

PONTIFICIA UNIVERSIDAD CATOLICA DEL ECUADOR

FACULTAD DE INGENIERÍA

ESCUELA DE SISTEMAS

**ANÁLISIS COMPARATIVO DE HERRAMIENTAS DE MENSAJERÍA ENTRE
DISPOSITIVOS MÓVILES A TRAVÉS DEL PROTOCOLO IP**

BRYAN OBANDO

CRISTIAN VEGA

Director:

Dr. Gustavo Chafra A.

Quito, Julio del 2017

Agradecimientos

Agradezco a las personas que estuvieron en los momentos más importantes de mi carrera y estuvieron aún más presentes en los momentos difíciles, que supieron darme un empujón para lograr seguir adelante con mis metas.

Agradezco a mi Director de Disertación Dr. Gustavo Chafla por la guía y por encaminarnos a culminar este proyecto de disertación; y darnos la oportunidad de integrarnos en sus ideas de proyectos dentro de la institución para obtener mayor conocimiento y poder enfrentarnos a los retos que nos esperan fuera de la universidad. A mis correctores Ingeniero Fabián de la Cruz e Ingeniero Oswaldo Espinosa por su arduo desempeño en encaminarnos igualmente sobre el desarrollo de este trabajo de disertación desarrollándolo con calidad y esmero.

Agradezco a mis padres, Francisco Vega e Inés Niama por todo el apoyo y sacrificio que realizaron para llevarme alto y brindarme la posibilidad de estudiar y alcanzar mi meta de ser un profesional, siempre estuvieron ahí y les agradezco por ello y por muchas cosas más, y espero poder recompensarles por todas las cosas que hicieron por mí.

Agradezco a mi hermano Daniel por siempre darme su apoyo en cualquier situación que se presentaba y requería de tu ayuda, siempre tendrás igualmente mi apoyo cuando lo necesites y nunca negaré en dártelo en todo lo que necesites.

A mi compañero de Disertación Bryan Obando, por todo el trabajo y tiempo que invertiste sobre esta labor que al final logramos finalizarlo con calidad y empeño; la dedicación que le diste al proyecto sin esperar algo a cambio, en serio amigo muchas gracias.

A mis compañeros de la universidad, algunos que conocí antes de estar en la universidad y a otros que los conocí en el transcurso de mi carrera, muchos fueron quedando a través de este tiempo y a otros, ahora los puedo llamar amigos.

Cristian Vega Niama

Agradezco a mis padres, por todo su sacrificio y esfuerzo, ya que me han apoyado incondicionalmente en mi vida personal y profesional, sentando bases de responsabilidad, humildad, respeto y sus deseos de superación. Gracias por todos esos consejos y por saber transmitir toda su sabiduría tanto a mis hermanos como a mí, ellos también han sido una parte muy importante en mi vida y en esta etapa que estoy culminando.

El camino para llegar hasta aquí no ha sido nada fácil y en éste he conocido a grandes personas, un reconocimiento especial a mis profesores por los conocimientos impartidos, su experiencia, valores y habilidades transmitidas en esta, una de las etapas más valiosas de mi vida. A mis amigos por su ayuda, compañerismo y amistad en todo momento.

Al Dr. Gustavo Chafra por la oportunidad brindada para hacer realidad este momento, de igual forma mi gratitud al Ing. Fabián de la Cruz y a Oswaldo Espinoza por su colaboración, para que este trabajo haya finalizado con éxito.

Un agradecimiento a mi compañero de disertación y amigo Cristian Vega, por su responsabilidad y sus aportes para que este momento sea una realidad, de igual manera a mis amigos Diego Ponce y Juan Prado por su apoyo y por todos los momentos compartidos en el transcurso de mi vida universitaria.

De manera muy especial quiero dedicar este trabajo a mis padres, a mis hermanos, a mi tía, así como también a una persona muy importante en mi vida, mi abuelita quien siempre estuvo al tanto de mi vida y fue un gran apoyo en todo momento, sus consejos, historias y anécdotas enseñándome que en la vida hay que luchar y siempre mirar hacia adelante. Gracias a todos, por todo.

Bryan Obando Acosta

Contenido

CAPITULO I	8
1. Introducción	9
1.1. Antecedentes	9
1.2. Planteamiento del problema	11
1.3. Justificación.....	13
1.4. Objetivos	14
1.5. Alcance.....	15
1.6. Funcionalidades.....	15
1.7. Metodología de desarrollo (eXtreme Programming)	17
CAPITULO II	20
2. MARCO TEÓRICO	20
2.1. Sistemas operativos.	21
2.2. Protocolos.....	45
2.3. Android Studio	52
2.4. GCM.....	59
2.5. Definiciones	61
2.6. Bases de datos	68
2.7. Servicios Web	69
2.8. Aplicación Web.....	76

CAPITULO III	91
3. Definición de plataformas.....	92
3.1. Sistema operativo del dispositivo.....	92
3.2. Versión de Android.....	93
3.3. Conectividad del Dispositivo.....	94
3.4. Análisis de plataformas.....	96
3.5. Cuadro Comparativo aplicaciones móviles.....	101
3.6. Elección y definición de la plataforma.....	102
CAPITULO IV	103
4. Definición del problema.....	103
4.1. Introducción.....	103
4.2. Identificación de actores (Roles).....	104
4.3. Historias de usuario.....	105
4.4. Estimación y análisis de historias de usuarios.....	107
4.5. Diagrama conceptual.....	121
4.6. Diagrama de clases.....	122
4.7. Requerimientos de software y hardware para el desarrollo y despliegue de la Administración del Sistema.....	123
4.8. Requerimientos de Hardware y Software para el desarrollo y despliegue del aplicativo móvil.....	125

4.9. Diccionario de datos.....	126
CAPITULO V.....	129
5. Diseño del prototipo.....	129
5.1. Interfaces.....	129
5.2. Estándares de codificación.....	139
CAPITULO VI.....	144
6. Desarrollo del prototipo.....	144
6.1. Servicios Implementados.....	144
6.2. Desarrollo del código fuente.....	147
6.3. Prototipo Final.....	160
6.4. Pruebas.....	166
6.5. Análisis Resultados en Redes Wi-Fi.....	179
6.6. Análisis Resultados en Redes Móviles.....	180
6.7. Análisis Resultados en Aplicaciones Web.....	181
CAPITULO VII.....	183
7. Conclusiones y Recomendaciones.....	183
7.1. Conclusiones.....	183
7.2. Recomendaciones.....	185
CAPITULO VIII.....	187
8. Anexos.....	187

8.1. Resultados Encuesta.....	187
8.2. Tabla de 100 mensajes, 50 caracteres, Redes Wi-Fi, WhatsApp.....	196
8.3. Tabla de 100 mensajes, 50 caracteres, Redes Wi-Fi, Messenger.....	199
8.4. Tabla de 100 mensajes, 50 caracteres, Redes Wi-Fi, Skype.	202
8.5. Tabla de 100 mensajes, 50 caracteres, Redes Wi-Fi, Prototipo.	205
8.6. Tabla de 100 mensajes, 50 caracteres, Redes Móviles, Whatsapp.	208
8.7. Tabla de 100 mensajes, 50 caracteres, Redes Móviles, Messenger.	211
8.8. Tabla de 100 mensajes, 50 caracteres, Redes Móviles, Skype.	214
8.9. Tabla de 100 mensajes, 50 caracteres, Redes Móviles, Prototipo.....	217
8.10. Tabla de 100 mensajes, 50 caracteres, PC, Whatsapp.	220
8.11. Tabla de 100 mensajes, 50 caracteres, PC, Messenger.	223
8.12. Tabla de 100 mensajes, 50 caracteres, PC, Skype.	226
8.13. Tabla de 100 mensajes, 50 caracteres, PC, Prototipo.....	229
Bibliografía	232

CAPITULO I

El desarrollo de este proyecto presenta un análisis entre herramientas de mensajería instantánea que utilizan el protocolo IP para el envío de paquetes; utilizadas con frecuencia por las personas en general para comunicarse a través de mensajes de texto, se conoce que estas aplicaciones contienen diferentes funcionalidades como envío de imágenes, videos, entre otras; pero para un análisis más específico está enfocado en el envío y recepción de mensajes de texto y la estructura que tienen dentro del paquete de información que es transmitido a través del protocolo.

Este análisis será realizado con un estudio de las tecnologías que utilizan estas aplicaciones y un proceso de pruebas que determinan la tecnología más utilizada; para un posterior desarrollo de un prototipo en una plataforma web y móvil que permita conocer más a profundidad el funcionamiento de la tecnología seleccionada. Las plataformas serán desarrolladas en lenguaje java; utilizando la tecnología Java Empresarial para web y alojarlo dentro de un servidor de aplicaciones Jboss, mientras que la plataforma móvil será generada con la tecnología que Android Studio nos permite implementar para el desarrollo de aplicaciones móviles.

En este capítulo se hace una breve descripción del proyecto, su justificación, el objetivo principal del trabajo, sus objetivos específicos, el alcance, las funcionalidades que tendrá el prototipo y cual metodología de desarrollo será utilizada para la implementación del mismo.

1. Introducción

1.1. Antecedentes

La comunicación siempre ha sido un punto importante en el desarrollo y la interacción entre sociedades, tomó un gran avance cuando una red de redes se diseñó y que ahora la conocemos como la internet, la manera tradicional de interacción fue evolucionando y ahora entregar un mensaje no tarda un tiempo mayor a segundos para transmitirlo. (Castells, 2000)

La tecnología ha avanzado en gran escala, se tienen varios medios para interactuar entre personas como los correos electrónicos, mensajería en base a notificaciones a través de un protocolo; en cuanto al área de los dispositivos móviles encontramos la mensajería por medio de datos y con la inclusión de la Internet se integró todas las maneras de comunicación anteriormente mencionadas. Como estudio dentro del intercambio de información, varias compañías han diseñado diferentes maneras de comunicarse y con el incremento exponencial en la adquisición de dispositivos móviles y el continuo aumento, se ha optado por analizar alternativas de enviar información por diferentes medios, compararlas definiendo ventajas y desventajas que ayuden a encontrar la mejor plataforma para realizar una comunicación y un intercambio de mensajes entre diferentes dispositivos, de una manera eficiente y que satisfaga al usuario del servicio. (Roig, 2012)

Como punto principal para el desarrollo de una red de comunicación, conocer la penetración de internet dentro de la sociedad afecta en la manera del consumidor de un servicio de intercambio de información, las personas se identifican como actores cibernéticos y la internet

reúne a estos actores a distancia para que exista una interacción; un punto que se debe recalcar es que la internet no es solo un medio de comunicación sino una combinación de diferentes áreas de interés para los actores. (García, 2005)

Cuando se habla de comunicación a través de un protocolo, hablamos de paquetes de datos donde se prioriza que la transferencia de información llegue exactamente igual como fue enviada, por este motivo la existencia de los NCS (Network Controlled Systems) como una forma de control en el retardo o la pérdida de información a través de la estabilización de las ondas de las redes es primordial para cumplir este objetivo, estas engloban 2 grandes campos: sistemas de control o controladores maestro y esclavo con motivo de reducción de los retrasos en la estabilidad de un sistema y el otro campo es encontrar la mejor forma de optimizar la transferencia de información a través de protocolos o dispositivos. Por concepto general la comunicación a través de las redes es bilateral, es decir, existe un canal de retorno de información como respuesta a una recepción de datos, a este tipo de paquetes se los conoce como ACK. (Dias-Cacho, 2010)

Tiempo real es un requisito en la transferencia de información por lo cual el envío de datos pequeños es primordial en un mensaje de texto, si la transferencia conlleva datos multimedia (videos, imágenes, etc.) los protocolos utilizados son diferentes los cuales son derivados de los RTP, siendo estos los encargados de enviar la información multimedia a través del protocolo IP; los protocolos RTP corren sobre UDP, donde los paquetes son enviados como datagramas a otros equipos que se encuentran en una red de protocolo IP. Muchas veces los datagramas son considerados paquetes, pero la diferencia radica en que los datagramas son reservados para paquetes que no es posible realizar una confirmación si la información llegó al receptor (mensaje

ACK). Un protocolo RTP trabaja en conjunto con un RTPC, el cual es el controlador de la calidad del datagrama al ser receptado y el sincronizador de los medios de transferencia continua (Streaming).

Por otro lado, TCP es otro protocolo utilizado en la transferencia de datos a través de la internet, diseñado para un intercambio de información más fiable, nos referimos con esto a que los paquetes tienen la posibilidad de “perderse”, llegar en desorden o duplicarse al ser enviados, TCP se encarga de solicitar nuevamente la transmisión de estos paquetes, ordenarlos y minimizar la congestión de la red y si persiste alguno de estos problemas, se notifica al remitente de lo ocurrido.

Cuando se maneja protocolos es preferible utilizar el adecuado, dependiendo del tipo de paquete que se envía para evitar problemas de conexión entre ambos puntos, agregando a esto miles de equipos que están conectados a la internet desde computadores de escritorios a teléfonos celulares, por este motivo el manejo adecuado de tamaños de transferencia debe ser primordial para no saturar la red.

1.2. Planteamiento del problema

En la actualidad los dispositivos móviles son la forma de intercomunicación más usada a nivel mundial, independientemente del dispositivo, sistema operativo o redes móviles; ocasiona que estas tecnologías evolucionen rápidamente implementando nuevos servicios y plataformas.

Las distintas herramientas de mensajería son parte fundamental para que este tipo de comunicación sea posible, a pesar de contar con protocolos que permiten el desarrollo y transporte de datos como TCP/IP, actualmente no contamos con una plataforma de mensajería estándar; existen varias tecnologías que cumplen dicha función, sin embargo, y por la gran demanda que existe de este servicio, las comunidades de desarrolladores han implementado múltiples herramientas para cubrir esta demanda. Varias de estas propuestas han ido abarcando gran parte del mercado, muchos de los servicios actualmente disponibles han hecho uso de estas herramientas para el desarrollo de diferentes aplicaciones.

Existen diferentes alternativas de transmitir información, varios desarrolladores han implementado diferentes maneras de lograrlo como mensajes “push” a través de un servicio web, mediante la utilización de una conexión creada por un servicio de terceros entre ellos GCM, Hangouts (Nava, 2015), entre otras, una gran parte de estos servicios son desarrollados utilizando el protocolo HTTP y XMPP para manejo de los paquetes que serán enviados a través del protocolo IP.

Para personas naturales, empresas, gobiernos, entidades militares, entre otras, la comunicación es un factor primordial en las áreas que se desempeñen, y más aún el tiempo de respuesta debe ser inmediato, por este motivo el tener una herramienta que sea eficiente en el uso de recursos en un envío de información y cumpla con las características necesarias para estas entidades conlleva a una mejora tanto en su producción como en el desempeño del personal, dando enfoque a una empresa, implica una mayor producción por existir un ahorro en el tiempo de notificar a las áreas de las tareas que se debe cumplir o el envío de documentos para nuevos

contratos, si existe personal que labore externamente a las instalaciones de la empresa; tener la capacidad de comunicarse remotamente con ellos produce una optimización de tiempo y recursos que sean necesarios para las actividades.

Con la mejora continua de equipos portátiles, específicamente teléfonos celulares, el progreso en la comunicación es inmensurable, por motivo de ser cambiante todo el tiempo e ir mejorando a través del tiempo

1.3. Justificación

A través de una selección de aplicaciones de mensajería, realización de pruebas de funcionamiento y consumo de recursos y, posteriormente realizar una comparación entre la información adquirida de cada una de las pruebas de las aplicaciones, se seleccionará la más óptima con los mejores resultados adquiridos; para realizar este análisis es necesario una investigación de los protocolos que son utilizados por los sistemas como son TCP, UDP y los protocolos que circulan en estos como HTTP y XMPP, tipos de paquetes manejados en la transferencia de información como datagramas y procedimientos que se utilicen para controlar como son transportados, se definirá las ventajas y desventajas de la utilización de cada una de las aplicaciones.

Posterior al análisis se seleccionará una plataforma de mensajería adecuada para realizar un prototipo funcional, este será enfocado a una plataforma web y a dispositivos móviles empleando la aptitud de los equipos que tienen acceso a datos móviles e internet. El prototipo será

desarrollado bajo lenguaje Java y al terminar su desarrollo se realizará pruebas dentro de este y los resultados serán puestos a comparación con la herramienta de mensajería seleccionada anteriormente.

1.4. Objetivos

1.4.1. Objetivo general

Analizar y comparar diferentes plataformas de intercambio de información entre dispositivos móviles a través del protocolo IP.

1.4.2. Objetivos específicos

- Realizar un estudio de las plataformas de intercambio de información a través de mensajería instantánea.
- Definir criterios de comparación entre las plataformas que serán estudiadas.
- Seleccionar una plataforma eficiente para la implementación de un prototipo.
- Realizar pruebas del funcionamiento del prototipo dentro de un área delimitada para la comunicación entre dispositivos.
- Analizar los resultados de la plataforma implementada a través de pruebas.

1.5. Alcance

Se iniciará con una investigación de las diferentes plataformas de transferencia de información entre dispositivos móviles a través de mensajería instantánea, sus ventajas, desventajas y la comparación entre estos, ayudando a seleccionar una plataforma eficiente entre las definidas, y poder aplicarla dentro de un prototipo funcional de mensajería, realizar pruebas, comparaciones con la plataforma seleccionada anteriormente y finalmente culminar el proyecto de disertación con la presentación de datos comparativos y un prototipo funcional.

1.6. Funcionalidades

1.6.1. Funcionalidades sistema web

1.6.1.1. Autenticación de usuarios

El sistema tiene la capacidad de realizar una identificación de los usuarios registrados en el sistema a través de la autenticación de sus datos, es un proceso de autenticación con el nombre de usuario y la contraseña respectiva de cada usuario.

1.6.1.2. Administración de usuarios

El sistema permite al administrador realizar acciones sobre los registros de los usuarios, incluye crear, modificar, eliminar usuarios y reestablecer la contraseña de todos estos.

1.6.1.3. Mensajería instantánea

El sistema incluye la funcionalidad de mensajería instantánea entre los usuarios que se encuentren registrados dentro del sistema, limitándose a mensajería entre usuarios del aplicativo móvil y el aplicativo web.

1.6.2. Funcionalidades aplicativo móvil

1.6.2.1. Autenticación de usuarios

El sistema permite una autenticación de los usuarios, anteriormente registrados en el sistema por un administrador dentro de la aplicación web. Este proceso consiste en verificar el nombre de usuario y la contraseña del usuario que va a ingresar al aplicativo móvil, con la información almacenada en la base de datos.

1.6.2.2. Visualización de usuarios

El sistema permite la actualización manual de los usuarios registrados en el sistema a través del consumo de un servicio web y almacenamiento posterior dentro de la base de datos local del dispositivo inteligente.

1.6.2.3. Mensajería instantánea

El sistema consta con un módulo de mensajería instantánea entre los usuarios administradores y los usuarios propietarios de un dispositivo móvil, y a la vez mensajería entre los usuarios propietarios de dispositivos móviles.

1.7. Metodología de desarrollo (eXtreme Programming)

XP es una metodología de desarrollo ágil que da una gran importancia a la adaptabilidad sobre la predictibilidad, propone que los cambios en el transcurso del desarrollo son ineludibles y es más realista, adaptarse a ellos y no intentar definir todos los requerimientos al inicio del proyecto y realizar un esfuerzo en controlar todos estos cambios cuando se presenten, de esta manera se apega más hacia las necesidades del cliente que las metodologías tradicionales y diseñar un software de igual o mejor calidad. Como punto importante de XP, utiliza diferentes formas de diseñar los requerimientos del cliente, en XP se diseñan inicialmente las “historias de usuario”, las cuales vienen a reemplazar los casos de uso de otras metodologías de desarrollo. (Wells, 2001)

Una característica de XP es la interacción con el cliente, se lo realiza en pequeñas iteraciones y de esta manera el cliente puede dar su opinión mientras se desarrolla el sistema, las metodologías de desarrollo tradicionales se enfocan también en la planificación a futuro o la reusabilidad del código, mientras que XP niega completamente estas ideas, por tener como principio que realizar grandes cambios al mismo tiempo no funciona correctamente. (SMAK, 2012)

1.7.1. Fase de planificación

El equipo de desarrollo se reúne con el cliente para crear las “historias de usuario” o requerimientos del proyecto, se convierte estas historias en iteraciones que detallan en resumen la funcionalidad de esa área del sistema, se realizan cálculos de tiempos, costos y se asigna los responsables de cada iteración.

1.7.2. Fase de diseño

Se define estándares para la nomenclatura y la codificación de los diferentes módulos del sistema, se diseña un prototipo de datos (información visible) que ayuda a los responsables del proyecto a incluir ideas o modificarlas para un mejor entendimiento del proceso de desarrollo.

1.7.3. Fase de codificación

Esta es la fase más importante dentro de XP, tiene una mayor prioridad que la documentación para que el cliente tenga un entregable al final de la iteración, basado en los estándares que se definieron en la fase anterior, se trabaja entre pares de desarrolladores enfocados a una producción de alta calidad en el módulo que sean asignados.

1.7.4. Fase de pruebas

Se trabaja con pruebas mientras se desarrolla el proyecto, estas no son realizadas al final de la fase de desarrollo, cada método tiene sus propias pruebas y deben pasarlas para poder ser implantadas en el producto final, también se toma en cuenta la satisfacción del cliente en base a las especificaciones que entregó a los desarrolladores y una demostración de la funcionalidad al cliente.

1.7.5. Fase de aceptación

La base de XP es la integración del cliente a través de opiniones durante la fase de desarrollo, además se considera los criterios del gerente del proyecto. Cada criterio de aceptación del cliente, equivale a la base de un nuevo diseño y es el proceso de las “pruebas en código” para que sean repetidas nuevamente, si el cliente se encuentra satisfecho, se da por terminada la iteración y el diseño de una nueva iteración empieza con un ciclo completamente nuevo.

CAPITULO II

Definiremos todos los conceptos y tecnologías que se utilizarán para realizar un prototipo de mensajería instantánea el cual será analizado en capítulos posteriores y puesto a prueba con servicios similares.

2. MARCO TEÓRICO

La rápida evolución que ha tenido la tecnología y la búsqueda de como facilitar las actividades que se realizan diariamente, ha llevado a la creación de dispositivos móviles con grandes capacidades para el procesamiento de información. El diseño de redes facilita el intercambio de datos, ya sean por cableado o por conexión inalámbrica, y como una necesidad principal la comunicación con personas alrededor del mundo, logrando así una comunicación casi instantánea (en tiempo real), por lo cual equipos especializados, diseñaron y crearon métodos y equipamiento capaz de manejar esta masiva cantidad de datos, almacenándolos y procesándolos de tal forma que sea accesible para cualquier usuario. Todo esto no sería posible sin la existencia de lo que actualmente conocemos como internet, el cual revolucionó la forma de transportar y dar acceso a la información creada alrededor del mundo.

Inicialmente el internet no era tomado como una opción para el manejo de la información por no tener una estructura estable y fácil de vulnerar, por lo cual ARPA¹ tomó como proyecto

¹ Advanced Research Projects Agency: Agencia Norte Americana ahora diseñadora del precursor del internet.

principal el diseño de una red que no tenga estas falencias, principalmente porque se manejaba una topología de tipo estrella la cual tiene un nodo central que controla los demás dispositivos conectados a éste, por este motivo, al ser vulnerado este equipo toda la red sufría daños; con base en esta deficiencia la organización puso empeño en crear una estructura más estable y con la llegada de los protocolos de comunicación, principalmente TCP/IP², el envío de mensajes como paquetes y la expansión de la red fue sencilla y ahora observamos la magnitud del internet que se ha convertido en una red de redes con acceso desde cualquier lugar del mundo. (Rouse, 2005)

Por otro lado los equipos que tienen acceso a la red antes mencionada, fueron a la par en el avance tecnológico logrando conectarse de una manera rápida y eficaz a través del protocolo antes mencionado (TCP/IP) y procesando esta información para el que el usuario pueda acceder a ella; empezando por equipos los cuales se conectaban a la red a través de cableado dentro de un área limitada (routers, computadores portátiles o de escritorio) y otros conectándose a través de redes inalámbricas (sistemas de posicionamiento global, dispositivos móviles), dieron acceso a esta información y abrieron campo al diseño de aplicaciones que gestionen estos datos y mantengan una conexión constante a la red de internet.

2.1. Sistemas operativos.

Un Sistema Operativo es un programa que gestiona el hardware³ de un dispositivo, brinda una plataforma base para ejecutar programas, aplicaciones, cálculos, soporte para periféricos, etc.

² Transmission Control Protocol/Internet Protocol: Protocolo básico de comunicación del internet, el cual maneja paquetes de información transmitidos a través del internet con un destino.

³ Hardware: Grupo de elementos materiales que forman una computadora o sistema.

Se comporta como intermediario entre el usuario y el hardware. El software⁴ por lo general viene precargado, dependiendo de las especificaciones se pueden realizar cambios, actualizaciones e instalación de uno o varios sistemas operativos siempre que el hardware cumpla con todos los requisitos del sistema. (McIver McHoes & Ida, 2011)

En la década 1950 aparecieron los primeros sistemas operativos su función principal fue realizar más fluida la interacción usuario y máquina, el tipo de procesamiento se realizaba por lotes de un solo flujo. Existía poca interacción del usuario con el proceso en ejecución y se lo realizaba mediante el uso de tarjetas perforadas, se podía desarrollar y ejecutar programas en lenguaje Fortran⁵ o ensamblador⁶. El tipo de almacenamiento era temporal con la finalidad de disminuir tiempo, el sistema operativo almacena la información utilizando un proceso llamado spooling⁷ el cual se encarga de transferir la información al buffer⁸. (McIver McHoes & Ida, 2011)

El desarrollo de sistemas operativos de tipo compartido y con capacidades de multiprocesamiento fueron denominados como la segunda generación y se dio lugar a mediados de los sesenta. El multiprocesamiento brinda la capacidad de utilizar varios procesadores en un solo sistema con el objetivo de mejorar el rendimiento de la máquina, para evitar interferencias en los distintos procesos se implementan dos tipos de arquitectura NUMA⁹ la cual no proporciona

⁴ Software: Grupo de programas y reglas para la ejecución de tareas en un dispositivo.

⁵ Fortran: Lenguaje de programación de alto nivel.

⁶ Ensamblador: Lenguaje de programación de bajo nivel.

⁷ Spooling: Simultaneous Peripheral Operations On-Line: Proceso de transferir datos poniéndolos en un área temporal de trabajo donde otro programa puede acceder a su proceso.

Buffer: Espacio determinado del disco que almacena información de forma temporal utilizando un sistema FIFO.

⁹ NUMA: Non-Uniform Memory Access.

un acceso uniforme a todas las zonas de memoria ya que hace uso de un bus de datos privado designado al proceso en ejecución, una de las características de esta arquitectura es que nos ofrece “escalabilidad ” de MPP¹⁰ muchas CPUs¹¹ independientes corren en paralelo para procesar un solo programa, es muy similar al procesamiento simétrico.

En la arquitectura SMP¹² el multiprocesamiento es en paralelo ya que todos los procesadores acceden a todas las zonas de la memoria y cualquier procesador puede realizar operaciones de cualquier otro, haciendo que las aplicaciones se dividan en subprocesos y éstos se ejecuten de manera concurrente. (McIver McHoes & Ida, 2011)

En 1970 se dio un gran avance al concepto de sistema operativo con el desarrollo de Multics¹³ y UNIX¹⁴, software que fue desarrollado en lenguajes de alto nivel por lo cual puede ser modificado para ser ejecutado en otros dispositivos volviéndolo un sistema operativo portable. Uno de los mayores problemas era el costo de desarrollo, su complejidad y la gran demanda de recursos necesarios para que el sistema funcione de manera adecuada. (McIver McHoes & Ida, 2011)

En la década de 1980 el desarrollo mejoró notablemente en relación costo/rendimiento, el hardware era más flexible permitiendo incluir muchas de las funcionalidades del sistema operativo

¹⁰ MPP: Massive Parallel Processing.

¹¹ CPU: Unidad central de proceso.

¹² SMP: Symmetric Parallel Processing.

¹³ Multics: Multiplexed Information and Computing Service.

¹⁴ UNIX: Sistema operativo portable.

directamente en el hardware, dando paso a un nuevo concepto llamado firmware¹⁵, sirve para indicarnos que un programa se encuentra de forma permanente en la memoria ROM.¹⁶ (McIver McHoes & Ida, 2011)

El multiprocesamiento permitió que se puedan diseñar lenguajes mucho más complejos con el objetivo de coordinar actividades de los procesadores que se encargaban de un solo trabajo, el resultado hizo posible la ejecución de programas en forma paralela, el multiprocesamiento con el paso del tiempo se transformó en una característica principal de la mayoría de dispositivos que cuentan con un sistema operativo. (McIver McHoes & Ida, 2011)

La gran demanda de computadores personales y la evolución en la velocidad de las comunicaciones originó el cambio hacia sistemas de redes y procesamiento distribuido, permitiendo que usuarios tengan la posibilidad de compartir recursos de hardware y software en distintas ubicaciones, este tipo de sistemas necesitaban un sistema operativo diferente, capaz de administrar varios conjuntos de subsistemas, como hardware que se encuentre localizado a kilómetros de distancia. (McIver McHoes & Ida, 2011)

Gracias a los sistemas operativos de red los usuarios podían acceder y manipular archivos en equipos de red distribuida sin importar su ubicación, estos sistemas eran muy parecidos a los sistemas operativos de procesador único ya que cada equipo ejecutaba su propio sistema operativo local y contaban con sus propios usuarios, mediante la adición de un controlador de interfaz de red

¹⁵ Firmware: Software encargado de gestionar físicamente el hardware.

¹⁶ ROM: (Read only memory).

con software de bajo nivel se logró controlar dicho sistema operativo local, tanto los programas como registros remotos y gestionar los archivos del equipo, a pesar de estas características añadidas la estructura básica del sistema operativo de red se asemeja a la de un sistema independiente. (McIver McHoes & Ida, 2011)

Los sistemas operativos de red presentan algunas desventajas como por ejemplo el requerimiento de más algoritmos de planificación, presencia de retrasos en la comunicación dentro de la red ocasionando que los algoritmos de planificación tengan que operar con información incompleta. (McIver McHoes & Ida, 2011)

A inicios de los noventa la gran demanda de servicios de internet desencadenó el incremento de la capacidad de red. La Red Global Mundial o World Wide Web creada por Tim Berners- Lee un científico de la computación británica, llamado el padre de la web logró establecer la primera comunicación entre un cliente y un servidor utilizando el protocolo HTTP¹⁷. Berners diseñó en un pedazo de papel el esquema que hizo posible realizar esta comunicación y lograr el acceso a internet de todos los usuarios de computadora alrededor del mundo. En diagrama mostrado a continuación representa el esquema para comunicación al internet. (McIver McHoes & Ida, 2011)

¹⁷ HTTP: Hypertext Transfer Protocol.

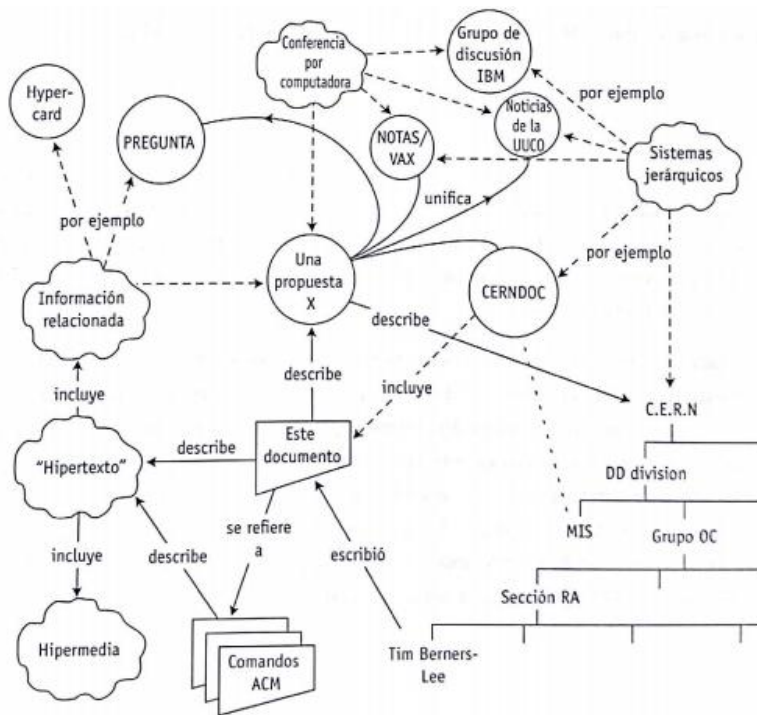


Diagrama II- 1 Esquema Internet

Elaborado por: Obando B. - Vega C.

Basado en: (McIver McHoes & Ida, 2011)

Siendo el acceso a la red y el uso de correo electrónico, servicios indispensables; estos se convirtieron rápidamente en características fundamentales de la mayoría de sistemas operativos. No obstante, el incremento de uso en estas redes generó la necesidad de mejorar la seguridad para proteger el acceso no autorizado al software del equipo conectado a internet. (McIver McHoes & Ida, 2011)

Los sistemas operativos siguen evolucionando continuamente para cumplir con las necesidades de los usuarios, en esta década se dio el incremento de aplicaciones multimedia las cuales demandaban muchos más recursos, potencia, flexibilidad, compatibilidad de dispositivos y

sistemas operativos. El consumo de este tipo de contenido fue el primer paso para implementar la creación y edición de archivos multimedia siendo necesario el uso de periféricos, como por ejemplo conexiones con cámaras digitales, lector de discos, altavoces, monitores adicionales, dispositivos para proyección, impresoras, micrófonos, cámaras web y otros. Muchos de estos periféricos necesitan software adicional para poder utilizarlos de manera adecuada en nuestro ordenador. (McIver McHoes & Ida, 2011)

Las aplicaciones multimedia requieren un aumento en las capacidades de almacenamiento y el sistema operativo debe manejarlo con elegancia. La reproducción de un segundo en video de calidad media y en pantalla completa requiere un promedio entre 30 y 50 MB¹⁸ de almacenamiento, para optimizar el acceso a este tipo de archivos se crearon programas que nos permitan comprimir los datos y minimizar el espacio en memoria del archivo disminuyendo el uso de hardware de los equipos. (McIver McHoes & Ida, 2011)

Para el siglo XX los sistemas operativos y dispositivos incrementaron mucho más su flexibilidad, confiabilidad y velocidad, de esta forma los equipos fueron capaces de permitir la instalación y ejecución de varios sistemas operativos al mismo tiempo y compartiendo recursos, este concepto fue llamado virtualización. (McIver McHoes & Ida, 2011)

La virtualización es la creación de particiones en un servidor único, cada partición puede alojar un sistema operativo diferente, convirtiendo un equipo físico en varios virtuales, para hacer realidad esta función el sistema operativo principal debe contar con un administrador intermedio

¹⁸ MB: Abreviatura para Megabyte, unidad de medida para datos informáticos.

el cual supervisa y gestiona cada sistema alojado en las particiones creadas, un solo procesador es capaz de ejecutar 64 sistemas independientes de manera simultánea siempre y cuando el procesador cuente con un hilo¹⁹ asignado para cada sistema. (McIver McHoes & Ida, 2011)

La velocidad de procesamiento se incrementa a la par del desarrollo de procesadores con más núcleos en un mismo chip, en el pasado se utilizaban obleas de silicio como base para los circuitos del chip y solo permitían alojar un solo CPU, con la llegada de los procesadores de doble núcleo en un solo chip permitió la ejecución de dos o varios conjuntos de cálculos al mismo tiempo optimizando el proceso y reduciendo el tiempo a la mitad. (McIver McHoes & Ida, 2011)

Los sistemas operativos siguen evolucionando y agregando funciones rápidamente, en un futuro no muy lejano la tecnología actual será obsoleta ya que los desarrollos de nuevos componentes reemplazarán y mejorarán el concepto de ordenador y sistema operativo creando dispositivos con características únicas y alcanzando velocidades de procesamiento abismales. (McIver McHoes & Ida, 2011)

2.1.1. Tipos de Sistemas Operativos para Computadoras Personales

2.1.1.1. Windows

Actualmente este sistema operativo es el más usado en ordenadores a nivel mundial, la primera versión de Windows 1.0 fue presentada en el año 1985, este sistema contaba con una

¹⁹ Hilo: secuencias de instrucciones.

interfaz gráfica GUI²⁰ y con una arquitectura x86²¹ en su sistema MS-DOS²², las primeras versiones de Windows no fueron muy populares ya que estaban limitadas por problemas legales que tenían con Apple. La primera versión que realmente abarcó un gran nicho en el mercado fue Windows 3.0, la multitarea y su sistema de ventanas implementado en su interfaz gráfica hicieron que sea un digno rival para Apple y su Macintosh.

Posteriormente se lanzaron actualizaciones del sistema operativo y nuevas versiones como Windows NT²³ destinado a estaciones de trabajo y servidor de red. En 1995 se presentó “Windows 95” con una renovada interfaz GUI híbrida entre 16 y 32 bits, el objetivo de esta versión fue sustituir a MS-DOS, entre sus principales características están la implementación de puertos tipo USB²⁴ y un sistema de archivos FAT32.²⁵ (Carcerca, 2001)

En 1998 se presentó Windows98 o también llamado Memphis esta versión fue diseñada específicamente para equipos de oficina y del hogar, contaba con una conexión compartida a internet la cual nos permitía compartir archivos a varios ordenadores, esta característica fue conocida como LAN²⁶; este sistema operativo brindó un mejor soporte para AGP²⁷ también se integró un navegador de Internet, la integración de este navegador llevó a Windows a ser acusado de monopolio. (Carcerca, 2001)

²⁰ GUI: Graphical User Interface.

²¹ X86: Termino para definir la arquitectura de un procesador marca Intel.

²² MS-DOS: MicroSoft Disk Operating System.

²³ NT: New Tecnology.

²⁴ US: Universal Serial Bus.

²⁵ FAT32: file allocation table.

²⁶ LAN: Local Area Network.

²⁷ AGP: Accelerated Graphics Port.

Con la llegada del año 2000 nos trajo una nueva versión de sistema operativo conocida como Windows 2000, ya no solo enfocada en hogares y equipos de oficina, este también contaba con una versión para servidores, el sector empresarial se vio atraído por la facilidad en la cual un usuario podía crear su propia cuenta, asignar recursos y privilegios, actuar como un servidor FTP²⁸ o contar con un sistema DNS²⁹. La principal característica de este sistema fue Active Directory³⁰, esta herramienta nos permitía administrar la infraestructura de una organización.

Windows 2000 contó con varias versiones destinadas a diferentes sectores y servicios como por ejemplo su versión Professional, Server, Avance Server y Datacenter Edition. En el mismo año se lanzó WindowsMe³¹, destinada a usuarios caseros, fue el último sistema operativo que utilizó kernel³² Windows y en utilizar MS-DOS. (Carcerca, 2001)

El año 2001 fue uno de los más trascendentales para Microsoft ya que en este se presentó Windows XP el sistema operativo más importante y demandado alrededor del mundo, las letras “XP” provienen de eXPerience, cuenta con soporte para procesadores x86 y se considera que existen más de 400 millones de copias funcionando. Microsoft dejó de brindar soporte a este sistema operativo en abril del 2014. (Carcerca, 2001)

Tras 6 años Microsoft presentó Windows Vista, este sistema operativo contó con muchos retrasos en su desarrollo y fue uno de los menos agradables para el usuario a pesar de contar con una interfaz gráfica renovada llamada Windows Aero, mucho más cuidada que su antecesor y no

²⁸ FTP: File Transfer Protocol.

²⁹ DNS: Domain Name System.

³⁰ Active Directory: Implementación de servicio de directorio.

³¹ WindowsMe: Millenium Edition

³² Kernel: Software que constituye parte del Sistema operativo, núcleo.

fue muy aceptada por los usuarios, Vista incluyó una gran cantidad de programas pre-instalados y el consumo de recursos por su interfaz hacían que este sistema operativo sea lento y presente incompatibilidad o mal funcionamiento de software para XP. (Carcerca, 2001)

Microsoft trabajó muy duro para enmendar los problemas que presentó Windows Vista y en tres años plazo para el desarrollo de su sucesor, en octubre del 2009 fue presentado Windows 7, su carta de presentación fue un arranque rápido, control de cuentas más eficiente, soporte para ventana Multi-touch³³, una mejora en la administración de redes Grupo Hogar y con menos aplicaciones preinstaladas, hicieron de este un sistema muy estable, rápido y agradable. (Carcerca, 2001)

A principios de 2010 los dispositivos con pantallas táctiles eran de gran demanda en el mercado es por eso que Microsoft apostó a esta tendencia y presentó Windows 8, un sistema operativo enfocado a ser utilizado de forma táctil y tradicional, esto lo consiguió con el desarrollo de una nueva interfaz llamada “Metro” la cual funcionaba a base de recuadros o mejor conocida como “Baldosas”, esta interfaz hizo desaparecer el famoso menú y botón de inicio que caracterizaban a Microsoft lo que ocasionó disgusto por parte de los usuarios ya que esta nueva interfaz a pesar de ser revolucionaria era muy difícil de manejar, poco intuitiva y carecía de profundidad. (Carcerca, 2001)

En 2014 Microsoft presentó Windows 10, actualmente es el último sistema operativo presentado por parte de Microsoft en el cual retornó el botón de inicio e integró una interfaz tanto

³³ Multi-Touch: Pantalla con tecnología táctil sensible a múltiples puntos a la vez.

para uso en pantallas táctiles como en tradicionales, Windows 10 cuenta con núcleo NT 6X³⁴ cuyo objetivo es tener un solo sistema operativo para diversos tipos de dispositivos, desde un teléfono hasta una consola de videojuegos; soporta distintas arquitecturas de 32 o 64 bits y presentó su asistente de voz personal llamado Cortana. (Carcerca, 2001)

2.1.1.2. Linux

El nombre de Linux es la combinación de Linus y Unix. Linus Torvalds es el nombre del creador de este sistema operativo y Unix es un sistema operativo desarrollado en 1969. En 1991 un estudiante de informática de origen finlandés llamado Linus Torvalds decide desarrollar un sistema operativo tomando como referencia a UNIX, pero con la característica de que este funcione en cualquier ordenador compatible, para que el sistema funcione correctamente el ordenador debía contar con un procesador i386 o superior ya que computadores más antiguos dificultaban el desarrollo de un sistema operativo que sea compatible con UNIX. (Miranda, 2002)

En la década de los 90 el internet era una tecnología revolucionaria y su gran expansión fue determinante para el desarrollo y aceptación de Linux, Internet facilitó el trabajo ya que muchas personas colaboraron con software que ya venían en UNIX, pero Linus no tenía como objetivo desarrollar un sistema con todos estos programas, su meta era crear un núcleo del Sistema Operativo que sea totalmente compatible con UNIX y que permitiera ejecutar todos sus programas

³⁴ NT 6x: Nombre y versión para interfaz gráfica de Sistema Operativo Windows.

desarrollados por la Free Software Foundation³⁵ la cual cuenta con licencia GNU³⁶. (Miranda, 2002)

Minix fue una versión reducida de Unix en la cual Linus se basó para la creación del núcleo, Minix fue desarrollado por el profesor Andy Tanenbaum y su fin fue educativo en el cual los estudiantes podían aprender de un sistema operativo real. La primera versión de Linux fue la 0.01 era un núcleo muy sencillo el cual solo podía leer y escribir archivos en la unidad de disquete. (Miranda, 2002)

El siguiente paso de Linus fue añadir funciones a su núcleo el cual para su versión 0.02 tenía la capacidad de ejecutar el intérprete de órdenes bash³⁷ y el compilador C³⁸, estas funciones eran básicas de GNU. Linux no contaba con ningún tipo de soporte, distribuciones ni documentación, pero Linus publicó las fuentes de Linux con el objetivo de que cualquier persona interesada pueda leer su código, modificarlo y mejorarlo consiguiendo que Linux pudiera ejecutar muchas más aplicaciones de GNU. (Miranda, 2002)

Con el gran aporte de la comunidad de desarrolladores, Linus lanzó la versión 0.10 y posteriormente alcanzó la versión 0.95, Linux se encontraba a un paso de ser estable y lanzar la versión 1.0 siendo esta teóricamente completa y sin errores, todo esto sucedió en 1992 y un año después nació Linux 1.0.

³⁵ Free Software Foundation: organización creada por Richard Stallman en 1985.

³⁶ GNU: También conocida GPL General Public License.

³⁷ Bash: conjunto de parámetros para la administración y configuración de un sistema.

³⁸ Compilador C: Programa informático para traducir lenguajes de programación.

Actualmente Linux cuenta con una gran cantidad de distribuciones, exactamente 31; cada una dirigida a diferentes sectores y usuarios, su gran comunidad sigue desarrollando distintas versiones y mejorando las existentes, Linux ocupa un espacio importante en el desarrollo de tecnologías, está situado como uno de los sistemas operativos más prometedores.

2.1.1.3. Mac Os

Macintosh Operating System o también llamado Mac es un sistema operativo creado por la compañía Apple fundada por Steve Jobs y Stephen Wozniak, este sistema operativo fue diseñado para los ordenadores Apple Macintosh. Mac Os sobresalió por su potencia, estabilidad y facilidad. Fue uno de los primeros sistemas operativos comerciales en contar con una interfaz gráfica dirigida al usuario común, su primera versión se presentó en enero de 1984 y era llamado el Sistema x.x. (Falla, 2007)

Los sistemas operativos Mac Os contaban con una gran parte de su sistema, almacenada en la memoria ROM ubicada en la placa madre cuyo objetivo era consumir menos espacio del que ofrecían los disquetes los cuales almacenaban el resto del sistema, esta característica se mantuvo en varias versiones hasta 1991, fue cuando Mac Os podía arrancar directamente desde la memoria ROM. Posteriormente el sistema operativo se dividió en dos familias el Mac Os “Clásico” que abarca desde su primera versión hasta la novena que fue presentada en 2001 y Mac OS X que es basado en BSD UNIX³⁹, específicamente en el sistema operativo NexStep, a pesar de contar con

³⁹ BSD UNIX:(Berkeley Software Distribución.

un nuevo desarrollo y núcleo esta versión conservó la mayoría de funciones de Mac OS 9. (Falla, 2007)

La X hace referencia a su décima entrega, a partir de esta versión en su mayoría llevan sobrenombres de felinos y actualmente cuenta con 13 distribuciones, su última versión lanzada al mercado es 10.12 Sierra, fue presentada en septiembre de 2016, fecha especial para la presentación de nuevos productos y software por parte de la empresa Apple, entre sus características fundamentales y que la diferencian de la competencia, contamos con una interfaz única en el mercado, el sistema operativo preferido por diseñadores, su estabilidad y seguridad son garantía por parte de Apple. (Falla, 2007)

2.1.2. Sistemas operativos dispositivos Móviles

2.1.2.1. Android

En 2003 con el auge de los dispositivos móviles inteligentes, Chris White, Andy Rubin y Rich Miner deciden fundar Android Inc. con el propósito de crear un sistema operativo para móviles el cual este basado en Linux. Google se interesó mucho en Android por su concepto a pesar de no ser reconocido lo adquirió por 50 millones de dólares en el año 2005, el equipo de desarrolladores de Android ya a cargo de Google y con la colaboración de otros fabricantes de dispositivos móviles como HTC⁴⁰ trabajaban en un Sistema Operativo fácil de usar y optimizado para utilizar pantallas táctiles. (Hernández, 2016)

⁴⁰ HTC: High Tech Computer Corporation.

El conglomerado de empresas que apoyaban la iniciativa de Google contó con el apoyo de muchos más fabricantes, para 2007 un total de 35 firmas se unieron a esta iniciativa llamada Open Handset Alliance⁴¹, esta alianza comercial tenía como objetivo la creación de un sistema operativo con estándares abiertos, esto generó que muchas más empresas se unieran por el potencial que contaba Android y su enfoque de uso libre. (Hernández, 2016)

Android 1.0 Apple Pie fue la primera versión estable lanzada al mercado en el 2007 siendo totalmente gratuito y Open Source entró a competir directamente con iOS de Apple en ese entonces software líder a nivel mundial en dispositivos móviles. Android estaba desarrollado con Kernel Linux 2.6 y la versión del SDK⁴² numero 23 fue presentada ese mismo año, Apple Pie tenía muchas cosas en que mejorar, pero aun así contaba con características que la distinguían de otros sistemas, como un menú desplegable de notificaciones, una tienda de aplicaciones con un catálogo limitado, pero en su mayoría gratuito. (Hernández, 2016)

En 2008 es lanzado el primer smartphone⁴³ con sistema operativo Android, el modelo de teléfono fue el HTC dream y sus ventas superaron las expectativas vendiendo más de un millón de equipos en Estados Unidos y alrededor de 600.000 en el resto del mundo, haciendo que el debut de Android en el mercado sea un éxito rotundo, La primera actualización se dio unos meses después de su lanzamiento, Android 1.1 o también conocido como Banana Bread corrigió

⁴¹ Open Handset Alliance: Alianza comercial de 84 compañías para desarrollar estándares en dispositivos móviles.

⁴² SDK: software development kit.

⁴³ Smartphone: Teléfono celular con funciones avanzadas similares a las de un ordenador.

numerosos fallos y bugs⁴⁴ de su antecesor, las actualizaciones automáticas y rápidas hacían de este un proceso muy fácil para el usuario logrando mantener el interés. (Hernández, 2016)

En el primer trimestre del 2009 fue presentado Android 2.0 Enclair, versión con cambios significativos tanto en su arquitectura como en características, este sistema operativo era capaz de funcionar en pantallas de distintos tamaños y resoluciones permitiendo a los fabricantes diversificar su oferta y mejorar la experiencia en sus dispositivos, la gran variedad de funciones nativas del sistema como compatibilidad con HTML5⁴⁵, rediseño de interfaces y compatibilidad con programas de terceros ocasionó que más fabricantes opten por Android como sistema operativo preferido para sus equipos, cada compañía desarrollaba su propias Roms⁴⁶. Esto marcaba la diferencia entre compañías, pero permitió la compatibilidad de las aplicaciones. (Hernández, 2016)

Samsung apuesta por Android e incluye este como sistema operativo para la presentación del teléfono más exitoso en ese entonces alcanzando un total de 10 millones de ventas a nivel mundial junto a las ventas de los equipos HTC, Android logró que por primera vez supere la cuota del mercado a iOS. En 2010 ya con nueva versión 2.2 denominada Froyo aparecen en el mercado equipos móviles de mayor tamaño llamadas tabletas, el concepto de estos equipos era similar al de un smartphone, pero con mayores dimensiones mejorando la experiencia del usuario y compitiendo directamente con el iPad. (Hernández, 2016)

⁴⁴ Bugs: Error o defecto de software o hardware que ocasiona un funcionamiento incorrecto.

⁴⁵ HTML: Lenguaje de marcado utilizado para el desarrollo de páginas web.

⁴⁶ Roms: contiene archivos y código necesario para que el dispositivo arranque y pueda ejecutar Android.

Android 2.3 Gingerbread fue la versión más extendida y populares desarrolladas por Google, esta versión incluyó un soporte para cámaras frontales y mejoró los modos de conexión entre dispositivos brindando soporte a chips NFC⁴⁷, Gingerbread fue un gran aliado de los desarrolladores ya que esta versión otorgó acceso de bajo nivel facilitando el desarrollo de juegos capaces de utilizar el hardware completo del equipo y mejorar la experiencia para el usuario, posteriormente fue lanzado Android 3.0 Honeycomb siendo esta una actualización específica para tabletas lo cual hacia que Gingerbread sea la versión más actual de su sistema para móviles hasta 2011. (Hernández, 2016)

Google seguía puliendo su sistema operativo y para finales del 2011 estaba preparando la versión de Android que revolucionó el mercado, esta fue Ice Cream Sandwich o Android 4.0 su gran éxito le hizo escalar puestos superando a BlackBerry OS y convirtiéndose en el sistema operativo móvil más usado en el mundo. Ice Cream Sandwich contaba con una interfaz denominada Holo, sistema de notificaciones mejorado, funciones de reconocimiento de rostro, soporte para Stylus⁴⁸, pero lo que realmente atrajo consumidores fue un sistema de multitarea renovado y nunca antes visto en Smartphones. (Hernández, 2016)

Tras dos presentaciones más Android 4.1 Jelly Bean y Android 4.4 Kit Kat en 2012 y 2013 respectivamente, en 2014 sale a la luz Android 5.0 Lollipop integrando un estándar en el diseño no solo dirigido para móviles si no a varios dispositivos y páginas web, este estándar es conocido como Material Design, interfaz adaptable para cualquier tamaño de dispositivo, fluida y muy

⁴⁷ NFC: Near field communication.

⁴⁸ Stylus: lápiz táctil.

intuitiva. Lollipop incrementó el desempeño del dispositivo y a su vez mejoró la autonomía de la batería con el proyecto Volta⁴⁹, también fue el primer sistema operativo de Google en dar soporte a sistemas de 64 bits permitiendo incluir una mayor capacidad en la memoria RAM⁵⁰. (Hernández, 2016)

En 2015 se anunció oficialmente Android MarshMallow incluyendo nuevas funcionalidades ya presentes en el mercado como soporte USB Tipo-C, lector de huella digital y la inclusión de Android Pay⁵¹. Google incluyó un sistema de permisos rediseñado y adaptado para todas sus tecnologías y servicios permitiendo al usuario decidir que funciones o características otorga a cada una de sus aplicaciones, esto fue fundamental ya que impedía a las aplicaciones acceder a contenido privado o hacer un mal uso del micrófono, cámara, función o cualquier periférico del dispositivo. (Hernández, 2016)

La última versión conocida actualmente es Android 7.0 Nougat muestra la evolución de un sistema operativo a través de los años, presentada en 2016 e incrementando sus posibilidades, brinda soporte para más de 100 idiomas nuevos, acceso a instant apps⁵², opciones de bloqueo para aplicaciones que se ejecuten en segundo plano, función de multiventana nativa entre otras han hecho que Android sea el sistema operativo robusto y muy seguro preferido por los usuarios de dispositivos móviles aventajando a la competencia con una gran diferencia. En 2017 se espera la presentación de su nueva versión llamada Android 8.0 Oreo. (Hernández, 2016)

⁴⁹ Proyecto Volta: Proyecto que mejora la tecnología en baterías para dispositivos móviles.

⁵⁰ RAM: Random Access Memory.

⁵¹ Android Pay: Método de pago electrónico utilizando sistema operativo Android.

⁵² Apps: aplicaciones que no necesitan ser instaladas ni descargadas.

2.1.2.2. iOS

En 2007 la empresa Apple Inc. con Steve Jobs a la cabeza decidieron poner su nombre en la historia una vez más creando un nuevo dispositivo llamado iPhone y por consecuencia un sistema operativo con identidad propia diseñado para dispositivos móviles llamado iOS. El iPhone ha ido evolucionando a través de los años no solo en su hardware sino también en su software, añadiendo funciones y mejorando el desempeño de sus equipos a tal punto que hoy forma parte no solo de iPhone, también lo podemos encontrar en el iPad y iwatch, con diferentes características y funcionalidades, pero en esencia el mismo objetivo, ser un sinónimo de calidad y utilidad para los usuarios como para Apple. (Rus, 2016)

En enero del 2007 es presentado el iPhone un dispositivo único y alucinante, el cual marcó una nueva línea de productos que posteriormente serian llamados smartphones. Este dispositivo contaba con la primera versión del sistema operativo llamado iPhone OS, el cual contaba con las funciones principales de un teléfono, una versión de Safari adaptada para móviles y una app para YouTube. Era un sistema muy sencillo con la capacidad de dar soporte a una pantalla multi-touch capacitiva, por estas razones el año siguiente se presentó una actualización que añadió varias funciones nuevas como el servicio de correo electrónico, mapas, un block de notas y la posibilidad de que el usuario personalice su dispositivo moviendo y creando nuevas pantallas para las aplicaciones. (Rus, 2016)

iOS 2 o iPhone OS 2 fue la segunda versión para el iPhone presentada en 2008, año el que también se presentó el SDK para iPhone, esta nueva entrega mostraba al mundo su nueva tienda virtual con aplicaciones para el iPhone y iPod llamada App Store. iPhone OS 3 llegó en el 2009, su lanzamiento fue en WWDC⁵³. Contó con varias novedades como el control por voz, la posibilidad de enviar mensajes multimedia, funciones de copiar, cortar y pegar entre otras. (Rus, 2016)

En 2010 fue presentado el primer iPad, como consecuencia Apple decidió cambiar el nombre de su sistema operativo a iOS, en ese entonces ya eran tres los dispositivos en contar con iOS por lo cual la evolución de este sistema fue necesaria y consiguiente fue presentado iOS 4, sistema en que la multitarea era ya una función nativa y muy esencial para versiones posteriores. En 2011 los asistentes de voz marcaron una tendencia en el avance de la tecnología y tuvo como protagonista a Siri, este asistente virtual fue presentado como parte de iOS 5 dando la posibilidad al usuario de interactuar con el dispositivo por medio de la voz. Siri contaba con una inteligencia artificial muy avanzada para su época y nunca antes vista la cual nos permitía realizar todo tipo de funciones tan solo utilizando comandos de voz. (Rus, 2016)

iOS 6 y la inclusión de su propio sistema de navegación llamado Apple Maps y con el objetivo de eliminar los servicios de Google del terminal fue la carta de presentación de esta versión del sistema en 2011, Apple Maps terminó siendo un desastre en Apple con varios errores de programación y un servicio mediocre mostrando ubicaciones erróneas y rutas incompletas o inexistentes, hicieron que los usuarios se vean obligados a instalar Google Maps en sus dispositivos. Tras el fracaso de esta versión Apple preparó iOS 7 esperando que este sea un sistema

⁵³ WWDC: Worldwide Developers Conference.

más estable y atractivo visualmente ya que adaptó a su interfaz gráfica un estado de colores degradados y una nueva barra de accesos directos desplegable desde la parte inferior de la pantalla, fue un mejor sistema y aprendió de los errores presentados de su antecesor. (Rus, 2016)

Un sistema profesional fue el lema de iOS 8, puliendo mucho más su interfaz , presentó una serie de mejoras para que no sea un sistema operativo tan limitado y así incrementando las posibilidades de conexión, logrando que los usuarios dejaran de lado el ordenador para trabajar directamente en su dispositivo móvil de preferencia como el iPad, gracias a su gran pantalla mejoró la experiencia notablemente, siendo el que más se asemejó a tener un ordenador totalmente portable. (Rus, 2016)

En 2015 Apple presentó un sistema operativo mucho más maduro, para muchos fue una versión casi perfecta por la serie de funciones que fueron añadidas y las posibilidades de iCloud mejoradas haciendo de este un sistema muy seguro, pero con grandes posibilidades de ser personalizado. Apple con el objetivo de recuperar terreno arrebatado por parte de Android, en 2016 lanzó la décima versión de su sistema operativo con un entorno mucho más amigable para los desarrolladores, un centro de widgets intuitivo y la posibilidad de eliminar aplicaciones nativas incluidas por parte de Apple, esto marcó un antes y un después para iOS. (Rus, 2016)

2.1.2.3. Windows Phone

Siendo los smartphones la tecnología de la última década son un mercado nuevo y con gran potencial para la inversión de compañías tecnológicas como lo hizo Google y Apple con Android

e iOS respectivamente, Microsoft no se quería quedar atrás y desarrolló su propio sistema operativo dedicado para smartphones llamado Windows Phone, con un 12% de la cuota de mercado mundial y el tercer sistema operativo más usado, este desapareció rápidamente del mercado, por la gran diferencia en funcionalidades y su pobre tienda de aplicaciones ocasionaron su fracaso. (Rabasco, 2015)

Windows Mobile fue la primera versión presentada por Microsoft en el año 2000 este sistema estaba desarrollado para los Pocket PC ⁵⁴los cuales eran equipos que se asemejaban a un ordenador y dirigidos al sector empresarial, era un sistema innovador pero que nunca llegó a madurar. En 2002 es presentado Windows Mobile 2002 una actualización de su sistema que no tuvo mucha relevancia al no incluir nuevas funcionalidades ni mejorar el desempeño en sus equipos. (Rabasco, 2015)

Windows decide cambiar de rumbo y en 2005 presenta la versión 5.0 de su sistema mejorado al contar con programas como Office Mobile⁵⁵ e incluir el reproductor Windows Media 10 Mobile⁵⁶, fue un sistema prometedor pero el mundo no estaba listo para utilizar este tipo de tecnologías ya que no estaba acorde a la economía para la época. Microsoft seguía trabajando en su sistema y lanzó dos nuevas versiones Windows Mobile 6 y 7 permitiendo a los usuarios poder manipular los equipos por medio de una pantalla táctil esta vez capacitiva a diferencia de sus versiones anteriores que contaban en su mayoría con pantallas resistivas⁵⁷. (Rabasco, 2015)

⁵⁴ Pocket PC: Computador de bolsillo.

⁵⁵ Office Mobile: Software para gestión documentos, hojas de cálculo, presentaciones etc.

⁵⁶ Windows Media 10 Mobile: Reproductor de contenido multimedia.

⁵⁷ Pantallas Resistivas: Tecnología para pantallas táctiles que funcionan con presión.

La llegada de Windows Phone 7 un sistema que podía haber sido exitoso en 2008 pero por su mala gestión y la lentitud en el desarrollo llegó tarde al mercado, para ese entonces competía directamente con el iOS 4 y Android 2.3 Gingerbread, sistemas ligeros y que contaban con interfaces mucho más amigables e intuitivas, hizo que los usuarios se inclinen por estas dos alternativas. El gran fracaso de este sistema se vio reflejado en sus funcionalidades como por ejemplo tareas básicas de copiar y pegar fueron implementadas en 2011 mientras que sus competidores ya contaban con estas funciones tres años antes. (Rabasco, 2015)

Para 2012 la idea de Microsoft era entrar de lleno al mercado móvil y decide comprar Nokia, transacción que no se terminó hasta el 2014. Tras la llegada de Windows 8 a los ordenadores Microsoft decide actualizar su sistema móvil y lo presenta ese mismo año. Su interfaz metro aclamada por la comunidad tecnológica prometió un gran desempeño, pero no cumplió y una tienda de aplicaciones lenta y escasa de contenido siguieron perjudicando a la empresa estadounidense y consecuente a Nokia haciendo que las ventas de sus equipos caigan drásticamente. (Rabasco, 2015)

Microsoft vuelve a cambiar el nombre de su sistema esta vez con la estrategia de integrar todos los productos en un mismo sistema llamado Windows 10, dicho sistema tuvo gran éxito en ordenadores y su línea de tabletas Surface, pero en el sector móvil seguían en el camino erróneo a pesar de contar con toda la gama Lumia de Nokia que cada vez más se acercaba a su fin. Actualmente Microsoft no da su brazo a torcer, en 2017 planea una nueva estrategia para triunfar con un dispositivo llamado Windows Surface Phone el cual contaría con Windows 10 y sería capaz

de correr todos los programas desarrollados para su versión de escritorio como si este se tratara de un ordenador.

2.2. Protocolos

2.2.1. ACK

Acuse de recibo, es un mensaje corto cuyo objetivo es informar al transmisor que los datos hayan llegado correctamente al destino esperado, el mensaje contiene información sobre el estado en que los datos han llegado. El acuse de reconocimiento es una señal entre procesos o dispositivos que establecen una comunicación. Cuando la fuente recepta una señal ACK, el dispositivo transmite el siguiente bloque de datos, si la señal ACK no es receptada correctamente se repite el envío de los datos o la transmisión finaliza dependiendo del protocolo por el cual los datos están siendo transmitidos. A continuación, el diagrama representa el funcionamiento del protocolo ACK (Rouse, 2006)

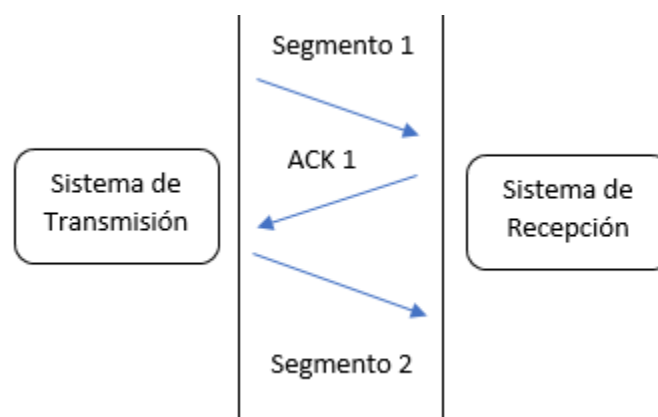


Diagrama II- 2 Esquema funcionamiento protocolo ACK

Elaborado por: Obando B. - Vega C.

Basado en: (Kioskea, 2014)

2.2.2. TCP

Protocolo de control de transmisión, protocolo de transporte que se encarga de administrar las conversaciones individuales entre emisor y receptor, el protocolo TCP primero fragmenta los mensajes HTTP en partes iguales denominados segmentos, para ser enviados al receptor, también es capaz de controlar el tamaño e intervalos en los que se pueden intercambiar mensajes. El protocolo TCP garantiza la transferencia de datos, a pesar de que usa el protocolo IP⁵⁸, que no incluye ningún tipo de control en la entrega de datos o paquetes. TCP tiene un ACK el cual permite al cliente y al servidor garantizar la recepción de datos. Al emitir un segmento, este es asociado con un número de secuencia. Al momento de recibir el segmento de datos, el receptor envía un mensaje ACK fijado con una etiqueta que equivale al número de secuencia anterior, tal como se muestra en la figura 5. (Kioskea, 2014)

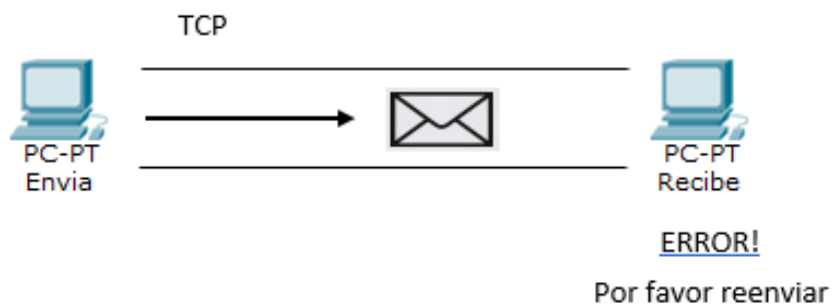


Diagrama II- 3 Representación protocolo TCP

Elaborado por: Obando B. - Vega C.

⁵⁸ IP: Protocolo para la transferencia de datos a través de internet.

2.2.3. HTTP

Protocolo de transferencia de hipertexto, encargado de controlar todas las transferencias de datos en la WWW⁵⁹. HTTP permite que tanto los clientes como servidores determinen el formato de los documentos que sean transferibles, si el receptor no puede acceder a los datos el usuario puede descargar un programa o complemento el cual le permita acceder al contenido.

Las cabeceras en HTTP brindan información sobre los objetos que se están transmitiendo a través de la web, la cabecera garantiza que un paquete sea entregado al receptor indicado, al estar basado en texto este protocolo no necesita de decodificación, tal como se muestra en la figura 6. (Bembibre, 2009)

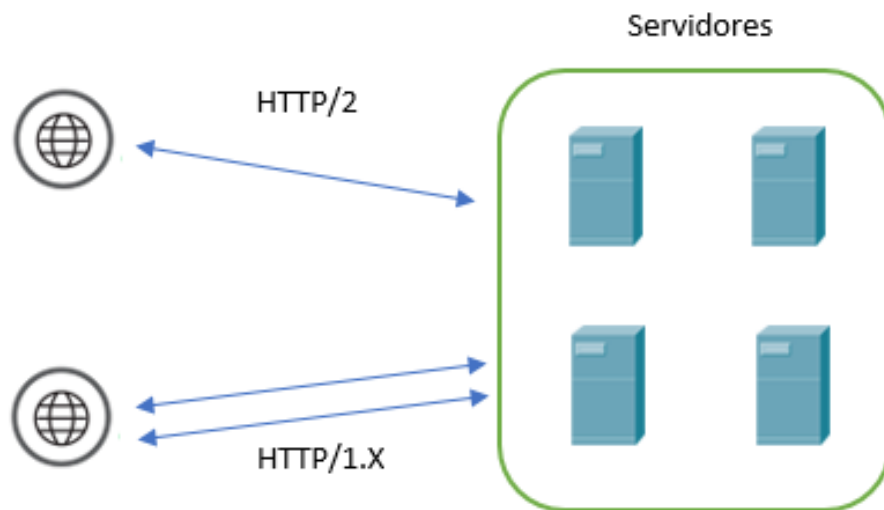


Diagrama II- 4 Transmisión protocolo HTTP

Elaborado por: Obando B. - Vega C.

Basado en: (Fores, 2016)

⁵⁹ WWW: Sigla de la expresión inglesa World Wide Web

2.2.4. XMPP

Protocolo extensible de mensajería y comunicación de presencia o también conocido como Jabber, es un protocolo abierto y extensible basado en XML⁶⁰, protocolo creado para control de mensajería instantánea, este protocolo fue desarrollado en el año 1998 por Jeremie Miller, a pesar de que en sus inicios no contaba con comunicaciones de servidor a servidor actualmente es un protocolo utilizado en aplicaciones de mensajería y VoIP⁶¹. Es un protocolo gratuito, abierto, público y sencillo, cuenta con estándares de aprobados por IETF⁶². XMPP cuenta con niveles de seguridad altos al ser aislado de la red pública y utilizar seguridad extra por capas en formatos SASL⁶³ y TLS⁶⁴, tal como se muestra en la figura 7. (Gonzales, 2014)

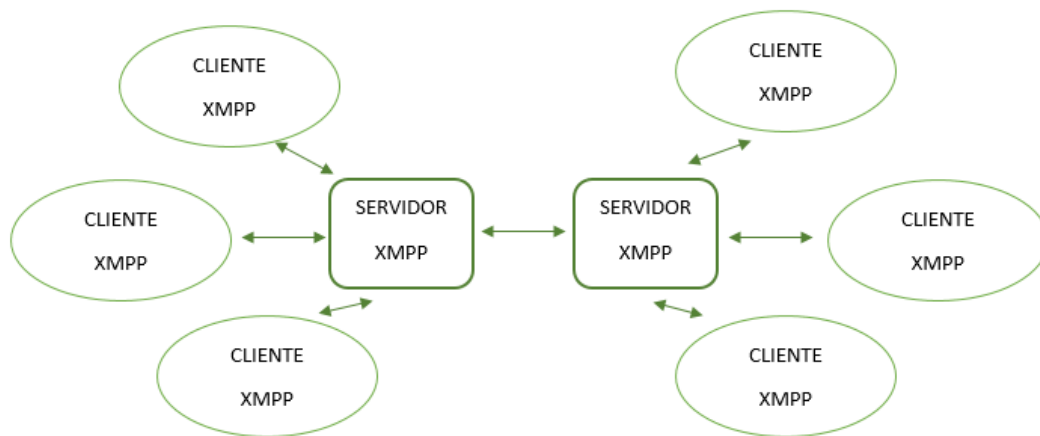


Diagrama II- 5 Protocolo XMPP

Elaborado por: Obando B. - Vega C.

⁶⁰ XML: eXtensible Markup Language.

⁶¹ VoIP: Protocolo voz sobre el protocolo de internet IP.

⁶² IETF: Internet Engineering Task Force.

⁶³ SASL: capa de seguridad y autenticación simple.

⁶⁴ TLS: seguridad de la capa de transporte.

Basado en: (Gonzales, 2014)

2.2.5. UDP

Protocolo de datos de usuario, encargado del transporte sin conexiones el cual permite a las aplicaciones contar con un mecanismo para enviar datagramas encapsulados de capa 4 y sin la necesidad de establecer una conexión, UDP cuenta con una cabecera de 8 bytes y 4 campos de los cuales dos son opcionales, los campos de puertos de origen son los que identifican al emisor y al receptor, UDP carece de un servidor de estado por lo cual este protocolo no solicita respuestas del estado del paquete . Este protocolo no permite la enumeración de los paquetes por lo cual no se puede garantizar que un paquete llegue duplicado o en desorden a su destino, como se puede observar en la figura 8. (Agesa, 2010),

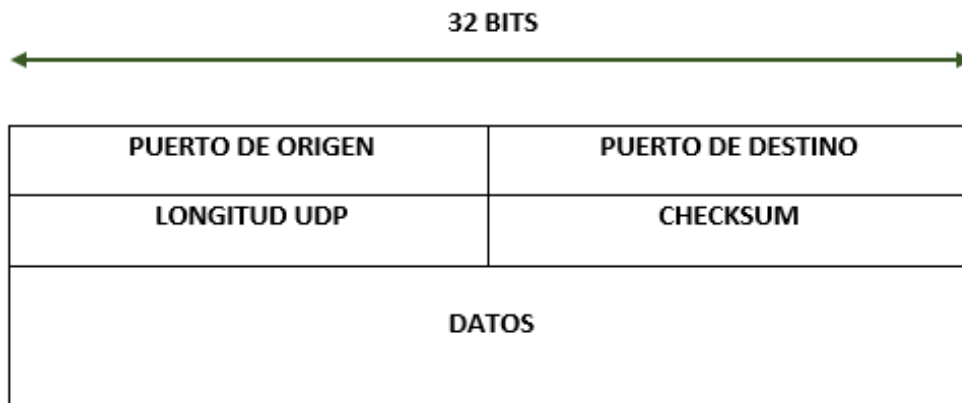


Diagrama II- Representación protocolo UDP

Elaborado por: Obando B. - Vega C.

Basado en: (Agesa, 2010)

2.2.6. IP

Internet Protocol. Este protocolo que hace posible la interconexión de redes en Internet, IP usualmente es usado junto al protocolo de transporte TCP. Las redes que utilizan este estándar utilizan direcciones IP de 32 bits con una capacidad de (2^{32}) direcciones diferentes, dividida en 4 segmentos de 8 bits, en decimal valores entre 0-255 cada uno separados por puntos, este tipo de dirección es conocida como IPv4, La sexta versión de este protocolo IPv6 con un número de direcciones en 16 bytes y 128 bits de longitud sustituirá progresivamente a IPv4. (Universidad Internacional de Valencia, 2016)

Los datos se encuentran organizados en paquetes y datagramas, cada paquete cuenta con información del origen, destino e información del mensaje, la cabecera tiene un espacio asignado de 20 bytes o 5 palabras de longitud. La palabra uno cuenta con la versión del protocolo (IPv4 o IPv6), longitud de la cabecera y el tipo de servicio. La segunda palabra sirve para identificar si el paquete se encuentra fragmentado y cuál es su número de identificación mediante la transmisión, si un paquete no puede ser fragmentado será indicado mediante una bandera, en la tercera palabra se define el tiempo de vida (TTL), su función es evitar bucles y la trayectoria del paquete a través de saltos, si este valor llega a cero el paquete se descarta. La dirección origen y destino del paquete se encuentran definidos en la cuarta y quinta palabra respectivamente. (Universidad Internacional de Valencia, 2016)

Las direcciones IP utilizan los dos primeros segmentos para asignar direcciones de red y los otros dos sirven para identificar un equipo dentro de la red, este sistema resultó inadecuado por

la asignación de direcciones, ya utilizadas para solucionar este problema se implementó una arquitectura de clases. (Universidad Internacional de Valencia, 2016)

- Clase A, cuenta con 126 redes desde 0.0.0.0 – 127.255.255.254
- Clase B, cuenta con 16384 redes desde 128.0.0.0 – 191.255.255.254
- Clase C, cuenta con 2097152 redes desde 192.0.0.0 – 223.255.255.254

En cada clase existen direcciones que no se encuentran asignadas denominadas direcciones privadas, estas direcciones son utilizadas por los hosts⁶⁵ para conectarse a Internet, las direcciones privadas de cada clase son:

- Clase A es una red de 8 bits y 24 bits para hosts que va desde la dirección 10.0.0.0 a 10.255.255.255.
- Clase B cuenta con 16 bits tanto para la red como para los hosts, van desde la dirección 172.16.0.0 a 172.31.255.255.
- Clase C es una red de 24 bits y 8 bits para hosts con direcciones desde 192.168.0.0 a 192.168.255.255.

El uso del protocolo IP es muy utilizado junto a protocolos de transmisión de información específica como TCP y UDP, todo el software de Internet, navegadores web, clientes y servidores FTP están basados en IP. (Universidad Internacional de Valencia, 2016)

⁶⁵ Hosts: Anfitrión de dispositivos conectados a una red.

2.3. Android Studio



Diagrama II- 6 Ventana Carga Android Studio

Elaborado por: Obando B. - Vega C.

Fuente: Android Studio versión 2.3.2

Es el entorno oficial de desarrollo integrado y creado por Google para la plataforma Android, fue anunciado en 2013 cuyo objetivo era reemplazar a Eclipse como IDE oficial para el desarrollo de aplicaciones. La principal característica que diferencia a Android Studio es la integración del SDK de este modo se logró que el desarrollo de aplicaciones sea específica o dedicada para cada versión de Android, tras un año de trabajo en diciembre de 2014 se presentó la primera versión estable y compatible con Windows, Mac OS y Linux tanto para arquitecturas de x86 y x64⁶⁶ bits. Actualmente Android Studio abarca el 80% de la cuota del mercado y es el IDE

⁶⁶ X64: Arquitectura de procesadores en 64 bits para ejecutar un conjunto de instrucciones.

preferido por desarrolladores, cuenta con una licencia Apache 2.0⁶⁷, es de uso libre y se encuentra en la versión 2.3.2. (Android, 2017)

Está basado en IntelliJ IDEA ⁶⁸el cual es un IDE para la plataforma de JVM⁶⁹, Android Studio tomó como referencia no solo su estructura sino la Interfaz Gráfica pero adaptada para el desarrollo de aplicaciones móviles, dando como resultado un híbrido entre IntelliJ IDEA y Eclipse. Android Studio ofrece aún más funciones que aumentan su productividad en el proceso de compilación de aplicaciones como las siguientes: (Android, 2017)

- Sistema de compilación Gradle⁷⁰.
- Emulador para cada versión, resolución de pantalla, periféricos y capacidad de procesamiento y memoria modificables.
- Entorno unificado para el desarrollo de todos los dispositivos Android.
- Instant Run generar cambios en el proceso de ejecución sin la necesidad de volver a compilar o crear un nuevo APK⁷¹.
- Plantillas de código e integración con GitHub⁷².
- Variedad de herramientas y frameworks⁷³ para mejorar la experiencia del desarrollador.
- Herramientas Lint para detectar código sospechoso, problemas de rendimiento, usabilidad, compatibilidad de versión, etc.

⁶⁷ Apache 2.0: licencia para el software de uso libre.

⁶⁸ IntelliJ IDEA: Es un ambiente de desarrollo IDE.

⁶⁹ JVM: Java Virtual Machine.

⁷⁰ Gradle: Herramienta para la automatización en la construcción de un Proyecto.

⁷¹ APK: Android Application Package.

⁷² GitHub: Plataforma de desarrollo colaborativo.

⁷³ Frameworks: Entorno o marco de trabajo para el desarrollo.

- Compatibilidad con C++ y NDK⁷⁴.
- Soporte para servicios de Google Cloud Platform⁷⁵, Google Cloud Messaging y App Engine⁷⁶.

Para el desarrollo de un proyecto en Android Studio se lo realiza utilizando módulos y paquetes los cuales cuentan con el código fuente, archivos y contenido multimedia, estos se encuentran organizados en 3 categorías definidas a continuación. (Android, 2017)

- Módulo de aplicaciones para Android: Contiene toda la información sobre las versiones de Android y esquema principal del proyecto.
- Bibliotecas: En esta sección se encuentran todas las librerías necesarias para que el proyecto funcione correctamente.
- Módulo de Google App Engine: Aloja todos los servicios de Google que brindarán soporte a nuestra aplicación.

La estructura está organizada de forma jerárquica la primera carpeta es “manifest” que contiene un fichero llamado “AndroidManifest.xml”, es un archivo de configuración y versionamiento para el código y nos permite otorgar permisos a la aplicación para usar diferentes características del dispositivo como su GPS⁷⁷, cámara, servicio de internet, etc. La siguiente

⁷⁴ NDK: Native Development Kit.

⁷⁵ Google Cloud Platform: Nube de Google.

⁷⁶ APP Engine: Servicio de alojamiento web.

⁷⁷ GPS: Sistema de posicionamiento global.

carpeta cuenta con los paquetes en los que se encuentran archivos de código fuente Java⁷⁸ denominados “Activities”, La última carpeta que encontramos se llama “res” y contiene todos los recursos, como diseños XML, imágenes, logos, colores, valores de variables, cadenas, dimensiones, etc. (Android, 2017)

Gradle es una herramienta encargada de automatizar y gestionar la construcción de los proyectos en Android, cuenta con una configuración por defecto la cual el desarrollador puede modificar, para que funcione correctamente este plugin⁷⁹ debe ser sincronizado cada vez que se realice un cambio ya que contiene las dependencias, instrucciones y librerías para la compilación, entre sus principales características contamos con la posibilidad de crear múltiples versiones de la aplicación como por ejemplo para móviles, tabletas determinar si una aplicación va a ser de pago o gratuita. (Android, 2017)

⁷⁸ Java: Lenguaje de programación orientado a objetos.

⁷⁹ Plugin: pequeña aplicación o programa que añade características al software principal.

2.3.1. Interfaz de Usuario.

En el siguiente diagrama se muestra la interfaz principal de Android Studio versión 2.3.2.

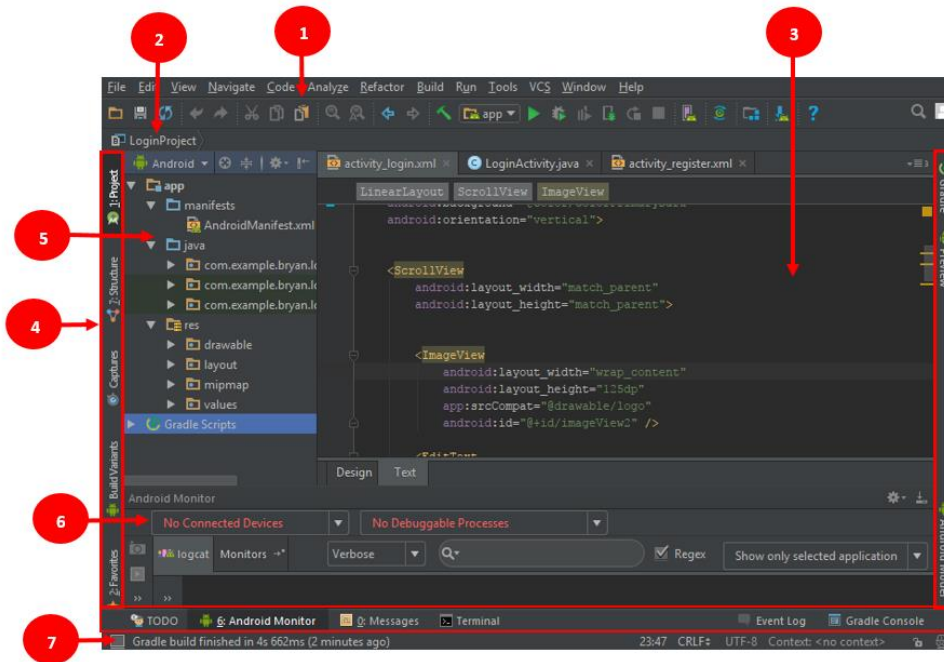


Diagrama II- 7 Interfaz Android Studio

Elaborado por: Obando B. - Vega C.

Basado en: (Android, 2017)

1. Barra de herramientas: Contiene las funciones principales para abrir, guardar, copiar pegar, deshacer, etc.
2. Barra de navegación: Nos permite visualizar la ubicación y carpetas que tiene nuestro proyecto, similar a un explorador de archivos.
3. Ventanas y pestañas del editor: Espacio en el cual se codifica el archivo seleccionado, nos brinda dos maneras de editar, una visual y otra en formato texto, las pestañas sirven para organizar los archivos que se van a modificar.

4. Barra de la ventana de herramientas: Se encuentra ubicada en los bordes de la interfaz del IDE, es una extensión de la barra de herramientas, pero con funciones específicas que permiten expandir o contraer distintas funciones o cuadros de dialogo.
5. Explorador de documentos: Esta casilla ayuda a visualizar el orden de la estructura del proyecto y cada una de sus carpetas y documentos, se puede modificar para buscar paquetes, proyectos y archivos según el desarrollador lo requiera.
6. Android monitor: Nos muestra el estado en el cual se encuentra el proyecto si está siendo compilado, en proceso de debug⁸⁰, los dispositivos que están conectados para pruebas y logcat⁸¹, el cual es un listado de los mensajes emitidos por el teléfono facilitando la detección de errores.
7. Barra de estado: Muestra el estado del proyecto y del IDE presentándonos una advertencia o mensaje en caso de que sucediera algún tipo de error.

Existen monitores de rendimiento los cuales sirven para dar seguimiento de manera sencilla el uso del CPU, GPU y memoria tanto del dispositivo móvil como del ordenador, busca objetos sin asignar, localiza pérdidas de memoria, controla las conexiones del dispositivo a periféricos o la red, cada una de estas funciones se puede visualizar directamente desde la ventana monitores de rendimiento o en la pestaña de anotaciones en Android Studio de una manera más general. (Android, 2017)

⁸⁰ Debug: Aplicación o herramienta para la ejecución controlada de un programa.

⁸¹ Logcat: Listado de mensajes emitidos por el teléfono, utilizado para encontrar errores en el software.

Android Studio es un entorno de desarrollo con muchas funcionalidades adicionales que lo convierten en un IDE completo para el desarrollo de aplicaciones, al contar con nuestro propio emulador integrado podemos realizar pruebas en prácticamente todas las versiones de Android disponibles actualmente, configurar el SDK según el desarrollador lo requiera, tener variantes de compilación para crear diferentes versiones de la misma aplicación desde un solo proyecto, sistema de depuración integrado mejora el proceso de revisión de código indicándonos variables integradas, objetos, métodos, expresiones lambda⁸² y operadores, etc. Android Studio continua en estado de desarrollo y periódicamente Google presenta actualizaciones para mejorar mucho más su productividad. (Android, 2017)

2.3.2. Requisitos del Sistema para Android Studio 2.3.2

En la siguiente tabla se presenta los requisitos para el funcionamiento de Android Studio en un ordenador.

	Windows	Mac OS X	Linux
Versión Sistema Operativo	Windows 7/8/10	Mac OS X 10 y posteriores	GNOME o KDE de escritorio. Ubuntu 12.04
Arquitectura PC	X86/ x64 bits		
RAM	3 GB mínimo, 8 GB recomendado		
ROM	2GB mínimo más 1,5 GB SDK y 1 GB emulador.		

⁸² lambda: expresiones para la simplificación de código en un IDE.

	Windows	Mac OS X	Linux
Resolución de pantalla	1280 x 800 pixeles mínimo, 1920 x 1080 pixeles.		
Versión Java	Java Development Kit 8		
Emulador	Sistema operativo de 64 bits y procesador Intel® compatible con Intel® VT-x, Intel® EM64T (Intel® 64) y la funcionalidad Execute Disable (XD) Bit. (No compatible con Mac)		

Tabla II- 1 Requisitos Android Studio

Elaborado por: Obando B. - Vega C.

Basado en: (Android, 2017)

2.4. GCM

Google Cloud Messaging. Es un servicio de notificaciones móviles desarrollado por Google que permite el uso a terceros para desarrollar aplicaciones con funciones de envío y recepción de información entre móviles u ordenadores, este servicio fue anunciado en 2012 y fue el sucesor de Android Cloud to Device Messaging, GCM eliminó las limitaciones para las APIS⁸³ incrementando el tamaño de los mensajes y mejorando las tasas de envío, pronto será sustituido por Google Firebase Cloud Messaging servicio de bases de datos no relacionadas en la nube. (Google, 2017)

⁸³ APIS: Application Programming Interface.

GCM fue compatible desde versión de Android 2.2 Froyo y toda su documentación fue presentada en el evento de tecnología Google I/O, esta plataforma tiene la habilidad de enviar notificaciones push⁸⁴ para el envío y recepción de mensajes en tiempo real, también nos facilita el despliegue de notificaciones cuando se recibe información directamente en el dispositivo, GCM funciona asignando un código único para cada dispositivo llamado register ID, este código se anexa directamente con otro servicio llamado Google Play Services que administra todos los servicios en el dispositivo, una vez asignado el register ID cuando el mensaje es enviado este es almacenado directamente en la nube de Google que se encarga de direccionar el mismo hacia el receptor, el objetivo de la nube es controlar cola de mensajes, mensajes repetidos, fallidos o enviar un mensaje a varios dispositivos, como se observa en la figura 11. (Google, 2017)

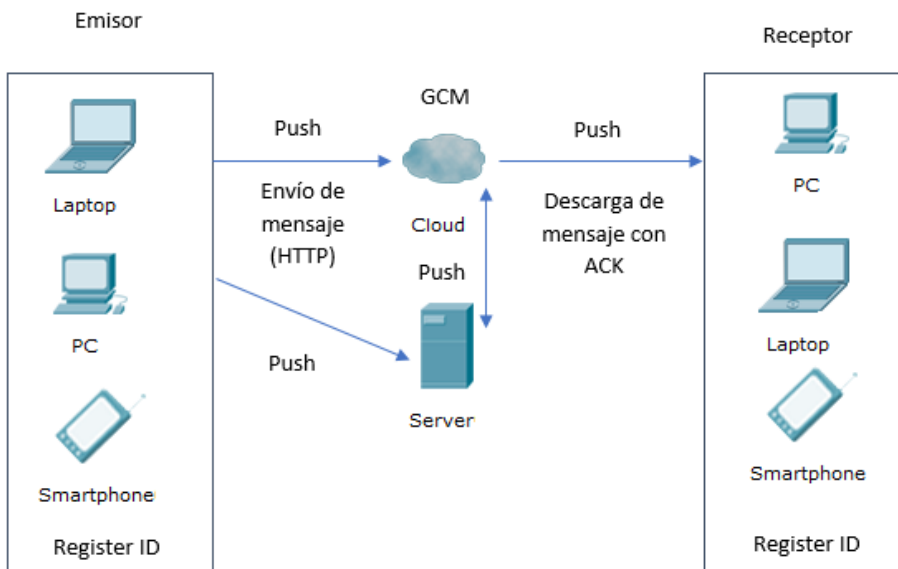


Diagrama II- 8 Funcionamiento GCM

Elaborado por: Obando B. - Vega C.

Basado en: (Google, Google Cloud Messaging, 2016)

⁸⁴ Push: Mensajes que se envían de forma directa a dispositivos móviles.

2.5. Definiciones

2.5.1. FTP

File Transfer Protocol. Protocolo de red para transferencia de archivos entre sistemas conectados por medio de una red o Internet en una arquitectura cliente-servidor, el cliente se puede conectar al servidor para acceder o enviar archivos sin importar el sistema operativo, al momento de la conexión hay dos canales de transmisión uno para control y el otro es el canal de datos. Los servidores FTP son utilizados para hosting⁸⁵, alojar páginas web, almacenar archivos personales en la nube. Una variación de este protocolo es Secure File Transfer Protocol (SFTP), utilizado para transmisión de datos cifrados, tal como se muestra en la figura 12. (CCM, 2017)

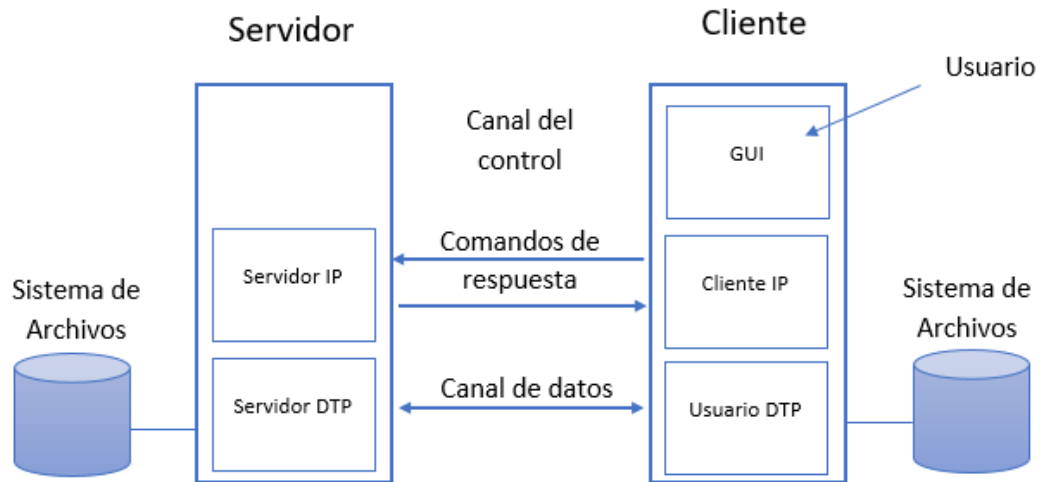


Diagrama II- 9 Funcionamiento Servidor FTP

Elaborado por: Obando B. - Vega C.

Basado en: (CCM, 2017)

⁸⁵ Hosting: servicio al cuál se asocia un dominio y brinda un alojamiento web.

Cuando el proceso de transferencia de datos (DTP) se lleva a cabo el intérprete de protocolo permite que DTP sea controlado por medio de los comandos del canal de control. El intérprete escucha los comandos del usuario a través del canal de control los cuales recibe el servidor FTP de esta forma se establece la conexión. Ftp cuenta con una serie de comandos para su manejo, también utiliza protocolos como Telnet que nos permite acceder a otro dispositivo de manera remota. (CCM, 2017)

2.5.2. JRE

Java Runtime Environment. Es un conjunto de utilidades que admiten la ejecución de programas java, controla el tiempo de ejecución en java virtual machine (JVM), esta máquina virtual interpreta el código y las librerías de clases son las que se implementan en el API de java. El usuario requiere el JRE para la ejecución si es necesario el desarrollo se requiere de un IDE y un JDK en el cual se va crear nuestro programa, el JRE es un intermediario entre el sistema y Java. (Latorre, 2010)

2.5.3. JDK

Java Development Kit es un conjunto de herramientas, programas y librerías que nos permiten compilar, ejecutar, documentar, etc. el desarrollo de programas en lenguaje Java. Existen varias versiones del JDK tanto para equipos con arquitecturas de 32 y 64. La empresa que se encarga de su distribución gratuita es Sun Microsystems y Oracle, los JDK cuentan con una herramienta para “debuggear” (término empleado para la ejecución y análisis de un programa en

desarrollo permitiéndonos conocer información sobre variables, métodos, clases y línea de código en cualquier momento de la ejecución del programa). Actualmente Oracle ha distribuido 8 versiones de esta herramienta. (Latorre, 2010)

2.5.4. SDK

Software Development Kit es un conjunto de herramientas diseñadas para facilitar el desarrollo de programas y aplicaciones, el SDK brinda soporte al desarrollador para encontrar errores y depurarlos de la manera más adecuada, la distribución de este software generalmente es gratuita y se lo hace a través del Internet, los desarrolladores muchas veces se convierten en colaboradores en la implementación de mejoras para el SDK en fases beta. (Tendencias MAG, 2016)

2.5.5. Navegador Web

Es un Software que nos permite visualizar contenido en internet por medio de un ordenador o dispositivo con acceso a este servicio, un navegador web interpreta código en HTML de una página web permitiéndonos navegar en ella mediante enlaces o hipervínculos, para realizar este proceso un navegador web se comunica con los servidores que alojan la página por medio del protocolo HTTP, la mayoría de navegadores admiten una gran variedad de protocolos como HTTPS⁸⁶, FTP, IP y muchos lenguajes de marcado para el contenido como XML, HTML5 y XHTML. Cada navegador cuenta con diferentes características y plugins que mejoran y añaden

⁸⁶ HTTPS: Protocolo de transferencia de hipertexto segura.

más funciones a los navegadores como formatos de audio, video, imágenes, capacidad de reproducir contenido flash⁸⁷, etc.

2.5.6. Wi-Fi

Es una tecnología para la conexión inalámbrica a una red local basada en el estándar IEEE⁸⁸ 802.11⁸⁹, la marca WI-FI Alliance⁹⁰ determina el uso y restricciones de esta tecnología por medio de Wi-Fi Certified⁹¹ para garantizar una conexión adecuada y funcional. Los dispositivos que cuentan con Wi-Fi se conectan a internet por medio de una WLAN⁹², las ventajas de establecer esta conexión es evitar el uso de cableado y facilitar la conectividad en sitios que no cuentan con puntos de acceso, el alcance de la señal es de aproximadamente 30 metros dependiendo de las antenas y el hardware incorporado en los equipos.

La frecuencia utilizada por Wi-Fi es de 2,4 GHz⁹³ y 5 GHz. Esta tecnología ha incorporado varias funciones para mejorar los tipos de conexión como Wi-Fi direct⁹⁴ o Wi-Fi ad-hoc⁹⁵ en las cuales no se necesita la conexión a un router⁹⁶ o punto de acceso, estos pueden conectarse directamente e intercambiar información. (Significados , 2017)

⁸⁷ Contenido flash: Formato para que el contenido sea indexado por los buscadores.

⁸⁸ IEEE: Organización sin fines de lucro, Instituto de ingenieros Eléctricos y Electrónicos.

⁸⁹ 802.11: Estándar IEEE. Utilizado para el funcionamiento de una red de área local inalámbrica.

⁹⁰ Wi-Fi Alliance: Organización sin fines de lucro que gestiona la tecnología Wi-fi.

⁹¹ Wi-Fi Certified: Tipo de certificado establecido por la Wi-Fi Alliance.

⁹² WLAN: Wireless local area network.

⁹³ GHZ: gigahertz.

⁹⁴ Wi-Fi Direct: Forma de conectividad Wi-Fi.

⁹⁵ Wi-Fi ad-hoc: Tipo de red inalámbrica descentralizada.

⁹⁶ Router: Dispositivo que proporciona conectividad a nivel de red.

2.5.7. Redes Móviles

Son redes pensadas para que los dispositivos móviles puedan comunicarse con otros en zonas donde exista cobertura, para establecer una comunicación existen varios protocolos los cuales mejoran la calidad de estas conexiones, una red móvil debe permitirnos mantener una conexión sin pérdidas en cualquier lugar que no exista interferencia, una de sus características es la estabilidad a pesar de movilizarnos a velocidades superiores a 300 Km/h. Los diferentes tipos de conexión son: (Temas Tecnológicos , 2017)

- GSM⁹⁷: Este fue el primer sistema de telefonía móvil totalmente digital que nos permitía realizar llamadas de voz y transferencia de información a 9.6 kbps⁹⁸, GSM opera en las bandas 900 MHz⁹⁹, 1.8 GHz para Europa y 1.9 GHz, 850 MHz para América. (Temas Tecnológicos , 2017)
- GPRS¹⁰⁰: Permitía alcanzar velocidades de transferencia hasta 40 kbps, este servicio está disponible en la mayoría de redes GSM. (Temas Tecnológicos , 2017)
- EDGE¹⁰¹: Es la evolución de GPRS la cual permitía aumentar la velocidad dependiendo de la distancia con la antena. (Temas Tecnológicos , 2017)
- Redes 2G y 3G fueron versiones mejoradas de GSM ofreciendo mayores velocidades para la transferencia de datos. (Temas Tecnológicos , 2017)

⁹⁷ GSM: Group Special Mobile.

⁹⁸ Kbps: kilobit por Segundo es una unidad de medida utilizada para determinar la velocidad en una red.

⁹⁹ MHz: Unidad de medida de la frecuencia.

¹⁰⁰ GPRS: General Packet Radio Service.

¹⁰¹ EDGE: Enhanced Data Rate for GSM Evolution.

- HSPA¹⁰²: Esta tecnología alcanzó velocidades de hasta 14 Mbps, superando a velocidades ADSL¹⁰³. (Temas Tecnológicos , 2017)
- LTE¹⁰⁴: es la cuarta generación en redes móviles, la más veloz actualmente a pesar de sus limitaciones en cuanto a frecuencias ofreciendo distintas velocidades dependientes, su máxima velocidad es de 400 Mbps¹⁰⁵, pero al contar con menos frecuencias esta puede ser de 75 Mbps. (Temas Tecnológicos , 2017)

2.5.8. Mensajería Instantánea.

La mensajería instantánea es una combinación entre los sistemas utilizados en servicios de chat y los mensajes que se envían por correo electrónico, este servicio tiene la capacidad que el emisor envíe un mensaje a un destinatario o a varios los cuales reciben el mensaje en tiempo real, el mensaje puede ser contestado al instante. Las aplicaciones de mensajería instantánea han añadido funciones adicionales para brindar un mejor servicio, entre esas funciones están el envío de archivos, contenido multimedia, videollamadas, telefonía, etc. (Lopez, 2015)

El servicio de mensajería instantánea utiliza la red de internet y sus protocolos para lograr la comunicación entre dispositivos. Actualmente este servicio cuenta con una gran cantidad de software y aplicaciones, por ejemplo: WhatsApp, Facebook Messenger, Skype, Line, Telegram,

¹⁰² HSPA: High Speed Packet Access.

¹⁰³ ADSL: Línea digital de banda ancha con gran capacidad para transmisión de datos.

¹⁰⁴ LTE: Long Term Evolution.

¹⁰⁵ Mbps: megabit por Segundo es una unidad de medida utilizada para determinar la velocidad en una red.

entre otros. Los servidores que hacen posible el servicio calculan que anualmente el tráfico de datos supera los 300 millones de mensajes tan solo en servidores de América. (Lopez, 2015)

2.5.9. Puertos de Red.

Es una interfaz cuyo objetivo es la comunicación utilizando una red, los puertos están numerados para posteriormente implementar protocolos y ser enviados, la asignación de puertos permite al ordenador entablar conexiones con diferentes dispositivos ya que los paquetes que se envían o reciben tienen la misma dirección, pero son dirigidos a puertos diferentes. El número de puertos tiene un tamaño de 16 bits¹⁰⁶ dando un total de 65535 puertos diferentes, la asignación de puertos está gestionada por IANA¹⁰⁷ que los organizó en 3 categorías:

- Puertos entre 0- 1024, estos puertos son usados por protocolos y están reservados para el sistema operativo, para utilizar estos puertos necesitamos que se encuentren disponible y contar con permisos de administrador en el equipo.
- Puertos entre 1024- 49151, estos puertos están registrados y pueden ser usados por cualquier software o por el usuario.
- Puertos entre 49152-65536, estos puertos son dinámicos o privados si son usados por el sistema operativo. (Pedra, 2016)

¹⁰⁶ 16 bits: Espacio que ocupan en memoria valores enteros o hexadecimales.

¹⁰⁷ IANA: Internet Assigned Numbers Authority.

2.6. Bases de datos

Una Base de datos permite al usuario acceder a la información, visualizarla, ingresar nuevos registros, actualizar o eliminar datos dependiendo de los derechos de acceso otorgados, es un método útil para almacenar y organizar información. Las bases de datos pueden ser locales, esto quiere decir que solo el usuario de un equipo tiene acceso, otro tipo de base de datos son las distribuidas, estas bases de datos cuentan con almacenamiento remoto y se puede acceder a través de una red. Una de sus ventajas es que varios usuarios puedan acceder al mismo tiempo. (CCM, 2017)

La administración de bases de datos se realiza con un sistema de Gestión de Bases de Datos o también llamado DBMS¹⁰⁸. Este sistema es un conjunto de servicios que permiten al usuario un acceso fácil e intuitivo a la información. El DBMS se divide en 3 subsistemas: uno para la administración de archivos; DBMS interno, para ordenar la información; DBMS externo, es la interfaz de usuario. (CCM, 2017)

Las características principales de los sistemas de bases de datos son:

- Independencia lógica y física a los datos.
- Evitar redundancia de datos.
- Acceso para múltiples usuarios a la vez.

¹⁰⁸ DBMS: sistemas de gestión de bases de datos.

- Integridad y seguridad de los datos.
- Consultas de datos optimizadas.
- Seguridad para el acceso y auditoría.
- Opciones de respaldo y recuperación de la información. (Valdes Perez, 2007)

2.7. Servicios Web

Para que un dispositivo pueda acceder a la información que se encuentra almacenada en la base de datos del servidor, existen diferentes maneras de crear un enlace de conexión; de forma directa utilizando conectores a las bases de datos desde el aplicativo en sí o a través de un servicio web el cual se encuentra alojado en el servidor de aplicaciones y el enlace se crea utilizando protocolos de conexión, estándares y un conjunto de métodos integrados dentro del código, que permitan adquirir los datos que necesite el dispositivo.

Para esto se crean los servicios web los cuales generan una pasarela entre los dispositivos y la base de datos transmitiendo la información cuando el dispositivo realice una llamada al servicio web, comúnmente se utiliza el protocolo HTTP por motivo que todos los navegadores de internet y servidores lo tienen integrado; se trabaja con diferentes operaciones que están definidas por un estándar las cuales son: PUT¹⁰⁹, GET¹¹⁰, POST¹¹¹ y DELETE¹¹², entre otras. Estas operaciones definen que debe realizar el código dentro del método que lo representa.

¹⁰⁹ Operación utilizada para el almacenamiento del recurso enviado dentro de un servicio web

¹¹⁰ Operación utilizada para la adquisición de recursos solicitados por la llamada a un servicio web

¹¹¹ Operación utilizada para la modificación de un recurso solicitado a través de un servicio web

¹¹² Operación utilizada para la eliminación de un recurso específico a través de una llamada del servicio web

Un servicio web puede ser implementado en diferente plataforma a los dispositivos que desean conectarse, es independiente del sistema operativo en el que se encuentre alojado y en el lenguaje que sea programado, por este motivo queda a disposición de los desarrolladores en que lenguaje implementarlo y dentro del sistema operativo que se desee, pero por estar expuestos al tráfico de la internet es necesario integrar altos niveles de seguridad para garantizar que la información que se transfiere de cualquiera de los dos extremos se encuentre protegida de cualquier atacante que desee adquirir esta información.

Como un servicio web es libre de ser desarrollado en diferentes plataformas y tecnologías a continuación se detallada una clasificación de estas:

- **SOAP:** Sus siglas corresponden a “Simple Object Access Protocol”, es un intercambio en base al protocolo HTTP y utiliza bloques de datos de extensión XML típicamente, es posible usar otro tipo de extensión, pero es más recomendable usar XML; estos definen las reglas para la estructura de los paquetes que se envíen a través de este servicio y los entrega con esta estructura, tal como se muestra en siguiente diagrama. (Quaine, 2007)

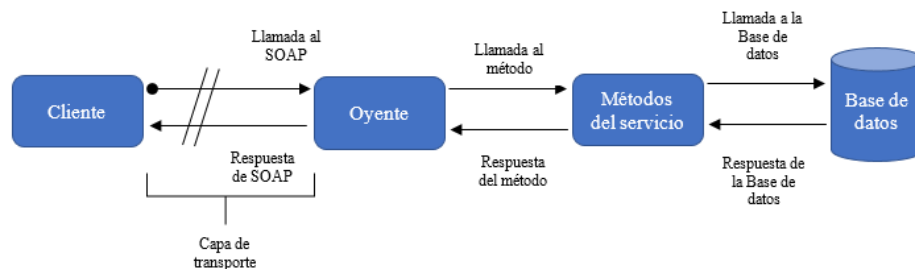


Diagrama II- 10 Funcionamiento de SOAP

Elaborado por: Obando B. - Vega C.

Basado en: (Quaine, 2007)

El proceso que sigue SOAP contiene cuatro elementos primordiales, los cuales son:

- **Cliente:** Elemento que hace la petición de información.
 - **Oyente:** Siempre está “escuchando” la capa de transporte por si existe alguna petición del servicio.
 - **Métodos del servicio:** Serán los encargados de adquirir la información desde la base de datos y enviarla al oyente para que este a su vez entregue como respuesta al cliente esta información.
 - **Base de datos:** Contiene toda la información necesaria lista para ser consumida.
- **RESTful:** Corresponde a “Representational State Transfer” el cual se denomina como una arquitectura que al ser implementada brinda un mejor desempeño, escalabilidad y facilidad de modificar para los servicios web, dentro de esta tecnología la información y sus funcionalidades son consideradas recursos y a estos se accede desde URIs ¹¹³, los cuales son direcciones a las funcionalidades que se encuentran dentro de la web. De la misma forma que otros servicios web tiene estándares en sus bloques de datos como son JSON¹¹⁴, XML, entre otros. El servicio web tendrá la función de ser el oyente de las peticiones del cliente, a continuación, diagrama de representación RESTful.

¹¹³ Uniform Resource Identifiers: Identificador de un recurso en base a denotaciones por el diseñador del mismo.

¹¹⁴ JavaScript Object Notation: Formato de intercambio de información.

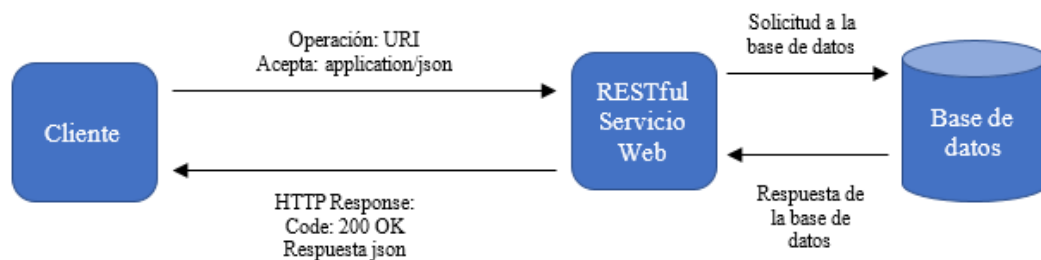


Diagrama II- 11 Arquitectura RESTful

Elaborado por: Obando B. - Vega C.

Basado en: (MikeKavis, 2013)

- **WSDL:** Corresponde a “Web Services Description Language”, se encarga de definir un listado estructurado de tipo XML que contiene los servicios de la red como puntos de acceso a los servicios web, utiliza varios elementos dentro de su estructura para identificar cada uno de ellos en niveles jerárquicos que tienen la capacidad de realizar un intercambio de mensajes. (Christensen, Curbera, Meredith, & Weerawarana, 2001)

 - **UDDI:** Sus siglas corresponden a “Universal Description, Discovery and Integration”, provee una descripción de los recursos de un servicio web y que servicios web que se ofrece a un cliente clasificándolos de forma similar a WSDL de una manera jerárquica desde sus proveedores como organizaciones a niveles más inferiores como que servicios ofrece y las interfaces con las que es posible acceder a estos servicios. Está estructurado con estándares entre ellos HTTP, XML, XML Schema y SOAP, listando si estos son de acceso público o internos de una organización. (Barry, 2003)
- Un UDDI consiste de 3 directorios que clasifican diferente tipo de información necesaria para definir los servicios web.

- **UDDI páginas blancas:** Contiene información de los proveedores de los servicios como datos descriptores de una organización.
- **UDDI páginas amarillas:** Contiene información más descriptiva de las organizaciones como que servicios ofrece al cliente, si desea realizar algún negocio con ellas.
- **UDDI páginas verdes:** Almacena la información técnica de los servicios y permite ligar a un servicio web; dentro de estas contiene las interfaces los URL e información requerida para encontrar a los recursos del servicio web y consumirlo.

La siguiente explicación estará enfocada en la adquisición de información de una base de datos a través de un servicio web de tipo REST, para otros tipos de servicios web su nomenclatura puede cambiar, pero realizan las mismas acciones. (ORACLE, 2013)

- **PUT:** Tiene la función de crear un nuevo registro, cuando es convocado a través del servicio web.
- **GET:** Adquiere toda la información de uno o varios registros que solicite el dispositivo cuando sea convocado a través del servicio web.
- **POST:** Transfiere un nuevo estado o de un registro ya existente, enfocado a una base de datos, una actualización del registro.
- **DELETE:** Elimina un registro que especifique el dispositivo.

Cuando se realiza una petición al servicio web, es necesario especificar la operación que se desea realizar, además de esto, dependiendo de la operación, incluir la información necesaria por el servicio web para realizarla.

Para realizar la comunicación, la información se transfiere con un formato que utiliza el servicio web, depende del servicio web que se utilice para elegir el formato, entre ellos se encuentra JSON, XML, HTTP, entre otros. Cada uno de estos tiene su diseño y si existiera un error en su estructura, el servicio web no sabría cómo manejarlo y devolvería inmediatamente un error al cliente, estos avisos también son utilizados para devolver una respuesta de varios tipos, al momento que un aplicativo crea una conexión al servicio web, este responde con un código que indica si fue posible o no realizar la conexión, esto afecta a todas las transacciones que se realicen entre el cliente y el servidor.

Por existir una gran variedad de códigos se especificarán las clases que engloban a los códigos, como indica la tabla II – 2.

CÓDIGO	ESTADO	DESCRIPCIÓN
1XX	Informational	Indica que la petición fue recibida por el servicio web. (IETF, 2017)
2XX	Success	Indica que la petición enviada al servicio web fue recibida, correcta, aceptada y procesada exitosamente. (IETF, 2017)
3XX	Redirection	Indica que la petición realizada al servicio web necesita de otra acción para ser completada, indica que puede que se esté direccionando a una URI antigua o que existen varias opciones a la solicitud. (IETF, 2017)

CÓDIGO	ESTADO	DESCRIPCIÓN
4XX	Client Error	Indica que la petición al salir del lado del cliente tiene alguna falla, como una estructura incorrecta, acceso no autorizado, etc. (IETF, 2017)
5XX	Server Error	Indica que la petición al llegar al lado del servidor no ha sido posible de procesar por algún motivo, puede suceder que los servidores no se encuentren en línea o las operaciones solicitadas por el cliente no existan, entre otras. (IETF, 2017)

Tabla II- 2 Códigos de respuesta HTTP

Elaborado por: Obando B. - Vega C.

2.8. Aplicación Web

El principal objetivo de la Internet es conectar la mayor cantidad de personas alrededor del mundo y poder distribuir cualquier tipo de información que sea posible, con este objetivo las aplicaciones web tomaron el papel de administrar y manejar este tipo de información, si se observa

desde un punto empresarial, la antigua manera de tanto distribuir y como adquirir datos era a través de las aplicaciones de escritorio siendo instaladas en los equipos de cada usuario, pero esto limitaba a conocer varios aspectos del cliente, desde si le era agradable y fácil de manejar, hasta como poder tenerlo como un cliente de una organización. Con el diseño de las aplicaciones web y la ayuda de la conexión entre los servidores y los clientes a través del internet, ahora las administraciones de estos recursos son limitados a un lugar y no afecta directamente al cliente al realizar cambios dentro de los servicios, esto se convierte en una ventaja para cada cliente de facilitar el acceso y uso de los servicios tan solo con tener una conexión a internet y la libertad de mejorar los servicios sin la necesidad de actualizar cada uno de las aplicaciones instaladas en los equipos de los clientes.

La creación de varias formas, unas más eficaces que otras, de realizar este proceso de distribución y adquisición de información, genera un amplio campo de desarrollo y con actualizaciones continuas de nuevas formas. Esto lleva a las aplicaciones web a ser un núcleo de diferentes aspectos de negocio.

2.8.1. Servidor de aplicaciones

Un servidor de aplicaciones está relacionado al concepto de un sistema distribuido, el tener una alta carga de conexiones a un servicio obliga a tener una capacidad que soporte esta carga, las ventajas de tener una función de esta magnitud en un servidor de aplicaciones es la facilidad de escalar el sistema si es necesario que existan nuevas funciones, el mantenimiento se vuelve más sencillo puesto que las modificaciones se realizan en un solo lugar, se actualiza, corrige errores y se despliega sin que los usuarios tengan el conocimiento de este proceso, ellos solo observarán los

cambios que se realizó cuando se encuentren en perfecto funcionamiento y la última ventaja que nos brinda es la alta disponibilidad, refiriéndose a esta como el funcionamiento ininterrumpido del servicio para que un usuario tenga acceso a cualquier hora del día, cualquier día del año (Universidad de Alicante, 2003).

Una función importante del servidor de aplicaciones es gestionar las conexiones que existan a un recurso y esto depende de la capacidad del equipo donde se encuentra alojado la aplicación, estos parámetros deben cambiar dependiendo del tráfico existente dentro de esta red ya que si supera las capacidades, el servicio puede quedar inutilizado hasta que la memoria sea liberada de una conexión ya existente o podrían verse afectadas todas las conexiones ya establecidas, provocando lentitud o que el servicio quede indisponible hasta que el administrador de servidor lo vuelva a ejecutar.

Existen diferentes empresas que han desarrollado servidores de aplicaciones, y ofrecen su servicio de manera gratuita, por este motivo depende de la carga que va a tener el aplicativo para seleccionar que servidor de aplicaciones podrá soportar este trabajo.

2.8.1.1. Servidor de aplicaciones JBOSS

JBoss ahora siendo parte de Red Hat¹¹⁵, es el área de soporte al programa de código abierto como servidor de aplicaciones, de esta división aparece el servidor de aplicaciones del mismo

¹¹⁵ Empresa Multinacional responsable de la distribución de varios productos de código abierto al área empresarial.

nombre, el cual fue diseñado en código Java y con capacidad de soportar aplicaciones de nivel empresarial y diseñadas en diferentes lenguajes de programación, aunque existen excepciones en los lenguajes su posibilidad de soportar un alto número de conexiones en paralelo y transacciones de nivel empresarial lo convierten en un servidor de aplicaciones bastante utilizado, adicional a esto puede ser ejecutado en cualquier sistema operativo donde exista Java (RedHat, 2016) .

Contiene características que Red Hat las describe:

- Soporta Java EE 7
- Agilidad entre operaciones de TI ¹¹⁶y desarrollo
- Utilización de recursos adaptable a las necesidades
- Despliegue flexible
- Libre uso y código abierto
- Desarrollo acelerado
- Listo para la nube
- Despliegue acelerado

2.8.2. Lenguaje de programación J EE

Java Enterprise Edition o por sus siglas J EE es una plataforma de desarrollo de software empresarial desplegado en un ambiente web, es una plataforma de uso libre utilizando el lenguaje de programación Java; en sus versiones iniciales fue conocido como J2EE pero posterior a la

¹¹⁶ Tecnología de la información: Area de estudio para la recolección, procesamiento de información.

versión 1.4 fue denominada J EE y sus versiones fueron cambiando aumentando el número de la versión en la cual se encontraba, actualmente la última versión es J EE 7 y se considera que al final del año 2017 saldrá su nueva versión J EE 8, el siguiente diagrama representa la arquitectura y niveles de J EE

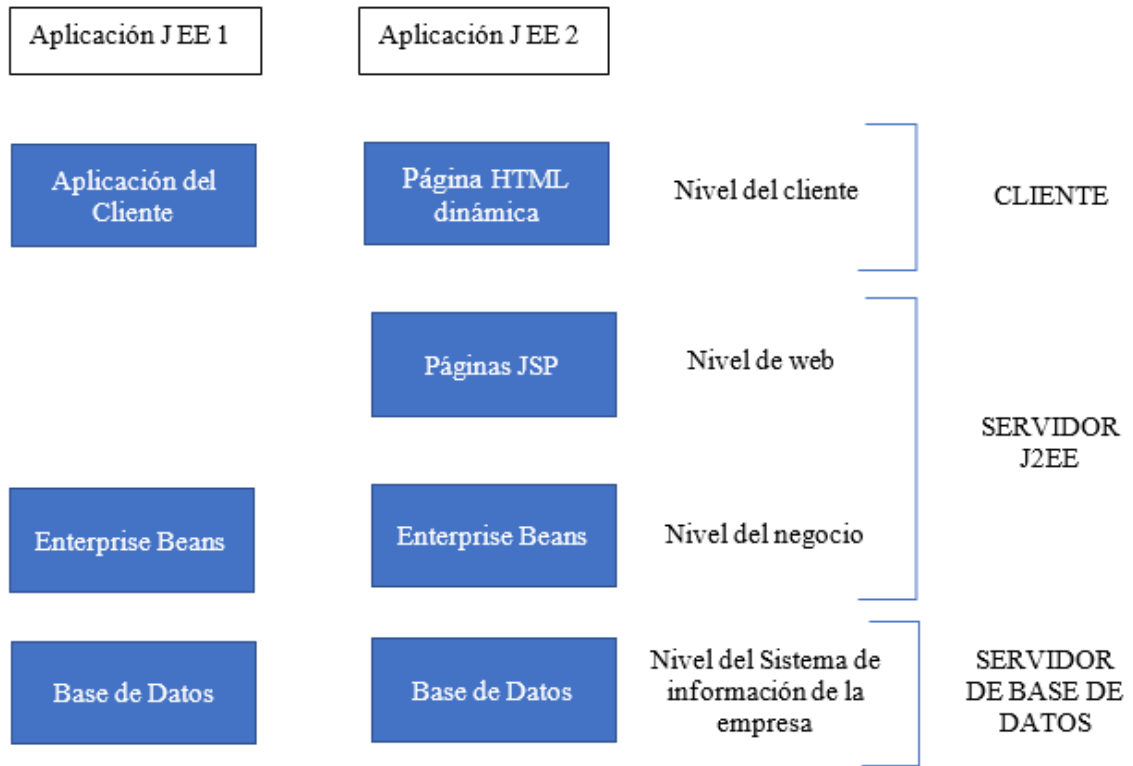


Diagrama II- 12 Arquitectura de dos aplicaciones diseñadas con J EE 2

Elaborado Por: Bryan Obando-Christian Vega

Basado en: (Pawlan, 2001)

En el diagrama II-3 observamos dos aplicaciones construidas con J EE 2, se observa que ciertas capas del sistema no son necesarias en el desarrollo, pero aun así se sigue respetando las jerarquías entre capas (GUI-DP-MD)

JEE provee de APIs para el manejo de entidades orientadas a objetos, servicios web y arquitectura por niveles; en la arquitectura por niveles se maneja similar a las capas que se tiene en una aplicación de escritorio GUI, DP ¹¹⁷ y MD¹¹⁸; en este caso tenemos el nivel del cliente el cual corresponde al medio por donde el usuario se comunica al aplicativo, un navegador, aplicaciones de escritorio, etc.; este se comunica con el nivel web el cual es la capa visible para el usuario, se encarga de adquirir cualquier información que el usuario ingrese, administrar la sesión del cliente, mantener los datos ingresados de manera temporal en componentes de tipo JavaBean¹¹⁹, a su vez se comunica con el nivel del medio, donde se encuentra la lógica del negocio, JEE se enfoca en esta capa para lograr una administración más sencilla, robusta y segura; procesa las solicitudes como información del aplicativo y las almacena a través del nivel de datos (siguiente capa del sistema), en una base de datos conectada a través del ultimo nivel. El manejar entidades como objetos facilita la comunicación entre el nivel de datos y la base de datos para realizar las operaciones que necesite la capa del medio, y existen componentes especializados para realizar estas actividades que serán especificados más adelante (ORACLE, 2012).

Dentro del nivel web existen tecnologías que ayudan al aplicativo a mantener la interacción entre el cliente y la lógica del negocio almacenando información temporalmente o mostrando una interfaz para el usuario.

- **Servlets:** Es una clase que contiene todas las interfaces y métodos necesarios para diseñar respuestas, administrar solicitudes utilizadas en páginas HTML

¹¹⁷ Capa de procesamiento de datos

¹¹⁸ Capa de manejo de datos hacia una base de datos.

¹¹⁹ Componente de la tecnología JEE, funciona como una clase contenedora de metodos dentro de una capa en el desarrollo de software.

- **JavaServer Faces (JSF):** Contiene clases para crear interfaces de usuario, componentes dentro del HTML y el usuario pueda ingresar la información necesaria, está encargada de la misma manera de almacenar la información en variables del lado del servidor.
- **JavaServer Faces Facelets:** Son de tipo JavaServerFaces, pero utilizando paginas XHTML en reemplazo de las JavaServer Pages (JSP).
- **Expressions Language:** Son anotaciones utilizadas dentro de JSP, JSF o JSF Facelets para nombrar a componentes que se encuentran dentro del J EE
- **JavaServer Pages (JSP):** Son documentos de tipo texto los cuales definen a qué nivel de dinamismo se puede añadir componentes dentro de las páginas HTML
- **JavaServer Pages Standard Tag Library:** Es una librería que contiene diferentes funciones utilizadas dentro de JSP
- **JavaBeans Components:** Estos componentes almacenan temporalmente la información ingresada en la interfaz de usuario y esta información sea utilizada cuando se realice una solicitud a los niveles inferiores

La interacción de los elementos mencionados anteriormente, permiten a los niveles inferiores realizar operaciones, entregar respuestas o almacenar la información dentro de una base de datos, interactuando entre sí y con el usuario.

Dentro del nivel del medio encontramos el nivel del negocio, este es más preciso para definirlo como el nivel que contiene la lógica del negocio. Las operaciones de un nivel web específico se encuentran en esta capa, aquí encontramos el manejo de toda la información que es transferida desde el nivel web y necesita ser gestionada a través de servicios, APIs y componentes que son parte del nivel del negocio (ORACLE, 2012).

- **Enterprise JavaBeans:** Componentes los cuales contienen las funcionalidades para realizar las transacciones hacia el nivel de datos, aquí se encuentran todas las operaciones que son posibles de realizar con los datos de un Managed Bean¹²⁰. Dentro de los Enterprise JavaBeans, por ser un componente que maneja la información lista para ser almacenada dentro de una base de datos es recomendable integrar seguridad al momento de realizar las transacciones entre los EJB¹²¹ y la base de datos.
- **JAX-RS RESTful web Services:** API con el cual es posible crear servicios web que responden a las operaciones HTTP mencionadas dentro de la

¹²⁰ Componente de JEE para el manejo de datos y expression language

¹²¹ Siglas de un Enterprise Java Bean

definición de los servicios web, los estándares que se manejan para crear un servicio web corresponden al tipo REST.

- **JAX-WS web Services endpoints:** API con la capacidad de crear y consumir servicios web de tipo SOAP.
- **Java Persistence API entities:** Este API permite la generación de las entidades que se encuentran dentro de una base de datos y transformarlas en objetos.
- **Java EE Managed Beans:** Son componentes donde se administra la información transferida del nivel web, dentro de estos se define las variables que se pueden utilizar como Expression Language dentro de una página JSP, JSF o JSF Facelets; es posible realizar operaciones dentro de estos Beans y considerar que existe una lógica del negocio de la misma forma que en los EJB, pero no disponen de la necesidad de existir seguridad en sus transacciones ya que la información que contenga un Managed Bean no es transferida directamente a la base de datos sin pasar por un Enterprise JavaBean.

El nivel del sistema de información empresarial contiene los servidores de base de datos, conexiones a estas que comúnmente se encuentran en un diferente lugar que el servidor de Java EE y pueden ser accedidas desde los componentes del nivel del negocio (ORACLE, 2012).

- **Java Database Connectivity API (JDBC):** API de bajo nivel que permite la conexión y la adquisición de la información almacenada dentro de una base de

datos, dentro de este API se diseñan los SQL queries¹²², donde el lenguaje utilizado para estos está definido por la base de datos que se utiliza.

- **Java Persistence API:** Encargado de tener la forma de cómo será LA estructura de las entidades al momento de transformarlas en objetos, se encuentre en un nivel superior al JDBC ¹²³y es encargado de no mostrar al usuario como la estructura de JDBC trabaja.
- **Java EE Connector Architecture:** Encargado de conectar a otros recursos que se encuentren dentro de la organización como los ERPs¹²⁴ o a software de administración de usuarios.
- **Java Transaction API (JTA):** Encargada de definir las transacciones cuando es un sistema distribuido o cuando existen varias bases de datos con las cuales se conecta la aplicación.

2.8.3. Framework para interfaces de usuario

Los frameworks para interfaces de usuario son diseñados para facilitar el desarrollo de las aplicaciones, consisten en un conjunto de componentes utilizados para reducir el tiempo de desarrollo de las interfaces y sus costos, ayuda a crear estructuras completas y entendibles con una terminología estandarizada en cada componente que maximiza la comprensión del desarrollador al momento de incluirlos en una interface de usuario, además tienen en sus estructura los estándares

¹²² Structured query Language: Lenguaje utilizado para realizar operaciones dentro de una base de datos.

¹²³ Java DataBase Connectivity: Define como un cliente accederá a una base de datos.

¹²⁴ Enterprise Resource Planning: Software de administración de todos los recursos de una organización.

necesarios de diseño visual, esto equivale a que los archivos CSS¹²⁵ ya tienen una estructura y permiten al desarrollador modificarlos si fuera necesario (Genaro, 2013).

La utilización de un framework para interfaces de usuario permite una mejor navegabilidad al usuario y un fácil entendimiento de las acciones que debe realizar cuando observe una aplicación de escritorio o web, además incluyen interfaces como eventos para realizar verificaciones, validaciones y páginas dinámicas de ser requeridas (Genaro, 2013).

Dentro del diseño de interfaces es considerado primordial la manipulación directa de componentes por parte del usuario como son botones, menús o íconos, por este motivo la responsabilidad que el desarrollador debe tener frente a este objetivo es que al momento que un usuario manipule la interfaz lo haga sin necesidad de una guía para que se sienta conforme al hacerlo, las interfaces deben cubrir todo aspecto necesario por el sistema para que sea posible realizar todas las operaciones que se encuentran en la parte de atrás del sistema (Duce, Gomes, Hopdog, & Lee, 2012)

2.8.3.1. Primefaces Framework

Primefaces es un framework diseñado principalmente para desarrolladores JSF, soporta HTML5 y contiene una gran variedad de componentes para interfaces de usuarios, tiene la capacidad de integración de AJAX¹²⁶ para el control de eventos o actualizaciones necesarias para

¹²⁵ Cascading Style Sheets: Describe la presentación visual de un archivo escrito en lenguaje de marcado (HTML)

¹²⁶ Asynchronous JavaScript and XML: Conjunto de acciones para diseñar aplicaciones web asincrónicas.

los componentes dentro de una página web y tiene compatibilidad con todos los navegadores de internet, su instalación se compone de un .JAR ¹²⁷y sus componentes son parte de una librería para JSF 2.0, de la misma forma íntegra tecnología responsive ¹²⁸Este framework fue desarrollado y le da soporte la corporación Primefaces Technology¹²⁹, la cual esa enfocada en facilitar el diseño de interfaces amigables tanto para aplicativos web como autónomas (Anghel, 2016).

Los componentes que se encuentran integrados en este framework, también han sido diseñados dentro de templates ¹³⁰desarrollados por Primefaces Technology, y da la opción a adquirirlos a través de su página principal. Con Primefaces es posible integrar una variedad de Frameworks de otras organizaciones que mejoran la capacidad de este framework y dentro de su documentación existen ejemplos de la utilización de estos frameworks de terceros, a la vez de como implementar el código necesario para su funcionamiento. Provee de una gran variedad de estilos e iconos que se pueden encontrar dentro de toda la documentación. Primefaces es actualizado constantemente con las nuevas tecnologías que se van destacando, ahora tienen integración a Angular2 ¹³¹para su uso en conjunto (Çivici, 2017).

¹²⁷ Extensión de archivos tipo Java

¹²⁸ Tecnología de las aplicaciones web y páginas web para modificar su estilo al cambiar de una resolución a otra.

¹²⁹ Empresa encargada del desarrollo y soporte de los productos de Primefaces

¹³⁰ Conjunto de componentes integrados dentro de un modelo visual para la creación de la interface gráfica para usuario.

¹³¹ Framework de manejo de la capa de GUI

2.8.4. Entorno de desarrollo Integrado

Un entorno de desarrollo es definido como un conjunto de herramientas integradas dentro de un paquete y convertido en aplicación para facilitar la implementación de proyectos de desarrollo de cualquier tipo, dentro de estos es posible encontrar asistentes para la creación de servicios web, estructuras de paquetes, APIs para diseño de entidades como objetos y una variedad de ayudas para realizar un despliegue de aplicaciones o para integrar nuevas herramientas dentro del entorno. Un IDE tiene su propia interfaz gráfica para facilitar el manejo de quien lo utilice y existe una gran variedad de ellos, dependiendo del lenguaje que estemos utilizando, para hacerse llamar IDE debe cumplir con ciertas características (fergarcia, 2013):

- Multiplataforma
- Soporte de variedad de lenguajes de programación
- Integración de sistemas de control de versiones
- Reconocimiento de sintaxis
- Extensiones y componentes para el IDE
- Integración de Frameworks
- Depurador
- Importación y exportación de proyectos
- Variedad de Idiomas
- Manual de usuario y opción de ayuda

2.8.4.1. Eclipse

Eclipse es un IDE utilizado por varios desarrolladores por su nivel de estabilidad y soporte constante por parte de The Eclipse Foundation¹³², existen varias actualizaciones donde se integran nuevos APIs, frameworks y soporte de lenguajes que son más utilizados en la actualidad, además de simplificar la manera de utilizar su IDE, logrando un manejo más instintivo por parte de los desarrolladores. Es muy popular dentro de la comunidad de Java ya que es software de libre uso, contiene su propio compilador Java y frameworks para el desarrollo de aplicaciones web, autónomas o móviles; dependiendo de la necesidad. Para el uso de IDE con otros lenguajes que no sea Java es necesario la instalación de extensiones, las cuales son fáciles de acceder dentro del buscador de extensiones que tiene integrado Eclipse (Adrian, 2014).

Después de ser parte de un proyecto de IBM¹³³, y se invierta una gran cantidad de dinero por parte de diferentes empresas que pertenecías a esta alianza, fue tomado como solución crear The Eclipse Foundation y se decidió en el año 2001 que sería software libre y que Eclipse trabaje bajo la licencia EPL¹³⁴, quiere decir que todo proyecto desarrollado con este IDE será liberado con esta licencia, existen desarrolladores de software que desean integrar otros tipos de licencia a sus proyectos, por lo cual incluye la licencia EDL¹³⁵, ambas licencias son de software libre pero existe incompatibilidad con las licencias de tipo GPL¹³⁶ (Burnette, 2005).

¹³² Fundación creada después de pasar Eclipse de manos de IBM a esta fundación, encargada de dar soporte al IDE Eclipse

¹³³ International Business Machine Corporation

¹³⁴ Eclipse Public License: Licencia de código abierto utilizada por Eclipse Foundation para su software.

¹³⁵ Eclipse Distribution License: Licencia utilizada por Eclipse Foundation para software generado con sus productos.

¹³⁶ General Public License: Licencia de software libre, que da varios permisos sobre un software.

Eclipse Foundation ha coordinado que se liberará una versión nueva de este IDE anualmente y han sido ya 13 versiones liberadas hasta el día de hoy. En el siguiente grafico se podrá observar la línea de tiempo de las versiones liberadas con su nombre y la fecha en la cual fueron puestas a disposición del público, el siguiente diagrama muestra una línea del tiempo de Eclipse.

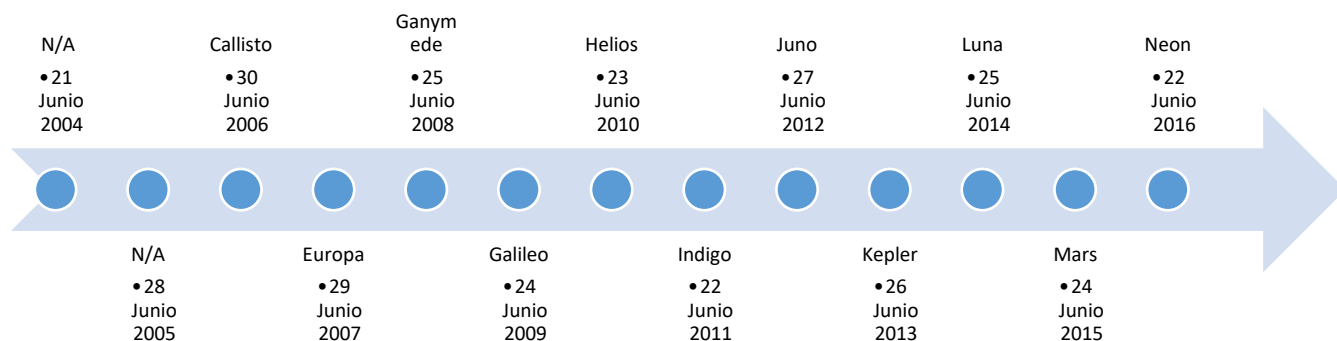


Diagrama II- 13 Línea de tiempo del IDE Eclipse y sus fechas de liberación.

Elaborado Por: Bryan Obando-Christian Vega

Basado en: (Code2Care, 2015)

CAPITULO III

En este capítulo analizaremos las aplicaciones de mensajería instantánea más utilizadas, para definir dichas aplicaciones se realizó una encuesta dirigida a estudiantes de la carrera en ingeniería de sistemas y computación de la Universidad Católica del Ecuador. El método utilizado en el cálculo del tamaño de la muestra será la fórmula propuesta por Murray y Larry detallada a continuación:

$$n = \frac{K^2 \times \sigma \times N}{(e^2 \times (N - 1)) + K^2 \times \sigma}$$

- N: población.
- K: Nivel de confianza
- e: Margen de error
- σ : Desviación estándar

La población utilizada fue de 200 estudiantes con un nivel de confianza del 90%, un margen de error del 5% y desviación estándar de 0.5. Con estos parámetros se determinó que el tamaño mínimo de la muestra es 116.

La encuesta cuenta con un total de 18 preguntas, de las cuales, la persona responderá como mínimo 2 preguntas y máximo 13. El número total de personas encuestadas fue de 120. A continuación, se realizará el análisis de los resultados que determinarán el software y las plataformas en las que se ejecutarán pruebas de mensajería instantánea, el contenido completo de la encuesta se encuentra

en la sección de anexos. Los datos totales de la encuesta se encuentran en el capítulo VIII, sección Anexos.

3. Definición de plataformas.

3.1. Sistema operativo del dispositivo.

El resultado obtenido fue que un total de 27 personas eligieron iOS, 76 seleccionaron Android, 1 Windows Mobile y 16 personas no cuentan con un dispositivo móvil. Por lo tanto, el sistema operativo que se utilizará es Android. Tal como se muestra en el Diagrama III-I Marca del dispositivo móvil.

Pregunta

- ¿Cuál es la marca de su dispositivo personal más utilizado?

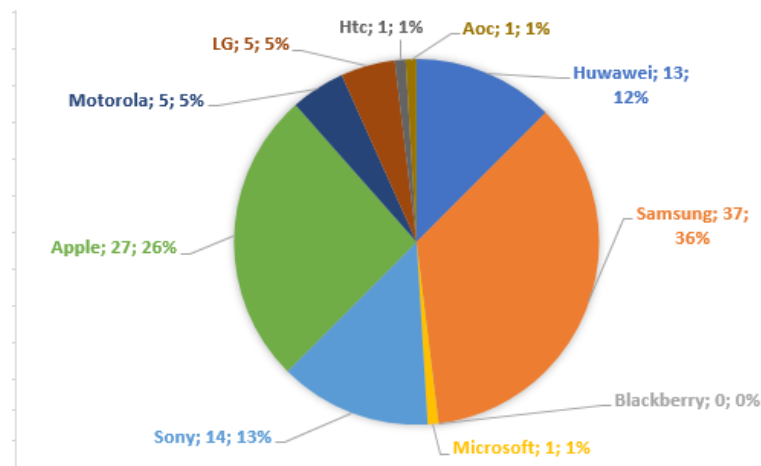


Diagrama III- 1 Marca del Dispositivo

Elaborado por: Obando B. - Vega C.

Consideraciones:

- El total de respuestas es de 104, 16 personas no cuentan con un dispositivo móvil.
- La suma de las marcas Motorola, LG, HTC, AOC, Huawei, Samsung y Sony cuentan con sistema operativo Android.

3.2. Versión de Android

La versión de Android con un total de 76 usuarios da como resultado que, 41 personas, representando el 53% desconoce la versión del sistema, por lo tanto, para el desarrollo del prototipo se usará el nivel api 15 (Android 4.0), para la fase de pruebas se utilizará cualquier versión superior al api antes mencionado. El resultado se muestra en el Diagrama III-2 Versión Adroid.

Pregunta

- ¿Cuál es la versión de Android en su dispositivo?

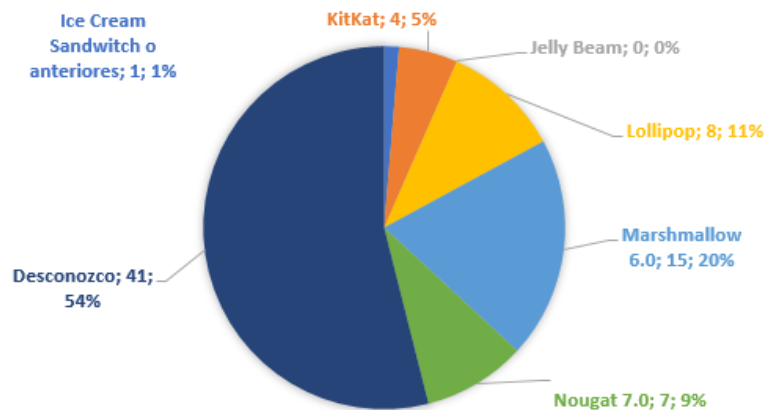


Diagrama III- 2 Versión Android

Elaborado por: Obando B. - Vega C.

Consideraciones:

- El total de respuestas es de 76, las cuales representan el 63.33% de la muestra.
- Si la mayoría de usuarios desconoce la versión del sistema operativo, se ocupará versiones de Android actuales.

3.3. Conectividad del Dispositivo

El tipo de conexión más utilizado fue 3G con un total del 41.7%, cabe mencionar que el tipo de conexión puede variar según la calidad de recepción de la señal móvil y la ubicación del dispositivo. A continuación, se muestran los resultados en el Diagrama III-3 Tipo de Conexión.

Pregunta

- ¿Cuál es el tipo de conexión de su dispositivo?

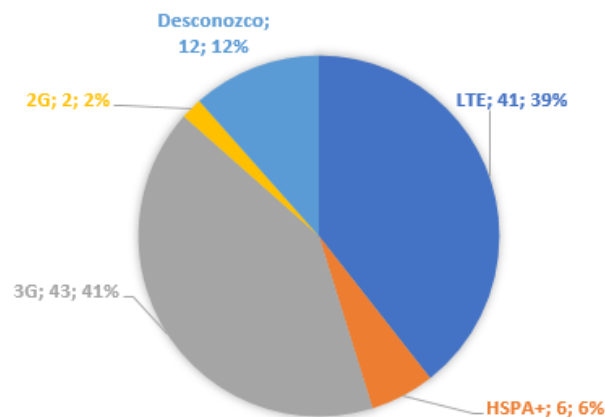


Diagrama III- 3 Tipo de Conexión

Elaborado por: Obando B. - Vega C.

Consideraciones:

- La calidad en la recepción de señal puede verse afectada por la ubicación del usuario.

Aplicaciones de mensajería.

Con un total de 104 respuestas, 98 personas utilizan WhatsApp, 69 usan Facebook Messenger y 16 Skype, por lo tanto, estas tres aplicaciones serán utilizadas en la fase de pruebas.

El Diagrama III- 4 Aplicaciones de Mensajería.

Pregunta

- ¿Cuáles aplicaciones de mensajería utiliza?

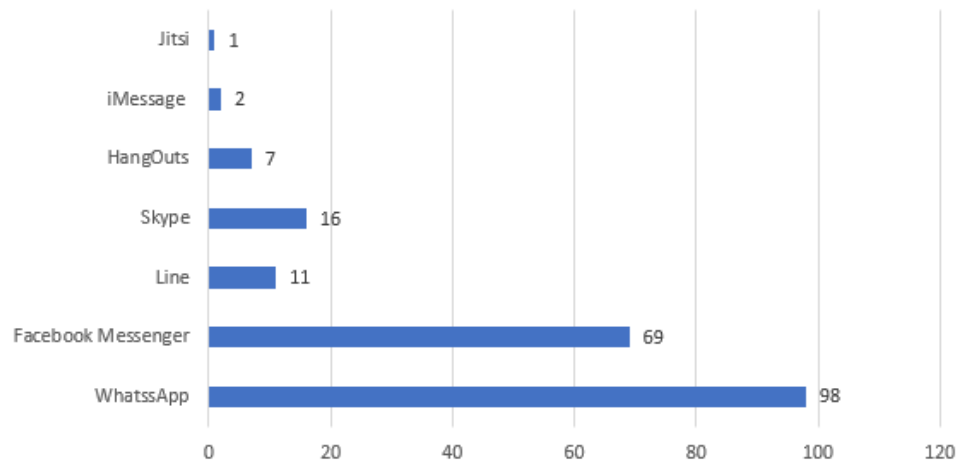


Diagrama III- 4 Aplicaciones de Mensajería

Elaborado por: Obando B. - Vega C. I

Consideraciones:

- La persona encuestada puede seleccionar varias opciones.
- Se toma en cuenta las 3 aplicaciones más utilizadas por los usuarios.

3.4. Análisis de plataformas

3.4.1. Especificaciones y características de las aplicaciones móviles.

3.4.1.1. WhatsApp

Fue fundada en 2009 por Jan Koum y Brian Acton, en 2014 se unió con Facebook, pero sigue siendo una aplicación independiente y enfocada en el servicio de mensajería. WhatsApp cuenta con software en distintas plataformas móviles, actualmente se distribuye de forma gratuita, a partir del año, el costo es de un dólar americano por el mismo período de tiempo, se encuentra en la versión 2.17 publicada el 7 de junio del 2017 y cuenta con más de mil millones de descargas realizadas en tienda de aplicaciones Play Store de Google, cuenta con una valoración de 4.4 sobre 5 según los usuarios, a continuación, más detalles técnicos de WhatsApp.

- Versión: 2.17.223
- Publicación: 2009
- Actualización: 7/7/2017
- Protocolo Mensajería: XMPP, TCP
- Plataformas: Android, iOS, Windows Phone, BlackBerry OS, Symbian.

3.4.1.2. Facebook Messenger

Aplicación de mensajería instantánea desarrollada por Facebook, inicialmente conocida como Chat de Facebook, pero en 2011 fue presentada como una aplicación independiente llamada Facebook Messenger y disponible para Android, iOS, Windows 10, Web. Su distribución es gratuita, se puede acceder mediante la url: www.messenger.com y una cuenta de Facebook activa, también podemos acceder a la aplicación móvil por medio de las tiendas virtuales Play Store y App store, actualmente se encuentra en la versión 126.0.0.9.84 publicada 10 de Julio de 2017 con más de mil millones de descargas.

- Versión: 126.0.0.9.84
- Publicación: 2011
- Actualización: 7/7/2017
- Protocolo Mensajería: TCP
- Plataformas: Android, iOS, Windows Phone, BlackBerry OS, Symbian.

3.4.1.3. Skype

Esta aplicación de mensajería instantánea también brinda servicios de voz y videollamadas siendo estos los más utilizados por los usuarios. Skype fue fundada en 2003 por Janus Friis y Niklas Zennström, desde 2013 Skype forma parte de Microsoft y se encuentra disponible para Windows 10, Android, iOS. La versión que analizaremos es Skype es 8.1.0.46539 publicada el 22

de junio de 2017 en la Play Store y cuenta con una calificación de 4.1 sobre 5 y con más de quinientos millones de descargas en todo el mundo.

- Versión: 8.1.0.46539
- Publicación: 2003
- Actualización: 6/22/2017
- Protocolo Mensajería: TCP
- Plataformas: Android, iOS, Windows Phone, BlackBerry OS, Symbian.

3.4.2. Especificaciones técnicas dispositivos móviles.

Para el desarrollo de las pruebas las características y especificaciones técnicas de los dispositivos móviles no serán tomadas en cuenta ya que es muy improbable que tanto el emisor como el receptor cuenten con una configuración idéntica cuando ocupan servicios de mensajería; a continuación, especificaremos los dispositivos utilizados en las pruebas según resultados en el apartado 3.2.1 Resultados de la encuesta.

3.4.2.1. Dispositivos.

Selección del primer dispositivo según versión de Android, características de red y disponibilidad de las aplicaciones; la Tabla III-1 muestra las especificaciones de los dispositivos móviles.

Dispositivo	Moto Z Force
Modelo	XT1650-02
Empresa	Lenovo
Lanzamiento	Junio de 2016
Redes	GSM 850 / 900 / 1800 / 1900 - HSDPA 850 / 900 / 1900 / 2100 - LTE band 2(1900), 3(1800), 4(1700/2100), 5(850), 7(2600), 13(700)
Wi-Fi	Wi-Fi 802.11 a/b/g/n/ac; Wi-Fi Direct; banda dual
GPS	GPS con soporte A-GPS y GLONASS
Dimensiones	155.9 x 75.8 x 7 mm
Peso	163 g
Pantalla	AMOLED, Touch, Resolución 1440 x 2560 pixeles, 5.5 pulgadas
Capacidad	32/64 GB
RAM	4 GB
Procesador	Procesador Qualcomm MSM8996 Snapdragon 820 quad-core
GPU	GPU Adreno 530
Gprs	Si
SO	Android 7.1 Nougat
Cámara	Principal: 21 MP, f/1.8 Secundaria: 5MP f/2.2
Batería	3500 mah Li-ion

Tabla III- 1 Características Técnicas dispositivo móvil 1

Elaborado por: Obando B. - Vega C.

La información de la Tabla III-2, corresponde al segundo dispositivo por marca, sistema operativo y disponibilidad de las aplicaciones.

Dispositivo	Galaxy S5
Modelo	G900M
Empresa	Samsung
Lanzamiento	Febrero de 2014
Redes	GSM 850 / 900 / 1800 / 1900 - HSPA 850 / 900 / 1900 / 2100 - LTE
Wi-Fi	Wi-Fi 802.11 a/b/g/n/ac; DLNA; Wi-Fi Direct; banda dual
GPS	A-GPS; GLONASS
Dimensiones	142 x 72.5 x 8.1 mm
Peso	145 g
Pantalla	Super AMOLED, Touch, Resolución 1080-1920 Píxeles, 5.1 pulgadas
Capacidad	16/32 GB
Ram	2 GB
Procesador	Procesador Qualcomm, Snapdragon 801 quad-core 2.5 GHz.
GPU	GPU Adreno 330
Gprs	Si
SO	Android 6.1 Marshmallow
Cámara	Principal: 16 MP Secundaria: 2.1MP
Batería	2800 mah Li-ion

Tabla III- 2 Características Técnicas dispositivo móvil 2

Elaborado Por: Obando B.- Vega C.

3.5. Cuadro Comparativo aplicaciones móviles.

En la Tabla III-3 analizaremos los resultados realizados en las pruebas, sus características y funciones principales, a pesar de que las tres aplicaciones a comparar cuentan con servicios de video, llamadas, envío de documentos estas características serán evaluadas, pero no influenciarán en el resultado para determinar la aplicación más eficaz para el envío de mensajes. Los resultados se muestran a continuación y son referentes a las versiones de Android.

Características Servicio	Aplicaciones Móviles		
	WhatsApp	Facebook Messenger	Skype
Mensajería	Si	Si	Si
Video	Si	Si	Si
Voz	Si	Si	Si
Notas de Voz	Si	Si	No
Chat en grupo	Si	Si	Si
Envío de contenido multimedia	Si	Si	Si
Copia de seguridad	Si	Si	Si
Estado de Conexión	Si	Si	Si
Cambio de Estados	Si	Si	Si
Versión Web o Pc	Si	Si	Si
Costo	Si	No	Si
Publicidad	No	No	Si

Características Técnicas	Aplicaciones Móviles		
	WhatsApp	Facebook Messenger	Skype
Versión	2.17.254	128.0.0.21.88	8.2.0.49163
Tamaño aplicativo	20MB	45MB	21MB
Proceso	com.whatsapp	com.facebook.orca	com.skype.raider
Protocolo de conexión TCP	Si	Si	Si
Protocolo mensajería	XMPP	TCP	TCP/VoIP
GCM	Si	Si	No
Encriptación TLS¹³⁷	Si	Si	Si
AES	Si	Si	Si
Telefonía Análoga	No	No	Si

Tabla III- 3 Cuadro comparativo Características Aplicaciones y resultados pruebas

Elaborado por: Obando B. - Vega C.

3.6. Elección y definición de la plataforma

Se decidió realizar el prototipo con un sistema operativo Android por los resultados adquiridos del DIAGRAMA III-1, donde se muestra que una gran cantidad de usuarios utilizan un equipo móvil con esta tecnología, y la elección de la tecnología para servicio de mensajería instantánea se utilizó como fuente los resultados adquiridos del DIAGRAMA III-4, considerando que las 2 primeras tienen tecnología GCM, esta será implementada dentro del prototipo.

¹³⁷ TLS Transport Layer Security

CAPITULO IV

Este capítulo contiene una breve descripción de la funcionalidad del prototipo y la interacción entre el servidor y el aplicativo móvil, posterior a esto se especificarán los actores o tipos de usuarios presentes en el desarrollo del proyecto, las historias de usuario que definen las funcionalidades existentes en los aplicativos, las estimaciones de tiempo y el análisis de las historias de usuario, los diagramas para definir la estructura de la base de datos y simplificar el entendimiento de las historias de usuario para el desarrollo, los métodos a utilizar dentro de las clases implementadas, los requerimientos de software y hardware para alojar el ambiente de desarrollo, la base de datos, el servidor de aplicaciones y el equipo de despliegue para la aplicación móvil y finalmente la estructura de las tablas de la base de datos (Diccionario de Datos).

4. Definición del problema

4.1. Introducción

Para un entendimiento más profundo de cómo funcionan las aplicaciones de mensajería se tomó la decisión de realizar un prototipo de un sistema de mensajería instantánea utilizando una de las tecnologías mencionadas en los capítulos anteriores, para esto se utilizará todo un proceso de desarrollo basado en la metodología “XP”, el cual al ser concluido, tendrá la capacidad de enviar y recibir mensajes a través de usuarios que tengan un dispositivo móvil y un servidor alojado en un computador de escritorio, además de esto deben encontrarse registrados en la base de datos que

será administrada igualmente desde un equipo de escritorio, para lo cual utilizaremos una administración de usuarios, donde será posible crear, eliminar y modificar usuarios.

4.2. Identificación de actores (Roles)

4.2.1. Administrador

El administrador es la persona encargada de gestionar los usuarios que se registrarán en el aplicativo móvil y podrán hacer uso de la mensajería instantánea. Los administradores tendrán la posibilidad de conversar con los usuarios registrados desde el aplicativo web.

4.2.2. Usuario Android

El usuario Android es uno de los roles que un administrador puede crear dentro del sistema, este rol tendrá la capacidad de mensajear entre usuarios del mismo rol, rol Ios rol administrador, y tendrá acceso al aplicativo móvil desde un equipo con sistema operativo Android.

4.2.3. Usuario Ios

El usuario Ios es otro rol que el administrador tiene la capacidad de crear dentro del sistema, tiene las mismas características que el usuario Android con la modificación que podrá registrarse dentro de un teléfono móvil con un sistema operativo iOS.

4.3. Historias de usuario

Es una descripción simple de las funcionalidades que se implementarán dentro del software, son descritas por los clientes en palabras entendibles para todo el equipo, estos sustituyen a los diagramas de casos de uso y su forma de redacción es al menor detalle. (Joskowicz, 2008)

El sistema que se ha implementado se encuentra dividido en dos partes; la Administración del Sistema, la cual fue desarrollada en un ambiente web accesible desde cualquier computador portátil o de escritorio, como funcionalidades tenemos la creación, eliminación y modificación de los usuarios y la mensajería entre administradores y usuarios Android o Ios; y el aplicativo móvil correspondiente los sistemas operativos de Android e Ios, como funcionalidad principal tendríamos la mensajería instantánea donde el usuario podrá comunicarse entre equipos móviles o entre computador y equipo móvil, dentro de ambas partes del sistema existe una autenticación del usuario que vaya a ingresar al sistema, tanto web como móvil. A continuación, se detallan las historias de usuario para la creación de las plataformas.

4.3.1. Autenticación de usuario (WEB)

Código: HUW01	Nombre Historia: Autenticación de usuario (WEB)
Usuario: Administrador	
Prioridad en Negocio: Alta	Riesgo en desarrollo: Baja
Descripción: Como usuario administrador quiero ingresar al sistema web a través de un control de personal.	
Validación: Los usuarios que deseen ingresar al sistema deberán registrarse con su nombre de usuario y su contraseña.	

Tabla IV- 1 Historias de usuarios 1: Autenticación de usuario(WEB)

Elaborado por: Obando B. - Vega C.

4.3.2. Administración de usuarios (WEB)

Código: HUW02	Nombre Historia: Administración de Usuarios (WEB)
Usuario: Administrador	
Prioridad en Negocio: Alta	Riesgo en desarrollo: Media
Descripción: Como usuario administrador debo tener la posibilidad de modificar la información de las personas registradas en el aplicativo o reestablecer su contraseña.	
Validación: El administrador del sistema tendrán la capacidad de ingresar, modificar, eliminar, visualizar y reestablecer la contraseña de todos los usuarios que se encuentren registrados dentro del sistema.	

Tabla IV- 2 Historias de usuarios 2: Administración de usuarios (WEB)

Elaborado por: Obando B. - Vega C.

4.3.3. Mensajería instantánea (WEB)

Código: HUW03	Nombre Historia: Mensajería instantánea (WEB)
Usuario: Administrador	
Prioridad en Negocio: Alta	Riesgo en desarrollo: Alta
Descripción: Como usuario administrador quiero poder mensajear con las personas registradas en el sistema.	
Validación: El administrador del sistema tendrán la capacidad de mensajear con cualquier usuario Android o Ios registrado en el sistema.	

Tabla IV- 3 Historias de usuarios 3: Mensajería instantánea (WEB)

Elaborado por: Obando B. - Vega C.

4.3.4. Autenticación de usuario (Móvil)

Código: HUM01	Nombre Historia: Autenticación de usuario (Móvil)
Usuario: Android/iOS	
Prioridad en Negocio: Alta	Riesgo en desarrollo: Baja
Descripción: Todo usuario que se encuentre registrado en el sistema debe autenticarse antes de poder acceder al aplicativo móvil.	
Validación: Los usuarios registrados anteriormente por un administrador en el sistema web, deben ingresar con su nombre de usuario y contraseña generados por el administrador del sistema.	

Tabla IV- 4 Historias de usuarios 4: Autenticación de usuario (Móvil)

Elaborado por: Obando B. - Vega C.

4.3.5. Visualización de contactos (Móvil)

Código: HUM02	Nombre Historia: Visualización de contactos (Móvil)
Usuario: Android/Ios	
Prioridad en Negocio: Baja	Riesgo en desarrollo: Baja
Descripción: Todo usuario que se encuentre registrado en el sistema podrá ver a todos los usuarios del sistema en una lista.	
Validación: Al momento de autenticarse el usuario podrá observar a todos los usuarios del sistema y podrá actualizar la vista manualmente desde una opción del menú.	

Tabla IV- 5 Historias de usuario 5: Visualización de contactos (Móvil)

Elaborado por: Obando B. - Vega C.

4.3.6. Mensajería instantánea (Móvil)

Código: HUM03	Nombre Historia: Mensajería instantánea (Móvil)
Usuario: Android/Ios	
Prioridad en Negocio: Alta	Riesgo en desarrollo: Alta
Descripción: Todo usuario que se encuentre registrado en el sistema podrá mensajear con cualquiera de los usuarios registrados.	
Validación: Al momento de autenticarse el usuario podrá observar a todos los usuarios del sistema y podrá actualizar la vista manualmente desde una opción del menú, además a esto podrá enviar y recibir mensajes de los demás usuarios en una ventana de “chat” por cada usuario.	

Tabla IV- 6 Historias de usuarios 6: Mensajería instantánea (Móvil)

Elaborado por: Obando B. - Vega C.

4.4. Estimación y análisis de historias de usuarios

Con las historias de usuarios ya definidas se procede a realizar una estimación de tiempos y consumo de recursos que se tengan a disposición; a continuación, detallamos los parámetros que debemos tomar en cuenta.

- Desarrolladores a disposición: 2
- Horas de desarrollo diario: 6
- Días a la semana de desarrollo: 5
- 1 punto equivale a una semana ideal de programación
- Velocidad de desarrollo: 3 puntos por iteración

4.4.1. Priorización de historias de usuarios

- **Prioridad Alta:** La implementación de la historia tiene la más alta importancia en el desarrollo.
- **Prioridad Media:** Depende de la implementación de otras historias de mayor prioridad, por lo cual se toman en cuenta cuando una historia de prioridad alta se ha finalizado y exista una relación entre estas.
- **Prioridad Baja:** No tienen un efecto fuerte dentro del desarrollo y serán tomadas en cuenta al final de todas las prioridades.

En la Tabla IV-7 se definirá la prioridad y el riesgo en desarrollo que cada historia de usuario tiene dentro del proyecto tomando en cuenta las prioridades antes mencionadas.

Código	Título	Prioridad	Riesgo en desarrollo
HUW01	Autenticación de usuario (WEB)	Media	Baja
HUW02	Administración de usuarios (WEB)	Alta	Media
HUW03	Mensajería instantánea (WEB)	Media	Alta
HUM01	Autenticación de usuario (Móvil)	Media	Baja

Código	Título	Prioridad	Riesgo en desarrollo
HUM02	Visualización de contactos (Móvil)	Baja	Baja
HUM03	Mensajería instantánea (Móvil)	Media	Alta

Tabla IV- 7 Priorización de historias de usuarios

Elaborado por: Obando B. - Vega C.

En base a la información de la Tabla IV-7 se define que los módulos de mayor importancia para la implementación serán aquellos que tengan como nivel de prioridad “Media” en adelante y al mismo tiempo un riesgo en desarrollo “Media” y posteriores.

Con esto se define que los módulos con código HUW03, HUW02 y HUM03 tienen la mayor prioridad de desarrollo dentro del proyecto; los módulos con código HUW01 y HUM02 se encontrarían en el siguiente nivel de prioridad y finalmente el módulo con código HUM01 tendría la prioridad más baja dentro del desarrollo del prototipo.

4.4.2. Estimación de esfuerzo

En la Tabla IV-8 se definirá que esfuerzo es necesario para cada una de los módulos que integran el sistema, para esto se utilizará una escala de 1 a 3 puntos donde, 1 punto equivale a un menor esfuerzo mientras que 3 puntos equivaldrá al mayor esfuerzo.

Código	Título	Esfuerzo ideal	Riesgo en desarrollo
HUW01	Autenticación de usuario (WEB)	1 punto	Baja
HUW02	Administración de usuarios (WEB)	1 punto	Media
HUW03	Mensajería instantánea (WEB)	3 puntos	Alta

Código	Título	Esfuerzo ideal	Riesgo en desarrollo
HUM01	Autenticación de usuario (Móvil)	1 punto	Baja
HUM02	Visualización de contactos (Móvil)	1 punto	Baja
HUM03	Mensajería instantánea (Móvil)	2 puntos	Alta

Tabla IV- 8 Estimación de esfuerzo

Elaborado por: Obando B. - Vega C.

4.4.3. Iteraciones

En base a los datos adquiridos de la tabla IV-8, se define que tomará 9 puntos y tomando el valor de nuestra velocidad de desarrollo definimos el número de iteraciones que van a existir.

$$\frac{\text{Número de puntos de las historias de usuarios}}{\text{Velocidad de desarrollo}} = \frac{9 \text{ puntos}}{3 \frac{\text{puntos}}{\text{iteración}}} = 3 \text{ iteraciones}$$

La Tabla IV-9 contendrá los módulos que serán desarrollados en cada iteración del proceso de implementación del prototipo, con su respectivo código y esfuerzo ideal adquirido de la Tabla IV-8.

Código	Título	Esfuerzo Ideal
Iteración 1		
HUW02	Administración de usuarios (WEB)	1 punto
HUW01	Autenticación de usuario (WEB)	1 punto
HUM01	Autenticación de usuario (Móvil)	1 punto

Código	Título	Esfuerzo Ideal
Iteración 2		
HUM02	Visualización de contactos (Móvil)	1 punto
HUM03	Mensajería instantánea (Móvil)	2 puntos
Iteración 3		
HUW03	Mensajería instantánea (WEB)	3 puntos

Tabla IV- 9 Iteraciones

Elaborado por: Obando B. - Vega C.

4.4.4. Análisis de historias de usuarios

Las historias de usuario contienen varios procesos que la persona puede realizar y deben ser desglosados para un mejor entendimiento de lo que engloban. Cada historia de usuario contiene información vital para el diseño de cada módulo, por este motivo, una historia de usuario debe ser diseñada lo más similar a la idea que tiene el cliente.

4.4.4.1. Autenticación de usuario (WEB)

Nombre	Autenticación de usuario (WEB)	
Código:	HUW01	
Actor:	Administrador	
Descripción:	El actor ingresa al sistema a través de sus credenciales (nombre de usuario y contraseña), registradas en la base de datos	
Precondiciones:	El actor debe estar registrado en el sistema	
Post-Condiciones:	Se actualiza el estado de conexión del actor que ingresó al sistema	
Flujo Principal:	ACTOR	SISTEMA
	1 El actor ingresa al URL destinado para el sistema	2 El sistema muestra la pantalla de autenticación de usuario
	3 El actor ingresa el nombre de usuario y la contraseña que le corresponde	4 El sistema comprueba que la información ingresada corresponda a los datos existentes en la base de datos
		5 El sistema muestra la pantalla de administración de usuarios
		6 El sistema muestra todos los usuarios registrados en el sistema

Flujo Principal:	ACTOR	SISTEMA
		7 El sistema actualiza el estado de conexión del actor dentro de la base de datos
Flujo alternativo:		4.1 El sistema solicita nuevamente las credenciales al ser incorrectas
		5.1 Si es la primera vez que el actor ingresa al sistema, el sistema mostrará la pantalla de cambio de contraseña, y posterior al cambio de contraseña el sistema muestra la pantalla de Administración de usuarios
Resultado:	El actor ingresa al sistema a la pantalla de administración de usuarios exitosamente	

Tabla IV- 10 Análisis de historias de usuarios 1: Autenticación de usuario (WEB)

Elaborado por: Obando B. - Vega C.

4.4.4.2. Administración de usuarios (WEB)

4.4.4.2.1. Ingresar usuario

Nombre	Ingresar usuario	
Código:	HUW02-01	
Actor:	Administrador	
Descripción:	El actor ingresa un nuevo usuario a través de un formulario que provee el sistema	
Precondiciones:	El actor debe estar registrado en el sistema El actor debe tener un rol de administrador	
Post-Condiciones:		
Flujo Principal:	ACTOR	SISTEMA
	1 El actor ingresa a la pantalla de administración de usuarios	2 El sistema muestra la pantalla de administración de usuarios
		3 El sistema muestra todos los usuarios registrados en el sistema
	4 El actor da clic en la opción de nuevo usuario	5 El sistema muestra el formulario con los campos necesarios para la creación de un nuevo usuario
	6 El actor ingresa los campos: “usuario, cedula de identidad, nombres, apellidos, email, imagen”	
	7 El actor da clic en guardar	8 El sistema almacena la información dentro de la base de datos y crea la contraseña automáticamente con el número de identificación de la persona
		9 El sistema actualiza la vista de la pantalla de administración de usuarios

Flujo alternativo:	ACTOR	SISTEMA
		8.1 Si existen campos obligatorios sin llenar el sistema mostrará un mensaje de error
Resultado:	Un nuevo usuario es almacenado en la base de datos	

Tabla IV- 11 Análisis de historias de usuarios 2: Administración de usuarios (WEB)-Ingresar usuario

Elaborado por: Obando B. - Vega C.

4.4.4.2.2. Modificar usuario

Nombre	Modificar usuario	
Código:	HUW02-02	
Actor:	Administrador	
Descripción:	El actor desea modificar la información de un usuario ya existente	
Precondiciones:	El actor debe estar registrado en el sistema El actor debe tener un rol de administrador	
Post-Condiciones:		
Flujo Principal:	ACTOR	SISTEMA
	1 El actor ingresa a la pantalla de administración de usuarios	2 El sistema muestra la pantalla de administración de usuarios
		3 El sistema muestra todos los usuarios registrados en el sistema
	4 El actor da clic en el botón de editar usuario, correspondiente al usuario que va a editar	5 El sistema muestra el formulario con la información del usuario a editar en cada uno de los campos
	6 El actor modifica los campos deseados	
	7 El actor da clic en guardar	8 El sistema actualiza la información del usuario dentro de la base de datos
		9 El sistema actualiza la vista de la pantalla de administración de usuarios
Flujo alternativo:		8.1 Si existen campos obligatorios sin llenar el sistema mostrará un mensaje de error
Resultado:	La información del usuario a modificar será actualizada dentro de la base de datos	

Tabla IV- 12 Análisis de historias de usuarios 3: Administración de usuarios (WEB)-Modificar usuario

Elaborado por: Obando B. - Vega C.

4.4.4.2.3. Eliminar usuario

Nombre	Eliminar usuario	
Código:	HUW02-03	
Actor:	Administrador	
Descripción:	El actor desea eliminar un usuario existente dentro del sistema	
Precondiciones:	El actor debe estar registrado en el sistema El actor debe tener un rol de administrador	
Post-Condiciones:		
Flujo Principal:	ACTOR	SISTEMA
	1 El actor ingresa a la pantalla de administración de usuarios	2 El sistema muestra la pantalla de administración de usuarios
		3 El sistema muestra todos los usuarios registrados en el sistema
	4 El actor da clic en el botón de eliminar usuario, correspondiente al usuario que va a eliminar	5 El sistema elimina toda información relacionada al usuario (mensajes) y posterior a ello elimina el usuario
Flujo alternativo:		
Resultado:	Toda la información del usuario a eliminar será borrada de la base de datos, incluyendo el usuario	

Tabla IV- 13 Análisis de historias de usuarios 4: Administración de usuarios (WEB)-Eliminar usuario

Elaborado por: Obando B. - Vega C.

4.4.4.2.4. Visualizar usuarios

Nombre	Visualizar usuario	
Código:	HUW02-04	
Actor:	Cualquier rol	
Descripción:	El actor desea visualizar los usuarios del sistema	
Precondiciones:	El actor debe estar registrado en el sistema	
Post-Condiciones:		
Flujo Principal:	ACTOR	SISTEMA
	1 El actor ingresa a la pantalla de administración de usuarios	2 El sistema muestra la pantalla de administración de usuarios
		3 El sistema muestra todos los usuarios registrados en el sistema
	4 El actor desea buscar un usuario específico y llena los campos de búsqueda	

Flujo Principal:	ACTOR	SISTEMA
	5 El actor da clic en buscar	6 El sistema filtra por los datos ingresados por el usuario
		7 El sistema muestra el usuario correspondiente
Flujo alternativo:		7.1 El sistema no muestra nada por no existir el usuario buscado
Resultado:	El actor visualiza el usuario buscado o todos los usuarios	

Tabla IV- 14 Análisis de historias de usuarios 5: Administración de usuarios (WEB)-Visualizar usuarios

Elaborado por: Obando B. - Vega C.

4.4.4.2.5. Reestablecer contraseña

Nombre	Reestablecer contraseña	
Código:	HUW02-05	
Actor:	Administrador	
Descripción:	El actor desea reestablecer la contraseña de un usuario registrado en el sistema	
Precondiciones:	El actor debe tener un rol de administrador	
Post-Condiciones:		
Flujo Principal:	ACTOR	SISTEMA
	1 El actor ingresa a la pantalla de administración de usuarios	2 El sistema muestra la pantalla de administración de usuarios
	3 El actor da clic en el botón de reestablecer contraseña, correspondiente al usuario al que va a reestablecer la contraseña	4 El sistema consulta a la base de datos el número de identificación del usuario y reestablece la contraseña con el número de identificación.
Flujo alternativo:		
Resultado:	La contraseña será reestablecida al valor por defecto siendo este el número de identificación del usuario	

Tabla IV- 15 Análisis de historias de usuarios 6: Administración de usuarios (WEB)-Reestablecer contraseña

Elaborado por: Obando B. - Vega C.

4.4.4.3. Mensajería instantánea (WEB)

4.4.4.3.1. Enviar mensaje

Nombre	Enviar mensaje	
Código:	HUW03-01	
Actor:	Administrador	
Descripción:	El actor desea enviar un mensaje a un usuario registrado en el sistema	
Precondiciones:	El actor debe estar registrado en el sistema El actor debe tener un rol de administrador	
Post-Condiciones:		
Flujo Principal:	ACTOR	SISTEMA
	1 El actor ingresa a la pantalla de mensajería	2 El sistema muestra la pantalla de mensajería
		3 El sistema muestra los usuarios registrados en una lista
	4 El actor da clic al usuario que desea enviar el mensaje	5 El sistema despliega el chat del usuario seleccionado con su imagen, nombre de usuario y los mensajes enviados
Nombre	Enviar mensaje	
	6 El actor escribe el mensaje a enviar	
	7 El actor da clic en enviar	8 El sistema envía el mensaje al usuario seleccionado a través de GCM
		9 El sistema almacena el mensaje en la base de datos y actualiza la vista de los mensajes
		10 El sistema móvil envía un mensaje de respuesta como comprobación de haber recibido el mensaje
Flujo alternativo:		10.1 Si el mensaje de confirmación no es recibido el mensaje es puesto en cola para envío
Resultado:	Envío del mensaje al usuario seleccionado	

Tabla IV- 16 Análisis de historias de usuarios 7: Mensajería instantánea (WEB)-Enviar mensaje

Elaborado por: Obando B. - Vega C.

4.4.4.3.2. Recibir mensaje

Nombre	Recibir mensaje	
Código:	HUW03-02	
Actor:	Administrador	
Descripción:	El actor recibe un mensaje de un usuario Android o Ios	
Precondiciones:	El actor debe estar registrado en el sistema El usuario Android o Ios debe estar registrado en el sistema El actor debe tener un rol de administrador	
Post-Condicion:		
Flujo Principal:	ACTOR	SISTEMA
		1 El sistema recibe un mensaje a través del servicio de GCM
		2 El sistema procesa el mensaje y lo almacena en la base de datos
		3 El sistema envía un mensaje de confirmación al usuario Android o Ios que envió el mensaje por haber recibido el mensaje
	4 El actor ingresa a la pantalla de mensajería	6 El sistema muestra la pantalla de mensajería
	6 El actor da clic en el usuario que le envió el mensaje	7 El sistema actualiza la vista de los mensajes entre el usuario Android o Ios y el administrador
Flujo alternativo:		
Resultado:	Recepción y visualización correcta del mensaje entre el usuario Android o Ios y el administrador	

Tabla IV- 17 Análisis de historias de usuarios 8: Mensajería instantánea (WEB)-Recibir mensaje

Elaborado por: Obando B. - Vega C.

4.4.4.4. Autenticación de usuario (Móvil)

Nombre	Autenticación de usuario (WEB)	
Código:	HUM01	
Actor:	Usuario Android/Ios	
Descripción:	El actor ingresa al sistema a través de sus credenciales (nombre de usuario y contraseña), registradas en la base de datos	
Precondiciones:	El actor debe estar registrado en el sistema	
Post-Condicion:	Se actualiza el estado de conexión del actor que ingresó al sistema	
Flujo Principal:	ACTOR	SISTEMA
	1 El actor abre el aplicativo móvil desde su dispositivo inteligente	2 El sistema muestra la pantalla de autenticación de usuario
	3 El actor ingresa el nombre de usuario y la contraseña que le corresponde	

Flujo Principal:	ACTOR	SISTEMA
	4 El actor presiona ingresar	5 El sistema consume el servicio web de obtención de usuarios registrados
		6 El sistema comprueba que la información ingresada corresponda a los datos existentes en la base de datos
		7 El sistema genera el identificador de GCM y lo almacena en la base de datos a través del servicio web
		8 El sistema muestra la pantalla principal del aplicativo móvil
		9 El sistema actualiza el estado de conexión del actor
Flujo alternativo:		6.1 El sistema solicita nuevamente las credenciales al ser estas incorrectas
		8.1 Si es la primera vez que el actor ingresa al sistema, el sistema mostrará la pantalla de cambio de contraseña, y posterior al cambio de contraseña el sistema muestra la pantalla principal del aplicativo móvil
Resultado:	El actor ingresa al sistema a la pantalla principal del aplicativo móvil exitosamente	

Tabla IV- 18 Análisis de historias de usuarios 9: Autenticación de usuario (Móvil)

Elaborado por: Obando B. - Vega C.

4.4.4.5. Visualización de contactos (Móvil)

Nombre	Autenticación de usuario (WEB)	
Código:	HUM02	
Actor:	Usuario Android/Ios	
Descripción:	El actor desea actualizar los contactos de la mensajería manualmente	
Precondiciones:	El actor debe estar registrado en el sistema	
Post-Condiciones:		
Flujo Principal:	ACTOR	SISTEMA
	1 El actor ingresa a la pantalla principal del aplicativo móvil	2 El sistema presenta la pantalla principal del aplicativo móvil sin contactos
	3 El actor presiona la opción de actualizar contactos del menú de opciones	4 El sistema consume el servicio web correspondiente a la adquisición de los usuarios registrados en el sistema
		5 El sistema almacena los contactos en la base de datos local del dispositivo inteligente

Flujo Principal:	ACTOR	SISTEMA
		6 El sistema muestra los contactos en la pantalla principal del aplicativo móvil
Flujo alternativo:		4.1 El servicio web no se encuentra disponible volver al paso 3
		4.2 No existe conexión a internet se muestra un mensaje para que encienda el WI-FI o consumo de datos y volver al paso 3
		5.1 Si los contactos ya existen procede a actualizar los identificadores GCM o comprueba la existencia de nuevos usuarios volver al paso 4
Resultado:	El actor visualiza los contactos en una lista en la pantalla principal del aplicativo móvil	

Tabla IV- 19 Análisis de historias de usuarios 10: Visualización de contactos (Móvil)

Elaborado por: Obando B. - Vega C.

4.4.4.6. Mensajería instantánea (Móvil)

4.4.4.6.1. Enviar mensaje

Nombre	Enviar mensaje	
Código:	HUM03-01	
Actor:	Usuario Android/Ios	
Descripción:	El actor desea enviar un mensaje a un usuario Android/Ios o a un administrador	
Precondiciones:	El actor debe estar registrado en el sistema El actor debe tener un identificador GCM	
Post-Condiciones:		
Flujo Principal:	ACTOR	SISTEMA
	1 El actor ingresa al menú y selecciona mensajería	2 El sistema muestra la pantalla de mensajería con los contactos registrados en el sistema
	3 El actor selecciona el contacto a enviar el mensaje	4 El sistema muestra la pantalla de mensajería entre el contacto y el actor
	5 El actor escribe el mensaje a enviar	
	6 El actor presiona enviar	7 El sistema almacena el mensaje en la base de datos local y envía el mensaje a través del servicio de GCM al contacto seleccionado

Flujo Principal:	ACTOR	SISTEMA
		8 El sistema receptor envía un mensaje de confirmación de haber recibido el mensaje
		9 El sistema muestra el mensaje enviado en la pantalla de mensajería entre el contacto y el actor
Flujo alternativo:		4.1 Si existen mensajes consume el servicio web y muestra todos los mensajes entre el contacto y el actor, caso contrario queda vacío
		8.1 Si el sistema no recibe un mensaje de confirmación de haber recibido el mensaje, este es enviado después de un lapso de tiempo
		7.1 Si el sistema no tiene una conexión de internet el mensaje es enviado después de un lapso de tiempo
Resultado:	El mensaje es enviado exitosamente y presentado por pantalla en ambos dispositivos	

Tabla IV- 20 Análisis de historias de usuarios 11: Mensajería instantánea (Móvil)-Enviar mensaje

Elaborado por: Obando B. - Vega C.

4.4.4.6.2. Recibir mensaje

Nombre	Recibir mensaje	
Código:	HUM03-02	
Actor:	Usuario Android/Ios	
Descripción:	El actor recibe un mensaje desde un usuario Android/Ios o a un administrador	
Precondiciones:	El actor debe estar registrado en el sistema El actor debe tener un identificador GCM Tener los contactos en la base local del dispositivo inteligente	
Post-Condiciones:		
Flujo Principal:	ACTOR	SISTEMA
		1 El sistema recibe un mensaje de otro contacto a través del servicio GCM
		2 El sistema identifica el contacto de donde proviene el mensaje y lo almacena en la base de datos local del dispositivo inteligente
		3 El sistema envía un mensaje de confirmación por haber recibido el mensaje
	4 El actor da clic en el contacto al cual pertenece el mensaje	5 El sistema carga los mensajes entre el actor y el contacto mostrando el nuevo mensaje

Flujo Principal:	ACTOR	SISTEMA
Flujo alternativo:		<p>3.1 Si el aplicativo no se encuentra abierto, muestra una notificación en el dispositivo inteligente de haber recibido un mensaje</p> <p>5.1 Si el aplicativo se encuentra abierto en la pantalla de mensajería entre el contacto y el actor, actualiza esta pantalla mostrando el nuevo mensaje</p>
Resultado:	El mensaje es recibido y visualizado por el actor exitosamente	

Tabla IV- 21 Análisis de historias de usuarios 12: Mensajería instantánea (Móvil)-Recibir mensaje

Elaborado por: Obando B. - Vega C.

4.5. Diagrama conceptual

En el Diagrama IV-13 se definirá las tablas que pertenecen a la base de datos para que el prototipo tenga un correcto funcionamiento en la administración de usuarios y mensajería instantánea, dentro de estas se almacenará información respecto a cada usuario y la información de los mensajes de texto que se envíen entre el aplicativo web y el móvil.

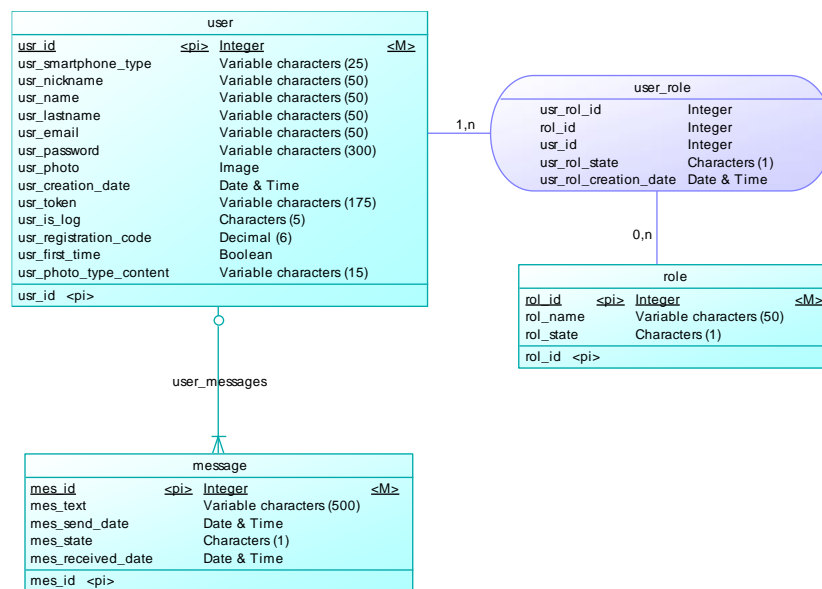


Diagrama IV- 1 Diagrama conceptual

Elaborado por: Obando B. - Vega C.

4.6. Diagrama de clases

El Diagrama IV-14 contendrá las variables y métodos que serán implementados en las clases que componen las entidades dentro del prototipo.

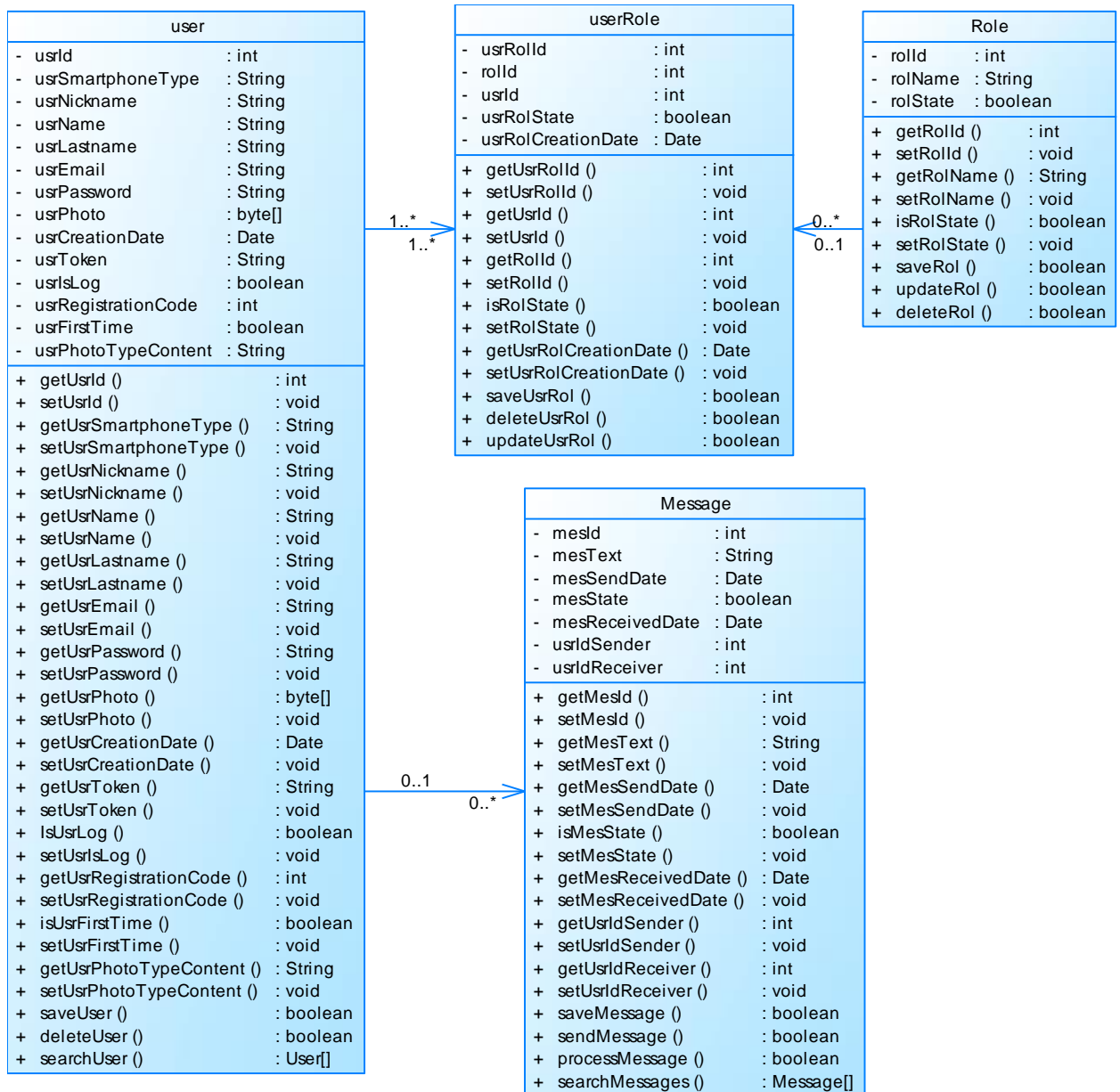


Diagrama IV- 2 Diagrama de clases

Elaborado por: Obando B. - Vega C.

4.7. Requerimientos de software y hardware para el desarrollo y despliegue de la Administración del Sistema

Es necesario equipamiento que tenga la capacidad de albergar la información dentro de un servidor de base de datos, equipos con los recursos necesarios para el desarrollo del prototipo y la realización de pruebas necesarias dentro de un ambiente controlado, por este motivo se ha definido parámetros para la adecuada implementación del aplicativo.

Requisito mínimo para la plataforma de Administración del Sistema (Servidor de base de datos)	
Categoría	Características
Hardware	Procesador Core 2 duo 1.06Ghz
	Tarjeta Fast Ethernet
	Memoria RAM 2Gb ddr3 1330Mhz
	Disco duro en RAID10 o RAID 0+1 espacio para software gestor base datos 5GB
	Espacio para Base de Datos 500GB
Software	Sistema Operativo Windows Server 2008 o CentOS
	Gestor y Administrador de Bases de datos (MySQL Enterprise Edition, PostgreSQL, ORACLE Database)

Tabla IV- 22 Requisito mínimo para la plataforma de Administración del Sistema (Servidor de base de datos)

Elaborado por: Obando B. - Vega C.

Requisito mínimo para la plataforma de Administración del Sistema (Servidor de Aplicaciones)	
Categoría	Características
Hardware	Procesador Intel Pentium 1Ghz (Core 2 duo para mejor rendimiento)
	Tarjeta Fast Ethernet
	Memoria RAM 4Gb ddr3 1330Mhz
	Espacio de Disco Duro para servidor de aplicaciones 4GB
Software	Sistema Operativo Windows Server 2008 o CentOS
	JVM versión 7.x en adelante
	JRE versión 7.x en adelante
	Jboss AS 7.x en adelante

Tabla IV- 23 Requisito mínimo para la plataforma de Administración del Sistema (Servidor de Aplicaciones)

Elaborado por: Obando B. - Vega C.

Requisito mínimo para la plataforma de Administración del Sistema (Equipo de desarrollo)	
Categoría	Características
Hardware	Procesador Intel Core i3 2.5Ghz
	Tarjeta Fast Ethernet
	Memoria RAM 8Gb ddr3 1330Mhz
	Espacio de Disco Duro para herramientas de desarrollo 10GB
Software	Sistema Operativo Windows 8.1 en adelante o CentOS
	JVM versión 7.x en adelante
	JRE versión 7.x en adelante
	Jboss AS 7.x en adelante
	Navegador Google Chrome, Mozilla Firefox en última actualización
	Eclipse 4.6 Neon

Tabla IV- 24 Requisito mínimo para la plataforma de Administración del Sistema (Equipo de desarrollo)

Elaborado por: Obando B. - Vega C.

4.8. Requerimientos de Hardware y Software para el desarrollo y despliegue del aplicativo móvil

De la misma forma que es requerido un mínimo de recursos para la parte de Administración del sistema, en el aplicativo móvil es necesario equipamiento para la realización de pruebas, desarrollo y despliegue de la aplicación.

Requisito mínimo para la plataforma móvil (Equipo de desarrollo)	
Hardware	Procesador Intel Core i3 2.5Ghz
	Tarjeta Fast Ethernet
	Memoria RAM 8Gb ddr3 1330Mhz
	Espacio de Disco Duro para herramientas de desarrollo 10GB
Software	Sistema Operativo Windows 8.1 en adelante o CentOS
	JVM versión 7.x en adelante
	JRE versión 7.x en adelante
	Android Studio

Tabla IV- 25 Requisito mínimo para la plataforma móvil (Equipo de desarrollo)

Elaborado por: Obando B. - Vega C.

Requisito mínimo para la plataforma móvil (Equipo de despliegue)	
Hardware	Chip WI-FI 802.11 a/b/g/n/ac
	Acceso a Internet Móvil
	Memoria RAM 1GB
	Procesador Dual-Core 1.2Ghz Cortex-A9
	Batería 1650mAh
	Pantalla de 4'
	Almacenamiento de 20MB
Software	Sistema Operativo Android Versión 4.0.x ICE CREAM
	Servicio GooglePlay 9.0.1

Tabla IV- 26 requisito mínimo para la plataforma móvil (Equipo de despliegue)

Elaborado por: Obando B. - Vega C.

4.9. Diccionario de datos

Contiene un listado de todos los datos de un sistema, utilizado para dar una mayor precisión sobre los datos que se manejan en el sistema, este es desarrollado durante el análisis de flujo de datos

Define los datos de entrada, salida, componentes del almacenamiento, flujos, etc. Guarda y organiza los detalles del diagrama de flujo de datos, es un conjunto de metadatos que contiene las características lógicas de los datos; su nombre, descripción, alias, contenido y organización, valores por defecto (si existieran), brindando un mejor entendimiento para que los analistas que participan en la determinación de los requerimientos conozcan a detalle los datos del sistema.

4.9.1. Tabla usuarios

Descripción: Base de datos que contendrá la información de los usuarios registrados en el sistema			
Campo	Tamaño	Tipo de Dato	Descripción
usr_id	11	int	Identificador del registro
usr_smartphone_type	25	varchar	Tipo de equipo celular
usr_nickname	50	varchar	Nombre de ingreso
usr_name	50	varchar	Nombre
usr_lastname	50	varchar	Apellido
usr_email	50	varchar	Correo Electrónico
usr_password	300	varchar	Contraseña de ingreso
usr_photo	-	longblob	Fotografía
usr_creation_date	-	datetime	Fecha de creación
usr_token	175	varchar	Identificador GCM
usr_is_log	5	char	Estado de conexión
usr_registration_code	(6,0)	decimal	Código de registro
usr_first_time	1	tinyint	Autenticación por primera vez
usr_photo_content_type	15	varchar	Tipo de dato de Fotografía
Relaciones		Campos Clave	
Tabla	Campo de relación	usr_id	Primary key (NOT NULL)
Message	usr_id_receiver		
Message	usr_id_sender		
user_role	usr_id		

Tabla IV- 27 Diccionario de Datos 1: Tabla usuarios

Elaborado por: Obando B. - Vega C.

4.9.2. Tabla rol

Descripción: Base de datos que contendrá los roles para los usuarios			
Campo	Tamaño	Tipo de dato	Descripción
rol_id	11	int	Identificador
rol_name	50	varchar	Nombre
rol_state	1	char	Estado de actividad
Relaciones		Campos clave	
Tabla	Campo de relación	rol_id	Primary key (NOT NULL)
user_role	rol_id		

Tabla IV- 28 Diccionario de Datos 2: Tabla rol

Elaborado por: Obando B. - Vega C.

4.9.3. Tabla relación usuario-rol

Descripción: Base de datos que contendrá la relación existente entre la tabla de usuarios y roles			
Campo	Tamaño	Tipo de dato	Descripción
usr_rol_id	11	int	Identificador
rol_id	11	int	Identificador del rol
usr_id	11	int	Identificador del usuario
usr_rol_state	1	char	Estado de actividad
usr_rol_creation_date	-	datetime	Fecha de creación
Relaciones		Campos clave	
Tabla	Campo de relación	usr_rol_id	Primary key (NOT NULL)
Role	rol_id	rol_id	Foreign key (rol)
User	usr_id	usr_id	Foreign key (user)

Tabla IV- 29 Diccionario de Datos 3: Tabla usuario-rol

Elaborado por: Obando B. - Vega C.

4.9.4. Tabla mensaje

Descripción: Base de datos que contendrá los mensajes enviados entre los usuarios			
Campo	Tamaño	Tipo de dato	Descripción
mes_id	11	int	Identificador
usr_id_receiver	11	int	Identificador del usuario receptor
usr_id_sender	11	int	Identificador del usuario emisor
mes_text	500	varchar	Contenido del mensaje
mes_send_date	-	datetime	Fecha de envío
mes_state	1	char	Estado
mes_received_date	-	datetime	Fecha de recepción
Relaciones		Campos clave	
Tabla	Campo de relación	mes_id	Primary key (NOT NULL)
User	usr_id	usr_id_receiver	Foreign key (user)
User	usr_id	usr_id_sender	Foreign key (user)

Tabla IV- 30 Diccionario de Datos 4: Tabla mensaje

Elaborado por: Obando B. - Vega C.

CAPITULO V

El capítulo 5 contiene toda la información necesaria para el diseño gráfico de las interfaces para ambos aplicativos tanto web como móvil, se desarrolla un prototipado de cómo se verán las interfaces gráficas y los componentes que estarán incluidos dentro de cada una.

5. Diseño del prototipo

Para el desarrollo es necesario establecer estándares dentro de la escritura del código, el diseño de interfaces y el manejo de los archivos generados dentro del proyecto del prototipo; para un mejor entendimiento de los desarrolladores se diseñó un molde de las interfaces que serán aplicadas dentro del prototipo las cuales serán transformadas en código.

5.1. Interfaces

Para el diseño de interfaces se tomó en cuenta la interacción que el usuario tiene con los dispositivos para crear un ambiente más amigable y de fácil manejo, se utilizó UI frameworks, en este caso los templates diseñados por Primefaces para el diseño de las interfaces del aplicativo web y los estándares de Material Design para las interfaces del aplicativo móvil, creando un diseño completo y utilizando librerías de apoyo para el control de usuarios. El objetivo de las interfaces es crear un uso instintivo para el usuario, de fácil entendimiento y manejo sin la necesidad de un manual o una capacitación para su uso.

5.1.1. Interfaces aplicativo web

5.1.1.1. Pantalla de autenticación

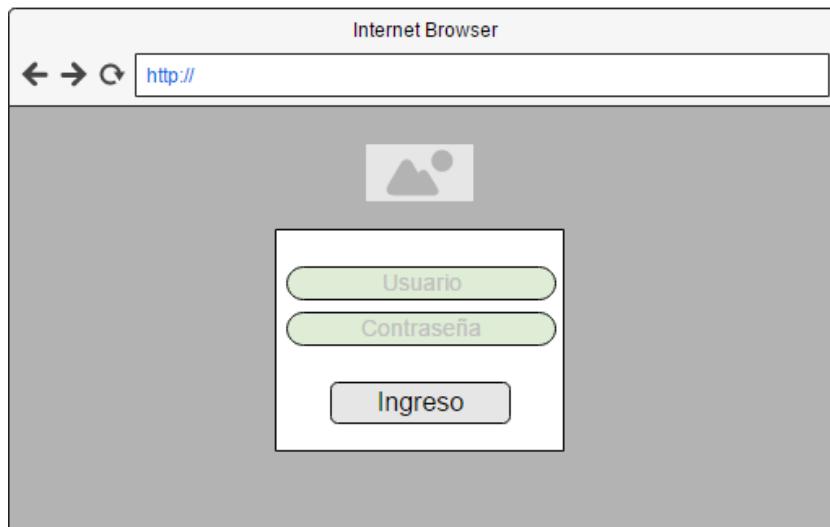


Diagrama V- 1 Interfaces aplicativo web 1: Pantalla de autenticación

Elaborado por: Obando B. - Vega C.

La composición de esta interfaz contiene el logo del aplicativo con las credenciales necesarias para la autenticación del usuario, donde esta información debe encontrarse registrada dentro de la base de datos del sistema para un correcto ingreso del usuario al sistema.

5.1.1.2. Pantalla de cambio de contraseña

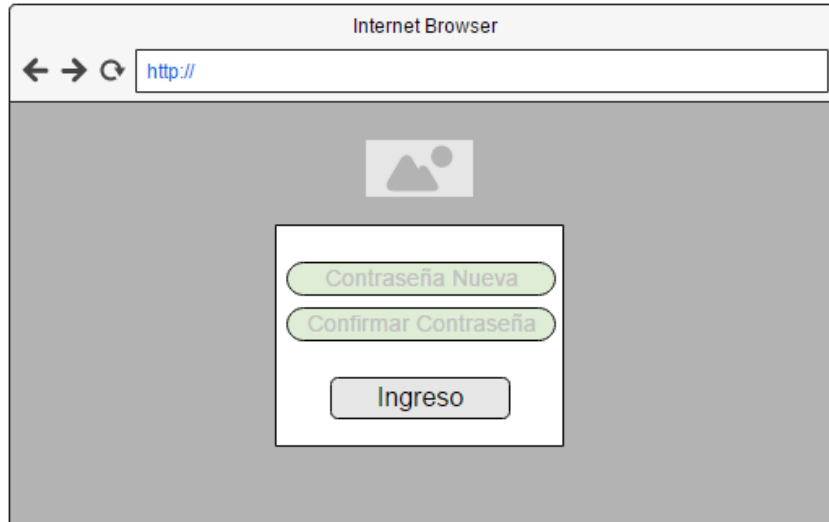


Diagrama V- 2 Interfaces aplicativo web 2: Pantalla de cambio de contraseña

Elaborado por: Obando B. - Vega C.

La interfaz de cambio de contraseña aparecerá si es la primera vez que el usuario ingresa al sistema, se solicitará inmediatamente la contraseña nueva y la confirmación de la misma se realizarán los controles necesarios para determinar si las contraseñas coinciden. Como punto extra la contraseña inicial no es solicitada al usuario, es autogenerada en base a información conocida por el usuario y deberá realizar el cambio de forma obligatoria.

5.1.1.3. Pantalla de administración de usuarios

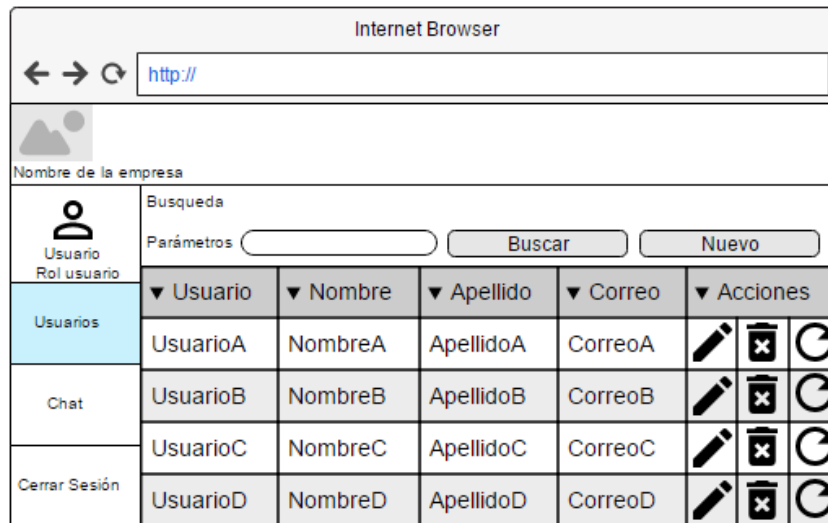


Diagrama V- 3 Interfaces aplicativo web 3: Pantalla de administración de usuarios

Elaborado por: Obando B. - Vega C.

Dentro de la interfaz de administración de usuarios encontraremos todos los usuarios registrados y las acciones que se pueden realizar sobre ellos, estas acciones serán la modificación, eliminación o el restablecimiento de la contraseña y serán visibles dependiendo del rol que tenga el usuario, si no tiene un rol de “Administrador” no se visualizarán estas acciones y solo podrá observar los usuarios que se encuentren registrados en el sistema. La interfaz está dividida en varias secciones:

- En la cabecera podemos encontrar el logo del aplicativo y a quien pertenece el sistema con su nombre respectivo.

- En la sección de la izquierda un menú para navegar a través de las funciones que integre el sistema, por motivo de disertación se integra la mensajería instantánea como una opción y es posible añadir cualquier otra funcionalidad necesaria.
- La parte central se encuentra la información necesaria para identificar los usuarios con las acciones especificadas anteriormente.
- Sobre la sección central encontramos un área de búsqueda en base a parámetros definidos los cuales son parte de la información que integra a cada usuario, para filtrar si existe la necesidad de encontrar un usuario específico si la lista es demasiado extensa.

5.1.1.4. Pantalla de mensajería instantánea

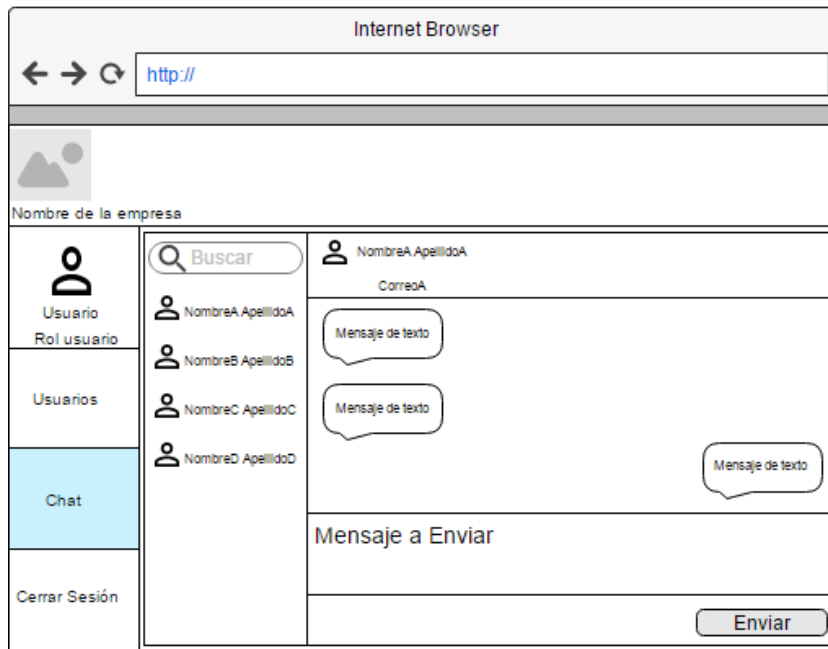


Diagrama V- 4 Interfaces aplicativo web 4: Pantalla de mensajería instantánea

Elaborado por: Obando B. - Vega C.

Dentro de la mensajería instantánea el usuario autenticado observará todos los usuarios que tengan un dispositivo móvil registrado en el sistema y podrá comunicarse a través de mensajes de texto con estos usuarios. Del mismo modo, solo usuarios con el rol de administrador tendrán acceso a esta pantalla dentro del sistema. Se consideró que exista como en la administración de usuarios un componente de búsqueda de usuarios para facilitar su manejo.

5.1.2. Interfaces aplicativo móvil

5.1.2.1. Pantalla de autenticación de usuario



Diagrama V- 5 Interfaces aplicativo móvil 1: Pantalla de autenticación de usuario

Elaborado por: Obando B. - Vega C.

En la interfaz de autenticación los componentes existentes son el usuario y la contraseña asignados por el administrador del aplicativo, los datos a ingresar deben encontrarse registrados dentro de la base de datos para realizar una correcta autenticación.

5.1.2.2. Pantalla de cambio de contraseña

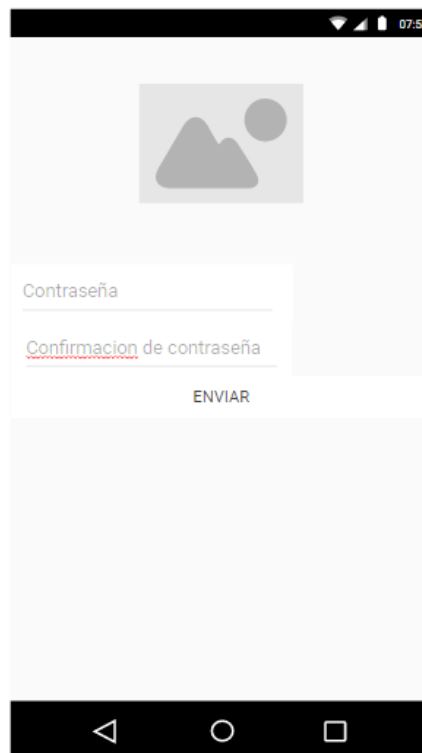


Diagrama V- 6 Interfaces aplicativo móvil 2: Pantalla de cambio de contraseña

Elaborado por: Obando B. - Vega C.

De la misma manera que funciona el aplicativo web, si es la primera vez que el usuario ingresa al aplicativo, se solicitará que cambie su contraseña, por motivos de seguridad ya que esta fue autogenerada en base a información del usuario adquirida al momento de registrarse en el sistema web.

5.1.2.3. Menú de navegación

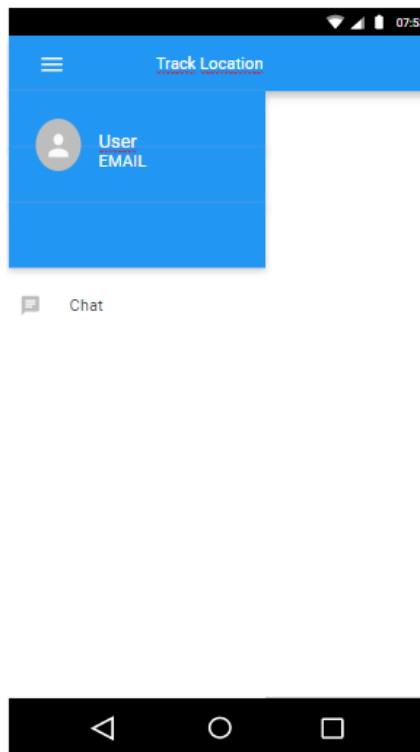


Diagrama V- 7 Interfaces aplicativo móvil 3: Menú de navegación

Elaborado por: Obando B. - Vega C.

El sistema de navegación contiene cualquier módulo existente dentro del aplicativo, por motivo de disertación, solo se ha incluido el chat entre usuarios y este puede ser utilizado por cualquier usuario que se encuentre registrado dentro del sistema.

5.1.2.4. Pantalla de usuarios y datos de mensajería

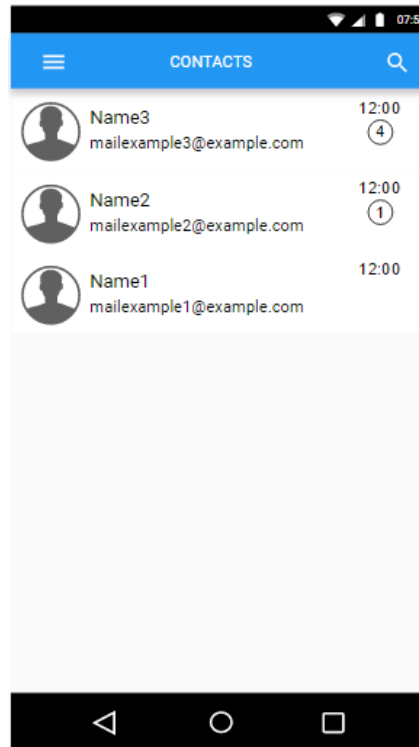


Diagrama V- 8 Interfaces aplicativo móvil 4: Pantalla de usuarios y datos de mensajería

Elaborado por: Obando B. - Vega C.

Aquí observaremos la información de los contactos que se encuentren registrados en el sistema, además de una notificación de los mensajes que se hayan enviado y no se encuentren revisados por el usuario dueño del dispositivo móvil, al hacer clic en cualquiera de estos usuarios despliega la pantalla de mensajería entre ese usuario y el dueño del móvil.

5.1.2.5. Pantalla de mensajería entre usuarios

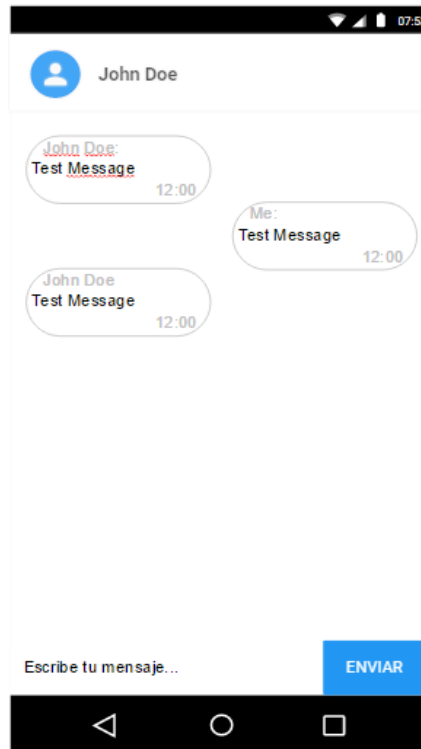


Diagrama V- 9 Interfaces aplicativo móvil 5: Pantalla de mensajería entre usuarios

Elaborado por: Obando B. - Vega C.

Esta pantalla será desplegada cuando un usuario sea seleccionado por el dueño del móvil, e iniciará un “chat” entre estos dos usuarios. Existe un área donde es posible escribir el mensaje, el botón de envío de mensajes y los mensajes ya existentes entre los dos usuarios aparecerán en la pantalla, identificándolos con la posición en la que se encuentran; los mensajes en la parte de la izquierda serán aquellos enviados por el dueño del móvil, y los mensajes en el lado derecho los mensajes enviados por el usuario con el cual se encuentra mensajando.

5.2. Estándares de codificación

Al momento de desarrollar cualquier aplicación, tener un estándar dentro de la codificación y todo el proyecto en sí, facilita varios aspectos del manejo de este como:

- **Orden en la jerarquía de paquetes y archivos:** El tener una forma fácil de acceder a los archivos que se necesiten, es una optimización del tiempo cuando se están manejando una gran cantidad de información dentro del proyecto.
- **Fácil entendimiento de código:** Que sea simple retomar una parte del código sin necesidad de recorrer todo su contenido, es una prioridad que los desarrolladores toman en cuenta al momento de escribir su código; a esto debemos incluir que otra persona pueda tener la capacidad de entender el código, si necesita trabajarlo o utilizarlo.
- **Creación de pruebas:** Cuando se desarrolla un proyecto, ningún código está libre de pruebas del tipo que lo necesiten, por este motivo la existencia de un estándar por ejemplo de nomenclatura ayudará a diseñar las pruebas correctas para el código.
- **Corrección de “bugs”:** Uno de los pasos principales en el ciclo de desarrollo, es la corrección de errores que se hayan encontrado en la transición de las pruebas, si se tiene un manejo correcto de archivos, nomenclatura y organización, para el desarrollador será sencillo encontrar en que sección de su código debe modificar para corregir un error.

- **Reutilización de código:** En muchos proyectos la reutilización del código es primordial para un ahorro de tiempo en desarrollos posteriores, si alguna sección del código es necesaria, manejando estándares será más sencillo identificarla y utilizarla.
- **Disminución de desperdicio:** Tener un código organizado disminuye el número de líneas de código, para lograr esto se siguen varios principios que permiten diseñar un proyecto lo más simple posible; entre estos podemos nombrar KISS, DRY, SOLID; estos principios enseñan la manera de cómo debemos organizar nuestro código al momento de escribirlo.

Después de conocer varias ventajas que nos proporciona el tener estándares de codificación dentro del proyecto, tomamos en cuenta el lenguaje que se va a utilizar, en este caso JAVA. Cada lenguaje tiene una convención de código que las podemos encontrar como documentos en el internet. (Google, Estándares de Codificación, 2011)

Estos estándares serán aplicados tanto en el desarrollo web como en el desarrollo dentro de la plataforma de Android.

5.2.1. Nomenclatura

Para esto debemos dividirlo en varios aspectos de la forma en como nombrar: métodos, clases, variables (estáticas y dinámicas), archivos, etc.

Tipo	Descripción
Clase	Toda clase será nombrada con UpperCase, quiere decir que será escrito todas las primeras letras de cada palabra que la defina con mayúscula, esto aplica también para los nombres de los archivos que la contengan (User, Message, etc.).
Método	Los métodos tendrán la nomenclatura camelCase, quiere decir que la primera palabra que lo defina estará escrita con minúscula y las posteriores serán escritas la primera letra con mayúscula y las demás con minúsculas; sin espacios en blanco o caracteres especiales, cada método debe tener una descripción de lo que el método realiza, si el método es solo de una palabra, esta será escrita en minúsculas. (sendMessage, processPacket, etc.).
Variable estática	Las variables estáticas tendrán la nomenclatura que todas las palabras deben encontrarse en mayúsculas y si su descripción contiene más de una palabra, serán separadas por guiones bajos “_” (GCM_SERVER, GCM_NAMESPACE, etc.).

Tipo	Descripción
Variable dinámica	Las variables dinámicas seguirán las mismas normas que utilizan los métodos, serán escritas en camelCase y su nombre tendrá una descripción de lo que contengan. (userAction, receiver, sender, userColor, etc.).
Bean	Los beans tendrán los mismos estándares que una clase, a diferencia que siempre tendrán la palabra “Bean” dentro del nombre de la clase y del archivo que la contenga (ChatBean, UserBean, etc.).
Converter	Los converters seguirán las mismas normas que las clases y los beans, contendrán la palabra “Converter” en el nombre de la clase y en el archivo que la contenga (UserConverter, RoleConverter, etc.).
Paquete	Los paquetes estarán escritos con todos los niveles que se recorren hasta llegar al necesario, se encontraran en minúsculas y cada nivel será separado por un punto “.” (src.main.java.com.map.services, src.main.java.com.map.entities, etc.).

Tipo	Descripción
Interface	Las interfaces seguirán los mismos estándares de las clases y los beans, quiere decir que contendrán la palabra “Interface” al final de su nombre. (GcmInterface).
HTML	Los archivos con extensión “html”, tendrán en su nombre la descripción de lo que contengan, si existe más de una palabra que lo describa, serán separadas con un guion “-” (chat.html, user.html, access-denied.html, etc.).
Servicios Web	Seguirán los mismos estándares que las clases y lo beans, con una descripción que realiza en el nombre del archivo, conteniendo al final de su nombre las siglas “RS” las cuales lo identifican como un “Resource” (RegisterAndLoginRS).

Tabla V- 1 Nomenclatura

Elaborado por: Obando B. - Vega C.

CAPITULO VI

Este contiene todos los elementos desarrollados dentro del prototipo desde la construcción de sus proyectos y los elementos más importantes dentro de ellos y hasta, las interfaces gráficas de ambos prototipos (web y móvil) y los servicios que se han integrado dentro de los proyectos.

También se incluyen las pruebas realizadas a cada una de las aplicaciones, el análisis de los resultados y la comparación con la plataforma seleccionada en el capítulo 3; obteniendo información que servirá para compararlas y obtener conclusiones de cual plataforma administra de una mejor manera el empaquetado a enviar por el protocolo TCP/IP.

6. Desarrollo del prototipo

6.1. Servicios Implementados

6.1.1. Google Cloud Messaging (GCM)

GCM es un servicio que ofrece Google libremente que permite enviar mensajes “push” entre servidor y aplicación del lado del cliente, lo realiza utilizando los protocolos XMPP (Extensible Messaging and Presence Protocol) y HTTP (Hypertext Transfer Protocol). (Google, Google Cloud Messaging, 2016)



Diagrama VI- 1 Arquitectura de comunicación GCM

Elaborado por: Obando B. - Vega C.

Basado en: (Google, Google Cloud Messaging, 2016)

Para tener un mejor entendimiento de cómo funciona GCM, tal como se indica en el Diagrama VI-1; los servidores de aplicaciones tienen implementado una conexión a través de los protocolos XMPP o HTTP, los cuales son utilizados para enviar un mensaje de tipo downstream o upstream al servidor de conexión GCM, este se encarga de gestionar en cola toda la información que llegue y enviarlos al aplicativo del lado del cliente, tomando en cuenta que éste se encuentra en un ambiente Android, Ios o en un navegador de internet o si la implementación del dispositivo trabaja con XMPP, es posible enviar mensajes normales o mensajes ACK del lado del cliente. (Google, Google Cloud Messaging, 2016)

Un mensaje ACK es una respuesta a una operación realizada a través de todo el proceso de GCM, si el cliente o el servidor reciben un mensaje, el receptor envía un mensaje ACK al emisor para hacerle conocer que la información fue gestionada con éxito o un mensaje NACK si la información no fue gestionada correctamente.

Al utilizar la plataforma de GCM en el desarrollo integramos la gestión de recepción y envío de mensajes entre cliente y servidor o viceversa, siempre y cuando los clientes se encuentren

registrados en el servidor. Para que los clientes se encuentren registrados, GCM maneja varios identificadores entre el aplicativo cliente y el aplicativo alojado en el servidor, estos dan a conocer al servidor de conexión de GCM la relación que existen entre ellos para que los mensajes sean transmitidos al receptor correcto.

- **Sender ID:** El proyecto que se desarrolle debe ser registrado dentro de los servicios de GCM, este será el identificador que permita enviar mensajes desde el lado del servidor al aplicativo del cliente.
- **Server Key:** Su función permite al aplicativo del lado del servidor tener acceso a los servicios de Google, en conjunto con el Sender ID. El Server Key funciona como una contraseña para autenticarse dentro de los servicios de Google, en este caso para utilizar GCM.
- **Application ID:** Es el identificador del lado del cliente, el cual es registrado de la misma forma que el Sender ID; es incluido dentro del proyecto del lado del cliente para poder recibir mensajes, es dependiente de la plataforma en la que se esté implementando.
- **Registration Token:** ID enviado por el servidor de conexión de GCM al aplicativo del lado del cliente para que tenga la capacidad de recibir mensajes.

Con estas credenciales es posible definir el proceso que se sigue entre los 3 elementos (servidor de aplicaciones, servidor de conexión de GCM y aplicativo del lado del cliente).

En el Diagrama VI-2 se explica de manera gráfica el proceso que se sigue, implementando los elementos mencionados anteriormente.

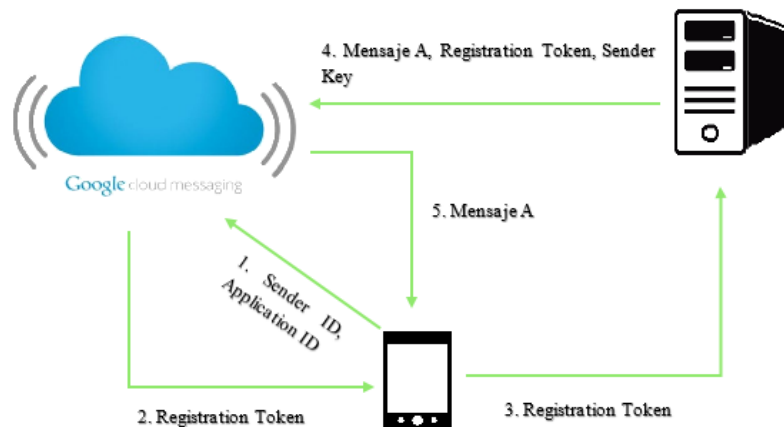


Diagrama VI- 2 Proceso de registro de dispositivos entre y mensajería

Elaborado por: Obando B. - Vega C.

Basado en: (Orero, 2016)

6.2. Desarrollo del código fuente

Estructura del proyecto del aplicativo Web

El desarrollo del prototipo se realizó con una estructura de paquetes manejable y de fácil acceso como se muestra en el Diagrama VI-3.

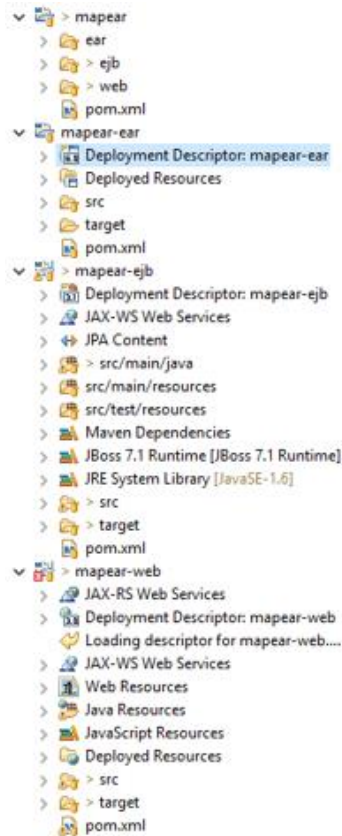


Diagrama VI- 3 Estructura de paquetes

Elaborado por: Obando B. - Vega C.

Donde se especifica que cada paquete se maneja de la siguiente manera:

mapear: Paquete encargado de albergar a los paquetes contenedores de los demás archivos, esta es la carpeta del proyecto en sí.

mapear-ear: Este directorio está encargado de almacenar los archivos de tipo .XML de configuración del despliegue del proyecto, además de almacenar las dependencias que sean necesarias para las funciones.

mapear-ejb: Dentro de este paquete encontramos todas las clases de tipo EJB, donde se tratan las operaciones de las diferentes entidades de la base de datos, las cuales fueron mapeadas con el framework Hibernate y la implementación de transacciones a través del lenguaje HQL, las operaciones pasan por la clase DAO para posterior comunicación con la base de datos.

La clase DAO contiene las instancias de las estructuras de las tablas generadas con Hibernate en forma de objetos del tipo necesario.

mapear-web: Este directorio contiene todas las vistas y los beans necesarios para la transacción entre el usuario y el aplicativo, para un posterior almacenamiento a través de los EJBs y los DAO de cada entidad.

6.2.1.1. Entidades

```
12 @Entity
13 @Table(name="message")
14 public class Message implements Serializable {
15     private static final long serialVersionUID = 1L;
16
17     @Id
18     @GeneratedValue(strategy=GenerationType.IDENTITY)
19     @Column(name="MES_ID")
20     private int mesId;
21
22     @Temporal(TemporalType.TIMESTAMP)
23     @Column(name="MES_RECEIVED_DATE")
24     private Date mesReceivedDate;
25
26     @Temporal(TemporalType.TIMESTAMP)
27     @Column(name="MES_SEND_DATE")
28     private Date mesSendDate;
29
30     @Column(name="MES_STATE")
31     private String mesState;
32
33     @Column(name="MES_TEXT")
34     private String mesText;
35
36     //bi-directional many-to-one association to User
37     @ManyToOne
38     @JoinColumn(name="USR_ID_RECEIVER")
39     private User user1;
40
41     //bi-directional many-to-one association to User
42     @ManyToOne
43     @JoinColumn(name="USR_ID_SENDER")
44     private User user2;
45
46     public Message(Date mesReceivedDate, Date mesSendDate, String mesState, String mesText, User user1, User user2) {
47         this.mesReceivedDate = mesReceivedDate;
48         this.mesSendDate = mesSendDate;
49         this.mesState = mesState;
50         this.mesText = mesText;
51         this.user1 = user1;
52         this.user2 = user2;
53     }
54 }
```

Diagrama VI- 4 Estructura de una entidad

Elaborado por: Obando B. - Vega C.

En el Diagrama VI-4 observamos cómo está estructurada una entidad mapeada con Hibernate, y tomamos en cuenta las anotaciones que se contiene cada variable:

- @Entity y @Table especifican que esta clase es tratada como una entidad y una tabla de la base de datos respectivamente. La anotación @Id señala que esta variable es la clave primaria de esta entidad y @GeneratedValue, que es generada automáticamente por la base de datos.
- @Column indica que la variable que se menciona posterior a la anotación es la columna de la base de datos y @JoinColumn indica que esta variable es adquirida de un cruce entre tablas.
- @Temporal debe ser utilizada si nuestro dato es de tipo tiempo o fecha, para indicarle a hibernate, específicamente de que tipo de variable se está manejando.
- @ManyToOne indica la relación que existe entre tablas cruzadas, si existiera alguna clave foránea.

6.2.1.2. EJB

Un EJB contiene todas las operaciones que se va a realizar en las transacciones con la base de datos, en el Diagrama VI-5 se muestra la estructura genérica de un EJB

```
@Stateless
@LocalBean
public class MessageEjb extends GenericDAOImpl<Message, Integer>{

    public MessageEjb(){

    }

}
```

Diagrama VI- 5 Estructura de un Ejb

Elaborado por: Obando B. - Vega C.

Donde es posible indicar los siguientes elementos:

- @Stateless es utilizado para indicar que la clase contiene operaciones independientes y no tiene un estado asociado a ella, y que es posible que esta clase mantenga su instancia.
- @LocalBean asigna a esta clase el estado de no estar asociada a ninguna interface existente.
- GenericDAOImpl<Object,Integer> es la clase generada a través de Hibernate que contiene las operaciones para la comunicación directa con la base de datos

6.2.1.3. Bean

Los Beans contienen las acciones que son posibles de realizar sin la intervención de la base de datos, es la sección donde prepararemos a los datos para ser enviados a un EJB y que sea almacenado u operaciones que no necesiten este tipo de transacción. El Diagrama VI-6 indica la estructura genérica de un Bean.

```

@ManagedBean(name = "chatBean")
@ViewScoped
public class ChatBean implements Serializable, GcmInterface, StanzalListener {

    private static final long serialVersionUID = -8990014463069759372L;
    public ChatBean(){

    }
}

```

Diagrama VI- 6 Estructura de un Bean

Elaborado por: Obando B. - Vega C.

Dentro de los Beans debemos considerar:

- @ManagedBean es una anotación que indica que esta clase es utilizada como expresiones de lenguaje dentro de las vistas de la aplicación, con esto permitimos que esta clase funcione como gestor de la información a través del front-end y el back-end.
- @ViewScoped indica que la instancia de esta clase seguirá existiendo mientras se realice transacciones en la página a la cual está asignada, si el usuario cambiara de vista, esta clase dejaría de existir.

6.2.1.4. Interfaz

Las interfaces, pantallas o vistas cumplen la función de permitir al usuario interactuar con el sistema, en el Diagrama VI-7 se muestra la estructura de una vista de Java Enterprise Edition.

```

<h:form id="chatForm" onsubmit="saveScrollPos()">
  <h:inputHidden id="scrollPos" />
  <h:panelGroup layout="block" class="ui" id="divisorChatWindow">
    <h:panelGroup layout="block" class="left-menu" id="divisorPeople">
      <p:selectOneListbox
        style="height:100% !important;width:300px !important"
        id="advanced" value="#{chatBean.userRole}"
        converter="userRoleConverter" var="t" filter="true"
        filterMatchMode="contains">
        <p:ajax listener="#{chatBean.onAppointmentTypeChange}" />
        <f:selectItems value="#{chatBean.userRoleList}" var="theme"
          itemLabel="#{theme.user.usrName} #{theme.user.usrLastname}" itemValue="#{theme}" />
        <p:column>
          <h:panelGroup class="avatar">
            <p:graphicImage value="#{imageBean.image}" height="50" width="75" style="border-color:#f1f3f1">
              <f:param name="personNickname" value="#{t.user.usrNickname}" />
            </p:graphicImage>
          </h:panelGroup>
        </p:column>
      </p:selectOneListbox>
    </h:panelGroup>
  </h:panelGroup>
</h:form>

```

Diagrama VI- 7 Estructura de una vista

Elaborado por: Obando B. - Vega C.

Esta estructura es realizada con Primefaces, por lo cual las anotaciones HTML son diferentes, existen elementos a considerar como los datos que se observan dentro de llaves y posterior a un “#”, esto se define como lenguaje de expresiones donde se cumple el ciclo de comunicación entre vista-managedBean-Ejb-DAO.

6.2.2. Desarrollo de módulos de la plataforma Móvil

6.2.2.1. Estructura del proyecto del aplicativo Móvil

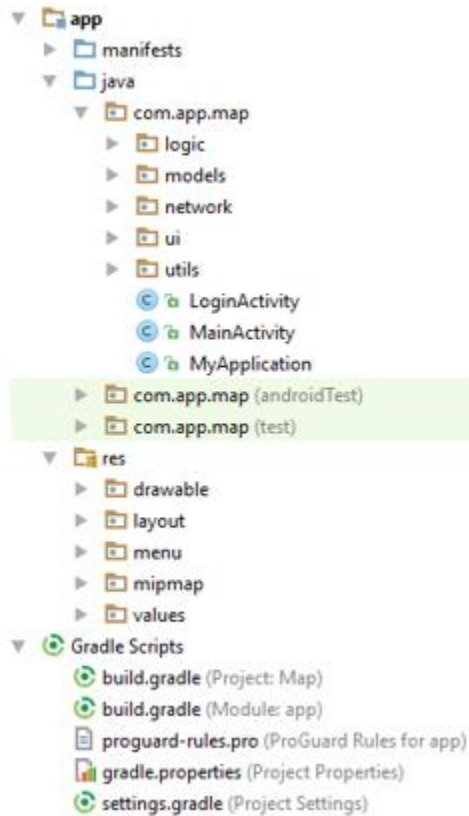


Diagrama VI- 8 Estructura de Paquetes

Elaborado por: Obando B. - Vega C.

Dentro del Diagrama VI-8 se define los directorios donde se encuentran todos los archivos para un correcto funcionamiento del aplicativo:

- **Manifest:** Se encuentra los archivos de configuración del proyecto de Android, el más importante, el “AdroidManifest.xml”, el cual contiene todos los activities, services y permisos de acceso que necesite solicitar la aplicación al equipo móvil.
- **Java:** Dentro de este se encuentra los archivos principales para el funcionamiento de la aplicación.

- **Logic:** Contiene los archivos donde se encuentran los CRUDs de las entidades del proyecto.
 - **Models:** Dentro encontramos los archivos de la capa del negocio.
 - **Network:** Aquí se definen las clases administradoras del servicio de GCM para poder utilizar la mensajería instantánea dentro de la aplicación.
 - **UI:** Aquí se almacenan los adapters y fragmentos de las vistas de la aplicación.
 - **Utils:** Se encuentran las clases donde es posible manejar la administración de sesiones, notificaciones y parámetros necesarios para el funcionamiento tanto de GCM como de otros componentes dentro de la aplicación.
- **res:** Dentro encontramos todos los componentes de las vistas de la aplicación, cada pantalla y los componentes que son parte de estas.

6.2.2.2. Activities

Un activity se define como la composición de un archivo .java y un archivo .xml, donde en el archivo .java se gestiona todas las acciones que se realizan sobre la parte gráfica de la aplicación y los comportamientos que debe tomar la misma y dentro del archivo .xml se encuentra el diseño de la parte gráfica perteneciente al activity.

6.2.2.3. Construcción de un activity

```
package com.app.map;

import ...

/**
 * A login screen that offers login via email/password.
 */
public class LoginActivity extends AppCompatActivity implements LoaderCallbacks<Cursor> {

    // UI references.
    private AutoCompleteTextView mEmailView;
    private EditText mPasswordView;
    private View mProgressView;
    private View mLoginFormView;
    private CallWebServices callWebServices;
    private BroadcastReceiver mRegistrationBroadcastReceiver;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_login);

        if(MyApplication.getInstance().getPrefManager().getUser() !=null){
            startActivity(new Intent(this, MainActivity.class));
            finish();
        }
    }
}
```

Diagrama VI- 9 Clase de un Activity

Elaborado por: Obando B. - Vega C.

Para considerarse una clase de un activity, debe extender un tipo de Activity o varios, como se observa en el Diagrama VI-9. Dependiendo del tipo de Activity importado, existen métodos que se importan de las interfaces, pero el método “onCreate” funciona para la creación del activity en sí.

La función setContentView hará referencia al layout con el cual tendrá interacción durante cualquier proceso realizado sobre este layout.

6.2.2.4. Interfaz gráfica de un activity

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:gravity="center_horizontal"
    android:orientation="vertical"
    android:paddingBottom="16dp"
    android:paddingLeft="16dp"
    android:paddingRight="16dp"
    android:paddingTop="16dp"
    tools:context="com.app.map.LoginActivity">

    <!-- Login progress -->
    <ProgressBar
        android:id="@+id/login_progress"
        style="?android:attr/progressBarStyleLarge"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginBottom="8dp"
        android:visibility="gone" />
```

Diagrama VI- 10 Estructura de una interfaz

Elaborado por: Obando B. - Vega C.

Toda interfaz realizada para Android, nativa, es estructurada dentro de un archivo .xml, aquí se definirán todos los componentes y propiedades de los mismos que se desean integrar a la pantalla; cada interfaz será ligada a un activity para el manejo del comportamiento de la pantalla. Esto es posible observar en el Diagrama V-10.

6.2.2.5. Consumo de servicios web

Es necesario definir que, para la adquisición de los usuarios y el ingreso del usuario a la aplicación, quienes se encuentran registrados en el sistema web fue necesario la implementación de servicios web, los cuales la aplicación móvil debe consumirlos para obtener y entregar la información necesaria para el funcionamiento.

```

HashMap<String, String> params = new HashMap<>();
params.put(Config.URL_REGISTER_EMAIL, email);
params.put(Config.URL_REGISTER_CODE, registerCode);
params.put(Config.URL_REGISTER_TOKEN, token);

JsonObjectRequest req = new JsonObjectRequest(url, new JSONObject(params),
(response) -> {
    try {
        JSONObject jsonResponse = response;
        if(!jsonResponse.getString("userName").equals("null")){
            User user = new User(jsonResponse.getString("userId"), jsonResponse.getString("userName"), jsonResponse.getString("usrLastname"),
                jsonResponse.getString("usrEmail"), jsonResponse.getString("usrNickname"));
            MyApplication.getInstance().getPrefManager().storeUser(user);

            Intent intent = new Intent(Config.LOGIN_RESPONSE);
            intent.putExtra("login", "true");
            LocalBroadcastManager.getInstance(context).sendBroadcast(intent);
        }else
            System.out.println("INGRESO FALLIDO");
    } catch (JSONException e) {
        System.out.println(e);
        e.printStackTrace();
    }
}, (error) -> { VolleyLog.e("Error: ", error.getMessage()); });

RequestQueue requestQueue = Volley.newRequestQueue(context);
requestQueue.add(req);

```

Diagrama VI- 11 Método para consumir el servicio web

Elaborado por: Obando B. - Vega C.

En el Diagrama VI-11 se encuentra en método necesario para la comunicación entre el equipo móvil y el servicio web implementado en la plataforma web.

6.2.2.6. Consumo del servicio GCM

Para realizar el consumo del servicio de GCM se debe seguir una serie de pasos:

1. Es necesario configurar dentro del archivo AndroidManifest.xml (archivo de configuración del proyecto) la creación de “services”, los cuales realizarán acciones dentro del plano secundario. Uno de estos servicios permitirá que los mensajes lleguen, aunque la aplicación no se encuentre abierta.
2. Otro “service” creado permite asignar un identificador único al equipo móvil, el cual permitirá comunicarse a través del servicio de GCM con la plataforma web, este

identificador tiene relación con los parámetros definidos dentro de la aplicación web sobre el servicio de GCM.

3. Crear un servicio que funcione como escuchador del canal de envío de paquetes al servicio de GCM, de esta manera si un mensaje es enviado desde otro dispositivo, este escuchador lo recibirá, solo si el identificador del paquete tiene como destino el equipo escuchador.
4. La creación de la clase que administra los mensajes a enviar hacia el servidor web, estos mensajes son conocidos como mensajes “Upstream”, los cuales contendrán diferentes parámetros que lo identificarán para que llegue al destinatario correcto y con la información que se desea incluir. Esto se observa en el Diagrama VI-12.

```
public GcmUpstream() { }

public void doGcmSendUpstreamMessage(Activity activity, String senderId, String msgId, String type, String message, String nickname)
    final GoogleCloudMessaging gcm = GoogleCloudMessaging.getInstance(activity);
    final String senderID = senderId;
    final String msgID = msgId;
    final String msgIdentificator = msgId;
    final Bundle data = new Bundle();
    SimpleDateFormat format = new SimpleDateFormat("yyyy-MM-dd hh:mm:ss");

    data.putString("message_type", type);
    data.putString("sentDate", format.format(Calendar.getInstance().getTime()));
    data.putString("messageText", message);
    data.putString("nickname", nickname);
```

Diagrama VI- 12 Mensaje Upstream

Elaborado por: Obando B. - Vega C.

5. La creación de la clase que administra los mensajes a enviar a un equipo móvil, estos mensajes a diferencia de los mensajes “Upstream”, son conocidos como mensajes

“Downstream”, tendrán de la misma forma que los mensajes “Upstream” parámetros identificadores para llegar a su destino. Esto se observa en el Diagrama VI-13.

```
public GcmDownstream() {}

public void doGcmSendDowstreamMessage(final Activity activity, final String senderId, final String registrationId, String msgId,

    final MessageGcmStructure.Builder messageBuilder = new MessageGcmStructure.Builder();
    //...
    messageBuilder.addData("Message", message);
    messageBuilder.addData("UserName", userName);
    messageBuilder.addData("UserLastname", userLastname);
    messageBuilder.addData("UserNickname", userNickname);
    messageBuilder.addData("Response", "true");
    messageBuilder.addData("Device", "MOBIL");
    messageBuilder.addData("content-available", "1");
    messageBuilder.addData("to", senderId);
    messageBuilder.addData("priority", "high");
    messageBuilder.notificationSound("default");
    messageBuilder.notificationBody(message);
    messageBuilder.notificationTitle(userName+" "+userLastname);
    messageBuilder.notificationTag("1");
```

Diagrama VI- 13 Mensaje Downstream

Elaborado por: Obando B. - Vega C.

6.3. Prototipo Final

6.3.1. Interfaces de la plataforma Web

6.3.1.1. Autenticación de usuario (WEB)

En el Diagrama VI-14, observamos la pantalla de autenticación de la plataforma web.

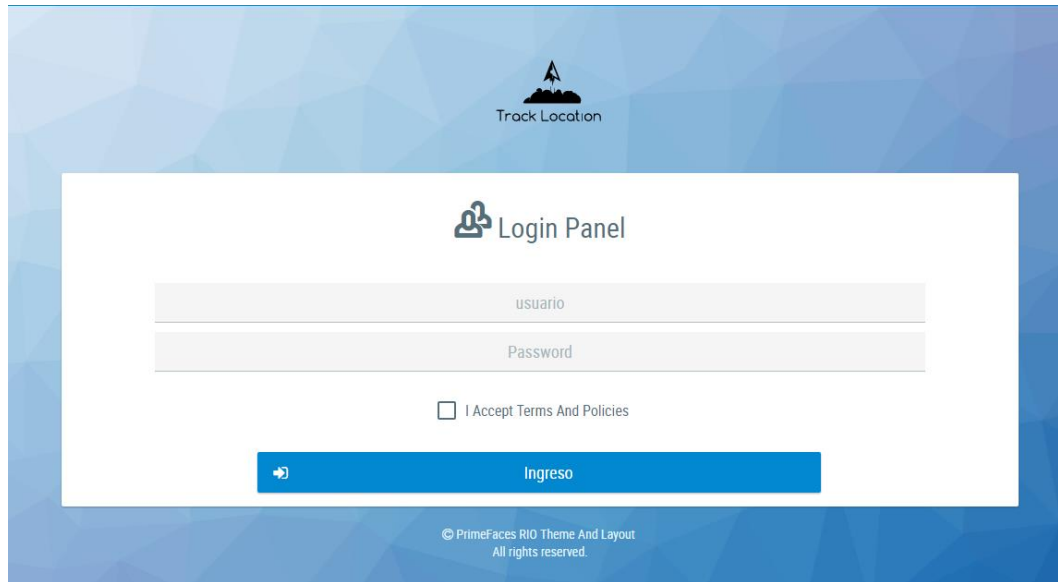


Diagrama VI- 14 Autenticación de usuarios

Elaborado por: Obando B. - Vega C.

6.3.1.2. Cambio de contraseña

En el Diagrama VI-15 se observa la pantalla de cambio de contraseña para la plataforma web.

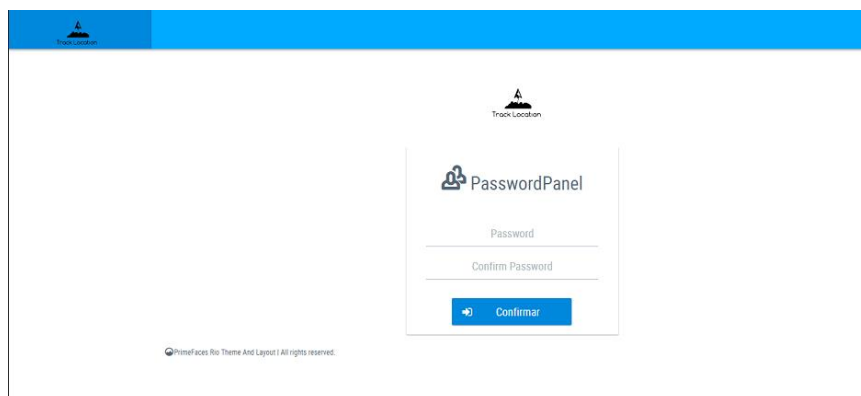
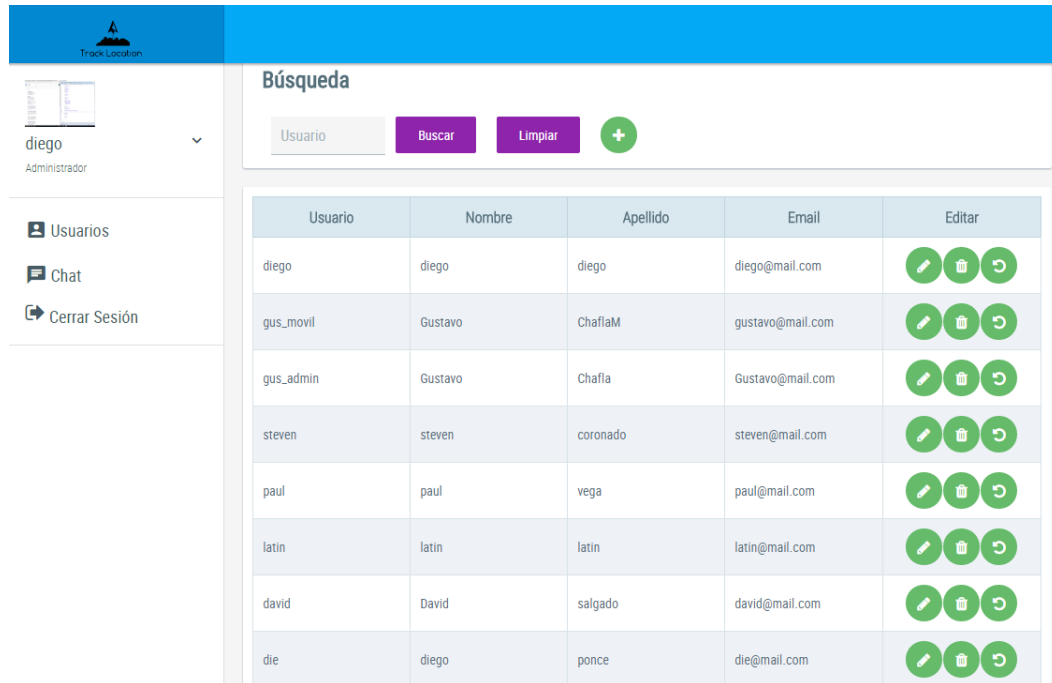


Diagrama VI- 15 Pantalla de Cambio de contraseña

Elaborado por: Obando B. - Vega C.

6.3.1.3. Administración de usuarios (WEB)

En el Diagrama VI-16 se encuentra como se ve la pantalla de la administración de usuarios con todas las acciones que se puede realizar sobre ellos.



























Usuario	Nombre	Apellido	Email	Editar
diego	diego	diego	diego@mail.com	  
gus_movil	Gustavo	ChafLaM	gustavo@mail.com	  
gus_admin	Gustavo	ChafLa	Gustavo@mail.com	  
steven	steven	coronado	steven@mail.com	  
paul	paul	vega	paul@mail.com	  
latin	latin	latin	latin@mail.com	  
david	David	salgado	david@mail.com	  
die	diego	ponce	die@mail.com	  

Diagrama VI- 16 Administración de usuarios

Elaborado por: Obando B. - Vega C.

6.3.1.4. Mensajería instantánea (WEB)

En el Diagrama VI-17 observamos la pantalla de mensajería instantánea la cual se utilizará para realizar las pruebas con la plataforma móvil.

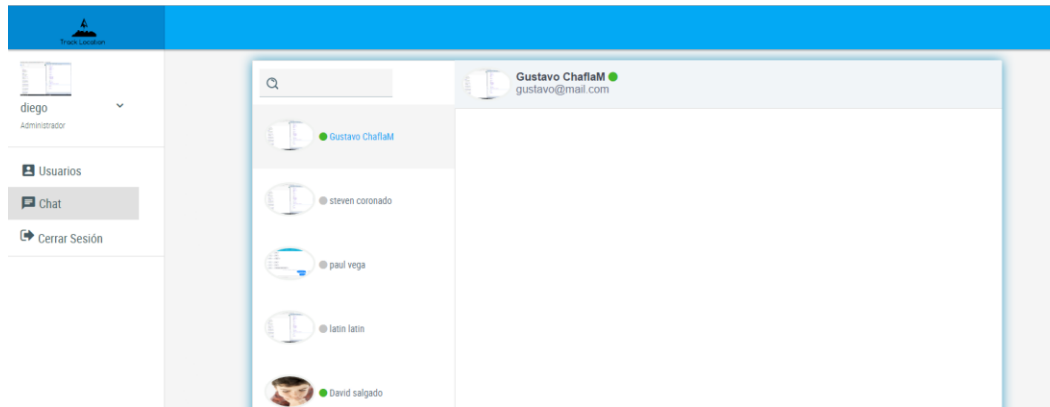


Diagrama VI- 17 Mensajería instantánea

Elaborado por: Obando B. - Vega C.

6.3.2. Interfaces de la plataforma Móvil

6.3.2.1. Autenticación de Usuario (Móvil)

En el Diagrama VI-18 se observa la pantalla de autenticación de usuarios de la plataforma móvil.

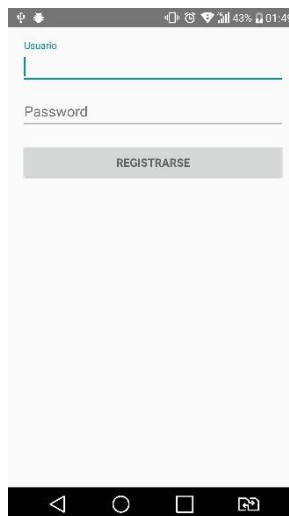


Diagrama VI- 18 Autenticación de usuario (Móvil)

Elaborado por: Obando B. - Vega C.

6.3.2.2. Pantalla de cambio de contraseña

En el Diagrama VI-19 se observa la pantalla de cambio de contraseña del usuario que ingresa al sistema.

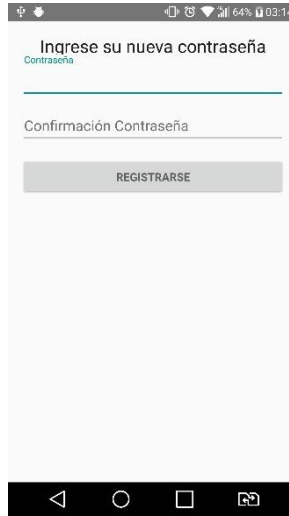


Diagrama VI- 19 Cambio de contraseña

Elaborado por: Obando B. - Vega C.

6.3.2.3. Menú de Navegación

En el Diagrama VI-20 se observa el menú de navegación de la aplicación, solo contiene la navegación a la mensajería instantánea que es parte de esta disertación.

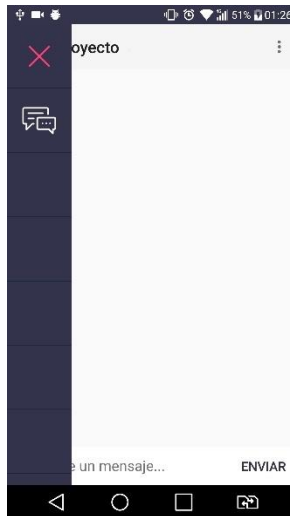


Diagrama VI- 20 Menú de navegación

Elaborado por: Obando B. - Vega C.

6.3.2.4. Pantalla de usuarios

En el Diagrama VI-21 se observa la pantalla de los usuarios con los que se puede contactar a través de la mensajería instantánea.

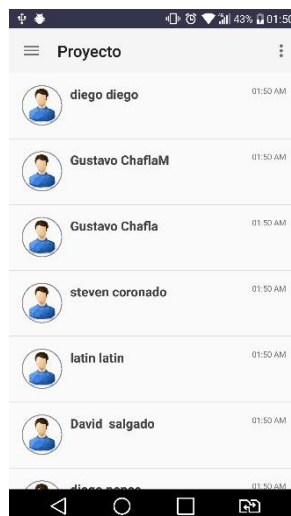


Diagrama VI- 21 Pantalla de usuarios

Elaborado por: Obando B. - Vega C.

6.3.2.5. Mensajería Instantánea

En el Diagrama VI-22 se observa la pantalla donde se despliegan los mensajes enviados entre dos usuarios.

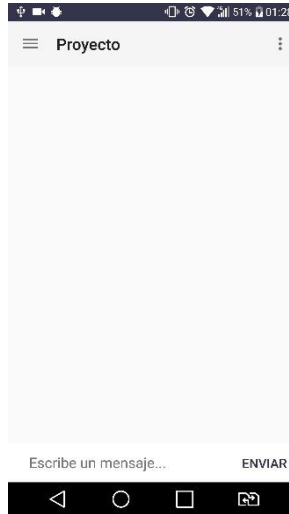


Diagrama VI- 22 Pantalla de mensajería

Elaborado por: Obando B. - Vega C.

6.4. Pruebas

Posterior al análisis de las aplicaciones de mensajería, la selección de la tecnología a implementar y el diseño del prototipo, se realizó una serie de pruebas entre las aplicaciones de mensajería elegidas en el capítulo III y el prototipo, para conocer su rendimiento se estableció un escenario de pruebas; hay que recalcar ciertos aspectos antes de analizar los resultados.

- Existen parámetros que solicita el servicio de GCM dentro del mensaje, estos vienen ya incluidos como valores por defecto de GCM y dentro del prototipo no fueron alterados.
- La implementación del prototipo fue para conocer el funcionamiento de la tecnología seleccionada y comparar los resultados entre aplicaciones de mensajería.
- Existen ciertos aspectos que no son visibles, por motivo que el creador de esta tecnología, no libera sus códigos fuente, lo codifica o encripta y no es posible realizar un análisis más profundo del funcionamiento.

Para obtener un resultado aceptable, se envió 100 mensajes dentro de cada prueba realizada y los resultados adquiridos fueron admisibles, por motivo que la variación en el consumo de banda de los mensajes no era notablemente alta.

6.4.1. Escenario de pruebas

Los parámetros que se utilizarán para realizar las pruebas son:

- El consumo de datos tanto en una red Wi-Fi como en una red móvil, esta puede ser LTE, HSPA+, 3G, etc.
- El mensaje contendrá un total de 50 caracteres, con un peso 0.05Kb, se analizará el consumo total de datos por mensaje mas no por número de caracteres.

El entorno para las pruebas Wi-Fi se realizará solo con dos dispositivos conectados (emisor y receptor), la velocidad de la red será superior a 1 MB, con respecto a la prueba en redes móviles

los dispositivos contarán con línea de un proveedor local y utilizarán la misma tecnología de conexión. Las aplicaciones estarán actualizadas a su última versión disponible en la fecha 07-07-2017. Los dispositivos a utilizar deberán contar con la versión de Android determinada en el Capítulo III, literal 3.2 Versión de Android (5.0 o superior). Para las versiones web de cada aplicación las pruebas se realizarán con un PC, navegador Google Chrome y el equipo estará conectado a una red Wi-Fi o Ethernet.

Los datos recopilados en las pruebas serán, el consumo de datos tanto del emisor como del receptor, tiempo (duración de las pruebas) y cantidad de mensajes. El tiempo de espera en el envío y recepción del mensaje no será considerado a menos que uno de los dispositivos pierda conexión ya que se trata de mensajería instantánea.

6.4.2. Pruebas red Wi-Fi.

En los cuadros mostrados a continuación se analiza el consumo de datos en una red Wi-Fi del emisor y receptor tanto de subida como de descarga, duración de las pruebas e IP de cada dispositivo. Los datos de las pruebas se encuentran en el capítulo VIII, sección de anexos.

6.4.2.1. Emisor

- **Teléfono:** Motorola Z Force (Las características técnicas del dispositivo se encuentran en el Capítulo 3).
- **IP:** La IP asignada por la red fue 192.168.100.8

- **Proceso:** Cada aplicación cuenta con un proceso específico para el sistema operativo el cual permite funcionamiento y transmisión de cada aplicación.
- **Duración:** Las pruebas tardaron un tiempo de 30 minutos por cada aplicación.

6.4.2.1.1. Consumo de datos emisor subida

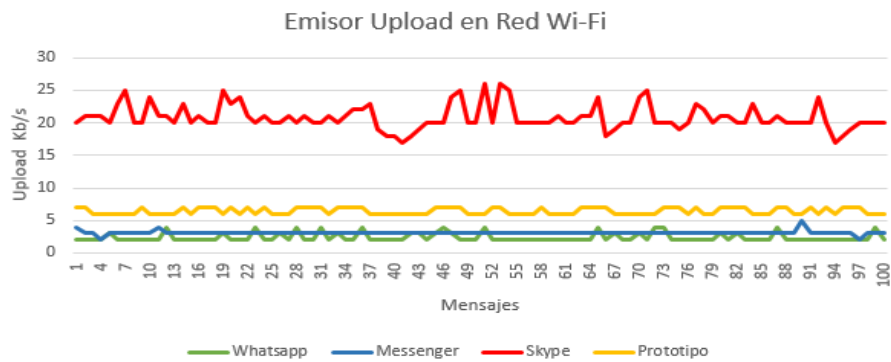


Diagrama VI- 23 Consumo Datos en Red Wi-Fi, Emisor Upload

Elaborado por: Obando B. - Vega C.

6.4.2.1.2. Consumo de datos emisor descarga

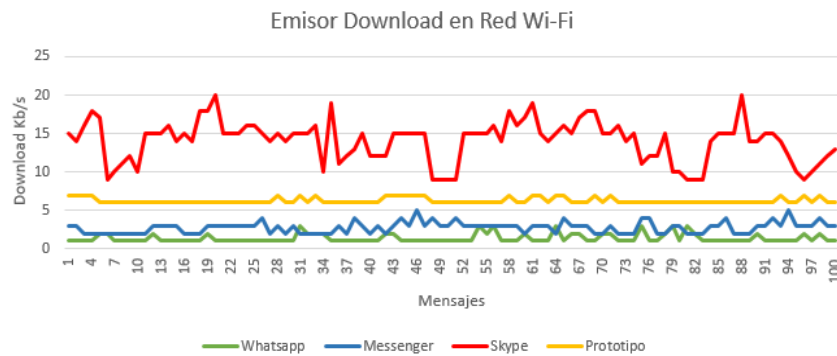


Diagrama VI- 24 Consumo Datos en Red Wi-Fi, Emisor Download

Elaborado por: Obando B. - Vega C.

6.4.2.1.3. Resultados Emisor

En la siguiente tabla se pueden observar cada uno de los resultados de las aplicaciones como emisor, teniendo como resultado según el menor consumo a WhatsApp con 370 Kb/s, Messenger con 580 Kb/s, Prototipo con 1267 Kb/s y Skype con 3485 Kb/s.

Emisor WiFi									
Aplicación	Whatsapp		Messenger		Skype		Prototipo		
Proceso	com.whatsapp		com.facebook.orca		com.skype.raider		com.map		
Valores en KB/s	Download	Upload	Download	Upload	Download	Upload	Download	Upload	
Promedio	1	2	3	3	14	21	6	6	
Mínimo	1	2	2	2	9	17	6	6	
Máximo	3	4	5	5	20	26	7	7	
Consumo	134	236	278	302	1403	2082	623	644	
Total	370		580		3485		1267		

Tabla VI- 1 Resultados Emisor red Wi- Fi.

Elaborado por: Obando B. - Vega C.

6.4.2.2. Receptor

- **Teléfono:** Samsung Galaxy S5 (Las características técnicas del dispositivo se encuentran en el Capítulo 3).
- **IP:** La IP asignada por la red fue 192.168.100.5
- **Proceso:** Cada aplicación cuenta con un proceso específico para el sistema operativo el cual permite funcionamiento y transmisión de cada aplicación.
- **Duración:** Las pruebas tardaron un tiempo de 30 minutos por cada aplicación.

6.4.2.2.1. Consumo de datos receptor subida

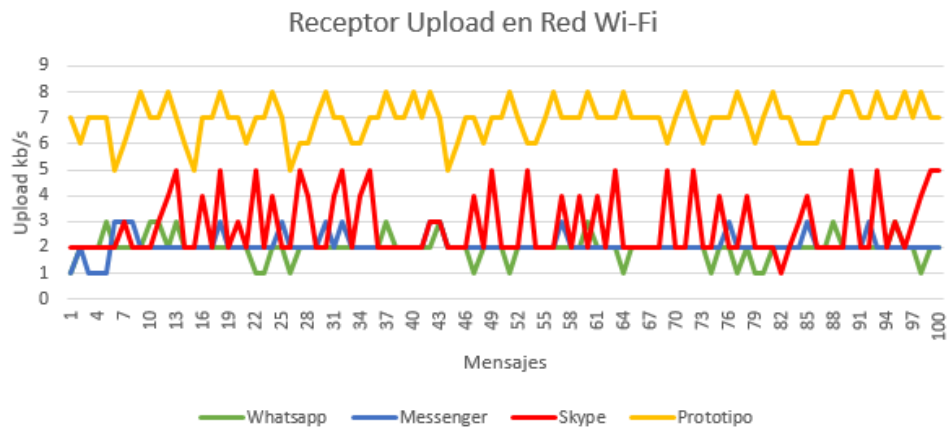


Diagrama VI- 25 Consumo Datos en Red Wi-Fi, Receptor Upload.

Elaborado por: Obando B. - Vega C.

6.4.2.2.2. Consumo de datos receptor descarga

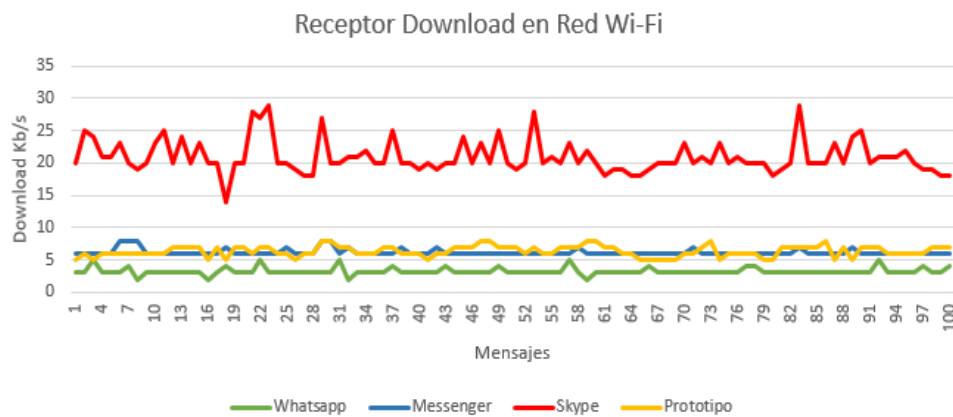


Diagrama VI- 26 Consumo Datos en Red Wi-Fi, Receptor Download.

Elaborado por: Obando B. - Vega C.

6.4.2.2.3. Resultados Receptor

En la siguiente tabla se pueden observar los resultados de las aplicaciones como receptoras, teniendo como resultado según el menor consumo a WhatsApp con 513 Kb/s, Messenger con 828 Kb/s, Prototipo con 1332 Kb/s y Skype con 2376 Kb/s.

Receptor WiFi								
Aplicación	Whatsapp		Messenger		Skype		Prototipo	
Proceso	com.whatsapp		com.facebook.orca		com.skype.raider		com.map	
Valores en KB/s	Download	Upload	Download	Upload	Download	Upload	Download	Upload
Promedio	3	2	6	2	21	3	6	7
Mínimo	2	1	6	1	14	1	5	5
Máximo	5	3	8	3	29	5	8	8
Consumo	316	197	619	209	2096	280	638	694
Total	513		828		2376		1332	

Tabla VI- 2 Resultados Receptor red Wi- Fi.

Elaborado por: Obando B. - Vega C.

6.4.3. Pruebas red Móvil.

En los cuadros que se presentan a continuación analizaremos el consumo de datos en una red Móvil la cual puede ser 4G, Hspa+, 3g, etc. (La tecnología utilizada dependerá de la disponibilidad en la conexión). Los datos recopilados serán del emisor y receptor tanto de subida como de descarga, duración de las pruebas e IP de cada dispositivo. Los datos de las pruebas se encuentran en el capítulo VIII, sección de anexos.

6.4.3.1. Emisor

- **Teléfono:** Motorola Z Force (Las características técnicas se encuentran en el Capítulo 3).
- **IP:** La IP asignada por la red fue 10.197.149.207
- **Proceso:** Cada aplicación cuenta con un proceso específico para el sistema operativo el cual permite funcionamiento y transmisión de cada aplicación.
- **Duración:** Las pruebas tardaron un tiempo de 30 minutos por cada aplicación.

6.4.3.1.1. Consumo de datos emisor subida

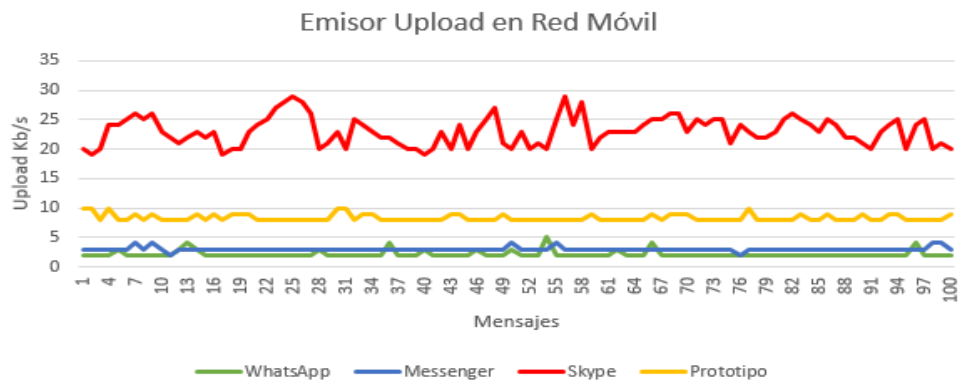


Diagrama VI- 27 Consumo Datos en Red Móvil, Emisor Upload.

Elaborado por: Obando B. - Vega C.

6.4.3.1.2. Consumo de datos emisor descarga.

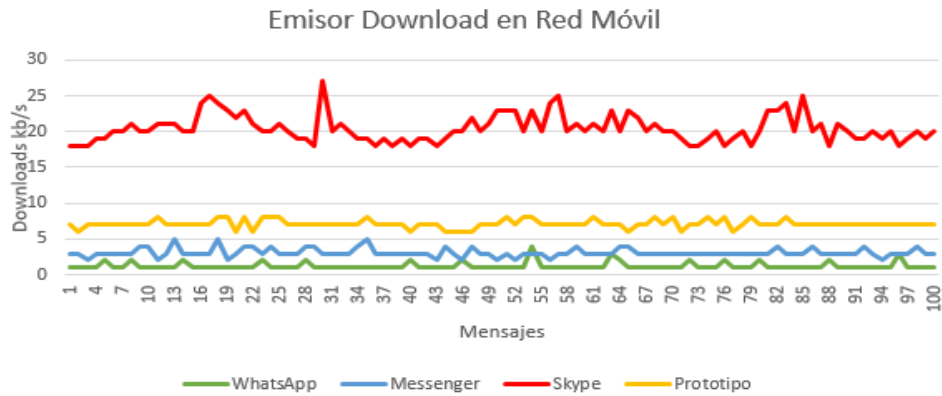


Diagrama VI- 28 Consumo Datos en Red Móvil, Emisor Download.

Elaborado por: Obando B. - Vega C.

6.4.3.1.3. Resultados Emisor

En la siguiente tabla se pueden observar los resultados de las aplicaciones como emisoras, teniendo como resultado según el menor consumo a WhatsApp con 338 Kb/s, Messenger con 618 Kb/s, Prototipo con 1542 Kb/s y Skype con 4343 Kb/s.

Emisor, red Móvil								
Aplicación	Whatsapp		Messenger		Skype		Prototipo	
Proceso	com.whatsapp		com.facebook.orca		com.skype.raider		com.map	
Valores en KB/s	Download	Upload	Download	Upload	Download	Upload	Download	Upload
Promedio	1	2	3	3	20	23	7	8
Mínimo	1	2	2	2	18	19	6	8
Máximo	4	5	5	4	27	29	8	10
Consumo	119	219	314	304	2040	2303	707	835
Total	338		618		4343		1542	

Tabla VI- 3 Resultados Emisor red Móvil.

Elaborado por: Obando B. - Vega C.

6.4.3.2. Receptor

- **Teléfono:** Samsung Galaxy S5 (Las características técnicas del dispositivo se encuentran en el Capítulo 3).
- **IP:** La IP asignada por la red fue 10.178.130.201
- **Proceso:** Cada aplicación cuenta con un proceso específico para el sistema operativo el cual permite funcionamiento y transmisión de cada aplicación.
- **Duración:** Las pruebas tardaron un tiempo de 30 minutos por cada aplicación.

6.4.3.2.1. Consumo de datos receptor subida.



Diagrama VI- 29 Consumo Datos en Red Móvil, Receptor Upload.

Elaborado por: Obando B. - Vega C.

6.4.3.2.2. Consumo de datos receptor descarga.

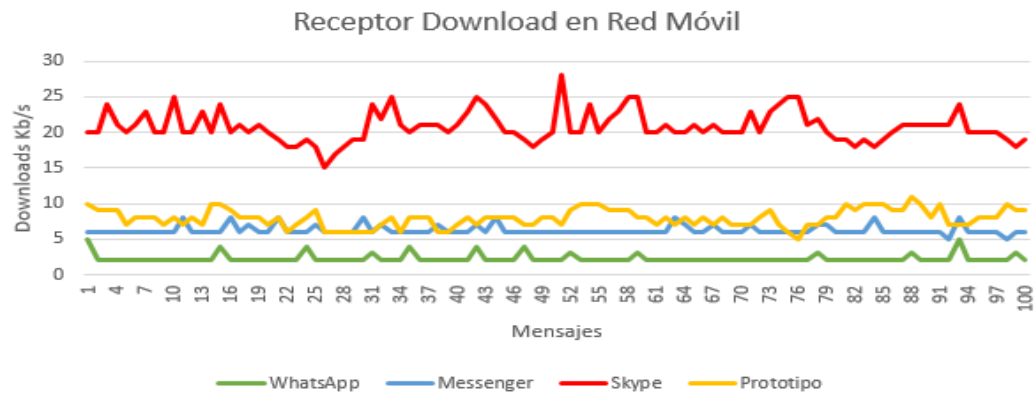


Diagrama VI- 30 Consumo Datos en Red Móvil, Receptor Download.

Elaborado por: Obando B. - Vega C.

6.4.3.2.3. Resultados Receptor.

En la siguiente tabla se pueden observar los resultados de las aplicaciones como receptoras, teniendo como resultado según el menor consumo a WhatsApp con 314 Kb/s, Messenger con 835 Kb/s, Prototipo con 1609 Kb/s y Skype con 2363 Kb/s.

Receptor, red Móvil								
Aplicación	Whatsapp		Messenger		Skype		Prototipo	
Proceso	com.whatsapp		com.facebook.orca		com.skype.raider		com.map	
Valores en KB/s	Download	Upload	Download	Upload	Download	Upload	Download	Upload
Promedio	2	1	6	2	21	3	8	8
Mínimo	2	1	5	1	15	1	5	6
Máximo	5	4	8	4	28	6	11	10
Consumo	222	125	624	211	2080	283	794	815
Total	314		835		2363		1609	

Tabla VI- 4 Resultados Receptor red Móvil.

Elaborado por: Obando B. - Vega C.

6.4.4. Pruebas Aplicación Web

En los cuadros mostrados a continuación se analiza el consumo de datos de los aplicativos en su versión web, para estas pruebas se utilizará un dispositivo móvil como emisor y el receptor será el ordenador. Los datos recopilados serán solo del receptor ya que el emisor al ser un dispositivo móvil ya ha sido analizado anteriormente. Los datos de las pruebas se encuentran en el capítulo VIII, sección de anexos.

6.4.4.1. Receptor

- **Dispositivo:** Ordenador marca HP. (Mismo utilizado para el desarrollo del prototipo)
- **IP:** La IP asignada por la red fue 192.168.1.16
- **Proceso:** Cada aplicación cuenta con un proceso específico para el sistema operativo el cual permite funcionamiento y transmisión de cada aplicación.
- **Duración:** Las pruebas tardaron un tiempo de 30 minutos por cada aplicación.

6.4.4.1.1. Consumo de datos receptor subida

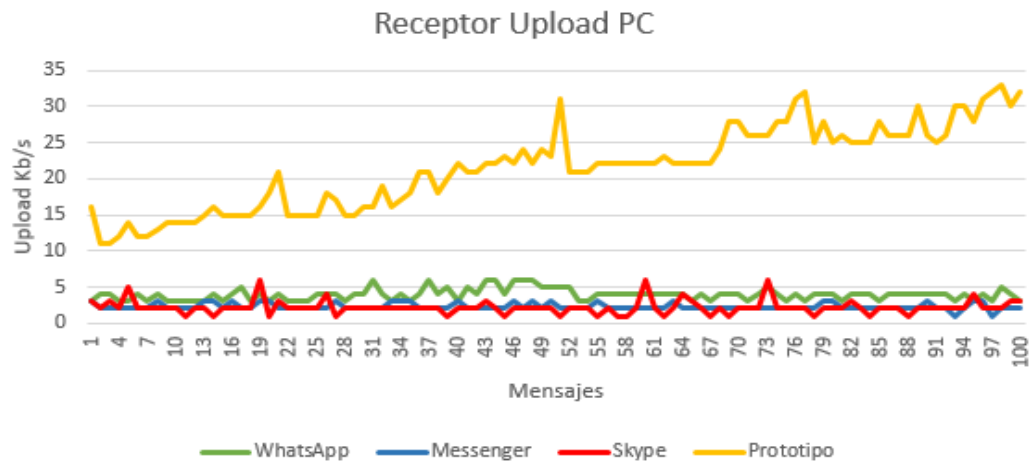


Diagrama VI- 31 Consumo Datos PC, Receptor Upload.

Elaborado por: Obando B. - Vega C.

6.4.4.1.2. Consumo de datos receptor descarga

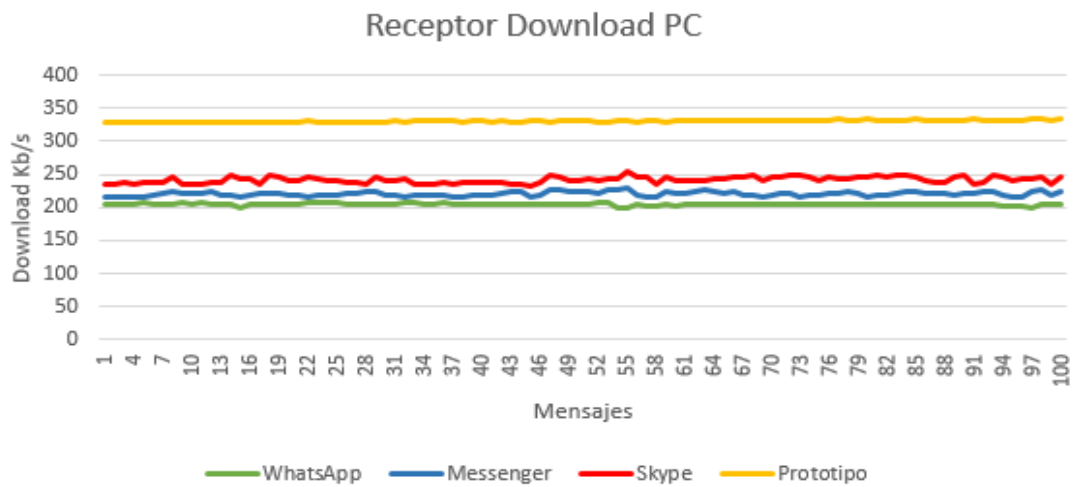


Diagrama VI- 32 Consumo Datos PC, Receptor Download.

Elaborado por: Obando B. - Vega C.

6.4.4.1.3. Resultados Receptor PC.

En la siguiente tabla se pueden observar los resultados de las versiones web en aplicaciones como receptoras, teniendo como resultado según el menor consumo a WhatsApp con 20934 Kb/s, Messenger con 22256 Kb/s, Skype con 24375 Kb/s y Prototipo con 35237 Kb/s.

Receptor, PC								
Aplicación	Whatsapp		Messenger		Skype		Prototipo	
Proceso	Google Chrome		Google Chrome		Skype 32 bits		Google Chrome	
Valores en KB/s	Download	Upload	Download	Upload	Download	Upload	Download	Upload
Promedio	205	4	220	2	242	2	331	22
Mínimo	200	3	215	1	231	1	328	11
Máximo	208	6	229	3	253	6	335	33
Consumo	20543	391	22036	220	24162	213	33085	2152
Total	20934		22256		24375		35237	

Tabla VI- 5 Resultados Receptor PC.

Elaborado por: Obando B. - Vega C.

6.5. Análisis Resultados en Redes Wi-Fi

Con los resultados recopilados anteriormente tanto como emisor y receptor se define lo siguiente:

- Se determina que el ancho de banda consumido para emitir y recibir mensajes es menor en aplicaciones con tecnología GCM.
- La estructura del paquete en Whatsapp, Messenger y Prototipo, que utilizan la misma tecnología GCM, están implementados de diferente manera y con un envío de contenido específico para cada aplicación generando que cada aplicación tenga un mayor o menor consumo.

6.6. Análisis Resultados en Redes Móviles.

Los resultados obtenidos en redes móviles son muy similares a los de una red Wi-Fi, la diferencia más radical es el uso de una diferente tecnología según la ubicación geográfica del emisor o receptor pudiendo variar entre tecnologías LTE, HSPA+, 3G. con esta consideración se define lo siguiente:

- Se observa que el paquete enviado tanto como emisor y receptor de la plataforma de Whatsapp, utilizando la misma tecnología implementada en Facebook Messenger y el Prototipo en cualquiera de las pruebas, consume una menor cantidad de datos, lo que indica que la construcción del paquete de Whatsapp está estructurada de una manera más eficiente para un consumo reducido de la banda móvil.
- Las pruebas fueron realizadas bajo un ambiente de conexión 4G, lo cual determina una mayor velocidad de transferencia de información, por lo cual si el paquete es de mayor tamaño el equipo utilizará una mayor banda para descargarlo o enviarlo, esto respalda los datos adquiridos en la Tabla VI-5, como el paquete enviado por Whatsapp es de menor tamaño, el consumo de banda será menor al consumo del resto de aplicaciones las cuales contienen un paquete de mayor tamaño.
- Se concluye que la estructura interna del mensaje y los parámetros ingresados de GCM afectan al peso del paquete enviado, esto se respalda con la variedad de posibilidades de construcción de este paquete desde el lado de GCM y como este interpreta la información para enviarlo a su destino.

6.7. Análisis Resultados en Aplicaciones Web

El resultado en las respectivas versiones Web y de escritorio de cada aplicación nos permite determinar que el consumo de datos es mucho mayor ya que al estar conectado a una red sin limitaciones, no solo se envía o recibe el mensaje, sino también la aplicación envía otro tipo de información que lamentablemente no es posible analizarla por encontrarse encriptada, y con un proceso en común siendo el navegador web en tres aplicaciones (Whatsapp, Messenger, Prototipo), Skype cuenta con versión de escritorio el consumo de datos fue mucho mayor.

6.7.1. Análisis final entre aplicaciones

Después de realizar una serie de pruebas que demuestren el consumo de ancho de banda tanto en el aplicativo de Whatsapp, Messenger, Skype y Prototipo se determinó que la aplicación más eficiente es Whatsapp, su gestión del empaquetado es óptima para un menor consumo de ancho de banda, lastimosamente no es posible analizar los motivos exactos del por qué consume una cantidad considerablemente menor ya que la empresa dueña de Whatsapp no libera sus algoritmos para empaquetado. Pero tomando como punto de referencia el desarrollo del prototipo y observando los resultados adquiridos y comparándolos con los resultados de Whatsapp es posible suponer los siguientes aspectos:

- Dentro del prototipo no se utilizó ninguna forma de compresión, codificación o encriptación de la información dentro del paquete que logrará reducir el tamaño del

mensaje o de los parámetros, lo cual esto podría funcionar como optimizador del tamaño del paquete.

- En la construcción del mensaje del prototipo se incluye información que es utilizada por nuestra parte para identificar ciertos parámetros del usuario, lo cual muy posiblemente existan otras maneras más eficaces de identificarlo.
- La estructura del mensaje de respuesta “ACK” que se envía al recibir un mensaje, tiene la estructura igualmente sencilla y con los parámetros por defecto, por lo cual eliminando ciertos parámetros que no se necesiten o utilizando métodos mencionados anteriormente para reducir su tamaño, existiría un menor consumo de ancho de banda.
- Una desventaja encontrada en ambos aspectos tanto del prototipo y de la plataforma Whatsapp, es la captura del paquete dentro de la red; por ser una tecnología bastante segura no es posible capturarlo a través de herramientas que monitoreen el tráfico de red y determinar de que manera la información está enviada.

CAPITULO VII

En este capítulo se define las conclusiones y recomendaciones obtenidas a través de todo el proceso de realización del trabajo de disertación.

7. Conclusiones y Recomendaciones

7.1. Conclusiones

- La gran cantidad de personas que utilizan la mensajería instantánea diariamente ha llevado a varias organizaciones a diseñar una variedad de aplicaciones que cumplan con esta tarea, incluyendo en ellas diferentes formas de comunicación, además con la existencia de la internet y los protocolos de transferencia de datos se facilitó el intercambio de información desde cualquier parte del mundo a través de paquetes estandarizados para un mejor manejo de los datos. Cada una de estas aplicaciones tiene su propia manera de administrar estos datos, utilizando protocolos compartidos entre aplicaciones ya que estos, fueron diseñados para cumplir con esta tarea.
- Cada aplicación tiene sus ventajas y desventajas frente a las demás, esto es posible definirlo en base a indicadores que muestran el desempeño de la aplicación en estudio, pero cada una de estas aplicaciones tiene el principal objetivo; que la información llegue exitosamente a su destino, en corto tiempo y que no existan pérdidas de datos en el proceso de transmisión.

- Para la transmisión de datos existen diferentes maneras de realizarlo, como la utilización de servicios web diseñados por las mismas organizaciones desarrolladoras del aplicativo o APIs creados por otras empresas y que brindan el servicio de envío de información como paquetes utilizando sus tecnologías y herramientas. Dependiendo de la tecnología que se utilice ya sea un servicio web o un API de terceros, ofrece al propietario del aplicativo ventajas frente a otras; como es el control absoluto de la administración del paquete enviado, si este es transferido a través de servicios web propios de la organización o la facilidad de implementar estándares ya establecidos por los APIs de terceros. De la misma forma existen desventajas ya que si la aplicación utiliza un servicio de terceros es dependiente que los servidores de estas organizaciones se encuentren en funcionamiento para utilizar su servicio, mientras que la organización propietaria tiene control frente a esta desventaja.
- Los procedimientos de envío de mensajería se rigen a que la información sea enviada de forma eficiente y sin fallos, por lo cual el control que se tiene es exhaustivo para cumplir esta meta; y es observable que muchas aplicaciones prefieren el consumir un servicio realizado por otra empresa, por razones que estas organizaciones enfocaron su completo esfuerzo a cumplir con estos parámetros, un ejemplo a considerar es Google y su servicio llamado GCM (Google Cloud Messaging), el cual tiene implementado estándares en el envío de los paquetes de información y en el proceso de transferencia de datos a través

de los protocolos XMPP y HTTP fáciles de utilizar por quien desee consumir su servicio.

- El intercambio de información, en este caso específico los mensajes de texto a través del protocolo IP tienen como necesidad la existencia de una conexión a internet a través de una red WI-FI o paquetes de datos ofrecidos por las empresas de telefonía para su correcto funcionamiento, la falta de este factor no permite la comunicación entre dispositivos y la transferencia de datos no podrá ser realizada.
- Los resultados adquiridos de la comparación de los valores finales de las pruebas, determinan que la plataforma seleccionada como la más óptima, tiene una gestión de su paquete a enviar a través del protocolo TCP/IP bastante adecuada para cualquier ámbito, sea este web o móvil, quiere decir que si se utiliza la aplicación que no está enfocada al ambiente utilizado (utilizar web en móvil) existirá un mayor consumo de ancho de banda, que el utilizado por la aplicación diseñada para ese ámbito específico.

7.2. Recomendaciones

- Al seleccionar la tecnología y herramienta para realizar una transferencia de datos a través de un medio es imprescindible conocer cómo se desea enviar esta información y hasta donde se quiere llegar, por lo que se recomienda identificar

que necesidades se tiene al momento de transferir un paquete de datos, los recursos necesarios para implementar este proceso tomando en cuenta los equipos donde se va a implantar la aplicación y el área que debe cubrir este servicio.

- Por la necesidad que exista una conexión a internet y el alto consumo de este servicio tanto por organizaciones o personas naturales, es recomendable que diferentes organizaciones tanto gubernamentales como privadas, ofrezcan este servicio sea por un costo o de forma gratuita y con una velocidad adecuada para garantizar que la transferencia de información se realice exitosamente
- Dentro de la implementación de un aplicativo de mensajería instantánea es posible integrar una variedad de servicios que se acoplen a la transferencia de paquetes, como es el envío de imágenes, VoIP, entre otras; por lo cual es recomendable que en base a las necesidades por las cuales se desarrolle una de estas aplicaciones se integre diferentes servicios, si son requeridos.

CAPITULO VIII

En este capítulo se adjunta toda información requerida para realizar los análisis y gráficas de los capítulos anteriores, enfocado en la selección de la aplicación a través de la encuesta y las tablas de información de las pruebas realizadas.

8. Anexos

8.1. Resultados Encuesta

Los resultados de la encuesta tuvieron como objetivo determinar las aplicaciones de mensajería, plataformas, sistema operativos y tecnologías de mensajería instantánea.

- ¿Cuenta con un computador?



Diagrama VIII- 1 Primera pregunta encuesta.

Elaborado por: Obando B. - Vega C.

- Seleccione el tipo de ordenador que posee.

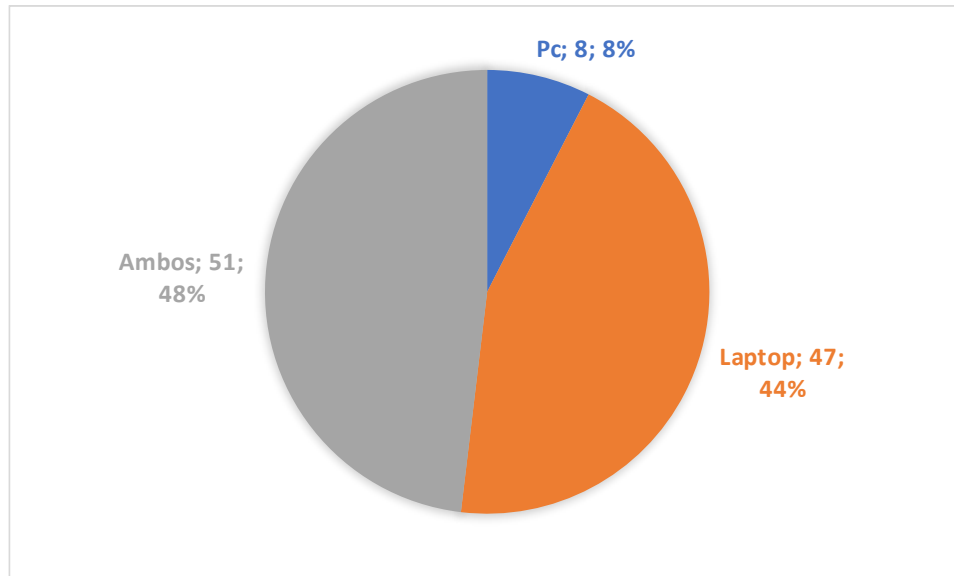


Diagrama VIII- 2 Segunda pregunta encuesta.

Elaborado por: Obando B. - Vega C.

- ¿Cuál es el sistema operativo de su ordenador?

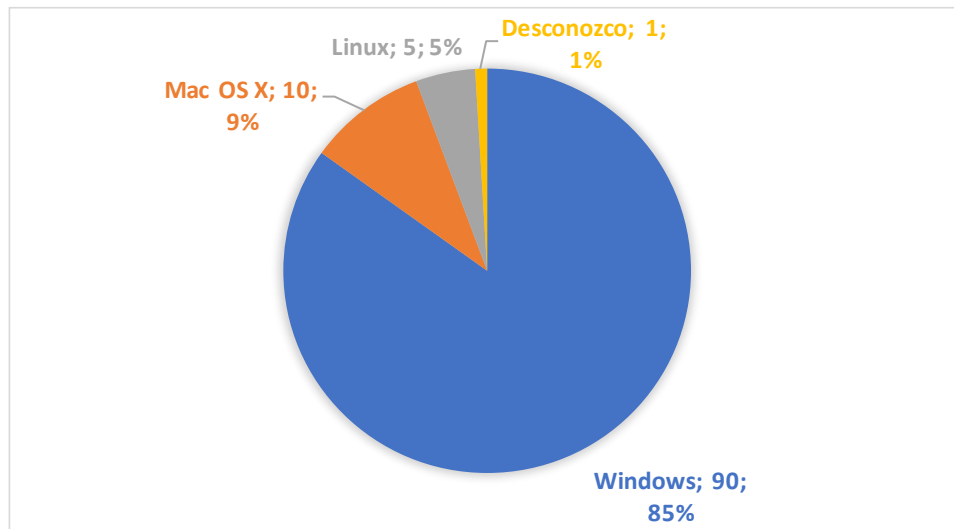


Diagrama VIII- 3 Tercera pregunta encuesta.

Elaborado por: Obando B. - Vega C.

- ¿Cuál es la versión de su Sistema Operativo Windows?
(Si elije Windows)

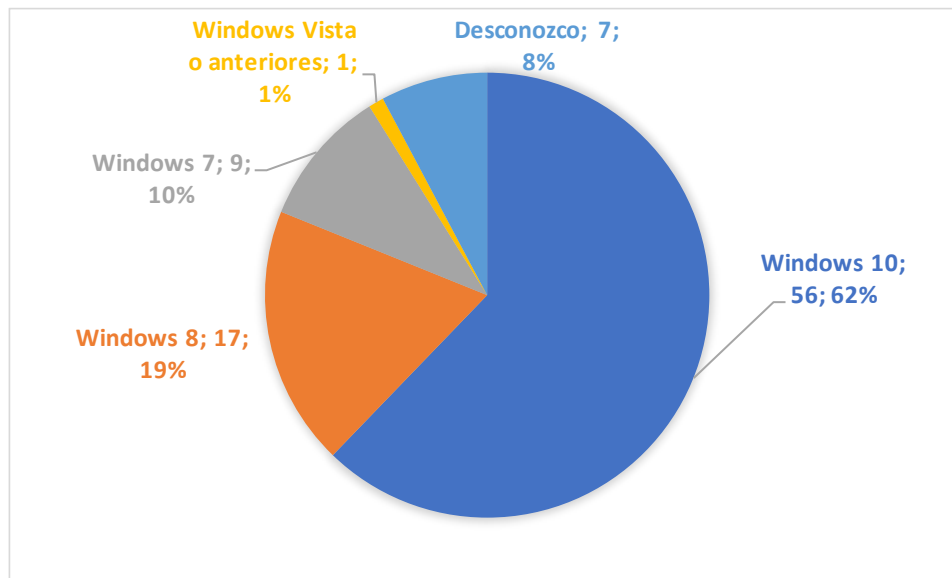


Diagrama VIII- 4 Cuarta pregunta encuesta.

Elaborado por: Obando B. - Vega C.

- ¿Cuál es la versión de su Sistema Operativo Mac OS X?
(Si elije Mak OS)

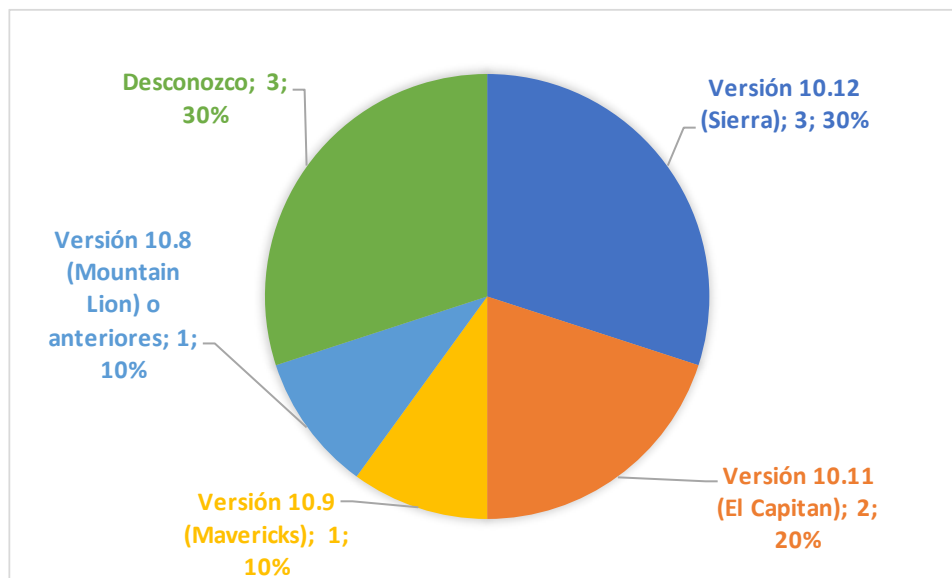


Diagrama VIII- 5 Quinta pregunta encuesta.

Elaborado por: Obando B. - Vega C.

- ¿Cuál es la distribución de su Sistema Operativo Linux ?
(Si Elije Linux)

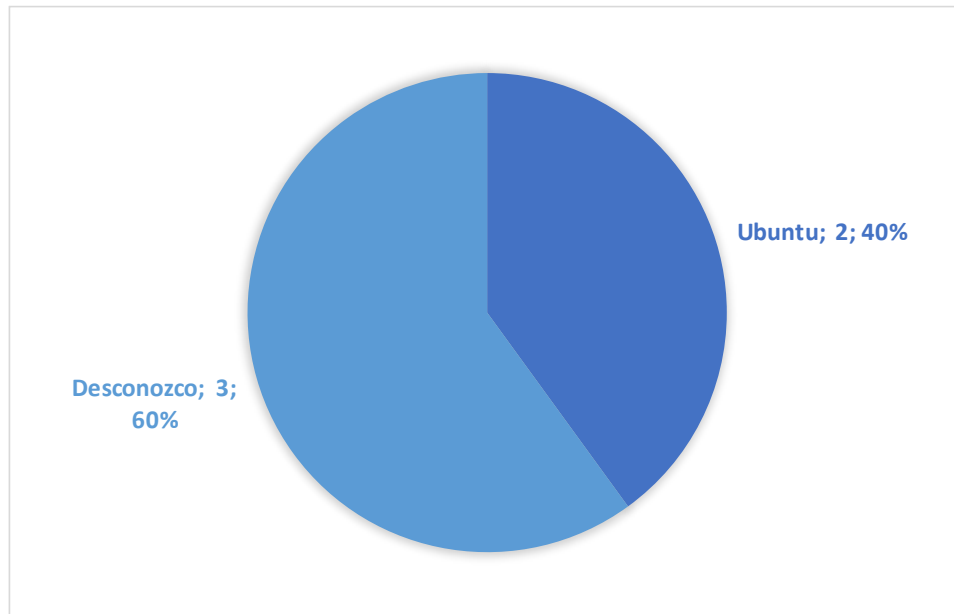


Diagrama VIII- 6 Sexta pregunta encuesta.

Elaborado por: Obando B. - Vega C.

- ¿Cual es su navegador web de preferencia?

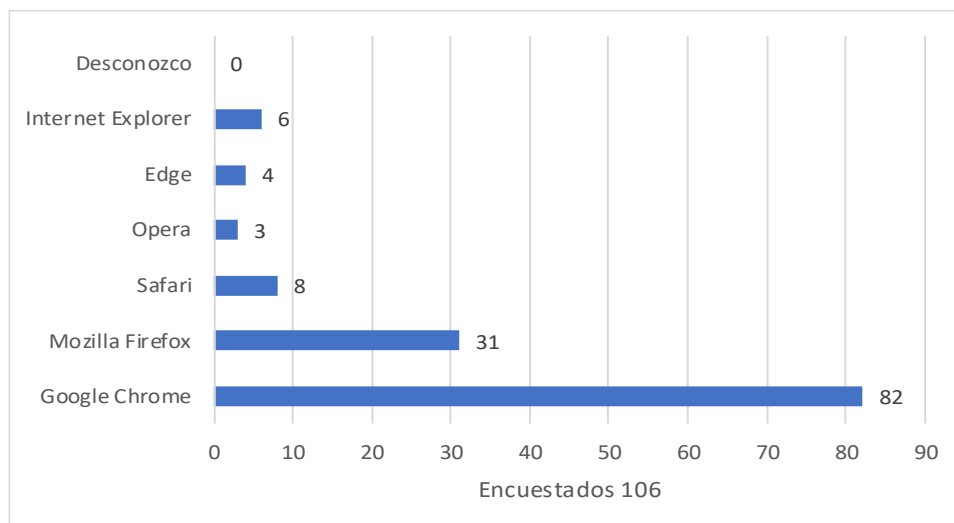


Diagrama VIII- 7 Séptima pregunta encuesta.

Elaborado por: Obando B. - Vega C.

- ¿Cuenta con uno o varios dispositivos móviles?

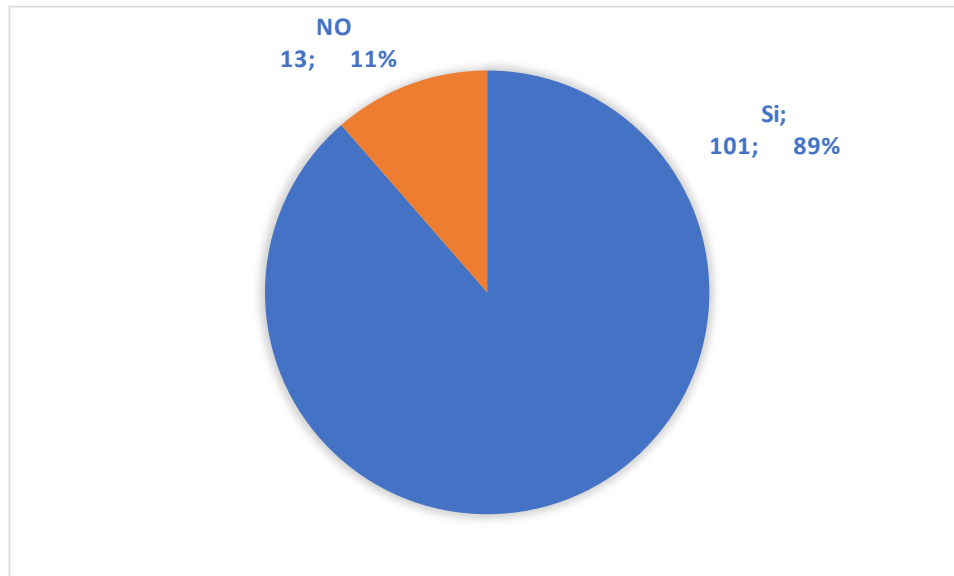


Diagrama VIII- 8 Octava pregunta encuesta.

Elaborado por: Obando B. - Vega C.

- Seleccione el tipo de dispositivos que posee.

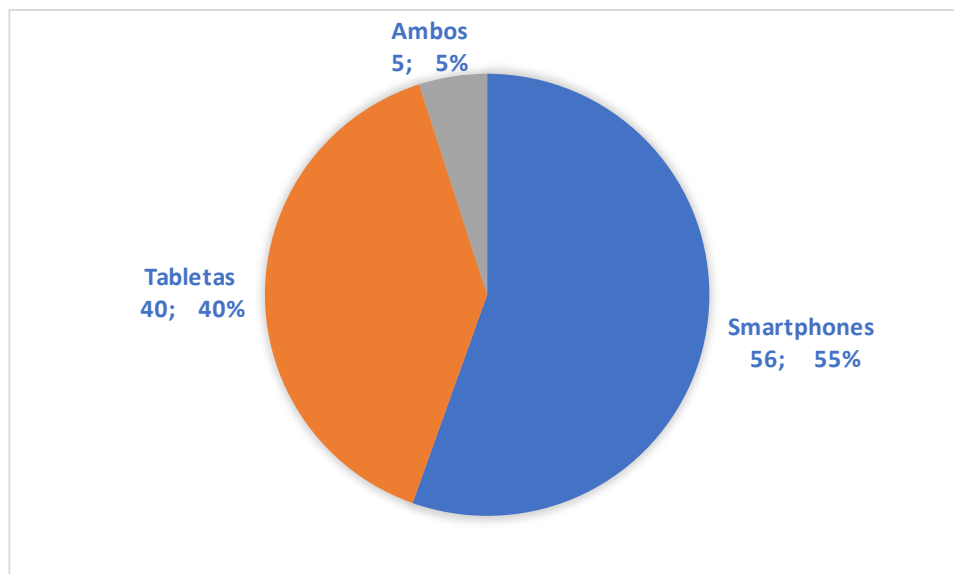


Diagrama VIII- 9 Novena pregunta encuesta.

Elaborado por: Obando B. - Vega C.

- ¿Cual es la marca de su dispositivo personal mas utilizado?

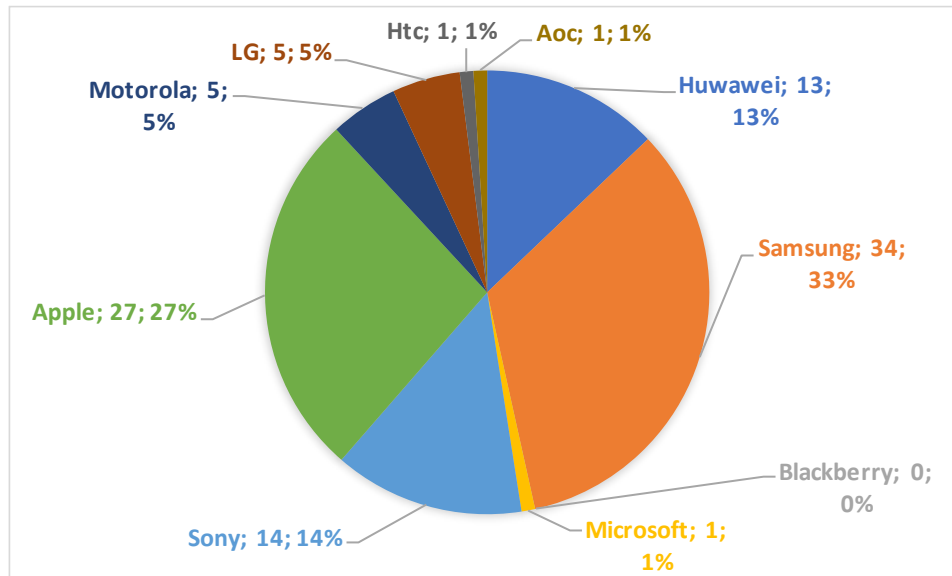


Diagrama VIII- 10 Decima pregunta encuesta.

Elaborado por: Obando B. - Vega C.

- ¿Cuál es la versión de android en su dispositivo?

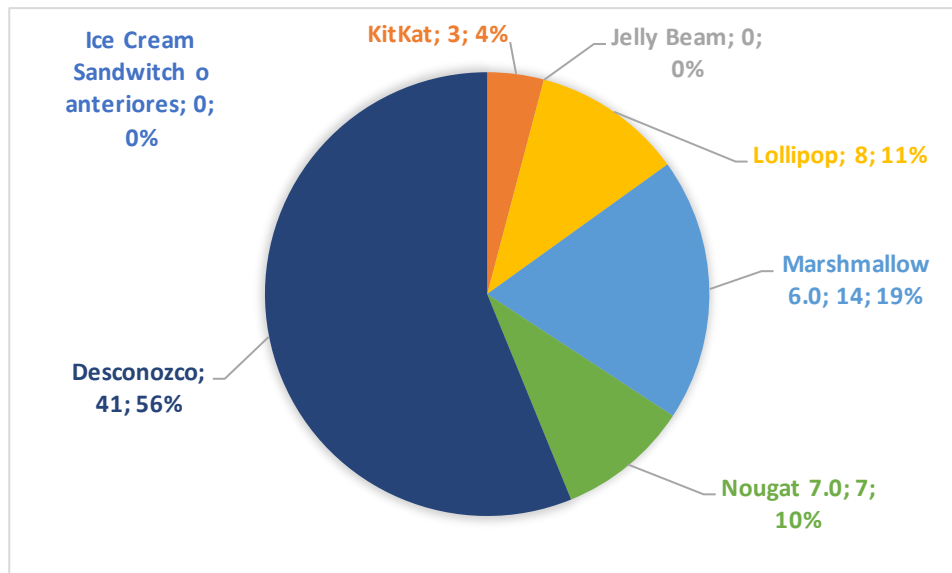


Diagrama VIII- 11 Décimo primera pregunta encuesta.

Elaborado por: Obando B. - Vega C.

- ¿Cuál es la versión de Windows en su dispositivo?
(Si elije microsoft)

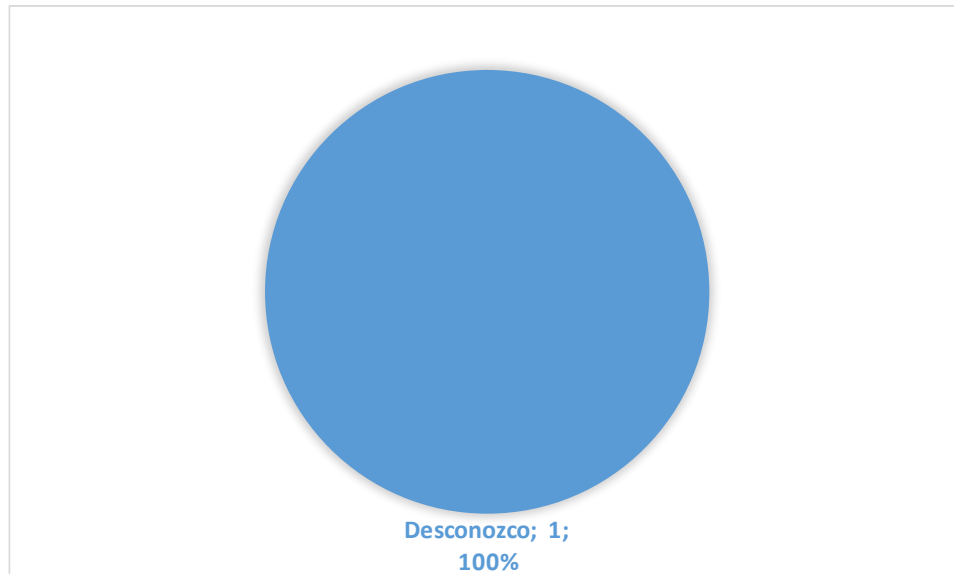


Diagrama VIII- 12 Décimo segunda pregunta encuesta.

Elaborado por: Obando B. - Vega C.

- ¿Cuál es la versión de iOS en su dispositivo?

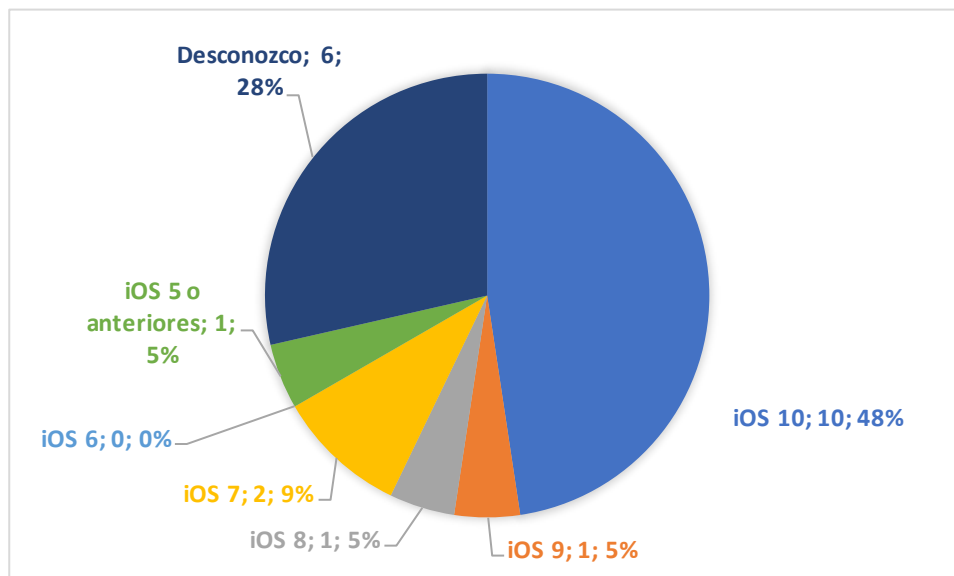


Diagrama VIII- 13 Décimo tercera pregunta encuesta.

Elaborado por: Obando B. - Vega C.

- ¿Cuál es el tipo de conexión de su dispositivo?

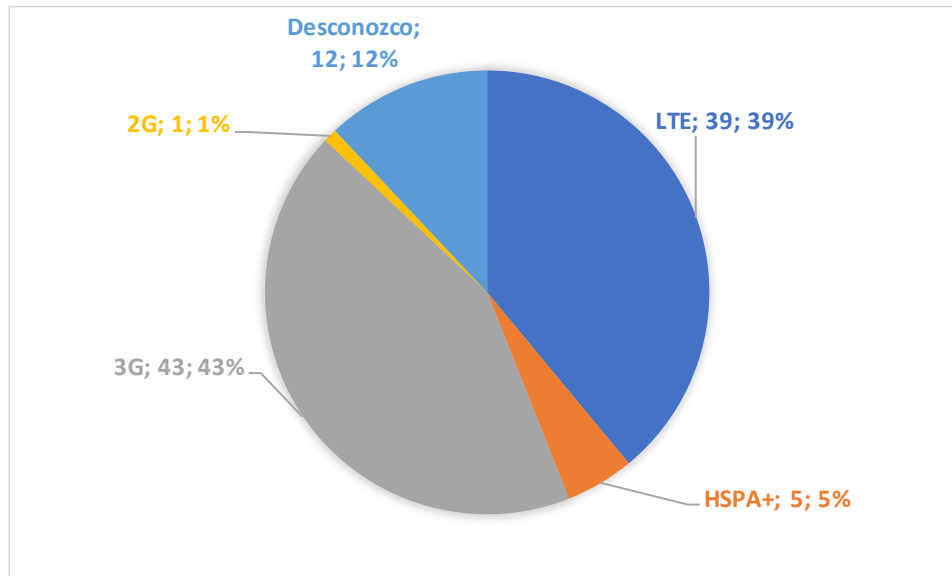


Diagrama VIII- 14 Décimo cuarta pregunta encuesta.

Elaborado por: Obando B. - Vega C.

- ¿Cuales aplicaciones de mensajería utiliza?

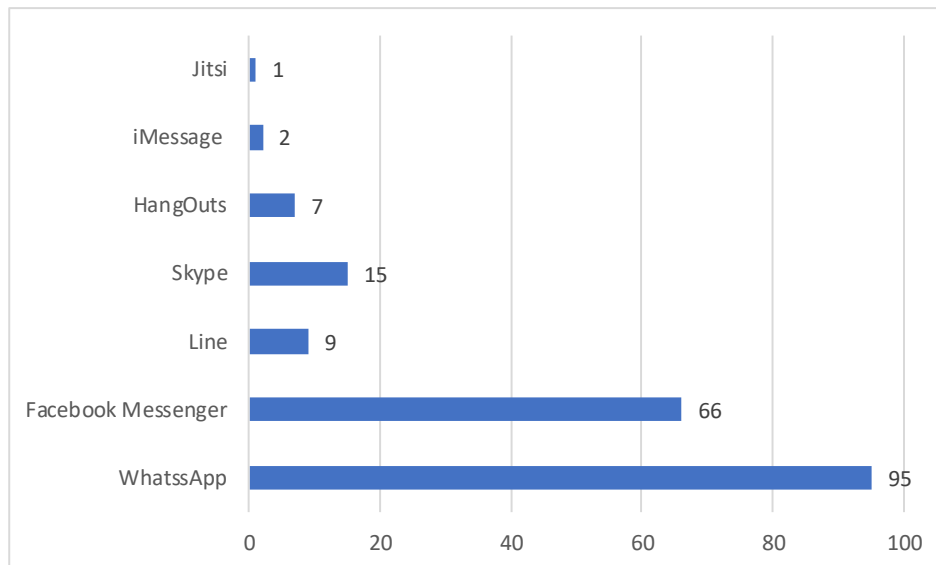


Diagrama VIII- 15 Décimo quinta pregunta encuesta.

Elaborado por: Obando B. - Vega C.

- Califique el servicio de estas aplicaciones.

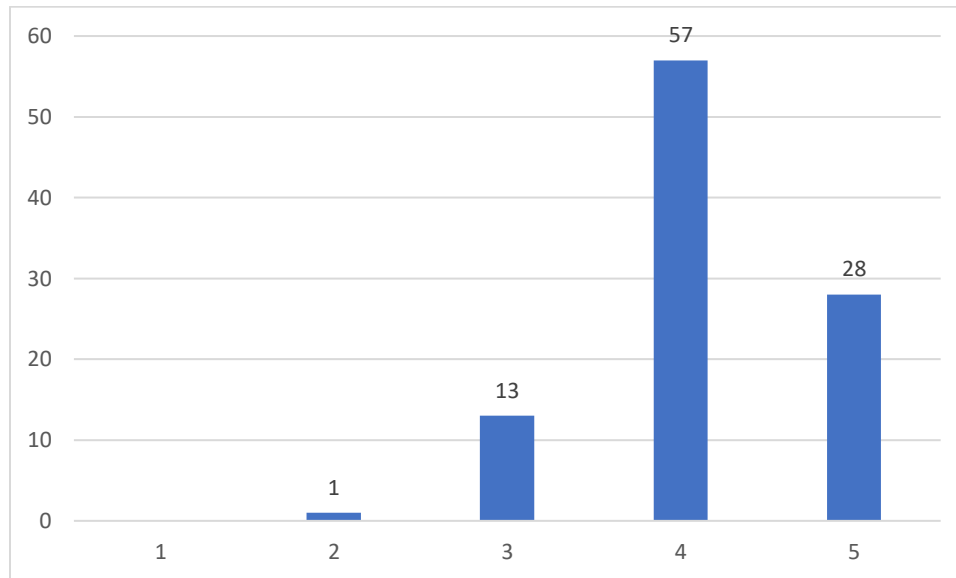


Diagrama VIII- 16 Décimo sexta pregunta encuesta.

Elaborado por: Obando B. - Vega C.

- ¿Utilizaría una nueva aplicación de mensajería instantánea?

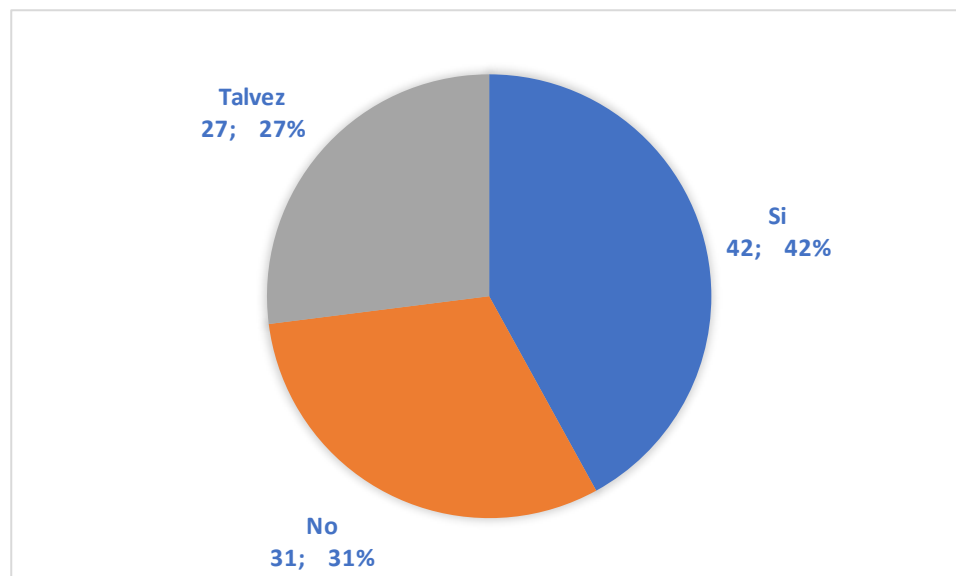


Diagrama VIII- 17 Décimo séptima pregunta encuesta.

Elaborado por: Obando B. - Vega C.

8.2. Tabla de 100 mensajes, 50 caracteres, Redes Wi-Fi, WhatsApp

Mensajes	Emisor		Receptor	
	Download Kb/s	Upload Kb/s	Download Kb/s	Upload Kb/s
1	1	2	3	1
2	1	2	3	2
3	1	2	5	2
4	1	2	3	2
5	2	3	3	3
6	2	2	3	2
7	1	2	4	2
8	1	2	2	2
9	1	2	3	2
10	1	2	3	3
11	1	2	3	3
12	2	4	3	2
13	1	2	3	3
14	1	2	3	2
15	1	2	3	2
16	1	2	2	2
17	1	2	3	2
18	1	2	4	2
19	2	3	3	2
20	1	2	3	2
21	1	2	3	2
22	1	2	5	1
23	1	4	3	1
24	1	2	3	2
25	1	2	3	2
26	1	3	3	1
27	1	2	3	2
28	1	4	3	2
29	1	2	3	2
30	1	2	3	2
31	3	4	5	2
32	2	2	2	2
33	2	3	3	2
34	2	2	3	2
35	1	2	3	2

Mensajes	Emisor		Receptor	
	Download Kb/s	Upload Kb/s	Download Kb/s	Upload Kb/s
36	1	4	3	2
37	1	2	4	3
38	1	2	3	2
39	1	2	3	2
40	1	2	3	2
41	1	2	3	2
42	2	3	3	2
43	2	3	4	3
44	1	2	3	2
45	1	3	3	2
46	1	4	3	2
47	1	3	3	1
48	1	2	3	2
49	1	2	4	2
50	1	2	3	2
51	1	4	3	1
52	1	2	3	2
53	1	2	3	2
54	3	2	3	2
55	2	2	3	2
56	3	2	3	2
57	1	2	5	2
58	1	2	3	2
59	1	2	2	2
60	2	2	3	3
61	1	2	3	2
62	1	2	3	2
63	1	2	3	2
64	3	2	3	1
65	1	4	3	2
66	2	2	4	2
67	2	3	3	2
68	1	2	3	2
69	1	2	3	2
70	2	3	3	2
71	2	2	3	2
72	1	4	3	2
73	1	4	3	2
74	1	2	3	1

Mensajes	Emisor		Receptor	
	Download Kb/s	Upload Kb/s	Download Kb/s	Upload Kb/s
75	3	2	3	2
76	1	2	3	2
77	1	2	4	1
78	2	2	4	2
79	3	2	3	1
80	1	3	3	1
81	3	2	3	2
82	2	3	3	2
83	1	2	3	2
84	1	2	3	2
85	1	2	3	2
86	1	2	3	2
87	1	4	3	2
88	1	2	3	3
89	1	2	3	2
90	2	2	3	2
91	1	2	3	2
92	1	2	5	2
93	1	2	3	2
94	1	2	3	2
95	1	2	3	3
96	2	2	3	2
97	1	2	4	2
98	2	2	3	1
99	1	4	3	2
100	1	2	4	2
Promedio	1	2	3	2
Mínimo	1	2	2	1
Máximo	3	4	5	3
Total	134	236	316	197

Tabla VIII- 1 Redes Wi-Fi, WhatsApp

Elaborado por: Obando B. - Vega C.

8.3. Tabla de 100 mensajes, 50 caracteres, Redes Wi-Fi, Messenger

Mensajes	Emisor		Receptor	
	Download Kb/s	Upload Kb/s	Download Kb/s	Upload Kb/s
1	3	4	6	1
2	3	3	6	2
3	2	3	6	1
4	2	2	6	1
5	2	3	6	1
6	2	3	8	3
7	2	3	8	3
8	2	3	8	3
9	2	3	6	2
10	2	3	6	2
11	2	4	6	2
12	3	3	6	2
13	3	3	6	2
14	3	3	6	2
15	3	3	6	2
16	2	3	6	2
17	2	3	6	2
18	2	3	7	3
19	3	3	6	2
20	3	3	6	2
21	3	3	6	2
22	3	3	6	2
23	3	3	6	2
24	3	3	6	2
25	3	3	7	3
26	4	3	6	2
27	2	3	6	2
28	3	3	6	2
29	2	3	8	2
30	3	3	8	3
31	2	3	6	2
32	2	3	7	3
33	2	3	6	2
34	2	3	6	2
35	2	3	6	2

Mensajes	Emisor		Receptor	
	Download Kb/s	Upload Kb/s	Download Kb/s	Upload Kb/s
36	3	3	6	2
37	2	3	6	2
38	4	3	7	2
39	3	3	6	2
40	2	3	6	2
41	3	3	6	2
42	2	3	7	3
43	3	3	6	3
44	4	3	6	2
45	3	3	6	2
46	5	3	6	2
47	3	3	6	2
48	4	3	6	2
49	3	3	6	2
50	3	3	6	2
51	4	3	6	2
52	3	3	6	2
53	3	3	6	2
54	3	3	6	2
55	3	3	6	2
56	3	3	6	2
57	3	3	6	3
58	3	3	7	2
59	3	3	6	2
60	2	3	6	2
61	3	3	6	2
62	3	3	6	2
63	3	3	6	2
64	2	3	6	2
65	4	3	6	2
66	3	3	6	2
67	3	3	6	2
68	3	3	6	2
69	2	3	6	2
70	2	3	6	2
71	3	3	7	2
72	2	3	6	2
73	2	3	6	2
74	2	3	6	2

Mensajes	Emisor		Receptor	
	Download Kb/s	Upload Kb/s	Download Kb/s	Upload Kb/s
75	4	3	6	2
76	4	3	6	3
77	2	3	6	2
78	2	3	6	2
79	3	3	6	2
80	3	3	6	2
81	2	3	6	2
82	2	3	6	2
83	2	3	7	2
84	3	3	6	2
85	3	3	6	3
86	4	3	6	2
87	2	3	6	2
88	2	3	6	2
89	2	3	7	2
90	3	5	6	2
91	3	3	6	2
92	4	3	6	3
93	3	3	6	2
94	5	3	6	2
95	3	3	6	2
96	3	3	6	2
97	3	2	6	2
98	4	3	6	2
99	3	3	6	2
100	3	3	6	2
Promedio	3	3	6	2
Mínimo	2	2	6	1
Máximo	5	5	8	3
Total	278	302	619	209

Tabla VIII- 2 Redes Wi-Fi, Messenger

Elaborado por: Obando B. - Vega C.

8.4. Tabla de 100 mensajes, 50 caracteres, Redes Wi-Fi, Skype.

Mensajes	Emisor		Receptor	
	Download Kb/s	Upload Kb/s	Download Kb/s	Upload Kb/s
1	15	20	20	2
2	14	21	25	2
3	16	21	24	2
4	18	21	21	2
5	17	20	21	2
6	9	23	23	2
7	10	25	20	3
8	11	20	19	2
9	12	20	20	2
10	10	24	23	2
11	15	21	25	3
12	15	21	20	4
13	15	20	24	5
14	16	23	20	2
15	14	20	23	2
16	15	21	20	4
17	14	20	20	2
18	18	20	14	5
19	18	25	20	2
20	20	23	20	3
21	15	24	28	2
22	15	21	27	5
23	15	20	29	2
24	16	21	20	4
25	16	20	20	2
26	15	20	19	2
27	14	21	18	5
28	15	20	18	4
29	14	21	27	2
30	15	20	20	2
31	15	20	20	4
32	15	21	21	5
33	16	20	21	2
34	10	21	22	4
35	19	22	20	5

Mensajes	Emisor		Receptor	
	Download Kb/s	Upload Kb/s	Download Kb/s	Upload Kb/s
36	11	22	20	2
37	12	23	25	2
38	13	19	20	2
39	15	18	20	2
40	12	18	19	2
41	12	17	20	2
42	12	18	19	3
43	15	19	20	3
44	15	20	20	2
45	15	20	24	2
46	15	20	20	2
47	15	24	23	4
48	9	25	20	2
49	9	20	25	5
50	9	20	20	2
51	9	26	19	2
52	15	20	20	2
53	15	26	28	5
54	15	25	20	2
55	15	20	21	2
56	16	20	20	2
57	14	20	23	4
58	18	20	20	2
59	16	20	22	4
60	17	21	20	2
61	19	20	18	4
62	15	20	19	2
63	14	21	19	5
64	15	21	18	2
65	16	24	18	2
66	15	18	19	2
67	17	19	20	2
68	18	20	20	2
69	18	20	20	5
70	15	24	23	2
71	15	25	20	2
72	16	20	21	5
73	14	20	20	2
74	15	20	23	2

Mensajes	Emisor		Receptor	
	Download Kb/s	Upload Kb/s	Download Kb/s	Upload Kb/s
75	11	19	20	4
76	12	20	21	2
77	12	23	20	2
78	15	22	20	4
79	10	20	20	2
80	10	21	18	2
81	9	21	19	2
82	9	20	20	1
83	9	20	29	2
84	14	23	20	3
85	15	20	20	4
86	15	20	20	2
87	15	21	23	2
88	20	20	20	2
89	14	20	24	2
90	14	20	25	5
91	15	20	20	2
92	15	24	21	2
93	14	20	21	5
94	12	17	21	2
95	10	18	22	3
96	9	19	20	2
97	10	20	19	3
98	11	20	19	4
99	12	20	18	5
100	13	20	18	5
Promedio	14	21	21	3
Mínimo	9	17	14	1
Máximo	20	26	29	5
Total	1403	2082	2096	280

Tabla VIII- 3 Redes Wi-Fi, Skype

Elaborado por: Obando B. - Vega C.

8.5. Tabla de 100 mensajes, 50 caracteres, Redes Wi-Fi, Prototipo.

Mensajes	Emisor		Receptor	
	Download Kb/s	Upload Kb/s	Download Kb/s	Upload Kb/s
1	7	7	5	7
2	7	7	6	6
3	7	6	5	7
4	7	6	6	7
5	6	6	6	7
6	6	6	6	5
7	6	6	6	6
8	6	6	6	7
9	6	7	6	8
10	6	6	6	7
11	6	6	6	7
12	6	6	7	8
13	6	6	7	7
14	6	7	7	6
15	6	6	7	5
16	6	7	5	7
17	6	7	7	7
18	6	7	5	8
19	6	6	7	7
20	6	7	7	7
21	6	6	6	6
22	6	7	7	7
23	6	6	7	7
24	6	7	6	8
25	6	6	6	7
26	6	6	5	5
27	6	6	6	6
28	7	7	6	6
29	6	7	8	7
30	6	7	8	8
31	7	7	7	7
32	6	6	7	7
33	7	7	6	6
34	6	7	6	6
35	6	7	6	7

Mensajes	Emisor		Receptor	
	Download Kb/s	Upload Kb/s	Download Kb/s	Upload Kb/s
36	6	7	7	7
37	6	6	7	8
38	6	6	6	7
39	6	6	6	7
40	6	6	6	8
41	6	6	5	7
42	7	6	6	8
43	7	6	6	7
44	7	6	7	5
45	7	7	7	6
46	7	7	7	7
47	7	7	8	7
48	6	7	8	6
49	6	6	7	7
50	6	6	7	7
51	6	6	7	8
52	6	7	6	7
53	6	7	7	6
54	6	6	6	6
55	6	6	6	7
56	6	6	7	8
57	6	6	7	7
58	7	7	7	7
59	6	6	8	7
60	6	6	8	8
61	7	6	7	7
62	7	6	7	7
63	6	7	6	7
64	7	7	6	8
65	7	7	5	7
66	6	7	5	7
67	6	6	5	7
68	6	6	5	7
69	7	6	5	6
70	6	6	6	7
71	7	6	6	8
72	6	6	7	7
73	6	7	8	6
74	6	7	5	7

Mensajes	Emisor		Receptor	
	Download Kb/s	Upload Kb/s	Download Kb/s	Upload Kb/s
75	6	7	6	7
76	6	6	6	7
77	6	7	6	8
78	6	6	6	7
79	6	6	5	6
80	6	7	5	7
81	6	7	7	8
82	6	7	7	7
83	6	7	7	7
84	6	6	7	6
85	6	6	7	6
86	6	6	8	6
87	6	7	5	7
88	6	7	7	7
89	6	6	5	8
90	6	6	7	8
91	6	7	7	7
92	6	6	7	7
93	7	7	6	8
94	6	6	6	7
95	6	7	6	7
96	7	7	6	8
97	6	7	6	7
98	7	6	7	8
99	6	6	7	7
100	6	6	7	7
Promedio	6	6	6	7
Mínimo	6	6	5	5
Máximo	7	7	8	8
Total	623	644	638	694

Tabla VIII- 4 Redes Wi-Fi, Prototipo

Elaborado por: Obando B. - Vega C.

8.6. Tabla de 100 mensajes, 50 caracteres, Redes Móviles, Whatsapp.

Mensajes	Emisor		Receptor	
	Download Kb/s	Upload Kb/s	Download Kb/s	Upload Kb/s
1	1	2	5	3
2	1	2	2	1
3	1	2	2	1
4	1	2	2	1
5	2	3	2	1
6	1	2	2	1
7	1	2	2	1
8	2	2	2	1
9	1	2	2	1
10	1	2	2	1
11	1	2	2	2
12	1	3	2	1
13	1	4	2	1
14	2	3	2	1
15	1	2	4	2
16	1	2	2	1
17	1	2	2	1
18	1	2	2	1
19	1	2	2	1
20	1	2	2	1
21	1	2	2	1
22	1	2	2	1
23	2	2	2	2
24	1	2	4	2
25	1	2	2	3
26	1	2	2	1
27	1	2	2	1
28	2	3	2	1
29	1	2	2	1
30	1	2	2	1
31	1	2	3	1
32	1	2	2	1
33	1	2	2	2
34	1	2	2	1
35	1	2	4	1

Mensajes	Emisor		Receptor	
	Download Kb/s	Upload Kb/s	Download Kb/s	Upload Kb/s
36	1	4	2	1
37	1	2	2	1
38	1	2	2	1
39	1	2	2	1
40	2	3	2	1
41	1	2	2	1
42	1	2	4	1
43	1	2	2	1
44	1	2	2	1
45	1	2	2	1
46	2	3	2	1
47	1	2	4	3
48	1	2	2	1
49	1	2	2	1
50	1	3	2	1
51	1	2	2	3
52	1	2	3	1
53	1	2	2	1
54	4	5	2	1
55	1	2	2	1
56	1	2	2	1
57	1	2	2	3
58	1	2	2	1
59	1	2	3	1
60	1	2	2	1
61	1	2	2	1
62	1	3	2	1
63	3	2	2	1
64	2	2	2	1
65	1	2	2	1
66	1	4	2	1
67	1	2	2	1
68	1	2	2	1
69	1	2	2	1
70	1	2	2	1
71	1	2	2	1
72	2	2	2	1
73	1	2	2	1
74	1	2	2	1

Mensajes	Emisor		Receptor	
	Download Kb/s	Upload Kb/s	Download Kb/s	Upload Kb/s
75	1	2	2	4
76	2	2	2	1
77	1	2	2	1
78	1	2	3	1
79	1	2	2	1
80	2	2	2	2
81	1	2	2	1
82	1	2	2	1
83	1	2	2	1
84	1	2	2	1
85	1	2	2	1
86	1	2	2	1
87	1	2	2	1
88	2	2	3	1
89	1	2	2	2
90	1	2	2	1
91	1	2	2	1
92	1	2	2	1
93	1	2	5	3
94	1	2	2	1
95	1	2	2	1
96	3	4	2	2
97	1	2	2	1
98	1	2	2	1
99	1	2	3	3
100	1	2	2	1
Promedio	1	2	2	1
Mínimo	1	2	2	1
Máximo	4	5	5	4
Total	119	219	222	125

Tabla VIII- 5 Redes Móviles, Whatsapp

Elaborado por: Obando B. - Vega C.

8.7. Tabla de 100 mensajes, 50 caracteres, Redes Móviles, Messenger.

Mensajes	Emisor		Receptor	
	Download Kb/s	Upload Kb/s	Download Kb/s	Upload Kb/s
1	3	3	6	2
2	3	3	6	2
3	2	3	6	2
4	3	3	6	2
5	3	3	6	2
6	3	3	6	1
7	3	4	6	1
8	3	3	6	1
9	4	4	6	2
10	4	3	6	1
11	2	2	8	2
12	3	3	6	2
13	5	3	6	2
14	3	3	6	2
15	3	3	6	2
16	3	3	8	2
17	3	3	6	2
18	5	3	7	1
19	2	3	6	2
20	3	3	6	2
21	4	3	8	2
22	4	3	6	2
23	3	3	6	3
24	4	3	6	3
25	3	3	7	3
26	3	3	6	2
27	3	3	6	3
28	4	3	6	2
29	4	3	6	2
30	3	3	8	2
31	3	3	6	2
32	3	3	7	2
33	3	3	6	2
34	4	3	6	1
35	5	3	6	2

Mensajes	Emisor		Receptor	
	Download Kb/s	Upload Kb/s	Download Kb/s	Upload Kb/s
36	3	3	6	2
37	3	3	6	2
38	3	3	7	2
39	3	3	6	2
40	3	3	6	2
41	3	3	6	3
42	3	3	7	3
43	2	3	6	1
44	4	3	8	2
45	3	3	6	2
46	2	3	6	2
47	4	3	6	2
48	3	3	6	4
49	3	3	6	2
50	2	4	6	2
51	3	3	6	2
52	2	3	6	2
53	3	3	6	2
54	3	3	6	2
55	3	4	6	3
56	2	3	6	2
57	3	3	6	2
58	3	3	6	2
59	4	3	6	2
60	3	3	6	3
61	3	3	6	3
62	3	3	6	2
63	3	3	8	2
64	4	3	7	2
65	4	3	6	2
66	3	3	6	3
67	3	3	7	2
68	3	3	6	3
69	3	3	6	2
70	3	3	6	2
71	3	3	7	2
72	3	3	6	3
73	3	3	6	2
74	3	3	6	2

Mensajes	Emisor		Receptor	
	Download Kb/s	Upload Kb/s	Download Kb/s	Upload Kb/s
75	3	3	6	2
76	3	2	6	3
77	3	3	6	2
78	3	3	7	2
79	3	3	7	2
80	3	3	6	2
81	3	3	6	2
82	4	3	6	2
83	3	3	6	2
84	3	3	8	2
85	3	3	6	3
86	4	3	6	2
87	3	3	6	2
88	3	3	6	2
89	3	3	6	2
90	3	3	6	2
91	3	3	6	2
92	4	3	5	2
93	3	3	8	2
94	2	3	6	3
95	3	3	6	2
96	3	3	6	2
97	3	3	6	2
98	4	4	5	3
99	3	4	6	2
100	3	3	6	2
Promedio	3	3	6	2
Mínimo	2	2	5	1
Máximo	5	4	8	4
Total	314	304	624	211

Tabla VIII- 6 Redes Móviles, Messenger

Elaborado por: Obando B. - Vega C.

8.8. Tabla de 100 mensajes, 50 caracteres, Redes Móviles, Skype.

Mensajes	Emisor		Receptor	
	Download Kb/s	Upload Kb/s	Download Kb/s	Upload Kb/s
1	18	20	20	2
2	18	19	20	3
3	18	20	24	2
4	19	24	21	2
5	19	24	20	2
6	20	25	21	3
7	20	26	23	2
8	21	25	20	2
9	20	26	20	2
10	20	23	25	3
11	21	22	20	2
12	21	21	20	2
13	21	22	23	2
14	20	23	20	3
15	20	22	24	2
16	24	23	20	2
17	25	19	21	4
18	24	20	20	2
19	23	20	21	5
20	22	23	20	2
21	23	24	19	2
22	21	25	18	6
23	20	27	18	2
24	20	28	19	5
25	21	29	18	2
26	20	28	15	5
27	19	26	17	2
28	19	20	18	2
29	18	21	19	4
30	27	23	19	4
31	20	20	24	5
32	21	25	22	5
33	20	24	25	5
34	19	23	21	5
35	19	22	20	2

Mensajes	Emisor		Receptor	
	Download Kb/s	Upload Kb/s	Download Kb/s	Upload Kb/s
36	18	22	21	2
37	19	21	21	2
38	18	20	21	3
39	19	20	20	2
40	18	19	21	4
41	19	20	23	2
42	19	23	25	5
43	18	20	24	5
44	19	24	22	5
45	20	20	20	2
46	20	23	20	2
47	22	25	19	2
48	20	27	18	2
49	21	21	19	3
50	23	20	20	2
51	23	23	28	1
52	23	20	20	2
53	20	21	20	1
54	23	20	24	2
55	20	25	20	3
56	24	29	22	2
57	25	24	23	3
58	20	28	25	2
59	21	20	25	2
60	20	22	20	3
61	21	23	20	4
62	20	23	21	5
63	23	23	20	4
64	20	23	20	4
65	23	24	21	4
66	22	25	20	2
67	20	25	21	2
68	21	26	20	2
69	20	26	20	3
70	20	23	20	2
71	19	25	23	2
72	18	24	20	2
73	18	25	23	3
74	19	25	24	2

Mensajes	Emisor		Receptor	
	Download Kb/s	Upload Kb/s	Download Kb/s	Upload Kb/s
75	20	21	25	4
76	18	24	25	2
77	19	23	21	2
78	20	22	22	2
79	18	22	20	5
80	20	23	19	2
81	23	25	19	5
82	23	26	18	2
83	24	25	19	2
84	20	24	18	2
85	25	23	19	2
86	20	25	20	3
87	21	24	21	3
88	18	22	21	2
89	21	22	21	4
90	20	21	21	2
91	19	20	21	5
92	19	23	21	2
93	20	24	24	2
94	19	25	20	4
95	20	20	20	2
96	18	24	20	3
97	19	25	20	2
98	20	20	19	2
99	19	21	18	2
100	20	20	19	5
Promedio	20	23	21	3
Mínimo	18	19	15	1
Máximo	27	29	28	6
Total	2040	2303	2080	283

Tabla VIII- 7 Redes Móviles, Skype

Elaborado por: Obando B. - Vega C.

8.9. Tabla de 100 mensajes, 50 caracteres, Redes Móviles, Prototipo.

Mensajes	Emisor		Receptor	
	Download Kb/s	Upload Kb/s	Download Kb/s	Upload Kb/s
1	7	10	10	7
2	6	10	9	8
3	7	8	9	7
4	7	10	9	7
5	7	8	7	7
6	7	8	8	7
7	7	9	8	8
8	7	8	8	7
9	7	9	7	7
10	7	8	8	8
11	8	8	7	9
12	7	8	8	7
13	7	8	7	7
14	7	9	10	8
15	7	8	10	9
16	7	9	9	9
17	7	8	8	9
18	8	9	8	7
19	8	9	8	9
20	6	9	7	7
21	8	8	8	9
22	6	8	6	9
23	8	8	7	10
24	8	8	8	10
25	8	8	9	10
26	7	8	6	9
27	7	8	6	9
28	7	8	6	9
29	7	8	6	10
30	7	10	6	10
31	7	10	6	8
32	7	8	7	8
33	7	9	8	8
34	7	9	6	7
35	8	8	8	7

Mensajes	Emisor		Receptor	
	Download Kb/s	Upload Kb/s	Download Kb/s	Upload Kb/s
36	7	8	8	8
37	7	8	8	6
38	7	8	6	7
39	7	8	6	7
40	6	8	7	8
41	7	8	8	7
42	7	8	7	8
43	7	9	8	7
44	6	9	8	8
45	6	8	8	7
46	6	8	8	8
47	6	8	7	8
48	7	8	7	8
49	7	9	8	8
50	7	8	8	9
51	8	8	7	9
52	7	8	9	10
53	8	8	10	9
54	8	8	10	9
55	7	8	10	9
56	7	8	9	9
57	7	8	9	10
58	7	8	9	9
59	7	9	8	9
60	7	8	8	9
61	8	8	7	9
62	7	8	8	10
63	7	8	7	7
64	7	8	8	7
65	6	8	7	8
66	7	9	8	8
67	7	8	7	8
68	8	9	8	9
69	7	9	7	9
70	8	9	7	10
71	6	8	7	10
72	7	8	8	7
73	7	8	9	7
74	8	8	7	7

Mensajes	Emisor		Receptor	
	Download Kb/s	Upload Kb/s	Download Kb/s	Upload Kb/s
75	7	8	6	8
76	8	8	5	9
77	6	10	7	6
78	7	8	7	8
79	8	8	8	8
80	7	8	8	6
81	7	8	10	7
82	7	8	9	7
83	8	9	10	8
84	7	8	10	8
85	7	8	10	9
86	7	9	9	9
87	7	8	9	9
88	7	8	11	9
89	7	8	10	10
90	7	9	8	10
91	7	8	10	8
92	7	8	7	8
93	7	9	7	7
94	7	9	7	7
95	7	8	8	7
96	7	8	8	8
97	7	8	8	8
98	7	8	10	8
99	7	8	9	7
100	7	9	9	7
Promedio	7	8	8	8
Mínimo	6	8	5	6
Máximo	8	10	11	10
Total	707	835	794	815

Tabla VIII- 8 Redes Móviles, Prototipo

Elaborado por: Obando B. - Vega C.

8.10. Tabla de 100 mensajes, 50 caracteres, PC, Whatsapp.

Mensajes	Emisor	Móvil	Receptor	PC
	Download Kb/s	Upload Kb/s	Download Kb/s	Upload Kb/s
1	3	4	206	3
2	2	4	206	4
3	2	4	206	4
4	2	4	206	3
5	3	4	207	3
6	3	4	206	4
7	3	4	206	3
8	2	5	206	4
9	2	5	208	3
10	3	5	206	3
11	2	4	207	3
12	2	6	206	3
13	3	5	206	3
14	2	4	206	4
15	2	5	200	3
16	2	5	204	4
17	3	4	205	5
18	3	5	205	3
19	3	4	205	4
20	2	5	206	3
21	3	5	206	4
22	3	5	208	3
23	2	5	208	3
24	4	5	208	3
25	3	4	208	4
26	2	4	206	4
27	4	4	206	4
28	3	4	206	3
29	2	4	206	4
30	2	4	206	4
31	2	4	206	6
32	2	4	207	4
33	3	5	207	3
34	2	4	206	4
35	2	4	206	3

Mensajes	Emisor		Receptor	
	Download Kb/s	Upload Kb/s	Download Kb/s	Upload Kb/s
36	2	5	207	4
37	2	4	206	6
38	2	5	204	4
39	3	5	205	5
40	3	4	206	3
41	3	5	206	5
42	3	5	206	4
43	3	4	206	6
44	3	4	206	6
45	3	4	206	4
46	3	4	206	6
47	3	6	206	6
48	2	4	206	6
49	3	5	206	5
50	2	4	206	5
51	3	4	206	5
52	3	4	208	5
53	3	4	208	3
54	4	4	200	3
55	3	4	200	4
56	3	5	204	4
57	3	4	202	4
58	3	6	202	4
59	3	5	204	4
60	3	4	202	4
61	3	4	206	4
62	4	4	206	4
63	2	5	205	4
64	3	4	205	4
65	3	4	206	3
66	3	4	206	4
67	4	5	205	3
68	3	4	206	4
69	2	5	206	4
70	3	6	206	4
71	3	5	206	3
72	3	4	206	4
73	3	5	206	5
74	4	4	206	4

Mensajes	Emisor		Receptor	
	Download Kb/s	Upload Kb/s	Download Kb/s	Upload Kb/s
75	3	4	204	3
76	3	4	205	4
77	3	4	205	3
78	2	4	204	4
79	3	5	206	4
80	4	4	206	4
81	3	4	206	3
82	3	5	206	4
83	3	4	206	4
84	4	4	205	4
85	3	5	205	3
86	3	4	205	4
87	3	6	205	4
88	2	5	205	4
89	4	4	206	4
90	3	4	206	4
91	3	4	206	4
92	3	5	206	4
93	3	4	204	3
94	3	5	202	4
95	3	4	202	3
96	3	4	203	4
97	3	4	200	3
98	3	6	205	5
99	3	5	206	4
100	3	4	206	3
Promedio	3	4	205	4
Mínimo	2	4	200	3
Máximo	4	6	208	6
Total	281	445	20543	391

Tabla VIII- 9 PC, Whatsapp

Elaborado por: Obando B. - Vega C.

8.11. Tabla de 100 mensajes, 50 caracteres, PC, Messenger.

Mensajes	Emisor	Móvil	Receptor	PC
	Download Kb/s	Upload Kb/s	Download Kb/s	Upload Kb/s
1	2	5	215	3
2	2	4	216	2
3	2	4	216	2
4	3	4	216	2
5	3	4	215	2
6	3	4	219	2
7	3	4	220	2
8	3	4	224	3
9	3	4	220	2
10	3	4	221	2
11	3	4	222	2
12	2	2	223	2
13	2	2	219	3
14	4	4	218	3
15	4	4	217	2
16	2	4	219	3
17	3	4	220	2
18	3	4	221	2
19	2	4	220	3
20	3	4	219	3
21	3	3	218	2
22	3	4	217	2
23	3	4	218	2
24	3	4	219	2
25	3	3	219	2
26	2	4	220	2
27	3	4	221	3
28	3	4	223	2
29	2	4	225	2
30	3	4	218	2
31	3	4	219	2
32	2	4	217	2
33	3	4	218	3
34	3	4	219	3
35	3	4	219	3

Mensajes	Emisor		Receptor	
	Download Kb/s	Upload Kb/s	Download Kb/s	Upload Kb/s
36	2	4	218	2
37	3	4	216	2
38	2	4	217	2
39	3	4	218	2
40	3	4	219	3
41	3	4	219	2
42	3	4	220	2
43	3	2	223	2
44	3	4	224	2
45	3	4	217	2
46	2	4	219	3
47	2	2	226	2
48	3	4	228	3
49	2	4	225	2
50	3	3	224	3
51	3	3	224	2
52	2	4	221	2
53	4	4	227	2
54	2	4	228	2
55	3	4	229	3
56	2	4	218	2
57	2	4	217	2
58	2	4	217	2
59	3	4	224	2
60	3	4	220	2
61	4	4	221	2
62	3	3	223	2
63	2	4	227	3
64	2	4	224	2
65	2	3	221	2
66	3	4	223	2
67	2	4	218	2
68	4	4	219	2
69	5	4	217	2
70	5	4	219	2
71	3	4	220	2
72	3	4	221	2
73	2	4	217	2
74	3	4	218	2

Mensajes	Emisor		Receptor	
	Download Kb/s	Upload Kb/s	Download Kb/s	Upload Kb/s
75	5	4	219	2
76	2	4	220	2
77	3	4	221	2
78	3	4	223	2
79	3	3	222	3
80	3	4	217	3
81	3	4	218	2
82	2	4	219	2
83	3	4	220	2
84	2	4	224	2
85	3	4	223	2
86	3	3	221	2
87	2	4	221	2
88	3	4	220	2
89	4	4	219	2
90	3	4	221	3
91	3	4	222	2
92	4	4	223	2
93	3	4	224	1
94	2	4	218	2
95	2	4	217	3
96	3	4	216	3
97	3	3	224	1
98	3	4	227	2
99	2	4	218	2
100	2	4	223	2
Promedio	3	4	220	2
Mínimo	2	2	215	1
Máximo	5	5	229	3
Total	280	384	22036	220

Tabla VIII- 10 PC, Messenger

Elaborado por: Obando B. - Vega C.

8.12. Tabla de 100 mensajes, 50 caracteres, PC, Skype.

Mensajes	Emisor	Móvil	Receptor	PC
	Download Kb/s	Upload Kb/s	Download Kb/s	Upload Kb/s
1	18	22	236	3
2	19	24	235	2
3	17	25	238	3
4	20	26	236	2
5	25	28	239	5
6	26	24	239	2
7	23	25	238	2
8	24	23	247	2
9	22	25	235	2
10	25	26	236	2
11	28	25	235	1
12	27	21	238	2
13	20	21	239	2
14	19	24	248	1
15	18	26	242	2
16	23	25	243	2
17	24	23	235	2
18	26	25	248	2
19	25	23	247	6
20	21	27	240	1
21	20	28	241	3
22	20	29	245	2
23	20	25	243	2
24	23	26	241	2
25	21	23	240	2
26	23	24	239	4
27	24	25	237	1
28	21	26	234	2
29	18	21	247	2
30	19	22	240	2
31	17	20	241	2
32	18	20	243	2
33	19	19	235	2
34	15	20	236	2
35	19	24	235	2

Mensajes	Emisor		Receptor	
	Download Kb/s	Upload Kb/s	Download Kb/s	Upload Kb/s
36	20	26	237	2
37	23	23	236	2
38	22	25	237	2
39	24	27	238	1
40	26	28	237	2
41	25	24	239	2
42	27	20	237	2
43	20	19	234	3
44	21	18	234	2
45	26	20	231	1
46	25	23	237	2
47	24	21	248	2
48	28	23	247	2
49	23	22	240	2
50	27	21	241	2
51	19	21	242	1
52	18	24	240	2
53	19	23	243	2
54	18	26	243	2
55	16	25	253	1
56	18	27	246	2
57	20	22	247	1
58	21	20	234	1
59	21	19	247	2
60	24	18	240	6
61	22	19	241	2
62	26	20	241	1
63	23	25	240	2
64	28	26	242	4
65	27	24	243	3
66	24	20	245	2
67	26	19	247	1
68	28	18	248	2
69	26	17	241	1
70	24	20	245	2
71	25	23	246	2
72	24	25	248	2
73	25	19	249	6
74	26	24	247	2

Mensajes	Emisor		Receptor	
	Download Kb/s	Upload Kb/s	Download Kb/s	Upload Kb/s
75	24	26	241	2
76	22	26	246	2
77	23	27	244	2
78	24	28	244	1
79	25	20	245	2
80	26	23	246	2
81	27	26	248	2
82	22	25	247	3
83	25	19	249	2
84	26	24	249	1
85	24	24	247	2
86	23	23	240	2
87	21	26	239	2
88	23	22	238	1
89	22	25	247	2
90	24	26	248	2
91	25	26	236	2
92	26	25	237	2
93	25	27	248	2
94	24	26	247	2
95	21	23	241	4
96	22	25	242	2
97	22	23	243	2
98	22	25	246	2
99	20	23	235	3
100	20	20	247	3
Promedio	23	23	242	2
Mínimo	15	17	231	1
Máximo	28	29	253	6

Tabla VIII- 11 PC, Skype

Elaborado por: Obando B. - Vega C.

8.13. Tabla de 100 mensajes, 50 caracteres, PC, Prototipo.

Mensajes	Emisor	Móvil	Receptor	PC
	Download Kb/s	Upload Kb/s	Download Kb/s	Upload Kb/s
1	7	10	329	16
2	6	10	329	11
3	7	8	328	11
4	7	10	329	12
5	7	8	329	14
6	7	8	328	12
7	7	9	330	12
8	7	8	328	13
9	7	9	328	14
10	7	8	330	14
11	8	8	329	14
12	7	8	328	14
13	7	8	330	15
14	7	9	329	16
15	7	8	328	15
16	7	9	330	15
17	7	8	329	15
18	8	9	329	15
19	8	9	330	16
20	6	9	330	18
21	8	8	329	21
22	6	8	331	15
23	8	8	329	15
24	8	8	328	15
25	8	8	330	15
26	7	8	329	18
27	7	8	330	17
28	7	8	330	15
29	7	8	330	15
30	7	10	330	16
31	7	10	331	16
32	7	8	330	19
33	7	9	332	16
34	7	9	331	17
35	8	8	331	18

Mensajes	Emisor		Receptor	
	Download Kb/s	Upload Kb/s	Download Kb/s	Upload Kb/s
36	7	8	331	21
37	7	8	331	21
38	7	8	330	18
39	7	8	332	20
40	6	8	331	22
41	7	8	330	21
42	7	8	332	21
43	7	9	330	22
44	6	9	330	22
45	6	8	332	23
46	6	8	331	22
47	6	8	330	24
48	7	8	332	22
49	7	9	331	24
50	7	8	332	23
51	8	8	332	31
52	7	8	330	21
53	8	8	330	21
54	8	8	332	21
55	7	8	331	22
56	7	8	330	22
57	7	8	331	22
58	7	8	331	22
59	7	9	330	22
60	7	8	331	22
61	8	8	331	22
62	7	8	331	23
63	7	8	332	22
64	7	8	331	22
65	6	8	331	22
66	7	9	331	22
67	7	8	331	22
68	8	9	332	24
69	7	9	332	28
70	8	9	331	28
71	6	8	332	26
72	7	8	332	26
73	7	8	331	26
74	8	8	332	28

Mensajes	Emisor		Receptor	
	Download Kb/s	Upload Kb/s	Download Kb/s	Upload Kb/s
75	7	8	332	28
76	8	8	331	31
77	6	10	333	32
78	7	8	332	25
79	8	8	332	28
80	7	8	333	25
81	7	8	332	26
82	7	8	332	25
83	8	9	332	25
84	7	8	332	25
85	7	8	333	28
86	7	9	332	26
87	7	8	332	26
88	7	8	332	26
89	7	8	332	30
90	7	9	332	26
91	7	8	333	25
92	7	8	332	26
93	7	9	332	30
94	7	9	332	30
95	7	8	332	28
96	7	8	332	31
97	7	8	333	32
98	7	8	335	33
99	7	8	332	30
100	7	9	334	32
Promedio	7	8	331	22
Mínimo	6	8	328	11
Máximo	8	10	335	33
Total	707	835	33085	2152

Tabla VIII- 12 PC, Prototipo

Elaborado por: Obando B. - Vega C.

Bibliografía

- Adrian, C. (24 de Enero de 2014). *Wordpress*. Obtenido de Entorno de Desarrollo Integrado (IDE):
<https://caleroadrian93.wordpress.com/2014/01/24/entorno-de-desarrollo-integrado-ide/>
- Agesa, L. (5 de Octubre de 2010). *Alegsa*. Obtenido de Definición de UDP:
<http://www.alegsa.com.ar/Dic/udp.php>
- Android. (2017). *Developer Android*. Obtenido de Conoce Android Studio:
<https://developer.android.com/studio/intro/index.html?hl=es-419>
- Anghel, L. (29 de Marzo de 2016). *What is Prime Faces?* Obtenido de developer.com:
<http://www.developer.com/java/data/what-is-primefaces.html>
- Barry, D. K. (1 de Marzo de 2003). *Service Architecture*. Obtenido de Universal Description, Discovery, and Integration (UDDI): http://www.service-architecture.com/articles/web-services/universal_description_discovery_and_integration_uddi.html
- Bembibre, V. (1 de Diciembre de 2009). *DefinicionABC*. Obtenido de Definición de HTTP:
<https://www.definicionabc.com/tecnologia/http.php>
- Burnette, E. (Agosto de 2005). *Eclipse IDE Pocket Guide*. Sebastopol, California, Estados Unidos de America. Obtenido de https://books.google.com.ec/books?id=MUQyvAIn5YQC&printsec=frontcover&dq=eclipse+IDE+book&hl=en&sa=X&redir_esc=y#v=onepage&q=IBM&f=false
- Carcerca. (9 de Diciembre de 2001). *Museo de informática*. Obtenido de El Sistema Operativo WINDOWS: desde sus inicios a la actualidad: <http://histinf.blogs.upv.es/2011/12/09/el-sistema-operativo-windows-desde-sus-inicios-a-la-actualidad/>
- Castells, M. (7 de Julio de 2000). Obtenido de academia.edu:
http://s3.amazonaws.com/academia.edu/documents/34314728/INTERNET_Y_LA_SOCIA

EDAD_RED.pdf?AWSAccessKeyId=AKIAIWOWYYGZ2Y53UL3A&Expires=1492465186&Signature=XEBWwKkUxad9S2g9srr8CiqUOjw%3D&response-content-disposition=inline%3B%20filename%3DINTERNET_Y_LA_SOCIED

CCM. (Marzo de 2017). *CCM*. Obtenido de Protocolo FTP (protocolo de transferencia de archivos): <http://es.ccm.net/contents/263-protocolo-ftp-protocolo-de-transferencia-de-archivos>

CCM. (Junio de 2017). *CCM*. Obtenido de Protocolo FTP (protocolo de transferencia de archivos): <http://es.ccm.net/contents/263-protocolo-ftp-protocolo-de-transferencia-de-archivos>

CCM. (Marzo de 2017). *CCM*. Obtenido de Introducción a las bases de datos: <http://es.ccm.net/contents/66-introduccion-a-las-bases-de-datos>

Christensen, E., Curbera, F., Meredith, G., & Weerawarana, S. (15 de Marzo de 2001). *W3C*. Obtenido de W3C Note: Web Services Description Language (WSDL) 1.1: <https://www.w3.org/TR/wsdl>

Çivici, Ç. (8 de Marzo de 2017). *PRIMEFACES User Guide*. Obtenido de https://www.primefaces.org/docs/guide/primefaces_user_guide_6_0.pdf

Code2Care. (9 de Octubre de 2015). *Code2Care*. Obtenido de List Of Eclipse IDE Versions and Future Releases: Mars and Neon: <http://code2care.org/2015/list-eclipse-ide-versions-future-releases-mars-neon/>

Dias-Cacho, B. R. (10 de Febrero de 2010). *ScienceDirect*. Obtenido de <http://www.sciencedirect.com/science/article/pii/S1697791210700303>

DINECOM. (27 de Octubre de 2014). *DINECOM*. Obtenido de Videoconferencia: ¿Qué se supone que es el protocolo RTCP?: <http://www.dinecom.cl/blogs/diccionario-de-conceptos/videoconferencia-que-se-supone-que-es-el-protocolo-rtcp/>

Dos Santos , E. A. (21 de Mayo de 2010). *3CX*. Obtenido de ¿Qué es RTP – Real Time Transport Protocol?: <https://www.3cx.es/voip-sip/rtp/>

Dos Santos , E. A. (21 de Mayo de 2010). *3CX*. Obtenido de ¿Qué es RTCP – Real Time Transport Protocol?: <https://www.3cx.es/voip-sip/rtcp/>

Duce, D., Gomes, M. R., Hopdog, F. A., & Lee, J. R. (6 de Diciembre de 2012). *User Interface Managment and Design*. Lisbon, Portugal. Obtenido de <https://books.google.com.ec/books?id=McOqCAAAQBAJ&pg=PA87&dq=user+interfac e+framework&hl=en&sa=X&ved=0ahUKEwipzufAgJbUAhWleCYKHQWSCUoQ6AE INTAE#v=onepage&q=user%20interface%20framework&f=false>

Falla, S. A. (26 de 10 de 2007). *Maestros del Web*. Obtenido de Historia y evolución del Sistema Operativo Mac OS: <http://www.maestrosdelweb.com/historia-y-evolucion-del-sistema-operativo-mac-os/>

fergarcia. (25 de Enero de 2013). *fergarcia WordPress*. Obtenido de Entorno de Desarrollo Integrado: <https://fergarcia.wordpress.com/2013/01/25/entorno-de-desarrollo-integrado-ide/>

Fores, M. (22 de Septiembre de 2016). *Advancedwebranking*. Obtenido de A Beginner's Guide to HTTP/2 and its Importance: <https://www.advancedwebranking.com/blog/beginners-guide-to-http2/>

García, G. L. (2005). *metabiblioteca*. Obtenido de <http://libros.metabiblioteca.org/handle/001/224>

Genaro, V. (13 de Noviembre de 2013). *TriPlet: A Conceptual Framework for Multidimensional Adaptation of User Interfaces to the Context of Use*. Louvain-la-Neuve, Walloon Brabant, Bélgica. Obtenido de <https://books.google.com.ec/books?id=rqgRAwAAQBAJ&pg=PA43&dq=user+interface>

+framework&hl=en&sa=X&ved=0ahUKEwipzufAgJbUAhWYeCYKHQWSCUoQ6AEI
KTAB#v=onepage&q=user%20interface%20framework&f=false

Gonzales, G. (29 de Julio de 2014). *Hipertextual*. Obtenido de ¿Qué es el protocolo XMPP y dónde se usa?: <https://hipertextual.com/archivo/2014/07/protocolo-xmpp/>

Google. (22 de Junio de 2011). *Estándares de Codificación*. Obtenido de Google Docs: https://docs.google.com/document/d/1rbxDFM0zsbFDNRZeM2FoXfRDbYSiSt6tCdbYP-A0qdzs/edit?hl=en_US#bookmark=id.f107a2341154

Google. (23 de Noviembre de 2016). *Google Cloud Messaging*. Obtenido de Google Inc Site: <https://developers.google.com/cloud-messaging/gcm>

Google. (13 de Septiembre de 2016). *Google Cloud Messaging*. Obtenido de Google Developers: <https://developers.google.com/cloud-messaging/server>

Google. (3 de Marzo de 2017). Android Studio. *Android Studio*. Silicon Valley, California, USA.

Google. (Marzo de 2017). *Google Developers*. Obtenido de Google Cloud Messaging: Overview: <https://developers.google.com/cloud-messaging/gcm>

Hernández, H. (17 de 11 de 2016). *Malavida*. Obtenido de La historia de Android: de Apple Pie 1.0 a Nougat 7.0: <http://www.malavida.com/es/analisis/la-historia-de-android#gref>

IETF. (14 de Abril de 2017). *Hypertext Transfer Protocol (HTTP) Status Code Registry*. Obtenido de IETF: <http://www.ietf.org/assignments/http-status-codes/http-status-codes.xml>

Joskowicz, I. J. (10 de Febrero de 2008). *Universidad de Vigo*. Obtenido de <http://iie.fing.edu.uy/~josej/docs/XP%20-%20Jose%20Joskowicz.pdf>

Kioskea. (Junio de 2014). *CMM*. Obtenido de Protocolo TCP: <http://es.ccm.net/contents/281-protocolo-tcp>

- Latorre, G. (22 de Marzo de 2010). *Programacion II*. Obtenido de JVM - JDK - JRE - Conceptos Fundamentales de la P.O.O.: <http://gl-epn-programacion-ii.blogspot.com/2010/03/jvm-jdk-jre-conceptos-fundamentales-de.html>
- Linares, N. A. (Septiembre de 2014). *Sistema Web-Móvil para llevar el seguimiento de tratamientos médicos en pacientes*. Obtenido de Universidad de JAEN, Escuela Politécnica Superior de Linares: <http://tauja.ujaen.es/bitstream/10953.1/1800/1/TFG-Nicolas.%20A-Serrano-Linares.pdf>
- Lopez, E. (Enero de 2015). *Sistios Argentina*. Obtenido de Pc Magazine : https://www.sitiosargentina.com.ar/webmaster/cursos%20y%20tutoriales/que_es_la_mesajeria_instantanea.htm
- McIver McHoes, A., & Ida, F. (2011). *Sistemas Operativos*. CENGAGE Learning.
- MikeKavis. (7 de Mayo de 2013). *Kavis Technology Consulting*. Obtenido de RESTful Services: The Key to cloud Computing: <http://www.kavistechnology.com/blog/restful-services-the-key-to-cloud-computing/>
- Miranda, J. (03 de Octubre de 2002). *iuma*. Obtenido de Breve historia de Linux: https://www.iuma.ulpgc.es/users/jmiranda/docencia/libro_ada/libro_ada_html/node133.htm
- Nava, E. (10 de 01 de 2015). *Pushy*. Obtenido de <https://eladnava.com/pushy-a-new-alternative-to-google-cloud-messaging/>
- ORACLE. (Abril de 2012). *Your First Cup*. Obtenido de documentación ORACLE: <http://docs.oracle.com/javaee/6/firstcup/doc/gcrky.html#gcroc>
- ORACLE. (January de 2013). *The Java EE 6 Tutorial*. Obtenido de docs oracle: <http://docs.oracle.com/javaee/6/tutorial/doc/gijqy.html>

- Orero, R. E. (Junio de 2016). Desarrollo de una aplicación móvil para la gestión de itinerancia en dispositivos Android. Santander, España. Obtenido de <https://repositorio.unican.es/xmlui/bitstream/handle/10902/9202/Barcena%20Orero%20Rebeca%20Elena.pdf?sequence=1&isAllowed=y>
- Pawlan, M. (23 de Marzo de 2001). *Introduction to the J2EE Architecture*. Obtenido de Personal Monica Pawlan Blog: <http://pawlan.com/monica/articles/j2earch/>
- Pedra, M. (15 de Enero de 2016). *MarceloPedra*. Obtenido de Listado de Puertos : <http://www.marcelopedra.com.ar/blog/2009/08/09/tabla-de-puertos-tcp/>
- Quaine, N. (2007). *SoapUser*. Obtenido de SOAP Basics: <http://www.soapuser.com/basics1.html>
- Rabasco. (14 de 6 de 2015). *espol*. Obtenido de Historia y evolución de Windows Phone: <http://blog.espol.edu.ec/rabasco/2015/06/14/historia-y-evolucion-de-windows-phone/>
- RedHat. (16 de Julio de 2016). *Red Hat Developer Program*. Obtenido de Red Hat JBoss Enterprise Application Program: <https://developers.redhat.com/products/eap/overview/?referrer=jbd>
- RedHat. (s.f.). *RedHat*. Obtenido de Jboss Installation Guide: https://access.redhat.com/documentation/en-US/Red_Hat_JBoss_Portal/5.1/html/Installation_Guide/Pre_Requisites-Hardware_and_Operating_System_Requirements.html
- Roig, X. C. (10 de 04 de 2012). *fundéUBBVA*. Obtenido de <http://www.fundeu.es/escribireninternet/el-origen-de-la-comunicacion-instantanea/>
- Rouse, M. (Septiembre de 2005). *TechTarget*. Obtenido de <http://searchnetworking.techtarget.com/definition/ARPA>

Rouse, M. (Agosto de 2006). *Search Networking*. Obtenido de ACK:
<http://searchnetworking.techtarget.com/definition/ACK>

Rus, C. (13 de 10 de 2016). *Applesfera*. Obtenido de La evolución de iOS desde sus orígenes: una carrera para ser el mejor sistema operativo móvil de la historia:
<https://www.applesfera.com/ios/la-evolucion-de-ios-desde-sus-origenes-una-carrera-para-ser-el-mejor-sistema-operativo-movil-de-la-historia>

Significados . (1 de Junio de 2017). *Significados*. Obtenido de Significado de Wifi:
<https://www.significados.com/wifi/>

SMAK, M. (17 de Marzo de 2012). *SlideShare*. Obtenido de
<https://www.slideshare.net/MrSMAK/extreme-programming-12047889>

Temas Tecnologicos . (20 de Abril de 2017). *Temas Tecnologicos de Interes* . Obtenido de Conceptos Básicos de Redes Móviles – ¿Qué necesito saber?:
<http://www.temastecnologicos.com/>

Tendencias MAG. (12 de Diciembre de 2016). *Tendencias Mag*. Obtenido de SDK: ¿Qué es y para qué sirve?: <http://www.tendenciasmag.com/sdk-que-es-y-para-que-sirve/>

Universidad de Alicante. (11 de Agosto de 2003). *Curso Abierto j2ee* . Obtenido de jtech Universidad de Alicante: <http://www.jtech.ua.es/j2ee/2003-2004/abierto-j2ee-2003-2004/sa/sesion1-apuntes.htm>

Universidad Internacional de Valencia. (25 de Agosto de 2016). *VIU*. Obtenido de Qué es y cómo funciona el protocolo ip: <http://www.viu.es/funciona-protocolo-ip/>

Valdes Perez, D. (26 de Octubre de 2007). *Maestros del Web*. Obtenido de Que son las bases de datos: <http://www.maestrosdelweb.com/que-son-las-bases-de-datos/>

Wells, J. D. (2001). *sourceforge*. Obtenido de Aplicación de eXtreme programming en ONess:
<http://oness.sourceforge.net/proyecto/html/ch05s02.html>