



# **PAUMIM**

## **Plataforma de Acceso Universal a Mensajería Instantánea Móvil**



**Darío Andrés Caicedo Moncayo**  
**Edwin Figueroa Losada**

*Universidad del Cauca*

**Facultad de Ingeniería Electrónica y Telecomunicaciones**

**Departamento de Telemática**

**Línea: Servicios y Aplicaciones de Internet**

**Popayán, Febrero de 2006**



# **PAUMIM**

## **Plataforma de Acceso Universal a Mensajería Instantánea Móvil**



Trabajo de Grado para optar al Título de Ingeniero en Electrónica y  
Telecomunicaciones

**Darío Andrés Caicedo Moncayo**  
**Edwin Figueroa Losada**

Director: Oscar Mauricio Caicedo

*Universidad del Cauca*

Facultad de Ingeniería Electrónica y Telecomunicaciones

**Departamento de Telemática**

**Línea: Servicios y Aplicaciones de Internet**

**Popayán, Febrero de 2006**



## Tabla de Contenido

<b>INTRODUCCIÓN.....</b>	<b>6</b>
<b>1 CAPITULO 1: MENSAJERIA INSTANTÁNEA .....</b>	<b>8</b>
1.1 Conceptos Generales.....	8
1.1.1 Contactos .....	8
1.1.2 Chat.....	8
1.1.3 Intercambio de archivos.....	9
1.1.4 Otros .....	9
1.2 Arquitectura .....	10
1.2.1 Componentes Básicos.....	10
1.2.1.1 Arquitectura Cliente – Servidor .....	10
1.2.1.2 Arquitectura Peer to Peer .....	11
1.2.2 Modo de operación.....	12
1.3 Mercado .....	13
1.3.1 Proveedores Desktop .....	13
1.3.1.1 ICQ .....	13
1.3.1.2 Yahoo! Messenger.....	14
1.3.1.3 MSN Messenger.....	14
1.3.1.4 AIM (AOL Instant Messenger).....	14
1.3.1.5 Google Talk .....	15
1.3.2 Grupos Consumidores.....	16
1.4 Normatividad y Regulación.....	18
1.4.1 Internacional.....	18
1.4.2 Nacional .....	18
1.5 Mensajería Instantánea Móvil.....	18
1.5.1 MSN y Vodafone .....	19
1.5.2 Yahoo!, AOL y Cingular .....	20
1.5.3 O2 Germany: Mobile ICQ Service.....	20
1.5.4 Mobber .....	20
1.6 Futuro.....	20
<b>2 CAPITULO 2: EL PROTOCOLO JABBER .....</b>	<b>22</b>
2.1 Jabber .....	22
2.1.1 Modelo de Mensajería [24] .....	25
2.1.2 Protocolo de Mensajería [25] .....	27
2.1.3 Protocolo de Presencia [25].....	29



2.1.4	Protocolo de Grupo de Chat [11] .....	31
2.1.5	Protocolo de Pregunta/Información (IQ) [11].....	31
2.1.6	Protocolo de Registro [11] .....	33
2.1.7	Protocolo de Autenticación [11] .....	34
2.1.8	Protocolo Roster [25].....	38
2.1.9	Transportes entre Jabber y otros Servidores de Mensajería Instantánea [11]	41
<b>3 CAPITULO 3: DESCRIPCIÓN DE LA PLATAFORMA DE ACCESO UNIVERSAL A MENSAJERIA INSTANTÁNEA MÓVIL (PAUMIM).....</b>		<b>42</b>
3.1	Planteamiento del Problema.....	42
3.2	Requerimientos para un sistema de Mensajería Instantánea Móvil .....	43
3.2.1	Cliente de mensajería instantánea móvil .....	43
3.2.2	Servidor de Mensajería Instantánea Móvil .....	44
3.3	Descripción de la solución .....	44
3.3.1	Módulo Administrativo .....	45
3.3.2	Módulo de Conexión Móvil.....	46
3.3.3	Módulo de Interoperabilidad .....	46
3.3.3.1	Servicios Jabber .....	46
3.3.3.2	Modulo de Transportes .....	47
3.3.4	Modulo de Control Central .....	47
3.3.4.1	Control.....	47
3.3.4.2	Tarificación .....	47
3.3.4.3	Gestión de Usuarios .....	47
3.4	Glosario.....	47
3.4.1	Actores .....	47
3.4.2	Vistas .....	48
3.5	Modelo de Casos de Uso .....	49
3.6	Diseño Detallado de Caso de Uso Iniciar Chat .....	50
3.6.1	Diagrama de Caso de Uso.....	50
3.6.2	Diagrama de Clases del Cliente Móvil .....	53
3.6.2.1	Diagrama de Clases del Cliente Móvil MIDP 1.0.....	53
3.6.2.2	Diagrama de Clases del Cliente Móvil MIDP 2.0.....	54
3.6.3	Diagrama de Clases del Servidor .....	56
3.6.3.1		56



3.6.4	Diagramas de Secuencia.....	56
3.6.5	Diagrama de Secuencia de Cliente Móvil .....	56
3.6.6	Diagrama de Secuencia del Servidor.....	57
3.6.7	Diagrama de Paquetes.....	57
3.6.8	Descripción de Paquetes.....	59
3.6.8.1	Servidor .....	59
3.6.8.2	Móvil.....	62
3.6.9	Diagrama de Implantación.....	62
3.6.10	Características Técnicas.....	63
3.6.10.1	Teléfono Móvil .....	63
3.6.10.2	Servidor PAUMIM .....	63
3.6.10.3	Data Base.....	64
<b>4</b>	<b>CAPITULO 4: EXPERIMENTACIÓN .....</b>	<b>65</b>
4.1	Mensajería Móvil con PAUMIM.....	65
4.2	Entorno de Ejecución .....	65
4.2.1	Hardware.....	65
4.2.1.1	Servidor .....	65
4.2.1.2	Móvil.....	65
4.2.2	Software .....	66
4.2.2.1	Servidor .....	66
4.2.2.2	Móvil.....	66
4.3	Cliente Móvil.....	67
	Procedimiento de implementación .....	67
4.3.1	Cliente MIDP1.0 .....	68
4.3.2	Cliente MIDP2.0 .....	68
4.3.3	Resultados .....	69
4.4	Módulo de Conexión Móvil .....	73
4.4.1	Módulo de Interoperabilidad .....	75
4.5	Análisis de costos de la plataforma .....	78
4.5.1	Costos de mantenimiento y administración.....	78
4.6	Mensajería Instantánea Móvil Tradicional.....	79
4.7	PAUMIM Vs SMS .....	80
4.7.1	Precio.....	80
	Con lo anterior se concluye que el precio por envío de mensajes en la plataforma PAUMIM es mucho más bajo con relación a SMS. ....	81



4.7.2	Acceso al servicio.....	81
4.7.2.1	SMS.....	81
4.7.2.2	PAUMIM .....	81
4.7.3	Interoperabilidad.....	82
<b>5</b>	<b>Conclusiones, Recomendaciones y Trabajos Futuros .....</b>	<b>83</b>
5.1.1	Conclusiones del Proyecto .....	83
5.1.2	Conclusiones Tecnológicas .....	84
5.1.3	Recomendaciones.....	85
5.1.4	Trabajos Futuros .....	85
<b>6</b>	<b>Referencias.....</b>	<b>86</b>
[32]	An Introduction to the Twisted Networking Framework. Itamar Shtull-Trauring. Enero 15 de 2004.....	90
<b>7</b>	<b>Glosario.....</b>	<b>91</b>
	<b>Plataforma de Acceso Universal a Mensajería Instantánea Móvil .....</b>	<b>92</b>
	<i>Universidad del Cauca.....</i>	<i>92</i>
	<b>Plataforma de Acceso Universal a Mensajería Instantánea Móvil .....</b>	<b>93</b>
	<i>Universidad del Cauca.....</i>	<i>93</i>

## Lista de Figuras

Figura 1.1	Arquitectura IM, Cliente-Servidor .....	11
Figura 1.2	Arquitectura IM, peer to peer.....	12
Figura 1.3	Secuencia de eventos del funcionamiento de la mensajería instantánea .....	13
Figura 2.1	Modelo básico de Mensajería Instantánea .....	25
Figura 3.1.	Arquitectura de PAUMIM .....	45
Figura 3.13.	Diagrama de Casos de Uso, vista de los actores .....	49
Figura 3.14.	Diagrama de Casos de Uso, vista del administrador .....	49
Figura 3.11.	Diagrama de Casos de Uso, vista del usuario.....	50
Figura 3.12.	Diagrama de Casos de Uso, vista del servidor .....	50
Figura 3.13.	Diagrama de Casos de Uso a implementar .....	51
Figura 3.13.	Diagrama de Clases del Cliente Móvil MIDP1.0 .....	54
Figura 3.14.	Diagrama de Clases del Cliente Móvil MIDP2.0 .....	55
Figura 3.15.	Diagrama de Clases del Cliente Servidor.....	56
Figura 3.16.	Diagrama de Secuencia del cliente móvil emisor .....	56



Figura 3.17 Diagrama de Secuencia del cliente móvil receptor .....	57
Figura 3.18. Diagrama de Secuencia del cliente móvil receptor .....	57
Figura 3.19 Diagrama de paquetes del servidor.....	58
Figura 3.20. Diagrama de paquetes del cliente móvil .....	59
Figura 3.21. Diagrama de implantación del sistema .....	62
Figura 4.1. Diagrama de pruebas de la plataforma .....	67
Figura 4.2. Consumo de memoria para el cliente MIDP 1.0 .....	70
Figura 4.3. Consumo de memoria para el cliente MIDP 2.0 .....	70
Figura 4.4. Consumo de memoria Cliente MIDP 1.0 .....	70
Figura 4.5. Consumo de memoria Cliente MIDP 2.0 .....	71
Figura 4.6. Tiempo de respuesta para el cliente MIDP 1.0.....	71
Figura 4.7. Tiempo de respuesta para el cliente MIDP 2.0.....	72
Figura 4.8. Tamaño de los mensajes para el Cliente Móvil (MIDP 1.0 y MIDP 2.0).....	73
Figura 4.9. Consumo Cliente MIDP 1.0 por concepto de sondeo .....	73
Figura 4.10 Diagrama de colaboración, inicio de sesión para Cliente MIDP1.0.....	74
Figura 4.11 Diagrama de colaboración, conexión e inicio de sesión para cliente MIDP2.0 .....	75
Figura 4.12 Uso de la CPU servidor Ejabberd.....	77
Figura 4.13 Consumo de memoria servidor Ejabberd .....	77

### **Lista de Tablas**

Tabla 1.1. Uso de las aplicaciones de Mensajería Instantánea .....	16
Tabla 1.2. Quiénes usan la mensajería instantánea.....	17
Tabla 2.1 Errores comunes del núcleo de Jabber .....	29
Tabla 4.1. Funcionalidades disponibles de los transportes Jabber.....	78
Tabla 4.2. Herramientas Hardware y Software para montar PAUMIM.....	78
Tabla 4.3 Gastos administrativos .....	79
Tabla 4.4. Precio de SMS para operadores nacionales de telefonía móvil.....	80
Tabla 4.5. Características del plan de datos Datos 10 .....	80
Tabla 4.6. Comparación de precios entre SMS y PAUMIM .....	81



## INTRODUCCIÓN

La gran mayoría de personas que ingresan a un café Internet o a la oficina de alguna empresa, se encuentra con un silencio generalizado en todo el recinto, sólo interrumpido por el ruido de las teclas de los computadores o una que otra pequeña risa. Pero la realidad es que estas personas se están comunicando con otras, mediante la mensajería instantánea, un servicio que ha revolucionado las telecomunicaciones del mundo al integrar las ventajas y funcionalidad del correo electrónico y el Chat [1].

Inicialmente se creía que la mensajería instantánea tendría acogida solamente dentro de las charlas cotidianas de ocio, como generalmente se usa el Chat; pero se ha convertido en una de las herramientas de trabajo primordiales en el marco empresarial, sin dejar de lado su utilidad como servicio lúdico de comunicación; teniendo como características principales, la agilidad de comunicación y los reducidos costos.

Es tan amplia su acogida y tan importante en la vida de los usuarios de Internet, que la IM (Instant Messaging), iguala e incluso supera el uso del servicio de correo electrónico, en un creciente porcentaje de la población [2].

Las ventajas del servicio de mensajería instantánea se pueden evidenciar de forma clara en ambientes empresariales y de entretenimiento. Al interior de una empresa, es vital contar con herramientas que permitan llevar a cabo la comunicación interactiva entre los trabajadores, compartiendo recursos, conocimientos e incrementando la agilidad laboral; y además, sirviendo de alternativa complementaria a los servicios de comunicación actuales como la telefonía convencional o móvil. En los últimos años, la utilización de este servicio ha cobrado mucha importancia en el campo del entretenimiento, ofreciendo a los usuarios un medio inmediato para ubicar y comunicarse con la familia, compañeros y amigos.

En la actualidad, existe un gran número de proveedores de servicios de mensajería instantánea, que prestan un excelente y completo servicio a sus usuarios pero que no permiten comunicarse con otros proveedores de mensajería. Además de esto, los usuarios hacen utilización de este servicio accediendo por medio de computadores de escritorio con conexión a Internet, lo que los limita en su movilidad.





Por ello, se crea la necesidad de brindar a los usuarios de mensajería instantánea un sistema que les permita comunicarse de forma libre con los proveedores del servicio, y además, brinde la facilidad de acceder desde cualquier lugar mediante dispositivos móviles. PAUMIM, es la Plataforma de Acceso Universal a Mensajería Instantánea Móvil, proyecto de grado desarrollado para dar solución al conjunto de limitaciones actuales del servicio de mensajería instantánea.

Para abordar el proyecto, iniciamos con una descripción de la Mensajería Instantánea en el Capítulo 1, en donde se tocan temas muy importantes como su definición, historia, mercado y regulación. En el Capítulo 2 se trata a fondo el protocolo de mensajería Jabber que es la base de la construcción de PAUMIM. En el capítulo 3 se describe la Plataforma en sí, resaltando su funcionalidad, análisis, y diseño. En el capítulo 4 se muestran los resultados obtenidos con las pruebas efectuadas a la plataforma y por último, en el capítulo 5 se muestran las conclusiones tecnológicas, del proyecto y los posibles trabajos futuros para continuar la investigación alrededor de la mensajería instantánea y sus infinitas posibilidades.



# 1 CAPITULO 1: MENSAJERIA INSTANTÁNEA

## 1.1 *Conceptos Generales*

Los programas de mensajería instantánea nos ofrecen una serie de características propias, las cuales lo convierten en un servicio muy singular que aporta nuevas funciones y ventajas para los usuarios, superando a la gran mayoría de los servicios comunicaciones existentes. A continuación se ven algunas de las características generales del servicio:

### 1.1.1 **Contactos**

Este es uno de los conceptos más importantes de la mensajería instantánea, siendo la razón de ser del servicio, un contacto es la representación de una persona o usuario, el cual, en algún momento podría establecer una comunicación con otro usuario del servicio. Los contactos manejan un concepto denominado presencia, del cual se puede deducir, que nos informa de la disponibilidad actual de un contacto para comunicarse. Estos estados generalmente son: disponible, no disponible, vuelvo enseguida, invisible, ocupado, entre otros.

La gestión de los contactos está en manos de usuario del servicio, el cual, para adicionar estos a su lista personal requiere de un acuerdo de ambas partes, debido a que la hora de hacer el registro de estos, se envía una solicitud al usuario que se desea incluir en la lista, siendo esta aceptada o rechazada de acuerdo al criterio del contacto solicitado.

Como características generales, los contactos usan un nombre y una imagen de identificación personal, las cuales sirven como caracterización individual de cada usuario, y estos se pueden agrupar según la conveniencia de cada usuario, en conjuntos familiares, de trabajo, de amigos, etc. [5].

### 1.1.2 **Chat**

El tan conocido intercambio de mensajes entre usuarios, el cual hereda la mensajería instantánea de las salas de Chat, es el principal argumento del crecimiento del servicio dentro de Internet. Sin embargo, la verdadera ventaja del servicio está en la capacidad de



mantener charlas solamente con las personas que el usuario desea, es decir, sus contactos.

Cuando un usuario ingresa a su sesión de MI, de lo primero que se entera es del estado de presencia de sus contactos, obteniendo así, información acerca de cuales usuarios están disponibles para establecer una comunicación en ese momento. Esta información de la presencia de los contactos, es actualizada continuamente, informando al usuario de los cambios de estado de toda su lista de contactos [6].

### **1.1.3 Intercambio de archivos**

Adicionalmente al envío de mensajes entre usuarios, los sistemas de mensajería instantánea permiten a los usuarios intercambiar archivos. Actualmente, los sistemas transfieren ficheros directamente entre usuarios, sin tener que pasar a través de algún servidor [7]. En otras palabras la técnica peer-to-peer (que se verá con más detalle en el apartado 1.3.1.2), es siempre usada para la transferencia de archivos, con el fin de eliminar las altas demandas de ancho de banda, que un servidor centralizado requeriría para este propósito.

Actualmente, ninguno de los más importantes sistemas de mensajería instantánea cifra los archivos que transfiere entre clientes IM. Si dos usuarios de una misma red corporativa, intercambian un archivo, este probablemente se mantendrá sobre esta, sin embargo, si uno de usuarios está afuera de la red, los archivos serán enviados a través de la Internet sin cifrar. Convirtiéndose en un gran problema de seguridad para la IM, en el cual los archivos que viajan a través de la Internet están a disposición de los husmeadores de la red.

### **1.1.4 Otros**

En respuesta a la gran competencia dentro del mercado de la mensajería instantánea, algunas compañías han agregado funcionalidad adicional a los clientes de mensajería para aumentar sus consumidores. Como ejemplo podríamos enumerar algunas de las nuevas características más relevantes dentro del servicio de la mensajería instantánea:



- **Video:** Las empresas proveedoras mensajería instantánea, han integrado dentro de sus clientes IM la posibilidad de enviar y recibir video desde cada uno de sus usuarios. Yahoo! fue el primero de los gigantes proveedores de IM que integró la transmisión de video dentro de sus clientes de mensajería instantánea [8], sin embargo, en este momento los clientes IM más populares, tienen nuevas capacidades adicionales al tradicional video-chat, como es el caso de AOL, que ofrece videos musicales, cortos de películas, noticias de entretenimiento y primicias deportivas [9].
- **Voz sobre IP:** Esta es otra característica importante que se ha agregado al servicio de mensajería instantánea la cual permite comunicación de voz en tiempo real entre usuarios IM; algunos clientes como Google Talk (ver sección 1.4.1) ofrecen este servicio, y recientemente Yahoo! anunció que planea actualizar su mensajero, con una nueva capacidad, que le permite a sus usuarios hacer llamadas hacia la red telefónica fija o móvil [10].

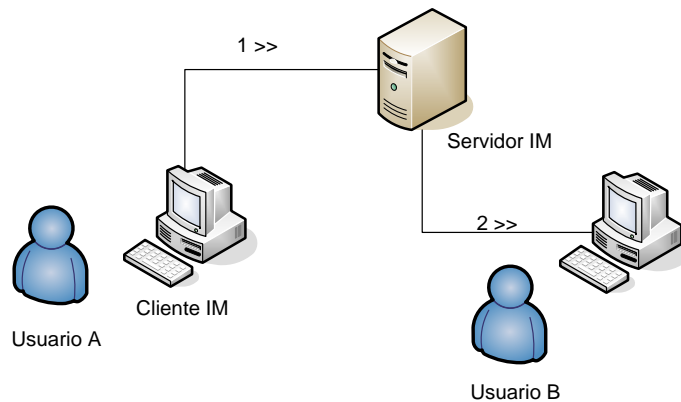
## 1.2 *Arquitectura*

### 1.2.1 Componentes Básicos

#### 1.2.1.1 Arquitectura Cliente – Servidor

Virtualmente todos los sistemas IM emplean la misma arquitectura básica cliente-servidor. Los usuarios instalan sus clientes de mensajería instantánea sobre sus computadores de escritorio, dispositivos inalámbricos, o PDAs (Personal Digital Assistance), los cuales se conectan a un servidor IM del proveedor, encargado de localizar a otros usuarios e intercambiar mensajes entre ellos. En la mayoría de los casos, los mensajes no son enviados directamente desde el computador del usuario emisor a el computador del receptor, porque estos son enviados primero a un servidor IM, y entonces desde este servidor se envía al destinatario, como se ve en la Figura 1.1.

En la mayoría de los sistemas de mensajería instantánea cliente-servidor, el intercambio de datos entre usuarios es fácilmente visible, haciéndolo susceptible a espías escondidos [7].

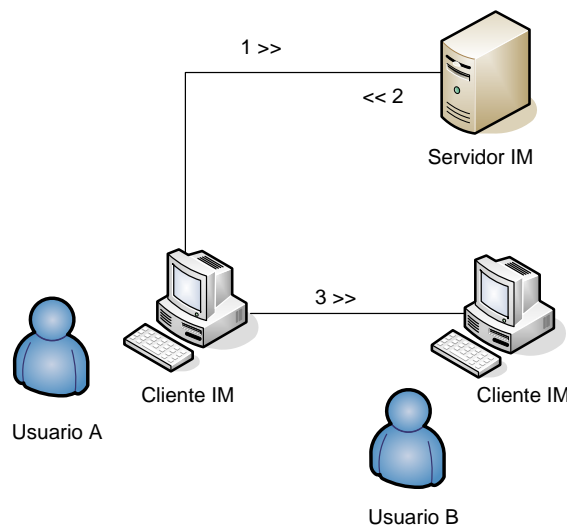


**Figura 1.1 Arquitectura IM, Cliente-Servidor**

### 1.2.1.2 Arquitectura Peer to Peer

Mientras la mayoría de los sistemas de mensajería instantánea usan servidores centralizados para transmitir todos los mensajes, algunos sistemas ofrecen mensajería peer-to-peer. En tales modelos, los clientes se conectan a un servidor para localizar a otros clientes, una vez localizados, estos se ponen en comunicación directa con el otro usuario, como se ve en la Figura 1.2.

El esquema peer-to-peer ofrece mejores características de seguridad que el esquema cliente-servidor cuando los dos clientes se encuentran sobre la misma red de área local, porque los mensajes no viajan sobre la red Internet. Sin embargo, si uno de los usuarios está localizado afuera de la red corporativa, los mensajes están expuestos a posibles espías escondidos, al igual que en el esquema cliente-servidor [7].





## Figura 1.2 Arquitectura IM, peer to peer

### 1.2.2 Modo de operación

La mensajería instantánea permite el intercambio mensajes de texto, y archivos, los cuales se transfieren entre usuarios que han iniciado una sesión de comunicación entre ellos. El procedimiento en que se hace esta comunicación varía entre los diferentes sistemas existentes, pero en general el modo de funcionamiento es el siguiente [11]:

- Un usuario inicia su sesión conectándose a un servidor de mensajería instantánea, en el cual se encuentran almacenados sus datos personales y lista de contactos.
- Una vez el servidor verifica la validez del usuario con un login y una contraseña, se establece su estado de disponibilidad dentro del servidor, el cual a su vez informa de la presencia a todos los contactos pertenecientes, a la lista del usuario que acaba de ingresar y que se encuentran conectados al servidor IM.
- El servidor envía la lista de contactos al usuario que recientemente ha ingresado y junto con esta, envía el estado de cada uno de ellos.
- A partir de ese momento, para establecer una comunicación, el usuario solamente debe elegir de su lista de contactos disponibles, al usuario con el cual desea establecer una charla.
- Cuando el usuario cierra su sesión de mensajería instantánea, el cliente IM informa al servidor IM de la terminación de la sesión del usuario; en el momento en que el servidor IM se entera de este evento, informa a todos los contactos conectados del usuario, acerca de la terminación de sesión y por lo tanto del cambio en la presencia de este. Acción similar a la que hizo el servidor de mensajería instantánea al inicio de la sesión. En la Figura 1.3 se puede ver esta interacción más claramente.

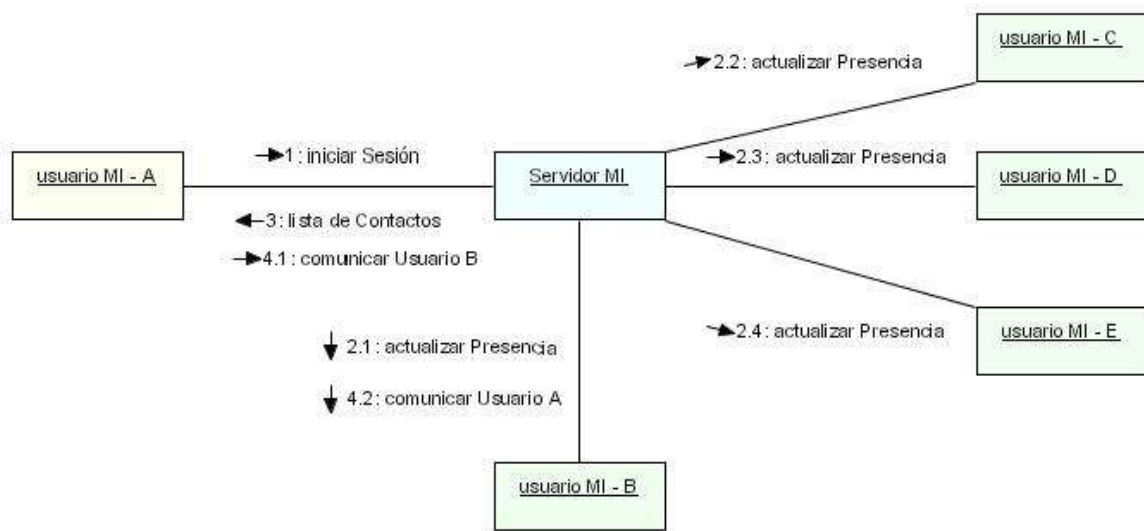


Figura 1.3 Secuencia de eventos del funcionamiento de la mensajería instantánea

## 1.3 Mercado

### 1.3.1 Proveedores Desktop

#### 1.3.1.1 ICQ

I seek you, como se pronuncia, o, lo que es lo mismo, te busco, fue la primera aplicación de mensajería instantánea y en ella se han inspirado las nuevas herramientas que conocemos hoy en día. Al igual que AOL Messenger, ICQ es también propiedad de America Online, que los distribuye por separado y son incompatibles entre sí. Incluye opciones curiosas ya en desuso en otros sistemas que agilizan la conversación a cambio de provocar situaciones comprometidas: envía cada carácter tecleado, en vez de remitir la frase entera, con lo que los errores quedan al descubierto.

ICQ fue el primer sistema de mensajería instantánea que existió en el mercado y, como dato curioso, sus versiones nunca han dejado de ser beta o de prueba. La última versión disponible, además de funciones como ICQphone que incorpora telefonía IP para hacer llamadas PC-a-PC o PC-a-teléfono, tecnología SMS (Short Message Service) para enviar mensajes a localizadores y dispositivos inalámbricos e integración con Outlook. Además, ofrece una nueva colección de iconos o "emoticons" que facilitan el envío de mensajes y



las intenciones de comunicación, además de una mayor integración con las herramientas de comunicación de AOL, empresa a la que pertenece la herramienta. Este instrumento también ha mejorado su sistema de operación dentro de redes que cuentan con Firewalls, o recursos de seguridad de la red, lo mismo que su integración con el sistema operativo Windows XP y las prestaciones para bloquear la recepción de correo no deseado, entre otras características [1].

### **1.3.1.2 Yahoo! Messenger**

El Messenger de Yahoo! es una de las herramientas de mensajería instantánea más completas y extendidas. Además del envío y recepción de mensajes y archivos, y la capacidad multimedia de audio y vídeo, incluye noticias, cotizaciones bursátiles y previsiones meteorológicas de todo el mundo [1].

### **1.3.1.3 MSN Messenger**

Lo emplean 130 millones de personas en todo el mundo y se ubica segundo en Estados Unidos con más de 15.7 millones de usuarios únicos al mes de mayo, a diferencia de países como México donde es el número uno con más de 3.8 millones de usuarios, MSN Messenger de Microsoft es un sistema eficiente cuya principal ventaja, además de su sencillez de uso, es su integración con Hotmail y MSN, la red de contenido de Microsoft, con la estrategia .Net [1].

### **1.3.1.4 AIM (AOL Instant Messenger)**

Es la más importante herramienta de mensajería instantánea en Estados Unidos, pero fuera de Estados Unidos la popularidad de la herramienta varía. Permite configurar alertas para notificar que sus amigos se han conectado, también ofrece la opción de compartir la lista de contactos con otros usuarios. Entre las prestaciones avanzadas están la de llamar a los contactos amigos y conversar de PC a PC o de PC a teléfono, para hacerlo requiere un proveedor que soporte el servicio. También permite configurar alertas para correo electrónico y leer mensajes de correo de cualquiera de las cuentas del usuario. Un aspecto que distingue al servicio de otros es la posibilidad de usar iconos de Amigos para animar sus mensajes instantáneos (personajes animados, símbolos, e imágenes). En septiembre AOL incorporará en la versión 5.0 de su cliente de mensajería instantánea AIM más funcionalidades que permitirán que el usuario personalice a gusto la





interfaz de la aplicación, y que se conocerá comercialmente como 'AIM Expressions". La versión 5.0 agrega además otra funcionalidad que hace rato está presente en el MSN Messenger y Yahoo! Messenger: indicar cuando el usuario con el que se está charlando está escribiendo un mensaje [1].

### 1.3.1.5 Google Talk

Google Talk es un programa de mensajería instantánea el cual usa el protocolo Jabber/XMPP, fue desarrollado por Google y funciona bajo el sistema operativo Windows. La versión beta de Google Talk fue lanzada el 24 de agosto de 2005. A pesar de que Google todavía es un proveedor IM aún muy joven, el número de usuarios que hacen uso de su buscador, lo convierten en uno de los competidores con mayor potencial. Esta aplicación se sustenta bajo el protocolo de interoperabilidad de Jabber y XMPP. El servicio está disponible para los usuarios de Gmail, los cuales pueden adicionar a más usuarios con sus invitaciones.

El hecho de que Google haya utilizado Jabber como protocolo, ha sido una gran noticia para mucha gente, que piensa que esto servirá para que Jabber se extienda y aumente sus prestaciones [12].

Para finalizar esta sección, en la Tabla 1.1 se ven algunas cifras de las principales aplicaciones de mensajería instantánea que se encuentran en el mercado [2].

Proveedores	Total de usuarios individuales ( x 1000)	Porcentaje del alcance entre usuarios de Internet (%)	Promedio de minutos por visitante en un mes	Promedio diario de usuarios ( x 1000)
AOL Instant Messega (Servicio propietario)	25.090	16.0	279.8	5715
Yahoo! Messenger	22.135	14.1	423.7	5218
AOL Instant Messenger (servicio AiM)	21.363	13.6	388.7	5956



<b>MSN Messenger</b>	17.167	10.9	217.1	3865
----------------------	--------	------	-------	------

**Tabla 1.1. Uso de las aplicaciones de Mensajería Instantánea**

### 1.3.2 Grupos Consumidores

El Radicati Group publicó un estudio sobre el mercado de la mensajería instantánea para el período 2004 a 2008, en el cual informan lo siguiente [2]:

- El 85% de todas las empresas en Estados Unidos utilizan la mensajería instantánea, bien sea para uso personal o empresarial.
- El 20% de todas las empresas a nivel global usan la mensajería instantánea y se espera que esta cifra ascienda al 80% para el 2008

Al final del 2008 habrá 72 millones de usuarios corporativos de mensajería instantánea, al finalizar el 2008, 88% de los usuarios corporativos de mensajería instantánea dependerán de una red pública de mensajería instantánea.

En el mismo estudio se informó que las razones citadas por las empresas para desplegar soluciones de mensajería instantánea fueron:

- 44% Aumentar las comunicaciones entre oficinas
- 33% Reducir costos de llamadas telefónicas de larga distancia
- 11% Aumentar productividad
- 11% Complementar sistemas actuales de correo y teléfono

Además, un estudio de The Pew Internet & American Life Project Tracking Survey [2] encuentra que el 42% de los usuarios de Internet usan la mensajería instantánea como medio de comunicación. De esos 53 millones de adultos Americanos, 12% usan mensajería instantánea en un día normal.

Aunque la mayoría de los usuarios de Internet usan aún más frecuentemente el correo electrónico que la mensajería instantánea, alrededor de la cuarta parte de los usuarios está usando más frecuentemente los mensajes instantáneos que el correo electrónico.



Dentro de los lugares de trabajo, la mensajería instantánea está tomando cada vez más fuerza y se está convirtiendo en una herramienta necesaria para los propósitos empresariales tal como lo muestran las cifras de este estudio, en donde se encuentra que cerca del 21% de los usuarios IM, o aproximadamente 11 millones de adultos Americanos, usan la mensajería instantánea en el trabajo. Al mismo tiempo, 77% de los usuarios IM usan sus programas de mensajería instantánea en el hogar.

En la Tabla 1.2 se pueden ver algunos datos estadísticos, los cuales revelan la importancia que tiene la mensajería instantánea dentro de la sociedad actual.

	Porcentaje de los usuarios de Internet, los cuales son usuarios IM (%)	Proporción de la popularidad de la IM dentro de cada grupo (%)
<b>Sexo</b>		
Hombres	42	50
Mujeres	42	50
<b>Raza o Étnia</b>		
Blancos	41	73
Negros	44	8
Latinos	52	9
Otros	40	10
<b>Edades</b>		
Generación X (18 a 27 años)	62	31
Generación Y (28 a 39 años)	37	28
40 a 49 años	33	20
50 a 58 años	29	12
59 a 68 años	25	7
Más de 68 años	29	3

Tabla 1.2. Quienes usan la mensajería instantánea



## **1.4 Normatividad y Regulación**

### **1.4.1 Internacional**

En los Estados Unidos, las políticas de la no regulación de los servicios de Internet, de la FCC (Federal Communication Commission), han permitido que la industria de los servicios de información, crezca ampliamente dentro del mercado de las telecomunicaciones, sin intervención del gobierno. Este organismo define la mensajería instantánea como un servicio de información y de valor agregado [16], el cual por sus características propias, está libre del arbitraje de los organismos reguladores del sector.

### **1.4.2 Nacional**

Este es uno de los temas que causan mayor controversia dentro del sector de las telecomunicaciones, principalmente, como consecuencia de la convergencia. Podemos decir que los grandes cambios hacia una regulación convergente se han dado en los países más desarrollados, entre ellos, los Estados Unidos, algunos países de la Unión Europea y Japón. Sin embargo, en Colombia este tema se queda corto para afrontar los modelos de las telecomunicaciones y de los servicios telemáticos; y peor aún, la regulación nacional no define nada cierto para el futuro del sector.

Dentro del banco de datos de la CRT, Comisión de Regulación de Telecomunicaciones y el Ministerio de Comunicaciones, de la República de Colombia se encuentran: el Decreto Número 600 del 14 de Marzo del 2003 [17] y el Decreto 3055 del mismo año [18], en los cuales se expiden normas sobre los servicios telemáticos y de valor agregado, y se modifica el decreto 600 del 2003, respectivamente. Estos son los únicos documentos claros con los que cuenta la regulación de los servicios telemáticos en Colombia, ofrecen una visión muy general de la temática, y no se comenta nada acerca de las aplicaciones sobre Internet y mucho menos sobre la mensajería instantánea.

## **1.5 Mensajería Instantánea Móvil**



El actual modelo del mercado de las telecomunicaciones, establece nuevos retos para los proveedores de servicios, dentro de los cuales se menciona, la necesidad de “...aplicaciones que enriquezcan la calidad de vida de los usuarios. El interés de los suscriptores en servicios combinados que mejoran su calidad de vida ha venido en aumento en la medida en que un creciente número de empleados trabaja desde sus hogares, en oficinas remotas o cuando están en movimiento” [19].

Teniendo en cuenta las características de movilidad que el usuario solicita, la mensajería instantánea con acceso desde terminales móviles, se convierte en uno de los servicios con mayores expectativas dentro del contexto laboral y social, mejorando aún más, las características iniciales del servicio. Pero la solución que el usuario requiere no es solamente el acceso móvil al servicio, sino que espera contar con las mismas particularidades de comunicación que tiene en un entorno de escritorio, incluyendo el uso de los mismos perfiles y cuentas de usuario; de esta manera, la IM se convierte verdaderamente en un servicio integrado, y no solamente en un conjunto servicios similares, a los cuales se tienen acceso desde diferentes medios y dispositivos.

El problema más grande que afronta la IM es la falta de interoperabilidad entre los diferentes proveedores del servicio, los cuales día tras día, cierran los protocolos aún más, justificando como políticas de seguridad; donde el único perjudicado termina siendo el usuario del servicio, quien tiene que instalar un variado grupo de aplicaciones clientes para cada una de sus cuentas IM [20]. Sin embargo, se han establecido algunas alianzas entre algunos de los principales proveedores mensajería instantánea y de telefonía móvil del mundo.

### **1.5.1 MSN y Vodafone**

Planean lanzar el servicio de mensajería instantánea entre computadores personales y teléfonos móviles, en el cual sus usuarios pueden ver la presencia de sus contactos y intercambiar mensajes entre un computador con MSN Messenger y un teléfono móvil con Vodafone Messenger. Teniendo en cuenta los 155 millones de usuarios de Vodafone alrededor del mundo, esta es una de las alianzas más importantes del mercado de la IM [13].



### **1.5.2 Yahoo!, AOL y Cingular**

Otra alianza entre proveedores IM y un proveedor de servicio de telefonía móvil, este servicio ofrece las capacidades básicas de la mensajería instantánea: manejo de la lista de contactos, de la presencia y como característica particular, el usuario sólo paga por los mensajes enviados y recibidos, y no por el mantenimiento de la sesión IM [14].

### **1.5.3 O2 Germany: Mobile ICQ Service**

Es una extensión móvil del servicio fijo de ICQ, el cual fue desarrollado como un acuerdo entre el grupo mmO2 y AOL para desarrollar versiones móviles de sus aplicaciones de mensajería instantánea, en las cuales los usuarios móviles necesitan haberse registrado previamente como un usuario ICQ y la aplicación le permite interactuar con toda la comunidad ICQ, como lo hace desde su terminal de escritorio [15].

### **1.5.4 Mobber**

Es un sistema de Mensajería Instantánea propietario basado en el protocolo Jabber, en la actualidad se encuentra en desarrollo, no hay versiones con funcionalidad estable y la documentación es muy escasa. El sistema Mobber está compuesto por un servidor propietario y una aplicación Java de libre distribución para el cliente móvil, desde la cual se puede establecer comunicación con contactos de diferentes proveedores de mensajería instantánea, con la restricción de que el acceso se debe hacer desde una cuenta del servidor Mobber [8].

## **1.6 Futuro**

El uso del servicio de mensajería instantánea traerá el acercamiento de las personas y el incremento dramático de la eficiencia de las empresas; la mensajería instantánea pronto puede volverse tan indispensable como el correo electrónico y el teléfono. Sin embargo, este crecimiento del servicio de MI colocará una carga pesada sobre las infraestructuras de seguridad correspondientes. Incluso los pequeños puntos débiles dentro de estas redes de comunicaciones pueden tener enormes repercusiones económicas y sociales.

Teniendo en cuenta el incremento de la penetración de los sistemas de mensajería instantánea dentro de nuestros hogares y oficinas, uno de sus principales objetivos para



un futuro próximo es fortalecer los sistemas de seguridad montados sobre la infraestructura de red.

De acuerdo a la consultora independiente International Data Corporation (IDC), se prevé que habrá más de 450 millones de usuarios de MI en todo el planeta para fines del 2007 [21]. Pero los usuarios ya no demandan solamente intercambio de mensajes de texto para sus comunicaciones, sino que requieren otros servicios adicionales y por lo tanto dirigen la visión de la mensajería instantánea hacia un servicio convergente, tal como el mercado lo pide.

Sin embargo, la convergencia del mercado no sólo requiere que la mensajería instantánea ofrezca un creciente conjunto de funcionalidades, sino también, un acceso en cualquier momento y en cualquier lugar, lo cual enfoca el futuro del servicio hacia el campo de los dispositivos móviles y las comunicaciones inalámbricas. Teniendo en cuenta, el número creciente de teléfonos celulares, PDAs, Smart phones y mirando los directrices del protocolo de Internet versión 6 acerca del direccionamiento IP para dispositivos móviles, es fácil intuir el nuevo modelo de comunicaciones siempre en línea que se adoptará para el futuro próximo y como se esparcirá aún mas el uso de los mensajes de instantáneos como medio de comunicación primordial para el entorno empresarial.

Además de la penetración de la mensajería instantánea en el mundo de los dispositivos móviles, la necesidad de interoperabilidad entre los diferentes proveedores del servicio se convierte en uno de los objetivos principales para hacer de la MI un servicio universal de comunicaciones, el cual debe ser independiente de las redes, los protocolos y las marcas.

Finalmente, se espera ver al sector empresarial Colombiano haciendo uso, o mejor aún, aprovechando las ventajas que le ofrece la mensajería instantánea como medio de comunicación ágil y económico, ayudando a alcanzar sus metas y objetivos; logrando un aporte al cierre de la brecha digital a partir del uso adecuado de las tecnologías de la información y de las comunicaciones; y por lo tanto; una colaboración con el crecimiento de la sociedad de la información.



## 2 CAPITULO 2: EL PROTOCOLO JABBER

### 2.1 Jabber

Jabber es un conjunto de protocolos XML de flujos de descarga y de tecnologías que permite que dos entidades en Internet intercambien mensajes, presencia, y otra información estructurada en tiempos cercanos al real. Jabber se encuentra soportado en miles de servidores de Internet y es usado por más de 6 millones de personas en todo el mundo. Algunas de las características más relevantes de este protocolo se nombran a continuación [22]:

- **Abierto:** Cualquier persona puede implementar un servidor o un cliente, y actualmente se cuenta con una extensa lista de implementaciones disponibles.
- **Basado en texto plano:** Ya que está basado en mensajes XML puede ser interpretado por distintos sistemas operativos y distintas plataformas.
- **No centralizado:** Cualquier persona puede implantar un servidor Jabber en su dominio, y si lo desea podrá interoperar perfectamente con los otros servidores distribuidos sobre Internet.
- **Extensible:** Se pueden añadir diferentes componentes, los cuales funcionan sobre el protocolo original, y mejoran o aumentan la funcionalidad de este, con lo cual, no hay una limitación solamente a los mensajes instantáneos. Las extensiones, son manejadas por la Jabber Software Foundation.
- **Seguro:** Los servidores Jabber soportan SSL (Secure Socket Layer) para comunicaciones Cliente/Servidor y algunos clientes la extensión GPG (GNU Privacy Guard) para firmar la presencia y cifrar las comunicaciones punto a punto usando modelo asimétrico.
- **Interoperabilidad con otras redes:** Los transportes, que adicionados a los servidores Jabber permiten que los usuarios puedan acceder a los contactos de otros sistemas de Mensajería Instantánea. Existen transportes para MSN, ICQ, AOL y Yahoo, entre otros.

Además de lo anterior, Jabber está diseñado desde el principio para ser extendido conforme avanza el tiempo. Ejemplo de esa extensibilidad son los cambios y mejoras que





ha ido sufriendo paulatinamente sin romper la compatibilidad hacia atrás. Por citar algunos:

- Soporte de cifrado SSL y GPG
- Información más precisa sobre el estado del mensaje: se está escribiendo uno, el contacto lo ha recibido, se le ha mostrado o se ha guardado en el servidor.
- Salas de Charla: el sistema de multiconferencia más potente en la actualidad, sobrepasando al propio IRC (Internet Relay Chat), ha sido diseñado y programado como una extensión a Jabber en los últimos dos años y está disponible de forma independiente al servidor Jabber.
- Los transportes a otros sistemas de mensajería muestran las ventajas de la extensibilidad: si algo no existe, puede añadirse como una extensión.

Existe una gran red de servidores Jabber interconectados entre sí e independientes los unos de los otros. La mayoría de estos servidores son privados, en el sentido de que son mantenidos por personas o asociaciones particulares; aunque de acceso por parte de los clientes sea público, por lo cual, cualquier usuario puede usar sus servicios sin ninguna restricción.

Jabber es descentralizado por definición. No existe por tanto un servidor Jabber único que provea el servicio a todo el mundo. Ni siquiera un grupo de servidores único y controlado. Jabber es una red de servidores, independientes pero a la vez interconectados como el correo electrónico.

Por tanto, el usuario de Jabber no solo escoge su nombre, sino también el servidor donde crear la cuenta. Lo que podría parecer un inconveniente, resulta una de las ventajas claves de Jabber frente a otros sistemas de mensajería, a continuación algunas de ellas [23]:

- **Distribución de la carga:** Al haber servidores repartidos por todo el mundo, divididos por geografía, por temática o por proveedor la carga generada por los usuarios

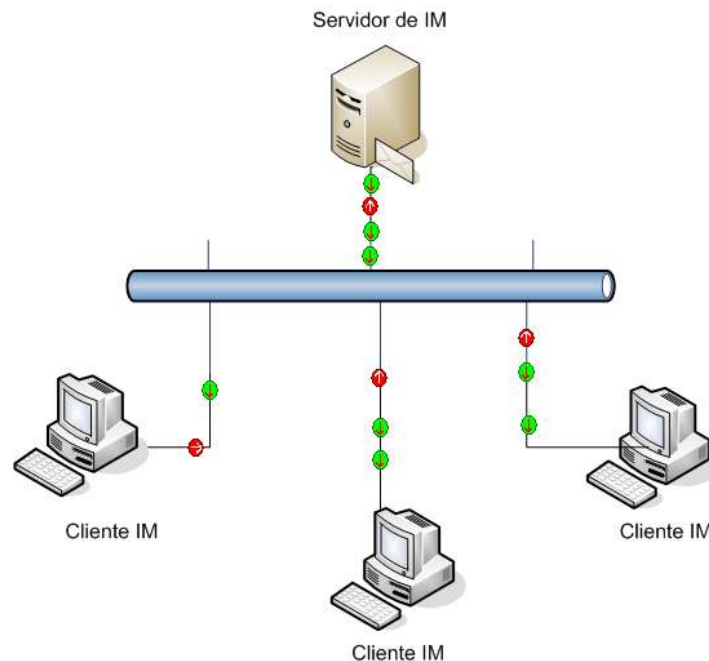


tiende a repartirse entre todos ellos. De esta forma se evita de una forma natural la formación de atascos, típicos cuando se usa un único recurso para todo el mundo.

- **Disponibilidad de nombres:** Al haber muchos servidores independientes, es más probable que el nombre escogido por el usuario esté disponible en el servidor que desea. Si el nombre ya estuviera reservado por otra persona, siempre tiene la posibilidad de probar en otro servidor.
- **Múltiples cuentas:** Una persona puede crear cuentas en varios servidores que sean de su interés, todas ellas con el mismo nombre de usuario. Puede mantener una cuenta para cada aspecto de su vida.
- **Confianza en el servidor:** El usuario puede elegir un servidor en el cual confíe, no tiene por qué usar un servidor que le genere dudas. Puede usar un servidor proporcionado por una empresa comercial de su confianza o incluso puede montar un servidor Jabber propio.
- **Conversaciones más seguras:** Cuando dos usuarios de Jabber están hablando, los mensajes se envían directamente entre sus respectivos servidores. No existe por tanto un servidor central mundial que tenga la posibilidad de escuchar las conversaciones de todos los usuarios. Entonces, la seguridad percibida por el usuario se basa en la confianza que tenga este, en el servidor y en el contacto con quien habla.
- **Supervivencia de la red:** Un servidor Jabber puede salir de servicio y dejar sin este a sus usuarios durante un determinado periodo de tiempo, mientras se resuelve el problema. Aunque no ocurra normalmente, es un hecho inevitable. No obstante, cuando un servidor Jabber se sale de servicio, lo hace solo; mientras el resto de la red Jabber permanece inmutable. Por lo tanto, solamente un número de usuarios relativamente pequeño se vería afectado por el percance.
- **Servicios Privados:** La independencia de los servidores permite establecer una estructura de mensajería instantánea sobre una intranet, sin necesidad de salir a la Internet. Este modelo es ideal para empresas que deseen alejar su tráfico de datos de servidores ajenos, incluso se podría montar una red aislada del resto de los servidores.

### 2.1.1 Modelo de Mensajería [24]

Para poder entender el modelo de mensajería, primero hay que tener claro los participantes básicos de una conexión Jabber como son el servidor, el cliente, la conexión y los paquetes. En la Figura 2.1 se puede apreciar un modelo de mensajería instantánea básico.



**Figura 2.1 Modelo básico de Mensajería Instantánea**

El servidor Jabber participa en todas las comunicaciones Jabber. Su principal responsabilidad es la de proveer servicios Jabber a los clientes de su dominio como pueden ser la redirección de sus paquetes y la gestión de sus cuentas de usuario. El cliente Jabber es quien interactúa directamente con el usuario, esta aplicación muestra las respuestas del servidor, recoge las peticiones del cliente y las envía al servidor para que este último las maneje. La conexión es realizada entre el cliente y el servidor, por la cual viajarán los paquetes con formato XML. Por último, pero los más importantes, los paquetes: son las peticiones y mensajes que envían o reciben los usuarios, en forma de fragmentos de XML correctamente formados. El protocolo Jabber es quien especifica el formato de estos paquetes.



El servidor Jabber, como cualquier servidor, es iniciado en una máquina y se mantiene a la espera de conexiones de usuarios en un puerto determinado. El estándar de Jabber establece el puerto 5222 como el puerto estándar de los servidores Jabber y el 5223 para conexiones cliente/servidor SSL seguras. Se pueden utilizar otros puertos, pero la mayoría de los clientes Jabber intentarán conectarse por defecto a estos dos puertos, por lo que deberán ser reconfigurados para que se conecten a puertos alternativos.

El cliente Jabber comienza una conexión mandando la etiqueta XML correspondiente al servidor, esto representa el inicio de una sesión del cliente con el servidor. Cuando un servidor Jabber acepta una conexión de un cliente responderá con otra etiqueta XML para confirmar el inicio de la sesión. Si hubiera un error en la sesión se enviaría una etiqueta XML de error explicando el problema y cerrando la conexión por parte del servidor.

Una vez que se ha establecido el flujo entre el cliente y el servidor, pueden enviarse paquetes de acuerdo con los múltiples tipos que define el protocolo Jabber. En la mayoría de los casos, el servidor sólo le permitirá al cliente enviar unos pocos tipos de mensajes hasta el momento en que se autentique; entre ellos, los paquetes de autenticación. La autenticación permite al servidor verificar que el cliente es quien dice ser para poder actuar bajo el Jabber ID con el cual se autentica. El estándar de Jabber no especifica el conjunto de protocolos disponibles para un usuario sin autenticar, pero como mínimo es normal que los usuarios puedan utilizar los protocolos de autenticación y registro. Tanto el servidor como el cliente pueden cerrar el flujo de datos en cualquier momento enviando una etiqueta de cierre. En ese momento, cada uno podrá cerrar la conexión y terminar la sesión Jabber.

El modelo cliente/servidor distribuido de Jabber tiene muchas ventajas. Una de las más importantes es el uso de un modelo simple y conocido por todos los desarrolladores de software para las comunicaciones en entornos de red. Este modelo permite al servidor del dominio centralizar el control de Jabber, pueden crearse políticas de seguridad y aplicarlas sin tener que modificar las aplicaciones clientes o tener que controlar conexiones directas entre los clientes debido a que todo el tráfico pasa por el servidor.



Como todo, el modelo cliente/servidor también tiene sus inconvenientes. Como la privacidad de cara a alguien que tenga acceso al servidor, debido a que en él, se almacenarán todos los mensajes que se reciban para las cuentas Jabber ID de los usuarios que no estén conectados. Otro gran problema de la arquitectura cliente/servidor es que si en la red empresarial sólo existe un servidor de Jabber y éste permanece inactivo durante un tiempo, los clientes no podrán comunicarse entre ellos. Este problema tiene una fácil solución: la instalación de un grupo de servidores de Jabber. Además de la gran seguridad de las conexiones cliente/servidor, en las que otro cliente nunca puede averiguar la dirección, esto también tiene un gran inconveniente: Conlleva a un cuello de botella que sólo se podrá ir solucionando ampliando el servidor de Jabber, añadiendo nuevos servidores o ampliando el ancho de banda de las conexiones con el servidor.

### 2.1.2 Protocolo de Mensajería [25]

Los mensajes son la parte más importante de la Mensajería Instantánea, por ello el protocolo de mensajes de Jabber es simple pero poderoso. El protocolo de mensajes es extremadamente sencillo, los paquetes son enviados por un usuario a un receptor. Por defecto, no se reciben confirmaciones del destinatario acerca de la llegada del mensaje, para reducir el tráfico en el servidor, además si el receptor no se encuentra disponible, el servidor guardará el mensaje hasta que este se conecte. El envío de mensajes es la mayor responsabilidad de cualquier sistema de mensajería Jabber, por ello se han desarrollado seis tipos de mensajes. Los cinco primeros son los tipos más normales de mensajes en los sistemas Jabber. Los mensajes “jabber:x:oob” se denominan Fuera de Banda y facilitan un mecanismo para intercambio de datos directamente entre dos usuarios.

A continuación se describen los seis tipos de mensajes definidos para por el protocolo Jabber:

- **Normal:** El tipo por defecto de mensaje en el protocolo es llamado “mensaje normal”. Está compuesto por un objetivo, remitente, destinatario y un mensaje. Este tipo de mensaje es creado y enviado normalmente usando interfaces similares a las usadas en las aplicaciones de correo electrónico, y a su semejanza, estos mensajes son enviados a los usuarios sin la necesidad de que el destinatario esté conectado.



- **Chat:** Los usuarios usan los mensajes de tipo Chat para enviarse entre ellos mensajes instantáneos. Estos mensajes suelen ser cortos, ya que se utilizan para llevar a cabo una conversación entre dos personas.
- **Groupchat:** Los mensajes del tipo groupchat son similares a los mensajes del tipo Chat, pero están diseñados para mantener conversaciones entre un grupo de personas sobre un tema en concreto. Por ello, cuando un cliente envía un mensaje al grupo, todos los clientes que se hayan unido a este, recibirán el mensaje, aunque también se podrán escribir mensajes de privados dirigidos a un usuario en concreto. Para formar parte de un grupo es necesario unirse a él con un sobrenombre, el cual será mostrado al resto de los usuarios. Esto está pensado para ocultar la identidad real de los participantes.
- **Headline:** Los mensajes del tipo headline están diseñados para mostrar información en la barra de estado o en otras partes de la interfaz de usuarios.
- **Error:** Este tipo de mensaje puede ocurrir cuando un cliente envía un mensaje y en el transcurso de la transmisión sucede un inconveniente, ya sea que el cliente al que va dirigido no existe, o que sencillamente el mensaje ha sido rechazado por el cliente destinatario. En la Tabla 2.1 se muestra los errores más comunes del núcleo de Jabber:

Tipo de Error	Significado
400	Petición errónea.
401	Desautorizado.
402	Servicio de pago.
403	Prohibido.
404	No encontrado.
405	No permitido.
406	No aceptable.
407	Registro requerido.
408	Timeout.
409	Conflicto.
500	Error interno del servidor.
502	Error del servidor remoto.
503	Servicio no disponible.
504	Timeout de servidor remoto.



**Tabla 2.1 Errores comunes del núcleo de Jabber**

- **Jabber:x:oob:** No son realmente mensajes convencionales de Jabber, en cambio, son una extensión de los mensajes que son enviados dentro de los mensajes normales. Un mensaje de este tipo contiene normalmente información, que en la mayoría de los casos es una URL, la cual, el cliente puede usar para realizar una transferencia de datos punto-a-punto sin pasar por el servidor. Los clientes normalmente suelen implementar este mensaje, iniciando un servidor Web o uno FTP, separadamente o como parte del cliente Jabber. Por lo tanto están dando a otro cliente información de la IP y del puerto que han abierto para la descarga el archivo.

Esto es ideal para el intercambio de ficheros entre clientes ya que con ello se consigue reducir el tráfico que pasa por el servidor. Por otra parte se corre un riesgo por que se está revelando la IP del cliente, por lo cual, si la persona a la que el usuario envía esta información es un usuario malintencionado podría utilizar la IP con otros fines muy distintos al inicial.

### 2.1.3 Protocolo de Presencia [25]

Entre otras cosas la Mensajería Instantánea se diferencia del correo electrónico en que los usuarios tienen la posibilidad de detectarse los unos a los otros y saber cuándo un usuario se encuentra conectado. Incluso, hoy en día se va más allá y podemos saber si un usuario, aunque conectado, en ese instante está ocupado o ausente, por lo tanto debemos esperar hasta que este disponible para comunicarnos en él. Además Jabber ofrece la posibilidad de poner tantos estados diferentes como desee el usuario, así siempre se sentirá más identificado con el estado en el que pone su sistema de IM.

Una de las prioridades más altas tenidas en cuenta en el protocolo Jabber es la intimidad de los usuarios. Por lo tanto, si un usuario quiere agregar a otro en su lista de contactos y recibir su presencia, deberá hacer una petición al servidor, y si el otro usuario acepta, entonces se podrá ver su estado.

El protocolo de presencia es usado principalmente en dos contextos:

- **Actualización de presencia:** cuando se presenta un cambio de estado del usuario.



- **Manejo de suscripción de presencia:** permite a los usuarios suscribirse a las actualizaciones de presencia de otros usuarios y controlar quien está accediendo a su propia presencia.

En ambos casos el servidor de Jabber actúa como un árbitro entre el emisor de la actualización de presencia y los destinatarios de la misma. El servidor tiene la obligación de hacer llegar el paquete de actualización de presencia a todos los contactos del usuario, pero sólo a los contactos que el emisor ha autorizado para recibir su presencia.

El servicio de actualización de presencia usa un mensaje unidireccional del emisor al servidor de su dominio, y este tendrá que copiar y reenviar el mensaje de presencia a todos los contactos del emisor. Para hacer esto, el servidor busca en el Roster (la lista de contactos), la cual es almacenada dentro de este. De esta forma, el cliente siempre tendrá su Roster actualizado, aunque cambie de aplicación o de máquina, gracias a que no es él quien guarda esos datos, sino el servidor de su dominio. Los clientes enviarán y recibirán mensajes de suscripción de presencia de dos tipos: “subscribe” y “unsubscribe”. Estos mensajes deben de ir llegando a su destinatario, pero sin embargo, el servidor deberá escuchar y actualizar la presencia de su cliente de acuerdo a estos mensajes. Este tipo de suscripción de presencia también es usada en los mensajes de grupo.

Dependiendo de sus atributos los mensajes de presencia, pueden ser de diferente tipo:

- **Available:** mensaje de actualización que indica que el usuario está listo para recibir mensajes.
- **Unavailable:** mensaje de actualización que indica que el usuario no está disponible para recibir mensajes.
- **Subscribe:** Mensaje de petición de suscripción de presencia, el usuario que lo envía desea suscribirse a la presencia del destinatario.
- **Unsubscribe:** Mensaje de cancelación de suscripción de presencia, el usuario que lo envía desea cancelar su suscripción a la presencia del destinatario.
- **Subscribed:** Respuesta que recibirá un usuario que ha realizado una petición de suscripción, que indica el estado actual de la suscripción.





- **Unsubscribed:** Respuesta que recibirá un usuario que ha realizado una petición de suscripción y le ha sido negada, o una petición de cancelación de la suscripción.
- **Error:** Mensaje estándar de error de Jabber que indica problemas en la presencia.
- **Probe:** Petición Servidor-a-Servidor que envía toda la información de presencia de un servidor a otro.

#### 2.1.4 Protocolo de Grupo de Chat [11]

Existen cuatro acciones fundamentales que un usuario debe ser capaz de hacer en un groupchat:

- **Unirse al grupo:** Enviando una presencia de tipo “available” al grupo.
- **Enviar mensajes a todo el grupo:** Enviando un mensaje de tipo groupchat al Jabber ID del grupo deseado, al cual previamente, el usuario ha debido de unirse.
- **Enviar mensajes a un único miembro del grupo:** Enviando un mensaje de tipo groupchat a una persona del grupo, poniendo tras el Jabber ID del grupo el carácter ‘/’ y el nickname de la persona.
- **Abandonar el grupo:** Enviando un mensaje “unavailable” de presencia al grupo.

Para unirse y abandonar un grupo se usa el protocolo de presencia, y para enviar mensajes, ya sean a todo el grupo, o mensajes privados se usa el protocolo de mensajes. Para unirse a un grupo, se usa una actualización de presencia especificando el nickname que el usuario va a utilizar a lo largo de la sesión.

#### 2.1.5 Protocolo de Pregunta/Información (IQ) [11]

Este tipo de mensajes como bien indica su nombre son de petición respuesta, así que cuando un cliente envía un mensaje IQ (Info/Query) siempre deberá recibir una respuesta del destinatario, aunque sea de tipo error. Un usuario en Jabber representa a una persona, los clientes Jabber actúan en representación de un determinado usuario, y un usuario en particular puede tener varios clientes activos simultáneamente. Los clientes que actúan en representación de los usuarios son llamados “resources” y se indican al final del Jabber ID del usuario.



El usuario es representado en el sistema Jabber por una cuenta de usuario almacenada y gestionada en el servidor Jabber del dominio. Los clientes se autentifican con el servidor Jabber usando la información de autenticación. Una vez el usuario se ha autenticado, el cliente puede descargar o actualizar la información de su cuenta de usuario.

Este protocolo se centra fundamentalmente en los procedimientos para crear y almacenar la información de un usuario en el servidor. Aunque la mayoría del tráfico de un sistema de Mensajería Instantánea Jabber está compuesto de mensajes y presencias, el mayor esfuerzo a la hora de implementar un cliente o servidor Jabber/XMPP radica en la administración de las cuentas de usuarios. Jabber ha implementado un protocolo de petición/respuesta para gestionar estas peticiones.

IQ es un protocolo sencillo y ampliable de petición/respuesta que permite a los usuarios hacer peticiones y almacenar o cambiar datos usando un sencillo protocolo. IQ es simplemente el conductor de esas peticiones y respuestas, y gestiona los datos que son necesarios de acuerdo con las necesidades particulares de cada servidor. La mayoría de las peticiones IQ son entre un cliente y un servidor. Sin embargo, hay algunos protocolos de IQ que van estrictamente de un cliente a otro, como el protocolo de petición de versión del cliente, en el cual, un cliente le pedirá a otro su versión del programa.

El protocolo IQ tiene dos participantes, el que genera la petición y el que la trata. Pueden darse dos situaciones, una comunicación entre un cliente y un servidor, en la cual, el cliente será quien envíe los paquetes IQ, o entre dos clientes, en cuyo caso un cliente realiza una petición a otro cliente, el cual, deberá responder.

El protocolo comienza con una petición de un cliente a otro cliente o servidor que gestionará su petición. El paquete IQ será encaminado por el sistema Jabber como cualquier otro paquete hasta ser entregado a su destinatario. Pero al contrario de un mensaje Jabber, el paquete no será almacenado si el receptor del paquete no se encuentra disponible. Además, los paquetes IQ que se dirigen a un usuario no pueden ser enviados al Jabber ID de un cliente, hay que especificar el cliente concreto de ese Jabber ID, es decir, habrá que especificar el "resource". A no ser que el destinatario sea un



servidor, que en tal caso el paquete si puede ir dirigido al servidor o incluso no mencionar el destinatario, si este es el servidor de su dominio.

Los paquetes IQ pueden ser de dos tipos, “set” o “get”: el primero modifica datos del destinatario del paquete, y el segundo los solicita. El destinatario tratará el paquete y siempre enviará una respuesta, en el primer caso responderá si se ha podido hacer la modificación correctamente o si ha ocurrido algún tipo de error; y en el segundo caso nos devolverá la información que hemos solicitado, o algún mensaje de error.

### **2.1.6 Protocolo de Registro [11]**

El primer paso para poder tener usuarios en el servidor es dejarlos crear sus cuentas mediante esta extensión del protocolo IQ. Esto no ocurrirá así, en un servidor de uso privado de una red empresarial, en el cual, uno de los factores importantes es la seguridad. Pero para cualquier servidor público de Jabber es necesario implementar el protocolo de registro para que cualquier persona que lo desee pueda registrarse y unirse a la comunidad de usuarios Jabber. Esta extensión del protocolo IQ es usada para tres propósitos:

- Crear cuentas de usuario.
- Actualizar la información de las cuentas de usuario.
- Borrar cuentas de usuario.

En los servidores públicos, y en su más pura esencia, las cuentas de usuario no son más que unos repositorios de credenciales que los clientes usan para autenticarse con el servidor. Además de estos datos básicos, muchos servidores asocian otros datos a la cuenta de usuario como pueden ser el e-mail, el nombre completo del usuario y su teléfono, así como otros datos que el servidor quiera recoger de sus usuarios.

Además muchos usuarios también almacenan “vCards” a su cuenta de usuario. Las “vCards” son como tarjetas de negocio, en formato XML, que contienen información de contacto sobre una persona. Los servidores tienen muchas formas de almacenar la información de un usuario. La forma de almacenar estos datos está fuera del alcance del



protocolo Jabber y por ello cada servidor lo realizará de la forma que mejor se adapte a sus necesidades. Una forma muy utilizada, y que también la utiliza el servidor de referencia de Jabber es usar ficheros XML con los datos de cada usuario dentro de un directorio. Otras implementaciones, por ejemplo pueden usar una base de datos o hacer persistente los datos.

Si el servidor no tiene implementado este protocolo de registro se deberán desarrollar utilidades para el tratado de las cuentas de usuario. Otros servidores tendrán implementado este protocolo y además tendrán otras herramientas para el manejo de las cuentas de usuario.

Aunque el manejo y almacenamiento de las cuentas de usuario puede ser algo complicado, la implementación del protocolo de registro no es así. El protocolo de registro normalmente suele ser junto con el de autenticación el único protocolo que un usuario sin autenticar puede usar de un servidor Jabber después de iniciar la conversación con la etiqueta

En general el protocolo implica una consulta “get” para comprobar el tipo de registro implementado en el servidor, y los datos necesarios para el registro, y un “set” para el registro en sí, en el que se deberán enviar al menos los datos obligatorios como son el nombre de usuario y la contraseña. Hacer la petición “get”, antes de registrarse no es obligatorio, pero dará información al cliente sobre cuales campos son soportados por el servidor en el registro, ya que el envío de campos no implementados por el servidor no está implementado en el protocolo, y se desconoce su respuesta.

### **2.1.7 Protocolo de Autenticación [11]**

El protocolo de autenticación de Jabber es como el de registro, una extensión del protocolo IQ. Es uno de los protocolos únicamente dedicados a la seguridad en Jabber, permite a los usuarios demostrar a su servidor que realmente son quien dicen ser, aunque muchas veces el cliente no tiene la certeza de que el servidor sea quien dice ser, debido a que dentro de una red, un cliente puede interceptar sus mensajes y engañarle, pero esto ya no es responsabilidad del protocolo Jabber, sino fallos que derivan directamente de las direcciones IP v.4.



Este sistema de autenticación pertenece al núcleo del protocolo Jabber, es decir, a los inicios del protocolo, desde que salió el estándar Jabber la autenticación se realiza mediante SASL (Simple Authentication and Security Layer), aunque siguen existiendo servidores que admiten este tipo de autenticación.

El sistema de autenticación y acceso a un servidor es simple: los clientes no autenticados tienen una serie de permisos, y los clientes autenticados tienen un completo acceso al uso de todos los protocolos de Jabber que estén implementados en el servidor del dominio al que pertenecen. Desafortunadamente, este simple modelo de autenticación es muchas veces limitado. Además, si un administrador de un sistema Jabber desea restringir los mensajes a ciertos dominios Jabber, limitar el tamaño de transferencias de ficheros, el tamaño de los paquetes u otras funciones administrativas, se verá ante la ausencia en el protocolo Jabber de estándares sobre este tipo de funciones administrativas. Otro problema es que Jabber/XMPP no distingue entre rangos o grupos de usuarios, todos los usuarios tienen según el protocolo el mismo derecho a usar todos los protocolos implementados en el servidor, sin embargo esto debería ser diferente.

Se están juntando dos conceptos diferentes como son el de autenticación y autorización, una persona autenticada no debería estar autorizada a hacer todo lo que deseara. Sin duda este aspecto se retomará cuando el protocolo esté en plena madurez.

Quitando estos pequeños inconvenientes que pueden ser solucionados por los desarrolladores aunque no estén tratados en el conjunto del protocolo Jabber, el protocolo de autenticación de Jabber aporta garantías suficientes para cualquier sistema de MI y sus actividades. Además el protocolo ofrece cuatro niveles diferentes de autenticación:

- **Anonymous:** Si el servidor admite usuarios anónimos basta con el envío de la petición "set" sin ningún otro tipo de datos. El servidor responde con un error si no soporta usuarios no autenticados. Pero si soporta usuarios anónimos devuelve como resultado del "set" un nombre aleatorio. El usuario podrá entonces enviar mensajes, donde su Jabber ID estará formado por el nombre del servidor seguido del carácter '/' y del recurso aleatorio que ha devuelto en la autenticación anónima.



- **Plain:** Este tipo de autenticación es el primero que provee de algún tipo de seguridad. Su principal ventaja es su extrema sencillez a la hora de su implementación. Funciona enviando dentro del XML de autenticación la contraseña en formato de texto sin codificar, del usuario a autenticar. El principal problema de este tipo de autenticación es que la contraseña es enviada de forma abierta al servidor y es fácil, para alguien que esta escuchando en la red, capturar la información de nombre y contraseña del usuario, y luego suplantarlos ante el servidor Jabber.
- **Digest:** Para evitar el envío de contraseñas en texto sin codificar, mediante este tipo de autenticación el servidor envía un identificador de sesión en el campo "id" junto con la etiqueta de inicio de sesión. Para generar la autenticación de tipo "digest" el cliente concatena el identificador de la sesión con la contraseña del usuario. La cadena resultante es codificada usando el algoritmo SHA-1 (Secure Hash Standard - 1). El único inconveniente de este tipo de autenticación es que la contraseña del usuario debe ser enviada durante el protocolo de registro como texto sin codificar. Además el servidor guarda la contraseña en formato de texto sin codificar. Un fallo en la seguridad del servidor puede poner en problemas la seguridad de las contraseñas de sus usuarios. Sin embargo, la contraseña sólo viaja por la red empresarial o Internet durante el registro, luego irá siempre codificada.
- **Zero-knowledge:** Es el formato más seguro y más complicado soportado por los protocolos Jabber, muchas veces lo podemos encontrar escrito como "Ok". El método de autenticación Ok es complejo y su adopción tanto en servidores como en clientes ralentiza el proceso de autenticación. Este tipo de autenticación ya no guarda la contraseña del usuario en el servidor. De hecho, la información que el servidor guarda son las credenciales que sólo sirven para una única autenticación del cliente. El servidor irá creando nuevas credenciales de un solo uso.

La técnica para generarlas utiliza cuatro tipos de información:

- **La contraseña del usuario:** Usada por el cliente para generar llaves de tipo Ok. La contraseña es almacenada en el cliente y nunca es enviada al servidor. La llave es creada con la combinación de la contraseña del usuario y el token.
- **Token:** Información generada de forma aleatoria, usada para crear la llave de Ok. El token es almacenado en el servidor, separando la contraseña y el token entre el



cliente y el servidor respectivamente se consigue que la llave creada sea única para el conjunto cliente/servidor.

- **Sequence:** Número que decrece automáticamente indicando, cual de las llaves está siendo usada en el momento.
- **Hash:** Llave particular en el conjunto de llaves identificada por el número de secuencia explicado anteriormente.

Inicialmente el cliente genera todas las piezas que se usan en el protocolo de registro. Para hacer esto, el cliente debe:

- Crear un mensaje de tipo digest con SHA-1 de la contraseña del usuario para crear el hashA. La salida se convierte a hexadecimal en minúsculas.
- Generar un token aleatorio.
- Generar el digest de la combinación de los resultados de los dos puntos anteriores para crear el hash 0 que es convertido a hexadecimal en minúsculas.
- Seleccionar arbitrariamente un número de secuencia.

Digest del hash-n para crear el hash-(n+1) y convertirlo en hexadecimal. Así hasta generar la secuencia m que será el número del paso 4.

El cliente envía el token, la secuencia, y el hash al servidor en el protocolo de registro si éste soporta autenticación Ok. Para autenticarse, el cliente sigue dos pasos. El primero consiste en enviar una prueba de autenticación y, el servidor devuelve el token y el número de secuencia menos uno. El cliente coge este token, y el número de secuencia y genera un nuevo hash (m-1). El cliente para ello sigue los mismos pasos que los explicados anteriormente, excepto la utilización del token y el número de secuencia dados, para enviarlos al servidor. El servidor coge el hash-(m-1) y genera el hash m mediante digest. Compara el enviado por el cliente y el que ha generado él, si coincide, el usuario es autenticado. Si no coincide devolverá un mensaje IQ de error. Si todo es correcto el servidor decrementa el número de secuencia y lo almacena. La próxima vez que el cliente se autentifique, el servidor envía el token M-2 al cliente. El proceso continúa hasta que el número de secuencia llegue a cero. Entonces, el cliente debe volver a usar



el protocolo de registro para actualizar el token y el número de secuencia antes de que éste llegue a cero.

Este tipo de autenticación tiene muchas ventajas:

- Las contraseñas nunca son enviadas por la red.
- Las contraseñas nunca son almacenadas en el servidor.
- Las contraseñas robadas dejan de ser válidas tan pronto como son usadas.
- La mayoría del procesamiento es para el cliente, liberando al servidor, lo cual evita generar un cuello de botella en el sistema.
- La única vulnerabilidad del sistema es la no actualización de los credenciales antes de que el número de secuencia llegue a cero. La actualización de los mismos sólo es posible una vez el cliente se haya autenticado.

Como el protocolo de registro, el protocolo de autenticación también consta de dos fases diferentes, una de consulta en la que el cliente podrá conocer cuáles de los tres tipos de autenticación que propone el estándar están implementados en el servidor, lo cual como resultado nos devolverá datos de autenticación usados para el método zero-knowledge, si es que el servidor lo soporta. La segunda fase consiste en la autenticación en sí. El cliente enviará una petición “set” que contendrá los datos necesarios para la autenticación según el tipo que el usuario elija. La prueba de autenticación utiliza una petición “get” en la que se deberá especificar el nombre del usuario que se pretende autenticar. El servidor retornará los métodos de autenticación disponibles para ese usuario en concreto y los campos que el usuario podría utilizar para autenticarse.

### **2.1.8 Protocolo Roster [25]**

Para no tener que enviar los cambios de presencia entre todos los usuarios del sistema Jabber ha creado el concepto de suscripción de presencia. Como su nombre indica, la suscripción de presencia determina los suscriptores que recibirán las actualizaciones de presencia de cada usuario.





Los suscriptores pedirán una suscripción a un usuario, y el usuario aceptará o denegará dicha suscripción. Cada usuario se suscribirá a los usuarios que desee y podrá aceptar que dichos usuarios u otros usuarios se suscriban a los cambios de su presencia. Para organizar y administrar las suscripciones de cada usuario, Jabber ha definido unas estructuras de datos estándar conocidas como Jabber roster. Jabber roster no es más que una lista de otros usuarios identificados por su Jabber ID. El protocolo de suscripción de presencia permite a los usuarios suscribirse a la presencia de otros usuarios, sea cual sea su dominio de Jabber.

Los servidores usan el protocolo de suscripción de presencia para sincronizar los rosters de sus usuarios tanto fuera como dentro de su dominio. Los diferentes tipos de relaciones de suscripción tratados por el roster son los siguientes:

- **To:** el usuario está interesado en recibir las actualizaciones de presencia del suscriptor.
- **From:** el suscriptor está interesado en recibir las actualizaciones de presencia del usuario.
- **Both:** el usuario y el suscriptor tienen un mutuo interés en recibir las presencias entre ellos.
- **None:** ni el usuario ni el suscriptor tienen interés por recibir presencias entre ellos.

Además de la información básica de suscripción, el roster permite al usuario almacenar información estándar de la interfaz de ese suscriptor. Esta información incluye un sobrenombre puesto por el usuario a su suscriptor y el grupo o grupos al que pertenece el suscriptor a la hora de mostrarlo en la interfaz.

Una de las cosas más importantes es que toda esta información puede ser administrada desde el servidor ya que se encuentra almacenada en él. Cualquier cambio realizado en el roster en uno de los clientes será automáticamente actualizado en los demás clientes iniciados con el mismo Jabber ID. El protocolo roster fue desarrollado para permitir a los clientes su administración.



A pesar de la estrecha relación entre el roster y la presencia, son conceptos diferentes. Digamos que el roster es todo el conjunto de contactos relacionados con la cuenta Jabber del cliente. Además se puede guardar datos de cada contacto como un sobrenombre puesto por el usuario o, el grupo o grupos al que pertenece para poder así buscar todos los contactos de un grupo, y no tener que recorrer todos los contactos. Como cada usuario tendrá su propio roster, si enviamos una actualización de presencia al servidor, éste buscará en el roster del usuario todos los contactos que tengan los tipos de suscripción “both” o “form” para enviarles solamente a ellos la actualización de la presencia.

Como todo se almacena en el servidor, el cliente comienza su sesión iniciando el proceso de autenticación y pidiendo todo el roster al servidor con un roster reset. El cliente muestra entonces todos los contactos del roster en la aplicación. Si más adelante el usuario desea hacer cambios en algún contacto del roster, envía un mensaje Roster update al servidor y se queda esperando un roster push del servidor para confirmar la actualización a todos los clientes abiertos con el mismo Jabber ID, ya que si uno de ellos realiza un cambio, éste se tiene que reflejar en el resto. Los cambios del roster pueden suceder en cualquier momento desde que un usuario se autentifica, por lo tanto los clientes deben estar preparados para recibir del servidor roster push y actualizar los contactos que muestran en ese momento. El servidor envía un mensaje Roster push cuando una actualización del roster cambie los atributos del mismo, y se deba actualizar los clientes mostrados o alguno de sus atributos o cuando por algún cambio en el tipo de suscripción de presencia se deba crear o borrar un contacto.

Como se ha visto, el protocolo de suscripción de presencia tiene grandes efectos sobre el roster, tantos, que una actualización del tipo de suscripción puede hacer desaparecer a un contacto de la pantalla del usuario, debido a que si el contacto no desea ser visto por el usuario y envía una cancelación de la suscripción de presencia poniéndola a “none” el usuario debe dejar de ver el estado de ese contacto lo antes posible. Los clientes sólo pueden modificar el apodo de sus contactos y el grupo o grupos a los que pertenecen mediante el protocolo roster. Además, como hemos visto pueden influir en el roster mediante el protocolo de suscripción de presencia.



El protocolo roster es una extensión del protocolo IQ, existen tres tipos básicos de protocolos roster:

- **Roster Get:** Permite a los clientes obtener una copia completa del roster almacenado en su servidor Jabber. Para ello se debe enviar un mensaje IQ de tipo get vacío especificando el protocolo Jabber roster.
- **Roster Update:** Usado por los clientes para actualizar el roster almacenado en el servidor. El servidor responderá a cada cliente con una respuesta indicando que la actualización se ha realizado correctamente o un mensaje de error en el caso contrario.
- **Roster Push:** Es enviado por el servidor a todos los clientes que se han conectado con el mismo Jabber ID tras una modificación en el roster por alguno de ellos, o por la modificación del tipo de suscripción de presencia de alguno de los contactos. Son los únicos mensajes IQ que no obtienen respuesta, sólo van del servidor a los clientes y éstos no tienen que responder.

### 2.1.9 Transportes entre Jabber y otros Servidores de Mensajería Instantánea [11]

Debido a que Jabber es un protocolo libre basado en el paso de paquetes en formato XML, los sistemas Jabber están preparados para el uso como un sistema genérico de transporte de Mensajería Instantánea. Su simple diseño ha sido explotado por servidores Jabber para conectarse con otros sistemas de IM como pueden ser el MSN Messenger y Yahoo! Messenger.

El servidor Jabber de referencia jabberd utiliza módulos llamados transports que proveen un puente entre Jabber y los demás sistemas de IM. Los transports tratan a cada sistema de IM propietario como un dominio Jabber con sus propios clientes identificándolos por su "Jabber ID", que en realidad sería el nombre del usuario en el sistema externo. Al enviar un mensaje Jabber a uno de esos módulos esos Jabber IDs especiales hacen que sean transportados por el módulo de transporte. Los módulos de transporte se conectan con el sistema de IM correspondiente y actúan como clientes de ese servidor para intercambiar mensajes y presencia entre los dos sistemas.



Por ejemplo, si se envía un mensaje a usuario@messenger.server.com el servidor reconocerá que messenger.server.com es un usuario de otro sistema y se lo enviará al módulo de transporte para que procese el paquete. El transporte iniciará la sesión en la red MSN Messenger y enviará el mensaje al interesado. Cada transporte deberá hacer el parseo de los mensajes Jabber a mensajes con el formato del sistema de IM correspondiente. En la comunicación con otros proveedores de mensajería instantánea se presenta un problema y radica en que los sistemas de IM propietarios no les interesa la interoperabilidad con otros proveedores, con el fin de mantener atados a sus clientes; entonces estos, tratan de cambiar el protocolo de manera regular, lo cual trae como consecuencia la realización trabajos de ingeniería inversa por parte de los desarrolladores de los módulos de transporte de cada servidor, para adaptarlos a los cambios y que sigan funcionando correctamente.

### **3 CAPITULO 3: DESCRIPCIÓN DE LA PLATAFORMA DE ACCESO UNIVERSAL A MENSAJERIA INSTANTÁNEA MÓVIL (PAUMIM)**

#### **3.1 *Planteamiento del Problema***

Existen diferentes proveedores de servicios de mensajería instantánea en el mundo que ofrecen a sus usuarios aplicaciones de calidad, que les permiten tener contactos con los cuales puedan utilizar las distintas funcionalidades ofrecidas, como el envío de mensajes de texto, intercambio de archivos, videoconferencia, entre otros. Algunas de las empresas más importantes que ofrecen el servicio de mensajería instantánea son: Microsoft con MSN (Microsoft Network) Messenger, American OnLine con ICQ (I Seek You) y AOL (American OnLine) Messenger y Yahoo con Yahoo Messenger.

El principal problema que presenta la mensajería instantánea es la coexistencia de varias redes, cada una de ellas con aplicaciones y protocolos distintos, lo que presenta una falta de interoperabilidad entre los diferentes proveedores, imposibilitando la comunicación universal entre los usuarios; además de esto, el acceso al servicio de mensajería instantánea convencional se lleva a cabo mediante computadores personales con



conexión a Internet, lo cual restringe la capacidad de movilidad de los usuarios. El presente proyecto busca solventar la falta de interoperabilidad entre los diferentes proveedores de mensajería instantánea y la movilidad de los usuarios por medio de la construcción de una Plataforma de Acceso Universal de Mensajería Instantánea Móvil (PAUMIM), la cual permitirá la comunicación entre dispositivos móviles, usando el ya mencionado servicio de mensajería instantánea y que además ofrecerá a los usuarios la interoperabilidad con los principales proveedores del servicio como MSN, Yahoo y AOL.

### **3.2 *Requerimientos para un sistema de Mensajería Instantánea Móvil***

#### **3.2.1 Cliente de mensajería instantánea móvil**

El cliente de Mensajería Instantánea Móvil es la parte del sistema que tiene contacto directo con los usuarios y les brinda toda la funcionalidad del sistema. El usuario puede utilizar el Sistema de Mensajería Instantánea móvil mediante la interfaz gráfica de la aplicación cliente instalada en el dispositivo móvil, esto de forma rápida y eficiente, con la posibilidad de enviar mensajes instantáneos, actualización de presencia, adición y eliminación de contactos, entre otros. El cliente cumple con los principales lineamientos de usabilidad, entre los que se tienen la satisfacción del usuario, la eficiencia y eficacia del cliente móvil. Para que el cliente móvil satisfaga en su totalidad al usuario se tiene en cuenta la facilidad en el aprendizaje de manejo, la disposición de las ayudas y la documentación necesaria para que el usuario pueda hacer un fácil, correcto e inmediato uso del cliente móvil. La eficiencia hace referencia a la relación entre recursos consumidos y grado de cumplimiento de las funcionalidades por parte del cliente, en donde se debe tener en cuenta las limitadas capacidades de los dispositivos móviles y la rápida respuesta que el usuario necesita al hacer uso del mismo. La eficacia viene definida por la precisión en que se ejecutan las tareas de la aplicación. En esta parte es muy importante llevar un adecuado manejo de errores, informando al usuario con mensajes dicentes el estado de la aplicación y cuál puede ser la razón por la cual se produjo dicho error [26].



### 3.2.2 Servidor de Mensajería Instantánea Móvil

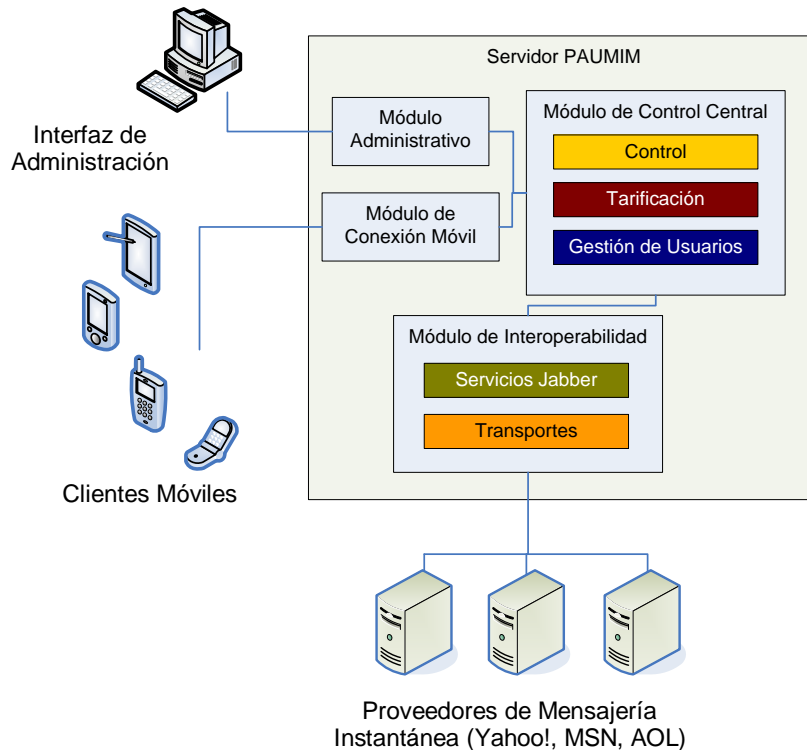
El servidor de Mensajería Instantánea Móvil es la parte del sistema que brinda a los clientes móviles todos los servicios mensajería instantánea. En este tipo de aplicaciones debe tenerse en consideración aspectos muy importantes como son la utilización masiva y concurrente de los usuarios, así como la portabilidad y posible migración a otras tecnologías. Para cumplir con lo mencionado, el servidor de mensajería instantánea debe contar con las características nombradas a continuación:

- **Alta disponibilidad:** El servidor de mensajería debe tener alta disponibilidad para atender a los usuarios, con lo cual genera buen servicio y confianza en el sistema. Para poder ofrecer un servicio de alta calidad desde el punto de vista de la disponibilidad del sistema, se debe tener en cuenta consideraciones y limitaciones en el hardware y el software. En la parte hardware, el servidor debe estar montado en una máquina dotada con un sistema de protección contra descargas eléctricas, regulación de voltaje y fuente de energía alterna que le permita llevar un continuo funcionamiento en caso de perder energía de su fuente principal. En la parte software, debe estar construido en una plataforma de desarrollo de aplicaciones de tipo empresarial que soporte una alta carga de usuarios concurrentes.
- **Seguridad:** Permite el manejo de seguridad en cuanto a la autenticación de los usuarios y la encriptación de la información tanto en el transporte como en el almacenamiento persistente de datos, esto con el objetivo de que no sea descifrada en caso de ser detectada en la red o que algún intruso pueda acceder a la base de datos.
- **Portabilidad:** Tiene la capacidad de implantarse con facilidad en múltiples plataformas y múltiples sistemas operativos.
- **Confiabilidad:** Maneja la integridad de la información de usuario sin dejar que terceros puedan capturar y conocer dicha información.
- **Escalabilidad:** Debe tener la capacidad de soportar un gran número de usuarios sin disminuir su eficiencia.

### 3.3 Descripción de la solución

La Figura 3.1 ilustra la arquitectura de la Plataforma de Acceso Universal a Mensajería Instantánea, la cual cumple con todas las necesidades de escalabilidad y accesibilidad

acordes a la evolución que debe tener un proyecto de esta envergadura, teniendo como pilares fundamentales los conceptos de herramientas libres, sistemas multiplataforma y mensajería instantánea. A continuación se describe la arquitectura de PAUMIM.



**Figura 3.1. Arquitectura de PAUMIM**

### 3.3.1 Módulo Administrativo

Permite realizar la administración y el mantenimiento de la plataforma PAUMIM mediante acceso Web. Entre las funcionalidades que brinda este módulo se encuentra la capacidad de gestionar usuarios, mediante su adición, edición y eliminación de la plataforma, y el establecimiento de comunicación directa entre el Administrador de la plataforma y el usuario por medio de mensajes. El administrador tendrá a su disposición información estadística referente al uso de la plataforma, mediante curvas y cifras podrá visualizar el número de usuarios conectados, tráfico cursado a través de la plataforma y tarificación del consumo de los usuarios.



Este módulo está constituido por componentes J2EE de capa de presentación del servidor, lógica de negocio y acceso a datos, bajo la utilización del Framework Struts y el patrón de diseño para el acceso a datos DAO (Data Access Object).

### **3.3.2 Módulo de Conexión Móvil**

Permite a los clientes móviles acceder a los servicios de mensajería instantánea que ofrece PAUMIM. Este módulo es el encargado de gestionar las conexiones y manejar la sesión de los clientes soportados (Cliente MIDP1.0 y Cliente MIDP2.0). Para realizar la gestión de conexiones, este módulo cuenta con dos submódulos. El primero, es el encargado de administrar las conexiones de los clientes MIDP1.0, cuenta con un componente de actualización de mensajes HTTP que mantiene en una cola los mensajes de intercambio entre el servidor y los cliente, quienes están en un continuo proceso de pregunta y respuesta para actualizar su estado. El segundo, encargado de administrar las conexiones de los clientes MIDP2.0, recibe peticiones de conexión y asigna conexiones por medio de Sockets TCP, de esta forma se logra establecer un circuito lógico de comunicación entre el servidor y cada cliente, de manera que la comunicación se lleva a cabo de forma directa.

### **3.3.3 Módulo de Interoperabilidad**

Es el encargado de llevar a cabo la implementación del protocolo de mensajería instantánea Jabber. Entre las funciones más importantes, se tiene el registro de usuarios, gestión información de presencia, mensajería, contactos de los usuarios y suministro de interoperabilidad con proveedores de mensajería externos. Cuenta con dos submódulos descritos a continuación:

#### **3.3.3.1 Servicios Jabber**

Encargado de prestar los servicios definidos por el protocolo Jabber, entre los más importantes se tiene la conexión, envío de mensajes y manejo de presencia. Se comunica con el módulo de control central para efectos de notificación, transporte de mensajes y presencia, y con el módulo transporte para establecer comunicación con los sistemas de mensajería instantánea MSN, AOL e ICQ.





### **3.3.3.2 Modulo de Transportes**

Es una extensión del módulo de servicios Jabber, que permite establecer comunicación con otras redes IM. Este modulo desempeña el papel de representación de los clientes móviles ante los servidores de mensajería instantánea propietarios, con el fin de dar a la plataforma una de sus principales características, la interoperabilidad.

### **3.3.4 Modulo de Control Central**

Este módulo se encarga de coordinar toda la funcionalidad de la plataforma. Se compone de tres submódulos descritos a continuación.

#### **3.3.4.1 Control**

Encargado de implementar la lógica del negocio del sistema y llevar a cabo la coordinación de mensajes entre los módulos restantes.

#### **3.3.4.2 Tarificación**

Lleva a cabo dos tareas de registro de datos y generación de información de utilidad para la administración de la plataforma. La primera tarea es registrar el uso de la plataforma por parte de los usuarios, en donde se tiene en cuenta el tráfico cursado y las horas pico de utilización. La segunda, es registrar el consumo por parte de los usuarios, en donde se tiene en cuenta el número de sesiones iniciadas, mensajes enviados, mensajes recibidos y sesiones de Chat establecidos.

#### **3.3.4.3 Gestión de Usuarios**

Encargado de gestionar la información de los usuarios. Provee las funcionalidades de registro, eliminación, edición y búsqueda de usuarios.

## **3.4 Glosario**

### **3.4.1 Actores**

- **Emisor:** este actor es la generalización de los 3 actores que se describen a continuación, los cuales realizan tareas comunes: cambio de presencia y el inicio de



chat. Por lo tanto, es un actor que se incluye dentro de este modelo para definir las características similares que tienen el usuario IM, el administrador y el servidor IM.

- **Usuario IM:** representa al usuario común del servicio de mensajería instantánea, el cual hace uso de toda la funcionalidad ofrecida, tanto de comunicación como de gestión de usuario y contactos.

Es el actor que inicia y termina las sesiones IM, modifica los datos personales y contraseña de su cuenta de usuario, adiciona y elimina los contactos de su lista, y principalmente envía y recibe mensajes instantáneos en el transcurso de una comunicación con sus contactos.

- **Administrador:** este actor representa a la persona que realiza todas las labores de administración de la plataforma, hace la gestión de usuarios de mensajería instantánea, evaluación de las estadísticas de tráfico y uso, así como el establecimiento de sesiones de chat con los usuarios IM.
- **Servidor IM:** este actor, a diferencia de los anteriores, representa un sistema que interactúa con PAUMIM. El servidor IM, simboliza a un servidor de mensajería instantánea propietario o libre, el cual realiza tareas de comunicación como: cambiar presencia e iniciar chats.

### 3.4.2 Vistas

- **Vista de los Actores:** especifica las relaciones de generalización descritas para los actores del sistema.
- **Vista del Administrador:** en esta vista se muestran los casos de uso relacionados con el administrador del sistema y que además están dentro del subsistema Módulo Administrativo.
- **Vista del Usuario:** debido a que muestra el subsistema más amplio en cuanto a interacción con el usuario, es la vista que incluye la mayor cantidad de casos de uso, los cuales definen toda la funcionalidad que ofrece el sistema para el usuario, y además representa al módulo de conexión móvil.

- **Vista del Servidor:** esta vista representa la funcionalidad del subsistema: módulo de interoperabilidad, en el cual se han identificado los acciones que se realizan entre el servidor IM y PAUMIM.

### 3.5 Modelo de Casos de Uso

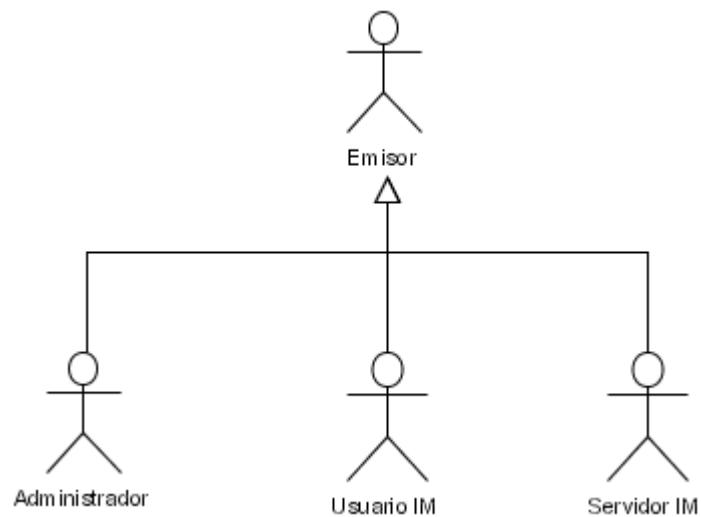


Figura 3.13. Diagrama de Casos de Uso, vista de los actores

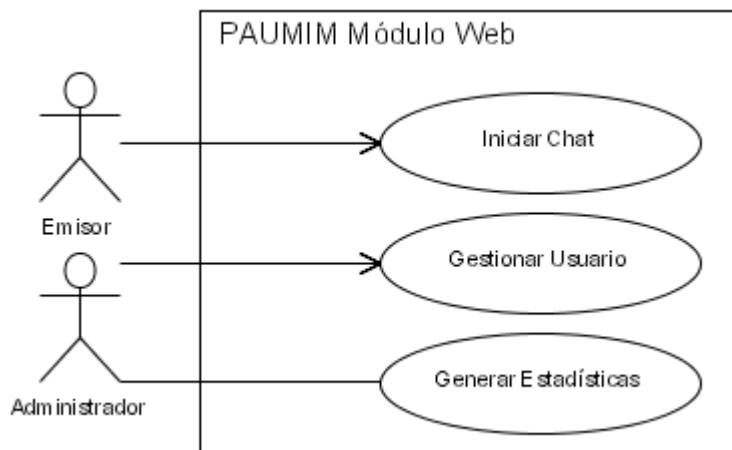


Figura 3.14. Diagrama de Casos de Uso, vista del administrador

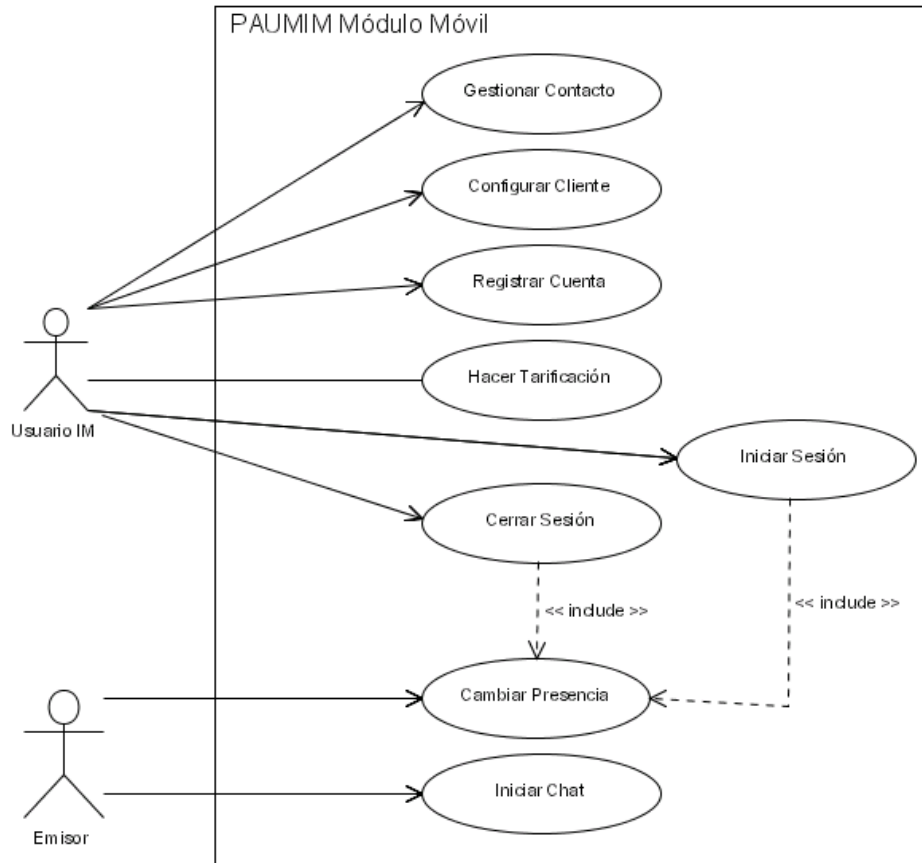


Figura 3.11. Diagrama de Casos de Uso, vista del usuario

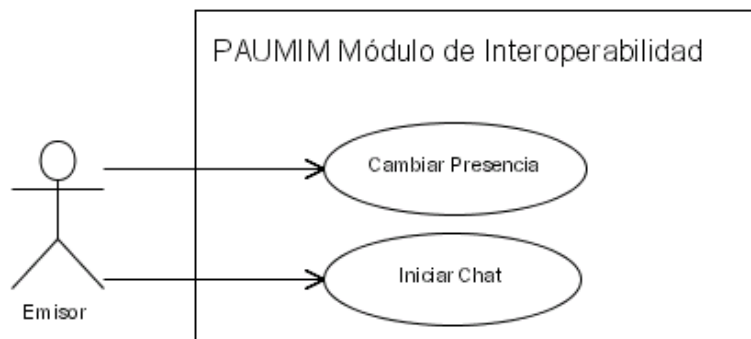
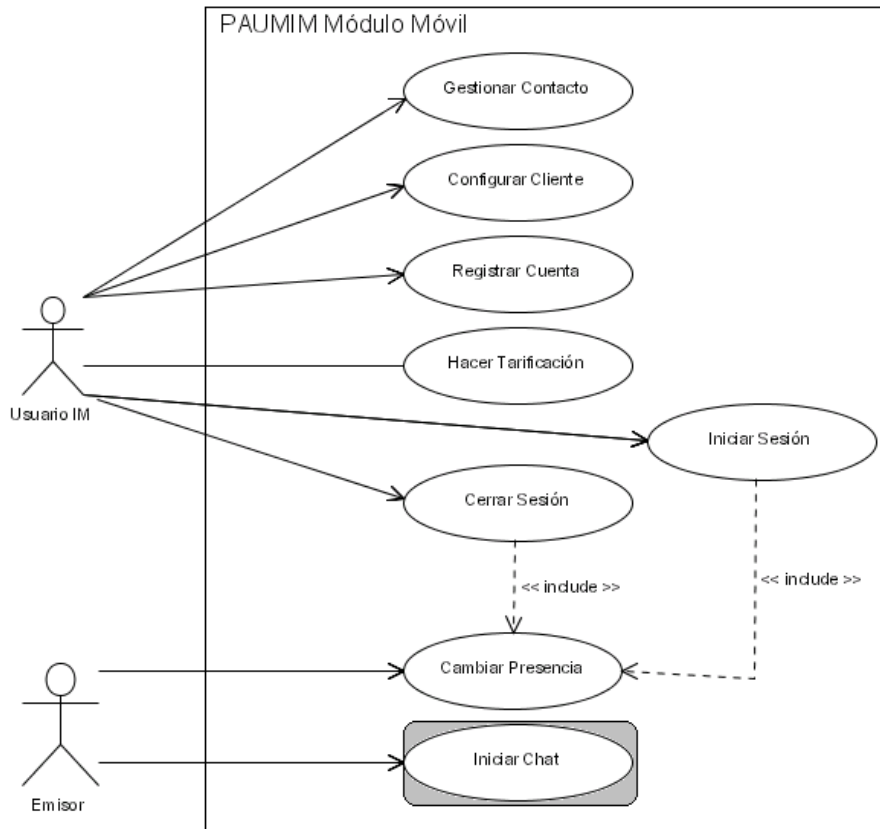


Figura 3.12. Diagrama de Casos de Uso, vista del servidor

### 3.6 Diseño Detallado de Caso de Uso Iniciar Chat

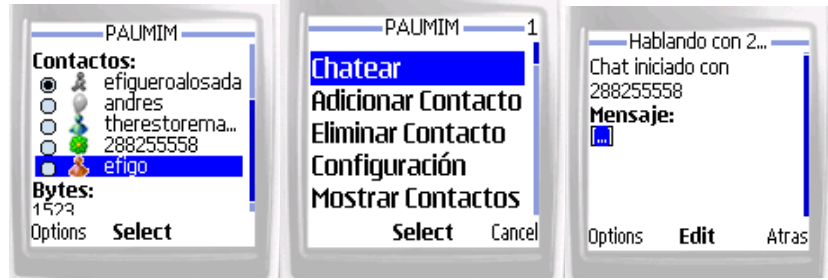
#### 3.6.1 Diagrama de Caso de Uso

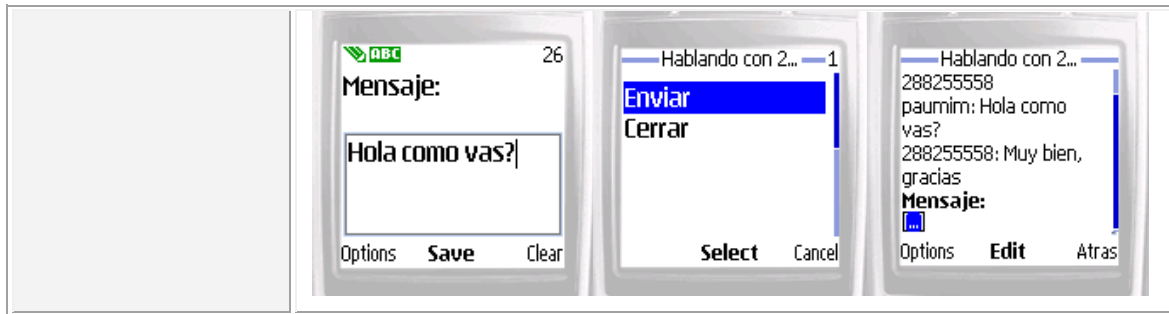


**Figura 3.13. Diagrama de Casos de Uso a implementar**

<b>Caso de Uso</b>	Iniciar Chat
<b>Actores</b>	Usuario IM(iniciador), Servidor IM
<b>Tipo</b>	Primario
<b>Resumen</b>	<p>El caso de uso comienza cuando un usuario IM, desde un dispositivo móvil, desea iniciar una conversación con uno de sus contactos. Entonces el usuario IM primero elige de su lista, al contacto elegido para tal propósito, seguidamente escribe el mensaje de texto y da la orden de enviar.</p> <p>El sistema por su parte, se encarga de manejar el direccionamiento del mensaje de acuerdo a la red a la cual pertenece el usuario de destino, y hacerlo llegar hasta su ventana de chat.</p>
<b>Precondiciones</b>	Iniciar Sesión, disponibilidad en la presencia del usuario IM receptor



<b>Poscondiciones</b>	Un nuevo mensaje ha llegado al usuario IM receptor.	
<b>Flujo Normal</b>	<b>Actores</b> <ol style="list-style-type: none"><li>1. El usuario IM elige el destinatario de su lista de contactos.</li><li>3. El usuario IM escribe el mensaje y da la orden de enviar.</li><li>8. El servidor IM reenvía el mensaje hacia el usuario IM receptor.</li></ol>	<b>Sistema</b> <ol style="list-style-type: none"><li>2. Muestra una pantalla de chat.</li><li>4. Agrega el mensaje a la ventana de chat del usuario IM emisor.</li><li>5. Envía el mensaje desde el dispositivo móvil hasta el servidor PAUMIM.</li><li>6. El servidor PAUMIM verifica la dirección de destino para encaminar el mensaje hacia la pasarela adecuada.</li><li>7. Una vez el mensaje llegue a la pasarela, se hace la conversión de protocolos para que el mensaje se adecue a la red de mensajería propietaria y llegue al servidor IM.</li><li>9. El mensaje llega al usuario IM receptor y se adiciona a la ventana de chat.</li></ol>
<b>Cursos Alternos</b>		
<b>GUIs Relacionadas</b>		



### 3.6.2 Diagrama de Clases del Cliente Móvil

#### 3.6.2.1 Diagrama de Clases del Cliente Móvil MIDP 1.0

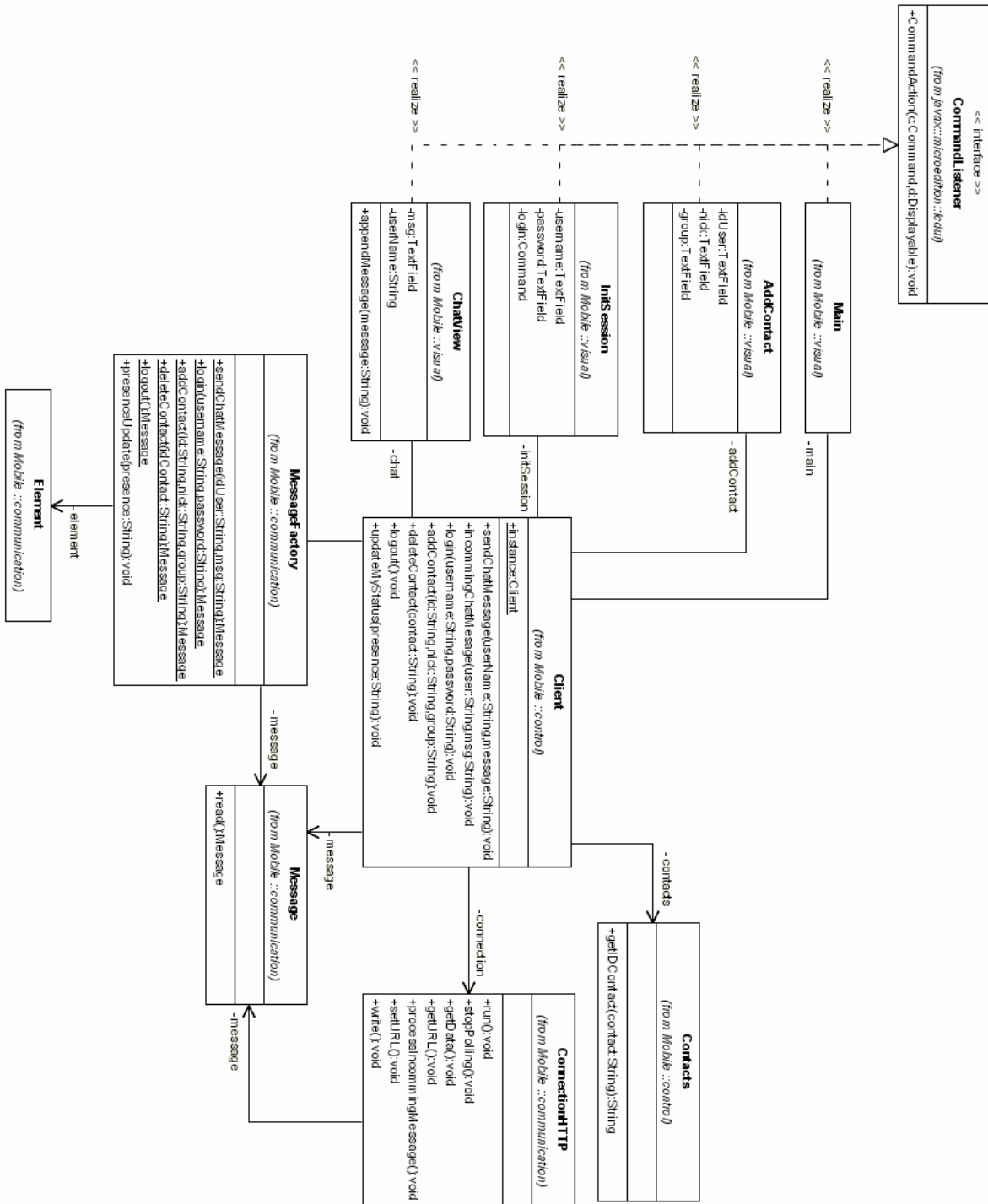


Figura 3.13. Diagrama de Clases del Cliente Móvil MIDP1.0

### 3.6.2.2 Diagrama de Clases del Cliente Móvil MIDP 2.0



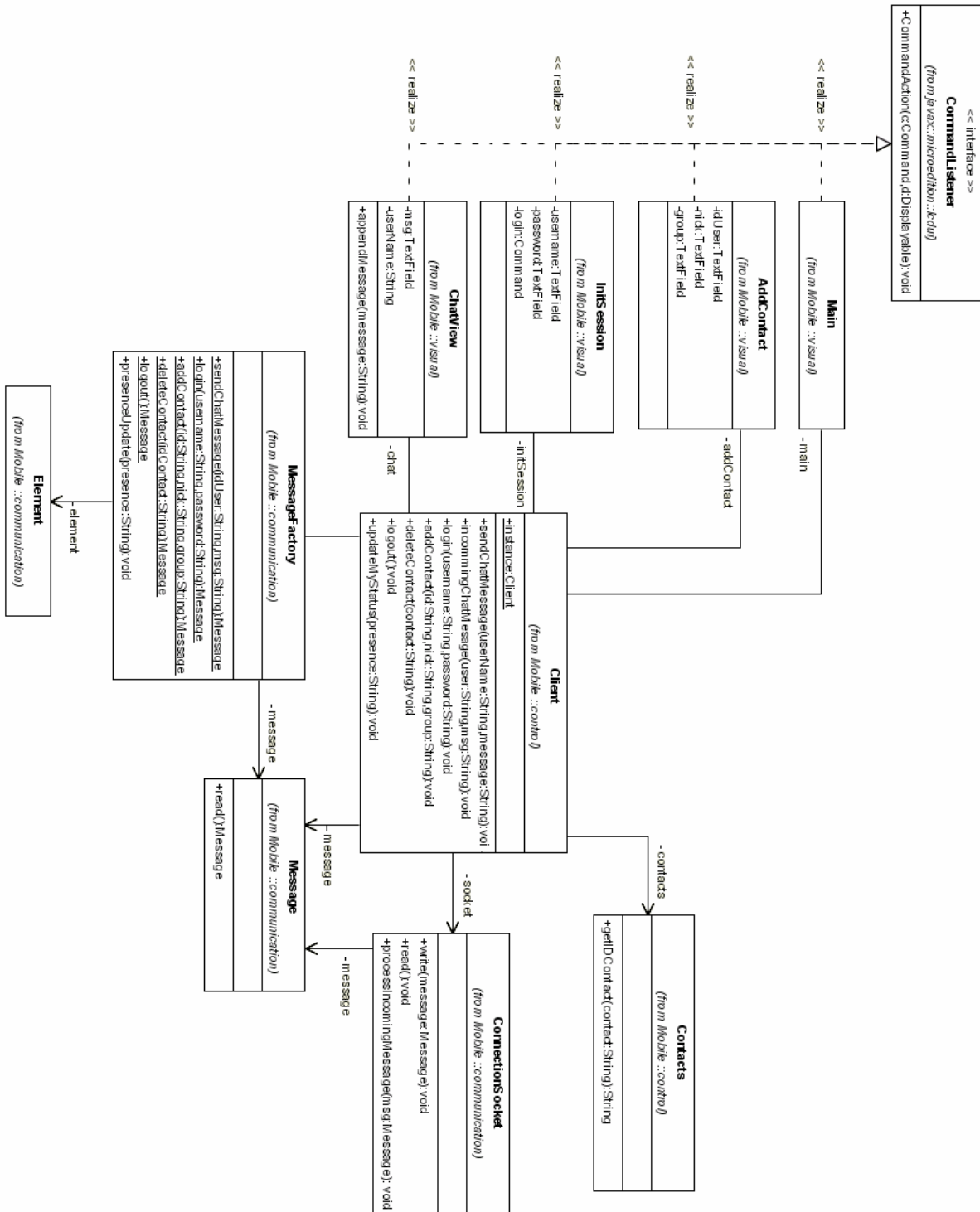


Figura 3.14. Diagrama de Clases del Cliente Móvil MIDP2.0

### 3.6.3 Diagrama de Clases del Servidor

#### 3.6.3.1

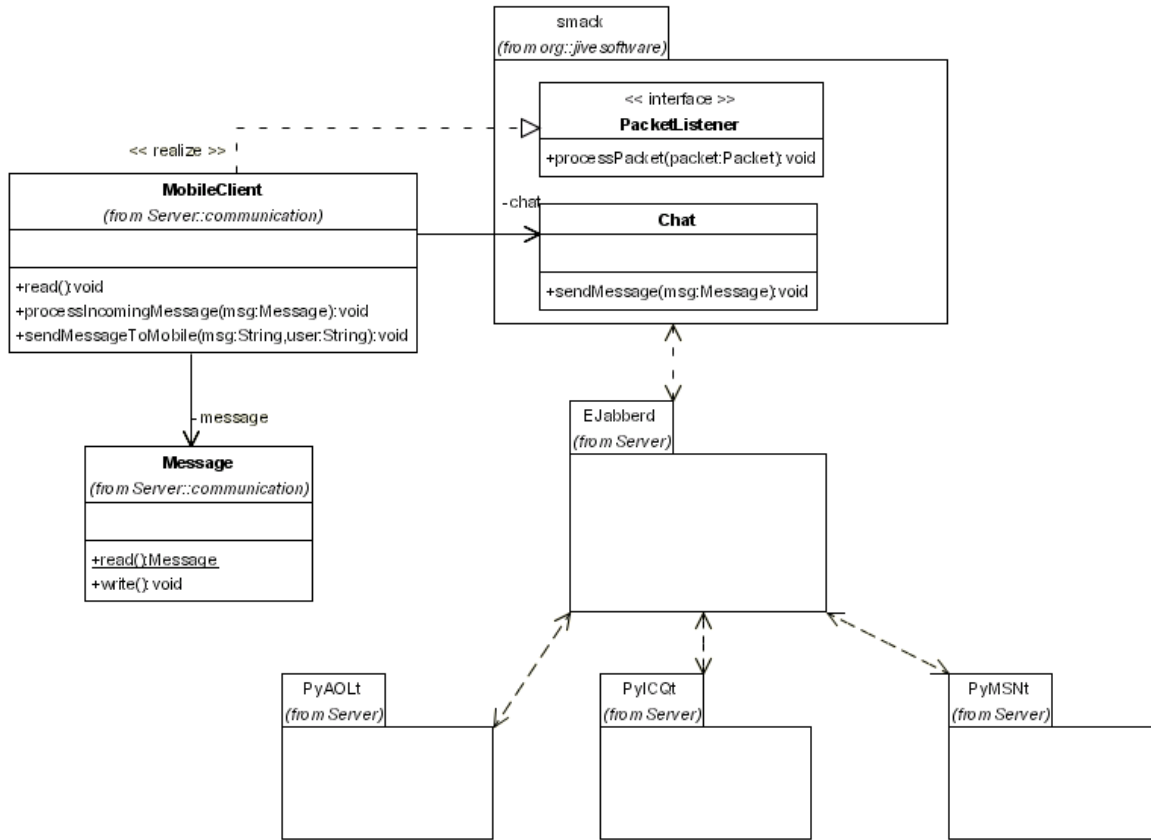


Figura 3.15. Diagrama de Clases del Cliente Servidor

### 3.6.4 Diagramas de Secuencia

#### 3.6.5 Diagrama de Secuencia de Cliente Móvil

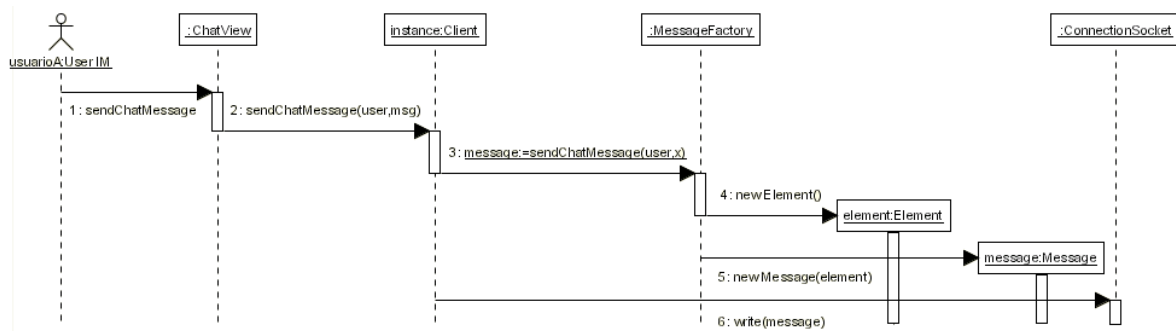


Figura 3.16. Diagrama de Secuencia del cliente móvil emisor

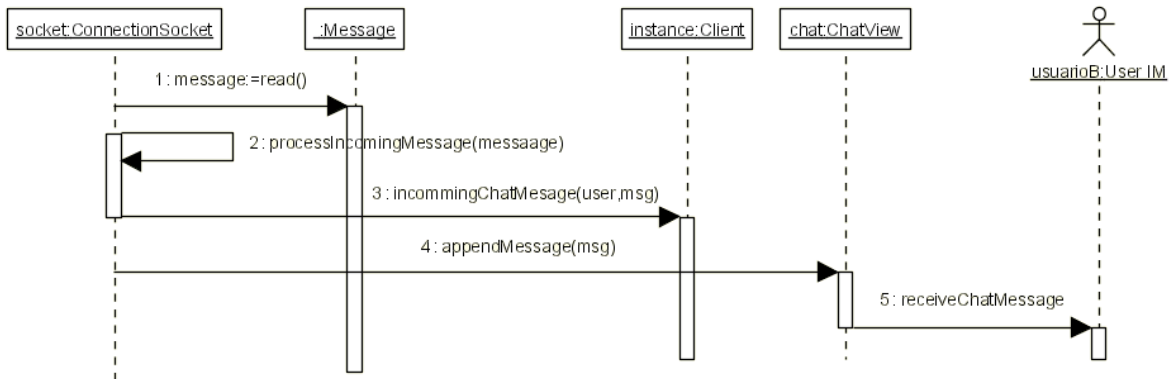


Figura 3.17 Diagrama de Secuencia del cliente móvil receptor

### 3.6.6 Diagrama de Secuencia del Servidor

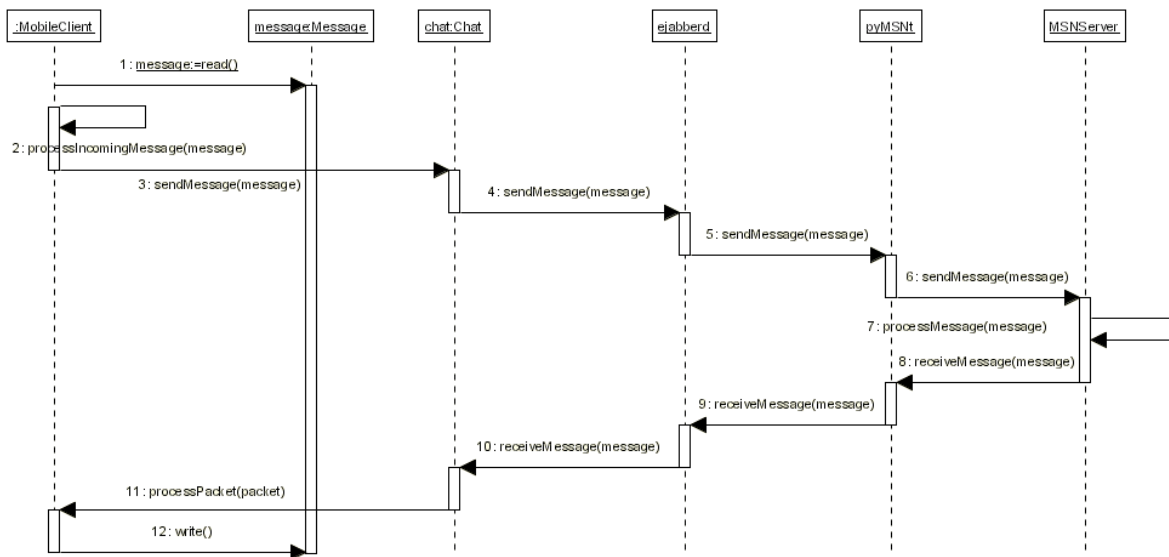


Figura 3.18. Diagrama de Secuencia del cliente móvil receptor

### 3.6.7 Diagrama de Paquetes

La arquitectura del sistema con base a los paquetes se muestra en las Figuras 3.23 y 3.24 correspondientes a los paquetes del móvil y del servidor. En estos diagramas de paquetes no se tuvo en cuenta los paquetes propios del núcleo de Java. Para el diagrama de paquetes del servidor se hizo una división de la arquitectura de tres niveles. En el primer nivel se encuentran los paquetes que corresponden a módulos funcionales de la plataforma externos al ambiente de desarrollo Java. El segundo nivel corresponde a



APIs de terceras partes adicionadas a manera de bibliotecas, y en el tercer nivel están los paquetes que contienen las clases que fueron implementadas por completo. Para el diagrama de paquetes del cliente se identificó un solo nivel, el correspondiente a las clases implementadas por completo.

### Servidor

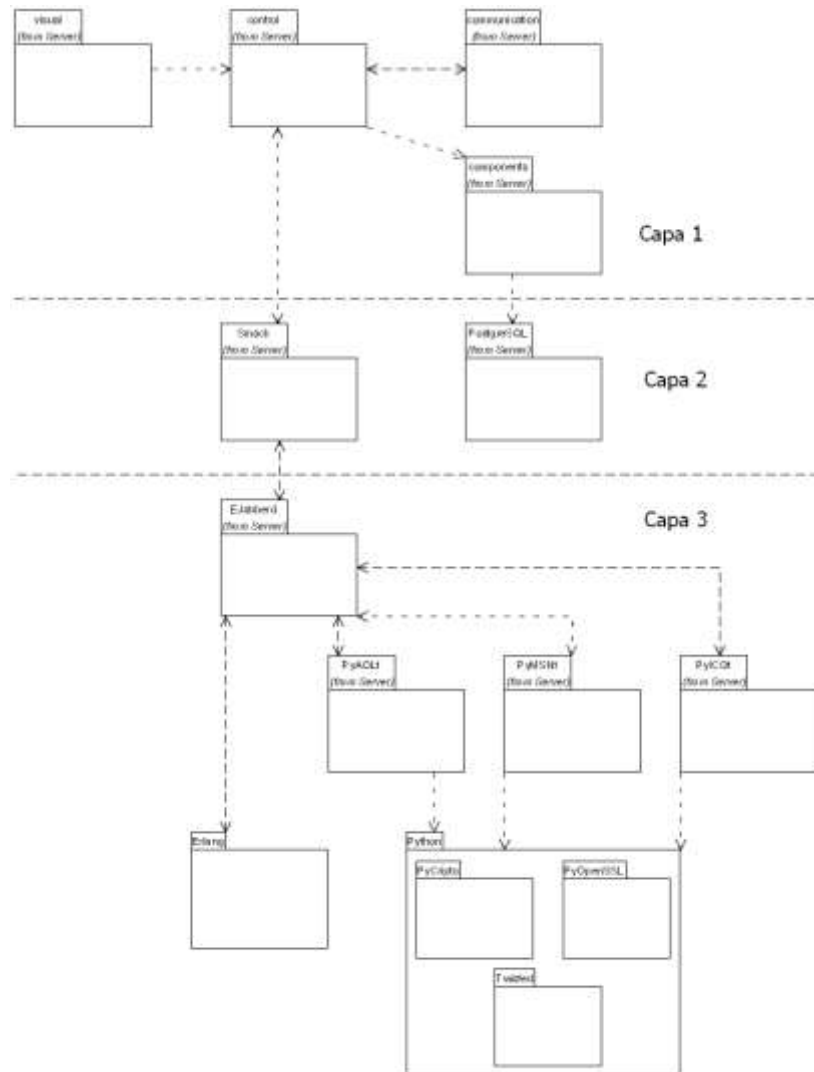


Figura 3.19 Diagrama de paquetes del servidor

### Móvil

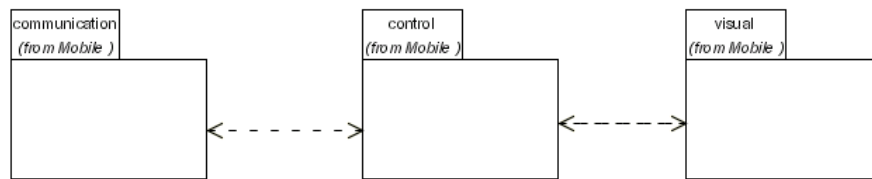


Figura 3.20. Diagrama de paquetes del cliente móvil

## 3.6.8 Descripción de Paquetes

### 3.6.8.1 Servidor

#### Primer nivel

**python:** Lenguaje interpretado utilizado para el desarrollo del módulo de transportes. Python se destaca por ser un lenguaje desarrollado como proyecto de código abierto, multiplataforma, permite dividir las aplicaciones en módulos reutilizables desde otros programas y está compuesto por una gran colección de módulos estándar que se pueden utilizar como base de las aplicaciones [29].

**pyOpenSSL:** Provee herramientas al Python para crear conexiones seguras. Conformar una capa de alto nivel alrededor de una subconfiguración de la librería OpenSSL, incluye Objetos de conexión SSL, reciclaje de métodos de los sockets portables de Python, Callbacks escritas en Python, Mecanismo extensivo de gestión de errores, reflejo de los códigos de error OpenSSL entre otros [30].

**pyCripto:** Provee herramientas de cifrado y descifrado de información a Python. Contiene recursos como el manejo de funciones hash: MD2, MD4, RIPEMD, SHA256, algoritmos de encriptación de bloques: AES, ARC2, Blowfish, CAST, DES, Triple-DES, IDEA, RC5, algoritmos de encriptación de 'streams': ARC4, simple XOR, algoritmos de clave pública: RSA, DSA, ElGamal, qNEW y protocolos de transformaciones 'todo-o-nada', 'chaffing/winning' [31].

**Twisted:** Es un Framework de código abierto implementado en Python especializado para el desarrollo de aplicaciones basadas en red. Sirve de soporte al paquete python para establecer y administrar conexiones de red. Está diseñado para correr en múltiples sistemas operativos y plataformas. Twisted es un intento de construcción de un



Framework capaz de soportar las nuevas y modernas aplicaciones basadas en redes, desde un simple protocolo hasta un sistema integrado, multiescala y multiprotocolo. Provee múltiples niveles de abstracción, comenzando desde eventos de red de bajo nivel, pasando por código de redes genérico, implementaciones de protocolos de red hasta Frameworks de alto nivel [32].

**Ejabberd:** Es un servidor Jabber multiplataforma. Desarrollado en el lenguaje de programación Erlang, cuenta con el soporte total de las características del protocolo Jabber [33].

**Erlang:** Lenguaje funcional utilizado especialmente para desarrollo de aplicaciones distribuidas. Tiene soporte para concurrencia, distribución y tolerancia de fallos. Utilizado para el desarrollo del servidor Ejabberd [34].

**PyMSNt:** Es un transporte desarrollado en Python. Provee una pasarela con la cual el servidor puede comunicarse con la red de MSN Messenger. El transporte debe estar instalado en el servidor Jabber, y su operación es transparente para el usuario. Este podrá interactuar con sus contactos MSN como si fueran contactos Jabber [35].

**PyAOL – t:** Es un transporte desarrollado en Python. Provee una pasarela con la cual el servidor puede comunicarse con la red de AOL. El transporte debe estar instalado en el servidor Jabber, y su operación es transparente para el usuario. Este podrá interactuar con sus contactos AOL como si fueran contactos Jabber [36].

**PyICQ – t:** Es un transporte desarrollado en Python. Provee una pasarela con la cual el servidor puede comunicarse con la red de ICQ Messenger. El transporte debe estar instalado en el servidor Jabber, y su operación es transparente para el usuario. Este podrá interactuar con sus contactos ICQ como si fueran contactos Jabber [37].

## Segundo nivel

**Smack API:** Es una librería de código abierto para construir clientes de mensajería instantánea en lenguaje de programación Java. Puede ser embebida en aplicaciones para crear componentes de presencia y mensajería instantánea basados en el protocolo



Jabber. Utilizada para establecer la comunicación entre el módulo de conexión móvil y el módulo de servicios Jabber [38].

**PostgresLibs:** Este conjunto de librerías permite a los programas Java conectarse al motor de base de datos PostgreSQL. Contiene un driver tipo IV construido en su totalidad en código Java que provee en gran parte la implementación de la especificación JDBC 3 en conjunto con otras extensiones específicas para PostgreSQL [39].

### **Tercer nivel**

**vista:** En este paquete se encuentran las páginas JSP, a través de las cuales se lleva a cabo la comunicación entre la plataforma y el administrador. Por medio de estas páginas, el administrador puede observar las variables de estado de la plataforma y así realizar funciones de administración. Todas las páginas JSP se comunican con el paquete de control.

**control:** Contiene las clases que sirven para hacer el control de la lógica de la aplicación y la coordinación de la comunicación entre módulos y entre la plataforma y el administrador.

**componentes:** Tiene inmersas las clases Java propietarias que implementan todas las funcionalidades de la plataforma. Cada una de las clases representa un componente encargado de llevar a cabo una funcionalidad específica. En este paquete se cuenta con componentes que llevan a cabo tareas de tarificación, gestión de usuarios, funcionalidades de administración y manejo de presencia de usuarios. El componente de tarificación permite almacenar las sesiones de tarificación de cada uno de los usuarios, cargar estadísticas de tarificación de usuario y de la plataforma. El componente de gestión de usuarios permite llevar a cabo las tareas de registro, eliminación y edición de cuentas de usuarios. El componente encargado de las funcionalidades de administración provee métodos para llevar a cabo la validación del administrador, inicio y detención del servicio de mensajería, consulta de información y estadísticas de uso de la plataforma. El componente de manejo de presencia es el encargado de registrar la presencia de cada uno de los usuarios del sistema, provee además la información de usuarios conectados y no conectados al administrador.



### 3.6.8.2 Móvil

**vista:** En este paquete se encuentran las clases que sirven para llevar a cabo la comunicación directa entre el sistema y el usuario, por medio de ellas, el usuario puede observar el estado de la aplicación y de la misma forma comunicarse con ella. Todas las clases se comunican de forma directa con la clase de control Client.

**control:** Contiene las clases que sirven para hacer el control de la lógica de la aplicación y la coordinación de la comunicación entre esta, el usuario y el servidor. La clase principal de este paquete permite manejar el ciclo de vida de la aplicación.

**comunicación:** En este paquete se ubican las clases involucradas en la comunicación entre el cliente móvil y el servidor PAUMIM. Para establecer dicha comunicación, se utilizan clases propietarias para el establecimiento y mantenimiento de la conexión y las clases definidas en JXME (JXTA para J2ME) para el ensamble y desensamble de los mensajes de comunicación.

### 3.6.9 Diagrama de Implantación

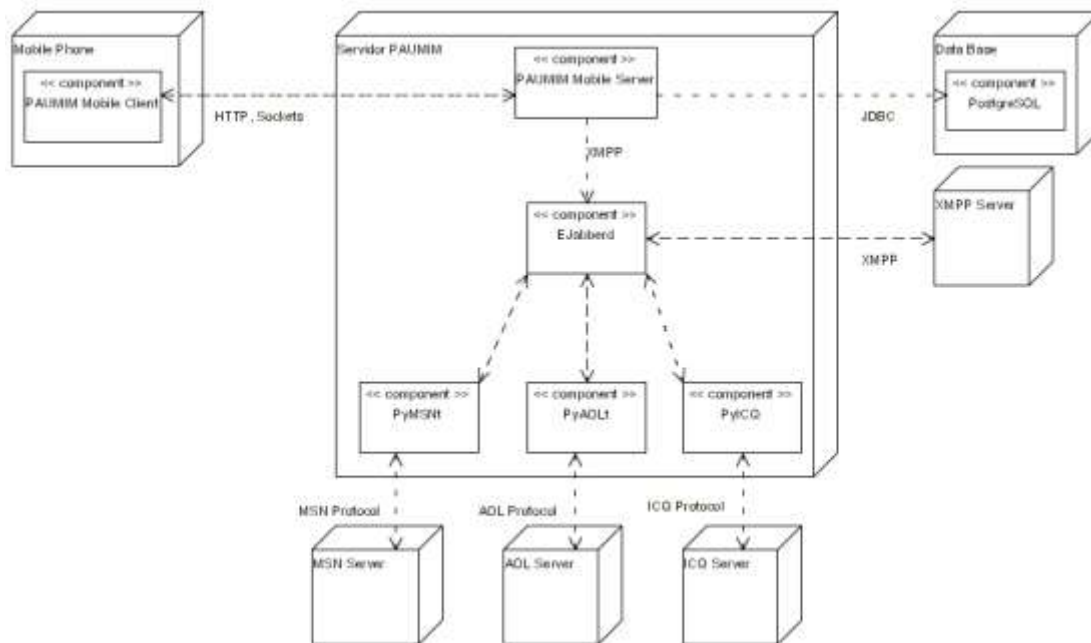


Figura 3.21. Diagrama de implantación del sistema





## 3.6.10 Características Técnicas

### 3.6.10.1 Teléfono Móvil

Este nodo representa al dispositivo móvil y el cliente PAUMIM. Es utilizado por los usuarios para acceder a los servicios de la plataforma mediante la utilización de dispositivos móviles como teléfonos celulares de gama media, alta, smartphones y computadores de mano tipo PDA.

Para el correcto funcionamiento de este nodo es necesario instalar la aplicación móvil. Para ello se debe tener en cuenta los mínimos requerimientos técnicos del dispositivo:

- Soporte para aplicaciones Java MIDP 1.0 o MIDP 2.0
- Pantalla a color
- Conexión a Internet

### 3.6.10.2 Servidor PAUMIM

Representa el servidor y su conjunto de componentes. Ofrece el servicio de acceso universal a mensajería instantánea para clientes móviles y la interoperabilidad con proveedores externos de mensajería instantánea.

Para llevar a cabo una adecuada instalación de los componentes del servidor y para garantizar el correcto funcionamiento, se debe tener en cuenta las siguientes versiones de los paquetes:

- **Entorno de Ejecución Java:** JRE 1.5
- **Servidor de Aplicaciones:** JBoss 4.0.1 SP1
- **Servidor Jabber:** Ejabberd v.1.0.0
- **Python:** Python v.2.3.5
- **Soporte Twisted para Python:** Twisted v.1.3.0
- **Soporte Cripto para Python:** PyCripto v.2.0
- **Soporte OpenSSL para Python:** PyOpenSSL v.0.4
- **Transporte AIM:** PyAIMt v.0.7b
- **Transporte ICQ:** PyICQt v.0.5a
- **Transporte MSN:** PyMSNt v.0.9.5



Se debe tener mucho cuidado con el manejo de versiones de los distintos módulos ya que no hay total compatibilidad entre ellas. El servidor se encuentra instalado en un computador personal con las siguientes características:

- **Procesador y tarjeta madre:** Athlon Semprom 2200+
- **Memoria RAM:** 512 Mb DDR 400Mhz,
- **Disco duro:** 40 Gb.
- **Sistema operativo:** Windows XP y Linux distribución Debian.

### 3.6.10.3 Data Base

Este nodo es el encargado de manejar la persistencia de la plataforma. Guarda datos de tarificación y cuentas de usuario, uso de la plataforma y configuración.

Está compuesto por un motor de bases de datos. En este nodo se instalaron las siguientes versiones de los componentes para la base de datos:

- **Motor de bases de datos:** PostgreSQL 8.0
- **Ciente de base de datos:** PGAdmin 1.4

El motor de bases de datos está instalado en el nodo servidor.



## **4 CAPITULO 4: EXPERIMENTACIÓN**

### **4.1 Mensajería Móvil con PAUMIM**

La Plataforma de Acceso Universal a Mensajería Instantánea Móvil ofrece el servicio de mensajería instantánea a través de dispositivos móviles, el cual viene acompañado de un completo sistema de presencia que brinda la posibilidad al usuario de establecer comunicaciones instantáneas e identificar el estado de presencia de sus contactos. En este capítulo se hace un análisis de las tecnologías utilizadas para la construcción de PAUMIM y se exponen los resultados obtenidos en el transcurso del proyecto.

### **4.2 Entorno de Ejecución**

El entorno de ejecución en el cual se realizó la construcción y las pruebas de la plataforma tiene las siguientes características:

#### **4.2.1 Hardware**

##### **4.2.1.1 Servidor**

- Computador personal, Sempron 2200+ con 512 Mb de memoria RAM DDR 400Mhz, disco duro de 40 Gb. Windows XP y Linux distribución Debian.
- Punto de acceso inalámbrico D-Link 2100AP 108Mbps 802.11 b/g

##### **4.2.1.2 Móvil**

- PDA Palm Drive.
- PDA Palm Tungsten T5
- Teléfono celular Sony Ericsson T610 y Nokia 6230.



## **4.2.2 Software**

### **4.2.2.1 Servidor**

#### **Entorno de desarrollo**

- J2SDK 1.5
- Eclipse 3.1
- MyEclipse Enterprise Workbench 4.0.3

#### **Base de datos**

- PostgreSQL 8.0
- PGAdmin 1.4

#### **Servidor de aplicaciones**

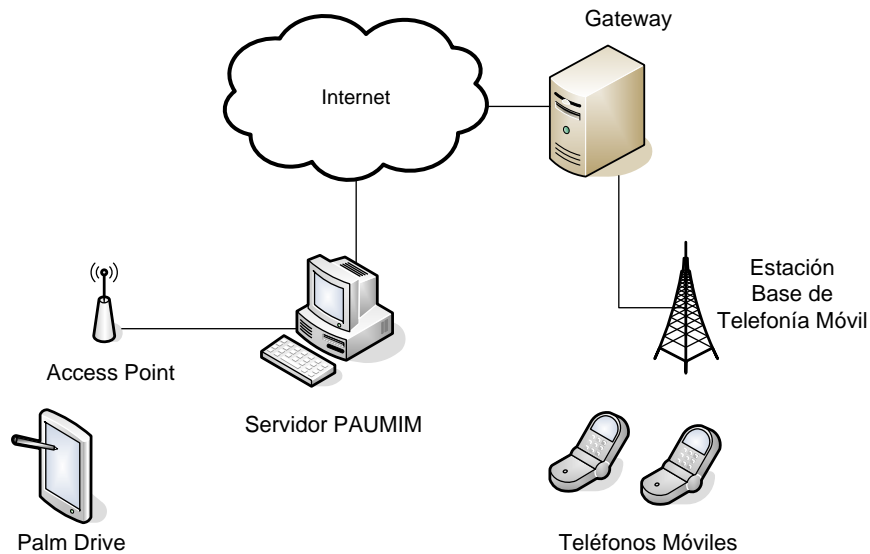
- JBoss 4.0

### **4.2.2.2 Móvil**

#### **Herramientas de desarrollo de aplicaciones para dispositivos móviles**

- Nokia Developer Suite 2.2
- Sony Ericsson Developer Kit 2.2.3
- Wireless Toolkit 2.2

En la Figura 4.1 se puede observar el diagrama de pruebas de la plataforma.



**Figura 4.1. Diagrama de pruebas de la plataforma**

### **4.3 Cliente Móvil**

#### **Procedimiento de implementación**

Construido con la tecnología Java 2 Micro Edition. El conjunto de APIs ofrecido por la tecnología fue suficiente para desarrollar una aplicación de alta calidad y bajo los lineamientos de usabilidad descritos en el capítulo dos (Ver sección 2.2.1). Para abordar la construcción de las dos versiones de clientes móviles se optó por construir una primera versión estándar, con las características necesarias para ser la base en la construcción de las aplicaciones. El diseño de esta primera versión se realizó aplicando patrones de diseño con el objetivo de dar la extensibilidad y adaptabilidad necesarias para extender hacia las dos versiones posteriores. La versión estándar consta del conjunto de componentes entre los cuales tenemos: interfaz gráfica, construida a partir de la librerías LCDUI utilizando el conjunto de clases definidas para MIDP 1.0; generación de mensajes de comunicación, construido con base a las clases definidas en la arquitectura JXME para en intercambio de mensajes entre cliente y servidor; manejo de errores y excepciones, encargado de notificar al usuario de eventos y comportamientos fuera de los normales, construido con clases de interfaz gráfica LCDUI y manejo de excepciones Java; por último la lógica de la aplicación, encargada de manejar la comunicación directa con el usuario y con el servidor PAUMIM. A partir de esta aplicación genérica se construyeron las dos versiones de clientes móviles descritos a continuación:



### **4.3.1 Cliente MIDP1.0**

A partir de la versión estándar, se incorporó un módulo de comunicación basado en conexiones HTTP, el cuál se adaptó fácilmente con el módulo de construcción de mensajes y la lógica de la aplicación. Este módulo es el encargado de realizar el sondeo al servidor PAUMIM con el objetivo de consultar los mensajes que el servidor almacena temporalmente. El periodo de sondeo depende del estado de utilización de la aplicación, si el usuario está chateando, el periodo de actualización es de 3 segundos, y si está en estado de espera, el periodo es de 10 segundos. Estos intervalos de tiempo dan la sensación al usuario de que la aplicación está actualizándose en tiempo real. El rendimiento de este módulo es muy bueno, desde el punto de vista técnico, no sobrecargó la memoria del dispositivo ni disminuyó el rendimiento de la aplicación en tiempo de ejecución. En cuanto a costos, no son muy altos debido a que el mensaje de sondeo tiene un tamaño de 60 bytes. En tiempo de espera, se envían mensajes de sondeo cada 10 segundos, por lo tanto en una hora el costo por mensajes de sondeo es de aproximadamente 80 pesos. Para más información ver las secciones siguientes.

### **4.3.2 Cliente MIDP2.0**

Para la adaptación del cliente al perfil MIDP 2.0 se realizaron pequeñas modificaciones en cuanto a interfaz gráfica, utilizando nuevos métodos definidos para esta especificación y que al hacer uso de ellos se incrementa la eficiencia de la aplicación con relación al Cliente Estándar. Adicional a esto, se adaptó el módulo de comunicaciones basado en una conexiones TCP/IP bajo la utilización de Sockets TCP definidos en la nueva especificación. Gracias a esta tecnología la comunicación se realiza con mayor eficiencia, ya que se establece un circuito lógico de comunicación entre el cliente y el servidor, por medio del cual se pueden intercambiar mensajes de forma sincrónica. La carga en memoria de este módulo es muy baja en comparación al incremento de eficiencia en la comunicación. En este tipo de comunicación existe la posibilidad de definir un atributo de conexión que especifica un intervalo de sondeo para garantizar la permanencia y validez del circuito lógico mediante la utilización de mensajes keepalive definidos por la especificación. Dependiendo de la calidad del servicio de red es conveniente o no utilizar este atributo. Si el servicio de red de datos es muy bueno, no hay necesidad de utilizarlo ya que se tiene gran nivel de confianza de que el circuito permanezca disponible el tiempo



que dure la conexión, además de esto depende del nivel de uso de la aplicación por parte del usuario en un tiempo determinado. Si la señal de datos es muy pobre, es necesario activar la propiedad para obtener así una mayor confiabilidad de que el circuito permanecerá activo y disponible cuando se lo necesite.

El módulo de comunicación está en la capacidad de reconectarse cuando pierde conexión, de tal forma que si la conexión se cae, el módulo lleva a cabo intentos de reconexión que en caso de ser exitosos, en gran medida, para el usuario no es perceptible el inconveniente de desconexión. Si por el contrario, el inconveniente de desconexión persiste, se informa directamente al usuario del error.

### **4.3.3 Resultados**

Para desarrollar y ejecutar la aplicación móvil se utilizó el computador personal y los kits de desarrollo para aplicaciones basadas en dispositivos móviles. Se realizaron tres tipos de pruebas para verificar la validez de la solución.

- La primera en emuladores de Nokia, Sony Ericsson y Wireless Toolkit de Sun, cada uno equipado con herramientas que permiten observar el consumo de memoria de la aplicación y la cantidad de datos enviados y recibidos por la red.
- La segunda en una WLAN (Wireless Local Area Network), montada con un punto de acceso inalámbrico y acceso a los servicios por medio de una PDA Palm Drive y Tungsten T5.
- La tercera bajo la plataforma de datos de la red celular, en donde se montó la aplicación servidora en una máquina con IP real y el acceso a través de teléfonos celulares Sony Ericsson T610 para el cliente MIDP 1.0 y Nokia 6230 para el cliente MIDP 2.0.

### **Consumo de memoria**

A continuación, se muestra el rendimiento experimental observado para los clientes MIDP 1.0 y 2.0. En las Figuras 4.2 y 4.3 presentan datos de consumo de memoria de la aplicación en tiempo de ejecución para las funcionalidades más relevantes:

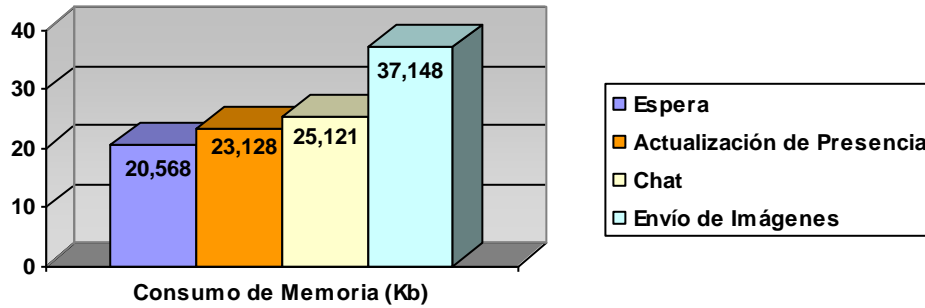


Figura 4.2. Consumo de memoria para el cliente MIDP 1.0

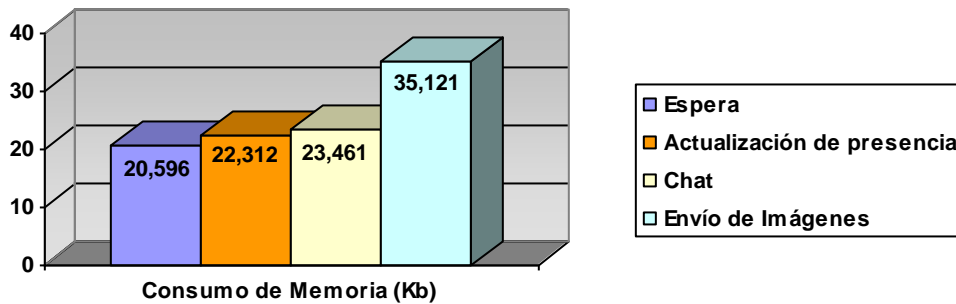


Figura 4.3. Consumo de memoria para el cliente MIDP 2.0

En las Figuras 4.4 y 4.5 se puede observar el consumo de memoria en el inicio de sesión de la aplicación y actualización de presencia en 4 ocasiones para los dos tipos de clientes:

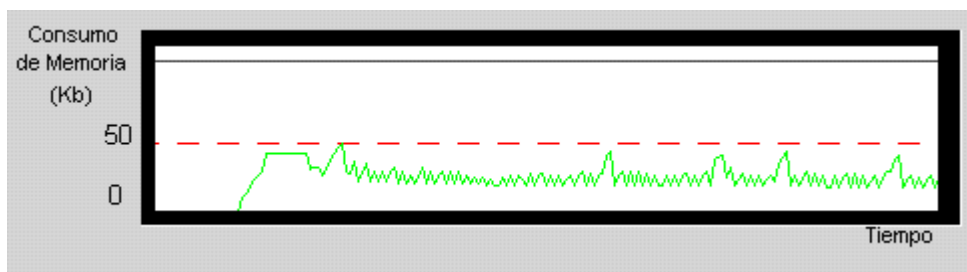
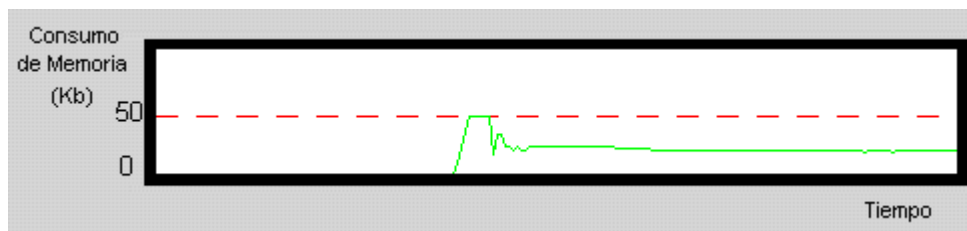


Figura 4.4. Consumo de memoria Cliente MIDP 1.0





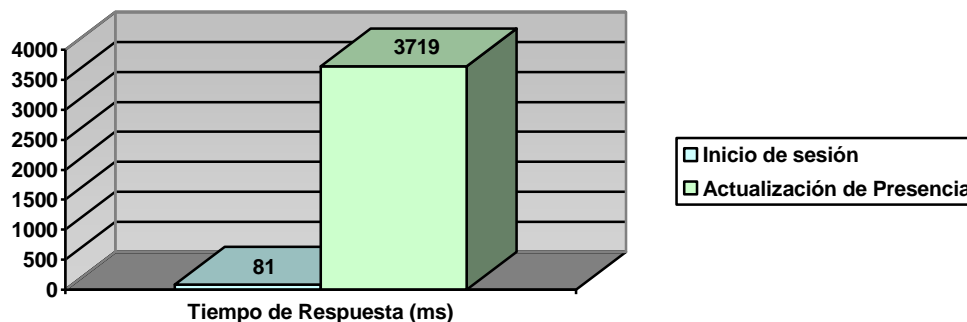


**Figura 4.5. Consumo de memoria Cliente MIDP 2.0**

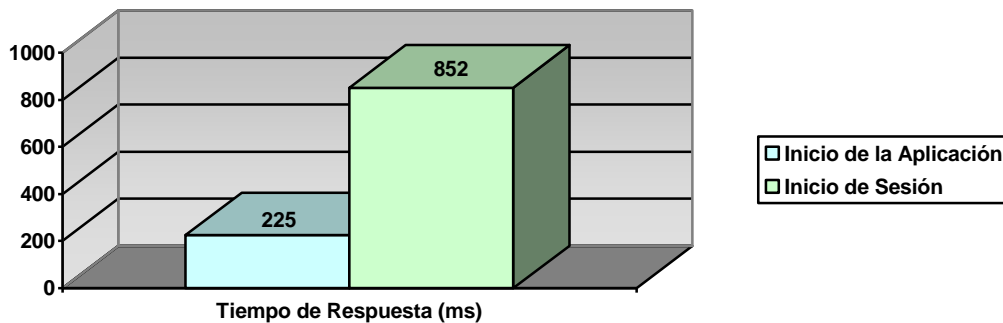
El principio de funcionamiento de los clientes MIDP 1.0 y 2.0 es distinto. Para los dos clientes hay un periodo de transición en el cual, el dispositivo carga en memoria la aplicación e inicia el periodo de conexión y validación de sesión. Una vez finalizada esta primera parte, el consumo de memoria toma cierta estabilidad dependiendo del tipo de cliente. En el cliente móvil MIDP1.0 se genera periódicamente un mensaje de sondeo y se envía al servidor, en este momento se crean objetos y se hace uso de la red, por esta razón la grafica muestra un comportamiento de diente de sierra ya que después de consumir recursos y enviar el mensaje al servidor, se limpia la memoria con el llamado al Garbage Collector. Para el cliente móvil MIDP 2.0 la gráfica de consumo de memoria es mucho más estable debido a que la comunicación con el servidor se realiza mediante un circuito lógico y no hay que crear objetos ni hacer uso de la red para conservar la conexión. En general el consumo de memoria para las dos aplicaciones es similar, teniendo mayor estabilidad y menor consumo el cliente MIDP2.0 por el principio de funcionamiento.

### Tiempo de respuesta

En las Figuras 4.6 y 4.7 se puede apreciar el tiempo de respuesta promedio de las aplicaciones para las funcionalidades más importantes en entorno de emulación.



**Figura 4.6. Tiempo de respuesta para el cliente MIDP 1.0**



**Figura 4.7. Tiempo de respuesta para el cliente MIDP 2.0**

El inicio de sesión consta del ensamble del mensaje con los datos de sesión para su envío al servidor, recepción de respuesta y actualización de la interfaz gráfica. Como se puede apreciar en las Figuras 4.6 y 4.7, el inicio de la aplicación es más rápido para el cliente MIDP1.0 pero su tiempo de respuesta al inicio de sesión es mucho más lento que para el cliente MIDP2.0. Lo anterior se debe a que el principio de funcionamiento para la recepción de conexiones por parte para los dos clientes es distinto. Para el cliente MIDP1.0, el servidor tiene que recibir y tratar información transportada en el protocolo HTTP, para lo cual debe empaquetar y desempaquetar la información según el stack de protocolos HTTP. Además de esto, el usuario debe descargar un mensaje por petición, lo cual, incrementa aún más el tiempo de respuesta de la aplicación. Para el cliente MIDP2.0 el servidor recibe la conexión TCP/IP y trata la información directamente sin necesidad de realizar empaquetamiento y los mensajes se envían a medida que se vayan creando, por lo tanto no hay cola de espera y el tiempo de retardo presente es únicamente el de transporte por la red.

### **Tamaño de mensajes**

En las Figura 4.8 se puede observar el tamaño de los mensajes para cada una de las funcionalidades del cliente móvil.

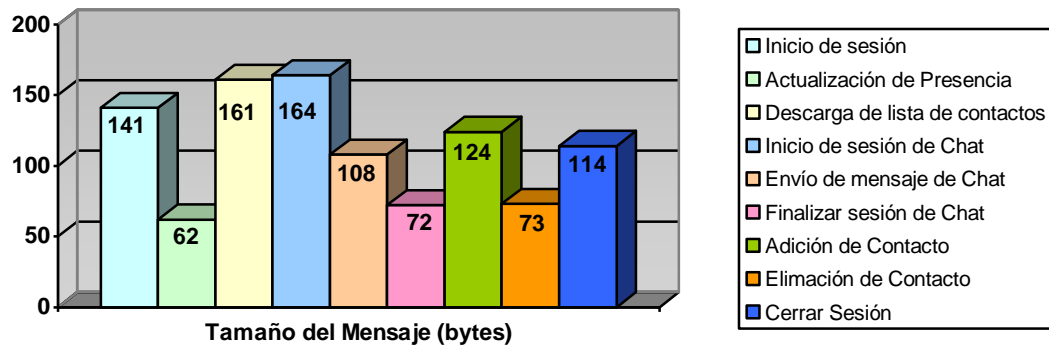


Figura 4.8. Tamaño de los mensajes para el Cliente Móvil (MIDP 1.0 y MIDP 2.0)

La diferencia entre el cliente MIDP1.0 y MIDP2.0 es que el primero debe realizar el sondeo al servidor y por ésta razón consume más ancho de banda. En la Figura 4.9 se puede observar el consumo del Cliente MIDP1.0 por concepto de sondeo en un minuto con relación a la funcionalidad específica.

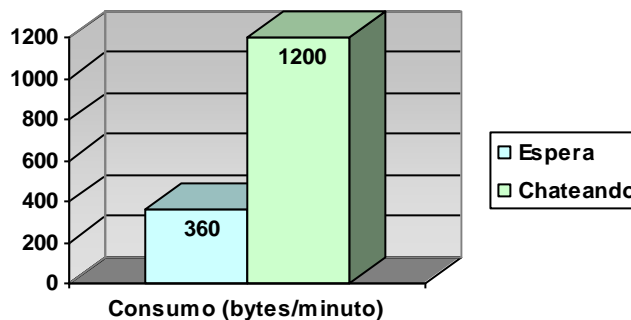


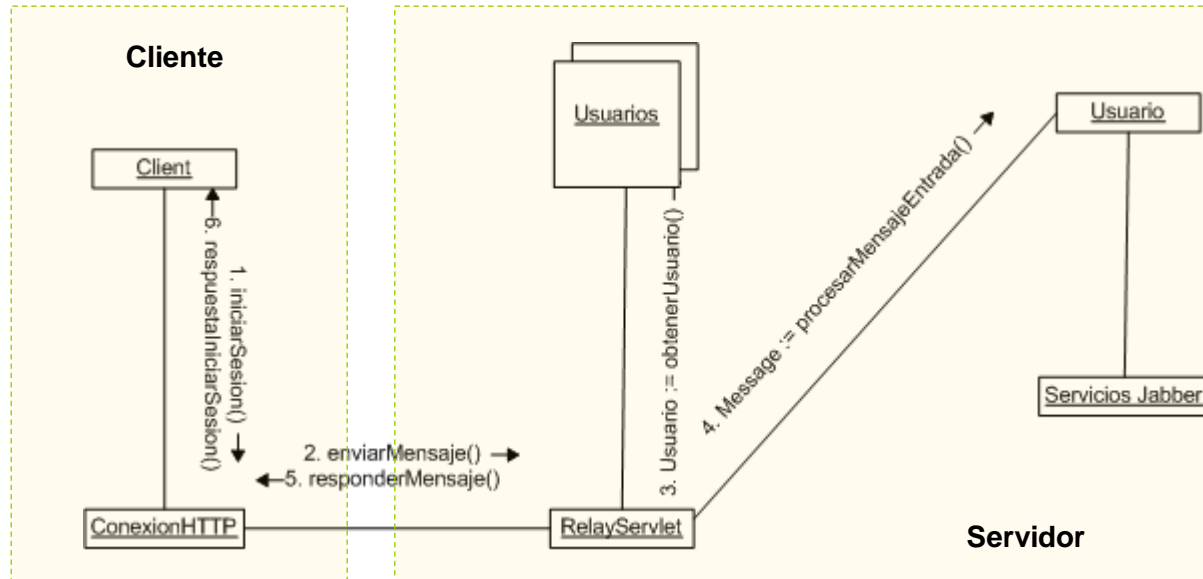
Figura 4.9. Consumo Cliente MIDP 1.0 por concepto de sondeo

#### 4.4 Módulo de Conexión Móvil

Este módulo consta de dos submódulos, cada uno para prestar el servicio de conexión a los clientes MIDP 1.0 y 2.0. A continuación se presenta una descripción de cada uno de ellos:

El primero, encargado de prestar el servicio de conexión a los clientes MIDP1.0 recibe peticiones HTTP por medio de un Servlet, el cual maneja la recepción y entrega de mensajes pertenecientes a cada uno de los clientes. Según la especificación J2EE, un

Servlet es un componente multihilo que puede atender un gran número de peticiones concurrentemente y debido a esto, las características de escalabilidad se cumplen para este módulo. En la Figura 4.8 se puede observar un diagrama de colaboración de inicio de sesión.



**Figura 4.10 Diagrama de colaboración, inicio de sesión para Cliente MIDP1.0.**

En la Figura 4.8 se puede apreciar que el cliente por medio de la clase ConexiónHTTP se conecta con el servidor, que a su vez por medio de la clase RelayServlet busca la referencia en el conjunto de usuarios disponibles. Una vez encontrada la referencia, la clase usuario inicia el procesamiento del mensaje de entrada, transformando los mensajes recibidos del móvil en mensajes Jabber para ser intercambiados con el módulo de servicios Jabber. El servidor retorna al móvil el primer mensaje que esté en la cola de espera. Cada que el cliente envíe un mensaje al servidor se repite este proceso.

El módulo encargado de atender las peticiones de los clientes MIDP 2.0 provee conexiones dedicadas a cada uno de los usuarios mediante Sockets TCP, y es capaz de manejar concurrentemente un gran número de usuarios definido por el administrador de la plataforma. En la gráfica 4.9 se puede observar el diagrama de inicio de sesión para el cliente MIDP2.0:

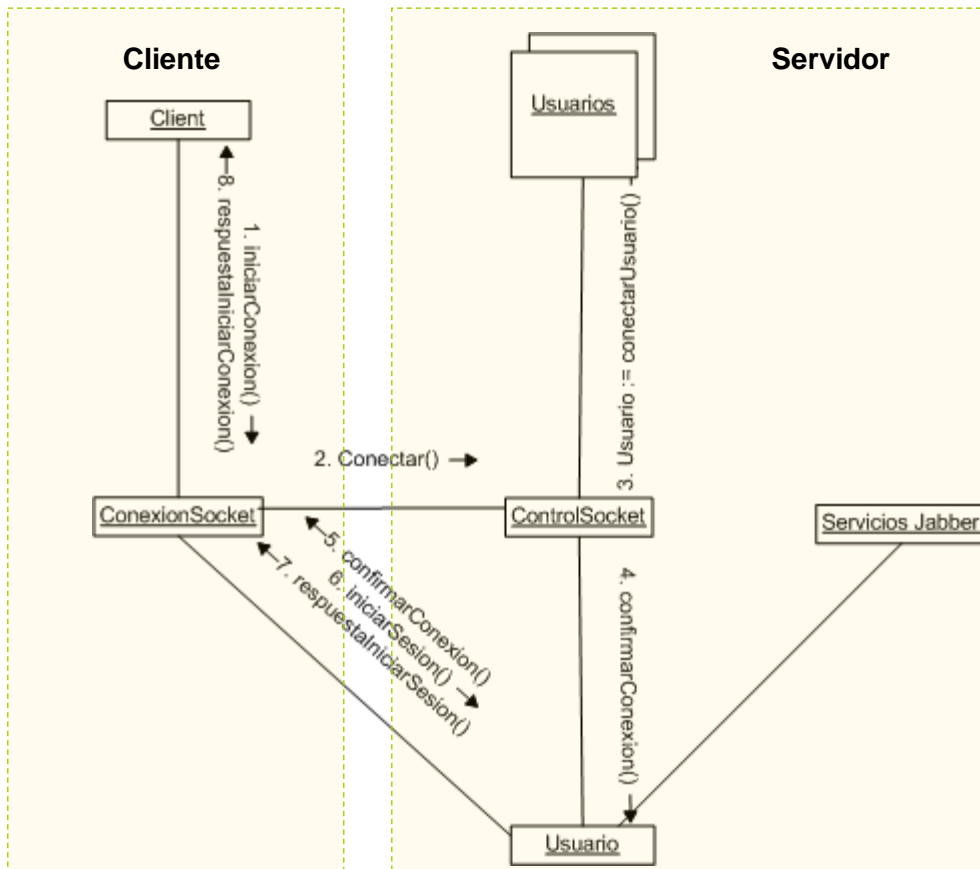


Figura 4.11 Diagrama de colaboración, conexión e inicio de sesión para cliente MIDP2.0

En la Figura 4.9 se puede apreciar la conexión y el inicio de sesión del cliente móvil. Una vez establecida la conexión, la comunicación entre el móvil y el servidor es directa por medio de las clases ConexionSocket en el móvil y Usuario en el servidor.

#### 4.4.1 Módulo de Interoperabilidad

Encargado de ofrecer los servicios de mensajería instantánea y proveer la comunicación con los proveedores externos de mensajería instantánea. Cuenta con dos submódulos descritos a continuación:

##### 4.4.1.1 Submódulo de Servicios Jabber

###### Consumo de Memoria y Uso de CPU

El servidor Ejabberd 1.0 se instaló en el sistema operativo Linux distribución Debian. El número de usuarios que el servidor soporta concurrentemente depende de la



configuración y del hardware en el cual se esté corriendo. En la parte de configuración, es necesario definir el número máximo de puertos habilitados por Erlang y el número máximo de conexiones. La eficiencia del servicio tiene dependencia directa con el ambiente de ejecución del servidor, si corre en una sola máquina o en un cluster de máquinas. Para ambos casos, la memoria RAM y la CPU son factores determinantes en el rendimiento. Cuando el servidor es configurado para correr en un cluster de máquinas, el rendimiento del sistema depende además de la configuración y disposición del cluster de máquinas.

En este proyecto se configuró el servidor Ejabberd para correr en una sola máquina, configurando el número máximo de puertos en 1000. Con una prueba de estrés, en la cual se inició sesiones anónimas en el servidor hasta que éste cerró la recepción de nuevas conexiones se determinó, que bajo el entorno de ejecución utilizado, que el servidor puede aceptar entre 720 y 770 usuarios dependiendo de la actividad de cada una de las sesiones de usuario. En las Figuras 4.12 y 4.13 se puede observar el uso de la CPU y el consumo de memoria respectivamente, en los tres intervalos en las cuales se ejecutó la prueba. El primero, es iniciar el servidor Ejabberd. El segundo es ejecutar la prueba de estrés, en donde se corrió una aplicación que inicia un número continuo de sesiones anónimas con el servidor y el tercero cerrar las sesiones de usuario iniciadas.

Se puede observar que el uso de la CPU es máximo cuando se inicia el servidor, en el registro de las cuentas y al cerrar el servidor. El consumo de memoria aumenta en la medida en que se van iniciando sesiones anónimas, hasta el punto en el cual el servidor ya no tiene la memoria suficiente, suspende la recepción de conexiones por parte de los clientes y deja de consumir memoria RAM.

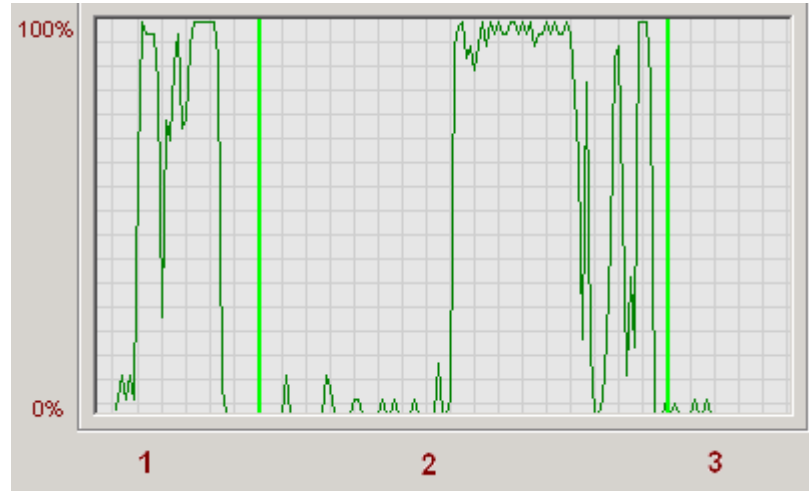


Figura 4.12 Uso de la CPU servidor Ejabberd

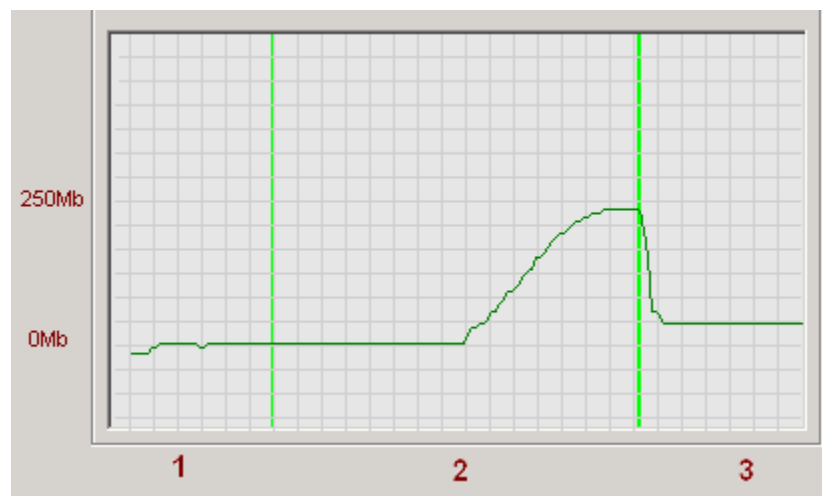


Figura 4.13 Consumo de memoria servidor Ejabberd

#### 4.4.1.2 Submódulo de Transportes

Cada uno de los transportes está en un estado de desarrollo diferente y no tienen todas las funcionalidades implementadas, a pesar de esto el desempeño de cada uno de los transportes fue muy bueno ya que permitieron llevar a cabo exitosamente la interoperabilidad con los proveedores de mensajería instantánea externos. En la Tabla 4.1 se puede observar las funcionalidades disponibles en el momento para cada uno de los transportes:

Funcionalidad	MSN	ICQ	AIM
---------------	-----	-----	-----



Mensajería	✓	✓	✓
Presencia	✓	✓	✓
Grupos de Chat	✗	✗	✓
Soporte para VCard	✓	✓	✓
Presencia invisible	✓	✗	✓
Notificaciones de escritura	✓	✓	✓
Mensajes HTML	✗	✗	✓

Tabla 4.1. Funcionalidades disponibles de los transportes Jabber

## 4.5 Análisis de costos de la plataforma

### 4.5.1 Costos de mantenimiento y administración

Para poder montar PAUMIM se debe contar con una serie de herramientas hardware y soporte para su mantenimiento y administración. A continuación se muestra en la Tabla 4.2 las herramientas básicas hardware y software necesarias para montar el servicio de PAUMIM, teniendo en cuenta que se va a recibir un número de usuarios que oscila entre 1 y 770 y que el rendimiento y eficiencia de la plataforma no va a verse comprometida bajo estas condiciones:

Nombre	Tipo de Herramienta	Valor (US\$/mes)
Máquina servidora con conexión a Internet	Hardware	150
Sistema Operativo Linux	Software	0

Tabla 4.2. Herramientas Hardware y Software para montar PAUMIM

Los precios del hosting varían dependiendo de las características de la máquina que se solicite. Entre mejor sea el rendimiento mayor es el precio a pagar. Uno de los beneficios que se tienen al contratar hosting en Internet es que vienen incluidos los gastos incluidos





de mantenimiento hardware incluyendo servicio de energía, refrigeración y protección contra descargas eléctricas.

En cuanto a los costos de administración se debe tener en cuenta los gastos mencionados en la Tabla 4.3:

Tipo de Gasto	Valor (\$/mes)
Administrador de la plataforma	2'000.000

**Tabla 4.3 Gastos administrativos**

El administrador de la plataforma es el encargado de realizar el mantenimiento Software y la futura expansión a nuevas tecnologías. De la misma forma deberá estar en constante investigación tecnológica para tener actualizadas las versiones software de los distintos componentes que conforman la plataforma, realizar adaptaciones a nuevas tecnologías y adoptar estrategias que permitan la correcta inclusión de dichas tecnologías en la plataforma.

#### **4.6 Mensajería Instantánea Móvil Tradicional**

En la actualidad, el servicio de mensajería instantánea móvil tradicional es prestado por los operadores de telefonía celular en todo el mundo a través de SMS (Short Message Service). SMS es un servicio inalámbrico que permite la transmisión de mensajes alfanuméricos entre suscriptores móviles y entre suscriptores móviles y sistemas externos tales como correo electrónico y los sistemas de correo de voz. El servicio de SMS se caracteriza por el envío de paquetes fuera de banda y la transferencia de mensajes de poco ancho de banda, que da lugar a uso de medios altamente eficientes para transmitir ráfagas cortas de datos. Dependiendo del estándar de mensajería utilizado por el operador, el formato varía y tiene una limitante en tamaño, alcanzando el mensaje típico una longitud de 160 caracteres. [40]

El auge de las comunicaciones móviles al interior del país impulsó a los operadores de telefonía móvil celular a ofrecer un mejor y más completo servicio, incluyendo por supuesto la mensajería instantánea basada en SMS. Cada operador ofrece a sus



usuarios distintos niveles de servicio, en los cuales se incluyen bolsas de mensajes, descuentos y la posibilidad de envío de mensajes a otros países, entre otras. Los precios dependen del plan al cual esté afiliado el usuario y si el mensaje se envía a otro operador. En la Tabla 4.4 se muestra el precio de SMS en cada operador nacional de telefonía móvil:

Operador	Valor SMS (Mismo operador)	Valor SMS (otro operador)
	en pesos	en pesos
Comcel S.A.	146	325
OLA	200	200
Movistar	200	200

**Tabla 4.4. Precio de SMS para operadores nacionales de telefonía móvil**

#### **4.7 PAUMIM Vs SMS**

Para realizar la comparación entre SMS y PAUMIM se van a abordar cuatro características muy importantes: precio, acceso al servicio e interoperabilidad, dándole el enfoque más importante al envío de mensajes, teniendo como objetivo al usuario consumidor. A continuación se analizan los tres aspectos:

##### **4.7.1 Precio**

Para realizar la comparación, se debe tener en cuenta los precios de envío de mensajes SMS y para el uso de PAUMIM. Para tener referencia de los precios para SMS se toma al operador de telefonía móvil COMCEL S.A. de la Tabla 4.4 y para el uso de PAUMIM un plan de datos de COMCEL con las características nombradas en la tabla 4.5:

Nombre	Valor Kb dentro del plan (pesos)	Valor Kb adicional (pesos)
Datos 10	3.73	9.3

**Tabla 4.5. Características del plan de datos Datos 10**

Con esta información se obtiene la Tabla comparativa 4.6:



Item	Precio SMS (pesos)	Precio envío mensaje PAUMIM (pesos)
Envío de mensaje(mismo operador)	146	2.23
Envío de mensaje(otro operador)	325	2.23

**Tabla 4.6. Comparación de precios entre SMS y PAUMIM**

Con lo anterior se concluye que el precio por envío de mensajes en la plataforma PAUMIM es mucho más bajo con relación a SMS.

#### **4.7.2 Acceso al servicio**

El acceso al servicio por parte de los usuarios es inmediato por medio de teléfono celular o PDA (Personal Digital Assistant) con las características adecuadas. A continuación se nombran las características más generales de los dispositivos para soportar cada tecnología:

##### **4.7.2.1 SMS**

Los teléfonos celulares que dispongan del servicio de telefonía celular tienen la posibilidad de enviar mensajes SMS de forma fácil y rápida, mediante el acceso a la aplicación por medio del menú del dispositivo. La mayoría de operadores en el mundo complementa el servicio de telefonía móvil con el de mensajería instantánea basada en SMS.

##### **4.7.2.2 PAUMIM**

Las características de los dispositivos para soportar el cliente de PAUMIM son los siguientes:

- Soporte para aplicaciones Java MIDP 1.0 o 2.0
- Acceso a la red de datos GPRS.

Uno de los principales aspectos a tener en cuenta en el acceso a la plataforma es que los clientes están soportados para teléfonos celulares de gama media, gama alta y PDAs que tengan acceso a la red de datos GPRS, lo que amplía el rango de dispositivos que



soportan la aplicación cliente. Es importante resaltar que para poder utilizar la aplicación cliente de PAUMIM, el teléfono celular debe tener acceso a la red de datos GPRS, disminuyendo notablemente el número de posibles usuarios de la plataforma ya que en el país los planes de datos son asignados únicamente a los usuarios que cuenten con un plan abierto de telefonía celular.

### **4.7.3 Interoperabilidad**

Uno de los objetivos principales de PAUMIM es brindar interoperabilidad entre múltiples proveedores de mensajería instantánea, mediante la adición de módulos funcionales a la plataforma. Por otro lado, para lograr interoperabilidad entre proveedores de servicio de SMS es necesario llegar a acuerdos comerciales de interconexión de redes. Dichos acuerdos son dependientes de las políticas de cada operador y de la regulación de cada país, siendo determinantes en el momento del establecimiento de dichos acuerdos.



## 5 Conclusiones, Recomendaciones y Trabajos Futuros

### 5.1.1 Conclusiones del Proyecto

- Se logró implementar una Plataforma de Acceso Universal a Mensajería Instantánea Móvil permitiendo la movilidad de los usuarios y el acceso a comunicaciones móviles a través de teléfonos celulares y computadores de mano tipo PALM.
- Se logró poner en funcionamiento PAUMIM en dos tipos de ambientes de ejecución. El primero en la red de telefonía móvil celular y el segundo en una red WLAN.
- Se logró establecer interoperabilidad entre PAUMIM y los proveedores de mensajería instantánea AOL, ICQ y MSN Messenger con muy buenos resultados.
- Se logró llevar a cabo la comunicación entre los usuarios móviles y el servidor PAUMIM con muy buenos resultados.
- Se implementó gran parte de la funcionalidad requerida en un cliente de mensajería instantánea móvil, logrando llevarle al usuario una aplicación altamente usable; sin embargo, faltaron funcionalidades adicionales como la transferencia de archivos y salas de charla entre diferentes proveedores, características muy importantes que no se implementaron por la limitación en los transportes.
- Con la construcción de dos tipos de clientes móviles se logró abarcar un número más amplio de posibles usuarios. El primer cliente va enfocado a gama media de teléfonos celulares con soporte Java MIDP 1.0 y el otro a la gama media, alta y computadores de mano con soporte Java MIDP 2.0.
- Se estableció una arquitectura replicable que permite tener un sistema altamente escalable, portable y administrable.
- Se estableció un estudio comparativo entre la mensajería corta móvil y la mensajería a través de PAUMIM, obteniendo resultados muy positivos para PAUMIM por su bajo precio, portabilidad, alta usabilidad, interoperabilidad y fácil acceso.



### 5.1.2 Conclusiones Tecnológicas

- A pesar de que J2ME es una especificación, las implementaciones de las máquinas virtuales para cada gama de dispositivos tiene algunas variantes y por lo tanto se deben tener en cuenta estas. Como ejemplo, los Input y Output Streams no funcionan correctamente en algunas de las primeras implementaciones de J2ME.
- Por medio de la construcción de un protocolo liviano de bajo consumo de ancho de banda se llevó a cabo la comunicación entre el cliente móvil y el servidor de forma eficiente.
- La utilización de protocolos universales como TCP/IP en la base de las comunicaciones entre los dispositivos móviles y el servidor PAUMIM hace que se puedan implementar y soportar diferentes tipos de clientes como son: Web y de escritorio, en una gran variedad de tecnologías de desarrollo como Java, PHP, Perl, Python, .Net, C#, VisualBasic, entre otras.
- El perfil MIDP1.0 es muy restringido en el área a comunicaciones, debido a que cuenta únicamente con conexiones HTTP, las cuales permiten establecer comunicaciones con el servidor pero de manera ineficiente, debido al uso de un sondeo con el servidor, lo que conlleva a un incremento del consumo ancho de banda, disminución la eficacia de la aplicación y aumento de los costos de utilización.
- Se identificó que el perfil MID 2.0 tiene serie de herramientas muy útiles que permiten construir aplicaciones de gran calidad. Cuenta con un framework de conexiones mucho más completo que su antecesor, lo cual permite implementar comunicaciones a bajo nivel por medio de Sockets TCP mediante el establecimiento de un circuito lógico entre el cliente móvil y el servidor, reduciendo costos e incrementando en gran medida la eficiencia de la aplicación. Además, ofrece herramientas para implementar la internacionalización de la aplicación.
- Se llevó a cabo la evaluación de servidores Jabber y se optó por el servidor Ejabberd por el completo soporte del protocolo Jabber, la capacidad de configuración y adaptabilidad de los transportes de comunicación.
- Se investigó y se evaluó las diferentes clases de transportes adaptables al servidor y se optó por PyMSN, PyICQ y PyAOL debido a sus características de libre distribución, alto rendimiento y fácil configuración.



### 5.1.3 Recomendaciones

- Utilizar una amplia gama de emuladores que permitan validar en cierta medida, las aplicaciones para cada uno de los proveedores de dispositivos móviles y para cada una de sus gamas.
- Al instalar y configurar el servidor Jabber y los transportes se debe tener mucho cuidado con las versiones, ya que existen versiones incompatibles y pueden llevar al mal funcionamiento o en el peor de los casos al no funcionamiento del servicio.

### 5.1.4 Trabajos Futuros

- Implementar los clientes para el sistema operativo Symbian y Windows Mobile ya que estos sistemas operativos tienen gran acogida en Europa y Estados Unidos.
- Llegar a cabo la interoperabilidad con el sistema de mensajería corta SMS.
- Incorporar un sistema de seguridad altamente eficiente que garantice a los usuarios de PAUMIM el transporte de su información de manera confiable, debido a que dentro de la plataforma el manejo de seguridad se hizo solamente entre el servidor de mensajería y los transportes, quedando por implementar conexiones seguras entre los clientes móviles y el servidor.



## 6 Referencias

[i] **Mobber, Mobile Jabber Communicator.** Grzegorz Graszka

Disponible en: <http://mobber.gryf.info/>

[1] **Mensajería Instantánea.** Ernesto López, PC Magazine.

Disponible en: <http://www.x-extrainternet.com/messengers.asp>

[2] **How Americans Use Instant Messaging.** Eulynn Shiu, Amanda Lenhart. PEW INTERNET & AMERICAN LIFE PROJECT. 1 de Septiembre del 2004.

Disponible en: [http://www.pewinternet.org/pdfs/PIP\\_Instanmessage\\_Report.pdf](http://www.pewinternet.org/pdfs/PIP_Instanmessage_Report.pdf)

[3] **AOL-Leadership. Innovation. Commitment to Members.** American Online. 2005.

Disponible en: <http://www.corp.aol.com/whoweare/history.shtml>

[4] **Proyecto CJR un cliente Jabber.** Raúl Rodríguez Aparicio, Universidad Rey Juan Carlos. 24 de Septiembre del 2002.

Disponible en: <http://gsyc.escet.urjc.es/~rrodrigu/pfc/isi/doc/memoria/node8.html>

[5] **Proyecto CJR un cliente Jabber.** Raúl Rodríguez Aparicio, Universidad Rey Juan Carlos. 24 de Septiembre del 2002.

Disponible en: <http://gsyc.escet.urjc.es/~rrodrigu/pfc/isi/doc/memoria/node9.html>

[6] **Mensajería Instantánea.** WIKIPEDIA. La Enciclopedia Libre.

Disponible en: [http://es.wikipedia.org/wiki/Mensajer%C3%ADa\\_instant%C3%A1nea](http://es.wikipedia.org/wiki/Mensajer%C3%ADa_instant%C3%A1nea)





[7] **Securing Instant Messaging.** Symantec Enterprise Security.

Disponible en:

<http://securityresponse.symantec.com/avcenter/reference/secure.instant.messaging.pdf>

[8] **Text IM Makes Way for Video.** Instant Messaging Planet.com. Christopher Sounder.

17 de Julio de 2003

Disponible en:

<http://www.instantmessagingplanet.com/enterprise/article.php/2242631>

[9] **AOL Streams IM Video.** Instant Messaging Planet.com. Tim Gray. 16 de Noviembre de 2003.

Disponible en: <http://www.instantmessagingplanet.com/public/article.php/3436561>

[10] **Yahoo Updates Messenger's Phone Features.** PC World. Juan Carlos Pérez. 8 de Diciembre de 2005

Disponible en: <http://www.instantmessagingplanet.com/public/article.php/3436561>

[11] **Jabber/XMPP.** Creative Commons. Óscar Lage Serrano. 15 de Junio de 2005

Disponible en: <http://www.jabberes.org/files/download/Jabber-XMPP.pdf>

[12] **Google Talk.** WIKIPEDIA. La Enciclopedia Libre.

Disponible en: [http://es.wikipedia.org/wiki/Google\\_Talk](http://es.wikipedia.org/wiki/Google_Talk)

[13] **MSN and Vodafone to Launch PC-to-Mobile Instant Messaging.** Microsoft PressPass – Information for Journalists. Redmond Wash. 29 de Junio de 2005

Disponible en:

<http://www.microsoft.com/presspass/press/2005/jun05/06-29VodafoneMSNPR.msp>

[14] **Yahoo! Messenger.** Cingular.

Disponible en: [http://www.cingular.com/media/instant\\_yahoo](http://www.cingular.com/media/instant_yahoo)



[15] **Instant Messaging service: analysis and challenges.** Autorité de Régulation des Télécommunications. Estudio conducido por la firma IDATE. Mayo de 2004

Disponible en: <http://www.art-telecom.fr/>

[16] **In the Matter of Applications of America Online.** Federal Communication Commission. 29 de Septiembre del 2000

Disponible en: <http://www.fcc.gov/mb/aoltw/imletter.txt>

[17] **Decreto Nº 600 del 14 de Marzo del 2003.** Martha Elena Pinto de Hart. Ministerio de Comunicaciones. 14 de Marzo del 2003

Disponible en:

<http://www.presidencia.gov.co/decretoslinea/2003/marzo/14/dec600030314.pdf>

[18] **Decreto 3055 de 2003.** Martha Elena Pinto de Hart. Ministerio de Comunicaciones. 27 de Octubre del 2003

Disponible en:

<http://www.presidencia.gov.co/decretoslinea/2003/octubre/27/dec3055271003.pdf>

[19] **La Convergencia de Servicios.** Sudhir Ahuja. Lucent Technologies. Octubre del 2005. RCT, Revista Colombiana de Telecomunicaciones

[20] **Interoperabilidad de IM y las amenazas.** Denken Uber. 12 de Octubre de 2005

Disponible en: <http://www.uberbin.net/archivos/colaboracion/interoperabilidad-de-im-y-las-amenazas.php>

[21] **Mensajería Instantánea: el nuevo modelo de comunicación personal.** Daniel Fajardo C. Ministerio de Comunicaciones. 27 de Octubre del 2003

Disponible en:

<http://www.presidencia.gov.co/decretoslinea/2003/octubre/27/dec3055271003.pdf>

[22] **Introducción a Jabber.** Kuasar y Raulex. Septiembre 25 de 2003.

Disponible en: <http://www.jabberes.org/introduccion>



[23] ***Servidores independientes e interconectados.*** Jamarier. Diciembre 8 de 2003

Disponible en: <http://www.jabberes.org/node/154>

[24] ***Model for Presence and Instant Messaging.*** Network Working Group, M. Day, Lotus. J. Rosenberg, dynamicsoft. H. Sugano, Fujitsu. Febrero de 2000

Disponible en: [www.ietf.org/rfc2778.txt](http://www.ietf.org/rfc2778.txt)

[25] ***Extensible Messaging and Presence Protocol (XMPP): Instant Messaging and Presence.*** Network Working Group. P. Saint-Andre. Octubre de 2004.

Disponible en: <http://www.xmpp.org/specs/rfc3921.html>

[26] ***Ergonomics of human-system interaction - Usability methods supporting human-centered design.*** International Organization for Standardization: ISO, ISO/TR 16982:1998: 1998.

Disponible en: <http://www.fees-network.org/resources/standards/isotc159.html>

[27] ***Plataforma Integrada de Desarrollo de Aplicaciones para Dispositivos Móviles con J2ME.*** Euler Trejo, Gabriel Vasquez. Universidad del Cauca. 2004

[28] ***The J2EE 1.4 Tutorial.*** Eric Armstrong, Jennifer Ball, Stephanie Bodoff, Debbie Bode Carson, Ian Evans, Dale Green, Kim Haase, Eric Jendrock. Addison-Wesley, 2001.

[29] **Python.** Wikipedia. La Enciclopedia Libre.

Disponible en: <http://es.wikipedia.org/wiki/Python>

[30] **PyOpenSSL.** Artículo.

Disponible en: [http://penguin.triumf.ca/comps/9/es\\_frames/rpms/pyOpenSSL.php](http://penguin.triumf.ca/comps/9/es_frames/rpms/pyOpenSSL.php)

[31] **PyCripto.** Noticia. 25 de agosto de 2004.

Disponible en: <http://www.pythonhispano.org/node/140>



[32] **An Introduction to the Twisted Networking Framework.** Itamar Shtull-Trauring. Enero 15 de 2004

Disponible en: [http://www.onlamp.com/pub/a/python/2004/01/15/twisted\\_intro.html](http://www.onlamp.com/pub/a/python/2004/01/15/twisted_intro.html)

[33] **Ejabberd.** Página Web.

Disponible en: <http://ejabberd.jabber.ru/>

[34] **Erlang.** Página Web.

Disponible en: [www.erlang.org](http://www.erlang.org)

[35] **PyMSNt.** Página Web.

Disponible en: <http://msn-transport.jabberstudio.org/>

[36] **PyAIM – t.** Página Web.

Disponible en: <http://pyaim-t.blathersource.org/>

[37] **PyICQ – t.** Página Web.

Disponible en: <http://pyicq-t.blathersource.org/>

[38] **Smack API: Simple and Powerful Java client API for XMPP.** Página Web.

Disponible en: <http://www.jivesoftware.org/smack/>

[39] **PostgreSQL JDBC Driver.** Página Web.

Disponible en: <http://jdbc.postgresql.org/>

[40] **Short Message Service.**

Disponible en:

[http://searchmobilecomputing.techtarget.com/sDefinition/0,,sid40\\_gci213660,00.html](http://searchmobilecomputing.techtarget.com/sDefinition/0,,sid40_gci213660,00.html)



## 7 Glosario