

**CONVERGENCIA DE SERVICIOS DE MENSAJERÍA MÓVIL E INSTANTÁNEA  
MEDIANTE UN ESB**



**Felipe Edmundo Burbano Burbano  
Juan Sebastián Embuz Hincapié**

**Universidad del Cauca  
Facultad de Ingeniería Electrónica y Telecomunicaciones  
Departamento de Telemática  
Popayán - Colombia**

**CONVERGENCIA DE SERVICIOS DE MENSAJERÍA MÓVIL E INSTANTÁNEA  
MEDIANTE UN ESB**



**Felipe Edmundo Burbano Burbano  
Juan Sebastián Embuz Hincapié**

**Monografía**

**Director: Javier Alexander Hurtado Guaca  
Ingeniero en Electrónica y Telecomunicaciones**

**Asesor: Oscar Mauricio Caicedo Rendón  
Magíster en Ingeniería Telemática**

**Universidad del Cauca  
Facultad de Ingeniería Electrónica y Telecomunicaciones  
Departamento de Telemática  
Línea de Investigación en Servicios Avanzados de Telecomunicaciones  
Popayán, Marzo de 2012**

## TABLA DE CONTENIDOS

CAPÍTULO 1.....	1
Introducción .....	1
1.1. Generalidades .....	1
1.1.1. Resumen del trabajo realizado.....	1
1.1.2. Contenido de la monografía .....	1
1.2. Planteamiento del Problema.....	2
1.3. Objetivos del Trabajo de Grado .....	3
CAPÍTULO 2.....	4
Marco Teórico.....	4
2.1. Conceptualización .....	4
2.1.1. Mensajería instantánea.....	4
2.1.2. Mensajería móvil.....	5
2.1.3. Integración de mensajería instantánea y mensajería móvil .....	7
2.1.3.1. Broker de mensajería .....	7
2.1.3.2. Enterprise Service Bus .....	8
2.1.3.2.1. Tipos de arquitecturas ESB.....	10
2.1.3.2.2. Comparación de software ESB .....	12
2.2. Proyectos Relacionados.....	15
2.2.1. A process modeling-based approach for web service management.....	15
2.2.2. A survey on web services in telecommunications.....	16
2.2.3. A UML model-driven business process development methodology.....	16
2.2.4. The enterprise service bus: Making service-oriented architecture real .....	16
2.2.5. Mensajería en sistemas de información .....	17
2.2.6. Integración de aplicaciones de seguridad .....	17
2.2.7. Contribuciones y diferencias con los proyectos relacionados.....	17
CAPÍTULO 3.....	19
Arquitectura para la Convergencia de Servicios de Mensajería mediante un Broker.....	19
3.1. Descripción de la Arquitectura .....	20
3.1.1. El patrón de arquitectura.....	21
3.1.2. Metodología para la descripción de la arquitectura .....	23
3.2. Vista Modular .....	24
3.2.1. Servidor de aplicaciones móviles - SAM .....	25
3.2.2. Servidor mail - SM .....	26
3.2.3. Servidor de bases de datos - SBD .....	26
3.2.4. Módulo enterprise service bus - ESB .....	27
3.2.5. Servidor de mensajería instantánea - SMI .....	27

3.2.6. Gateway de mensajería móvil - GMM .....	28
3.2.7. Módulo de clientes .....	28
3.3. Vista de Diseño .....	29
3.3.1. Servidor de aplicaciones móviles .....	29
3.3.2. Servidor mail.....	30
3.3.3. Servidor de bases de datos.....	31
3.3.4. Enterprise Service Bus.....	32
3.3.5. Servidor de Mensajería Instantánea.....	33
3.3.6. Gateway de Mensajería Móvil .....	34
3.4. Vista de Implantación .....	36
CAPÍTULO 4.....	38
Implementación del Broker de Mensajería y Evaluación .....	38
4.1. Implementación de Referencia .....	38
4.1.1. Diagrama de casos de uso.....	38
4.1.1.1. Casos de uso iniciados por el usuario móvil .....	39
4.1.1.2. Casos de uso iniciados por usuario Web .....	39
4.1.1.3. Casos de uso comunes a los 2 actores .....	41
4.1.2. Diagrama de paquetes general .....	41
4.1.3. Análisis por subsistemas.....	43
4.1.3.1. Gateway de mensajería móvil.....	43
4.1.3.1.1. Diagrama de paquetes de análisis del subsistema.....	43
4.1.3.1.2. Diagrama de paquetes de diseño del subsistema .....	43
4.1.3.1.3. Diagrama de clases del subsistema .....	44
4.1.3.2. Servidor de mensajería instantánea .....	47
4.1.3.2.1. Diagrama de paquetes de análisis del subsistema.....	47
4.1.3.2.2. Diagrama de paquetes de diseño del subsistema.....	48
4.1.3.2.3. Diagrama de clases del subsistema .....	48
4.1.3.3 Servidor mail .....	50
4.1.3.3.1. Diagrama de paquetes de análisis del subsistema.....	50
4.1.3.3.2. Diagrama de paquetes de diseño del subsistema .....	50
4.1.3.3.3. Diagrama de clases del subsistema .....	51
4.1.3.4. Servidor de aplicaciones móviles.....	51
4.1.3.4.1. Diagrama de paquetes de análisis del subsistema.....	51
4.1.3.4.2. Diagrama de paquetes de diseño del subsistema.....	52
4.1.3.4.3. Diagrama de clases del subsistema .....	52
4.1.3.5. Servidor de base de datos.....	53
4.1.3.5.1. Diagrama de paquetes de análisis del subsistema.....	53
4.1.3.5.2. Diagrama de paquetes de diseño del subsistema .....	54
4.1.3.5.3. Diagrama de clases del subsistema .....	55
4.2. Arquitectura de la Implementación .....	56
4.2.1. Interacción entre componentes .....	56
4.2.1.1. Usuario a través de SMS .....	57
4.2.1.2. Usuarios a través de un cliente MI.....	60

CAPÍTULO 5.....	62
Servicio de Prueba.....	62
5.1. Descripción del Escenario de Despliegue.....	63
5.2. Descripción del piloto .....	64
5.2.1. Base de datos del sistema .....	64
5.2.2. Método de prueba.....	65
5.2.3. Establecimiento del canal de escucha ESB .....	66
5.2.4. Consumo de memoria.....	69
5.2.5. Tiempos de respuesta.....	71
5.2.6. Comunicación y tamaño de datos de la aplicación para móvil.....	72
5.2.7. Concurrencia .....	75
CAPITULO 6.....	78
Aportes, Conclusiones y Trabajo Futuro .....	78
6.1. Aportes.....	78
6.2. Conclusiones.....	78
6.3. Trabajo Futuro.....	80
Referencias Bibliográficas.....	81

## Lista de Figuras

Figura 1. Arquitectura ESB .....	8
Figura 2. Message bus architecture .....	11
Figura 3. Protocol switch.....	11
Figura 4. Gateway.....	12
Figura 5. Comparación de productos ESB 1 .....	13
Figura 6. Comparación de productos ESB 2 .....	13
Figura 7. Comparación de productos ESB 3 .....	14
Figura 8. Módulos del ESB .....	23
Figura 9. Arquitectura modular de referencia .....	25
Figura 10. SAM.....	26
Figura 11. SM.....	26
Figura 12. SBD.....	26
Figura 13. ESB_M.....	27
Figura 14. SMI.....	28
Figura 15. GMM.....	28
Figura 16. Módulo de clientes .....	29
Figura 17. Diagrama de diseño SAM .....	30
Figura 18. Diagrama de diseño SM.....	31
Figura 19. Diagrama de diseño SBD.....	32
Figura 20. Diagrama de diseño ESB_M.....	32
Figura 21. Diagrama de diseño SMI.....	34
Figura 22. Diagrama de diseño GMM .....	36
Figura 23. Diagrama general de implantación.....	37
Figura 24. Casos de uso.....	38
Figura 25. Diagrama de paquetes del sistema .....	42
Figura 26. Diagrama de paquetes de análisis GMM.....	43
Figura 27. Diagrama de paquetes de diseño GMM.....	44
Figura 28. Diagrama de clases módem GSM.....	45
Figura 29. Diagrama de clases SMSC.....	46
Figura 30. Diagrama de paquetes de análisis SMI.....	47
Figura 31. Diagrama de paquetes de diseño SMI .....	48
Figura 32. Diagrama de clases SMI .....	49
Figura 33. Diagrama de paquetes de análisis SM.....	50
Figura 34. Diagrama de paquetes de diseño SM .....	50
Figura 35. Diagrama de clases SM.....	51
Figura 36. Diagrama de paquetes de análisis SAM.....	52
Figura 37. Diagrama de paquetes de diseño SAM.....	52
Figura 38. Diagrama de clases SAM.....	53
Figura 39. Diagrama de paquetes de análisis SBD.....	54
Figura 40. Diagrama de paquetes de diseño SBD .....	54
Figura 41. Diagrama de clases SBD.....	55
Figura 42. Arquitectura de implementación.....	56
Figura 43. Diagrama de secuencia Login y Logout .....	57
Figura 44. Diagrama de secuencia envío de mensaje MI.....	59
Figura 45. Diagrama de secuencia clientes MI.....	60
Figura 46. Ambiente de evaluación.....	63
Figura 47. Diagrama de la base de datos del sistema.....	65
Figura 48. Panel principal multimensajes SimSMSC.....	66

<i>Figura 49. Establecimiento de la comunicación con ESB, WhiresShark.....</i>	<i>66</i>
<i>Figura 50. Establecimiento de la comunicación con ESB, consola de Windows .....</i>	<i>67</i>
<i>Figura 51. Petición de login del SMI.....</i>	<i>67</i>
<i>Figura 52. Respuesta del ESB.....</i>	<i>68</i>
<i>Figura 53. Consola petición login usuario no registrado.....</i>	<i>69</i>
<i>Figura 54. Consumo de memoria ESB, SBD, SM y SAM.....</i>	<i>69</i>
<i>Figura 55. Consumo de memoria SMI y GMM.....</i>	<i>70</i>
<i>Figura 56. Consumo de memoria para tráfico ESB .....</i>	<i>70</i>
<i>Figura 57. Proceso para entrada de SMS.....</i>	<i>71</i>
<i>Figura 58. Comparación de tiempos de respuesta por mensaje.....</i>	<i>72</i>
<i>Figura 59. Comunicación SMI - Cliente móvil.....</i>	<i>73</i>
<i>Figura 60. Estadística de servidor de mensajería instantánea .....</i>	<i>76</i>
<i>Figura 61. Comportamiento RAM con 14000 usuarios.....</i>	<i>77</i>

## Lista de Tablas

<i>Tabla 1. Comparativa de compatibilidad de software para implementaciones ESB.....</i>	<i>15</i>
<i>Tabla 2. Proyectos relacionados.....</i>	<i>18</i>
<i>Tabla 3. Características de las arquitecturas ESB.....</i>	<i>20</i>
<i>Tabla 4. Casos de uso iniciados por el usuario móvil.....</i>	<i>39</i>
<i>Tabla 5. Casos de uso iniciados por el usuario Web.....</i>	<i>41</i>
<i>Tabla 6. Casos de uso iniciados por los dos actores.....</i>	<i>41</i>
<i>Tabla 7. Características equipo de prueba.....</i>	<i>63</i>
<i>Tabla 8. Puertas de entrada.....</i>	<i>64</i>
<i>Tabla 9. Características de los equipos servidores y puertos.....</i>	<i>64</i>
<i>Tabla 10. Tiempo de respuesta login.....</i>	<i>71</i>
<i>Tabla 11. Tiempo de respuesta logout.....</i>	<i>72</i>
<i>Tabla 12. Tiempo de respuesta mensaje.....</i>	<i>72</i>
<i>Tabla 13. Consumo de bytes solicitud login Roster.....</i>	<i>74</i>
<i>Tabla 14. Consumo de bytes solicitud login sin Roster.....</i>	<i>74</i>
<i>Tabla 15. Consumo de bytes de las solicitudes del protocolo jabber.....</i>	<i>75</i>
<i>Tabla 16. Referencia de precios operadores nacionales.....</i>	<i>75</i>

# **CAPÍTULO 1.**

## **Introducción**

### **1.1. Generalidades**

#### **1.1.1. Resumen del trabajo realizado**

En la actualidad, las organizaciones buscan la integración entre sistemas heterogéneos de forma flexible, escalable y reutilizable, además de la necesidad de estandarizar la comunicación entre diferentes plataformas, llevando a la creación de servicios soportados sobre Internet, que mediante las aplicaciones puedan ser descritos, publicados e invocados a través de la red, con una interfaz pública bien definida [1]. Debido a la necesidad de extender las capacidades de descripción de los servicios soportados sobre Internet, ha surgido recientemente la tecnología de servicios web que, permite evolucionar hacia la automatización de varias tareas, como el descubrimiento, selección y composición de servicios.

En ese sentido, surge el concepto de Arquitectura Orientada a Servicios (Service Oriented Architecture, SOA) que se basa en modelar el “negocio” a través de la definición de servicios, implementados comúnmente como servicios sobre Internet. Otro concepto importante que surge y tiene una relación cercana a SOA es el Bus de Servicios Empresariales (Enterprise Service Bus, ESB). El ESB ofrece la infraestructura necesaria para que aplicaciones y sistemas heterogéneos se comuniquen de forma segura, confiable y desacoplada. Dada esta infraestructura, los clientes invocan el servicio deseado y el ESB se encarga de encaminar y transformar la invocación hacia la implementación del servicio.

Por lo tanto en el presente proyecto, se propone extender una implementación desde ESB como punto de vista investigativo, a los efectos de ofrecer capacidad de registro, exploración, invocación e integración de servicios de mensajería móvil y mensajería instantánea, con el fin de proveer una mayor experiencia al usuario final. Se propone una arquitectura para éste fin y se establece su implementación para que, proveedores y consumidores de servicios, puedan hacer uso de la plataforma.

La solución propuesta es adaptable a distintos dominios de aplicación y presenta un diseño que permite ser extensible a nuevas funciones. La validación fue realizada mediante el desarrollo de un posible caso de estudio para la Universidad del Cauca mediante la plataforma JBossESB que es la implementación de ESB provista por JBoss. En la validación se realizó una evaluación a manera de ejemplo para el caso de estudio, ya que no fue posible hacerlo de forma real.

#### **1.1.2. Contenido de la monografía**

Este documento contiene toda la información relevante al desarrollo del proyecto de convergencia de mensajería móvil e instantánea mediante un ESB, y se ha dividido de tal forma que satisfaga la realización de un objetivo específico por capítulo, es decir, que los

capítulos 3, 4 y 5 proporcionan los resultados del trabajo realizado para cumplir los tres objetivos específicos de este proyecto.

El capítulo 2 aporta toda la información teórica necesaria para la comprensión del desarrollo del broker basado en ESB en los capítulos posteriores, además de la escogencia del software de integración para su implementación. El capítulo 3 presenta una serie de vistas para la descripción de la arquitectura, incluida la referencia del patrón de arquitectura. El capítulo 4 presenta el análisis UML del prototipo a través de casos de uso, diagramas de paquetes y diagramas de clases, para la posterior descripción de la interacción de sus componentes. El capítulo 5 presenta la descripción del escenario de pruebas para el prototipo y el detalle de las mismas, para afirmar con cabalidad la efectividad del sistema.

En los anexos se complementa la información del capítulo dos, en cuanto a otros paradigmas de integración y otras herramientas para implementar un ESB; para el capítulo tres se proporciona otra vista descriptiva de las abstracciones del sistema; para el capítulo cuatro se complementan los diagramas UML; se finaliza con una guía para la instalación y la configuración de componentes del sistema para el capítulo cinco, además de un manual de usuario.

## **1.2. Planteamiento del Problema**

La mensajería en general, es una de las herramientas de comunicación de gran popularidad en la Internet y las redes de telecomunicaciones [1]. A finales de 1996, la IM (Instant Messaging) se abrió camino al compás de Internet, proporcionando inmediatez y eficiencia en el diálogo, haciéndolo más amistoso, veloz, preciso y sobre todo informal. Las conversaciones son más interactivas porque se produce en un ambiente que reduce los malos entendidos (común en el e-mail) y promueve el humor y las emociones. Eso ayuda a crear un contexto ameno, en donde se puede bromear o transmitir estados de ánimo que solamente tenían sentido hablando cara a cara. Está claro, que este tipo de comunicación es ideal por ejemplo para usos financieros, equipos de proyectos distribuidos (freelance, teletrabajo), reuniones de negocio a distancia, servicio y atención al cliente, educación a distancia, entre otros [2].

De otro lado, existe un servicio de mensajería móvil, que logró adaptarse a las diferentes necesidades surgidas con el desarrollo de la Sociedad de la Información, este es el SMS (Short Message Service). SMS hace frente a los chats y al messenger, de hecho, se creyó que conforme evolucionaran las posibilidades de comunicación en Internet, incluso ahora con las redes sociales, su uso disminuiría de forma progresiva hasta desaparecer [3], sobre todo, cuando apareció en escena en el 2001 el MMS (Multimedia Messaging System), un hermano más avanzado que permite incorporar sonidos, vídeos o fotos. Sin embargo, mientras algunos servicios como EMS (Enhanced Messaging Service) y el buscapersonas han sido transformados, sustituidos o simplemente desaparecieron poco tiempo después de su nacimiento, los mensajes cortos pueden presumir que llevan más de 16 años en el contexto de las aplicaciones móviles. Además el SMS ha contribuido a la transmutación hacia el mundo digital, estableciendo una nueva forma de comunicación, modificando el lenguaje, las costumbres y la manera de relacionarnos con los demás [3].

La convergencia de la mensajería móvil y la mensajería instantánea, ha posibilitado una situación en la que existe un gran interés en promover servicios de valor agregado, para generar nuevas experiencias a los usuarios finales y modelos de negocio informáticos.

Para facilitar la integración de estos servicios, se tiene en cuenta una de las arquitecturas más utilizadas actualmente, SOA (Service Oriented Architecture), la cual proporciona facilidades en estos procesos (ej. facilidad para la integración de tecnologías disímiles, facilidad para abordar modelos de negocios basados en colaboración con otros entes, etc.).

En el mismo contexto, un ESB es una solución de integración distribuida, basada en mensajes y en estándares abiertos [5], y su función es proporcionar una comunicación fiable entre distintos recursos tecnológicos tales como aplicaciones, plataformas y servicios, que están distribuidos en múltiples sistemas por toda la empresa, que constituye los cimientos de una arquitectura orientada a servicios, los cuales pueden complementarse con capacidades de productividad adicionales [4].

Por lo tanto en los últimos años, se ha hecho más evidente en las comunidades empresariales u organizaciones, la necesidad de integrar servicios como el correo electrónico, la mensajería instantánea y la mensajería móvil, con el fin de proveer un mayor acercamiento entre los usuarios del sistema, lo cual se traduce en una mayor efectividad en el uso del tiempo para sus labores. El principal desafío en este tipo de integración es, cómo resolver la escalabilidad de las conexiones punto a punto, en donde el número de conexiones crece exponencialmente por cada aplicación que se añade. Por lo tanto, es necesario proporcionar mecanismos que permitan la integración de estos servicios, en donde se conecten solo una vez a una infraestructura troncal común, soportando mensajería distribuida, transparencia en la ubicación de los servicios, transparencia en el transporte y enrutamiento basado en contenidos, aspectos que son descritos como características de un ESB. Esto lleva a plantear la búsqueda de las características para la efectiva convergencia de servicios de mensajería a través de un ESB como referencia investigativa, proporcionando la pregunta de exploración para el proyecto: ¿Cómo integrar servicios de mensajería utilizando un ESB?

### **1.3. Objetivos del Trabajo de Grado**

Los objetivos propuestos para este proyecto son los citados a continuación.

**Objetivo general:** Integrar servicios de mensajería instantánea y móvil utilizando un broker basado en ESB.

**Objetivos específicos:** Se establecen tres:

- Adaptar una arquitectura, que permita definir las características de un broker de mensajería basado en ESB, para proporcionar servicios de mensajería móvil y mensajería instantánea.
- Implementar el broker de mensajería móvil y mensajería instantánea basado en ESB.
- Evaluar el broker de mensajería, basado en ESB, a través de un piloto de servicios de mensajería móvil y mensajería instantánea en la Universidad del Cauca.

**Palabras clave:** Enterprise Service bus, Servicios sobre internet, JBossESB, Comunidad académica virtual, Arquitectura de implementación, Mensajería móvil, Mensajería instantánea, Servidor de mensajería.

## CAPÍTULO 2.

### Marco Teórico

#### 2.1. Conceptualización

##### 2.1.1. Mensajería instantánea

El termino mensajería instantánea (IM) hace referencia en Internet a la posibilidad de establecer conversaciones de texto en tiempo real entre individuos. Pero a diferencia del chat, esta comunicación no se establece en una sala en la que hay más personas, sino de forma exclusiva entre dos o un grupo de individuos involucrados, por lo tanto se considera como un punto intermedio entre los sistemas de chat y los mensajes de correo electrónico [6].

La mensajería instantánea, sin considerar quien la provee (ICQ, Yahoo, Microsoft, etc.) es un medio que permite mantener una comunicación constante y a bajo costo, es un servicio sin horarios ni restricciones que ofrece la comodidad de aparecer y desaparecer tan solo presionando un botón [7]. Las herramientas de mensajería instantánea son programas regularmente gratuitos y versátiles, de fácil instalación y uso, que requieren una conexión a internet para su activación. Dichas herramientas pueden ser utilizadas en computadores o dispositivos móviles en los cuales una aplicación permite a estos clientes acceder a los servicios de mensajería.

**Características:** Los sistemas de mensajería tienen unas funciones básicas además de mostrar los usuarios que hay conectados para enviarles mensajes o chatear [8]. Algunas funciones son comunes a todos o casi todos los clientes o protocolos y otras son menos comunes, como las dichas a continuación.

- a. **Contactos:** Los usuarios conectados al sistema pueden mostrar varios estados: Disponible, Disponible para hablar, Sin actividad, No disponible, Vuelvo enseguida, Invisible o No conectado. Con el estado invisible se puede ver a los demás, pero los demás no verán al usuario conectado.

Así mismo el usuario puede mostrar un mensaje de estado cuando se accede al sistema, el cual es una palabra o frase que le es visible a los demás usuarios conectados (contactos añadidos previamente), y que aparece junto a su propio nombre de usuario. A veces, es usado por sistemas automáticos para mostrar la temperatura, o la canción que se está escuchando, sin molestar con mensajes o peticiones de chat continuos. También se puede dejar un mensaje de estado en el servidor para cuando se esté desconectado.

Para registrar usuarios en la lista de contactos propia, al solicitar la inclusión, se puede enviar un mensaje explicando los motivos para la admisión. También al rechazar un contacto discretamente, para que no lo vea al usuario cuando se conecta, se puede realizar sin dejar de estar en su lista de contactos. Solo se deja de avisar cuando se conecta.

A veces puede ser conveniente agrupar los contactos en listas personalizadas: Familia, Trabajo, Facultad, etc. Así mismo se puede usar un avatar, una imagen que lo identifique a uno. No tiene porqué ser la foto del mismo usuario.

**b. Conversación:** Puede haber varios tipos de mensajes:

- **Aviso:** Envía únicamente un mensaje. No es una invitación a mantener la conversación, solo se quiere enviar una información; un ejemplo sería el mensaje del día o MOTD ofrecido por el servidor.
- **Invitación a chatear:** Se invita a mantener una conversación en tiempo real.
- **Mensaje emergente:** Es un aviso que se despliega unos segundos y se vuelve a cerrar. No requiere atención si no se desea. Sirve como aviso breve que molesta lo mínimo posible. Por ejemplo, "ya lo encontré, gracias".

Muchas veces es útil mostrar cuando un contacto está escribiendo un mensaje al usuario, lo cual depende del sistema de mensajería. También dependiendo de la aplicación, se pueden usar emoticonos, y establecer charlas en grupo al estilo IRC MultiUser Chat:

- Se pueden crear salas (grupos de charla), públicas y privadas y también permanentes o que desaparezcan al quedarse sin usuarios.
- Es posible restringir el acceso a salas mediante invitaciones certificadas, para invitar solo a quien uno quiera.

**c. Otras funciones:**

**Envío de archivos:** La mayoría de los clientes de mensajería instantánea, también permiten el envío de archivos, generalmente usando protocolos punto a punto (P2P). Existe la posibilidad de usar también otros sistemas de comunicación, como una pizarra electrónica, o abrir otros programas, como por ejemplo para establecer una videoconferencia.

Los servicios de la mensajería instantánea se utilizan, generalmente de persona a persona, pero se pueden utilizar, mediante un robot de software para que se puedan utilizar de persona a equipo, y en este caso, se trata de servicios sobre la mensajería instantánea, un tipo de aplicación que cada vez es más utilizado, por su rapidez de respuesta y versatilidad de servicios.

### **2.1.2. Mensajería móvil**

El uso de los mensajes de texto cortos o SMS se ha popularizado de tal manera que no es extraño, ni inusual ver que en los programas de televisión o de radio los concursantes participen a través de su teléfono móvil enviando una palabra clave a un número determinado. La masificación de este servicio está altamente ligada a que es prestado a todos los dispositivos móviles debido a sus características técnicas. Sin embargo, con la migración de las redes celulares a generación 3G y generación 3.5G, las tendencias del mercado cambian y de la misma manera las capacidades y características de los equipos, resultando en que los usuarios además de enviar sólo texto estén en la capacidad de transmitir audio, video e imágenes. De esta manera empiezan a tomar fuerza servicios como los de mensajería multimedia o MMS sobre los cuales es fundamental tener conocimiento para estar a la vanguardia en la concepción y desarrollo de aplicaciones [9].

En mensajería móvil existen varios tipos de mensajes: Mensajes SMS, mensajes WAP Push y mensajes MMS [2]. Un mensaje SMS es una cadena alfanumérica de hasta 160 caracteres de 7 bits, y cuyo encapsulado incluye una serie de parámetros. En principio, se emplean para enviar y recibir mensajes de texto normal, pero existen extensiones del protocolo básico que permiten incluir otros tipos de contenido, dar formato a los mensajes o encadenar varios mensajes de texto, dando una mayor longitud [10].

MMS es un estándar de mensajería que permite enviar y recibir contenidos multimedia (incorporando sonido, video, fotos, etc.). La mensajería multimedia permite el envío de estos contenidos, además a cuentas de correo electrónico, ampliando las posibilidades de la comunicación móvil. El límite de cada mensaje multimedia suele ser de 100 o 300 KB, está definido por el operador o las características del terminal y no del protocolo [11]. Es una evolución de los SMS.

Un WAP-PUSH permite acceder mediante uno o varios SMS a la descarga de contenido alojado en un servidor WAP, tal como: aplicaciones JAVA, imágenes, melodías polifónicas, videos, etc. El funcionamiento se basa en una estructura XML, que contiene la dirección o URL del contenido, la cual es compilada y enviada como un SMS binario. Este mensaje es interpretado como una página WAP, que es abierta por el Navegador del teléfono móvil, permitiendo descargar el contenido enlazado [12].

Actualmente, las nuevas formas de comunicación, incluyendo la mensajería instantánea y las redes sociales, son paulatinamente apartadas por los jóvenes tras acceder éstos al mercado de trabajo. El tener menos tiempo libre, más dinero, una más amplia libertad de movimiento y una mayor distancia de su círculo de amigos, conducen necesariamente al abandono del chat, los foros o el blog, y a incrementar el uso del e-mail, el SMS y especialmente la telefonía móvil [13].

Los sistemas de mensajería SMS permiten su integración en múltiples plataformas para el control y gestión de sus procesos productivos. La comunicación de una empresa con sus empleados, clientes y proveedores es la clave de su éxito, y la integración de soluciones de mensajería SMS ofrece grandes posibilidades debido a los siguientes casos:

- Trabajadores itinerantes: En muchos casos la forma de comunicarse con trabajadores que se encuentran fuera de la oficina se realiza mediante llamadas telefónicas. Mediante un adecuado sistema de mensajería SMS es posible enviar información de forma rápida y eficaz a uno o varios trabajadores ahorrando tiempo y dinero en llamadas telefónicas.
- Confirmación a clientes: Confirmar la entrega de un pedido, la recepción de un material o una simple cita es mucho más sencillo y barato mediante mensajes SMS, y además demuestra más profesionalidad y mejora la imagen de la empresa.
- Alertas y avisos: Recibir una alerta o aviso a tiempo puede ahorrarle muchos problemas y dinero. La mensajería SMS puede integrarse con sus sistemas de facturación, alarmas, recepción de pedidos, cadenas de montaje, etc.

Por lo tanto, ayudar a integrar sistemas de mensajería SMS a nivel organizacional puede otorgar el aprovechamiento de las ventajas que brinda este sistema de una forma fácil, rápida y económica.

### **2.1.3. Integración de mensajería instantánea y mensajería móvil**

Las siguientes son las formas de integración, en el caso de los servicios para MI y SMS, que se tendrán en consideración para el cumplimiento de las metas del proyecto. En el anexo A, se ven de forma general los paradigmas de integración, que pudieran usarse para una implementación diferente de la arquitectura modular descrita posteriormente. Cabe resaltar que no se realiza un exhaustivo diagnóstico de otras formas de integración debido a que se contempla la investigación con ESB como el eje del proyecto.

#### **2.1.3.1. Broker de mensajería**

Este tipo de servicio se basa en el empleo de una pasarela, entre una red privada y la red de telefonía móvil, capaz de obtener a partir de peticiones de los usuarios la información que estos requieran. Además, permite a las aplicaciones cliente descubrir las capacidades de la red móvil de forma segura para el operador de telecomunicaciones. De esta manera, el operador permite abrir su red a sistemas y servidores externos, proporcionando un entorno flexible y dinámico en el que se puedan incorporar nuevas capacidades de servicio [14].

El broker debe ser capaz de traducir los mensajes del operador de telefonía en peticiones Web, que le permitan resolver las solicitudes hechas por los usuarios. Una vez resueltos, la plataforma deberá filtrarlos y procesarlos con el objetivo de generar uno o varios mensajes que contengan la información solicitada [14]. En el corazón del servicio del mensaje es el broker, quien encamina los mensajes y los entrega fiablemente, autentifica usuarios, y recoge los datos de rendimiento:

- Para encaminar los mensajes y entregarlos, el broker pone los mensajes entrantes en sus destinos respectivos y dirige la circulación de mensajes dentro y fuera de sus destinos.
- Para suministrar entrega confiable, el broker usa una memoria persistente guardando información y mensajes de estado hasta que son recibidos. Si la conexión o el broker falla, la información salvada permite que el broker restaure su estado y vuelva a procesar las operaciones.
- Suministrar seguridad para los datos que están cambiando, que el Broker usa para autenticación de conexiones. Opcionalmente los datos pueden ser cifrados con el protocolo de seguridad SSL. El broker también usa y dirige un depósito de información sobre usuarios y los datos o las operaciones a las que pueden acceder. El broker autentifica usuarios que piden servicios y autoriza las operaciones que quieren llevar representando la información en su repositorio.
- Para monitorear el sistema, el broker genera las métricas y la información de diagnóstico a las que un administrador puede acceder, medir el rendimiento y mejorar su funcionamiento. Las métricas para las que la información es programable también permiten a aplicaciones ajustar la circulación del mensaje y que se mejoren el rendimiento.

El servicio de cola de mensaje provee una variedad de herramientas administrativas que el administrador puede usar para configurar el soporte del broker.

### 2.1.3.2. Enterprise Service Bus

La intersección de SOA con la integración de aplicaciones y el modelado de procesos de negocio, dan lugar a un nuevo producto denominado ESB. En informática un bus de servicios empresariales consiste en un modelo de arquitectura de software que proporciona servicios fundamentales para arquitecturas complejas a través de un sistema de mensajes (el bus) basado en las normas y que responde a eventos [15]. Dado que el componente investigativo de éste proyecto contempla ESB como base para el desarrollo del mismo, no se hace necesario realizar una investigación para determinar el mejor paradigma de investigación.

Los desarrolladores normalmente implementan un ESB utilizando tecnologías de productos de infraestructura de middleware que se basan en normas reconocidas. Es un concepto antiguo, pero que adquiere fuerza de nuevo gracias a la facilidad y bajo costo con el que se puede implementar actualmente [16] [17].

ESB es un middleware, que proporciona servicios para la construcción de arquitecturas más complejas, basadas en eventos y un motor de mensajería (el BUS). ESB permite la integración de aplicaciones en forma rápida, directa y basada en estándares. Es una suite de productos independientes de la infraestructura que facilita el proceso, la transformación de datos, el enrutamiento y la orquestación de procesos usando Web Services [16] [17]. La **figura 1** muestra la arquitectura ESB. En este sentido, los principales beneficios de ESB [15] son:

- La integración de sistemas existentes es más rápido y a menor costo.
- Tiene mayor flexibilidad, es más fácil de cambiar si hay nuevos requisitos.
- Es basado en estándares.
- Posibilidad de escalar desde soluciones puntuales hasta implementaciones de empresa (bus distribuido).
- Tipos de servicio listos-para-funcionar (ready-to-use) predefinidos.
- Mayor configuración en vez de tener que codificar la integración.
- Parches incrementales con tiempo de apagado instantáneo, es decir, la empresa se hace "refactorizable".

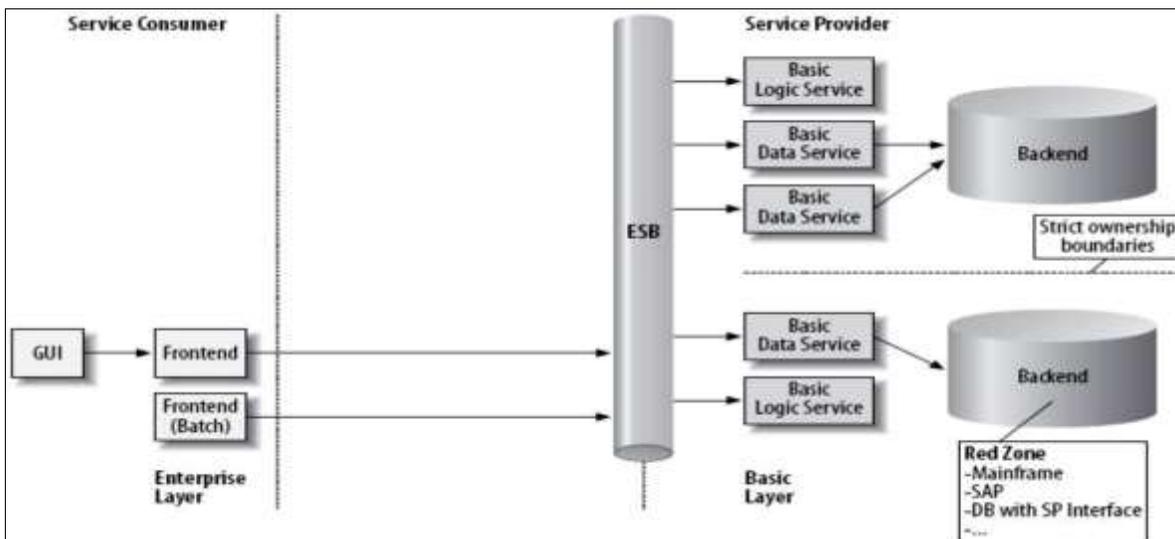


Figura 1. Arquitectura ESB

Examinando las capacidades del bus de servicios a escala empresarial, cada una de las siguientes funciones es un elemento esencial para la satisfactoria integración de una SOA. Juntos, estos elementos resuelven los problemas a los que se enfrentan los clientes y los proveedores de servicios en un entorno de SOA [4]:

**Mensajería distribuida:** El núcleo del ESB lo constituye una aplicación de middleware orientada al mensaje. Este núcleo proporciona un método de transporte fiable y distribuido que emplea un mecanismo de almacenamiento y reenvío gracias al cual se garantiza la entrega de los mensajes incluso en caso de anomalías en la red.

**Transparencia de las ubicaciones:** Con la mediación entre servicios, un servicio cliente que invoque al proveedor de servicio solo necesita saber que el servicio existe; el cliente no necesita saber dónde se está ejecutando el servicio. El ESB localiza el servicio cuando se invoca. Esto proporciona un cierto nivel de virtualización de los servicios y de transparencia de las ubicaciones, de forma que si un equipo falla, o si se cambia la ubicación de un proveedor de servicio, no es preciso notificar el cambio a cada uno de los clientes individuales. Esto puede contribuir significativamente a la reducción de los costes de gestión de las TI y a minimizar los riesgos.

**Transparencia del transporte:** En los enfoques tradicionales de la integración punto a punto todos los componentes y objetos están muy estrechamente acoplados. En SOA, los servicios están repartidos por todo el entorno de TI y su acoplamiento es menos estricto, gracias a la transparencia de las ubicaciones. Además de apoyarse en la transparencia de las ubicaciones para conectar clientes y proveedores de servicios, el ESB también proporciona protocolo de transporte físico para hacer posible la comunicación entre servicios utilizando transportes diferentes.

**Soporte multiprotocolo:** Debido a que plantea requerimientos de fiabilidad inherentes y solamente funciona bien con patrones de intercambio de mensajes (MEP) sincrónicos, el modelo de transporte HTTP no satisface los requisitos de todos los servicios y aplicaciones. Por ejemplo, el servicio de mensajes de Java (JMS) además de poseer características asincrónicas, ofrece más fiabilidad en el transporte que HTTP. Para compatibilizar el comportamiento de las aplicaciones individuales, algunos sistemas recurren a SOAP a través de JMS. También se usan otros tipos de modelos de transporte, entre los que se cuentan los sistemas de transporte propietarios de algunos de los principales proveedores de sistemas y soluciones de planificación de recursos empresariales. ESB necesita, por lo tanto, ser capaz de soportar muchos tipos de sistemas de transporte para integrar sistemas dispares y gestionar el transporte de comunicaciones complejas eficazmente.

**Calidad de servicio:** En las aplicaciones empresariales, la calidad de servicio (QOS) hace referencia, fundamentalmente, a su fiabilidad. La entrega de los mensajes y la fiabilidad de servicios de invocación son funciones de misión crítica de cualquier sistema. Aun los servicios sobre internet por sí solos no ofrecen un servicio de entrega garantizada. Un ESB, por otro lado, puede proporcionar un servicio de alta fiabilidad garantizando la entrega del mensaje de principio a fin que va más allá de la fiabilidad que puede ofrecer transportes como JMS. Asimismo, los métodos empleados para alcanzar un alto nivel de QOS deben satisfacer los estándares existentes, como, por ejemplo, ser compatibles con la especificación WS-ReliableMessaging.

**Patrones de intercambio de mensajes:** En la actualidad, la mayoría de los ESB se basan en un paradigma de solicitud/respuesta usando SOAP sobre HTTP; esto significa que el servicio cliente lanza un mensaje de solicitud al usuario y espera a recibir la respuesta. Esto se conoce como un MEP sincrónico. Sin embargo, en el MEP de publicación/suscripción, el servicio cliente puede enviar un mensaje y suscribirse a la respuesta, en lugar de esperar a recibirla. El MEP de publicación/suscripción puede responder de forma más eficaz a eventos en un contexto empresarial, en particular cuando el ciclo de vida de una acción de servicio tiene lugar durante períodos de tiempo prolongados. Un ESB debe ser capaz de manejar ambos paradigmas.

**Enrutamiento basado en el contenido:** Existen dos tipos de enrutamiento dentro de un ESB. El primer servicio de enrutamiento se produce cuando la invocación de un servicio entra en el ESB y éste encamina la respuesta al proveedor de servicio apropiado, sin necesidad de que el servicio cliente conozca la ubicación del proveedor del servicio. Así es cómo se logra la transparencia de las ubicaciones que antes se ha comentado.

El otro tipo, enrutamiento basado en el contenido, introduce una serie de reglas o una lógica de negocio que se aplica al contenido del mensaje en la etapa del enrutamiento y que hacen posible que el ESB encamine los mensajes a proveedores de servicio específicos basándose en su contenido; dando prioridad, por ejemplo, a los pedidos de determinados clientes o marcando los pedidos de gran tamaño para darles un tratamiento especial. Esto ofrece a las empresas un servicio muy valioso, ya que puede contribuir a reducir el coste de la gestión de la Información, garantiza que se respeten los acuerdos a nivel de servicio y permite a las empresas centrarse en actividades para mejorar la satisfacción de sus clientes.

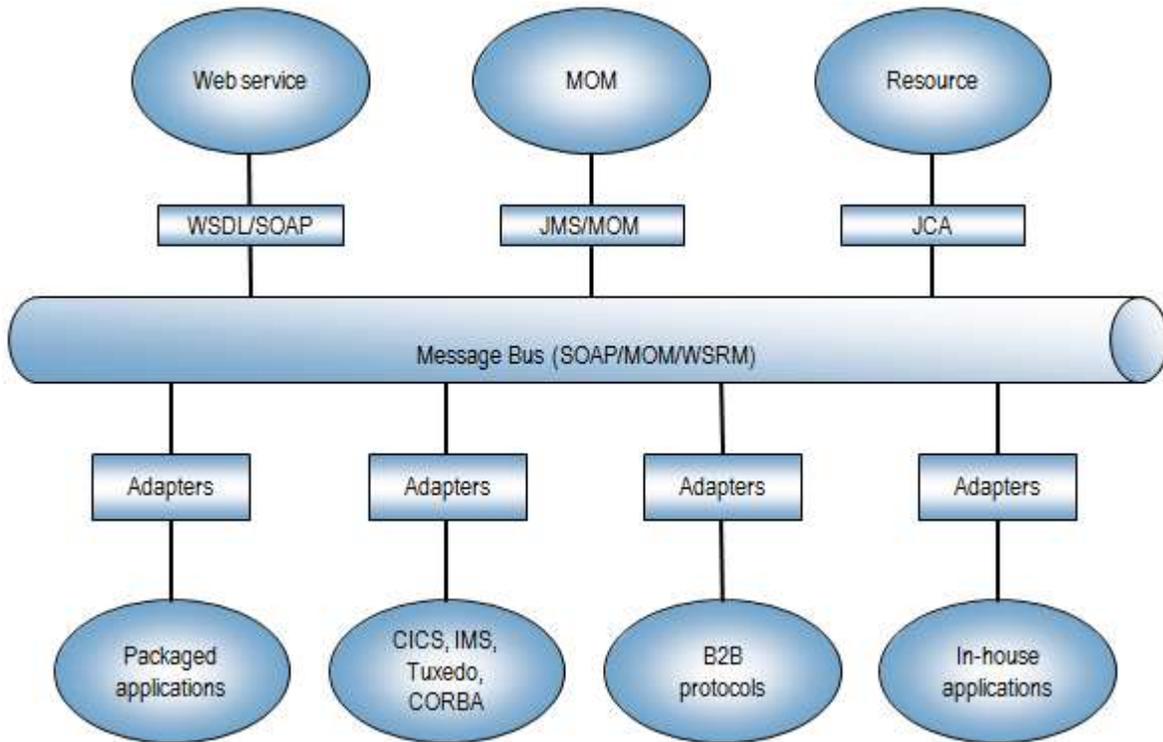
**Transformación:** Si bien la tarea de un ESB es dirigir mensajes de un servicio al siguiente, hay ocasiones en que el formato de los datos de un servicio no satisface los requisitos del siguiente servicio. Por ese motivo, el ESB debe ser capaz de transformar los datos de un formato a otro.

Por lo tanto, la necesidad de un ESB para la arquitectura que se plantea en el siguiente capítulo, surge de divisar la complejidad de las organizaciones que deben coordinar e integrar sus procesos de negocio, sistemas operacionales y datos sin renunciar a la innovación tecnológica imprescindible para ser competitivos, mediante una arquitectura que permita mantener integrados los sistemas, nuevos y legados, en un estilo completamente distribuido e interoperable.

#### **2.1.3.2.1. Tipos de arquitecturas ESB**

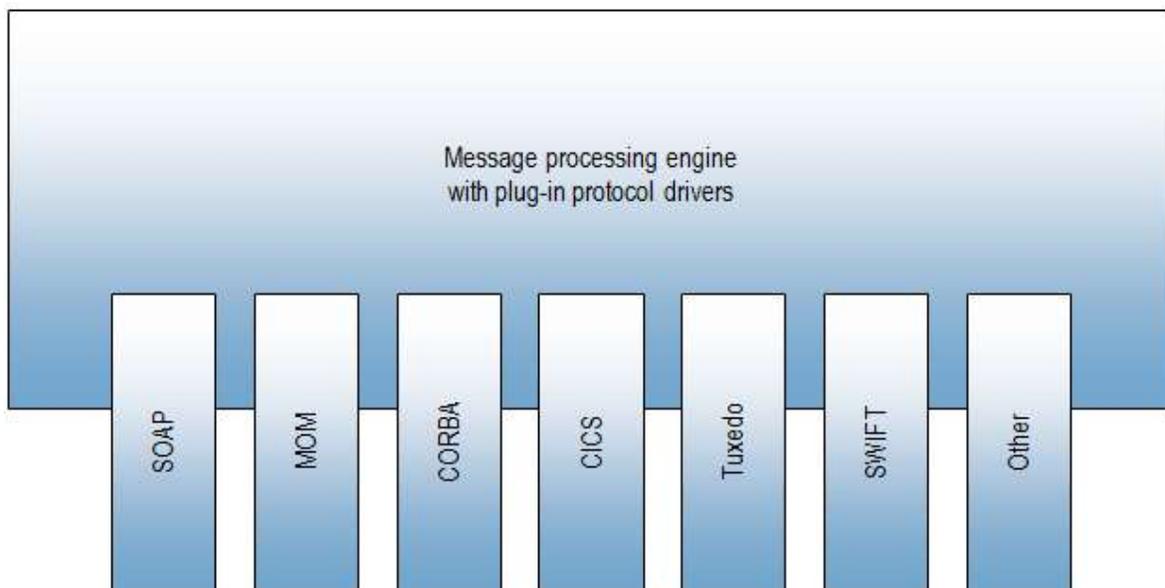
Existen principalmente tres tipos de arquitecturas posibles para desplegar una plataforma ESB [18], representadas a continuación. De los tres tipos de arquitecturas, es posible observar que Message Bus Architecture es la que proporciona los mecanismos necesarios para la funcionalidad del proyecto, descritos en el planteamiento del problema y que se detallan en el capítulo 3. En ese sentido, dicha arquitectura se tomará como referencia para describir la modularidad del sistema en el siguiente capítulo, debido a su capacidad de soportar la mayoría de las características SOA y ESB a diferencia de las demás.

a. Arquitectura ESB: Message Bus Architecture (**figura 2**)



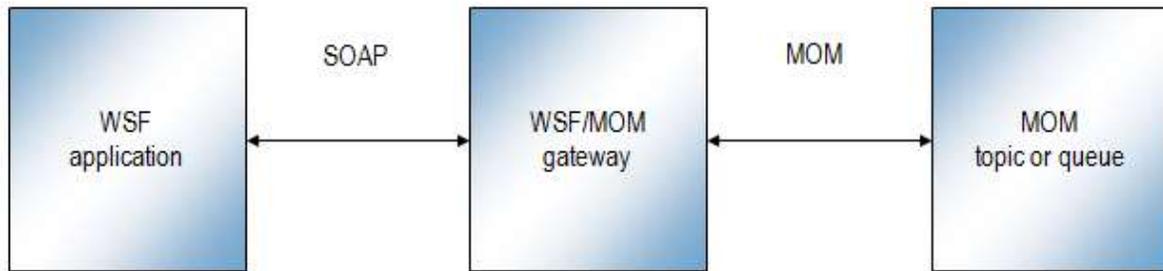
**Figura 2.** Message bus architecture

b. Arquitectura ESB: Protocol Switch (**figura 3**)



**Figura 3.** Protocol switch

c. Arquitectura ESB: Gateway (**figura 4**)



**Figura 4.** Gateway

### 2.1.3.2.2. Comparación de software ESB

La comparación puede ser hecha desde 2 puntos de vista:

- a. Si de la experiencia obtenida de la implementación de diferentes herramientas ESB para la misma arquitectura, se obtienen las características relevantes de las herramientas y esto da la posibilidad para la escogencia de alguna de ellas (o varias) para futuros escenarios.
- b. Si se escoge la evaluación realizada de la implementación de varias herramientas ESB por parte de un autor (o varios autores), para a partir de ésta, escoger la herramienta que se utilizará en un escenario.

A partir de lo anterior y teniendo en cuenta el hecho de que existen varias herramientas ESB, las cuales han sido comparadas para determinar diferentes funcionalidades o requisitos esperados para determinados servicios, se tendrá en cuenta la evaluación de las herramientas ESB realizadas por otros autores o empresas en conjunto con la experiencia obtenida en la realización del proyecto, para a partir de esto, concluir del porque de la escogencia de las herramientas ESB para este proyecto y la definitiva para su implementación o servicio de prueba [19] [20].

Consiguientemente, las funcionalidades que se deben tener en cuenta en las capacidades de una herramienta ESB para su escogencia pueden ser [21] [22]:

- Poseer interacciones basadas en eventos.
- Funcionar en ambientes heterogéneos.
- Separar la lógica de Integración de la lógica del negocio.
- Proveer independencia de los lenguajes que implementan los contratos de los servicios.
- Manejo de transacciones globales.
- Control de versiones.
- Soportar procesamiento sincrónico y asincrónico.
- Soportar encriptación, autenticación y autorización.
- Exponer servicios tanto interna como externamente.
- Poseer un registro de servicios.
- Proveer transformación y enriquecimiento de mensajes.

- Realizar enrutamiento dinámico basado en contenido.
- Proveer orquestación de servicios.
- Proveer coreografía de servicios.
- Permitir el acceso a sistemas legados.
- Proveer administración y monitoreo.
- Proveer acceso distribuido a repositorios de datos (persistencia).
- Proveer acceso a servicios externos.
- Utilizar estándares.
- Alto uso de la configuración en vez de la codificación.
- Componer servicios a partir de servicios ya existentes a través de lenguajes no procedurales, metalenguajes, etc.
- Message mapping.

Por lo tanto, en un estudio [23] citado a manera de ejemplo, se invitó a 3 proveedores ESB y brindaron el apoyo para llenar las pautas de la evaluación. Estos productos son:

- a. Biztalk 2006
- b. Open ESB
- c. Sonic ESB

Las siguientes gráficas muestran algunos resultados interesantes:

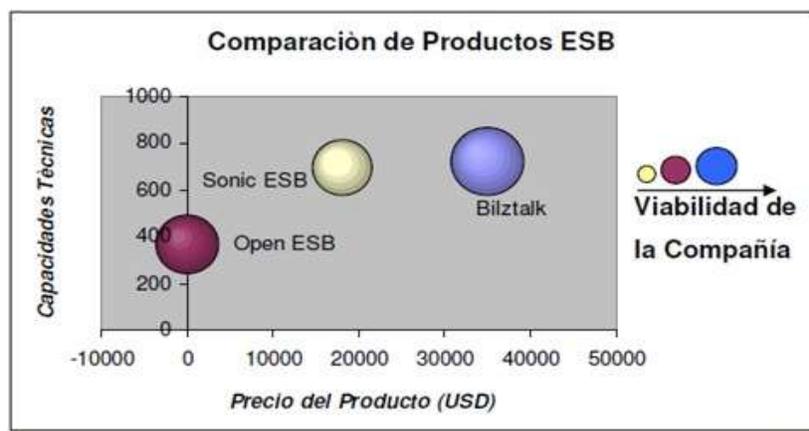


Figura 5. Comparación de productos ESB 1

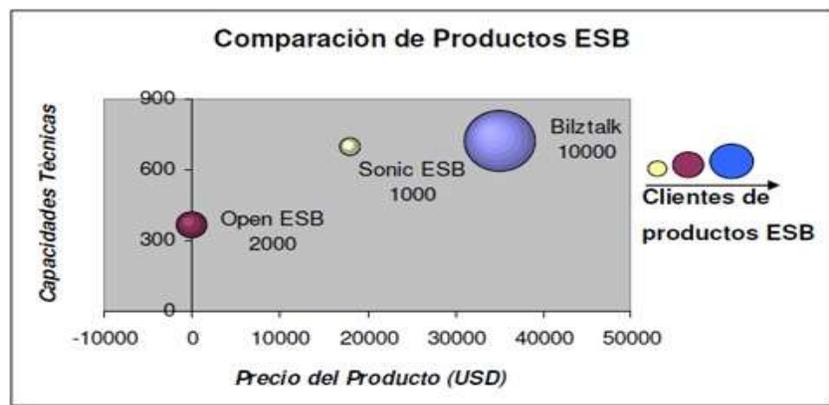


Figura 6. Comparación de productos ESB 2

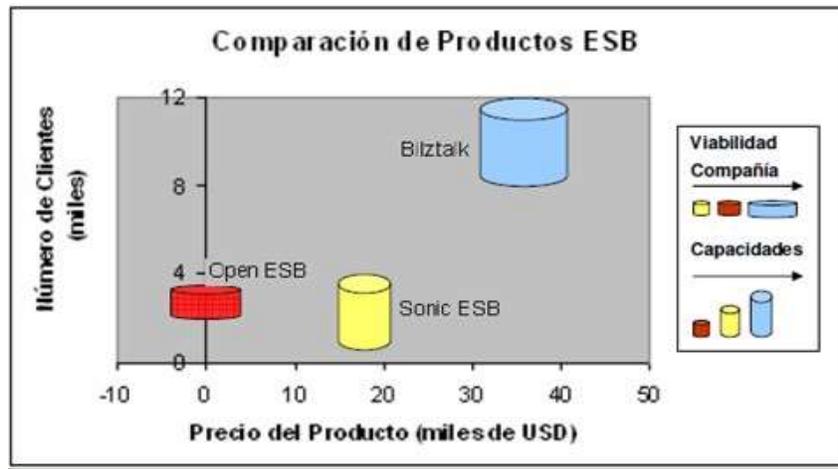


Figura 7. Comparación de productos ESB 3

Consecuentemente los resultados del estudio fueron:

Biztalk Server 2006 R2, es el proveedor más consolidado de los comparados; ofrece mayores capacidades y funcionalidades; tiene el mayor número de usuarios; pero es el producto más costoso de los evaluados.

Open ESB v2, tiene un proveedor consolidado; posee relativamente pocas capacidades; tiene buena presencia en el mercado; y no tiene costo de adquisición, aunque su "Total Costo of Ownership" claramente no es cero.

Sonic ESB 7.6, proviene de la compañía creadora del concepto, por lo que tiene buena presencia en el mercado; posee relativamente pocas capacidades comparada con el resto de los productos valuados; pero es un producto altamente interoperable, capaz de integrarse con varias tecnologías; su costo es el menor de los productos bajo licencia, aunque se prevé un aumento del costo para futuras versiones.

De lo anterior, se puede observar que la comparativa de software de integración empresarial [24] puede resumirse de varias herramientas a unas pocas que sean las más relevantes para este proyecto, y dado que no es necesario el aspecto comercial de la herramienta ya que esta proyecto es para una ambiente académico, un ámbito de funcionalidades breve de esta nueva comparación sería:

- Implementaciones de la Arquitectura Orientada a Servicios.
- Middleware orientado a mensaje y brokers de mensajería.
- Implementaciones del Enterprise Service Bus.
- Implementaciones BPEL.
- Software EAI (Enterprise application integration).

Entonces como se puede ver en la **tabla 1**, el resultado de la evaluación de algunas herramientas para implementar un ESB en relación a compatibilidades es:

Software de integración empresarial	Creador	Compatibilidad con SO (Windows, Mac, Linux Red Hat)	Soporte de BD (API JDBC, Microsoft SQL Server, MySQL, Oracle)
<b>JBoss ESB</b>	JBoss Inc.	Todos	Todos
<b>BizTalk Server</b>	Microsoft	Windows	Microsoft SQL Server
<b>Mule ESB</b>	Mule Source	Todos	Todos
<b>Open ESB</b>	Sun Microsystems	Todos	Todos
<b>Oracle ESB</b>	Oracle Corporation	Todos	Oracle
<b>ServiceMix</b>	Apache Software Foundation	Windows y Unix	Todos
<b>WebSphere Message Broker</b>	IBM	Todos	API JDBC, Microsoft SQL Server, Oracle

*Tabla 1. Comparativa de compatibilidad de software para implementaciones ESB*

En el anexo B se da una descripción general de cada herramienta ESB. Seguidamente, como se buscaba una herramienta que cumpliera con las siguientes características:

- Estabilidad de la aplicación, que no sea beta.
- Debe poseer documentación efectiva, con completos manuales de usuario.
- No restricción para la ejecución de la herramienta, es decir, que sea software libre.
- En relación a otras herramientas, debe ser simple su utilización.
- Las funcionalidades que brinda deben ser afines con los mecanismos necesarios para la implementación del proyecto.

Se establece que JBoss ESB es la herramienta que cumple con los requisitos para llevar a cabo la puesta en marcha del proyecto [25][26]. Esta información es útil para la temática del capítulo 5.

## 2.2. Proyectos Relacionados

A continuación se presentan los trabajos relacionados y sus aportes para este proyecto.

### 2.2.1. A process modeling-based approach for web service management

La calidad en los servicios individuales es importante para garantizar un alto rendimiento y disponibilidad de los sistemas orientados a servicios. Esto requiere un proceso de diagnóstico de la calidad de servicio y responder a este de manera oportuna. Se presenta un enfoque basado en modelado, para coordinar el proceso de administración de servicios de la Web, configurar los parámetros o invocar a otras tareas. Este enfoque utiliza procesos ejecutables para representar la lógica de diagnóstico [27]. Se implementa una arquitectura basada en ESB que permite integrar servicios y redirigir los mensajes

con poca penalización en el rendimiento. Se demuestra la eficacia de este enfoque utilizando un servicio de préstamo de intermediación Web en ESB.

### **2.2.2. A survey on web services in telecommunications**

Este artículo presenta un estudio sobre una arquitectura orientada a servicios y los servicios web dentro de las telecomunicaciones. A través de los últimos años las telecomunicaciones han estado en un estado de constante flujo, con los cambios en las regulaciones, una mayor competencia, y el progreso tecnológico. El artículo describe estas fuerzas en detalle y muestra que la necesidad de la adopción exitosa de SOA dentro de las telecomunicaciones se ha convertido en un importante tema para la mayoría de los operadores de redes de telecomunicaciones. El artículo, procede a describir SOA y sus herramientas clave en las telecomunicaciones como servicios web, arquitecturas orientadas a eventos, especificaciones Parlay X/ECMA y el Enterprise Service Bus [28], y se realiza un análisis en profundidad de SOA y de su aplicación para los servicios Web en las telecomunicaciones.

### **2.2.3. A UML model-driven business process development methodology for a virtual enterprise using SOA & ESB**

El propósito de este trabajo es demostrar como una Integración Empresarial Virtual (IEV) usando Arquitectura Orientada a Servicios (SOA) y Enterprise Service Bus (ESB) puede ser efectivamente llevada a cabo por un equipo virtual de corredores de servicio. En la actualidad, IEV se logra a través de SOA y ESB utilizando servicios web y los motores de los procesos de negocio que ejecutan WSDL y WS-BPEL. Para rediseñar un Virtual Enterprise (VE) con base en uno o más componentes heredados, conocidos como un legado de VE, las partes abstractas y concretas de los procesos de negocio relevantes de la VE necesitan ser diseñados para un alto nivel de abstracción [29]. Con base en los resultados de ingeniería inversa y directa de dos casos de empresa virtual, las directrices, que utilizan UML como un proyecto con puntos de vista multi-arquitectura, ayuda a los agentes del servicio (o brokers) a comprender el proceso subyacente de la arquitectura y la organización de un empresa virtual que se ha construido utilizando el concepto de SOA y ESB contemporáneos.

### **2.2.4. The enterprise service bus: Making service-oriented architecture real**

Enterprise Service Bus (ESB) es la infraestructura que sustenta plenamente una arquitectura integrada y flexible de extremo a extremo orientada a servicios (SOA). En este artículo se detalla esencialmente los meta-datos y las capacidades de ESB. Se presenta un resumen de los conceptos clave de la ESB y define el modelo de integración para él, incluyendo claves de usuario roles. Estas funciones se cumplen con los meta-datos que describen los criterios de valoración de servicios, tales como la interfaz de servicio y las exigencias de políticas y capacidades. La ESB gestiona estos meta-datos a través de un registro, que admite configuración, conexión, emparejamiento, y el descubrimiento de los extremos de servicio. Algunos patrones típicos de la mediación que se utilizan para satisfacer las políticas de punto final se exploran, y el uso de patrones se describe en cual ESB se utiliza para implementar SOAs reales [30]. Se identifica la gestión del servicio de meta-datos a través de un registro de servicios como una clave componente del ESB, lo que permite la integración de especialistas y administradores para crear y administrar soluciones orientadas a servicios.

### **2.2.5. Mensajería en sistemas de información**

Explica las soluciones basadas en mensajería, para esto analiza qué es mensajería, cuáles son sus aportes a los sistemas de información, contextos de aplicación y las características que esta tecnología presenta. Estudia tecnologías y herramientas que facilitan la implementación de este tipo de soluciones y los patrones de mensajería. Estas herramientas van desde soluciones comerciales, pasando por Frameworks que facilitan su implementación, hasta soluciones basadas en estándares independientes de la plataforma subyacente [31]. Se brinda en un marco teórico para entender las soluciones basadas en mensajería y el estudio de patrones de mensajería brindando lineamientos para la implementación de estos en un contexto orientado a servicios.

### **2.2.6. Integración de aplicaciones de seguridad**

Poco a poco la informática ha inundado las empresas y en estos momentos muchas de las mejores se hacen añadiendo servicios a una arquitectura ya de por sí muy complicada. El crecimiento de las arquitecturas ha hecho que surja un nuevo problema, la administración, comunicación e interoperabilidad entre todos los servicios [32]. Los ESB, solucionan este problema ofreciendo una gran capacidad de integración y administración de los servicios. La tecnología de ESB a día de hoy como un pilar principal de las arquitecturas orientadas a servicios (SOA), y aun y así, debe entenderse esta como una tecnología incipiente y no se puede saber hasta qué punto puede llegar en el futuro. Se incluye y documenta desde la recopilación de todas las herramientas que se han escogido, probado y comparado, hasta dos casos de uso complejos donde se aprovecha el uso de la tecnología.

### **2.2.7. Contribuciones y diferencias con los proyectos relacionados**

En la **tabla 2** se listan los aportes o contribuciones y las brechas de los anteriores proyectos con respecto a este trabajo.

Proyecto	Aporte	Brecha
A process modeling-based approach for web service management	Este enfoque utiliza modelos de procesos ejecutables para representar la lógica de diagnóstico y orquestar la sustitución. Los modelos de proceso pueden interactuar con el sistema y aceptar instrucciones de los administradores a nivel de proceso.	No integra las alternativas a sistemas orientados a servicios de mensajería SMS e IM. Nuestra propuesta busca la convergencia de dichos servicios.
A survey on web services in telecommunications	El artículo describe en detalle y muestra que la necesidad de la adopción exitosa de SOA dentro de las telecomunicaciones se ha convertido en un tema importante para la mayoría de los operadores de redes de telecomunicaciones.	Da una explicación de SOA en las Telecomunicaciones actualmente, pero no describe la implementación específica de un sistema.
A UML model-driven business process development methodology for a virtual enterprise using SOA & ESB	Demuestra como un sistema de integración Virtual Enterprise (IVE) utilizando SOA y ESB puede realizarse eficazmente por un equipo virtual de corredores de servicio.	Se enfoca en Virtual Enterprise y en procesos de negocio para demostrar una integración empresarial.
The enterprise service bus: Making service-oriented architecture real	Detalla los datos esenciales y capacidades de ESB. Se presenta un resumen de los conceptos claves de la ESB y define el modelo de integración para ella, incluyendo los roles clave de usuario.	No explora herramientas para la implementación de un ESB, aunque da una muy detallada conceptualización de lo que es ESB.
Mensajería en sistemas de información	Se centra en el estudio exhaustivo de los patrones de mensajería y se analiza la implementación de éstos en base a las herramientas y tecnologías que menciona y se proponen categorías para agrupar a los mismos.	Se centran en soluciones para servicios web. La propuesta actual, busca integrar servicios sobre Internet con la mensajería de texto e instantánea basados en ESB.
Integración de aplicaciones de seguridad	Presenta las características más relevantes para la decisión en la escogencia de una herramienta de software ESB.	Se centra en una implementación de ESB ligada hacia la seguridad de la información, a diferencia de esta propuesta que se centra en los servicios de mensajería.

**Tabla 2.** Proyectos relacionados

## **CAPÍTULO 3.**

### **Arquitectura para la Convergencia de Servicios de Mensajería Instantánea y Móvil mediante un Broker basado en ESB**

La propuesta descrita a continuación parte de una caracterización realizada con el fin de comprender y construir una base de conocimiento del estado actual de la arquitectura de integración de servicios para este proyecto, que reforzará lo presentado en el capítulo previo. Así, se identificaron: Las características generales, los servicios ofertados y las tecnologías usadas. A partir de esta base conceptual y teniendo en cuenta la definición que provee el IEEE acerca de una arquitectura, la cual se refiere a esta como: “una organización fundamental de un sistema enfatizando en sus componentes, las relaciones entre ellos y el ambiente y los principios que orientan su diseño” [33] [34], se describe la propuesta para la implementación de un Broker basado en ESB para integrar servicios de mensajería instantánea y móvil, iniciando por la especificación del patrón de arquitectura, continuando con la descripción de una serie de vistas que posibilitan el fácil entendimiento y comprensión modular de la misma, con un conjunto de tecnologías sugeridas para su implementación.

En primer lugar en el proceso de adaptación, se determinan las necesidades que debe suplir la arquitectura que permita converger servicios de mensajería móvil e instantánea mediante un broker basado en ESB:

- Necesidad de un escenario acorde a las características de una comunidad virtual académica.
- Un sistema con conexión a la red móvil por intermedio de un modem GSM.
- Proporcionar el envío y recepción de SMS.
- Permitir el acceso de usuarios al chat mediante un servidor de mensajería instantánea.
- Los usuarios deben ingresar al chat haciendo uso de SMS.
- Un usuario debe también ingresar al chat mediante una aplicación instalada en el móvil.
- El sistema debe permitir crear un grupo especial de usuarios, con los cuales una persona se comunica constantemente y que no cambia con los semestres.
- Deben poder enviarse mensajes de notificación o urgentes, a uno o un grupo de usuarios sin necesidad de que ellos estén conectados.
- Los mensajes de notificación deben llegar a un usuario a su móvil a través SMS, si el mensaje instantáneo lo envían cuando éste está desconectado y previamente ha aceptado recibir este tipo de mensaje.
- El usuario SMS puede iniciar sesión, cerrar sesión y enviar mensajes instantáneos.
- Los mensajes instantáneos deben ser enviados a un teléfono móvil a través de un SMS.

Adicionalmente se presentan de forma explícita las características mencionadas en la definición del planteamiento del problema, necesarias en una arquitectura para la efectiva

convergencia de los servicios de la mensajería móvil e instantánea, y que son usadas como guía para el proceso de adaptación:

- ✓ Integración de tecnologías disímiles.
- ✓ Mensajería distribuida.
- ✓ Conexión a una infraestructura común.
- ✓ Transparencia de las ubicaciones y virtualización.
- ✓ Enrutamiento basado en contenidos.

En la **tabla 3** se relaciona cada arquitectura de despliegue junto a las 5 características requeridas para el prototipo, con el fin de establecer cuáles son las características que cumple cada arquitectura.

Tipo de despliegue ESB	Características de la arquitectura de despliegue
<b>Gateway</b>	<ul style="list-style-type: none"> <li>✓ Integración de tecnologías disímiles.</li> <li>✓ Conexión a una infraestructura común.</li> </ul>
<b>Protocol Switch</b>	<ul style="list-style-type: none"> <li>✓ Integración de tecnologías disímiles.</li> <li>✓ Conexión a una infraestructura común.</li> <li>✓ Transparencia de las ubicaciones.</li> <li>✓ Enrutamiento basado en contenidos.</li> </ul>
<b>Message Bus Architecture</b>	<ul style="list-style-type: none"> <li>✓ Integración de tecnologías disímiles.</li> <li>✓ Mensajería distribuida.</li> <li>✓ Conexión a una infraestructura común.</li> <li>✓ Transparencia de las ubicaciones.</li> <li>✓ Enrutamiento basado en contenidos.</li> </ul>

**Tabla 3.** Características de las arquitecturas ESB

Gateway, debido a su despliegue realizado mediante una puerta de enlace, permite la integración de tecnologías disímiles y la conexión a una infraestructura común. Sin embargo no maneja mensajería distribuida debido a que para eso es necesario poseer de un núcleo middleware orientado a mensajes, el cual es requisito para el enrutamiento basado en contenidos. La transparencia de las ubicaciones no es posible debido a que no tiene suficientes servicios que se invoquen.

Protocol switch, en cambio permite 4 de las características del prototipo, sin embargo con relación a la mensajería distribuida, no es soportada debido a que éste tipo de arquitectura conmuta protocolos a través de un motor de procesamiento mediante plugins, sin especificar el uso de middleware orientado a mensajes.

Pero MBA (Message Bus Architecture), si contempla el cumplimiento de las 5 características, debido a que posee una troncal común, que permite integración de tecnologías disímiles. Por lo tanto, mediante una implementación del núcleo del ESB, se pueden brindar transparencia de las ubicaciones y enrutamiento basado en contenidos; logrando prevalecer por encima de las demás despliegues debido a que su forma estructural da para proveer de mayores funcionalidades al prototipo.

### 3.1. Descripción de la Arquitectura

Para la descripción de la arquitectura se buscó un esquema de organización que permitiera plasmar la base de conocimiento, de tal forma que facilitara expresar los subsistemas necesarios, los métodos de comunicación y la colaboración entre ellos, en

una estructura de fácil comprensión [34]. Por lo tanto se tuvieron en cuenta dos referentes para su descripción:

- Teniendo en cuenta los tipos de arquitectura ESB, se escoge message bus architecture, como se menciona anteriormente, porque provee de un diseño que permite la integración de servicios, en donde se conecten solo una vez a una infraestructura troncal común, soportando mensajería distribuida, para otorgar transparencia en la ubicación de los servicios, transparencia en el transporte y enrutamiento basado en contenidos.
- De acuerdo al estándar IEEE 1471, denominado “Recommended practice for architectural description of software intensive systems” el cual provee un marco conceptual para la descripción de arquitecturas de software [26], incluidas una serie de vistas para su propósito.

### **3.1.1. El patrón de arquitectura**

Considerando que el patrón arquitectónico a usar debía brindar un alto nivel de interoperabilidad entre aplicaciones internas y externas y esbozar un escenario extensible, SOA (Service Oriented Architecture, Arquitectura Orientada a Servicios), que es un estilo de arquitectura de software para sistemas distribuidos basado en la definición de servicios reutilizables, con interfaces bien definidas, donde los proveedores y consumidores de servicios interactúan en forma desacoplada para realizar procesos [33], lo cual como se ha indicado anteriormente, es detallado luego de llevar a cabo una implementación ESB.

Explícitamente la W3C define SOA como: “un conjunto de componentes que pueden ser invocados, cuyas descripciones de interfaces se pueden publicar y descubrir”. Según CBDI, empresa especializada en la investigación de SOA, quien proporciona una definición más amplia que la suministrada por la W3C, SOA se precisa como: “Un estilo resultante de políticas, prácticas y frameworks que permiten que las funcionalidades de una aplicación se puedan proveer y consumir como conjuntos de servicios, con una granularidad relevante para el consumidor. Los servicios pueden invocarse, publicarse y descubrirse y están abstraídos de su implementación utilizando forma estándar de interfaces” [34]. Esta arquitectura se basa en 3 componentes esenciales describe 3 tipos de interacciones entre estos componentes (localizar servicios, publicar servicios y acceder o interconectar servicio) [35] [36].

Es necesario precisar que SOA como patrón arquitectónico no define un conjunto de tecnologías a las que una implementación deba ceñirse, en este sentido, si bien los WS (Web Services, Servicios Web) basados en SOAP (Simple Object Access Protocol, Protocolo de Acceso Simple a Objetos) y RPC (Remote Procedure Call, Llamado a Procedimiento Remoto) son una interface programática o más bien una implementación adecuada en conformidad con SOA, no son obligatorios para la implementación de una arquitectura orientada a servicios [37] [43].

Aclarado lo anterior, es preciso mencionar que en la arquitectura descrita más adelante los servicios se rigen bajo los principios de REST (Representation State Transfer, Transferencia de Estado Representacional) [42], la cual es un estilo arquitectónico orientado a recursos, introducida por Roy Fielding, a la hora de implementar SOA, en su tesis de doctorado en el año 2000. Entre el conjunto de principios que rigen a REST se encuentran [38]:

- Los estados, datos y funcionalidades son representados como recursos. Esto significa que cada entidad (tal como un consumidor, localización o una dirección) es direccionada como un dato y es representado como un recurso. Cada recurso es identificado con una dirección única a través de una sintaxis universal basada en identificadores uniforme de recursos (URI).
- Todos los recursos son compartidos uniformemente a través de una interfaz simple que consta de un conjunto de operaciones y tipos de datos bien definidos, basados típicamente en el protocolo HTTP.
- La información entre el servidor y el cliente es transferida a través de XML, JSON (JavaScript Object Notation, Notación de Objetos de JavaScript) o ambos.

El uso de REST como estilo arquitectural, fue seleccionado debido a las enormes ventajas que este ofrece a un ambiente móvil en tanto al consumo de ancho de banda necesario para el envío y recepción de mensajes, limitado uso de recursos gracias a un esquema de mensajes compacto y soporte global a dispositivos debido al uso de HTTP [39] [41].

De acuerdo a las necesidades del prototipo y la escogencia de MBA como patrón de arquitectura para desplegar, se realiza la búsqueda y especificación de las características restantes para la adaptación. Como se conocen las 5 características base, se determina de acuerdo a la definición de ESB, las demás características que llevarían a una correcta implementación, y ellas son: Soporte multiprotocolo, adaptación, transformación y enrutamiento.

A partir de lo anterior, para lograr conceptualizar una arquitectura, se establecen las funcionalidades de servicios que debe tener la misma, a través de los servidores que se utilizarían posteriormente en el prototipo:

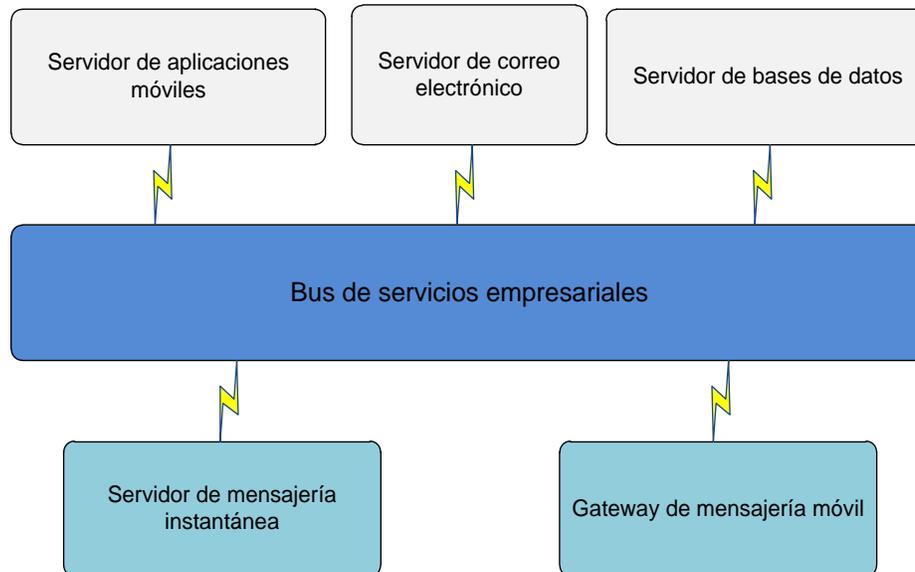
- Servidor de aplicaciones móviles: Necesario para hacerse cargo de la toma de decisiones respecto a un mensaje.
- Servidor de bases de datos: Encargado de mantener en orden los datos.

Por otra parte, se necesitaba un medio de acceso por parte de los usuarios a las funcionalidades de la arquitectura, logrando establecer dos componentes:

- Servidor de mensajería instantánea: El encargado de proveer la comunicación instantánea a usuarios de escritorio o móviles.
- Gateway de mensajería móvil: Debe establecer la comunicación entre el ESB y la red de un proveedor de telefonía móvil.

Por simplicidad se toma cada servidor como un módulo para describirlo más detalladamente en secciones posteriores, y de ésta forma darle una mayor versatilidad a la arquitectura. De ésta forma es posible concluir que se pueden adicionar otros servidores como módulos para aumentar el número de servicios al usuario. Por lo tanto se adiciona un servidor de correo electrónico para mejorar la experiencia del usuario en relación a los servicios de mensajería.

En la **figura 8** se puede observar las 5 partes modulares, conformadas como componentes individuales, de tal forma que sea “versátil” su configuración y se logre transparencia de ubicación en los servicios del sistema, además de virtualización.



**Figura 8.** Módulos del ESB

Como se observa, de un lado se tienen los componentes modulares que proporcionan servicios proveedores, como lo son las consultas de bases de datos, el e-mail y la oferta móvil.

Del otro lado se tienen los componentes modulares que son los servicios consumidores, como lo son el servidor de mensajería instantánea y el Gateway de mensajería móvil, los cuales proporcionan el medio de acceso a los usuarios para que hagan uso de los servicios proveedores.

### 3.1.2. Metodología para la descripción de la arquitectura

Para facilitar el entendimiento de la arquitectura propuesta se utilizó el modelo definido por Rational Software Corporation (afín a la propuesta conceptual en el estándar IEEE 1471) [40], el cual afirma que la descripción arquitectónica de un sistema puede ser detallada a través de cuatro vistas:

- Vista modular (Process View): Identifica las principales unidades de implementación, junto con sus relaciones más importantes.
- Vista de análisis (Logical View): Determinar las abstracciones esenciales del sistema a desarrollar.
- Vista de diseño (Development View): La organización estática del software en su entorno de desarrollo.
- Vista de implantación (Physical View): Describe cual es la distribución física de los módulos software en componentes hardware.

Debido al refinamiento de la vista de análisis en relación a la vista de diseño, se presenta en éste capítulo solo la vista de diseño. En caso de consulta, la vista de análisis se puede observar en el anexo C.

### 3.2. Vista Modular

El diseño modular de la propuesta de arquitectura para el bróker de mensajería basado en ESB definida en este trabajo de grado, está compuesto por seis módulos (**figura 9**):

- Módulos que facilitan la consecución de los servicios ofrecidos:

**SAM:** Servidor de aplicaciones móviles. Es el servidor que permite la oferta móvil el cual recibe un mensaje y toma la decisión respecto a su contenido.

**SM:** Servidor de e-mail. Es el que permite el uso del correo electrónico, a través de la configuración de los datos que reciba desde el ESB.

**SBD:** Servidor de Bases de Datos. Es el elemento encargado de guardar o consultar los datos de forma ordenada.

- Mediador

**ESB\_M:** Módulo enterprise service bus. El cual constituye el núcleo de la arquitectura y es el encargado de integrar los servicios ofrecidos por los demás módulos para luego ofrecerlos a los diferentes medios de acceso de usuario.

- Módulos que permiten la interacción de los usuarios con los servicios proporcionados:

**SIM:** Servidor de mensajería instantánea. Es el encargado de proveer comunicación instantánea a un cliente móvil o a un laptop/desktop.

**GMM:** Gateway Móvil, servidor de mensajería móvil SMS. Es el elemento que permite la interacción con un controlador SMS para proveer mensajería a clientes móviles.

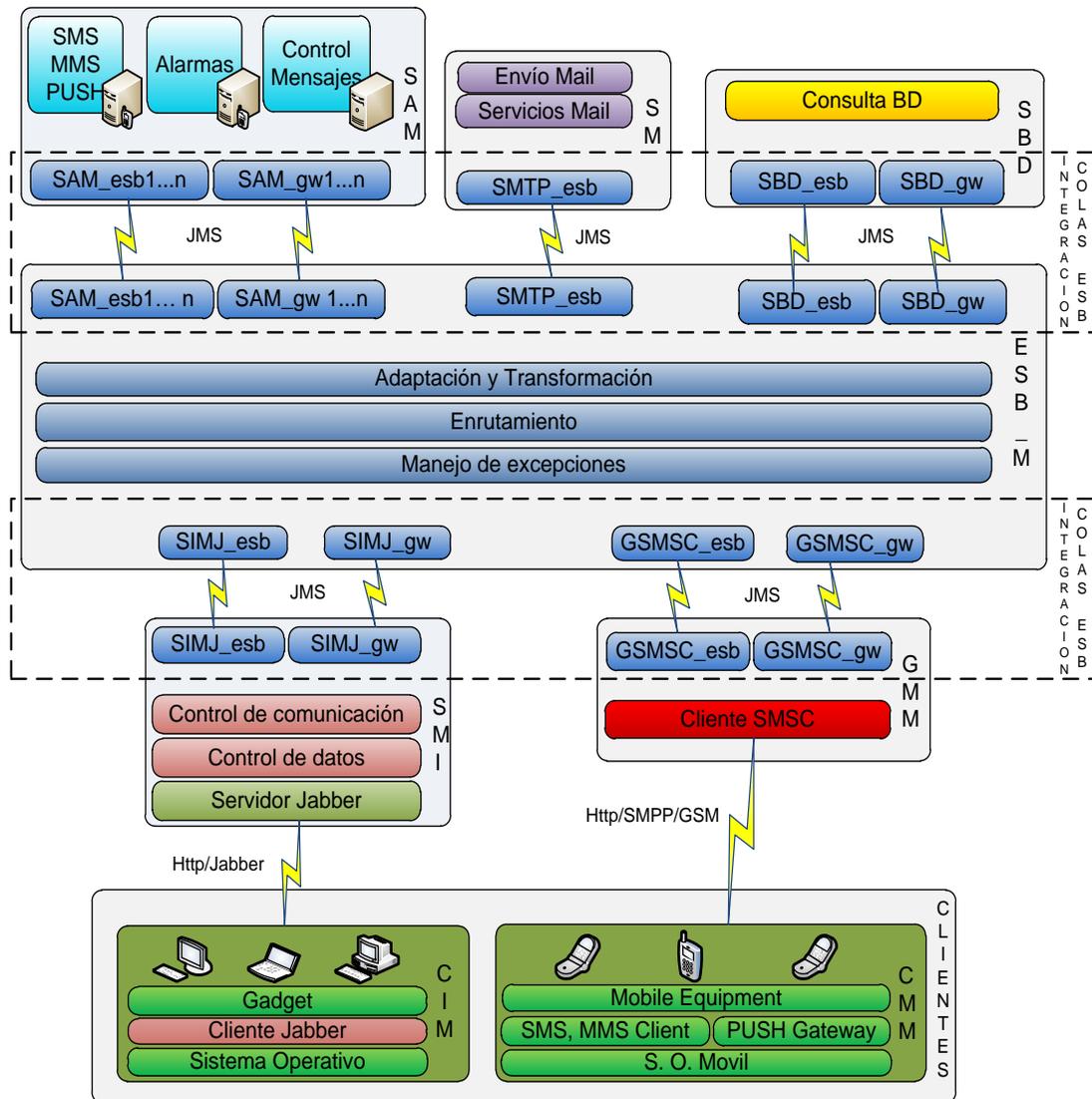


Figura 9. Arquitectura modular de referencia

- Clientes

**CM:** Módulo de Clientes; los dispositivos y sus aplicaciones que le dan medios de acceso a los usuarios.

A continuación se proporciona una descripción de las partes de los módulos, y se complementa con las siguientes vistas de la arquitectura.

### 3.2.1. Servidor de aplicaciones móviles - SAM

**Servicios SMS, PUSH, MMS:** Este servicio posibilita la creación de mensajes de texto, mensajes multimedia y mensajes WAP para brindar mayores posibilidades de comunicación y de interacción a los usuarios. Ver figura 10.

**Servicio de alarmas:** Este servicio da la posibilidad de programar un recordatorio mediante un SMS de cualquier evento que programe el usuario.

**Lógica del mensaje:** Es el encargado de definir la comunicación dependiendo del tipo de mensaje, ya que puede ser SMS, MMS, PUSH, mensajes login, logout.

**Compuertas de enlace ESB SAM:** Son las entidades de escucha que permiten la comunicación de mensajes de servicios entre el SAM y el ESB.



Figura 10. SAM

### 3.2.2. Servidor mail - SM

**Envío mail:** Es el encargado de proporcionar el envío de correo electrónico para los clientes de mensajería móvil e instantánea. Ver figura 11.

**Compuerta de enlace ESB SM:** Son las entidades de escucha que permiten la comunicación de mensajes de servicios entre el SM y el ESB.



Figura 11. SM

### 3.2.3. Servidor de bases de datos - SBD

**Consulta BD:** Este servicio realiza búsqueda de información en la base de datos y devuelve la información si se encuentra allí. Ver figura 12

**Compuerta de enlace ESB SBD:** Es la entidad de escucha que permiten la comunicación de mensajes de servicios entre el SBD y el ESB.



Figura 12. SBD

### 3.2.4. Módulo enterprise service bus - ESB

**Enlace SAM:** Es el servicio que provee comunicación hacia el SAM mediante su compuerta de escucha. Ver **figura 13**.

**Enlace SM:** Es el servicio que provee comunicación hacia el SM mediante su compuerta de escucha.

**SBD\_ESB:** Es el servicio que provee comunicación hacia el SBD mediante su compuerta de escucha.

**Adaptación y transformación:** Es lo que realiza el middleware para satisfacer los requisitos de los servicios, al momento de dirigir mensajes entre servicios.

**Enrutamiento:** Es la entidad encargada de encaminar la respuesta al proveedor de servicio apropiado, sin necesidad de que el servicio cliente conozca la ubicación del proveedor del servicio.

**Enlace SMI:** Es el servicio que provee comunicación hacia el SMI mediante su compuerta de escucha.

**Enlace GM:** Es el servicio que provee comunicación hacia el GMM mediante su compuerta de escucha.

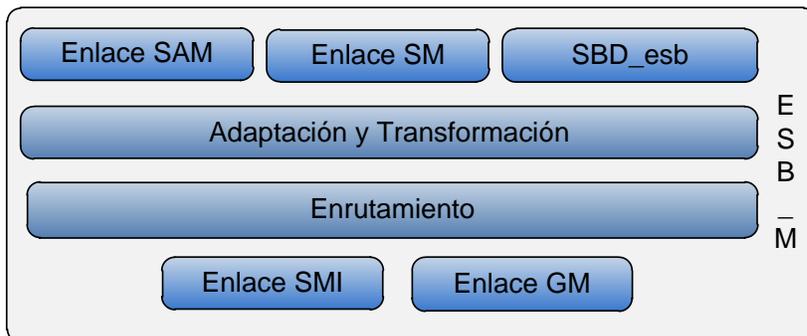


Figura 13. ESB\_M

### 3.2.5. Servidor de mensajería instantánea - SMI

**Compuertas de enlace ESB SMI:** Son las entidades de escucha que permiten la comunicación de mensajes de servicios entre el SMI y el ESB. Ver **figura 14**.

**Adaptación ESB:** Satisface los requisitos de los servicios del SMI.

**Integración XMPP:** Integra otros protocolos de mensajería instantánea.

**Servidor XMPP:** Proporciona los servicios del protocolo XML para el intercambio real de mensajes.

**Control de usuarios:** Realiza el control de los usuarios ingresados en el sistema.

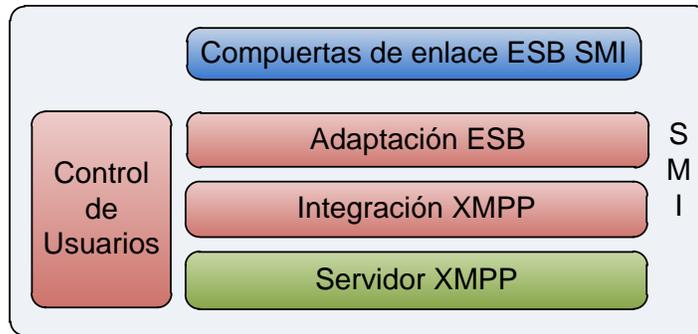


Figura 14. SMI

### 3.2.6. Gateway de mensajería móvil - GMM

**Compuertas de enlace ESB GM:** Son las entidades de escucha que permiten la comunicación de mensajes de servicios entre el GMM y el ESB. Ver **figura 15**.

**Control de adaptación ESB:** Satisface los requisitos de los servicios del GMM.

**Control de conexión gateway:** Realiza la conexión con el controlador de mensajería del sistema móvil celular.

**Cliente SMSC:** Realiza la conexión con los clientes que utilizan telefonía celular.

**Cliente módem:** Realiza la conexión con clientes conectados a través de un módem de telefonía celular.

**Control de mensajes:** Es el servicio encargado de decidir hacia qué tipo de cliente se debe realizar el envío del mensaje.



Figura 15. GMM

### 3.2.7. Módulo de clientes

Este módulo en realidad no es tenido en cuenta para el diseño, ya que son las formas en que los clientes tienen acceso a los servicios proporcionados por la arquitectura, y estas tecnologías dependen de plataformas externas (**figura 16**).

**CIM:** Cliente de mensajería instantánea, es el usuario que puede acceder a los servicios a través de aplicaciones de escritorio.

**CMM:** Cliente de mensajería móvil, es el usuario que puede acceder a los servicios a través de la mensajería de su sistema celular.

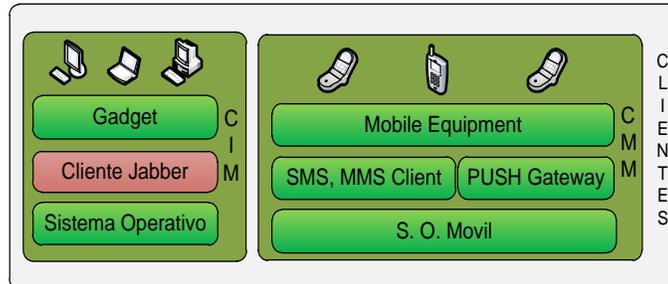


Figura 16. Módulo de clientes

### 3.3. Vista de Diseño

La vista de diseño busca aterrizar el esquema de paquetes que se describe en la etapa de análisis (Anexo C), tomando como referencia un esquema de implementación basado en Java, pero dejando algunos componentes de libre elección. Una implementación de referencia de la arquitectura planteada de un bróker basado en ESB se presenta en el siguiente capítulo.

#### 3.3.1. Servidor de aplicaciones móviles

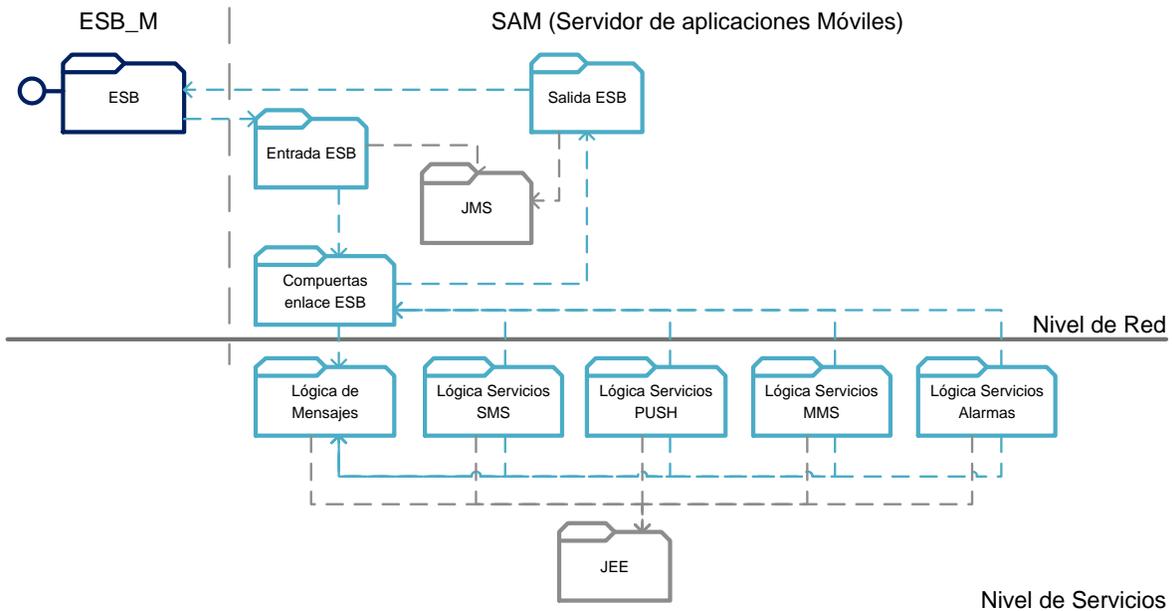
##### Nivel de red

- **Entrada ESB:** Es la compuerta de entrada del ESB basada en JMS. Ver figura 17.
- **Salida ESB:** Es la compuerta que dirige una petición al ESB. Esta compuerta genera peticiones de forma asíncrona y está basada en JMS.
- **JMS:** Es una solución creada para el uso de colas de mensajes, que permite a los componentes de aplicaciones enviar, recibir y leer mensajes.
- **Compuerta enlace ESB:** Es la encargada de administrar todos los procesos que requieran una comunicación con el ESB.

##### Nivel de servicios

- **Lógica de mensajes:** El ESB\_M dirige mensajes provenientes del GMM (solicitudes hechas a través de SMS), en la cual la lógica del servicio analiza el tipo de mensaje según su contenido y resuelve el proceso a seguir. Otro tipo de peticiones es el que llega desde otros servidores para el envío de mensajes a través del GMM. Al analizar su contenido son enviados a los diferentes procesos (SMS, MMS, Push).
- **Lógica de servicios SMS:** Es el encargado de encapsular un mensaje entrante para convertirlo en un SMS.
- **Lógica de servicios PUSH:** Crea las condiciones necesarias para el envío de un WAP-Push. Este tipo de mensajes es muy utilizado para la descarga de contenidos al móvil.

- **Lógica de servicios MMS:** Es el encargado de encapsular un mensaje entrante para convertirlo en un MMS.
- **Lógica de servicios Alarmas:** En este módulo se crean diferentes servicios de monitoreo y alarmas, por ejemplo para enviar mensajes de recordatorio de eventos, enviar mensajes de condición de algún servidor.
- **JEE:** Hace uso de varias de las librerías y APIs que se encuentran en ésta plataforma.



**Figura 17.** Diagrama de diseño SAM

### 3.3.2. Servidor mail

Este módulo es el que permite el envío de correos electrónicos a los diferentes usuarios, previa solicitud del ESB. Solo tiene una compuerta de enlace con el ESB, ya que el no va a hacer peticiones al ESB. La compuerta utiliza el API JMS (Java Message Service) y para el envío de mensajes se utiliza el API Java Mail que se encuentra dentro de la plataforma JEE. La configuración del servidor de mensajería por el cual va a ir el mensaje viene embebida en la petición del ESB. En la **figura 18** muestra el diagrama de diseño de SM.

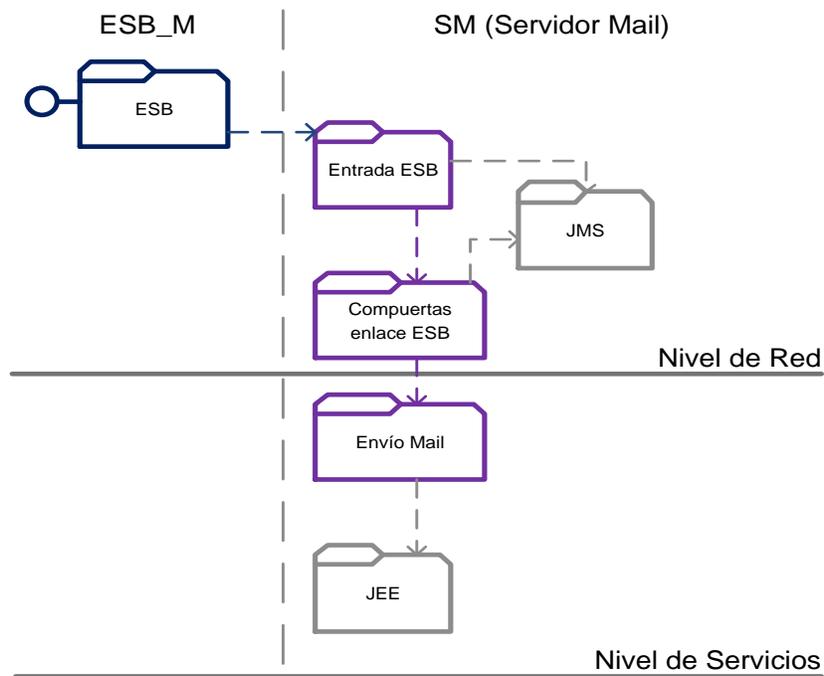


Figura 18. Diagrama de diseño SM

### 3.3.3. Servidor de bases de datos

Es el encargado de suministrar toda la información necesaria para el funcionamiento de todos los módulos. Esta información será entregada previa petición del ESB. Tiene dos compuertas para la comunicación con el ESB, una de entrada que recibe las peticiones y una de salida que envía los datos al ESB de forma asíncrona. Estos enlaces utilizan el API JMS para la comunicación con el ESB. Para obtener los datos se utiliza el pool de conexiones de Java Proxool. Ver **figura 19**.

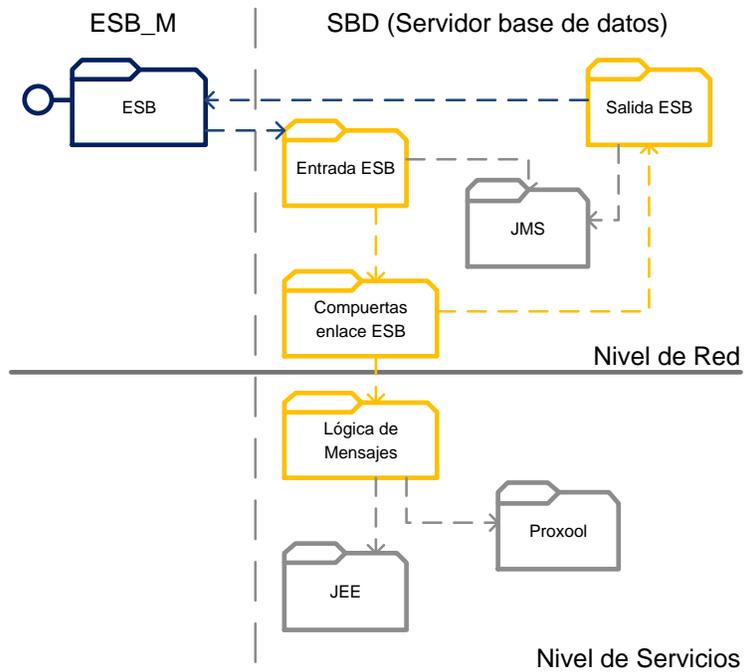


Figura 19. Diagrama de diseño SBD

### 3.3.4. Enterprise Service Bus

Este componente se comporta como un Gateway entre el bus de servicios y todos los servidores añadidos a él. Es el encargado de hacer solicitudes y comunicar la respuesta a los componentes que la requieren. Establece una conexión entre servidor-ESB para consumir servicios necesitados y establece una conexión ESB servidor para notificar de una solicitud del servicio. Éste módulo hace uso de las librerías incluidas en la especificación JBossESB 4.7. La **figura 20** muestra el diagrama de diseño del ESB.

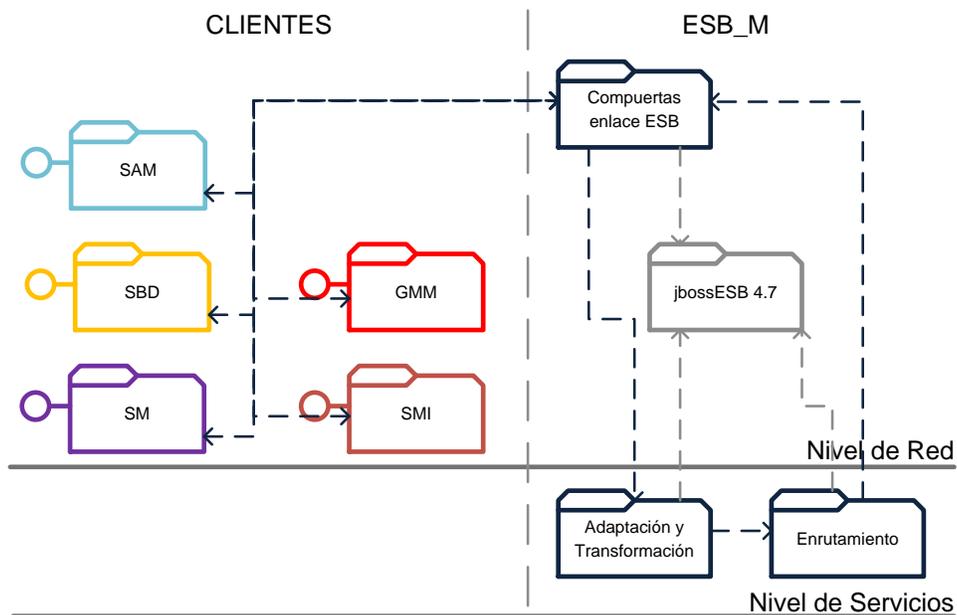


Figura 20. Diagrama de diseño ESB\_M

### 3.3.5. Servidor de Mensajería Instantánea

#### Nivel de aplicación

- **Cliente IM:** Son los clientes que pueden ingresar directamente al servidor mensajería instantánea a través aplicaciones que soporten la comunicación a través del protocolo XMPP. Ver **figura 21**.

#### Nivel de servicios

- **Servidor XMPP:** Es un sistema de mensajería instantánea que utiliza el protocolo XMPP.
- **Integración XMPP:** Resuelve las solicitudes enviadas por el ESB, de usuarios que están por fuera del servidor de mensajería instantánea, y quieren hacer uso de los recursos del servidor.
- **Servicio de autenticación:** Verifica las peticiones de registro y autenticación de usuarios. Si es necesario se hará una solicitud de datos mediante el Control de usuarios.
- **Servicio de presencia:** Toma el consolidado de contactos y los diferentes grupos que tiene el usuario, además de ver el estado en que se encuentra.
- **Envío de mensajes:** Controla el envío de los mensajes hacia usuarios de mensajería instantánea, mensajería móvil o correo electrónico.
- **Dom4j:** Es un framework que permite manipular (leer, crear, editar) archivos XML.
- **XMPP:** Es un protocolo abierto y extensible basado en XML. Con este protocolo se establece una plataforma para el intercambio de datos XML que es utilizada en aplicaciones de mensajería instantánea.
- **Jivesoftware/Openfire:** Son librerías para la creación de servidores de mensajería instantánea.

#### Nivel de datos

- **Control de usuarios:** Realiza solicitudes al ESB para la obtención de datos de un usuario.
- **Proxool:** Es pool de conexiones Java, que utiliza el API JDBC para la conexión a base de datos.

#### Nivel de red

- **Adaptación ESB:** Dependiendo del tipo de solicitud, decide de qué forma va a comunicarse con el ESB, si de forma síncrona o asíncrona. Contiene las compuertas de escucha con el ESB, estas compuertas pueden ser temporales para las comunicaciones síncronas o permanentes para comunicación asíncrona.
- **Entrada ESB:** Es la compuerta de entrada del ESB basada en JMS.
- **Salida ESB:** Es la compuerta que dirige una petición al ESB. Esta compuerta genera peticiones de forma síncrona o asíncrona y está basada en JMS.
- **JMS:** Es una solución creada para el uso de colas de mensajes, que permite a los componentes de aplicaciones enviar, recibir y leer mensajes.

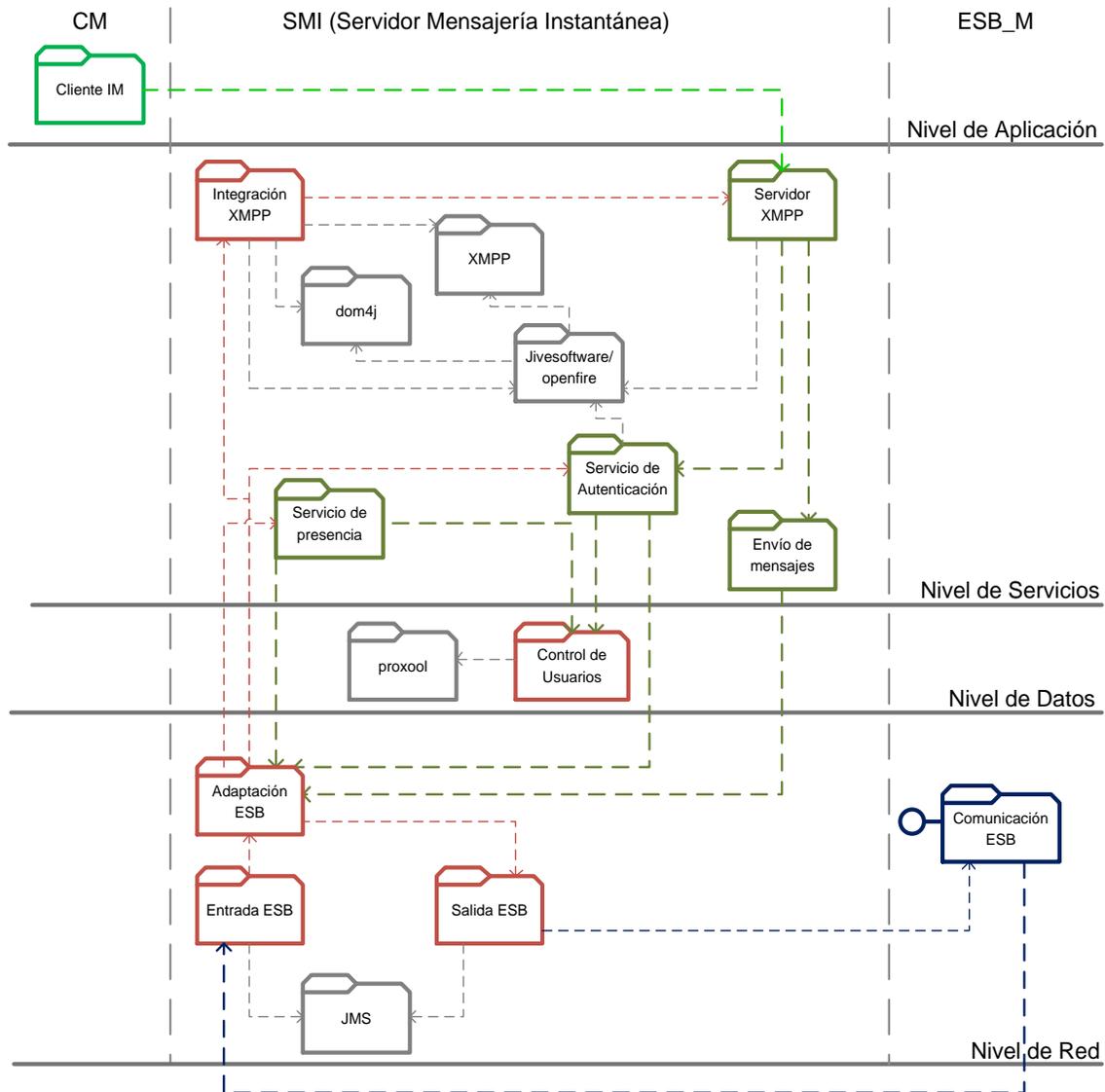


Figura 21. Diagrama de diseño SMI

### 3.3.6. Gateway de Mensajería Móvil

#### Nivel de aplicación

- **Administrador:** Es la aplicación que permite gestionar el Gateway Móvil ya sea para la conexión con el SMSC o para iniciar la escucha del modem GSM. Ver figura 22.

#### Nivel de vista

- **Interfaz de conexión:** Son las interfaces que permiten la administración del Gateway Móvil a través de la aplicación.

### Nivel de servicios

- **Control conexión GW:** Resuelve las solicitudes provenientes de la aplicación de administración.
- **GW modem:** Permite la conexión, desconexión y envío de mensajes a través del modem GSM.
- **GW SMSC:** Permite la conexión, desconexión y envío de mensajes a través del SMSC.
- **Cliente SMS:** Cliente móvil celular que tiene la capacidad de envío de mensajería corta.
- **Iniciar GW:** Activa la escucha del modem GSM.
- **Escuchar modem:** Sirve para escuchar los mensajes entrantes en el modem GSM.
- **Enviar modem:** Sirve para enviar los mensajes SMS en el modem GSM.
- **SMSlib:** Es una librería pequeña basada en Java que permite enviar/recibir mensajes de SMS mediante un modem GSM.
- **JEE Servlet:** Es un objeto que se ejecuta en un servidor o contenedor JEE, especialmente diseñado para ofrecer contenido dinámico desde un servidor web, generalmente HTML.
- **Escuchar SMSC:** Escucha las peticiones entrantes por el SMSC.
- **Enviar SMSC:** Enviar solicitudes de mensajes SMS al SMSC.
- **SMPP:** Es un protocolo estándar de telecomunicaciones pensado para el intercambio de mensajes SMS entre equipos y el SMSC. El protocolo se basa en el intercambio petición/respuesta de pares PDU. El intercambio de datos puede realizarse de manera síncrona, espero cada parte la respuesta/petición del otro o asíncrona donde cada envío y recepción van a través de distintos hilos.

### Nivel de datos

- **Control de mensajes:** Es el encargado de guardar registros para el control de los mensajes SMS enviados y recibidos.
- **Proxool:** Es pool de conexiones Java, que utiliza el API JDBC para la conexión a base de datos.

### Nivel de red

- **Control de adaptación ESB:** Crea la comunicación asíncrona y contiene las compuertas de escucha con el ESB.
- **Entrada ESB:** Es la compuerta de entrada del ESB basada en JMS.
- **Salida ESB:** Es la compuerta que dirige una petición al ESB. Esta compuerta genera peticiones de forma síncrona o asíncrona y está basada en JMS.
- **JMS:** Es una solución creada para el uso de colas de mensajes, que permite a los componentes de aplicaciones enviar, recibir y leer mensajes.

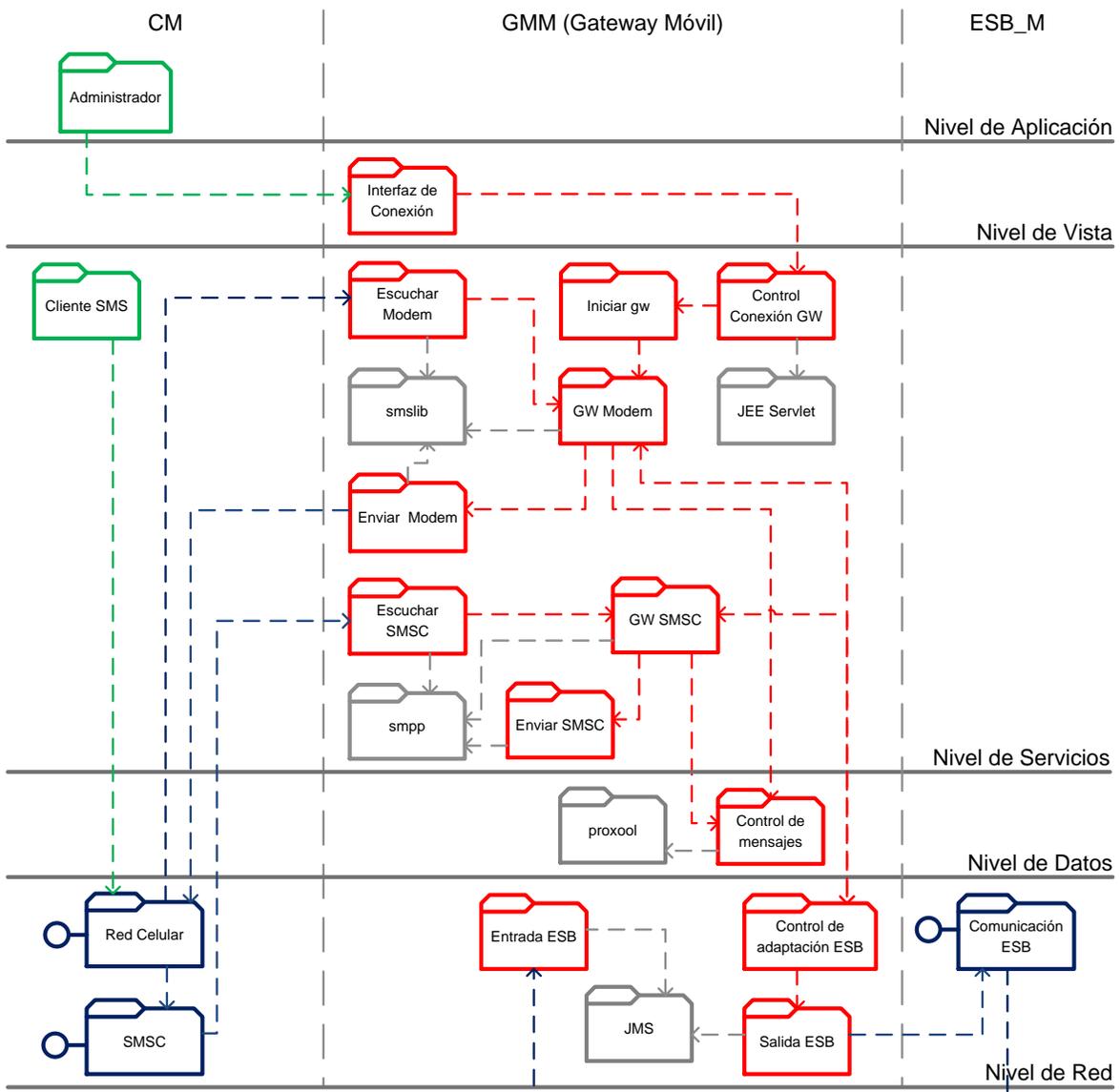


Figura 22. Diagrama de diseño GMM

### 3.4. Vista de Implantación

En la **figura 23** se presenta un diagrama general de implantación definido para la arquitectura de referencia, en el cual se muestra la disposición de los servidores necesarios para la composición de la plataforma general, y determinando como estos se comunican entre sí [39].

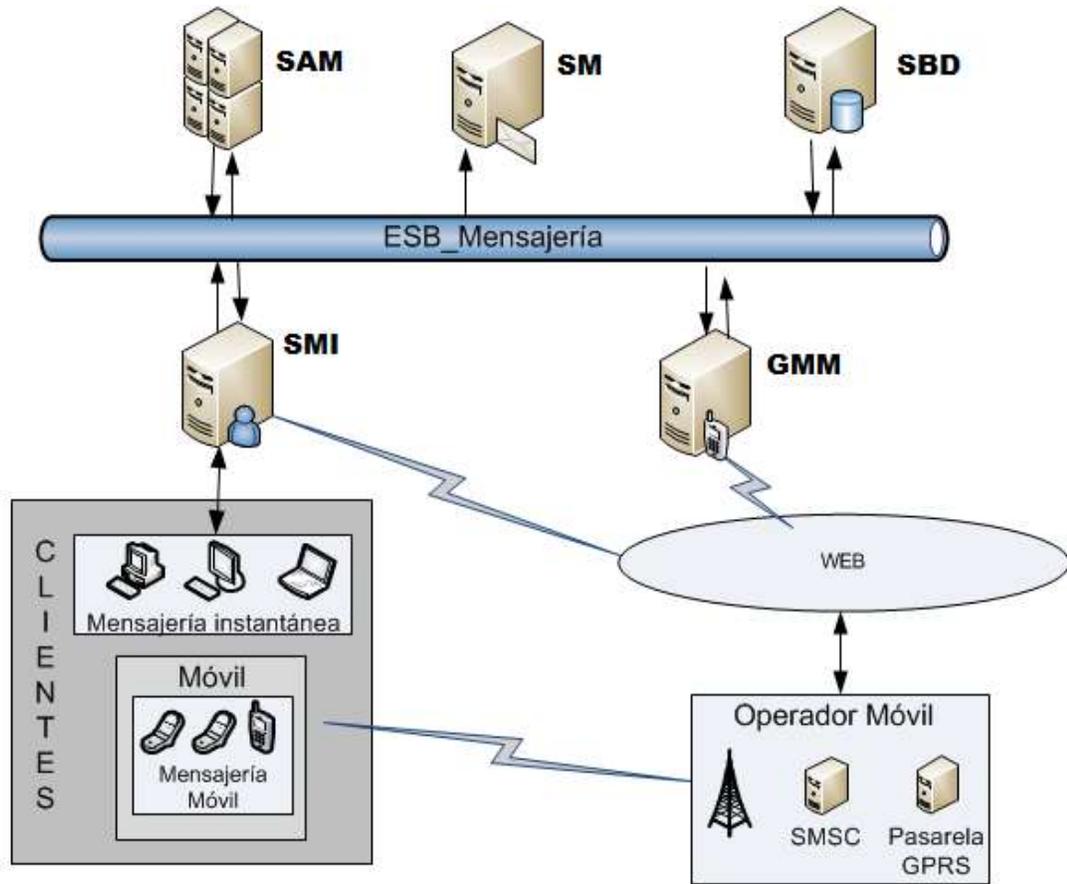


Figura 23. Diagrama general de implantación

## CAPÍTULO 4.

### Implementación del Broker de Mensajería y Evaluación

#### 4.1. Implementación de Referencia

A partir de la arquitectura, se realizó la implementación de las funcionalidades principales, construida con el fin de evaluar la solución planteada, y además servir como orientación a quienes deseen hacer nuevas implementaciones en base a la mensajería móvil e instantánea [44]. A continuación se hace una descripción mediante UML del prototipo [45] y estableciendo partes más detalladas en el anexo D y la documentación digital del anexo [46].

##### 4.1.1. Diagrama de casos de uso



Figura 24. Casos de uso

La **figura 24** muestra el diagrama general de casos de uso para la aplicación.

Los actores que hacen parte de este sistema son:

- **Usuario Móvil:** representa al usuario quien interactúa con los servicios de mensajería en su dispositivo móvil mediante SMS.
- **Usuario Web:** representa al usuario que esta interactuando con el sistema desde su computador personal o una aplicación.

#### 4.1.1.1. Casos de uso iniciados por el usuario móvil

En la **tabla 3**, podemos ver la explicación del caso de uso iniciado por el usuario móvil.

Enviar Mensaje	
<b>Iniciador</b>	Usuario Móvil
<b>Tipo</b>	Primario
<b>Propósito</b>	Hacer uso del sistema
<b>Descripción</b>	El usuario envía un mensaje de texto para hacer uso de la aplicación, ya sea para hacer login, logout, enviar mensaje o saber el estado de sus compañeros.
Procesar SMS recibido	
<b>Tipo</b>	Primario
<b>Propósito</b>	Procesar SMS entrante
<b>Descripción</b>	Este caso de uso es el encargado recibir los SMS enviados por el Gateway móvil y procesarlos para que sean legibles en el SMI.

**Tabla 4.** Casos de uso iniciados por el usuario móvil

#### 4.1.1.2. Casos de uso iniciados por usuario Web

La **tabla 5**, muestra la descripción de los casos de uso iniciados por el usuario Web.

Validar usuario	
<b>Iniciador</b>	Usuario Web
<b>Tipo</b>	Primario
<b>Propósito</b>	Registrar al usuario dentro del servidor de mensajería instantánea
<b>Descripción</b>	El usuario a través del cliente de mensajería instantánea debe validar su Username y Password, para poder hacer uso de los servicios de mensajería instantánea.
Enviar notificación	
<b>Iniciador</b>	Usuario Web
<b>Tipo</b>	Secundario
<b>Propósito</b>	Envía una notificación a uno o a un grupo de usuarios
<b>Descripción</b>	El usuario envía una notificación o mensaje urgente a uno o a un grupo de usuarios. Este mensaje será enviado a los destinatarios de diferentes formas: <ul style="list-style-type: none"> <li>- Si está conectado le llegara el mensaje por intermedio del chat.</li> <li>- Si esta desconectado y tiene habilitado el envío de SMS, le llegara a su móvil.</li> <li>- Si esta desconectado y no tiene habilitado el envío de SMS, le llegará como mensaje de correo electrónico.</li> </ul>
Enviar mensaje instantáneo	
<b>Iniciador</b>	Usuario Web

<b>Tipo</b>	Primario
<b>Propósito</b>	Envío de un mensaje a un contacto
<b>Descripción</b>	El mensaje será enviado al contacto a través del chat o será enviado al móvil del usuario, si ha accedido al sistema por medio de SMS.
<b>Invitar amigos</b>	
<b>Iniciador</b>	Usuario Web
<b>Tipo</b>	Secundario
<b>Propósito</b>	Permitir que un usuario web agregue contactos
<b>Descripción</b>	El usuario puede agregar como amigos a otros usuarios de mensajería instantánea, enviándoles una solicitud de amistad que deberá ser confirmada. Estos amigos serán agregados a un grupo especial de contactos el cual no puede ser editado.
<b>Editar perfil</b>	
<b>Iniciador</b>	Usuario Web
<b>Tipo</b>	Secundario
<b>Propósito</b>	Permitir que un usuario web edite su perfil
<b>Descripción</b>	El usuario puede editar su perfil, con prioridad en el punto en donde se acepta recibir mensajes de notificación vía SMS en su móvil.
<b>Actualizar datos</b>	
<b>Tipo</b>	Primario
<b>Propósito</b>	Actualizar datos de contactos
<b>Descripción</b>	Esta acción se debe realizar la primera vez que inicie una sesión o cada nuevo semestre. Se requiere para actualizar los contactos según las materias matriculadas.
<b>Actualizar password</b>	
<b>Tipo</b>	Secundario
<b>Propósito</b>	Actualizar password de usuario
<b>Descripción</b>	Esta acción se realiza cuando el usuario ha cambiado de contraseña en el portal Web.
<b>Solicitar SMS</b>	
<b>Tipo</b>	Primario
<b>Propósito</b>	Enviar mensaje SMS a los contactos
<b>Descripción</b>	Esta acción se debe realizar cuando un mensaje requiera el envío de un SMS a un contacto, ya sea por una notificación o que el usuario haya accedido al sistema por medio de SMS.
<b>Enviar mail</b>	
<b>Tipo</b>	Primario
<b>Propósito</b>	Envía un correo electrónico a un usuario desconectado
<b>Descripción</b>	Cuando se envía un mensaje de notificación o urgente desde el SMI y se comprueba que el contacto esta desconectado y no permite que se le envíe un SMS, el ESB enruta en mensaje para ser enviado a través de un e-Mail.
<b>Proveer datos usuario</b>	
<b>Tipo</b>	Primario
<b>Propósito</b>	Tomar datos necesarios para la validación de un usuario
<b>Descripción</b>	Esta consulta se debe hacer cuando el usuario se valida por primera vez en el servidor de mensajería instantánea o cuando a través del portal web cambia su contraseña.
<b>Actualizar materias</b>	
<b>Tipo</b>	Primario
<b>Propósito</b>	Suministrar materias y contactos de esa materia a un usuario
<b>Descripción</b>	Esta consulta se hace cuando el usuario se valida por primera vez o cuando ha comenzado un nuevo periodo académico. Suministra las materias matriculadas y las personas matriculadas en estas, que servirán como contactos al usuario.

Actualizar semestre	
<b>Tipo</b>	Secundario
<b>Propósito</b>	Actualiza el periodo académico.
<b>Descripción</b>	Esta consulta se hace temporalmente para saber si se ha cambiado el periodo académico.
Procesar envío SMS	
<b>Tipo</b>	Primario
<b>Propósito</b>	Envía un correo electrónico a un usuario desconectado
<b>Descripción</b>	Genera la lógica para el envío de a través del Gateway móvil.

**Tabla 5.** Casos de uso iniciados por el usuario Web

#### 4.1.1.3. Casos de uso comunes a los 2 actores

La **tabla 6** muestra la descripción de los casos de uso iniciados por los dos actores.

Login	
<b>Iniciador</b>	Usuario Móvil, Usuario Web
<b>Tipo</b>	Primario
<b>Propósito</b>	Identificarse como usuario ante el sistema.
<b>Descripción</b>	El usuario envía un mensaje de texto con la estructura: "login Username Password". De esta forma, el Gateway móvil enviará el mensaje hacia el ESB, para que pueda hacer uso los servicios de mensajería.
Logout	
<b>Iniciador</b>	Usuario Móvil, Usuario Web
<b>Tipo</b>	Primario
<b>Propósito</b>	Salir del sistema.
<b>Descripción</b>	El usuario envía un mensaje de texto con la estructura: "logout Username". De esta forma, el Gateway móvil enviará el mensaje hacia el ESB, para que el usuario sea sacado de los servicios de mensajería.
Enviar mensaje	
<b>Iniciador</b>	Usuario Móvil, Usuario Web
<b>Tipo</b>	Primario
<b>Propósito</b>	Envío de un mensaje a un usuario conectado
<b>Descripción</b>	El Usuario envía un mensaje de texto con la estructura: "logout UsernameDestino mensaje". El mensaje será enviado a un usuario conectado en el servicio de mensajería instantánea.
Ver usuarios conectados	
<b>Iniciador</b>	Usuario Móvil, Usuario Web
<b>Tipo</b>	Primario
<b>Propósito</b>	Solicitar la lista de contactos conectados
<b>Descripción</b>	El usuario envía un mensaje de texto con la palabra: "lista". Para poder recibir la lista de contactos conectados el usuario debe haber previamente accedido al sistema. Este listado se presenta con el UsernameDestino de cada amigo.

**Tabla 6.** Casos de uso iniciados por los dos actores

#### 4.1.2. Diagrama de paquetes general

La **figura 25** nos muestra el diagrama de paquetes general del sistema, en ella se observa los 5 diferentes subsistemas definidos para el desarrollo del proyecto, los cuales son nombrados a continuación:

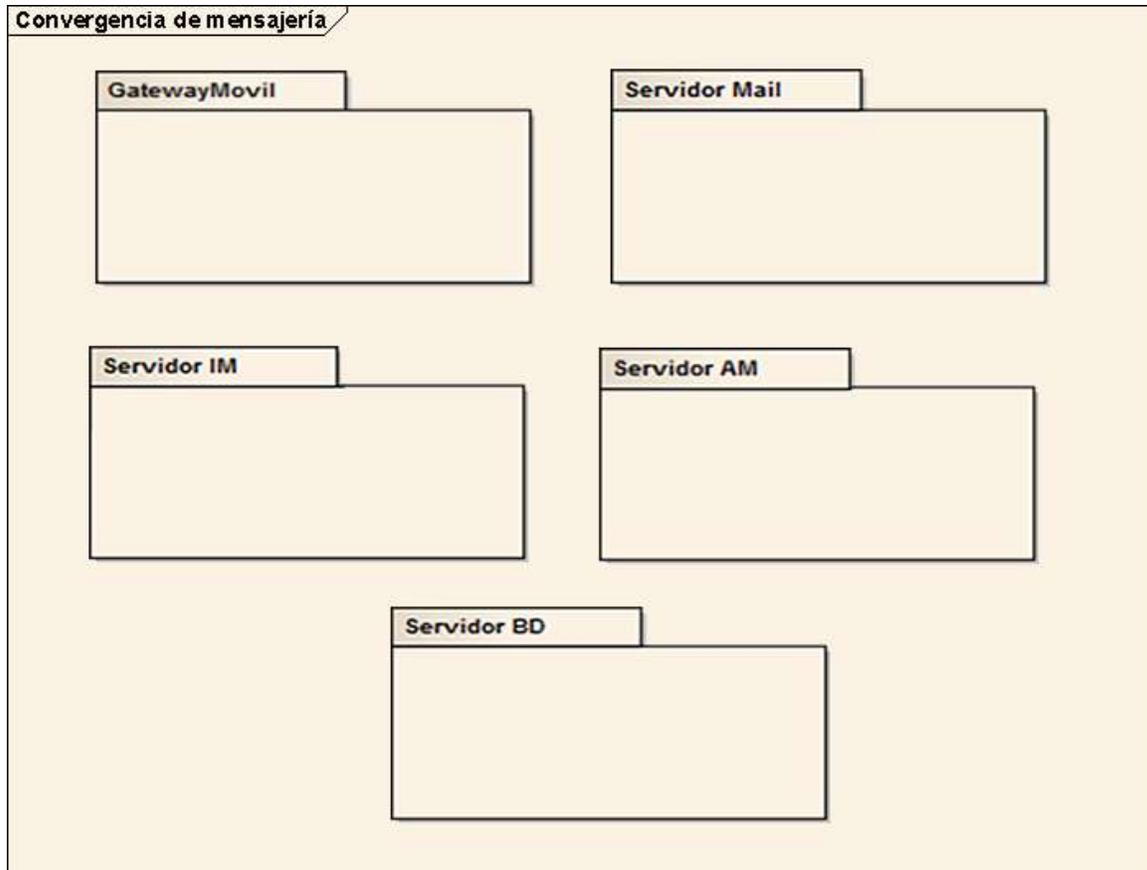


Figura 25. Diagrama de paquetes del sistema

**GatewayMovil:** se encarga de la interacción de mensajería SMS con la red móvil celular.

**Servidor IM:** Se encarga de la gestión de los usuarios que quieran hacer uso de la mensajería instantánea.

**Servidor Mail:** Realiza los procesos para el envío de correos electrónicos.

**Servidor AM:** Procesa los mensajes cortos (SMS) para su envío o recepción.

**Servidor BD:** Es el encargado de gestionar y proveer los datos necesarios para un usuario

Se hará el análisis a cada uno de los subsistemas de la siguiente manera:

- Casos de uso contenidos en el subsistema.
- Diagrama de paquetes de análisis del subsistema.
- Diagrama de paquetes de diseño del subsistema.
- Diagrama de clases del subsistema.

### 4.1.3. Análisis por subsistemas.

#### 4.1.3.1. Gateway de mensajería móvil

##### 4.1.3.1.1. Diagrama de paquetes de análisis del subsistema

Este subsistema tiene relacionado el siguiente caso de uso:

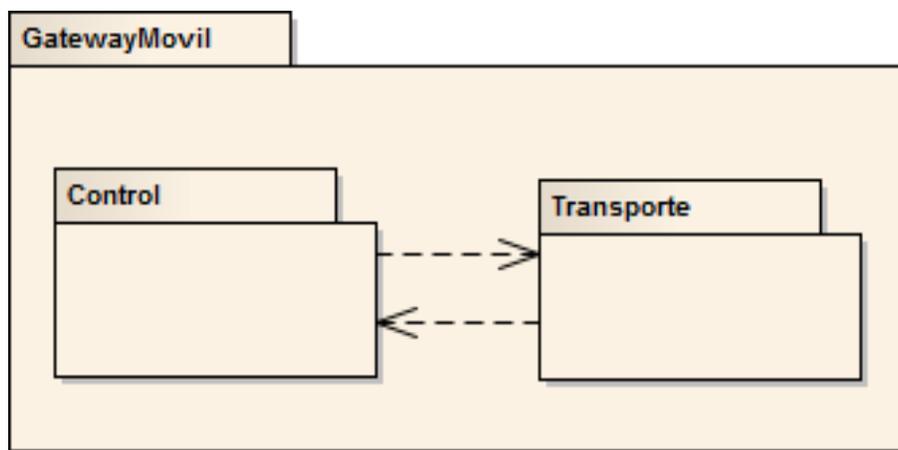
- Enviar SMS.

Los paquetes de análisis para el subsistema que se muestran en la **figura 26** son:

**Control:** Hace referencia al niveles de vista, servicio y datos tratados en el diagrama de paquetes del GMM.

- Controla la llegada de mensajes por el modem GSM o la conexión al SMSC.
- Gestión el envío de mensajes ya sea por el modem GSM o la conexión al SMSC.
- Guarda registro de los mensajes entrantes y salientes del Gateway Móvil.

**Transporte:** Hace referencia a la configuración realizada para poder comunicarse con el ESB, ya sea mensajes entrantes o salientes.



*Figura 26. Diagrama de paquetes de análisis GMM*

##### 4.1.3.1.2. Diagrama de paquetes de diseño del subsistema

La **figura 27** muestra el diagrama de paquetes de diseño, el indica las herramientas y librerías que se utilizaron para la implementación de cada paquete:

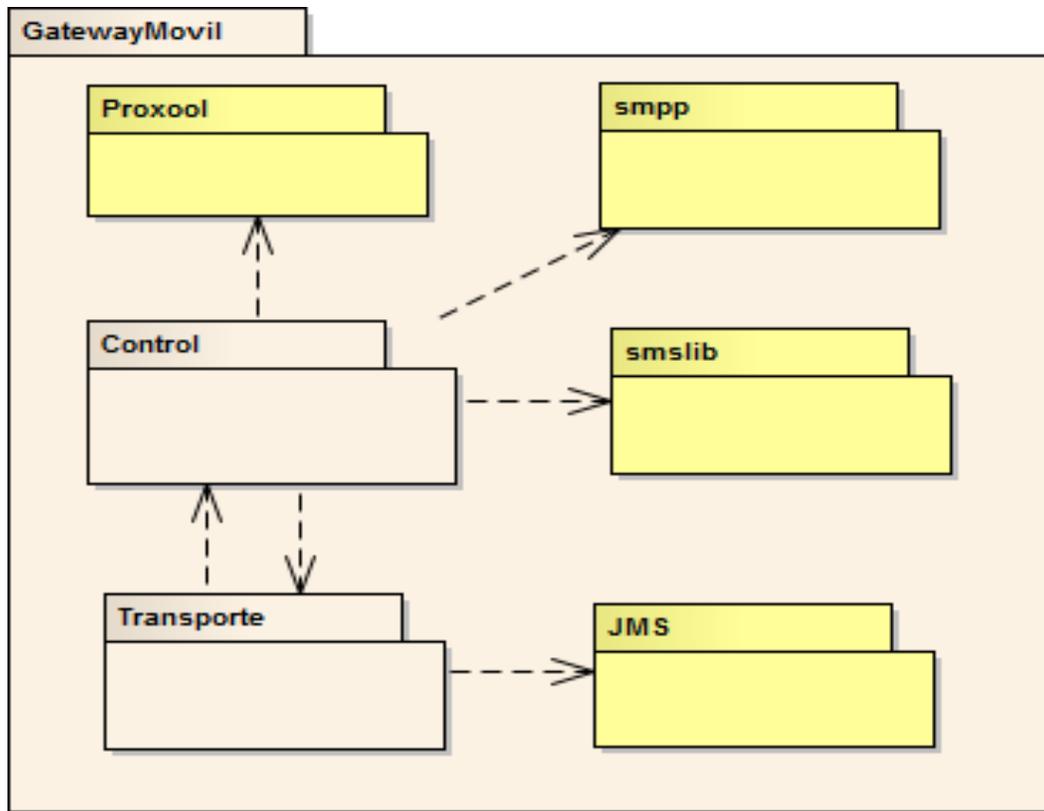


Figura 27. Diagrama de paquetes de diseño GMM

**Control:** Para este paquete se utilizaron las siguientes librerías:

- Proxool: Para la conexión a la base de datos PostgreSQL [47].
- SMPP: Para el intercambio de mensajes SMS entre el servidor y la SMSC.
- SMSLib: Para envío de SMS a través de modem GSM o teléfono móvil.

**Transporte:** JMS es el API utilizado para la comunicación entre el GMM y el ESB.

#### 4.1.3.1.3. Diagrama de clases del subsistema

En la **figura 28** podemos ver el diagrama de clases generado para comunicación por el modem GSM.

La **figura 29** muestra el diagrama de clases para la comunicación con el SMSC.

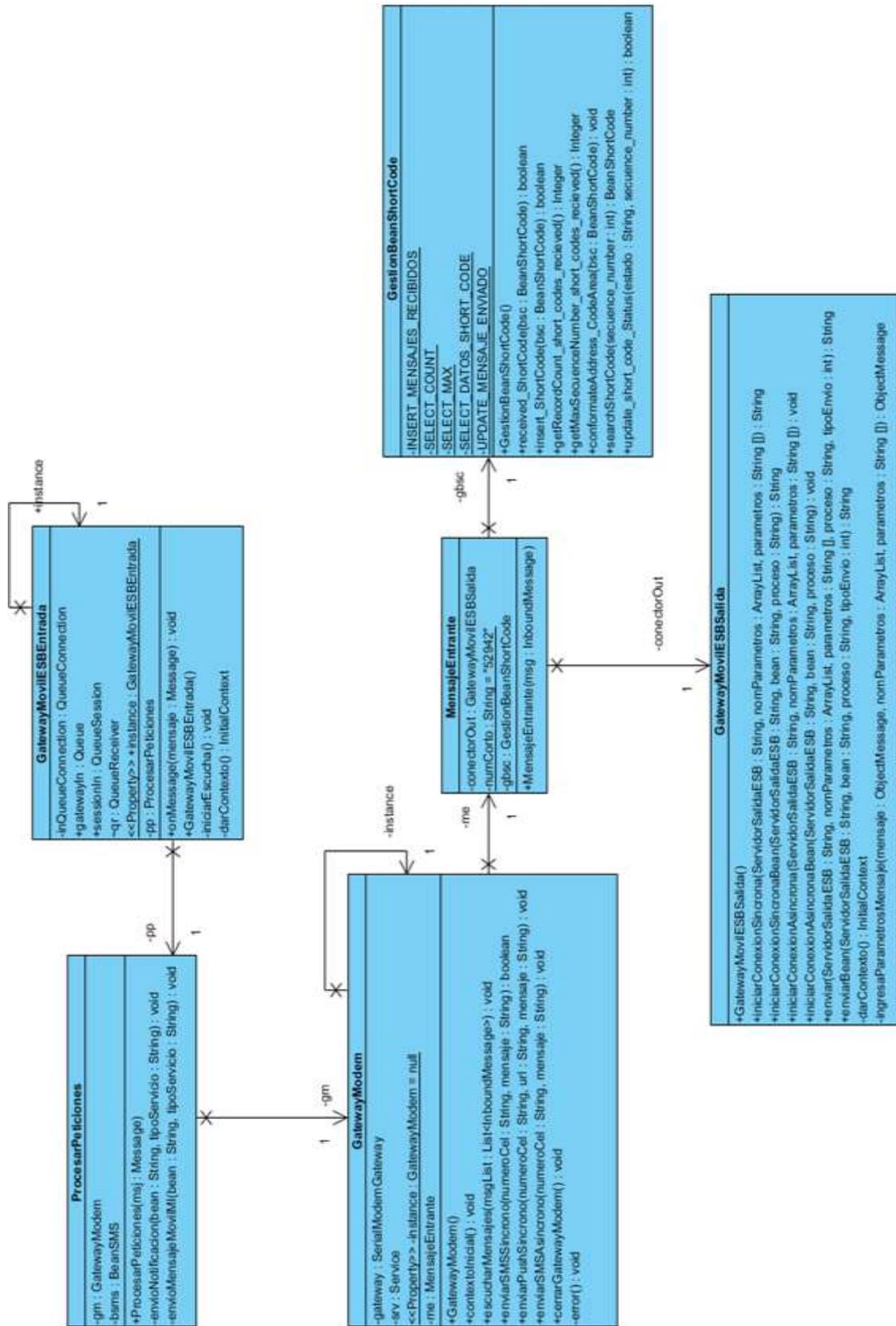


Figura 28. Diagrama de clases módem GSM

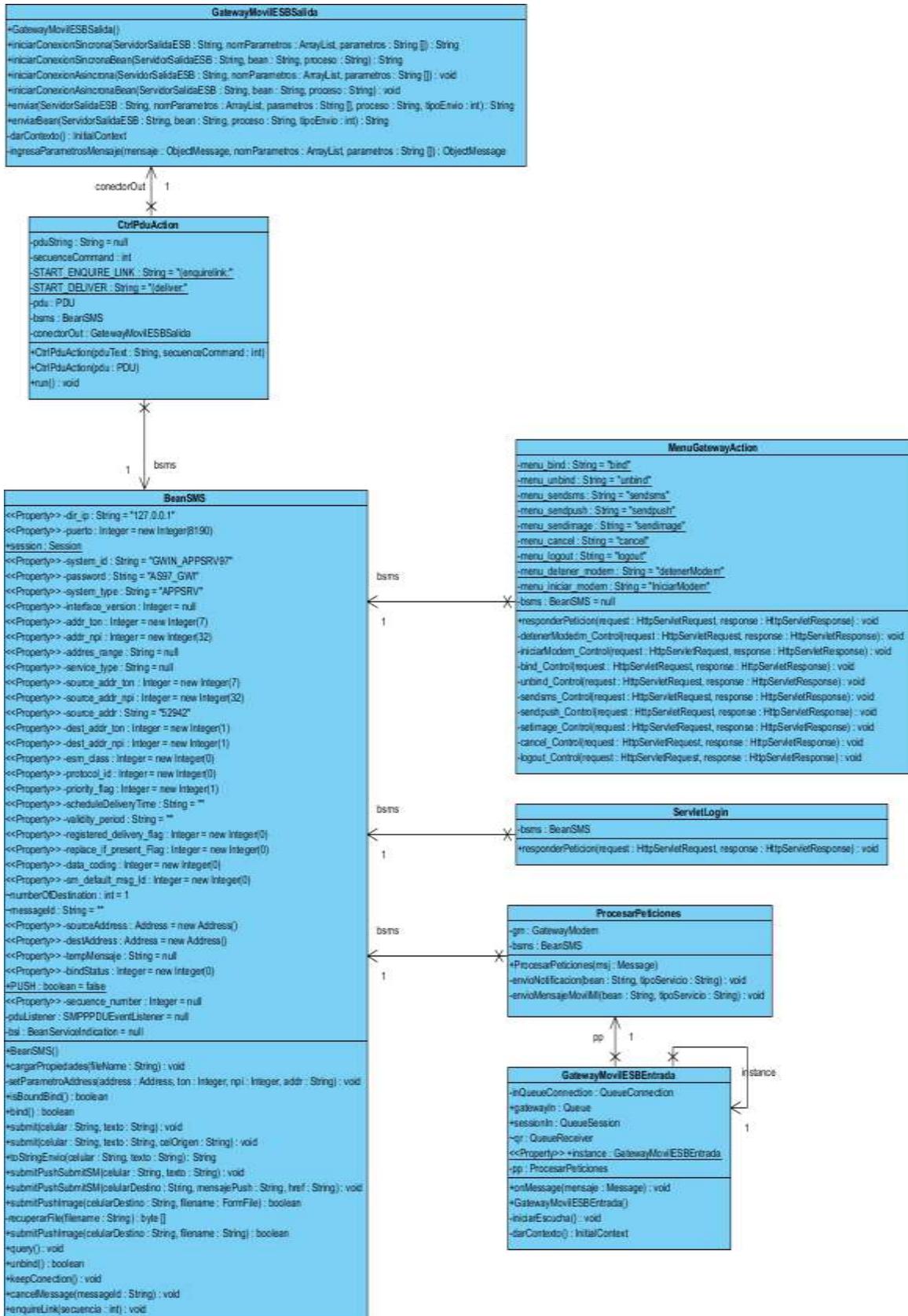


Figura 29. Diagrama de clases SMSC

#### 4.1.3.2. Servidor de mensajería instantánea

##### 4.1.3.2.1. Diagrama de paquetes de análisis del subsistema

El subsistema tiene relacionados los siguientes casos de uso:

- Editar perfil.
- Invitar amigos.
- Enviar notificación.
- Enviar mensaje instantáneo.
- Solicitar SMS
- Login
- Logout
- Enviar mensaje
- Ver usuarios conectados
- Validar usuarios
- Actualizar datos

Los paquetes de análisis para el subsistema son (**Figura 30**):

**Control:** Hace referencia al niveles de servicio y datos tratados en el diagrama de paquetes del SMI.

- Controla el registro de usuarios del servidor de mensajería instantánea.
- Gestiona las comunicación entre contactos, ya sea enviándolas al chat o mediante un SMS a un usuario móvil.
- Guarda los datos de cada usuario.
- Controla la entrada de nuevos mensajes provenientes del ESB.

**Transporte:** hace referencia a la configuración realizada para poder comunicarse con el ESB, ya sea mensajes entrantes o salientes.

**Aplicación:** este paquete contiene a la aplicación de escritorio que permite a un usuario conectarse al servidor de mensajería instantánea, el cual básicamente consiste en un cliente Jabber modificado para hacer uso de todos los servicios de este servidor.

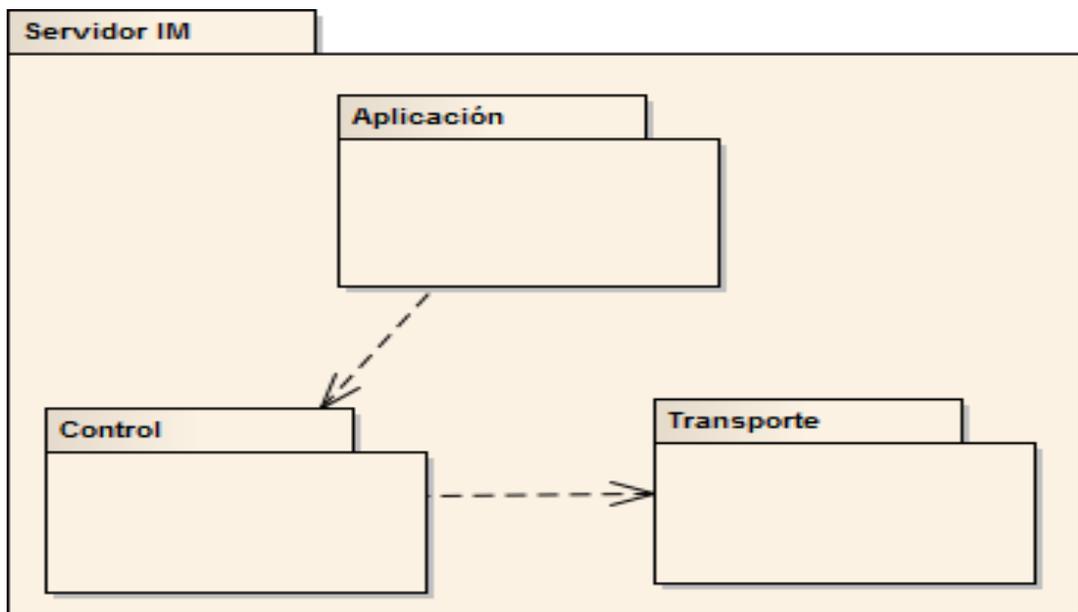
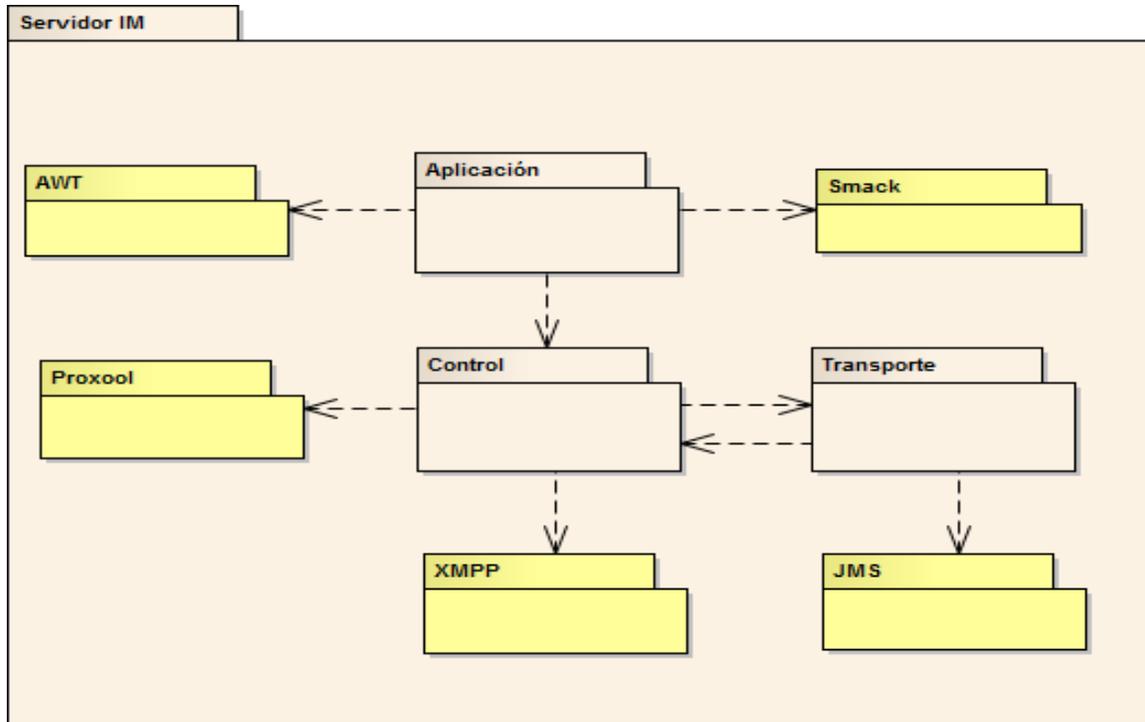


Figura 30. Diagrama de paquetes de análisis SMI

#### 4.1.3.2.2. Diagrama de paquetes de diseño del subsistema

La **figura 31** muestra el diagrama de paquetes de diseño, el indica las herramientas y librerías que se utilizaron para la implementación de cada paquete:



**Figura 31.** Diagrama de paquetes de diseño SMI

**Control:** Para este paquete se utilizaron las siguientes librerías:

- Proxool: Para la conexión a la base de datos del servidor de mensajería instantánea la cual está hecha en PostgreSQL.
- XMPP: Protocolo extensible, abierto basado en XML para el intercambio en tiempo real de mensajes y presencia entre dos puntos en Internet.

**Transporte:** JMS es el Api utilizado para la comunicación entre el SMI y el ESB.

**Aplicación:**

- AWT: Librería para la implementación de las interfaces gráficas.
- Smack: Librería para crear cliente de mensajería Instantánea basados en XMPP.

#### 4.1.3.2.3. Diagrama de clases del subsistema

La **figura 32** muestra el diagrama de clases generado para el Servidor de mensajería instantánea SMI. Para mayor claridad, la figura se puede ver en los anexos digitales.



### 4.1.3.3 Servidor mail

#### 4.1.3.3.1. Diagrama de paquetes de análisis del subsistema

El subsistema tiene relacionado el siguiente caso de uso:

- Enviar Mail.

Los paquetes de análisis para el subsistema son (**Figura 33**):

**Transporte:** hace referencia a la configuración realizada para poder comunicarse con el ESB, ya sea mensajes entrantes o salientes y a la configuración enviada por el ESB para el envío de correos electrónicos.

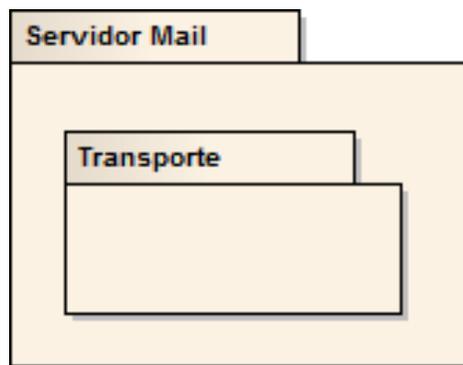


Figura 33. Diagrama de paquetes de análisis SM

#### 4.1.3.3.2. Diagrama de paquetes de diseño del subsistema

La **figura 34** muestra el diagrama de paquetes de diseño, el indica las herramientas y librerías que se utilizaron para la implementación de cada paquete:

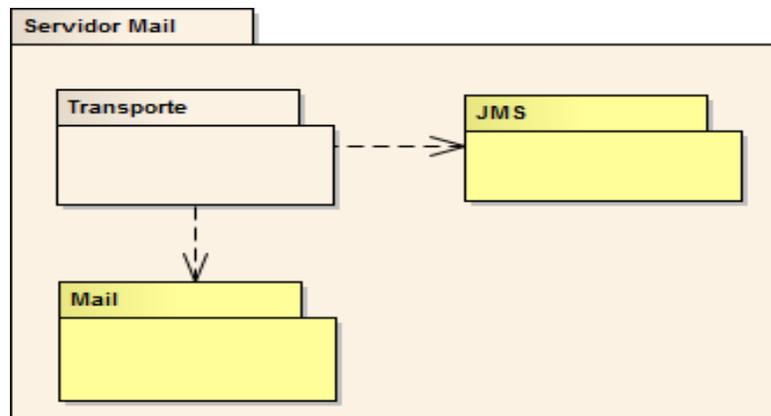


Figura 34. Diagrama de paquetes de diseño SM

**Transporte:**

- JMS es el Api utilizado para la comunicación entre el SM y el ESB.
- La librería Mail, utilizada para el envío de correos electrónicos incluida en J2EE [48].

#### 4.1.3.3.3. Diagrama de clases del subsistema

La **figura 35** muestra el diagrama de clases generado para el Servidor mail SM.

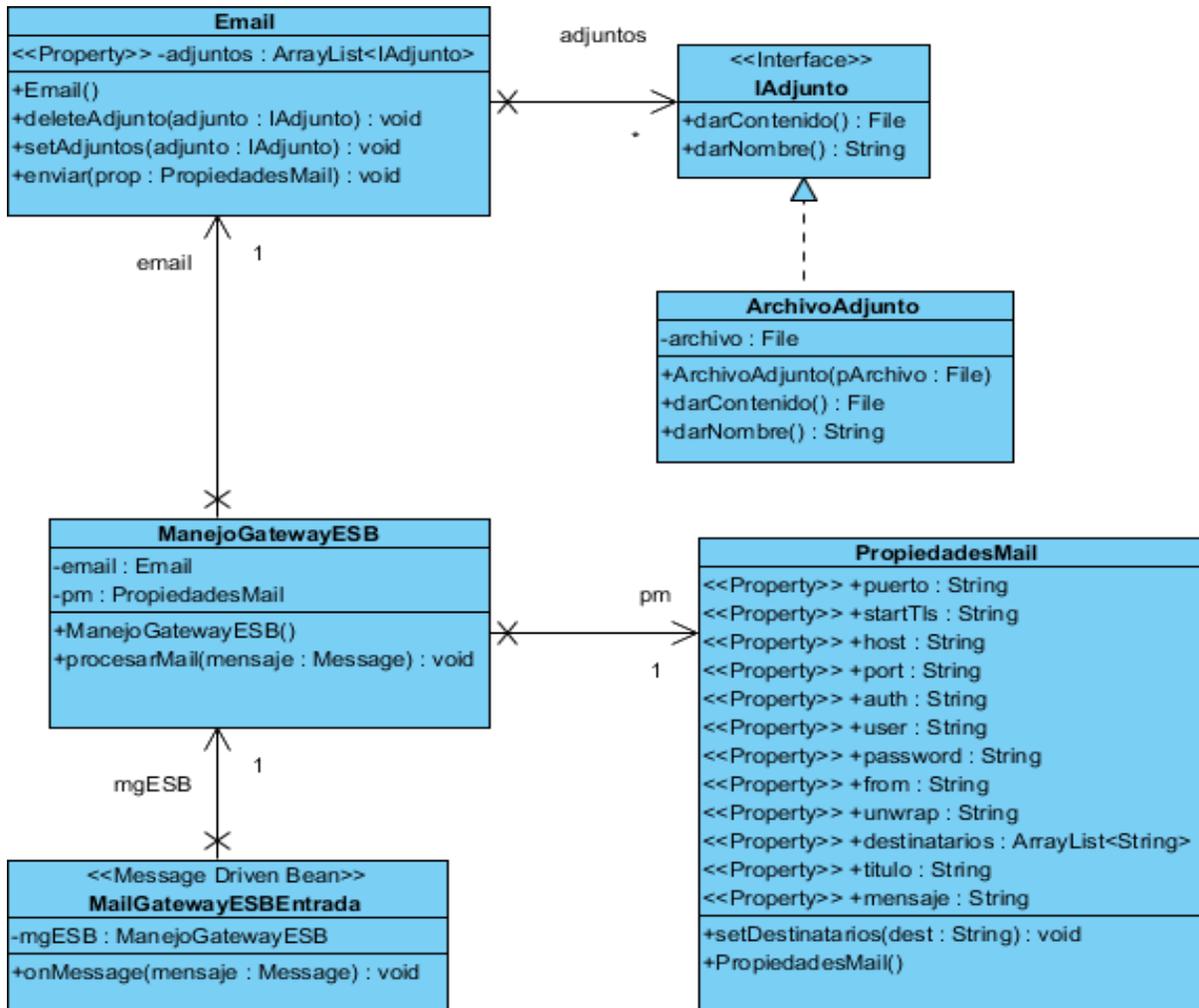


Figura 35. Diagrama de clases SM

#### 4.1.3.4. Servidor de aplicaciones móviles

##### 4.1.3.4.1. Diagrama de paquetes de análisis del subsistema

El subsistema tiene relacionado los siguientes casos de uso:

- Procesar envío SMS.
- Procesar SMS recibido.

Los paquetes de análisis para el subsistema son (**Figura 36**):

**Control:** Hace referencia al niveles de servicio en el diagrama de paquetes del SAM nombrado en el anterior capítulo.

- Genera la lógica del mensaje móvil.

- Crea parámetros de configuración para el envío de mensajes.
- Gestiona los mensajes entrantes para que sean legibles en el SMI.

**Transporte:** hace referencia a la configuración realizada para poder comunicarse con el ESB, ya sea mensajes entrantes o salientes.

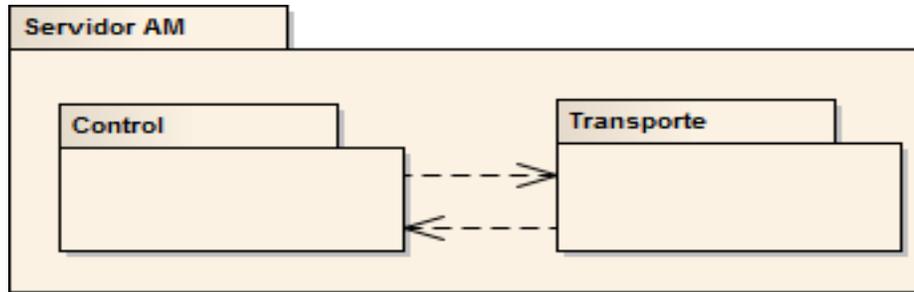


Figura 36. Diagrama de paquetes de análisis SAM

#### 4.1.3.4.2. Diagrama de paquetes de diseño del subsistema

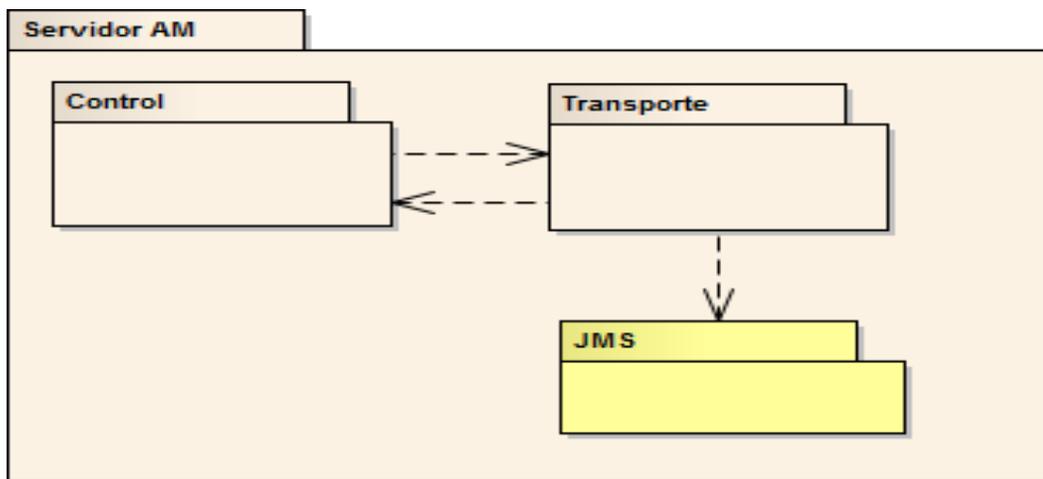


Figura 37. Diagrama de paquetes de diseño SAM

Las librerías utilizadas para este subsistema se muestran en la **figura 37**, estas son:

**Control:** Utiliza las librerías estándar de J2EE.

**Transporte:** JMS es el API utilizado para la comunicación entre SAM y el ESB.

#### 4.1.3.4.3. Diagrama de clases del subsistema

La **figura 38** muestra el diagrama de clases generado para el Servidor de Aplicaciones Móviles SAM.

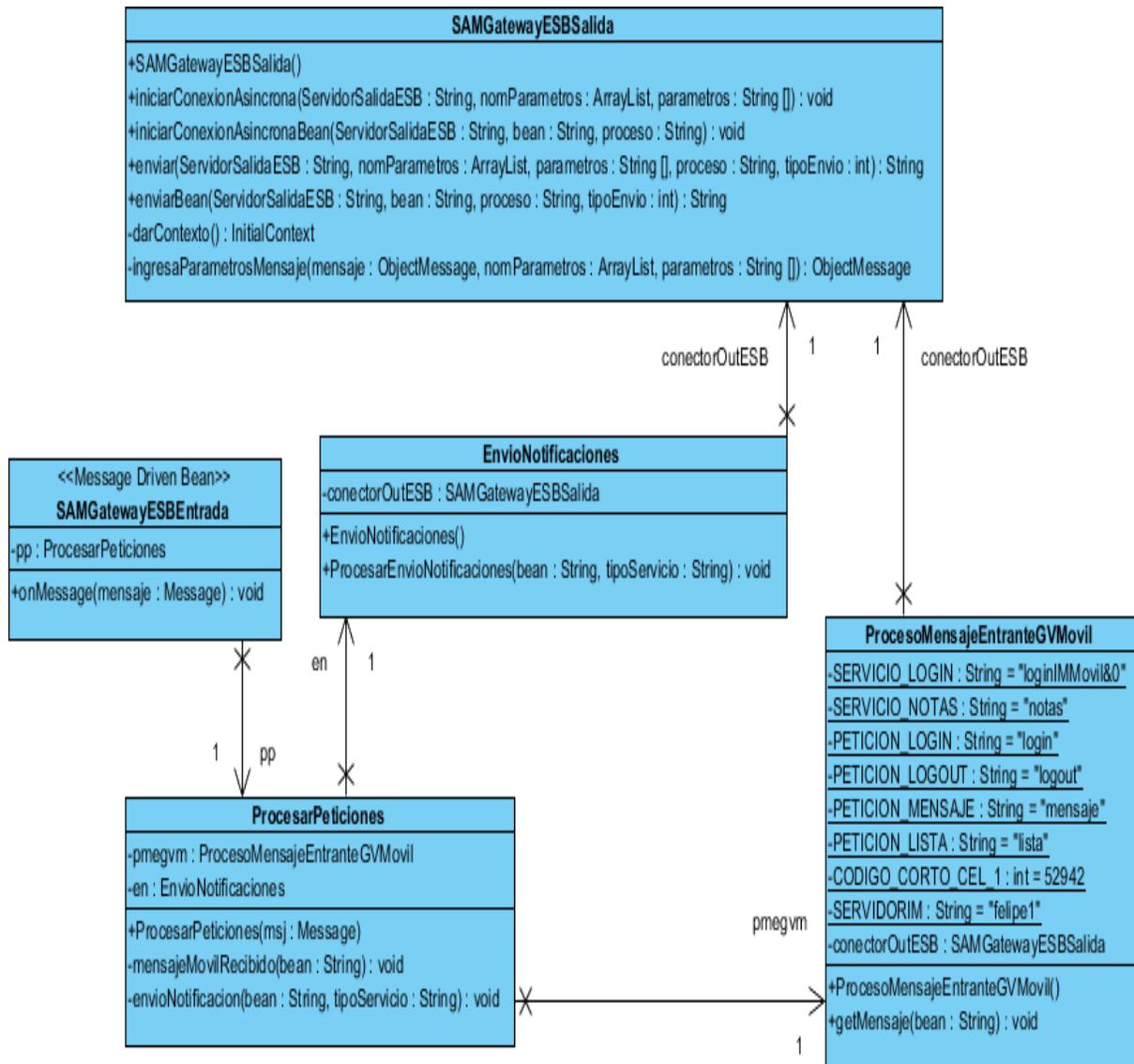


Figura 38. Diagrama de clases SAM

#### 4.1.3.5. Servidor de base de datos

##### 4.1.3.5.1. Diagrama de paquetes de análisis del subsistema

Los casos de uso relacionados con este subsistema son:

- Actualizar password
- Proveer datos usuario
- Actualizar semestre
- Actualizar materias

Los paquetes de análisis para el subsistema son (Figura 39):

**Control:** Hace referencia al nivel de servicio en el diagrama de paquetes de SBD nombrado en el anterior capítulo.

- Suministra datos necesarios para la validación de un usuario.
- Suministra la información de materias y contactos para un usuario del servidor de mensajería instantánea.
- Gestiona los periodos académicos para cambio de datos.

**Transporte:** hace referencia a la configuración realizada para poder comunicarse con el ESB, ya sea mensajes entrantes o salientes, que contienen toda la información de un usuario.

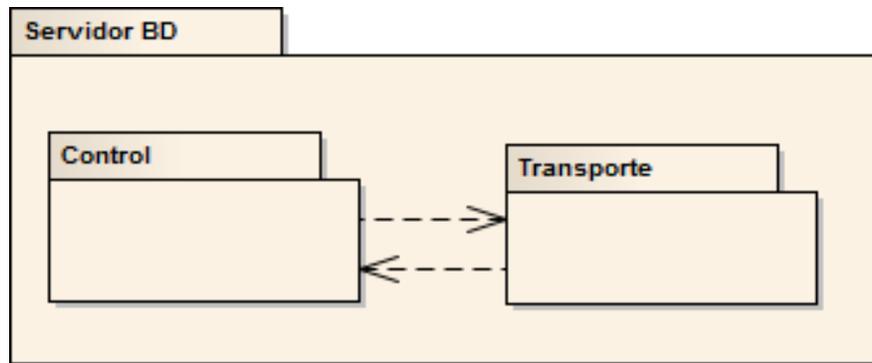


Figura 39. Diagrama de paquetes de análisis SBD

#### 4.1.3.5.2. Diagrama de paquetes de diseño del subsistema

Las librerías utilizadas para este subsistema se muestran en la **figura 40**, estas son

**Control:** Para este paquete se utilizaron las siguientes librerías:

- Proxool: Para la conexión a la base de datos del SBD.

**Transporte:** JMS es el Api utilizado para la comunicación entre el SBD y el ESB.

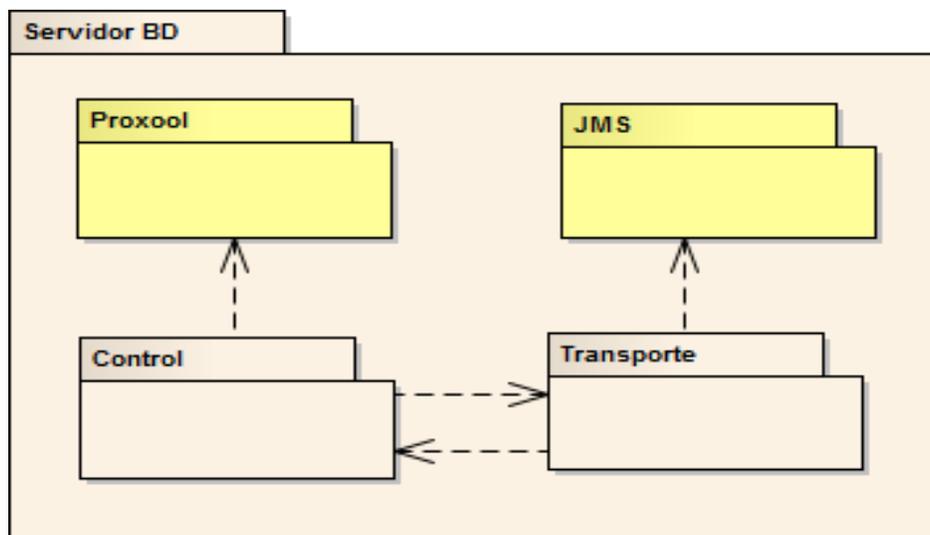


Figura 40. Diagrama de paquetes de diseño SBD

#### 4.1.3.5.3. Diagrama de clases del subsistema

La **figura 41**, muestra el diagrama de clases del servidor de mensajería instantánea.

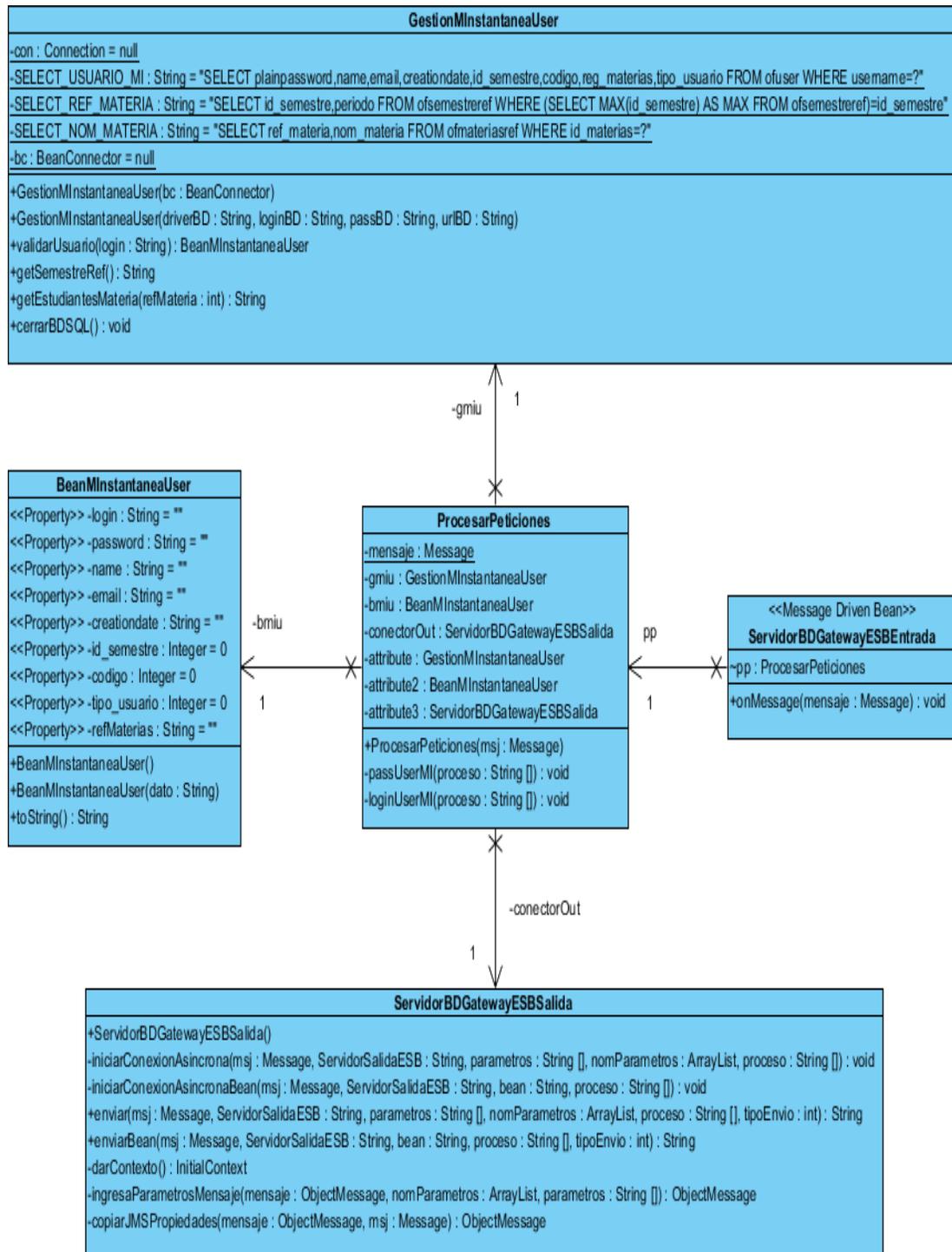
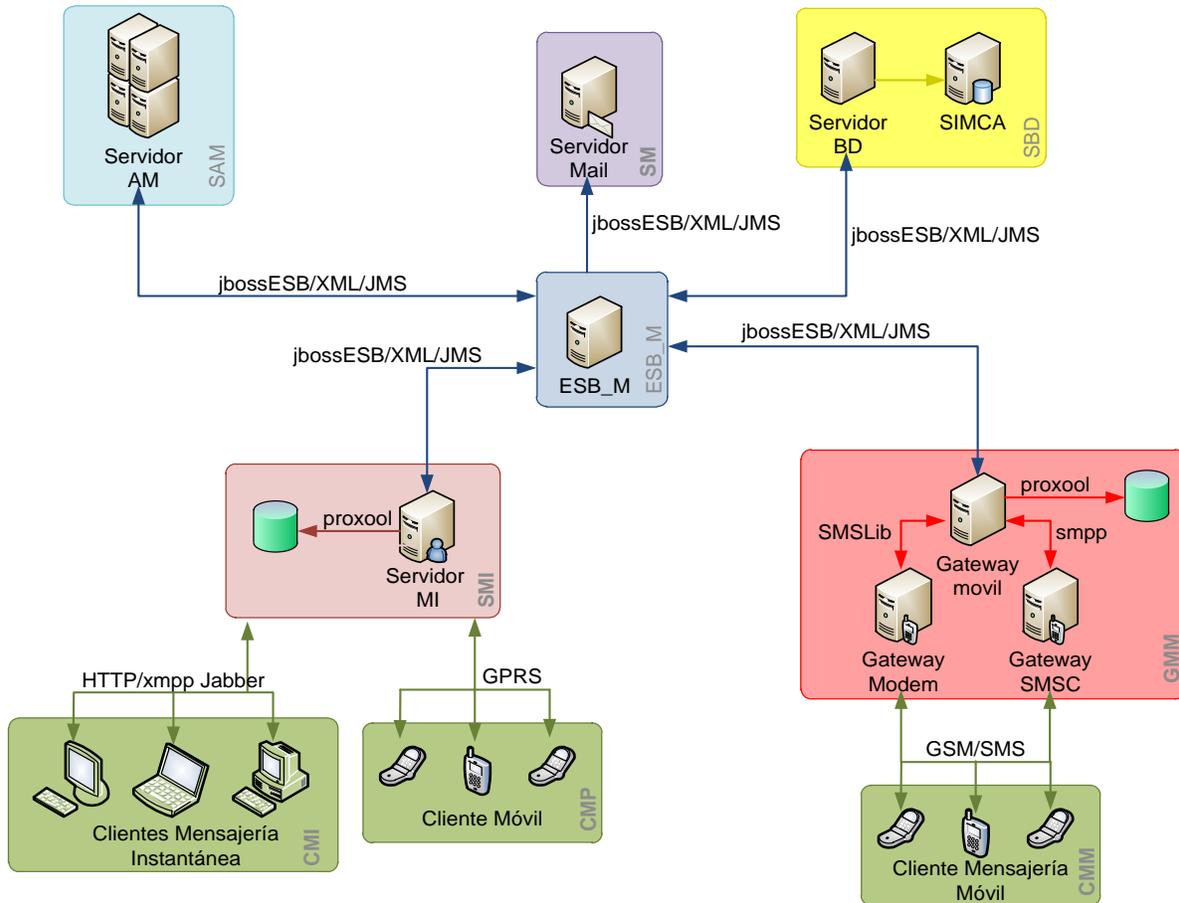


Figura 41. Diagrama de clases SBD

## 4.2. Arquitectura de la Implementación

En la **figura 42** se representa el modelo de implantación de referencia para el sistema:



**Figura 42.** Arquitectura de implementación

### 4.2.1. Interacción entre componentes

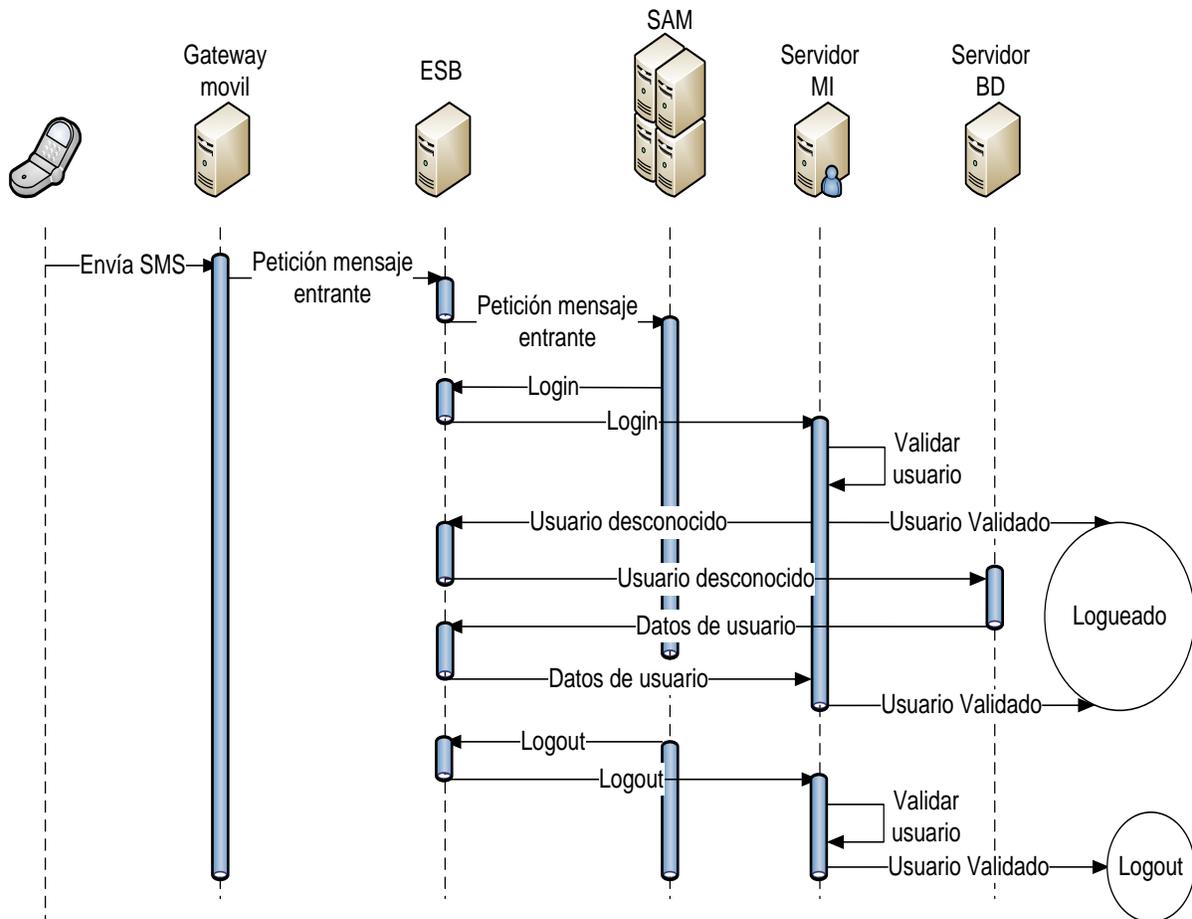
En este proyecto se desea crear un conglomerado de servicios de mensajería para establecer una comunidad virtual académica dentro de la Universidad del Cauca. Para ello se realiza la unión de la mensajería instantánea ordinaria, con los servicios que se pueden acceder mediante un teléfono móvil [49].

Por lo tanto un usuario con cuenta de correo activa en la Universidad del Cauca podría comunicarse de forma interactiva con otro usuario que también posea una cuenta a través de una aplicación de mensajería para PC [50]. Bajo este nuevo servicio, los usuarios de un operador de telefonía celular tendrían la posibilidad de conectarse a través de sus móviles y chatear a través de mensajes de texto SMS o una aplicación móvil, como si estuvieran frente a un PC.

Se mostrará a continuación los diagramas de secuencia que describen los servicios que se implementan para la posterior realización de pruebas en este proyecto.

#### 4.2.1.1. Usuario a través de SMS

Un usuario desde cualquier móvil puede conectarse al Servidor MI, haciendo uso de los mensajes de texto. La **figura 43** muestra un diagrama de secuencia entre los servidores, para crear el login y el logout de dicho usuario.



**Figura 43.** Diagrama de secuencia Login y Logout

Secuencia de la interacción:

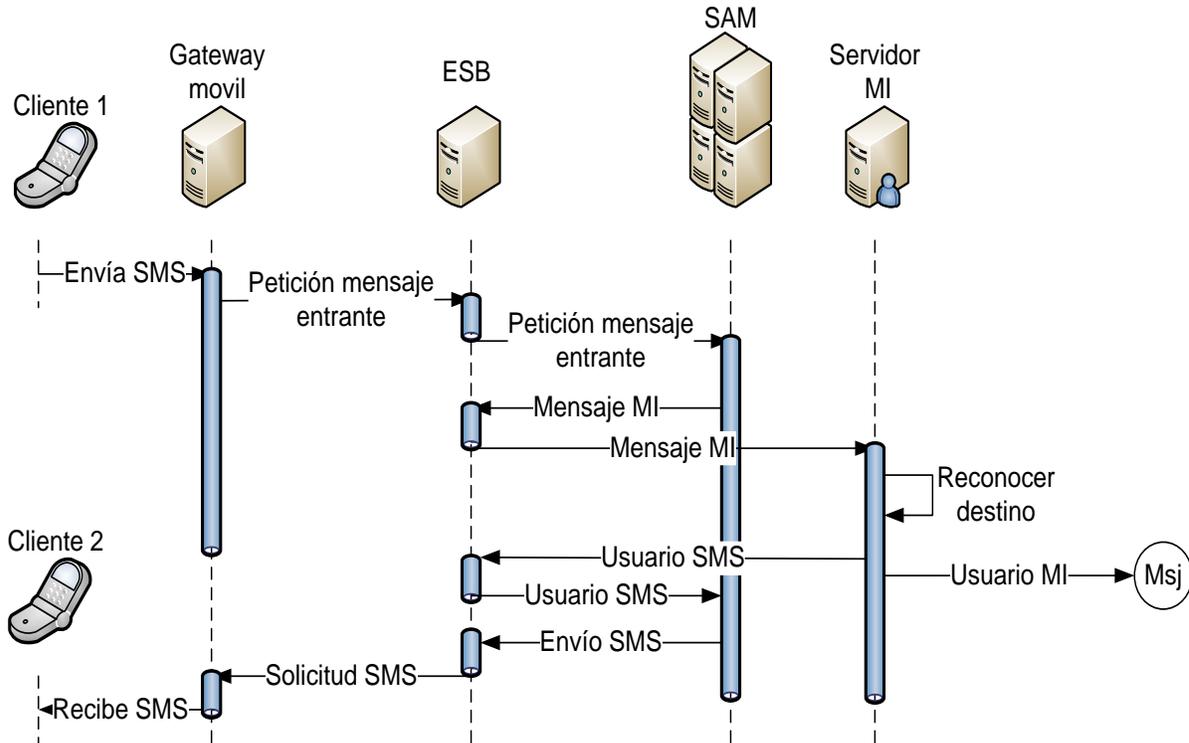
1. El usuario envía un SMS de la siguiente manera:
  - **Login:** envía la palabra login seguida por el userName y el password, separadas con un espacio entre palabras (ejemplo: login ana ana).
  - **Logout:** envía la palabra logout seguida por el userName, separadas con un espacio entre palabras (ejemplo: logout ana).
2. El GMM recibe el mensaje y hace la petición de mensaje entrante al ESB.
  - **Tipo de Servicio:** *mensajeRecibido*, **Estado:** *0. mensajeRecibido&0*.

- Mensaje asíncrono.
  - **Cuerpo del mensaje:**  
 codigoCorto,texto,numCelUsuario,fechaRecibido,estado,numSecuencia,idCodCorto,códigoÁrea,"final"
    - codigoCorto: Es el número al que se está enviando el mensaje.
    - texto: El texto del mensaje enviado por el usuario.
    - numCelUsuario: Es el número celular desde donde se envió el mensaje.
    - fechaRecibido: Fecha en que se recibió el mensaje
    - estado: es el estado en que se encuentra el mensaje (recibido, respondido, etc.)
    - numSecuencia: Numero de registro único que tiene un SMS entrante.
    - códigoÁrea: código de área de un país en general.
    - "final": indica el final del cuerpo del mensaje.
3. El ESB recibe el mensaje y lo enruta hacia el Servidor de Aplicaciones Móviles.
  4. SAM, en este caso sirve como puente de datos, ya que los servicios creados para responder al SMS, van dirigidos al SMI.
  5. El ESB recibe los datos y los enruta hacia el SMI.
  6. SMI recibe los datos.
    - **Logout:** Verifica que el usuario este registrado previamente, si es así realiza el logout indicado. Finaliza la secuencia para logout.
    - **Login:** Valida el usuario.
      - Si el usuario esta actualizado en la base de datos del SMI, y los datos concuerdan, hace el registro e ingresa al usuario al chat. Finaliza la secuencia para login.
      - En caso de no encontrar un registro o esta que el usuario este desactualizado, por ejemplo en el semestre, se hace una solicitud al ESB (Paso 7).
  7. Solicitud de datos nuevo usuario o usuario no actualizado.
    - **Tipo de Servicio:** *loginUserMI*, **Estado:** 0. *loginUserMI&0*.
    - Mensaje síncrono.
    - **Cuerpo del mensaje:** userName,"fin".
      - userName: nombre del usuario dentro del servidor IM.
      - "fin": indica el final del cuerpo del mensaje.
  8. El ESB recibe el mensaje, por el tipo de servicio y el estado en que se encuentra, lo enruta hacia el SBD.
  9. El SBD recopila los datos del usuario dependiendo del Username y da una respuesta al ESB.
    - **Tipo de Servicio:** *loginUserMI*, **Estado:** 1. *loginUserMI&1*.
    - Respuesta de mensaje síncrono.
    - **Cuerpo del mensaje:** [datos de usuario]%%[datos de semestre]%%[datos de materia 1]%% ... %%[datos de materia n]
      - Datos de materia:  
 login,password,nombre,email,fechaCreacion,código,tipoUsuario,"fin".
      - Datos semestre: idSemestre,periodo.
      - Datos de materia:  
 refMateria,nombreMateria,nomUsuario1&&&tipoUsuario1&&&codigo1, ..., nomUsuarioN&&&tipoUsuarioN&&&codigoN.

Los datos de materia se repiten por el numero de materias que tenga matriculadas en un semestre dado y donde *tipoUsuario#* depende del número de estudiantes matriculados en esa materia
  10. El ESB da la respuesta al SMI.

11. El SMI hace el registro o la actualización de los datos del usuario y por último hace la validación del usuario, si los datos concuerdan se ingresa al chat. Finaliza la secuencia 2 para login.

La **figura 44** muestra la secuencia del envío de un mensaje a otro usuario MI.



**Figura 44.** Diagrama de secuencia envío de mensaje MI

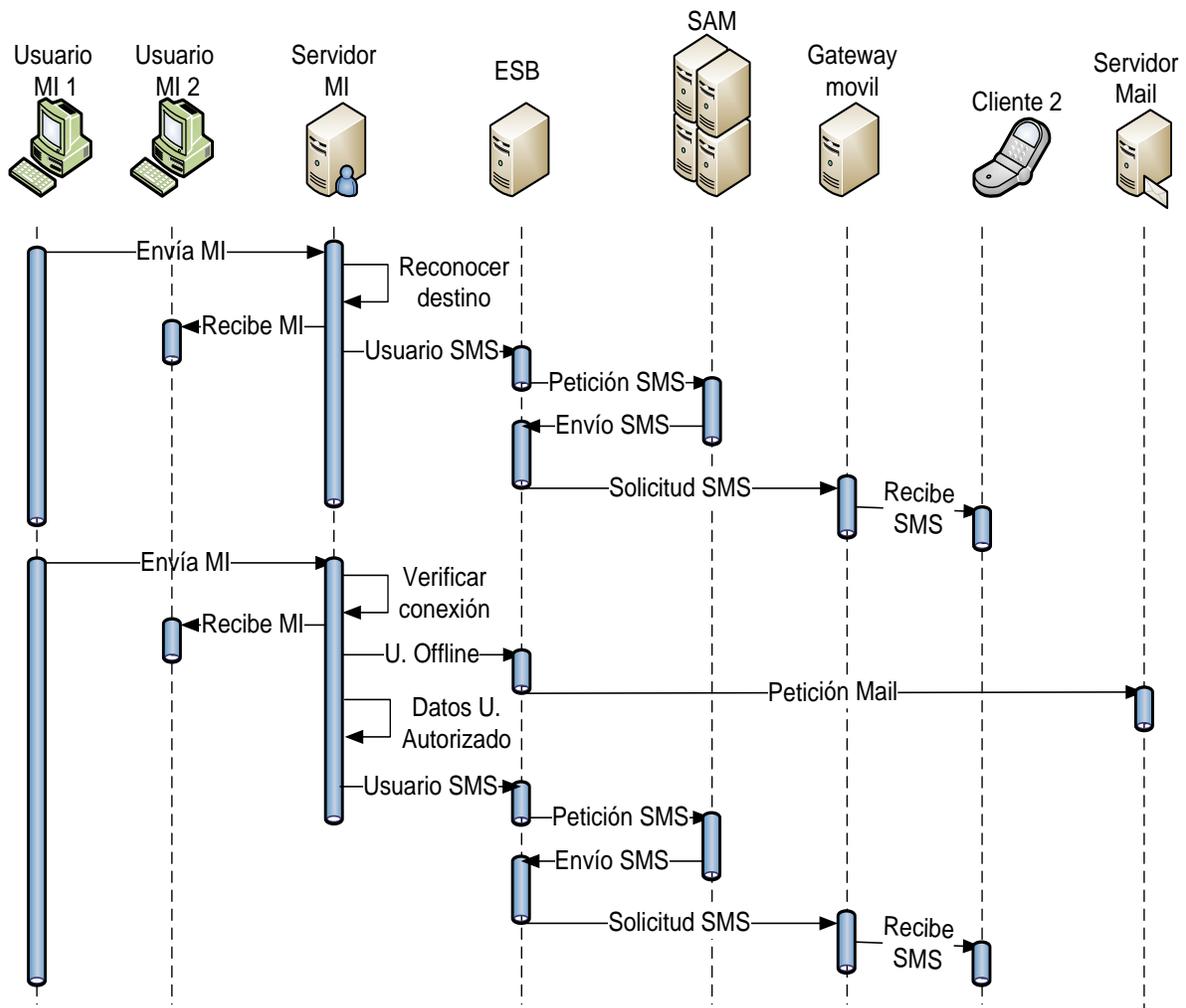
Secuencia de la interacción:

1. El usuario envía un SMS de la siguiente manera:
  - **Mensaje:** envía la palabra mensaje seguida por el userName de la persona a la que va dirigido y el texto a enviar, separadas con un espacio entre palabras (ejemplo: mensaje Ana hola como estas).
2. Los pasos 2 al 5 son similares a los pasos de login y logout. (ver **figura 43**).
3. Al recibir el mensaje, SMI verifica que tipo de conexión tiene el usuario de destino:
  - **UsuarioMI:** Es el usuario que está conectado directamente al servidor, esta conexión puede ser por la aplicación de escritorio o a través de GPRS por una aplicación móvil. En este caso se envía directamente el mensaje. Finaliza la secuencia de mensaje a través de SMS.
  - **Usuario SMS:** Si el usuario está conectado a través de un móvil vía SMS, se hace la solicitud al ESB para el envío de mensaje corto.
4. Solicitud de envío de un SMS:
  - **Tipo de Servicio:** *mensajeMovilMI*, **Estado:** 0. *mensajeMovilMI&0*
  - Mensaje asíncrono.
  - **Cuerpo del mensaje:** "MensajeSMS",to,movilTo,login,mensaje.
    - "MensajeSMS": me indica el tipo de mensaje a enviar.

- to: indica el userName a quien va dirigido el mensaje.
  - movilTo: indica el número móvil de destino.
  - login: UserName de la persona que manda el mensaje.
  - Mensaje: es el contenido del mensaje.
5. SAM, en este sirve como puente de datos de los mensajes que van al Gateway Móvil.
  6. El Gateway Móvil procesa y envía los datos a su destino. Finaliza la secuencia 2 de mensaje a través de SMS.

#### 4.2.1.2. Usuarios a través de un cliente MI

La **figura 45** muestra el diagrama de secuencia de los clientes MI conectados desde un terminal de datos, ya sea un computador personal o un móvil con una aplicación instalada.



**Figura 45.** Diagrama de secuencia clientes MI

Secuencia de la interacción, mensaje instantáneo:

1. El usuario de la aplicación de escritorio o de una aplicación móvil envía un MI.

2. Se realizan los pasos 3 a 6 para el envío de un mensaje MI a través de un SMS (ver **figura 44**).

Secuencia de la interacción, mensaje de notificación:

El otro tipo de mensaje que se puede enviar a través de la aplicación instalada en un computador personal, es el mensaje de notificación o urgente, siendo un tipo de mensaje para ser recibido con rapidez, este mensaje se podrá enviar a uno o un grupo de usuarios (un grupo de una asignatura académica), sin necesidad que estén conectados y les llegara en un tiempo corto de tiempo.

1. El usuario de la aplicación de escritorio o de una aplicación móvil envía un mensaje de notificación.
2. El servidor SMI verifica la conexión de los usuarios a quien va dirigido el mensaje.
  - **On-line:** Es el usuario que está conectado directamente al servidor. En este caso se envía directamente el mensaje a los usuarios de destino. Finaliza la secuencia de mensaje a través de SMS.
  - **Off-line:** Si los usuarios no están conectados, hay que verificar cuales han autorizado que este tipo de mensajes le lleguen al móvil a través de SMS.
    - **Autorizados:** Se realizan los pasos 3 a 6 del envío de un mensaje MI a través de un SMS (ver **figura 44**). Pero haciendo la salvedad que el mensaje puede estar dirigido a múltiples usuarios. Finaliza la secuencia de mensaje de notificación a través de SMS.
    - **No Autorizados:** Se hace la solicitud al ESB para el envío de correo electrónico
3. Solicitud de envío e-mail:
  - **Tipo de Servicio:** notifMessMI, **Estado:** 0. notifMessMI&0
  - Mensaje asíncrono.
  - **Cuerpo del mensaje:** permitidos%&desconectados%&texto.
    - permitidos: usuarios que permiten el envío de este mensaje a través de SMS
    - desconectados: lista de usuarios desconectados.
    - texto: contenido del mensaje.
4. El ESB recibe el mensaje, por el tipo de servicio y el estado en que se encuentra, lo enruta hacia el SM.
5. SM toma a los usuarios que no están en línea y que no autorizaron este mensaje a través de SMS y les envía un mail. Finaliza la secuencia de mensaje de notificación a través del correo electrónico.

## CAPÍTULO 5.

### Servicio de Prueba

El sistema es proyectado inicialmente, para ser usado en un entorno universitario como la Universidad del Cauca, aprovechando que la gran parte de los datos del estudiantado, profesores y administrativos están centralizados en una base de datos y/o en plataformas como SIMCA - Sistema Integrado de Matrículas y Control Académico - Plataforma de la División de Tecnologías de Información y Comunicación de la Universidad del Cauca. Aún así, la importancia de usar un escenario como la Universidad del Cauca, radica no en la interacción con los miembros de la comunidad universitaria, sino, en contar con acceso a la información académica necesaria para el funcionamiento y despliegue adecuado del prototipo [51]. En los anexos se presenta un manual de usuario con las interfaces típicas del prototipo y a continuación se enlistan sus características:

- Establece un escenario acorde a las características de una comunidad virtual académica.
- El prototipo se conecta a la red móvil por intermedio de un modem GSM.
- El prototipo simula una conexión con la SMSC de la red móvil.
- Permite recibir y enviar SMS.
- Permite el acceso de usuarios al chat mediante un servidor de mensajería instantánea.
- Usuarios con cualquier tipo de teléfono móvil pueden ingresar al chat haciendo uso de SMS.
- Un usuario puede ingresar al chat mediante una aplicación instalada en el móvil (celulares gama media).
- Los grupos en el chat son semejantes a las materias que tenga matriculadas en el semestre un usuario.
- La lista de usuarios de los diferentes grupos, está dada por todas las personas que están matriculadas en esa materia en el respectivo semestre.
- Las listas de usuarios se actualizan cuando un usuario entra al chat por primera vez, o cuando haya un cambio de semestre.
- Se permite crear un grupo especial de usuarios, con los cuales una persona se comunica constantemente y que no cambia con los semestres.
- La comunicación entre 2 usuarios puede hacerse entre una combinación de equipos de escritorio, aplicación móvil o SMS, sin que los usuarios necesiten saber cómo está conectado el otro usuario.
- En la configuración del perfil, un usuario puede aceptar que lleguen mensajes de notificación o urgentes a su móvil.
- Los mensajes de notificación o urgentes, se pueden enviar a uno o un grupo de usuarios sin necesidad de que ellos estén conectados.
- Los mensajes de notificación le llegaran a un usuario a su móvil a través SMS, si el mensaje instantáneo lo envían cuando este está desconectado y previamente ha aceptado recibir este tipo de mensaje.
- Los mensajes de notificación llegarán al correo electrónico, si el usuario esta desconectado en el momento del envío y no ha aceptado recibirlo en el móvil.
- El usuario SMS pude hacer las siguientes peticiones: Login, Logout y enviar un MI.

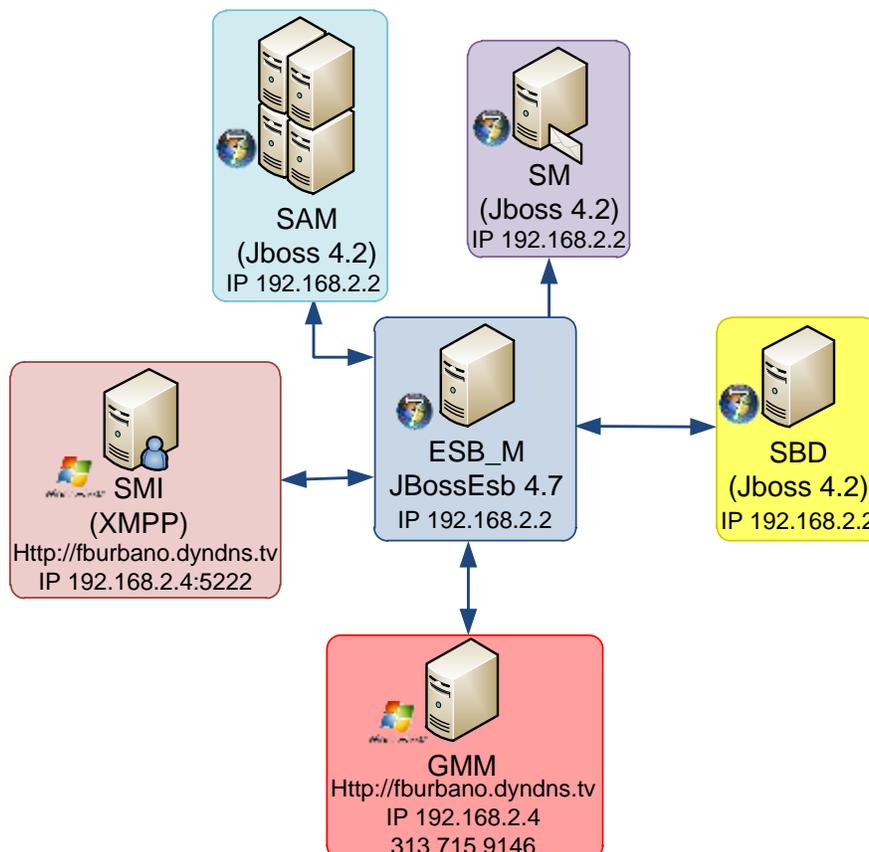
### 5.1. Descripción del Escenario de Despliegue

Para crear el escenario de pruebas de la Convergencia de servicios de mensajería móvil e instantánea mediante un ESB, se contó con dos equipos portátiles para cumplir con el papel de servidor y cliente; estos equipos se conectaron entre sí por un Router Inalámbrico creando así la Red del Servicio. Las características de los equipos se muestran en la **tabla 7**.

Marca	Procesador	RAM	Sistema Operativo
HP	AMD Atholon(tm) X2 Dual-Core QL-65 2.10 GHz	3 GB	Windows Vista Home Basic
Lenovo	Intel(R) Core(TM) i3 CPU M350 2.27 GHz	3 GB	Windows 7 Home Basic

**Tabla 7.** Características equipo de prueba

En la **figura 46** se muestra como fueron desplegados los diferentes módulos en los servidores JBoss 4.2, JBossESB 4.7, Openfire 3.7 sobre sistemas operativos Windows Vista Service Pack 1 y Windows 7. Este escenario fue tenido en cuenta como referencia para el esquema de pruebas presentado en la siguiente sección.



**Figura 46.** Ambiente de evaluación

- **Mecanismos o puertas de entrada al sistema:** La **tabla 8** muestra los mecanismos de entrada utilizados para crear el ambiente de pruebas.

	Mecanismos de entrada	Descripción
1.	http://fburbano.dyndns.tv	DNS dinámico para permitir el ingreso de clientes externos a la LAN
2.	192.168.2.4	Dirección IP que permite el ingreso clientes dentro de la LAN
3.	313 715 9146	Número de prueba para el Gateway módem GSM

**Tabla 8.** Puertas de entrada

- **Servidores y Puertos:** La **tabla 9** presenta la ubicación de los diferentes módulos del sistema y los puertos utilizados por estos para su comunicación.

Servidor	Dirección	Puertos
SMI	192.168.2.4	- 5222 Cliente – Servidor. - 9090 Consola de administración. - 1099 Puerto ESB.
GMM	192.168.2.4	- 8190 Comunicación SMSC. - 8080 Servicio para ligar al SMSC y al ESB. - 1099 Puerto ESB.
ESB	192.168.2.2	- 1099 Puerto ESB.
SBD	192.168.2.2	- 1099 Puerto ESB.
SAM	192.168.2.2	- 1099 Puerto ESB.
SM	192.168.2.2	- 1099 Puerto ESB.

**Tabla 9.** Características de los equipos servidores y puertos

## 5.2. Descripción del piloto

El sistema de pruebas para la descripción del piloto, crea una comunidad virtual académica de mensajería, basada en las materias matriculadas cada semestre por un estudiante, en la cual tendrá como amigos de chat, en el servidor de mensajería instantánea, a los otros estudiantes matriculados en los mismos cursos.

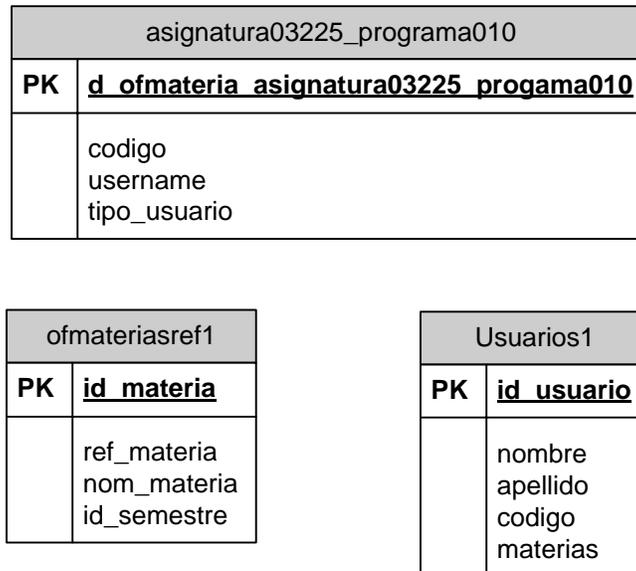
### 5.2.1. Base de datos del sistema

Para ubicar el entorno de pruebas dentro de la comunidad universitaria se pensó en obtener los datos desde SIMCA, pero debido a que no era viable debido a políticas de la División de TIC, dada la naturaleza de la información, se optó por crear una base de datos que simule un entorno Universitario donde haya estudiantes, docentes, cursos a matricular, etc. que brinde unas condiciones similares a los que ofrecería SIMCA.

El servidor SBD encierra toda la conexión y consultas a la base de datos. Para utilizar el sistema implementado en éste proyecto, en cualquier entorno ya sea en una empresa o en una comunidad académica, debe configurarse este servidor (drivers, consultas) con las características de la base de datos con que se cuenta; por ejemplo, para el caso de la Universidad del Cauca integrar el acceso a SIMCA con el SBD.

Para este servicio de prueba, se creó una base de datos con: 14510 estudiantes, 3000 materias y 20 programas, en donde las materias tienen capacidad hasta de 20 estudiantes matriculados por vez, y cada materia depende de un programa. Se utilizaron nombres genéricos para llenar estos datos. La **figura 47** muestra el esquema de la base

de datos utilizada para esta prueba. La tabla **asignatura03225\_programa010**, presenta un ejemplo de tabla para cada una de las materias creadas.



**Figura 47.** Diagrama de la base de datos del sistema

Normalmente, una base de datos se describe a través de un diagrama entidad relación. La base de datos utilizada en el prototipo, no corresponde con un ambiente real para un sistema de información, sino que se usan ciertas entidades de datos que permiten simular ese ambiente del sistema de información y que por lo tanto, las entidades no guardan una relación directa entre sí. Algunos campos e incluso el nombre de la entidad de datos, se usan para optimizar los procesos de búsqueda de información y evitar la complejidad de usar una base de datos relacional real.

Si bien, la estructura de una base de datos académica como la de SIMCA puede llegar a ser extremadamente compleja, para el sistema propuesto, solo se tomaron en cuenta las entidades de datos estrictamente necesarias para su correcto funcionamiento. Esta estructura de datos simplifica el proceso de simulación de una comunidad virtual académica.

### 5.2.2. Método de prueba

En varias partes del sistema se necesita generar solicitudes de varios usuarios (estudiantes) en un corto tiempo. Como es difícil reunir a un grupo grande de estudiantes, se optó por generar esas solicitudes a través de un hilo de peticiones generado por el simulador de SMSC creado para este proyecto. Este simulador nos permite enviar mensajes al GMM ya sea de login, logout, mensajes IM, etc. programando el tipo de mensaje, cantidad de mensajes por segundo; el teléfono de origen, etc. y el texto del mensaje lo creará el Simulador SMSC [57]. La **figura 48** muestra el panel principal del simulador para originar múltiples mensajes.

Cliente	GWIN_APPSRV97	Deliver sm	
phone	3122580000		
text	login		
destAdrr	52942		
<input checked="" type="checkbox"/> Multi-Msj	Cantidad:	10	Multi Dlvr sm
To:	Msjx Seg:	5	
		Tf:	

Figura 48. Panel principal multimensajes SimSMSC

### 5.2.3. Establecimiento del canal de escucha ESB

Para poder utilizar este sistema en primera instancia debe probarse que la comunicación entre los servidores y el ESB se realice correctamente. Para esto se utiliza la herramienta **WireShark** la cual es una herramienta gráfica utilizada para capturar y analizar el tráfico de red en un momento determinado. Para la captura de paquetes de forma más precisa, se utiliza un filtro con la IP del ESB y el puerto de escucha 1099.

Para recrear el establecimiento de la comunicación con el ESB se toma como ejemplo la solicitud de conexión que hace el SMI. La **figura 49** muestra los paquetes de comunicación entre estos 2 servidores, para establecer su comunicación. Los frame 1-2 hacen la solicitud de conexión, en el cuadro rojo se puede ver como el **SMI** solicita la conexión para el puerto `JMS queue.serviMI_reply_gw`. Los frame 3-4 son las respuestas provenientes del ESB, y así queda establecida la conexión.

En la **figura 50** se muestra en la consola de Windows como se establece esta comunicación. En ella se ve claramente como el **SMI** se prepara para el establecimiento de la conexión hasta que llega el mensaje de aceptación desde el ESB.

Source	Destination	Protocol	Info
192.168.2.4	192.168.2.2	TCP	61305 > rmiactivation [ACK] Seq=2123 Ack=2101 win=17516 Len=1460
192.168.2.4	192.168.2.2	TCP	61305 > rmiactivation [PSH, ACK] Seq=3583 Ack=2101 win=17516 Len=193
192.168.2.2	192.168.2.4	TCP	rmiactivation > 61305 [ACK] Seq=2101 Ack=3776 win=65536 Len=0
192.168.2.2	192.168.2.4	TCP	rmiactivation > 61305 [PSH, ACK] Seq=2101 Ack=3776 win=65536 Len=597
0000	00 26 82 9e 91 a2 00 26 82 07 c4 3f 08 00 45 00		..&.....& ...?.E.
0010	00 e9 27 3c 40 00 80 06 4d 7c c0 a8 02 04 c0 a8		.. '<@... M .....
0020	02 02 ef 79 04 4a 4f c2 80 6d 73 5d 07 b0 50 18		...y.JO. .ms]..P.
0030	11 1b 32 65 00 00 70 2e 6a 61 72 78 72 00 14 6a		..2e..p. jarxr..j
0040	61 76 61 2e 75 74 69 6c 2e 50 72 6f 70 65 72 74		ava.util .Propert
0050	69 65 73 39 12 d0 7a 70 36 3e 98 02 00 01 4c 00		ies9..zp 6>....L.
0060	08 64 65 66 61 75 6c 74 73 74 00 16 4c 6a 61 76		.default st..Ljav
0070	61 2f 75 74 69 6c 2f 50 72 6f 70 65 72 74 69 65		a/util/P ropertie
0080	73 3b 70 78 72 00 13 6a 61 76 61 2e 75 74 69 6c		s;pxr..j ava.util
0090	2e 48 61 73 68 74 61 62 6c 65 13 bb 0f 25 21 4a		.Hashtable le...%J
00a0	e4 b8 03 00 02 46 00 0a 6c 6f 61 64 46 61 63 74		.....F.. loadFact
00b0	6f 72 49 00 09 74 68 72 65 73 68 6f 6c 64 70 78		orI..thr esholdpx
00c0	70 3f 40 00 00 00 00 00 08 77 08 00 00 00 0b 00		p?@..... .w.....
00d0	00 00 00 78 70 77 04 00 00 00 02 74 00 05 71 75		...xpw.. ...t..qu
00e0	65 75 65 74 00 10 73 65 72 76 69 4d 49 5f 72 65		euet..se rviMI_re
00f0	70 6c 79 5f 67 77 78		ply_gwx

Figura 49. Establecimiento de la comunicación con ESB, WhiresShark

```
c:\workspace\ServidorMensajeriaInstantanea\target\openfire\lib>java -jar startup.jar
name: felipe1
Openfire 3.6.4 [15-nov-2011 10:55:14]
DEBUG [main] (SpyConnectionFactoryObjectFactory.java:56) - Extracting SpyConnectionFactory from reference
DEBUG [main] (SpyConnectionFactoryObjectFactory.java:64) - The GenericConnectionFactory is: GenericConnecti
ce=org.jboss.mq.il.uil2.UILClientILService, UIL_TCPNODELAY_KEY=yes, UIL_CHUNKSIZE_KEY=1000000, UIL_CONNECTA
UIL_ADDRESS_KEY=192.168.2.2, UIL_PORT_KEY=8093, PingPeriod=60000, UIL_BUFFERSIZE_KEY=2048}]
DEBUG [main] (Connection.java:202) - Setting the clockDaemon's thread factory
DEBUG [UIL2.SocketManager.ReadTask#1 client=192.168.2.2:8093] (SocketManager.java:323) - Begin ReadTask.run
DEBUG [UIL2.SocketManager.WriteTask#2 client=192.168.2.2:8093] (SocketManager.java:526) - Begin WriteTask.r
DEBUG [UIL2.SocketManager.WriteTask#2 client=192.168.2.2:8093] (SocketManager.java:532) - Created ObjectOut
DEBUG [UIL2.SocketManager.ReadTask#1 client=192.168.2.2:8093] (SocketManager.java:328) - Created ObjectInpu
DEBUG [main] (UILClientILService.java:154) - Starting
DEBUG [main] (SpyDestinationObjectFactory.java:58) - SpyDestinationObjectFactory->getObjectInstance()
Se ha inicializado la escucha de: QUEUE.serviMI_reply_gw
Consola de administración escuchando en:
http://felipe1:9090
https://felipe1:9091
Starting Monitoring Plugin
```

Figura 50. Establecimiento de la comunicación con ESB, consola de Windows

Este tipo de conexión se hace para cada uno de los servidores que tienen que estar en frente de un servicio soportado sobre Internet, en este prototipo se puede ver en el SMI y en el GMM.

Después de establecer la conexión con el ESB, se prueba que la comunicación o conexión entre los servidores es correcta. Para esta prueba se utilizó la solicitud que hace el SMI al momento de registrar un usuario nuevo, del cual no se poseen los datos y hay que traerlos del SBD a través del ESB. Esta comunicación es síncrona. El SMI espera la respuesta que le llega vía ESB por lo tanto da una visión más detallada de esta comunicación [53]. La figura 51 muestra la solicitud que hace el SMI, en el recuadro se aprecia el puerto de escucha (serviMI\_route), el tipo de servicio (loginUserMI), el nickname del usuario (ana), y el servidor de destino (servidorBD). En la figura 52 se muestra la respuesta que recibe el SMI a la petición anterior, resaltado de rojo están los datos pertenecientes al nickname (username, password, lista de materias, lista de alumnos matriculados en la materias, etc.), en verde se observan el tipo de servicio y el servidor de destino.

Source	Destination	Protocol	Info
192.168.2.4	192.168.2.2	TCP	49505 > 8093 [PSH, ACK] Seq=294 Ack=105 win=17416 Len=224
0000	00 26 82 9e 91 a2 00 26 82 07 c4 3f 08 00 45 00		.&....& ...?..E.
0010	01 08 30 ca 40 00 80 06 43 cf c0 a8 02 04 c0 a8		..0.@... C.....
0020	02 02 c1 61 1f 9d 36 7d fe 31 25 96 a5 8e 50 18		...a..6} .1%...P.
0030	44 08 30 32 00 00 77 dd 02 00 00 00 86 00 01 01		D.02..w. ....
0040	02 00 0a 73 65 72 76 69 4d 49 5f 67 77 00 00 00		...servi MI_gw...
0050	02 00 00 00 00 00 00 00 00 00 00 00 04 07 00 14		.....
0060	49 44 3a 33 31 2d 31 33 32 31 34 36 32 33 30 34		ID:31-13 21462304
0070	35 35 32 31 00 00 01 33 ad 4a 8f 28 01 07 00 10		5521...3 .J.(....
0080	51 75 69 63 6b 73 74 61 72 74 49 64 5b 34 34 5d		Quicksta rtId[44]
0090	ff ff ff ff 02 00 0d 73 65 72 76 69 4d 49 5f 72		.....s erviMI_r
00a0	6f 75 74 65 09 00 01 00 07 00 05 49 44 3a 33 31		oute.... ...ID:31
00b0	00 00 00 03 00 0c 74 69 70 6f 53 65 72 76 69 63		.....ti poServic
00c0	69 6f 07 00 0d 6c 6f 67 69 6e 55 73 65 72 4d 49		io...log inuserMI
00d0	26 30 00 04 62 65 61 6e 07 00 07 61 6e 61 2c 66		&0..bean ...ana,f
00e0	69 6e 00 0e 73 65 72 76 69 64 6f 72 53 61 6c 69		in..serv idorSali
00f0	64 61 07 00 0a 73 65 72 76 69 64 6f 72 42 44 00		da...ser vidorBD.
0100	00 00 00 11 ac ed 00 05 74 00 0a 73 65 72 76 69		..... t..servi
0110	64 6f 72 42 44 79		dorBDy

Figura 51. Petición de login del SMI

```

Source          Destination      Protocol  Info
192.168.2.2    192.168.2.4    TCP      8093 > 49505 [PSH, ACK] Seq=147 Ack=939 win=63318 Len=904
0000 00 26 82 07 c4 3f 00 26 82 9e 91 a2 08 00 45 00 .&...?.& .....E.
0010 03 b0 65 03 40 00 80 06 0c ee c0 a8 02 02 c0 a8 ..e.@... ..
0020 02 04 1f 9d c1 61 25 96 a5 b8 36 7e 00 b6 50 18 ....a%. ..6~.P.
0030 f7 56 60 3e 00 00 7a 00 00 03 82 19 00 00 00 00 .V>..Z. ....
0040 00 00 00 00 01 04 02 00 0d 73 65 72 76 69 4d 49 .....serviMI
0050 54 6d 70 5f 67 77 00 00 00 02 00 00 00 00 00 00 Tmp_gw... ..
0060 00 00 00 00 00 04 07 00 15 49 44 3a 31 33 2d 31 .....ID:13-1
0070 33 32 31 34 36 32 33 31 36 31 30 31 31 34 00 00 32146231 610114..
0080 01 33 ad 4a bc 45 01 07 00 10 51 75 69 63 6b 73 .3.J.E...Quicks
0090 74 61 72 74 49 64 5b 34 34 5d ff ff ff ff 02 00 tartId[4].....
00a0 0d 73 65 72 76 69 4d 49 5f 72 6f 75 74 65 09 00 .serviMI_route..
00b0 00 01 07 00 05 49 44 3a 31 33 00 00 00 04 00 04 ....ID: 13.....
00c0 62 65 61 6e 07 02 89 61 6e 61 2c 61 6e 61 2c 61 bean...a na,ana,a
00d0 6e 61 2c 61 6e 61 40 75 6e 69 63 61 75 63 61 2e na,ana@ nicauca.
00e0 65 64 75 2e 63 6f 2c 30 2c 31 30 31 31 31 34 33 edu.co,0 ,1011143
00f0 30 33 2c 30 2c 66 69 6e 25 26 25 33 2c 49 2d 32 03,o,fin %&&3,I-2
0100 30 31 31 25 26 25 61 73 69 67 6e 61 74 75 72 61 011%&&as ignatura
0110 30 32 39 38 30 5f 70 72 6f 67 61 6d 61 31 39 2c 02980_pr ogama19,
0120 61 73 69 67 6e 61 74 75 72 61 30 32 39 38 30 2c asignatu ra02980,
0130 61 6c 75 6d 6e 6f 31 33 36 34 37 26 26 26 26 26 31 alumno13 647&&&&1
0140 31 30 30 26 26 26 26 31 30 31 31 31 33 36 34 37 100&&&&1 01113647
0150 2c 61 6c 75 6d 6e 6f 31 33 35 39 31 26 26 26 26 ,alumno1 3591&&&&
0160 31 31 30 30 26 26 26 26 31 30 31 31 31 33 35 39 1100&&&& 10111359
0170 31 2c 61 6c 75 6d 6e 6f 31 33 37 34 35 26 26 26 26 1,alumno 13745&&&
0180 26 31 31 30 30 26 26 26 26 26 31 30 31 31 31 33 37 &1100&&& &1011137
0190 34 35 2c 61 6c 75 6d 6e 6f 31 33 37 36 31 26 26 26 45,alumn oi3761&&
01a0 26 26 31 31 30 30 26 26 26 26 26 31 30 31 31 31 33 &&1100&& &&101113
01b0 37 36 31 2c 61 6c 75 6d 6e 6f 31 33 38 37 36 26 761,alumn oi3876&
01c0 26 26 26 31 31 30 30 26 26 26 26 31 30 31 31 31 &&&1100& &&&10111
01d0 33 38 37 36 2c 61 6c 75 6d 6e 6f 31 33 39 38 30 3876,alu mno13980
01e0 26 26 26 26 31 31 30 30 26 26 26 26 31 30 31 31 &&&&1100 &&&&1011
01f0 31 33 39 38 30 2c 61 6c 75 6d 6e 6f 31 34 32 38 13980,al umno1428
0200 38 26 26 26 26 31 31 30 30 26 26 26 26 31 30 31 8&&&&110 0&&&&101
0210 31 31 34 32 38 38 25 26 25 61 73 69 67 6e 61 74 1142888% assignat
0220 75 72 61 30 32 39 34 34 5f 70 72 6f 67 61 6d 61 ura02944 _progama
0230 31 39 2c 61 73 69 67 6e 61 74 75 72 61 30 32 39 19,asign atura029
0240 34 34 2c 61 6c 75 6d 6e 6f 31 33 36 36 37 26 26 44,alumn oi3667&&
0250 26 26 31 31 30 30 26 26 26 26 31 30 31 31 33 &&1100&& &&101113
0260 36 36 37 2c 61 6c 75 6d 6e 6f 31 33 39 31 35 26 667,alumn oi3915&
0270 26 26 26 31 31 30 30 26 26 26 26 31 30 31 31 31 &&&1100& &&&10111
0280 33 39 31 35 2c 61 6c 75 6d 6e 6f 31 33 38 38 33 3915,alu mno13883
0290 26 26 26 26 31 31 30 30 26 26 26 26 31 30 31 31 &&&&1100 &&&&1011
02a0 31 33 38 38 33 2c 61 6c 75 6d 6e 6f 31 33 38 39 13883,al umno1389
02b0 33 26 26 26 26 31 31 30 30 26 26 26 26 31 30 31 3&&&&110 0&&&&101
02c0 31 31 33 38 39 33 2c 61 6c 75 6d 6e 6f 31 33 39 113893,a lumno139
02d0 35 30 26 26 26 26 31 31 30 30 26 26 26 26 31 30 50&&&&11 00&&&&10
02e0 31 31 31 33 39 35 30 2c 61 6c 75 6d 6e 6f 31 34 1113950, alumno14
02f0 30 36 35 26 26 26 26 31 31 30 30 26 26 26 26 31 065&&&&1 100&&&&1
0300 30 31 31 31 34 30 36 35 2c 61 6c 75 6d 6e 6f 31 01114065 ,alumno1
0310 34 32 31 31 26 26 26 26 31 31 30 30 26 26 26 26 4211&&&& 1100&&&&
0320 31 30 31 31 31 34 32 31 31 2c 61 6c 75 6d 6e 6f 10111421 1,alumno
0330 21 34 32 39 35 26 26 26 26 26 31 31 30 30 26 26 26 14295&&& &1100&&&
0340 26 31 30 31 31 31 34 32 39 35 25 26 25 66 69 6e &1011142 95&&&fin
0350 00 0c 74 69 70 6f 53 65 72 76 69 63 69 6f 07 00 .tipoSe rvicio..
0360 0d 6c 6f 67 69 6e 55 73 65 72 4d 49 26 32 00 0c .loginus erMI&2..
0370 6e 6f 74 69 66 69 63 61 63 69 6f 6e 07 00 0b 6e notifica cion...n
0380 6f 74 69 66 69 63 61 72 49 4d 00 0e 73 65 72 76 otificar IM..serv
0390 69 64 6f 72 53 61 6c 69 64 61 07 00 0a 53 65 72 idorSali da...Ser
03a0 76 69 64 6f 72 4d 49 08 00 00 00 01 00 0a 53 65 vidorMI. ....Se
03b0 72 76 69 64 6f 72 4d 49 01 80 00 00 00 79 rvidorMI .....v

```

Figura 52. Respuesta del ESB

Por último, en la **figura 53** se observa en consola el proceso que tiene la petición de login de un usuario no registrado en el SMI (SMI - ESB - SBD - ESB - SMI). En el cuadro enmarcado con rojo se puede ver el momento que el ESB recibe y procesa la petición que viene del SMI, la solicitud es del tipo *loginUserMI*, finaliza al ser enviada al SBD [54]. En el marco amarillo se puede ver los procesos que hace el mensaje dentro del SBD, se inicializa la base de datos, toma los datos del usuario (después que ha hecho login en el sistema) incluyendo las materias que tiene matriculadas en ese momento, para finalmente enviarlo de nuevo al ESB y este en el recuadro verde, direccionarlo de nuevo al SMI.



El mismo proceso se hace en el otro equipo que sirve como servidor. En la **figura 55** se muestra el proceso de carga que hay al iniciar los servidores. Se empieza con una carga del 45% de la capacidad del equipo, al arrancar el servidor de mensajería instantánea SMI sube al 48% y al lanzar la Gateway móvil, la carga de memoria llega al 55%, teniendo un pico de 57% en el momento de activar el puerto de escucha con el ESB y el modem GSM.

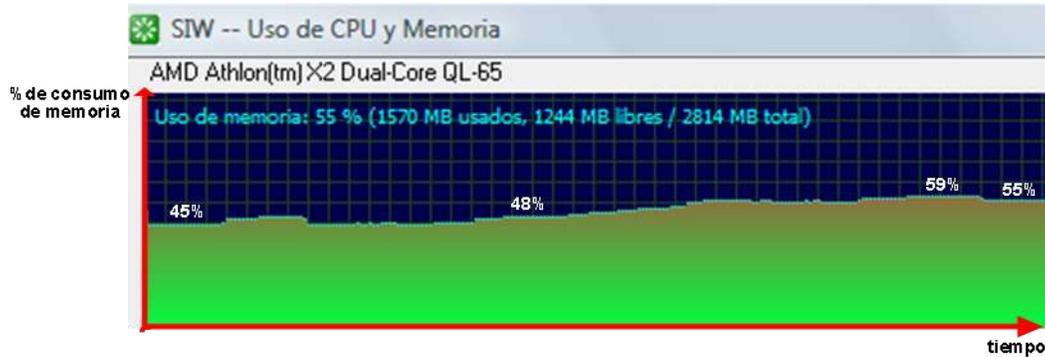


Figura 55. Consumo de memoria SMI y GMM

Otra prueba en el consumo de memoria era ver el comportamiento de las máquinas cuando se genera tráfico entre los servidores ya que esto implica una cantidad de procesos de atención de los usuarios, activados al mismo tiempo. Para ello se lanza una solicitud de login desde simSMS (simulador de SMS) para 10, 50 y 100 usuarios en un segundo. La **figura 56** muestra el comportamiento que tiene el equipo donde se encuentra alojado el ESB. A pesar de aumentar el número de peticiones por segundo, el comportamiento del servidor, en cuestión de memoria, es muy similar siempre, consumiendo aproximadamente el 4% (120 Mb aproximadamente) de memoria del equipo mientras está dando respuesta a las diferentes solicitudes.

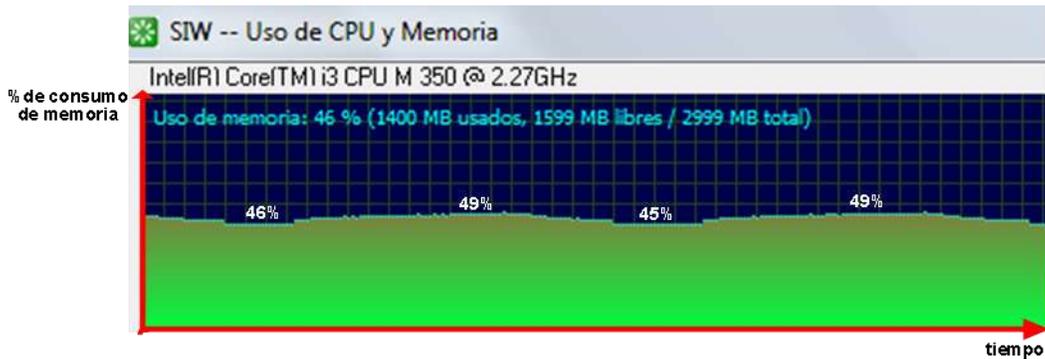
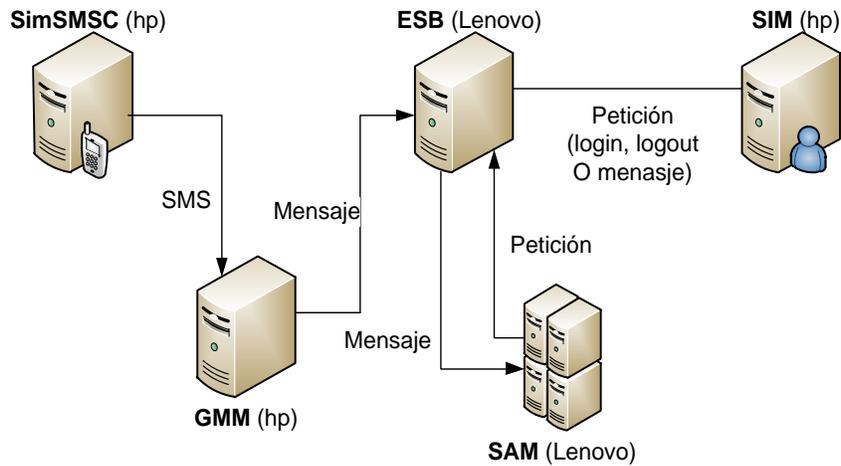


Figura 56. Consumo de memoria para tráfico ESB

Es de notar para este caso y con los equipos que se utilizaron para realizarlo, que el consumo de memoria es muy estable sin importar la carga inicial de memoria de los equipos. Se recomienda que los servidores dispongan de al menos 3 GB o más de memoria RAM y que se establezcan por separado el ESB y los servicios soportados sobre Internet. Lo ideal sería tener equipos servidores especializados.

### 5.2.5. Tiempos de respuesta

Es necesario analizar el tiempo promedio que el sistema tarda en responder una solicitud; ya que el desempeño de un sistema se mide por el tiempo que tarda el sistema en responder a una petición. Para esto se utilizará el SimSMSC para crear una cantidad de peticiones por segundo, las peticiones serán del tipo login, logout y mensaje, las cuales en términos lógicos serán las más utilizadas por los usuarios. La **figura 57** muestra el esquema de la prueba de solicitud que se realiza.



**Figura 57.** Proceso para entrada de SMS

Las **tablas 10, 11 y 12** muestran el tiempo que tarda el sistema en responder a los diferentes mensajes. La cantidad de peticiones varía en número de peticiones por segundo (peticiones/segundo), la prueba se realizó en varias ocasiones y se calcula un promedio con los tiempos que proyecta para cada cantidad y tipo de mensaje. Los tiempos se dan en (minutos: segundos: milésimas de segundo). La **figura 58** muestra una comparación de los tiempos de respuesta por mensaje.

Según los especialistas en usabilidad, diez segundos (10 segundos) es el tiempo máximo que un usuario puede permanecer atento a una tarea. Si el tiempo de respuesta es mayor, los usuarios tienden a ocupar su tiempo en otras cosas [52]. Según esto, el sistema está dentro de un rango de respuesta aceptable, el cual puede variar al usar equipos con más capacidades o especializados para estas tareas.

Mensaje Login		
Peticiones	Tiempo/Respuesta (mm:ss:ms)	Promedio/Respuesta (mm:ss:ms)
1	00:07:24	00:07:240
10	00:46:28	00:04:628
100	05:43:01	00:05:430
Promedio total		00:05:099

**Tabla 10.** Tiempo de respuesta login

Mensaje Logout		
Peticiones	Tiempo/Respuesta (mm:ss:ms)	Promedio/Respuesta (mm:ss:ms)
1	00:07:240	00:05:780
10	00:39:347	00:03:934
100	05:37:300	00:05:373
Promedio total		00:04:362

Tabla 11. Tiempo de respuesta logout

Mensaje IM		
Peticiones	Tiempo/Respuesta (mm:ss:ms)	Promedio/Respuesta (mm:ss:ms)
1	00:06:780	00:06:780
10	00:58:623	00:05:862
100	05:30:100	00:05:301
Promedio total		00:05:072

Tabla 12. Tiempo de respuesta mensaje

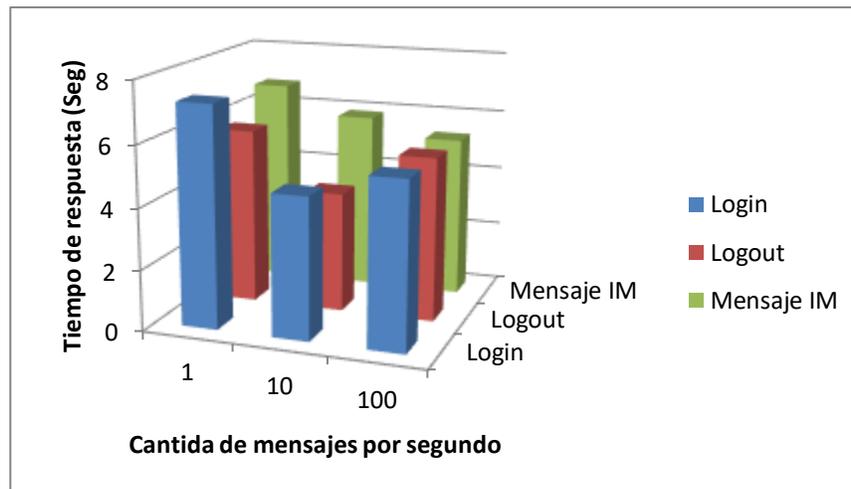


Figura 58. Comparación de tiempos de respuesta por mensaje

## 5.2.6. Comunicación y tamaño de datos de la aplicación para móvil

El objetivo de este punto consiste en analizar el tráfico de una comunicación básica entre un usuario móvil y el SMI, para así dar un estimado del consumo en bytes y poder analizar la viabilidad para un usuario que utiliza el sistema. Por lo tanto es importante establecer el tamaño de los mensajes que se intercambian entre un móvil y el servidor de mensajería instantánea. Para este fin se hace uso de **Wireless Toolkit** de Sun Microsystems, con el cual se genera el tráfico entre la aplicación y el servidor [56].

La figura 59, muestra el proceso de comunicación jabber entre la aplicación móvil y el SMI. En la primera parte enmarcado en rojo, muestra la solicitud de login, en donde se pueden ver la petición y la respuesta jabber <stream:stram>, lo cual indica el inicio de sesión de un cliente. Luego se ve la solicitud de autenticación y por último el protocolo roster, el cual sirve para traer la lista de alumnos en cada materia que está matriculado este usuario.

En el marco verde se pueden ver los mensajes que llegan a la aplicación móvil para notificar que un alumno se acaba de conectar o desconectar, esto utilizando el protocolo *presence*.

El protocolo *message* es extremadamente sencillo, los paquetes son enviados por un usuario a un receptor. Por defecto, no se reciben confirmaciones de que el mensaje ha sido recibido por el destinatario para reducir el tráfico en el servidor, esto se puede ver en el recuadro amarillo. Por último el cuadro azul muestra la solicitud de logout, del cliente hacia el servidor [42].

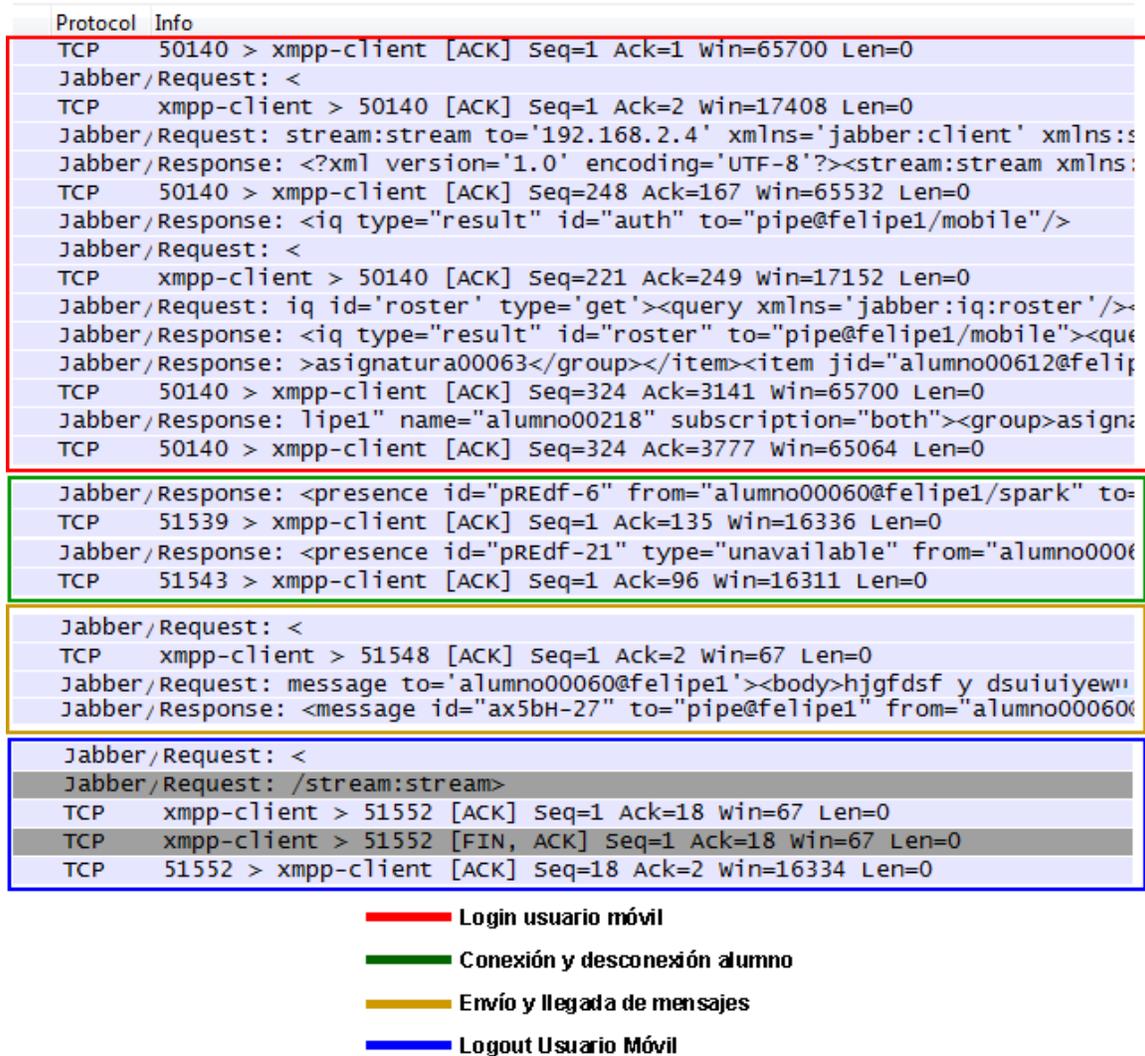


Figura 59. Comunicación SMI - Cliente móvil

La base de datos que se creó para esta prueba, generó un máximo de 3 asignaturas matriculadas por alumno y cada asignatura tiene hasta 20 personas matriculadas; el consumo de Kilobyte va a variar de un usuario a otro, dependiendo de la cantidad de compañeros que tenga en las materias matriculadas para ese semestre. En la **tabla 13**, es posible observar tamaño del flujo de bytes originado al momento de registrar, por primera vez o al hacer la actualización semestral, los datos dentro de la aplicación móvil.

En la **tabla 13** también se muestra la información de 5 alumnos (5 primeras líneas) al azar que tengan 1, 2 o 3 materias matriculadas. Se observa el tamaño de los datos haciendo uso de Wireshark y los protocolos jabber de comunicación entre el usuario y el servidor. Además se puede ver que el tamaño de los frame no varía mucho y se puede decir que es estándar para todos.

La variación de Bytes está dada por el protocolo roster [55], el cual devuelve la lista de “compañeros”, que tenga el usuario en cada semestre. El tamaño o la cantidad de estos hará variar el tamaño total de bytes que se consumen. Cabe resaltar que este proceso se realizará 1 vez cada inicio de semestre. Mientras el semestre transcurre y ya se ha hecho esta primera solicitud de login, el tamaño del paquete de datos es de 335 bytes cada vez que ingrese a la aplicación, ya que no se usaría el protocolo roster. Los datos se muestran en la **tabla 14**.

# Alumnos				Byte									
Por materia				Bytes enviados			Bytes recibidos				Total	Por compañero	
1	2	3	Total	Stream	Roster	Total	Stream	Auth	Roster	Total			
17	10	19	46	369	184	553	220	115	4049	4384	4937	88,02	
10	10	16	36	369	184	553	220	115	3833	4168	4721	106,47	
9	12	11	32	369	184	553	220	115	3401	3736	4289	106,28	
10	6	6	22	369	184	553	220	115	2591	2926	3479	117,77	
13	9	13	35	355	184	539	220	108	3718	4046	4585	106,23	
15	6		21	369	184	553	219	115	2267	2601	3154	107,95	
8	8		16	369	184	553	220	115	1717	2052	2605	107,31	
12	13		25	369	184	553	220	115	2699	3034	3587	107,96	
10	15		25	369	184	553	220	115	2699	3034	3587	107,96	
7	8		15	369	184	553	220	115	1827	2162	2715	121,80	
9			9	369	184	553	220	115	1025	1360	1913	113,89	
16			16	369	184	553	220	115	1835	2170	2723	114,69	
14			14	369	184	553	220	115	1619	1954	2507	115,64	
10			10	369	184	553	220	115	1133	1468	2021	113,30	
15			15	369	184	553	220	115	1720	2055	2608	114,66	

**Tabla 13.** Consumo de bytes solicitud login Roster

Byte							
Byte Bajada			Byte Subida				Total
Stream	Roster	Total	Stream	Auth	Roster	Total	
369	119	488	220	115	0	335	823

**Tabla 14.** Consumo de bytes solicitud login sin Roster

La **tabla 15** muestra el consumo de byte para las otras solicitudes jabber en la aplicación móvil. Para el envío y recepción de mensajes instantáneos, el consumo de bytes depende del tamaño del mensaje, la prueba se hizo para mensajes de 1, 100 y 200 letras tanto en mensajes de entrada como de salida. El protocolo *presence* de *jabber* es el que permite saber si un compañero se conecta (in) o se desconecta del chat (out).

Solicitud		Total (bytes)
Presence	in	188 - 195
	out	149 - 156
Mensaje in	1	239 - 246
	100	345 - 351
	200	442 - 445
Mensaje out	1	111 - 112
	100	213 - 229
	200	308 - 322
Logout		124

**Tabla 15.** Consumo de bytes de las solicitudes del protocolo jabber

Para medir la viabilidad económica de este sistema, se opta por un cliente prepago, sin plan de datos que se conecta a la aplicación móvil ya sea por primera vez o en una ocasión posterior. Estando conectado, dos compañeros ingresan al chat (presence in) y dos compañeros salen de esta (presence out), entabla una conversación con un compañero de 3 mensajes de entrada y 3 de salida, por ultimo sale de la aplicación. La **tabla 16** muestra la cantidad de bytes que se consumen por cada uno de los anteriores pasos, el valor del kilobyte por demanda (sin ningún plan de datos prepago) para 2 operadores de telefonía móvil colombiana [58] [59] y el precio total de toda esta transacción.

Comunicación Jabber (Kilo Bytes)									
Login		Presence		Mensajes		Logout	Total		
Roster	Sin Roster	In X 2	Out X 3	In X 3	Out X 3		Roster	Sin Roster	
4,937	0,823	0,585	0,312	1,053	0,687	0,124	7,698	3,584	
						<b>Precio (Pesos Colombianos)</b>			
						<b>Operador</b>	KB demanda	Roster	Sin Roster
						Comcel	19,28	148,417	69,0995
						Tigo	12	92,376	43,008

**Tabla 16.** Referencia de precios operadores nacionales

El precio total que genera el servicio para un cliente prepago es poco, lo cual lo hace viable incluso para usuarios prepago que consumen Kilobytes por demanda [60] [61]. Los planes y paquetes de datos e internet móvil que ofrecen los operadores de telefonía, hará más sencillo el uso de esta aplicación ya que ellos dan una cantidad limitada de Gigabytes por un precio moderado y como ya se analizó el consumo de bytes de la aplicación es muy bajo.

### 5.2.7. Concurrencia

Ya que este proyecto está pensado para un ámbito académico como la Universidad del Cauca, se quiere probar si el servidor de mensajería tiene la capacidad de manejar aproximadamente 14000. Para esto se crea un ciclo en el cual se enviarán 14000 solicitudes de *login*, finalizado este proceso se crearon conversaciones entre los

participantes al chat y por último se enviarán solicitudes de logout. Se desea ver cuál es el comportamiento del SMI y el rendimiento del equipo al hacer esta prueba.

En la **figura 60** y utilizando el plugin de monitoreo del servidor de mensajería, se puede ver las estadísticas arrojadas por la prueba. En la primer gráfica *conexiones de clientes*, en la barra vertical está el número de clientes conectados y en la barra horizontal la hora del día, se puede ver 2 pruebas, la conexión de 5000 usuarios, con una duración de 10 minutos aproximadamente, 8 minutos para el proceso de login y 2 minutos para desconectarlos. La segunda prueba es para conectar los 14000 usuarios la cual duro aproximadamente 23 minutos para registrarlos, 7 minutos para hacer pruebas de mensajes entre ellos, se establecen 500 conexiones entre usuarios con intercambio de mensajes; por último el proceso de desconexión de los 14000 usuarios tiene una duración de 12 minutos. El segundo gráfico muestra el número de paquetes por minuto mientras se hacen los procesos anteriores, siendo visible que es el proceso de conexión o login el que más genera tráfico al SMI.

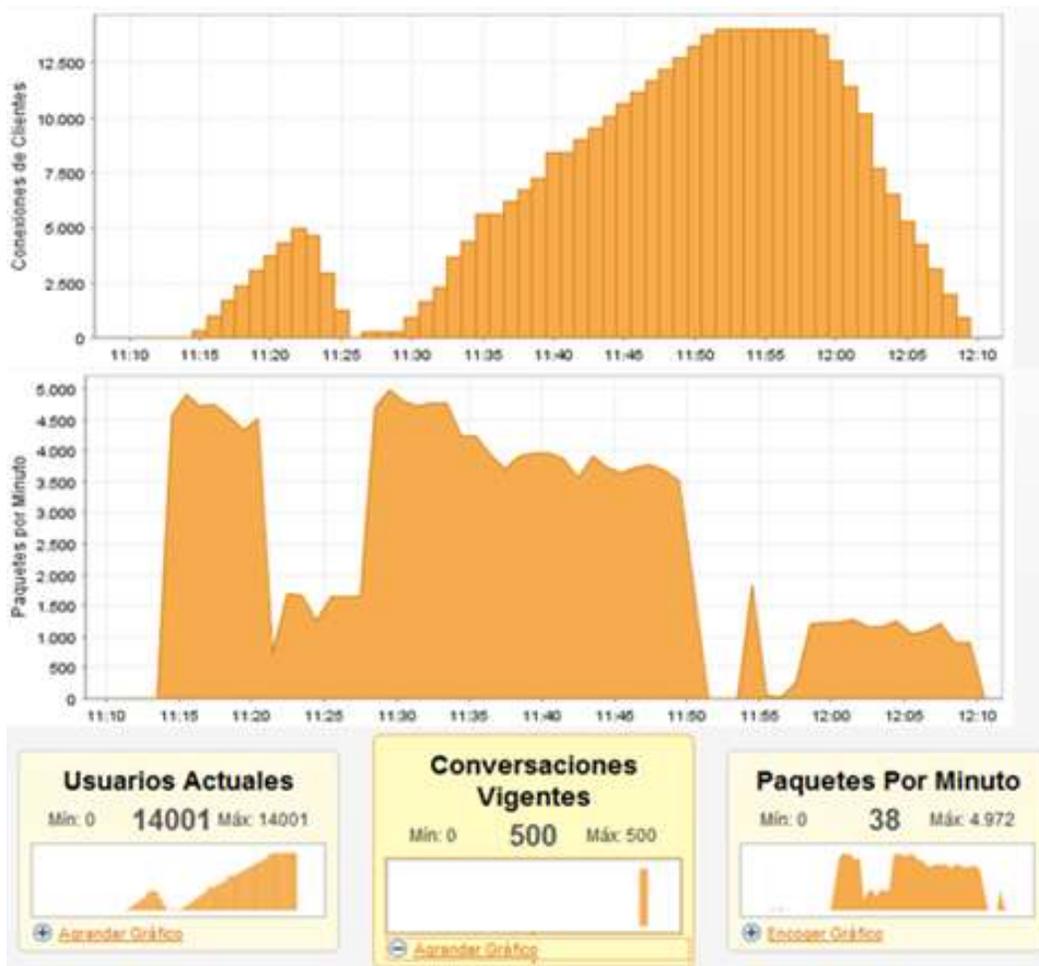


Figura 60. Estadística de servidor de mensajería instantánea

La **figura 61** muestra el comportamiento del consumo de memoria RAM durante esta prueba. La carga inicial de memoria al iniciar la prueba es del 56%, esta va aumentando en un 1% a medida que los estudiantes se hacen *login*. Al final de la prueba se ha subido

un 8% de la capacidad del servidor. Al final de la prueba, mientras los estudiantes hacen el *logout* del sistema, el consumo de memoria vuelve a descender hasta llegar al punto inicial de 56% de RAM.

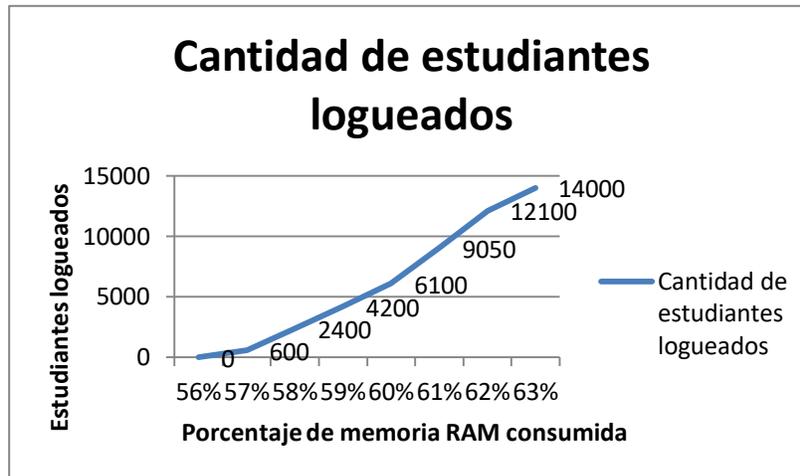


Figura 61. Comportamiento RAM con 14000 usuarios

El tiempo total de la prueba fue de aproximadamente 40 minutos, de lo que se concluye que en promedio, una petición puede ser resuelta en 105,26 milisegundos. El pico de paquetes por minuto que maneja el servidor fue de 4972. Por último es de notar, que aunque las características del equipo no son las mejores, se puede soportar sin inconvenientes los 14000 estudiantes que se crearon para esta prueba. Pero de ser necesario, si el servidor no soporta de manera eficiente el sistema, se puede optar en usar un clúster de servidores para mitigar las cargas en un solo equipo o garantizar la alta disponibilidad del servicio.

## **CAPITULO 6.**

### **Aportes, Conclusiones y Trabajo Futuro**

#### **6.1. Aportes**

Los aportes que proporciona la realización del proyecto son:

- Una arquitectura adaptada para la implementación convergente de servicios de mensajería instantánea y móvil, basada en el paradigma de integración Enterprise Service Bus, la cual provee una infraestructura troncal común, soportando mensajería distribuida, transparencia en la ubicación de los servicios, transparencia en el transporte y enrutamiento basado en contenidos.
- Un prototipo de evaluación desarrollado en Java y usando las librerías de JBoss ESB para la validación de un servicio de prueba que pueda ser implementado en la Universidad del Cauca, que permite la creación de una comunidad virtual académica.
- La descripción de los diferentes paradigmas de integración y mediadores de software, lo cual establece las funcionalidades de soporte para la implementación de un nuevo sistema con otro patrón de integración.
- La búsqueda y puesta en práctica de las principales herramientas software para llevar a cabo una implementación de servicios de mensajería que requieren ser convergentes, y el establecimiento de las funcionalidades necesarios para la escogencia de la más apropiada para el caso un estudio.
- Una plataforma con diseño e implementación modular, lo que permite que pueda ser fácilmente extensible a nuevas funciones y servicios, y que puede ser refactorizable, es decir, que se pueda cambiar detalles en su estructura interna sin cambiar su comportamiento externo.
- Un artículo técnico, el cual fue presentado en el concurso de papers del IEEE Región 9 (Latinoamérica) versión 2011, en el cual se logró obtener el segundo puesto en el área de telecomunicaciones.

#### **6.2. Conclusiones**

Las conclusiones que brinda la realización del proyecto son:

- Debido a los resultados obtenidos en el caso de estudio, la arquitectura implementada permite mantener integrados los sistemas nuevos y legados en un estilo completamente distribuido e interoperable, es decir que para usar el sistema no es necesario tener un teléfono inteligente o algún teléfono móvil de gama alta; basta con utilizar un teléfono móvil de gama media que permita mensajería SMS para ingresar al sistema.

- A partir de la escogencia de la arquitectura de bus de mensajes para suplir las necesidades de convergencia del proyecto, se pueden determinar las características arquitecturales que se requieran adicionalmente para la adaptación de los módulos de bases de datos, correo electrónico y el de aplicaciones móviles en el caso de los servicios proveedores, y los módulos de mensajería instantánea y el Gateway de mensajería para los servicios consumidores.
- Se destaca que en la arquitectura modular del sistema, la solución propuesta permite la integración de diferentes sistemas de mensajería, incluso el correo electrónico que no se consideró desde el inicio del proyecto, pero que luego tuvo una cabida importante en el mismo contribuyendo en una mayor experiencia del usuario.
- La arquitectura permite su refactorización, es decir, que se puede reestructurar su código fuente y alterar parte de su estructura interna, sin cambiar su comportamiento externo, lo cual conlleva a poder añadir módulos a través de sus respectivos adaptadores sin necesidad de vincular a otros módulos de la arquitectura.
- En los módulos se puede lograr extender sus funcionalidades con nuevos servicios consumidores o proveedores, como alarmas e incluso el uso de sus propias bases de datos, de ser necesario, para lograr un mejor desempeño y funcionalidad del prototipo.
- Debido a la refactorización también es posible la virtualización, de forma que si un equipo falla, o si se cambia de proveedor del servicio de mensajería móvil, no es necesario notificar el cambio a cada uno de los usuarios del sistema; esto contribuye a la reducción de riesgos de disponibilidad en el entorno de TI.
- Los recursos software consumidos y los recursos hardware necesarios para ejecutar un servidor de mensajería instantánea, un servidor de correo electrónico, un servidor de base de datos, un servidor de aplicaciones móviles y un gateway de mensajería móvil no son altos, y dependiendo de la capacidad hardware con la que se cuente, se pueden implementar más funcionalidades al sistema.
- Para un mejor funcionamiento del ESB, desde el punto de vista de la capacidad RAM, es mejor implementar un servidor para uso exclusivo del mismo o implementarlo en un servidor donde no estén habilitados servicios web o servicios sobre internet.
- Al determinar varios paradigmas de integración se destaca ESB ya que además de ser un middleware que se basa en SOA, permite funcionalidades como la mensajería distribuida y el soporte multiprotocolo, que permiten al bróker una mayor eficiencia de los recursos, debido a que provee una infraestructura troncal común, soportando mensajería distribuida, lo cual es necesario para proporcionar los mecanismos que lleven a cabo la convergencia de servicios.
- Luego de probar varias herramientas software y de revisar publicaciones de otras fuentes, se concluye que JBossESB por ser estable, estar bien documentado, no tener restricciones de uso y por permitir afinidad entre sus funcionalidades y los mecanismos de integración necesarios del proyecto, es la mejor para éste caso de investigación; aún así, dependiendo de los servicios y requisitos funcionales de una organización empresarial y del valor comercial que se esté dispuesto a invertir en una

herramienta software, se pueden encontrar otras herramientas mejores para obtener soluciones acordes a los requerimientos.

- Con este trabajo se comprueba la posibilidad de establecer un escenario para la creación de una comunidad académica virtual, entre los estudiantes, los profesores y los administrativos con el fin de proveer una mejor interacción entre ellos y permitir crear mejores relaciones sociales, además de permitirle a dichos usuarios manejar mejor su tiempo debido a las ofertas de comunicación.
- Puede ser necesario, si el servidor de base de datos no soporta de manera eficiente el sistema, optar en usar un clúster de servidores para mitigar las cargas en un solo equipo y/o garantizar la alta disponibilidad del servicio.

### **6.3. Trabajo Futuro**

La arquitectura del bróker de mensajería basado en ESB, permite desarrollar la convergencia de servicios de mensajería a una escala empresarial, lo cual puede ser implementado no solamente en una red organizacional como en la Universidad del Cauca, sino que también puede ser implementada y adaptada a otro tipo de organizaciones que requieran facilidades en sus servicios prestados a sus usuarios, por lo tanto otros posibles trabajos futuros provenientes de éste proyecto pueden derivarse de:

- Adaptar la arquitectura propuesta a otro tipo de entorno organizacional, mediante ESB u otro paradigma de integración, como Pub/Sub o MOM en un entorno comercial.
- El estudio de los servicios de valor agregado, a nivel de e-commerce, que se pueden implementar al poner en marcha el sistema propuesto, y obtener ingresos mediante un sistema de pago móvil.
- Implementar una comunidad virtual académica con interacción real entre los estudiantes y analizar el tráfico de la red desde el punto de vista de un operador de telefonía móvil.
- Evaluar la arquitectura con nuevas funcionalidades que le permitan dar la a conocer la seguridad en red y la calidad del servicio.
- De la relación proveniente de la distribución de la energía eléctrica con las telecomunicaciones, denominado redes inteligentes o smart grids, surge la necesidad de establecer la comunicación entre los consumidores eléctricos y sus aparatos eléctricos, con el fin de medir y usar racionalmente la energía, en donde una plataforma de integración empresarial como la propuesta, podría hacer parte del medio para llevar a cabo esta comunicación.

## Referencias Bibliográficas

[1] ZHU Zhongwen. A new IMS based community messaging service. 2009 IEEE International Conference on Intelligence in Next Generation Networks. Montreal, Quebec, Canada. [Consulta: Septiembre de 2011].

[2] FERNANDEZ, Marcelo F. Mensajería Instantánea en Internet, La nueva forma de comunicarse. 2001. Disponible en: [www.marcelofernandez.info/publicaciones/Mensajería%20Instantánea%20en%20Internet.pdf](http://www.marcelofernandez.info/publicaciones/Mensajería%20Instantánea%20en%20Internet.pdf) [Consulta: Enero de 2011].

[3] GARCÍA, Eliana. Fundación Telefónica. SMS historia, lenguaje y aplicaciones. 2009. Disponible en: <http://sociedadinformacion.fundacion.telefonica.com/url-direct/pdf-generator?tipoContenido=articulo&idContenido=2009100116310208> [Consulta: Febrero de 2011].

[4] TIBCO, The power of now. El papel de un bus de servicios empresariales (ESB) en una SOA. Disponible en: [www.willydev.net/InsiteCreation/v1.0/WillyCrawler/2008.06.08.Articulo.El%20papel%20de%20un%20bus%20de%20servicios%20empresariales%20\(ESB\)%20en%20una%20SOA.pdf](http://www.willydev.net/InsiteCreation/v1.0/WillyCrawler/2008.06.08.Articulo.El%20papel%20de%20un%20bus%20de%20servicios%20empresariales%20(ESB)%20en%20una%20SOA.pdf) [Consulta: Octubre de 2011].

[5] DUART Josep. Aprender sin distancias. 2000. Universidad Oberta de Cataluña, España. Disponible en: [www.uoc.edu/web/esp/articles/josep\\_maria\\_duart.html](http://www.uoc.edu/web/esp/articles/josep_maria_duart.html) [Consulta: Octubre de 2011].

[6] TEBAR José Antonio, NAVARRO Andrés. Implementación de una plataforma de comunicaciones unificada. En: RedIRIS. Boletín No 82 -83 (Abril 2008). Disponible en: [www.rediris.es/difusion/publicaciones/boletin/82-83/ponencia2.4D.pdf](http://www.rediris.es/difusion/publicaciones/boletin/82-83/ponencia2.4D.pdf) [Consulta: Octubre de 2011].

[7] DÍAZ José Carlos. Estudio del Protocolo XMPP de Mensajería Instantánea, de sus Antecedentes, y de sus Aplicaciones Civiles y Militares. Madrid, España, 2008, 235 h. Trabajo de Grado (Ingeniero de sistema). Universidad Politécnica de Madrid. Facultad de informática. Disponible en el catalogo en línea de la Universidad Politécnica de Madrid: <http://oa.upm.es>. [Consulta: Septiembre de 2011].

[8] Curry, Edward; Chambers, Desmond; Lyons, Gerard; "Extending Message-Oriented Middleware using Interception". Disponible en: [www.edwardcurry.org/web\\_publications/curry\\_DEBS\\_04.pdf](http://www.edwardcurry.org/web_publications/curry_DEBS_04.pdf) [Consulta: Enero de 2012].

[9] CINTEL Curso SMS Y MMS, [www.cintel.org.co/cintel/export/sites/default/cintel/inicio/capacitacion/cursoscintel/smsymms.html](http://www.cintel.org.co/cintel/export/sites/default/cintel/inicio/capacitacion/cursoscintel/smsymms.html) [Consulta: Enero de 2011].

[10] HERNÁNDEZ Haydée. El SMS y la mensajería instantánea, Marketing para jóvenes del siglo XXI. En: comunicación, ciudadanía y periodismo social [En línea]. No 3 (2006). Galicia, España. Disponible en: <http://213.60.252.126/mediateca/observatorio/xornadas/200701/comunicacions/artigo20.pdf> [Consulta: Enero de 2011].

[11] MARTÍNEZ Eustaquio Alcides. J2ME Wireless Toolkit 2.2 en un sistema de billeteaje electrónico para buses. En: Revista Científica Politécnica, Universidad Nacional del Este. No. 4 (2008). P. 19 – 34. Disponible en: [www.fpune.edu.py/docs/revista2008.pdf](http://www.fpune.edu.py/docs/revista2008.pdf) [Consulta: Septiembre de 2011].

[12] CISNEROS, Fanny Carlota, CELI, Danny Lenin. Portal de Navegación de la FIEC mediante el uso de teléfonos móviles. Guayaquil, Ecuador, 2008, 101 h. Trabajo de grado (Ingeniero en computación. Disponible en el catalogo en línea de la Escuela Superior Politécnica Del Litoral: [www.dspace.espol.edu.ec/handle/123456789/8001](http://www.dspace.espol.edu.ec/handle/123456789/8001) [Consulta: Octubre de 2011].

[13] El móvil continuará liderando el consumo de tecnologías de la comunicación [www.tendencias21.net/El-movil-continuara-liderando-el-consumo-de-tecnologias-de-la-comunicacion\\_a2764.html](http://www.tendencias21.net/El-movil-continuara-liderando-el-consumo-de-tecnologias-de-la-comunicacion_a2764.html) [Consulta: Enero de 2011].

[14] SECO Andrés. Pasarelas de mensajería SMS. 2009. Disponible en: <http://ftp.knoppix.nl/os/Linux/doc/LuCaS/Presentaciones/200103hispalinux/seco/pdf/pasarelas-sms.pdf> [Consulta: Octubre de 2011].

[15] RICHARDS, Mark. “The Role of the Enterprise Service Bus”. Disponible en: [www.infoq.com/presentations/Enterprise-Service-Bus](http://www.infoq.com/presentations/Enterprise-Service-Bus) [Consulta: Enero de 2012].

[16] DE LEUSSE Pierre, PERIORELLIS Panos, WATSON Paul. Enterprise Service Bus: An overview. Newcastle upon Tyne: University of Newcastle upon Tyne: Computing Science, 2007. University of Newcastle upon Tyne, Computing Science, Technical Report Series, No. CS-TR-1037. [Consulta: Septiembre de 2011].

[17] HÉRAULT Colombe, THOMAS Gaël, LALANDA Philippe. Mediation and Enterprise Service Bus, A position paper. En : First International Workshop on Mediation in Semantic Web Services conjunction with the 3rd International Conference on Service-Oriented Computing (1°, 2005, Amsterdam, Holanda, pag 67 – 80. [Consulta: Septiembre de 2011].

[18] ARIAS, Jorge; ESB: Enterprise Services Bus “La siguiente generación de plataformas para la integración empresarial de aplicaciones”; Novell [Consulta: Noviembre de 2011].

[19] GUINEA Alejandro, JORRÍN Sergio. Arquitectura SOA para la integración de Software libre y Software propietario en entornos mixtos. Conferencia 13. En: Jornadas de SIG libre (I: 2007, Gerona). Servicio de Sistema de Información Geográfico y teledetección 2007, I Jornadas de SIG libre. Universidad de Gerona. Gerona, España. [En línea]. Disponible en: [www.sigte.udg.es/jornadassiglibre2007/comun/1pdf/13.pdf](http://www.sigte.udg.es/jornadassiglibre2007/comun/1pdf/13.pdf). [Consulta: Mayo 3, 2010].

[20] FRANTZ Rafael. Integración de Aplicaciones: Un lenguaje específico de dominio para el diseño de soluciones de integración. Universidad de Sevilla, España: The Distributed Group ETSI Informática, 2008. 139 p. Disponible en: <[www.lsi.us.es/docs/doctorado/memorias/Zancan-Frantz-Rafael-Memoria-Investigacion.pdf](http://www.lsi.us.es/docs/doctorado/memorias/Zancan-Frantz-Rafael-Memoria-Investigacion.pdf)>. [Consulta: Abril 11, 2010].

[21] HERNÁNDEZ MARTÍNEZ Ricardo, QUINTERO HERRERA Erkin Delver. Estudio para el desarrollo de telecomunicaciones de nueva generación en Colombia. Bucaramanga, Colombia, 2008, 117 h. Trabajo de Grado (Ingeniero Electrónico) Universidad Industrial De Santander. Facultad Ingenierías Físico-Mecánicas, Escuela De Ingeniería De Sistemas E Informática. Disponible en: <<http://radiogis.uis.edu.co/gestion/Biblioteca/Articulos%20NGN/Tesis%20Pregrado%20Servivios%20NGN%20-%20Erkin%20y%20Ricardo.pdf>>. [Consulta: Febrero 15, 2010].

[22] WANG Zhenqi, GUO Ziyang, WANG Yue. Design and Implementation of Short Message Query System for Academic. En: International Colloquium on Computing, Communication, Control, and Management (2008, Guangzhou, China). ISECS International Colloquium on Computing, Communication, Control, and Management, CCCM '08. Washington, DC, USA: IEEE Computer Society, 2009. P 509 – 513. [Consulta: Marzo 11, 2010].

[23] Wordpress: Resultados del estudio de comparación de productos ESB. <<http://jpgarcia69.wordpress.com/2008/10/15/resultados-del-estudio-de-comparacion-de-productos-esb/>> [Consulta: Noviembre de 2011].

[24] Guía de ESB (Enterprise Service Bus). MSDN. Disponible en: <[http://msdn.microsoft.com/es-es/library/dd257004\(v=BTS.10\).aspx](http://msdn.microsoft.com/es-es/library/dd257004(v=BTS.10).aspx)> [Consulta: Agosto de 2011].

[25] JBOSS Inc. JBossESB 4.7, Programmers Guide. Boston: JBoss Inc. h 159. Disponible en: <[www.jboss.org/jbossesb/downloads](http://www.jboss.org/jbossesb/downloads)>. [Consulta: Mayo 15, 2011].

[26] FAGGIANO Janina Antonela, HERRERA Federico Maximiliano, PIDRE Silvana. Arquitectura de servicios semánticos ASESE. Montevideo, Uruguay, 2008, 103 h. Trabajo de grado (Ingeniero de sistemas). Universidad de la República UDELAR. Facultad de Ingeniería. Disponible en: <[www.fing.edu.uy/~pgsemws/recursos/documentos/InformeFinal.pdf](http://www.fing.edu.uy/~pgsemws/recursos/documentos/InformeFinal.pdf)>. [Consulta: Febrero 20, 2010]

[27] LIU Yan. A process modeling-based approach for web service management. En 2009 IEEE International Conference on Web Services (5, 2009, Washington, DC, USA). Proceedings of the 2009 IEEE International Conference on Web Services. Washington, DC, USA: IEEE Computer Society, 2009. P 928 – 935. [Consulta: Septiembre de 2011].

[28] GRIFFIN Donna, PESCH Dirk. A Survey on Web Services in Telecommunications. Cork Institute of Technology, Irlanda. Julio de 2007. IEEE Communications Magazine. [Consulta: Enero de 2011].

**[29]** CHUNG Sam, DAVALOS Sergio, NIIYAMA Craig, WON Daehee, BAEG Seung-Ho, PARK Sangdeok. A UML Model-Driven Business Process Development Methodology for a Virtual Enterprise using SOA & ESB. University of Washington, USA - Korea Institute of Industrial Technology, Korea. 2009 IEEE Asia-Pacific Services Computing Conference (IEEE APSCC). [Consulta: Enero de 2011].

**[30]** SCHMIDT Marc-Thomas, HUTCHISON Beth, LAMBROS Peter, PHIPPEN Rob. The Enterprise Service Bus: Making service-oriented architecture real. 2005 IBM Systems Journal, Vol 44, No 4. [Consulta: Enero de 2011].

**[31]** CAPONI Marcelo, RODRÍGUEZ Pablo, ZAMUDIO Pablo. Mensajería en sistemas de información. Montevideo Uruguay, 2008, 102 h. Trabajo de grado (Ingeniero de sistemas). Universidad de la República UDELAR. Facultad de Ingeniería. Disponible en: [http://proyecto-grado-masi.googlecode.com/files/Informe\\_Proyecto\\_Mensajeria\\_SI.pdf](http://proyecto-grado-masi.googlecode.com/files/Informe_Proyecto_Mensajeria_SI.pdf). [Consulta: Febrero de 2011].

**[32]** RUIZ Jonatan, FERNANDEZ Francisco. Integración de aplicaciones de seguridad. Vol 1. Arquitectura de Computadores. Febrero de 2009. [Consulta: Febrero de 2011].

**[33]** IEEE, "Recommended Practice for Architectural Description of Software Intensive Systems". Disponible en: <http://ieeexplore.ieee.org/Xplore/login.jsp?url=http://ieeexplore.ieee.org/iel5/7040/18957/00875998.pdf%3Farnumber%3D875998&authDecision=-203> [Consulta: Febrero de 2011].

**[34]** Gegos George, Panagiotopoulos Vasileios, Paraskevaïdis Dimitrios y Eleftherios Stefanis, "A Survey of Modern Mobile Social Applications: Services and Architectures", Department of Computer Science, Athens University of Economics and Business. Disponible en <http://nes.aueb.gr/~efstath/social.pdf> [Consulta: Enero de 2012]

**[35]** Boyd Danah y Ellison Nicole, "Social Network Sites: Definition, History, and Scholarship", School of Information University of California-Berkeley y Department of Telecommunication, Information Studies, and Media Michigan State University, 2007. [Consulta: Enero de 2012]

**[36]** DELGADO, Andrea; GONZALES, Laura y PIEDRABUENA, Federico; "Desarrollo de aplicaciones con enfoque SOA", Universidad de la Republica, Facultad de Ingeniería, Monterey Uruguay. Disponible en [www.fing.edu.uy/~fpiedrab/downloads/JIISIC06\\_SOA\\_AD\\_v6.6\\_.pdf](http://www.fing.edu.uy/~fpiedrab/downloads/JIISIC06_SOA_AD_v6.6_.pdf) [Consulta: Enero de 2011].

**[37]** BILLY Reynoso, "Arquitecturas de Software – Arquitecturas Orientadas a Servicios SOA", Universidad de Buenos Aires, Argentina. Disponible en [www.scribd.com/doc/4294229/Arquitectura-de-Software-Arquitecturas-Orientada-a-Servicios-SOA](http://www.scribd.com/doc/4294229/Arquitectura-de-Software-Arquitecturas-Orientada-a-Servicios-SOA) [Consulta: Enero de 2011].

**[38]** Isaac Gutiérrez Gómez y Salvador Otón Tortosa, "Service Oriented Architecture", Universidad del Alcalá, España. Disponible en: <http://ftp.informatik.rwth-aachen.de/Publications/CEUR-WS/Vol-132/paper09.pdf> [Consulta: Enero de 2011].

- [39] CERON Hader Ricardo, VELA Gustavo Adolfo, CAICEDO Oscar Mauricio. Arquitectura de referencia para sitios de redes sociales en ambientes móviles. Trabajo de Grado. Abril de 2009 [Consulta: Diciembre de 2011]
- [40] NAVARRO, Rafael; “REST vs Web Services”, 2007. Disponible en <<http://users.dsic.upv.es/~rnavarro/NewWeb/docs/RestVsWebServices.pdf>> [Consulta: Enero de 2011].
- [41] DE SETA, Leonardo; “Introducción a los Servicios Web RESTful”, Noviembre 2008 <[www.dosideas.com/](http://www.dosideas.com/)> [Consulta: Diciembre de 2011]
- [42] KLEIMOLA Jari, “A RESTful Interface to Mobile Phone”, Helsinki University of Technology, Finlandia. Disponible en: <[www.hiit.fi/~kleimola/net/pamp/pampREST\\_paper.pdf](http://www.hiit.fi/~kleimola/net/pamp/pampREST_paper.pdf)> [Consulta: Diciembre de 2011]
- [43] PHILIPPE Kruchten, “Architectural Blueprints – The 4+1 View Model of Software Architecture”, Rational Software Corporation. Disponible en: <[www.cs.ubc.ca/~gregor/teaching/papers/4+1view-architecture.pdf](http://www.cs.ubc.ca/~gregor/teaching/papers/4+1view-architecture.pdf)> [Consulta: Diciembre de 2011]
- [44] DE LA CRUZ Edgar, TAIMAL Hernán, CAICEDO Oscar Mauricio. Middleware de seguridad para el acceso a servicios IMS en un entorno 802.11. Trabajo de Grado. Abril de 2009 [Consulta: Diciembre de 2011]
- [45] Object Management Group – UML. Disponible en: <<http://www.uml.org>> [Consulta: Diciembre de 2011]
- [46] SERRANO, Carlos. “Modelo Integral para el Profesional en Ingeniería”. Popayán: Editorial Universidad del Cauca, 2005. [Consulta: Enero de 2012]
- [47] PostgreSQL. Disponible en: <[www.postgresql.org](http://www.postgresql.org)> [Consulta: Enero de 2012]
- [48] Java 2 Platform, Enterprise Edition - J2EE. Disponible en: <<http://java.sun.com/j2ee/overview.html>> [Consulta: Enero de 2012]
- [49] Bleecker Julian, “What’s your social doing in my mobile Design Patterns for Mobile Social Software”, Research Fellow, Annenberg Center for Communication, University of Southern California, Los Angeles. 2006. [Consulta: Enero de 2012]
- [50] Jordan Ken, Hauser Jan and Foster Steven, “The Augmented Social Network: Building identity and trust into the next-generation Internet”. [Consulta: Enero de 2012]
- [51] Boyd Danah y Ellison Nicole, “Social Network Sites: Definition, History, and Scholarship”, School of Information University of California-Berkeley y Department of Telecommunication, Information Studies, and Media Michigan State University, 2007. [Consulta: Enero de 2012]
- [52] NIELSEN, Jakob; Usabilidad, Diseño de Sitios Web, Edición 2008 [Consulta: Enero de 2012]

**[53]** SERRANO Oscar. Jabber/XMPP. Universidad de Deusto, Facultad de ingeniería ESIDE, España. Disponible en: <[www.jabberes.org/node/529](http://www.jabberes.org/node/529)> [Consulta: Enero de 2012]

**[54]** SHIGEOKA Iain. Instant Messaging In Java, the Jabber protocols. Manning Publications Co. Disponible en:  
<<http://books.google.com.co/books?id=iQnSsyMlezQC&pg=PA221&lpg=PA221&dq=roster+protocol&source=bl&ots=HVk43PYDRA&sig=r7sZeEr7zkNfjYeXkXDM9MikKQA&hl=es-419&sa=X&ei=HbAYT8OsOoatgweS7Y34Cw&ved=0CEkQ6AEwBA>>  
[Consulta: Diciembre, 2011]

**[55]** Guevara, C.; Mera, F. “Criterios para establecer políticas de seguridad de la información y plan de contingencia, caso de estudio el centro de datos de la universidad del cauca”. Tesis de Pregrado. Facultad de Ingeniería Electrónica y Telecomunicaciones, Universidad del Cauca, Popayán, Colombia, Marzo 2008. [Consulta: Enero de 2012]

**[56]** Wireshark, disponible en <[www.wireshark.org](http://www.wireshark.org)> [Consulta: Enero de 2012]

**[57]** JDK, disponible en <<http://java.sun.com>> [Consulta: Enero de 2012]

**[58]** Comcel Colombia. Disponible en: <[www.comcel.com](http://www.comcel.com)> [Consulta: Enero de 2012]

**[59]** Tigo Colombia. Disponible en: <[www.tigo.com.co/personas/internet-movil/prepago/paquetes](http://www.tigo.com.co/personas/internet-movil/prepago/paquetes)> [Consulta: Enero de 2012]

**[60]** Comisión de Regulación de Telecomunicaciones CRT, “Informe Sectorial de Telecomunicaciones”. Disponible en  
<[www.crt.gov.co/Documentos/BibliotecaVirtual/InformeSectorial/Informe\\_sectorial\\_11.pdf](http://www.crt.gov.co/Documentos/BibliotecaVirtual/InformeSectorial/Informe_sectorial_11.pdf)>  
[Consulta: Enero de 2012]

**[61]** Comisión de Regulación de Telecomunicaciones CRT, “Terminales móviles homologados en el país”. Disponible en  
<[www.crt.gov.co/Paginas/InfTecnica/TerminalesHomologados/MovilesHomologados.htm](http://www.crt.gov.co/Paginas/InfTecnica/TerminalesHomologados/MovilesHomologados.htm)>  
[Consulta: Enero de 2012]