



UNIVERSIDAD TECNOLÓGICA EQUINOCCIAL

**FACULTAD DE CIENCIAS DE LA INGENIERÍA
CARRERA DE INGENIERÍA INFORMÁTICA Y
CIENCIAS DE LA COMPUTACIÓN**

**DESARROLLO DE UNA APLICACIÓN MÓVIL EN ANDROID
QUE PERMITA AGENDAR TUTORÍAS ENTRE ESTUDIANTES
Y DOCENTES DE LA UTE**

**TRABAJO PREVIO A LA OBTENCIÓN DEL TÍTULO
DE INGENIERO EN INFORMÁTICA Y CIENCIAS DE LA COMPUTACIÓN**

JUAN ALFONSO GONZÁLEZ LÓPEZ

DIRECTOR: ING. VICTOR HUGO GÁLVEZ CAZA

Quito, Julio 2017

DERECHOS DE AUTOR

© Universidad Tecnológica Equinoccial. 2017
Reservados todos los derechos de reproducción

FORMULARIO DE REGISTRO BIBLIOGRÁFICO

PROYECTO DE TITULACIÓN

DATOS DE CONTACTO	
CÉDULA DE IDENTIDAD:	1718543935
APELLIDO Y NOMBRES:	GONZALEZ LOPEZ JUAN ALFONSO
DIRECCIÓN:	HUMBERTO ALBORNOZ OE5171 Y M. DE NIZA
EMAIL:	glja1024109@ute.edu.ec
TELÉFONO FIJO:	02 320 2859
TELÉFONO MOVIL:	09 9733 2639

DATOS DE LA OBRA	
TÍTULO:	Desarrollo de una Aplicación móvil en Android que permita agendar tutorías entre estudiantes y docentes de la UTE.
AUTOR O AUTORES:	JUAN GONZALEZ
FECHA DE ENTREGA DEL PROYECTO DE TITULACIÓN:	14 DE JULIO 2017
DIRECTOR DEL PROYECTO DE TITULACIÓN:	ING. VICTOR GALVEZ
PROGRAMA	PREGRADO <input checked="" type="checkbox"/> POSGRADO <input type="checkbox"/>
TÍTULO POR EL QUE OPTA:	INGENIERO EN INFORMATICA Y CIENCIAS DE LA COMPUTACIÓN
RESUMEN:	Se desarrolló una aplicación móvil para sistemas operativos Android como un prototipo funcional que permita agendar citas de tutorías académicas entre docentes y estudiantes para apoyar al proceso de aprendizaje de los estudiantes. La Universidad Tecnológica Equinoccial ofrece tutorías a los alumnos que las requieran, sin embargo, entre las dificultades para acceder a ellas se encuentran: la falta de organización de tiempo y la comunicación entre docentes y estudiantes.

	<p>La aplicación presentada en este trabajo de investigación se desarrolló mediante metodología basada en prototipos y funciona de la siguiente manera: Se conecta a una base de datos desarrollada en MySQL mediante un webservice SOAP programado en Java que corre en un servidor Apache Tomcat; Estos dos componentes actúan como servidor de la aplicación.</p> <p>Como requisito para el desarrollo de la aplicación en Android se presentan dos interfaces gráficas; una para alumnos y otra para docentes a las que se designaron respectivamente como módulo de alumnos y módulo de docentes.</p> <p>En el módulo de alumnos se tiene la posibilidad de llenar encuestas preliminares que ayudarán a los docente a hacer una valoración de la situación actual del estudiante, visualizar información personal y concretamente poder agendar citas con los docentes tutores por cada materia en los horarios asignados a inicio del periodo lectivo, pudiendo también calificar su nivel de satisfacción con sesión de tutoría a la que asistió.</p> <p>Por otra parte el módulo de los docentes permite visualizar todas las citas que han sido agendadas por los estudiantes en sus materias, pudiendo aceptarlas, cancelarlas, reprogramarlas en caso de que no le sea posible atender la cita en el horario seleccionado o transferirlas a departamentos de apoyo según amerite el caso, además de visualizar información relevante del alumno que agenda una cita y generar reportes como apoyo de su gestión en base a las tutorías académicas.</p>
<p>PALABRAS CLAVES:</p>	<p>aplicación móvil; agendar tutorías; webservice SOAP; Java; Android</p>

ABSTRACT:

A mobile application for Android operating systems was developed as a functional prototype that allows scheduling appointments of academic tutorials between teachers and students to contribute to the learning process of students. The "Universidad Tecnológica Equinoccial" offers tutoring to students who need them, however, there are difficulties to access them, like the lack of time organization and communication between teachers and students.

The application described in this research was developed using prototype-based methodology and works as follows: Connects to a database developed in MySQL using a webservice SOAP programmed in Java that runs on an Apache Tomcat server; These two components works as the application server.


As a requirement for the development of the application in Android, two graphical interfaces are presented; One for students and another for teachers, respectively designated as a module for students and a module for teachers.

In the students module you have the possibility to fill preliminary surveys that will help the teachers to make an assessment of the current situation of the student, to visualize personal information and specifically to be able to schedule appointments with the tutors for each subject in the schedules assigned to start of the academic period, being able also to qualify its level of satisfaction with tutoring session to which it attended.

The teachers module allows you to view all the appointments that have been scheduled by the students in their subjects, being able to

	accept them, cancel them, reprogram them in case it is not possible to attend the appointment at the selected timetable or transfer them to support departments according to each case, besides visualizing relevant information of the student who schedules an appointment and generate reports as support of its management based on the academic tutorials.
KEYWORDS	mobile application (app); tutorial class schedule; webservice SOAP; JAVA; Android

Se autoriza la publicación de este Proyecto de Titulación en el Repositorio Digital de la Institución.


f: 
GONZALEZ LOPEZ JUAN ALFONSO
C.I. 1718543935

DECLARACIÓN Y AUTORIZACIÓN

Yo, **GONZALEZ LOPEZ JUAN ALFOSNO**, CI: 1718543935 autor del proyecto titulado: **DESARROLLO DE UNA APLICACIÓN MÓVIL EN ANDROID QUE PERMITA AGENDAR TUTORÍAS ENTRE ESTUDIANTES Y DOCENTES DE LA UTE** previo a la obtención del título de **INGENIERO EN INFORMÁTICA Y CIENCIAS DE LA COMPUTACIÓN** en la Universidad Tecnológica Equinoccial.

1. Declaro tener pleno conocimiento de la obligación que tienen las Instituciones de Educación Superior, 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 BIBLIOTECA de la Universidad Tecnológica Equinoccial a tener una copia del referido trabajo de graduación con el propósito de generar un Repositorio que democratice la información, respetando las políticas de propiedad intelectual vigentes.


Quito, 14 de Julio del 2017

f: 
GONZALEZ LOPEZ JUAN ALFONSO
C.I. 1718543935

DECLARACIÓN

Yo **GONZALEZ LOPEZ JUAN ALFONSO**, declaro que el trabajo aquí descrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

La Universidad Tecnológica Equinoccial puede hacer uso de los derechos correspondientes a este trabajo, según lo establecido por la Ley de Propiedad Intelectual, por su Reglamento y por la normativa institucional vigente.

f. 

GONZALEZ LOPEZ JUAN ALFONSO

C.I. 1718543935

CERTIFICACIÓN

Certifico que el presente trabajo que lleva por título “**Desarrollo de una Aplicación móvil en Android que permita agendar tutorías entre estudiantes y docentes de la UTE**”, que, para aspirar al título de **Ingeniero en Informática y Ciencias de la Computación** fue desarrollado por **Juan Alfonso González López**, bajo mi dirección y supervisión, en la Facultad de Ciencias de la Ingeniería; y cumple con las condiciones requeridas por el reglamento de Trabajos de Titulación artículos 19, 27 y 28.



Ing. Víctor Hugo Gálvez Caza

DIRECTOR DEL TRABAJO

C.I. 1707796536

ÍNDICE DE CONTENIDOS

	Página
RESUMEN.....	v
ABSTRACT	vi
1. INTRODUCCIÓN.....	1
2. MARCO TEÓRICO	3
2.1 ESTADO DEL ARTE	3
2.1.1 GOOGLE CALENDAR.....	4
2.1.1.1 Ventajas	4
2.1.1.2 Desventajas	5
2.1.2 CALENDARIO BUSINESS AGENDA	5
2.1.2.1 Ventajas	6
2.1.2.2 Desventajas	6
2.1.3 CAL: ANY.DO CALENDARIO	6
2.1.3.1 Ventajas	7
2.1.3.2 Desventajas	7
2.2 CONCEPTOS BÁSICOS.....	7
2.2.1 PROTOCOLOS DE COMUNICACIÓN	7
2.2.1.1 Protocolo TCP/IP.....	7
2.2.1.2 HTTP (Hypertext Transfer Protocol).....	8
2.2.1.3 Webservice SOAP.....	9
2.2.2 BASE DE DATOS DE LA APLICACIÓN	10
2.2.2.1 MySQL Server.....	10
2.2.3 SERVIDOR DE APLICACIÓN	10
2.2.3.1 Apache TOMCAT	10
2.2.4 ENTORNOS DE DESARROLLO DE LA APLICACIÓN	11
2.2.4.1 NetBeans IDE.....	11
2.2.4.2 Android Studio.....	11
2.2.5 TUTORÍA ACADÉMICA	112
3. METODOLOGÍA.....	13

3.1 METODOLOGÍA DE DESARROLLO BASADA EN PROTOTIPOS ...	14
3.1.1 COMUNICACIÓN	14
3.1.2 PLANEACIÓN Y MODELADO	14
3.1.3 CONSTRUCCIÓN DEL PROTOTIPO.....	14
3.1.4 DESPLIEGUE	15
4. RESULTADOS Y DISCUSIÓN.....	16
4.1 RESULTADOS	16
4.1.1 COMUNICACIÓN	16
4.1.1.1 Consideraciones sobre los procesos.....	16
4.1.1.2 Proceso para agendar tutorías basada en la recolección de requisitos:.....	17
4.1.1.3 Requisitos funcionales	17
4.1.1.4 Requisitos no funcionales	18
4.1.2 PLANEACIÓN Y MODELADO	19
4.1.3 CONSTRUCCIÓN DEL PROTOTIPO.....	21
4.1.3.1 Base de Datos.....	21
4.1.3.2 Webservice	22
4.1.3.3 Aplicación Android.....	24
4.1.4 DESPLIEGUE	26
4.1.4.1 Entrega del Prototipo.....	26
4.1.4.2 Pruebas y Retroalimentación	32
4.2 DISCUSIÓN	33
5. CONCLUSIONES Y RECOMENDACIONES	34
5.1 CONCLUSIONES.....	34
5.2 RECOMENDACIONES	35
BIBLIOGRAFÍA.....	36

ÍNDICE DE TABLAS

	Página
Tabla 1. Requisitos funcionales.....	18
Tabla 2. Requisitos no funcionales.....	19
Tabla 3. Pruebas realizadas al software.....	32

ÍNDICE DE FIGURAS

	Página
Figura 1. Google Calendar.....	4
Figura 2. Calendario Business Agenda.....	5
Figura 3. Cal: Any.do Calendario.	6
Figura 4. Flujo de la Información a través de las capas del protocolo TCP/IP	8
Figura 5. Flujo del Proceso Evolutivo del Desarrollo de Software.....	13
Figura 6. Diagrama de Actividades a realizar en el primer prototipo.....	17
Figura 7. Storyboard del módulo de Estudiante	20
Figura 8. Modelo relacional de la Base de Datos.....	22
Figura 9. Metodos implementados en webservice SOAP mostrados por la herramienta de pruebas “SoapUI“	23
Figura 10. Envío de peticiones y muestra de resultados del webservice en SoapUI.....	24
Figura 11. Estructura de la aplicación dentro del IDE Android Studio.....	25
Figura 12. Diagrama de Secuencia: Peticiones desde la Aplicación	26
Figura 13. Prototipo Funcional. “Splash screen” y pantalla de inicio de sesión	27
Figura 14. Prototipo Funcional – Módulo Estudiantes.....	28
Figura 15. Prototipo Funcional – Módulo Docentes. Pantalla de Inicio y Citas	30
Figura 16. Prototipo Funcional – Módulo Docentes. Pantallas de Informes.	31

RESUMEN

Se desarrolló una aplicación móvil para sistemas operativos Android como un prototipo funcional que permita agendar citas de tutorías académicas entre docentes y estudiantes para aportar al proceso de aprendizaje de los estudiantes. La Universidad Tecnológica Equinoccial ofrece tutorías a los alumnos que las requieran, sin embargo, entre las dificultades para acceder a ellas se encuentran la falta de organización de tiempo y la comunicación entre docentes y estudiantes.

La aplicación presentada en este trabajo de investigación se desarrolló mediante una metodología basada en prototipos y funciona de la siguiente manera: Se conecta a una base de datos desarrollada en MySQL mediante un webservice SOAP programado en Java que corre en un servidor Apache Tomcat; Estos dos componentes actúan como servidor de la aplicación.

Como requisito para el desarrollo de la aplicación en Android se presentan dos interfaces gráficas; una para alumnos y otra para docentes a las que se designaron respectivamente como módulo de alumnos y módulo de docentes.

En el módulo de alumnos se tiene la posibilidad de llenar encuestas preliminares que ayudarán a los docente a hacer una valoración de la situación actual del estudiante, visualizar información personal y concretamente poder agendar citas con los docentes tutores por cada materia en los horarios asignados a inicio del periodo lectivo, pudiendo también calificar su nivel de satisfacción con sesión de tutoría a la que asistió.

Por otra parte el módulo de los docentes permite visualizar todas las citas que han sido agendadas por los estudiantes en sus materias, pudiendo aceptarlas, cancelarlas, reprogramarlas en caso de que no le sea posible atender la cita en el horario seleccionado o transferirlas a departamentos de apoyo según amerite el caso, además de visualizar información relevante del alumno que agenda una cita y generar reportes como apoyo de su gestión en base a las tutorías académicas.

Palabras clave: Aplicación móvil; Android; Tutorías; Webservice; Agendar; Java.

ABSTRACT

A mobile application for Android operating systems was developed as a functional prototype that allows scheduling appointments of academic tutorials between teachers and students to contribute to the learning process of students. The “Universidad Tecnológica Equinoccial” offers tutoring to students who need them, however, there are difficulties to access them, like the lack of time organization and communication between teachers and students.

The application described in this research was developed using prototype-based methodology and works as follows: Connects to a database developed in MySQL using a webservice SOAP programmed in Java that runs on an Apache Tomcat server; These two components works as the application server.

As a requirement for the development of the application in Android, two graphical interfaces are presented; One for students and another for teachers, respectively designated as a module for students and a module for teachers.

In the students module you have the possibility to fill preliminary surveys that will help the teachers to make an assessment of the current situation of the student, to visualize personal information and specifically to be able to schedule appointments with the tutors for each subject in the schedules assigned to start of the academic period, being able also to qualify its level of satisfaction with tutoring session to which it attended.

The teachers module allows you to view all the appointments that have been scheduled by the students in their subjects, being able to accept them, cancel them, reprogram them in case it is not possible to attend the appointment at the selected timetable or transfer them to support departments according to each case, besides visualizing relevant information of the student who schedules an appointment and generate reports as support of its management based on the academic tutorials.

Keywords: Android; mobile application; Tutorials; Webservice; Schedule; Java.

1. INTRODUCCIÓN

Las tutorías académicas son un apoyo al proceso formativo dentro de la Universidad y se vuelven sumamente importantes cuando el estudiante atraviesa por dificultades en el proceso de aprendizaje. La Universidad Tecnológica Equinoccial “UTE” cuenta con un departamento dedicado para esta labor en cada una de sus facultades el cual estaba deslindado del aspecto tecnológico que podría mejorar la efectividad del mismo y en el último año se implementó un sistema de tutorías dentro de la plataforma informática universitaria. Sin embargo, una aplicación móvil provee de grandes ventajas cuando se trata de comunicar o coordinar, entre ellas se destaca su facilidad de uso y accesibilidad, además de que la conectividad a internet inalámbrico se ha extendido dentro de la comunidad universitaria ya sea mediante WiFi o con uso del paquete de datos móviles, disponibilidad de tiempo para realizar estas tareas se reduce gracias a la portabilidad de los smartphones, es decir, que se lo puede realizar en cualquier lugar donde se disponga de conectividad a internet.

El objetivo general del proyecto es desarrollar una aplicación móvil en Android que podría implementarse en la universidad para agendar tutorías y así facilitar la comunicación durante este proceso y que sea de utilidad para docentes y estudiantes.

Los objetivos específicos son: a) Documentar los actuales procesos que se requieren para agendar tutorías. b) Automatizar los procesos documentados y aplicarlos en el desarrollo de la aplicación móvil. c) Lograr una comunicación óptima entre la aplicación cliente de docentes y estudiantes con el servidor.

Para cumplir con la realización la aplicación se utilizó la metodología de desarrollo ágil basada en prototipos.

La fase de desarrollo inició con el desarrollo de un prototipo por medio de un storyboard que muestra el funcionamiento inicial de la aplicación y se basa en los requisitos preliminares. Partiendo de ahí, se continuó con la creación de la base de datos que se diseñó con el modelo entidad-relación en MySQL

Server mediante la herramienta MySQL Workbench, se desarrolló el Webservice SOAP (Simple Object Access Protocol) en lenguaje Java mediante el IDE (Integrated Development Environment) NetBeans que se ejecuta en un servidor Apache Tomcat, mismo que comunica la aplicación con la base de datos.

La aplicación móvil como tal, se desarrolló en Android Studio que es el IDE oficial de Google para la creación de aplicaciones móviles para Android.

El trabajo realizado muestra cómo se puede mejorar un proceso mediante el desarrollo tecnológico de la universidad y apoyar al mejor desempeño de las actividades académicas con el uso de una aplicación móvil.

2. MARCO TEÓRICO

Para sustentar el desarrollo de una aplicación móvil se hace necesario conocer aspectos básicos sobre la plataforma de ejecución, así como aplicaciones similares. En el presente capítulo, se recogen conceptos generales de la plataforma Android, investigaciones relacionadas con el tema de desarrollo y se presentan ampliaciones similares como referencia.

Android es un sistema operativo para smartphones, tablets y también disponible en muchos dispositivos orientados a la web como smart TV, smart watch, autos, entre otros y está basado en un kernel Linux. Android nació como un proyecto de código abierto AOSP (Android Open Source Project), que posteriormente fue adquirido por la multinacional informática Google, la cual lidera su desarrollo y distribución. (Darwin, 2017)

Según la empresa consultora de investigación y análisis de datos “Strategy Analytics”, más del 88% del mercado mundial de smartphones está ocupado por Android a finales del 2016 (Strategy Analytics, 2016), manteniéndose como el Sistema Operativo para dispositivos móviles más usado del mundo, motivo por el cual la investigación se basa en esta plataforma de desarrollo.

2.1 ESTADO DEL ARTE

Existen varias aplicaciones de calendario en la tienda de aplicaciones de Android que permiten a los usuarios agendar citas y tener control de su calendario de actividades, sin embargo en el caso de las tutorías se necesita que la aplicación provea la comunicación adecuada entre docentes y alumnos además de presentar la información académica necesaria para poder coordinar y planificar las tutorías. A continuación se presentan algunas de las aplicaciones de agenda-calendario más relevantes y sus características importantes.

2.1.1 GOOGLE CALENDAR

Google Calendar es una de las aplicaciones de uso más extendido, principalmente porque lleva el nombre "Google", esta app se sincroniza con nuestra cuenta principal de google y nos permite crear, editar y eliminar eventos, con la posibilidad de notificar y compartir recordatorio vía email.

La Figura 1 muestra la organización de la agenda personal en día, con las fechas importantes que se hayan marcado o que estén predeterminadamente marcadas como importantes, al tocar la fecha importante nos mostrará una descripción de la fecha y si está el día sin marcar, creará un evento para nosotros, permitiéndonos ingresar nombre y horarios del evento, con la posibilidad de agregar ubicación, notificaciones, e invitar a más personas. (Google, 2017)

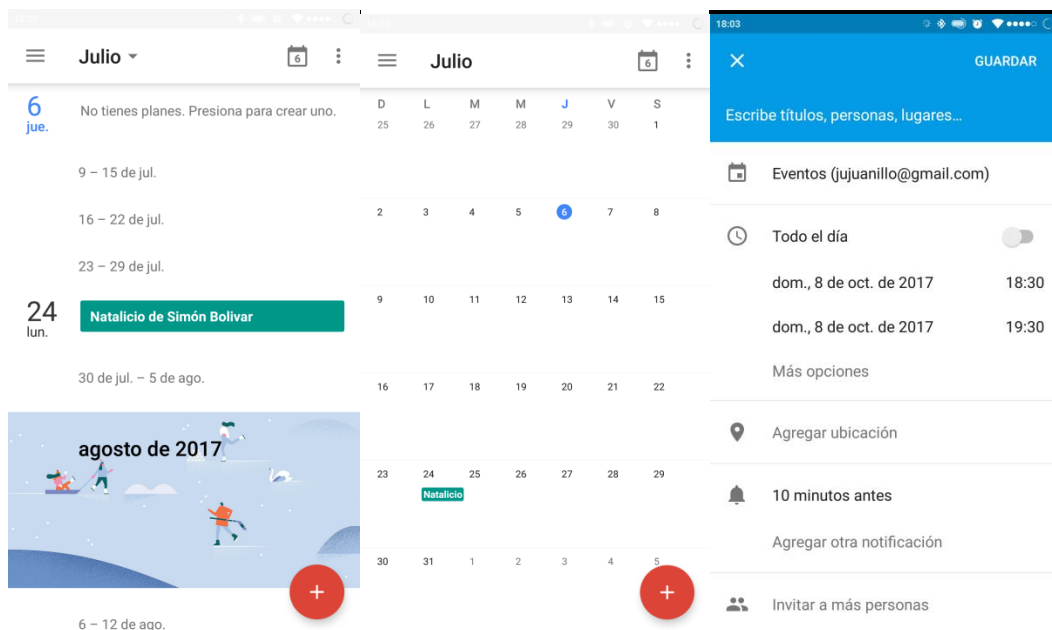


Figura 1. Google Calendar.

2.1.1.1 Ventajas

Venir de fábrica en dispositivos con Android stock (es decir un dispositivo que no tiene modificaciones ni capas de personalización en su sistema operativo Android), contar con la marca Google en su nombre la hace una aplicación confiable y atractiva para el usuario, además se sincroniza por defecto con la información que tenemos almacenada en la cuenta principal de Google del dispositivo.

2.1.1.2 Desventajas

No se puede personalizar su interfaz y muestra días importantes dependiendo de la ubicación, esto podría ser un problema para quien desee ser notificado de fechas importantes en otra nación diferente a la que reside.

2.1.2 CALENDARIO BUSINESS AGENDA

Esta aplicación de calendario desarrollada por Appgenix Software según la tienda google (Play Store), se sincroniza con los contactos del dispositivo y con la cuenta de google del mismo para mostrarnos fechas importantes que se encuentren en estos apartados, además de permitir la creación de eventos y gestionar tareas, mostrando recordatorios si así el usuario lo requiere. (Appgenix, 2017)

En la Figura 2, se puede observar una interfaz de calendario similar a Google calendar para mostrar todos los días de un mes, así como para la agendación de un evento en el calendario.

Sin embargo lo que más llama la atención de esta app, es estabilidad, navegación intuitiva, notificaciones personalizables, entre otras características que la hacen una de las favoritas de los usuarios según la Play Store de Android, otorgándole una calificación de 4.5/5 estrellas.

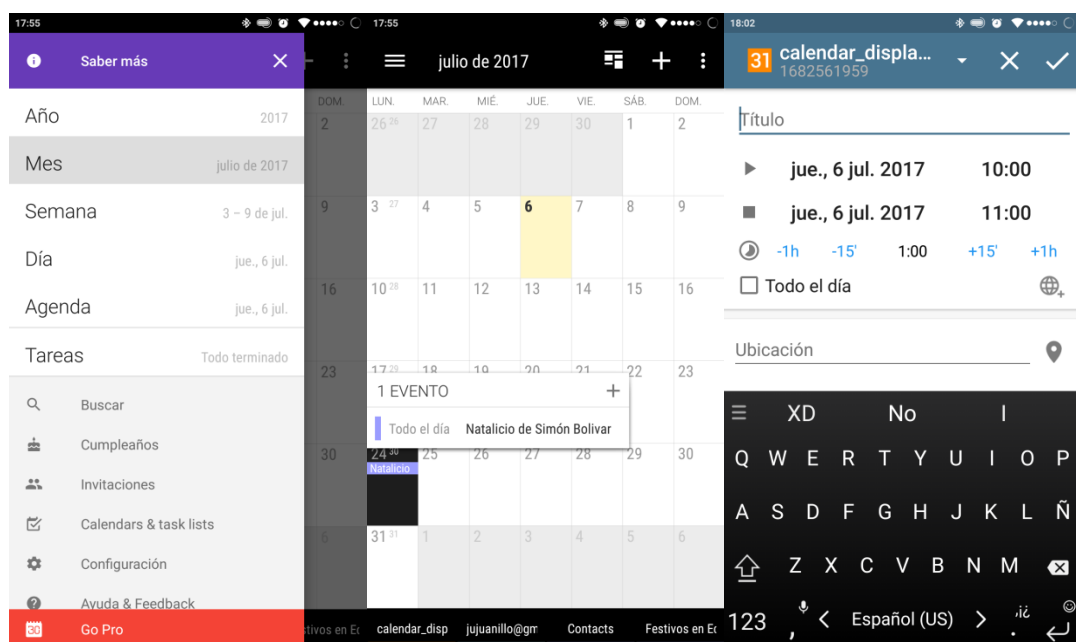


Figura 2. Calendario Business Agenda.

2.1.2.1 Ventajas

De interfaz sencilla, no posee anuncios indeseables y su combinación de colores la hacen muy agradable y fácil de usar.

2.1.2.2 Desventajas

No permite compartir eventos entre los usuarios de la aplicación, y no se sincroniza con otros clientes de correo que no sean Gmail o Exchange.

2.1.3 CAL: ANY.DO CALENDARIO

Aplicación desarrollada por Any.do y disponible en la tienda Play Store de Android, esta aplicación de calendario posee una interfaz sencilla con opciones de integración con Facebook, lo que la hace bastante atractiva ya que puede sincronizarse con días importantes no solo relacionados con el dispositivo donde se instala sino también de los amigos en Facebook.

Como se puede observar en la Figura 3, Cal: Any.do presenta una interfaz algo diferente a las vistas anteriormente, es mucho más sencilla sin muchas opciones y con colores y diseños más artísticos, pero bastante efectiva para su propósito de mantener control y organización de las fechas importantes y eventos. (Any.do, 2017)

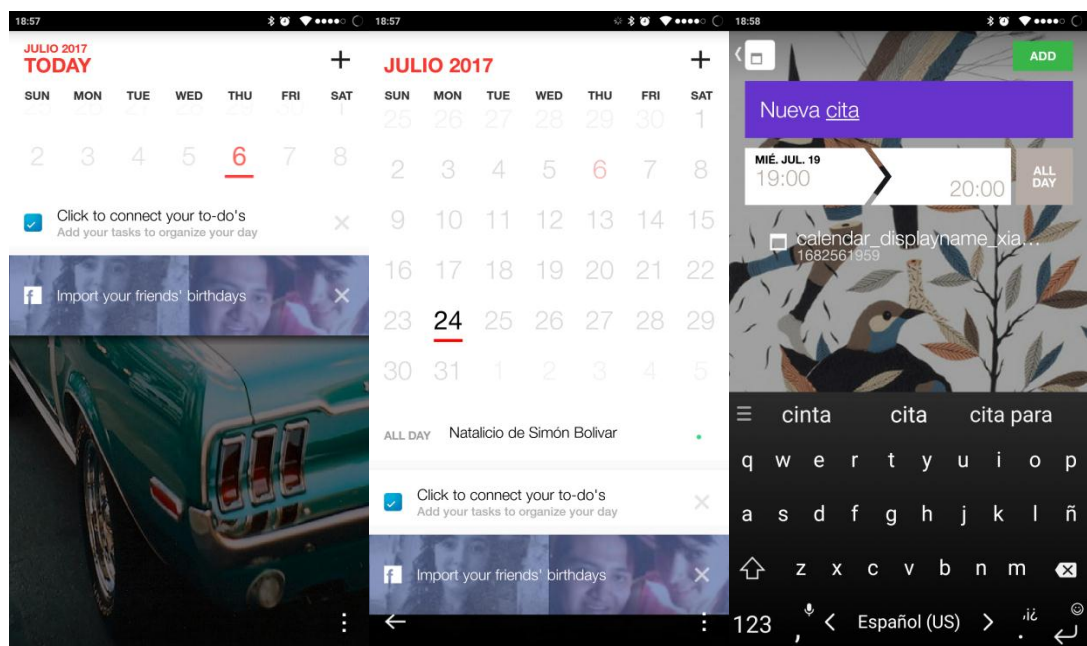


Figura 3. Cal: Any.do Calendario.

2.1.3.1 Ventajas

Su principal ventaja es poder sincronizarse con los contactos y eventos de Facebook, que es una de las más redes sociales más importantes del mundo, es así que puede contener mucha información respecto de los intereses, fechas y eventos del usuario y de sus amigos. Además también se sincroniza automáticamente con la cuenta de Google y los contactos de la misma. Otra ventaja reside en su interfaz sencilla y eficaz.

2.1.3.2 Desventajas

Dada su sencillez carece de opciones como compartir eventos o crear alarmas personalizables y también su fortaleza, que reside en conectarse con Facebook puede suponer una desventaja al no poder filtrar el contenido que recibe de los contactos y eso supondría bastantes fechas y alarmas indeseables en el calendario.

2.2 CONCEPTOS BÁSICOS

2.2.1 PROTOCOLOS DE COMUNICACIÓN

Se definen como reglas y normas que en conjunto permiten la transmisión de señales o datos entre equipos, facilitando su comunicación. Los protocolos de comunicación tienen especificaciones tanto en hardware como en software, y operan por canales definidos para cada especificación. (Rodríguez, 2005)

2.2.1.1 Protocolo TCP/IP

En realidad el TCP/IP (Transfer Control Protocol / Internet Protocol) es un conjunto de protocolos de comunicación de redes, es decir que los protocolos realizan transferencias multidireccionales de mensajes entre dispositivos y sistemas, el mismo que posee una arquitectura con 5 capas como muestra la Figura 4 que son: la capa red física que es el hardware que utiliza la red, la capa de interfaz de red que vienen a ser los protocolos de

enlace de datos, la capa de internet que se identifica por los protocolos representativos IPv4 e IPv6, la capa de transporte que asegura que los paquetes lleguen de manera ordenada minimizando errores y la de aplicación donde se definen las aplicaciones y servicios de red que utilizará el usuario. (Comer, 1996)

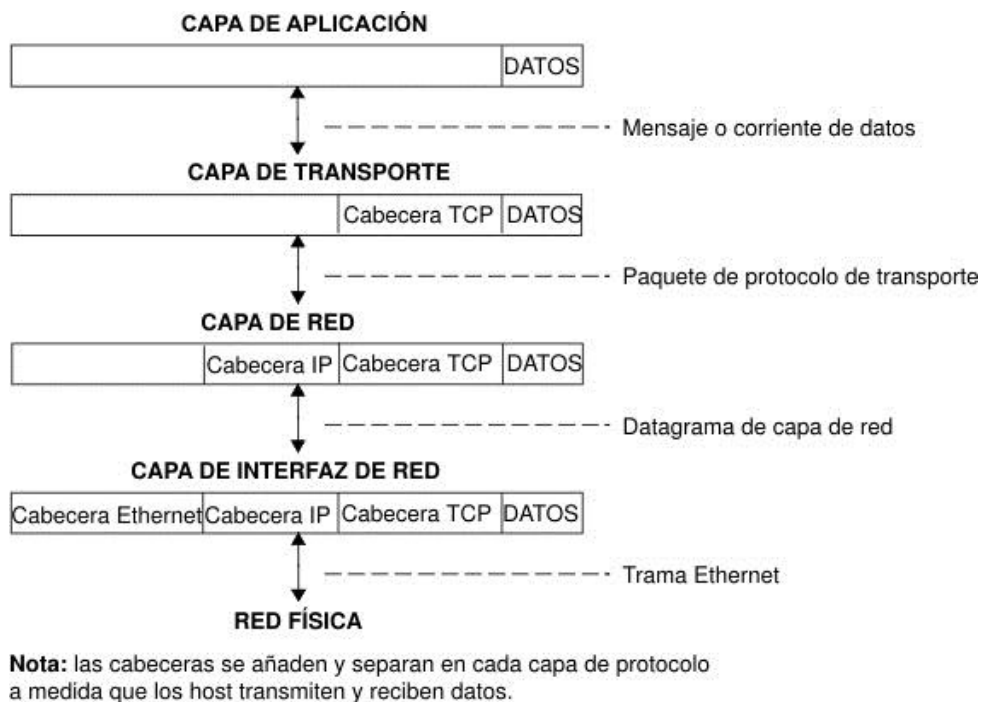


Figura 4. Flujo de la Información a través de las capas del protocolo TCP/IP. (IBM, 2017)

2.2.1.2 HTTP (Hypertext Transfer Protocol)

HTTP es un protocolo genérico a nivel de la capa de aplicación que funciona a base de peticiones y respuestas entre terminales a manera de cliente-servidor. Mediante HTTP el cliente envía una petición al servidor y este devuelve una respuesta al cliente, dada su simpleza y facilidad de uso se convirtió desde su creación en prácticamente un estándar de la red mundial, el internet. (Berners-Lee, 1996)

2.2.1.3 Webservice SOAP

Cabe destacar que un webservice como tal, es un conjunto de protocolos que configuran una interfaz accesible desde la red para el intercambio de datos entre aplicaciones, independientemente del sistema informático que usen. (Tidwell , Snell, & Kulchenko , 2001)

Y un Webservice SOAP (Simple Object Access Protocol) es un protocolo de acceso a objetos y para entenderlo es necesario saber que SOAP es una aplicación de la especificación XML (eXtensible Markup Language) que es una meta-lenguaje estandarizada y ampliamente usada en internet por su sencillez e independencia de plataformas, lenguajes de programación y aplicaciones. Es por eso que puede ser usado en cualquier entorno y ser intercambiado por cualquier sistema sin necesidad de ser traducido a otro lenguaje.

Es así que SOAP envía mensajes estructurados de la siguiente manera:

- Envelope, que contiene la información que se necesita para el procesamiento del mensaje y a todos los elementos “hijos” del mensaje.
- Header, que tiene el propósito de brindar información acerca del mensaje y que contiene a su vez bloques que pueden contener la identificación del mensaje, credenciales de autenticación o información del ruteo del mensaje.
- Body, que contiene el mensaje, es decir peticiones, información, respuestas o contenido útil del mensaje SOAP.

Bajo esta estructura de mensajes estandarizada y una dirección WSDL (Webservices Description Language) la cual describe los métodos implementados en el webservice de manera estructurada en XML, es posible para SOAP ejecutar los métodos remotos enviando las peticiones sobre HTTP, a estos se los conoce como “SOAP Services”.

2.2.2 BASE DE DATOS DE LA APLICACIÓN

2.2.2.1 MySQL Server

Es un Sistema de Gestión de Bases de Datos licenciada por la Corporación Oracle bajo GPL (General Public License) para uso libre de ciertas versiones y con una licencia comercial para versiones corporativas.

Como su nombre lo indica MySQL server se basa en lenguaje SQL (Structured Query Language) que es un estándar como lenguaje para bases de datos relacionales, sin embargo tiene sus propias características del lenguaje. Aunque se basa en consultas, puede llegar a ser bastante complejo al momento de tratar y manipular datos lo que lo hace muy veloz, multi-hilos, multi-usuarios y muy robusto. (Oracle, 2017)

Existe una herramienta llamada MySQL Workbench que presenta una interfaz gráfica amigable y sencilla para poder conectarse con una base de datos MySQL de manera local o remota, para facilitar el uso de este sistema de gestión de bases de datos.

2.2.3 SERVIDOR DE APLICACIÓN

2.2.3.1 Apache TOMCAT

Es un servidor de código abierto para aplicaciones JAVA web como JSP (JavaServer Pages), Java SE (Standard Edition) y Servlets. La implementación de SOAP en este servidor, permite invocar los servicios de SOAP mediante dos métodos:

RPC (Remote Procedure Call), que está basado en mensajes y usa el modelo cliente- servidor para ejecutar remotamente los servicios de SOAP de manera síncrona.

SMTP (Simple Mail Transfer Protocol) el cual es un modelo basado en mensajes de igual manera y también transporta los servicios SOAP al servidor, pero a diferencia del otro método este lo hace de manera asíncrona

ya que puede almacenar un mensaje para entregar más tarde. (Goodwill, 2002)

2.2.4 ENTORNOS DE DESARROLLO DE LA APLICACIÓN

2.2.4.1 NetBeans IDE

Este entorno de desarrollo para Java y otros lenguajes, es de código abierto y está basado en Framework RCP (Rich-Client Platform), por tanto no requiere de instalar plugins o módulos externos para funcionar, sino que incluye toda su funcionalidad en un solo paquete. Es multiplataforma e integra herramientas diversas para la creación de aplicaciones orientadas a servicios como Apache Tomcat o Glassfish para aplicaciones web y módulos de conectividad con bases de datos como MySQL de Oracle o SQL Server de Microsoft. Por su soporte y su flexibilidad se adapta a necesidades de uso educativo tanto como para desarrollos profesionales. (Bordeau T., 2017)

2.2.4.2 Android Studio

Es el entorno de desarrollo oficial para aplicaciones Android, el cual provee a los desarrolladores de todas las herramientas necesarias para crear aplicaciones para dispositivos Android desde el código a la interfaz de usuario, gracias a su propio repositorio SDK (Software Development Kit) y un emulador integrado que permite ejecutar aplicaciones en tiempo real en un sistema operativo android virtualizado. (Cruz, 2016)

Para crear aplicaciones con este IDE se necesitan conocimientos del lenguaje Java en el que se basa, pero con ciertas particularidades que son exclusivas de Android. La estructura de un proyecto en Android Studio consta principalmente de dos partes; los módulos con el código fuente y los archivos de recursos como archivos externos, librerías, etc. (Android Developers, 2017)

2.2.5 TUTORÍA ACADÉMICA

La tutoría es el proceso de asistencia y seguimiento que realiza un profesor a un estudiante, orientado a mejorar su formación integral.

Basado en el “Reglamento de Tutoría Académica de la Universidad Tecnológica Equinoccial”, el objetivo de la misma es el de brindar el apoyo al proceso de formación del estudiante desde la institución proveyéndolo de los recursos y herramientas que faciliten esta labor, orientándolo en aspectos académicos y personales de manera personalizada. (Consejo Universitario UTE, 2017)

Las Tutorías pueden ser de tres tipos:

Académica: La cual se orienta principalmente a la personalización de la tutoría brindando seguimiento y apoyo individual según las características y requerimientos del estudiante.

De Asignatura: Que corresponde a los procesos de nivelación y refuerzo del conocimiento en materias o cursos tomados por el estudiante.

Especial: En la cual intervienen departamentos de apoyo de bienestar estudiantil (médicos, psicológicos, etc.) y se orienta en resolver asuntos intrapersonales e interpersonales.

Estos tres tipos de tutorías se relacionan entre sí como complementos las unas de las otras, dado que todas son necesarias para cumplir el objetivo planteado en el documento de referencia.

3. METODOLOGÍA

La investigación del presente trabajo se realizó mediante el método deductivo de la investigación científica, el cual permite recolectar información sobre el objeto de estudio e identificar el problema a solucionar. En este caso los procesos implícitos y explícitos para agendar citas de tutorías en la Universidad Tecnológica Equinoccial se encuentran parcialmente definidos y fueron documentados mediante diagramas de flujo, se logró abstraer información relevante para comprender estos procesos, para posteriormente plantear la aplicación móvil como posible solución al problema de investigación y finalizando con la experimentación mediante el prototipo funcional y datos que se acerquen a la realidad para simular un entorno de trabajo real. (INTECO, 2009)

Para el desarrollo de la aplicación se utilizó el modelo evolutivo basado en prototipos para el desarrollo de software. Por ser una metodología ágil que permite mejorar la calidad de los requisitos y experimentar con ellos, iterando hasta llegar a concluir con una aplicación estable y funcional, este modelo está definido por Roger Pressman como muestra la Figura 5.

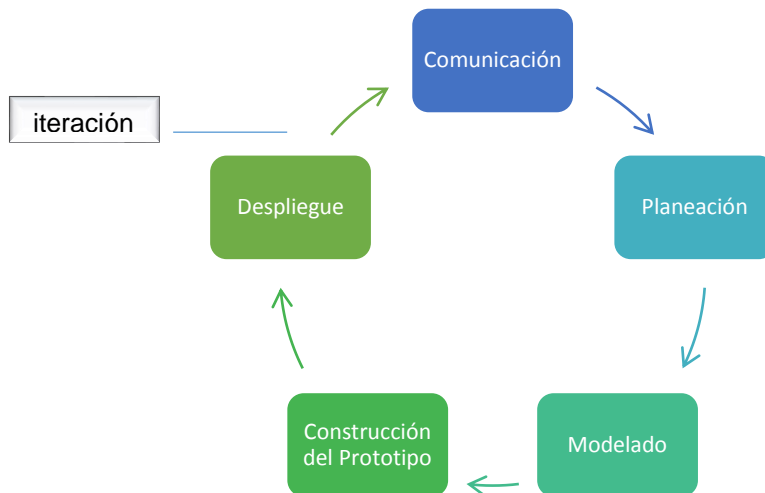


Figura 5. Flujo del Proceso Evolutivo del Desarrollo de Software

3.1 METODOLOGÍA DE DESARROLLO BASADA EN PROTOTIPOS

La metodología de desarrollo basada en prototipos contiene etapas definidas, en las que se llevaron a cabo actividades específicas para realizar el prototipo las mismas que se fueron refinando hasta definir el software por completo.

3.1.1 COMUNICACIÓN

En esta fase se recolectaron los requerimientos preliminares para iniciar el desarrollo, mediante observación de los sistemas y procesos actuales, así como el uso de la experiencia, obtenida a través de los años como estudiante y entrevistas a algunos docentes con conocimiento de campo en el área de tutorías.

3.1.2 PLANEACIÓN Y MODELADO

Durante esta fase se definieron las funcionalidades preliminares del software con la ayuda de la técnica de prototipado Storyboard, que permite visualizar los el producto a desarrollar con la funcionalidad en cada una de sus interfaces. Esta etapa permite conocer el tipo de sistema que se va a desarrollar basado en los requerimientos y en la tecnología disponible.

3.1.3 CONSTRUCCIÓN DEL PROTOTIPO

En esta fase se construyeron los componentes del sistema; base de datos en MySQL server, webservice SOAP alojada en un servidor Apache Tomcat y la aplicación móvil desarrollada en el IDE Android Studio. Se fueron implementando funcionalidades y cambios en estos tres componentes según evolucionaba el desarrollo hasta construir un prototipo que satisficiera los requerimientos planteados.

3.1.4 DESPLIEGUE

En esta fase se entrega un prototipo con la funcionalidad implementada en la fase de construcción y mediante pruebas de caja negra y se recibe una retroalimentación, que en el caso de ser negativa, sirve para estabilizar el software y corregir errores, y en el caso de ser positiva, se asocia a los cambios evolutivos y mejoras para reiniciar el ciclo.

4. RESULTADOS Y DISCUSIÓN

4.1 RESULTADOS

Los resultados obtenidos se ajustan al objetivo planteado, que es el de Desarrollar una aplicación móvil en Android, que podría ser implementada en la universidad para agendar tutorías entre estudiantes y docentes de la UTE.

A continuación se muestran los resultados obtenidos en cada etapa de la metodología de desarrollo descrita anteriormente.

4.1.1 COMUNICACIÓN

Mediante la observación del sistema actual implementado en la plataforma web de la UTE "SITUTOR", así como situaciones de la vida estudiantil recolectadas por medio de la experiencia y entrevistas realizadas a docentes como el Ing. Luis Hidalgo, director del departamento de tutorías de la Facultad de Ciencias de la Ingeniería de la UTE y el Ing. Víctor Hugo Gálvez, docente de la carrera de Ingeniería Informática y Ciencias de la Computación se determinó un proceso general para agendar tutorías.

4.1.1.1 Consideraciones sobre los procesos

- Los docentes tienen asignado el horario de tutorías desde el inicio del periodo lectivo o semestre.
- Los docentes deber dar prioridad a alumnos en los primeros niveles y a estudiantes que se encuentran con tercera matrícula.
- Los alumnos buscan tutorías en las materias que encuentran dificultad.
- Las tutorías son personales, es decir que para el desarrollo del presente proyecto se tomó en cuenta esa modalidad de tutorías.

4.1.1.2 Proceso para agendar tutorías basada en la recolección de requisitos:

Siguiendo el desarrollo evolutivo de la metodología y partiendo del proceso definido en el Diagrama de Actividades de la Figura 6, mismo que no es definitivo y se fue refinando mediante los requisitos funcionales y no funcionales.

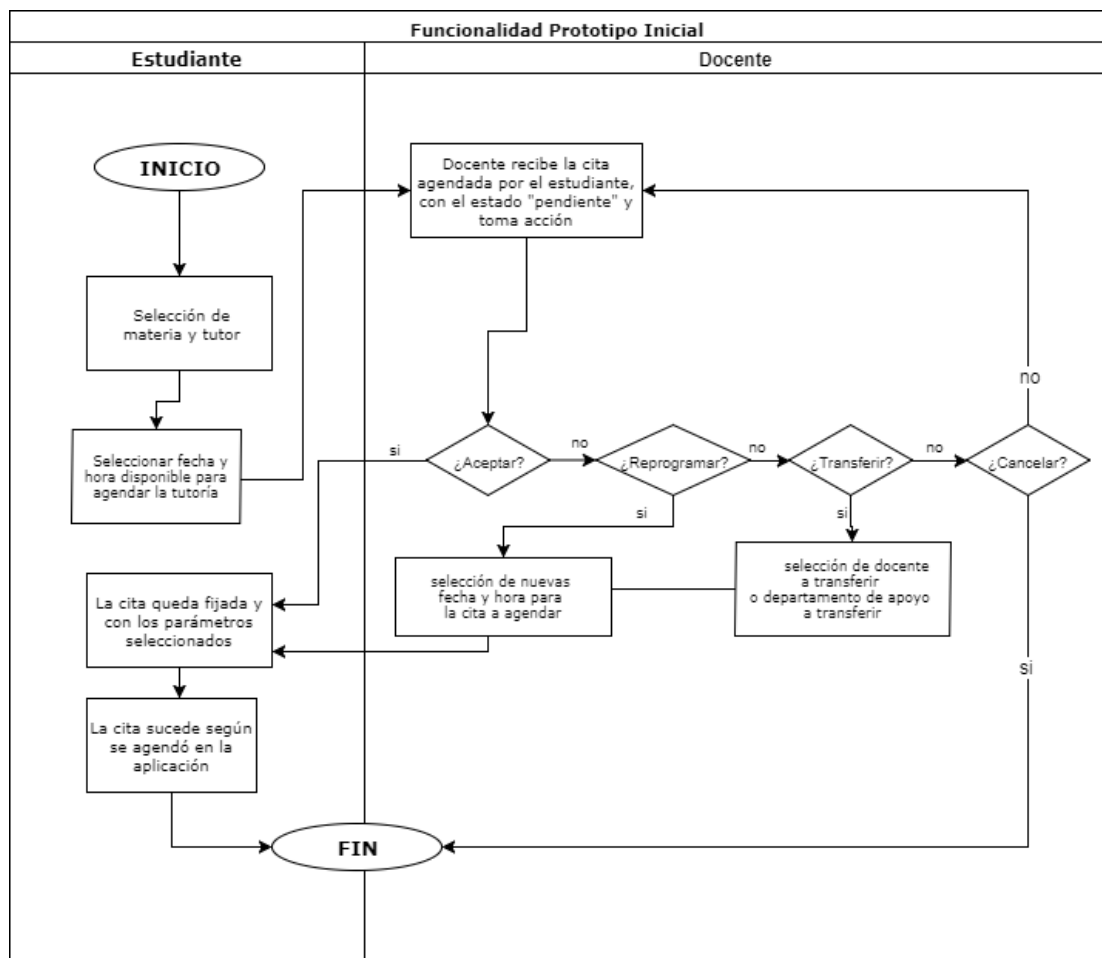


Figura 6. Diagrama de Actividades a realizar en el primer prototipo

4.1.1.3 Requisitos funcionales

Estos requisitos se implementaron para el correcto funcionamiento de la aplicación y que cumpla con su finalidad de brindar el apoyo tecnológico y práctico de una ampliación móvil como herramienta al servicio de la educación, para esto se elaboró la Tabla1.

Tabla 1. Requisitos funcionales

Identificador	Nombre	Descripción	Prioridad
RF01	Encuesta para estudiantes.	En el módulo de Estudiantes, se agregó una encuesta de tipo socioeconómica y de hábitos de estudio, misma que debe ser llenada en caso de recibir tutorías y esta información estará disponible para el docente tutor.	Alta
RF02	Generación de Informes	Los registros de las citas deben almacenarse para generar reportes, es así que se agregó una sección de informes en el módulo de Docentes para que puedan tener control sobre la gestión de tutorías.	Alta
RF03	Observaciones acerca de las citas atendidas	Los docentes tienen la posibilidad de agregar observaciones sobre las citas de tutorías atendidas así como enviar correos para convocar una nueva cita en caso de que no haya sido atendida.	Alta
RF04	Satisfacción del estudiante	Los estudiantes pueden calificar su experiencia con la tutoría recibida en una escala de satisfacción del 1 al 5, siendo 1 un bajo nivel satisfacción y 5 el nivel de satisfacción más alto.	Alta
RF05	Transferencia a departamentos de apoyo	La transferencia de Tutorías puede también hacerse a departamentos de apoyo como: el departamento médico, bienestar estudiantil y orientación psicológica. En caso de que el docente lo crea pertinente.	Alta

4.1.1.4 Requisitos no funcionales

Estos requisitos están implícitos en el desarrollo y se implementan principalmente para crear sensaciones positivas en el usuario y que se sienta el apoyo de la aplicación en el cumplimiento de su propósito, mostrados en la Tabla 2.

Tabla 2. Requisitos no funcionales

Identificador	Nombre	Descripción	Prioridad
RNF01	Interfaz de usuario	De fácil uso y con colores que identifiquen a la aplicación con la Universidad.	Alta
RNF02	Desempeño	Fluidez y velocidad de procesamientos en consultas y mensajes de respuesta.	Alta
RNF03	Rendimiento	Se espera que la aplicación consuma los recursos del dispositivo y de red de manera justa y que permita el correcto funcionamiento del resto de funciones.	Alta
RNF04	Disponibilidad	Se debe garantizar al usuario un correcto funcionamiento de la aplicación, siempre que disponga de conexión a internet y los servidores se encuentren operativos.	Alta
RNF05	Seguridad	Se contempló un inicio de sesión con procesos de autenticación mediante nombre de usuario y contraseña únicos, para identificar al usuario de la aplicación	Alta

4.1.2 PLANEACIÓN Y MODELADO

Esta etapa tiene como fin la de conceptualizar el software, para conocer la funcionalidad requerida y la que verdaderamente se puede implementar, es por eso que mediante la técnica de prototipado Storyboard se pueden conocer muchos aspectos del software y de su funcionamiento general, ya que para la realización de determinadas tareas, se debe contar con una plataforma tecnológica que lo soporte.

La Figura 7 muestra pantallas que simulan el comportamiento de la aplicación en el módulo del estudiante, desde que inicia sesión hasta que finalmente agenda su cita, pasando por el proceso de selección de materia, selección de docente, fechas y horarios disponibles.

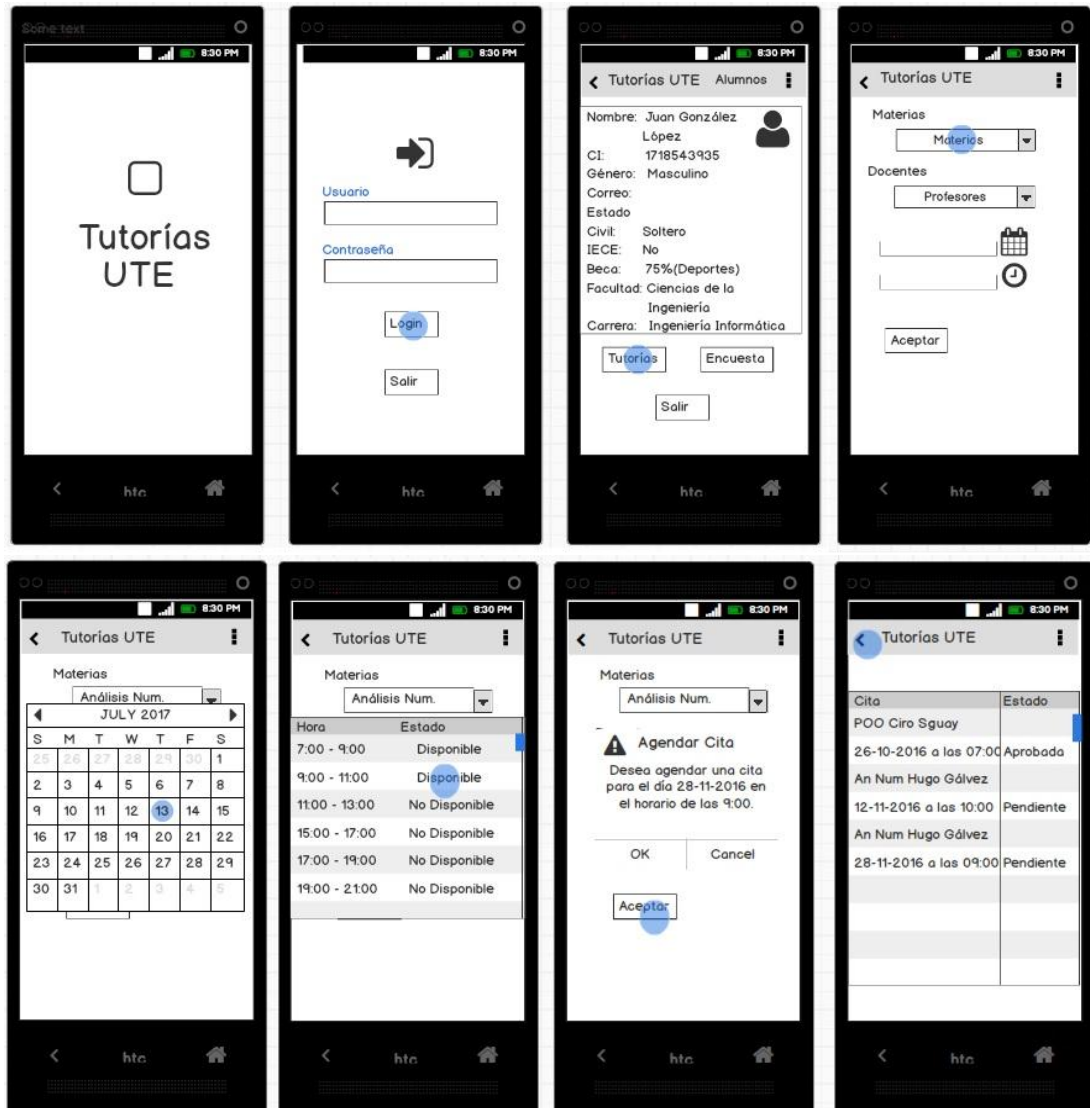


Figura 7. Storyboard del módulo de Estudiante

Para dar el soporte tecnológico a esta aplicación móvil se conceptualizó la plataforma como un sistema de tres componentes principales:

- La base de datos está alojada en un servidor remoto accesible desde internet en el SGBDD MySQL que se encarga de alimentar el software con datos aproximados a los reales para poder comprobar el correcto funcionamiento del software.

- El webservice SOAP desarrollado en Java procesa las consultas en la base de datos para ser interpretadas por la aplicación, este se aloja en otro servidor remoto accesible desde internet.
- En la aplicación se muestran todas las interfaces que controlan la funcionalidad del software. Las acciones realizadas en la aplicación se envían por internet al Webservice SOAP mediante la librería KSOAP para Android. Mientras el Webservice procesa las consultas a la base de datos mediante la clase BDD Manager y entrega los resultados que se envían de nuevo por el Webservice a la aplicación y los muestra dentro de su interfaz según se haya solicitado.

4.1.3 CONSTRUCCIÓN DEL PROTOTIPO

En la construcción de la aplicación intervinieron tres plataformas: Base de Datos y WebService que conforman el servidor y la aplicación móvil que es el cliente. Haciendo uso del prototipo en storyboard como referencia se desarrolló el servidor en estas dos plataformas que actúan conjuntamente para satisfacer las peticiones del cliente.

4.1.3.1 Base de Datos

En la Figura 8 se representa el modelo de la base de datos realizado con la herramienta MySQL Workbench. Se puede apreciar cómo se relacionan las tablas y sus atributos entre sí y mediante consultas que se convierten en peticiones y mensajes de respuesta, interactúan con el webservice y este a su vez con la aplicación.

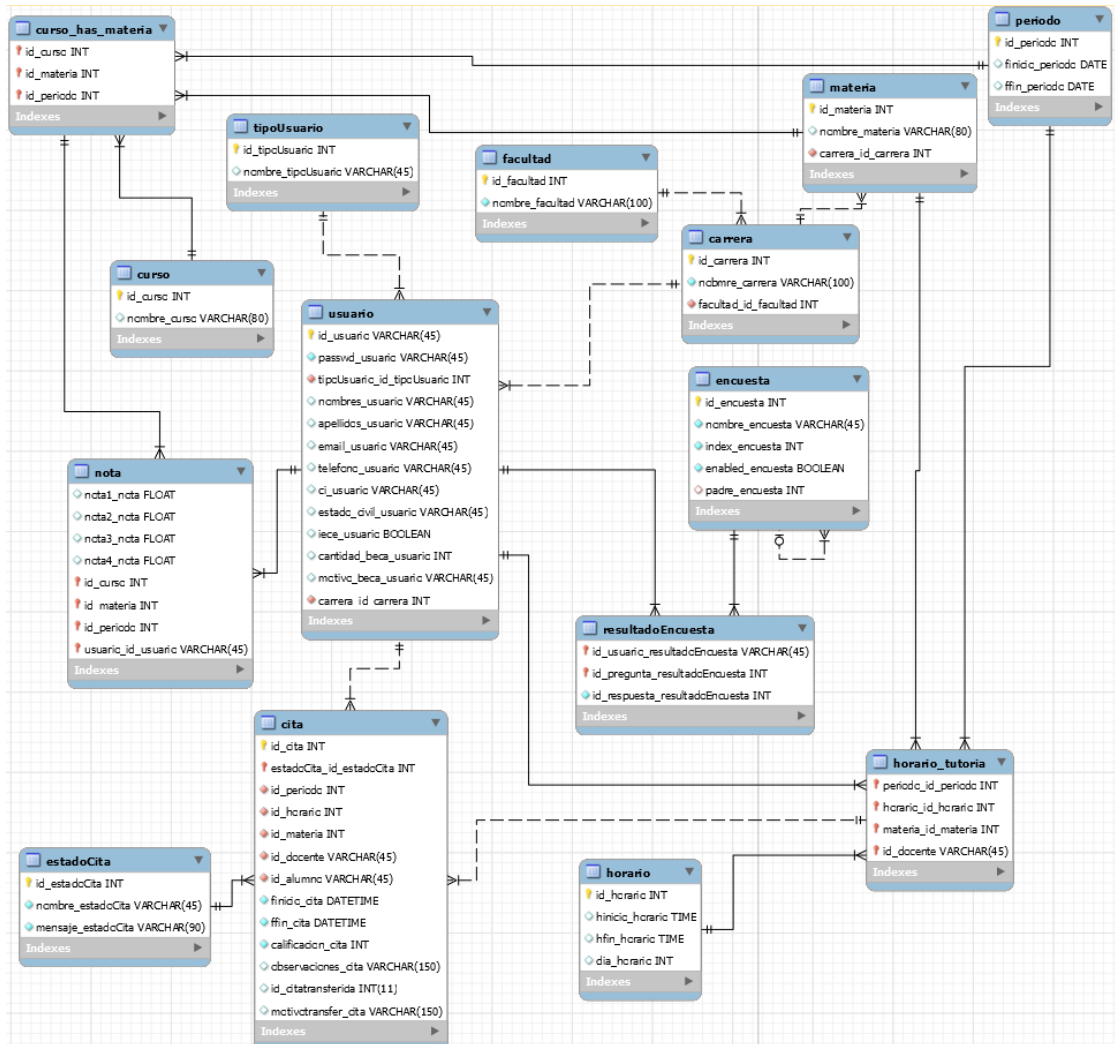


Figura 8. Modelo relacional de la Base de Datos

4.1.3.2 Webservice

Este componente es el vínculo entre la base de datos y la aplicación, tanto para enviar peticiones o consultas como para transportar y entregar los resultados.

Mediante una herramienta utilitaria de nombre SoapUI cuya finalidad es la probar aplicaciones orientadas a servicios (SOA), se comprobó el correcto funcionamiento del webservice alojado en el servidor Apache Tomcat. (Goodwill, 2002)

La herramienta recibe la dirección WSDL para inmediatamente obtener la descripción de los métodos que ejecuta SOAP en el servidor como muestra la “Figura 10”. Existen dos clases muy importantes que son:

TutoríasWS.- Esta clase representa todas las funciones disponibles para la aplicación.

BDDManager.- Esta clase es la que interactúa con la Base de datos. Su fuente de datos son las sentencias SQL que directamente ejecutan consultas, la Figura 9 muestra los métodos que representan a las sentencias, el webservice se alimenta de las sentencias y la base responde, y es cuando esta clase distribuye a otras clases los resultados.

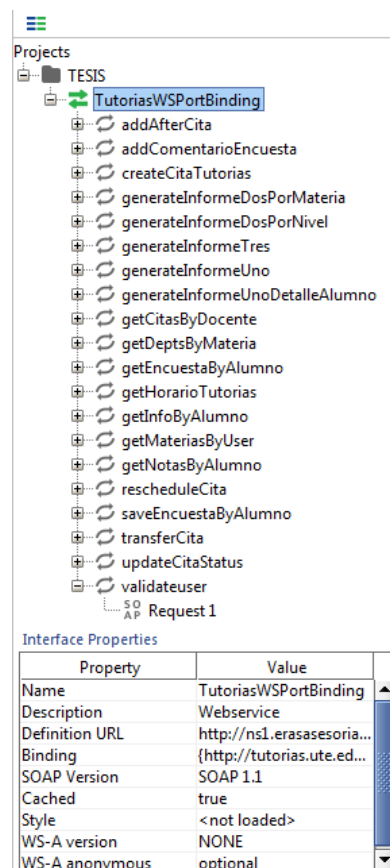


Figura 9. Metodos implementados en webservice SOAP mostrados por la herramienta de pruebas "SoapUI".

Al seleccionar las peticiones que muestra cada método del servicio aparece una ventana interna donde se realiza la petición al servidor y este envía el resultado respectivo, ambas acciones se presentan en formato XML como muestra la Figura 10.

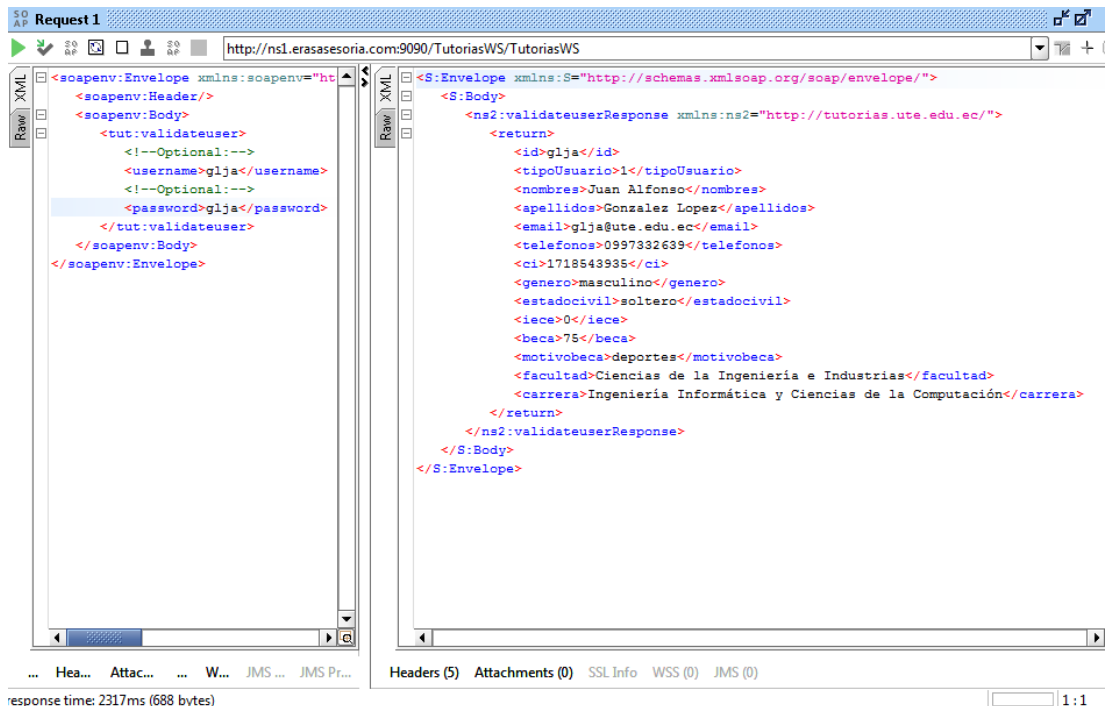


Figura 10. Envío de peticiones y muestra de resultados del webservice en SoapUI.

4.1.3.3 Aplicación Android

La aplicación móvil se desarrolló en la última versión hasta la fecha de Android Studio y funciona en plataformas Android 4.0.3 Kit Kat o superior. En la Figura 11 se observa la estructura del proyecto dentro del IDE que se muestra en una estructura de raíz con carpetas que contienen archivos que conforman los diferentes componentes, clases y recursos de la aplicación.

Estos son los más importantes:

- **AndroidManifest.xml:** archivo que se encuentra contenido en la carpeta manifest y este detalla los permisos que requiere la aplicación para ejecutarse dentro del dispositivo, en este caso se le otorgaron permisos del uso de internet para su interacción con el servidor y permisos para obtener cuentas y leer contactos para enviar correos electrónicos, adicionalmente se definieron aquí algunas características de la interfaz como algunos estilos y orientación de las pantallas.

- **ec.edu.ute.tutoriasute:** carpeta contenida dentro de la carpeta java, y a su vez contenedor de las actividades principales TutoriasUTEapp y SplashActivity, además de los siguientes paquetes:
 - **entities:** contiene las clases representativas de los objetos que interactúan con el webservice.
 - **gui:** donde se encuentran las actividades que interactúan con la interfaz, así como adaptadores que se encargan de presentar adecuadamente los datos en las diferentes vistas.
 - **util:** que contiene clases utilitarias para formatear o presentar datos y contiene una clase para crear diálogos personalizados.
 - **wsclient:** donde se encuentran las clases más importantes de la aplicación, es decir las que proveen de funcionalidad y se encargan de gestionar las comunicaciones y los métodos o funciones que ejecutan peticiones y reciben mensajes para interpretarlos como respuestas a las peticiones.
- **res:** carpeta que contiene recursos de la aplicación, así como diseños de la interfaz descritos en xml.

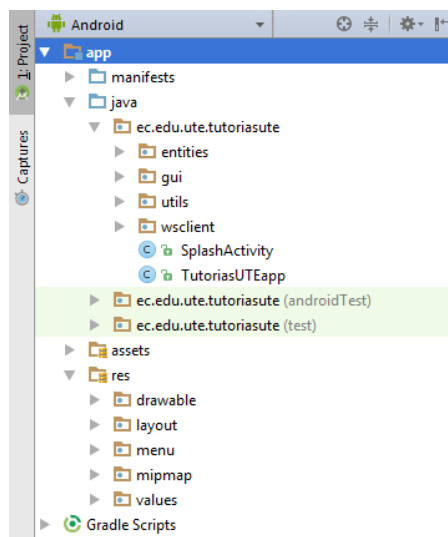


Figura 11. Estructura de la aplicación dentro del IDE Android Studio

Para visualizar el funcionamiento del software desde la aplicación, se puede observar el diagrama de secuencia de la Figura 12 partiendo de que la aplicación interactúa desde su interfaz con el usuario y este a su vez ejecuta las funciones descritas en la aplicación que interactúan con el servicio web y este con la base de datos.

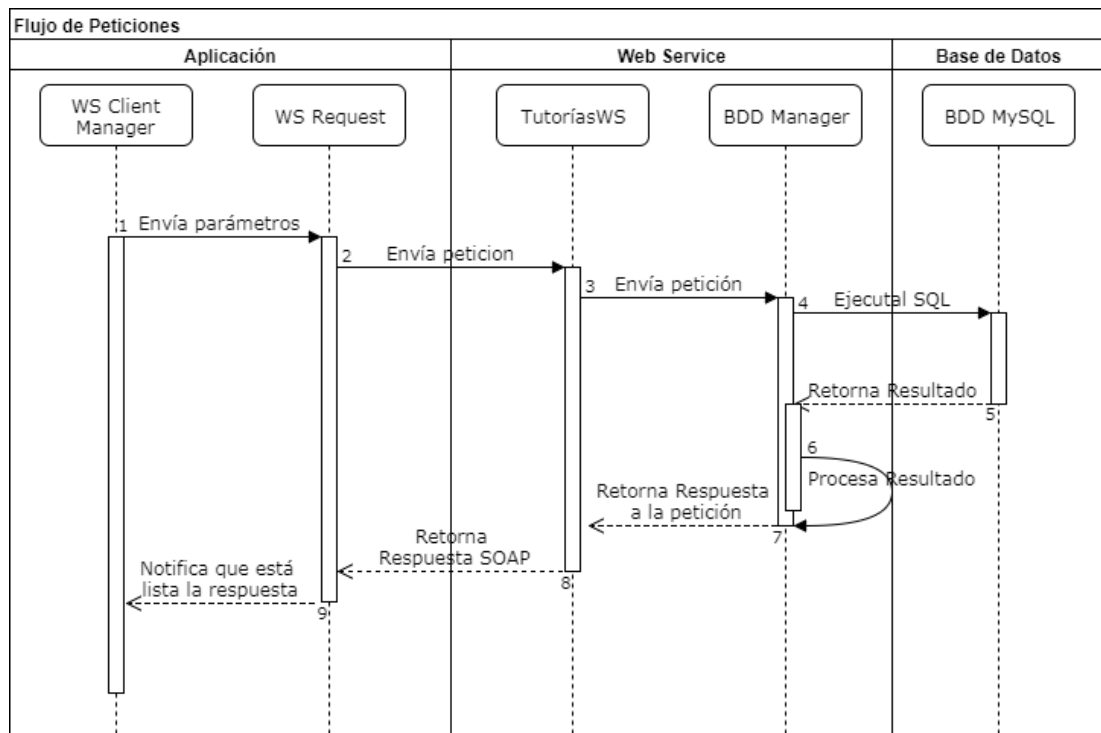


Figura 12. Diagrama de Secuencia: Peticiones desde la Aplicación

4.1.4 DESPLIEGUE

4.1.4.1 Entrega del Prototipo

El prototipo funcional se fue refinando con el tiempo hasta llegar a su etapa final. A continuación se muestran capturas de pantalla del prototipo en funcionamiento.

En la Figura 13 se puede observar la pantalla de inicio y la pantalla de inicio de sesión donde se ingresan los datos de usuario, sea este docente o estudiante.



Figura 13. Prototipo Funcional. “Splash screen” y pantalla de inicio de sesión

En la Figura 14 se pueden observar las pantallas representativas del módulo del Estudiante:

- Pantalla donde se visualiza información básica y el menú.
- Pantalla que muestra un listado de las citas creadas, misma que muestra a cada cita con el nombre de la materia, el docente asignado, la fecha y hora de inicio de la cita y su estado que puede variar entre: pendiente, aceptado, reprogramado, transferido, cancelado y finalizado atendido.
- La pantalla de creación de citas.
- La pantalla que permite llenar la encuesta socioeconómica y de hábitos de estudio.

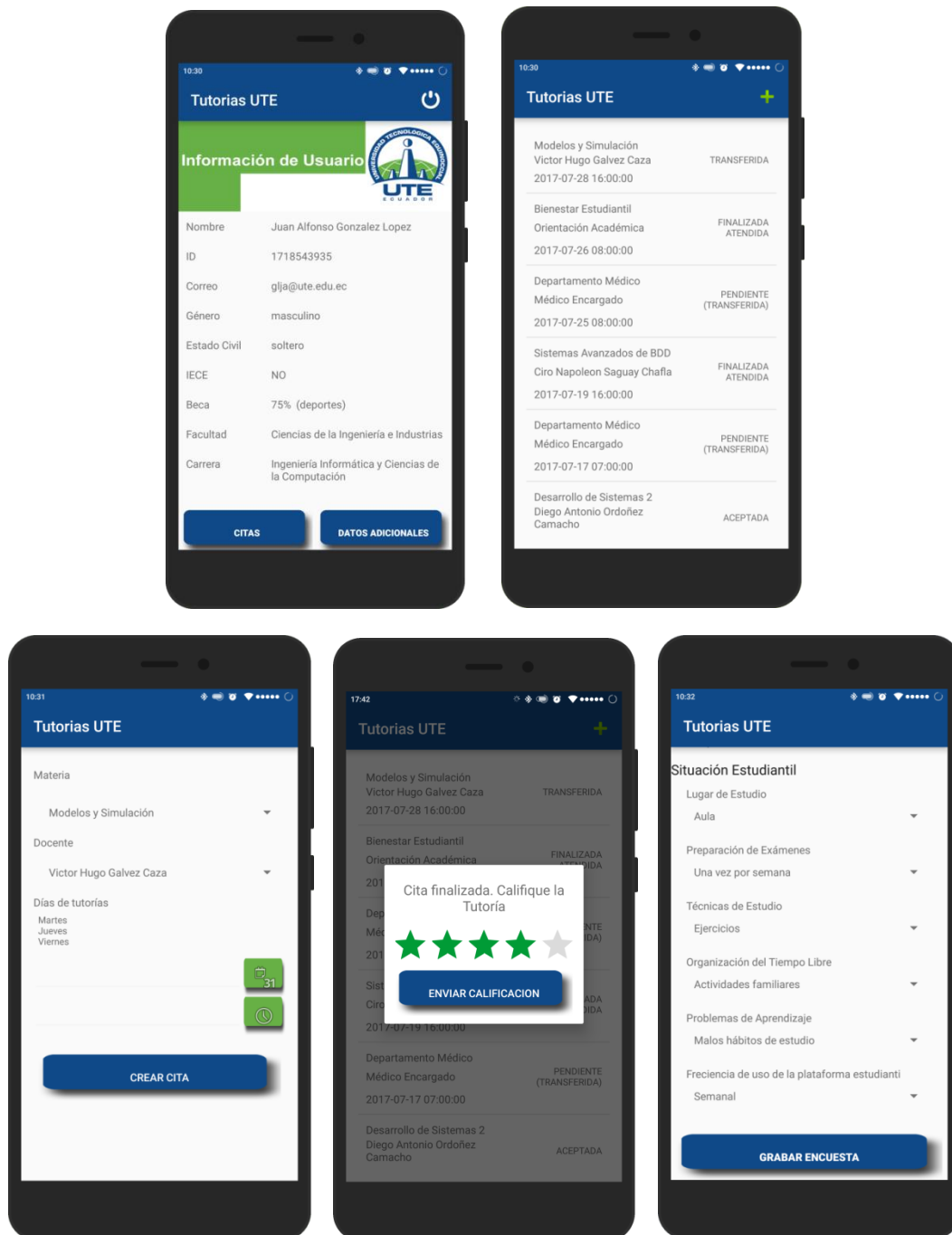


Figura 14. Prototipo Funcional – Módulo Estudiantes.

El prototipo funcional en su módulo de Docente muestra una pantalla inicial similar a la del estudiante ya que ambos tienen un botón para acceder a la actividad de gestión de sus citas, sin embargo el módulo de docente incluye un generador de reportes basado en los datos de las citas atendidas y la información de los alumnos como nivel y materias.

Características de la Pantalla de Citas:

La Figura 15 muestra el módulo de docentes con las siguientes características:

- En la pantalla de citas se muestra un listado de citas agendadas por sus distintos alumnos, agrupados por fecha en orden descendente para una mejor gestión de las mismas y en cada una se detalla el nombre de la materia, el estado de la cita, la hora de inicio y si es transferida se muestra el motivo de la transferencia
- Cuando se toque esta información se mostrará un cuadro de dialogo que le permitirá al docente cambiar el estado de la cita según corresponda sus necesidades y las del estudiante, con la particularidad que al transferirla, se incluye en la lista de materias, a los departamentos de apoyo.
- En la parte derecha, un botón con el nombre del estudiante que agendó la cita, y el color de su botón muestra una característica específica, si el color es azul, el estudiante no tiene terceras matrículas, por otra parte, si es rojo indica que tiene tercera matrícula en alguna materia. Este botón lleva al perfil del estudiante que contiene su información y los datos de la encuesta, además de la posibilidad de dejar un comentario basado en la entrevista de la primera cita que queda registrado y será visible únicamente para los docentes que revisen el perfil de este estudiante desde la aplicación, este mensaje se actualiza en caso de que el docente decida hacerlo, es decir solo puede escribir un comentario por estudiante.

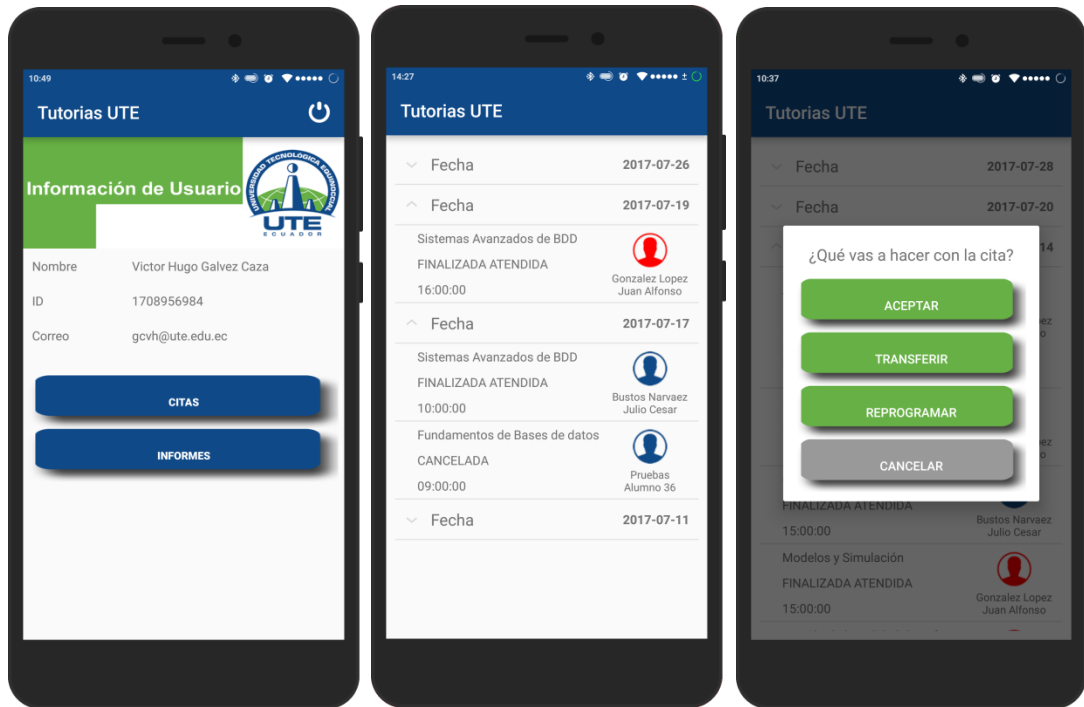


Figura 15. Prototipo Funcional – Módulo Docentes. Pantalla de Inicio y Citas

Características de la Pantalla de Informes:

En la Figura 16 de muestra la pantalla de informes con un menú de tres opciones que llevan al generador de informes que indica cada opción.

- **Informe General de Citas:** este informe hace un conteo del total de alumnos que tomaron citas independientemente de si fueron atendidos o no, sin embargo se contabilizará el total de citas atendidas y se desglosa por alumno el total de citas en cada estado. Adicionalmente al tocar en un alumno que tenga citas atendidas se despliega un informe de detalle de citas tomado por cada alumno que queda como evidencia de la cita e incluye las observaciones del docente de cada una.
- **Informe de Estudiantes Atendidos por Nivel:** este informe recopila el total de estudiantes que tomaron citas y los agrupa dependiendo del nivel al que pertenecen en determinada materia con un porcentaje del total de alumnos atendidos por cada nivel. Adicionalmente al tocar el nivel presenta una contabilización de alumnos y los agrupa por materia.

- Informe de Citas Transferidas:** en este informe se muestran todas las citas que fueron transferidas por el docente que muestran la fecha a la que transfirió la cita, el nombre del alumno, su nivel, al docente o departamento que fue transferido y el estado de la cita transferida.



Figura 16. Prototipo Funcional – Módulo Docentes. Pantallas de Informes

4.1.4.2 Pruebas y Retroalimentación

Para probar el prototipo se utilizaron pruebas de caja negra, que son funcionales, es decir que mediante datos y acciones de entrada-salida se determina si el software está cumpliendo correctamente con sus funciones. Concretamente se generaron datos como usuarios de tipo estudiantes y docentes, a los que se le asignaron cursos, se crearon periodos lectivos, facultades, carreras y materias además de los departamentos de apoyo, se definieron también horarios y estructuras similares a las que se podrían usar dentro de la universidad para poder simular el entorno en el que podría operar la aplicación.

En las pruebas realizadas se buscó tener un balance correcto del prototipo entre, funcionalidad, facilidad de uso, interacción con el dispositivo y uso adecuado de recursos, en la Tabla 3 se muestran estas pruebas.

Tabla 3. Pruebas realizadas al software

Tipo de Prueba	Requerimiento	Cumple con el requerimiento	
		SI	NO
Interacción con el dispositivo	Conexión a internet mediante redes WiFi y móviles	✓	
	Uso del cliente de mail del dispositivo para envío de correo a estudiantes	✓	
Funcionalidad de la aplicación	Inicio correcto	✓	
	Inicio de sesión persistente	✓	
	Consulta de información y correcta interpretación de datos	✓	
	Creación y recuperación correcta de citas de tutorías	✓	
	Correcto funcionamiento de los métodos para reprogramar y transferir citas de tutorías	✓	
	Adecuado almacenamiento de las respuestas de los estudiantes en la encuesta	✓	
Interfaces	Uso de colores institucionales y facilidad de uso	✓	
	Funcionamiento correcto en resoluciones de pantalla HD y Full HD	✓	
	Presentación de informes comprensible y uso correcto de la pantalla	✓	

La retroalimentación se logró por el uso de la aplicación con los métodos de prueba y error para corregir problemas que se suscitaron durante el desarrollo, además de la redefinición de requisitos que se presentaron en cada entrega del prototipo, que se refinó hasta lograr esta versión funcional y estable.

4.2 DISCUSIÓN

Desde un punto de vista práctico y funcional, todas las aplicaciones de calendario y agenda para Android mencionados la sección 2.1 carecen de una funcionalidad importante del desarrollo presentado en este trabajo, la cual es de simular la integración a una base de datos de estudiantes, docentes, horarios, cursos, etc. Este aspecto es muy importante a la hora presentar datos, generar reportes y en el propio proceso de agendar citas de citas, por el simple hecho de que la aplicación hace manejo de una cantidad de datos considerables que tienen que configurarse de manera que presenten correctamente la información que se solicite.

Como parte negativa y de manera objetiva, la aplicación desarrollada carece de alguna funcionalidad que sí tienen las aplicaciones de calendario antes vistas, como el de presentar notificaciones o crear alarmas.

Como punto diferencial, se puede decir que la aplicación no necesita integrarse con el calendario del dispositivo ni con los contactos como lo hacen las otras aplicaciones de calendario y esto es porque la red de actores y las funciones de uso exclusivo para actividades educativas y no sociales.

Por tanto dentro del ámbito institucional y educativo, tomando en cuenta la funcionalidad se logró configurar una aplicación adecuada y útil para el apoyo de las tareas que involucran el proceso de agenda de citas para tutorías.

5. CONCLUSIONES Y RECOMENDACIONES

En este capítulo se describen las conclusiones y recomendaciones que se obtuvieron del desarrollo del presente trabajo.

5.1 CONCLUSIONES

- El proceso para agendar de citas es bastante simple, sin embargo, se puede decir que se mejoró el proceso haciéndolo más orientado a resultados y no solo a un almacén de datos, el prototipo presentado cumple con el objetivo general del proyecto, es decir que la aplicación desarrollada, en efecto podría implementarse en la universidad para agendar tutorías.
- El uso de la librería externa de Android, KSOAP, para poder comunicarse con el webservice SOAP permite una rápida implementación de este protocolo en la aplicación ya que no es nativo y de otra manera se hubiesen tenido que programar los métodos para comunicarse y el desarrollo se hubiese extendido aún más, esto dio como resultado una óptima comunicación entre los servidores y la aplicación.
- Al haber usado la metodología por prototipos, siendo esta de tipo evolutivo, se pudieron implementar las correcciones y mejoras en el software sobre la marcha, sin tener que hacer grandes cambios en su estructura, más bien agregando las mejoras y haciendo cambios menores.
- Aunque la interfaz de usuario es simple, cumple con su propósito de identificar a la institución en ella, y además es de uso fácil e intuitivo, debido al uso de controles sencillos como botones con colores claramente identificables y una experiencia táctil a la que usuarios de dispositivos móviles ya están acostumbrados.
- Por medio de la experimentación, se definió a la aplicación como funcional dentro de los parámetros de pruebas, que aunque fueron limitadas sobre todo por la cantidad de datos que tuvo que procesar,

se puede decir que cumplió con los requisitos, tanto los definidos al inicio, como los que se fueron agregando mediante cada iteración de la metodología.

- Para implementar esta aplicación en la universidad, se deben consumir los webservices de los sistemas informáticos de la universidad.

5.2 RECOMENDACIONES

- Para la implementación de este prototipo en un entorno real se recomienda integrar protocolos de seguridad para lograr que la información sea enteramente confidencial y mantenga su integridad y fiabilidad.
- El usuario debe tener una conexión estable a internet, sea por wifi o por red móvil para poder hacer uso de esta aplicación ya que prácticamente toda su funcionalidad está basada en envío y recepción de mensajes al servidor por medio de la red.
- Es deseable que se incluyan notificaciones personalizadas y alarmas a configurables en la creación de citas, para darle un extra tecnológico, que puede ser complementario a la funcionalidad actual.
- Se recomienda incluir a los procesos involucrados con las tutorías de la universidad, en documentos oficiales y manuales de buenas prácticas, tanto para alumnos como docentes, con la finalidad de que se sociabilicen y que se extienda su uso como herramienta complementaria del aprendizaje.
- Para el uso de la aplicación descrita en el presente proyecto se sugiere al usuario estudiante usarla con responsabilidad y estar consiente que una cita representa un compromiso que se debe respetar con seriedad y puntualidad.

BIBLIOGRAFÍA

- Darwin I. (2017). Android Cookbook: Problems and Solutions for Android Developers. California, Estados Unidos. Ed. O'Reilly Media.
Recuperado de <https://books.google.com.ec/books?id=AdPTDgAAQBAJ>
- Strategy Analytics. (2016). Android Captures Record 88 Percent Share of Global Smartphone Shipments in Q3 2016. Boston, Estados Unidos.
Recuperado de <https://www.strategyanalytics.com>
- Google Cloud (2017). Google Calendar. Google Inc. Recuperado de https://gsuite.google.com/intl/es-419/products/calendar/?utm_source=google&utm_medium=cpc
- Appgenix. (2017). Business Calendar. Appgenix Software. Recuperado de <http://www.businesscalendar.de/en/features>
- Any.do (2017). Any.do Cal. Any.do Recuperado de <https://www.any.do/cal/>
- Rodríguez L. (2005). Internet y Teleinformática. Universidad Rey Juan Carlos. Madrid, España. Recuperado de <https://previa.uclm.es/profesorado/licesio/ib.html>
- Berners-Lee T., Fielding R., Frystyc H. (1996). Hypertext Transfer Protocol – HTTP/1.0. Network Working Group. Recuperado de <https://www.rfc-editor.org/rfc/pdf/rfc1945.txt.pdf>
- Tidwell D., Snell J., Kulchenko P. (2001). Programming Webservices with SOAP. California, Estados Unidos. Ed. O'Reilly Media.

- Oracle Corporation. (2017). MySQL 5.7 Reference Manual. California, Estados Unidos. Recuperado de <https://dev.mysql.com/doc/refman/5.7/en/introduction.html>
- Comer D. (1996). Redes globales de información con Internet y TCP/IP: principios básicos, protocolos y arquitectura. Londres, Inglaterra. Ed. Pearson Educación
- Goodwill J. (2002). Using SOAP with Tomcat. California, Estados Unidos. O'Reilly Media. Recuperado de <http://www.onjava.com/pub/a/onjava/2002/02/27/tomcat.html>
- INTECO (2009). Ingeniería del Software: Metodologías y Ciclos de Vida. León, España. Ed. Instituto Nacional de Tecnologías de la Comunicación de España
- Cruz B. (2016). Android Studio 2 Essentials. Birmingham, Reino Unido. Ed. Packt Publishing. Recuperado de <https://books.google.com.ec/books?id=VwFwDQAAQBAJ>
- Bordeau T., Glick J., Grene S., Spurlin V., Woehr J. (2002). NetBeans: The Definitive Guide. California, Estados Unidos. O'Reilly Media. Recuperado de <https://books.google.es/books?hl=es&lr=&id=JCMBWozfcJUC>
- Documentación de Android Developers (2017). Conoce Android Studio. California, Estados Unidos. Android Studio. Recuperado de <https://developer.android.com/studio/intro/index.html>

Consejo Universitario UTE. (2017). Reglamento de Tutoría Académica.
Quito, Ecuador. Universidad Tecnológica Equinoccial. Recuperado de
<https://app.ute.edu.ec/lexute/lexute/REGLAMENTO%20DE%20TUTOR%C3%8DA%20ACAD%C3%89MICA%20R.%20No.%20084-SE-06-CU-UTE-2017.pdf>

Pressman R. (2010). Ingeniería del Software. Un Enfoque Práctico 7ma Ed.
México DF, México. McGraw-Hill Interamericana Editores.