



Universidad de las Ciencias
Informáticas

Facultad 2

Implementación de protocolos para la comunicación con
Centros de Mensajería Inalámbrica.

Trabajo de Diploma

Autores: José Carlos Alonso Maceda.
Darién Jesús Álvarez de la Cruz.

Tutor: Lic. Tania Gual Martínez.

24 de mayo de 2007.
“Año 49 de la Revolución”

Agradecimientos

A la revolución, a nuestro comandante Fidel y a la UCI por permitirnos formar parte del proyecto futuro y por formarnos como profesionales revolucionarios

A nuestra tutora, a Abel, al viejo Damián F. y al personal de Procyon en general.

Darién y José Carlos

A mi amiguita Kariné y a Denys que fueron como mi familia aquí en la UCI.

A mi segundo padre Pepe, que me ayudo mucho.

A la profe Rosa Vázquez y a Roberto, sin ellos no hubiera podido terminar la universidad.

A mis padres, a mis abuelos, a mi familia en general, a mi querida esposa Dayma, a todos los que me ayudaron, incluso a los que no me ayudaron, ellos me hicieron ser más fuerte.

Darién

A mis amigos Ionian, Yunisel, mis compañeros de aula.

A mi madre María Eugenia, a mis hermanos Ernesto y Leonardo por el apoyo que me brindaron.

A mi hermano y amigo Ricardo, a Anita, y toda las personas que me ayudaron y que conocí.

José Carlos

Dedicatoria

A mi nenesa por ser mi apoyo y darme fuerzas para seguir adelante, en todos los momentos difíciles por los que he pasado en la UCI.

A mis padres por inculcarme el espíritu de lucha y no dejarme vencer nunca demostrándome que la única barrera para lograr lo imposible es el miedo y el cansancio.

A mi abuelito Juansi que es lo más lindo de mi vida y para Tata que también la quiero mucho y para mi hermanita linda, para toda la familia en general.

A mis compañeros, a los que están y en especial a los que no pudieron llegar al final.

A mis amigos Orlando, Yannys, José, Yanet y René son poco pero valen por miles.

A todos los que me han ayudado o saben que significan algo para mí.

Darién

A mi madre, en especial, por brindarme tanta felicidad, a David. A mi familia en general

A mis hermanos que supieron guiarme y orientarme.

A Fina e Ibrahim Nápoles familia que conocí y me ayudaron en esta universidad.

A Ricardo, Ionian y a todos aquellos que saben que siempre pensaré en ellos.

José Carlos

Resumen

Los servicios de mensajería inalámbrica como SMS, MMS y Wap Push, tienen actualmente un gran impacto en el mercado. Haciendo uso de estos, se puede generar una amplia variedad de servicios de valor agregado permitiendo a las diferentes empresas de este sector obtener ganancias. Existen diferentes protocolos que permiten la transferencia de estos mensajes a los Centros de Mensajería, actualmente la única especificación que estandariza las interfaces para la transferencia de SMS, MMS y WAP Push es SAMS-M, pero esta carece de drivers que implementen los protocolos existentes.

Por lo que, el objetivo de este trabajo es, desarrollar un *driver* para el protocolo PDU, para ser integrado en la implementación de SAMS-M, que permitan la transferencia de mensajes SMS, MMS y Wap Push entre las aplicaciones desarrolladas por las Empresas Proveedoras de Servicios de Valor Agregado y los Centros de Mensajería.

Este documento recoge los resultados obtenidos en el desarrollo de este trabajo. Se presentan las principales características del protocolo PDU centrado en el servicio de SMS y MMS. Además, se realiza el análisis y el diseño del driver, donde se obtiene la implementación del driver para envío y recepción de SMS. Al software desarrollado se le ejecutan las pruebas de aceptación y se realiza un análisis de los resultados obtenidos, y se elabora un análisis de la factibilidad del proyecto utilizando un método de estimación. Se establecen las conclusiones obtenidas del desarrollo del trabajo y a partir de esto, se realizan una serie de recomendaciones, que deben ser tomadas en cuenta para continuar la investigación.

Índice

Introducción	1
1. Fundamentación Teórica	4
1.1. Introducción	4
1.2. Estado del Arte.....	4
1.2.1. Comunicaciones Móviles.....	4
1.2.2. SAMS-M.....	7
1.2.3. Protocolos de Comunicación	10
1.2.4. PDU	11
1.2.4.1. PDU y el Servicio de Mensajería Corta.....	11
1.2.4.2. PDU y el Servicio de Mensajería Multimedia	14
1.2.5. Gestión del Módem GPRS/GSM.	17
1.2.5.1. Java Comm API.....	18
1.3. Estudio de alternativas de solución.	19
1.3.1. Puertas de Enlace	19
1.3.1.1. NowSMS	19
1.3.1.2. Kannel	20
1.3.2. APIs	21
1.4. Metodología para el desarrollo de la solución.....	23
1.4.1. Proceso de desarrollo de software.	23
1.4.2. Proceso Unificado de Desarrollo de Software	23
1.5. Herramientas para el Desarrollo de la solución.	25
1.5.1. Herramienta de Modelado UML.....	25
1.5.1.1 Rational Rose Enterprise Edition	25
1.5.2. Entornos Integrados de Desarrollo	25
1.5.2.1 Eclipse.....	26
1.5.3. Control de Versiones.	26
1.5.3.1 SVN.....	26
1.5.3.2. SubEclipse	27

1.6. Conclusiones del Capítulo.....	27
2. Características del sistema	28
2.1. Introducción	28
2.2. Objetivos estratégicos	28
2.3. Flujo de los procesos involucrados en el campo de acción.	29
2.4. Análisis crítico	32
2.5. Objeto de automatización.....	33
2.6. Propuesta de Sistema	33
2.7. Modelo del Dominio	34
2.7.1. Diagrama de Clases del Dominio	35
2.7.2. Descripción de los conceptos del Dominio	35
2.8. Requisitos	36
2.8.1. Requisitos Funcionales	36
2.8.2. Requisitos No Funcionales	40
2.9. Modelo de Casos de Uso del sistema.	42
2.9.1. Definición de los actores del sistema.....	43
2.9.2. Casos de Uso del sistema.	43
2.9.3. Diagrama de Casos de Uso.....	46
2.9.4. Distribución de Casos de Uso por ciclo de desarrollo	47
2.9.5. Casos de Uso extendidos.....	48
2.10. Conclusiones del Capítulo.....	57
3. Análisis y diseño del sistema	58
3.1. Introducción	58
3.2. Modelo de Análisis	58
3.2.1. Diagrama de clases del Análisis	59
3.3. Modelo de Diseño	60
3.3.1. Métodos de diseño de drivers.....	60
3.3.2. Subsistemas del Diseño	62
3.3.4. Diagramas de Clases del Diseño.....	63
3.3.5. Diagramas de Interacción.....	65

3.3.6. Patrones de Diseño.....	65
3.3.6.1. Patrones GRASP.....	66
3.3.6.2. Patrones GoF.....	66
3.3.7. Lineamientos para el código fuente.....	67
3.4. Conclusiones del Capítulo.....	67
4. Implementación y prueba del sistema.....	68
4.1. Introducción.....	68
4.2. Modelo de Implementación.....	68
4.2.1. Diagrama de Despliegue.....	69
4.2.2. Diagrama de Componentes.....	70
4.3. Modelo de prueba.....	71
4.3.1. Casos de Prueba.....	71
4.3.1.1. Caso de prueba. Enviar mensaje SMS.....	71
4.3.1.2. Análisis de los defectos.....	72
4.3.2.1. Caso de prueba: Recibir mensaje SMS.....	73
4.3.2.2. Análisis de los defectos.....	72
4.4. Conclusiones del Capítulo.....	74
5. Estudio de Factibilidad.....	75
5.1. Introducción.....	75
5.2. Método utilizado estimación basada en Puntos de Casos de Uso.....	75
5.2.1. Cálculo de los Puntos de Casos de Uso sin ajustar.....	75
5.2.2. Cálculo de los Puntos de Casos de Uso ajustados.....	76
5.2.3. Estimación de esfuerzo a través de los Puntos de Casos de Uso.....	78
5.2.3.1. Calcular esfuerzo de todo el proyecto.....	79
5.2.4. Beneficios tangibles e intangibles.....	79
5.2.5. Análisis de costos y beneficios.....	80
5.3. Conclusiones del Capítulo.....	80
Conclusiones.....	81
Recomendaciones.....	82
Bibliografía.....	83
Anexos.....	86

Anexo 1. Encapsulación de PDU por capas del Modelo OSI.....	86
Anexo 2. Encapsulación del datagrama PDU.....	87
Anexo 3. Diagramas de Colaboración del Análisis.	89
Anexo 4. Diagramas de Secuencia del Diseño.....	93
Anexo 5. MMS en formato PDU.	101
Anexo 6. Muestra del error identificado.	102
Glosario de Términos.....	103

Figuras

Figura 1 Entorno J2SE de SAMS-M.....	8
Figura 2 Entorno J2EE de SAMS-M.....	9
Figura 3 Tipos de mensajes TPDU entre un SME y un SMSC	12
Figura 4 Estructura de un mensaje TPDU.....	13
Figura 5 Flujo de mensajes para el envío y recepción de mensajes MMS	15
Figura 6 Flujo de mensajes para cancelar mensajes MMS	16
Figura 7 Estructura de un mensaje MMS	17
Figura 8 Entrada y salida de un proceso de desarrollo de software	23
Figura 9 Arquitectura global de RUP	24
Figura 10 Entorno de aplicaciones para transferencia de mensajes a Centros de Mensajería.	30
Figura 11 Entorno de la entidad Procyon para transferencia de mensajes utilizando NowSMS	31
Figura 12 Entorno aplicación de SAMS-M con un driver para el protocolo PDU	34
Figura 13 Modelo del Dominio del Driver para el protocolo PDU.....	35
Figura 14 Diagrama de Casos de Uso del sistema	46
Figura 15 Diagrama de Clases del Análisis.....	59
Figura 16 Subsistemas del Diseño.....	62
Figura 17 Diagrama de Clases del Diseño. Subsistema SMS.....	63
Figura 18 Diagrama de Clases del Diseño. Subsistema MMS.	64
Figura 19 Diagrama de Clases del Diseño. Subsistema HardwareHandler.....	65
Figura 20 Diagrama de Despliegue.....	69
Figura 21 Diagrama de Componentes.	70
Figura 22. Componentes de un datagrama del protocolo PDU	87
Figura 23. Procesamiento de un datagrama para transmisión.	87
Figura 24. Procesamiento de un datagrama para recepción	88
Figura 25. Diagrama de Colaboración del Análisis. Enviar SMS.	89
Figura 26. Diagrama de Colaboración del Análisis. Enviar MMS.....	89
Figura 27. Diagrama de Colaboración del Análisis. Recibir SMS.	90
Figura 28. Diagrama de Colaboración del Análisis. Recibir MMS.....	90

Figura 29. Diagrama de Colaboración del Análisis. Cancelar SMS.	91
Figura 30. Diagrama de Colaboración del Análisis. Cancelar MMS.	91
Figura 31. Diagrama de Colaboración del Análisis. Administrar Módem (Iniciar conexión GSM).	92
Figura 32. Diagrama de Colaboración del Análisis. Administrar Módem (Iniciar conexión GPRS).	92
Figura 33. Diagrama de Colaboración del Análisis. Estado del SMS.	92
Figura 34. Diagrama de Secuencia del Diseño. Cancelar MMS.	93
Figura 35. Diagrama de Secuencia del Diseño. Cancelar SMS.	94
Figura 36. Diagrama de Secuencia del Diseño. Administrar Módem.	95
Figura 37. Diagrama de Secuencia del Diseño. Recibir MMS.	96
Figura 38. Diagrama de Secuencia del Diseño. Recibir SMS.	97
Figura 39. Diagrama de Secuencia del Diseño. Enviar MMS.	98
Figura 40. Diagrama de Secuencia del Diseño. Enviar SMS.	99
Figura 41. Diagrama de Secuencia del Diseño. Estado del SMS.	100
Figura 42. MMS en formato PDU (M-Send.req)	101
Figura 43. Error identificado.	102

Tablas

Tabla 1 Justificación de los actores del sistema.....	43
Tabla 2 Caso de Uso del Sistema. Enviar mensaje SMS.....	43
Tabla 3 Caso de Uso del Sistema. Recibir mensaje SMS.....	44
Tabla 4 Caso de Uso del Sistema. Enviar mensaje MMS.....	44
Tabla 5 Caso de Uso del Sistema. Recibir mensaje MMS.....	44
Tabla 6 Caso de Uso del Sistema. Consultar estado del SMS.....	45
Tabla 7 Caso de Uso del Sistema. Cancelar mensaje SMS.....	45
Tabla 8 Caso de Uso del Sistema. Cancelar mensaje MMS.....	45
Tabla 9 Caso de Uso del Sistema. Administrar Módem.....	45
Tabla 10 Ciclo de Desarrollo 1.....	47
Tabla 11 Ciclo de Desarrollo 2.....	47
Tabla 13 Ciclo de Desarrollo 3.....	48
Tabla 14 Expansión del Caso de Uso. Enviar mensaje SMS.....	49
Tabla 15 Expansión del Caso de Uso. Recibir mensaje SMS.....	50
Tabla 16 Expansión del Caso de Uso. Enviar mensaje MMS.....	51
Tabla 17 Expansión del Caso de Uso. Recibir mensaje MMS.....	52
Tabla 18 Expansión del Caso de Uso. Consultar estado del SMS.....	53
Tabla 19 Expansión del Caso de Uso. Cancelar SMS.....	54
Tabla 20 Expansión del Caso de Uso. Cancelar mensaje MMS.....	55
Tabla 21 Expansión del Caso de Uso. Administrar Módem.....	56
Tabla 22 Caso de Prueba. Enviar Mensaje SMS.....	72
Tabla 23 Análisis de Defectos. Enviar Mensaje SMS.....	72
Tabla 24 Caso de Prueba. Recepción de Mensaje SMS.....	73
Tabla 25 Factor de peso de los actores sin ajustar.....	76
Tabla 26 Factor de peso de los casos de uso sin ajustar.....	76
Tabla 27 Factor de complejidad técnica.....	77
Tabla 28 Factor de ambiente.....	78
Tabla 29 Esfuerzo del proyecto.....	79

Introducción

En el devenir de estos últimos 30 años las comunicaciones inalámbricas han tomado gran fuerza mundialmente. Esto ha derivado en el desarrollo de nuevas tecnologías de comunicación donde una de las de mayor fuerza es la telefonía celular. Producto a este auge y desarrollo alcanzado en tan corto periodo en la Telefonía Inalámbrica existe una amplia variedad de servicios, entre ellos los servicios de mensajería, que reportan ganancias millonarias a las empresas proveedoras de servicios de telefonía celular, los “Operadores”. Estos servicios han abierto un mercado muy provechoso a nivel mundial para empresas proveedoras de servicios de valor agregado VASP (*del inglés: Value Added Service Provider*), las cuales desarrollan aplicaciones que ofrecen servicios de mensajería como chat, horóscopo, postales, estado del tiempo, votaciones, noticias, entre otros, que por ser de gran aceptación entre los clientes, representan una alta fuente de ingresos.

Desde el punto de vista de los VASP, la dificultad radica en realizar aplicaciones que de manera estándar puedan acceder a los Centros de Mensajería, entidades responsables de manejar los mensajes a través de la red inalámbrica. Esto se debe a que, con el desarrollo acelerado del software y el hardware, estos servicios han evolucionado y se han creado múltiples protocolos de comunicación con los Centros de Mensajería. Cada protocolo, permite el acceso a uno o varios servicios, implementa sus propias funcionalidades y los datos se reciben y envían de diferentes maneras.

Múltiples intentos de desarrollar estándares fueron llevados a cabo por distintas empresas, sin llegar a un punto de convergencia donde los servicios de mensajería fueran independientes de los Operadores. En el año 2004 surge SAMS-M (*del inglés: Server APIs for Mobile Services - Messaging*) JSR-212 especificación desarrollada principalmente por Nokia y Sun. Esta especificación plantea una arquitectura estándar para el envío y recepción de mensajes cortos SM (*del inglés: short message*), mensajes multimedia MM (*del inglés: multimedia message*) y notificaciones WAP Push (*del inglés: Wireless Application Protocol Push*). La arquitectura de esta API propone la implementación de *drivers* para cada protocolo de comunicación, permitiendo a los VASP abstraerse de la forma de conexión con los Centros de Mensajería y del manejo de transferencia de los mensajes.

JSR-212 SAMS-M fue implementada por la empresa Nokia, pero su uso no es gratuito y los drivers que fueron desarrollados responden solo a los intereses de dicha compañía. En la entidad Procyon, ubicada en la UCI, se realizó una nueva implementación de SAMS-M, de código abierto, a pedido de la compañía Sueca fabricante de tecnología celular "Sony Ericsson", pero los drivers para los protocolos que resultan de interés para dicha empresa no están disponibles en el mercado mundial.

Utilizando una implementación concreta de SAMS-M se logra estandarizar la interface para envío y recepción de mensajes SMS, MMS y WAP Push. Esta implementación aumenta su valor agregado según la cantidad de drivers implementados cubriendo los distintos protocolos existentes, este trabajo está orientado al desarrollo de un driver para el protocolo PDU (*del inglés: Protocol Data Unit*). Existen diferentes soluciones que implementan este protocolo a nivel mundial pero su código no es libre en la mayoría de los casos y resulta muy difícil adaptarlo a SAMS-M.

Por tanto el **problema** queda formulado de la siguiente manera: ¿Cómo desarrollar un driver para SAMS-M que permitan la transferencia de mensajes entre los diferentes Centros de Mensajería y las aplicaciones de los VASP, a través del protocolo PDU?

El **objeto de estudio** de este trabajo es el desarrollo de un driver del protocolo PDU para ser integrados a una implementación concreta SAMS-M.

De aquí se deriva el **campo de acción** que abarca el desarrollo de un driver para el protocolo PDU que garantice el proceso de recepción y envío de mensajes, a través de un módem con conexión GPRS (*del inglés: General Packet Radio Service*), desde y hacia los Centros de Mensajería Inalámbrica.

El **objetivo general** del trabajo es desarrollar un *driver* para el protocolo PDU, para ser integrado en la implementación de SAMS-M, que permitan la transferencia de mensajes SMS, MMS y Wap Push entre las aplicaciones desarrolladas por los VASP y los Centros de Mensajería.

Para el desarrollo del trabajo se han formulado las siguientes **preguntas de investigación**:
¿Cuáles son las características del protocolo PDU?

¿Cómo integrar una implementación del protocolo PDU a SAMS-M?

¿Cómo realizar la comunicación con el módem GPRS a través de puertos físicos de la computadora?

¿Cómo comprobar la factibilidad del driver?

Para el cumplimiento del objetivo general y dar respuesta a los cuestionamientos planteados se han trazado las siguientes **tareas de investigación**:

1. Caracterizar el protocolo PDU haciendo un análisis de las funcionalidades que brinda para el envío y recepción de SMS, MMS y WAP Push.
2. Elaborar un driver para SAMS-M utilizando el protocolo PDU.
3. Probar el driver en ambiente real para el envío y recepción de SMS.
4. Realizar un estudio de la factibilidad del driver utilizando un método de estimación de costo.

Este trabajo ha sido estructurado por capítulos. En el Capítulo I, titulado “Fundamentación teórica”, se realiza un análisis sobre el protocolo PDU y sus características. Se mencionan los principales conceptos referentes a los mensajes para teléfonos celulares, se analizan las tecnologías a utilizar, realizando la selección de las más adecuadas, así como, las herramientas y la metodología para el desarrollo del sistema; además fundamenta las bases para el desarrollo del Capítulo II “Características del sistema” donde se propone una solución realizando una descripción detallada a través de un Modelo de Dominio y los requisitos funcionales y no funcionales del sistema. Los principales aspectos relacionados con la construcción de la solución propuesta se abordan en el Capítulo III, “Análisis y diseño del sistema”, en este se modelan los diagramas de clases de análisis y de diseño, y se especifican los principios la implementación. En el Capítulo IV, titulado “Implementación y prueba del sistema”, se especifican los detalles de la implementación, a través del Modelo de Implementación, y se analizan los resultados obtenidos en las pruebas efectuadas. Por último, en el Capítulo V “Estudio de Factibilidad” se muestra el costo estimado del proyecto y un análisis de los beneficios.

Entre los resultados obtenidos en esta investigación podemos mencionar el análisis y el diseño de un driver para el protocolo PDU teniendo en cuenta las operaciones soportadas por este protocolo en mensajes del tipo SMS, MMS y WAP Push. Se realizó la implementación del envío y recepción de mensajes SMS y WAP Push, además se logró integrar esta implementación a SAMS-M realizando las pruebas pertinentes para su utilización.

1

1. Fundamentación Teórica

1.1. Introducción

En este capítulo se tratan los principales conceptos y términos abordados en la investigación, enfatizando en SAMS-M y los aspectos relevantes de su arquitectura para la realización de un *driver* para el protocolo PDU. Este protocolo será analizado vinculado específicamente a los servicios de SMS (*del inglés: Short Message Service*) y MMS (*del inglés: Multimedia Message Service*). Se mencionan además, las tecnologías, herramientas y metodología utilizadas en el desarrollo de la solución.

1.2. Estado del Arte

1.2.1. Comunicaciones Móviles

Con el surgimiento de la Telefonía Inalámbrica nació una nueva rama para el desarrollo de las Telecomunicaciones. Una amplia gama de estándares de red para teléfonos móviles, protocolos de comunicación y servicios, fueron surgiendo con la evolución del hardware y el software.

Uno de los estándares más difundidos a nivel mundial es Sistema Global para Comunicaciones Móviles **GSM** (*del inglés: Global System for Mobile Communications*). GSM soporta múltiples servicios entre ellos se destaca el servicio de mensajería corta SMS. Es un servicio de telefonía móvil universalmente aceptado, que posibilita la transmisión de mensajes cortos SM alfanuméricos entre usuarios móviles y

sistemas externos como correo electrónico, *paging* y sistemas de correos de voz. **(International Engineering Consortium)**

Se puede mencionar que un SM es un mensaje de texto breve enviado desde un dispositivo móvil MO (*del ingles: Mobile Originated*), hacia un dispositivo móvil MT (*del ingles: Mobile Terminated*), a través de un SMS. Un SM estándar consiste en 160 caracteres alfanuméricos máximo, aunque se pueden enviar mensajes con mayor cantidad de caracteres utilizando compresión de datos. **(TechTarget)**

Desde el punto de vista de los Operadores de Telefonía Celular los SMS no se envían punto a punto, por lo cual, se necesita una entidad en la red inalámbrica encargada del proceso. Esta entidad conocida, como Centro de Servicio de Mensajes Cortos SMSC (*del inglés: Short Message Service Center*), es responsable de la transmisión, almacenamiento y reenvío de SM entre entidades de mensajería corta SME (*del inglés: Short Message Entity*). **(International Engineering Consortium)**

SMS es uno de los servicios que brindan las redes GSM. El estándar GSM continuó su desarrollo manteniendo la compatibilidad y con la versión 97 del este estándar desarrollada por 3GPP (*del ingles: 3rd Generation Partnership Project*) surgió el Servicio General de Radio-Paquetes **GPRS**, donde se adicionaron capacidades de empaquetamiento de datos. Con el surgimiento de GPRS se adicionaron nuevos servicios, entre los más conocidos se puede mencionar el Servicio de Mensajes Multimedia MMS, diseñado específicamente para GPRS.

MMS es una tecnología de comunicaciones que permite a los usuarios intercambiar mensajes multimedia entre teléfonos móviles aptos y otros dispositivos. MMS es una extensión de SMS, que define una manera de enviar y recibir, casi instantáneamente, Mensajes Multimedia MM. **(TechTarget, 2005)**

Los MM pueden contener imágenes, audio, video y texto. La presentación del mensaje esta descrita en un “archivo de presentación” específicamente un archivo SMIL (*del inglés: Synchronized Multimedia Integration Language*) de manera que las imágenes, sonidos, videos y texto se muestran en el orden establecido en ese archivo. **(TechTarget, 2005)**

Al igual que SMS el servicio MMS necesita una entidad encargada de la transferencia de los mensajes, esta se conoce como Centro de Servicio de Mensajes Multimedia MMSC (del inglés: Multimedia Message Service Center). Provee facilidades de almacenamiento y reenvío de MM a través, de la red Inalámbrica. Suministra la funcionalidad de darle formato y optimizar los mensajes para ser enviados a los diferentes dispositivos según sus características como tamaño de pantalla, cantidad de colores, formato de sonido soportado, entre otras. **(MPirical)**

Otro de los servicios relevantes que soporta el estándar GSM y por consiguiente GPRS son las Notificaciones WAP (*WAP Push*) estas típicamente hacen referencia a un MMS, email, ó para descargas de contenidos como imágenes, juegos, entre otros. **(Wikipedia)**

Específicamente se puede decir que un WAP Push es un SMS que consiste en dos secciones, un mensaje de texto y un URL relacionado a una página WAP (*del inglés: Wireless Application Protocol*), donde el teléfono móvil receptor del mensaje muestra el texto y provee al usuario la opción de conectarse al URL con solo presionar una tecla. **(AME Info, 2003)**

La entidad de red encargada de la administracion de este servicio se denomina PPG (*del inglés: Push Proxy Gateway*) que envía notificaciones con un URL a un teléfono móvil. **(Wikipedia)**

Los mensajes SMS, MMS y WAP Push, fueron utilizados de forma tradicional hasta el instante que diferentes empresas desarrollaron los primeros servicios de valor agregado, VAS (*del inglés: Value Added Service*), en este campo. A partir de ese momento, un simple mensaje SMS que se enviaba desde un teléfono celular a otro se convirtió en toda una industria de aplicaciones con diversos servicios que incluían noticias, votaciones, horóscopo, chat, reporte del tiempo, entre muchos otros. Con el surgimiento de los MMS, los usuarios obtuvieron nuevas aplicaciones que combinaron imágenes, sonidos, videos y texto donde experimentaron una experiencia más agradable.

La existencia de múltiples protocolos de comunicación y diferentes servicios de mensajería como SMS, MMS y WAP Push representa un problema para los desarrolladores de estos productos. Esta industria necesitaba un estándar que regulara las interfaces para el desarrollo de aplicaciones con servicios de

SMS, MMS y WAP Push con una arquitectura que permitiera agregar implementaciones para cada uno de los protocolos de comunicación con Centros de Mensajería, entiéndase por Centros de Mensajería Inalámbrica los SMSC, MMSC y PPG, este estándar fue llamado API Servidora para Servicios Móviles de Mensajería **SAMS-M** y no fue desarrollado hasta el 2004, como una alternativa a la variedad de soluciones propietarias que existían en el mercado donde cada una de estas definía cuales protocolos implementar. SAMS-M representa el primer estándar que se define para homogeneizar las interfaces de envío y recepción para las aplicaciones desarrolladas por VASP.

Este API (*del inglés: Application Programming Interface*) ha sido implementada por la entidad Procyon en la Universidad de las Ciencias Informáticas a petición de la empresa Sony Ericsson, pero se carece de las implementaciones para los protocolos de comunicación con los Centros de Mensajería Inalámbrica, por lo cual este trabajo propone el desarrollo de una implementación **driver** para el protocolo PDU que se integre a SAMS-M.

Es importante aclarar que esta investigación no tiene como objetivo explicar en detalle la especificación SAMS-M, sin embargo se exponen algunos detalles de la arquitectura y la implementación que son relevantes para este trabajo.

1.2.2. SAMS-M

Cuando se menciona el término SAMS-M nos referimos al JSR-212 (*del inglés: Java Specification Request-212*) que define un API estándar para SMS, MMS y WAP Push independiente del protocolo y portable, que permite enviar y recibir mensajes cortos, mensajes multimedia y notificaciones WAP. (**Java Community Process, 2005**)

Este API de alto nivel es independiente de los protocolos de red, del formato de datos y de las plataformas para la comunicación con Centros de Mensajería. Su arquitectura permite la adición en tiempo de ejecución de drivers para protocolos de comunicación, y se ofrece la posibilidad de especificar que driver, con implementaciones específicas para SMS, MMS y WAP Push debe ser usado. Esta arquitectura permite la adición de nuevos estándares de mensajes además de SMS, MMS y WAP Push.

La especificación fue desarrollada originalmente por Nokia y BEA, actualmente le dan soporte Nokia, BEA y Sun Microsystem y se diseñó para las plataformas J2SE (*Java 2 Standard Edition*) y J2EE (*Java 2 Enterprise Edition*). La siguiente figura muestra la arquitectura de SAMS-M en un entorno J2SE.

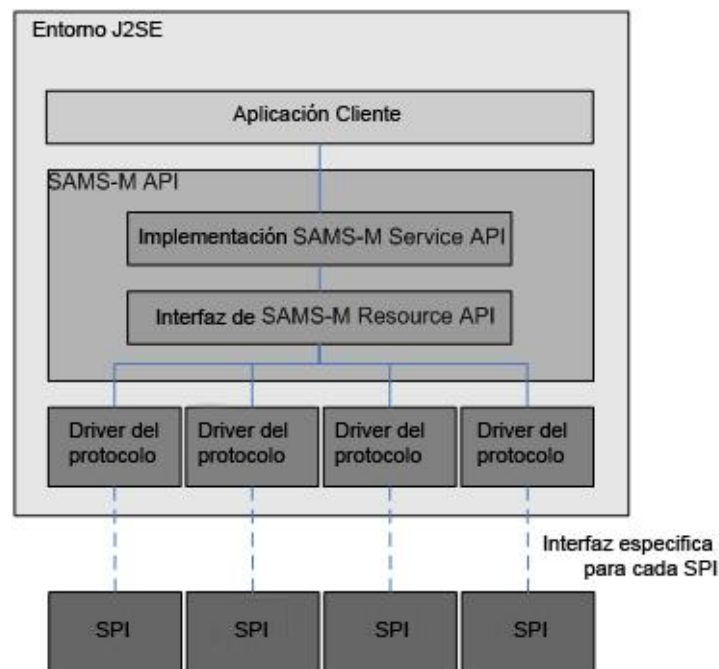


Figura 1 Entorno J2SE de SAMS-M

Como muestra la figura 1, SAMS-M está compuesta por dos capas fundamentales: la **capa de servicios** que provee las interfaces necesarias a las aplicaciones clientes para la transferencia de mensajes y, la **capa de recursos** que proporciona las interfaces para las operaciones que deben implementar los drivers. Para cada protocolo se desarrolla un **driver** que es capaz de realizar las operaciones de envío y recepción a través de las Interfaces de los Proveedores de Servicios SPI (*del inglés: Service Provider Interface*).

Como se había mencionado anteriormente SAMS-M provee soporte para la plataforma J2EE separando la **capa de servicios** dentro de una aplicación Enterprise Java Bean, EJB, y la **capa de recursos** y los **drivers** en una Arquitectura Conectora de Java JCA (*del inglés: Java Connector Architecture*). La siguiente figura muestra la arquitectura de SAMS-M en un entorno J2EE.

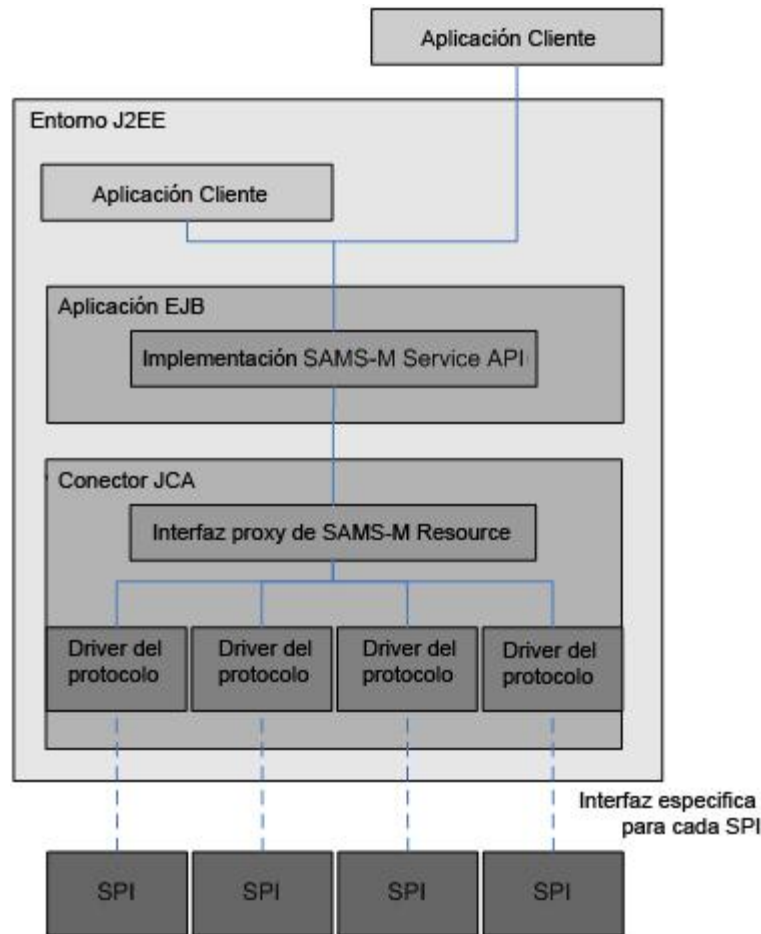


Figura 2 Entorno J2EE de SAMS-M

Las arquitecturas planteadas para entornos J2SE y J2EE hacen de SAMS-M una solución factible para aquellas empresas que utilizan Java como plataforma de desarrollo. SAMS-M no puede utilizarse con aplicaciones clientes desarrolladas en otras plataformas que no sean las mencionadas, aunque esto es una limitante no podemos considerarlo como una restricción para el desarrollo de la solución debido a que no forma parte de los requerimientos que se han planteado.

Para el desarrollo de un driver basado en la implementación concreta de SAMS-M y las interfaces que provee, es necesario: construir **clases concretas** que implementen las siguientes interfaces que forman parte de la capa de recursos de SAMS-M: MMSConnectionFactory, SmsConnectionFactory según el

patrón de diseño Factory Method y son encargadas de construir instancias de MMSConnection y SmsConnection respectivamente, estas a su vez son interfaces de las que se deben construir clases concretas. Estas clases contienen los métodos necesarios para enviar, cancelar, reemplazar y consultar el estado de los mensajes, además un método para enviar notificaciones de entrega. La implementación de la interfaz ResourceAdapter inicializa las clases encargadas de la recepción de los mensajes. Otro de los requisitos para construir un driver es: conocer con profundidad el protocolo de comunicación que se debe implementar.

1.2.3. Protocolos de Comunicación

Para la comunicación con los Centros de Mensajería y el intercambio de SMS, MMS y Wap Push existen una amplia gama de protocolos para la comunicación con SMSC podemos encontrar: CIMD, PDU, CIMD 2, EMI/UCP, SMPP, OIS y OISD. En el caso del servicio de MMS y la conexión con MMSC algunos de los más conocidos son: MM7, MM1 y EAIF.

Cada uno de estos protocolos define su propio formato, algunos de estos se comunican a través de conexión por socket, otros envían los datos por HTTP, además también difieren en la forma de enviar los parámetros, algunos utilizan un formato XML, otros un flujo de caracteres codificados, en resumen existen múltiples vías dependiendo cada una del protocolo y los datos que se envían.

El propósito de este trabajo es obtener un driver de SAMS-M para el Protocolo de Unidad de Dato PDU que permita el envío y recepción de SMS, MMS y WAP Push, por lo que resulta relevante señalar algunos de los motivos que nos conducen a escoger este protocolo entre las diversas opciones existentes.

1. La empresa Sony Ericsson como cliente especificó que se debía implementar ese protocolo con posibilidades de manejar SMS, MMS y WAP Push.
2. Este protocolo no necesita contratos especiales con Operadores de Telefonía.
3. Resulta muy barato para los VASP utilizar este protocolo para el desarrollo de aplicaciones con servicios de mensajería.

Básicamente se puede decir que para que una aplicación cuente con servicios de SMS, MMS y WAP Push solo se necesita una implementación del protocolo PDU, un teléfono celular ó módem que soporte el

estándar GPRS/GSM y un contrato tradicional con un Operador de Telefonía, es decir, un número telefónico.

Un módem GPRS/GSM es un tipo de módem especializado el cual acepta tarjetas SIM (*del inglés: Subscriber Identity Module*), y opera sobre una suscripción a un operador de telefonía, igual que un teléfono celular. Desde la perspectiva de un Operador de Telefonía, un módem GPRS/GSM luce igual que un teléfono celular. No obstante, este muestra una interface que permite a las aplicaciones enviar y recibir mensajes sobre la interfaz del módem. Este tipo de módem utiliza el protocolo PDU para transmitir los mensajes. **(Now Wireless Limited)**

1.2.4. PDU

En telecomunicaciones el término PDU (**Federal Standard 1037C, 1996**) tiene los siguientes significados:

- Información que es entregada como una unidad entre un par de entidades en una red y que debe contener información de control, dirección o datos.
- En sistemas por capas, una unidad de datos contiene el mensaje de la capa superior y la información del protocolo de control de información PCI (*del inglés: Protocol Control Information*) de la capa actual.

Este protocolo es utilizado para la transferencia de datos entre las distintas capas del modelo OSI (*del inglés: Open Interface Specification*) (*Anexo 1. Encapsulación de PDU por capas del Modelo OSI*), todos los datos se encapsulan según las especificaciones del protocolo (*Anexo 2. Encapsulación del datagrama PDU*). Este trabajo se concentra en la relación de PDU con los servicios de mensajería inalámbrica, por lo cual los aspectos relevantes para la investigación se enmarcan en los servicios de mensajería corta y mensajería multimedia.

1.2.4.1. PDU y el Servicio de Mensajería Corta

La pila de protocolos SMS esta compuesta por cuatro capas: **capa de aplicación, capa de transferencia, capa de transmisión y capa de enlace**. Aplicaciones basadas en SMS son directamente basadas en la capa de transferencia, consecuentemente, cualquier ingeniero que tenga la intención de desarrollar aplicaciones con servicios de SMS, necesita administrar la capa de transferencia.

En la **capa de transferencia**, el mensaje es considerado como una secuencia de octetos que contienen información como la longitud del mensaje, el origen o el destino del mensaje, fecha de recepción, entre otros. La capa de transferencia se conoce frecuentemente como *capa de transferencia de mensajes cortos SM-TL* (del inglés: *short message – transfer layer*). (Le Bodic, 2003)

Básicamente existen seis tipos de mensajes PDU en el caso del servicio de mensajería corta SMS, estos se enumeran a continuación:

- SMS-Submit: esta transacción corresponde al envío de un segmento de mensaje desde un SME a un SMSC, finalizado este paso el SMSC reconoce el envío retornando un SMS-Submit-Report al SME.
- SMS-Deliver: esta transacción corresponde a la entrega de un segmento de mensaje desde un SMSC a un SME, después de haber realizado la entrega el SME retorna un SMS-Delivery-Report reconociendo la entrega del segmento de mensaje.
- SMS-Status-Report: esta transacción corresponde a la entrega de un reporte de estado desde un SMSC a un SME.
- SMS-Command: esta transacción corresponde a la petición de SME de ejecución de un comando específico en un SMSC.

Para ilustrar este proceso proponemos la próxima figura donde se muestra el origen y el destino de cada uno de estos tipos de mensaje:

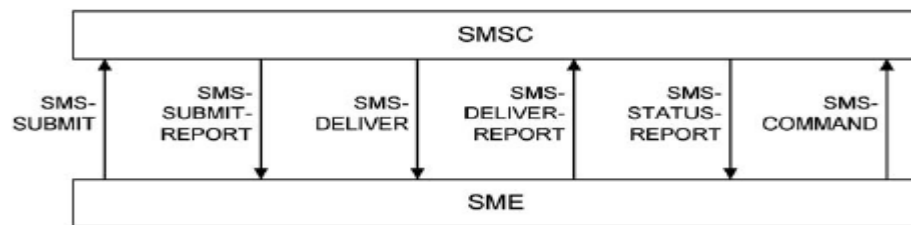


Figura 3 Tipos de mensajes TPDU entre un SME y un SMSC

Con el propósito de realizar el transporte, una aplicación mapea el contenido del mensaje y las instrucciones de entregas asociadas al Protocolo de Transferencia de Unidad de Datos TPDU (del inglés: *Transfer Protocol Data Unit*) en la capa de transferencia (3GPP Specification TS 23.040, 2006). La siguiente figura muestra la estructura de un TPDU.

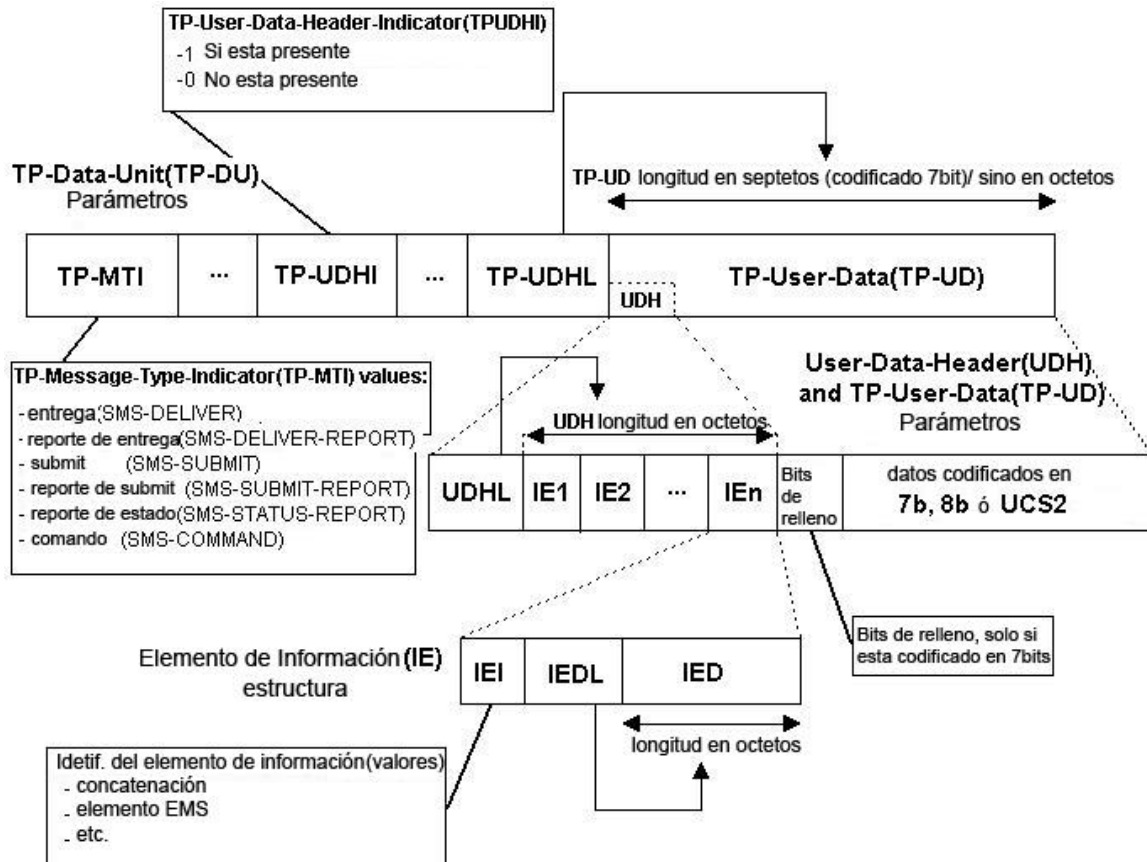


Figura 4 Estructura de un mensaje TPDU

Un **TPDU** esta compuesto por varios parámetros indicando el tipo de mensaje, especificando si se desea solicitar o no reporte de estado, contiene la parte de texto del mensaje, entre otros, por tanto si se desea enviar un SMS a través del protocolo PDU es necesario realizar este proceso. A cada parámetro se le establece como prefijo TP de protocolo de transferencia, algunos de estos parámetros son TP-Message_Indicator (TP-MTI), TP-Status-Report-Indication (TP-SRI), entre otros. Es posible que el contenido de un mensaje sea mayor que el tamaño soportado por un TPDU por lo cual es necesario dividir el mensaje en varios segmentos y enviarlo de esa forma, algo similar ocurre en la recepción donde los segmentos deben concatenarse para formar un mensaje SMS.

PDU para el Servicio de Mensajería Corta difiere del que se conforma para el Servicio de Mensajería Multimedia, incluso la forma de comunicación también varia, en el siguiente epígrafe se explica en detalle esta afirmación.

1.2.4.2. PDU y el Servicio de Mensajería Multimedia

En el servicio de mensajería multimedia MMS el protocolo PDU tiene el propósito de encapsular el mensaje. **(WAP Forum, 2002)** El servicio WAP WSP/HTTP es utilizado para transferir mensajes multimedia entre el terminal y el MMSC. El Protocolo de Sesión Inalámbrica WSP (*del ingles: Wireless Session Protocol*) forma parte de la pila de protocolos WAP1. **(WAP Forum, 2001)**

Básicamente existen ocho tipos de mensajes en el protocolo PDU para MMS especificados en WAP MMS Client Transactions **(WAP Forum, 2002)**, estos se enumeran a continuación haciendo referencia a la operación que realizan:

- M-Send.req y M-Send.conf. (Envío de mensajes al MMS Proxy-Relay).
- WSP/HTTP Get.req y M-Retrieve.conf. (Recibir mensajes desde el MMS Proxy-Relay).
- M-Notification.ind y M-NotifyResp.ind. (Notificación sobre un nuevo mensaje MMS).
- M-Delivery.ind. (Reporte de entrega sobre un mensaje enviado).
- M-Acknowledge.req. (Acuse de recibo de mensaje entregado).

La siguiente figura muestra como ocurren las transacciones para estos tipos de mensajes entre un MMS Proxy Relay que es la interfaz de un MMSC y un cliente con soporte para MMS.

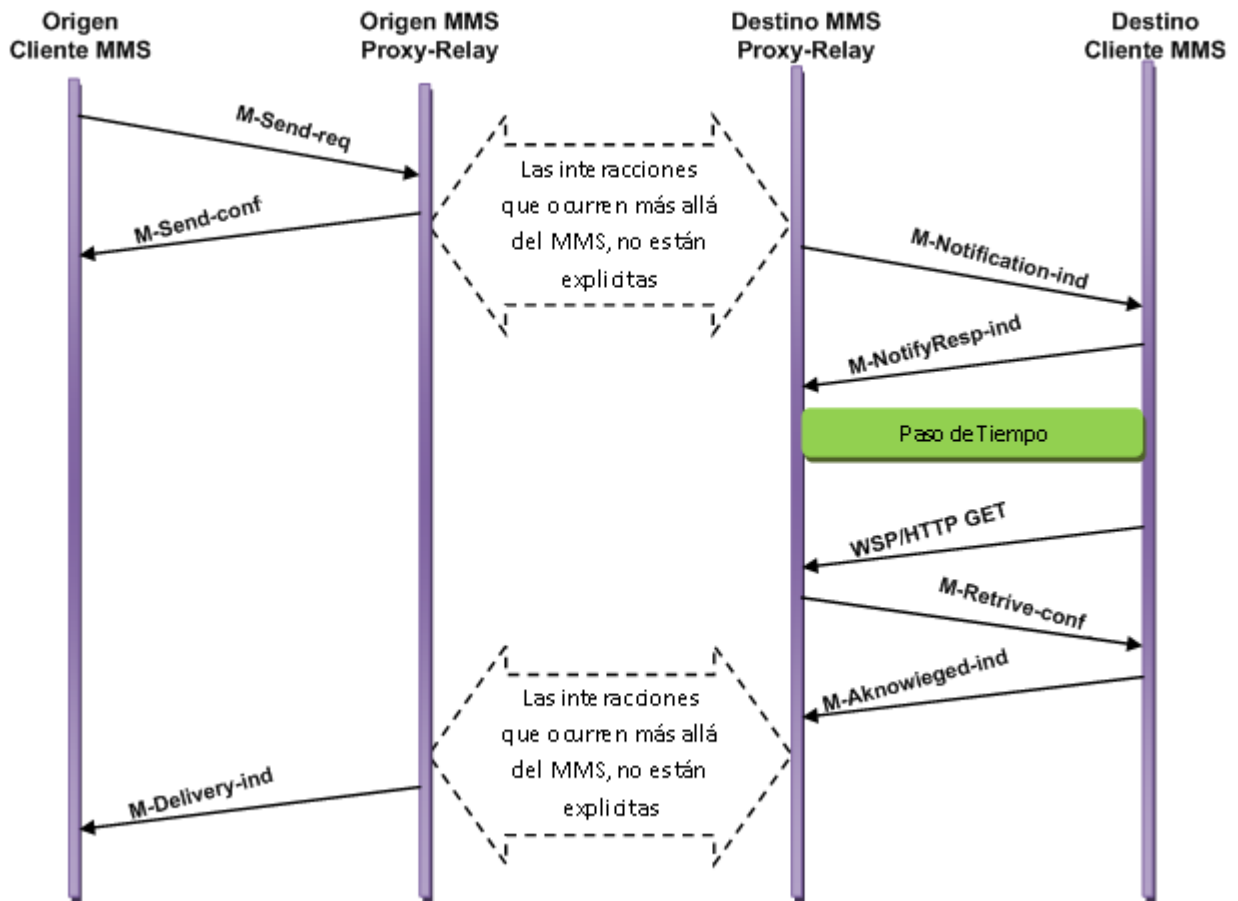


Figura 5 Flujo de mensajes para el envío y recepción de mensajes MMS

Existen otros tipos de mensajes en el protocolo PDU para MMS según el estándar de OMA Multimedia Messaging Service Client Transactions (**OMA, sep2005**), la operación de cancelar un mensaje MMS, esta resulta de interés debido a que es una acción requerida según los métodos que ofrecen las interfaces de recursos de SAMS-M.

- M-Cancel.req y M-Cancel.conf. (Cancelar un mensaje MMS).

En el caso de la operación de cancelación de un MMS, se puede observar el flujo de mensajes en el siguiente gráfico.

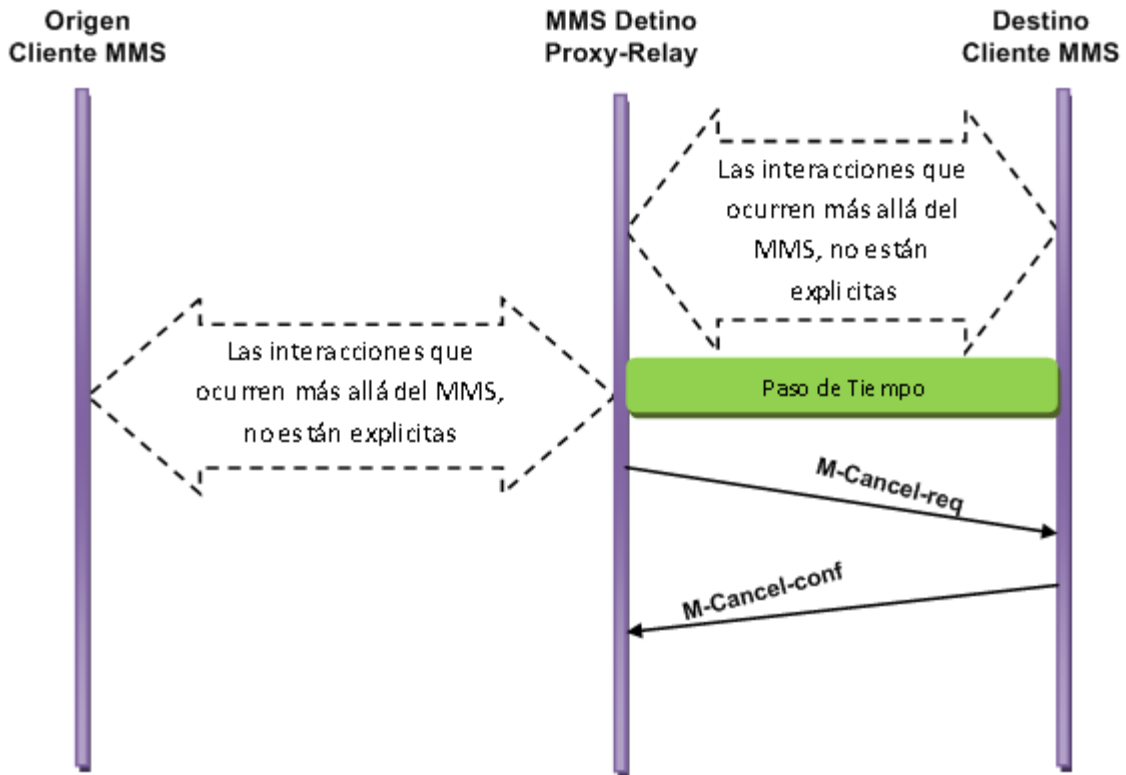


Figura 6 Flujo de mensajes para cancelar mensajes MMS

Lógicamente el PDU consiste en un conjunto de encabezados y un cuerpo conocido como *multipart*, el cuerpo del mensaje solamente es transmitido en el caso del envío de un mensaje multimedia (M-Send.req) o de la recepción satisfactoria de un mensaje (WSP/HTTP Get.req). El cuerpo del mensaje debe contener cualquier tipo de contenido multimedia, incluyendo los contenidos pre asignados especificados en el estándar WAPWSP. Un mensaje MMS se representa con el tipo *application/vnd.wap.mms-message*. La siguiente figura muestra la estructura de un mensaje MMS.

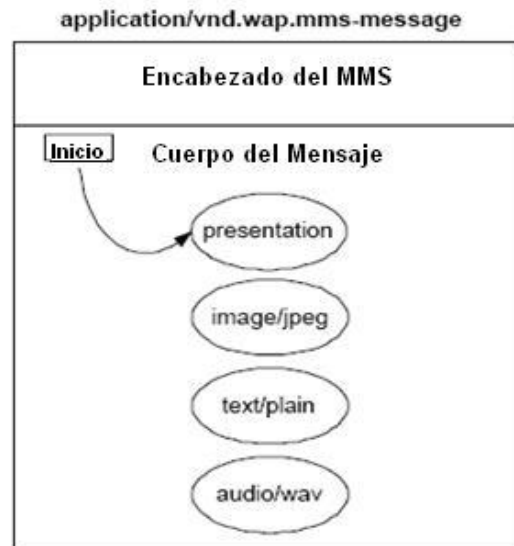


Figura 7 Estructura de un mensaje MMS

Cada uno de los encabezados y partes del cuerpo del mensaje son codificados según el protocolo PDU para MMS, utilizando el conjunto de reglas definidas para cada uno de los tipos de mensajes. (*Anexo 5. MMS en formato PDU.*)

Como se ha mencionado se necesita un módem GPRS para el envío y recepción de mensajes utilizando el protocolo PDU. El siguiente epígrafe hace alusión a los elementos necesarios para la gestión de la comunicación con el módem.

1.2.5. Gestión del Módem GPRS/GSM.

Para gestionar la comunicación con el módem se necesita un API. El principal propósito de un API consiste en proporcionar un conjunto de funciones de uso general y que los programadores se beneficien de estas, evitándose el trabajo de programar todo desde el principio.

De esta forma, existen diferentes APIs para administrar la entrada y salida de datos por los puertos de la PC que proporcionan diversos paquetes brindando funcionalidades muy útiles, como el manejo de los **puertos de serie** (RS-232), **paralelo** (IEEE-1284), y **USB**, que resultan de interés para la gestión del módem.

Algunas de estas se utilizan para diferentes tareas como: **RCX Java API** utilizada en la robótica, **jUSB: Java USB** utilizada para la conexión con cámaras digitales, mp3s, entre otros, haciendo uso del puerto USB pero sin una versión estable para Windows; **Java™ USB API** y **Java Comm API** que implementan la especificación **JSR-80**, esta última implementa una serie de funcionalidades y paquetes que proporciona todas las características necesarias para este trabajo, de las cuales hacemos mención.

1.2.5.1. Java Comm API

El paquete Java Comm API nos permite realizar comunicaciones con los puertos serie RS-232 (*del inglés: Recommended Standard-232*) y el paralelo IEEE-1284 (*del inglés: Standard Signaling Method for a Bi-directional Parallel Peripheral Interface for Personal Computers*). Esto posibilita realizar aplicaciones que utilizan los puertos de comunicaciones (tarjetas inteligentes, fax, módem) independientemente de la plataforma, por tanto Java Comm API reúne un grupo de funcionalidades que nos facilita el trabajo.

Este API, proporciona control total de todos los parámetros de los puertos serie: velocidad en baudios, bits de stop, paridad, bits por trama; así como control manual o automático de las líneas de control de flujo. Con el uso de la especificación RS-232, hay dos líneas de señal y el resto son líneas de control; dependiendo del tipo de comunicación, síncrona o asíncrona, el número de líneas de control seleccionadas puede variar. El API proporciona acceso a las líneas de control fundamentales.

Añadir a esto que el uso del paquete de comunicaciones *javax.comm* existente en esta API tenemos una serie de clases que nos permiten tratar varios niveles de programación y varios servicios tales como:

- Obtener los puertos disponibles así como sus características.
- Abrir y mantener una comunicación en los puertos.
- Resolver colisiones entre aplicaciones. Con este servicio se puede tener varias aplicaciones Java funcionando y utilizando los mismos puertos, además, no solo se conocerá si el puerto esta ocupado, sino que, se sabrá que aplicación lo esta utilizando.

Después de haber analizado los principales servicios, el protocolo, y el API de comunicación a utilizar, sobre los que se enfoca el desarrollo de este trabajo, se realizará un estudio del tema y los productos

desarrollados a nivel mundial, efectuando un análisis de algunas de las principales alternativas que existen paralelamente a este desarrollo.

1.3. Estudio de alternativas de solución.

Entre los diferentes productos analizados se pueden mencionar algunas plataformas como OSA Parlay (del inglés: Open Service API Parlay) ó IPX (del inglés: Internet Payment Exchange) en las que no se hará mucho hincapié debido a que no son los más difundidos. A modo de ejemplo, se puede decir que IPX es una tecnología propietaria de la Sony Ericsson, por lo tanto, solo los Centros de Mensajería fabricados por esta empresa la soportan. OSA Parlay es una especificación que maneja los servicios de los Centros de Mensajería a “bajo nivel”, es decir, provee interfaces para las distintas prestaciones del hardware a través de un Parlay Gateway, por lo tanto, los Operadores deben implementar un Parlay Gateway y los desarrolladores implementar el OSA Parlay. Otra alternativa sería utilizar una implementación propietaria donde por supuesto se deben pagar costosas licencias para su uso.

Se puede destacar entre los productos analizados algunas APIs y Puertas de Enlace (Gateways) que son ampliamente usados a nivel mundial, para estos productos se realiza una breve explicación mencionando las software más utilizados.

1.3.1. Puertas de Enlace

Estos productos proveen a los usuarios una forma de conexión con protocolos de comunicación bien conocidos como HTTP de esta forma se posibilita la transferencia de mensajes ocultando al usuario final los distintos protocolos de comunicación con los Centros de Mensajería para los servicios de SMS, MMS y WAP Push. Algunos de estos implementan sus propios SMSC o MMSC de manera que pueden comunicarse con SMSC reales utilizando los protocolos específicos establecidos para ese tipo de comunicación. Los productos de este tipo líderes en el mercado son NowSMS y Kannel sobre estos se realiza un breve análisis a continuación.

1.3.1.1. NowSMS

NowSMS es un SMS/MMS Gateway que provee una solución para la entrega de contenidos SMS y MMS. Representa una manera rápida para construir y desplegar soluciones de SMS, MMS y Wap Push. Es fácil de instalar y esta desarrollado para los sistemas operativos Windows NT/2000/XP.

NowSMS incluye las características siguientes:

- Soporta la conectividad para SMS por uno o más módems GSM (o teléfonos GSM conectados por un puerto de serie a la PC), o conexiones TCP/IP usando protocolos tales como SMPP, CIMD, UCP/EMI y/o del HTTP.
- Soporta el envío y recepción de MMS, a través de la interfaz de MM1 con el protocolo PDU se puede realizar la comunicación con el MMSC utilizando un módem de GPRS/GSM sin una cuenta especial del operador.
- Es capaz de generar y hacer una transferencia mensajes de MMS, e incluye un compilador de MMS además de un encabezado binario así como el formato para mensaje, requeridos para el contenido de MMS.

NowSMS es una solución muy práctica para empresas que se inician en el mundo de la mensajería para móviles debido a que proporciona una interfaz de usuario muy amigable para la configuración, es fácil de utilizar ya que cuenta con una interfaz WEB que administra los diferentes servicios como SMS, MMS y WAP Push, además suministra una implementación del protocolo PDU que permite utilizar un Módem GSM/GPRS para la comunicación. Su desventaja radica en que para acceder a los servicios del NowSMS es necesario utilizar la interfaz Web y esta no tiene servicios para cancelar mensajes, eliminar mensajes, además no brinda reportes de estado o de entregas del mensaje. Otras desventajas son que esta desarrollado en C++ por lo tanto no se ajusta a la arquitectura de SAMS-M y es código propietario, hay que pagar licencia para su uso.

1.3.1.2. Kannel

Kannel es un Wap Gateway de código abierto. Intenta proveer una parte de la infraestructura WAP sin tener que pagar licencia para su uso, tratando de potenciar el mercado de servicios WAP, tanto para el operador como para el proveedor de servicios de valor agregado, materializando este intercambio de la manera más eficiente.

Kannel también trabaja como un SMS Gateway para redes GSM. Casi todos los teléfonos GSM pueden recibir y enviar mensajes SMS. Es una forma de servir a muchos más clientes que solo usando teléfonos con servicio de navegación WAP. Además opera como un PPG haciendo posible para los servidores de contenido enviar sus datos a los teléfonos. Este tipo de servicio WAP tiene aplicaciones muy interesantes.

Algunos de las características que soporta son:

- Completamente compatible con la especificación WAP 1.1.
- Además implementa algunas características de WAP 1.2 y de WAP 2.0.
- Soporta WAP Push SI y SL
 - Soporta gran variedad de protocolos de comunicación con SMSC, entre ellos UCP/EMI 4.0 y 3.5, CIMD 1.37 and 2.0, SMPP 3.4, SMS2000 OIS 4.0, 5.0 and 5.8 y PDU utilizando Módems GSM.
- Soporte completo para MO mensajes y MT mensajes.

Kannel ha sido reconocido como una de las mejores soluciones de este tipo en el mercado con la gran ventaja de ser un proyecto (Open Source) con licencia BSD. Sus desventajas radican en que, su código esta desarrollado en C por lo tanto no se ajusta a la arquitectura de SAMS-M, es difícil de configurar no provee interfaces amigables a los usuarios, a pesar que es multiplataforma resulta difícil de instalar sobre Windows.

Otra de las alternativas utilizadas a nivel mundial son las APIs, estas implementan algunos de los protocolos de comunicación para los servicios de SMS, MMS y WAP Push.

1.3.2. APIs

API conjunto de especificaciones de comunicación entre componentes de software necesarias que facilitan el desarrollo de un aplicación. A continuación se enumeran algunas de las APIs que se tuvieron en cuenta al realizar un estudio de alternativas a este desarrollo.

- jSMS API: Provee servicios de envío y recepción de SMS, MMS y Wap Push. Desarrollada en Java. Su principal desventaja es el costo de su licencia que oscila entre 40 y 2495 USD

anualmente en dependencia de los servicios que se necesiten. Algunas de las funcionalidades que brinda se muestran a continuación:

- Funciona con la mayoría de los dispositivos GSM que soportan en envío y recepción de SMS sobre RS-232.
- Implementa los protocolos UCP, CIMD2 y SMPP.
- Envío y recepción de MMS usando el protocolo PDU sobre MM1.
- SwiftNote SMSJDK: permite el envío y recepción de SMS y Wap Push sobre diferentes protocolos. Su desventaja radica en que no soporta servicios para los MMS y no es gratuita. Algunas de las principales funcionalidades se enumeran a continuación:
 - Implementa los protocolos UCP, CIMD2, SMPPP 1.1 y SMPP 3.3/3.4.
 - Envío y recepción síncrona y asíncrona de mensajes.
 - Soporte para control de flujos y monitoreo automático de la conexión con el SMSC y provee facilidades de reconexión cuando esta se pierde.
- SMSLIB es un API con licencia LGPL, que permite el envío y recepción de SMS a través de módems GSM. Es un API muy simple y no permite envío de MMS, en eso radica su principal desventaja, pues sería necesario implementar el soporte para MMS o en otro caso habría que utilizar otra API como complemento.

Se confirmó que en el mercado existen diferentes productos que proveen de forma parcial una solución a la situación problemática planteada, pero en resumen poseen las siguientes restricciones:

1. La mayoría de estos productos son de código propietario y se necesita pagar las licencias por ellos.
2. Algunos de ellos proveen soluciones parciales, se necesita más de un producto para cubrir nuestras necesidades.
3. Cada uno de estos productos proveen interfaces diferentes para la comunicación.
4. No se acogen a ninguna normativa o estándar para los parámetros, las interfaces de comunicación, entre otros, para comunicarse con las aplicaciones desarrolladas por los VASP.

Por estos motivos se considera realizar la implementación de un driver para el protocolo PDU integrado a SAMS-M como interface para las aplicaciones clientes.

1.4. Metodología para el desarrollo de la solución.

Todo desarrollo de software es riesgoso y difícil de controlar, si no se utiliza una metodología que guíe el proceso, se obtienen clientes insatisfechos con los resultados y desarrolladores aún más insatisfechos.

1.4.1. Proceso de desarrollo de software.

Un proceso de Desarrollo de Software define QUIEN esta haciendo QUE, CUANDO y COMO alcanzar un objetivo. En la ingeniería de Software el objetivo es construir un producto de software o mejorar uno existente. Un proceso efectivo proporciona normas para el desarrollo eficiente de software de calidad. Captura y presenta las mejores prácticas que el estado actual de las tecnologías permite. En consecuencia reduce el riesgo y hace el proyecto más predecible. El efecto global es el fomento de una visión y una cultura comunes. **(Jacobson, y otros)**

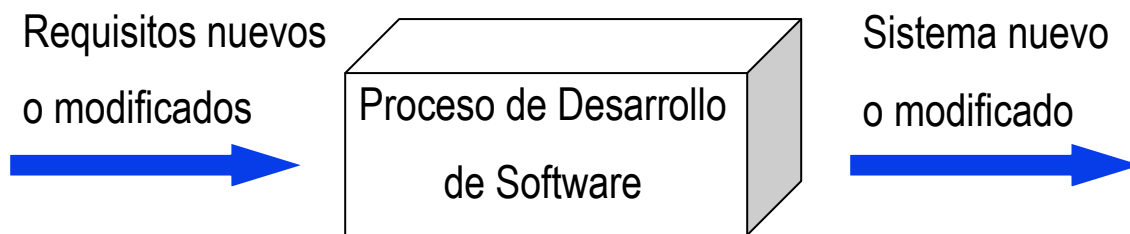


Figura 8 Entrada y salida de un proceso de desarrollo de software

No existe un Proceso de Software Universal, esto se debe a las características de cada proyecto, los equipos de desarrollo, los recursos, los riesgos, entre otros, por lo tanto se exige que los procesos sean configurables.

Es necesario usar la metodología adecuada para reducir el tiempo de desarrollo del software y aumentar la calidad del producto, obviamente se debe haber llegado a un acuerdo formal con el cliente, al inicio del proyecto, de tal manera que cada cambio no represente un problema. La metodología que se utiliza es el Proceso Unificado de Desarrollo de Software.

1.4.2. Proceso Unificado de Desarrollo de Software

El Proceso Unificado de Desarrollo de Software RUP (*del inglés: Rational Unified Process*), surgido en 1988 define una metodología aplicable a proyectos de gran envergadura aunque puede ser ajustada para

trabajos más pequeños. Es un proceso en dos dimensiones donde encontramos Fases que son: Inicio, Elaboración, Construcción y Transición, cada una de ellas compuesta de una o varias iteraciones y los Flujos de Trabajo que son: Modelado del Negocio, Requisitos, Análisis y Diseño, Implementación, Pruebas y Despliegue, como flujo de apoyo encontramos Entorno, Gestión de Proyectos, Gestión de Configuración y Cambios. **(Jacobson, y otros)**

Estas fases revelan que para producir una versión del producto en desarrollo se emplean todas las actividades de ingeniería pero con diferente énfasis en cada flujo de trabajo; en las primeras versiones se hace más énfasis en el modelado del negocio, requisitos, análisis y diseño; mientras en las posteriores el énfasis recae sobre las actividades de implementación, pruebas y despliegue.

En la siguiente figura se muestra el esfuerzo que se emplea en cada Flujo de Trabajo según la Fase de Desarrollo del Proyecto.

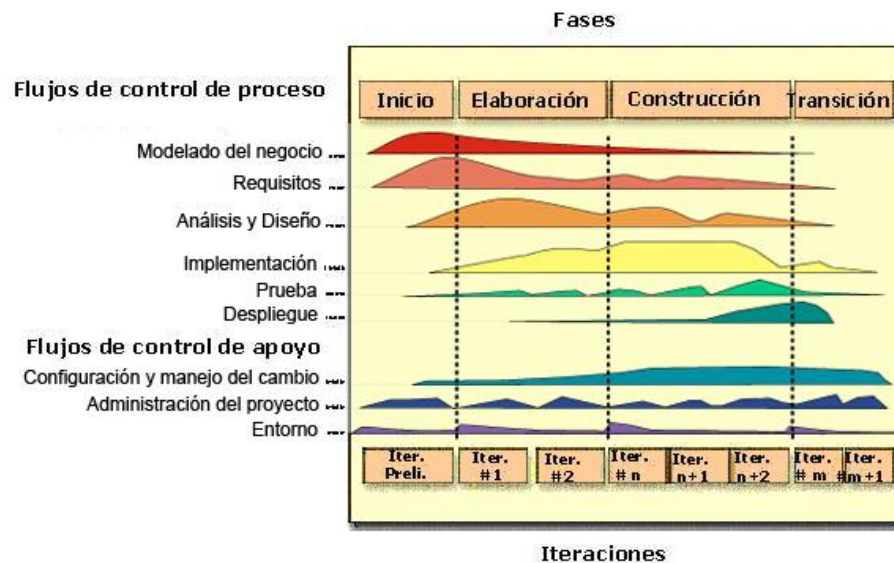


Figura 9 Arquitectura global de RUP

Es necesario mencionar la existencia de herramientas que ayudan a perfeccionar el resultado, haciendo más simple el uso de la metodología obteniendo un producto con calidad.

1.5. Herramientas para el Desarrollo de la solución.

Las tecnologías actuales ayudan considerablemente a solventar muchos detalles que años pasados retrasaban el desarrollo, mejores IDEs, herramientas CASE, sistemas operativos y hardware logran que al integrarse se aumente el desempeño de un grupo de desarrollo.

Con dicha integración la productividad aumenta, pues gracias a funcionalidades como la verificación de código sin compilar, depuración por pasos, generación de documentación automática y control de versiones ayudan de manera significativa al desarrollo de aplicaciones robustas. A continuación se menciona las distintas herramientas que han sido utilizadas para el desarrollo de este trabajo.

1.5.1. Herramienta de Modelado UML

Como herramienta de modelado se utilizó Rational Rose Enterprise Edition 2003, herramienta líder en esta rama, en la que posee mayor experiencia por parte del equipo de desarrollo de este trabajo.

1.5.1.1 Rational Rose Enterprise Edition

Rational Rose Enterprise Edition 2003, importante para el Modelado Visual mediante UML de sistemas software. Permite Especificar, Analizar, Diseñar el sistema antes de codificarlo. Mantiene la consistencia de los modelos del sistema software. Chequeo de la sintaxis UML. Generación Documentación automáticamente. Generación de Código a partir de los Modelos. Ingeniería Inversa (crear modelo a partir código); entre otras funcionalidades, facilita el desarrollo de un proceso cooperativo en el que todos los agentes tienen sus propias vistas de información (vista de Casos de Uso, vista Lógica, vista de Componentes y vista de Despliegue). Permite que los arquitectos y diseñadores practiquen el desarrollo orientado al modelado, permitiéndoles producir modelos independientes a la, arquitectura de la plataforma del software y necesidades del negocio.

1.5.2. Entornos Integrados de Desarrollo

Se conoce como IDE (*del inglés: Integrated Development Environment*) a un entorno integrado de desarrollo, consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica. En este trabajo se utilizará Eclipse como IDE para la implementación del sistema, a continuación exponemos algunas de las características del mismo.

1.5.2.1 Eclipse

Eclipse es una plataforma que proporciona un marco de trabajo o framework para desarrollar aplicaciones. Esto significa que distintos fabricantes pueden desarrollar e integrar sus herramientas con el WorkBench existente. El WorkBench es un conjunto de frameworks de Java y herramientas de desarrollo equipadas para constructores de herramientas. Eclipse se considera como uno de los IDE más poderosos para el desarrollo de aplicaciones en Java, provee soporte mediante *plugins* para trabajar con entornos J2EE y J2SE utilizados en el desarrollo de este trabajo. **(Eclipse)**

Dentro de las funcionalidades del Eclipse, la adición de Plugins puede facilitar la automatización de diferentes tareas, es el caso de un plugin que permite la administrar el *control de versiones* de un proyecto, seguidamente abordaremos más sobre el tema.

1.5.3. Control de Versiones.

Es el proceso de gestión de las versiones (revisiones) de todos los elementos de configuración que forman la línea base de un producto o una configuración del mismo. Para la automatización de este proceso se emplea básicamente SVN, herramienta libre que provee una serie funcionalidades necesarias para el control de versiones.

1.5.3.1 SVN

Subversion provee un sistema de control de versiones; ofrece muchas funcionalidades como: integración con Eclipse, JDeveloper y Netbeans, presenta la mayoría de las facilidades de CVS y manejo eficiente de ficheros binarios, entre otras. La experiencia con que cuenta el equipo de desarrollo fue otro motivo para la selección de esta herramienta debido a que este es el sistema utilizado por la entidad Procyon. **(Tigris.org)**

SVN se utiliza como servidor para el almacenamiento de los elementos de configuración, para acceder a este servidor de manera fácil se utiliza un cliente denominado SubEclipse.

1.5.3.2. SubEclipse

Esta herramienta no es más que un *plugging* que integra el Eclipse con el SVN. Haciendo uso de este software podremos facilitar el control de versiones desde el IDE Eclipse de forma tal que se pueden hacer operaciones de commit, update, delete, entre otras. Esta herramienta es libre y es ampliamente utilizada por los desarrolladores de software que utilizan el Eclipse como IDE a nivel mundial. **(SubEclipse, 2004)**.

1.6. Conclusiones del Capítulo

Se ha hecho mención de los principales conceptos que se abordaron a lo largo de todo este trabajo, introduciendo al lector en la terminología utilizada en el campo de las aplicaciones de mensajería para móviles. Se profundizó en las soluciones alternativas que existen a este proyecto en el mercado internacional, así como, algunas de sus ventajas y las principales desventajas que influyen directamente en el problema a resolver. Se realiza un análisis de SAMS-M y su arquitectura enfocado directamente al desarrollo de drivers para esta API y se profundiza en el protocolo PDU específicamente para los servicios de SMS y MMS. Además, se abordaron las herramientas y la metodología de desarrollo para la construcción de la solución.

2

2. Características del sistema

2.1. Introducción

En este capítulo se tratarán las características generales del sistema observadas en la modelación del dominio, que fueron recogidas en el proceso de captura de requisitos funcionales y no funcionales del sistema. Como punto de partida se abordan los objetivos estratégicos de la entidad Procyon y los VASP en general a nivel internacional. Se menciona el actual flujo de procesos y se realiza un análisis crítico de estos, sentando las bases para definir el objeto de automatización y la propuesta de solución que será desarrollada.

2.2. Objetivos estratégicos

En la actualidad la entidad Procyon necesita insertarse en el mercado de las aplicaciones para móviles, dominado actualmente por diversas empresas con mucho prestigio como Altecom, Filmac y Elibom, entre otras. Los productos desarrollados estas compañías, proveedoras de servicios de valor agregado, deben ofrecer grandes facilidades de comunicación con los diferentes sistemas disponibles en este sector de las Telecomunicaciones, con el fin de lograr contratos con una amplia gama de Operadores, y por consiguiente, que el servicio agregado que estas compañías ofrecen sea consumido por un mayor número de clientes finales.

Las grandes ganancias que ofrecen estos servicios han hecho que un gran número de compañías dirijan su atención hacia este mercado, el cual, como resultado de lo antes mencionado, se ha convertido en un

campo de gran competitividad. Todo esto, ligado al constante cambio de las tecnologías, hace necesario que los VASP desarrollen aplicaciones con mayor rapidez, calidad y mejores prestaciones.

Una de las soluciones que permite a los desarrolladores de aplicaciones que ofrecen servicios de mensajería concentrarse en la lógica del negocio y abstraerse de la comunicación con los Centros de Mensajería, es utilizando una implementación de SAMS-M para acceder a los servicios de red.

Con la utilización de una implementación concreta de SAMS-M y el driver para el protocolo PDU es posible proveer servicios de mensajería a diferentes clientes sin poseer contratos especiales con proveedores de servicios. Esto es especialmente útil si la empresa se esta iniciando en estos tipos de servicios, ó si no se han logrado establecer los contratos especiales con los Operadores para la comunicación con los Centros de Mensajería, permitiendo, una penetración paulatina en el mercado, así como, un marco para probar los productos en un entorno real, con clientes reales que esperan un servicio de calidad. Al mismo tiempo, se va ganando en conocimiento y prestigio necesario para establecer contratos con Operadores de Telefonía Inalámbrica, con productos de mayor calidad y así competir con las diferentes compañías que comercializan productos similares a nivel mundial.

2.3. Flujo de los procesos involucrados en el campo de acción.

A nivel mundial se utilizan una amplia gama de productos para garantizar la transferencia de mensajes a los Centros de Mensajería a través de los distintos protocolos de comunicación establecidos. Entre los productos se pueden encontrar APIs, Gateway, Plataformas como IPX y OSA Parlay, Bulk Servers entre otros, la siguiente figura ilustra el entorno en el que se enmarcan estos productos.

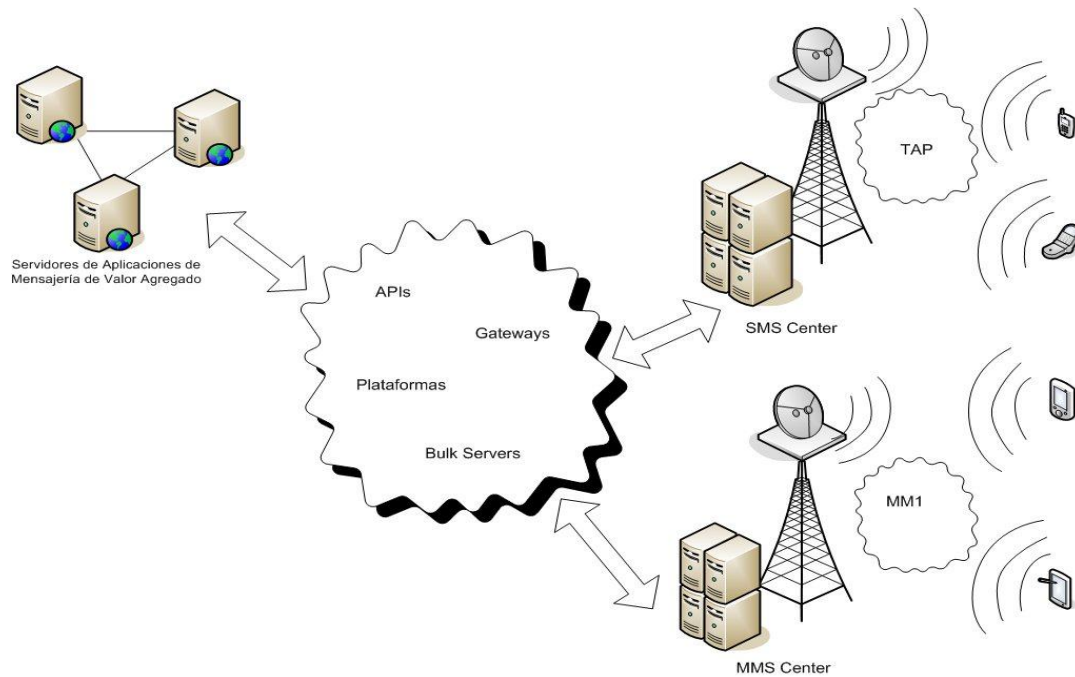


Figura 10 Entorno de aplicaciones para transferencia de mensajes a Centros de Mensajería.

El envío y recepción de mensajes SMS y MMS en la entidad Procyon se ha realizado a través, del NowSMS Gateway, este producto provee una capa de abstracción para los protocolos de comunicación con los Operadores de Telefonía Celular. El proceso de comunicación de una aplicación desarrollada por un VASP con el NowSMS se puede realizar fácilmente a través del protocolo HTTP que es bien conocido. Además, provee varias interfaces Web que facilitan el envío personalizado de mensajes. En la siguiente figura se muestra como se realiza el proceso hasta el momento.

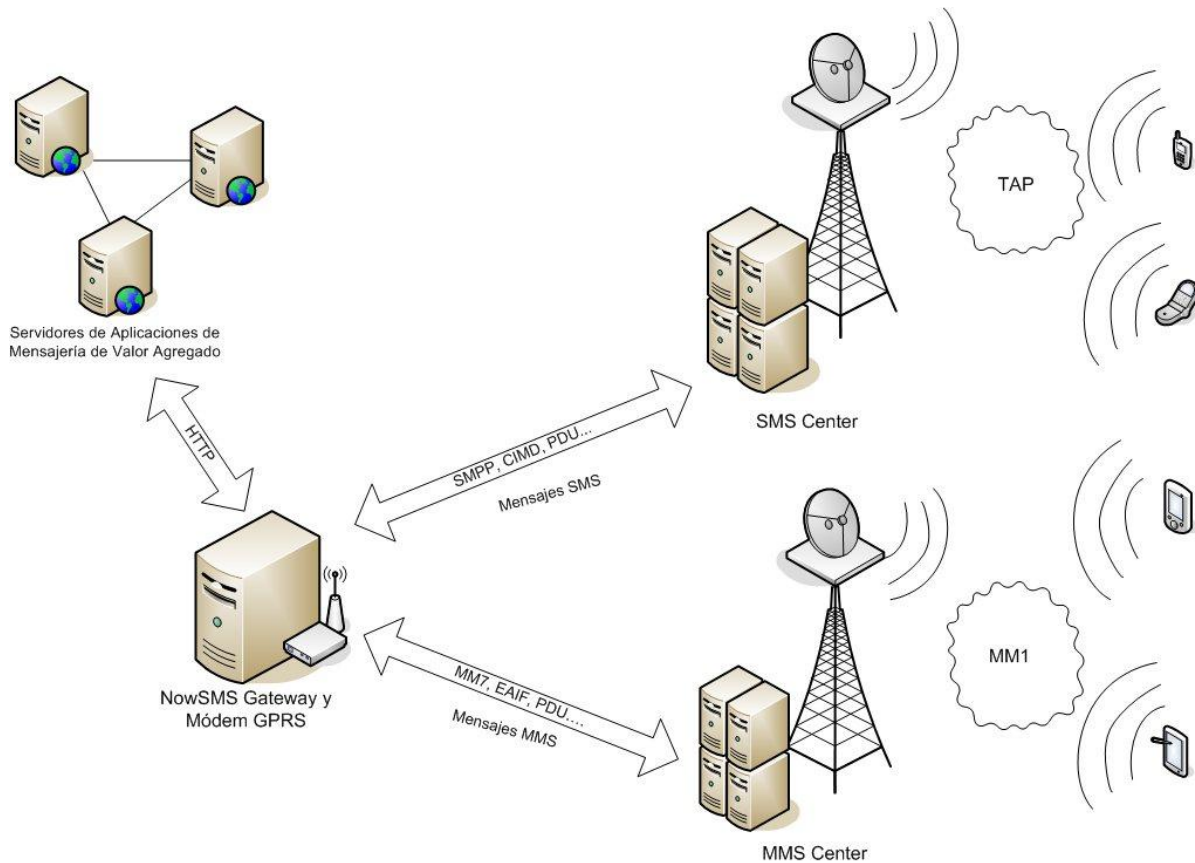


Figura 11 Entorno de la entidad Procyon para transferencia de mensajes utilizando NowSMS

Como se puede observar una aplicación desarrollada por un VASP se comunica con el NowSMS, a través, de Internet utilizando el protocolo HTTP. El NowSMS se encarga de la comunicación con los SMSC y MMSC para el envío y recepción de los mensajes SMS, Wap Push y MMS respectivamente mediante distintos protocolos como SMPP, CIMD, MM7, según los contratos establecidos con los Operadores de Telefonía Inalámbrica, en caso de no haberse establecido ningún contrato, la única alternativa es utilizar un módem GPRS y el protocolo PDU. Los SMSC y MMSC se encargan de la comunicación con los dispositivos móviles como teléfonos celulares, Smartphone, entre otros.

Luego de haberse analizado como es el flujo de procesos a nivel mundial y en la entidad Procyon se puede llegar a distintas conclusiones que serán analizadas con mayor profundidad en el siguiente epígrafe, donde se realiza un análisis crítico del entorno y las soluciones existentes en el mercado.

2.4. Análisis crítico

Varios productos como APIs, plataformas, entre otros, se han desarrollado internacionalmente. Estos productos proveen sus propias interfaces de comunicación, la gran mayoría son propietarios y se deben pagar licencias para su uso, están desarrollados en diferentes lenguajes de programación y no implementan la mayoría de los protocolos que existen en el campo de la mensajería para teléfonos celulares, además, poseen limitantes en cuanto a Sistemas Operativos sobre los que se ejecutan, por lo cual no se ajustan a las necesidades que tienen los VASP de estandarizar las interfaces para SMS, MMS y WAP Push.

En la entidad Procyon la utilización del NowSMS ha posibilitado el envío y recepción de mensajes SMS pero este producto tiene muchas limitantes que nos han conllevado al desarrollo de una solución propia adaptándose a un estándar bien establecido. Estas limitantes son las siguientes:

- NowSMS es una solución de código propietario.
- La licencia de NowSMS es costosa.
- Para la recepción de MMS es necesario fabricar un software que gestione todos los archivos que se reciben en una carpeta y sea capaz de enviarlos a través de internet, debido a que el NowSMS se limita a colocar los archivos en una carpeta local.
- No es posible tener conocimiento de cuantos mensajes de los enviados a través del NowSMS fueron recibidos correctamente por su destinatario.
- No es posible eliminar, cancelar o reemplazar mensajes.
- NowSMS solo puede utilizarse sobre Windows NT/2000/XP.

Existen otros motivos que se tuvieron en cuenta al desechar la posibilidad de utilizar NowSMS, por ejemplo, no tiene soporte para plataformas como IPX, OSA/Parlay, entre otras y no es posible adicionarle ese soporte, no implementa todos los protocolos existentes y es una solución propietaria desarrollada en C que se ejecuta solamente sobre el sistema operativo Windows.

2.5. Objeto de automatización

En la entidad Procyon se desea automatizar el proceso de envío y recepción de SMS, MMS y Wap Push, a través de la utilización de una interfaz genérica, especificada por un estándar bien establecido en el mundo de los servicios de mensajería para móvil, como es JSR-212 SAMS-M. El protocolo que se debe implementar es PDU, para permitir la transferencia de mensajes SMS, MMS y WAP Push a los Centros de Mensajería Inalámbrica, ajustándose a las funcionalidades que brinda SAMS-M como son envío, recepción, cancelación, reemplazo y obtención del estado de los mensajes. De estas funcionalidades se implementarán aquellas que sean permitidas por el protocolo PDU.

2.6. Propuesta de Sistema

Se propone realizar un driver para el API SAMS-M implementado en la entidad Procyon. Se ha decidido implementar el protocolo PDU (Protocol Data Unit) específicamente para SMS, y el diseño de la solución para el servicio de MMS.

El driver será diseñado utilizando un patrón de arquitectura de dos capas, identificando en la capa superior la lógica de implementación del protocolo PDU, y la capa inferior responsable de la interacción con el hardware. Esto posibilita la escalabilidad de la aplicación y facilita el mantenimiento.

Se ha identificado el protocolo PDU como objeto de desarrollo debido a que no se necesitan contratos especiales para realizar la comunicación. Utilizando un teléfono celular compatible con GSM/GPRS haciendo función de módem, ó un módem GPRS, conectado a la PC y una suscripción estándar a un Operador de Telefonía Inalámbrica, pueden enviarse y recibirse mensajes sobre la red GSM y GPRS. Para lograr un mejor entendimiento se propone la siguiente figura donde se ilustra el entorno de SAMS-M y el driver que será desarrollado.



Figura 12 Entorno aplicación de SAMS-M con un driver para el protocolo PDU

En la figura se ilustra un servidor de Aplicaciones de Mensajería capaz de generar SMS y MMS en un entorno J2EE, que permite que SAMS-M esté desplegado en un servidor independiente y este conectado por cable a través de un puerto de serie a un módem GPRS este será capaz de recibir y enviar los mensajes a través de la red inalámbrica hacia los Centros de Mensajería encargados de entregarlos a sus destinatarios.

2.7. Modelo del Dominio

El flujo de trabajo de modelación del negocio es un paso importante en la realización de un producto de software, ya que favorece en gran medida el entendimiento de lo que se desea automatizar ayudando a la producción de un sistema que responda a las necesidades de los posibles usuarios finales. Esta actividad esta dividida en modelo de negocio y en modelo de objeto los que se componen por los casos de uso, entidades, actores y trabajadores involucrados. En el sistema que se desea realizar no se identifican de forma clara los actores, ni los trabajadores, ni los procesos del negocio por lo cual se decide realizar un modelo de dominio el cual es un subconjunto del modelo de objeto del negocio donde se representan los conceptos y eventos fundamentales que se expresan como clases con la cardinalidad que existe entre ellas.

2.7.1. Diagrama de Clases del Dominio

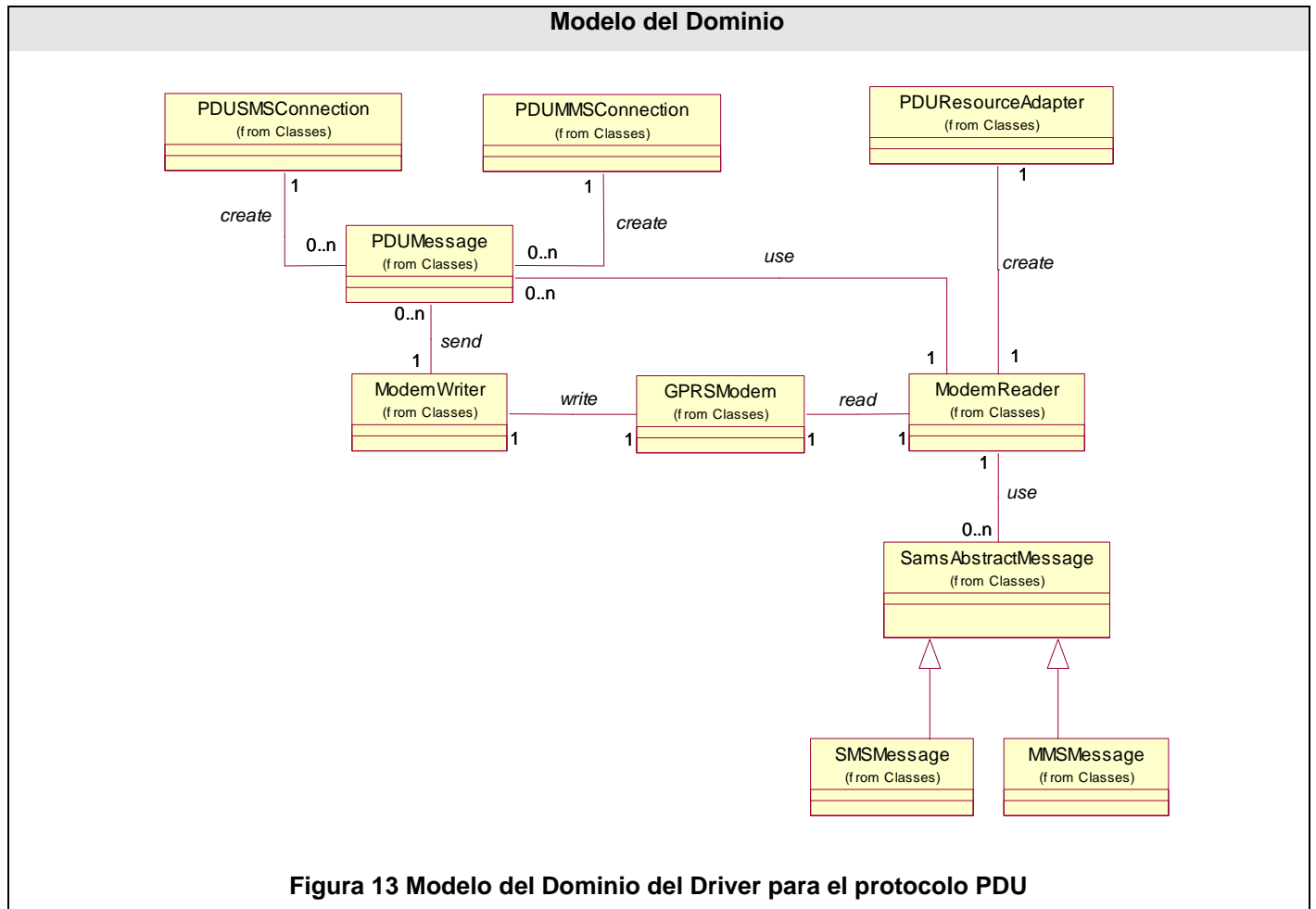


Figura 13 Modelo del Dominio del Driver para el protocolo PDU

A continuación se muestra una descripción de los conceptos representados en el diagrama del Modelo del Dominio, con el objetivo de comprender cada una de las relaciones que se establecen entre estas entidades.

2.7.2. Descripción de los conceptos del Dominio

PDUSMSConnection: Responsable de manejar las operaciones de envío, cancelación y solicitud de estado para los mensajes SMS.

PDUMMSConnection: Responsable de manejar las operaciones de envío y cancelación para los mensajes MMS.

PDUResourceAdapter: Responsable de manejar la recepción de los mensajes SMS y MMS.

PDUMessage: Representa un mensaje SMS ó MMS en formato PDU para ser enviado por el módem

ModemWriter: Responsable de escribir flujos de bytes en el módem.

ModemReader: Responsable de leer flujos de bytes del módem.

GPRSModem: Representa un módem GPRS que efectúa la comunicación con los Centros de Mensajería.

SamsAbstractMessage: Representa un mensaje genérico del API SAMS-M.

SMSMessage: Representa un mensaje SMS del API SAMS-M.

MMSMessage: Representa un mensaje MMS del API SAMS-M.

2.8. Requisitos

Como resultado del proceso de Captura de Requisitos para el proyecto SAMS-GSM, que elabora un driver para el protocolo PDU, se exponen los siguientes requisitos que debe cumplir la aplicación a desarrollar. La identificación de estos requisitos, se ha realizado a partir de las necesidades reales de la entidad Procyon y los posibles usuarios de este driver.

2.8.1. Requisitos Funcionales

RF1.- Enviar mensaje SMS.

Consiste en enviar un SMSMessage a través del módem GPRS a su destinatario.

RF1.1.- Transformar objeto.

Transformar un objeto *SMSMessage* de SAMS-M en un PDU de tipo *SMS-Submit*.

RF1.2.- Segmentar mensaje

Dividir en varios segmentos con el mismo identificador (*Message_Id*) en caso que el mensaje sea mayor a 140 caracteres.

RF1.3.- Enviar segmentos.

Enviar los segmentos de mensaje a través de módem GPRS, escribiendo del PDU en el módem.

RF1.4.- Recibir reporte.

Recibir un PDU del tipo *SMS-Submit-Report* a través del módem enviado desde el SMSC.

RF2.- Recibir mensaje SMS.

Consiste en conformar un *SMSMessage* a partir de los datos recibidos por el módem y entregarlo a través de las interfaces de SAMS-M.

RF2.1.- Procesar mensajes.

Procesar los diferentes PDU de tipo *SMS-Deliver* que se reciben a través del módem GPRS.

RF2.2.- Concatenar mensajes

Concatenar los mensajes divididos en múltiples segmentos según el (*Message_Id*) y el orden del mensaje.

RF2.3.- Entregar mensaje

Consiste en crear un *SMSMessage* y enviarlos al destino (*end-point*) que activa SAMS-M.

RF2.4.- Enviar reporte de entrega.

Conformar un PDU del tipo *SMS-Deliver-Report* y enviarlo a través del módem GPRS para confirmar la entrega.

RF3.- Enviar Mensaje MMS.

Consiste en enviar un *MMSMessage* a través del módem GPRS a su destinatario.

RF3.1.- Transformar objeto.

Transformar un objeto *MMSMessage* de SAMS-M en el PDU correspondiente al tipo *M-Send.req*.

RF3.2.- Enviar mensaje.

Enviar el PDU a través del módem en un *WSP/HTTP Post*.

RF3.3.- Recibir confirmación.

Recibir una confirmación del MMSC con un PDU del tipo *M-Send.conf*.

RF3.4.- Confirmar entrega.

Confirmar la entrega del mensaje a través del envío de un PDU de tipo *M-Delivery.Ind*.

RF4.- Recibir Mensaje MMS.

Consiste en conformar un *MMSMessage* a partir de los datos recibidos por el módem y entregarlo a través de las interfaces de SAMS-M.

RF4.1.- Recibir notificación.

Recibir un WAP Push y procesarlo para obtener la URL donde se encuentra el mensaje.

RF4.2.- Enviar confirmación de la notificación.

Confirmar la notificación enviando un PDU del tipo *M-NotifyResp.ind* a través del módem.

RF4.3.- Recibir mensaje.

Enviar un *WSP/HTTP Get* a la URL recibida y obtener el mensaje desde el MMSC.

RF4.4.- Recibir confirmación de entrega.

Procesar PDU del tipo del tipo *M-Retrieve.conf* enviado desde el SMSC con reporte del estado de la entrega.

RF4.5.- Entregar mensaje.

Conformar un *MMSMessage* y enviarlo al destino (end-point) especificado por SAMS-M.

RF5.- Consultar estado del mensaje SMS.

Consultar el estado de un mensaje SMS que está almacenado en el SMSC.

RF5.1.- Conformar comando.

Conformar el PDU correspondiente al tipo *SMS-Command* con la acción status utilizando el identificador del mensaje (*Message_Id*) que provee SAMS-M.

RF5.2.- Enviar comando.

Enviar el PDU a través del módem GPRS.

RF5.3.- Recibir reporte

Recibir un PDU del tipo *SMS-Status-Report* a través del módem enviado desde el SMSC.

RF6.- Cancelar mensaje SMS.

Proceso de cancelar un mensaje SMS cuando ha expirado o cuando se estime conveniente.

RF6.1.- Conformar comando.

Conformar el PDU correspondiente al tipo *SMS-Command*, con la acción cancel utilizando el identificador del mensaje (*Message_Id*) que provee SAMS-M.

RF6.2.- Enviar comando.

Enviar el PDU a través del módem GPRS.

RF7.- Cancelar Mensaje MMS.

Proceso de cancelar un mensaje MMS cuando ha expirado o cuando se estime conveniente.

RF7.1.- Conformar comando.

Conformar el PDU en correspondencia al tipo de mensaje *M-Cancel.req* utilizando el identificador del mensaje (*Message_Id*) que provee SAMS-M.

RF7.2.- Enviar comando.

Enviar el PDU a través del módem GPRS.

RF7.3.- Recibir confirmación

Recibir confirmación del MMSC con un PDU del tipo *M-Cancel.conf*, a través del módem.

RF8.- Administrar módem GPRS.

Proceso de administrar la inicialización y el cierre de las comunicaciones con el módem, así como los datos de entrada y salida en el puerto de comunicaciones.

RF8.1.- Inicializar y cerrar el puerto de comunicaciones.

Se inicializa el puerto de comunicaciones con los parámetros necesarios, incluyendo la velocidad en baudios (*baud-rate*), la paridad (*parity*) entre otros.

RF8.2.- Inicializar y terminar conexión GPRS.

Comprende el proceso de inicializar una conexión GPRS con un MMSC y terminar la conexión para liberar el módem

RF8.3.- Administrar entrada/salida.

Administrar la escritura y lectura de datos a través del puerto en el módem GPRS.

2.8.2. Requisitos No Funcionales

Los requisitos no funcionales especificados a continuación representan las cualidades o propiedades que debe cumplir el driver, estos se adecuan a las necesidades y las restricciones impuestas por el cliente y la entidad Procyon.

RNF 1.- Usabilidad.

RNF 1.1.- Proveer la personalización del driver y la configuración mediante un fichero de propiedades con parámetros bien definidos.

RNF 2.- Confiabilidad.

RNF 2.1.- Posibilitar la recuperación de posibles errores en la inicialización del módem, cierre de la conexión, puerto de comunicaciones ocupado, fallos en la conexión GPRS.

RNF 2.2.- Garantizar un tratamiento adecuado de las excepciones.

RNF 3.- Rendimiento.

RNF 3.1.- Proporcionar las mayores tasas de transferencia de mensajes posible medidas en mensajes/minuto. Los aspectos que deben ser tomados en cuenta para hacer una aproximación de la cantidad de mensajes que puede enviar y recibir el driver según las operaciones necesarias de cada tipo de mensaje, se enumeran a continuación:

- Velocidad del módem en baudios.
- Velocidad del puerto de comunicaciones en baudios.
- Tiempo de respuesta del módem en milisegundos.
- Latencia de la red GSM/GPRS en milisegundos.

Estos parámetros determinan el rendimiento del driver ya que el proceso de generar los mensajes es mucho más rápido que su transmisión, se debe resaltar que estos parámetros pueden variar en el tiempo, incluso no representan todas las variables que pueden influir en la transferencia de mensajes por lo tanto solo sirven para realizar aproximaciones al rendimiento que debe tener el driver.

RNF 4.- Soporte.

RNF 4.1.- Seguir los Estándares de Código de la entidad Procyon (*Anexo 5.*)

RNF 4.2.- Utilizar la estructura de paquetes propuesta por la empresa Sony Ericsson, de esta forma se facilita el mantenimiento y posibles actualizaciones al driver.

RNF 5.- Restricciones de diseño.

Para garantizar la calidad del producto y la interoperabilidad se han tomado en cuenta los siguientes estándares:

RNF 5.1.- Implementar las interfaces necesarias de la Capa de Recursos del estándar JSR-212 SAMS-M.

RNF 5.2.- Implementar el sistema utilizando Java específicamente la plataforma J2SE.

Según la especificación de SAMS-M, no existen restricciones para el diseño del driver excepto que se adapte a la arquitectura JCA de SAMS-M.

RNF 6.- Software

Para la utilización del driver se requieren los siguientes componentes de software, en los que se han tenido en cuenta las versiones inferiores con los que puede funcionar el driver.

RNF 6.1.- Utilizar una implementación concreta de SAMS-M.

RNF 6.2.- Utilizar Java Comm API versión 2.0.

RNF 6.3.- Utilizar la Máquina Virtual de Java versión 1.5 o superior.

RNF 7.- Hardware

Los requerimientos de hardware no están marcados específicamente por el consumo de recursos del driver, sino por la plataforma y el sistema operativo requerido. Para el driver específicamente se necesita el siguiente:

RNF 7.1.- Usar módem GPRS/GSM compatible con el estándar GSM 07.07 (**3GPP Specification TS 07.05, 2007**) y GSM 07.05 (**3GPP Specification TS 07.05, 2007**) respectivamente.

RNF 7.2.- Emplear los requerimientos de hardware de la Máquina Virtual de Java JVM (*del inglés: Java Virtual Machine*). (**SUN Microsystems**)

RNF 8.- Requerimiento de ayuda y documentación.

RNF 8.1.- Confeccionar un manual de instalación

RNF 8.2.- Realizar una descripción de las clases del driver.

RNF 8.3.- Descripción de los parámetros de configuración del driver.

RNF 9.- Portabilidad.

El producto ha sido diseñado para ser usado bajo el Sistema Operativo Windows, puede ser extendido a distintas distribuciones de Linux y Solaris:

RNF 9.1.- Utilizar Java Comm API con una implementación específica para cada sistema operativo.

RNF 10.- Interfaz interna.

RNF 11.1.- Interactuar con los siguientes componentes de software:

- Implementación concreta de SAMS-M.
- Java Comm API.

RNF 11.- Legales

RNF 14.1.- Emplear componentes de software y herramientas libres, para el desarrollo de la solución.

A manera de resumen podemos expresar que los requisitos especificados representan la base para el Modelo de Casos de Uso del sistema. En el próximo epígrafe se enumeran los actores del sistema y una descripción de los Casos de Uso identificados.

2.9. Modelo de Casos de Uso del sistema.

El modelo de Casos de Uso del sistema representa las funcionalidades deseadas y el entorno del sistema a través de actores y casos de uso, además sirve como un contrato entre los clientes y los desarrolladores. Este sienta las bases necesarias para el desarrollo del análisis y el diseño del sistema.

A continuación se muestra una descripción para los actores que interactúan con el sistema que debe ser desarrollado.

2.9.1. Definición de los actores del sistema.

Un actor puede representar personas o cosas externas al sistema que de alguna forma interactúan con el. En la siguiente tabla se describen los actores del sistema que debe ser desarrollado.

Tabla 1 Justificación de los actores del sistema

Actores	Justificación
SAMS	Representa el API SAMS-M responsable de iniciar los eventos de envío, recepción, estado y cancelar un mensaje.

Luego de haberse realizado la descripción de los actores damos paso al proceso de identificación de los Casos de Uso apoyados en los requisitos del sistema.

2.9.2. Casos de Uso del sistema.

Los casos de uso son artefactos que describen, en forma de acciones y reacciones, el comportamiento del sistema desde el punto de vista del usuario, las siguientes tablas presentan los casos de uso determinados para satisfacer los requisitos funcionales de sistema.

Tabla 2 Caso de Uso del Sistema. Enviar mensaje SMS

Código:	CU-1	Nombre del Caso de Uso:	Enviar mensaje SMS
Actor:	SAMS		
Descripción:	Consiste en enviar un <i>SMSMessage</i> de SAMS-M a su destinatario, en este proceso se convierte el <i>SMSMessage</i> en un PDU y se transmite a través del módem GPRS.		
Referencia:	R1		

Tabla 3 Caso de Uso del Sistema. Recibir mensaje SMS

Código:	CU-2	Nombre del Caso de Uso:	Recibir mensaje SMS.
Actor:	SAMS		
Descripción:	Consiste en conformar un objeto <i>SMSMessage</i> de SAMS-M a partir de los mensajes PDU recibidos por el módem y entregarlo a través de las interfaces de SAMS-M.		
Referencia:	R2		

Tabla 4 Caso de Uso del Sistema. Enviar mensaje MMS

Código:	CU-3	Nombre del Caso de Uso:	Enviar Mensaje MMS
Actor:	SAMS		
Descripción:	Consiste en enviar un <i>MMSMessage</i> de SAMS-M a su destinatario, en el proceso se encapsula el MMS en un mensaje PDU y se transmite a través del módem GPRS.		
Referencia:	R3		

Tabla 5 Caso de Uso del Sistema. Recibir mensaje MMS

Código:	CU-4	Nombre del Caso de Uso:	Recibir Mensaje MMS
Actor:	SAMS		
Descripción:	Consiste en conformar un objeto <i>MMSMessage</i> de SAMS-M a partir de los mensajes PDU recibidos por el módem y entregarlo a través de las interfaces de SAMS-M.		
Referencia:	R4		

Tabla 6 Caso de Uso del Sistema. Consultar estado del SMS

Código:	CU-5	Nombre del Caso de Uso:	Consultar estado del SMS
Actor:	SAMS		
Descripción:	Consiste en consultar el estado de un mensaje SMS que está almacenado en el SMSC, a través del envío de un comando en forma de un mensaje PDU a través del módem GPRS.		
Referencia:	R5		

Tabla 7 Caso de Uso del Sistema. Cancelar mensaje SMS

Código:	CU-6	Nombre del Caso de Uso:	Cancelar mensaje SMS
Actor:	SAMS		
Descripción:	Consiste en el proceso de cancelar un mensaje SMS, a través del envío de un comando en forma de mensaje PDU a través del módem GPRS.		
Referencia:	R6		

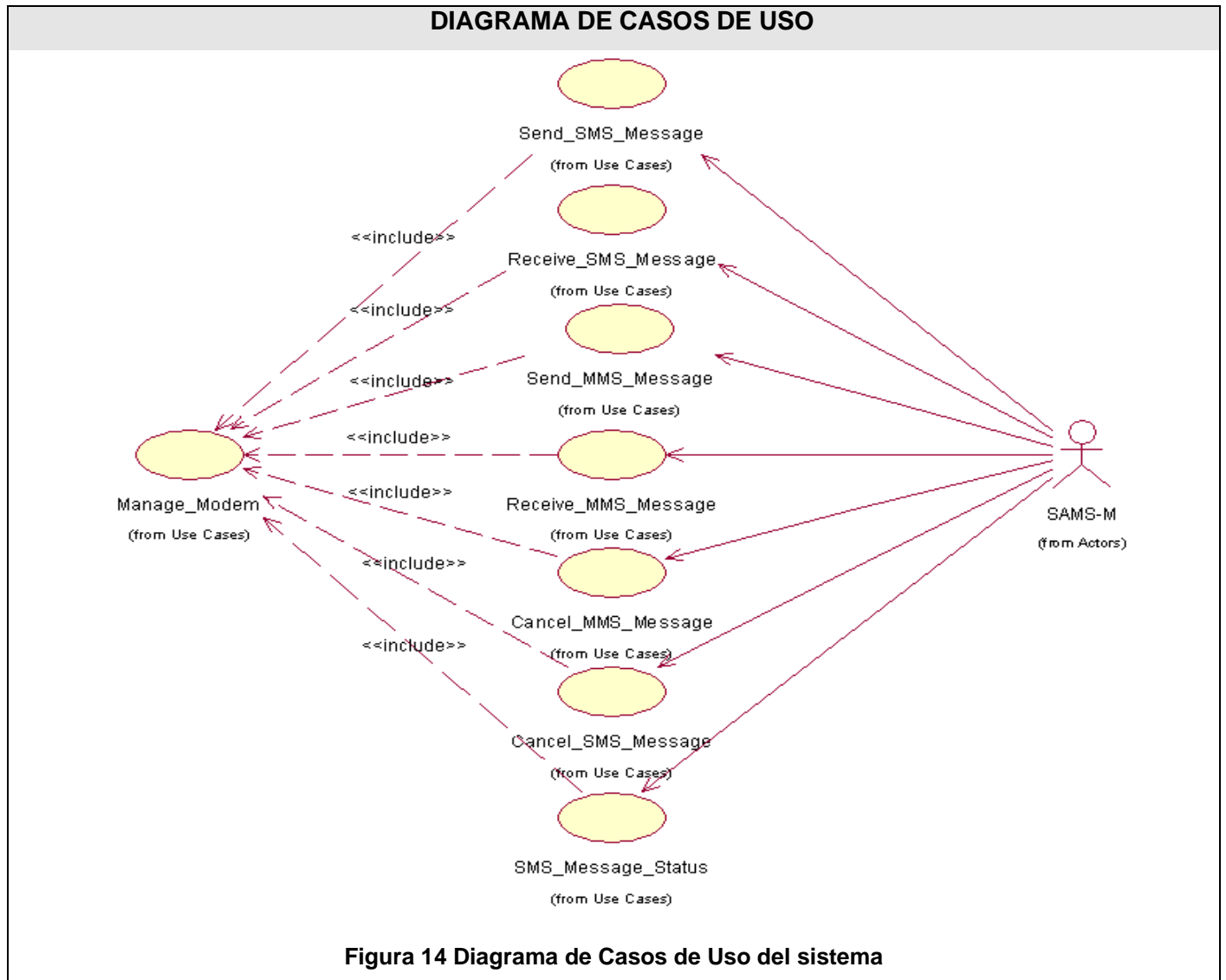
Tabla 8 Caso de Uso del Sistema. Cancelar mensaje MMS

Código:	CU-7	Nombre del Caso de Uso:	Cancelar Mensaje MMS
Actor:	SAMS		
Descripción:	Consiste en el proceso de cancelar un mensaje MMS, a través del envío de un comando en forma de mensaje PDU a través del módem GPRS.		
Referencia:	R7		

Tabla 9 Caso de Uso del Sistema. Administrar Módem.

Código:	CU-8	Nombre del Caso de Uso:	Administrar Módem
Actor:	SAMS		
Descripción:	Administra las operaciones del módem, a través del envío de comandos AT correspondientes al estándar GSM 07.05 y GSM 07.07.		
Referencia:	R8		

2.9.3. Diagrama de Casos de Uso.



2.9.4. Distribución de Casos de Uso por ciclo de desarrollo

Tabla 10 Ciclo de Desarrollo 1

Ciclo de Desarrollo 1			
Código	Nombre de caso de uso	Paquete	Justificación de la selección.
CU-8	Administrar Módem	common	Se necesita la realización de este caso de uso crítico como base para todas las operaciones que se puedan implementar en el driver para SMS. En este ciclo se desarrollaran las operaciones de acuerdo al estándar GSM 07.05.
CU-1	Enviar mensaje SMS	sms	Caso de uso crítico para el sistema debe ser desarrollado en el primer ciclo.
CU-2	Recibir mensaje SMS.	sms	Caso de uso crítico para el sistema debe ser desarrollado en el primer ciclo.

Tabla 11 Ciclo de Desarrollo 2

Ciclo de Desarrollo 2			
Código	Nombre de caso de uso	Paquete	Justificación de la selección.
CU-8	Administrar Módem	common	Se necesita la realización de este caso de uso crítico como base para todas las operaciones que se puedan implementar en el driver para el servicio de MMS. En este ciclo se desarrollaran las operaciones de acuerdo al estándar GSM 07.07.
CU-3	Enviar Mensaje MMS	mms	Caso de uso crítico para el sistema, se decide implementar en este ciclo debido a la complejidad del protocolo.

Tabla 12 Ciclo de Desarrollo 3

Ciclo de Desarrollo 3			
Código	Nombre de caso de uso	Paquete	Justificación de la selección.
CU-4	Recibir Mensaje MMS	mms	Caso de uso crítico para el sistema, se decide implementar en este ciclo debido a la complejidad del protocolo y las operaciones a realizar.
CU-5	Consultar estado del SMS	sms	Caso de uso secundario para el sistema. Se desarrolla en este ciclo para terminar las funcionalidades para el servicio de SMS.
CU-6	Cancelar mensaje SMS	sms	Caso de uso secundario para el sistema. Se desarrolla en este ciclo para terminar las funcionalidades para el servicio de SMS.
CU-7	Cancelar Mensaje MMS	mms	Caso de uso secundario para el sistema. Se desarrolla en este ciclo para terminar las funcionalidades para el servicio de MMS.

2.9.5. Casos de Uso extendidos

En este epígrafe presentamos una descripción del flujo de eventos iniciados por la acción del actor con la consecuente respuesta del sistema para cada uno de los Casos de Uso identificados.

Tabla 13 Expansion del Caso de Uso. Enviar mensaje SMS

Caso de uso	
CU-1	Enviar mensaje SMS
Propósito	Permitir el envío de SMS a través del módem GPRS.
Actores:	SAMS
Resumen:	Consiste en enviar un <i>SMSMessage</i> de SAMS-M a su destinatario, en este proceso se convierte el <i>SMSMessage</i> en un mensaje PDU y se transmite a través del módem GPRS.
Referencias	R1
Prioridad	Crítico
Precondiciones:	
Post-Condiciones:	
Acción del actor	Respuesta del sistema
1.- Enviar mensaje SMS	1.1.- Transformar el objeto <i>SMSMessage</i> en el mensaje PDU correspondiente.
	1.2.- Punto de extensión 1.
	1.3.- Enviar mensaje PDU SMS-Submit escribiéndolo en el módem.
	1.4.- Recibir mensaje PDU de confirmación SMS-Submit-Report.
	1.5.- Procesar mensaje PDU de tipo SMS-Submit-Report.
	1.6.- Completar la transferencia del SMS.
Flujo alternativo	
Acción del actor	Respuesta del sistema
Puntos de extensión.	
Punto de extensión 1.- Línea 2 Se inicializa conexión con el módem. Ver Caso de Uso 8 Administrar Módem	

Tabla 14 Expansión del Caso de Uso. Recibir mensaje SMS

Caso de uso	
CU-2	Recibir mensaje SMS.
Propósito	Recibir un mensaje SMS a través de un módem GPRS.
Actores:	SAMS
Resumen:	Consiste en conformar un objeto <i>SMSMessage</i> de SAMS-M a partir de los mensajes PDU recibidos por el módem GPRS y entregarlo a través de las interfaces de SAMS-M.
Referencias	R2
Prioridad	Crítico
Precondiciones:	
Post-Condiciones:	
Acción del actor	Respuesta del sistema
1.- Activar destino.	1.1.- Punto de extensión 1.
	1.2.- Leer mensajes PDU de tipo SMS-Deliver a través del módem.
	1.3.- Transformar mensaje PDU de tipo SMS-Deliver en un <i>SMSMessage</i> .
	1.4.- Enviar <i>SMSMessage</i> a través de la interfaz de SAMS-M.
	1.5.- Enviar mensaje PDU de tipo SMS-Deliver-Report a través del módem.
Flujo alternativo	
Acción del actor	Respuesta del sistema
Puntos de extensión.	
Línea 1 Se inicializa conexión con el módem. Ver Caso de Uso 8 Administrar Módem	

Tabla 15 Expansión del Caso de Uso. Enviar mensaje MMS

Caso de uso	
CU-3	Enviar Mensaje MMS
Propósito	Permitir el envío de MMS a través del módem GPRS.
Actores:	SAMS
Resumen:	Consiste en enviar un <i>MMSMessage</i> de SAMS-M a su destinatario, en el proceso se encapsula el MMS en un mensaje PDU y se transmite a través del módem GPRS.
Referencias	R3
Prioridad	Crítico
Precondiciones:	
Post-Condiciones:	
Acción del actor	Respuesta del sistema
1.- Enviar Mensaje MMS	1.1.- Transformar el objeto <i>MMSMessage</i> en un mensaje PDU M-Send.Req.
	1.2.- Punto de extensión 1.
	1.3.- Punto de extensión 2.
	1.4.- Enviar mensaje PDU de tipo M-Send.Req a través de la conexión GPRS.
	1.5.- Recibir mensaje PDU de tipo M-Send.conf como confirmación de envío.
	1.6.- Completar la transferencia del MMS.
Flujo alternativo	
Acción del actor	Respuesta del sistema
Puntos de extensión.	
Punto de extensión 1.- Línea 2 Se inicializa conexión con el módem. Ver Caso de Uso 8 Administrar Módem.	
Punto de extensión 2.- Línea 3 Establecer conexión GPRS. Ver Caso de Uso 8 Administrar Módem.	

Tabla 16 Expansión del Caso de Uso. Recibir mensaje MMS

Caso de uso	
CU-4	Recibir Mensaje MMS
Propósito	Recibir un mensaje MMS a través de un módem GPRS.
Actores:	SAMS
Resumen:	Consiste en conformar un objeto <i>MMSMessage</i> de SAMS-M a partir de los mensajes PDU recibidos por el módem y entregarlo a través de las interfaces de SAMS-M.
Referencias	R4
Prioridad	Crítico
Precondiciones:	
Post-Condiciones:	
Acción del actor	Respuesta del sistema
1.- Activar destino.	1.1.- Punto de extensión 1.
	1.2.- Punto de extensión 2.
	1.3.- Realizar WSP/HTTP Get a través del módem.
	1.4.- Recibir mensaje PDU de tipo M-Retrieve.conf, a través de la conexión GPRS.
	1.5.- Transformar mensaje PDU de tipo M-Retrieve.conf en un <i>MMSMessage</i> .
	1.6.- Enviar <i>MMSMessage</i> a través de la interfaz de SAMS-M.
	1.7.- Enviar mensaje PDU de tipo M-Acknowledge.ind, a través de la conexión GPRS.
Flujo alternativo	
Acción del actor	Respuesta del sistema
Puntos de extensión.	
Punto de extensión 1.- Línea 1 Se inicializa conexión con el módem. Ver CU 8 Administrar Módem.	
Punto de extensión 2.- Línea 2 Establecer conexión GPRS. Ver CU. 8 Administrar Módem.	

Tabla 17 Expansión del Caso de Uso. Consultar estado del SMS

Caso de uso	
CU-5	Consultar estado del SMS
Propósito	Consultar el estado de un mensaje MMS a través de un módem GPRS.
Actores:	SAMS
Resumen:	Consiste en consultar el estado de un mensaje SMS que está almacenado en el SMSC, a través del envío de un comando en forma de mensaje PDU a través del módem GPRS.
Referencias	R5
Prioridad	Secundario
Precondiciones:	
Post-Condiciones:	
Acción del actor	Respuesta del sistema
1.- Consultar estado del SMS	1.1.- Punto de extensión 1.
	1.2.- Generar mensaje PDU de tipo SMS-Command con el <i>Message_Id</i> .
	1.3.- Enviar mensaje PDU de tipo SMS-Command, a través del módem.
	1.4.- Recibir mensaje PDU de tipo SMS-Status-Report, a través del módem.
	1.5.- Procesar mensaje PDU de tipo SMS-Status-Report.
	1.6.- Enviar reporte de estado, a través de las interfaces de SAMS-M
Flujo alternativo	
Acción del actor	Respuesta del sistema
Puntos de extensión.	
Punto de extensión 1.- Línea 1 Se inicializa conexión con el módem. Ver Caso de Uso 8 Administrar Módem.	

Tabla 18 Expansión del Caso de Uso. Cancelar SMS

Caso de uso	
CU-6	Cancelar mensaje SMS
Propósito	Cancelar un mensaje SMS, a través de un módem GPRS.
Actores:	SAMS
Resumen:	Consiste en el proceso de cancelar un mensaje SMS, a través del envío de un comando en forma de mensaje PDU a través del módem GPRS.
Referencias	R6
Prioridad	Secundario
Precondiciones:	
Post-Condicionas:	
Acción del actor	Respuesta del sistema
1.- Cancelar mensaje SMS	1.1.- Punto de extensión 1.
	1.2.- Generar mensaje PDU de tipo SMS-Command con el <i>Message_Id</i> .
	1.3.- Enviar mensaje PDU de tipo SMS-Command, a través del módem.
	1.4.- Completar la cancelación del mensaje SMS.
Flujo alternativo	
Acción del actor	Respuesta del sistema
Puntos de extensión.	
Punto de extensión 1.- Línea 1 Se inicializa conexión con el módem. Ver Caso de Uso 8 Administrar Módem.	

Tabla 19 Expansión del Caso de Uso. Cancelar mensaje MMS

Caso de uso	
CU-7	Cancelar mensaje MMS
Propósito	Cancelar un mensaje MMS, a través de un módem GPRS.
Actores:	SAMS
Resumen:	Consiste en el proceso de cancelar un mensaje MMS, a través del envío de un comando en forma de mensaje PDU a través del módem GPRS.
Referencias	R7
Prioridad	Secundario
Precondiciones:	
Post-Condicion:	
Acción del actor	Respuesta del sistema
1.- Cancelar mensaje MMS	1.1.- Genera el mensaje PDU M-Cancel.req con el <i>Message_Id</i> .
	1.2.- Punto de extensión 1.
	1.3.- Punto de extensión 2.
	1.4.- Enviar el mensaje PDU de tipo M-Cancel.req, a través de la conexión GPRS.
	1.5.- Recibir mensaje PDU de tipo M-Cancel.conf, a través de la conexión GPRS.
	1.6.- Completar la cancelación del mensaje MMS.
Flujo alternativo	
Acción del actor	Respuesta del sistema
Puntos de extensión.	
Punto de extensión 1.- Línea 2 Se inicializa conexión con el módem. Ver Caso de Uso 8 Administrar Módem.	
Punto de extensión 2.- Línea 3 Establecer conexión GPRS. Ver Caso de Uso 8 Administrar Módem.	

Tabla 20 Expansión del Caso de Uso. Administrar Módem.

Caso de uso	
CU-8	Administrar Módem
Propósito	
Actores:	SAMS
Resumen:	Administrar las operaciones del módem, a través del envío de comandos AT correspondientes al estándar GSM 07.05 y GSM 07.07.
Referencias	R8
Prioridad	Crítico
Precondiciones:	
Post-Condiciones:	
Acción del actor	Respuesta del sistema
1.- Inicializa el módem para conexión GSM ó para conexión GPRS.	1.1.- Se realiza algunas de las siguientes acciones: a) Si se decide realizar envío, recepción o consulta de estado de un SMS, ir a la sección "Conexión GSM". b) Si se decide realizar envío, recepción o consulta de estado de un MMS, ir a la sección "Conexión GPRS".
Sección "Conexión GSM"	
Acción del actor	Respuesta del sistema
2.-Establecer conexión GSM	2.1.- Leer del fichero de configuración parámetros de inicialización del módem
	2.2.- Enviar los comandos AT correspondientes, a través del puerto.
	2.4.- Leer datos del puerto enviados desde el módem.
3.- Finalizar módem.	3.1.- Enviar los comandos AT correspondientes.

Sección “Conexión GPRS”	
Acción del actor	Respuesta del sistema
2.- Establecer conexión GPRS.	2.1.- Leer del fichero de configuración parámetros de la conexión GPRS.
	2.2.- Enviar los comandos AT correspondientes, a través del puerto.
	2.3.- Recibir respuesta del módem.
	2.4.- Establecer conexión GPRS.
3.- Finalizar módem.	3.1.- Enviar los comandos AT correspondientes.
Flujo alternativo	
Acción del actor	Respuesta del sistema
Puntos de extensión.	

2.10. Conclusiones del Capítulo

En este capítulo se ha desarrollado la propuesta de solución basado en un análisis crítico de la situación a nivel internacional y en la entidad Procyon para la implementación de protocolos que permitan la comunicación con Centros de Mensajería y posibiliten la transmisión y recepción de mensajes SMS y MMS. Se obtuvieron los principales conceptos del Dominio y se registraron las funcionalidades que deben ser desarrolladas en el proceso de Captura de Requisitos realizándose una descripción de cada uno de estos. Se identificaron y describieron los Actores y Casos de uso estableciéndose las relaciones correspondientes entre cada uno de ellos en el Diagrama de Casos de Uso.

3

3. Análisis y diseño del sistema

3.1. Introducción

El capítulo tiene como objetivo principal realizar el modelo de análisis y diseño del software SAMS-GSM, efectuando los diagramas de clases y la realización de los casos de uso teniendo en cuenta los patrones de diseño, lo cual permite que el producto sea escalable, reutilizable y flexible, reduciendo así, los esfuerzos de desarrollo y mantenimiento. Este flujo de trabajo tiene una gran importancia en el ciclo de desarrollo pues traduce los requisitos definidos a funcionalidades que debe realizar el producto las cuales van a formar parte de la arquitectura candidata, el modelo de diseño que se obtiene como resultado es una entrada principal para la disciplina de implementación.

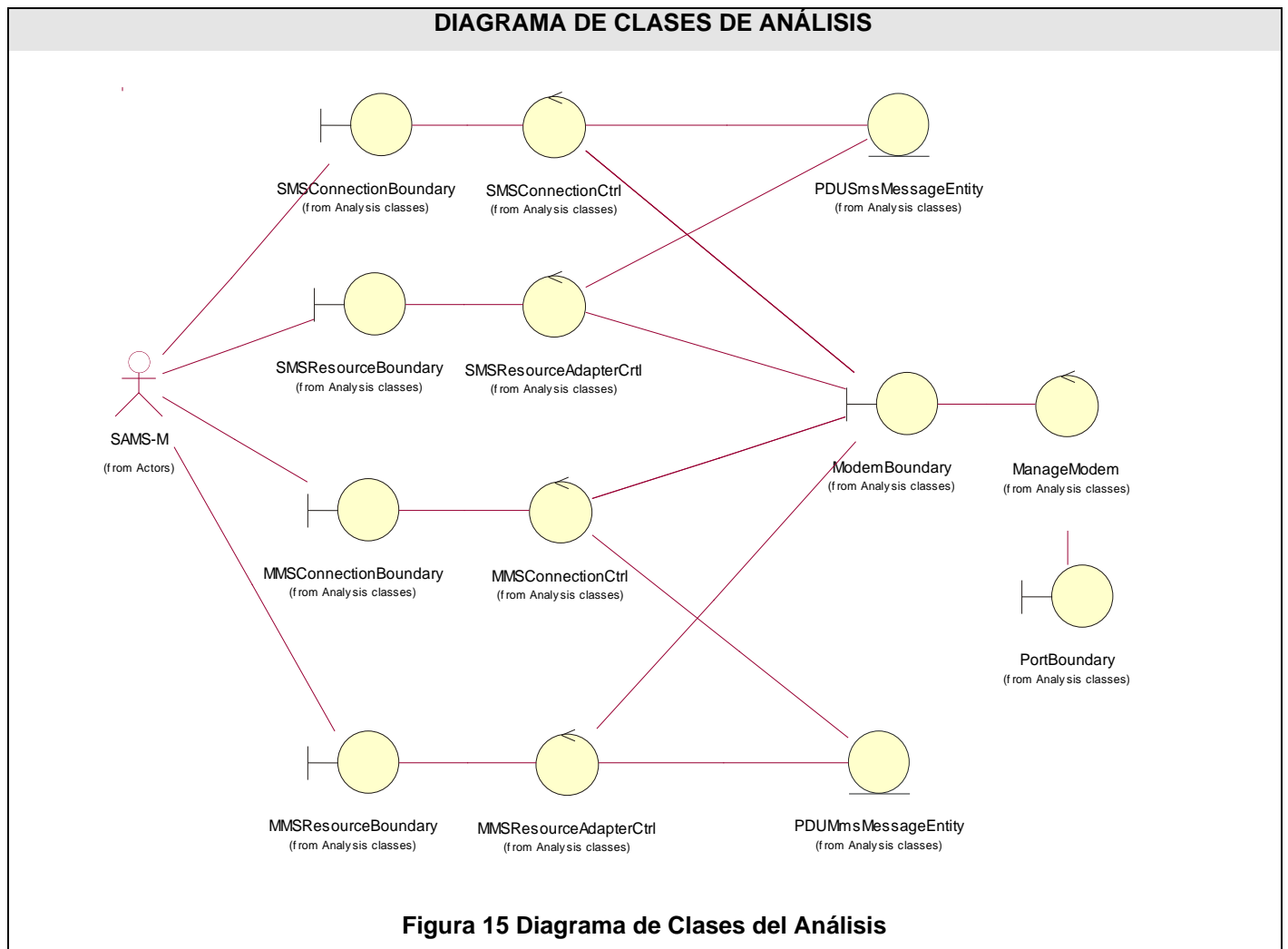
3.2. Modelo de Análisis

La disciplina de análisis facilita la realización del diseño, se basa en el refinamiento de los casos de uso del sistema y su objetivo fundamental es la asignación de las funcionalidades especificadas por los requisitos funcionales, en objetos, que proporcionan el comportamiento independientemente del lenguaje de programación que se desee utilizar. El modelo del análisis se puede considerar una abstracción del Modelo de Diseño.

Uno de los artefactos generados en el modelo del análisis es el Diagrama de Clases del Análisis. Mediante este diagrama se representa los conceptos en el dominio del problema. Las clases del análisis se centran en los requisitos funcionales representando conceptos y relaciones del dominio. Estas tienen

atributos y entre ellas se pueden establecer relaciones de asociación, agregación, composición, generalización y especialización. Según RUP estas clases se pueden clasificar en Entidad, Interfaz y Control. En la siguiente figura se muestra una vista del sistema mediante la elaboración del diagrama de clases del análisis.

3.2.1. Diagrama de clases del Análisis



Para establecer una vista dinámica de la interacción entre las clases, las relaciones y las llamadas entre cada una de las clases se realizan los diagramas de secuencia y colaboración del análisis. (*Anexo 3. Diagramas de Colaboración del Análisis.*)

3.3. Modelo de Diseño

En la fase de diseño se modela el sistema de forma tal que cubra todos los requisitos funcionales y no funcionales que fueron identificados en fases iniciales del proyecto. Se identifican los requisitos, subsistemas individuales, interfaces, clases y se toma en cuenta el lenguaje de programación que será utilizado para el desarrollo del sistema con el propósito de crear un punto de partida para las actividades de implementación subsiguientes.

En esta fase se elaboran diagramas de interacción que modelan la comunicación entre las instancias de clases para realizar las actividades especificadas en los requerimientos. Además se desarrollan diagramas de clases que representan las relaciones entre las clases y las interfaces del sistema.

Para el desarrollo de drivers existen métodos que de forma general dan soporte para la realización de estos artefactos y para la identificación subsistemas que encapsulen funcionalidades y establecer las relaciones entre estos subsistemas.

3.3.1. Métodos de diseño de drivers.

Tradicionalmente, desarrollar drivers para dispositivos implicaba libertad para generar código sin una estrategia de diseño. En el apuro de probar un nuevo modelo de hardware, las pruebas de unidad al código llego a ser la base para finalizar el driver. Afortunadamente los días de desarrollar aplicaciones sin una estrategia de diseño han terminado. Entre complejos sistemas operativos y hardware complicado no existe espacio para diseñar al mismo tiempo que se va desarrollando el código. Implementar drivers en general tiene una curva de aprendizaje pronunciada, por lo que se requiere una buena ingeniería de software y práctica en el desarrollo de este tipo de software satisfactoriamente.

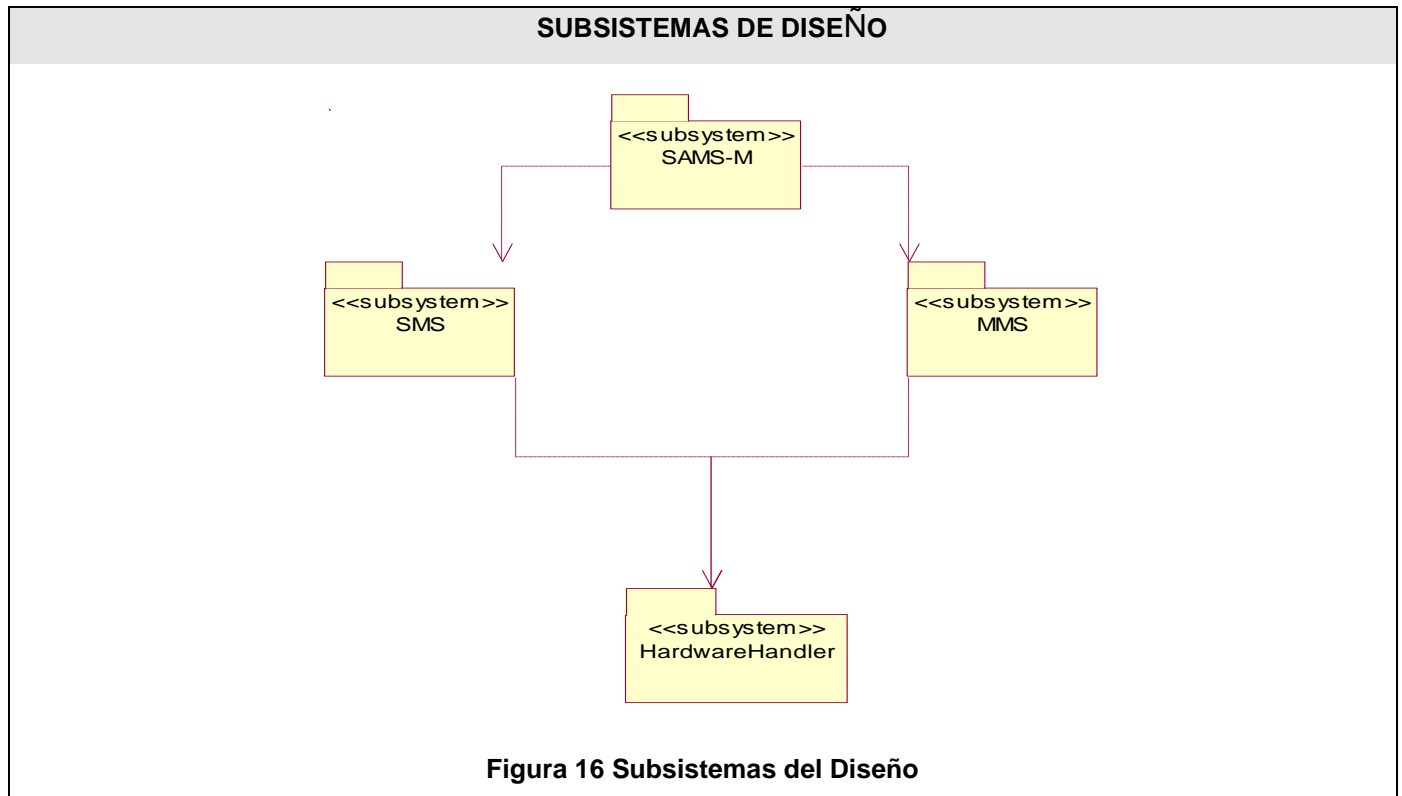
A continuación se muestra una pequeña lista de técnicas de diseño que deben ser tomadas en cuenta a la hora de desarrollar drivers, muchas de estas son tomadas de metodologías de diseño en tiempo real.

- Diagrama de flujo de datos, pueden ayudar a dividir un driver en unidades funcionales discretas. Este diagrama puede facilitar la visualización de cómo las unidades funcionales en un driver se relacionan con las otras, y como estas transforman los datos de entrada en datos de salida.
- Modelo de máquina de estado, es otra buena forma de describir el flujo de control en un driver, especialmente en uno que administre un elaborado hardware o protocolo de software. En el proceso de verificación de la máquina de estados se pueden identificar tareas potenciales de sincronización.
- Un análisis de la entrada de datos esperados o respuestas obligatorias de entrada-salida, brindará un conjunto de requerimientos de tiempo cuantitativos. Esto es importante para establecer la visión global de los objetivos generales de rendimiento del driver.
- Otra herramienta muy útil, es una lista explícita de eventos externos y las acciones que debe desarrollar el driver según esos eventos. Esta lista debe incluir eventos del hardware del dispositivo y entrada-salida de datos por software desde los usuarios.

Usando estas técnicas se facilita la descomposición del driver en unidades funcionales bien identificadas. En algunos casos, esto debe significar la ruptura de un único driver monolítico en un par de drivers que manejen las funciones dependiente del hardware e independientes del hardware. En cualquier caso, el tiempo empleado en analizar el diseño de un driver al inicio del proyecto se puede recobrar en la reducción del tiempo de depurado y mantenimiento.

Luego de realizar estas acciones se pueden identificar subsistemas del diseño que resultan de gran utilidad para el desarrollo, a su vez establecen un punto de partida para organizar aplicar el patrón Capas (*del inglés: Layers*) a la Arquitectura agrupando cada uno de estos subsistemas.

3.3.2. Subsistemas del Diseño



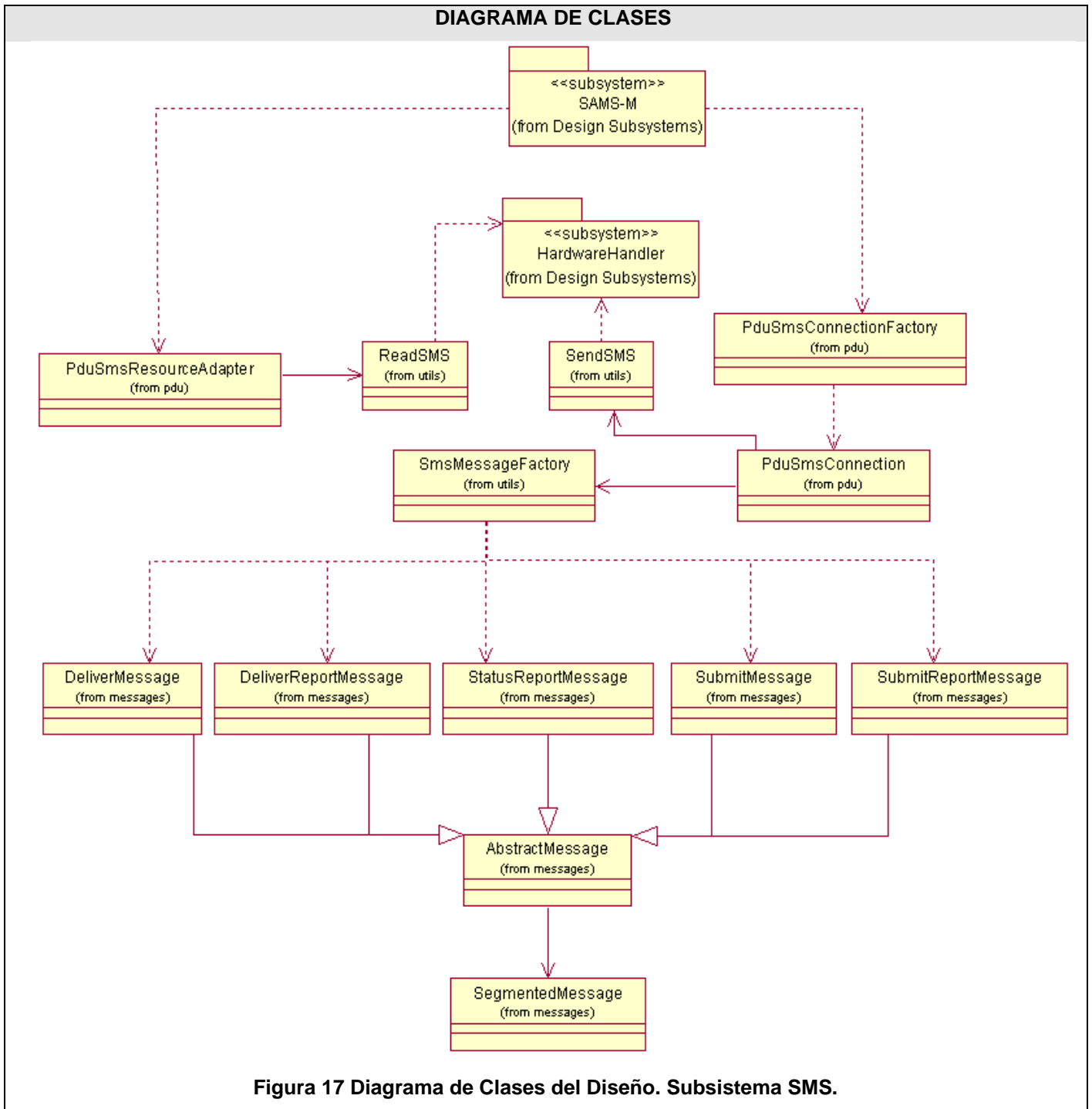
Cada uno de los subsistemas identificados engloba las funcionalidades requeridas en el driver. A continuación se muestra una vista de estos subsistemas con los diagramas de clases de cada uno de ellos, donde se representan las relaciones entre las principales clases.

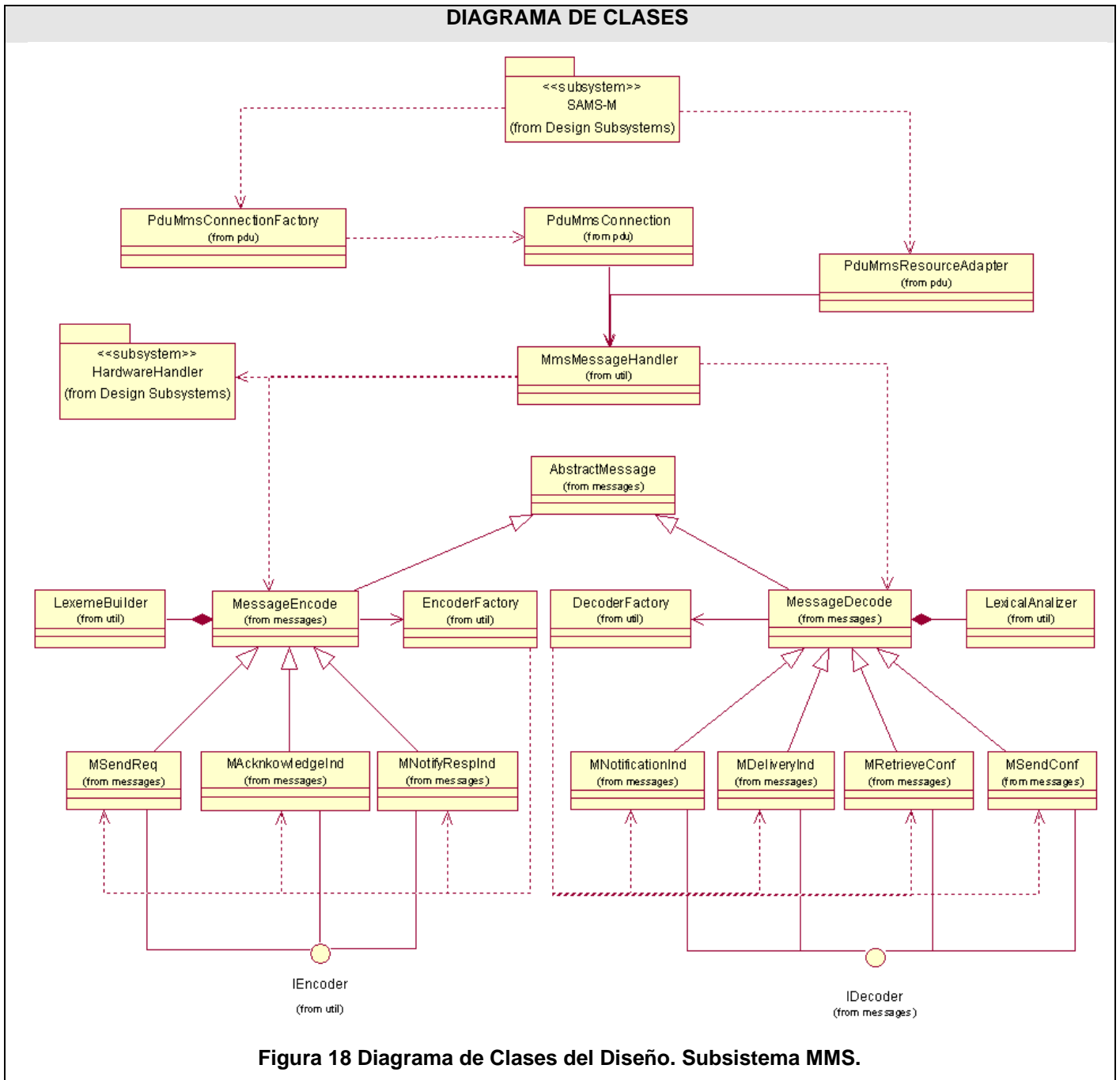
Se utiliza el Patrón de Arquitectura por Capas (*del inglés: Layers*), este estructura la aplicación de modo que puede ser descompuesta en grupos de sub tareas cada una de ellas responde a un nivel de abstracción dentro de la aplicación.

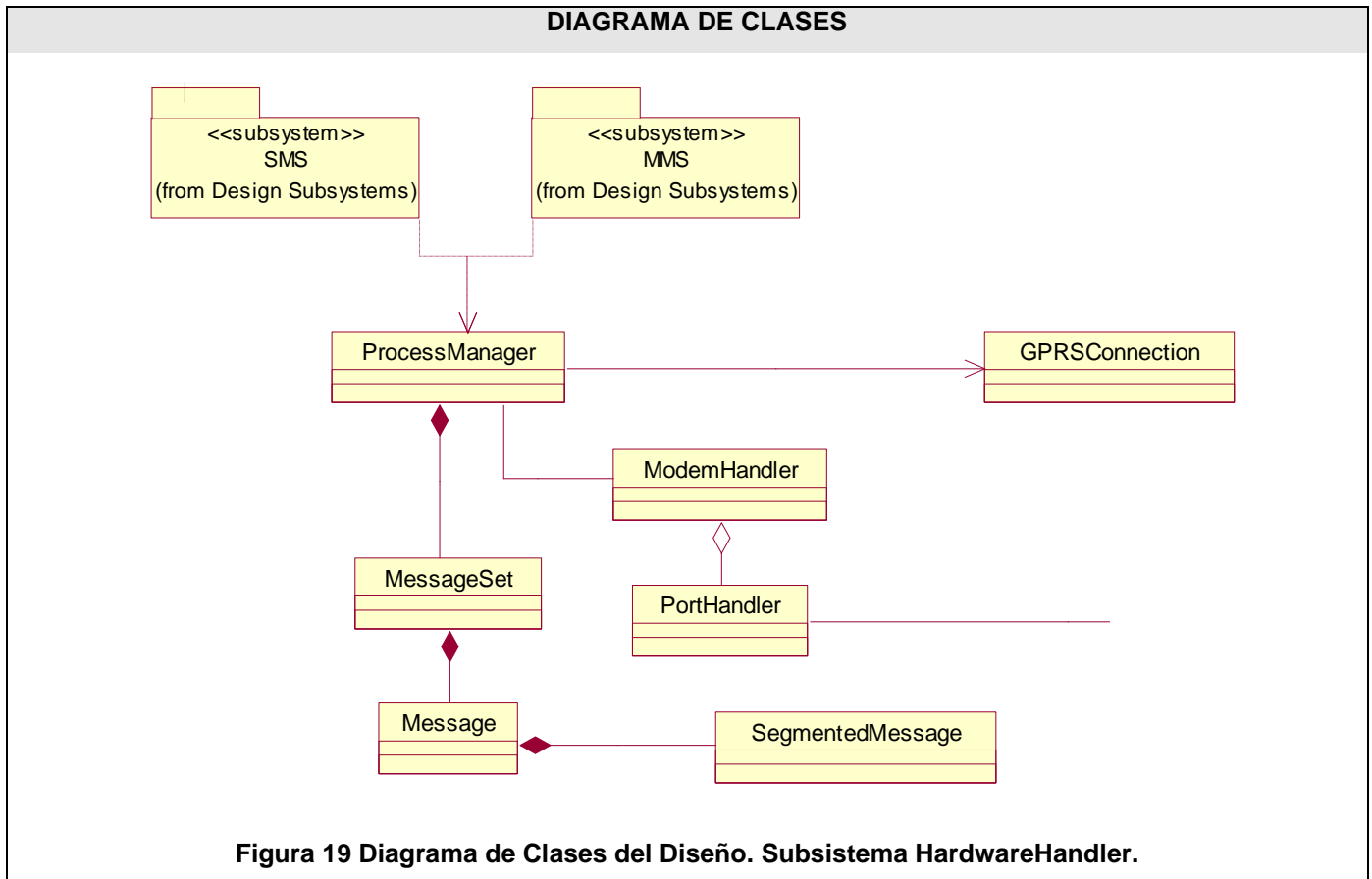
Según este patrón se agrupan los subsistemas SMS y MMS dentro de la capa “Protocol Services” y el subsistema HardwareHandler en la capa “Protocol I/O” de esta manera se obtiene un sistema de dos capas que facilita el mantenimiento y la escalabilidad de la aplicación.

En el próximo epígrafe se muestran los diagramas de clases para cada uno de los subsistemas del diseño identificados.

3.3.4. Diagramas de Clases del Diseño.







3.3.5. Diagramas de Interacción

Los diagramas de interacción modelan el comportamiento dinámico del sistema. En este trabajo se presentan los diagramas de secuencia, para cada uno de los casos de uso identificados, con el fin de mostrar una vista de la interacción entre los objetos, y sus relaciones con los mensajes que se envían entre ellos. (*Anexo 4. Diagramas de Secuencia del Diseño.*)

3.3.6. Patrones de Diseño.

Para el desarrollo de la solución se aplicaron diferentes patrones de diseño, con el objetivo de facilitar el mantenimiento del software y contribuir a la realización de un producto reutilizable y escalable.

Entre los diferentes patrones podemos mencionar los referidos a asignación de responsabilidades GRASP (*del inglés: General Responsibility Assignment Software Patterns*), estos fueron tomados en cuenta para

el diseño de clases y métodos de cada una de ellas. A continuación se mencionan algunos de los patrones utilizados en el diseño del sistema.

3.3.6.1. Patrones GRASP

- Alta cohesión. (Low coupling) Facilita el mantenimiento y la reusabilidad. Ayuda a manejar la complejidad de una clase asignándole responsabilidades (cohesivas) altamente relacionadas a una clase.
- Bajo acoplamiento. (High cohesion) Facilita el mantenimiento y la reusabilidad. Establece una manera de asignar responsabilidades a las clases para lograr, una baja dependencia entre clases, bajo impacto de los cambios que se producen en una clase para otras las demás clases que se relacionan con ella.
- Controlador. (Controller): Encargado de manejar los eventos del sistema que representan la visión general del sistema o de un caso de uso

Otro conjunto de patrones bien conocidos son los patrones de la pandilla de los cuatro GoF (*del inglés: Gang of Four*) estos se dividen en 3 grupos fundamentales: creacionales, estructurales y de comportamiento. Algunos de estos fueron aplicados directamente en el diseño de este sistema, a continuación se menciona el nombre del patrón y las clases donde se aplicaron directamente.

3.3.6.2. Patrones GoF

- Creacionales
 - Método Factoría. (Factory Method): Crea una interface para la instanciación de objetos, pero delega a las subclases la responsabilidad de decidir que clase debe ser instanciada. Este patrón se pone de manifiesto en las clases EncoderFactory, DecoderFactory y SmsMessageFactory.
 - Instancia única. (Singleton): Asegura que se obtenga una instancia única de una clase y provee un punto de acceso global a esa instancia. Este patrón se pone de manifiesto en las clases ModemHandler y PortHandler.

- Comportamiento.
 - Cadena de Responsabilidades. (Chain of responsibility): Evita el acoplamiento entre el emisor y el receptor de una petición posibilitando que más de un objeto pueda manejar la petición, envía la petición través de la cadena de objetos hasta que alguno de ellos maneje la petición. Se pone de manifiesto en las clases que heredan de MessageDecode y MessageEncode cada una de estas clases se procesa la petición según el tipo de procesamiento requerido.
- Estructurales
 - Adaptador. (Adapter): Convierte la interface de una clase en lo que los clientes de la interfaz esperan. Posibilita que clases con interfaces incompatibles puedan interactuar. Este patrón se pone de manifiesto en las clases PduMmsResourceAdapter y PduSmsResourceAdapter.

3.3.7. Lineamientos para el código fuente.

Como complemento para la realización del diseño del software con la calidad requerida por la empresa Sony Ericsson, se utilizaron estándares de códigos establecidos por la entidad Procyon y aceptados por los clientes del producto. Estos facilitan el mantenimiento del producto y la comprensión del código. (Vázquez Méndez-Quintero, et al., 2006)

3.4. Conclusiones del Capítulo

En este capítulo se obtuvo el modelo de análisis y diseño de la solución propuesta realizándose los diagramas de clases y de interacción para cada caso de uso del sistema identificado en el flujo de trabajo Requisitos. Se han enumerando algunas de las técnicas para el diseño de drivers y se realizó un bosquejo sobre los patrones de diseño aplicados, así como los estándares establecidos para la construcción del código, de forma tal, que este trabajo pueda tener un desarrollo con la calidad requerida y entendimiento para futuros desarrolladores.

4

4. Implementación y prueba del sistema.

4.1. Introducción

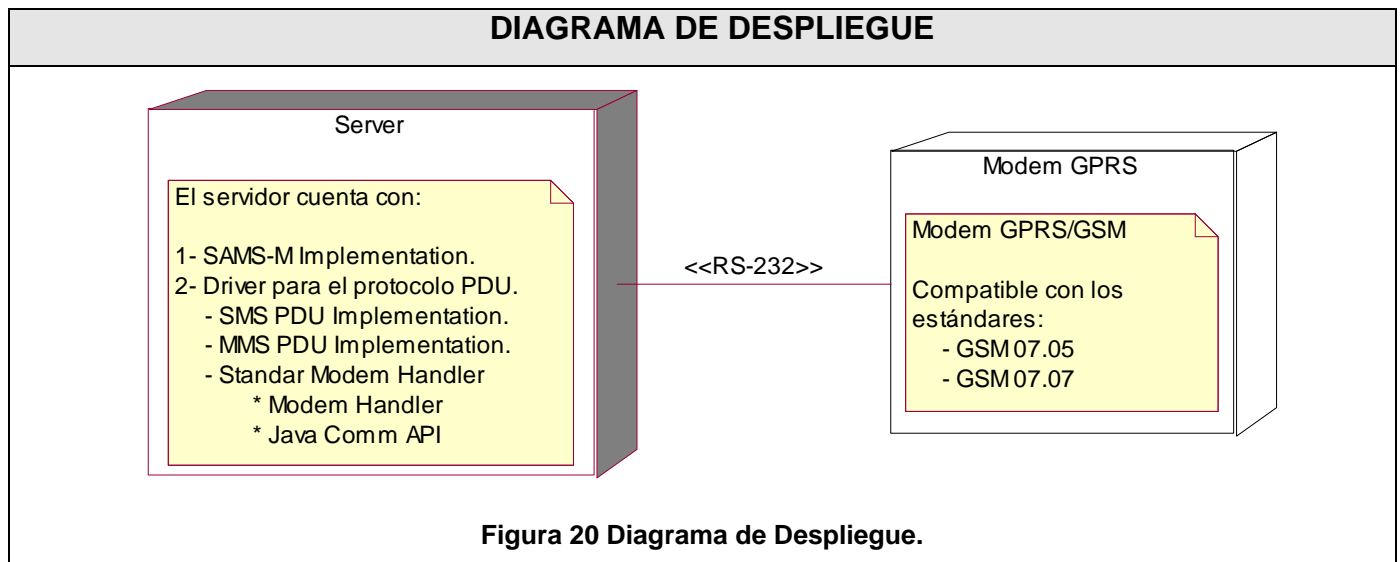
Este capítulo aborda aspectos como implementación y prueba de la solución propuesta. Se describe el modelo de implementación utilizado y se muestran los diagramas de componente y de despliegue. Se realiza, además, el modelo de prueba a través de los casos de prueba aplicados en el desarrollo de este driver.

4.2. Modelo de Implementación.

El modelo de implementación describe cómo los elementos del modelo del diseño se implementan en términos de componentes y cómo estos se organizan de acuerdo a los nodos específicos en el modelo de despliegue.

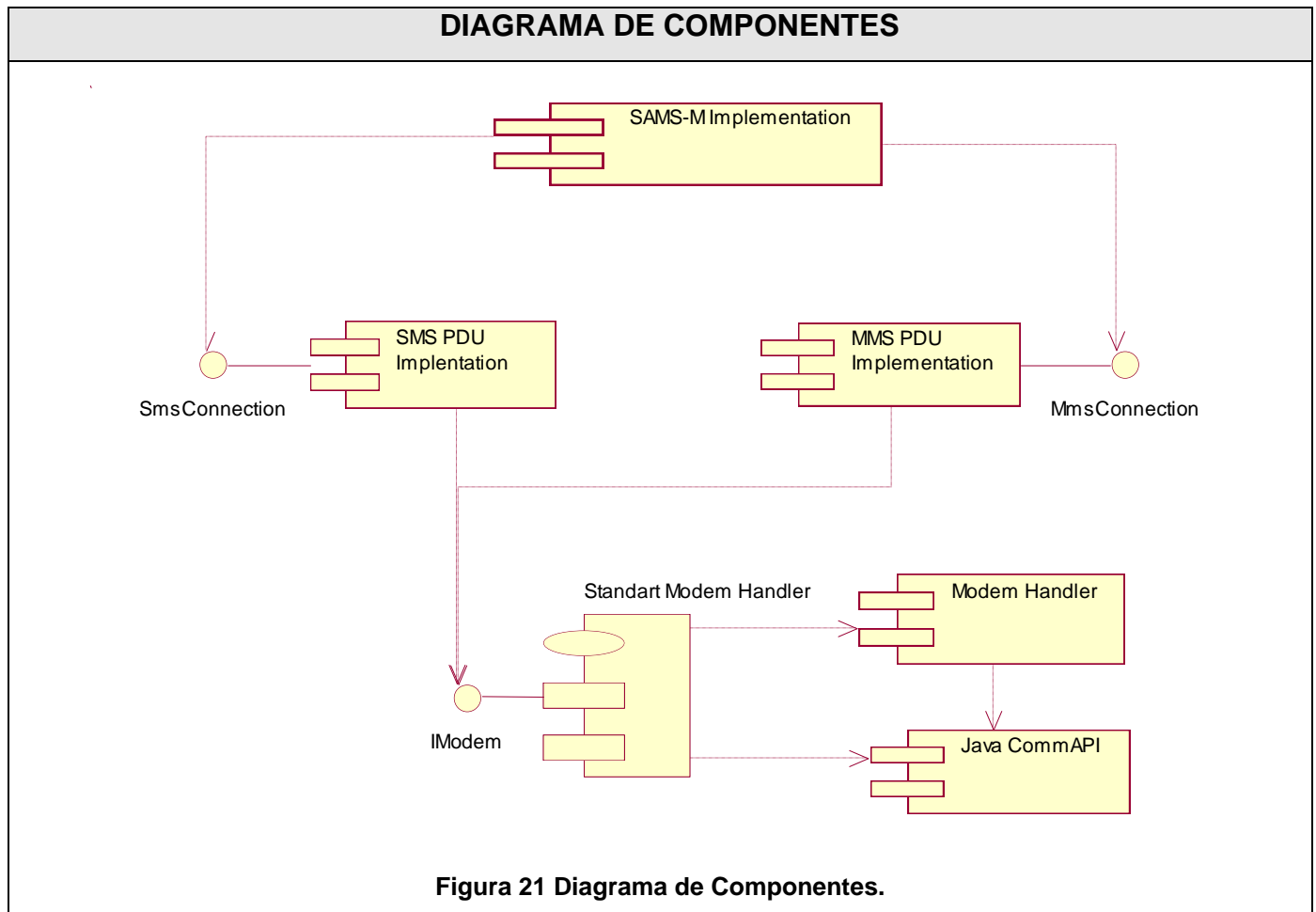
4.2.1. Diagrama de Despliegue

El diagrama de despliegue muestra la distribución de los componentes de software desarrollados en el entorno donde será aplicada la solución. La siguiente figura muestra como será desplegado el driver desarrollado en conjunto con la implementación de SAMS-M y la interacción con el módem GPRS que requiere el driver.



4.2.2. Diagrama de Componentes

El diagrama de componentes muestra las relaciones de dependencia entre las partes modulares del sistema desarrollado, que encapsulan la implementación y proporcionan las interfaces necesarias para la utilización de sus funcionalidades. A continuación se muestran los componentes identificados y las interfaces que proveen para su uso.



Un aspecto de suma importancia en el ciclo de desarrollo del software lo constituyen las pruebas ya que esta disciplina posibilita detectar los errores desde etapas tempranas, aumentando de esta forma la calidad del producto final.

4.3. Modelo de prueba

En este epígrafe se desarrolla el modelo de pruebas para el primer ciclo de desarrollo del producto. Los errores que son encontrados se documentan para realizar su corrección en el próximo ciclo, efectuando un análisis y clasificación de acuerdo a las prioridades que posean.

Ningún cuerpo de código complejo puede estar completamente libre de errores. Los programadores están familiarizados con el fenómeno de que corregir errores, introduce nuevos errores. Los errores no pueden ser detenidos; solo pueden ser contenidos. Esto es una realidad palpable cuando un software interactúa con hardware ó software de otro fabricante.

La prueba de drivers es igual en muchas formas a todas las pruebas de software: se desarrollan los casos de prueba y se miden los resultados de ejercer pruebas en condiciones de límite y estrés. A un nivel más práctico se puede decir que las pruebas a los drivers requieren innovación, conocimiento del hardware y sobre todo mucha paciencia.

4.3.1. Casos de Prueba

Para la realización del Modelo de Prueba se identificaron dos casos de prueba que son detallados a continuación.

4.3.1.1. Caso de prueba. Enviar mensaje SMS

Caso de uso: Enviar mensaje SMS.

Procedimiento:

1. Se selecciona la opción de enviar un mensaje SMS.
2. Se introduce la cantidad de destinatarios con sus números telefónicos.
3. Se introduce el texto.

Condiciones

Inicializar el módem.

Tabla 21 Caso de Prueba. Enviar Mensaje SMS

Clase válida	Clase no válida	Resultado esperado	Resultado
SAMS inicializa el evento de enviar un mensaje SMS con menos de 160 caracteres.		Se envía el mensaje	Satisfactorio
SAMS inicializa el evento de enviar un mensaje SMS con más de 160 caracteres.		Se envía el mensaje	Satisfactorio
SAMS inicializa el evento de enviar un mensaje SMS con caracteres extraños.		Se envía el mensaje correctamente	No satisfactorio <i>(Anexo 6)</i>

4.3.1.2. Análisis de los defectos

Con el análisis de los defectos se puede llevar un registro de aquellas fallas encontradas durante las pruebas. Podemos ver un ejemplo en la tabla mostrada a continuación.

Tabla 22 Análisis de Defectos. Enviar Mensaje SMS.

No.	Descripción de la No Conformidad	Fuente (Opción de la Aplicación)	Prioridad	Severidad	Estado
1	Cuando se envía el mensaje SMS con caracteres extraños se recibe con caracteres diferentes a los que fueron enviados.	Opción 1.- Enviar mensaje.	Normal	Normal	Pendiente

4.3.2.1. Caso de prueba: Recibir mensaje SMS

Caso de uso: Recibir mensaje SMS.

Procedimiento:

1. Seleccionar la opción de recibir mensajes SMS.
2. Se selecciona la opción de detener la recepción de los mensajes SMS.

Condiciones

Inicializar el módem.

Tabla 23 Caso de Prueba. Recepción de Mensaje SMS

Clase válida	Clase no válida	Resultado esperado	Resultado
SAMS inicializa el evento de recibir un mensaje SMS con menos de 160.		Se recibe el mensaje SMS.	Satisfactorio
SAMS inicializa el evento de recibir un mensaje SMS con más de 160 caracteres.		Se recibe el mensaje SMS.	Satisfactorio
Detener la recepción de mensajes SMS.		Se detiene la recepción.	Satisfactorio

Según los casos de prueba realizados, se identifica un error de severidad normal. Este no afecta en gran medida la calidad del driver, por lo que se puede valorar como satisfactorio, el trabajo realizado en el primer ciclo de desarrollo del driver atendiendo a los resultados arrojados por las pruebas específicamente.

4.4. Conclusiones del Capítulo

En este capítulo se abordaron los flujos de trabajo de implementación y pruebas propuestos por RUP. Se elaboraron los artefactos más importantes correspondientes a estos flujos como el diagrama de despliegue, el diagrama de componentes. Además se realizó el modelo de pruebas al software utilizando la técnica de caja negra permitiendo de esta forma la detección de errores desde los primeros ciclos de desarrollo del proyecto. Por último, se obtuvieron y se describieron las no conformidades, así como, su clasificación.

5

5. Estudio de Factibilidad

5.1. Introducción

Dentro de todo desarrollo de software se debe documentar y analizar sobre los beneficios que el producto reportará, uno de los factores más significativo es el costo. Gracias a este estudio se podrá obtener el tiempo de desarrollo del proyecto, así como la cantidad de personas que intervendrán y sobre todo, cuanto hay que invertir, sobre este tema se aborda en este capítulo.

5.2. Método utilizado estimación basada en Puntos de Casos de Uso.

Se trata de un método de estimación del tiempo de desarrollo de un proyecto mediante la asignación de "valores" a cierto número de factores que lo afectan, para finalmente, contabilizar el tiempo total estimado para el proyecto a partir de esos factores.

5.2.1. Cálculo de los Puntos de Casos de Uso sin ajustar.

El cálculo se realiza dada esta formula $UUCP=UAW+UUCW$, donde: $UUCP$, puntos de casos de uso sin ajustar; UAW , factor de peso de los actores sin ajustar; $UUCW$, factor de peso de los casos de uso sin ajustar.

Las siguientes tablas muestran los cálculos de los valores necesarios para la solución:

Tabla 24 Factor de peso de los actores sin ajustar.

Tipo de actor	Descripción	Factor de peso	Actores	Total
Simple	Sistema con sistema a través de interfaz de programación.	1	1	1
Medio	Sistema con sistema mediante protocolo de interfaz basada en texto.	2	0	0
Complejo	Persona que interactúa con el sistema mediante interfaz gráfica.	3		0
	UAW = $\sum(\text{Factor} * \text{Actores})$		UAW	1

UWA toma valor 1, dada las características.

Tabla 25 Factor de peso de los casos de uso sin ajustar.

Tipo de CU	Descripción	Peso	CantCU	Total
Simple	El caso de uso tiene de 1 a 3 transacciones.	5	8	40
Medio	El caso de uso tiene de 4 a 7 transacciones.	10	0	0
Complejo	El caso de uso tiene más de 8 transacciones.	15	0	0
	UUCW = $\sum(\text{Factor} * \text{CantCU})$		UUCW	40

UUCW toma valor 40, por lo tanto se realiza la suma dando le un valor de 41 a UUCP. (UAW = 1; UUCW = 40; UUCP = 41)

5.2.2. Cálculo de los Puntos de Casos de Uso ajustados.

De igual forma que en el primer paso utilizamos una fórmula ($UCP=UUCP*TCP*EF$) con sus respectivas variables, (UCP , Puntos de Casos de Uso ajustados; $UUCP$, sin ajustar valor que se obtuvo en el paso

anterior; TCP, factor de complejidad técnica; *EF*, factor de ambiente.) las cuales se calculan de las siguientes formas:

Cada factor tiene a su vez un conjunto de factores que se cuantifica en un valor que va desde 0 (irrelevante) hasta 5 (relevante).

El Factor de Complejidad Técnica (TCF) se determina mediante la cuantificación de un conjunto de factores y predeterminados que determinan la complejidad técnica del sistema.

Tabla 26 Factor de complejidad técnica

Factor	Descripción	Peso	Valor asignado	Total
T1	Sistema distribuido	2	0	0
T2	Tiempo de respuesta	1	4	4
T3	Eficiencia del usuario final	1	0	0
T4	Funcionamiento Interno complejo	1	4	4
T5	El código debe ser reutilizable	1	4	4
T6	Facilidad de instalación	0.5	3	1.5
T7	Facilidad de uso	0.5	5	2.5
T8	Portabilidad	2	5	10
T9	Facilidad de cambio	1	3	3
T10	Concurrencia	1	0	0
T11	Incluye objetivos especiales de seguridad	1	0	0
T12	Provee acceso directo a terceras partes	1	0	0
T13	Se requieren facilidades especiales de entrenamiento de usuarios	1	0	0
			Σ	29
	TCF = 0.6 + 0.01 * Σ (Peso * Valor)		TCF	0.89

$$TCF = 0.6 + 0.01 * 29 = 0.6 + 0.29 = 0.89$$

El Factor de Ambiente (EF) está relacionado con las habilidades y entrenamiento del equipo de desarrollo.

Tabla 27 Factor de ambiente.

Factor	Descripción	Peso	Valor asignado	Total
E1	Familiaridad con el modelo de proyecto utilizado	1.5	0	0
E2	Experiencia en la aplicación	0.5	0	0
E3	Experiencia en la orientación a objetivos.	1	4	4
E4	Capacidad del analista líder.	0.5	3	1.5
E5	Motivación.	1	5	5
E6	Estabilidad de requerimientos	2	5	10
E7	Personal Part-Time	-1	0	0
E8	Dificultad del lenguaje de programación	-1	3	-3
			Σ	17.5
	EF = 1.4 - 0.03 * Σ(Peso * Valor)		EF	0.875

$$EF = 1.4 - 0,03 * 17.5 = 1.4 - 0.525 = 0.875$$

Luego del resultado hallado se hace el cálculo, para completar la ecuación.

$$UCP = UUCP * TCF * EF = 41 * 0.89 * 0.875 = 31.92875$$

5.2.3. Estimación de esfuerzo a través de los Puntos de Casos de Uso.

Según lo planteado en este método, para el cálculo del esfuerzo $E=UCP*CF$ el factor de conversión CF, se obtiene contando cuantos valores de los que afectan el Factor Ambiente, están por debajo de la media (3), y los que están por arriba de la media. Si el total es 2 o menos se utiliza el factor de conversión 20 Horas-Hombre / Punto de Casos de uso. Si el total es 3 o 4 se utiliza el factor de conversión 28 Horas-Hombre / Punto de Casos de uso. Si el total es mayor o igual que 5 se recomienda efectuar cambios en el proyecto o desecharlo, por su alto riesgo de fracaso.

Sobre lo planteado se puede decir que CF toma un valor 20 horas-hombre/Punto de Casos de Uso. ($E = 31.92835 * 20 = 638.575$ horas-hombres)

5.2.3.1. Calcular esfuerzo de todo el proyecto.

Tabla 28 Esfuerzo del proyecto.

Actividad	Porcentaje %	Horas-Hombres
Análisis	30	478.9313
Diseño	20	319.2875
Implementación	40	638.575
Pruebas	10	159.6438
Sobrecarga (otras actividades)	0	0
Total	100	1596.438

Si $E_{hm} = 1596.438$ horas-hombre y se estima que cada mes tiene como promedio 240 horas laborables, eso daría un $E_{mh} = 7.982188$ mes-hombre.

Esto quiere decir que 1 persona puede realizar la tarea en 8 meses aproximadamente.

Se asume como salario promedio mensual \$100, personal 2 (CH), dada la estimación del tiempo (total del proyecto), se obtiene, el costo-hombres-mes ($CHM = 2 * \text{Salario Promedio}$) 200 \$/mes, el costo total del proyecto $\text{Costo} = CHM * E_{mh} = 200.00 * 7.982188 = 1596.438$

De los resultados obtenidos se interpreta que, con **2 hombres** trabajando en el proyecto, el mismo se desarrolla en **4 meses**, redondeado, y su costo total se estima que sea **\$1596.438**.

5.2.4. Beneficios tangibles e intangibles.

El *driver* para el protocolo PDU es un producto con fines comerciales. Puede ser utilizado por los VASP para desarrollar aplicaciones con servicios de mensajería SMS, MMS y WAP Push. El beneficio fundamental es contar con un sistema que a través del protocolo PDU, sea capaz de efectuar el envío y recepción de mensajes, a través, de un módem GPRS/GSM como se ha explicado en capítulos anteriores.

Como beneficio tangible se puede mencionar que este software forma parte de un paquete de drivers para SAMS-M solicitados por la compañía Sony Ericsson, por los cuales, se obtienen un importe de 80 mil dólares.

Algunos de los beneficios intangibles que este driver provee son:

- Reducción del tiempo de desarrollo de aplicaciones con servicios de mensajería.
- Posibilita el envío y recepción de mensajes sin contratos especiales con Operadores de Telefonía Inalámbrica.
- Reducción del costo por adquisición de licencias.

5.2.5. Análisis de costos y beneficios.

Cualquier producto informático es caro, por lo tanto, determinar si se lleva a cabo ó no, depende de varios factores como por ejemplo: el impacto económico, que valora los aportes y beneficios, reducción de costos, entre otros. En este caso el sistema ha sido desarrollado a petición de la compañía Sony Ericsson, el mismo será distribuido con licencia de Código Abierto (Open Source), por lo que no es necesaria la adquisición de licencias para el uso de este producto en la entidad Procyon lo cual puede resultar muy difícil dadas las condiciones de Cuba como país bloqueado.

Se puede decir, que el estudio de factibilidad realizado correspondiente al desarrollo de este *driver*, teniendo en cuenta el costo estimado, y los beneficios que reportará, mejorará el proceso de desarrollo en los proyectos orientados a esta tecnología.

5.3. Conclusiones del Capítulo

En este capítulo se obtuvieron los estimados de horas por hombre que se requieren para desarrollar la solución, y el costo aproximado teniendo en cuenta el salario hipotético que reciben los miembros del equipo de trabajo. Se realizó un análisis de los distintos indicadores y se concluyó que era factible la realización de este driver.

Conclusiones

Con el desarrollo de este trabajo, se profundizó en el conocimiento del protocolo PDU y se aplicó la metodología RUP, para guiar el proceso de desarrollo de software, con el cual se obtuvo la implementación de un driver para este protocolo con las funcionalidades previstas para el primer ciclo de desarrollo de la solución. El diseño y la implementación se rigieron por los estándares internacionales establecidos para este protocolo, los estándares de código de la entidad Procyon y se integró con SAMS-M, implementando las interfaces establecidas por esta API, aplicando diferentes patrones de diseño y el patrón de arquitectura Capas. Se definieron los parámetros de configuración con la documentación correspondiente, así como, un manual de instalación y una aplicación simple de ejemplo. Se puede concluir que se ha cumplido satisfactoriamente el objetivo trazado para este trabajo enfatizando en los siguientes puntos:

- Se implementaron las funcionalidades para envío y recepción de SMS de texto.
- Se desarrollaron las implementaciones para la codificación de 7bits, 8 bits y 16 bits según la especificación del protocolo.
- Se utilizó el Java Comm API para manejar el puerto de comunicaciones de la PC.
- Se emplearon los comandos AT compatibles con la especificación GSM 07.07 y GSM 07.05 para la comunicación con el módem.
- Se provee la posibilidad de enviar y recibir SMS de texto largo con más de 160 caracteres.
- Se realizaron pruebas comprobando la posibilidad de utilizar diferentes dispositivos compatibles con la especificación GSM 07.07 y GSM 07.05 para el envío y recepción de mensajes SMS obteniendo resultados satisfactorios.
- Se realizaron pruebas del driver por diferentes puertos físicos de comunicación de la PC obteniendo resultados satisfactorios.

Seguidamente se realizan una serie de recomendaciones que han de tomarse en cuenta para la continuación de la investigación y el desarrollo de los próximos ciclos de desarrollo.

Recomendaciones

A continuación se mencionan algunas recomendaciones con el objetivo de realizar un seguimiento y mejora de esta investigación.

- Culminar los ciclos de desarrollo según la planificación propuesta en este trabajo y guiándose por el diseño realizado.
- Planificar pruebas al driver en condiciones de estrés, evaluando posibles herramientas para automatizar este proceso.
- Realizar pruebas con Operadores de Telefonía Internacionales.

Bibliografía

- 3GPP Specification TS 07.05. 2007.** 3GPP TS 07.05. *3GPP TS 07.05.* [En línea] 2007. [Citado el: 7 de 1 de 2007.] <http://www.3gpp.org/ftp/Specs/html-info/0705.htm>.
- 3GPP Specification TS 07.07. 2007.** 3GPP TS 07.07. *3GPP TS 07.07.* [En línea] 25 de 5 de 2007. [Citado el: 3 de 1 de 2007.] <http://www.3gpp.org/ftp/Specs/html-info/0707.htm>.
- 3GPP Specification TS 23.040. 2006.** 3GPP TS 23.040. *3GPP TS 23.040.* [En línea] 3 de 2006. [Citado el: 10 de 1 de 2007.] <http://www.3gpp.org/ftp/Specs/html-info/23040.htm>.
- Advance Wireless Planet.** SMS Text Mode. [En línea] [Citado el: 15 de 12 de 2006.] <http://www.gsm-modem.de/sms-text-mode.html>.
- AME Info. 2003.** Wap Push. [En línea] 8 de 12 de 2003. [Citado el: 14 de 3 de 2007.] <http://www.ameinfo.com/31971.html>.
- Baker, Art y Lozano, Jerry. nov2000.** *The Windows 2000 Device Driver Book. A Guide for Programmers. Second Edition.* nov2000. 0-13-020431-5.
- Corbet, Jonathan, Kroah-Hartman, Greg y Rubini, Alessandro. feb2005.** *Linux Device Drivers 3rd Edition.* feb2005. 0-596-00590-3.
- Eclipse.** Eclipse.org. [En línea] [Citado el: 5 de 10 de 2006.] <http://www.eclipse.org/>.
- Federal Standard 1037C. 1996.** Telecommunications: Glossary of Telecommunication Terms. *Telecommunications: Glossary of Telecommunication Terms.* [En línea] 7 de 8 de 1996. [Citado el: 4 de 12 de 2006.] <http://www.its.bldrdoc.gov/fs-1037/fs-1037c.htm>.
- Granada, Ailec y Borges, Liudmila. abr2006.** *Sistema de Mensajería Instantánea .YaBEEP.* abr2006.
- International Engineering Consortium.** Wireless Short Message Service. [En línea] [Citado el: 26 de 12 de 2006.] <http://www.iec.org>.
- Jacobson, Igor, Booch, Grady y Rumbaugh, James.** *El proceso unificado desarrollo de software.* s.l. : Madrid : Pearson Education S.A., 2000. 84-7829-036-2.
- Java Community Process. 2005.** JSR 212: SAMS-M. *JSR 212: Server API for Mobile Services: Messaging - SAMS: Messaging.* [En línea] 12 de 1 de 2005. [Citado el: 10 de 12 de 2006.] <http://jcp.org/en/jsr/detail?id=212>.

- Le Bodic, Gwenael. 2003.** *Mobile Messaging Technologies and Services SMS EMS and MMS*. Alcatel, France : s.n., 2003.
- . **2003.** *Multimedia Messaging Service An Engineering Approach to MMS*. Alcatel, France : s.n., 2003.
- MPirical.** MMSC - Multimedia Messaging Service Centre. [En línea] [Citado el: 2 de 2 de 2007.] http://www.mpirical.com/companion/Multi_Tech/MMSCCentre.htm.
- NOKIA. jun2003.** *How to Create MMS Services*. jun2003.
- Now Wireless Limited.** What is NowSMS? *NowSMS*. [En línea] [Citado el: 20 de 2 de 2006.] <http://www.nowsms.com/whatisnowsms.htm#whynowsms>.
- OMA. oct2005.** *MMS Conformance Document*. oct2005.
- . **sep2005.** *Multimedia Messaging Service Client Transactions*. sep2005.
- . **sep2005.** *Multimedia Messaging Service. Encapsulation Protocol*. sep2005.
- Phone Scoop.** SMS. [En línea] [Citado el: 11 de 12 de 2006.] <http://www.phonescoop.com/glossary/term.php?gid=86>.
- PhoneSoope.** MMS. [En línea] [Citado el: 1 de 2 de 2007.] <http://www.phonescoop.com/glossary/term.php?fid=43>.
- SubEclipse. 2004.** SubEclipse. *SubEclipse*. [En línea] 2004. [Citado el: 20 de 10 de 2006.] (<http://subclipse.tigris.org/>).
- SUN Microsystems.** Java Virtual Machine. *Java Virtual Machine*. [En línea] [Citado el: 15 de 10 de 2006.] http://java.sun.com/javase/downloads/index_jdk5.jsp.
- TechTarget. 2005.** Multimedia Messaging Service. [En línea] 20 de 9 de 2005. [Citado el: 1 de 2 de 2007.] <http://searchmobilecomputing.techtarget.com/>.
- . Short Message. [En línea] [Citado el: 1 de 2 de 2007.] <http://searchmobilecomputing.techtarget.com/>.
- Tigris.org.** Projects Subversion Open Source Software Engineering Tools. [En línea] [Citado el: 23 de 11 de 2006.] <http://subversion.tigris.org/>.
- Vázquez Méndez-Quintero, Emilio, García, Javier Ramón y Schlingman, Martin. 2006.** *The PROCYON Coding Standards*. Ciudad de la Habana : s.n., 2006.
- WAP Forum. 2002.** *WAP MMS Client Transactions*. 2002. WAP-206-MMSCRT-20020115-a.
- . **2002.** WAP-209-MMSEncapsulation. *Wireless Application Protocol MMS Encapsulation Protocol*. [En línea] 5 de 1 de 2002. [Citado el: 20 de 1 de 2007.] <http://www.wapforum.org/what/technical.htm>.

—. 2001. WAP-230-WSP. *Wireless Application Protocol - Wireless Session Protocol Specification*. [En línea] 5 de 7 de 2001. [Citado el: 14 de 2 de 2007.] <http://www.wapforum.org/what/technical.htm>.

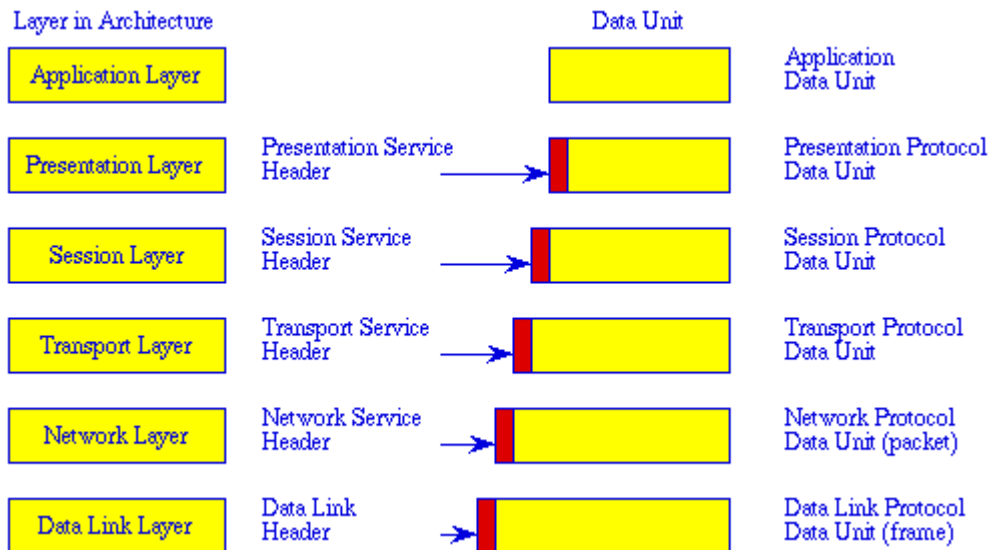
Wikipedia. Push Proxy Gateway. [En línea] [Citado el: 10 de 4 de 2007.] http://en.wikipedia.org/wiki/Push_Proxy_Gateway.

Windle, Daniel R. y Abreo, L. Rene. aug2002. *Software Requirements Using the Unified Process: A Practical Approach*. aug2002. 0-13-096972-9.

Anexos

Anexo 1. Encapsulación de PDU por capas del Modelo OSI

El proceso consiste en que cuando un paquete es enviado por un proceso de una aplicación a la capa, esta lo procesa. Se adiciona al inicio un protocolo de encabezado y se envía el paquete (junto con el encabezado) a la capa inferior. La capa en turno procesa los datos y se suma un nuevo encabezado de manera adicional. Cada capa trata la información acumulada como datos de la capa superior y no se preocupa sobre el contenido. El proceso continúa hasta que el paquete alcanza la capa física del Modelo OSI. La siguiente figura muestra los encabezados asociados con cada una de las capas del Modelo OSI.



La capa física serializa el paquete (convierte el paquete a una serie de bits) y lo envía a través del cable o circuitos de telecomunicación al sistema destino o a un intermediario. En el receptor, el sistema remoto re ensambla la serie de bits para formar un paquete y lo reenvía para que sea procesado por la capa de enlace, esta elimina el encabezado de la capa de enlace y lo envía a la próxima capa. El proceso continúa hasta que finalmente el paquete de datos original es enviado al proceso de la aplicación remota.

Anexo 2. Encapsulación del datagrama PDU

La Encapsulación consiste en la adición de PCI y de un PDU. La encapsulación adiciona los encabezados al inicio del PDU. Para comprender este proceso se muestra la siguiente figura.

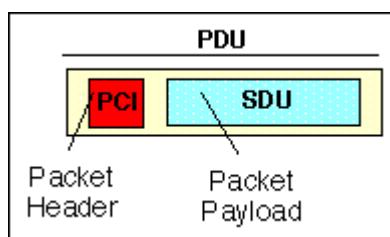


Figura 22. Componentes de un datagrama del protocolo PDU

La encapsulación ocurre en cada una de las capas del Modelo OSI y para explicar la encapsulación OSI define el concepto de “servicio” y servicio de unidad de dato SDU (*del inglés: Service Data Unit*).

Un SDU es una pieza de información pasada de una capa superior (Capa N+1) a la capa actual (Capa N) para transmisión utilizando el servicio de esa capa. Para transportar el SDU, la capa actual encapsula el SDU adicionándole un protocolo de encabezado PCI. La combinación de PCI y SDU se conoce como el PDU que pertenece a esa capa, este a su vez forma el SDU de la capa inferior. Este proceso se conoce como encapsulación. La siguiente figura muestra el procesamiento para transmisión por la capa N del protocolo de encabezado.

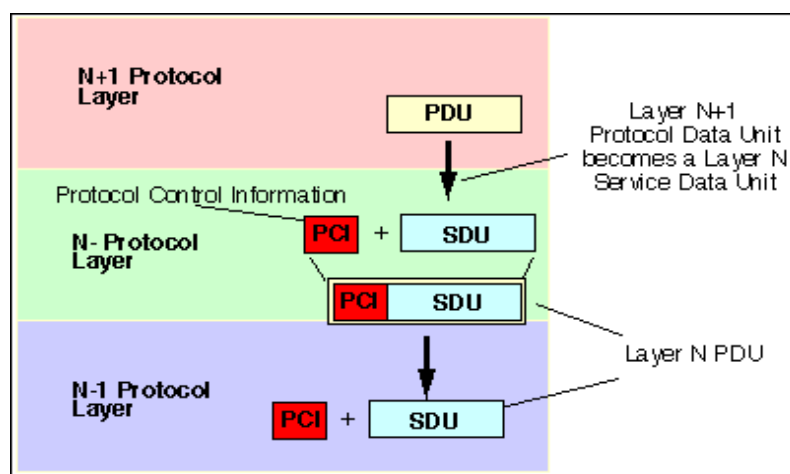


Figura 23. Procesamiento de un datagrama para transmisión.

La siguiente figura muestra el proceso de recepción, básicamente es el proceso inverso a la transmisión donde la capa N separa el protocolo de encabezado y pasa el SDU a la capa superior este SDU forma el PDU de la capa N+1.

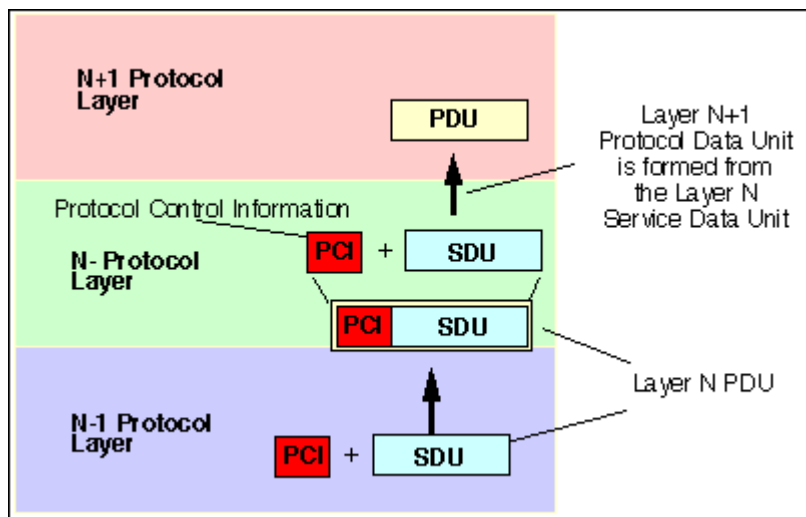


Figura 24. Procesamiento de un datagrama para recepción

Anexo 3. Diagramas de Colaboración del Análisis.

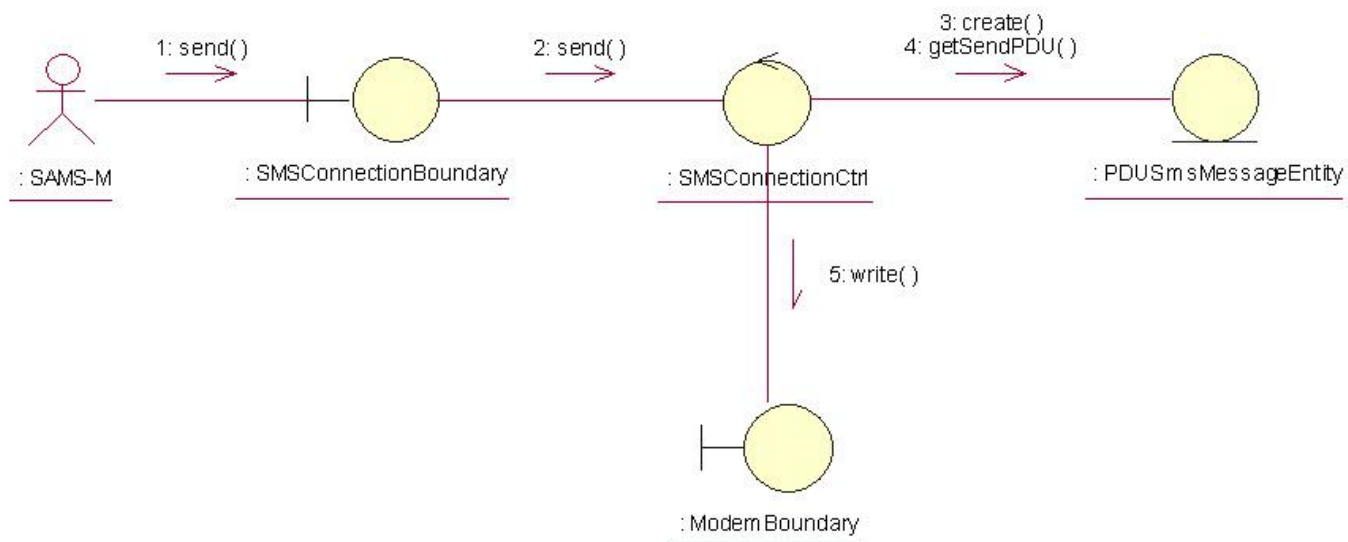


Figura 25. Diagrama de Colaboración del Análisis. Enviar SMS.

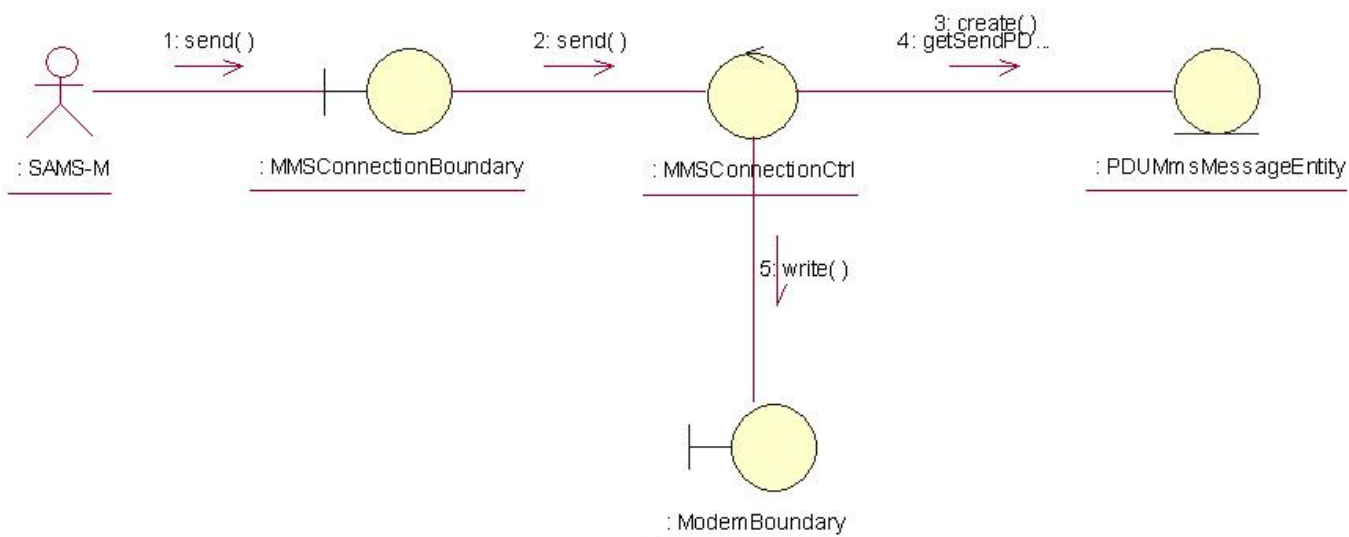


Figura 26. Diagrama de Colaboración del Análisis. Enviar MMS.

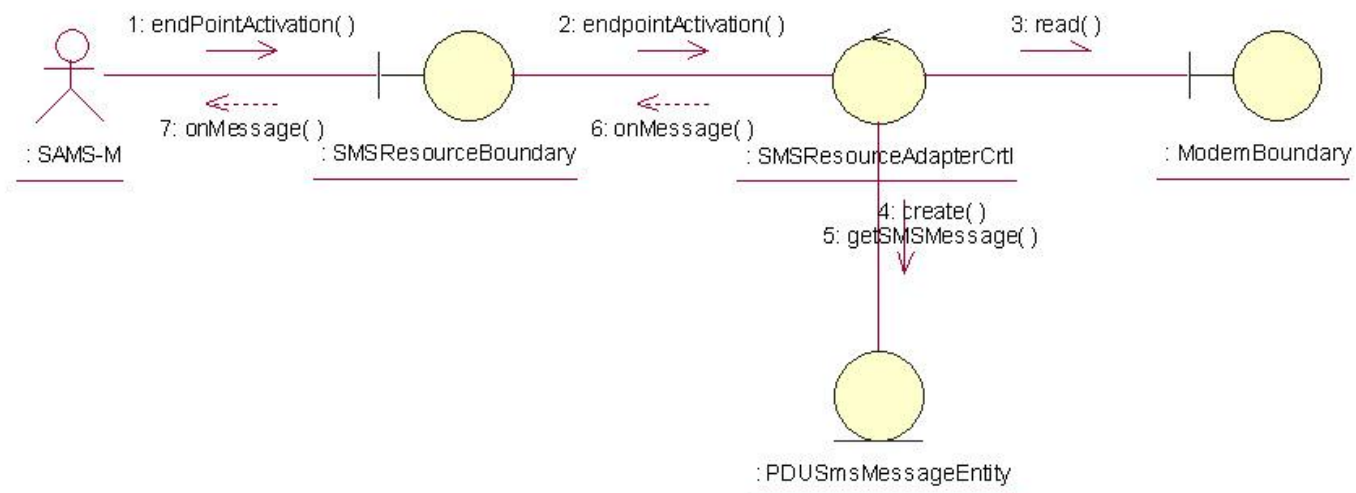


Figura 27. Diagrama de Colaboración del Análisis. Recibir SMS.

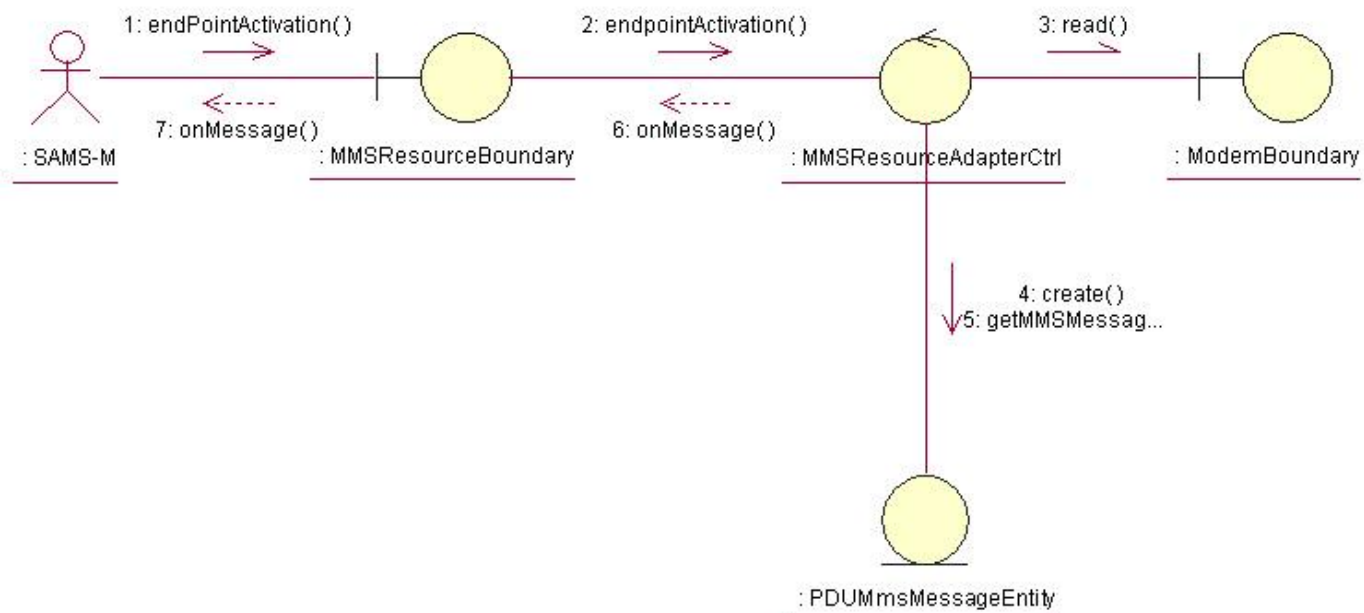


Figura 28. Diagrama de Colaboración del Análisis. Recibir MMS.

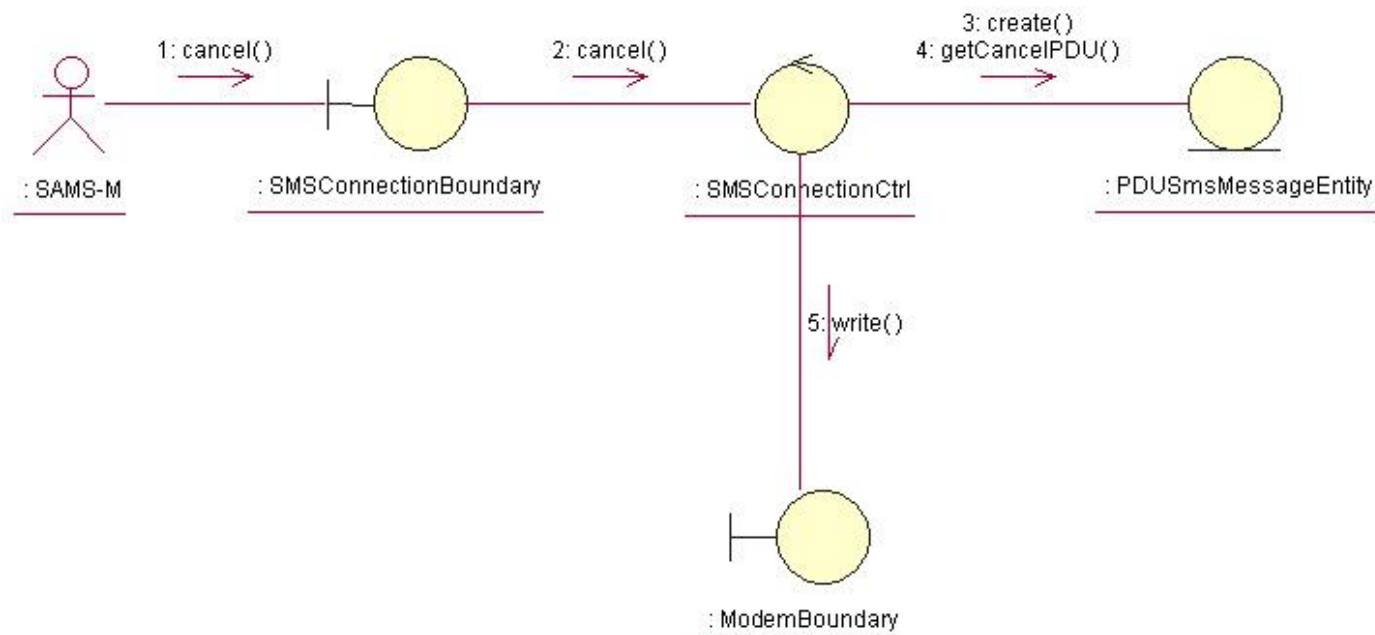


Figura 29. Diagrama de Colaboración del Análisis. Cancelar SMS.

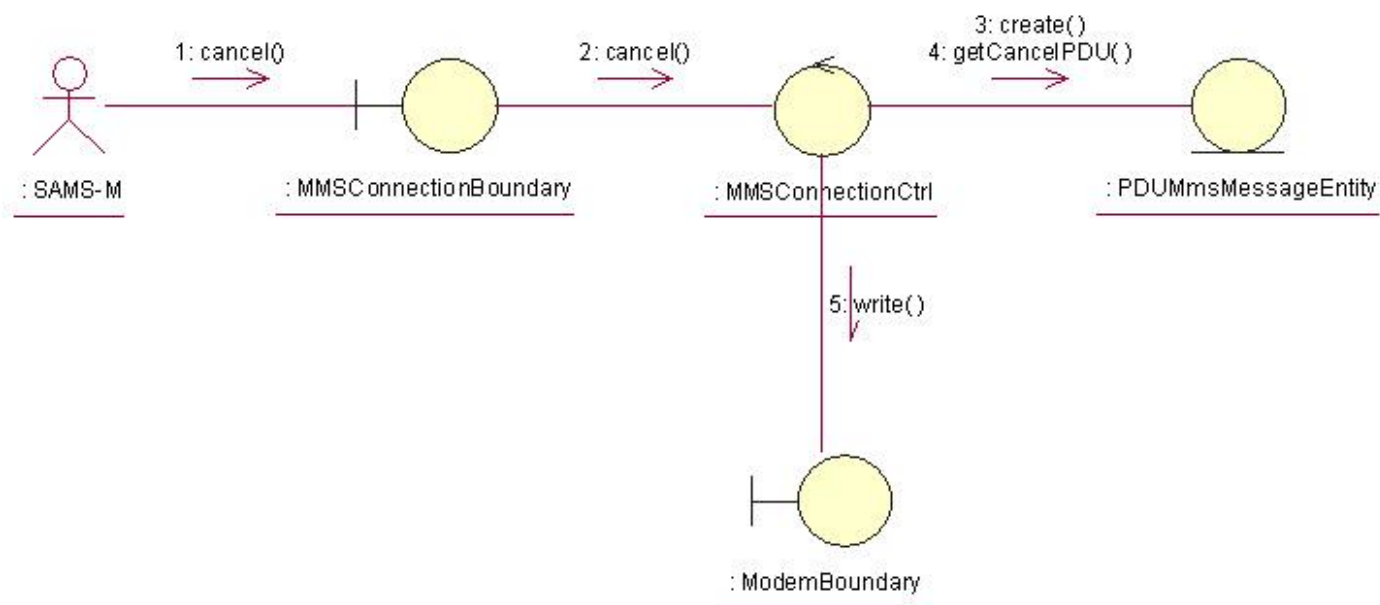


Figura 30. Diagrama de Colaboración del Análisis. Cancelar MMS.

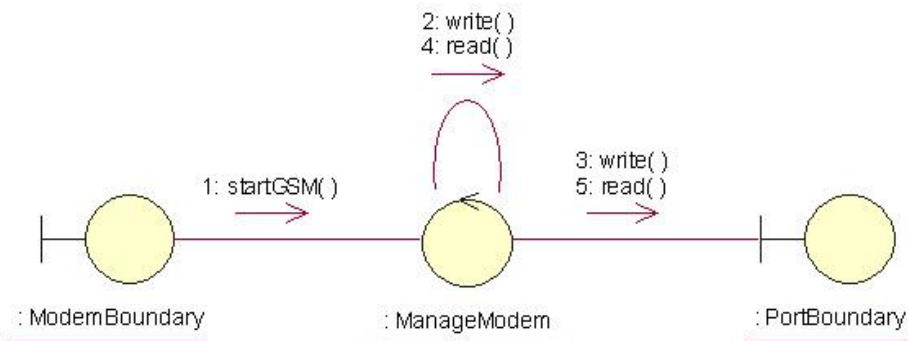


Figura 31. Diagrama de Colaboración del Análisis. Administrar Módem (Iniciar conexión GSM).

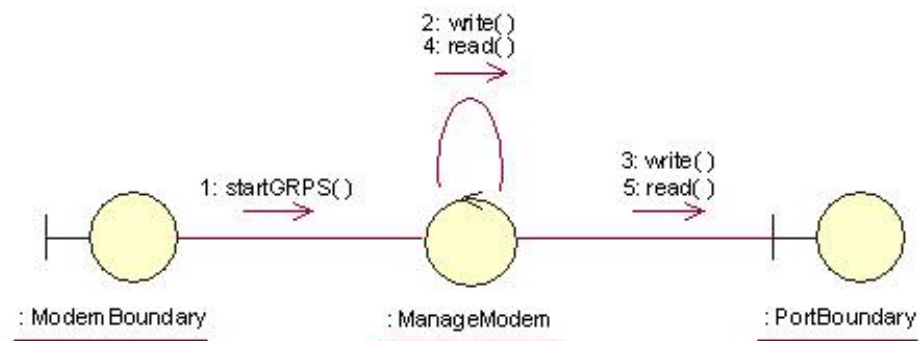


Figura 32. Diagrama de Colaboración del Análisis. Administrar Módem (Iniciar conexión GPRS).

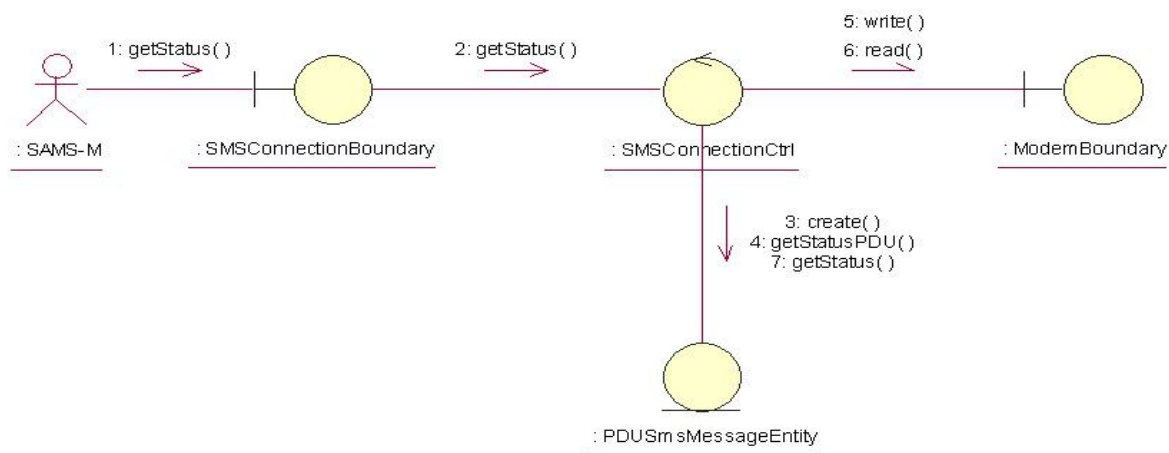


Figura 33. Diagrama de Colaboración del Análisis. Estado del SMS.

Anexo 4. Diagramas de Secuencia del Diseño.

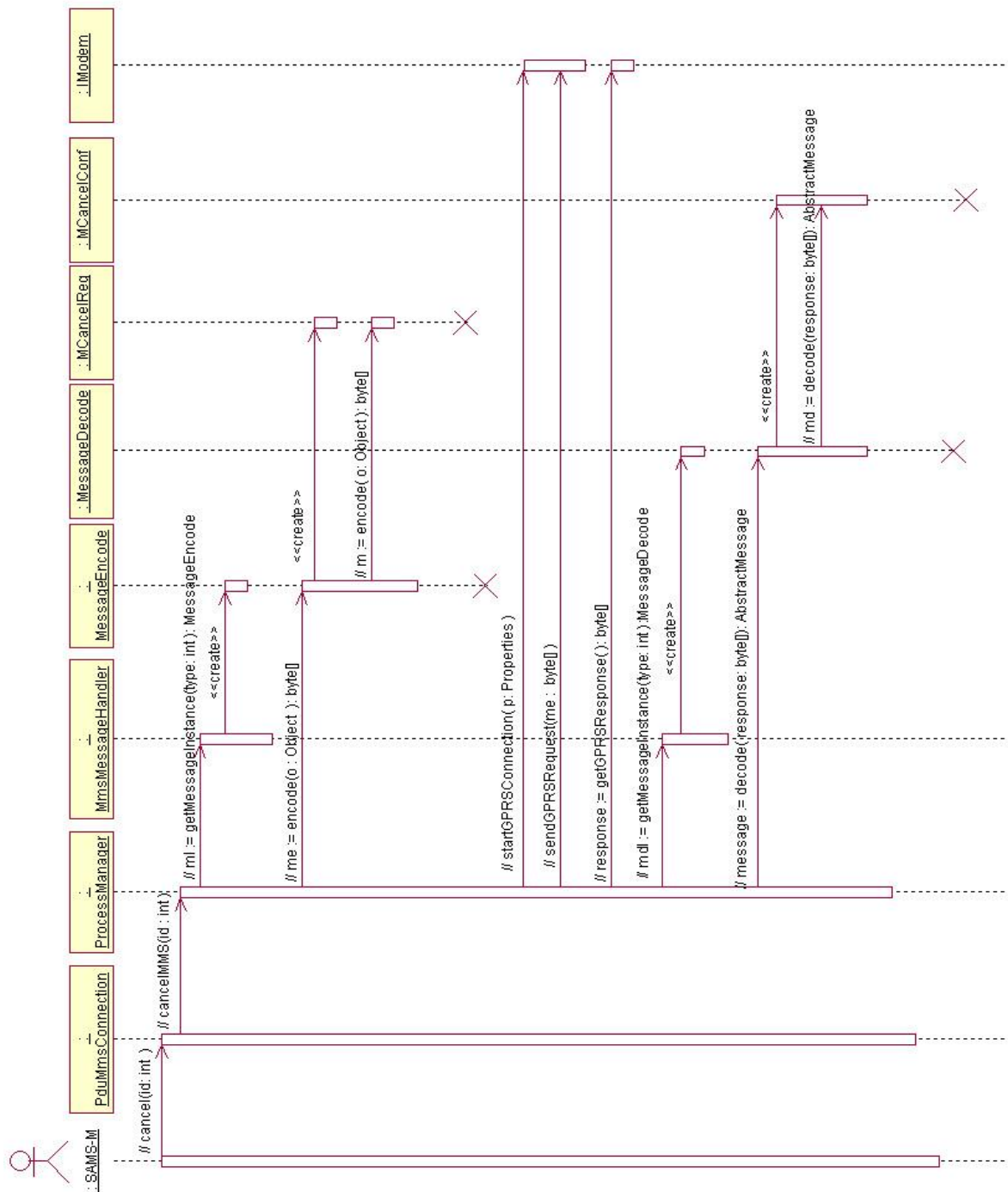


Figura 34. Diagrama de Secuencia del Diseño. Cancelar MMS.

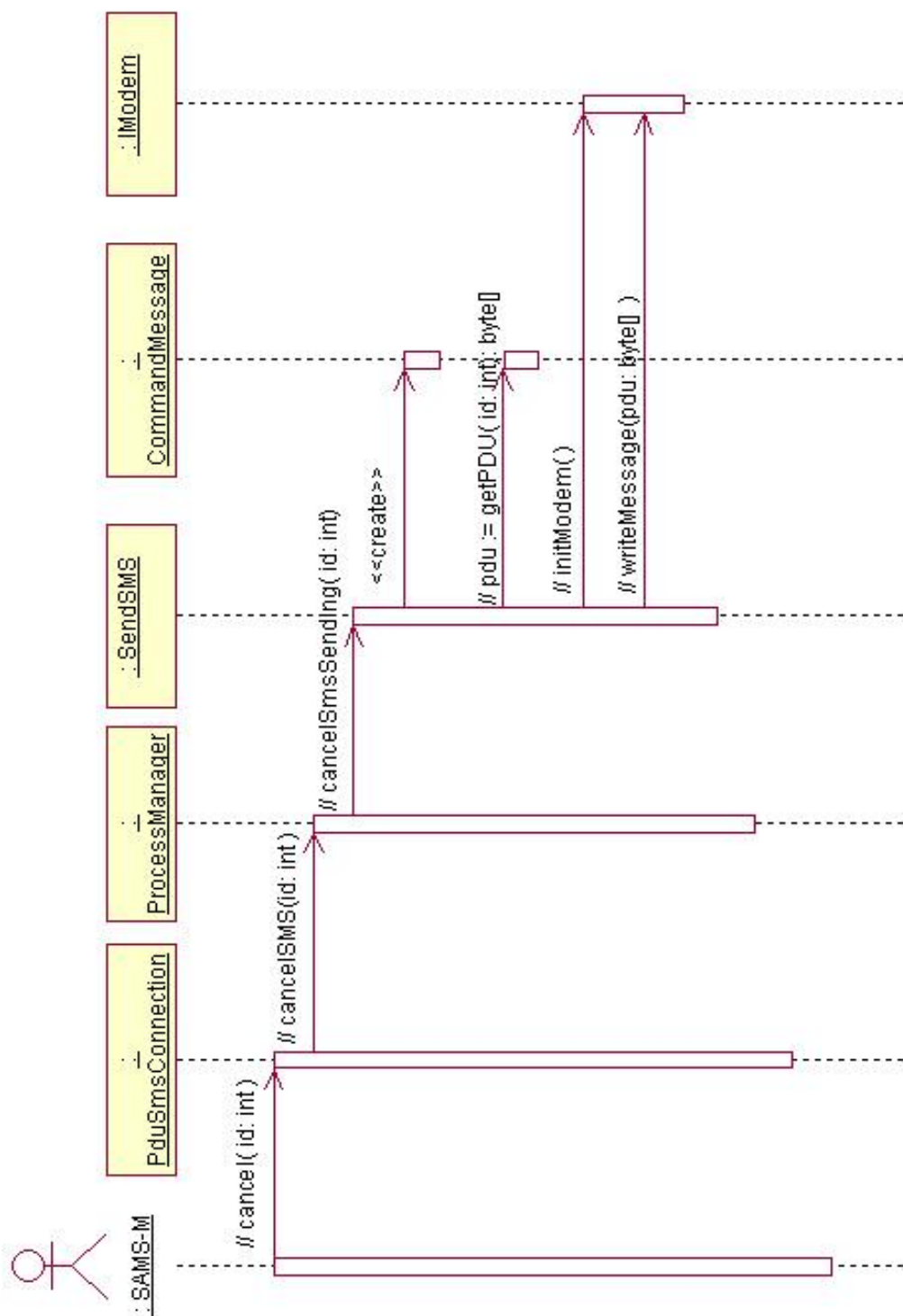


Figura 35. Diagrama de Secuencia del Diseño. Cancelar SMS.

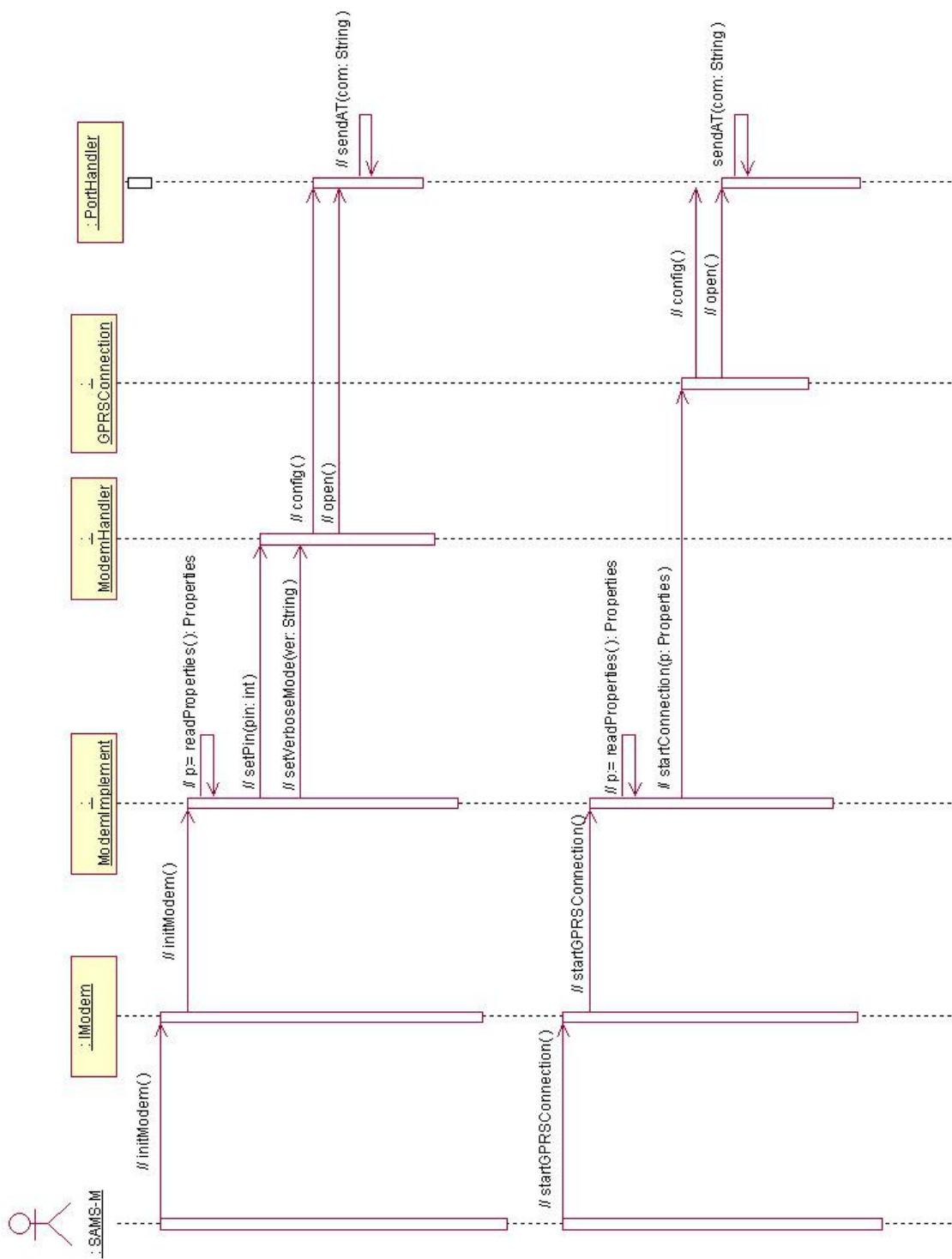


Figura 36. Diagrama de Secuencia del Diseño. Administrar Módem

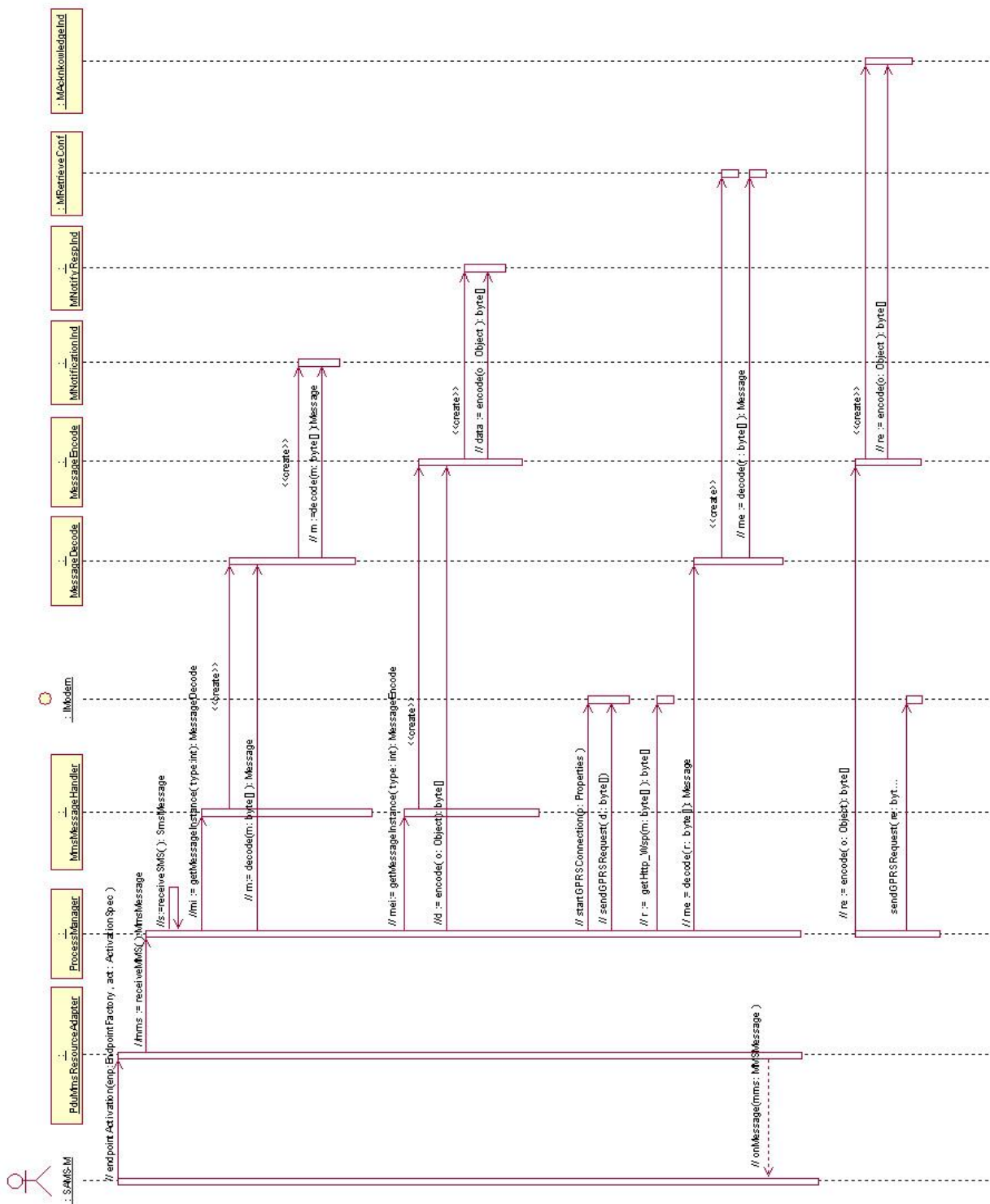


Figura 37. Diagrama de Secuencia del Diseño. Recibir MMS.

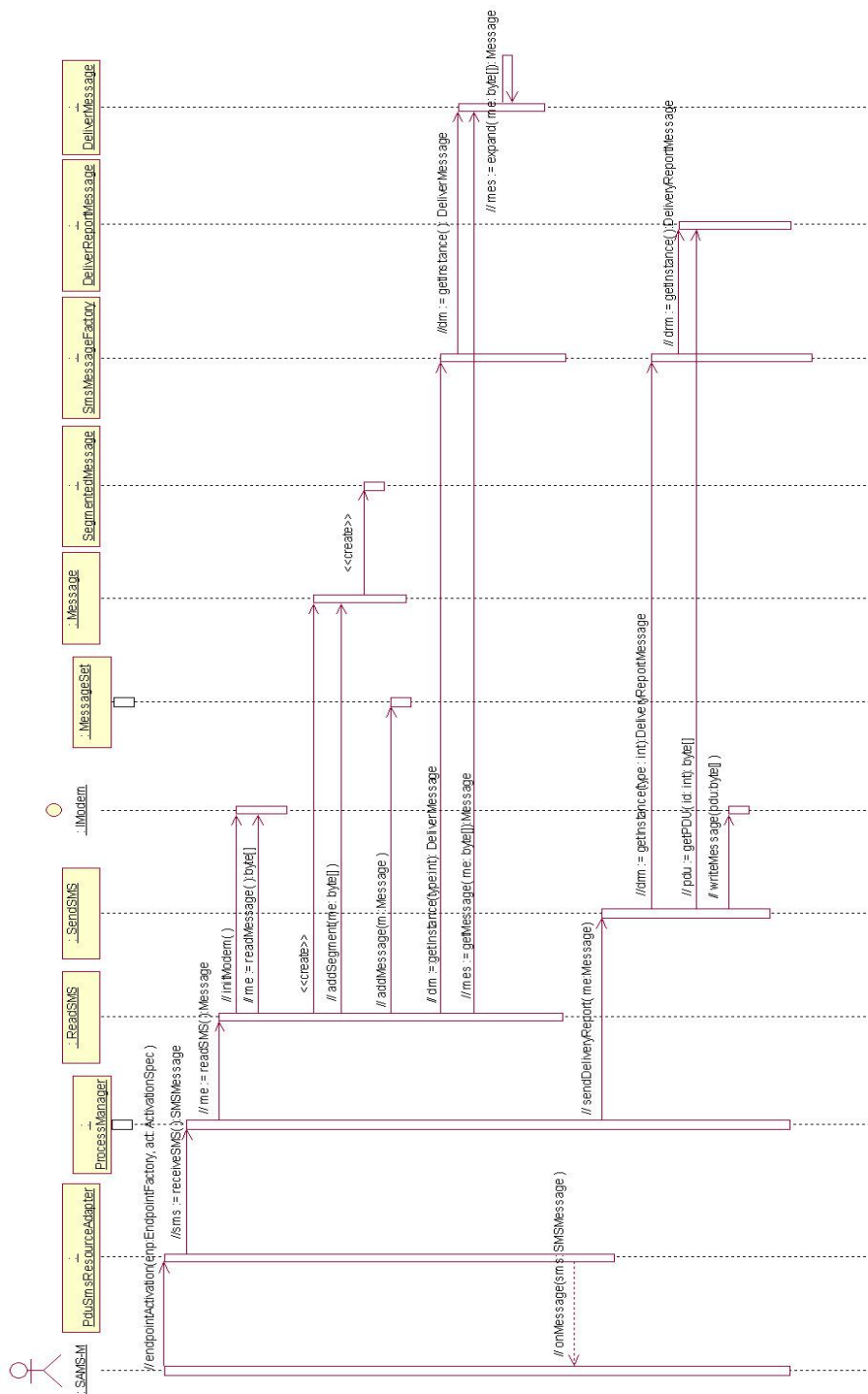


Figura 38. Diagrama de Secuencia del Diseño. Recibir SMS.

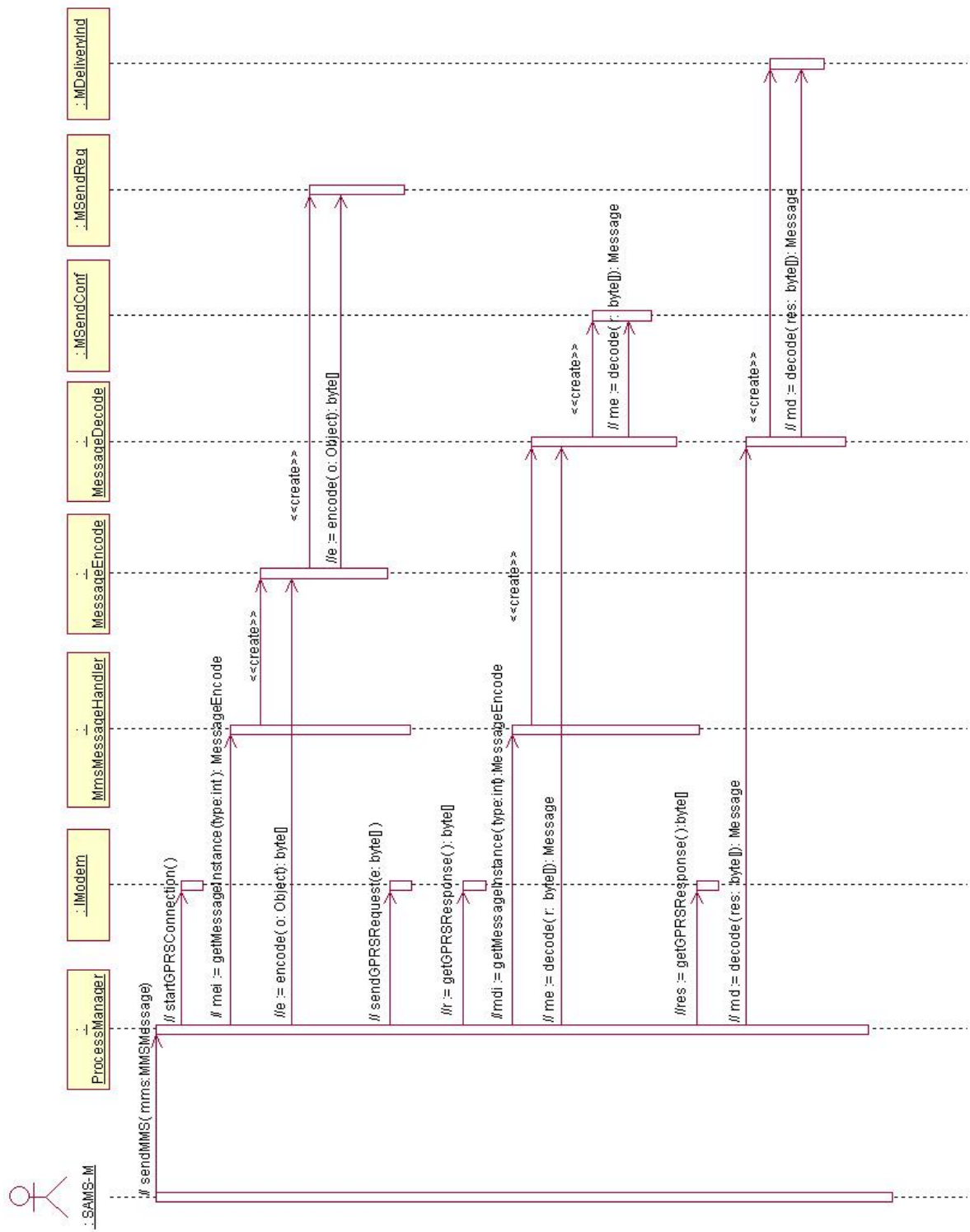


Figura 39. Diagrama de Secuencia del Diseño. Enviar MMS.

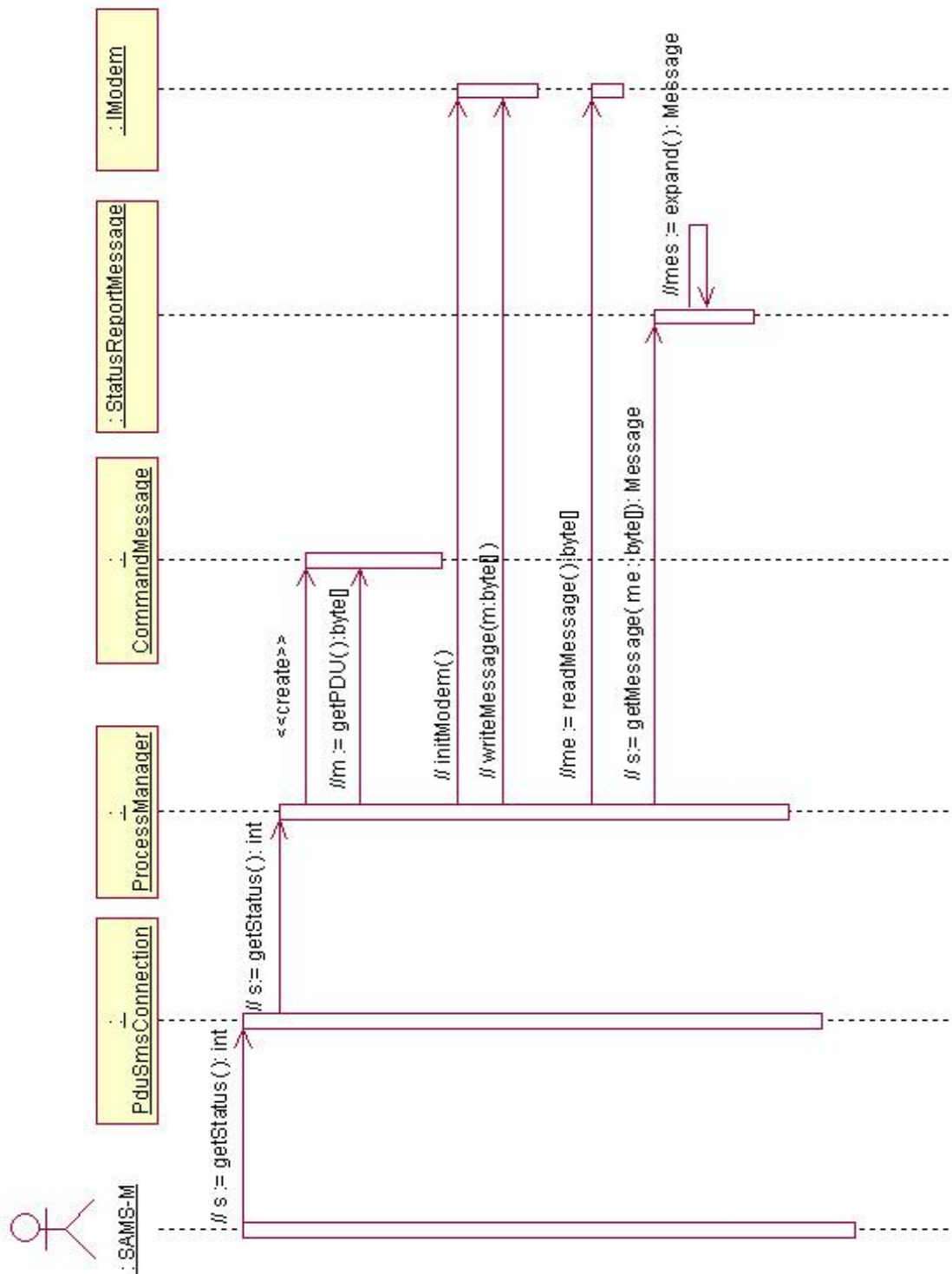


Figura 41. Diagrama de Secuencia del Diseño. Estado del SMS.

Anexo 5. MMS en formato PDU.

```

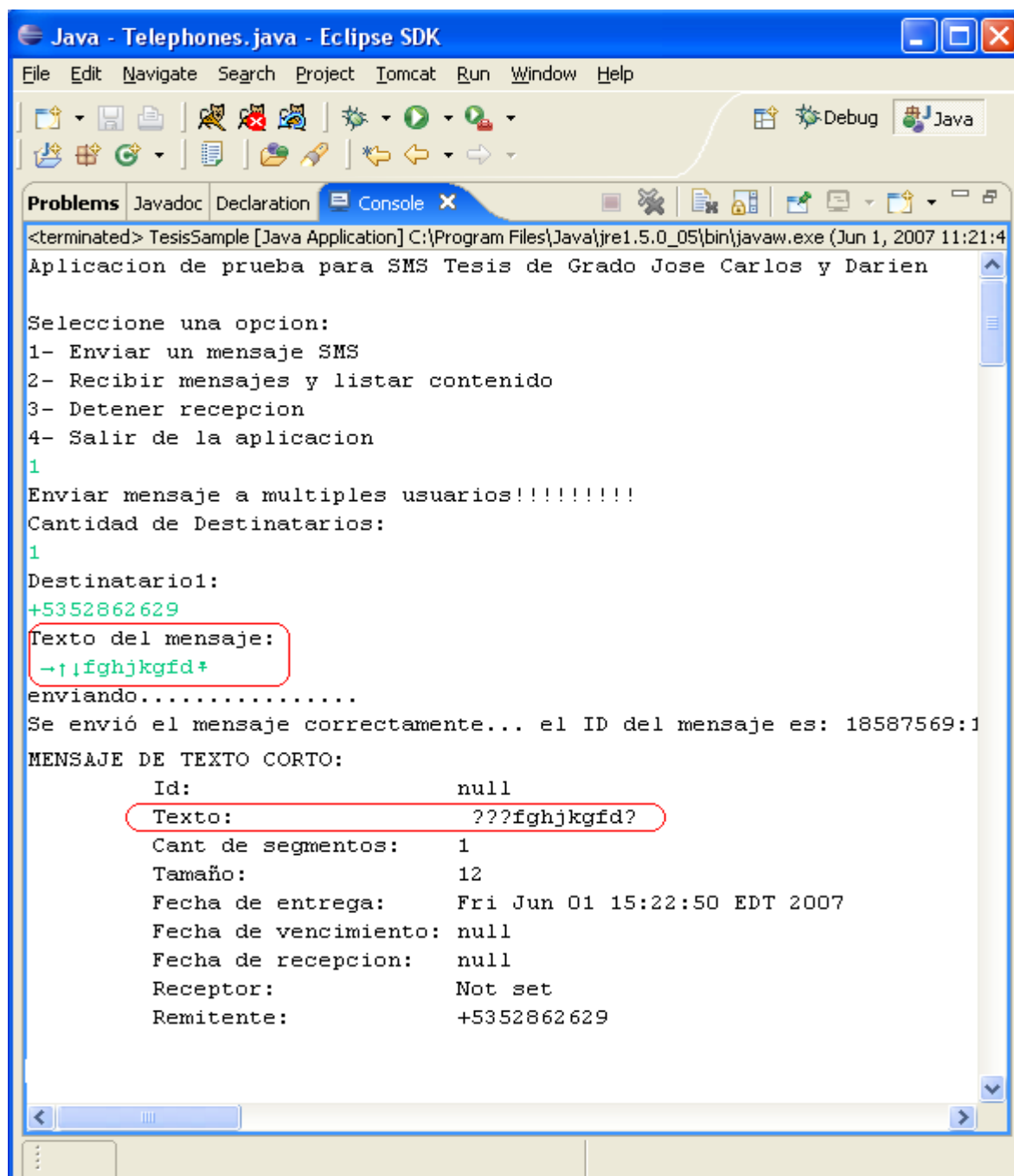
0000: 8C80 9830 3132 3334 3536 3738 3900 8D90
0010: 8910 802B 3132 332F 5459 5045 3D50 4C4D
0020: 4E00 972B 3435 362F 5459 5045 3D50 4C4D
0030: 4E00 964D 7920 6669 7273 7420 7465 7374
0040: 206D 6573 7361 6765 2100 841B B389 6170
0050: 706C 6963 6174 696F 6E2F 736D 696C 008A
0060: 3C30 3030 303E 0007 2385 1D61 7070 6C69
0070: 6361 7469 6F6E 2F73 6D69 6C00 436F 6E74
0080: 656E 742D 4944 003C 3030 3030 3E00 3C73
0090: 6D69 6C20 786D 6C6E 733D 2268 7474 703A
00A0: 2F2F 7777 772E 7733 2E6F 7267 2F32 3030
00B0: 312F 534D 494C 3230 2F4C 616E 6775 6167
00C0: 6522 3E0D 0A20 203C 6865 6164 3E0D 0A20
00D0: 2020 203C 6C61 796F 7574 3E0D 0A20 2020
00E0: 2020 203C 726F 6F74 2D6C 6179 6F75 7420
00F0: 7769 6474 683D 2231 3630 2220 6865 6967
0100: 6874 3D22 3134 3022 2F3E 0D0A 2020 2020
0110: 2020 3C72 6567 696F 6E20 6964 3D22 496D
0120: 6167 6522 2077 6964 7468 3D22 3136 3022
0130: 2068 6569 6768 743D 2231 3230 2220 6C65
0140: 6674 3D22 3022 2074 6F70 3D22 3022 2F3E
0150: 0D0A 2020 2020 2020 3C72 6567 696F 6E20
0160: 6964 3D22 5465 7874 2220 7769 6474 683D
0170: 2231 3630 2220 6865 6967 6874 3D22 3230
0180: 2220 6C65 6674 3D22 3022 2074 6F70 3D22
0190: 3132 3022 2F3E 0D0A 2020 2020 3C2F 6C61
01A0: 796F 7574 3E0D 0A20 203C 2F68 6561 643E
01B0: 0D0A 0D0A 2020 3C62 6F64 793E 0D0A 2020
01C0: 2020 3C70 6172 2064 7572 3D22 3573 223E
01D0: 0D0A 2020 2020 2020 3C69 6D67 2073 7263
01E0: 3D22 536D 696C 6579 4661 6365 2E67 6966
01F0: 2220 7265 6769 6F6E 3D22 496D 6167 6522
0200: 202F 3E0D 0A20 2020 2020 203C 7465 7874
0210: 2073 7263 3D22 4865 6C6C 6F57 6F72 6C64
0220: 2E74 7874 2220 7265 6769 6F6E 3D22 5465
0230: 7874 2220 2F3E 0D0A 2020 2020 2020 3C61
0240: 7564 696F 2073 7263 3D22 4865 6C6C 6F57
0250: 6F72 6C64 2E61 6D72 2220 2F3E 0D0A 2020
0260: 2020 3C2F 7061 723E 0D0A 2020 2020 3C70
0270: 6172 2064 7572 3D22 3130 7322 3E0D 0A20
0280: 2020 2020 203C 696D 6720 7372 633D 2263
0290: 6964 3A54 6865 456E 642E 6769 6622 2072
02A0: 6567 696F 6E3D 2249 6D61 6765 2220 2F3E
02B0: 0D0A 2020 2020 2020 3C74 6578 7420 7372
02C0: 633D 2263 6964 3A54 6865 456E 642E 7478
02D0: 7422 2072 6567 696F 6E3D 2254 6578 7422
02E0: 202F 3E0D 0A20 2020 2020 203C 6175 6469
02F0: 6F20 7372 633D 2263 6964 3A59 4342 4E59
0300: 482E 616D 7222 202F 3E0D 0A20 2020 203C
0310: 2F70 6172 3E0D 0A20 203C 2F62 6F64 793E
0320: 0D0A 3C2F 736D 696C 3E0D 0A11 8A7B 9D8E
0330: 536D 696C 6579 4661 6365 2E67 6966 0047
0340: 4946 3839 61A0 0077 00F7 0000 0000 0000
.
.
.
08B0: F8E6 ABEF BE03 0504 003B 110C 838E 4865
08C0: 6C6C 6F57 6F72 6C64 2E74 7874 0048 656C
08D0: 6C6F 2057 6F72 6C64 211A 8729 6175 6469
08E0: 6F2F 616D 7200 8E48 656C 6C6F 576F 726C
08F0: 642E 616D 7200 2321 414D 520A 0C85 68BF
.
.
.
0C90: 7C7C 7C44 E463 25A1 727C 7C7C 7C7C 7C19
0CA0: 8B68 9D43 6F6E 7465 6E74 2D49 4400 3C54
0CB0: 6865 456E 642E 6769 663E 0047 4946 3839
.
.
.
1290: 4827 ADF4 D24C 37ED F4D3 5047 2D35 4B01
12A0: 0100 3B19 0883 436F 6E74 656E 742D 4944
12B0: 003C 5468 6545 6E64 2E74 7874 3E00 5468
12C0: 6520 656E 642E 2289 5D61 7564 696F 2F61
12D0: 6D72 0043 6F6E 7465 6E74 2D49 4400 3C59
12E0: 4342 4E59 482E 616D 723E 0023 2141 4D52
.
.
.
17B0: 8322 5972 7C7C 7C7C 7C7C 7C44 E470 D259
17C0: 727C 7C7C 7C7C 7C7C
.
.
.
...0123456789...
...+123/TYPE=PLM
N..+456/TYPE=PLM
N..My first test
message!....ap
plication/smil..
<0000>..#.appli
cation/smil.Cont
ent-ID.<0000>.<s
mil xmlns="http
://www.w3.org/200
1/SMIL20/Languag
e">... <head>...
<layout>...
<root-layout
width="160" heig
ht="140"/>...
<region id="Im
age" width="160"
height="120" le
ft="0" top="0"/>
..
<region
id="Text" width=
"160" height="20
" left="0" top="
120"/>... </la
yout>... </head>
.... <body>...
<par dur="5s">
..
<img src
="SmileyFace.gif
" region="Image"
/>... <text
src="HelloWorld
.txt" region="Te
xt" />... <a
udio src="HelloW
orld.amr" />...
</par>... <p
ar dur="10s">...

..
<text sr
c="cid:TheEnd.tx
t" region="Text"
/>... <audi
o src="cid:YCBNY
H.amr" />... <
/par>... </body>
...</smil>...{..
SmileyFace.gif.G
IF89a..w.....
.
.
.
.....;....He
lloWorld.txt.Hel
lo World!..)audi
o/amr..HelloWorl
d.amr.#!AMR...h.
.
.
.
|||D.c%.r| |||||.
.h.Content-ID.<T
heEnd.gif>.GIF89
.
.
.
H'...L7...PG-5K.
...Content-ID
.<TheEnd.txt>.Th
e end.".]audio/a
mr.Content-ID.<Y
CBNYH.amr>.#!AMR
.
.
.
."Yr |||||D.p.Y
r |||||

```

Figura 42. MMS en formato PDU (M-Send.req)

Anexo 6. Muestra del error identificado.



```
<terminated> TesisSample [Java Application] C:\Program Files\Java\jre1.5.0_05\bin\javaw.exe (Jun 1, 2007 11:21:4
Aplicacion de prueba para SMS Tesis de Grado Jose Carlos y Darien

Seleccione una opcion:
1- Enviar un mensaje SMS
2- Recibir mensajes y listar contenido
3- Detener recepcion
4- Salir de la aplicacion
1
Enviar mensaje a multiples usuarios!!!!!!!!!!
Cantidad de Destinatarios:
1
Destinatario1:
+5352862629
Texto del mensaje:
->|fghjkgfd*
enviando.....
Se envi  el mensaje correctamente... el ID del mensaje es: 18587569:1
MENSAJE DE TEXTO CORTO:
      Id:                null
      Texto:              ???fghjkgfd?
      Cant de segmentos:  1
      Tama o:             12
      Fecha de entrega:   Fri Jun 01 15:22:50 EDT 2007
      Fecha de vencimiento: null
      Fecha de recepcion: null
      Receptor:          Not set
      Remitente:         +5352862629
```

Figura 43. Error identificado.

Glosario de Términos

- **3GPP** (del inglés: *3rd Generation Partnership Project*; Consorcio o grupo de desarrolladores que aboga por la estandarización de las comunicaciones para móviles.)
- **API** (del inglés: *Application Programming Interface*, en español, *Interfaz de Programación de Aplicaciones*)
- **Centros de Mensajería** (*SMSC, MMSC*)
- **CIMD, CIMD 2** (del inglés: *Computer Interface for Message Distribution*; *Protocolo de comunicación desarrollado por NOKIA*)
- **Driver** (Programa diseñado para hacer que un dispositivo concreto de hardware interactúe con un sistema operativo o software.)
- **EAIF** (*External Application InterFace*)
- **EJB** (del inglés: *Enterprise Java Bean*; *Los EJB son componentes distribuidos, quiere decir, componentes que se ejecutan en diferentes servidores, contra diferentes bases de datos, o no, es una decisión del analista-desarrollador*)
- **EMI** (*External Machine Interface*)
- **EMI/UCP** (*External Machine Interface / Universal Computer Protocol*; *Protocolo desarrollado por LogicaCMG*)
- **Framework** (Es una estructura de soporte definida en la cual otro proyecto de software puede ser organizado y desarrollado)
- **Gateway** (*Puerta de Enlace*; *Es un servidor que actúa como intermediario para otro servidor; programa o dispositivo de comunicaciones que transfiere datos entre redes que tienen funciones similares pero operativas diferentes.*)
- **GPRS** (del inglés: *General Packet Radio Service*; *Extensión de GSM, Transmisión inalámbrica de paquetes de datos.*)
- **GSM** (del inglés: *Global System for Mobile Communications*, en español, *Sistema Global de Comunicaciones para Móviles*; *Sistema de telefonía celular digital para comunicaciones móviles de segunda generación desarrollado en Europa*)

- **IDE** (del inglés: *Integrated Development Environment*, en español, *Entorno Integrado de Desarrollo*)
- **IEEE-1284** (del inglés: *Standard Signaling Method for a Bi-directional Parallel Peripheral Interface for Personal Computers*, en español, *Estándar del Método de Señalización para una Interfase Paralela Bidireccional Periférica para Computadoras Personales*)
- **IPX** (del inglés: *Internet Payment Exchange*; *IPX* (del inglés: *Internet Payment eXchange*) de Ericsson es un servicio global de administración de pagos y entrega de mensajes para SMS Premium o no Premium, MMS y contenidos Web y Wap.)
- **J2EE** (*Java 2 Enterprise Edition*; La edición empresarial del paquete Java, comprenden un conjunto de especificaciones y funcionalidades orientadas al desarrollo de aplicaciones empresariales.)
- **J2SE** (*Java 2 Standard Edition*; Plataforma de desarrollo, contiene colecciones de APIs utilizada por cualquier programa que utilice Java.)
- **JCA** (del inglés: *Java Connector Architecture*; Define una arquitectura estándar para conectar la plataforma e-business a sistemas empresariales.)
- **JSR-212** (del inglés: *Java Specification Request-212*, Especificación que proporciona un API para trabajar con mensajes SMS y MMS desde aplicaciones J2EEy J2SE)
- **JSR-80** (del inglés: *Java Specification Request-80*; Especificación que define un estándar para el acceso a dispositivos USB utilizando el lenguaje Java.)
- **MM** (del inglés: *Multimedia Message*, en español, *Mensajes Multimedia*; Mensaje que maneja voz, video y datos.)
- **MM1, MM4 (SMTP), MM7:** (Protocolos para el envío y recepción de mensajes MMS; MM1: Define la comunicación entre el MMS Relay/Server y el User Agent (cliente MMS terminal); MM4: Define la comunicación entre los distintos MMSC; MM7: Es la interfaz con los servicios de valor añadido.)
- **MMS** (del inglés: *Multimedia Message Service*, en español, *Servicio de Mensajes Multimedia*; Envío y recepción de un MM.)
- **MMSC** (del inglés: *Multimedia Message Service Center*, en español, *Central de Servicio de Mensajes Multimedia*; Central que gestiona los MMS)
- **MO** (del inglés: *Mobile Originated*; Transmisor)

- **Módem** (Consiste en un dispositivo que se conecta al ordenador ya una línea telefónica y que permite poner en contacto dos ordenadores u otros dispositivos.)
- **MT** (del inglés: *Mobile Terminated; Receptor*)
- **OIS, OISD** (*Open Interface Specification; Protocolos desarrollados por SEMA Group*)
- **Open Source** (Código abierto es el término por el que se conoce al software distribuido y desarrollado en una determinada forma.)
- **OSA Parlay** (del inglés: *Open Service API Parlay; Define un conjunto de API que permite a los proveedores de software, desarrollar aplicaciones que utilizan las funciones actuales y las emergentes de las redes de telecomunicaciones*)
- **PCI** (del inglés: *Protocol Control Information; Define la información de protocolo que añade una entidad a la SDU procedente de la capa superior, constituyendo en conjunto el protocolo PDU.*)
- **PDU** (del inglés: *Protocol Data Unit*)
- **PPG** (del inglés: *Push Proxy Gateway; Pasarela proxy para servicios de notificaciones.*)
- **RS-232** (del inglés: *Recommended Standard-232; Puerto de serie, constituye la interfaz que designa una norma para el intercambio serie de datos binarios.*)
- **RUP** (del inglés: *Rational Unified Process; Metodología para el desarrollo de software. Modelación UML*)
- **SAMS-M** (del inglés: *Server API for Mobile Services-Messaging*)
- **SDU** (del inglés: *Service Data Unit; Paquete de la capa subordinada para cada caso.*)
- **SIM** (del inglés: *Subscriber Identity Module; Pequeña Tarjeta Inteligente que contiene datos del suscriptor.*)
- **SM** (del inglés: *Short Message, en español, Mensajes Cortos; Mensajes cortos de texto hasta 160 caracteres.*)
- **SM-TL** (del inglés: *Short Message – Transfer Layer; Servicio de de tranferencia y entrega de un mensaje cortos, obteniéndose un informe sobre lo ocurrido*)
- **SME** (del inglés: *Short Message Entity, en español, Entidad de Mensajes Cortos; Entidad que puede enviar o recibir mensajes cortos, pudiendo estar localizada en la red fija, una estación móvil, u otro centro de servicio.*)
- **SMIL** (del inglés: *Synchronized Multimedia Integration Language; Lenguaje de diseño, que permite la integración de una colección de objetos multimedia de manera sincronizada.*)

- **SMPP** (del inglés: *Short Message Peer to Peer; Protocolo estándar de telecomunicaciones pensado para el intercambio de mensajes SMS entre equipos que gestionan los mensajes*)
- **SMS** (del inglés: *Short Message Service, en español, Servicio de Mensajería Corta; Envío y recepción de SM*)
- **SMSC** (del inglés: *Short Message Service Center; Central que gestiona los SMS*)
- **Socket** (Un objeto de software utilizado por un cliente para conectarse a un servidor; los componentes básicos incluyen el número de puerto y la dirección de red del host local.)
- **SPI** (del inglés: *Service Provider Interface, en español, Interfaz Proveedores de Servicios; Contiene el entorno donde los programas usan el API SAMS y ejecutan las aplicaciones*).
- **SVN** (*Subversion, herramienta utilizada para el control de versiones*)
- **TPDU** (del inglés: *Transfer Protocol Data Unit; Unidad de datos en la capa de transporte.*)
- **UCP** (del inglés: *Universal Computer Protocol; Protocolo desarrollado por LógicaCMG*)
- **VAS** (del inglés: *Value Added Service, en español, Servicio de Valor Añadido*)
- **VASP** (del inglés: *Value Added Service Provider, en español, Proveedores de Servicio de Valor Añadido*)
- **WAP** (del inglés: *Wireless Application Protocol; Estándar abierto internacional para aplicaciones que utilizan las comunicaciones inalámbricas*)
- **WAP Push, WAP Push SI, WAP Push SL** (*Notificaciones WAP*)
- **WorkBench** (Un conjunto integrado de herramientas que dan soporte a la automatización del proceso completo de desarrollo del sistema informático)
- **WSP** (del inglés: *Wireless Session Protocol, en español, Protocolo Inalámbrico de Sesión; Constituye la capa que se sitúa por debajo de la capa de Aplicación, proporcionando la capacidad necesaria para: Establecer una conexión fiable entre el cliente y el servidor, y liberar esta conexión de una forma ordenada. Intercambiar contenido entre el cliente y el servidor utilizando codificación compacta. Suspende y recupera la sesión.*)
- **XML** (del inglés: *Extensible Markup Language; Lenguaje de descripción de páginas de Internet, diseñado con la intención de reemplazar al estándar actual HTML*).