"Software para el intercambio seguro de mensajes y archivos entre dispositivos móviles a través de la Internet"

Proyecto de Título presentado en conformidad a los requisitos para obtener el título de Ingeniero Ejecución en Computación e Informática.

Alumno:
Bryan Ramirez

Profesor Guía:
Patricio Galdames Sepulveda

Fecha:
22 de Julio de 2016

Resumen

Este proyecto se presenta para dar conformidad a los requisitos exigidos por la Universidad de Bío-Bío en el proceso de titulación para a la carrera de Ingeniería de Ejecución en Computación e Informática. Y su título es “Software para el intercambio seguro de mensajes y archivos entre dispositivos móviles a través de la Internet”.

El intercambio de mensajes de texto/imagenes desde un dispositivo móvil conectado a la Internet mediante una red inalámbrica, podría sufrir serios problemas de privacidad de datos. La mayoría de las aplicaciones de chat existentes en el mercado no protegen la privacidad de los datos. Aquellas que si protegen la privacidad, encriptan todos los mensajes incluso aquellos cuyo carácter es publico. Por lo tanto estas últimas podrían generar un consumo de energía innecesario.

Como consecuencia de ello, este proyecto de tesis propone e implementa una aplicación cliente-servidor. El cliente usa criptografía asimétrica y simétrica para la transmisión segura de los datos. De este modo ofrece confidencialidad e integridad a los usuarios.El servidor actua como relay permitiendo el intecambio de datos desde y hacia multiples usuarios desde su aplicación cliente móvil. Cabe destacar que el servidor solo se encarga de almacenar y distribuir las llaves públicas de los usuarios y en ningún caso aquella información sensible de los usuarios como contraseñas y llaves privada.

Contenido

Índice de Tablas

Indice de figuras

1. Introducción

Los programas de chat disponibles para Android como WhatsApp y Line, solo por nombrar los más populares, han facilitado y fomentado el intercambio de mensajes de texto e imágenes entre usuarios de redes inalámbricas. Existen al menos 1.000 millones de usuarios que usan dicha aplicación [1]. Muchos de estos usuarios intercambian a través de estas aplicaciones, todo tipo de información como por ejemplo contraseñas, datos de sus tarjeta de crédito.

Lamentablemente en sus inicios estos programas no protegian la privacidad de sus usuarios de forma efectiva. Ya que los datos se transmiten en texto plano, un individuo malintencionado que se encuentre escuchando el tráfico estos datos inalámbricos, puede capturarlos y potencialmente podría usarlos para cometer robos y fraudes económicos. Mientras se desarrollaba este proyecto de titulo, WhatsApp ha decidido incorporar la encriptacion automáticamente de cada uno de los mensajes de texto enviados por un usuario para proteger la privacidad pero no asi con las imágenes.

Si bien la encriptación nos ayuda a proteger la privacidad, su costo es el mayor consumo de energía por la ejecución de diversas operaciones matematicas en las que se basa la encriptación. Este costo es importante en teléfonos digitales cuya fuente de energía, su batería, esta limitada a un periodo de tiempo corto.

Es debido a estas observaciones que el presente trabajo diseñó e implementó una aplicación de red para teléfonos Android llamada CriptoChat. La base de este proyecto yace en el uso de herramientas criptográficas para asegurar una transmisión privada de datos en la Internet. La criptografía es la ciencia que se ocupa de las técnicas de cifrado o codificado destinadas a alterar las representaciones lingüísticas de ciertos mensajes con el fin de hacerlos ininteligibles a receptores no autorizados [2].

CriptoChat permite el intercambio de texto/imagenes encriptadas a través de una red inalámbrica y la Internet a partir de un servicio de relay desarrollado para la misma aplicación. Por ende esta aplicacion puede ser ejecutada incluso desde usuarios que están detrás de un router que opera con NAT. Con el fin de limitar el potencial costo energético de la encriptación, dejamos al usuario la libertad de decidir usar o no el uso de la encriptación para transmitir mensajes/imágenes que el considere que son privadas.

2. Definición de La empresa o Institución

2.1.Descripción de la empresa

2.1.1. Antecedentes generales

La Universidad del Bio-Bio es una corporación de derecho público autónoma, con patrimonio propio, dedicada a la enseñanza y el cultivo de las ciencias, las tecnologías, las letras y las artes. Fue creada por ley Nª 18.744, publicada en el Diario Oficial del 29 de Septiembre de 1988. Dispone de una infraestructura de 68.000 m2 construidos, quecomprenden aulas, laboratorios, talleres, bibliotecas, salas de estudio y recintos deportivos, instalados en tres campus: Concepción, Fernando May y La Castilla, éstos dos últimos localizados en la ciudad.

Visión

Ser reconocida a nivel nacional como una Universidad estatal, pública, regional, autónoma, compleja e innovadora con énfasis en la formación de capital humano, vinculada al desarrollo sustentable de la Región del Biobío y que aporta a la sociedad del conocimiento y al desarrollo armónico del país.

Misión

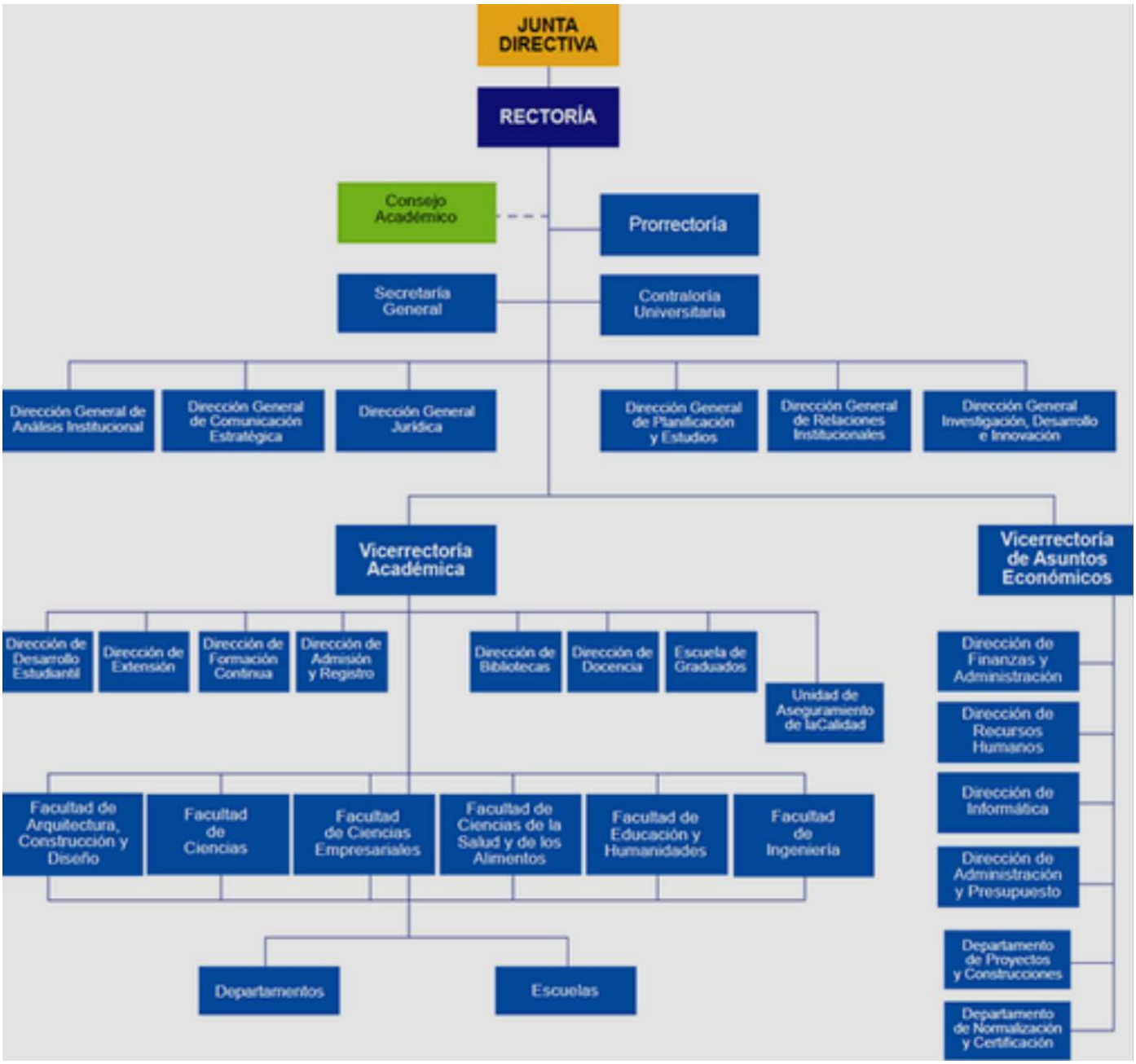
La Universidad del Bío-Bío es una institución de educación superior, pública, estatal y autónoma, de carácter regional, que se ha propuesto por misión:

Formar profesionales de excelencia capaces de dar respuesta a los desafíos de futuro, con un modelo educativo cuyo propósito es la formación integral del estudiante a partir de su realidad y sus potencialidades, promoviendo la movilidad social y la realización personal.

Fomentar la generación de conocimiento avanzado mediante la realización y la integración de actividades de formación de postgrado e investigación fundamental, aplicada y de desarrollo, vinculadas con el sector productivo, orientadas a áreas estratégicas regionales y nacionales.

Contribuir al desarrollo armónico y sustentable de la Región del Biobío, a través de la aplicación del conocimiento, formación continua y extensión, contribuyendo a la innovación, productividad y competitividad de organizaciones, ampliando el capital cultural de las personas, actuando de manera interactiva con el entorno y procurando la igualdad de oportunidades.Desarrollar una gestión académica y administrativa moderna, eficiente, eficaz y oportuna, centrada en el estudiante, con estándares de calidad certificada que le permiten destacarse a nivel nacional y avanzar en la internacionalización.

Estructura organizativa



Ilustracion 1: Organigrama de la universidade del Bio-Bio ([3])

2.1.2. Entorno

Competencia directa

La universidad del Bio-bio se centra principalmente en los estudiantes egresados de la enseñanza media de la región del Bio-bio, y en menor grado, profesionales para Post-grados.

Son muchas las instituciones que compiten por atraer nuevos alumnos cada año, siendo la competencia directa las Universidades del consejo de rectores y las Instituciones privadas. En menor grado las instituciones Técnicas, asi como tambien el resto de universidades del país.

Las principales competencias se mencionan a continuación:

Institución	**Rector**
Universidad de Concepción.	Sr. Sergio Lavanchy Merino.
Universidad Católica de la Santisima Concepcion.	Sr. Juan Miguel Cancino Cancino.
Universidad Técnica Federico Santa María.	Sr. José Rodríguez Pérez.
Universidad del Desarrollo.	Sr. Federico Valdés Lafontaine.
Universidad San Sebastián.	Sr. Ricardo Riesco Jaramillo.
Instituto Inacap.	Sr. Gonzalo Vargas Otte.

Tabla 1: Competencia.

Cuota de Mercado

Aunque la universidad del bio-bio no es la principal universidad de la región, cada año acapara un número elevado de nuevos matriculados, considerando que según las consultasa distintos medios de comunicación, un total de 233.284 personas rindieron la Prueba de Selección Universitaria (PSU) este año, a lo largo del país, y 34.000 aproximadamente en la región del Bio-bio

Para la Universidad del Bio Bio, 2294 fueron los alumnos matriculados en los 3 campus, siendo una cifra alta, comparada con años anteriores, además si se considera el porcentaje de alumnos que no logró entrar a una Universidad del consejo de rectores, la cifra podría alcanzar alrededor del 10% del mercado.

2.2.Descripción del área de estudio

Como indica el título del documento, el sistema está orientado al ambiente universitario, principalmente a los estudiantes, es por esto que se enmarca en el área de Desarrollo Estudiantil, y en segunda instancia en el administrador que leventará el servidor, debido a que es necesario establecer una conexión que actué como intermediario entre los estudiantes. Aun así, no son solo estos actores los encargados de la mantención y actualización del sistema, ya que cada usuario, sea profesor o de otro departamento contará con acceso al sistema donde podráintercambiar mensajes de manera segura.

2.3.Descripción de la problemática

Los programas de chat disponibles para Android como WhatsApp y Line, solo por nombrar los más populares, han facilitado y fomentado el intercambio de mensajes de texto e imágenes entre usuario de redes inalámbricas. Sin embargo, estos programas no protegen la privacidad de sus usuarios de forma efectiva. Algunos transmiten en texto plano los mensajes entre usuarios, otros permiten que aplicaciones externas puedan acceder a las llaves usadas en la encriptacion de los datos, de manera que la integridad y confidencialidad de los mensajes se ve realmente comprometida

3. Marco Teórico

TCP

La sigla TCP, se refiere a "***Transmission Control Protocol***", en español como "**Protocolo de Control de Transmisión"**, es uno de los protocolos fundamentales en Internet. Fue creado entre los años 1973 y 1974 por Vint Cerf y Robert Kahn.
Muchos programas dentro de una red de datos compuesta por redes de computadoras, pueden usar TCP para crear "conexiones" entre sí a través de las cuales puede enviarse un flujo de datos. El protocolo garantiza que los datos serán entregados en su destino sin errores y en el mismo orden en que se transmitieron. También proporciona un mecanismo para distinguir distintas aplicaciones dentro de una misma máquina, a través del concepto de Puerto.
Con el uso de protocolo TCP, las aplicaciones pueden comunicarse en forma segura (gracias al de acuse de recibo -ACK- del protocolo TCP) independientemente de las capas inferiores. Esto significa que los *routers* (que funcionan en la capa de Internet) sólo tiene que enviar los datos en forma de datagrama, sin preocuparse con el monitoreo de datos porque esta función la cumple la capa de transporte (o más específicamente el protocolo TCP).

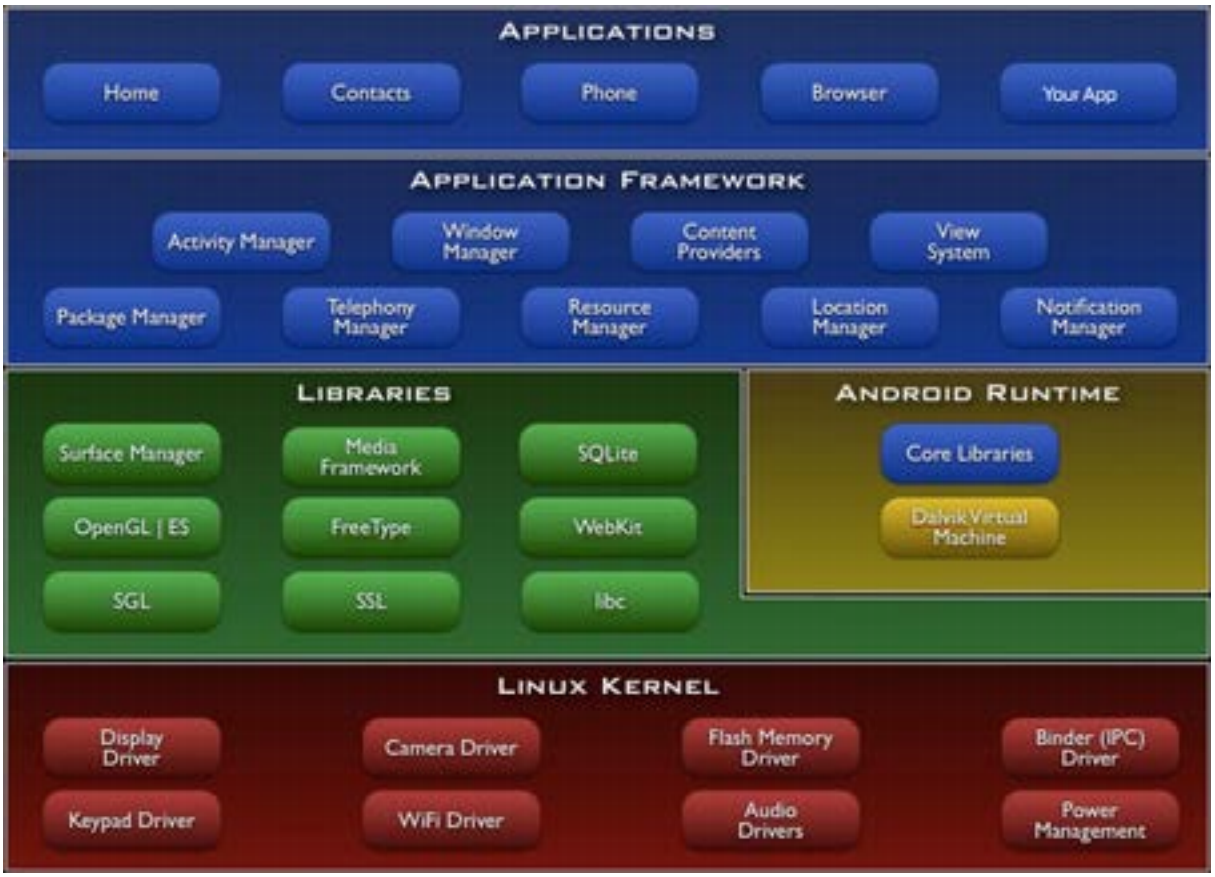
Sockets

"Socket" designa un concepto abstracto por el cual dos programas (posiblemente situados en computadoras distintas) pueden intercambiar cualquier flujo de datos, generalmente de manera fiable y ordenada.El término socket es también usado como el nombre de una interfaz de programación de aplicaciones (API) para la familia de protocolos de Internet TCP/IP, provista usualmente por el sistema operativo. Los sockets de Internet constituyen el mecanismo para la entrega de paquetes de datos provenientes de la tarjeta de red a los procesos o hilos apropiados. Un socket queda definido por un par de direcciones IP local y remota, un protocolo de transporte y un par de números de puerto local y remoto.[4]

TCP y Android

Android es un sistema operativo principalmente desarrollado para gracias a que al ser touchscreen no requieren de teclados adicionales ni algún otro hardware externo para su correcto funcionamiento, lo que se traducen en mayor espacio para mostrar información en una pantalla al usuario, en un espacio reducido (portable).

El principal potencial de este S.O. es que nos permite acceder a las diversas funcionalidades del hardware, el cual ha ido evolucionando con el avance tecnológico, actualmente hay diversos sensores de temperatura, velocidad, magnetismo, etc. esta el GPS, el WIFI, etc. Debido a este avance es que android también evoluciona junto con la tecnología, y nos permite el correcto funcionamiento de esta en nuestro dispositivo móvil. Para el correcto funcionamiento de Android, este se basa en la siguiente arquitectura



Ilustracion 2: Arquitectura de la aplicacion movil Android ([5])

De esto, nos centraremos en el Application Framework, ya que aquí se encuentran las funciones Activity manager, Notificacion Manager, así como también las Librerias SQLite y nuestras dependencias de protocolo de transmision TCP, todo estos recursos necesarios para nuestra aplicación.

Activity Manager: se encarga del correcto funcionamiento de las actividades, creación, destrucción, pausa, etc, es esencial para crear nuestra aplicación, que será mostrada en pantalla (a diferencia de un service que son tareas en segundo plano).

Notificacion Manager: Contiene todo lo necesario para manejar las notificaciones del celular con respecto al chat, con tal de notificar al usuario mediante vibraciones, shows o Toast de nuestro dispositivo móvil.

SQLite: Una base de datos independiente que permite a cada dispositivo contar con un registro independiente de datos almacenados.

Otros puntos a considerar son el uso de las APIs http de android, las cuales permiten la conexión cliente-servidor a través de HttpClient(Interface que encapsula todo lo necesario para ejecutar un HTTP request), HttpPost(el cual utiliza la URL del servidor para direccionar la conexión), HttpResponse(es un interface que permite acceder a los distintos datos de larespuesta del servidor), HttpEntity(usado para darle parámetros a la petición al servidor, como también a la respuesta).

3.2. Bases de datos SQLite

SQLite es un sistema de gestión de bases de datos relacional compatible con ACID, contenida en una relativamente pequeña biblioteca. A diferencia de los sistema de gestión de bases de datos cliente-servidor, el motor de SQLite no es un proceso independiente con el que el programa principal se comunica. En lugar de eso, la biblioteca SQLite se enlaza con el programa pasando a ser parte integral del mismo. El programa utiliza la funcionalidad de SQLite a través de llamadas simples a subrutinas y funciones.[2]
Este tipo de base de datos son necesarias en nuestro proyecto para almacenar las llaves privadas ahorrando de esta manera la memoria en nuestro dispositivo, al actuar con consultas independientes se vela la integridad de la llave privada en cada dispositivo movil asi mismo como al desintalar la aplicación se borra automaticamente toda la información contenida en el.[6]

4. Definición Del Proyecto

4.1.Objetivos del proyecto

4.1.1. Objetivo General:

Implementar una aplicación cliente servidor que permita a usuarios de dispositivos inalámbricos establecer sesiones de intercambio de texto e imágenes sobre un canal de comunicación seguro.

4.1.2. Objetivos Específicos:

- Implementar una técnica de criptografía simétrica para la encriptación/des encriptación de mensajes de texto e imágenes.
- Diseñar e implementar una función cliente para un dispositivo inalámbrico que permita el intercambio seguro de una llave de criptografia simetrica entre dos usuarios.
- Implementar una aplicación cliente para un dispositivo inalámbrico que permita la encriptación/desencriptación de mensajes de texto/imágenes transmitidas entre dos usuarios.
- Desarrollar una aplicación servidor que gestione los contactos de los usuarios y les notifica a los clientes que usuarios están online.
- Desarrollar una aplicación servidor que actúa como Relay entre diversos usuarios que desean comunicarse entre si.
- Desarrollar una aplicación que permita el almacenamiento seguro de llaves de criptografia asimetrica.
- Implementar una Firma Digital para validar el origen de un mensaje y verificar la autenticidad del emisor (llave Secreta).

4.2.Ambiente de Ingeniería de Software

Modelo de desarrollo orientado a prototipos

El Modelo de prototipos, pertenece a los modelos de desarrollo evolutivo. El prototipo debe ser construido en poco tiempo, usando los programas adecuados y no se debe utilizar muchos recursos.

Su uso se centra en la idea de ayudar a comprender los requisitos que plantea el usuario sobre todo si este no tiene una idea acabada de lo que se desea. Además puede utilizarse cuando el ingeniero en software tiene dudas acerca de la viabilidad de la solución pensada.

Se recomienda su uso cuando:

- Los requerimientos no son conocidos al principio
- Los usuarios desean visualizar el sistema final para así poder clarificar los requerimientos
- Los usuarios están constantemente añadiendo y probando el sistema o se ven muy indecisos en los requerimientos
- También se usa para lograr crear sistemas complejos (ya que se les añaden muchas cosas así mismo cuesta saber sin un prototipo que cosas van faltando)

Ventajas:

Las ventajas de un enfoque de desarrollo orientado a prototipos están dadas por: reducción de la incertidumbre y del riesgo, reducción de tiempo y de costos, incrementos en la aceptación del nuevo sistema, mejoras en la administración de proyectos, mejoras en la comunicación entre desarrolladores y clientes, etc.

Desventajas:

Si bien, el desarrollo orientado a prototipos tiene considerables ventajas, también presenta desventajas como: la dependencia de las herramientas de software (librerías o dependencias) para el éxito ya que la necesidad de disminución de incertidumbre depende de las iteraciones del prototipo, entre más iteraciones exista mejor y esto último se logra mediante el uso de mejores herramientas lo que hace a este proceso dependiente de las mismas. También, no es posible aplicar la metodología a todos los proyectos de software y, finalmente, la mala interpretación que pueden hacer los usuarios del prototipo, al cual pueden confundir con el sistema terminado.

Justificación:

Cuando se trabaja en una tecnología nueva, la reducción de tiempo y del riesgo son puntos importantes dentro del desarrollo, en este caso, la metodología orientada a prototipos nos brinda una ventaja considerable por sobre las otras metodologías.

Las características de esta aplicación, permite una gran gama de posibilidades, y el tiempo de desarrollo es reducido, aun así, se considera continuar desarrollando actualizaciones a futuro e iterar la mayor cantidad de veces posibles, probando nuevos requerimientos y adecuándose a la comodidad de los usuarios.

Aminorando las desventajas de la metodología, podemos decir que tenemos la capacidad de replantear requerimientos y reformular las diferentes opciones y configuraciones a medida que se va ejecutando y conociendo las técnicas de la criptografía. Al ser desarrollado personalmente en su casi totalidad, nos preocupamos de mantener las herramientas utilizadas desde el principio, además al tener tantas opciones de continuar el desarrollo de software, no nos estancamos, sino que aprovecharemos al máximo el tiempo con el que disponemos ya que a futuro incluso lo podemos desarrollar para fines comerciales. Otra desventaja a aminorar es que debido a que comprendemos en su totalidad el sistema, y a su vez podemos ser usuarios de este, no hay espacio a malas interpretaciones. Debido todo esto, es que la metodología de prototipos es la que de mejor forma calza con lo que necesitamos.

4.3.Definiciones, Siglas y Abreviaciones

TCP: (*Transmission Control Protocol*), en español como "**Protocolo de Control deTransmisión"**, es uno de los protocolos fundamentales en Internet del cual nos permitecrear "conexiones" y establecer las comunicaciones.

Android :es una de las plataformaspara dispositivos móviles touch mas conocida a nivel mundial. Android es un sistema operativo basado en linux, que debido a la gran diversidad de hardware y las amplias opciones de desarrollo dentro de su software, brinda a los usuarios una versatilidad enorme además de su portabilidad.

Base de datos: Base de datos que contiene informaciónón de usuarios para la gestion y/o autentificacion de clientes/usuarios, asi como el registro de nuevos usuarios y almacenamiento de las llaves publicas.

Cliente: es una parte de la arquitectura cliente-servidor que envía datos a un servidor para recibir una determinada información.

Servidor:es la segunda parte de la arquitectura cliente-servidor (basado en socket de Java) la cual recibe datos del cliente, (desde Android) y luego retorna a este mismo, la información que este requiere.

API: Application Programming Interface - Interfaz de Programación de Aplicaciones, sirve para comunicar distintos componentes de software.
PHP: (Personal Home Page Tools o Hypertext Preprocessor) es un lenguaje de programación del lado del servidor, usado para el desarrollo de web dinámicas complementando el desarrollo en HTML.

BD: Base de Datos

Licencia BSD: Licencia que se otorga a los sistemas BSD(Berkeley Software Distribution), es una licencia de software libre permisiva

MySQL:es un sistema de base de datos objeto-relacional, con licencia open source

OpenSource: El software bajo la licencia de Opensource, tienen su codigo abierto al publico general, y puede ser modificado con la idea de mejorarlo

Touchscreen: Pantalla táctil

S.O.: Sistema Operativo

APP: Abreviatura usada para referirse a una "aplicacion".

IP: Internet protocol, en español como "protocolo de internet", direccion que identifica de manera logica y jerarquica una interfaz de red, permite asi la conexion a un servidor mediante un Puerto establecido.

WiFi: Nombre otorgado por la*Wi-Fi Alliance*a la conexión inalámbrica de dispositivos a internet

HTTP: Hyper-text Transfer Protocol, se traduce como Protocolo de Transferencia de Hipertexto.

HTTP Request: Solicitud de información por parte del cliente a un servidor web.

SQLite: Gestion de base de datos independiente relacional soportado para el desarrollo de Android.

Login: Acción de ingresar por medio de un usuario y contraseña, a un sistema, lo cual otorga ciertos privilegios dentro de este.

Netbeans: Es un ambiente de desarrollo multilenguaje (IDE) al cual se programa en Java (Servidor) para el desarrollo de este proyecto.

Apache: Segun *Wikipedia***:** Es un servidor web HTTP de código abierto, para plataformas Unix.

Android Studio: Un ambiente de desarrollo que permite desarrollar aplicaciones moviles basado en el lenguaje Android.

5. Factibilidad

5.1.Factibilidad técnica

El desarrollo de aplicaciones móviles, poco a poco ha ido tomando fuerza, y hoy en día, es una de las herramientas mejor cotizadas, debido a la alta demanda de dispositivos y aplicaciones útiles para el diario vivir. Es así como cada dia aparecen herramientas útiles para el desarrollo de estas.

Para nuestro proyecto utilizo la herramienta Android Studio 2.1 con el desarrollo de la versión 2.2 API 16 proporcionado por Google, publicados bajo la licencia de IntelliJ.

Para la aplicación web (web service) que actua de petición/respuesta se desarrolla bajo el entorno de Sublime Text 2 v2.0.1 para windows 7 y para la creación de la base de datos se utiliza el gestor de base de datos MySQL v8.4 y Apache.

En relación al equipamiento y dispositivos a utilizar, se utiliza un computador y/o laptop de gama media con Sistema Operativo Windows 7 y/o Mac Book OSX El Capitan para actuar como Servido. Las pruebas de la aplicación cliente, se realizan en un dispositivo móvil Samsung Galaxy Note con sistema android 2.3 (Gingerbread).Para la operación del sistema se necesita contar con conexión a internet en todos los dispositivos asi como sus permisos respectivos.

Con respecto a las tecnologías de desarrollo utilizadas, la programación de la aplicación android se realiza principalmente en Java asi como el desarrollo del servidor que también esta basado en Java, junto con la base de datos y la programación PHP, fueron conocimientos adquiridos durante la carrera, que al aplicarlos se consideran de rápido aprendizaje, ya que su investigación es de fácil acceso.

Con todo esto se puede concluir que el sistema es factible técnicamente, ya que se cuenta con el equipamiento y conocimiento necesarios para desarrollarlo sin mayores imprevistos, y guardando su buen funcionamiento y buen uso de la información.

5.2.Factibilidad Operativa

Es ya conocido el fuerte avance técnico de los sistemas Android, actualmente se puede acceder a cualquier información en cualquier parte, es por esto que este sistema, está pensado como una opción a lo que actualmente existe en el ámbito de la transmisión de información inalámbrica. Aunque su base principal es representar la seguridad y la confianza entre los intercambios de mensajes a los alumnos y otros individuos de la universidad mediante un chat grupal, su mayor impacto en cuanto a usabilidad se notara mucho más a la hora de enviar un mensaje privado a un receptor determinado, debido a que se velará encarecidamente a que sea seguro y privado. Muchas veces los alumnos y/o usuarios no saben si alguien puede estar espiando su información a la hora de enviar una información privada como el numero de una tarjeta bancaria, ahora en cambio, la aplicación permitirá que intercambie textos sin que la integridad del mensaje se vea involucrada.

Otra utilidad a la que este sistema apoya es el uso de cifrado, es decir, aunque un administrador opte por “espiar” desde el servidor, estos llegarán encriptado, de manera que sera irreconocible al mismo, de forma que se omite la preocupación y la necesidad de que un agente externo pueda aprovecharse y/o involucrarse.

5.3.Factibilidad Económica

Todo el hardware necesario para el desarrollo de este sistema se encuentra a disposición en la Universidad del Bío-Bío, así como también se optó por el desarrollo del software con programas de licencia libre, por lo cual no existe mayor costo que el humano por parte de mi. Salvo la integridad de los datos.

Beneficios del Sistema

El sistema implantado de forma correcta, se puede traducir a una mayor deseabilidad y confianza de la aplicación a la hora de intercambiar mensajes personales, ya que la información perteneciente a la integridad personal del usuario, se transmite y se obtiene de forma cómoda,a cualquier hora y sin incurrir en burocracia y la desconfianza de espionaje, la cual a veces es incómoda, tanto para funcionarios, como para estudiantes.

El impacto generado por esta aplicación, es difícil de dimensionar en primera instancia, pero ya que se tiene nociones de la cuota de mercado, a futuro, y si es que el sistema es implementado de forma óptima, y a su vez manteniendo la exclusividad dentro de la universidad del Bio Bio, se podrá dimensionar más fácilmenteya que si la demanda o el registro de usuarios tiene un aumento significativo en el uso de la aplicación, puede saturarse el sistema por alto trafico.

5.4 Conclusión de la factibilidad

Luego del análisis presentado para cada una de las factibilidades, se puede concluir que el proyecto es totalmente factible para ser implementado en la Universidad del Bio bio o bien como alternativa aparte para el intercambio de mensajes que no sea WhatsAp o Telegram, entre otros, ya que no muestra ningún tipo de dificultad a la hora de su implementación ni vulnera la integridad de la base de datos, salvo que haya una considerable demanda masiva de usuarios conectados.

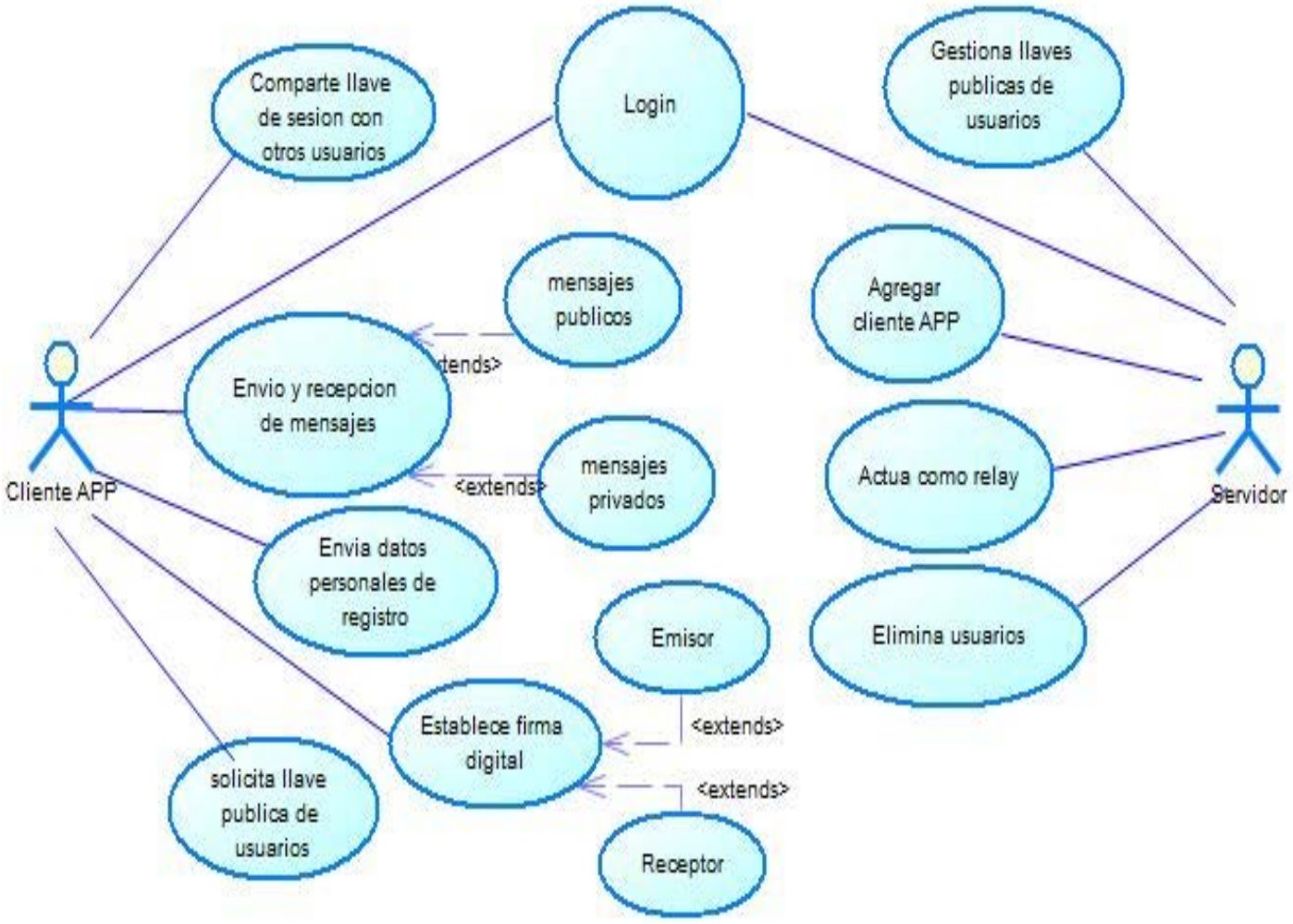
6. Análisis

6.0 Proceso de negócios Futuros.

En este caso, el proyecto que se ha desarrollado tiene un carácter de investigación, por lo tanto este ítem no aplica `para lo desarrollado.

6.1.Diagrama de Casos de Uso

Se presenta el diagrama de casos de uso, el cual muestra las funciones y usos que tiene el sistema.



Ilustracion 3: Caso de uso.

6.1.1. Actores

Servidor:

- Rol: La función del Administrador Servidor es establecer la conexión entre los clientes para que puedan interactuar entre si, asi como la gestión de una lista de usuarios disponibles y los comandos establecidos para dar información a los usuarios. Tambien será el encargado del mantenimiento de usuarios registrados y de sus llaves publicas al usar la aplicacion, tendrá una cuenta única con nombre de usuario, contraseña y llave pública respectiva, será alojada en el servidor con un entorno MySQL-PHP, entorno en el cual podrá Ingresar a su cuenta única (Login).

- Nivel de conocimientos técnicos requeridos: El conocimiento computacional y técnico debe ser alto, debido al manejo de base de datos.

- Nivel de privilegios: Tiene nivel alto de privilegios, capaz de agregar, modificar y eliminar información, tales como los usuarios y su llave pública.

Cliente APP:

- Rol: Hace uso de la aplicación para dispositivos móviles, su función es conectarse al servidor e intercambiar mensajes, sea públicos o privados, asi como también solicitar información o activar comandos.

- Nivel de conocimientos técnicos requeridos: El nivel de conocimiento tecnico y computacional son medios.

- Nivel de Privilegios: Tiene sólo privilegios de lectura y escritura, lo que significa que de nivel medio,ya que sólo necesita iniciar la APP, registrarse en caso de que no este registrado, y luego comenzar a intercambiar mensajes.

6.1.2. Especificación de casos de uso

1. Login

Nombre	Login
Descripción	Permite autentificarse en el sistema, de tal manera que otorgue los permisos necesarios para usar la app.
Pre-condición	- El usuario debe estar creado.
Post-Condición	- El usuario estará dentro del sistema
Flujo normal	
Accion del Actor	Acción del Sistema
1.- Ingresar al sistema	
2.- Rellenar campo de usuario y contraseña	
	3. Validar Datos ingresados.
	4. Conecta el usuario ingresado a la app y pide establecer una conexion a cualquier servidor externo.
Flujo Alternativo	
Accion del Actor	Accion del Sistema
	3a. Datos Incorrectos
	4a. No establece ninguna conexion de la APP con el servidor.
Post-Condicion	- No se pudo ingresar al Sistema.

Tabla 2: Caso de Uso Login

2. Enviar datos personales de registro.

Nombre	Envia datos personales de registro.
Descripción	Permite al usuario de la APP, registrarse en el sistema, de tal manera que asi pueda tener permisos necesarios para usar la app.
Pre-condición	- La app debe estar abierta y en la pantalla de registro
Post-Condición	- El usuario estará registrado en el sistema
Flujo normal	
Accion del Actor	Acción del Sistema
1.- Solicita la pantalla de registro.	
2.- Rellenar campo de usuario y contraseña	
	3. Valida Datos ingresados.
	4. Verifica si no hay usuario duplicado y registra los datos al Sistema.
Flujo Alternativo	
Accion del Actor	Accion del Sistema
	3a. Datos Incorrectos o duplicados
Post-Condicion	- No se pudo registrar al usuario.

Tabla 3: Caso de Uso Enviar datos personales de registro.

3.- Envio y recepción de mensajes.

Nombre	Envio y recepcion de mensajes
Descripción	Permite al usuario enviar mensajes publicos en un chat global o enviar mensajes privados a un usuario en particular, segun lo que prefiera.
Pre-condición	- El usuario debe estar autentificado y conectado al servidor.
Post-Condición	- El usuario envio el mensaje publico o privado.
Flujo normal	
Accion del Actor	Acción del Sistema
1.- Escribir mensaje publico o activar comando para mensaje privado.	
2.- Pulsar boton de Enviar	
	3. Si hay conexion establecida al servidor, recibe mensaje y determina si es publico o privado.
	4. Dependiendo del tipo de mensaje, entrega el mensaje publico a los demas usuarios o el privado al usuario ingresado previamente.
Flujo Alternativo	
Accion del Actor	Accion del Sistema
1.- Escribir mensaje publico o activar comando para mensaje privado.	
2.- Pulsar botón enviar	
	3.- Si no hay servidor establecido, no recibe por lo tanto no redicciona ningún mensaje.
Post-Condicion	- No se entrego ningun mensaje.

Tabla 4: Caso de Uso Envio y recepcion de mensajes

4.- Gestiona llaves Publicas de usuarios

Nombre	Gestiona llaves publicas de usuarios.
Descripción	Permite al Sistema web almacenar un registro de llaves publicas por cada usuario a medida que el usuario se registra.
Pre-condición	- El usuario debe haber registrado sus datos previamente.
Post-Condición	- Se disponen las llaves publicas en caso de ser utilizadas.
Flujo normal	
Accion del Actor	Acción del Sistema
1.- Ingresar al sistema	
2.- Rellenar campo de usuario y contraseña	
	2. Validar Datos ingresados.
	3. Se dispone de una llave publica para el usuario.
Flujo Alternativo	
Accion del Actor	Accion del Sistema
	3a. Datos Incorrectos
Post-Condicion	- No se establecio ninguna llave publica.

Tabla 5: Caso de Uso Gestiona llaves publicas de usuarios

5.- Agregar cliente App

Nombre	Agregar ClienteApp
Descripción	Permite al Servidor registrar un usuario.
Pre-condición	- El usuario debe haber abierto la app y haber ingresado sus datos correctamente.
Post-Condición	- El usuario estará registrado dentro del sistema
Flujo normal	
Accion del Actor	Acción del Sistema
	1. Validar Datos ingresados.
	2. Si no existen duplicados, registra al usuario al Servidor.
Flujo Alternativo	
Accion del Actor	Accion del Sistema
	3a. Datos duplicados e incorrectos.
Post-Condicion	- No se pudo registrar usuario al Sistema.

Tabla 6: Caso de Uso Agregar cliente App

6.- Actua como Relay

Nombre	Actua como Relay
Descripción	El servidor espera el ingreso de usuarios registrados para luego actuar como intermediario en el intercambio de mensajes.
Pre-condición	- El usuario debe estar creado y haber ingresado la IP del servidor al que conectarse previamente.
Post-Condición	- El usuario estará dentro del servidor.
Flujo normal	
Accion del Actor	Acción del Sistema
1.- Abre la APP.	
2.- Rellenar campo ingresando sus datos y la IP del servidor al que desea conectarse.	

	3. Recibe los datos.
	4. Conecta al usuario ingresado a la app y establece una conexion para intercambiar mensajes.
Flujo Alternativo	
Accion del Actor	Accion del Sistema
	3a. Datos Incorrectos
	4a. No establece ninguna conexion de la APP con el servidor.
Post-Condicion	- No se pudo conectar a ningun servidor.

Tabla 7: Caso de Uso Actua como Relay

7.- Elimina usuarios

Nombre	Elimina Usuarios.
Descripción	El servidor elimina a un usuario registrado en el servidor.
Pre-condición	- El usuario debe existir, o al menos debe existir un usuario registrado en el servidor.
Post-Condición	- El usuario estará eliminado del servidor.
Flujo normal	
Accion del Actor	Acción del Sistema
1.- Seleccionar usuario para eliminar	
	3. Elimina al usuario.
Flujo Alternativo	
Accion del Actor	Accion del Sistema
	3a. Si no existe usuario creado o no coincide.
	4a. Datos Incorrectos
Post-Condicion	- No se pudo eliminar al usuario del servidor.

Tabla 8: Caso de Uso Elimina usuários.

8.- Comparte llave de sesión con otros usuarios.

Nombre	Comparte llave de session con otros usuarios
Descripción	Durante la transmission de mensajes privados, permite compartir su llave de session a otro usuario, de modo que, pueda usarse para cifrar/descifrar sus mensajes.
Pre-condición	- Tanto el usuario emisor como el receptor debe existir y estar conectado en el servidor.
Post-Condición	- El usuario comparte su llave de session, ambos lo poseen.
Flujo normal	
Accion del Actor	Acción del Sistema
1.- Inicia comando para mensaje privado.	
	2.-Se prepara para recibir la llave de sesión encriptada.
3.- Establece nombre del receptor y envia el mensaje al nombre correspondiente	
	4.- Recibe y reenvia la llave de sesion encriptada al usuario con el nombre ingresado previamente.
Flujo Alternativo	
Accion del Actor	Accion del Sistema
	3a. Si no existe usuario creado o no esta en linea.
Post-Condicion	- Se perdió el mensaje.

Tabla 9: Caso de Uso Comparte llave de sesion con otros usuarios.

9.- Solicita llave pública de usuarios

Nombre	Actua como Relay
Descripción	El servidor le otorga al cliente APP la llave publica de un usuario determinado, para proceder a cifrar la llave de session.
Pre-condición	- El usuario debe estar en linea y haber ingresado la IP del servidor al que conectarse previamente.
Post-Condición	- El usuario tendra la llave publica del receptor.
Flujo normal	
Accion del Actor	Acción del Sistema
1.- Inicia comando para mensaje privado.	
2.- Ingresa nombre del receptor y el mensaje que entregará.	
	3. Recibe el nombre del receptor.
	4. Le envia la llave publica del receptor ingresado previamente.
Flujo Alternativo	
Accion del Actor	Accion del Sistema
	3a. Si no existe el receptor.
	4a. No retorna ninguna llave pública.
Post-Condicion	- No se pudo solicitar la llave pública.

Tabla 10: Caso de Uso Solicita llave publica de usuarios.

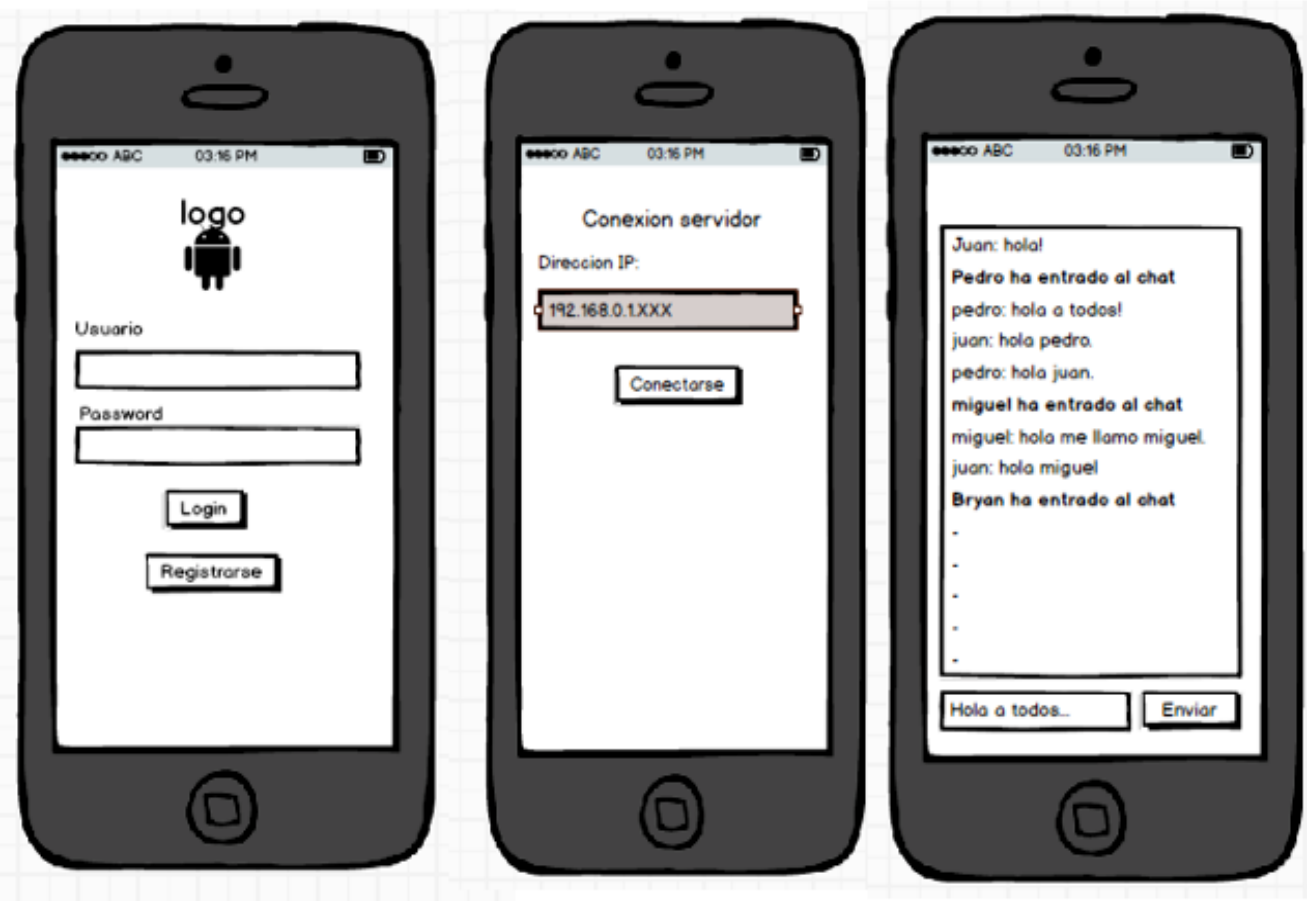
6.2.Modelamiento de datos

Como se ha explicado en los puntos anteriores a este documento, el trabajo realizado esta orientado a un tema de investigación por lo que no presenta ningún modelamiento de datos, sin embargo, y a modo de proposición, mas adelante se presentará, durante los propotipos, el modelo entidad relación a medida que se transcurre el desarrollo y el análisis según se requiera.

6.3.Descripción global del producto.

6.3.1. Interfaz de usuario:

La interfaz de usuario se compone de diferentes etapas según lo que se requiera al usar la aplicación.El formato visual del dispositivo móvil tendrá una arquitectura como se muestra en la siguiente gráfica:



Ilustracion 4: Interfaz global del desarrollo móvil.

Como se aprecia en la figura, primero despliega un login para la autentificación del usuario, si es correcto procede a ingresar una dirección IP valida para establecer un relay en el servidor, se puede optar por hacerlo automáticamente pero se da preferencia a la comodidad del usuario para elegir entre diferentes servidores disponibles si los hay, finalmente entra a la ultima ventana de la aplicación, donde podrá intercambiar mensajes públicos o privados, según sea el caso.

La siguiente interfaz es simple, comoda y solo cumple la función de una ventana adicional para rellenar el formulario y registrar al usuario en el sistema para tener permisos de acceso a la aplicación.

Ilustracion 5: Interfaz movil de Login

6.3.2. Interfaz de Hardware

Para la implementación del prototipo, y del sistema en general, se necesita los siguientes equipos:

1. Servidor
2. Dispositivo Móvil Android y conexión a Internet

6.3.3. Interfaz de Software

Para la implementación del sistema se requiere de los siguientes softwares:

a) Servidor
- Sistema Operativo Mac Osx El Capitan.
- Servidor Apache
- MySQL 8.4
-TCP (Sockets)
- Desarrollo y dependencia de lenguaje Java.

b) Cliente
- Android 5.
-Java 8

6.3.4. Interfaces de comunicación

Las interfaces de comunicación utilizadas son:

- HTTP, protocolo utilizado por el entorno web.

6.3.5. Atributos del producto

Funcionalidad – Seguridad: El sistema debe mantener la seguridad de la información, para esto se implementa un control de usuarios y contraseña, además de no permitir usuarios duplicados.

Usabilidad – Comprensibilidad: Otro de los atributos del producto es fácil manejo, por esto debe presentar mensajes de error adecuados, botones de acceso claros para realizar las operaciones necesarias.

Portabilidad – Adaptabilidad: El servidor puede ser utilizado en cualquier computador con conexión a internet, no así la aplicación cliente, que por ahora, solo se manejara en dispositivo con S.O Android, esto deja fuera a los usuarios de otros tipo de dispositivos móviles basado en iOS y Windows Phone.

Mantenibilidad – Estabilidad: El servidor debe estar siempre activo para el correcto funcionamiento de la aplicación, así como también la aplicación requerirá de permisos para la 34 ndroid 34 a internet y flujo de datos permitido. El uso de Usuarios registrados para la autentificación facilita enormemente la tarea de mantención, además de una retroalimentación del sistema.

7. Especificación de Requerimientos de Software y prototipos

7.1 PROTOTIPO 1

7.1.1. Alcances

El primer prototipo cuenta con un servidor implementado para actuar de relay entre los clientes, la cual también se cuenta con una APP de Android desde el cliente de manera que permita comunicarse entre ellos. Toda la información de los mensajes y/o de los nombres de usuario se desplegará en el dispositivo móvil.

El prototipo no proporciona imágenes a la descripción de las infraestructuras, solo solo se intenta priorizar la comunicación de datos entre los clientes, además el margen de error observado en este prototipo es considerable. Tampoco tiene la facultad de ingresar o controlara los usuarios.

7.1.2. Objetivo Del Software

7.1.2.1. Objetivo general

El objetivo de este prototipo es principalmente para poner en práctica las nuevas tecnologías utilizadas, una de ellas es la programación Android y de socket mediante java, incluyendo la utilización de un servidor y probar la conexión entre clientes, asi como probar y gestionar la respuesta/petición del mismo, los tiempos que toma realizar estas conexiones, además tener una idea de cuán cómodo resulta este tipo de aplicaciones para un usuario común.

7.1.2.2. Objetivos específicos

- Montar un servidor que contenga sockets disponibles para actuar de relay entre diferentes clientes.

- Programar una aplicación cliente para el sistema operativo android, capaz de realizar la conexión entre cliente y servidor y ser capaz de enviar sus mensajes de texto tanto públicos como privados.

7.1.3.Requerimientos Específicos

7.1.3.1 Interfaces externas de entrada

Para este primer prototipo solo se necesita un tipo de dato externo de entrada al sistema, el cual es la IP de acceso al servidor, que se obtiene mediante la configuración del mismo, esta configuracion tiene un formato de IP:puerto.

ID	Nombre	Detalle de datos
IE_01	IP	Direccion IP:Puerto
IE_02	Nombre (Nick)	Nick del usuario.

Tabla 11: Interfaz de Entrada

7.1.3.2. Interfaces externas de Salida

Luego de recibir los datos de acceso al servidor, el mismo servidor realiza las consultas pertinentes estableciendo una conexión a el,solicitando un nombre (Nick) para usarse y genera la siguiente interfaz de salida:

ID	Nombre	Detalle de datos	Medio de salida
IS_01	Conexión	Bienvenida	Pantalla cliente

Tabla 12: Interfaz de Salida

7.1.4. Diseño y Construcción

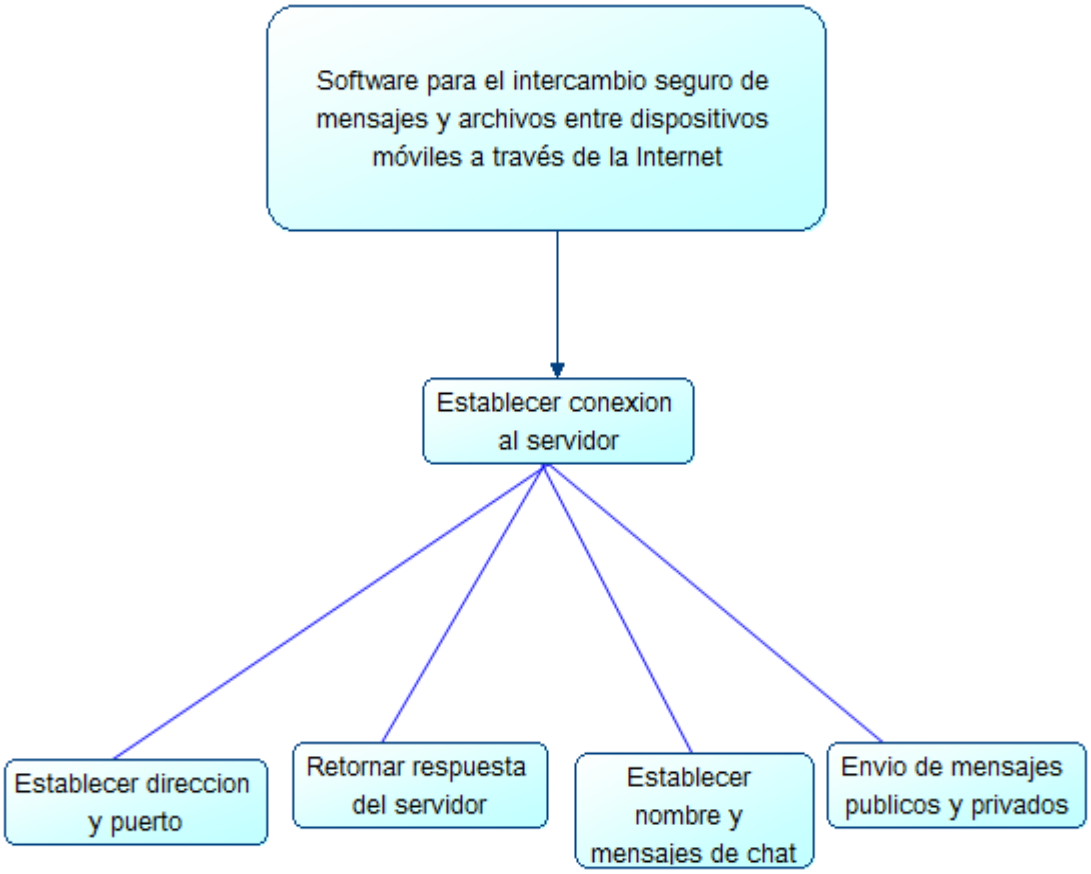
7.1.4.1. Diseño Físico de la Base de datos.

No se cuenta con un diseño que se presente para el prototipo 1, salvo el servidor, el que actualmente solo tiene los siguientes atributos estimados para establecer una conexion y aceptar clientes

Atributos:

- IP: (Direccion IP)
- Puerto (puerto establecido para la conexion)
- name (nombre del cliente)
- line (nombre del texto enviado por cliente)
- lista_user (lista de usuarios conectados)

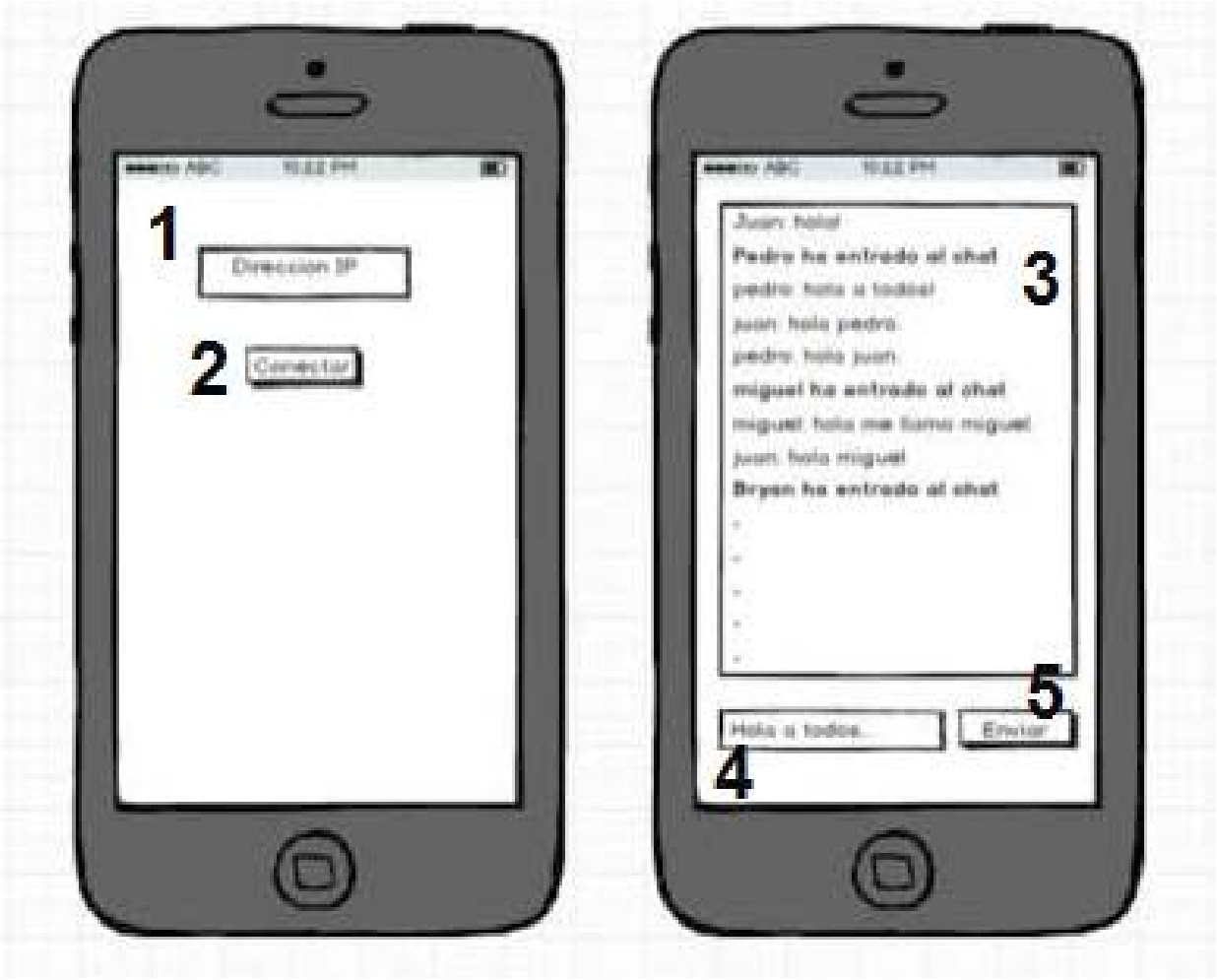
7.1.4.2. Diseño de la arquitectura funcional



Ilustracion 6: Diseño arquitectura funcional protótipo 1.

7.1.4.3. Diseño Interfaz y navegación

Se presenta la interfaz de navegación que presenta el prototipo 1 para el usuario cliente, esto es en el dispositivo móvil.



Ilustracion 7: Interfaz movil del protótipo 1.

Como se aprecia, no hay una robusta navegación manual excepto el ingreso de la dirección IP y conectarse, es solo una muestra de información en texto plano.

1) **Direccion IP:** Permite el ingreso de una direccion IP valida para conectarse a un servidor
2) **Conectarse:** Boton el cual al pulsar y validar la direccion IP, navega hasta la siguiente pantalla para visualizar mensajes y clientes.
3) **Pantalla de mensajes:** Una lista enumerada de todos los mensajes publicos enviados por los usuarios.
4) **Panel de escritura de mensajes:** Espacio en blanco que permite escribir tu propio mensaje.
5) **Enviar:** Boton que permite el envio de los mensajes introducidos.

7.1.4.4. Especificación de módulos

Módulos:

Nombre Módulo: Conectar servidor..			
Parámetros de entrada		**Parámetros de Salida**	
Nombre: Direccion servidor, puerto.	**Tipo de dato:**String	**Nombre:**respuesta servidor, puerto 222. En caso de error: “Error de conexión con el servidor.”	**Tipo de dato:**String
Nombre Módulo: Establecer nombrede usuario. (Nick)			
Parámetros de entrada		**Parámetros de Salida**	
Nombre: Name.	**Tipo de dato:** String	**Nombre:**TextView Android para visualizar la respuesta.en caso de error: “Nombre no puede contener @”.	**Tipo de dato:**TextVi ew(St ring)
Nombre Módulo: Intercambiar mensajes.			
Parámetros de entrada		**Parámetros de Salida**	
Nombre: Line.	**Tipo de dato:** String	**Nombre:**TextView Android para visualizar la respuesta.	**Tipo de dato:**TextVi ew(St ring)
Nombre Módulo: Retornar respuesta del servidor			
Parámetros de entrada		**Parámetros de Salida**	
Nombre: Respuesta servidor	**Tipo de dato:** String	**Nombre:**TextView Android para visualizar la respuesta.	**Tipo de dato:** TextView(St ring)

Tabla 13: Especificacion de módulos.

7.1.5 Evaluación y Puebas.

7.1.5.1. Elementos de prueba

A continuación se harán pruebas a funcionalidad de los distintos módulos ya definidos y a la suma de módulos que conforman la aplicación Android Cliente. Para Identificarlos definimos un código de Identificación para cada elemento de prueba.

1. Establecer dirección y puerto.
2. Retornar respuesta del servidor
3. Establecer nombre y mensaje de chat
4. Envio de mensajes públicos y privados
5. Conexion Total de los módulos

7.1.5.2. Especificación de pruebas.

ID	Caracter ísticas a probar	Nivel de prueba	Objetivo de la Prueba	Enfoque para la definición de casos de prueba	Técnicas para la definición de casos de prueba	Actividades de prueba	Criterios de cumplimiento
1	Funcionalidad y desempeño	Unidad	Probar la configuracion	Caja Negra	Comparar con respuesta esperada	a- Ingreso correctamente de los datos de acceso al servidor. b- Datos erroneos	- Muestra dirección y puerto.Si formato es correcto o incorrecto.
2	Funcionalid ad	Unidad	Existe el servidor con los datos ingresados	Caja Negra	Conexión exitosa con el servidor según corresponda	a- Pruebas con conexión adecuada a internet b- prueba con conexión débil a internet.	- El servidor retorna conexión exitosa si correspond e - El servidor retorna Conexión Fallida si no corresponde.

3	Funcionalidad .	Unidad	Verificar que el cliente recibe la respuesta del servidor adecuadamente y establecer su Nick.	Caja Negra.	Comparar si la respuesta vista en el cliente es igual a la que respondió en servidor en la prueba 2.	a- Prueba de ingreso de Nick con @ b- Prueba de Nick sin @.	a- Dar la bienvenida al usuario sin uso de su nombre con @ b-Nick incluye @, no se permite ese uso.
4	Funcionalidad	Unidad	Verificar que hay conexión y hay flujo de datos para la transmisión de mensajes.	Caja Negra.	Comparar con respuesta esperada.	a- Prueba de ingreso de mensajes publicos b- Prueba con mensaes privados	- Que lo mostrado en el TextView de la APP, sea idéntico a Loenviado desde la APP.

5	Funcionalidad y Desempeño	Sistema	Verificar si el sistema entrega la respuesta esperada, con la conexión establecida entre servidor y la APP para luego intercambiar mensajes entre multiples clientes.	Caja Negra	Comparar la información entregada desde la APP al servidor donde se establece la conexión.	a.- Ingresar datos de dirección de acceso al servidor desde la APP. b.- Abrir la aplicación y solicitar la petición/respuesta. c.- intercambar mensajes públicos y activar comando para un privado	- Que la información mostrada por la APP coincida con la Información de respuesta del servidor. - Que se configure el servidor y este listo para actuar como relay. - Que distintos usuarios conectados al servidor pueden intercambiar mensajes y se muestre correctamente.

Tabla 14: Especificacion de pruebas

7.1.5.3. Responsables de las pruebas

El principal responsable de realizar las pruebas en este prototipo de modo general, no es mas que el autor del informe de la Tesis, Bryan Ramirez.

7.1.5.4. Calendario de pruebas

Las pruebas fueron realizadas luego de concluido el primer prototipo, las cuales se detallan en la siguiente tabla:

Nombre de Prueba	**Fecha**
Establecer direccon y puerto.	27 de mayo de 2016
Retornar respuesta del servidor	27 de mayo de 2016
Establecer nombre y mensaje de chat.	27 de mayo de 2016
Envio de mensajes publicos y privados	27 de mayo de 2016
Conexion Total de los módulos	27 de mayo de 2016

Tabla 15: Calendario de pruebas

7.1.5.5. Detalle de las pruebas

Pruebas de Unidad

1.- Establecer dirección y puerto: Este módulo será probado con WIFI, se establece una dirección IP valida y existente, se evaluara el uso del wifi activado o desactivado tanto para local o remoto en una APP Android.

ID Caso De Prueba	Caracterísitcas a Probar	Datos de Entrada	Salida esperada	Salida Obtenida	Éxito / Fracaso	Observaciones
1.a	Funcionalidad y Desempeño	Wifi, Direccion IP (local y remoto).	Wifi activado.	- Sent 192.168.0.xxx	Éxito	Con el wifi activado, se recibe respuesta al ingreso.

1.b	Funcionalidad y Desempeño	Wifi, Direccion IP (local y remoto)	Wifi desactivado	-ERROR	Fracaso	Tener wifi desactivado o no podrá esperar una respuesta ni se recibe ningún ingreso.

2.- Retornar respuesta del servidor: Para lograr la conexión con el servidor, el dispositivo móvil debe estar conectado a internet, y la aplicación debe poder acceder a estos permisos. Ademas el servidor debe estar Funcionando en linea, con un módulo de pruebas que ante la entrega de una respuesta correcta retorne “Done”.

ID Caso De Prueba	Características a Probar	Datos de Entrada	Salida esperada	Salida Obtenida	Éxito / Fracaso	Observaciones
2.a	Funcionalidad	String IP correcto, Wifi activado.	Conecting Sent 192.168.0.xxx”	“Done”	Éxito	Ingresando formato correcto y existente, recibimos respuesta y conexion
2.b	Funcionalidad	String IP incorrecto, Wifi activado.	“Error de conexión con el servidor”	“ERROR”	Fracaso	Si el formato de ingreso de IP es incorrecta, no recibe conexion.

3.- Establecer nombre y mensaje de chat: Verificamos si la consulta responde adecuadamente en el servidor. Esta prueba se realizará solo en el servidor, por lo que se necesita de un Módulo de pruebas que ejecute un printf que muestre en pantalla la respuesta del servidor ante ciertos parámetros que enviamos desde la APP..

ID Caso De Prueba	Características a Probar	Datos de Entrada	Salida esperada	Salida Obtenida	Éxito / Fracaso	Observaciones
3.a	Funcionalidad	String name String Line. .	“”Cliente Name esta conectado Mensaje: line””	“”Cliente Name esta conectado Mensaje: line””	Éxito	Si registramos un nombre y una línea de mensaje de saludo se podrá usar la APP
3.b	Funcionalidad	String name String Line	“”	“”	Fracaso	Esta respuesta vacia no registra al cliente ni tampoco permite mensajes correctos.
3.c	Funcionalidad	String@name String Line	“El nombre no debe contener @”	“El nombre no debe contener @”	Fracaso	Un nombre no debe contener @. Por lo que se pedirá que reintente con otro Nick.

4.- Envio de mensajes públicos y privados: Este módulo será probado siempre con el WIFI activado, no es necesario que hayan muchos clientes conectados, pero al menos debe existir 2 clientes conectados. Las principales diferencias entre es que asi se pueden visualizar los cambios que ocurren con el otro cliente nuevo, como por ejemplo la notificación del ingreso de un nuevo cliente o los comandos de mensajes públicos y privados.

ID Caso De Prueba	Características a Probar	Datos de Entrada	Salida esperada	Salida Obtenida	Éxito / Fracaso	Observaciones
4.a	Funcionalidad	String Line= “Hola a todos”. .	“<name> Hola a todos”	“<name> Hola a todos”	Éxito	Si enviamos un mensaje de saludo o publico, ser{a visible a todos los clientes conectados.
4.b	Funcionalidad	String Line= “”.	“<name> “””	<name> “””	Exito	Si enviamos un mensaje de saludo vacio o publico vacio, se visualizara como mensaje vacio.
4.c	Funcionalidad	String @name	“El nombre no debe contener @”	“El nombre no debe contener @”	Fracaso	Un nombre no debe contener @. Por lo que se pedirá que reintente con otro Nick.

4.d	Funcionalidad	String line="/prv"	Activo de comando para mensaje privado	"Ingrese Nick para enviar MP:"	Exito	Al envíar un comando "/Prv" activo el comando para enviar mensajes privados.

Pruebas de Sistema

5.- Interacción entre todos los módulos: Para esto, el servidor debe estar en línea. Y el dispositivo móvil debe tener la aplicacion Android instalada, ademásde obviar que debe tener el WiFi activado y con conexión a internet.

ID Caso De Prueba	Características a Probar	Datos de Entrada	Salida esperada	Salida Obtenida	Éxito / Fracaso	Observaciones
5.a	Funcionalidad	String IP="192.168.0.107" . .	Sent: 192.168.0.xxx Done	Sent: 192.168.0.xxx Done	Éxito	Se establece una conexión al servidor con una IP valida y existente.
5.b	Funcionalidad	String Line="Hola mundo" String Line="Que tal?" String Line="bye"	"<name>Hola mundo "<name>que tal? "<name>Bye"	"<name>Hola mundo "<name>que tal? "<name>Bye"	Exito	Mensajes públicos tanto vacios como con texto se visualizan en el chat. .

5.c	Funcionalidad	String Line="/Prv" String nombre String mensaje	Ingrese Nick: Ingrese mensaje privado:	Ingrese Nick: Ingrese mensaje privado:	Exito	Activo comando de mensaje privado y envía el mensaje al Nick establecido.
5.d	Funcionalidad	String Line="/Prv" String nombre String mensaje	Ingrese Nick: Ingrese mensaje privado:	Ingrese Nick: Ingrese mensaje privado:	Fracaso	Activo comando de mensaje privado y si nombre ingresado es incorrecto o no existe Nick en línea, no se envía el mensaje.

Tabla 16: Detalle de pruebas

7.1.5.6. Conclusiones de las pruebas

Las pruebas realizadas nos muestran que sin tener el WIFI activado e internet, no es posible acceder de ninguna manera a un servidor remoto para conectarse y establecer una conexión, por lo que también debemos tener cuidado con la falsa alarma, pues no solo hay datos incorrectos sino también debemos procurar tener una IP valida, ya que si nos equivocamos en el ingreso de un numero nos estamos referiendo a otra dirección, por lo tanto, a un servidor equivocado. Por otra parte también concluimos las pruebas sobre el ingreso de mensajes públicos entre los usuarios y con la activación de un comando especial dedicado a los mensajes privados para que llegue solo al destino respectivo.

7.2 PROTOTIPO 2

7.2.1. Alcances

Este prototipo tiene como principal característica añadida, el manejo de usuarios (Login) [7] y la administración de llaves públicas, lo cual trae consigo la utilización de un servidor MySQL en la que permitirá registrar usuarios asi como las llaves publicas o eliminarlas.

Otro punto a considerar es que en este segundo prototipo se mejoró notablemente el diseño y la interfaz de la APP para el uso del cliente.

7.2.2. Objetivo del Software

7.2.2.1. Objetivo general

El objetivo de este prototipo es alcanzar el 100% del objetivo general propuesto, vale decir, tener en funcionamiento ambas partes tanto del servidor como cliente de forma que estén correctamente configuradas, establezcan el relay y el intercambio de mensajes, en este ámbito se aplicara el uso de criptografía [8] y el login.

7.2.2.2. Objetivos específicos

-Implementar una técnica de criptografía simétrica para la encriptación/des encriptación de mensajes de texto e imágenes.

-Diseñar e implementar una aplicación cliente para un dispositivo inalámbrico que permita el intercambio seguro de una llave de criptografia simetrica entre dos usuarios

-Desarrollar una aplicación que permita el almacenamiento seguro de llaves de criptografia asimetrica.

-Diseñar una interfaz desde la aplicación móvil que permita a los usuarios registrarse para acceder al uso de la aplicacion.

-Diseñar un login desde la aplicación móvil que permita a los usuarios registrados loguearse para acceder al uso de la aplicación e intercambiar mensajes.

7.2.4.Requerimientos Específicos

7.2.4.1 Interfaces externas de entrada

Estas son las entradas con las que contará el sistema.

a) Cliente

ID	Nombre	Detalle de datos
IE_01	Nombre, password	Datos del usuario para loguearse al usar la aplicación
IE_02	Direccion IP	Dato de la dirección IP que permitirá al usuario conectarse a un servidor.

Tabla 17: Interfaz entrada cliente 2

b) Servidor

ID	Nombre	Detalles de datos
IE_01	Datos de usuario, llave publica	Datos de los usuarios registrados, Nombre de usuario, password y llave publica almacenada en un servidor MYSQL.
IE_02	Datos de Usuario, llave privada.	Dato del usuario que contiene la llave privada, esta no se almacena en un servidor remoto, sino propio como SQLite. (independiente en su móvil)

Tabla 18: Interfaz entrada servidor 2

7.2.4.2. Interfaces externas de Salida

Al igual que la interfaz de entrada, la interfaz de salida se divide en 2 partes

a) Cliente

ID	Nombre	Detalle de datos	Medio de salida
IS_01	Login	Nombre, password correctos o incorrecto	Pantalla cliente
IS_02	Saludo	Bienvenida	Pantalla cliente

Tabla 19: Interfaz salida cliente 2

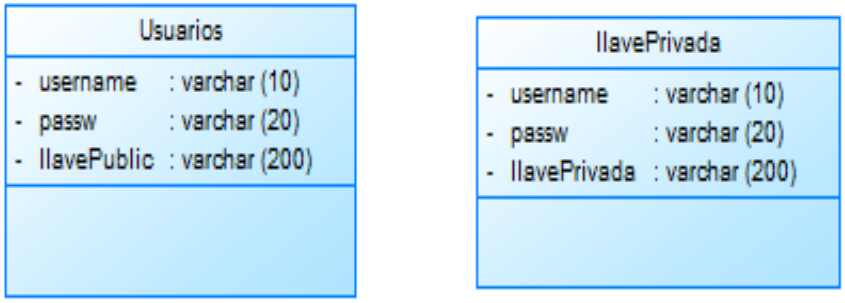
b) Servidor

ID	Nombre	Detalle de datos	Medio de salida
IS_03	Conexion	Direccion IP	Pantalla servidor
IS_04	nick	Nombre	Pantalla servidor
IS_05	Mensajes	Line	Pantalla servidor.

Tabla 20: Interfaz salida servidor 2

7.2.5. Diseño y Construcción

7.2.5.1. Diseño Físico de la Base de datos.



Ilustracion 8: Modelo entidad-Relacion

Como se puede ver, se agregó una tabla en la base de datos que permitirá el registro de usuarios y la administración de llaves públicas. Asi como también una tabla “llavePrvada” que estará almacenado en SQLITE. Ambas tablas no se relacionan de ninguna forma, son totalmente independientes.

DroidLogin: Es el nombre de la base de datos que contiene a la tabla Usuarios.

Usuarios: Esta tabla contiene los datos de conexión de los usuarios del sistema.

Atributos

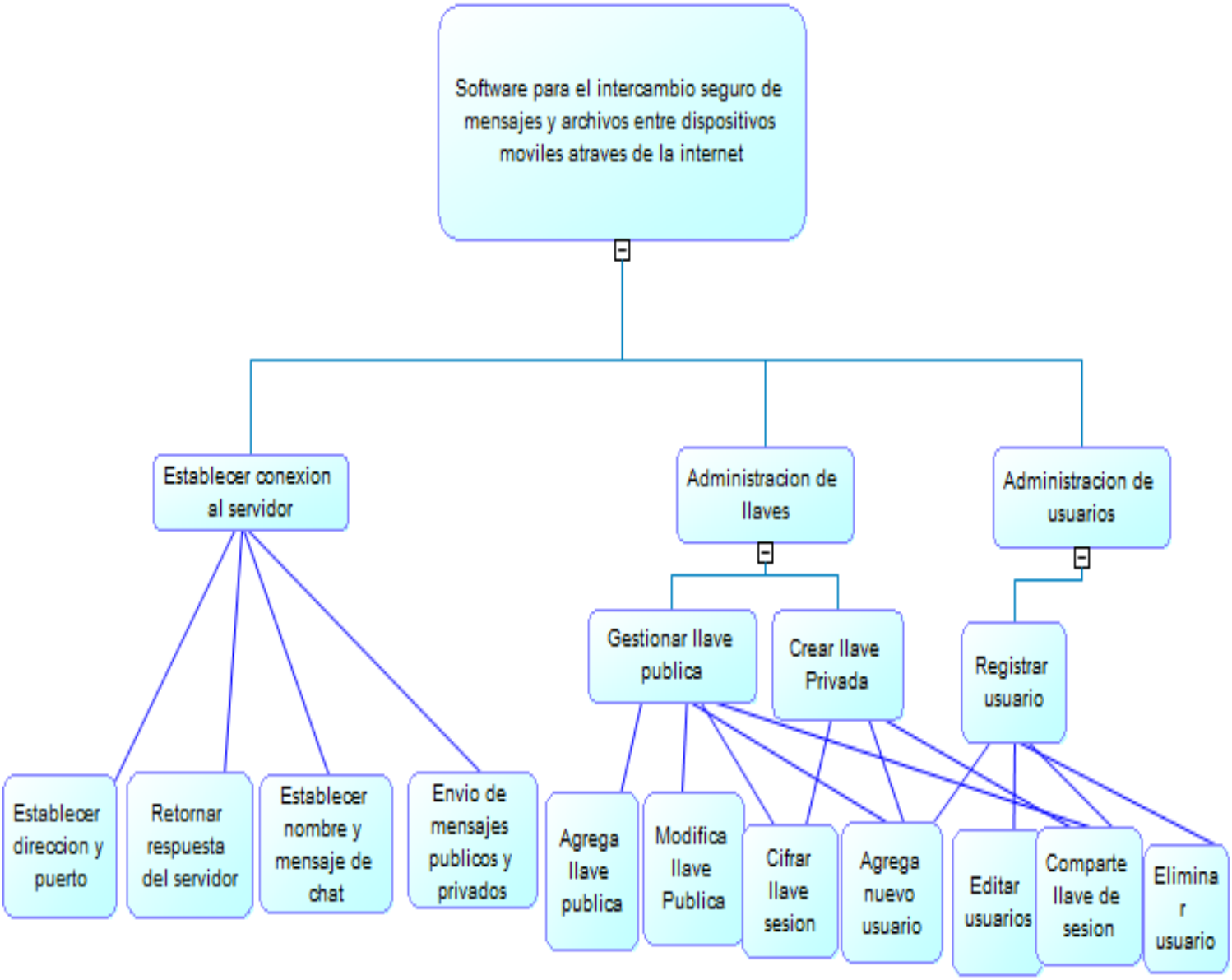
- username
- passw
- llavePublic

llavePrivada: Esta tabla contiene los datos de la llave privada del usuario para compartir la llave de sesión.

Atributos

- username
- llavePrivada

7.2.5.2. Diseño de la arquitectura funcional



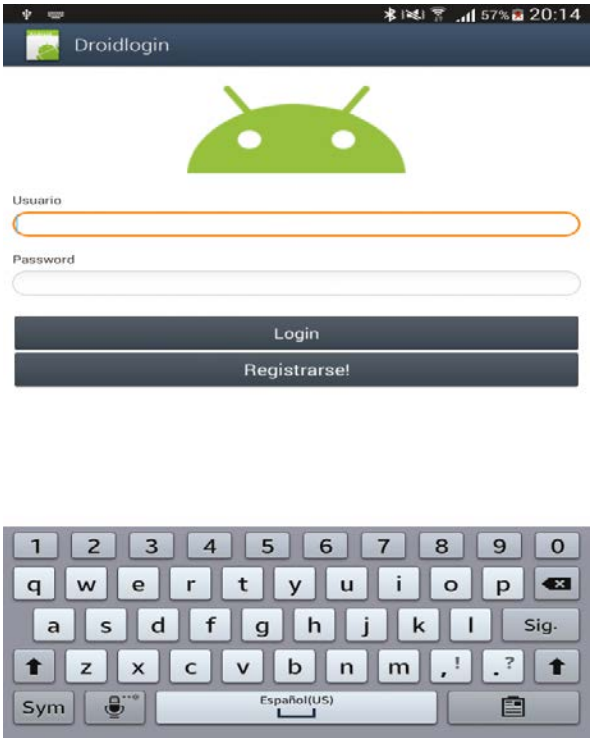
Ilustracion 9: Diseño arquitectura funcional protótipo 2.

7.2.5.3. Diseño Interfaz y navegación

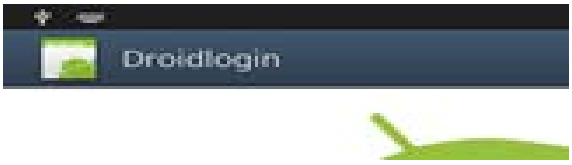
En este prototipo, basado en las funciones del prototipo 1, separamos la interfaz y navegación en 2 partes, como anteriormente se ha señalado, estas partes corresponden a la aplicación Android y servidor

Aplicación Android

Se presenta la interfaz correspondiente al Usuario cliente, ya conocemos anteriormente, por el prototipo 1, las funciones de cada estructura, por lo cual, en este caso vamos a detallar su proposito y funclonalidad, ademas podemos observar un login para el usuario en su dispositivo móvil:

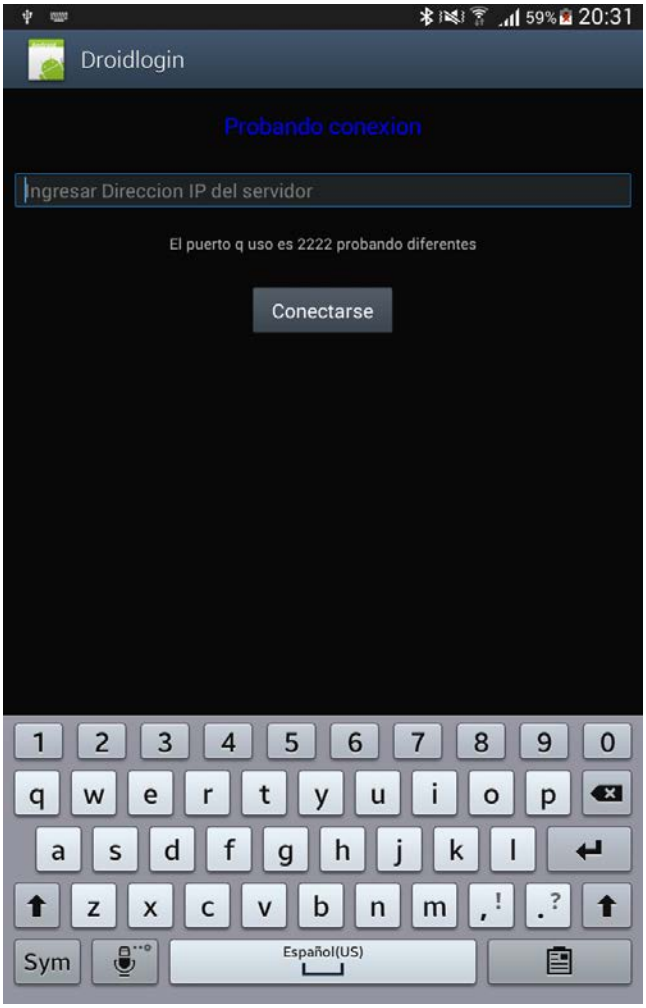


Ilustracion 10: Interfaz Login.



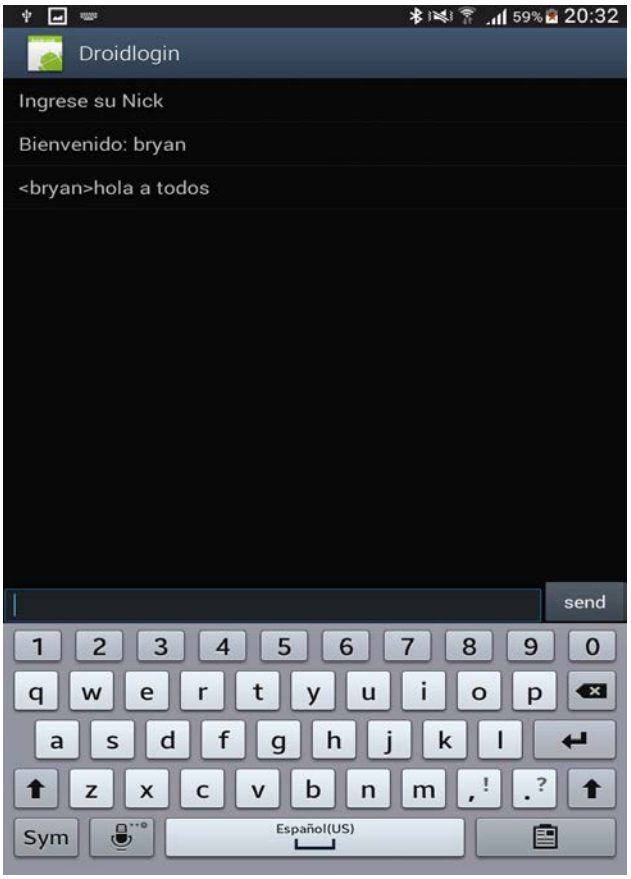
Ilustracion 11: Interfaz Titulo.

En la parte superior de la aplicación del login se puede ver el título/nombre_de_aplicación, esto emula un “head” como aplicación móvil.



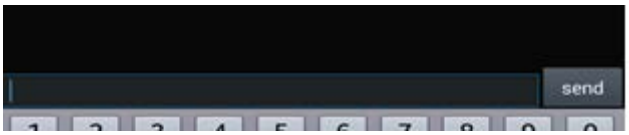
Ilustracion 12: Interfaz contenido medio.

El contenido medio contiene la información referente a la dirección IP que necesita para establecer una conexión al servidor. Este ocurre una vez que el usuario se haya logueado correctamente para luego proceder a la interfaz del chat:



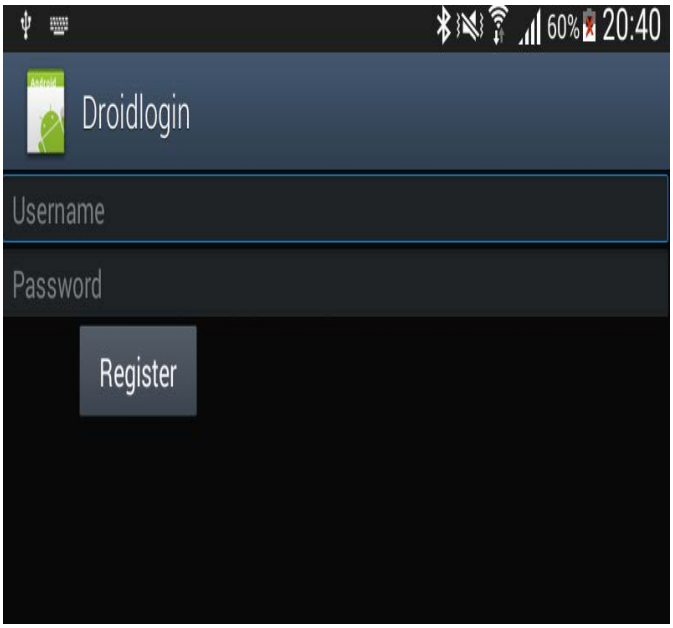
Ilustracion 13: Interfaz Chat room.

Habiendo ingresado una IP correcta y existente, se le pide el ingreso de un Nick y posteriormente puede usar toda la pantalla para charlar tanto mensajes públicos o privados.



Ilustracion 14: Interfaz Pie Envio mensaje.

El pie de la aplicación muestra, para el primer prototipo, un textView para ingresar cualquier texto, comando o mensaje para enviarse a los demás usuarios conectado.



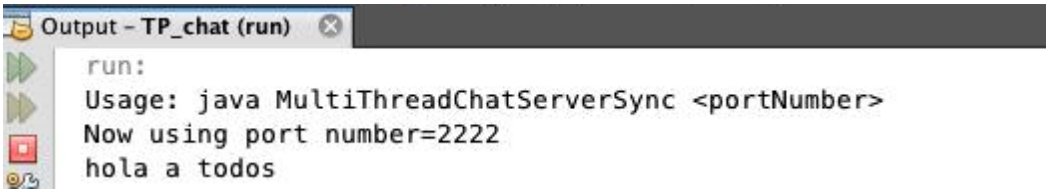
Ilustracion 15: Interfaz Registro de Usuario.

Una vez finalizado todo el procedimiento lógico y correcto al pulsar el botón “Login”, volvemos al inicio, y esta vez en lugar de dar “login”, pulsamos el botón “Registrarse” por lo que nos llevara a un campo con dos EditText el cual permitirá el ingreso de un username y una contraseña para luego ser parte del sistema y asi loguearnos, si no queremos eso o ya tenemos una cuenta y se desea volver a la aplicación principal, se debe tocar el botón “Regresar” de tu dispositivo móvil.

*Nota: las fotos del ejemplo fueron realizadas por un Tablet Samsung Galaxy tab 3 y por mi mismo con mis propios datos para realizar pruebas por lo que difiere el diseño en cada dispositivo móvil.

Interfaz Servidor.

Desde el lado servidor no presenta interfaz ni diseño, sino que simplemente es importante recalcar que servidor muestra los mensajes (“hola a todos”)ya sean públicos o privados (cifrados) para dar veracidad a las pruebas de que los mensajes llegan encriptados (base64) y son desencriptados con escritura visible posteriormente durante el transcurso de la aplicación móvil.



Ilustracion 16: Interfaz Servidor

7.2.5.4. Especificación de módulos

A Continuación Se detallan los nuevos módulos agregados en este Prototipo.

Nombre Módulo: Agrega llave publica.			
Parámetros de entrada		**Parámetros de Salida**	
Nombre: base64 (llave Publica en formato base64)	**Tipo de dato:** String	**Nombre:**llavePublic (convertido en base64)	**Tipo de dato:**Varchar (200)
Nombre Módulo: Modificar llave publica			
Parámetros de entrada		**Parámetros de Salida**	
Nombre: llavePublic (convertido en base64)	**Tipo de dato:**Varch ar(200)	**Nombre:**llavePublic Actualizado en el servidor.	**Tipo de dato:**Varchar (200)
Nombre Módulo: Cifrar llave sesión.			
Parámetros de entrada		**Parámetros de Salida**	
Nombre: llavePublic	**Tipo de dato:** String	**Nombre:**String a cifrar como variable local.	**Tipo de dato:**String
Nombre Módulo: Agregar nuevo usuario			
Parámetros de entrada		**Parámetros de Salida**	

Nombre: usuario, pass	**Tipo de dato:** String	**Nombre:** String con los datos a ingresar a la BD	**Tipo de dato:** Varchar
Nombre Módulo: Editar Usuario			
Parámetros de entrada		**Parámetros de Salida**	
Nombre: username, passw	**Tipo de dato:** Varchar	**Nombre:**String con los Datos editados y no editados de algún usuario.	**Tipo de dato:**String
Nombre Módulo: Comparte llave de sesión.			
Parámetros de entrada		**Parámetros de Salida**	
Nombre: LlaveAES	**Tipo de dato:**String	**Nombre:** Encriptado para ser usado como variable local.	**Tipo de dato:**String
Nombre Módulo: Eliminar usuario.			
Parámetros de entrada		**Parámetros de Salida**	
Nombre: username.	**Tipo de dato:**Varch ar	**Nombre:**Consulta SQL para eliminar al usuario seleccionado..	**Tipo de dato:**String

Tabla 21: Especificacion de módulos 2

7.2.6. Evaluación y pruebas

7.2.6.1. Elementos de prueba

A continuación se harán pruebas a funcionalidad de los distintos módulos ya definidos y a la suma de módulos que conforman la aplicación Android Cliente. Para Identificarlos definimos un código de Identificación para cada elemento de prueba.

1. Registrar Usuario.
2. Login del Usuario
3. Modificar usuario.
4. Gestion de llaves publicas y privadas.

7.2.6.2. Especificación de pruebas.

ID	Caracter ísticas a probar	Nivel de prueba	Objetivo de la Prueba	Enfoque para la definición de casos de prueba	Técnicas para la definición de casos de prueba	Actividades de prueba	Criterios de cumplimiento
1	Funcionalidad	Unidad	Probar si el usuario se ingresa correctamente.	Caja Negra	Particiones	a- Abrir aplicación. b- Llenar formulario y guardar datos.	-Verificar aviso de éxito en la APP.
2	Funcionalidad	Unidad	Probar si en el login se valida el usuario correctamente	Caja Negra	Particiones	a- Abrir aplicacion b- Intentar loguearse ingresando sus datos de acceso..	- Comprobar si el usuario ingresado es correcto con el usuario existente en el servidor web.

3	Funcion alidad .	Unidad	Cambiar los datos de usuario.	Caja Negra.	Particiones	Usuario a modificar en servidor web.	Comprobar si los datos se actualizaron
4	Funcionalid ad	Unidad	Verificar si se almacenan correctame nte las llaves publicas y privadas.	Caja Negra.	Particiones	En el registro de usuarios se agrega las llaves.	- Que existan llaves publicas y privadas validas en base64.

Tabla 22: Especificacion de pruebas 2

7.2.6.3. Responsables de las pruebas

El principal responsable de realizar las pruebas en este prototipo 2, no es mas que el autor del informe de la Tesis, Bryan Ramirez.

7.2.6.4. Calendario de pruebas

Las pruebas fueron realizadas luego de concluido el primer prototipo, las cuales se detallan en la siguiente tabla:

Nombre de Prueba	**Fecha**
Registrar usuario.	30 de Junio de 2016
Login de usuario	30 de Junio de 2016
Modificar usuario.	30 de Junio de 2016
Gestion de llaves publicas y privadas.	30 de Junio de 2016

Tabla 23: Calendario pruebas 2

7.2.6.5. Detalle de las pruebas

1.- Registrar usuario: Este módulo será probado con WIFI, se establece una configuracion de IP valida al servidor web, siempre que exista y se disponga uno, asi mismo como los datos de acceso al servidor deben ser válidamente correctos y no permitir datos duplicados.

ID Caso De Prueba	Características a Probar	Datos de Entrada	Salida esperada	Salida Obtenida	Éxito / Fracaso	Observaciones
1.a	Funcionalidad y Desempeño	bryan, bryan (Username, password)	Se registro correctamente	-Se registro correctamente	Éxito	Se agrego al usuario nuevo.
1.b	Funcionalidad y Desempeño	Bryan, bryan (valores duplicados)	.Usuario existente	-Error, nombre duplicado	Exito	Evita usernames duplicados.
1.c	Funcionalidad y Desempeño	Asadadadas, adsdasdasd (datos aleatorios y largos)	Se registro correctamente.	Se registro correctamente	Exito	Se prueba con username de largo alto.

2.- Login usuario Para lograr la conexión con el servidor, el dispositivo móvil debe estar conectado a internet y configurada correctamente, de lo contrario no se tendra validez en la prueba sin conexión.

ID Caso De Prueba	Características a Probar	Datos de Entrada	Salida esperada	Salida Obtenida	Éxito / Fracaso	Observaciones
2.a	Funcionalidad	Bryan, bryan (usuario existe)	Login correcto.	“Bienvenido”	Éxito	Ingresando un usuario valido y existente.
2.b	Funcionalidad	Bryan12, bryan (usuario no existe)	“login incorrecto	“No existe usuario o datos incorrectos”	Exito	Ingresando un usuario valido pero q no existe en el servidor..
2.c	Funcionalidad y desempeño	Bryan, bryan12	“login incorrecto”	No existe usuario o datos incorrectos”	Exito	Ingresar usuario existente pero con datos erroneos (en este caso password).

3.- Modificar usuario: Verificamos si la consulta de modificar usuario responde adecuadamente en el servidor. Esta prueba se realizará solo en el servidor, por lo que se necesita solo una modificacion desde servidor web.

ID Caso De Prueba	Características a Probar	Datos de Entrada	Salida esperada	Salida Obtenida	Éxito / Fracaso	Observaciones
3.a	Funcionalidad	Bryan, bryan.	Usuario existe	Usuario duplicado.	Éxito	Si registramos un nombre existente, servidor verifica duplicados.
3.b	Funcionalidad	Bryan, bryan546	“”Usuario modificado	“Se realizo la consulta	Exito	Una vez validado, se verifica si se modifica y se actualiza el password del usuario. .
3.c	Funcionalidad	Bryan, bryan, 452DFS	No se puede modificar la llave publica.	“usuario modificado	Fracaso	Modificamos una llave publica del registro de usuario a un formato incorrecto.

4.- Gestion de llaves publicas y privadas: Este módulo será probado siempre con el WIFI activado y la configuracion correcta de servidor, no es necesario que hayan muchos clientes conectados y registrados, pero al menos debe existir 2 clientes conectados.

ID Caso De Prueba	Características a Probar	Datos de Entrada	Salida esperada	Salida Obtenida	Éxito / Fracaso	Observaciones
4.a	Funcionalidad	“1234567890abctthf”	Formato de llave incorrecta	ExceptionFailedkey	Éxito	Una llave publica de largo incorrecto al codificar un mensaje, fuera de 128 bits
4.b	Funcionalidad	“1234567890abcdtv”.	Formato de llave correcta		Exito	Una llave publica de largo correcto al codificar un mensaje, de 128 bits .

Tabla 24: Detalle de las pruebas 2

7.2.6.6. Conclusiones de las pruebas

Las pruebas realizadas en este ámbito, es la implementacion de un login y de un sistema de registro de usuarios para acceder a la aplicación asi mismo registrar su llave publica y privada respectivamente para poder usarse a cifrar mensajes posteriormente, se procura llevar la configuracion del web service de manera correcta y bien configurada, de lo contrario, es imposible llevar a cabo toda la operación, ademas de ello, nos permite proceder con el uso de la criptografia, el intercambio de la llave de sesion y el cifrado durante el intercambio de mensajes entre diferentes usuarios sea publico y privado.

8. Plan de Capacitación y entrenamiento

Usuarios a capacitar

Los usuarios a los cuales se les realizará una capacitación, son los usuarios que usaran la aplicacion, pues son los encargados de la utilizacion volátil del sistema, vale decir, el manejo de información del intercambio de mensajes tanto públicos como privados, asi como el sistema de autentificación..

Se les indicarán cada una de las funciones que tiene la aplicación y el correcto funcionamiento de las opciones.

Tipo de capacitación

El tipo de capacitación se hará por medio de una presentación tipo conferencia, como es un sistema autónomo, es decir, que todo proceso funciona internamente, no es necesario una complejidad alta, sino solo capacitarse en un ámbito de interfaz y navegación para usarse correctamente, para esto se enviará una invitación a todos los posibles ausuarios que tendrá a disposición la universidad, el aprendizaje se hará con un ejemplo real.

Como último recurso, se dejará un correo para futuras consultas y peticiones de los Usuarios.

Funcionalidad y aspectos que serán abordados

Las principales funciones que serán tratadas son el inicio de sesión o Login, en el cual se hará hincapié en la seguridad, siendo un aspecto importante decir que la la contraseña es personal, asi mismo como la inclusión de una llave privada independiente en su propio móvil.

La otra funcionalidad es el intercambio de mensajes, se enseñará cómo acceder al sistema, al registro y a activar el comando para un mensaje privado.

Tiempo estimado

Las horas estimadas en el aprendizaje se detalla de la siguiente forma

Módulo	**Horas Requeridas Aproximadas**
Conferencia*	1 c/u
- Introducción	0.1
- Manejo de la aplicación (Registro y Login)	0.5
- Intercambio de mensajes públicos y privados.	1
-Interfaz de diseño y navegación.	0.5
Total	3.1

Tabla 25: Tiempo estimado

Para este punto se consideran 2 clases de capacitación, cada una con distintas personas pues no se requiere mucha cantidad de personas para entender el funcionamiento.

9. Plan de Implantación y Puesta en marcha

Al ser un modelo de prototipos, la puesta en marcha va sobre el desarrollo de este sistema, pensando en la marcha blanca como la finalización del prototipo 2, vale decir, se pone a disposición de los alumnos la aplicación móvil para sistema operativo Android.

Para este punto, se necesita tener a disposición un servidor de la Universidad, el cual es facilitado por el laboratorio de redes de la facultad de ciencias empresariales, aunque de forma temporal, sin embargo se está tramitando la opción de implantar un servidor propio dentro de la universidad, para el futuro uso de la aplicación.

Aún así la liberación oficial de la aplicación, requiere la revisión y autorización por parte de la Universidad del bio-bio, lo que conlleva nuevas iteraciones, que hagan aún más amigable y útil el uso de este sistema.

Para completar el plan de implantación se seguirán los siguientes pasos.

Instalación de MySQL y Netbeans:en esta etapa se requiere la instalación de estos 2 software, en el cual actuara como un servidor y un relay.

Configuración de servicio web: Esta se realiza, también en el servidor del laboratorio de redes, parte fundamental del sistema, ya que sin este paso, no se podría administrar, para ello se crea una aplicación en lenguaje PHP configurando sus parametros correctos.

Ingreso de Datos reales:este ingreso se hizo durante el desarrollo del segundo prototipo

Difusión de la aplicación cliente: Corresponde a enviar por correo o un enlace de descarga dentro de la página oficial de la Universidad, para que los alumnos puedan instalarla, se estima que esa difusión podría tardar a lo menos un mes.

Capacitación: De acuerdo a lo señalado anteriormente, se estima en una semana.

10. Resumen Esfuerzo requerido

El esfuerzo requerido en horas, se representa en la siguiente tabla, separándolas por actividades o fases

Actividades/faces	**Horas**
Conocimiento Socket-Servidor	40
Estudio Java	40
Estudio Android	40
Estudio Bases de datos MYSQL	20
Definición del Proyecto	20
Factibilidad	10
Estudio de criptografia	45
Pruebas con prototipo 1	25
Modelamiento de datos	10
Diseño	20
Codificación	300
Pruebas prototipo 2	60
Reuniones con profesor guia	25
Poblar la base de Datos	25
Estudio SQLite	10
Puesta en Marcha	10
Total	**700**

Tabla 26: Resumen esfuerzo requerido

11. Conclusiones

Uno de mis principales objetivos tras la realización de este proyecto de Titulo, era el de aplicar en su mayoría los conocimientos que la universidad del Bio Bio me ha otorgado durante mi formación. Conocimiento que me permitió hacer posible el desarrollo de este sistema, el cual resulta novedoso en nuestro país, ya que utiliza tecnología nueva y es en sí una idea que tiene el potencial de extenderse a un sinfín de áreas, o al menos, en el ambito de una aplicación de mensajeria muy similar a WhatsApp o Line, debido a que es una forma de conocer y comprender el desarrollo de los cliente de mensajería, lo cual se alinea directamente a mi carrera abriéndonos un nuevo abanico de posibilidades.

Concluyendo directamente con respecto al sistema, a través del uso de éste, se completo la mayoría de los objetivos específicos, en cuanto a criptografía simétrica y asimétrica se refiere, es decir, el intercambio de llaves, encriptación/desencriptación de los mensajes y de la llave de sesión, asi como también la implementación de la Firma Digital, entre otros. Aquello que no se pudo implementar fue la creación de grupos y de encriptación para archivos más pesados como imágenes o el intercambio de archivos binarios que podría implementarse en un futuro cercano, con respecto a la opinión personal de la implementación de todo lo anteriormente mencionado, se puede recalcar que siente más seguro y confiable el uso de la criptografia durante el intercambio de los mensajes en la aplicación, esto se determina al intercambiar, por ejemplo, el numero de cuenta bancaria, de mensajes personales o simplemente de la integridad del mismo usuario tranquilizandolo al momento de usar la aplicación, obviamente ésta podria mejorarse añadiendose multiples tecnicas de mejora, como el uso de la firma digital, o el uso de Hash, asi como la encriptacion de la gestion de usuarios, md5, etc. Esto último se espera implementar durante el transcurso de un futuro, añadiendo esto, el sistema tendrá mucho potencial para seguir su desarrollo a futuro, siendo incluso implementado en empresas que requieran su propio sistema o servidor con mensajeria segura. Pero esto último también debe ir de la mano con que la disponibilidad de las tecnologías, permitan a la mayor parte de las personas contar con internet móvil de libre acceso en su celular.

Y como punto final a modo de complementar mi formación profesional, este proyecto de título, me brindó una experiencia nunca antes obtenida a lo largo de mi carrera universitaria, la cual fue desarrollar un proyecto de forma íntegra y divertida permitiendome ser autodidacta para aprender los desarrollos y las tecnologias mas novedosas como el lenguaje Android, SQLite y su complementación y sincronización con los Web Services, permitiendome sentir en carne propia lo que significa desarrollar un sistema en base a prototipos como tambien a la utilidad de la planificación y organización, sin mencionar lo necesaria que se hacen las herramientas de trabajo para el desarrollador.

12. Bibliografía

[1] https://es.wikipedia.org/wiki/WhatsApp
Párrafo 1, linea 5.
[Consulta 2016]

[2] Pino Caballero Gil. “Introducción a la criptografía”,RA-MA S.A. Editorial y Publicaciones, 2da. Edición, 2002.

[3] Fuente Imagen: http://ubiobio.cl/miweb/web2012.php?id_pagina=5152

[4] Java Socket y ServerSocket

http://codigoprogramacion.com/cursos/java/103-sockets-en-java-con-cliente-y-servidor.html#.V48W7Pm7iko
[Consulta 2016]

[5] Fuente Imagen: http://elinux.org/images/c/c2/Android-system-architecture.jpg

[6]. https://es.wikipedia.org/wiki/SQLite
[Consulta 2016]

[7] Uso de WebServices desde Android con Mysql.
http://picarcodigo.blogspot.cl/2014/05/webservice-conexiones-base-de-datos.html
[Consulta 2016]

[8] Criptografia simétrica y asimétrica.
https://infosegur.wordpress.com/unidad-4/criptografia-simetrica-y-asimetrica/
[Consulta 2016]

13. ANEXO: Planificación inicial del proyecto

14 ANEXO: RESULTADOS DE ITERACIONES EN EL DESARROLLO

Se desarrollaron 2 prototipos en el desarrollo de este sistema.

14.1 Primer Prototipo

Este primer prototipo fue más bien hecho con el objetivo de ver que alcance podría tener el software en lo que es el manejo de los dispositivos móviles, y que tan preciso y ágil podría ser el intercambio de los mensajes asi como el uso de un servidor como relay con un dispositivo movil. De este prototipo se logro concluir que era posible la realización del sistema entre dispositivos moviles, ya que se pueden obtener resultados bordeando lo eficaz, pero por otro lado se aleja mucho de nuestro proposito, ya que los mensajes solo se envian en texto plano sin cumplir el proposito de este proyecto.

14.2 Segundo Prototipo

El segundo prototipo se enfocó en desarrollar las tecnicas de criptografia durante la utilizacion de la aplicación en el envio de mensajes privados. además de cambiar ciertos nombres y modificar detalles minimos en lo que respecta a lo visible de la aplicación cliente. a partir de este prototipo, sehace visible la necesidad de agregar un Login y una gestion de usuarios, ya que en ciertas ocasiones se necesitaba un servidor que administre las claves publicas, asi mismo otro que administre de manera personal las claves privadas.

15 a) ANEXO: DICCIONARIO DE DATOS DEL MODELO DE DATOS

Las tablas correspondientes al modelo de datos es la siguiente:

Nombre: Usuarios

Descripción: Esta tabla contiene la información asociada a la gestión de los usuarios del sistema.

Atributo	Tipo	Dimensión	Descripción
username	Varchar	10	Contiene el nombre de usuario del cliente.
passw	Varchar	20	Contiene la contraseña del usuario.
llavePublic	Varchar	200	Llave publica única asignada al usuario registrado.

Tabla 27: Diccionario datos usuario

Nombre: llavePrivada

Descripción: Tabla que esta alojada en SQLite, su función es almacenar la llavePrivada independiente en cada dispositivo móvil.

Atributo	Tipo	Dimensión	Descripción
username	Varchar	10	Contiene el nombre de usuario del cliente.
LlavePrivada	Varchar	200	Llave privada única asignada al usuario que usa su dispositivo móvil.

Tabla 28: Diccionario datos llavePrivada

a) MANUAL DE USUARIO

1 INTRODUCCIÓN

En este documento se describirá los objetivos e información clara y concisa de cómo utilizar el "**Software para el intercambio seguro de mensajes y archivos entre dispositivos móviles"** para su buen y correcto funcionamiento.

El **Software para el intercambio seguro de mensajes y archivos entre dispositivos móviles** fue creado con el objetivo de brindar facilidades y seguridad a los usuarios y/o clientes para consultar sobre el uso que se le da a las características instaladas en el uso de la aplicación, sus datos de registros, comandos, Login, entre otras opciones. Es de mucha importancia consultar este manual antes y/o durante la visualización de la aplicación, ya que esto lo guiará paso a paso en el manejo de las funciones en él, con el fin de facilitar la comprensión del manual, se incluye imágenes explicativas.

Si bien, la aplicación trata todo de manera automática, como el intercambio de llaves, la desencriptacion/encriptación del texto plano, la firma digital y el envió de los mensajes, no se requiere mucha petición y solicitud de parte del usuario salvo teclear mensajes y enviarlo, registrar su cuenta así como loguearse, es por eso que debido a que no se requiere la participación exhaustiva del usuario, de igual manera se incluirá todo lo posible del funcionamiento para un mejor entendimiento.

2 OBJETIVOS DEL MANUAL

El Objetivo primordial de éste manual es ayudar y guiar al usuario a utilizar el **Software para el intercambio seguro de mensajes y archivos entre dispositivos móviles** obteniendo información deseada para llevar una buena toma de decisiones al momento de escoger los software que requieren renovación.

Por lo tanto el manual comprende lo siguiente:

- Guía para acceder al **Software para el intercambio seguro de mensajes y archivos entre dispositivos móviles**
- Conocer cómo utilizar el sistema, mediante una descripción detallada e ilustrada de las opciones.
- Conocer el alcance de toda la información por medio de una explicación detallada e ilustrada de cada una de las páginas que lo conforman.

- Conocer la visualización de los mensajes públicos y privados recibidos con protección o sin.

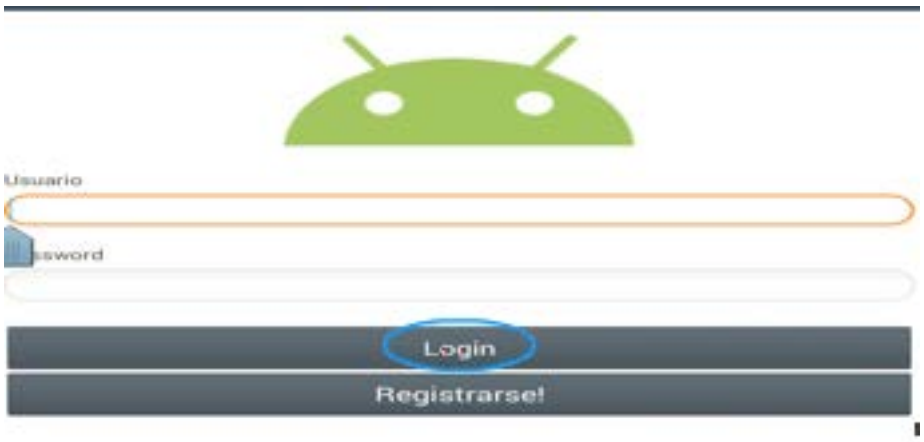
3 LO QUE DEBE CONOCER

Los conocimientos mínimos que deben tener las personas que operarán sistema web y este manual son:

- Conocimientos básicos acerca de Programas Utilitarios.
- Conocimientos básicos de Aplicación Móvil Android.
- Conocimiento básico de Internet.
- Conocimiento básico del termino usado “IP”
- Conocimientos básico para procesos lógicos (percatarse de los mensajes)

4 INGRESO AL SISTEMA

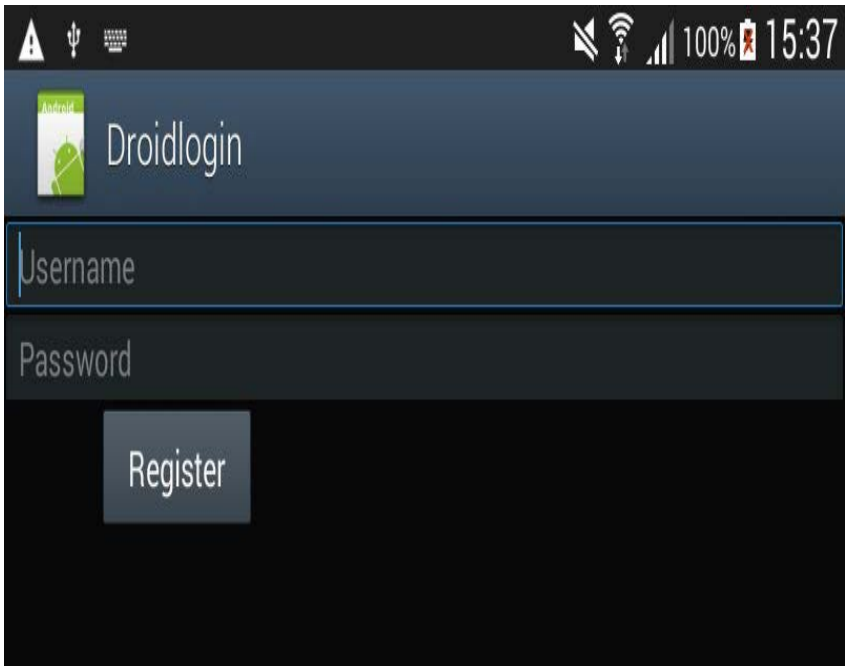
Para hacer ingreso al **Software para el intercambio seguro de mensajes y archivos**, primeramente **SE DEBE** estar registrado en el servidor (haberse creado una cuenta personal) De lo contrario, no se puede acceder. Si ya tenemos una cuenta registrada entonces rellenamos nuestros datos personales (Usuario y Password) y pulsamos "Login". Esperamos que se autentifique y procederá a pedirnos la IP.



5 REGISTRO EN EL SISTEMA

Pero antes, primero hay que considerar ciertos aspectos importantes antes de registrar un usuario al sistema, el servidor debe estar funcional y bien configurado, (para esto debe estar instalado XAMPP(*) o cualquier otro gestor de base de datos MySQL junto con Apache) Considerando que la maquina/servidor el cual se configuro la dirección IP y/o datos del WebService existe y esta funcionando, procedemos a pulsar el botón “Registrarse!”

Una vez dentro, habiendo pulsado el botón “Registrarse!”, se procede a rellenar los campo de datos solicitados ingresando nuestro username y password para luego finalmente pulsar el botón “Register”.



Evidentemente una vez pulsado el botón “Register” saldrá el aviso de que todo ha estado bien, mientras que si rellenamos datos equivocados o duplicados, igualmente saldrá el aviso de Éxito, sin embargo, al ser duplicado, estos no se ingresaran al servidor.

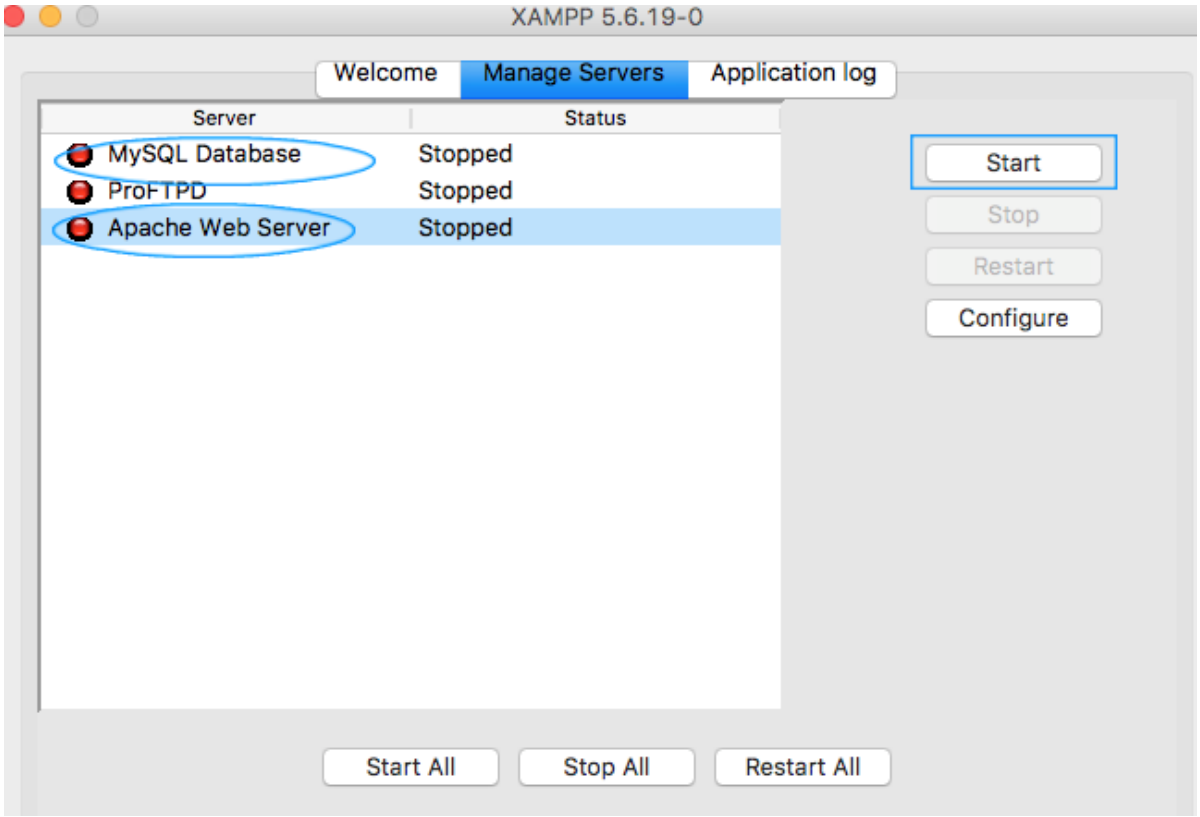
Verificamos si efectivamente se ingreso a la base de datos (web) en la siguiente imagen:

Opciones

	id_usuario	username	passw	llavePublic
Editar Copiar Borrar	58	pato	pato	MFwwDQYJKoZIhvcNAQEBBQADSwAwSAJBAPWAeLNI7D3GVNhCUW...
Editar Copiar Borrar	59	bry	bry	MFwwDQYJKoZIhvcNAQEBBQADSwAwSAJBAM8+QC26ml1eGnQRA7...

Como vemos, se asigno el username, y el password, asi como también su respectiva llave Publica para uso futuro al cifrar mensajes.

NOTA(*): Se requiere instalar XAMPP, para esto, vamos a la pagina de descarga de XAMPP en el siguiente link: https://www.apachefriends.org/es/index.html
Posteriormente descargamos la versión que nos acomode a nuestro sistema operativo, en mi caso, Mac OSX. Finalmente después que se descargue y se instale, ejecutamos XAMPP y ponemos a correr las funciones.



Seleccionamos MySQL Database (marcado con circulo celeste) y pulsamos el botón “Start” (marcado con un cuadrado celeste), se repite el proceso para “Apache Web Server”. Todo funcionara bien cuando ambas características cambien de color Rojo a Verde.

6 ELEGIR SERVIDOR

6.1 Establecer conexión

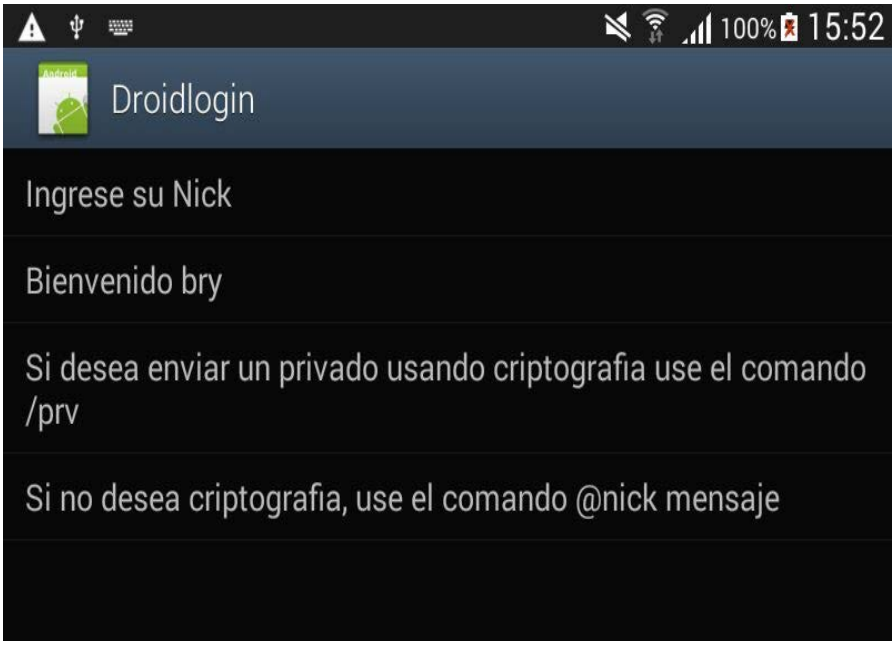
Despues de crear una cuenta y loguearnos correctamente en la aplicación, procedemos a establecer una conexión a un servidor ingresando una dirección IP valida y finalmente pulsamos el botón “Conectarse” para llevar a cabo la tarea.

7 CHAT DEL SISTEMA (Cliente)

7.1 Bienvenida

El chat del sistema se refiere a la "ventana" o "visualización" del chat, en donde se leerán todos los mensajes públicos y privados que los usuarios envían.

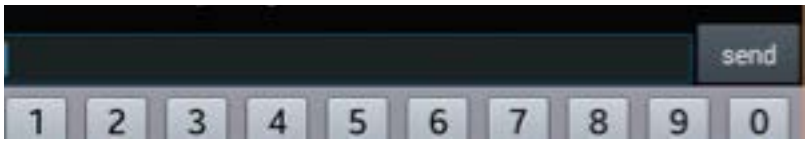
Pre-Requisito: Despues de ingresar una dirección de IP valida y conectarse a un servidor, se entra a la ventana del chat, donde se muestra la Bienvenida.



Tal como se ve en la imagen, hay dos comandos posibles, el envió de un mensaje privado sin criptografía (Sin Protección) así como también con criptografía (Con protección).

7.2 Envio de mensaje

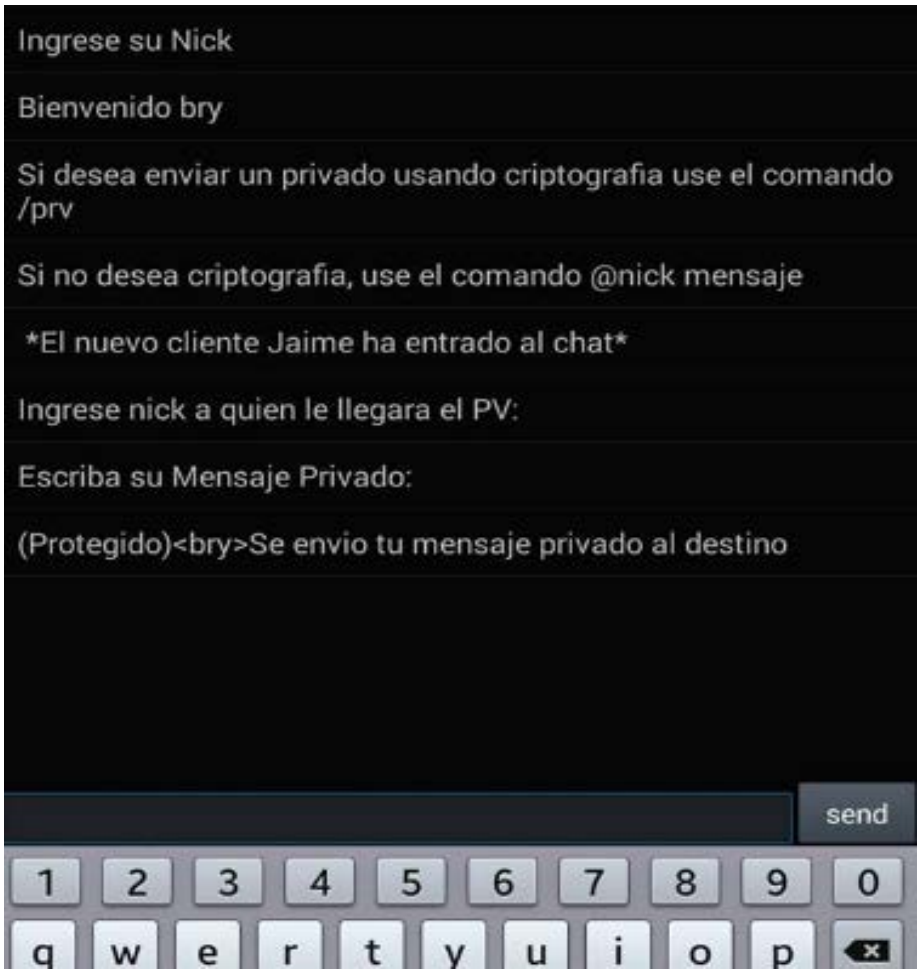
Tal como se indica en la imagen abajo, hay un campo donde permite teclear cualquier mensaje o símbolos, de manera que al enviar dicho mensaje, basta con pulsar el botón “Send” (Enviar)



7.3 Comando de mensaje Privado con Criptografia

Mientras se visualizan los mensajes públicos que los demás usuarios envían, se puede optar por enviar un mensaje privado CON CRIPTOGRAFIA a un determinado usuario. Por ejemplo a “Jaime”. Por lo tanto usamos el campo para rellenar texto y escribimos el comando y seguimos estos pasos:

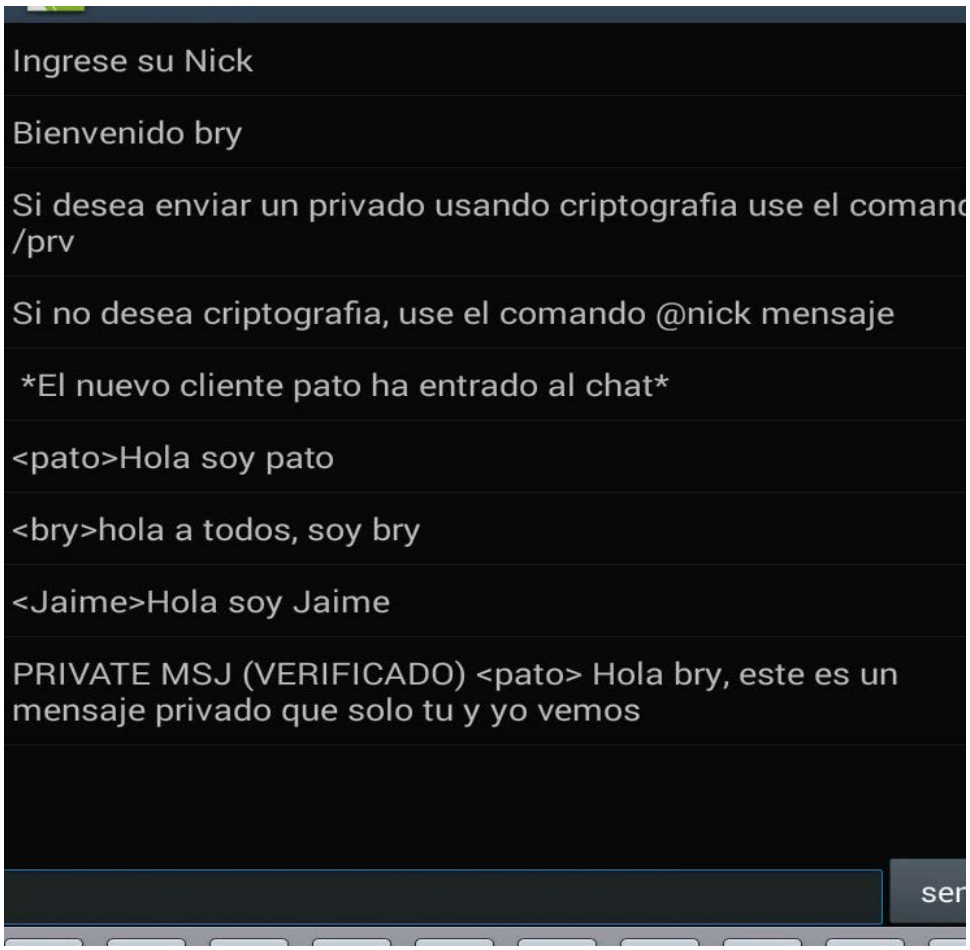
- Escribimos: “/prv” (comando para avisar al sistema que queremos enviar un privado con criptografía)
- Escribimos: “Jaime” (el sistema nos pedirá un Nick a quien enviarle el mensaje)
- Escribimos: “Hola jaime”(El sistema nos pedirá que escribamos el mensaje que queremos enviarle, en este caso, uso el ejemplo Hola Jaime)



NOTA: La visualización del Nick “Jaime” y el mensaje “Hola Jaime” han sido introducidas previamente y no aparecen en la lista.

7.4 Visualización de la llegada de un mensaje privado (Con Criptografía)

A continuación tenemos un ejemplo visual donde me logueo como Bry y me llega un mensaje privado (Con Criptografía) de pato, el cual luego sigue el mensaje enviado. Las palabras que contienen entre los <> Significa que es el Nick de la persona que lo envió, en este caso "pato", En este apartado no hay nada que hacer, salvo leer el mensaje y darse cuenta que pato me envió dicho mensaje.



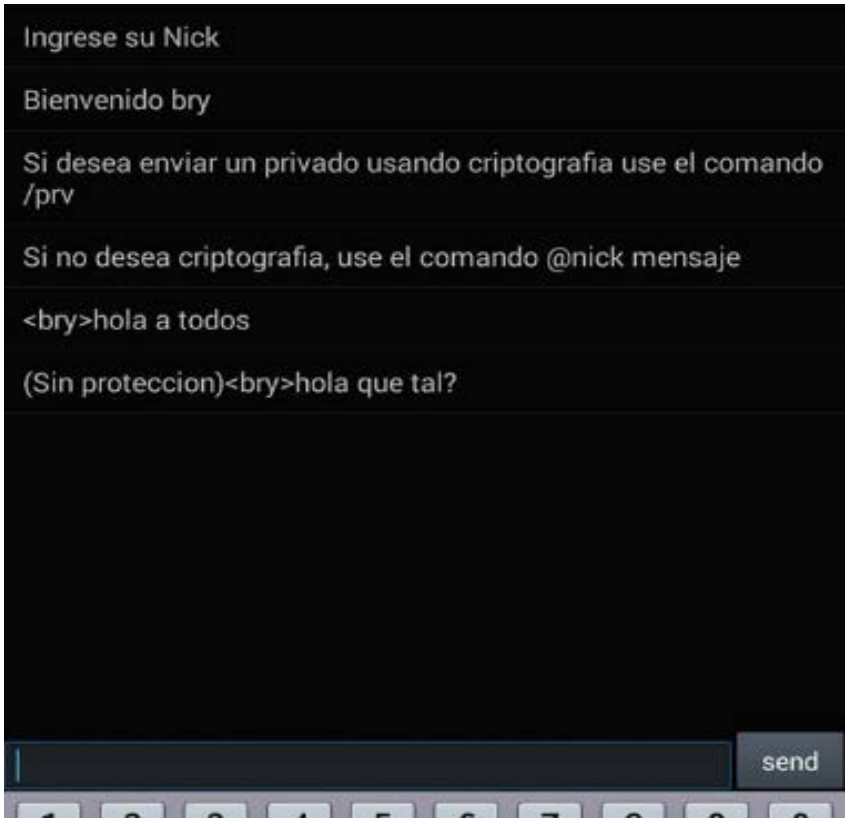
7.5 Comando de mensaje privado SIN Criptografía.

A continuación, a diferencia del uso con criptografía, este caso, es SIN criptografía, para esto, hacemos lo siguiente:

- Escribimos "@nick mensaje"

Donde Nick es el Nick de la persona a quien le enviamos un mensaje privado, y mensaje es el texto que le queremos enviar. Por ejemplo. "@Jaime hola que tal?", en donde "@Jaime" es el Nick destino y "hola que tal?" es el mensaje que le enviaremos.

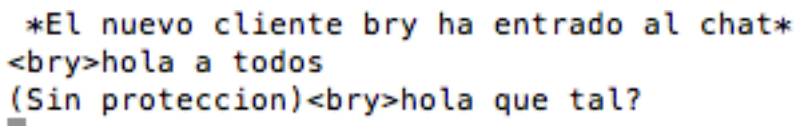
- Luego pulsamos el botón "Send" para enviarlo.
- Visualizamos el mensaje que enviamos y notamos que se envió sin criptografía (SIN PROTECCIÓN).



NOTA: La visualización del Nick "Jaime" y el mensaje "Hola que tal? " han sido introducidas previamente y no aparecen en la lista.

7.6 Visualización de la llegada de un mensaje privado SIN Criptografía.

Ya que no usamos criptografía, y no hay nada más que hacer que solo visualizar el mensaje sin protección, podemos verla en la terminal, o bien en un dispositivo móvil:



Tal como se ve en la imagen, la frase "(Sin protección)" Indica que se envió el mensaje privado que viene de bry sin el uso de Criptografía para ahorrar batería y no consumir recursos.

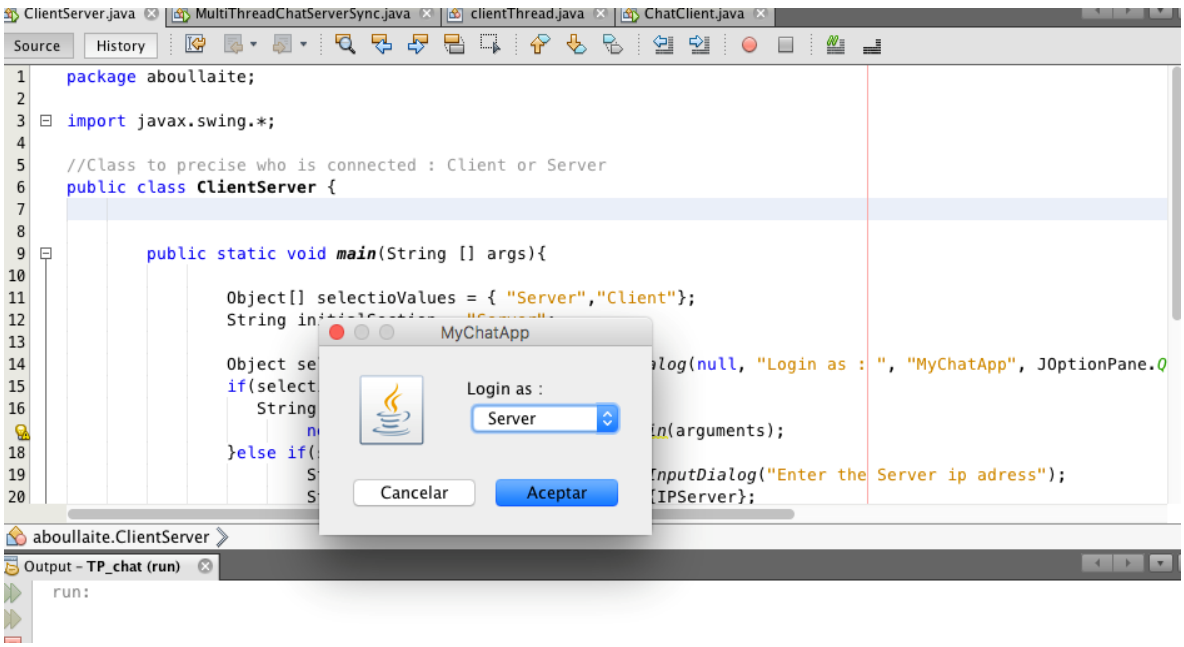
8 CHAT DEL SISTEMA (Servidor)

8.1 Servidor Relay

Ya que necesitamos un servidor que actué de Relay y acepte múltiples clientes, ejecutamos el servidor, para ello abrimos el programa Netbeans desde:

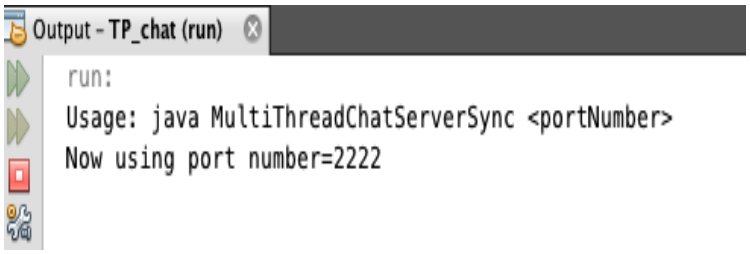
- Inicio
- tecleamos “Netbeans” y damos doble click.
- Buscamos la carpeta del proyecto “TP_chat”
- Click en la clase “ClientServer.java” y ejecutarlo.

Nos aparecerá una ventana de la siguiente forma:

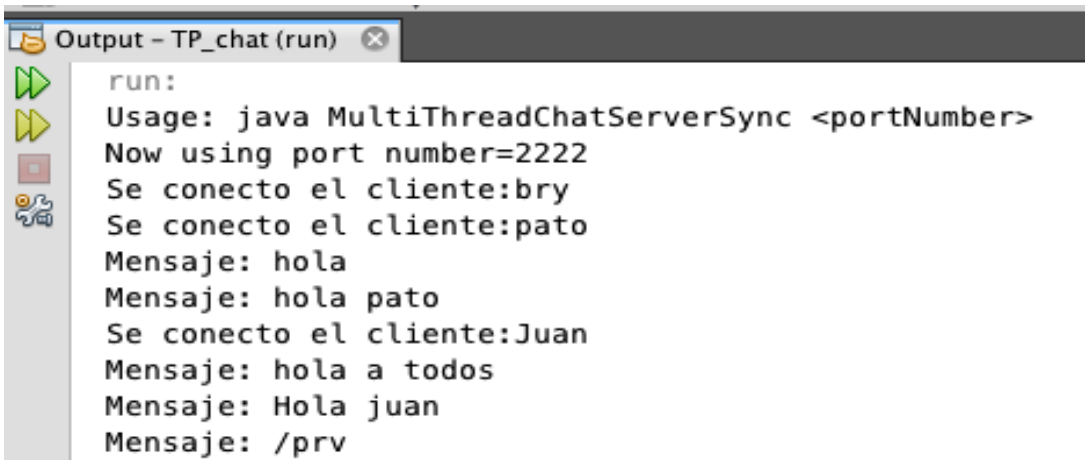


Le damos click a Aceptar siempre manteniendo la opción de “Server”.

Si todo sale bien, finalmente nos indica que se esta ejecutando sin problemas, preparado para recibir peticiones en la siguiente imagen:



Además, como visualización, también nos indica todos los mensajes que pasa por el, un registro tanto de textos como de usuarios que entran al servidor como se muestra en la siguiente imagen:

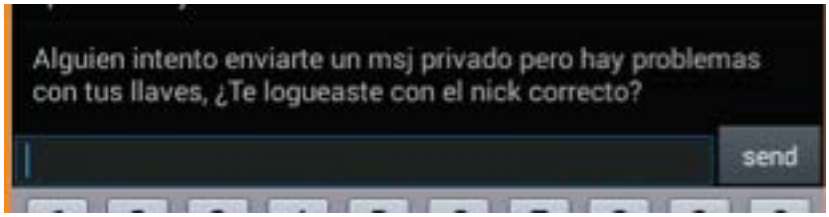


NOTA: El mensaje “/prv” solo es visible como que se ejecuto un mensaje privado, pero servidor no tiene conocimiento del contenido del mensaje.

9 Firma Digital (Visualización)

9.1 Fraude

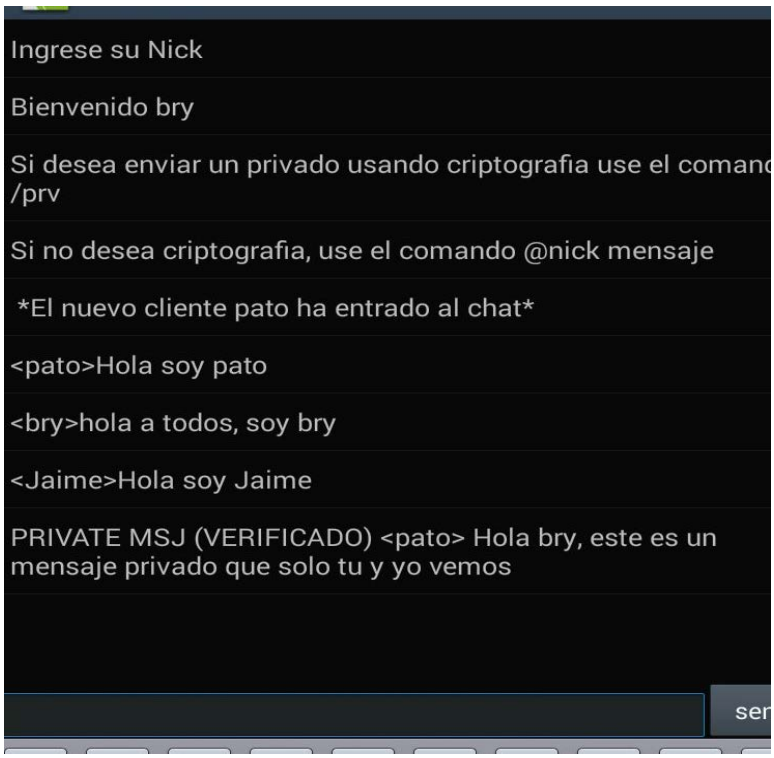
Si bien la firma digital no requiere nuestra participación puesto que es automática y autónoma, vamos a visualizar en la siguiente imagen lo que sucede cuando hay problema en las llaves o bien en desencriptar algún mensaje y/o se duda la identidad del usuario por posible fraude.



Cuando hay un fraude y/o un problema, aparece el aviso a la otra persona de que hay un error.

9.2 Éxito de firma Digital

Cuando no hay ningún problema en la unión de dos llaves o en la veracidad de la identidad de quien envía el mensaje, tenemos la siguiente imagen:



EN PRIVATE MSJ (VERIFICADO) el texto entre paréntesis, es decir, “VERIFICADO”, es suficiente para determinar que no ha habido ningún problema y se comprueba el origen del mensaje.