

**CONVERGENCIA DE SERVICIOS DE MENSAJERÍA MÓVIL E INSTANTÁNEA
MEDIANTE UN ENTERPRISE SERVICE BUS**



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

Anexos

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

**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 CONTENIDO

Anexo A: Paradigmas de Integración	2
A.1. P2P	2
A.2. Publisher-Subscriber	2
A.3. EAI	3
A.4. Middleware	3
A.5. MOM	4
A.6. ESB	4
Anexo B: Herramientas Software para la Implementación de un ESB	6
B.1. GlassFish ESB	6
B.2. IBM WebSphere Message Broker	7
B.3. Microsoft Biztalk Server	8
B.4. Mule ESB	9
B.5. Oracle ESB	10
B.6. ServiceMix	11
B.7. JBossESB	12
Anexo C: Vista de Análisis para la Descripción de la Arquitectura	15
C.1. Servidor de Aplicaciones Móviles	15
C.2. Servidor Mail	16
C.3. Servidor de Bases de Datos	16
C.4. Enterprise Service Bus	17
C.5. Servidor de Mensajería Instantánea	18
C.6. Gateway de Mensajería Móvil	19
Anexo D: Diagramas UML	23
D.1. Gateway de Mensajería Móvil	23
D.1.1. Casos de uso	23
D.1.2. Diagrama de paquetes	23
D.1.3. Diagramas de clases	24
D.1.4. Diagramas de secuencia	26
D.2. Servidor de Mensajería Instantánea	29
D.2.1. Casos de uso	29
D.2.2. Diagrama de paquetes	29
D.2.3. Diagramas de clases	30
D.2.4. Diagramas de secuencia	33
D.3. Servidor Mail	36
D.3.1. Casos de uso	36
D.3.2. Diagrama de paquetes	36
D.3.3. Diagrama de clases	37
D.3.4. Diagrama de secuencia	37
D.4. Servidor de Aplicaciones Móviles	38
D.4.1. Casos de uso	38
D.4.2. Diagrama de paquetes	39
D.4.3. Diagrama de clases	39
D.4.4. Diagrama de secuencia	40
D.5. Servidor Base de Datos	41
D.5.1. Casos de uso	41
D.5.2. Diagrama de paquetes	41
D.5.3. Diagrama de clases	42
D.5.4. Diagrama de secuencia	43

Anexo E: Instalación y Configuración de Componentes	45
E.1. JBossESB	45
E.1.1. Requisitos mínimos	45
E.1.2. Instalación y configuración JDK.....	45
E.1.2. Instalación y configuración de apache ANT.....	46
E.1.3. Instalación y configuración de JBoss 4.2.2.....	46
E.1.4. Instalación y configuración de JBoss 4.7	48
E.2. Configuración del ESB de mensajería	49
E.3. Configuración del Servidor de Mensajería Instantánea.....	50
E.4. Configuración de los Servidores del Proyecto	52
E.5. Creación del Ejecutable del Cliente de Mensajería.....	52
Anexo F: Manual de Usuario.....	60
F.1. Instalación	60
F.2. Iniciando UnicaucaSpark	61
F.3. Conociendo UnicaucaSpark	61
F.4. Editar Perfil.....	62
F.5. Mensajes a Grupos	63
F.6. Agregar Contactos.....	63
Referencias Bibliográficas.....	65

Lista de Figuras

Figura 1. Arquitectura GlassFish.....	6
Figura 2. Arquitectura IBM WebSphere.....	7
Figura 3. Arquitectura BizTalk Microsoft.....	8
Figura 4. Arquitectura ESB según MULE.	10
Figura 5. Arquitectura Oracle Weblogic ESB.....	11
Figura 6. Arquitectura ServiceMix.	12
Figura 7. Arquitectura JBossESB.....	13
Figura 8. Paquetes SAM.....	15
Figura 9. Paquetes SM.....	16
Figura 10. Paquetes SBD.	17
Figura 11. Paquetes ESB.....	18
Figura 12. Paquetes SMI.....	19
Figura 13. Paquetes GMM.	21
Figura 14. Diagrama de casos de uso GMM.	23
Figura 15. Diagrama de paquetes GMM.....	23
Figura 16. Diagrama de clases GMM desde el módem GSM.....	24
Figura 17. Diagrama de clases GMM desde la conexión al SMSC.	25
Figura 18. Diagrama de secuencia SMS entrante.....	26
Figura 19. Diagrama de secuencia envío de mensaje por el módem GSM.	27
Figura 20. Diagrama de secuencia envío de mensaje por el SMSC.....	27
Figura 21. Diagrama de secuencia comportamiento de la conexión al SMSC.....	28
Figura 22. Diagrama de casos de uso SMI.....	29
Figura 23. Diagrama de paquetes SMI.....	29
Figura 24. Diagrama de clases SMI 1.	30
Figura 25. Diagrama de clases SMI 2.....	31
Figura 26. Diagrama de clases SMI 3.....	32
Figura 27. Diagrama de secuencia mensaje entrante Login.....	33
Figura 28. Diagrama de secuencia mensaje entrante Logout.....	35
Figura 29. Diagrama de secuencia mensaje entrante Lista.....	36
Figura 30. Diagrama de caso de uso SM.....	36
Figura 31. Diagrama de paquete SM.....	36
Figura 32. Diagrama de clases SM.....	37
Figura 33. Diagrama de secuencia SM.....	38
Figura 34. Diagrama de casos de uso SAM.....	38
Figura 35. Diagrama de paquetes SAM.....	39
Figura 36. Diagrama de clases SAM.....	39
Figura 37. Diagrama de secuencia SAM.....	40
Figura 38. Diagrama de casos de uso SBD.	41
Figura 39. Diagrama de paquetes SBD.....	41
Figura 40. Diagrama de clases SBD.	42
Figura 41. Diagrama de secuencia SBD.	43
Figura 42. Configuración de variables de entorno JDK.....	45
Figura 43. Prueba de versión del SDK.....	46
Figura 44. Variables de entorno para ANT.....	46
Figura 45. Verificación ANT.	46
Figura 46. Variable de entorno para JBoss.....	47
Figura 47. Ventana de inicio JBoss.....	47
Figura 48. Página de inicio JBoss.....	47

Figura 49. Variable de entorno JBossESB	48
Figura 50. Modificación en deployment-properties.....	48
Figura 51. Despliegue de ESB en JBoss	49
Figura 52. Cambio de URL de los servidores en el ESB	49
Figura 53. Cambios SMI.	50
Figura 54. Configuración del dominio SMI	51
Figura 55. Configuración SBD	51
Figura 56. Cambio servidor de mensajería	52
Figura 57. Application Info	53
Figura 58. Compiler Variables.....	53
Figura 59. Media File Options	54
Figura 60. Launcher - Executable Info	54
Figura 61. Editar versión instalador.....	55
Figura 62. Editar opciones Windows.....	55
Figura 63. Directorio de instalación.....	56
Figura 64. Máquina virtual para el instalador.....	56
Figura 65. Configuración de idioma del ejecutable.....	57
Figura 66. Eliminación de ejecutables.....	57
Figura 67. Terminar configuración media.....	57
Figura 68. Creación del ejecutable.....	58
Figura 69. Pasos para la instalación de UnicaucaSpark 1.....	60
Figura 70. Pasos para la instalación de UnicaucaSpark 2.....	60
Figura 71. Pasos para la instalación de UnicaucaSpark 3.....	61
Figura 72. UnicaucaSpark.....	61
Figura 73. Vista funcional de UnicaucaSpark.....	62
Figura 74. Editar perfil.....	62
Figura 75. Enviar mensajes a grupos.....	63
Figura 76. Agregar contacto.....	64
Figura 77. Copiar contacto de materia	64

ANEXO A

PARADIGMAS DE INTEGRACIÓN

Anexo A: Paradigmas de Integración

A continuación se listan algunos paradigmas de integración descritos de forma breve. Este anexo hace referencia a la temática del capítulo 2.

A.1. P2P

Una red peer-to-peer o red de pares o red entre iguales o red entre pares o red punto a punto (P2P, por sus siglas en inglés) es una red de computadoras en la que todos o algunos aspectos funcionan sin clientes ni servidores fijos, sino una serie de nodos que se comportan como iguales entre sí. Es decir, actúan simultáneamente como clientes y servidores respecto a los demás nodos de la red. Las redes P2P permiten el intercambio directo de información, en cualquier formato, entre los ordenadores interconectados.

El hecho de que sirvan para compartir e intercambiar información de forma directa entre dos o más usuarios ha propiciado que parte de los usuarios lo utilicen para intercambiar archivos cuyo contenido está sujeto a las leyes de copyright, lo que ha generado una gran polémica entre defensores y detractores de estos sistemas.

Las redes peer-to-peer aprovechan, administran y optimizan el uso del ancho de banda de los demás usuarios de la red por medio de la conectividad entre los mismos, y obtienen así más rendimiento en las conexiones y transferencias que con algunos métodos centralizados convencionales, donde una cantidad relativamente pequeña de servidores provee el total del ancho de banda y recursos compartidos para un servicio o aplicación.

Dichas redes son útiles para diversos propósitos. A menudo se usan para compartir ficheros de cualquier tipo (por ejemplo, audio, vídeo o software). Este tipo de red también suele usarse en telefonía VoIP para hacer más eficiente la transmisión de datos en tiempo real.

La eficacia de los nodos en el enlace y transmisión de datos puede variar según su configuración local (cortafuegos, NAT, routers, etc.), velocidad de proceso, disponibilidad de ancho de banda de su conexión a la red y capacidad de almacenamiento en disco [1].

A.2. Publisher-Subscriber

Publish-Subscribe (o pub/sub) es un patrón de mensajería, donde los remitentes (publishers) de los mensajes no programan los mensajes que se envían directamente a receptores específicos (subscribers). Más bien, los mensajes publicados se caracterizan en clases, sin el conocimiento de lo que, si es el caso, existan suscriptores. Los suscriptores expresan su interés en una o más clases, y solo se reciben mensajes que son de interés, sin el conocimiento de lo que, en su caso, existan los editores. Esta disociación de los editores y los suscriptores pueden permitir una mayor escalabilidad y una topología de red más dinámica.

Pub/sub es un hermano del paradigma queue de mensajes, y por lo general una parte de un largo sistema de middleware orientado a mensajes. La mayoría de los sistemas de mensajería soporta tanto el pub/sub y los modelos de colas de mensajes en su API, por ejemplo, Java Message Service (JMS) [2].

A.3. EAI

La integración de aplicaciones empresariales o EAI (siglas en inglés de enterprise application integration) o se define como el uso de software y principios de arquitectura de sistemas para integrar un conjunto de aplicaciones dentro de cualquier empresa.

Es el proceso de conectar las aplicaciones con otras para intercambiar información operativa o financiera. Cuando dichos sistemas no pueden compartir su información efectivamente se crean cuellos de botella que requieren de la intervención humana en la forma de toma de decisiones o en el ingreso mismo de la información. Con una arquitectura EAI correctamente implementada, las organizaciones pueden enfocar la mayoría de sus esfuerzos en la creación de competencias que generen valor en lugar de enfocarse en la coordinación de labores operativas.

Durante varias generaciones, los sistemas de las empresas han servido para un propósito específico a un único usuario o grupo de usuarios, los cuales actúan como la interfaz de dicho sistema con el resto de la organización, con lo cual se ha limitado su conexión con otros sistemas modernos o más amplios en la empresa, más aún por la creciente demanda de las empresas por compartir datos y usarlos en sus procesos sin tener que realizar cambios en sus aplicaciones o en sus estructuras de datos.

Uno de los retos que encaran las organizaciones modernas es dar a sus empleados información completa en tiempo real. Muchas de las aplicaciones en uso actualmente se apoyan en tecnologías antiguas, por lo cual esos sistemas enfrentan dificultades a la hora de mover esta información entre las aplicaciones.

EAI, como una disciplina, busca solventar muchos de esos problemas, así como crear nuevos paradigmas para ciertamente mejorar las organizaciones, tratando de trascender el objetivo de conectar las aplicaciones individuales para buscar ser un mecanismo de incrementar el conocimiento de la organización y crear ventajas competitivas futuras a la empresa [3].

A.4. Middleware

El middleware es un software de conectividad que ofrece un conjunto de servicios que hacen posible el funcionamiento de aplicaciones distribuidas sobre plataformas heterogéneas. Funciona como una capa de abstracción de software distribuida, que se sitúa entre las capas de aplicaciones y las capas inferiores (sistema operativo y red). El middleware abstrae de la complejidad y heterogeneidad de las redes de comunicaciones subyacentes, así como de los sistemas operativos y lenguajes de programación, proporcionando una API para la fácil programación y manejo de aplicaciones distribuidas. Dependiendo del problema que resolver y de las funciones necesarias, serán útiles diferentes tipo de servicios de middleware.

Por lo general el middleware del lado cliente está implementado por el sistema operativo subyacente, el cual posee las bibliotecas que implementan todas las funcionalidades para la comunicación a través de la red [4].

A.5. MOM

Middleware orientado a mensajes (MOM) es la infraestructura de software o hardware centrada en el envío y recepción de mensajes entre sistemas distribuidos. MOM permite a los módulos de aplicación distribuirse a través de plataformas heterogéneas, y reduce la complejidad del desarrollo de aplicaciones que abarcan múltiples sistemas operativos y protocolos de red mediante el aislamiento del desarrollador de la aplicación de los detalles del sistema operativo e interfaces de red. APIs que se extienden a través de diversas plataformas y redes suelen ser proporcionados por MOM.

MOM es un software que reside en las dos partes de una arquitectura cliente/servidor y por lo general soporta las llamadas asincrónicas entre las aplicaciones cliente y servidor. MOM reduce la participación de los desarrolladores de aplicaciones con la complejidad de la naturaleza master-slave del mecanismo cliente/servidor [5].

A.6. ESB

Un bus de servicios de empresa (ESB) 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. Los desarrolladores normalmente implementan un ESB utilizando tecnologías de productos de infraestructura de middleware que se basan en normas reconocidas.

Un ESB generalmente proporciona una capa de abstracción construida sobre una implementación de un sistema de mensajes de empresa que permita a los expertos en integración explotar el valor del envío de mensajes sin tener que escribir código. Al contrario que sucede con la clásica integración de aplicaciones de empresa (EAI) que se basa en una pila monolítica sobre una arquitectura hub and spoke, un bus de servicio de empresa se construye sobre unas funciones base que se dividen en sus partes constituyentes, con una implantación distribuida cuando se hace necesario, de modo que trabajen armoniosamente según la demanda.

Un ESB no implementa en sí mismo una arquitectura orientada a servicios (SOA), sino que proporciona las características mediante las cuales sí se puede implementar. Un ESB debería basarse en normas y proporcionar flexibilidad, dando cobertura a distintos medios de transporte que sean capaces de implementar tanto patrones de SOA tradicionales como arquitectura de negocios con una SOA 2.0 enriquecida. El ESB trata de aislar el acoplamiento entre el servicio solicitado y el medio de transporte. La mayoría de los proveedores de ESB incorporan principios de SOA y permiten formatos de mensaje independientes [6].

ANEXO B
HERRAMIENTAS SOFTWARE PARA LA
IMPLEMENTACIÓN DE UN ESB

Anexo B: Herramientas Software para la Implementación de un ESB

Todo producto va evolucionando al largo de su existencia y durante el tiempo transcurrido del desarrollo del proyecto, se investigó y probó herramientas ESB, de las cuales algunas han cambiado pero otras siguen igual. Teniendo en cuenta desde el primer momento que es una tecnología incipiente, a continuación se describen las más importantes. No se han hecho las mismas pruebas sobre todas las herramientas software, pero si se hicieron algunas pruebas comunes [7] [8]. Este anexo hace referencia a la temática del capítulo 2.

B.1. GlassFish ESB

Nace de la unión de un ESB existente (OPEN ESB), el Application Server de SUN (GLASSFISH) y el entorno de desarrollo integrado de SUN (NETBEANS). Desde esta unión se ha separado una versión comercial de una versión de la comunidad, por lo cual solo se da soporte sobre la versión comercial.

La meta de proporcionar una versión gratuita es para que la comunidad vaya testeando la versión y creando módulos nuevos. Aunque ya tiene muchos creados quieren proporcionar muchos más y aprovecharse de las ideas de la comunidad Open Source. Respecto a esto han dejado claro que solo se dará soporte a los módulos hechos por SUN. Algunas de las propiedades destacables son:

- Ligero, ágil y pequeño.
- Tiene una arquitectura Microkernel para conectar multitud de componentes.

La última release encontrada data de noviembre de 2009. Este ESB se destaca por tener un conjunto de partners, entre los cuales Bostech Corporation que tiene un ESB propio y un grupo de consultoras. Además tiene el apoyo de SUN con la unión con su Application Server. Además existe otro proyecto ligado a GlassFish ESB que es muy interesante, FUJI Milestone. Este será el nuevo interfaz del ESB, y servirá para configurarlo más fácilmente [9].

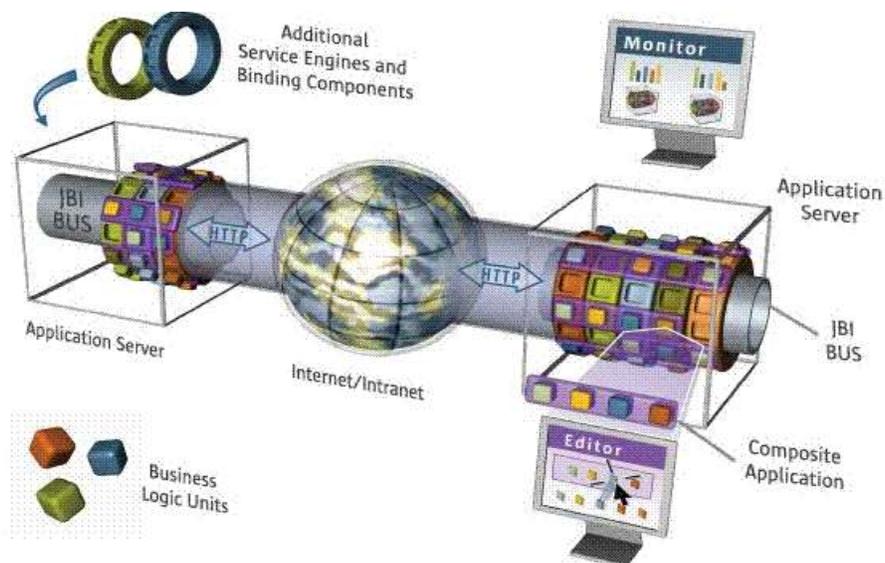


Figura 1. Arquitectura GlassFish

B.2. IBM WebSphere Message Broker

IBM es una empresa líder de consultoría, y como tal parece que tiene cierto interés en que sus aplicaciones no sean completamente abiertas. La autentica compatibilidad del ESB es con las aplicaciones de WebSphere de la propia IBM.

Este proporciona un enfoque inteligente a la arquitectura orientada a servicios, ofreciendo conectividad basada en estándares y una solución de integración que le permite crear y desplegar rápida y fácilmente interacciones entre aplicaciones y servicios, con un número reducido de interfaces menos complicadas.

Ofrece herramientas fáciles para usar que requieren habilidades de programación mínimas y es fácil de instalar, configurar, crear y gestionar. Además da soporte a muchas soluciones ISV a través de WebSphere Adapters.

Ofrece liderazgo en estándares de arquitectura orientada a servicios para composición, mediación y alojamiento de servicios. Aumenta la agilidad y la flexibilidad empresarial pudiéndose ampliar fácilmente a un modelo ESB federado.

Se puede volver a configurar de forma dinámica para que satisfaga las cargas de procesos empresariales en constante evolución. Proporciona fáciles con cualquier aplicación JMS y HTTP.

Se integra a la perfección con la plataforma WebSphere, así como con productos de la arquitectura orientada a servicios de IBM. Los sistemas operativos admitidos: HP Unix, Linux, Sun Solaris, Windows [9].

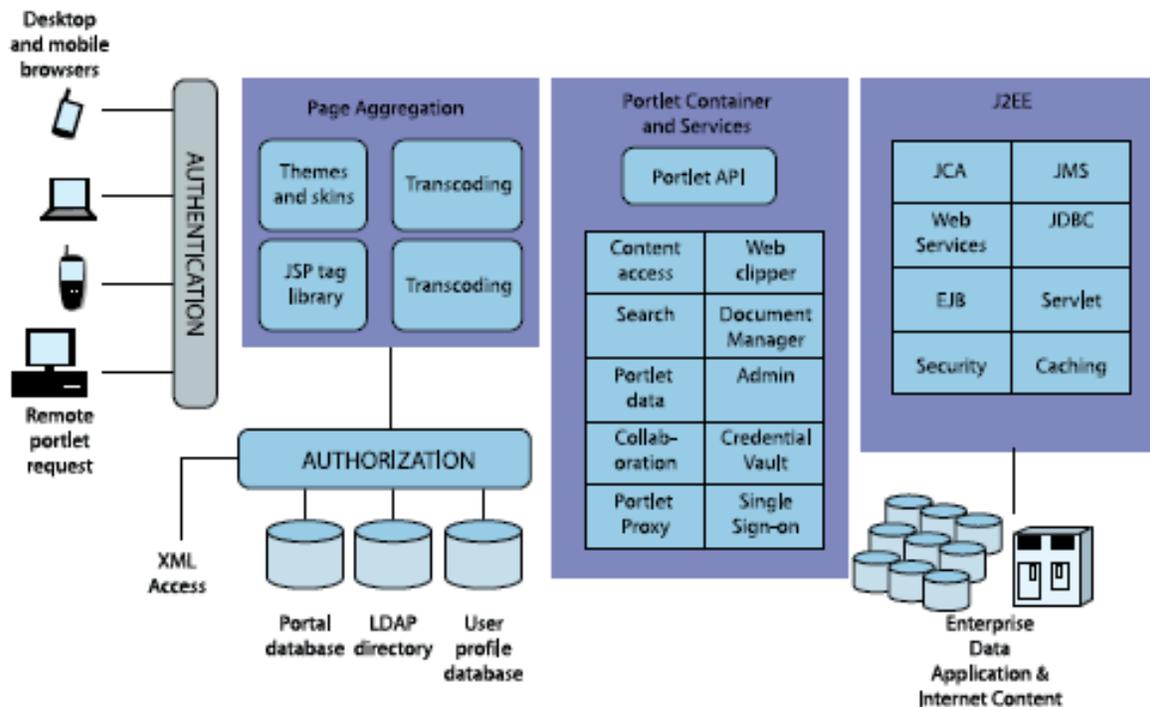


Figura 2. Arquitectura IBM WebSphere

B.3. Microsoft Biztalk Server

Microsoft para ESB crea BizTalk Server 2009 para admitir una arquitectura de mensajería de acoplamiento flexible. BizTalk Server incluye un mecanismo de publicación y suscripción muy eficaz para aplicaciones de mensajería que funciona creando y rellenando suscripciones. De esta forma, proporciona una plataforma muy eficaz y escalable para aplicaciones de arquitectura orientada a servicios (SOA).

Microsoft amplía la funcionalidad de BizTalk Server a fin de proporcionar una serie de nuevas capacidades enfocadas a la creación de aplicaciones robustas, conectadas y orientadas a servicios que incorporan la invocación de servicios basada en el itinerario para composición ligera de servicios, la resolución dinámica de los extremos y las asignaciones, el servicio web y la integración WS, la administración y generación de informes de errores, así como la integración con soluciones de gobierno SOA de terceros [10].

La combinación de distintos sistemas en procesos empresariales efectivos supone todo un desafío. Por consiguiente, BizTalk Server incluye una amplia variedad de tecnologías. La siguiente ilustración muestra los componentes más destacados del producto.

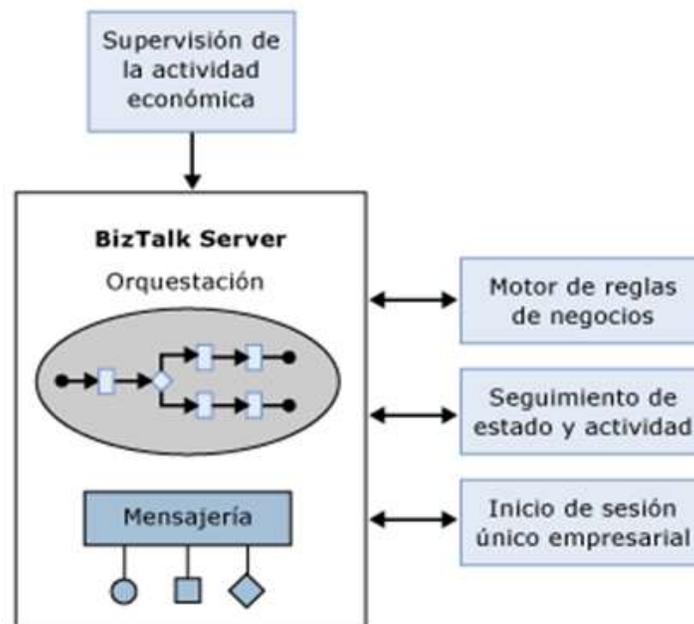


Figura 3. Arquitectura BizTalk Microsoft

Tal y como sugiere la ilustración, el motor de BizTalk Server es el componente principal del producto. Este motor cuenta con dos piezas principales:

- Un componente de mensajería que proporciona la capacidad de establecer una comunicación con otros tipos de software. Dado que está basado en adaptadores para distintos tipos de comunicación, el motor admite una amplia gama de protocolos y formatos de datos, como, por ejemplo, los servicios Web, entre otros.
- Un soporte para la creación y ejecución de procesos definidos gráficamente, las orquestaciones. Las orquestaciones, integradas sobre los componentes de

mensajería del motor, implementan la lógica que dirige un proceso empresarial completo o parte de éste.

Además, se pueden usar otros componentes de BizTalk junto con el motor, incluidos los siguientes:

- Un motor de reglas de negocios que evalúa conjuntos complejos de reglas.
- Un concentrador de grupo que permite que los programadores y administradores supervisen y administren el motor y las orquestaciones que este ejecute.
- Una función de inicio de sesión único (SSO) empresarial que proporciona la capacidad de asignar datos de autenticación entre Windows y otros sistemas ajenos.

Además de todo esto, BizTalk Server incluye la supervisión de la actividad económica, utilizada por los trabajadores para supervisar los procesos empresariales en ejecución. El tipo de información que se muestre dependerá de las preferencias de los usuarios empresariales y se expresará en términos empresariales, en lugar de técnicos.

BizTalk Server utiliza también las últimas versiones del resto de las tecnologías de Microsoft. Se basa en la versión 3.5 de .NET Framework, por ejemplo, y las herramientas de los programadores se hospedan en Microsoft Visual Studio 2008. Por lo que respecta al almacenamiento, BizTalk Server puede utilizar SQL Server 2008, la última versión del producto estrella de base de datos de Microsoft. También se puede usar BizTalk Server 2009 en servidores Windows de 64 bits, aprovechando así la memoria de mayor capacidad y el resto de las ventajas que ofrece esta nueva generación de hardware [11].

B.4. Mule ESB

Ofrece una aplicación neutral que se puede integrar con las aplicaciones más convenientes, por esta razón es mucho más ligero que la mayoría de los que existen en el mercado. Estos están montados sobre aplicaciones de la misma compañía, como por ejemplo los comerciales.

Especialmente este software se destaca por tener muchos proyectos a su alrededor, casi todos dedicados al desarrollo de componentes para darle más funcionalidades, entre ellos contiene una herramienta administrativa. Esta es muy útil porque la versión estándar del software no contiene consola.

Otras propiedades que se destacan son:

- Basado en framework de mensajería JAVA para la conexión e intercambio de datos.
- Arquitectura orientada a servicios.
- Integración de una gran cantidad de protocolos.
- Basado en las ideas de la arquitectura del Enterprise Service Bus.
- Facilidad para añadir nuevos componentes.
- Posibilidad de cualquier formato en los mensajes: SOAP, image, etc
- Reusabilidad de componentes.
- Mantienen una versión Enterprise (comercial).

La última relase que se ha encontrado data en Diciembre del 2008. El equipo que se dedica al desarrollo de MULE es de Estados Unidos, Malta y de Reino Unido. No se ha encontrado ninguna relación directa con empresas comerciales del sector, pero se conoce que hay potentes inversoras detrás [9].



Figura 4. Arquitectura ESB según MULE

B.5. Oracle ESB

Hace poco que ORACLE compro BEA y se hizo suyos todos los productos. Oracle ya contaba con algunos productos propios, aunque casi todos los ha sustituido o unido con los de BEA. El ESB que tenía Oracle era más bien simple, en cambio el de BEA es muy completo. Hace un tiempo que aun se podía instalar el ESB de BEA, ahora ya está a nombre de Oracle y con la única peculiaridad del cambio de logos y de nombre.

A este ESB se le podrían atribuir muchas propiedades pero las más destacables son:

- Rápida configuración con reglas básicas sin necesidad de código.
- Transformaciones básicas basadas en XQuery o XSLT, aplicadas dinámicamente.
- Publicación de servicios automatizados y dirigidos por la administración.
- Soporte de múltiples protocolos y estándares.
- Posibilidad de crear transportes propios.
- Configuración totalmente ajustable según parámetros del mensaje.

Por otra parte cabe destacar la consola de administración. Controla todos los cambios incluyendo sesiones para saber en todo momento quien está modificando las configuraciones. Se pueden parar y encender servicios de manera totalmente dinámica. Incluso se pueden testear configuraciones desde el punto que sea necesario [9].

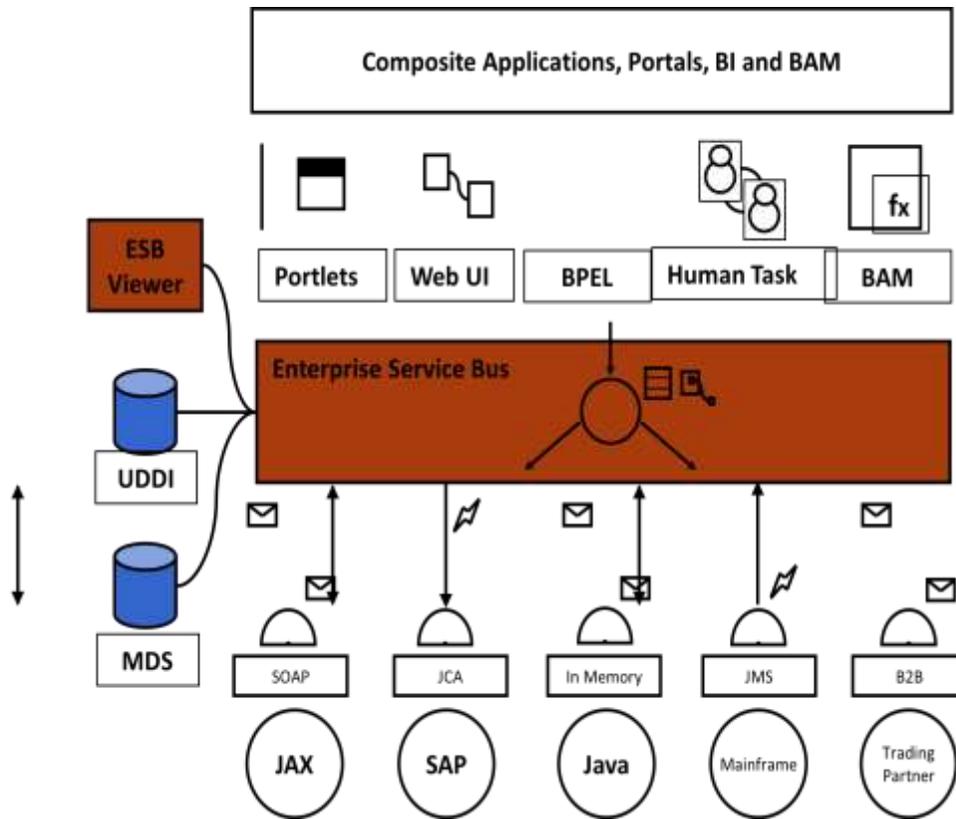


Figura 5. Arquitectura Oracle Weblogic ESB

B.6. ServiceMix

Es uno de los dos ESB de código libre que corren bajo la licencia Apache. Existen muchas empresas que tienen sus productos Open Source enmascarados bajo esta licencia, y hay muchos grupos de desarrolladores que se han ido a desarrollar sus proyectos bajo Apache.

Lo definen como un ESB ágil, por combinar SOA con EDA. Como se conoce éste tendría que ser un requisito básico, en cambio tanto éste como muchos otros lo ponen como una propiedad especial llamada ágil.

Utiliza un gestor de colas Active MQ, otro software Apache. Con él distribuyen los errores, ofrecen fiabilidad y control remoto. Como en el caso de Mule hacen hincapié en que es una arquitectura sin ataduras y se puede montar sobre diferentes arquitecturas mayores.

La última relase encontrada data de Diciembre de 2008. El equipo que se dedica al desarrollo de ServiceMix destaca la inclusión de gente que está en proyectos de IBM e IONA. Como es un proyecto Open Source de la comunidad Apache se ve como las empresas que hay detrás ya no solo financian el proyecto sino que incluyen su personal para nutrirse del conocimiento adquirido [9]. Otras de las propiedades que destacan son:

- Ha integrado SPRING
- Soporta EJB y POJO
- Se puede personalizar y así acomodarlo a las necesidades.

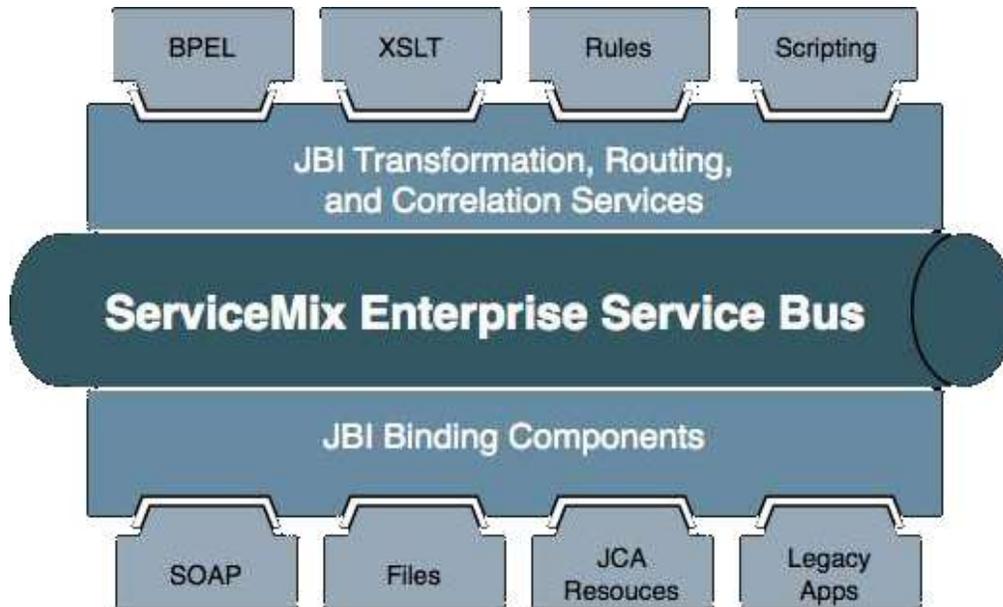


Figura 6. Arquitectura ServiceMix

B.7. JBossESB

JBossESB es la próxima generación de EAI, por lo tanto, tiene muchas de las capacidades similares a las ofertas existentes de EAI: Monitoreo de procesos de negocio, entorno de desarrollo integrado, interfaz humana de flujo de trabajo del usuario, gestión de procesos de negocios, conectores, administrador de transacciones, la seguridad, contenedor de aplicaciones, servicio de mensajería, repositorio de metadatos, servicio directorio y arquitectura de computación distribuida. Además JBossESB es parte de un SOI (Infraestructura Orientada a Servicios).

Sin embargo, SOA no es simplemente una tecnología o un producto: es un estilo de diseño, con muchos aspectos (como la arquitectura, metodológicos y organizativos) que no guardan relación con la tecnología actual [12]. Características:

- Apoyo para el marco de la notificación general. Transportes soportados incluyen JMS (HornetQ, JBoss Messaging, MRG, JBossMQ, AQ Oracle y MQSeries), InVm, TCP/IP del sistema, correo electrónico, base de datos o archivos. JBoss mensajería 1.4.5GA ahora es el que recomendamos la implementación de JMS.
- JMS y la integración de transacciones de SQL.
- Más integración cuando se despliega en JBossAS.
- Integración jBPM.
- Servicios Web de apoyo (JBossws nativos y JBossws cxf).
- Implementación de la mejora y la configuración, usando un determinado servidor ESB.
- Soporte Groovy.
- Muchos ejemplos de inicio rápido para empezar.
- Apoyo a las transformaciones de datos utilizando Smooks 1.2.5 o XSLT.
- Oyentes y el modelo de acción en apoyo acoplamiento suelto de los pasos de interacción.
- Enrutamiento basado en contenido utilizando Drools, regexp o XPath.

- Apoyo para los registros, utilizando JAX-R y jUDDI fuera de la caja. Ahora el apoyo a la SOA y registro UDDI también.
- Pasarelas para permitir el tráfico de ESB no cuenta con el flujo en la ESB.
- Editor de configuración gráfica.
- Alto rendimiento y fiabilidad.

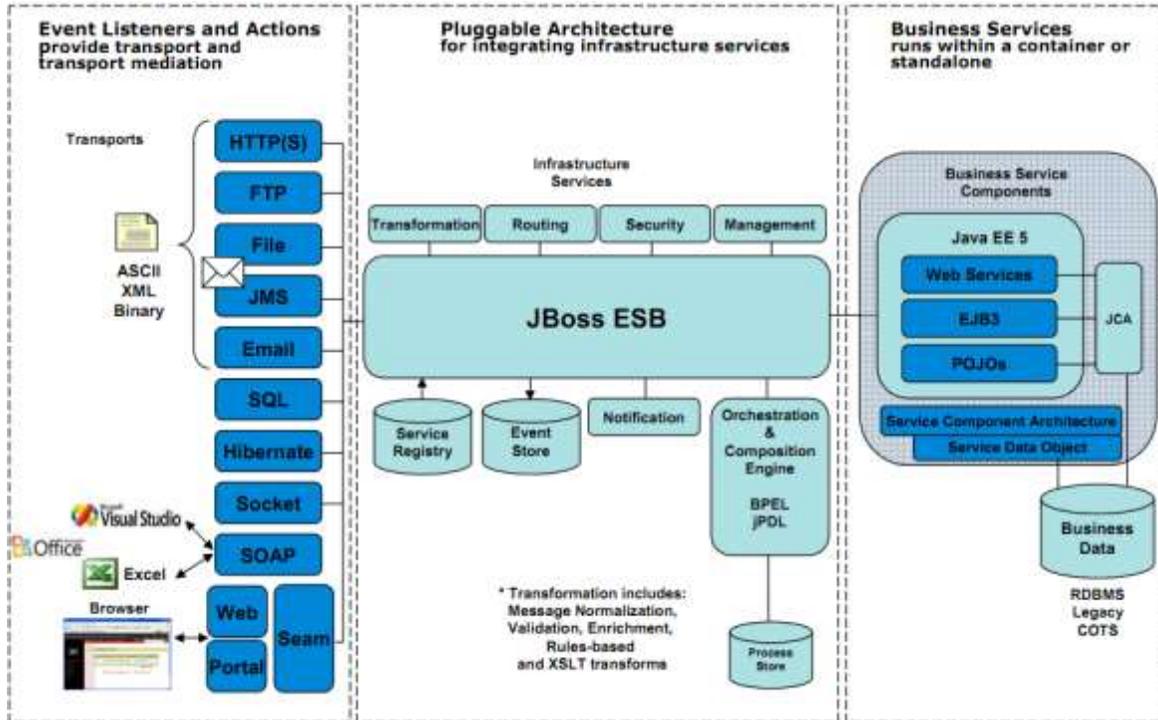


Figura 7. Arquitectura JBossESB

ANEXO C
VISTA DE ANÁLISIS PARA LA DESCRIPCIÓN DE LA
ARQUITECTURA

Anexo C: Vista de Análisis para la Descripción de la Arquitectura

Las clases obtenidas para cada módulo de la arquitectura fueron agrupadas en diferentes paquetes clasificados de acuerdo a un grupo de niveles según su funcionalidad, los cuales se obtuvieron a partir de cada uno de los diagramas detallados de la vista modular y sus relaciones. Esta organización de paquetes representa la composición genérica a nivel de análisis de la arquitectura planteada [13]. Este anexo hace referencia a la temática del capítulo 3.

C.1. Servidor de Aplicaciones Móviles

Nivel de Red

- **Compuertas enlace ESB:** En este módulo se encuentran las compuertas que permiten la comunicación con el ESB_M. Se crean dos enlaces, uno de entrada por el cual se recibe todos los mensajes del ESB y uno de salida por donde se responde de manera asíncrona a cada una de estas solicitudes.

Nivel de servicios

- **Lógica de mensajes:** El ESB_M dirige mensajes provenientes del GMVL (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 GMVL. 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 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.

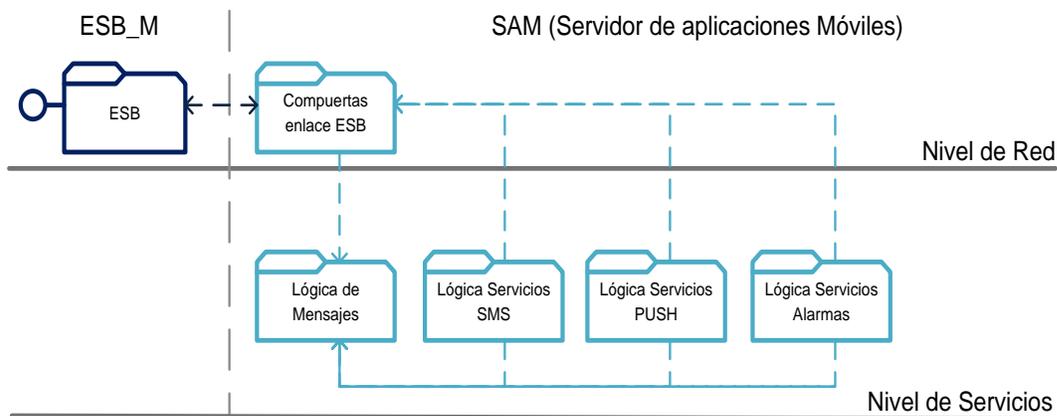


Figura 8. Paquetes SAM

C.2. Servidor Mail

Nivel de red

- **Compuertas de enlace ESB:** En este módulo se encuentra la compuerta que permite escuchar el ESB_M. Este módulo no utiliza ningún servicio expuesto por otro componente, por ésta razón el conector se encuentra diseñado para recibir peticiones.
- **Envío mail:** Este componente permite la composición de correos electrónicos y el envío de éstos. Recibe del ESB la configuración del cliente de correo electrónico (username, password, host, puerto) para ser enviado el mensaje hacia el destino a través de él.

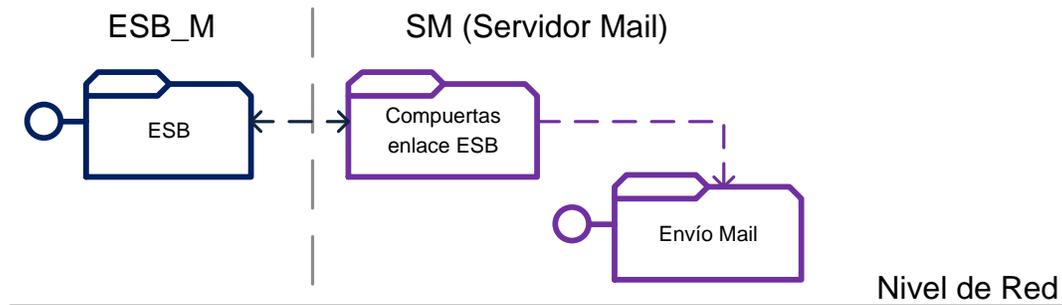


Figura 9. Paquetes SM

C.3. Servidor de Bases de Datos

Nivel de red

- **Compuertas de enlace ESB:** En este módulo se encuentran dos compuertas que permiten la comunicación con el ESB_M. Una compuerta expuesta a las peticiones de todos los demás servicios, escucha del ESB todos los requerimientos de información para ser enviados a resolver, y la otra envía las respuestas hacia los demás servicios mediante el ESB.

Nivel de servicios

- **Consulta BD:** Recibe la petición hecha por los demás servicios, la resuelve, genera la consulta de los datos y retorna la información para ser enviada al destino.

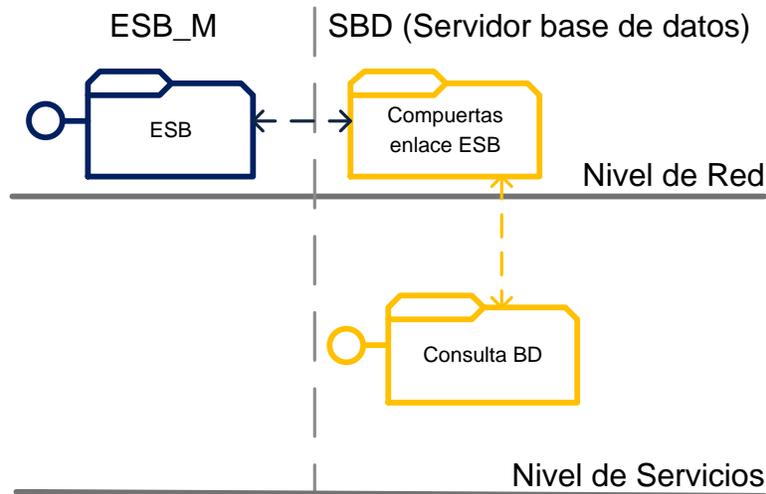


Figura 10. Paquetes SBD

C.4. Enterprise Service Bus

Nivel de red

- **Compuertas de enlace ESB:** Contiene las compuertas de entrada y salida hacia los demás módulos, haciendo del ESB el mediador de solicitudes entre servidores, permitiendo la visibilidad requerida para solicitar cualquier petición dentro de ésta arquitectura.

Nivel de servicios

- **Adaptación y transformación:** Recibe la petición para ser resuelta, en caso de ser requerido. Se encarga de complementar la información de un mensaje o de cambiar el formato del mensaje para ser recibido correctamente en el módulo al que va dirigido.
- **Enrutamiento:** Verifica la información del mensaje para decidir hacia que servidor debe dirigirse, utilizando las compuertas de enlace.

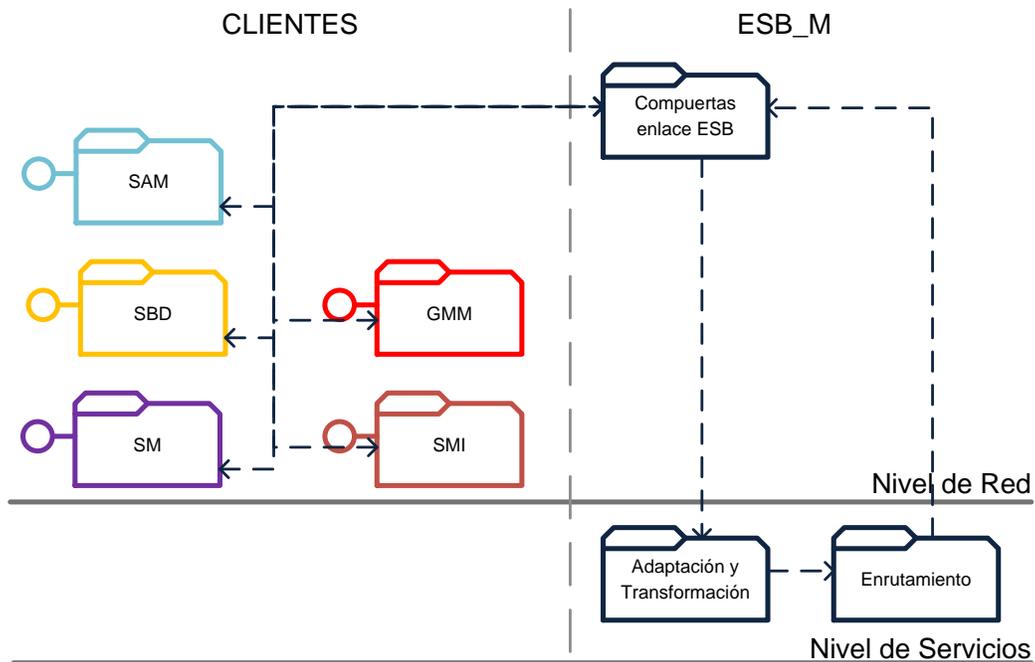


Figura 11. Paquetes ESB

C.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.

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.

Nivel de datos

- **Control de usuarios:** Realiza solicitudes al ESB para la obtención de datos de un usuario.

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.

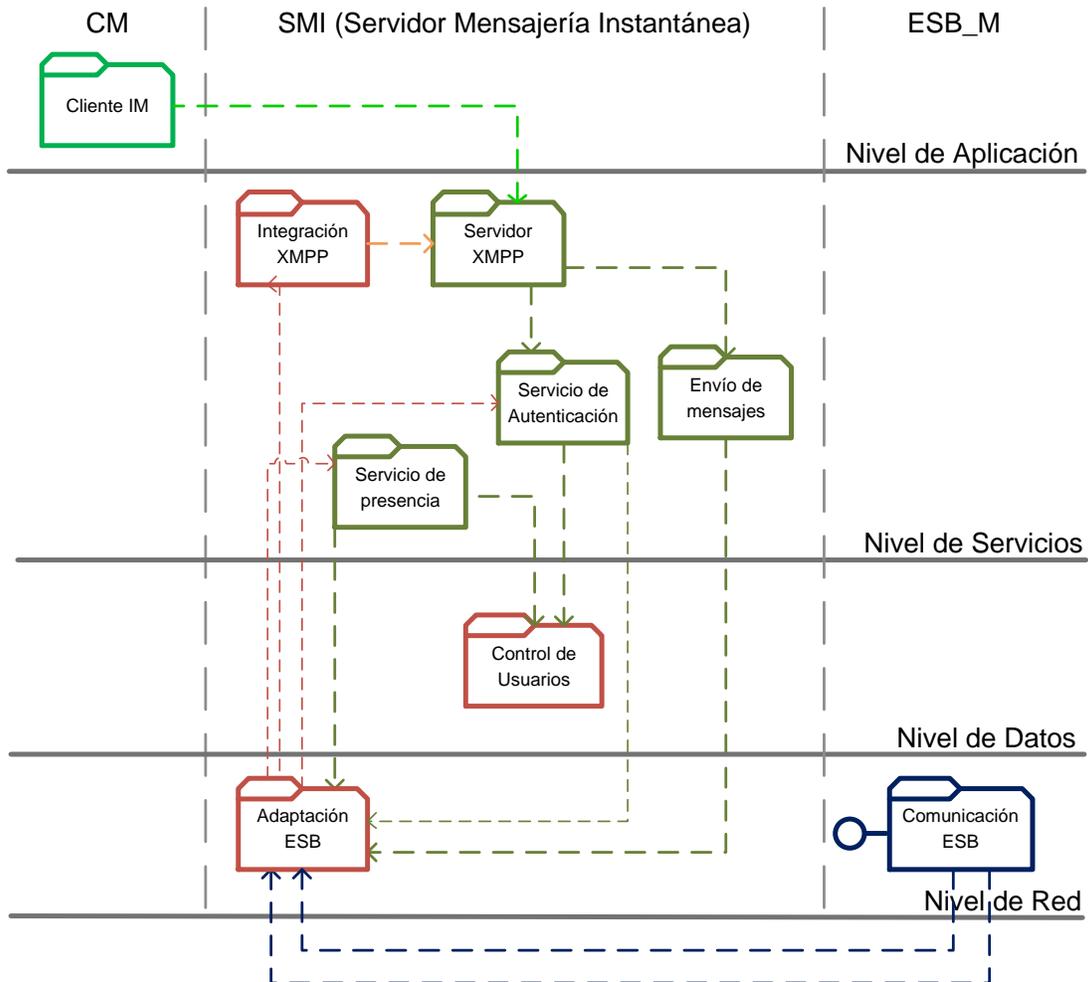


Figura 12. Paquetes SMI

C.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.

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 módem:** 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.

Nivel de datos

- **Control de mensajes:** Es el encargado de guardar registros para el control de los mensajes SMS enviados y recibidos.

Nivel de red

- **Control de adaptación ESB:** Crea la comunicación asíncrona y contiene las compuertas de escucha con el ESB.

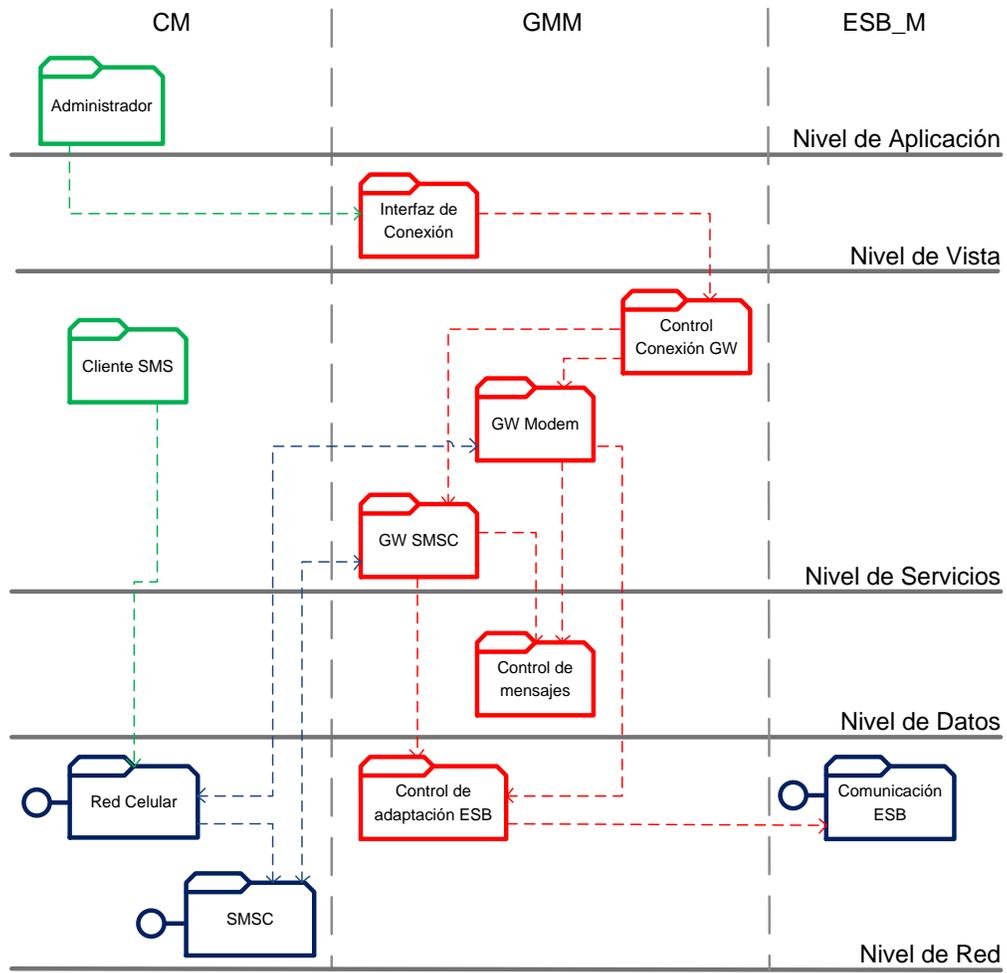


Figura 13. Paquetes GMM

ANEXO D

DIAGRAMAS UML

Anexo D: Diagramas UML

A continuación se presentan los diagramas UML referentes al proyecto [14]. Este anexo hace referencia a la temática del capítulo 4.

D.1. Gateway de Mensajería Móvil

D.1.1. Casos de uso

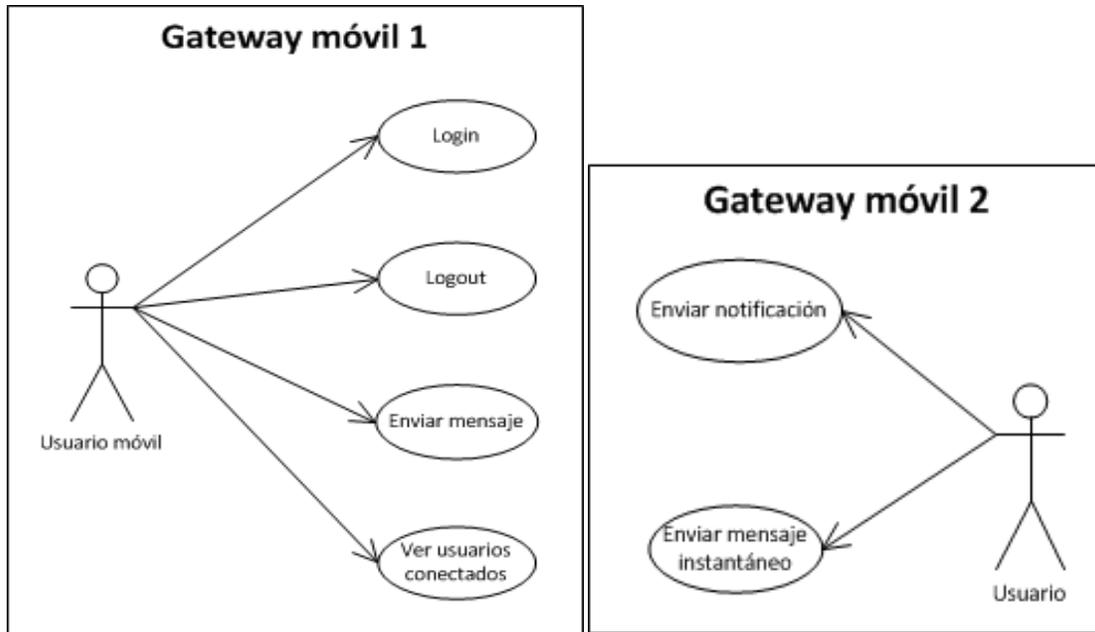


Figura 14. Diagrama de casos de uso GMM

El actor Usuario describe el cliente que puede ser tanto un Usuario Móvil o un Usuario Web.

D.1.2. Diagrama de paquetes

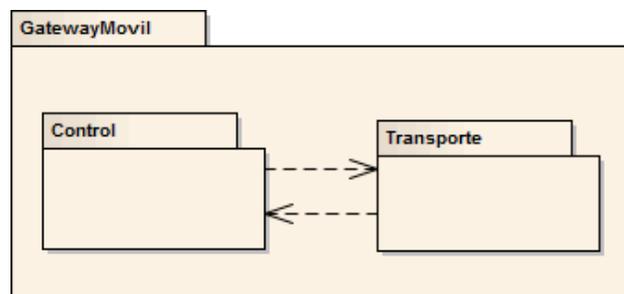


Figura 15. Diagrama de paquetes GMM

D.1.3. Diagramas de clases

Gateway GSM:

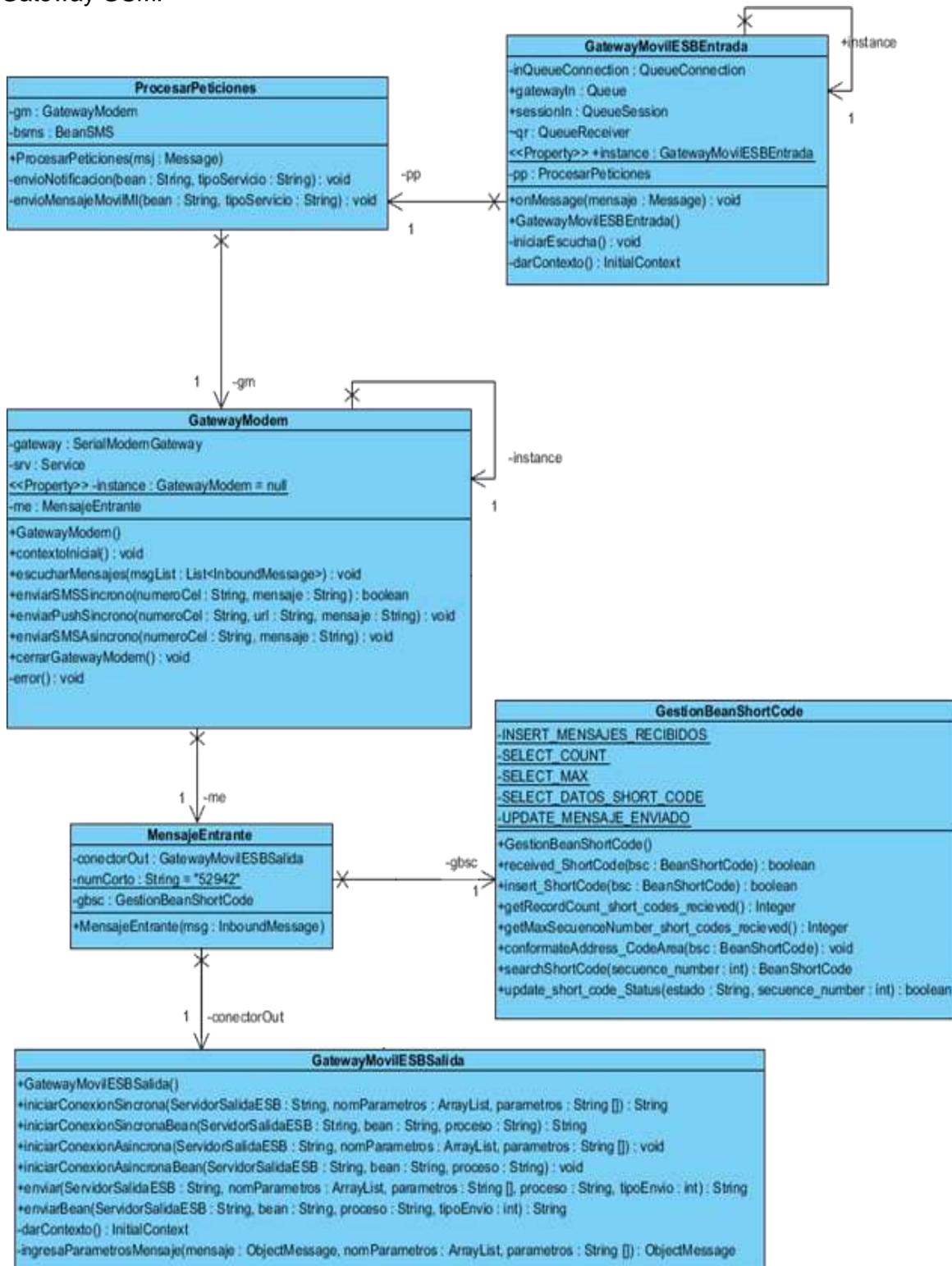


Figura 16. Diagrama de clases GMM desde el módem GSM

Conexión SMSC:

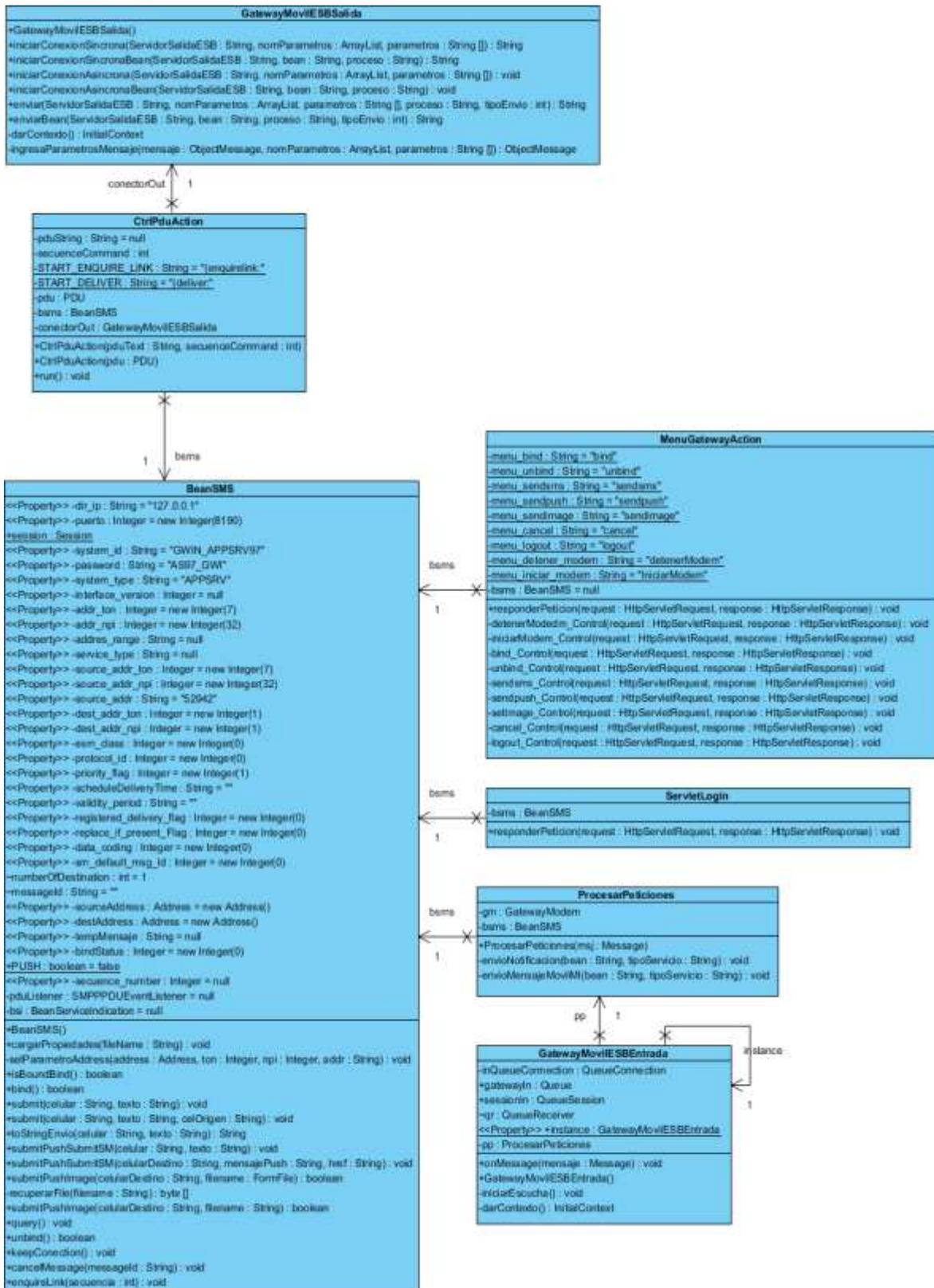


Figura 17. Diagrama de clases GMM desde la conexión al SMSC

D.1.4. Diagramas de secuencia

D.1.4.1. SMS Entrante

La figura 48, muestra el diagrama de secuencia general para un SMS entrante, ya sea por la conexión con el SMSC o por el Modem GSM.

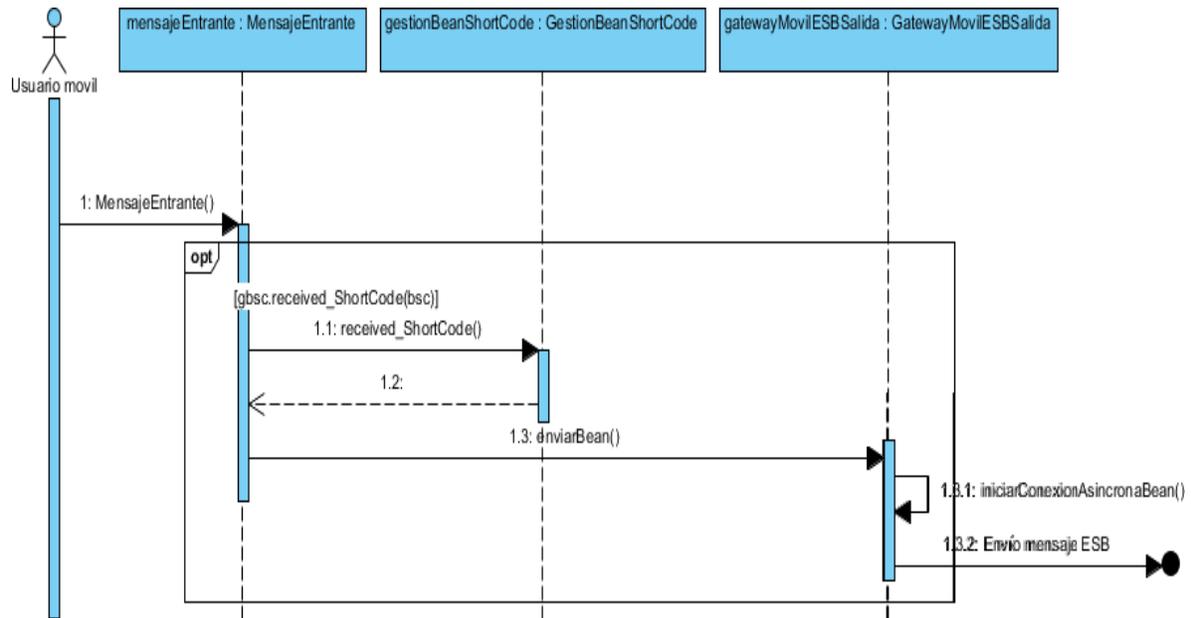


Figura 18. Diagrama de secuencia SMS entrante

- El usuario envía un SMS.
- El mensaje entrante es guardado para el control de los mensajes recibidos.
- Se hace la solicitud a la compuerta de salida del ESB.
- Envía la solicitud de forma asíncrona al ESB.

D.1.4.2. Petición de envío de mensaje SMS a través del modem GSM

- Recibe el mensaje proveniente del ESB.
- Se analiza el tipo de mensaje recibido y se toma la decisión del proceso a seguir.
- Si el mensaje es de tipo notificación, separa la lista de usuarios a los que debe ser enviado y se configura un SMS para cada uno de ellos
- Si es del tipo mensaje móvil MI, se configura el SMS al usuario deseado.
- Se crea el proceso de envío del o los SMS.

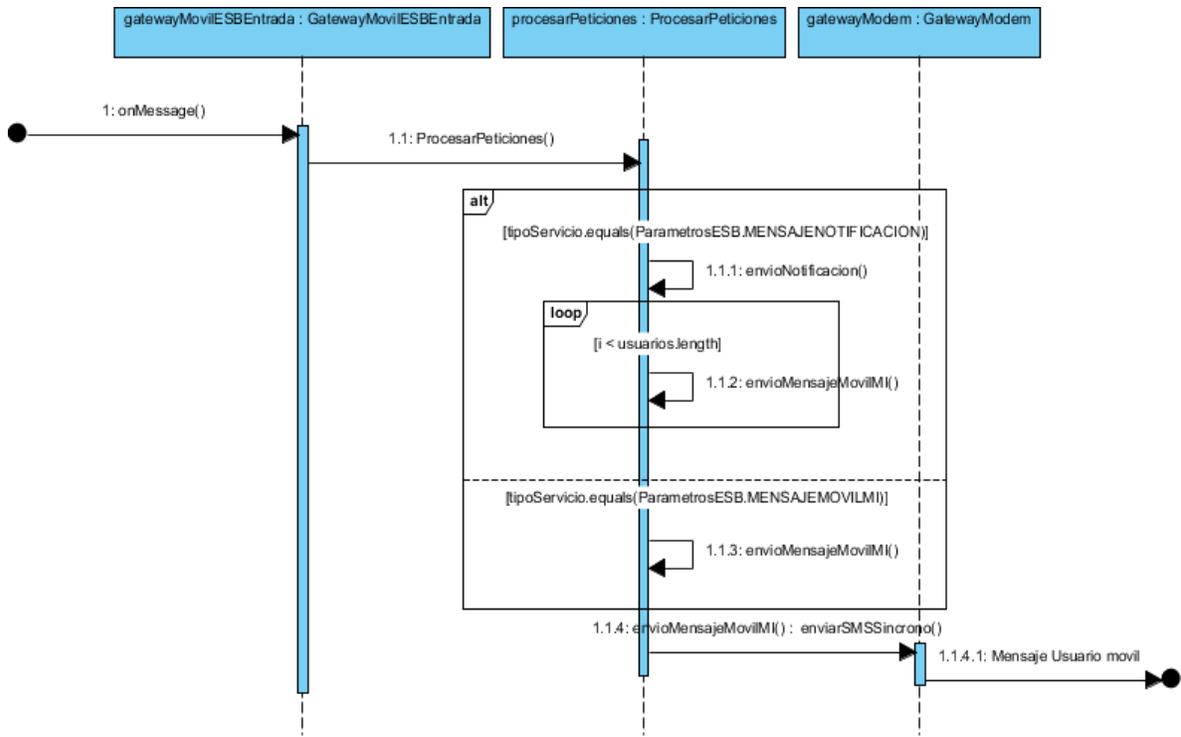


Figura 19. Diagrama de secuencia envío de mensaje por el módem GSM

D.1.4.3. Petición de envío de mensaje SMS a través de la conexión al SMSC

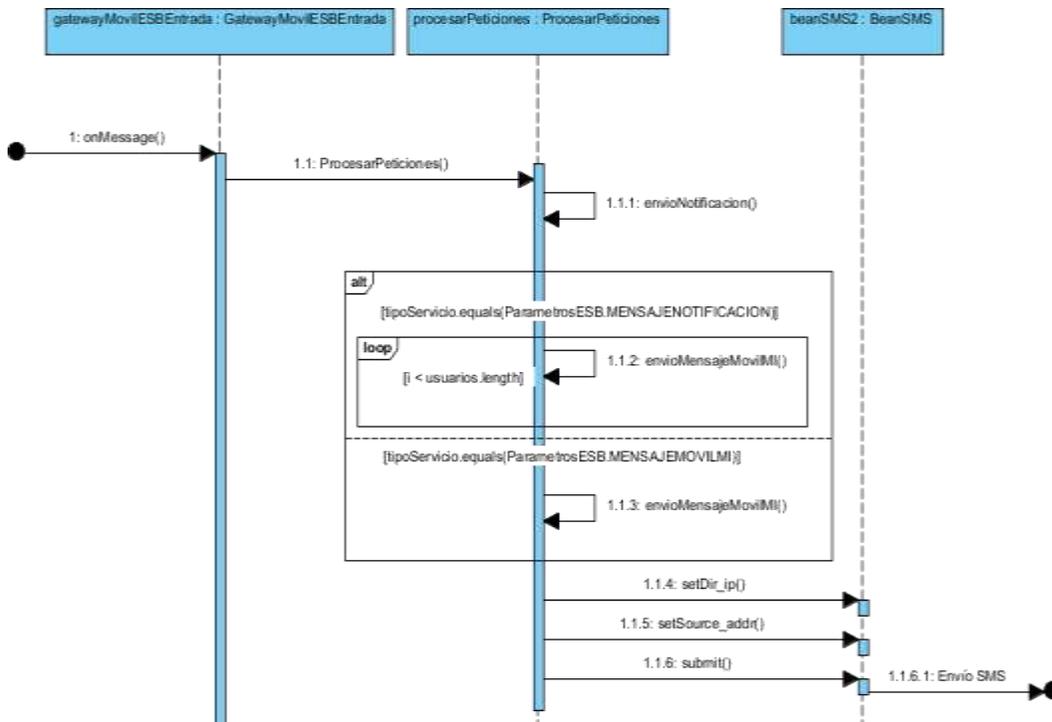


Figura 20. Diagrama de secuencia envío de mensaje por el SMSC

- Recibe el mensaje proveniente del ESB.
- Se analiza el tipo de mensaje recibido y se toma la decisión del proceso a seguir.

- Si el mensaje es de tipo notificación, separa la lista de usuarios a los que debe ser enviado y se configura un SMS para cada uno de ellos
- Si es del tipo mensaje móvil MI, se configura el SMS al usuario deseado.
- Se configura la dirección IP en la que está ubicada la SMSC.
- Se anexa el número de teléfono de quien envía el mensaje, para efectos de cobro.
- Se envía el mensaje.

D.1.4.4. Comportamiento de la conexión al SMSC

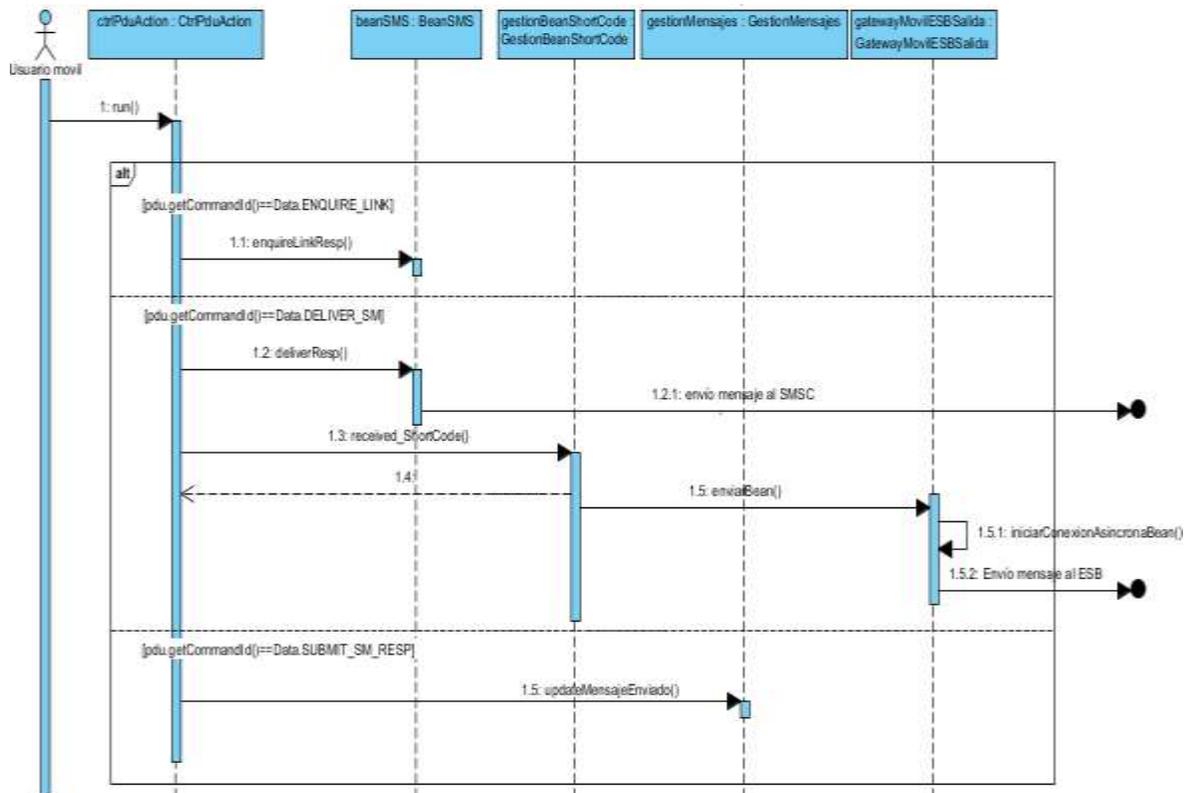


Figura 21. Diagrama de secuencia comportamiento de la conexión al SMSC

- Para empezar debe crear la conexión *enquirelink* con el SMSC.
- Una vez creada la conexión, se puede hacer solicitudes *DELIVER_SM* para el envío de mensajes al SMS al SMSC.
- El mensaje entrante es escuchado a través de *received_Shortcode*.
- Se hace la solicitud a la compuerta de salida del ESB.
- Envía la solicitud de forma asíncrona al ESB.
- *SUBMIT_SM_RESP*, recibe la información de respuesta del SMSC para un SMS enviado.

D.2. Servidor de Mensajería Instantánea

D.2.1. Casos de uso

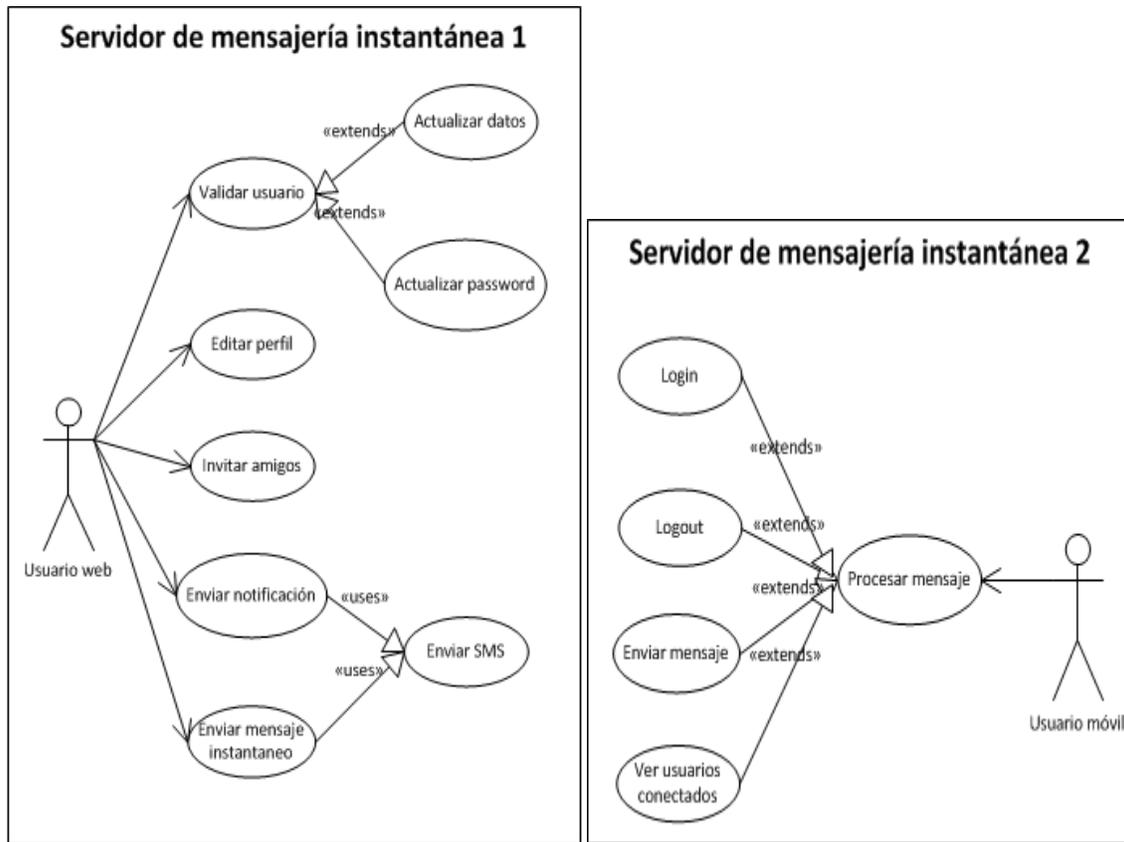


Figura 22. Diagrama de casos de uso SMI

D.2.2. Diagrama de paquetes

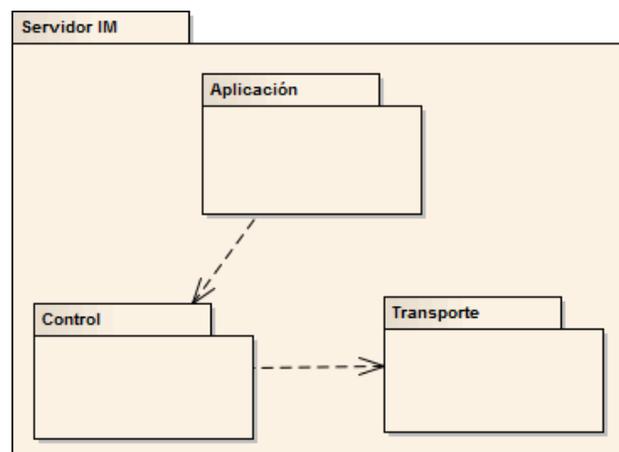


Figura 23. Diagrama de paquetes SMI

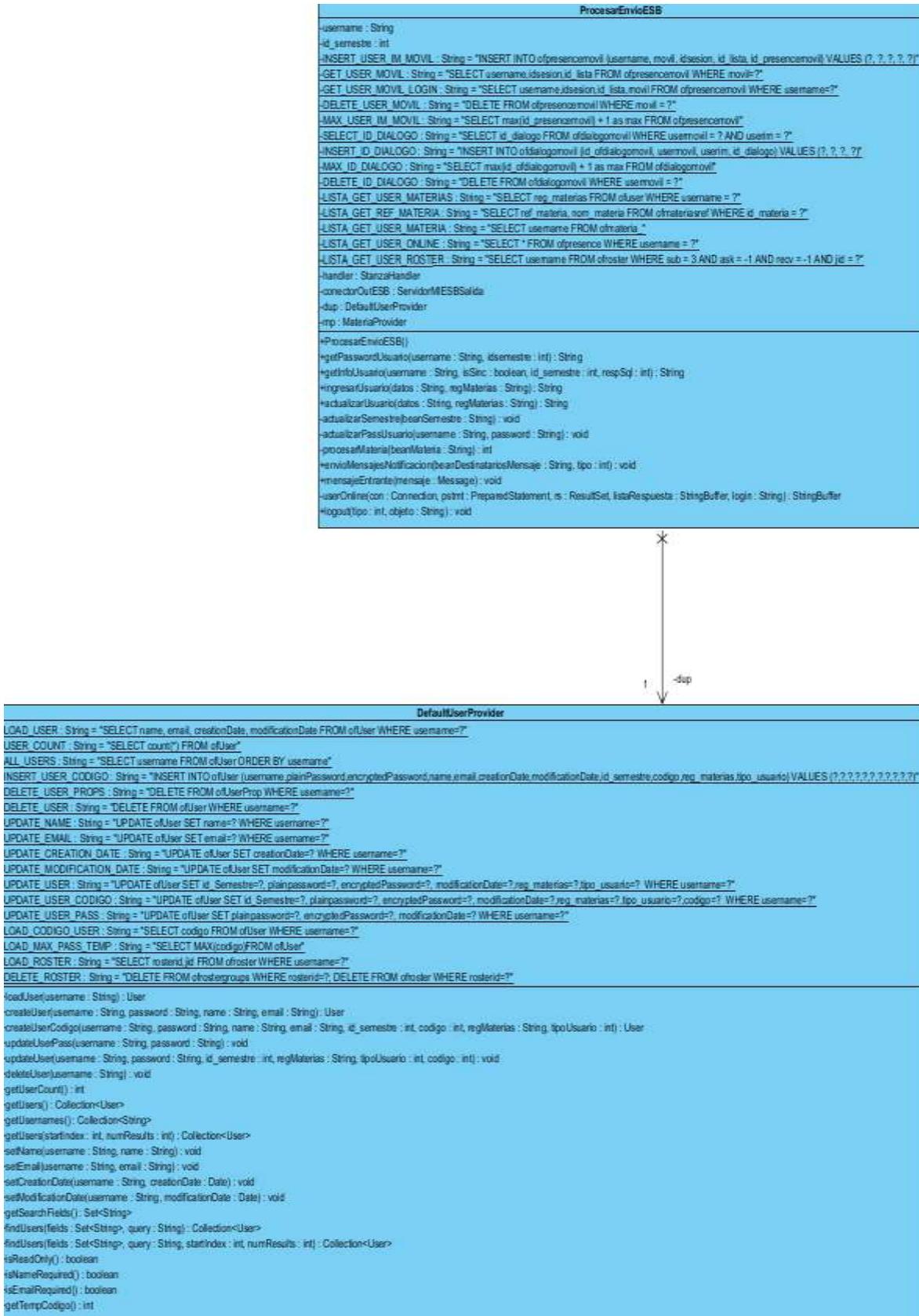


Figura 25. Diagrama de clases SMI 2



Figura 26. Diagrama de clases SMI 3

D.2.4. Diagramas de secuencia

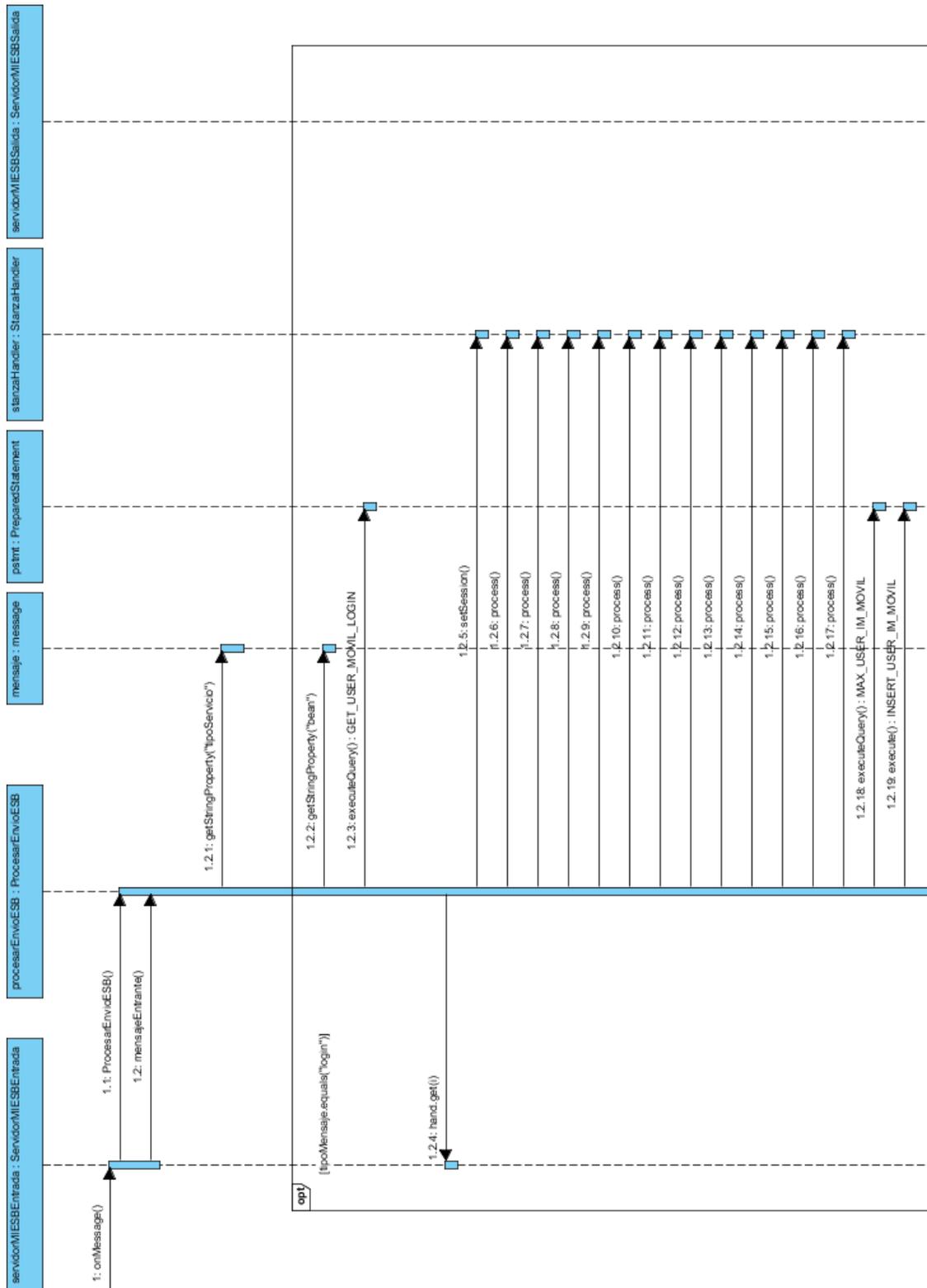


Figura 27. Diagrama de secuencia mensaje entrante Login

D.2.4.1. Diagrama de secuencia Mensaje entrante

- El mensaje llega por medio del ESB
- Se define el tipo de mensaje entrante.
- Los datos son extraídos del cuerpo del mensaje.

Tipo Login

- Inicia el proceso de verificación del usuario que envió el mensaje, mira si este usuario existe en la base de datos del servidor.
- Crea los *process*, para que sea identificado como usuario XMPP.
- Guarda el usuario como un cliente Móvil.

Tipo Logout

- Se toman los datos de usuario móvil del cliente, si existe como usuario móvil continua el proceso.
- Crea los *process*, para cerrar sesión de este cliente. Borra los datos del usuario como un cliente Móvil.

Tipo Mensaje

- Se toman los datos de usuario móvil del cliente, si existe como usuario móvil continua el proceso.
- Mira qué tipo de cliente es la persona a la que va dirigida el mensaje.
- Si el usuario está conectado a través de una aplicación al servidor de mensajería, crea los *process* para enviar el mensaje tipo chat.
- Por otra parte si es un usuario tipo móvil crea los parámetros necesarios para hacer una solicitud de SMS al ESB.

Tipo Lista

- Se toman los datos de usuario móvil del cliente, si existe como usuario móvil continua el proceso.
- Mira en la base de datos los clientes conectados para ese usuario y crea los parámetros necesarios para enviar una respuesta a través del ESB.

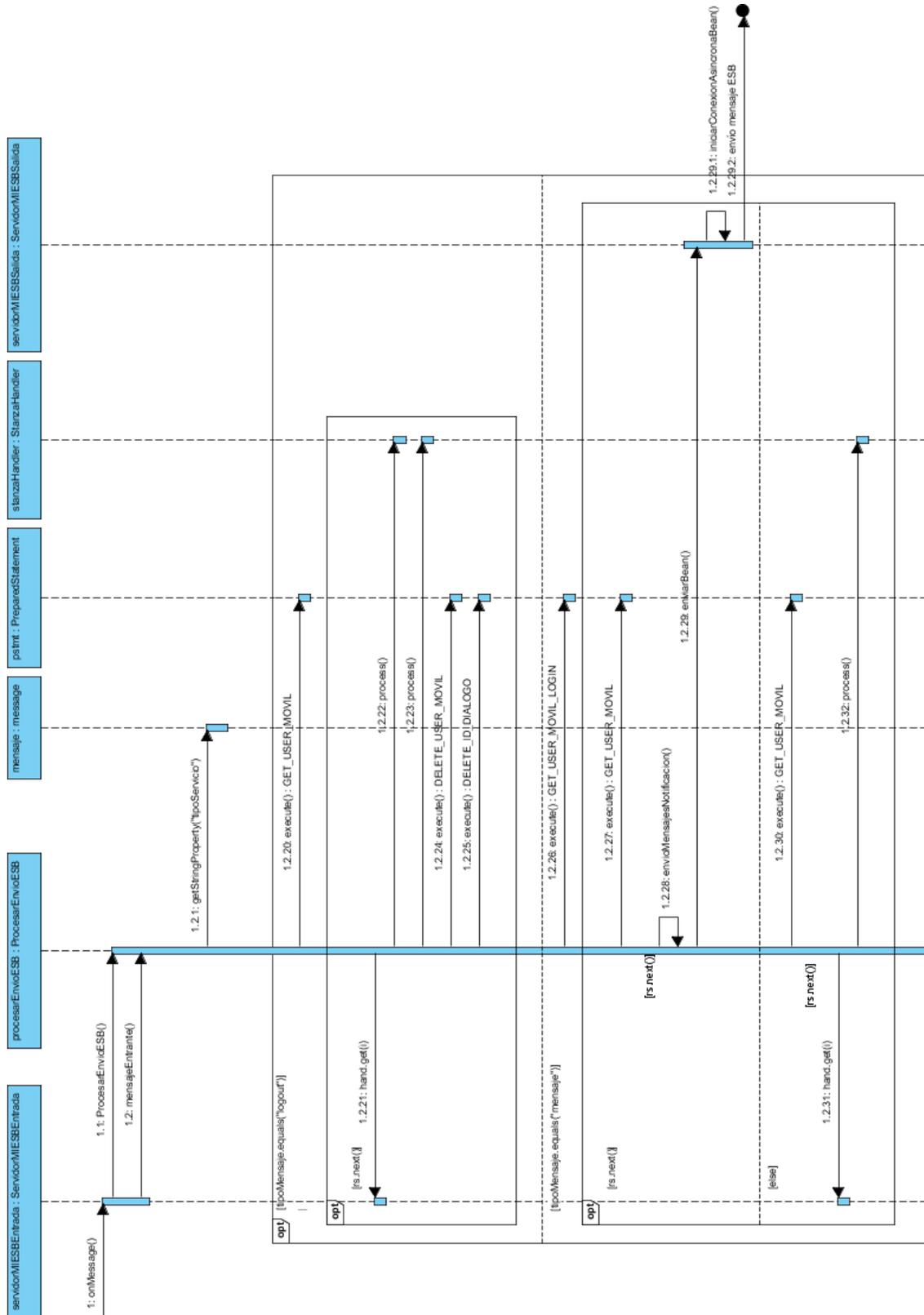


Figura 28. Diagrama de secuencia mensaje entrante Logout

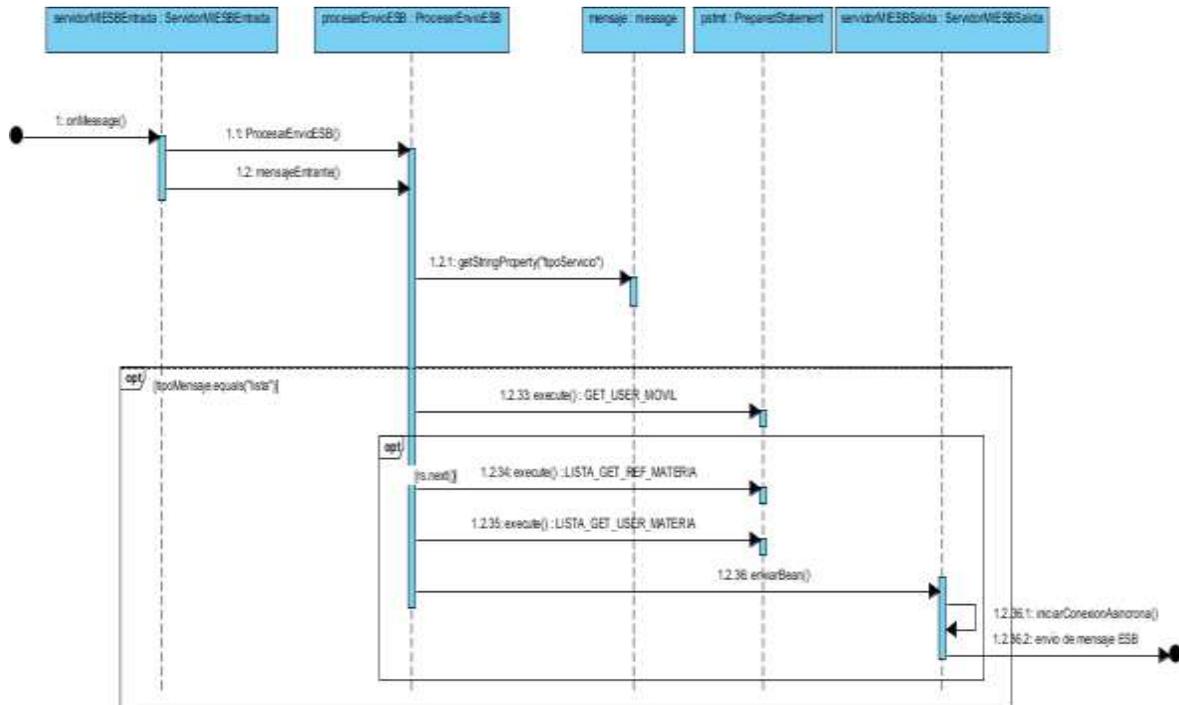


Figura 29. Diagrama de secuencia mensaje entrante Lista

D.3. Servidor Mail

D.3.1. Casos de uso

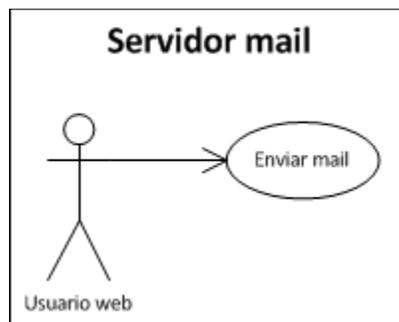


Figura 30. Diagrama de caso de uso SM

D.3.2. Diagrama de paquetes

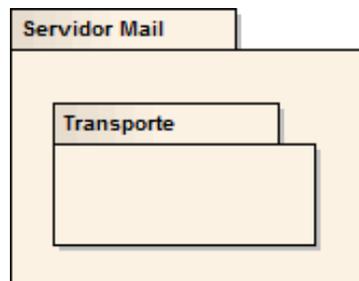


Figura 31. Diagrama de paquete SM

D.3.3. Diagrama de clases

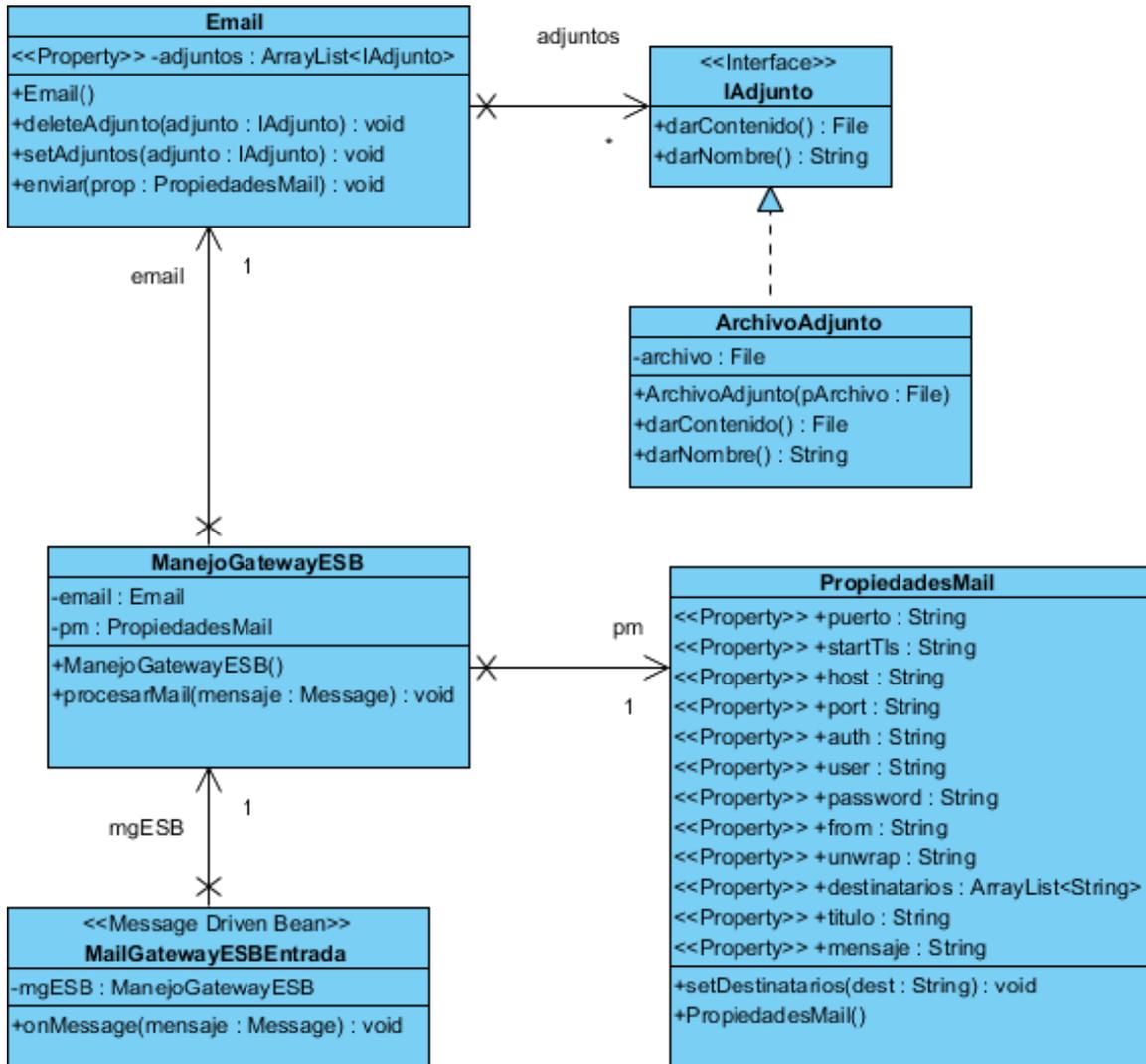


Figura 32. Diagrama de clases SM

D.3.4. Diagrama de secuencia

- La petición llega por medio del ESB.
- Se toman los datos relevantes del cuerpo del mensaje y se decide el tipo mensaje que será enviado.
- Se configura el servidor por el cual va ha ser enviado el correo electrónico.
- Se ingresa los usuarios a los que se les enviara el correo electrónico.
- Se envía el mensaje.

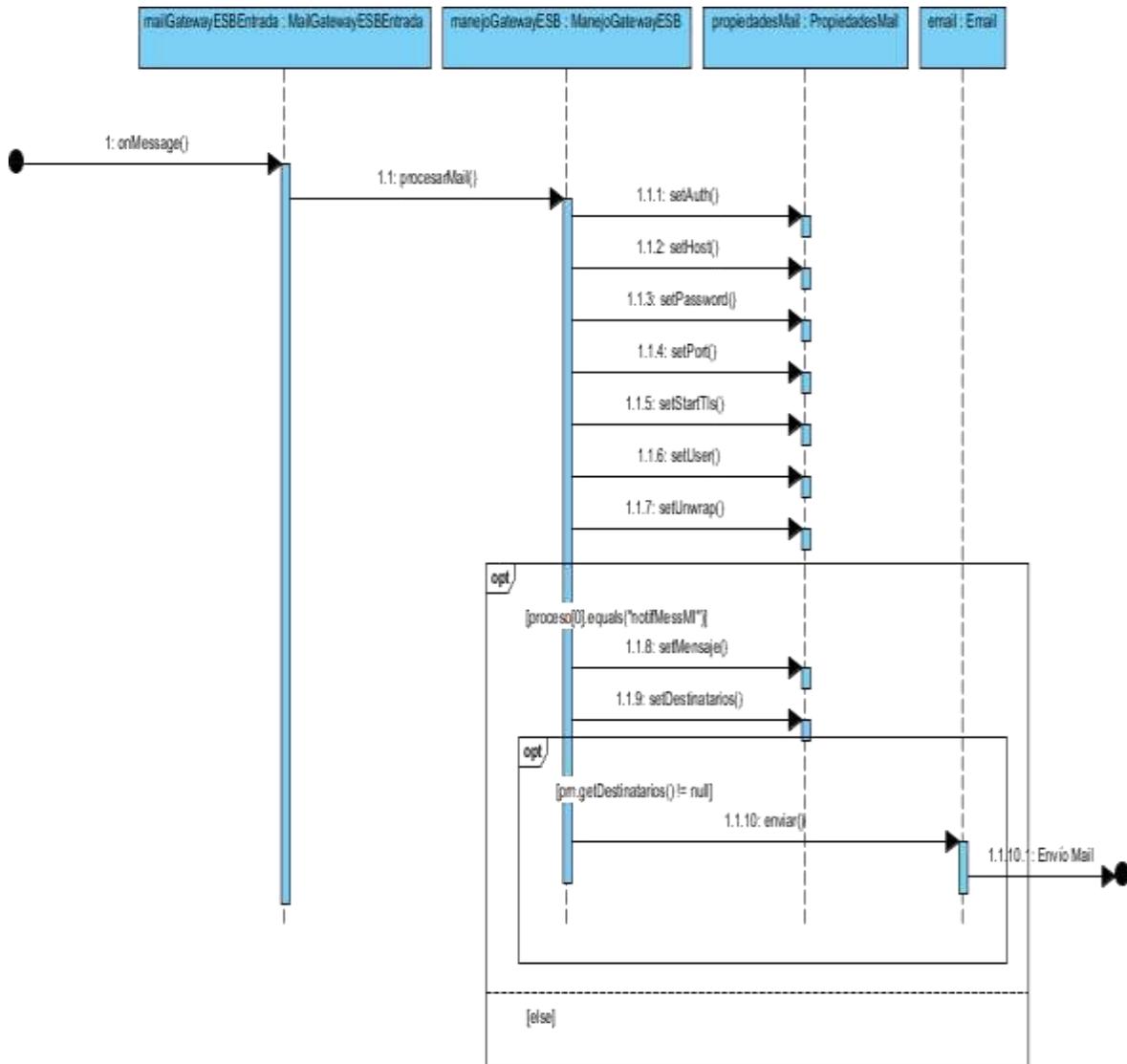


Figura 33. Diagrama de secuencia SM

D.4. Servidor de Aplicaciones Móviles

D.4.1. Casos de uso

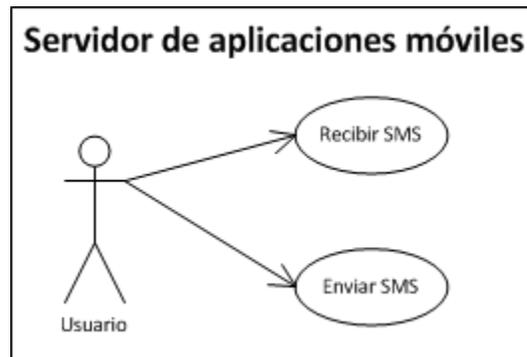


Figura 34. Diagrama casos de uso SAM

D.4.2. Diagrama de paquetes

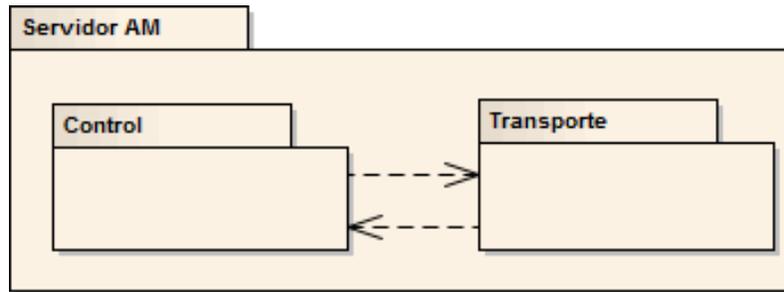


Figura 35. Diagrama de paquetes SAM

D.4.3. Diagrama de clases

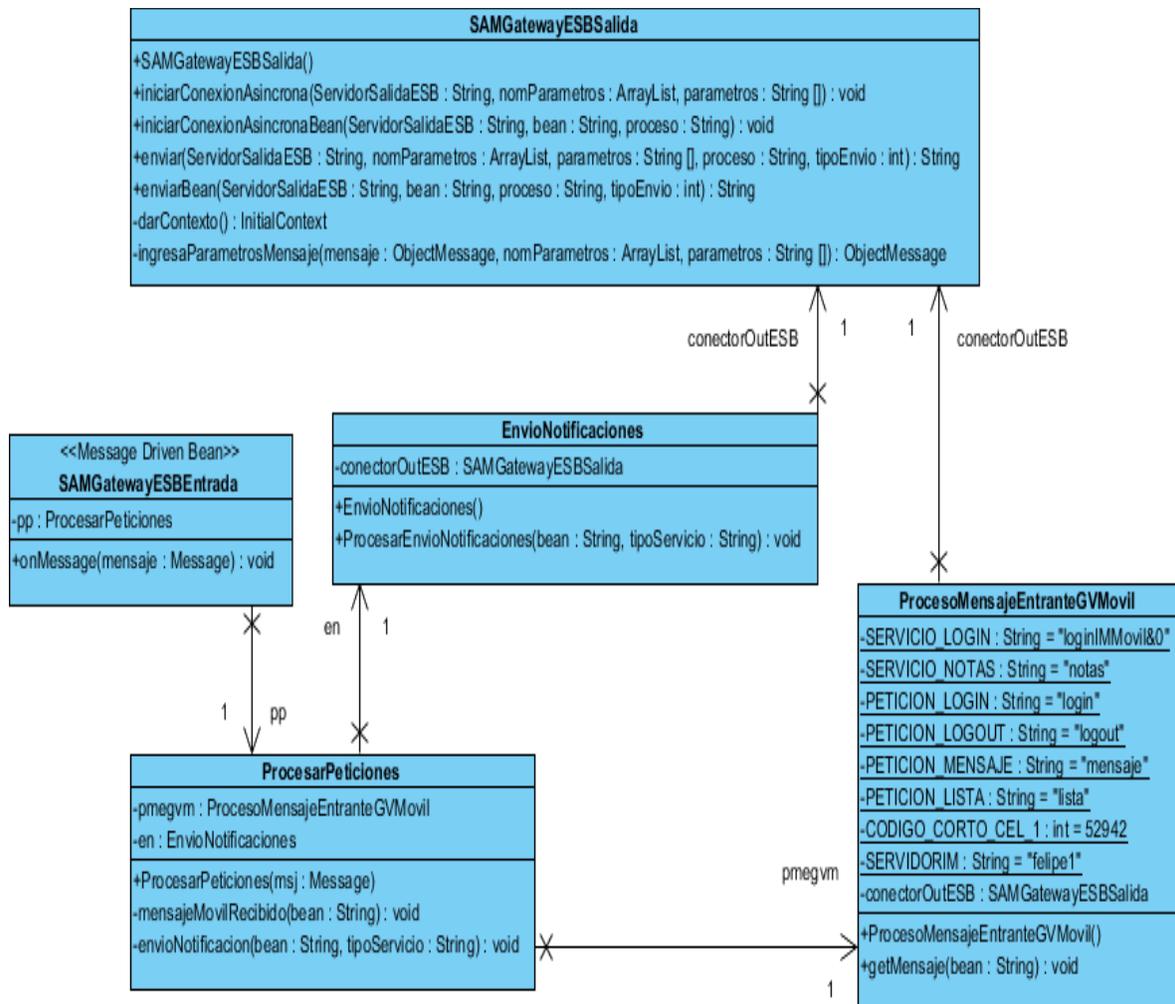


Figura 36. Diagrama de clases SAM

D.4.4. Diagrama de secuencia

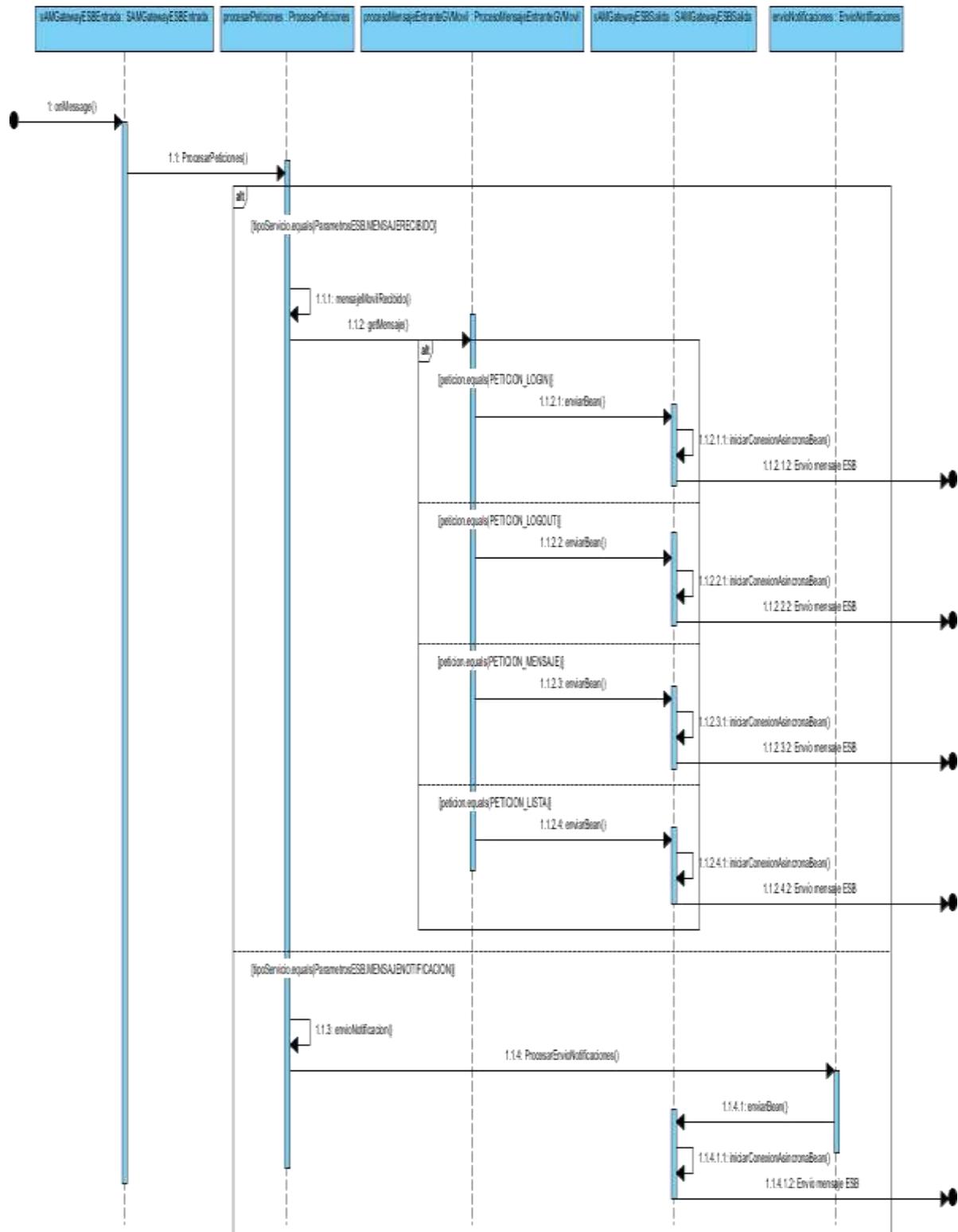


Figura 37. Diagrama de secuencia SAM

- La petición llega por medio del ESB.
- Se toman los datos relevantes del cuerpo del mensaje y se decide el tipo mensaje que será enviado.
- Según el tipo de mensaje arma los paquetes para ser enviados por el ESB.
- Se crean los parámetros necesarios para enviar de forma asíncrona la solicitud al ESB.

D.5. Servidor Base de Datos

D.5.1. Casos de uso

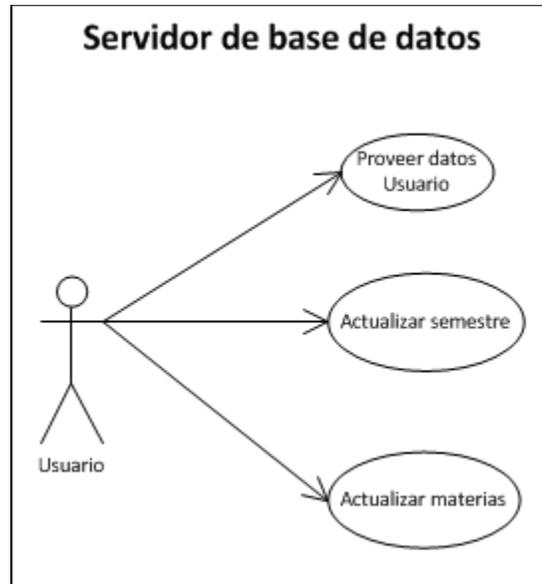


Figura 38. Diagrama casos de uso SBD

D.5.2. Diagrama de paquetes

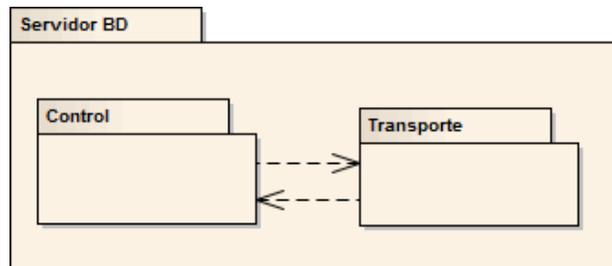


Figura 39. Diagrama de paquetes SBD

D.5.3. Diagrama de clases

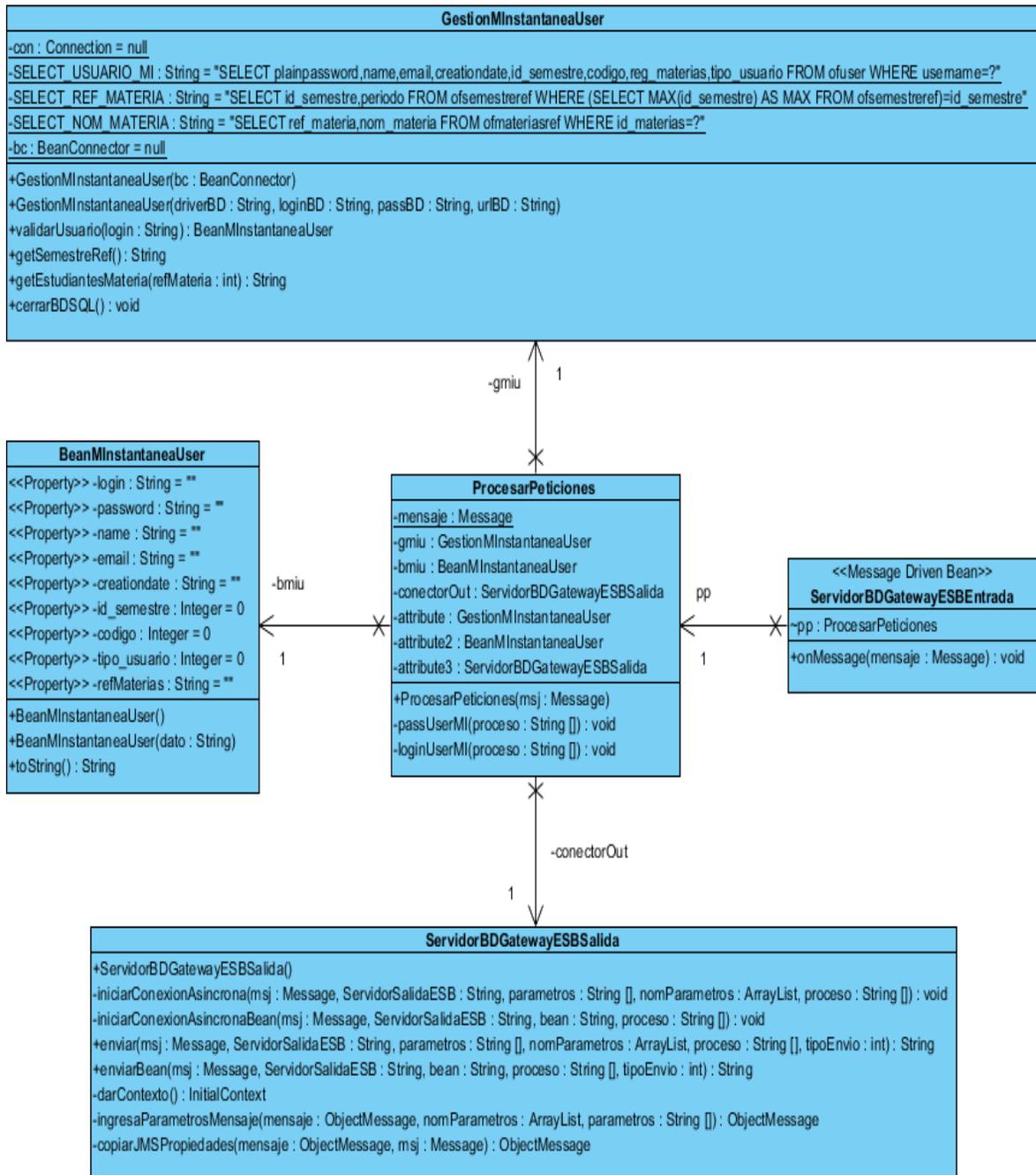


Figura 40. Diagrama de clases SBD

D.5.4. Diagrama de secuencia

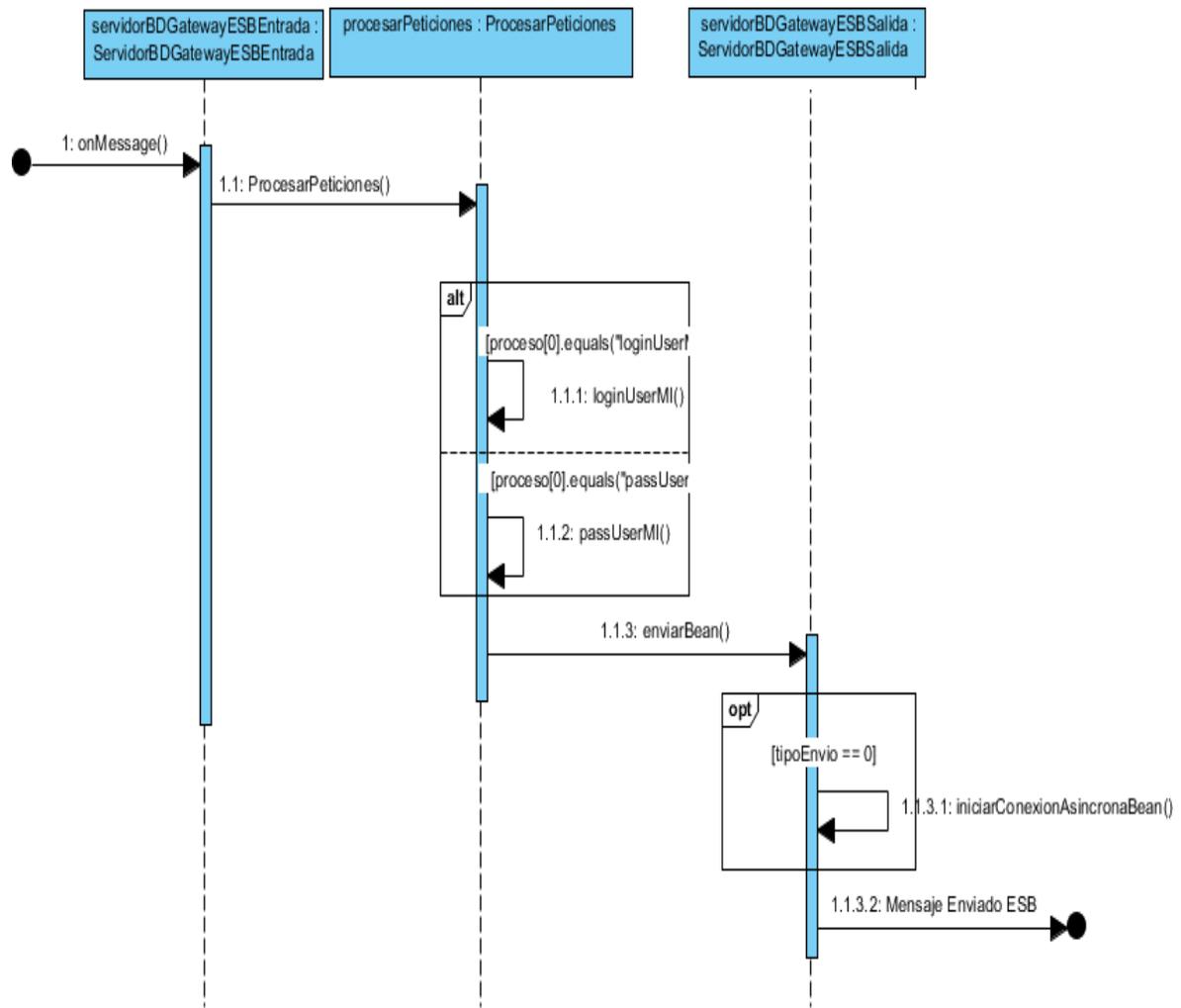


Figura 41. Diagrama de secuencia SBD

- La petición llega por medio del ESB.
- Se toman los datos relevantes del cuerpo del mensaje y se decide el tipo de consulta que debe realizarse.
- *Consulta login*. Se consultan los siguientes datos: cursos matriculados, contactos del curso, semestre en curso, datos generales de registro.
- *Consulta pass*. Se consultan los siguientes datos: password del usuario, semestre en curso.
- Se crean los parámetros necesarios para enviar la respuesta a través del ESB.

ANEXO E

INSTALACIÓN Y CONFIGURACIÓN DE COMPONENTES

Anexo E: Instalación y Configuración de Componentes

El objetivo de este anexo es lograr la instalación, y de manera cómoda, todos los requerimientos del sistema a implementar, que se necesitan para que funcionen las aplicaciones basadas en JBossESB y los diferentes servidores de este proyecto de grado. Este anexo hace referencia a la temática del capítulo 5.

E.1. JBossESB

A continuación se explican los pasos necesarios para instalar JBossESB 4.7 en el servidor de aplicaciones JBoss. En este caso los componentes ESB se implementan directamente dentro del servidor de aplicaciones.

E.1.1. Requisitos mínimos

- JDK 5 (v1.5.0_06 o superiores)
- ANT (v1.6.5 o superiores)
- JBoss Application Server 4.2.x.GA

E.1.2. Instalación y configuración JDK

- Descargar el JDK (Java 2 Development Kit) desde la página web de Oracle: www.oracle.com/technetwork/java/javase/downloads
- Se ejecuta el instalador y se siguen los pasos de instalación.
- Instalada la aplicación, se crean las variables de entorno JAVA_HOME y CLASSPATH y se edita la variable Path. Se deben crear con la ruta de instalación del SDK.

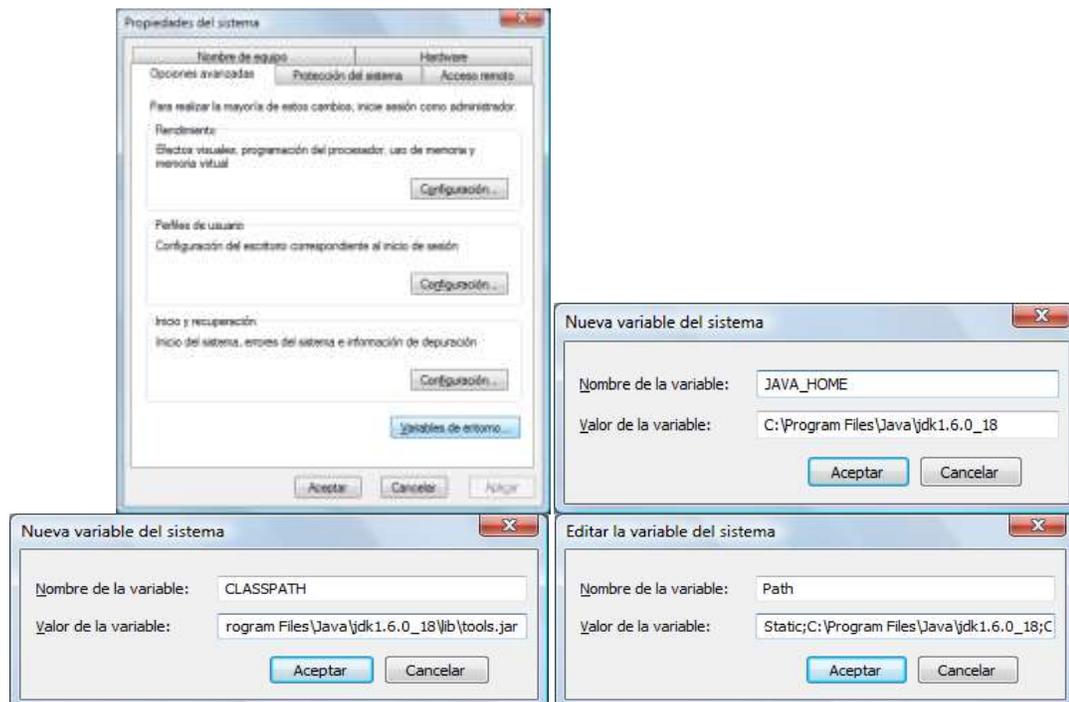
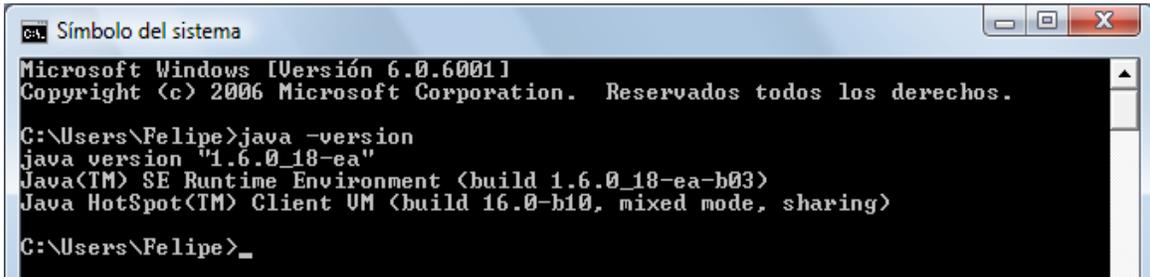


Figura 42. Configuración de variables de entorno JDK

- Para verificar la instalación del SDK, en una ventana de comandos se ejecuta `java -version`.



```
Microsoft Windows [Versión 6.0.6001]
Copyright (c) 2006 Microsoft Corporation. Reservados todos los derechos.

C:\Users\Felipe>java -version
java version "1.6.0_18-ea"
Java(TM) SE Runtime Environment (build 1.6.0_18-ea-b03)
Java HotSpot(TM) Client VM (build 16.0-b10, mixed mode, sharing)

C:\Users\Felipe>_
```

Figura 43. Prueba de versión del SDK

E.1.2. Instalación y configuración de apache ANT

- Se descarga la versión binaria de ANT desde la página de Apache: <http://ant.apache.org/bindownload>
- Crear un directorio *Servidor* en la partición C y se descomprime el archivo descargado.
- Se crea la variable de entorno ANT_HOME y se edita la variable Path.

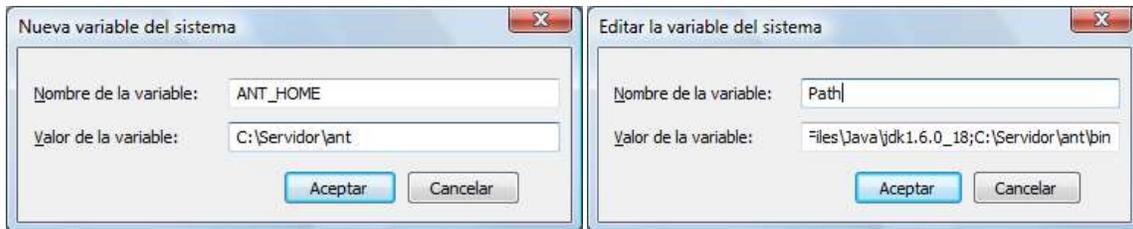
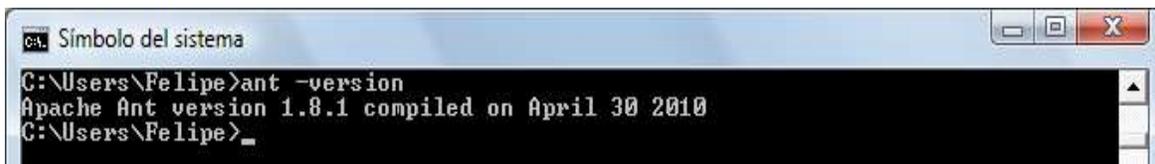


Figura 44. Variables de entorno para ANT

- Se verifica la instalación de ANT ejecutando `ant -version` en una ventana de comandos.



```
Microsoft Windows [Versión 6.0.6001]
Copyright (c) 2006 Microsoft Corporation. Reservados todos los derechos.

C:\Users\Felipe>ant -version
Apache Ant version 1.8.1 compiled on April 30 2010

C:\Users\Felipe>_
```

Figura 45. Verificación ANT

E.1.3. Instalación y configuración de JBoss 4.2.2

- Descargar la versión binaria de JBoss 4.2.2 desde la página oficial de JBoss: www.jboss.org/jbossas/downloads
- En el directorio *Servidor* creado para ANT se descomprime el archivo descargado.
- Ahora se crea la variable de entorno JBOSS_HOME.

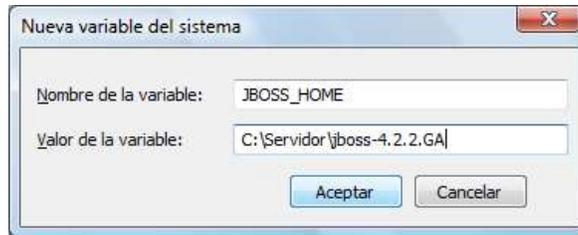


Figura 46. Variable de entorno para JBoss

- Se arranca el servidor JBoss. En una ventana de comandos, se ubica el subdirectorio `JBOSS_HOME/bin` de la distribución JBoss e iniciamos el servidor con el comando `run.bat -b 0.0.0.0`.

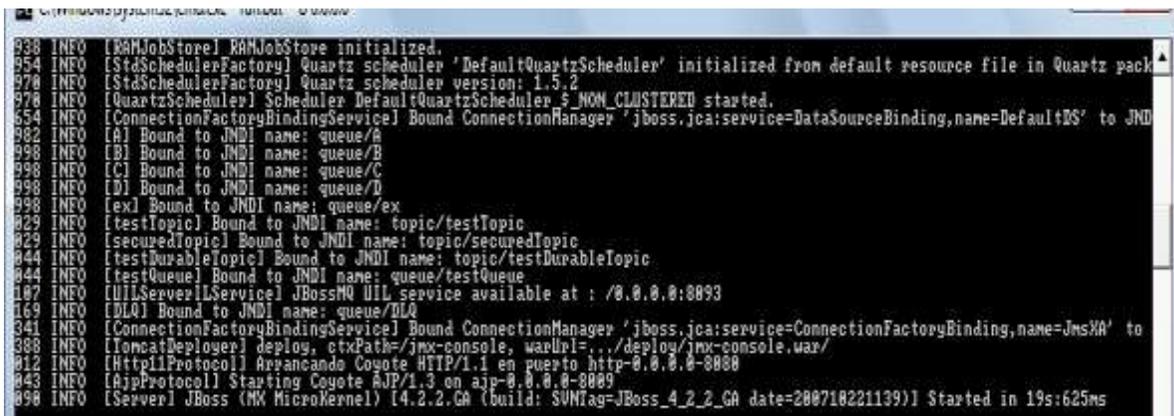


Figura 47. Ventana de inicio JBoss

- Para comprobar el funcionamiento de JBoss abrimos un explorador de Internet y le damos la ruta: <http://localhost:8080/> en la cual debe aparecer la página principal de JBoss.

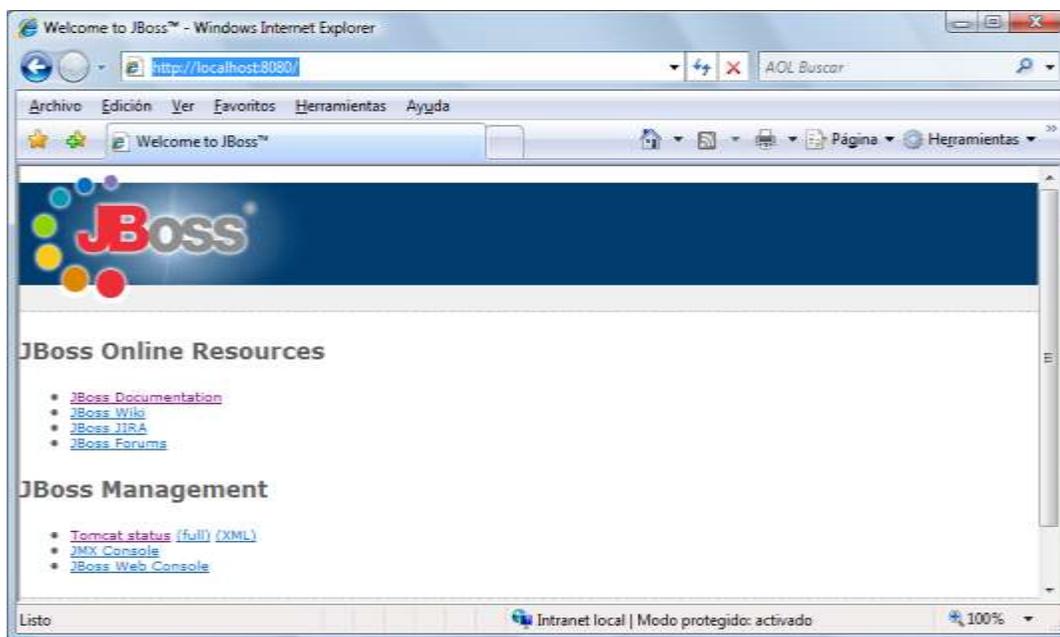


Figura 48. Página de inicio JBoss

E.1.4. Instalación y configuración de JBoss 4.7

- Descargar la versión binaria de JBossESB 4.7 desde la página oficial de JBossESB: www.jboss.org/jbossesb/downloads
- En el directorio *Servidor* creado para ANT se descomprime el archivo descargado. Es necesario que JBossESB se descomprima en el directorio padre en el que se instaló la versión de JBoss.
- Crear la variable de entorno `JBOSS_ESB_HOME`.

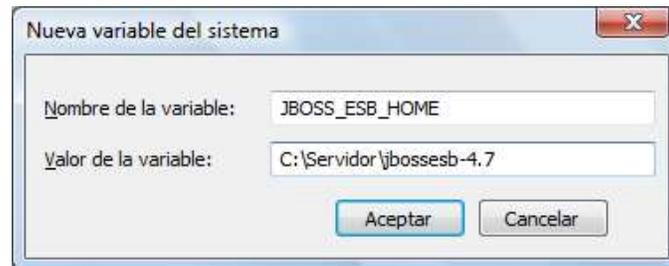


Figura 49. Variable de entorno JBossESB

- La distribución del ESB viene con un servidor de aplicaciones configurado de forma mínima para la instalación y ejecución de servicios ESB. Sin embargo esta versión no incluye los paquetes de instalación de EJBs ni componentes web. Por lo anterior se necesita instalar el ESB en un servidor de aplicaciones completo y utilizarlo desde allí. Con este fin, se localiza el archivo:

`JBOSS_ESB_HOME/install/ deployment.properties-example`

Se copia el archivo en la misma carpeta con el nombre *deployment-properties* y se modifican las propiedades de este archivo para que apunten al servidor de aplicaciones en la configuración *default*. Es decir, hay que modificar las siguientes propiedades y eliminar lo relacionado a Tomcat.

```
# application server root directory
org.jboss.esb.server.home=/jboss-4.2.2.GA
# the instance of jboss you are running(default)
org.jboss.esb.server.config=default
```

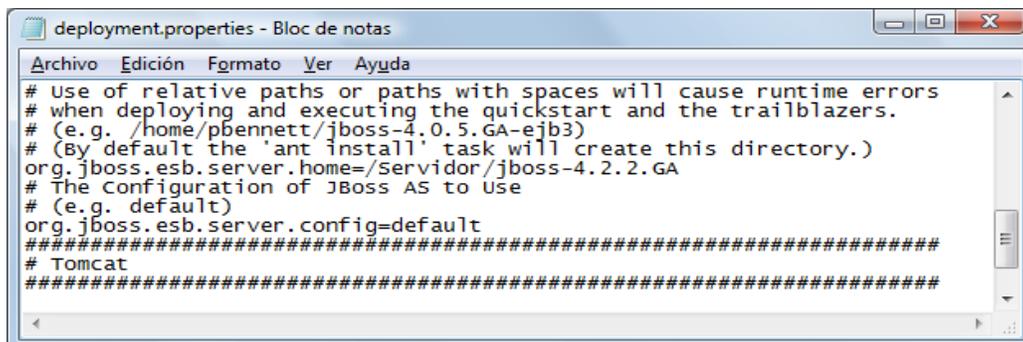


Figura 50. Modificación en *deployment-properties*

- Ahora hay que desplegar el ESB en el servidor de aplicaciones. Se inicia una ventana de comandos de Windows, ubicada en el directorio `JBOSS_ESB_HOME/install` y se ejecuta la instrucción `ant deploy`.

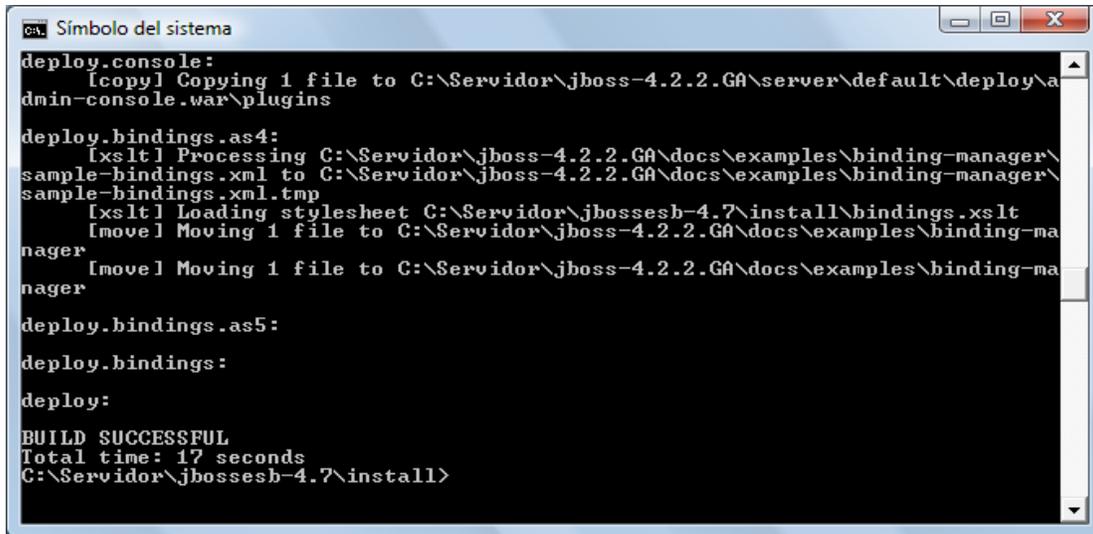


Figura 51. Despliegue de ESB en JBoss

- Al finalizar estos pasos, ya se puede utilizar los recursos ESB en el servidor JBoss instalado.

E.2. Configuración de la mensajería del ESB

Utilizando el entorno de desarrollo integrado Eclipse Ganymede (el cual se puede descargar de www.eclipse.org/ganymede) se procede a hacer los cambios de configuración necesarios para adaptarlos al nuevo entorno de ejecución del proyecto.

- Se crea una carpeta en la que se copia el proyecto. Si se quiere se utiliza el Workspace del Eclipse Ganymede.
- En el fichero `jboss-esb.xml`, se cambia donde sea necesario las URL donde se encuentran los otros servidores.

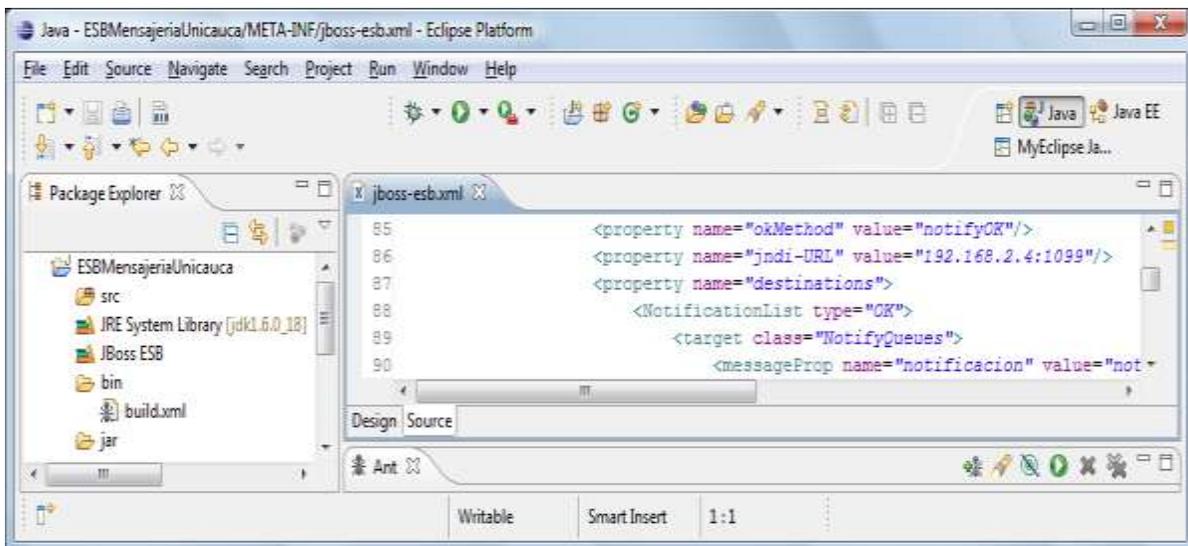


Figura 52. Cambio de URL de los servidores en el ESB

- Se hace un deploy de la aplicación, corriendo el servidor ANT de eclipse.

E.3. Configuración del Servidor de Mensajería Instantánea

- Se crea una carpeta en la que se copia el Código Fuente del proyecto. Si se quiere se utiliza el Workspace del Eclipse Ganymede.
- Se abre el proyecto en el entorno de desarrollo y se modifica en el paquete *org.unicauca.servidorMI.esb*, Clase *ParametrosESB* el parámetro *URL_ESB* con la nueva dirección en donde se encuentra el ESB.

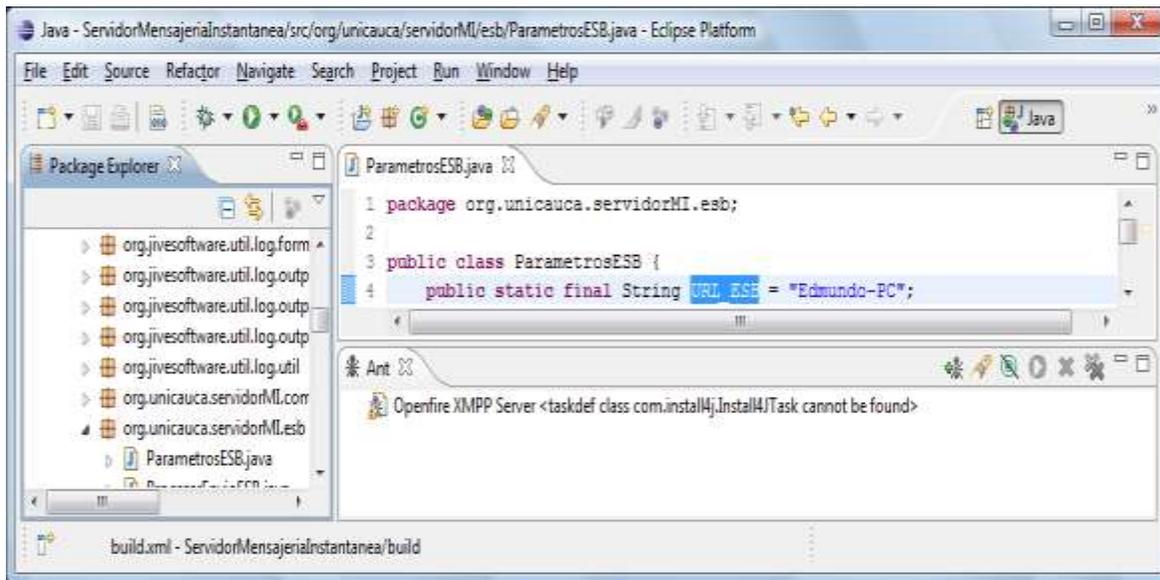


Figura 53. Cambios SMI

- Se hace un deploy de la aplicación, corriendo el servidor ANT de eclipse.
- En PostgreSQL se crea una base de datos con el nombre SERIM.
- Utilizando una ventana de comandos Windows, en el directorio raíz del SMI, se busca la dirección */target/openfire/lib* y se corre por primera vez la aplicación utilizando el comando *java -jar startup.jar*.
- Para configurar el SMI, abrimos un navegador web en la URL: <http://localhost:9090> y se siguen los pasos para la configuración.
- Se configura el idioma del servidor.
- Se escoge el nombre del dominio del servidor, los otros datos se dejan por defecto.



Figura 54. Configuración del dominio SMI

- Se escoge el tipo de conexión a la base de datos, para este caso elegimos la conexión estándar. En la siguiente pantalla de configuración, se define el servidor de base de datos, en este caso PostgreSQL, y se configura la URL con el nombre de la base de datos creada anteriormente. Por último se coloca el username y el password del servidor.

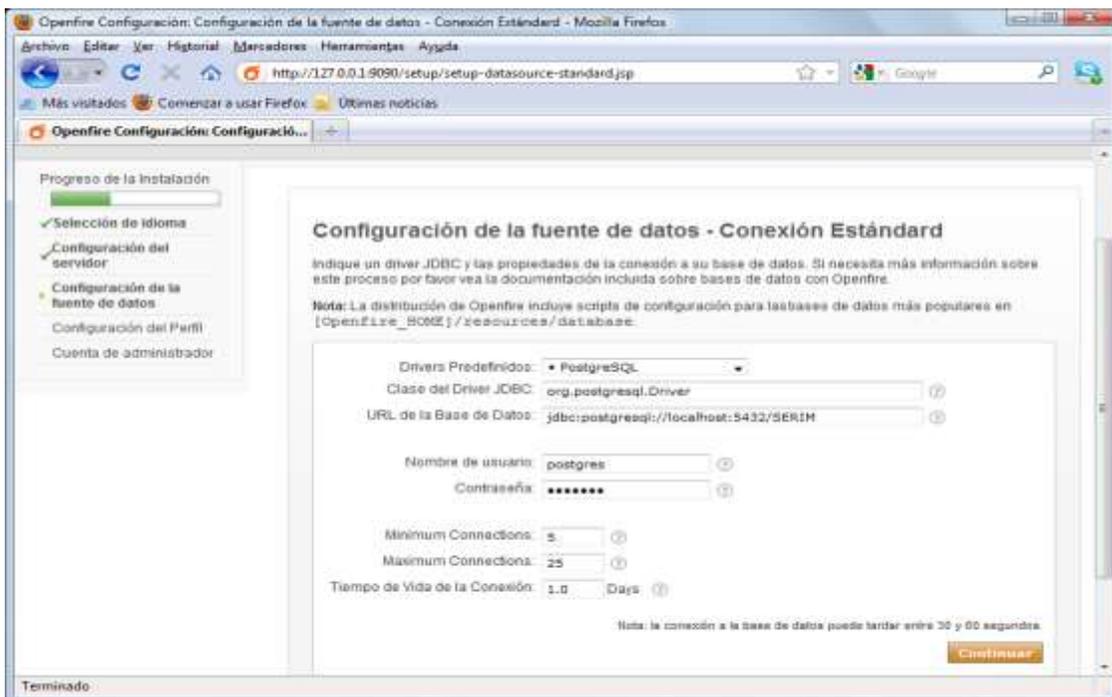


Figura 55. Configuración del servidor base de datos

- Configurar la clave del administrador del sistema. Es importante crear el usuario servidor y guardar sus datos, para poder validarse en la consola de administración.
- Reiniciar el servidor para que haga efecto la configuración creada.

E.4. Configuración de los Servidores del Proyecto

Para configurar los demás servidores se debe tener en cuenta que para el Gateway Móvil, SAM (Servidor de Aplicaciones Móviles) y SBD (Servidor Base de Datos) se debe hacer el deploy en MyEclipse; para el SM (Servidor Mail) se hace el deploy con Eclipse Ganymede, asegurándose de que el archivo creado quede dentro de la carpeta *deploy* del servidor JBoss. Teniendo en cuenta lo anterior, se debe modificar en el paquete ESB, la clase *ParametrosESB*, el parámetro *URL_ESB* con la nueva dirección en donde se encuentra.

E.5. Creación del Ejecutable del Cliente de Mensajería

- Con la ayuda del entorno de desarrollo NetBeans, cambiar el parámetro *servidorMensajeria* con la dirección IP o URL en donde se encuentra el Servidor de mensajería instantánea. Este parámetro se encuentra en el paquete *package org.jivesoftware*, clase *LoginDialog*. Y finalmente construir la aplicación teniendo en cuenta que el servidor ANT debe ser versión 1.6 o 1.7.

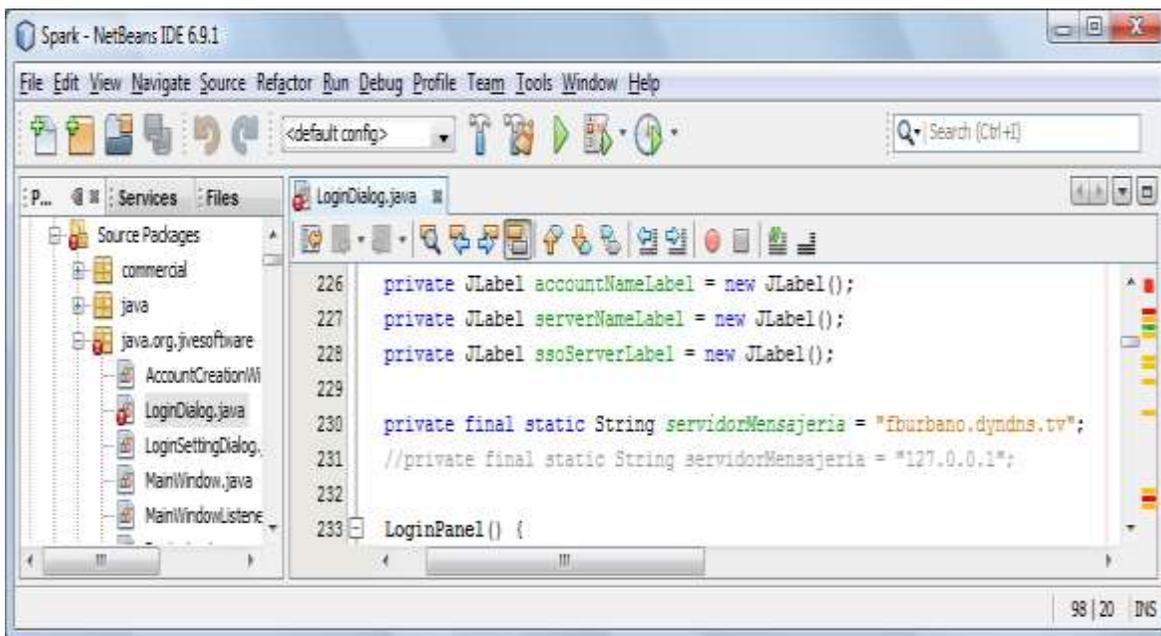


Figura 56. Cambio servidor de mensajería

- Para generar el ejecutable del cliente se ha seleccionado “install4j” de EJ-Technologies, y se trata de un wizard que partiendo del código fuente compilado y personalizando una serie de opciones acaba generando un paquete de instalación. Este software no es gratuito, pero desde la página web del fabricante se puede descargar una versión de evaluación de 90 días:
www.ej-technologies.com/download/install4j/files.html

- Con esta aplicación abriremos el proyecto *unicaucaSpark.install4j* que se localiza en el path: *builder\installer*. Los siguientes pasos ya están configurados en este proyecto de instalación, pero se nombran en caso de necesitar configurar algún parámetro.
- Dentro de *General Settings*, en la pestaña de *Application Info* se podrán personalizar los siguientes parámetros:

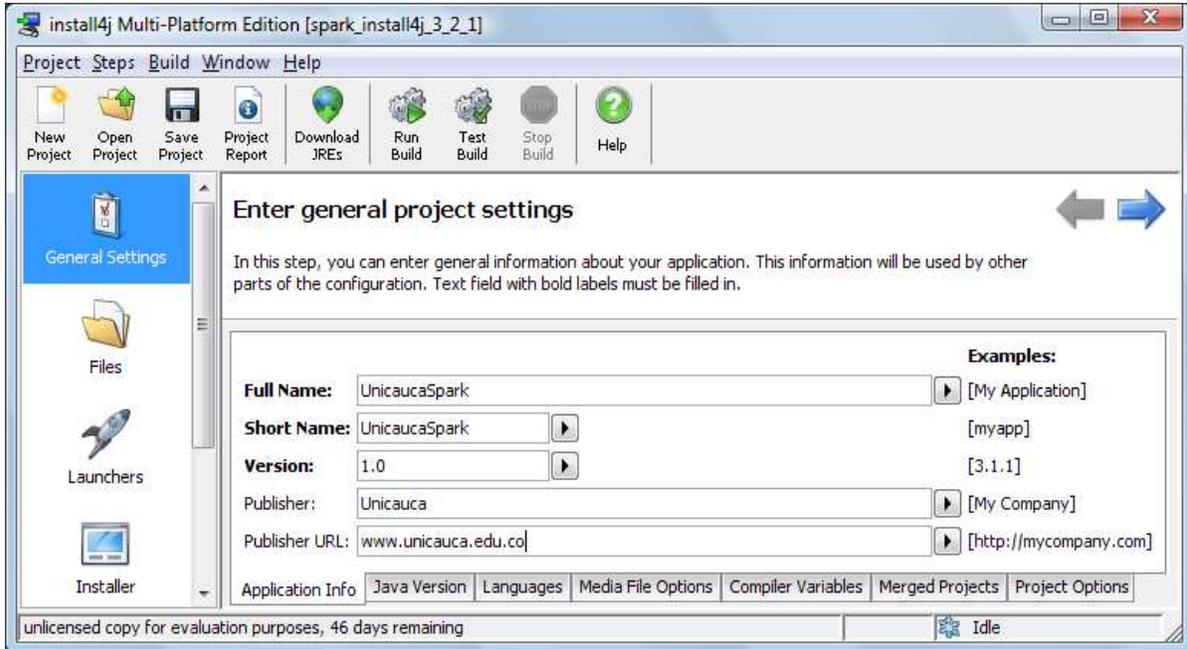


Figura 57. Application Info

- En la pestaña de *Compiler Variables* se podrán personalizar los siguientes parámetros:

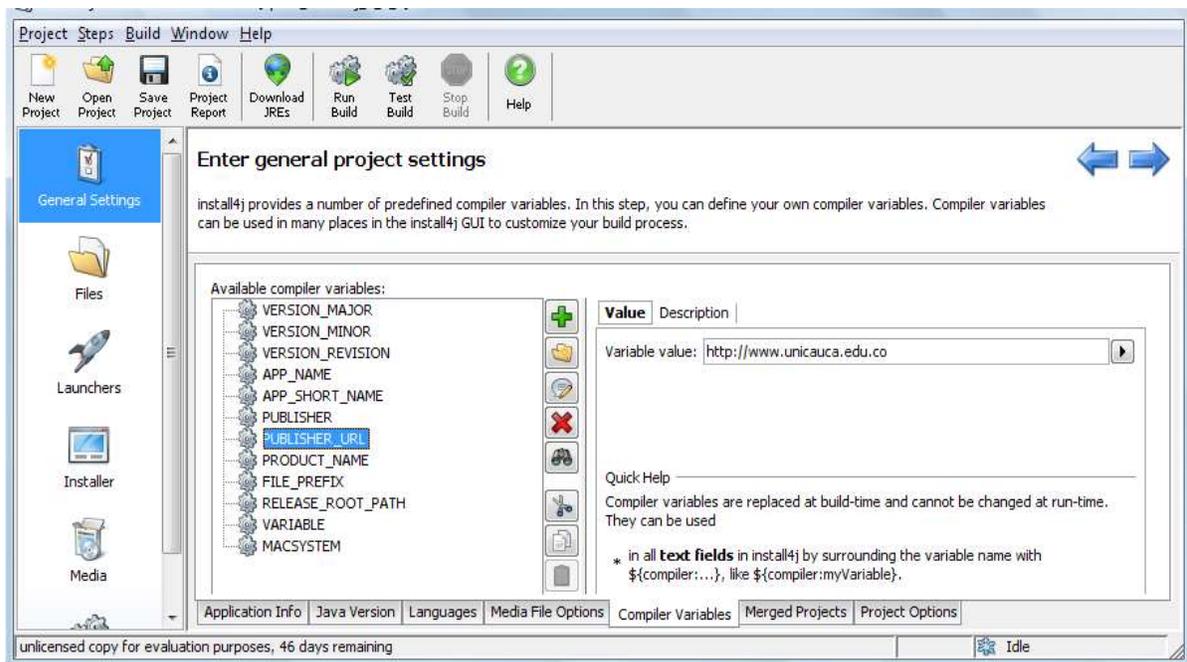


Figura 58. Compiler Variables

- En la pestaña de *Media File Options* se podrán personalizar los siguientes parámetros:

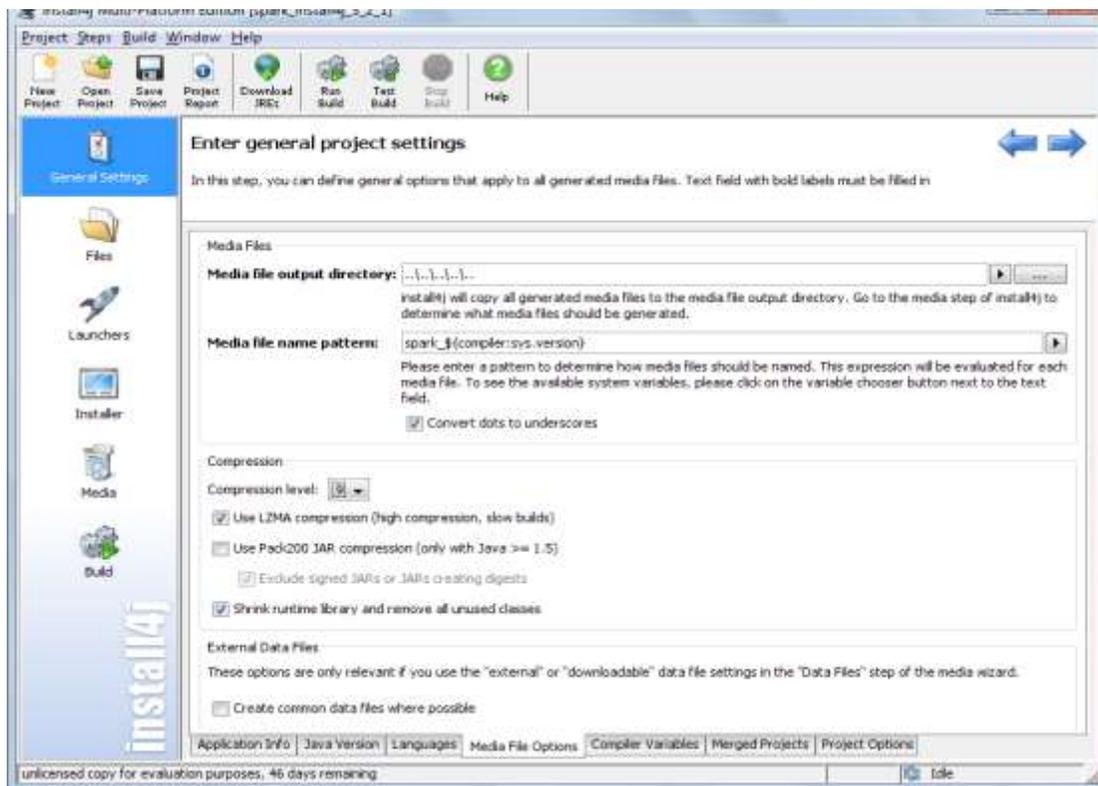


Figura 59. Media File Options

- En el menú de *Launcher*, se edita las propiedades del instalable Spark.
- En el menú *Executable Info* se puede cambiar el nombre del instalable si se desea.

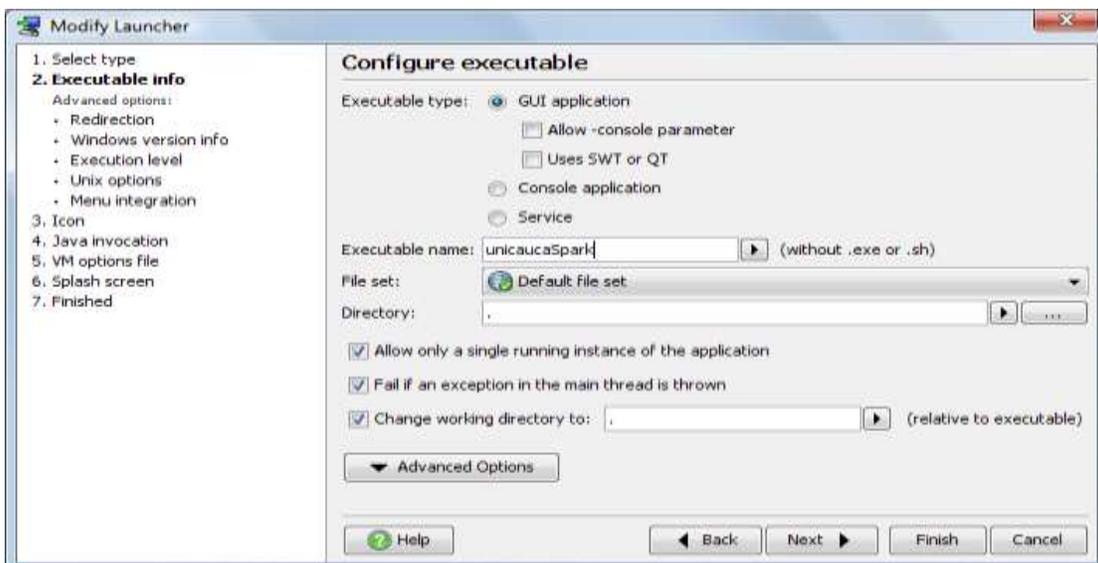


Figura 60. Launcher - Executable Info

- Se edita la versión del nuevo instalador.

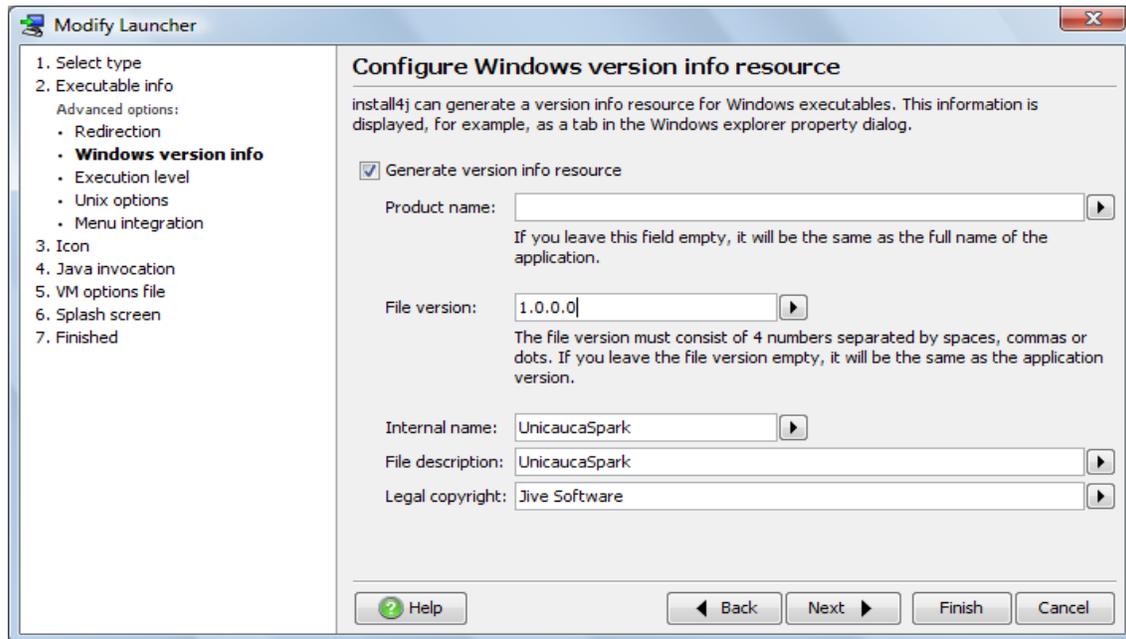


Figura 61. Editar versión instalador

- En Media, se editan las propiedades de Windows ya que al tratarse de una versión de evaluación solo permite generar Wizards de instalación para Windows.

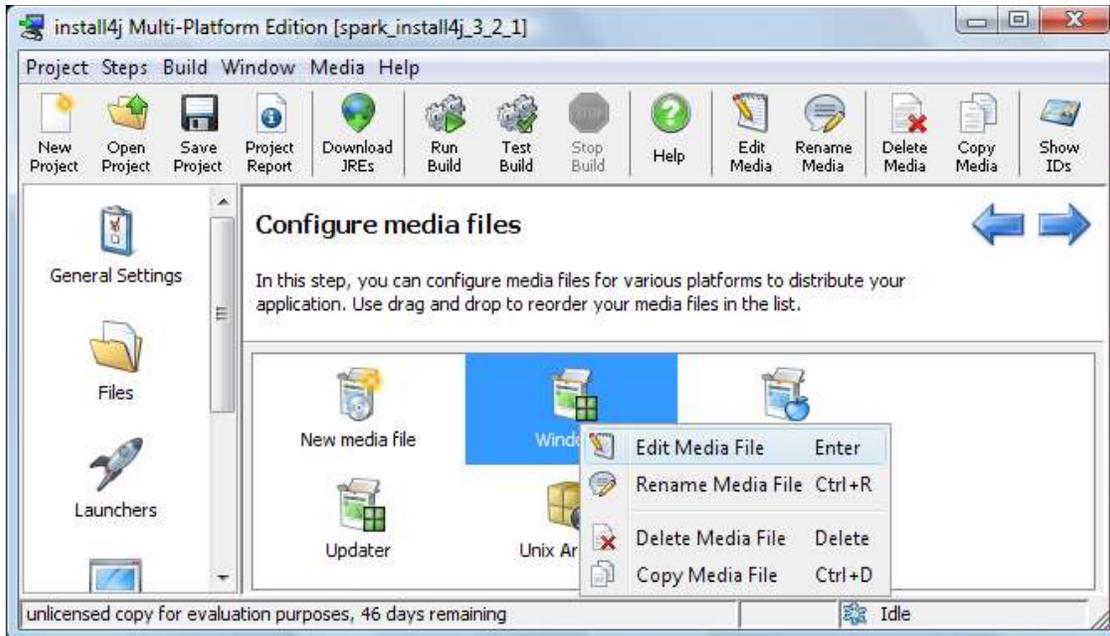


Figura 62. Editar opciones Windows

- Se edita el nombre del directorio de instalación.



Figura 63. Directorio de instalación

- Seleccionar una versión de la máquina virtual Java a instalar con el mismo paquete, ya que para que funcione UnicaucaSpark se requiere de la versión 1.6 o 1.7.

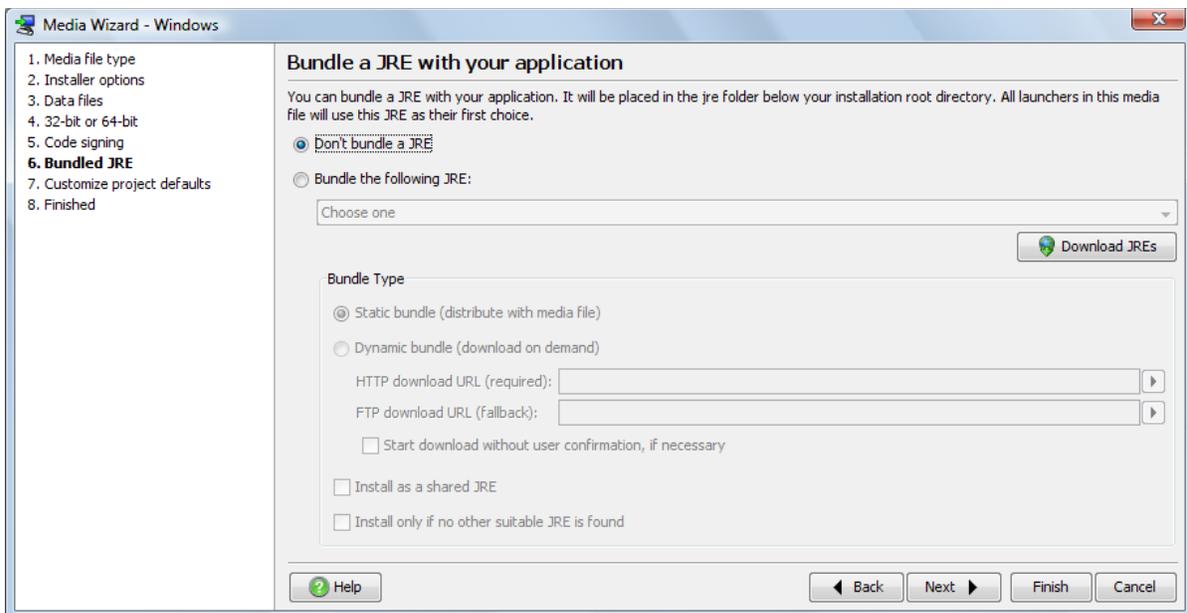


Figura 64. Máquina virtual para el instalador

- Se configura el idioma del paquete de instalación.



Figura 65. Configuración de idioma del ejecutable

- En la sección *Exclude launchers* se selecciona la opción starter.

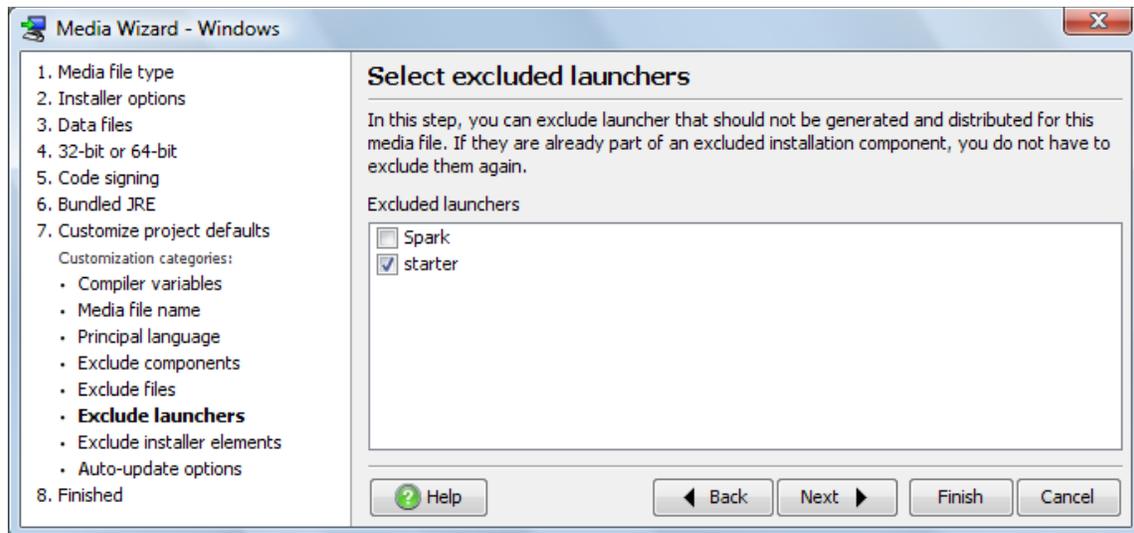


Figura 66. Eliminación de ejecutables

- Para terminar la configuración de *Media* se pulsa el botón OK.

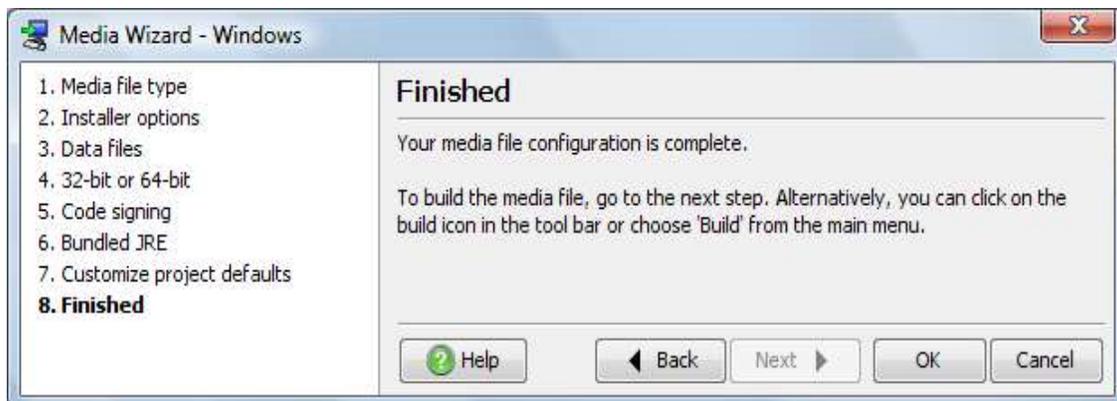


Figura 67. Terminar configuración media

- Finalmente, se inicia la creación del ejecutable, pulsando el botón *Start Build*.

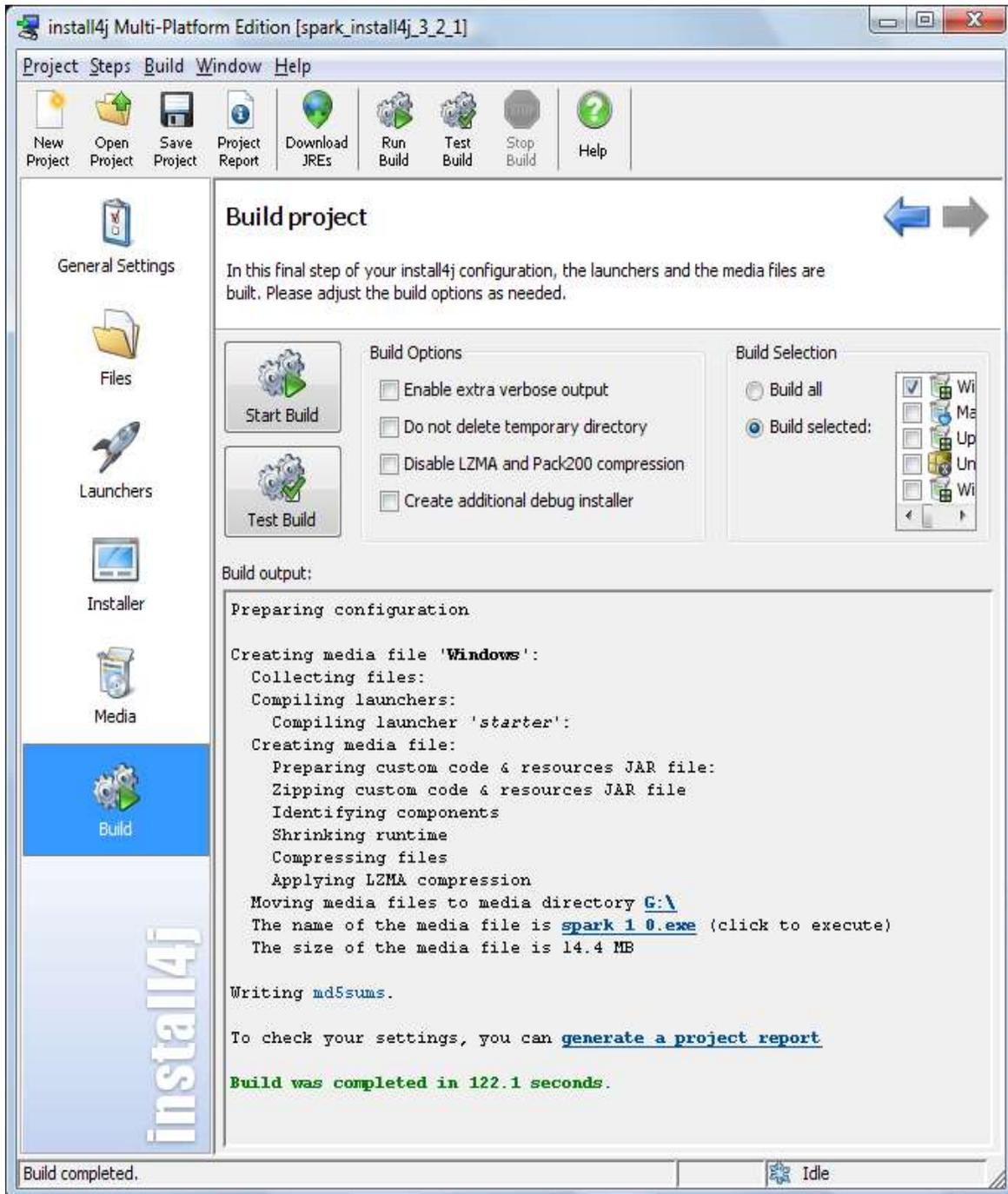


Figura 68. Creación del ejecutable

ANEXO F

MANUAL DE USUARIO

Anexo F: Manual de Usuario

Esta guía proporcionará los conocimientos básicos para usar el cliente de mensajería instantánea UnicaucaSpark. Se podrá conectar un usuario de forma predeterminada al servidor de mensajería y así acceder a los servicios que este proporciona. Este anexo hace referencia al prototipo final del proyecto.

F.1. Instalación

Para poder utilizar los servicios de mensajería instantánea, se debe ejecutar el archivo UnicaucaSpark.exe y seguir los pasos de instalación. Las figuras 69, 70 y 71 muestran los diferentes pasos de instalación.

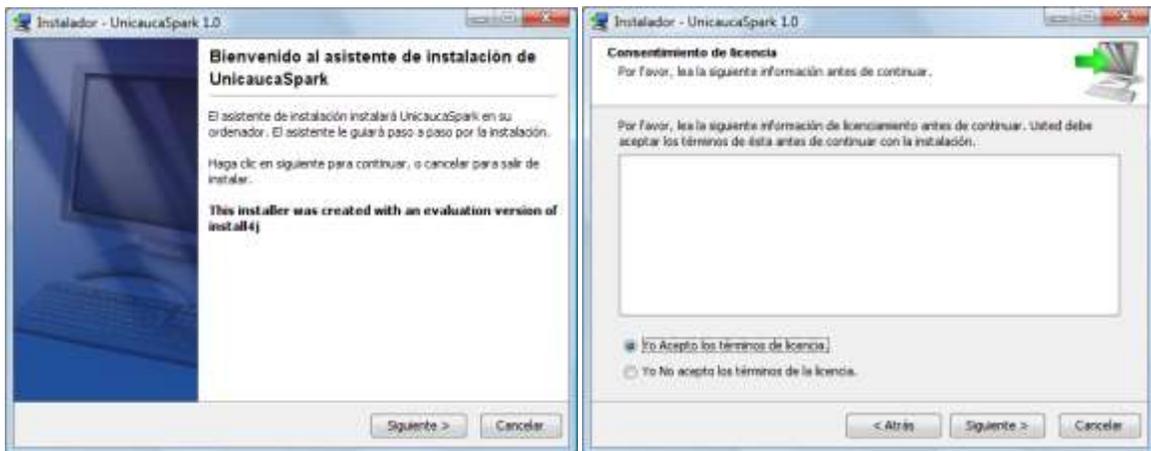


Figura 69. Pasos para la instalación de UnicaucaSpark 1

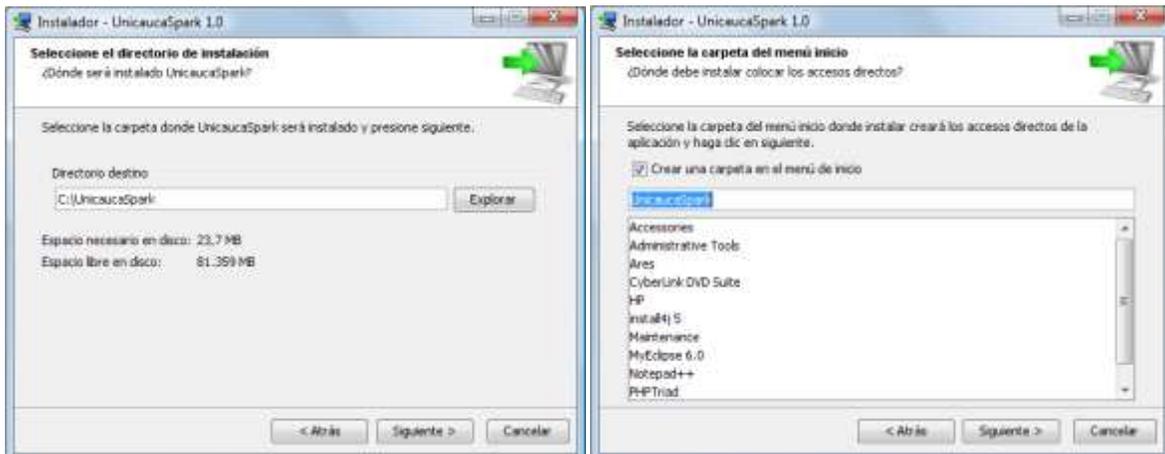


Figura 70. Pasos para la instalación de UnicaucaSpark 2

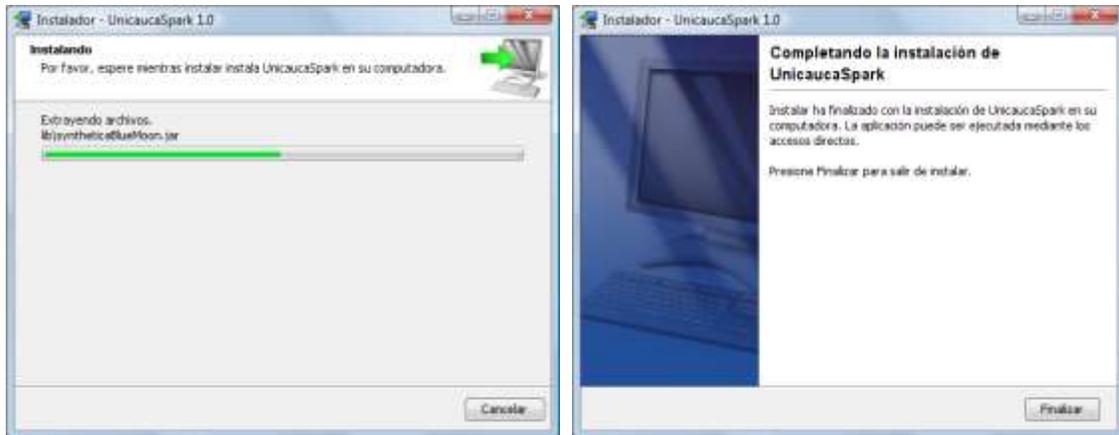


Figura 71. Pasos para la instalación de UnicaucaSpark 3

F.2. Iniciando UnicaucaSpark

Una vez instalado se inicia UnicaucaSpark, se introduce la información de login, username (nombre de usuario) y password (clave), y se pulsa la opción ingresar. La información del servidor o dominio del SMI será proporcionada de forma automática por la aplicación.



Figura 72. UnicaucaSpark

F.3. Conociendo UnicaucaSpark

Si se tiene noción en el uso de aplicaciones de mensajería instantánea, se encontrará todo muy familiar, y si no, se encontrará que es muy fácil de usar UnicaucaSpark.

Una vez conectado un usuario con UnicaucaSpark, se puede observar la pantalla principal. Después de que aparezca la lista de contactos pueden comenzarse sesiones de chat, y se encontrarán las cosas necesarias de forma fácil y rápida.

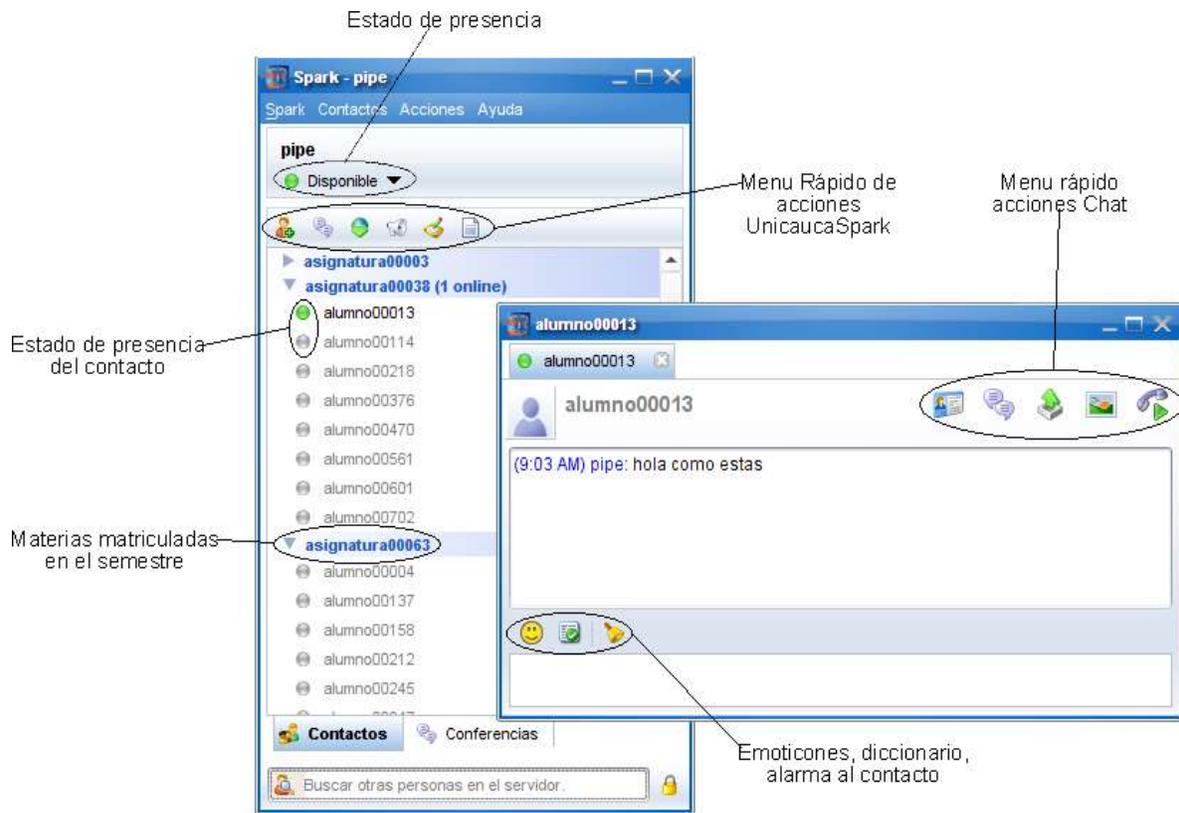


Figura 73. Vista funcional de UnicaucaSpark

F.4. Editar Perfil

El perfil es la forma de presentación de un usuario a los demás. Para cambiar dicha información se debe dar click en *Editar mi perfil* del menú *Spark*.

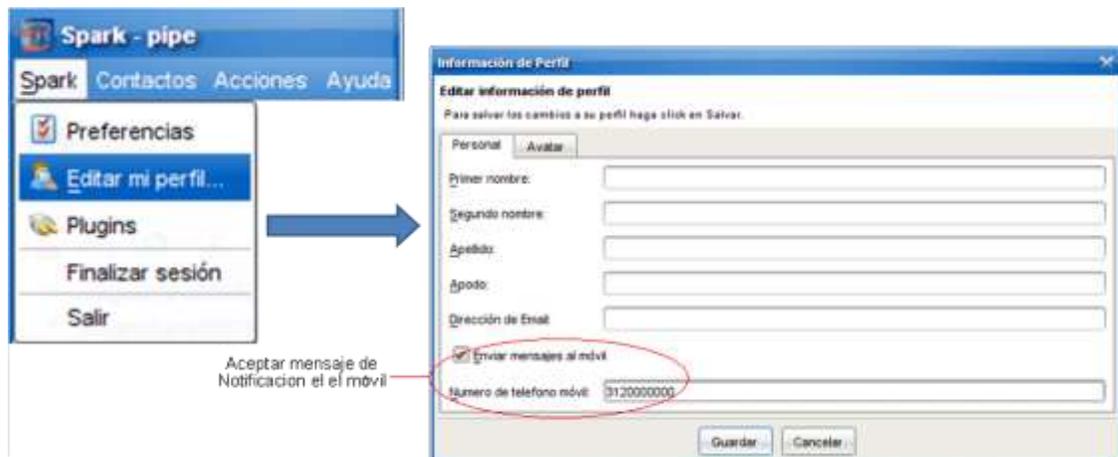


Figura 74. Editar perfil

En esta parte se puede elegir si se desea que lleguen mensajes de notificación al móvil, para hacerlo se debe chequear la opción *Enviar mensajes al móvil* y digitar el número de teléfono móvil.

F.5. Mensajes a Grupos

Cuando se tiene algo que decir a todos o varios de los contactos de la lista, se puede emitir un mensaje de broadcast. Para emitir estos mensajes se debe pulsar el botón de broadcast o en el menú de acciones dar *click* en *Difundir mensaje*:

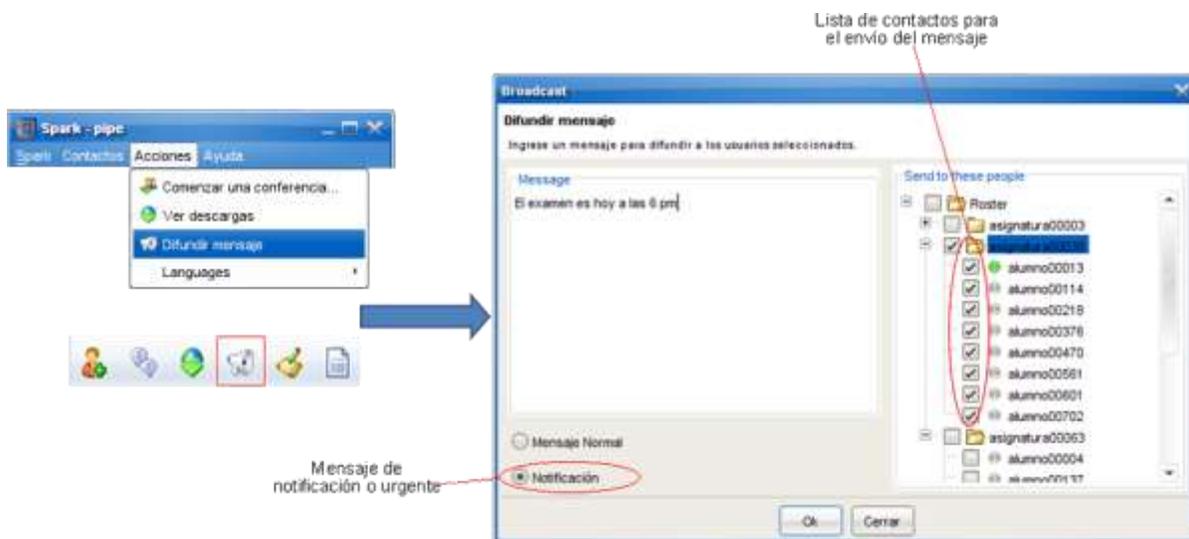


Figura 75. Enviar mensajes a grupos

Existen dos formas de difundir un mensaje. El primero es un mensaje normal o de importancia baja, el cual será enviado a los usuarios conectados en ese momento y le llegara a los no disponibles en el momento que inicien sesión. El segundo tipo de mensaje es de notificación o urgente en el cual a los usuarios que no estén conectados les llegará al móvil mediante un SMS (previa autorización en editar perfil), o se les enviará un correo electrónico con la información.

F.6. Agregar Contactos

La aplicación trae a la lista de contactos todas las personas que están matriculadas en las mismas materias que el usuario. Este tipo de usuarios no se podrán editar o borrar. Adicional a esto, se pueden agregar nuevos contactos, y para esto se debe dar click en el botón agregar contactos o en el menú *Contactos* señalar *Agregar contacto*.



Figura 76. Agregar contactos

Debe escribirse el username del contacto con el cual se quiere agregar y el apodo con el cual será reconocido. El grupo al cual va a pertenecer esta persona viene definido por la aplicación. Si se desea también, se puede copiar al grupo (que por defecto es privado) contactos que están en la lista de materias que entrega el sistema.

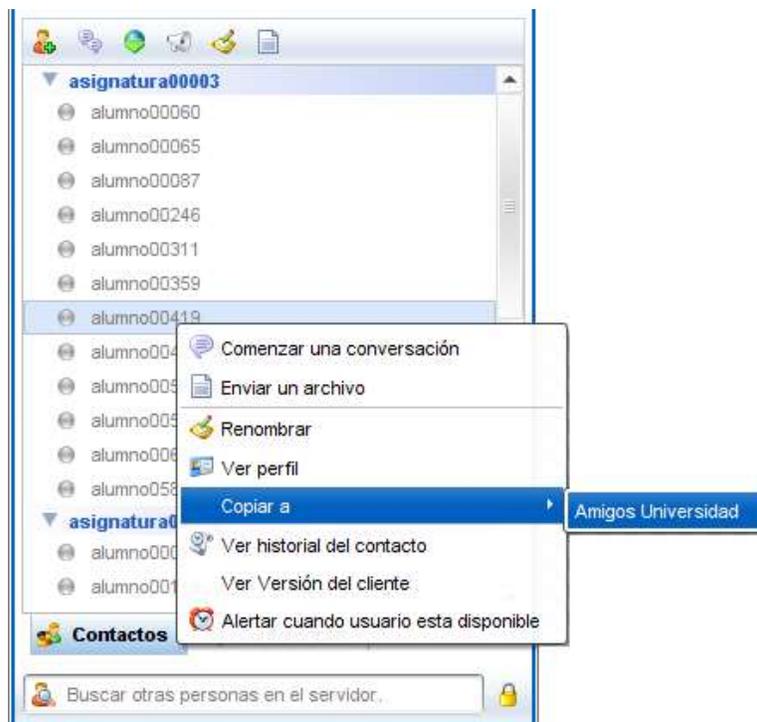


Figura 77. Copiar contacto de materia

Referencias Bibliográficas

- [1] Herlitz, Heinz; "Transversabilidad en NAT/Firewall". Disponible en: <<http://biblioteca.uct.cl/tesis/heinz-herlitz/tesis.pdf>> [Consulta: Enero de 2012].
- [2] Hamilton, Keith; "A Distributed Publish/Subscribe Event System". Disponible en: <<http://pubsub.codeplex.com/>> [Consulta: Enero de 2012].
- [3] Understanding Enterprise Application Integration - The Benefits of ESB for EAI. Disponible en: <<http://www.mulesoft.org/enterprise-application-integration-eai-and-esb>> [Consulta: Enero de 2012].
- [4] Bakken, David; "Middleware". Disponible en: <<http://www.dia.uniroma3.it/~cabibbo/ids/altrui/middleware-bakken.pdf>> [Consulta: Enero de 2012].
- [5] Curry, Edward; Chambers, Desmond; Lyons, Gerard; "Extending Message-Oriented Middleware using Interception". Disponible en: <http://www.edwardcurry.org/web_publications/curry_DEBS_04.pdf> [Consulta: Enero de 2012].
- [6] Richards, Mark. "The Role of the Enterprise Service Bus". Disponible en: <<http://www.infoq.com/presentations/Enterprise-Service-Bus>> [Consulta: Enero de 2012].
- [7] Daryl, Plummer; David, McCoy; Charles, Abrams; "Magic Quadrant for the Integrated Service Environment Market". Gartner Research. [Consulta: Agosto de 2011].
- [8] Jess, Thompson; Daniel, Sholler; Yefim, Natis; Massimo, Pezzin; Kimihiko, Lijima; "Gartner RAS Core Research". [Consulta: Enero de 2012].
- [9] Ruiz, Jonatan; Fernández, Francisco; "Integración de aplicaciones de seguridad". Arquitectura de Computadores. Febrero de 2009. [Consulta: Agosto de 2011].
- [10] 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].
- [11] Introducción a BizTalk Server 2009. MSDN. Disponible en: <[http://msdn.microsoft.com/es-ES/library/aa547058\(v=BTS.10\).aspx](http://msdn.microsoft.com/es-ES/library/aa547058(v=BTS.10).aspx)> [Consulta: Enero de 2012].
- [12] JBoss ESB Overview. JBoss Community. Disponible en: <<http://www.jboss.org/jbossesb>> [Consulta: Agosto de 2011].
- [13] Cerón, Ricardo; Vela, Gustavo; Caicedo, Oscar; "Arquitectura de referencia para sitios de redes sociales en ambientes móviles". Abril de 2009 [Consulta: Agosto de 2011].
- [14] De la Cruz, Edgar; Taimal, Hernán; Caicedo, Oscar; "Middleware de seguridad para el acceso a servicios IMS en un entorno 802.11". Abril de 2009 [Consulta: Agosto de 2011].