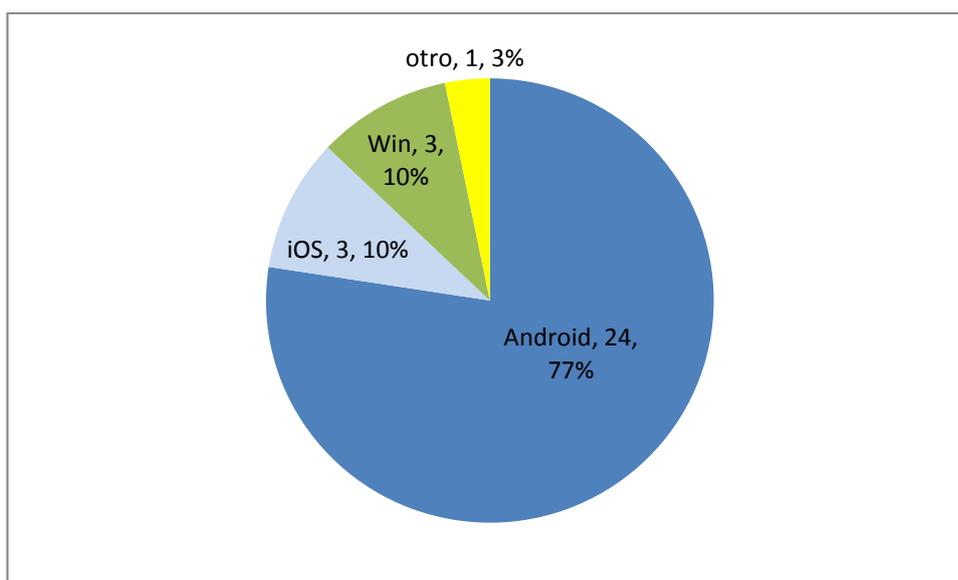


Figura 20 Smartphones según sistema operativo (Elaboración propia)

Tomando nuevamente al grupo de 31 personas que indicaron usar un smartphone, se les consultó si estarían dispuestos a usar una aplicación con funcionalidades similares al SIGU, y como muestra la figura 21, 90% (28 personas) la usarían. Aquí no se aplicó la segmentación por sistema

operativo pues era más importante conocer la disposición de los estudiantes.

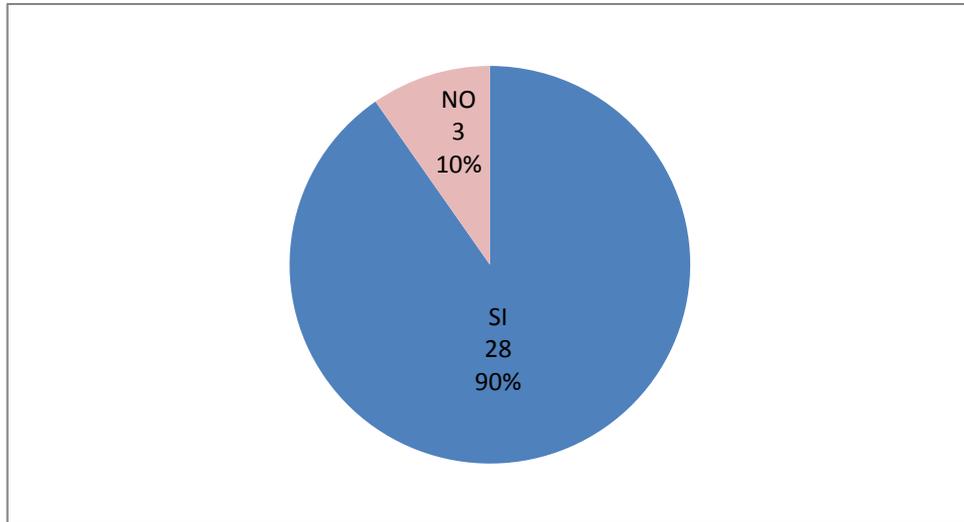


Figura 21 Estudiantes que utilizarían la aplicación (Elaboración propia)

Finalizando el análisis de la encuesta, se preguntó a los estudiantes que eligieran entre múltiples funcionalidades que les gustaría ver en la aplicación prototipo. Como se presenta en la figura 22 todas las funcionalidades sugeridas fueron elegidas por la mayoría de encuestados, lo cual es un buen indicador ya que la aplicación prototipo cumple con estas funcionalidades.

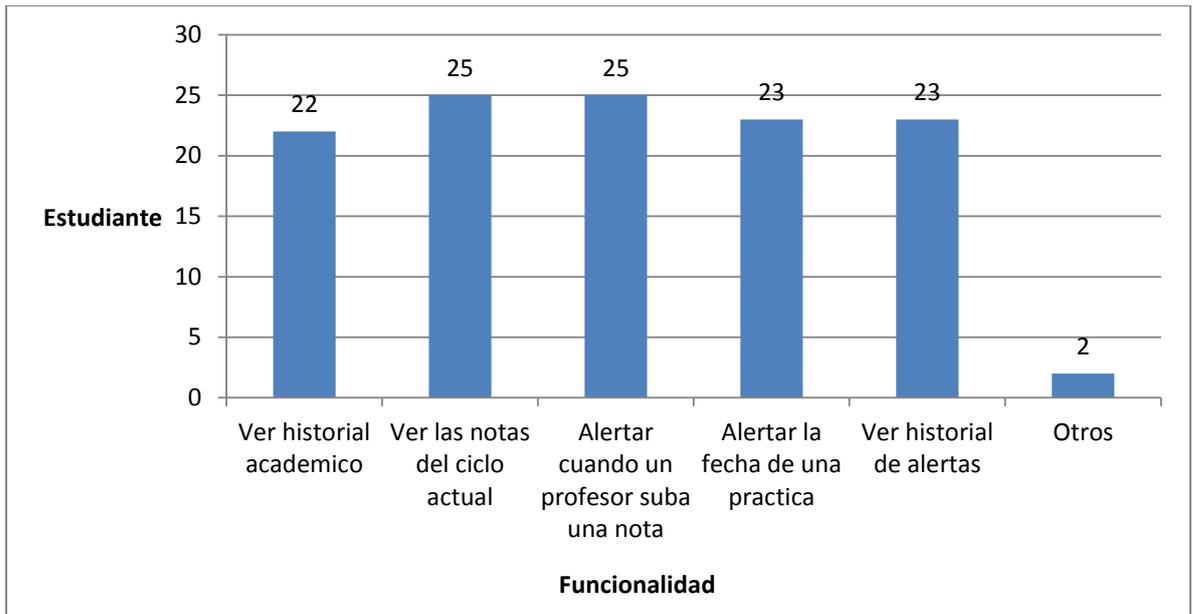


Figura 22 Funcionalidades requeridas por estudiantes (Elaboración propia)

3.3.2. Revisión del prototipo

El resultado final del presente trabajo es un prototipo de aplicación Android que se sincroniza con un servidor web, es por ello que parte del desarrollo del proceso SCRUM comprendió la realización de una aplicación web que no solo interactúa con la base de datos principal, sino que permite registrar calificaciones y enviar mensajes a estudiantes, esto para demostrar el funcionamiento de GCM conjuntamente con el prototipo.

Para comprobar que el prototipo cumple con los requerimientos descritos en el Product Backlog, se realizó una serie de casos de prueba de caja negra, los que serán descritos a continuación.

Caso de prueba 1

Objetivo: comprobar que el login solo permite el ingreso a usuarios existentes.

Código estudiante	Clave	Resultado esperado
2009200132	estudiante	ingresa
2009100145	estudiante	no ingresa
' or '1'='1	estudiante	no ingresa

Tabla 3 Entradas caso de prueba 1 (Elaboración propia)

Los resultados para el primer grupo de entradas se presentan en la figura 23. El prototipo encuentra al usuario y lo redirige al menú principal.

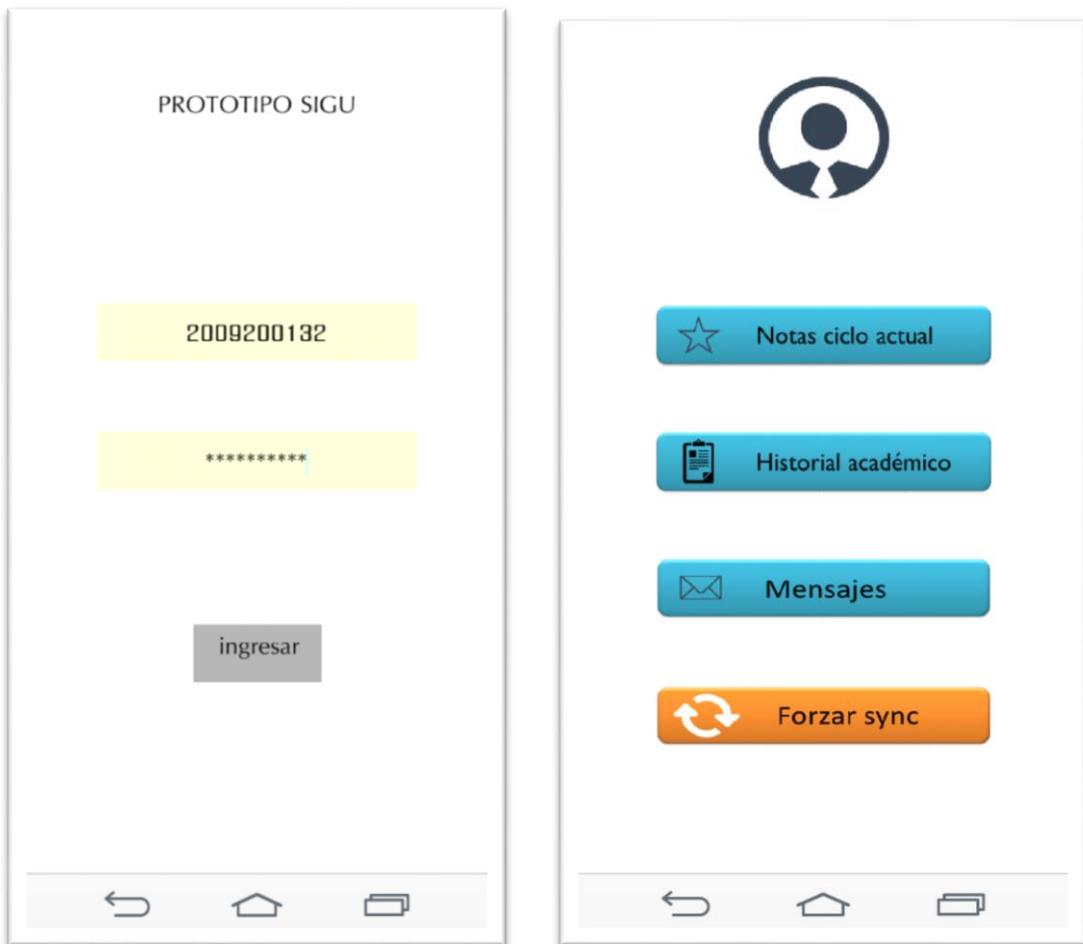


Figura 23 Resultados de la primera entrada del caso de prueba 1 (Elaboración propia)

Para la segunda y tercera entrada, el prototipo no permite ingresar al usuario y muestra un mensaje de “Usuario inexistente”. Los resultados se presentan en la figura 24.

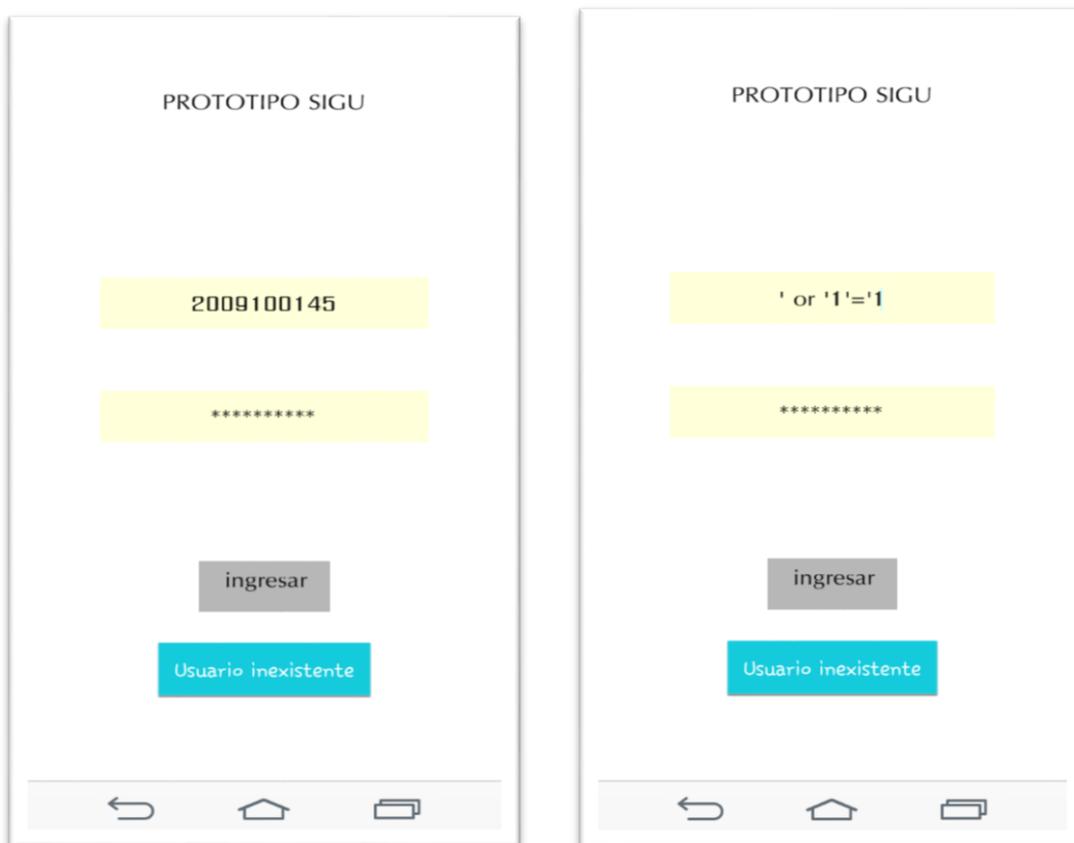


Figura 24 Resultados de la segunda y tercera entrada del caso de prueba 1

Caso de prueba 2

Objetivo: comprobar que el prototipo se sincronizó correctamente con el servidor y extrajo todos los datos relevantes del usuario.

Acción	Resultado esperado
Pulsar el logo de usuario	Ver la información personal del usuario
Pulsar el botón "Notas del ciclo actual"	Ver las notas del último ciclo que el estudiante este cursando
Pulsar el botón "Historial de notificaciones"	Ver el historial académico del usuario
Pulsar el botón "Mensajes"	Ver la lista de mensajes enviados por los profesores

Tabla 4 Acciones Caso de prueba 2 (Elaboración propia)

Como muestra la figura 25, al pulsar el logo de usuario, la información personal del usuario (en este caso el estudiante) es presentada en una vista del prototipo.



Figura 25 Información personal del usuario (Elaboración propia)

Al pulsar el botón “Notas del ciclo actual” el prototipo redirige al usuario a una vista donde aparecen las calificaciones de todos los cursos del último ciclo que el estudiante este llevando. Para observar todo el contenido de esta vista el usuario debe hacer un toque deslizante hacia la parte inferior o superior de la pantalla. Los resultados de esta acción aparecen en la figura 26.



Figura 26 Notas del ciclo actual (Elaboración propia)

Cuando el usuario presiona el botón de “Historial académico” se despliega una interfaz gráfica donde están listados los periodos con sus respectivos cursos y promedios finales que el estudiante ha acumulado a lo largo de su estadía en la universidad. Esta vista cambia la orientación de la pantalla para que la información sea mejor apreciada y al igual que la anterior vista se realiza un toque deslizando sobre la pantalla para ver todos los datos extraídos. La figura 27 muestra el resultado de esta acción.



Figura 27 Historial académico (Elaboración propia)

Pulsando el botón “Mensajes” aparece en pantalla la lista de mensajes enviados por los profesores. Esta lista esta ordenada según la hora de envió ubicando a los más recientes al inicio. La vista resultante se presenta en la figura 28.

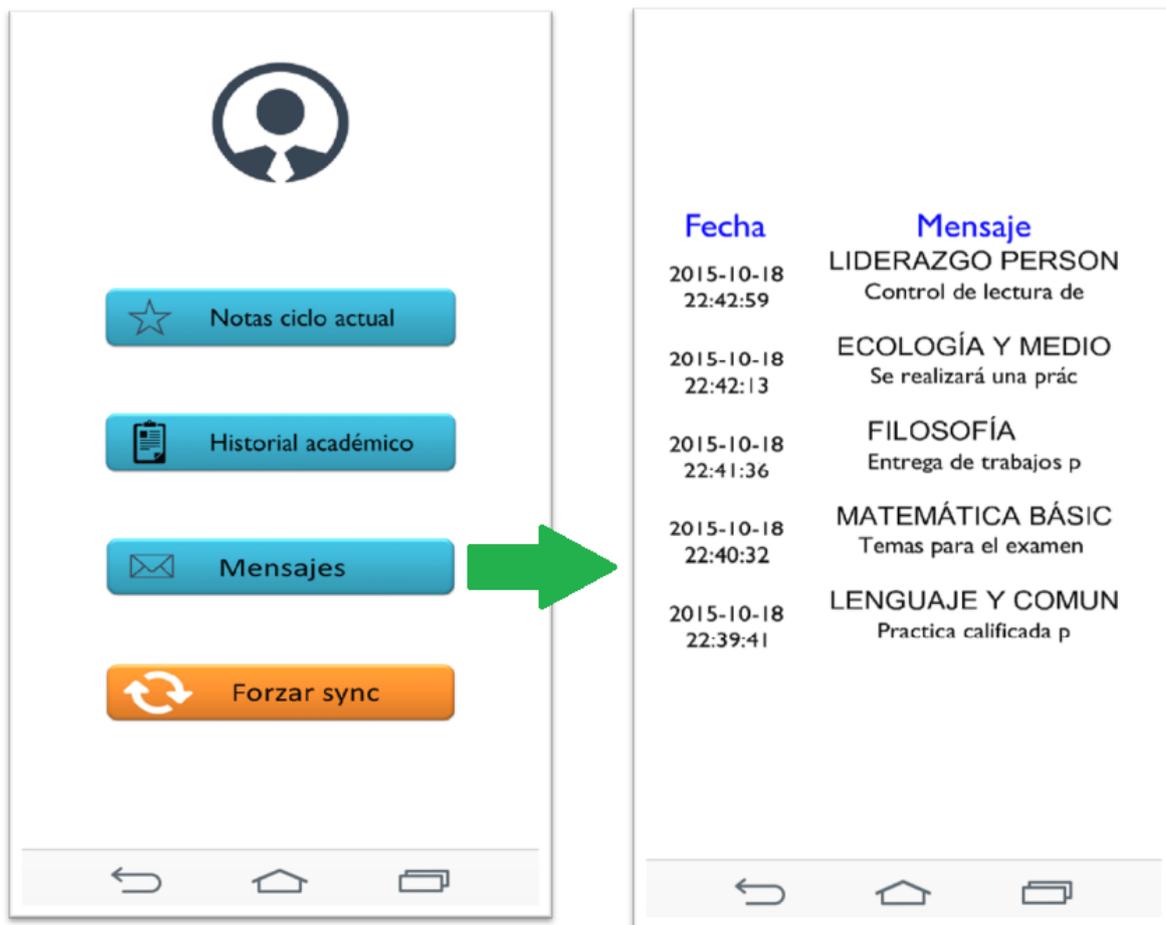


Figura 28 Lista de mensajes (Elaboración propia)

Caso de prueba 3

Objetivo: Demostrar el funcionamiento de GCM con el prototipo.

Acción	Resultado esperado
Actualizar las calificaciones de un estudiante desde la interfaz web	La aplicación recibe una notificación y las notas son actualizadas
Redactar un mensaje y enviarlo desde la interfaz web	La aplicación recibe una notificación y se agrega un nuevo mensaje

Tabla 5 Acciones Caso de prueba 3 (Elaboración propia)

Desde la interfaz web del docente se realizan los procedimientos para actualizar las notas de los estudiantes de un curso y la redacción de mensajes. Ambas acciones actualizan la base de datos del servidor web y emiten notificaciones hacia GCM el cual las reenviará a los

dispositivos de los estudiantes del curso que el docente dicta, permitiendo que la base de datos SQLite contenida en éstos también se actualice. Las figuras 29 y 30 representan a las acciones de registro de notas y redacción de mensajes respectivamente.

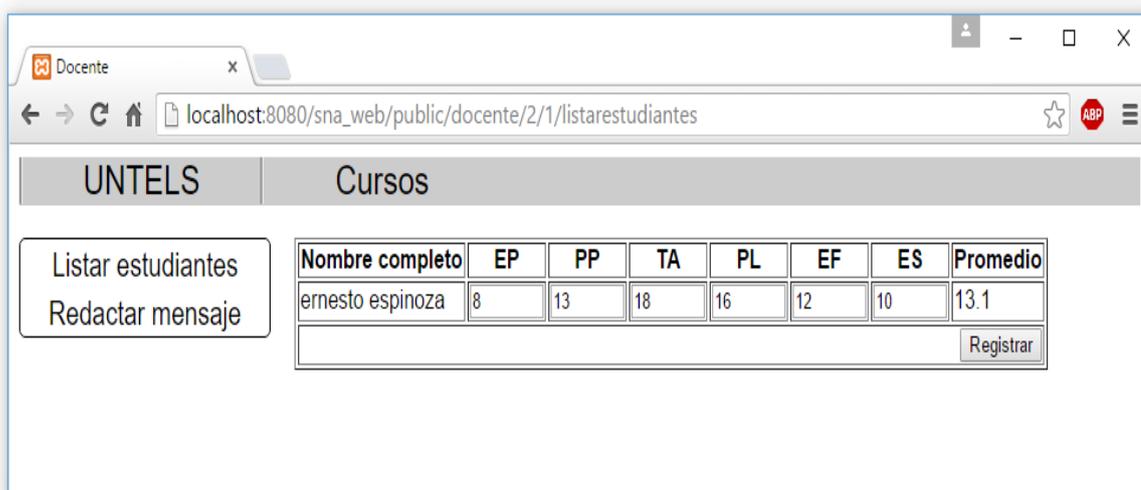


Figura 29 Interfaz web de registro de notas (Elaboración propia)



Figura 30 Interfaz web de redacción de mensajes (Elaboración propia)

Las notificaciones recibidas se acumulan a modo de cola como se puede ver en la figura 31, cada una de estas tiene una descripción que le indica la acción de la que proceden. También se ve como el icono de la aplicación muestra un contador que indica la cantidad de notificaciones que el usuario no ha revisado.

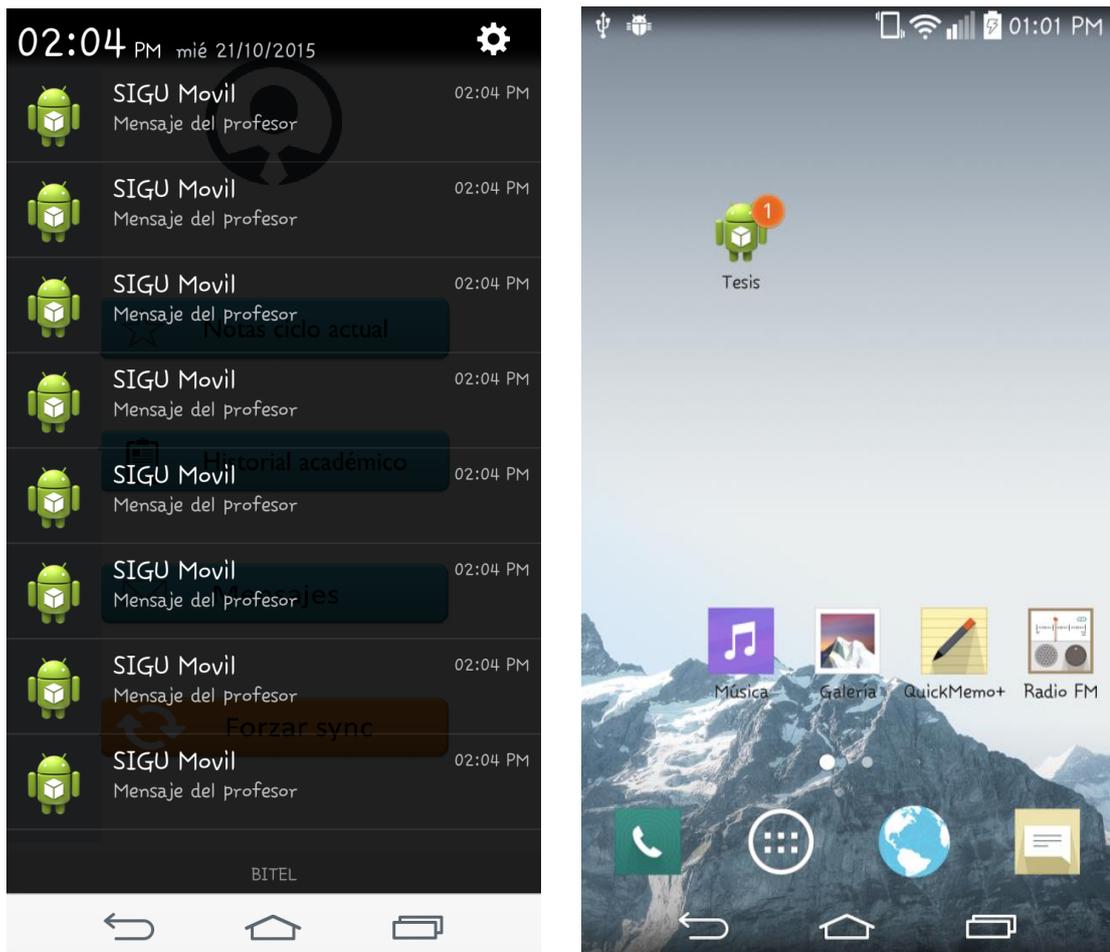


Figura 31 Cola de notificaciones GCM en el dispositivo (Elaboración propia)

Caso de prueba 4

Objetivo: comprobar los detalles finales del prototipo

Acción	Resultado esperado
Pulsar el botón “Descargar” en la interfaz de historial académico.	Se muestra un dialogo de confirmación e inicia la descarga del historial como PDF al aceptar el dialogo.
Presionar una fila de la lista de mensajes.	El detalle del mensaje aparece a modo de dialogo.
Presionar el botón “Forzar sync”.	Se superpone en pantalla la interfaz de sincronización y se extrae toda la información del usuario desde el servidor web.

Tabla 6 Acciones Caso de prueba 4 (Elaboración propia)

Si el usuario se dirige a la interfaz de “Historial académico” tendrá la opción de descargarlo en formato PDF. De aceptar la descarga, un archivo con el nombre “historial.pdf” aparecerá en la carpeta de la aplicación, la cual puede ser encontrada en el almacenamiento interno como “com.yakusoft.sigu”. Las figuras 32 y 33 muestran tanto el dialogo de descarga como el archivo generado.

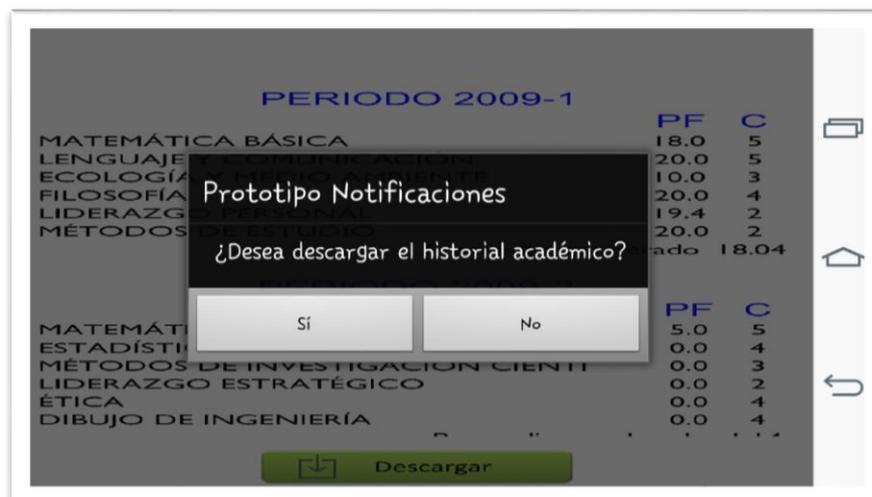


Figura 32 Descarga del historial académico del prototipo (Elaboración propia)

		UNIVERSIDAD NACIONAL TECNOLÓGICA DE LIMA SUR Ingeniería de Sistemas Historial académico	
CÓDIGO DE ESTUDIANTE: 2009200132 APELLIDOS Y NOMBRES: ernesto espinoza			
<u>SEMESTRE ACADÉMICO 2009-2</u>			
CICLO	CURSO	CREDITOS	NOTA
I	MATEMÁTICA BÁSICA	5	20.0
I	LENGUAJE Y COMUNICACIÓN	5	20.0
I	ECOLOGÍA Y MEDIO AMBIENTE	3	20.0
I	FILOSOFÍA	4	20.0
I	LIDERAZGO PERSONAL	2	20.0
I	MÉTODOS DE ESTUDIO	2	20.0
		21	20
Créditos Aprobados: 21 Créditos Desaprobados: 0 Cursos Aprobados: 6 Cursos Desaprobados: 0			
<u>SEMESTRE ACADÉMICO 2010-1</u>			
CICLO	CURSO	CREDITOS	NOTA
II	MATEMÁTICA I	5	20.0
II	ESTADÍSTICA DESCRIPTIVA	4	20.0
II	MÉTODOS DE INVESTIGACIÓN CIENTÍFICA	3	20.0
II	LIDERAZGO ESTRATÉGICO	2	20.0
II	ÉTICA	4	20.0
II	DIBUJO DE INGENIERÍA	4	20.0
		22	20
Créditos Aprobados: 22 Créditos Desaprobados: 0 Cursos Aprobados: 6 Cursos Desaprobados: 0			
<u>SEMESTRE ACADÉMICO 2010-2</u>			
CICLO	CURSO	CREDITOS	NOTA
III	MATEMÁTICA II	5	20.0
III	ESTADÍSTICA Y DISEÑO EXPERIMENTAL	4	20.0
III	QUÍMICA GENERAL	4	20.0
III	FÍSICA I	4	20.0
III	INTRODUCCIÓN A LA INGENIERÍA DE SISTEMAS	3	20.0
III	INFORMES TÉCNICOS	2	20.0
III	DESARROLLO DE LA CREATIVIDAD E INNOVACIÓN	2	20.0
		24	20
Créditos Aprobados: 24 Créditos Desaprobados: 0 Cursos Aprobados: 7 Cursos Desaprobados: 0			
<u>SEMESTRE ACADÉMICO 2011-1</u>			

Figura 33 PDF generado del historial (Elaboración propia)

La lista de mensajes permite ver solo un extracto de los mismos, si el usuario quiere ver el detalle del mensaje solo tiene que presionar sobre la fila donde se ubica y un dialogo con el texto completo aparecerá en pantalla tal como se ve en la figura 34.



Figura 34 Detalle de mensaje de un curso (Elaboración propia)

El último detalle que se agregó al prototipo fue le forzado de sincronización con el servidor. Este proceso extrae todos los datos relacionados con el historial académico del usuario desde el servidor, borrando toda la que se encuentra actualmente almacenada en la base de datos SQLite de la aplicación, como tal, esta acción puede demorar y se le consulta al usuario si desea continuar con la sincronización a través de un dialogo de confirmación.

Esta funcionalidad se desarrolló como contramedida cuando GCM deje de funcionar por lo que el usuario debe utilizarla prudentemente. La figura 35 presenta este proceso.

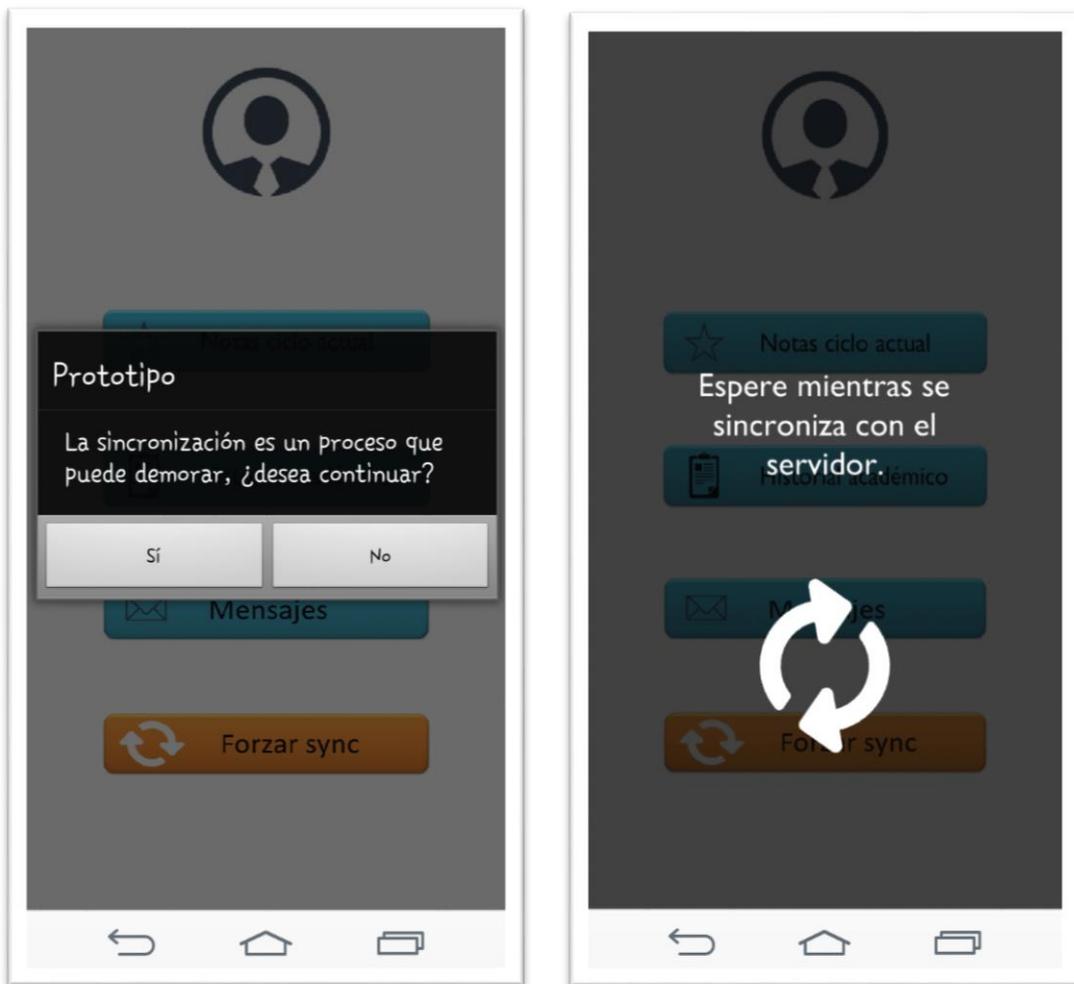


Figura 35 Forzado de sincronización (Elaboración propia)

CONCLUSIONES

- Con el presente trabajo se demuestra que las tecnologías móviles tienen un gran potencial de aplicación que puede ser usado con fines académicos.
- El SIGU es un sistema valorado por los estudiantes, sin embargo, ellos mismos reconocen la existencia de problemas en su funcionamiento.
- La finalidad del prototipo desarrollado no es reemplazar al actual SIGU sino complementar sus funciones, cubriendo las falencias encontradas.
- GCM dotó al prototipo de eficiencia y eficacia, pues por medio de sus notificaciones un usuario sabrá el momento oportuno cuando revisar sus registros.
- Con el prototipo no es el usuario quien acude al sistema, sino el sistema es quien acude al usuario.
- SQLite es un buen respaldo para almacenar la información del usuario, con esta no es necesario realizar transacciones hacia el servidor cada vez que el usuario requiera ver sus datos, si no se realizan transacciones a la base de datos del dispositivo.

RECOMENDACIONES

- Quien quiera realizar un trabajo similar deberá contemplar entre sus requerimientos el “UX design” (diseño de experiencia de usuario), característica que el prototipo presente carece.
- Animo a los interesados en desarrollar software a utilizar SCRUM como metodología, pues con cada iteración apreciaba como la funcionalidad del prototipo se incrementaba; en lo posible sigan sus lineamientos.
- Este trabajo fue hecho para demostrar las bondades de la tecnología móvil por lo que a pesar de ser funcional sigue siendo limitado, debe ser tratado como un ejemplo o base para trabajos similares futuros.

BIBLIOGRAFÍA

- Aguirre Chacón, L. T., & Sinche Ricra, H. J. (2013). *Diseño de una aplicación móvil para la consulta académica de la FIIS-UTP*. Obtenido de http://pis1.wikispaces.com/file/view/Presentacion+Final_Tesis+I.pdf
- Alegsa. (2010). *¿Qué son los prototipos de un sistema informático?* Obtenido de <http://www.alegsa.com.ar/Diccionario/C/7664.php>
- Alegsa. (2015). *Definición de smartphone*. Obtenido de <http://www.alegsa.com.ar/Dic/smartphone.php>
- Android Developers. (2015). *Android Studio*. Obtenido de <http://developer.android.com/intl/es/tools/studio/index.html>
- Android Developers. (2015). *System Architecture*. Obtenido de <http://developer.android.com/images/system-architecture.jpg>
- Ávila Mejía, O. (2012). *Android*. Obtenido de <http://www.izt.uam.mx/newpage/contactos/revista/83/pdfs/android.pdf>
- Blazek, M. (2011). *Google Android APK application package file format description*. Obtenido de <http://www.file-extensions.org/article/android-apk-file-format-description>
- Cancela Garcia, S., & Ostos Lobo, S. (2015). *Arquitectura Android*. Obtenido de <https://sites.google.com/site/swcuc3m/home/android/generalidades/2-2-arquitectura-de-android>
- Casillas Santillán, L. A., Gibert Ginesta, M., & Pérez Mora, Ó. (2008). *Bases de datos en MySQL*. Obtenido de http://ocw.uoc.edu/computer-science-technology-and-multimedia/bases-de-datos/bases-de-datos/P06_M2109_02151.pdf
- Caudevilla, G. (2010). *Metodología Ágil: SCRUM*. Obtenido de <http://developing.frogtek.org/wp-content/uploads/2010/07/Imagen-122.png>
- Caudevilla, G. (2010). *Waterfall vs SCRUM vs Kanban (II)*. Obtenido de <http://developing.frogtek.org/tag/daily-scrum/>
- Departamento de Informática de la Universidad de Valladolid. (2006). *El modelo Cliente/Servidor*. Obtenido de http://www.infor.uva.es/~fdiaz/sd/2005_06/doc/SD_TE02_20060305.pdf
- Gómez Matesanz, A. (2014). *Aplicación Android para la empresa Travelling-Service*. Obtenido de https://repositorio.uam.es/bitstream/handle/10486/662281/gomez_matesanz_alfonso_tfg.pdf?sequence=1

- González Mérida, I. (2012). *Desarrollo de una Aplicación Móvil: Caso Universiada 2012*. Obtenido de <http://cdigital.uv.mx/bitstream/123456789/32382/1/gonzalezmerida.pdf>
- Google Developers. (2015). *Android, the world's most popular mobile platform*. Obtenido de <http://developer.android.com/about/android.html>
- Google Developers. (2015). *Architectural Overview*. Obtenido de <https://developers.google.com/cloud-messaging/images/GCM-arch.png>
- Google Developers. (2015). *Cloud Messaging*. Obtenido de <https://developers.google.com/cloud-messaging/>
- Google Developers. (2015). *Google Developers Console*. Obtenido de <https://console.developers.google.com>
- MySQL. (2015). *About MySQL*. Obtenido de <https://www.mysql.com/about/>
- MySQL. (2015). *MySQL Workbench*. Obtenido de <https://www.mysql.com/products/workbench/>
- Open Handset Alliance. (2015). *Android*. Obtenido de http://www.openhandsetalliance.com/android_overview.html
- Perez Garcia, A. (2013). *Gamification: Encuestas de Satisfacción de los Servicios de Alimentación Javerianos*. Obtenido de <http://repository.javeriana.edu.co/bitstream/10554/12643/1/PerezGarciaAlejandro2013.pdf>
- Rodríguez, A. (2011). *La máquina virtual Dalvik*. Obtenido de <http://androideity.com/2011/07/07/la-maquina-virtual-dalvik/>
- Schwaber, K., & Sutherland, J. (2013). *La Guía de Scrum*. Obtenido de <http://www.scrumguides.org/docs/scrumguide/v1/Scrum-Guide-ES.pdf>
- Sinnathamby, M. (2012). *Stack based vs Register based Virtual Machine Architecture, and the Dalvik VM*. Obtenido de <https://markfaction.wordpress.com/2012/07/15/stack-based-vs-register-based-virtual-machine-architecture-and-the-dalvik-vm/>
- SourceForge. (2015). <http://sourceforge.net/projects/itext/>.
- SQLite Org. (2015). *About SQLite*. Obtenido de <https://www.sqlite.org/about.html>
- SQLite Studio. (2015). *SQLite Studio*. Obtenido de <http://sqlitestudio.pl/>
- Torres Huesca, M. A. (2011). *Propuesta de una aplicación (Web) móvil en la*. Obtenido de <http://core.ac.uk/download/pdf/16306177.pdf>

Universidad de El Salvador Facultad de Ingeniería y Arquitectura Escuela de Ingeniería de Sistemas Informáticos. (2014). *Introducción a SQLite*. Obtenido de http://aula.fia.ues.edu.sv/materialpublico/pdm115/anyos_anteriores/2014/PDM115_guia_lab05.pdf

Universidad de Lima. (2014). *App Ulima está lista para ser descargada*. Obtenido de <http://www.ulima.edu.pe/ulima/noticias/app-ulima-esta-lista-para-ser-descargada>

Universidad Peruana de Ciencias Aplicadas. (2014). *UPC Móvil*. Obtenido de www.upc.edu.pe/app-upc-movil/

ANEXOS

Historias de usuario

1		MOCKUPS	
Como cliente quiero ver un bosquejo inicial de las interfaces con las que el estudiante interactuará al usar la aplicación móvil.			
Estimación	4	Dependencia	-
Prioridad	1		

Pruebas de aceptación:

- Imágenes de cada interfaz a modo de mockup.

2		Base de datos servidor	
<p>Como cliente requiero que los datos de los estudiantes se encuentren almacenados en una base de datos principal</p>			
Estimación	8	Dependencia	-
Prioridad	1		

Pruebas de aceptación:

- Registrar los datos de un estudiante de Ingeniería de Sistemas durante 10 ciclos.
- Entrega del script que genere la base de datos.

3**Registro de notas**

Como cliente requiero una interfaz que permita a los docentes registrar calificaciones de los estudiantes inscritos en sus cursos.

Estimación	28	Dependencia	2
Prioridad	1		

Pruebas de aceptación:

- No dejar que se registre una nota que no esté entre 0 y 20.
- Al seleccionar un curso solo deben aparecer los estudiantes inscritos.
- No se debe permitir registrar letras.
- Si se ingresan los datos correctos las calificaciones deben aparecer actualizadas.

4 Enviar notificaciones a estudiantes			
<p>Como cliente quiero que cuando un profesor cambie las notas de un estudiante se le envíe una notificación a su Smartphone.</p>			
Estimación	8	Dependencia	3
Prioridad	2		

Pruebas de aceptación:

- En el smartphone del estudiante debe vibrar y mostrar una notificación después de que un profesor cualquiera actualice sus notas.

5 Base de datos Smartphone			
Como cliente necesito que la aplicación posea su propia fuente de datos interna que almacene todos los datos del estudiante que la use.			
Estimación	2	Dependencia	-
Prioridad	2		

Pruebas de aceptación:

- Registrar el historial académico de un estudiante.
- Registrar los mensajes del profesor.
- Entregar script de generación de base de datos.

6		Login de estudiante	
<p>Como cliente requiero que los estudiantes ingresen su código y contraseña para poder usar la aplicación con normalidad.</p>			
Estimación	13	Dependencia	5,3
Prioridad	2		

Pruebas de aceptación:

- Al ingresar código y clave incorrectos la aplicación debe permanecer en la interfaz de login.
- Al ingresar los datos correctos la aplicación sale del login y muestra el menú principal.

7**Ver notas del ciclo actual**

Como cliente quiero que los estudiantes puedan ver las notas del ciclo en que se encuentran actualmente matriculados.

Estimación	9	Dependencia	5,6
Prioridad	2		

Pruebas de aceptación:

- Cuando se presiona el botón de “ver notas del ciclo actual” aparece una interfaz con los cursos y sus calificaciones correspondientes a los rubros de prácticas, exámenes parciales, laboratorio, sustitutorio, promedio final.

8**Ver historial académico**

Como cliente requiero que los estudiantes que usen la aplicación puedan ver la totalidad de su record académico.

Estimación	9	Dependencia	5,6
Prioridad	2		

Pruebas de aceptación:

- Cuando se presiona el botón “historial académico” la aplicación debe mostrar una interfaz donde aparecen los cursos con sus promedios finales de todos los periodos que el estudiante haya estado matriculado.

9

Descargar historial

Como cliente quiero que un estudiante pueda descargar su historial académico en formato PDF.

Estimación	6	Dependencia	5,6,8
Prioridad	2		

Pruebas de aceptación:

- Cuando el estudiante presione el botón de “descargar historial” en la interfaz de historial académico debe mostrarse un mensaje de confirmación, de aceptar se inicia la descarga.

10

Mensajes del docente

Como cliente requiero que la interfaz del docente le permita redactar mensajes para sus estudiantes de un curso determinado y que una vez enviados puedan ser leídos por sus estudiantes.

Estimación	11	Dependencia	3
Prioridad	2		

Pruebas de aceptación:

- Al seleccionar redactar mensaje desde la interfaz docente se muestra un formulario donde se ingresa un texto. Después de enviarlo el estudiante debe recibir una notificación con el mensaje del docente.

11

Ver información personal

Como cliente quiero que los estudiantes puedan ver su información personal en una interfaz de la aplicación.

Estimación	2	Dependencia	5,6
Prioridad	3		

Pruebas de aceptación:

- Al presionar el icono de foto de estudiante se muestra una interfaz con sus datos personales.

12 Mostrar contador en icono

Como cliente quiero que encima del icono de la aplicación aparezca un contador que representa la cantidad de notificaciones no revisadas.

Estimación	1	Dependencia	4,10
Prioridad	3		

Pruebas de aceptación:

- Al actualizar notas y redactar mensajes el contador aumentara de uno en uno.
- Al abrirse la aplicación el contador regresa a 0 y ya no se muestra en el logo.

13**Forzar sincronización**

Como cliente quiero que se habilite la opción de sincronización forzada para que el estudiante pueda ver sus notas actualizadas si es que las notificaciones están fallando.

Estimación	3	Dependencia	6,3
Prioridad	3		

Pruebas de aceptación:

- Al presionar botón “forzar sync” debe aparecer un mensaje de confirmación que de aceptar inicia la sincronización.
- La acción de sincronizar debe borrar todos los datos almacenados hasta el momento en la aplicación.

Respuestas de la encuesta realizada

ID	HISTORIAS DE USUARIO	ESTIMACIÓN	PRIORIDAD	CÓDIGO	REQUERIMIENTOS	ESTIMACIÓN
1	Mockups	4	1	REQ-101	Desarrollo de mockups	4
2	Base de datos servidor	8	1	REQ-102	Modelado de la base de datos	4
				REQ-103	Creación de la base de datos	4
3	Registro de notas	28	1	REQ-104	Desarrollo de scripts PHP para el registro de notas	24
				REQ-105	Maquetado de la interfaz web	4
4	Enviar notificaciones a estudiantes	8	2	REQ-106	Conectar con GCM desde los scripts PHP	4
				REQ-107	Enviar mensajes a GCM después de una actualización de notas	2
				REQ-117	Generar una notificación local por cada notificación de GCM recibida	2
5	Base de datos Smartphone	2	2	REQ-108	Modelar y crear la base de datos SQLite para la aplicación móvil	2
6	Login de estudiante	13	2	REQ-109	Desarrollar las clases java que administren el login	3
				REQ-110	Sincronizar con el servidor automáticamente después del login	8
				REQ-111	Desarrollar la interfaz gráfica para el login del estudiante	2
7	Ver notas del ciclo actual	9	2	REQ-112	Desarrollar las clases java que se permitan ver las notas del ciclo actual.	5

				REQ-113	Desarrollar la interfaz gráfica para las notas del ciclo actual	4
8	Ver historial académico	9	2	REQ-114	Desarrollar las clases java que se encarguen del historial académico	5
				REQ-115	Desarrollar la interfaz gráfica para el historial académico	4
9	Descargar historial	6	2	REQ-116	Implementar la descarga del historial en formato PDF	6
10	Mensajes del docente	11	2	REQ-118	Habilitar la redacción y envío de mensajes del docente desde la interfaz web	3
				REQ-119	Enviar notificación a GCM después de redactar un mensaje	1
				REQ-120	Almacenar las mensajes entrantes en la BD SQLite	3
				REQ-121	Programar las clases java para mostrar los mensajes	2
				REQ-122	Diseñar la interfaz gráfica para el historial de mensajes	2
11	Ver información personal	2	3	REQ-123	Desarrollar las clases para visualizar la información personal	1
				REQ-124	Diseñar la interfaz de información personal	1
12	Mostrar contador en icono	1	3	REQ-125	Mostrar un contador de notificaciones en el ícono de la aplicación	1
13	Forzar sincronización	3	3	REQ-126	Agregar la opción de sincronización forzada	3

Los siguientes anexos fueron grabados a modo de archivos en un CD-ROM el cual está adjunto con el documento impreso del presente trabajo:

- Aplicación web PHP.
- Proyecto Android Java.
- Script para generar base de datos MySQL.
- Script para generar base de datos SQLite.
- Respuestas de la encuesta realizada.