

**ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL**



**Facultad de Ingeniería en Electricidad y Computación**

“ANÁLISIS E IMPLEMENTACIÓN DE UN SERVIDOR DE PROTOCOLO DE MENSAJERÍA ESCRITA PUNTO A PUNTO (SMPP) VERSIÓN 3.4 EN LINUX QUE INTERACTÚE CON UN CLIENTE SMPP EN EL ENVÍO Y RECEPCIÓN DE MENSAJES CORTOS (SMS's) Y QUE GENERE ARCHIVOS DE REGISTROS DE DATOS DE LLAMADAS, CDRs”

**INFORME DE MATERIA DE GRADUACIÓN**

**PREVIO A LA OBTENCIÓN DEL TÍTULO DE:**

**INGENIERO EN ELECTRÓNICA Y TELECOMUNICACIONES**

Presentado por:

**CÉSAR ALEJANDRO MACÍAS DUARTE**

**HENRRY DANILO VARGAS MONGE**

GUAYAQUIL – ECUADOR

AÑO 2015

## **AGRADECIMIENTO**

Agradezco a Dios, y mi familia por sus bendiciones que me permitieron lograr y cumplir mis metas.

A la Universidad Escuela Superior Politécnica del Litoral por brindarme la oportunidad de estudiar y ser un profesional.

A mis profesores, quienes durante mi etapa de estudios aportaron a la formación de mi carrera universitaria.

A todas las personas que han formado parte de mi vida profesional por su amistad, apoyo y compañía.

**César**

Agradezco a Dios, que ante tantas circunstancias, ha estado iluminándome en todo momento para poder afrontar las adversidades que se me ha presentado en la vida y así poder lograr mis metas.

Mi familia, que amo con todo mi corazón, mi papá Klever y mi mamá Nancy son los pilares fundamentales de este logro. Mis hermanos que siempre han estado ahí y son el reflejo de lo que soy.

Mis amigos y profesores que siempre fueron un apoyo para cualquier situación que se me haya presentado.

La Escuela Superior Politécnica del Litoral por haberme formado como un profesional en todas las áreas, académicas y personales.

**Henry**

## DEDICATORIA

Mi proyecto lo dedico a mi amada esposa **Diana**, por su amor, por creer en mis capacidades, por sus palabras de aliento y su confianza en mí. Por ayudarme a ser perseverante y no dejarme decaer, para seguir adelante y cumplir mis ideales.

A mi Madre, hermanas, tías y primos porque en todo momento me han brindado su cariño incondicional y su apoyo para realizarme profesionalmente.

A mis amigos, quienes sin esperar nada a cambio me compartieron sus conocimientos, experiencias y de alguna manera contribuyeron al logro de mis objetivos.

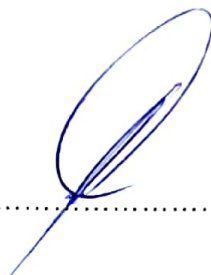
**César**

Dedico este proyecto y mi gran esfuerzo principalmente a mis padres, que son los que me han sabido guiar y gracias a ellos he llegado a ser un profesional. Mi recompensa es esta, el triunfo es de ellos y la alegría es mía.

Mis hermanos, porque siempre son un apoyo y un incentivo al ser yo un ejemplo a seguir.

**Henry**

## TRIBUNAL DE SUSTENTACIÓN



MSc. José Miguel Menéndez

PROFESOR DE LA MATERIA DE GRADUACIÓN



Ing. Jorge Rodríguez

PROFESOR DELEGADO POR LA UNIDAD ACADÉMICA

## DECLARACIÓN EXPRESA

"La responsabilidad del contenido de este Informe, nos corresponde exclusivamente; y el patrimonio intelectual de la misma a la Escuela Superior Politécnica del Litoral"



César Alejandro Macías Duarte



Henry Danilo Vargas Monge

## RESUMEN

El presente proyecto está dirigido para cualquier estudiante o profesor de ingeniería en telecomunicaciones que desee entender el protocolo SMPP de una forma didáctica, detallando los procesos principales, su funcionamiento y mostrando resultados puntuales.

Se detalla el protocolo SMPP desde el punto de vista del servidor, es decir una implementación limitada de una central de servicios de mensajes cortos (SMSC) y para realizar pruebas se interactúa con una entidad externa (ESME).

Los dos primeros capítulos relatan los antecedentes del protocolo SMPP, los objetivos del proyecto, las limitaciones y el marco teórico del protocolo SMPP v3.4 para posteriormente con los conocimientos adquiridos llegar a una implementación del SMSC en una máquina virtual mediante lenguajes de programación que se detallan en el capítulo 3. Por último, en el capítulo 4 se detalla un análisis de los mensajes enviados entre ESME y SMSC observándose los resultados de toda la implementación limitada del SMSC.



## ÍNDICE GENERAL

AGRADECIMIENTO .....	II
DEDICATORIA .....	IV
TRIBUNAL DE SUSTENTACIÓN .....	VI
DECLARACIÓN EXPRESA .....	VII
RESUMEN.....	VIII
ÍNDICE GENERAL .....	IX
ABREVIATURAS Y SIMBOLOGÍA .....	XII
ÍNDICE DE FIGURAS.....	XIV
INTRODUCCIÓN.....	XVII
CAPÍTULO 1.....	1
1 ANTECEDENTES Y JUSTIFICACIÓN.....	1
1.1 Antecedentes de SMPP.....	1
1.2 Objetivo General.....	2
1.3 Objetivo Específico.....	2
1.4 Justificación.....	2
1.5 Alcance.....	4
1.6 Limitaciones .....	4
1.7 Metodología.....	5
CAPÍTULO 2.....	6
2 DESCRIPCIÓN DE SMPP .....	6
2.1 Definición SMPP.....	6

2.1.1 SMSC.....	7
2.1.2 ESME .....	7
2.2 Aplicaciones de SMPP .....	8
2.3 Sesiones SMPP.....	9
2.3.1 Tipos de Sesión ESME.....	9
2.3.2 Iniciando una Sesión SMPP .....	10
2.3.3 Asíncrono.....	13
2.4 Operaciones PDUs de SMPP .....	14
2.4.1 Definiciones de los PDUs de SMPP .....	14
2.5 Formato PDU de SMPP.....	19
2.5.1 Comandos de la Cabecera PDU de SMPP.....	20
2.5.2 Parámetros SMPP Mandatorios – Cuerpo PDU .....	23
2.6 Ejemplos de Sesiones SMPP .....	31
2.7 Máquinas Virtuales y su utilidad para el proyecto .....	32
CAPÍTULO 3.....	34
3 IMPLEMENTACIÓN DE UN SMSC .....	34
3.1 Breve descripción de las herramientas utilizadas .....	34
3.1.1 Servidor Linux Ubuntu Server 14.04.....	35
3.1.2 Lenguaje Perl – Programación del Protocolo SMPP.....	35
3.1.3 Lenguaje PHP –Interfaz Gráfica .....	36
3.1.4 Base de datos MariaDB.....	36
3.1.5 Requisitos y versiones de lenguajes de programación .....	37
3.2 Instalación de la Máquina Virtual VMWare Player .....	37
3.3 Descripción General de la Implementación del SMSC.....	42

3.4 Funcionamiento del SMSC .....	45
CAPÍTULO 4.....	53
4 ANÁLISIS DERESULTADOS.....	53
4.1 Análisis General .....	53
4.1.1 Conexión del ESME hacia el SMSC en modo bind_receiver .....	54
4.1.2 Conexión del ESME hacia el SMSC en modo bind_transmitter .....	54
4.1.3 Desconexión de un ESME con Unbind.....	55
4.2 Pruebas de Errores .....	56
4.3 Detalle de Parámetros de mensajes PDU.....	58
CONCLUSIONES Y RECOMENDACIONES .....	63
BIBLIOGRAFÍA.....	66
ANEXO .....	70

## ABREVIATURAS Y SIMBOLOGÍA

ACK	Acknowledgement (Acuse de recibo)
CCITT	Comité Consultivo Internacional Telegráfico y Telefónico
ESME	External Short Message Entity (Entidad Externa de Mensajes Cortos)
GPS	Global Positioning System (Sistema de Posicionamiento Global)
IP	Internet Protocol (Protocolo de Internet)
ITU-T	International Telecommunication Union–Telecommunication (Unión Internacional de Telecomunicaciones – Telecomunicaciones)
MS	Mobile Station (Estación Móvil)
NACK	Negative Acknowledgement (Acuse de recibo Negativo)
NPI	Numeric Plan Indicator (Indicador del Plan Numérico)
PDU	Protocol Data Unit (Unidad de Datos del Protocolo)
PERL	Practical Extraction and Report Language (Lenguaje Práctico de Extracción y de Informes)
PHP	Personal Home Page (Página Personal)
SMPP	Short Message Peer to Peer (Mensaje Corto Punto a Punto)
SME	Short Messaging Entity (Entidad de Mensajería Corta)
SMS	Short Message Service (Servicio de Mensajería Corta)

SMSC	Short Message Service Center (Centro de Servicios de Mensajes Cortos)
SS7	Signaling System #7 (Sistema de Señalización #7)
TCP	Transmission Control Protocol (Protocolo de Control de Transmisión)
TON	Type Of Number (Tipo de Número)
UIT	Unión Internacional de Telecomunicaciones

## ÍNDICE DE FIGURAS

Figura 2.1. Formato PDU de SMPP [12] [13] .....	19
Figura 2.2. PDUs del campo command_id [6].....	21
Figura 2.3. Comandos de Errores SMPP [6] [15].....	22
Figura 2.4. Parámetros bind [6].....	23
Figura 2.5. Parámetros bind_resp [6].....	23
Figura 2.6. Parámetros Outbind [6].....	24
Figura 2.7. Parámetros submit_sm y delivery_sm [6].....	25
Figura 2.8. Parámetros submit_sm_resp y delivery_sm_resp [6].....	25
Figura 2.9. Parámetros submit_multi [6] .....	26
Figura 2.10. Parámetros submit_multi_resp [6].....	26
Figura 2.11. Parámetros data_sm [6].....	27
Figura 2.12. Parámetros data_sm_resp [6].....	27
Figura 2.13. Parámetros alert_notification [6].....	28
Figura 2.14. Parámetros query_sm [6].....	28
Figura 2.15. Parámetros query_sm_resp [6].....	29
Figura 2.16. Parámetros cancel_sm [6] .....	29
Figura 2.17. Parámetros replace_sm [6] .....	30
Figura 2.18. Ejemplo de una sesión SMPP Transmitter [5] .....	31
Figura 2.19. Ejemplo de una sesión SMPP Receiver [5].....	31
Figura 2.20. Ejemplo de una sesión SMPP Transceiver [5] .....	32
Figura 3.1. Creación de Máquina Virtual VMWare Player .....	38

Figura 3.2. Selección de ISO Ubuntu Server 14.04.1.....	38
Figura 3.3. Datos del usuario administrador Linux .....	39
Figura 3.4. Selección de tamaño del disco virtual .....	39
Figura 3.5. Información de la Máquina Virtual antes de crear .....	40
Figura 3.6. Instalación en curso de la Máquina Virtual.....	40
Figura 3.7. Instalación de herramientas de VMWare .....	41
Figura 3.8. Máquina Virtual instalada con éxito.....	42
Figura 3.9. Ubuntu 14.04 .....	42
Figura 3.10. Máquina Virtual VMware .....	43
Figura 3.11. Descripción General del Proyecto.....	43
Figura 3.12. tb_sh_codes .....	44
Figura 3.13. tb_esme.....	45
Figura 3.14. Diagrama del funcionamiento del SMSC.....	45
Figura 3.15. Proceso Escuchar (Listen) .....	46
Figura 3.16. Código del Proceso Escuchar (Listen) .....	46
Figura 3.17. ESME hizo conexión como transmitter.....	47
Figura 3.18. ESME hizo conexión como receiver o transceiver .....	48
Figura 3.19. Tipos de PDU programados en Perl .....	48
Figura 3.20. Código del manejo de conexión con el ESME.....	49
Figura 3.21. Código del manejo de desconexión .....	50
Figura 3.22. Código del manejo de envío de mensajes al ESME.....	50
Figura 4.1. Conexión de un ESME tipo bind_receiver.....	54

Figura 4.2. Conexión de un ESME tipo bind_transmitter.....	55
Figura 4.3. Desconexión de un ESME .....	56
Figura 4.4. Resumen de mensajes PDU del proyecto.....	56
Figura 4.5. Falla de estado bind (bind failed) .....	57
Figura 4.6. Clave incorrecta (invalid password) .....	57
Figura 4.7. Parámetros bind_transceiver .....	58
Figura 4.8. Parámetros bind_transceiver_resp .....	59
Figura 4.9. Parámetros submit_sm .....	60
Figura 4.10. Parámetros submit_sm_resp .....	60
Figura 4.11. Parámetros deliver_sm .....	61
Figura 4.12. Parámetros deliver_sm_resp .....	61
Figura 4.13. Parámetros enquire_link .....	62
Figura 4.14. Parámetros enquire_link_resp .....	62



## INTRODUCCIÓN

Este proyecto está basado principalmente en la explicación del funcionamiento del protocolo de Mensaje Corto Punto a Punto (Short Message Peer to Peer, SMPP) mediante la implementación de un Centro de Servicios de Mensajes Cortos (Short Message Service Center, SMSC) limitado, es decir, algo didáctico para un mejor entendimiento.

Se requiere de una Entidad Externa de Mensajes Cortos (External Short Message Entity, ESME) o un cliente SMPP para realizar las pruebas de los envíos y recepción de mensajes cortos. Sin embargo, este análisis se enfoca en el servidor SMSC y adicionalmente se ha realizado una simulación de un sencillo ESME para probar la implementación del SMSC.

# **CAPÍTULO 1**

## **1 ANTECEDENTES Y JUSTIFICACIÓN**

### **1.1 Antecedentes de SMPP**

SMPP fue desarrollado con el objetivo de hacer la comunicación sencilla entre entidades o empresas, enviando mensajes cortos y que soporta tecnologías celulares [1] [2].

Muchas personas en la actualidad están familiarizadas con la recepción de mensajes escritos a sus celulares, sean estos promociones o información que alguna empresa les envíe [3]. De estas empresas, se destaca Acision, que en el 2010 fue reconocida como número uno del mundo en servicios de mensajería [4].

## **1.2 Objetivo General**

El objetivo principal es dar una explicación didáctica del uso del protocolo SMPP como un servidor mediante la utilización de una plataforma SMSC limitada.

## **1.3 Objetivo Específico**

Los objetivos específicos son:

- Describir el funcionamiento y operación del protocolo SMPP v3.4 de una forma sencilla para una comprensión rápida de cualquier estudiante o profesor de ingeniería en telecomunicaciones.
- Implementar una plataforma que actúe como Centro de Servicio de Mensajería Corta (SMSC) para la interacción con un ESME o cliente SMPP v3.4.

## **1.4 Justificación**

Hoy en día la mensajería es usada cada vez más y esto se refleja en el hecho de que es una de las grandes fuentes de ingresos para las empresas telefónicas.

**En la actualidad la mensajería sigue siendo la principal fuente de los ingresos de datos para los operadores móviles, e Informa Telecoms&Media prevé que los ingresos por mensajería seguirán creciendo en los próximos cinco años, donde solo con los SMS se generarán 73.000 millones de libras esterlinas (118.300 millones de dólares estadounidenses) para 2014 [4].**

La citación realizada en el párrafo anterior nos da un indicativo acerca de la importancia de este servicio de mensajería, no solo para las operadoras, sino para el usuario que de algún modo por ejemplo tiene la necesidad de estar informado mediante mensajes cortos de texto al suscribirse a una entidad externa ESME.

Por lo tanto, el protocolo SMPP sirve como solución para dar una buena administración del flujo de los mensajes enviados por entidades externas, tales como: proveedores de contenidos, notificaciones de la banca, noticias, entre otros.

Para el estudio del Protocolo SMPP y de un análisis e implementación de un SMSC del presente trabajo, es dar una explicación sencilla del funcionamiento de estos, porque actualmente se utiliza protocolos de mensajería escrita corta como SMPP y resulta ser un ingreso económico muy bueno para las operadoras telefónicas y entidades externas.

## 1.5 Alcance

El alcance de la implementación del SMSC está dado por la explicación del funcionamiento del protocolo SMPP empleándose sobre sistema operativo Ubuntu una aplicación en Lenguaje Práctico de Extracción y de Informes (Practical Extraction and Report Language, PERL) y otra en lenguaje dinámico para Páginas Personales (Personal Home Page, PHP). Ambas aplicaciones están interconectadas mediante una base de datos MariaDB, que es una derivación de MySQL.

## 1.6 Limitaciones

En las limitaciones encontramos principalmente lo siguiente:

- Solo se consideran los siguientes mensajes de Unidad de Datos del Protocolo (Protocol Data Unit, PDU) de SMPP: Transmitter, Receiver, Transceiver, Submit\_sm, Deliver\_sm, Unbind y Enquire\_link.
- Los errores considerados por el ESME son los siguientes: no hay error, clave incorrecta y falla de estado de enlace (bind).
- Para fines de explicación del protocolo SMPP, se considera que todos los SMS recibidos por este protocolo ya han sido tarifados y pueden ser enviados a su destinatario.
- Para la decisión de a que ESME se debe enviar un SMS, se implementó una tabla de rutas dentro de la base de datos.

- Los SMS recibidos por el SMSC se los procesa con la metodología de almacenar y luego enviar.

## **1.7 Metodología**

El estudio realizado en el presente trabajo tiene una investigación descriptiva, puesto que no se modifica lo observado y se ha hecho uso de fuentes de páginas web y también de recursos brindados por el tutor para ampliar los conocimientos del uso del protocolo SMPP y así poder explicarlo de manera fácil mediante una implementación básica de lo que es un SMSC.

Este trabajo tiene un fin didáctico y no busca cambiar ningún enunciado, por tal motivo en el capítulo 2 se detalla teóricamente el protocolo SMPP, en el capítulo 3 se procede a la implementación básica del SMSC con una breve descripción de las herramientas utilizadas para su propósito y por último, los capítulos 4 y 5 relatan los análisis, resultados, conclusiones y recomendaciones de toda la investigación descriptiva de SMPP e implementación SMSC.

## **CAPÍTULO 2**

### **2 DESCRIPCIÓN DE SMPP**

#### **2.1 Definición SMPP**

El protocolo de Mensajes Cortos Punto a Punto (Short Message Peer to Peer) más conocido como SMPP, es un protocolo abierto y estándar de las telecomunicaciones, para la comunicación de mensajes cortos fuera de la red móvil entre entidades, tales como, Entidades Externas de Mensajes Cortos (ESME) que se conectan con el Centro de Servicios de Mensajes Cortos (SMSC). El SMSC es el servidor y el ESME el cliente [5] [6].

Para que exista la comunicación, se la realiza mediante una conexión del Protocolo de Control de Transmisión/Protocolo de Internet (Transmission

Control Protocol/Internet Protocol, TCP/IP) o similar, en cual el protocolo SMPP hace la función de intercambiar mensajes cortos entre ESME y SMSC con los llamados PDU. Cada petición PDU que se envía de una entidad tiene su respectiva respuesta PDU [5] [6].

### **2.1.1 SMSC**

El SMSC (Short Message Service Center) es la central de mensajería corta de una red celular que realiza la organización de todos los mensajes escritos que se envían por parte de los usuarios o alguna entidad externa ESME. Este SMSC recibe los mensajes y se encarga de enviarlos a su destinatario siempre que esté disponible. Por lo tanto, un mensaje queda almacenado en el SMSC por un cierto tiempo, el cual es configurado por el administrador o también el usuario puede dar este límite de tiempo si así lo desea [7].

### **2.1.2 ESME**

El ESME es una entidad de red externa que puede enviar y recibir mensajes, pero primero el ESME debe conectarse a un SMSC con una conexión TCP/IP o similar y mediante un protocolo que pueda



intercambiar mensajes (petición/respuesta) que puede ser SMPP [8].

Un ESME es considerado también una entidad no móvil. Unos ejemplos de ESME, son las empresas que envían anuncios, concursos de trivia, votaciones, entre otros. Un usuario probablemente reconocería que un ESME le ha enviado un mensaje, cuando el número de procedencia no es un número celular habitual o el mensaje escrito no es deseado.

## **2.2 Aplicaciones de SMPP**

Con SMPP se puede tener un sin número de aplicaciones de mensajería. Algunos ejemplos de estas se nombran a continuación [5] [6]:

- Alerta de mensajes de voz.
- Servicios de posicionamiento o localización (GPS).
- Mensajería Instantánea.
- Servicios de Información.
- Mensajes de difusión (Broadcast).

## **2.3 Sesiones SMPP**

La sesión se establece entre el ESME y el SMSC. Esta sesión es iniciada normalmente por el ESME, quien emite una solicitud de enlace (Bind Request) para abrir una sesión SMPP. La conexión se trabaja en la capa de aplicación TCP/IP o similar [9].

### **2.3.1 Tipos de Sesión ESME**

Un ESME envía y recibe mensajes estableciendo dos conexiones de redes que pueden ser TCP/IP o similares y dos sesiones de SMPP que son el Transmitter y el Receiver. En la versión 3.4 del protocolo SMPP, un ESME puede establecer una sesión SMPP Transceiver sobre una sola conexión de red [10].

En resumen hay tres formas en que el ESME inicia sesión:

- Transmitter.
- Receiver.
- Transceiver.

#### **Transmitter**

El inicio de sesión como Transmitter, hace que el ESME pueda enviar mensajes cortos al SMSC.

### **Receiver**

El inicio de sesión como Receiver, hace que el ESME pueda recibir mensajes cortos del SMSC.

### **Transceiver**

El inicio de sesión como Transceiver, hace que el ESME pueda enviar y recibir mensajes cortos del SMSC. La diferencia con respecto a los otros inicios de sesión donde para un Transmitter hacen una conexión y para un Receiver hacen otra conexión, con Transceiver se hace ambas en una sola conexión.

### **2.3.2 Iniciando una Sesión SMPP**

Como ya se mencionó, para que exista una sesión SMPP, el ESME normalmente debe conectarse con el SMSC. Este ESME puede enviar una serie de peticiones al SMSC que se mantiene escuchando las conexiones por uno o más puertos de TCP/IP o similar (El puerto 2775 normalmente se usa, pero esto puede variar dependiendo del vendedor del SMSC), y el SMSC enviará al ESME las respuestas adecuadas a cada una de estas peticiones. El SMSC

también puede hacer peticiones al ESME, y este también responderá de forma adecuada [9].

### **Estados de una Sesión SMPP**

Estos estados son [9]:

- Open.
- Bound\_TX.
- Bound\_RX.
- Bound\_TRX.
- Closed.
- Outbind.

#### **Open**

En este estado el ESME establece una conexión de red TCP/IP o similar con el SMSC, pero aún no se realiza la solicitud de enlace (bind\_request).

Una vez realizada la conexión, vienen los estados Bound según el tipo de sesión del ESME.

### **Bound\_TX**

El ESME ha iniciado sesión como un Transmitter. ESME envió una solicitud (bind\_request) para unirse (bind\_transmitter PDU fue enviado al SMSC) y recibió la respuesta a esta solicitud (bind\_transmitter\_resp PDU fue recibido del SMSC).

### **Bound\_RX**

El ESME ha iniciado sesión como un Receiver. ESME envió una solicitud (bind\_request) para unirse (bind\_receiver PDU fue enviado al SMSC) y recibió la respuesta a esta solicitud (bind\_receiver\_resp PDU fue recibido del SMSC).

### **Bound\_TRX**

El ESME ha iniciado sesión como un Transceiver. ESME envió una solicitud (bind\_request) para unirse (bind\_transceiver PDU fue enviado al SMSC) y recibió la respuesta a esta solicitud (bind\_transceiver\_resp PDU fue recibido del SMSC).

### **Closed**

Este estado se da cuando se ha solicitado el cierre de conexión de red entre el ESME el SMSC.

### **Outbind**

Se ha mencionado que normalmente el ESME es quien inicia sesión, pero en este estado lo ha realizado el SMSC (recordando que el SMSC también puede iniciar una sesión SMPP) enviando un mensaje outbind al ESME. El objetivo que se persigue es hacer que el ESME tenga una conexión de red como un Receiver (bind\_receiver) para que el SMSC pueda entregar los mensajes pendientes que tiene hacia el ESME.

### **2.3.3 Asíncrono**

El protocolo SMPP es asíncrono. Esto ayuda a las entidades externas ESME o al SMSC enviar varias solicitudes al mismo tiempo. Existe un campo llamado sequence\_number en la cabecera PDU de SMPP (ver en subcapítulo 2.5 sobre formato PDU de SMPP), el cual lleva un número de secuencia para poder correlacionar cada solicitud con su respectiva respuesta, logrando así un protocolo asíncrono [5].

## **2.4 Operaciones PDUs de SMPP**

Las operaciones son las siguientes [11]:

- PDU Bind.
- PDU OutBind.
- PDU Unbind.
- PDU Generic\_Nack.
- PDU Enquire\_link.
- PDU Submit\_sm.
- PDU Submit\_Multi.
- PDU Deliver\_sm.
- PDU Data\_sm.
- PDU Alert\_Notification.
- PDU Cancel\_sm.
- PDU Query\_sm.
- PDU Replace\_sm.

### **2.4.1 Definiciones de los PDUs de SMPP**

Se presentan a continuación la definición de cada PDU de SMPP.

## **PDU Bind**

PDU Bind hace el registro de una instancia entre el ESME con el SMSC y así establecer una sesión SMPP.

El PDU Bind se divide en tres PDU según el tipo de inicio de sesión del ESME:

- bind\_transmitter.
- bind\_receiver.
- bind\_transceiver.

## **PDU OutBind**

Con esta operación el SMSC informa al ESME para que inicie sesión como bind\_receiver en caso de que el SMSC tenga mensajes pendientes que entregar al ESME.

## **PDU Generic\_Nack**

Se origina un NACK a un mensaje PDU de SMPP que ha sido enviado con una cabecera inválida. Este PDU generic\_nack se produce cuando el campo command\_length es demasiado corto o



tiene demasiada longitud, o cuando el campo `command_id` es desconocido (ver en subcapítulo 2.5 sobre formato PDU de SMPP).

### **PDU UnBind**

PDU UnBind cierra la sesión SMPP y la conexión TCP o similar. Esta solicitud de desenlace UnBind puede ser enviado por el ESME o SMSC.

### **PDU Enquire\_link**

Esta operación realiza una revisión de la conexión a nivel de aplicación entre el ESME y SMSC, con el objetivo de verificar si existe comunicación entre ellos. La entidad que recibe esta petición, ESME o SMSC, responderá con un `enquire_link_resp`.

### **PDU Submit\_sm**

PDU Submit\_sm es el mensaje SM que envía el ESME hacia el SMSC. Su respectiva respuesta es el PDU `submit_sm_resp` indicando que el mensaje ha sido recibido por el SMSC, quien posteriormente se encargará de enviar este SM a su destino.

Este PDU Submit\_sm es usado cuando el ESME inicia sesión como un Transmitter o un Transceiver.

### **PDU Submit\_Multi**

La operación PDU submit\_multi envía un mensaje SMPP a múltiples destinatarios.

### **PDU Deliver\_sm**

PDU Deliver\_sm es el mensaje SM que envía el SMSC hacia el ESME. Su respectiva respuesta es el PDU deliver\_sm\_resp indicando que el mensaje ha sido recibido por el ESME, el destino final, ya que normalmente este tipo de SM tiene como origen en una terminal o estación móvil y es el SMSC quien direcciona estos mensajes. Se lo utiliza cuando el ESME inicia sesión como un Receiver o un Transceiver.

### **PDU Data\_sm**

Sirve para transferir datos entre ambas entidades SMSC y ESME.

Este PDU fue añadido para el uso de aplicaciones interactivas. Es una alternativa para los PDUs submit\_sm y delivery\_sm.

### **PDU Alert\_Notification**

Este PDU es enviado por el SMSC al ESME cuando se ha detectado que un suscriptor móvil en particular está disponible y una bandera pendiente de entregar ha sido establecido por el suscriptor de un PDU data\_sm anterior.

### **PDU Cancel\_sm**

Sirve para cancelar uno o más mensajes que hayan sido enviados anteriormente (aquellos mensajes enviados con el PDU submit\_sm) por el ESME hacia el SMSC. Es el ESME quien emite este PDU Cancel\_sm.

### **PDU Query\_sm**

Utilizado para consultar el estado de un mensaje que ha sido enviado anteriormente por el ESME hacia el SMSC. Es el ESME quien emite este PDU Query\_sm.

## PDU Replace\_sm

Reemplaza un mensaje que ha sido enviado anteriormente por el ESME hacia el SMSC. Es el ESME quien emite este PDU Replace\_sm.

### 2.5 Formato PDU de SMPP

A continuación, se describe el formato PDU de SMPP mostrado en la Figura 2.1:

Cabecera PDU (Mandatorio)				Cuerpo PDU	
command length	command id	command status	sequence number	Parámetros mandatorios	Parámetros opcionales

Figura 2.1. Formato PDU de SMPP [12] [13]

La cabecera PDU de SMPP contiene 4 campos, los cuales tienen los parámetros mandatorios que deben estar siempre presentes en cada PDU de SMPP. El cuerpo PDU de SMPP es una parte opcional del formato.

Cada campo de la cabecera PDU de SMPP tiene de longitud 4 octetos (4 bytes o 32 bits). El cuerpo PDU de SMPP tiene longitud variable dependiendo lo que indique el campo command\_id, ya que este campo indica el tipo de operación PDU de SMPP (bind, submit, unbind, entre otros) y por lo tanto los parámetros varían.

En esta sección se mencionan solo los parámetros mandatorios del cuerpo PDU de SMPP. Los parámetros opcionales con más detalle en SMPP Developers Forum [6].

### **2.5.1 Comandos de la Cabecera PDU de SMPP**

A continuación, se describen los comandos o campos de la cabecera PDU de SMPP:

#### **Command\_length**

Este campo indica la longitud en octetos de todo el mensaje SMPP (cabecera y cuerpo), incluyéndose a sí mismo. Debe ser mayor o igual que 16 octetos, ya que los 4 campos de la cabecera deben estar presentes siempre en el mensaje SMPP y suman 16 octetos [14].

Conociendo la longitud en octetos del mensaje SMPP dado por `command_length`, si se desea conocer la longitud en octetos del cuerpo PDU de SMPP, se la puede deducir de la siguiente forma:

$$\text{Longitud cuerpo} = \text{Longitud mensaje SMPP} - 16 \text{ octetos} \quad (2.1)$$

## Command\_id

Indica el tipo de operación PDU de SMPP, puede ser un PDU bind, submit, unbind, entre otros. El listado de los posibles valores de este campo se muestra en la Figura 2.2.

Campo command_id	valor en hexadecimal
generic_nack	0x80000000
bind_receiver	0x00000001
bind_receiver_resp	0x80000001
bind_transmitter	0x00000002
bind_transmitter_resp	0x80000002
query_sm	0x00000003
query_sm_resp	0x80000003
submit_sm	0x00000004
submit_sm_resp	0x80000004
deliver_sm	0x00000005
deliver_sm_resp	0x80000005
unbind	0x00000006
unbind_resp	0x80000006
replace_sm	0x00000007
replace_sm_resp	0x80000007
cancel_sm	0x00000008
cancel_sm_resp	0x80000008
bind_transceiver	0x00000009
bind_transceiver_resp	0x80000009
outbind	0x0000000B
enquire_link	0x00000015
enquire_link_resp	0x80000015
submit_multi	0x00000021
submit_multi_resp	0x80000021

Figura 2.2. PDUs del campo command\_id [6]

## Command\_status

Indica si una solicitud PDU de SMPP es exitosa o con algún error.

Los tipos de errores se detallan en la Figura 2.3.

Tipo de Error	Valor en Hex	Descripción
ESME_ROK	0x00000000	No hay error
ESME_RINVMSGLEN	0x00000001	Longitud del mensaje inválido
ESME_RINVCMDLEN	0x00000002	Command Length es inválido
ESME_RINVCMDID	0x00000003	Command ID es inválido
ESME_RINVBNDSTS	0x00000004	Bind incorrecto por un comando dado
ESME_RALYBND	0x00000005	ESME ya en estado Bound
ESME_RSYSERR	0x00000008	Error del sistema
ESME_RINVSRCADR	0x0000000A	Dirección de origen inválido
ESME_RINVDSTADR	0x0000000B	Dirección de destino inválido
ESME_RINVMSGID	0x0000000C	Message ID es inválido
ESME_RBINDFAIL	0x0000000D	Fracaso de Bind
ESME_RINVPASWD	0x0000000E	Clave inválida
ESME_RINVSYID	0x0000000F	System ID es inválido
ESME_RCANCELFAIL	0x00000011	Fracaso de Cancel SM
ESME_RREPLACEFAIL	0x00000013	Fracaso de Replace SM
ESME_RINVSRCTON	0x00000048	Dirección de origen TON inválido
ESME_RINVSRCNPI	0x00000049	Dirección de origen NPI inválido
ESME_RINVDSTTON	0x00000050	Dirección de destino TON inválido
ESME_RINVDSTNPI	0x00000051	Dirección de destino NPI inválido
ESME_RQUERYFAIL	0x00000067	Fracaso de query sm

Figura 2.3. Comandos de Errores SMPP [6] [15]

### Sequence\_number

Recordando que el protocolo SMPP es asíncrono, con este campo `sequence_number` se tiene el número de secuencia de cada solicitud y respuesta PDU de SMPP.

Si una solicitud tiene como número de secuencia un valor  $n$ , la respuesta será con el mismo número de secuencia de la solicitud realizada.

## 2.5.2 Parámetros SMPP Mandatorios – Cuerpo PDU

A continuación, se describen los parámetros del cuerpo para cada PDU de SMPP:

### Bind

Los parámetros bind se describen en la Figura 2.4.

Cuerpo PDU SMPP - Parámetros bind		
Nombre del Parámetro	Tamaño en octetos	Descripción
system_id	Varía, máx. 16	Identifica al ESME haciendo bind ya sea como transmitter, receiver o transceiver
password	Varía, máx. 9	SMSC usa una clave por razones de seguridad y así autentica al ESME que solicita una operación bind
system_type	Varía, 13	Tipo de sistema del ESME haciendo bind
interface_version	1	Indica la versión del protocolo SMPP que tiene el ESME
addr_ton	1	Indica el tipo de número TON de la dirección del ESME. Si no lo conoce, envía NULL
addr_npi	1	Indica que tipo de plan es el número NPI para la dirección del ESME. Si no lo conoce, envía NULL
address_range	Varía, máx. 41	Dirección del ESME. Si no lo conoce, envía NULL

Figura 2.4. Parámetros bind [6]

Los parámetros en respuesta a bind se describen en la Figura 2.5.

Cuerpo PDU SMPP - Parámetros bind_resp		
Nombre del Parámetro	Tamaño en octetos	Descripción
system_id	Varía, máx. 16	Identificador SMSC. Da una identificación del SMSC al ESME

Figura 2.5. Parámetros bind\_resp [6]



## Outbind

Los parámetros Outbind se describen en la Figura 2.6.

Cuerpo PDU SMPP - Parámetros Outbind		
Nombre del Parámetro	Tamaño en octetos	Descripción
system_id	Varía, máx. 16	Identificador SMSC. Da una identificación del SMSC al ESME
password	Varía, máx. 9	La contraseña es usada por el ESME por razones de seguridad para autenticar al SMSC que está enviando un outbind

Figura 2.6. Parámetros Outbind [6]

## Unbind

Tanto la operación unbind como el unbind\_resp, no tiene cuerpo PDU de SMPP, solo contiene la cabecera PDU de SMPP. En el campo command\_id se especifica el tipo de operación PDU: unbind o unbind\_resp.

## Generic\_Nack

No tiene cuerpo PDU de SMPP, solo contiene la cabecera PDU de SMPP.

## Submit\_sm y Delivery\_sm

Los parámetros Submit\_sm y Delivery\_sm se describen en la Figura 2.7.

Cuerpo PDU SMPP - Parámetros submit_sm y delivery_sm		
Nombre del Parámetro	Tamaño en octetos	Descripción
service_type	Varía máx. 6	Identifica el servicio de aplicación SMS asociado con el mensaje. Por defecto se pone en NULL en la configuración SMSC.
source_addr_ton	1	TON para dirección origen. Si no se conoce se pone en NULL.
source_addr_npi	1	NPI para dirección origen. Si no se conoce se pone en NULL.
source_addr	Varía máx. 21	Dirección SME quien origina ese mensaje. Si no se conoce se pone en NULL.
dest_addr_ton	1	TON para destino.
dest_addr_npi	1	NPI para destino.
destination_addr	Varía máx. 21	Dirección destino del mensaje corto.
.	.	.
.	.	.
.	.	.
protocol_id	1	Identificador de protocolo.
sm_length	1	Longitud en octetos de short_message
short_message	Varía de 0 - 254	Se permite hasta 254 octetos de mensaje corto de datos del usuario.

Figura 2.7. Parámetros submit\_sm y delivery\_sm [6]

Los parámetros en respuesta a submit\_sm y delivery\_sm se describen en la Figura 2.8.

Cuerpo PDU SMPP - Parámetros submit_sm_resp y delivery_sm_resp		
Nombre del Parámetro	Tamaño en octetos	Descripción
message_id	Máx. 65 en submit y solo 1 con delivery.	Este campo contiene el message_id SMSC del mensaje enviado. Este campo no es usado por delivery_sm y se debe poner en NULL en tal caso.

Figura 2.8. Parámetros submit\_sm\_resp y delivery\_sm\_resp [6]

## Submit\_Multi

Los parámetros Submit\_Multi se describen en la Figura 2.9.

Cuerpo PDU SMPP - Parámetros submit_multi		
Nombre del Parámetro	Tamaño en octetos	Descripción
service_type	Máx. 6	Identifica el servicio de aplicación SMS asociado con el mensaje. Por defecto se pone en NULL en la configuración SMSC.
source_addr_ton	1	TON para dirección origen. Si no se conoce se pone en NULL.
source_addr_npi	1	NPI para dirección origen. Si no se conoce se pone en NULL.
source_addr	Máx. 21	Dirección SME quien origina ese mensaje. Si no se conoce se pone en NULL.
number_of_dests	1	Números de direcciones destino. Indica el número de estructura de dest_address que se van a seguir.
dest_address	2-24	Contiene una o más (number_of_dests) direcciones SME y/o una lista de distribución.
.	.	.
protocol_id	1	Identificador de protocolo.
sm_length	1	Longitud en octetos de short_message
short_message	Varía de 0 - 254	Se permite hasta 254 octetos de mensaje corto de datos del usuario.

Figura 2.9. Parámetros submit\_multi [6]

Los parámetros en respuesta a submit\_multi se describen en la Figura 2.10.

Cuerpo PDU SMPP - Parámetros submit_multi_resp		
Nombre del Parámetro	Tamaño en octetos	Descripción
message_id	Máx. 65	Es el mensaje identificador SMSC del mensaje enviado.
no_unsuccess	1	Es el número de mensajes a direcciones SME destinos que fueron enviados sin éxito al SMSC.
unsuccess_sme	7-27	Contiene una o más (no_unsuccess) direcciones ESME o lista de distribución cual envió fue sin éxito.

Figura 2.10. Parámetros submit\_multi\_resp [6]

## Data\_sm

Los parámetros Data\_sm se describen en la Figura 2.11.

Cuerpo PDU SMPP - Parámetros data_sm		
Nombre del Parámetro	Tamaño en octetos	Descripción
service_type	Máx. 6	Identifica el servicio de aplicación SMS asociado con el mensaje.
source_addr_ton	1	TON para dirección origen. Si no se conoce se pone en 0x00.
source_addr_npi	1	NPI para dirección origen. Si no se conoce se pone en 0x00.
source_addr	Máx. 65	Dirección SME quien originó ese mensaje.
dest_addr_ton	1	TON para destino.
dest_addr_npi	1	NPI para destino.
destination_addr	Máx. 65	Dirección destino del mensaje corto.
esm_class	1	Indica el modo de mensaje y tipo de mensaje.
registered_delivery	1	Indica si un SMSC delivery receipt o un SME acknowledgement es requerido.
data_coding	1	Define el esquema de codificación de los datos de carga útil.

Figura 2.11. Parámetros data\_sm [6]

Los parámetros en respuesta a data\_sm se describen en la Figura 2.12.

Cuerpo PDU SMPP - Parámetros data_sm_resp		
Nombre del Parámetro	Tamaño en octetos	Descripción
message_id	Máx. 65	Este campo contiene el message_id SMSC del mensaje corto.

Figura 2.12. Parámetros data\_sm\_resp [6]

## Alert\_Notification

Los parámetros Alert\_Notification se describen en la Figura 2.13.

Cuerpo PDU SMPP – alert_notification		
Nombre del Parámetro	Tamaño en octetos	Descripción
source_addr_ton	1	TON para la MS que cambió a disponible. Si no se conoce se pone en NULL.
source_addr_npi	1	NPI para la MS que cambió a disponible. Si no se conoce se pone en NULL.
source_addr	Máx. 65	Dirección de la MS que cambió a disponible.
esme_addr_ton	1	TON para la dirección de destino que solicitó una alerta sobre una MS en particular que cambió a disponible. Si no se conoce se pone en NULL.
esme_addr_npi	1	NPI para la dirección de destino que solicitó una alerta sobre una MS en particular que cambió a disponible. Si no se conoce se pone en NULL.
esme_addr	Máx. 65	Dirección del ESME que solicitó una alerta sobre una MS en particular que cambió a disponible.

Figura 2.13. Parámetros alert\_notification [6]

Nota: alert\_notification no tiene una respuesta PDU.

## Query\_sm

Los parámetros Query\_sm se describen en la Figura 2.14.

Cuerpo PDU SMPP - Parámetros query_sm		
Nombre del Parámetro	Tamaño en octetos	Descripción
message_id	Máx. 65	Identificación del mensaje que se va a consultar su estado
source_addr_ton	1	TON de mensaje originador. Es utilizado para fines de verificación, Si no se conoce se establece en NULL.
source_addr_npi	1	NPI de mensaje originador. Es utilizado para fines de verificación, Si no se conoce se establece en NULL.
source_addr	Máx. 21	Dirección de mensaje originador. Es utilizado para fines de verificación, Si no se conoce se establece en NULL.

Figura 2.14. Parámetros query\_sm [6]

Los parámetros en respuesta a query\_sm se describen en la Figura 2.15.

Cuerpo PDU SMPP - Parámetros query_sm_resp		
Nombre del Parámetro	Tamaño en octetos	Descripción
message_id	Máx. 65	Identificación del mensaje que se va a consultar su estado.
final_date	1 o 17	Es la fecha y hora cuando el mensaje requerido llegó a un estado final.
message_state	1	Especifica el estado del mensaje corto requerido.
error_code	1	Código de error de la red que define la razón de la falta de mensaje de entrega.

Figura 2.15. Parámetros query\_sm\_resp [6]

## Cancel\_sm

Los parámetros Cancel\_sm se describen en la Figura 2.16.

Cuerpo PDU SMPP - Parámetros cancel_sm		
Nombre del Parámetro	Tamaño en octetos	Descripción
service_type	Máx. 6	Identifica el servicio de aplicación SMS si la cancelación de un grupo de aplicaciones SMS es deseada, de lo contrario se establece en NULL.
message_id	Máx. 65	Identificación del mensaje a ser cancelado. Se establece en NULL si se cancela un grupo de mensajes.
source_addr_ton	1	TON de mensaje origen. Si no se conoce se pone en NULL.
source_addr_npi	1	NPI de mensaje origen. Si no se conoce se pone en NULL.
source_addr	Máx. 21	Dirección de mensaje origen a ser cancelado.
dest_addr_ton	1	TON de mensaje destino SME a ser cancelado. Puede ser NULL si se proporciona el message_id.
dest_addr_npi	1	NPI de mensaje destino SME a ser cancelado. Puede ser NULL si se proporciona el message_id.
destination_addr	Máx. 21	Dirección de mensaje destino a ser cancelado. Puede ser NULL si se proporciona el message_id.

Figura 2.16. Parámetros cancel\_sm [6]

Nota: Cancel\_sm tiene respuesta Cancel\_sm\_resp, pero la respuesta no tiene cuerpo PDU de SMPP.

## Replace\_sm

Los parámetros Replace\_sm se describen en la Figura 2.17.

Cuerpo PDU SMPP – Replace_sm		
Nombre del Parámetro	Tamaño en octetos	Descripción
message_id	Max. 65	Identificación del mensaje que va a ser reemplazado
source_addr_ton	1	TON para mensaje origen. Si no se conoce se pone en NULL.
source_addr_npi	1	NPI para mensaje origen. Si no se conoce se pone en NULL.
source_addr	1 - 21	Dirección del mensaje origen que será reemplazado.
schedule_delivery_time	1 o 17	Nuevo calendario de entrega para el mensaje corto. Se establece en NULL si la actualización del mensaje original no es deseada.
validity_period	1 o 17	Nuevo tiempo de expiración para el mensaje corto. Se establece en NULL si la actualización del mensaje original no requiere de tiempo de expiración.
registered_delivery	1	Nuevo registro de configuración del parámetro de entrega.
sm_default_msg_id	1	Nuevo pre-definido del mensaje identificador.
sm_length	1	Nuevo longitud en octetos de short_message.
short_message	Varía de 0 - 254	Nuevo mensaje corto que reemplaza al mensaje existente.

Figura 2.17. Parámetros replace\_sm [6]

Nota: Replace\_sm tiene respuesta Replace\_sm\_resp, pero solo tiene cabecera PDU de SMPP.

## Enquire\_link

Tanto la operación enquire\_link como el enquire\_link\_resp, no tiene cuerpo PDU de SMPP, solo contiene la cabecera PDU de SMPP.

En el campo command\_id se especifica el tipo de operación PDU: enquire\_link o enquire\_link\_resp.

## 2.6 Ejemplos de Sesiones SMPP

El inicio de sesión como Transmitter se describe en la Figura 2.18.

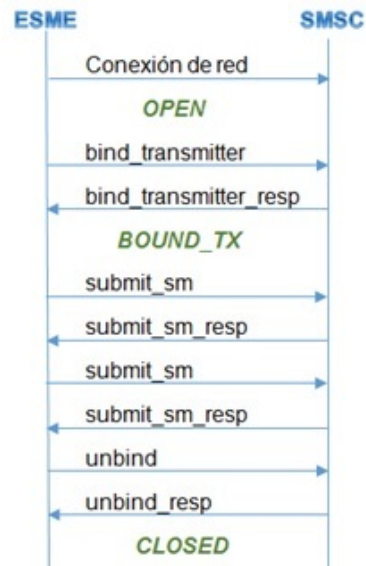


Figura 2.18. Ejemplo de una sesión SMPP Transmitter [5]

El inicio de sesión como Receiver se describe en la Figura 2.19.

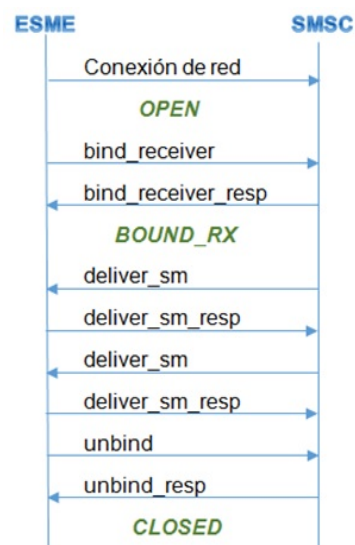


Figura 2.19. Ejemplo de una sesión SMPP Receiver [5]



El inicio de sesión como Transceiver se describe en la Figura 2.20.

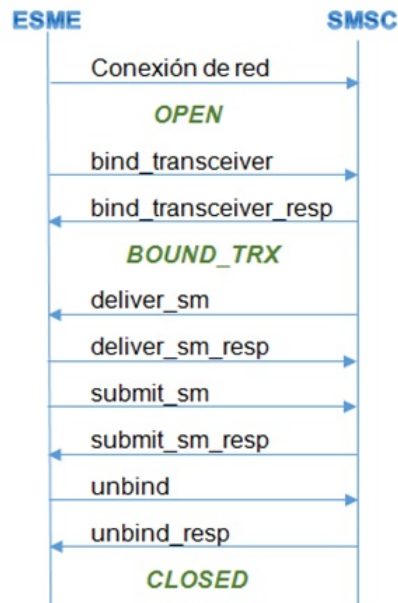


Figura 2.20. Ejemplo de una sesión SMPP Transceiver [5]

## 2.7 Máquinas Virtuales y su utilidad para el proyecto

La máquina virtual es un programa que se instala en cualquier computadora y emula características de una computadora real. La persona que hace la instalación, le da características según su conveniencia a la máquina virtual. Se puede tener más de una máquina virtual en una computadora y son compatibles con Windows y Linux [18].

Existen algunos programas para crear máquinas virtuales, entre ellos, VMWare. Este permite crear y configurar máquinas con diferentes sistemas operativos. VMWare es de utilidad para este proyecto debido a

la necesidad de tener otras PCs para hacer pruebas de los mensajes PDU de SMPP. Esto se logra con una conexión de una red entre la máquina virtual y PC. La versión gratuita es VMWare Player y se la puede descargar de la página web [www.vmware.com](http://www.vmware.com).

## **CAPÍTULO 3**

### **3 IMPLEMENTACIÓN DE UN SMSC**

#### **3.1 Breve descripción de las herramientas utilizadas**

Se requiere un ambiente de programación sobre un servidor Ubuntu 14.04. Este servidor es una máquina virtual instalada en una computadora.

El SMSC se implementó como un servicio (demonio) el cual realiza las funciones por medio del protocolo SMPP. Adicionalmente, se implementó una aplicación para la visualización de los mensajes SMPP y su administración. Ambas aplicaciones interactúan con la misma base de datos.

Los lenguajes de programación utilizados son: Lenguaje Práctico de Extracción y de Informes (Practical Extraction and Report Language, Perl) para la implementación del SMSC y el lenguaje dinámico para Páginas Personales (Personal Home Page, PHP) para la aplicación web que transmite y recibe los mensajes PDU. La base de datos utilizada es MariaDB.

### **3.1.1 Servidor Linux Ubuntu Server 14.04**

Este sistema operativo Ubuntu esta basado en Linux y es un software de código abierto. La versión 14.04 fue liberada en abril del 2014 y tiene soporte de escritorio y servidor [16].

Para la implementación de este proyecto, se puede usar cualquier distribución y versión de Linux, siempre que sea compatible con las versiones de los lenguajes de programación y sus librerías usadas.

### **3.1.2 Lenguaje Perl – Programación del Protocolo SMPP**

Perl es un lenguaje de interpretación basado en el lenguaje C, que tiene un alto desarrollo desde 1987. Además, consume menos recursos que otros de su tipo y tiene un amplio desarrollo de librerías realizados por su comunidad.

Se usa el lenguaje Perl para la implementación del demonio que realiza las funciones de un SMSC por medio del protocolo SMPP.

### **3.1.3 Lenguaje PHP –Interfaz Gráfica**

PHP es un lenguaje de desarrollo web con ejecución del código en el servidor y envío de la página web resultante. Fue creado en 1995 y tiene una gran aceptación a nivel mundial, y es usado en sitios como Facebook.

Se usa el lenguaje PHP para la implementación de la aplicación web, donde se visualiza los mensajes PDU de SMPP, y se puede administrar algunas funciones del servicio SMSC como:

- Desconectar un ESME conectado.
- Visualizar, modificar y crear los datos de conexión de los ESME.
- Manipular la tabla de rutas, que determina el ESME responsable de código corto de SMS.

### **3.1.4 Base de datos MariaDB**

La base de datos MariaDB es una continuidad en código abierto de la base de datos Mysql que fue adquirida por Oracle.

Las funciones de la base de datos son:

- Registrar los mensajes de interacción del SMSC con los diferentes ESME.
- Maneja el estado de las conexiones de los ESME con el SMSC.
- Almacena la tabla de rutas, que se usa para decidir a que ESME se debe enviar un SMS.

### **3.1.5 Requisitos y versiones de lenguajes de programación**

Para poder realizar este proyecto, se tienen los siguientes requisitos:

- Lenguaje Perl; versión 5.18 o superior.
- Lenguaje PHP; versión 5.5.
- Librerías de Perl: Net::SMPP 1.19 y DBI.
- Librerías de PHP: MySQL.

## **3.2 Instalación de la Máquina Virtual VMWare Player**

La instalación de la máquina virtual se detalla a continuación:

- Para la creación de la máquina virtual se usó la plataforma de virtualización gratuita VMWare Player 7.1 x64bit. Para crear una

nueva máquina virtual se presiona sobre la opción de *Create New Virtual Machine* como se describe en la Figura 3.1.

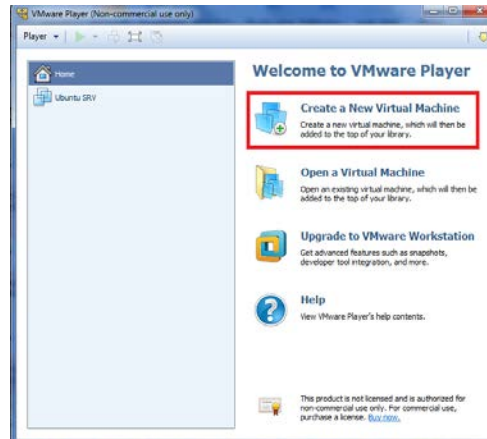


Figura 3.1. Creación de Máquina Virtual VMWare Player

- Ahí aparecerá una nueva ventana donde se debe indicar la ubicación del ISO del sistema a instalar (Figura 3.2), para este proyecto se usó *ubuntu-14.04.1-server-amd64.iso*.

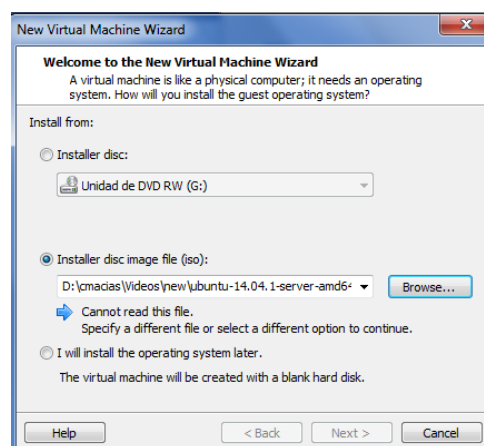


Figura 3.2. Selección de ISO Ubuntu Server 14.04.1

- Como es una versión soportada por VMWare, automáticamente se inicia la instalación fácil. Donde nos solicitan los datos del usuario administrador antes de iniciar la instalación, Figura 3.3.

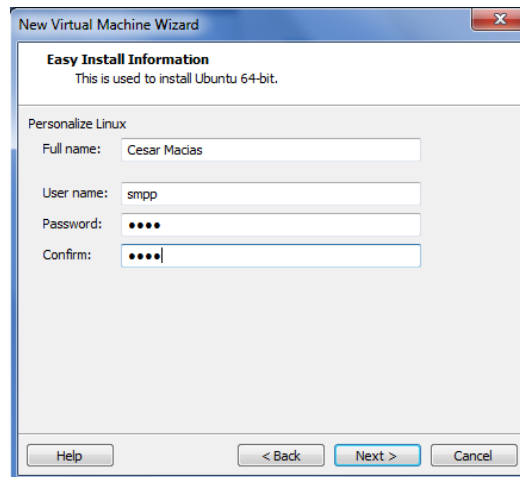


Figura 3.3. Datos del usuario administrador Linux

- El último paso para la creación de la máquina, es definir el tamaño del disco y como se describe en la Figura 3.4, se seleccionó un solo archivo de 8GB que es suficiente para el proyecto.

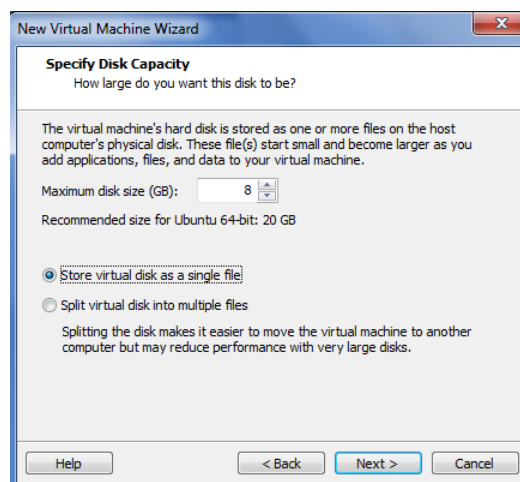


Figura 3.4. Selección de tamaño del disco virtual



- Antes de iniciar la instalación, se muestra la información de la máquina a crear, Figura 3.5.

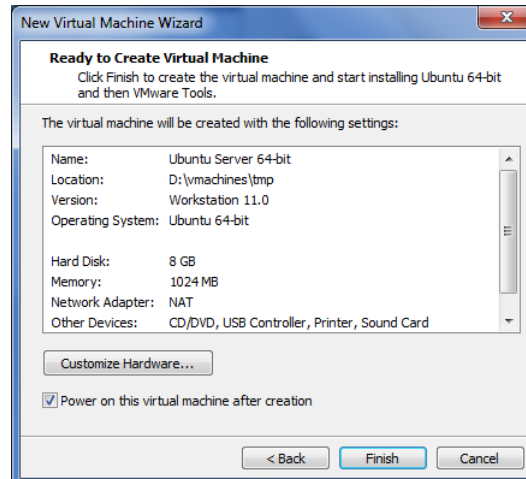


Figura 3.5. Información de la Máquina Virtual antes de crear

- La instalación se inicia automáticamente sin necesidad de intervención del usuario (Figura 3.6), ya que es administrada por VMWare directamente. Se debe esperar que la instalación termine.

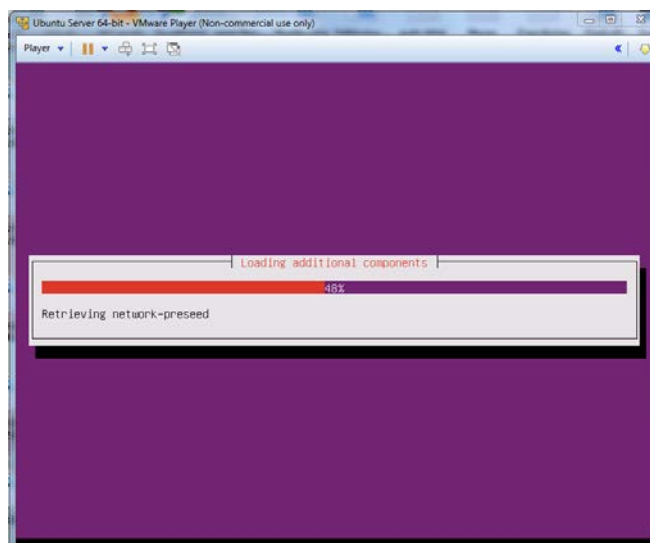
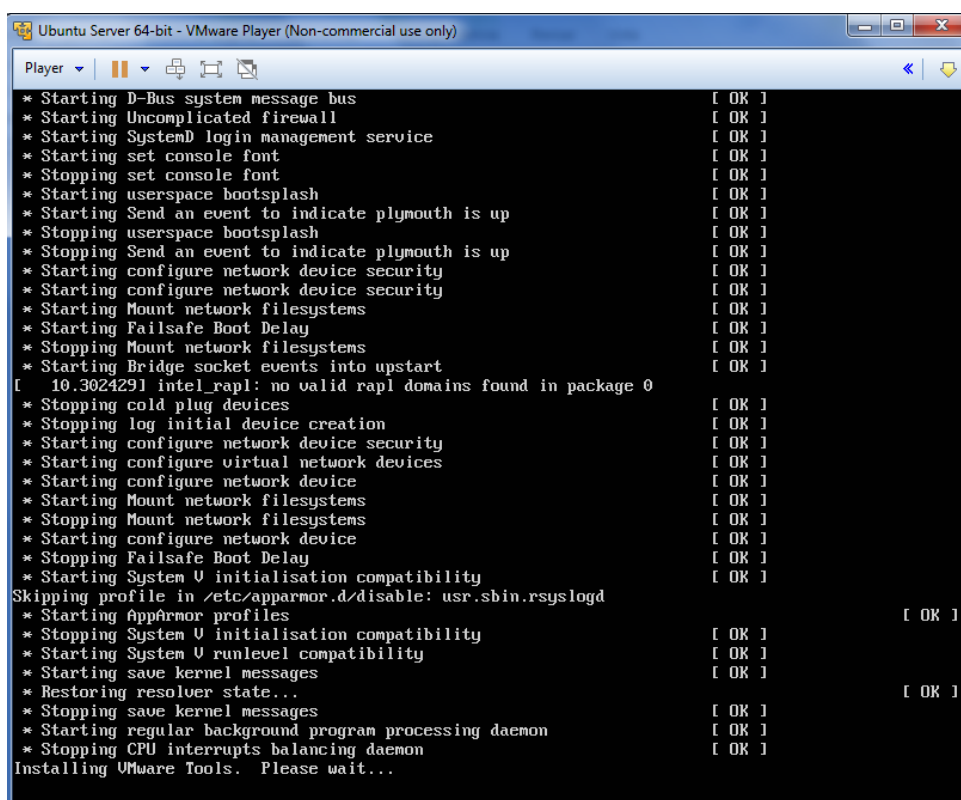


Figura 3.6. Instalación en curso de la Máquina Virtual

- Terminada la instalación, automáticamente se inicia el sistema operativo e inicia el proceso de instalación de las herramientas de VMWare (*VMWare Tools*) como se observa en la Figura 3.7. Esto permite una mejor comunicación e interacción entre la máquina virtual y la máquina física.



```
Ubuntu Server 64-bit - VMware Player (Non-commercial use only)
Player
* Starting D-Bus system message bus [ OK ]
* Starting Uncomplicated firewall [ OK ]
* Starting SystemD login management service [ OK ]
* Starting set console font [ OK ]
* Stopping set console font [ OK ]
* Starting userspace bootsplash [ OK ]
* Starting Send an event to indicate plymouth is up [ OK ]
* Stopping userspace bootsplash [ OK ]
* Stopping Send an event to indicate plymouth is up [ OK ]
* Starting configure network device security [ OK ]
* Starting configure network device security [ OK ]
* Starting Mount network filesystems [ OK ]
* Starting Failsafe Boot Delay [ OK ]
* Stopping Mount network filesystems [ OK ]
* Starting Bridge socket events into upstart [ OK ]
[ 10.302429] intel_rapl: no valid rapl domains found in package 0
* Stopping cold plug devices [ OK ]
* Stopping log initial device creation [ OK ]
* Starting configure network device security [ OK ]
* Starting configure virtual network devices [ OK ]
* Starting configure network device [ OK ]
* Starting Mount network filesystems [ OK ]
* Stopping Mount network filesystems [ OK ]
* Starting configure network device [ OK ]
* Stopping Failsafe Boot Delay [ OK ]
* Starting System U initialisation compatibility [ OK ]
Skipping profile in /etc/apparmor.d/disable: usr.sbin.rsyslogd
* Starting AppArmor profiles [ OK ]
* Stopping System U initialisation compatibility [ OK ]
* Starting System U runlevel compatibility [ OK ]
* Starting save kernel messages [ OK ]
* Restoring resolver state... [ OK ]
* Stopping save kernel messages [ OK ]
* Starting regular background program processing daemon [ OK ]
* Stopping CPU interrupts balancing daemon [ OK ]
Installing VMWare Tools. Please wait...
```

Figura 3.7. Instalación de herramientas de VMWare

- La creación e instalación de la máquina virtual ha terminado y está operativa completamente, Figura 3.8.

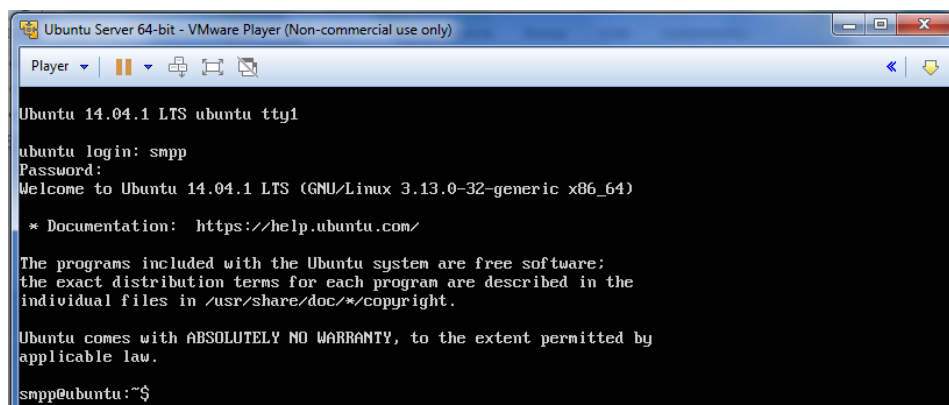


Figura 3.8. Máquina Virtual instalada con éxito

### 3.3 Descripción General de la Implementación del SMSC

Una vez realizada una breve explicación de las herramientas utilizadas para la implementación del proyecto, en esta sección se explica un marco general de cómo está estructurado.

El sistema operativo Ubuntu se muestra en la Figura 3.9.

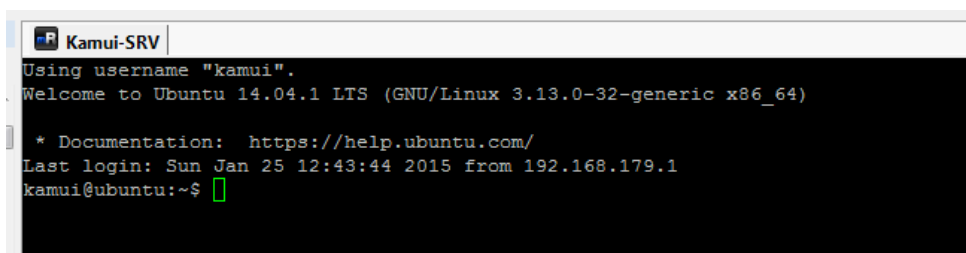


Figura 3.9. Ubuntu 14.04

La máquina virtual que se ejecuta sobre la plataforma de VMWare se muestra en la Figura 3.10.

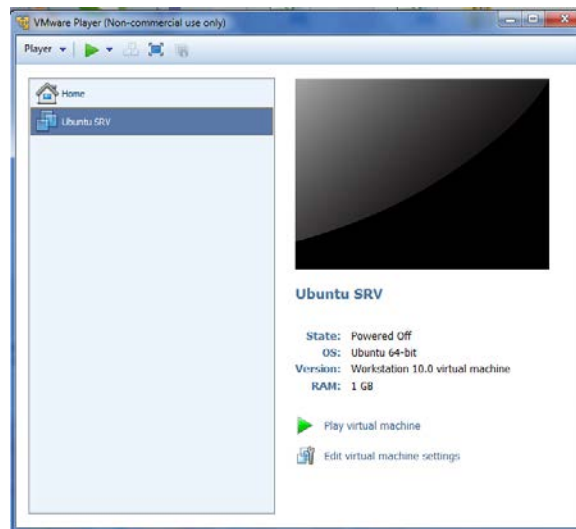


Figura 3.10. Máquina Virtual VMware

En la Figura 3.11 se presenta un gráfico en donde se tiene un ESME, un bloque de color rojo que representa el servicio SMSC, un bloque de color verde que representa la aplicación web y la base de datos MariaDB.

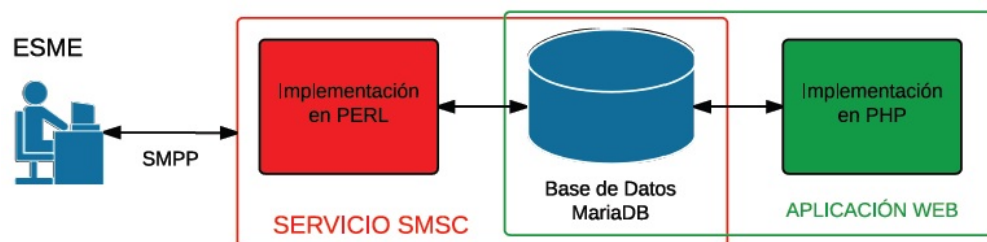


Figura 3.11. Descripción General del Proyecto

El ESME es cualquier cliente que se conecte a este SMSC vía SMPP. Se ha considerado el bloque del ESME en la Figura 3.11 con el fin de ilustrar la estructura de la comunicación entre ESME y SMSC.

El dominio total del SMSC lo tiene el servicio implementado en PERL, Figura 3.11. Para el efecto, se ha hecho uso de una librería en Perl para el manejo de los mensajes SMPP. Esta librería es Net::SMPP y se la encuentra en la página web de CPAN [17], donde se pueden encontrar alguna librerías de Perl para otros usos.

En la base de datos se guarda la información del número corto del ESME (short code), claves de ESME e interacción entre el programa de manejo del protocolo SMPP y el ambiente visual de gestión.

La decisión de qué ESME es el destinatario del SMS recibido y almacenado en el SMSC, se lo realiza por medio de la tabla *tb\_sh\_codes* en la base de datos, Figura 3.12. Ahí se realiza una relación del número corto de destinatario al identificador numérico del ESME creado en la tabla *tb\_esme*, Figura 3.13.

<input type="checkbox"/>	id_sc	short_code	id_esme
<input type="checkbox"/>	1	2020	1
<input type="checkbox"/>	2	8080	2
<input type="checkbox"/>	3	5050	3
<input type="checkbox"/>	4	6060	4
<input type="checkbox"/>	5	4040	5
*	(Auto)	(NULL)	(NULL)

Figura 3.12. *tb\_sh\_codes*

Tabla tb\_esme:

id_esme	system_id	password	Descripcion
1	abc	123	ESME Test1
2	abcd	1234	ESME Test2
3	telfonica	123456	
4	cnt	123456	CNI EP - Movil
5	claro	123456	Concecel - Claro
*	(Auto)	(NULL)	(NULL)

Figura 3.13. tb\_esme

La aplicación web se implementó usando el lenguaje PHP, esta se conecta con la base de datos como se observa en el bloque de color verde en la Figura 3.11.

### 3.4 Funcionamiento del SMSC

El servicio implementado en Perl es la parte más importante del proyecto y se puede dividir en cuatro procesos o tres niveles detallados en la Figura 3.14.

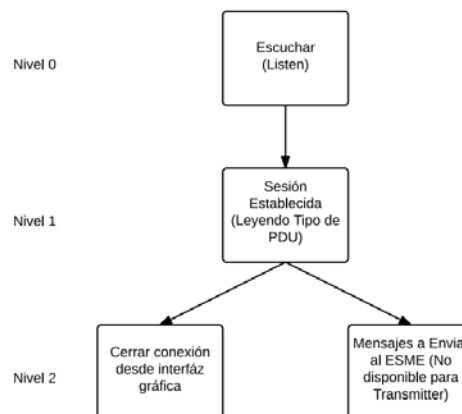


Figura 3.14. Diagrama del funcionamiento del SMSC

El proceso Escuchar (Listen) o también llamado Nivel 0, es el primer proceso en donde el SMSC está escuchando por el puerto 2251 esperando por una conexión de algún ESME (Figura 3.15), esto se implementó haciendo uso de la librería para SMPP.

```
kamui@ubuntu:~$ sudo netstat -tnlp
[sudo] Password for kamui:
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address          Foreign Address        State       PID/Program name
tcp        0      0 0.0.0.0:*               0.0.0.0:*              LISTEN      1618/mysql
tcp        0      0 0.0.0.0:2251           0.0.0.0:*              LISTEN      1644/perl
tcp        0      0 0.0.0.0:22             0.0.0.0:*              LISTEN      959/sshd
tcp6       0      0 :::80                  :::*                    LISTEN      1134/apache2
tcp6       0      0 :::22                  :::*                    LISTEN      959/sshd
kamui@ubuntu:~$
```

Figura 3.15. Proceso Escuchar (Listen)

En la figura 3.15, se puede observar el servicio en ejecución y un segmento del código se detalla en la Figura 3.16.

```
smc-v9.pl
123
124  if ($listen) {
125      while (1) {
126          # proceso padre de esperar conecciones de ESME
127          $smc = $listen->accept;
128          !defined $smc ? next : fork or last;
129      }

```

Figura 3.16. Código del Proceso Escuchar (Listen)

Se ha denominado también al proceso Escuchar como proceso Padre, porque en el momento en que un ESME hace conexión con el SMSC se derivan y crean más sub-procesos.

En el proceso Hijo o Nivel 1 ya se ha establecido una conexión con algún ESME específico. Este se encuentra leyendo el tipo de PDU, maneja el envío/recepción de mensajes en esa sesión y dependiendo del tipo de conexión del ESME crea hasta dos procesos.

Estos dos procesos están en el Nivel 2 y se detallan a continuación:

- Proceso Cerrar Conexión: Maneja la desconexión del ESME por medio del cambio de un estado en la base de datos, cambio que es realizado en la aplicación de administración.
- Envío de Mensajes: Maneja el envío de mensajes al ESME conectado como tipo receiver o transceiver. Estos mensajes han sido enviados por otro ESME ya sea previamente o durante la conexión del ESME destinatario.

Si el ESME se conecta como un transmitter, solo se desglosa el proceso de desconexiones. La razón se da porque el ESME como transmitter envía mensajes al SMSC y no las recibe. Se ha enmarcado de color verde el proceso de nivel 1 y 2 respectivamente en la Figura 3.17.

```

Kamui-SRV Kamui-SRV Kamui-SRV
kamui@ubuntu:~$ ps aux | grep smsc
kamui    1959  0.0  1.1 132564 11940 pts/0    S+   14:23   0:00 perl smsc-v9.pl
kamui    1977  0.0  0.0  11744   916 pts/1    S+   14:26   0:00 grep --color=auto smsc
kamui@ubuntu:~$ ps aux | grep smsc
kamui    1959  0.0  1.1 132564 11940 pts/0    S+   14:23   0:00 perl smsc-v9.pl
kamui    1978  0.0  0.9 132564  9832 pts/0    S+   14:26   0:00 perl smsc-v9.pl
kamui    1980  0.0  0.9 132564  9464 pts/0    S+   14:26   0:00 perl smsc-v9.pl
kamui    1982  0.0  0.0  11744   916 pts/1    S+   14:26   0:00 grep --color=auto smsc

```

Figura 3.17. ESME hizo conexión como transmitter

Por otro lado, si el ESME se conecta como un receiver o transceiver, se desglosa los dos procesos que están en el Nivel 2. Se ha enmarcado de color verde el proceso de nivel 1 y 2 respectivamente en la Figura 3.18.



```
kamui@ubuntu:~$ ps aux | grep smsc
kamui 1959 0.0 1.1 132564 11940 pts/0 S+ 14:23 0:00 perl smac-v9.pl
kamui 1985 0.0 0.9 132564 9832 pts/0 S+ 14:27 0:00 perl smac-v9.pl
kamui 1986 0.0 0.9 132564 9456 pts/0 S+ 14:27 0:00 perl smac-v9.pl
kamui 1987 0.0 0.9 132564 9464 pts/0 S+ 14:27 0:00 perl smac-v9.pl
kamui 1989 0.0 0.0 11744 916 pts/1 S+ 14:27 0:00 grep --color=auto smac
kamui@ubuntu:~$
```

Figura 3.18. ESME hizo conexión como receiver o transceiver

A continuación, se presenta en la Figura 3.19 los tipos de PDU utilizados en el programa Perl:

```
62 use constant cmd_tab => {
63     0x80000000 => { cmd => 'generic_ack', reply => undef, },
64     0x80000001 => { cmd => 'bind_receiver', reply => _receiver, },
65     0x80000001 => { cmd => 'bind_receiver_resp', reply => undef, },
66     0x80000002 => { cmd => 'bind_transmitter', reply => _transmitter, },
67     0x80000002 => { cmd => 'bind_transmitter_resp', reply => undef, },
68     0x80000003 => { cmd => 'query_sm', reply => undef, },
69     0x80000003 => { cmd => 'query_sm_resp', reply => undef, },
70     0x80000004 => { cmd => 'submit_sm', reply => _submit_sm, },
71     0x80000005 => { cmd => 'deliver_sm', reply => undef, },
72     0x80000004 => { cmd => 'submit_sm_resp', reply => undef, },
73     0x80000005 => { cmd => 'deliver_sm_resp', reply => undef, },
74     0x80000006 => { cmd => 'unbind', reply => _unbind, },
75     0x80000006 => { cmd => 'unbind_resp', reply => undef, },
76     0x80000007 => { cmd => 'replace_sm', reply => undef, },
77     0x80000007 => { cmd => 'replace_sm_resp', reply => undef, },
78     0x80000008 => { cmd => 'cancel_sm', reply => undef, },
79     0x80000008 => { cmd => 'cancel_sm_resp', reply => undef, },
80     0x80000009 => { cmd => 'bind_transceiver', reply => _transceiver, },
81     0x80000009 => { cmd => 'bind_transceiver_resp', reply => undef, },
82     0x8000000b => { cmd => 'outbind', reply => undef, },
83     0x80000015 => { cmd => 'enquire_link', reply => _enquire_link, },
84     0x80000015 => { cmd => 'enquire_link_resp', reply => undef, },
85 };
```

Figura 3.19. Tipos de PDU programados en Perl

Para una fácil selección de que función se debe ejecutar al recibir un PDU, se creó la tabla `cmd_tab` en la base de datos, lo que hace la relación código en hexadecimal del PDU con su nombre de código y la función que manejará el PDU recibido.

La Figura 3.20, muestra el código del proceso hijo de nivel 1 que establece la conexión con el ESME, crea los 2 procesos hijos por medio

de *fork* y entra en un lazo infinito que espera recibir los mensajes de PDU del ESME ya conectado.

```

130 # proceso child-11, manejo de la conexión del ESME
131 until $!listen;
132 log("log:ESME conectado");
133 my $pdu = $smc->read_pdu() or die;
134 if (defined cmd_tab->{$pdu->{cmd}} and ( cmd_tab->{$pdu->{cmd}}{cmd} eq "bind_transceiver" or cmd_tab->{$pdu->{cmd}}{cmd} eq "bind_transmitter" or cmd_tab->{$pdu->{cmd}}{cmd} eq "bin
135 _log("recibido:", cmd_tab->{$pdu->{cmd}}{cmd} . ":", $pdu->{seq} . ":", $pdu->{status} . ":", $pdu->{system_id} . "/" . $pdu->{password} . ":", unpack('H*', $pdu->{data}));
136 $esme_sysid = ${cmd_tab->{$pdu->{cmd}}{reply}}{$smc, $pdu, \%conf};
137 if (defined $esme_sysid) {
138     log("sesion:", $pdu->{system_id} . ":", cmd_tab->{$pdu->{cmd}}{cmd} . ":", $);
139     $cid = esme_connection(\%conf, $pdu->{system_id}, cmd_tab->{$pdu->{cmd}}{cmd});
140 }
141 } else {
142     _log("error:PDU rechazado: No es inicio de sesion:", $pdu->{cmd} , cmd_tab->{$pdu->{cmd}}{cmd});
143     exit;
144 }
145 $conf{'cid'} = $cid;
146 $conf{'pid'} = $$;
147 if (fork()) {
148     if (fork()) {
149         # proceso child-12, parent, manejo de PDU
150         while (1) {
151             $pdu = $smc->read_pdu() or esme_disconnect(\%conf);
152             if (defined cmd_tab->{$pdu->{cmd}}) {
153                 _log("recibido:", cmd_tab->{$pdu->{cmd}}{cmd} . ":", $pdu->{seq} . ":", $pdu->{status} . ":", ":", unpack('H*', $pdu->{data}));
154                 if (defined cmd_tab->{$pdu->{cmd}}{reply}) { &{cmd_tab->{$pdu->{cmd}}{reply}}($smc, $pdu, \%conf);
155             } else {
156                 _log("error:PDU rechazado: No definido:", $pdu->{cmd});
157                 sleep 1;
158             }
159         }
160     }
161 }

```

Figura 3.20. Código del manejo de conexión con el ESME

Se usa la tabla `cmd_tab` para ejecutar la función que procesará el PDU recibido. De esta manera no es necesario crear una condición por cada tipo de PDU para procesar la respuesta. Lo que reduce el código, su tiempo de ejecución y uso de recursos de CPU.

La figura 3.21, muestra el código del proceso hijo de nivel 2 que maneja las desconexiones del ESME. Esto se realiza por medio de una consulta a la base de datos, preguntando por el estado de la conexión que maneja. Si el estado es 2 indica que se debe desconectar este ESME, y la desconexión se la realiza matando al proceso padre por medio de una interrupción. El otro sub-proceso, al quedar huérfano se cierra, así evitado que queden procesos en ejecución que ya no son usados.

```

161 # proceso child2-L2, disconnect
162 my $dbh = db_connect(\%conf);
163 while (1) {
164     unless (kill 0, $conf{'pid'}) { exit; }
165     sleep 1;
166     my $sql = qq(SELECT `process` FROM `tb_connections` WHERE id = $cid AND `status` = 2);
167     my @ary = $dbh->selectrow_array($sql);
168     if ( defined $ary[0] and $ary[0] == $conf{'pid'}) {
169         kill INT => $conf{'pid'};
170         exit;
171     }
172 }
173 }

```

Figura 3.21. Código del manejo de desconexión

La figura 3.22, muestra el proceso hijo de nivel 2 de envío de SMS al ESME conectado. Esto se realiza por medio un lazo infinito donde se realiza una consulta a la base de datos, por si existen mensajes que tiene como destinatario el ESME a cargo. En ese caso se envían los mensajes por medio del PDU deliver\_sm y se actualiza la tabla de mensajes cambiando el estado ha enviado, para que no se vuelva a procesar dichos mensajes. La única manera de que este proceso se termine es por la muerte o finalización de su proceso padre, que es el proceso de nivel 1.

```

175 # proceso child1-L2, envío de SMS a ESME
176 my $dbh = db_connect(\%conf);
177 while (1) {
178     unless (kill 0, $conf{'pid'}) { exit; }
179     sleep 1;
180     if ( getppid == 1 or cmd_tab->{$pdu->{cmd}}{cmd} eq "bind_transmitter" ) {
181         $dbh->disconnect();
182         exit;
183     }
184     my $sql = qq(SELECT id,mensaje,numero_A,numero_B,msg_id FROM `tb_cola_sms` WHERE numero_B =
185         ( SELECT `short_code` FROM `tb_esme` LEFT JOIN `tb_sh_codes`
186           USING (id_esme) WHERE system_id = \'$esme_sysid\' )
187         AND estado = 0);
188     my $sms = $dbh->selectall_hashref($sql,'id');
189
190     $sql = qq(UPDATE `tb_cola_sms` SET estado = 1 , seq = ? WHERE id = ? );
191     my $stm = $dbh->prepare($sql);
192
193     foreach my $id (keys %{$sms}) {
194         if ( my $seq = smsc_deliver_sm($smc,$sms->{$id}) ) {
195             $stm->execute($seq,$id);
196         }
197     }
198 }
199 }

```

Figura 3.22. Código del manejo de envío de mensajes al ESME

En resumen los pasos a seguir para la implementación del SMSC son:

1. Instalación del servidor Linux.
2. Instalación de Perl 5.8 o superior.
3. Instalación de la librería de perl Net::SMPP.
4. Instalación de la librería de perl DBI:MySQL.
5. Instalación de apache 2.2 o 2.4.
6. Instalación de PHP 5 o superior con soporte a MySQL.
7. Instalación de MariaDB 5.5 o superior.
8. Creación de base de datos, y tablas necesarias para el funcionamiento del demonio.
  - a. Tabla de los diferentes ESME, system\_id y contraseña.
  - b. Tabla de números cortos, donde relaciona a cada uno con un ESME.
  - c. Tabla de registro de los mensajes SMPP enviados y recibidos.
9. Aplicación o servicio en PERL usando la librería Net::SMPP, para creación del demonio SMSC.
  - a. Iniciar el protocolo SMPP, con la función *smpp\_start* indicando el puerto sobre el que se trabajará.
  - b. Cuando exista el inicio de conexión de un ESME se debe dar un *listen->accept*.
  - c. Para recibir los PDU se usa la función *read\_pdu*.
  - d. Para validar el system\_id y contraseña del ESME, se realiza una consulta a la base de datos a la tabla de ESME.

- e. La librería ya cuenta con una función de cada tipo de mensaje del protocolo SMPP, por lo que se debe programar es que función usar en caso de recibir un PDU específico.
  - f. Para poder hacer varias acciones en paralelo, como esperar PDU, enviar SMS, aceptar la conexión de otro ESME, se crean nuevos procesos por medio de la función *fork*.
10. Aplicación en PHP, para ambiente Web que permite la administración del demonio SMSC.
- a. El programa en PHP debe visualizar 3 tablas antes especificadas.
  - b. Debe permitir el editar y crear registros en la tabla de ESME y de números cortos.

## **CAPÍTULO 4**

### **4 ANÁLISIS DERESULTADOS**

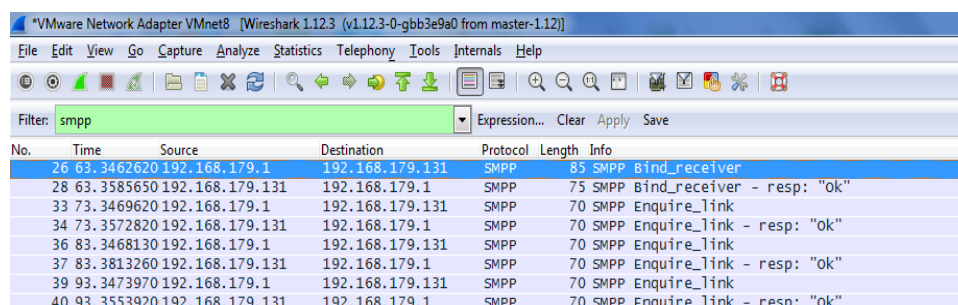
#### **4.1 Análisis General**

Para un análisis de resultados sobre la implementación del SMSC y su buen funcionamiento mediante el protocolo SMPP, se utiliza la herramienta Wireshark, para analizar el tráfico del protocolo SMPP.

Para el presente trabajo, el análisis se realizó con un ESME de prueba configurado en una computadora con dirección ip 192.168.179.1 y un SMSC implementado en una maquina virtual VMware instalado en la misma computadora donde se encuentra el ESME de prueba pero que tiene dirección ip 192.168.179.131.

#### 4.1.1 Conexión del ESME hacia el SMSC en modo bind\_receiver

En la Figura 4.1 observamos que el ESME ha solicitado un inicio de sesión de tipo bind\_receiver en el paquete 26 y su respuesta consecuentemente es un bind\_receiver\_resp en el paquete 28 enviado por el SMSC. Posteriormente, el ESME está a la espera de algún mensaje y envía enquire\_link para verificar que aún haya conexión entre ESME y SMSC. Para cada enquire\_link existe su enquire\_link\_resp y esto asegura que la conexión aún existe.



The screenshot shows a Wireshark capture of SMPP traffic. The filter is set to 'smpp'. The packet list table is as follows:

No.	Time	Source	Destination	Protocol	Length	Info
26	63.3462620	192.168.179.1	192.168.179.131	SMPP	85	SMPP Bind_receiver
28	63.3585650	192.168.179.131	192.168.179.1	SMPP	75	SMPP Bind_receiver - resp: "ok"
33	73.3469620	192.168.179.1	192.168.179.131	SMPP	70	SMPP Enquire_link
34	73.3572820	192.168.179.131	192.168.179.1	SMPP	70	SMPP Enquire_link - resp: "ok"
36	83.3468130	192.168.179.1	192.168.179.131	SMPP	70	SMPP Enquire_link
37	83.3813260	192.168.179.131	192.168.179.1	SMPP	70	SMPP Enquire_link - resp: "ok"
39	93.3473970	192.168.179.1	192.168.179.131	SMPP	70	SMPP Enquire_link
40	93.3553920	192.168.179.131	192.168.179.1	SMPP	70	SMPP Enquire_link - resp: "ok"

Figura 4.1. Conexión de un ESME tipo bind\_receiver

#### 4.1.2 Conexión del ESME hacia el SMSC en modo bind\_transmitter

Podemos observar en la Figura 4.2, que el ESME genera otra conexión para hacer un inicio de sesión ahora como bind\_transmitter en el paquete 56 y su respectiva respuesta bind\_transmitter\_resp está dada en el paquete 58 que emite el SMSC, lo cual indica un ESME como transmitter exitoso. Ahora en la misma Figura 4.2, en el paquete 63 se muestra un mensaje PDU

de tipo submit\_sm por parte del ESME. El SMSC recibe este mensaje submit\_sm y le responde con un submit\_sm\_resp con el paquete 65. Este SMSC se encarga ahora de enviar el mensaje al mismo ESME mencionado (así se configuró para esta prueba) mediante el mensaje PDU deliver\_sm con el paquete 67 y su respectiva respuesta deliver\_sm\_resp por parte del ESME es dada por el paquete 68. Como aún no se cierra la conexión, el ESME sigue verificando que aún haya dicha conexión con enquire\_link en el paquete 70.

No.	Time	Source	Destination	Protocol	Length	Info
53	113.347427	192.168.179.1	192.168.179.131	SMPP	70	SMPP Enquire_link
54	113.355160	192.168.179.131	192.168.179.1	SMPP	70	SMPP Enquire_link - resp: "ok"
56	113.719604	192.168.179.1	192.168.179.131	SMPP	83	SMPP Bind_transmitter
58	113.732292	192.168.179.131	192.168.179.1	SMPP	75	SMPP Bind_transmitter - resp: "ok"
63	116.927408	192.168.179.1	192.168.179.131	SMPP	107	SMPP Submit_sm
65	116.978057	192.168.179.131	192.168.179.1	SMPP	81	SMPP Submit_sm - resp: "ok"
67	117.477142	192.168.179.131	192.168.179.1	SMPP	107	SMPP Deliver_sm
68	117.593242	192.168.179.1	192.168.179.131	SMPP	71	SMPP Deliver_sm - resp: "ok"
70	126.927801	192.168.179.1	192.168.179.131	SMPP	70	SMPP Enquire_link
72	126.966278	192.168.179.131	192.168.179.1	SMPP	70	SMPP Enquire_link - resp: "ok"

Figura 4.2. Conexión de un ESME tipo bind\_transmitter

#### 4.1.3 Desconexión de un ESME con Unbind

En la Figura 4.3 se muestra el intercambio de mensajes de la desconexión. El ESME ha solicitado la desconexión hacia el SMSC mediante el mensaje PDU unbind, y esta SMSC le responde con un unbind\_resp indicando desconexión exitosa.



175432	285.790790	192.168.179.1	192.168.179.131	SMPP	70 SMPP Unbind
175433	285.800298	192.168.179.131	192.168.179.1	SMPP	70 SMPP Unbind - resp: "ok"

Figura 4.3. Desconexión de un ESME

Como un breve resumen de lo detallado en este análisis, se hizo un diagrama explicativo de lo revisado en todo este subcapítulo, Figura 4.4.

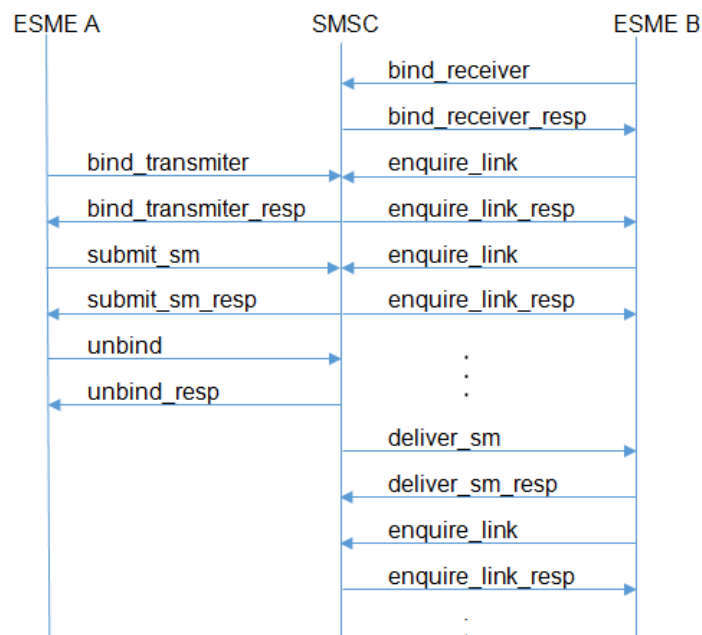


Figura 4.4. Resumen de mensajes PDU del proyecto

## 4.2 Pruebas de Errores

Como se mencionó en las limitaciones del proyecto, se analizan tres tipos de errores: no hay error, clave incorrecta y falla de estado de enlace (bind). Cuando no hay error se ha venido mencionando con la palabra

OK en las respuestas a cada mensaje PDU, por lo tanto ya se ha analizado este punto.

Se analizan ahora los otros dos errores restantes. Como se observa en la Figura 4.5, existe ya un ESME conectado como bind\_receiver e intencionalmente se envió otro bind\_receiver con el objetivo de que el SMSC responda con un mensaje de error, en este caso el error de falla de estado bind (bind failed).

175293	227.343777	192.168.179.1	192.168.179.131	SMPP	87	SMPP	Bind_receiver	
175300	227.355514	192.168.179.131	192.168.179.1	SMPP	75	SMPP	Bind_receiver	- resp: "bind failed"
175305	227.357221	192.168.179.1	192.168.179.131	SMPP	87	SMPP	Bind_receiver	
175312	227.368027	192.168.179.131	192.168.179.1	SMPP	75	SMPP	Bind_receiver	- resp: "bind failed"
175317	227.369805	192.168.179.1	192.168.179.131	SMPP	87	SMPP	Bind_receiver	
175324	227.379964	192.168.179.131	192.168.179.1	SMPP	75	SMPP	Bind_receiver	- resp: "bind failed"
175329	227.381638	192.168.179.1	192.168.179.131	SMPP	87	SMPP	Bind_receiver	
175336	227.391916	192.168.179.131	192.168.179.1	SMPP	75	SMPP	Bind_receiver	- resp: "bind failed"
175341	227.393899	192.168.179.1	192.168.179.131	SMPP	87	SMPP	Bind_receiver	

Figura 4.5. Falla de estado bind (bind failed)

Para el análisis del error de conexión de un ESME por clave incorrecta, en la Figura 4.6, se muestra en el paquete 183565 que el SMSC responde con un bind\_receiver\_resp indicando en uno de sus parámetros que la clave ha sido incorrecta (invalid password) por parte del ESME.

183565	327.083314	192.168.179.131	192.168.179.1	SMPP	75	SMPP	Bind_receiver	- resp: "invalid password"
183570	327.084952	192.168.179.1	192.168.179.131	SMPP	88	SMPP	Bind_receiver	
183577	327.093790	192.168.179.131	192.168.179.1	SMPP	75	SMPP	Bind_receiver	- resp: "invalid password"
183582	327.095556	192.168.179.1	192.168.179.131	SMPP	88	SMPP	Bind_receiver	
183589	327.105192	192.168.179.131	192.168.179.1	SMPP	75	SMPP	Bind_receiver	- resp: "invalid password"
183594	327.106925	192.168.179.1	192.168.179.131	SMPP	88	SMPP	Bind_receiver	

Figura 4.6. Clave incorrecta (invalid password)

### 4.3 Detalle de Parámetros de mensajes PDU

A continuación, se presenta y analiza un detalle breve de los parámetros de algunos de los mensajes PDU que se han utilizado en el presente trabajo.

#### PDU bind\_transceiver

En la Figura 4.7, se observa los parámetros del PDU bind\_transceiver. En estos parámetros están incluidos los comandos de la cabecera PDU y los parámetros mandatorios del cuerpo PDU. Los parámetros observados son: Length, Operation, Sequence, System ID, Password, Version, Type of Number (TON), Numbering plan indicator (NPI).

The screenshot shows a network traffic analysis tool interface. The top part displays a list of captured packets with columns for No., Time, Source, Destination, Protocol, Length, and Info. The selected packet (No. 101386) is highlighted in blue. Below the list, the detailed view of the selected packet is shown, including the Ethernet II header, Internet Protocol Version 4 header, and the Short Message Peer to Peer (SMPP) body. The SMPP body details are as follows:

Field	Value
Length	29
Operation	Bind_transceiver (0x00000009)
Sequence #	0
System ID	abc
Password	123
Version (1f)	0.3
Type of number	International (0x01)
Numbering plan indicator	ISDN (E163/E164) (0x01)

Figura 4.7. Parámetros bind\_transceiver

La respuesta a este PDU bind\_transceiver se observa en la Figura 4.8. Los parámetros se limitan a los siguientes: Length, Operation, Result, Sequence y System ID.

No.	Time	Source	Destination	Protocol	Length	Info
101386	194.078111	192.168.179.1	192.168.179.131	SMPP	83	SMPP Bind_transceiver
101388	194.099528	192.168.179.131	192.168.179.1	SMPP	75	SMPP Bind_transceiver - resp: "ok"
101393	195.464178	192.168.179.1	192.168.179.131	SMPP	107	SMPP Submit_sm
101394	195.488677	192.168.179.131	192.168.179.1	SMPP	81	SMPP Submit_sm - resp: "ok"
101396	196.116382	192.168.179.131	192.168.179.1	SMPP	107	SMPP Deliver_sm
101397	196.188795	192.168.179.1	192.168.179.131	SMPP	71	SMPP Deliver_sm - resp: "ok"
101401	196.192504	192.168.179.1	192.168.179.131	SMPP	70	SMPP Enquire_link

Frame 101388: 75 bytes on wire (600 bits), 75 bytes captured (600 bits) on interface 0  
 Ethernet II, Src: Vmware\_65:65:1f (00:0c:29:65:65:1f), Dst: Vmware\_c0:00:08 (00:50:56:c0:00:08)  
 Internet Protocol Version 4, Src: 192.168.179.131 (192.168.179.131), Dst: 192.168.179.1 (192.168.179.1)  
 Transmission Control Protocol, Src Port: 2251 (2251), Dst Port: 24307 (24307), Seq: 1, Ack: 30, Len: 21  
 Short Message Peer to Peer, Command: Bind\_transceiver - resp, Status: "ok", Seq: 0, Len: 21  
 Length: 21  
 Operation: Bind\_transceiver - resp (0x80000009)  
 Result: ok (0x00000000)  
 Sequence #: 0  
 System ID: CAMD

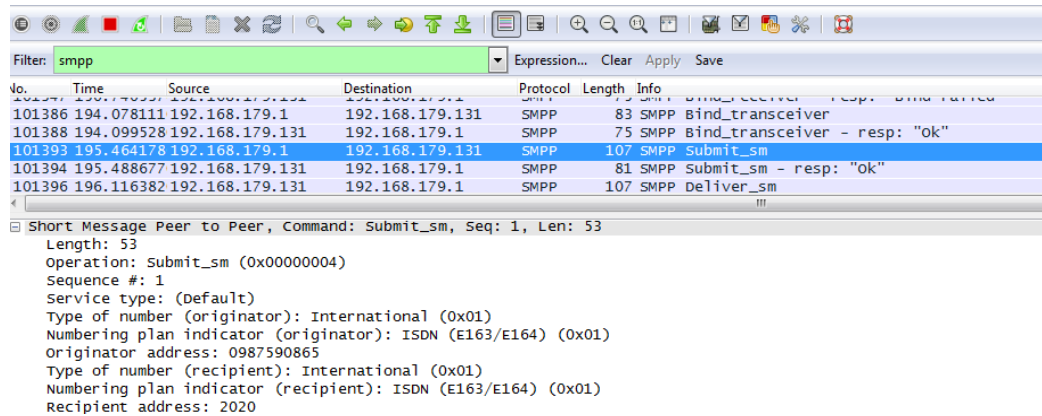
Figura 4.8. Parámetros bind\_transceiver\_resp

### PDU Submit\_sm y Deliver\_sm

Los parámetros para estos PDU submit\_sm (Figura 4.9) y deliver\_sm (Figura 4.11) son más que para cualquier otro PDU. Esto se debe a que aquí se encuentra el mensaje corto y el tamaño del campo length tiene 53 octetos, comparado con otros PDU en que su tamaño es alrededor de 20 octetos.

Las respuestas para estos PDU son submit\_sm\_resp (Figura 4.10) y deliver\_sm\_resp (Figura 4.12) respectivamente y tienen menos tamaño

en sus mensajes. Este hecho se da porque una respuesta no lleva el mensaje corto, sino que puede ser un aviso de éxito o de fracaso de que el mensaje ha llegado al SMSC.



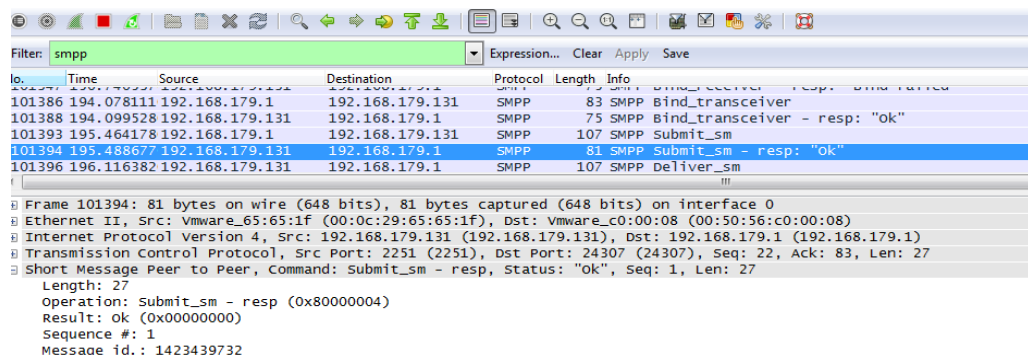
No.	Time	Source	Destination	Protocol	Length	Info
101387	194.740337	192.168.179.131	192.168.179.1	SMPP	75	SMPP Bind_transceiver - resp: "ok"
101386	194.078111	192.168.179.1	192.168.179.131	SMPP	83	SMPP Bind_transceiver
101388	194.099528	192.168.179.131	192.168.179.1	SMPP	75	SMPP Bind_transceiver - resp: "ok"
101393	195.464178	192.168.179.1	192.168.179.131	SMPP	107	SMPP Submit_sm
101394	195.488677	192.168.179.131	192.168.179.1	SMPP	81	SMPP Submit_sm - resp: "ok"
101396	196.116382	192.168.179.131	192.168.179.1	SMPP	107	SMPP Deliver_sm

```

Short Message Peer to Peer, Command: Submit_sm, Seq: 1, Len: 53
Length: 53
Operation: Submit_sm (0x00000004)
Sequence #: 1
Service type: (Default)
Type of number (originator): International (0x01)
Numbering plan indicator (originator): ISDN (E163/E164) (0x01)
Originator address: 0987590865
Type of number (recipient): International (0x01)
Numbering plan indicator (recipient): ISDN (E163/E164) (0x01)
Recipient address: 2020
  
```

Figura 4.9. Parámetros submit\_sm

A continuación, se muestran los parámetros en respuesta a submit\_sm en la Figura 4.10:



No.	Time	Source	Destination	Protocol	Length	Info
101387	194.740337	192.168.179.131	192.168.179.1	SMPP	75	SMPP Bind_transceiver - resp: "ok"
101386	194.078111	192.168.179.1	192.168.179.131	SMPP	83	SMPP Bind_transceiver
101388	194.099528	192.168.179.131	192.168.179.1	SMPP	75	SMPP Bind_transceiver - resp: "ok"
101393	195.464178	192.168.179.1	192.168.179.131	SMPP	107	SMPP Submit_sm
101394	195.488677	192.168.179.131	192.168.179.1	SMPP	81	SMPP Submit_sm - resp: "ok"
101396	196.116382	192.168.179.131	192.168.179.1	SMPP	107	SMPP Deliver_sm

```

Frame 101394: 81 bytes on wire (648 bits), 81 bytes captured (648 bits) on interface 0
Ethernet II, Src: Vmware_65:65:1f (00:0c:29:65:65:1f), Dst: Vmware_c0:00:08 (00:50:56:c0:00:08)
Internet Protocol Version 4, Src: 192.168.179.131 (192.168.179.131), Dst: 192.168.179.1 (192.168.179.1)
Transmission Control Protocol, Src Port: 2251 (2251), Dst Port: 24307 (24307), Seq: 22, Ack: 83, Len: 27
Short Message Peer to Peer, Command: Submit_sm - resp, Status: "ok", Seq: 1, Len: 27
Length: 27
Operation: Submit_sm - resp (0x80000004)
Result: ok (0x00000000)
Sequence #: 1
Message id.: 1423439732
  
```

Figura 4.10. Parámetros submit\_sm\_resp

Los parámetros deliver\_sm se describe en la Figura 4.11.

o.	Time	Source	Destination	Protocol	Length	Info
101394	195.488677	192.168.179.131	192.168.179.1	SMPP	81	SMPP Submit_sm - resp: "ok"
101396	196.116382	192.168.179.131	192.168.179.1	SMPP	107	SMPP Deliver_sm
101397	196.188795	192.168.179.1	192.168.179.131	SMPP	71	SMPP Deliver_sm - resp: "ok"
101401	206.192594	192.168.179.1	192.168.179.131	SMPP	70	SMPP Enquire_link
101403	206.199852	192.168.179.131	192.168.179.1	SMPP	70	SMPP Enquire_link - resp: "ok"

Short Message Peer to Peer, Command: Deliver\_sm, Seq: 1, Len: 53  
 Length: 53  
 Operation: Deliver\_sm (0x00000005)  
 Sequence #: 1  
 Service type: (Default)  
 Type of number (originator): National (0x02)  
 Numbering plan indicator (originator): Internet (IP) (0x0e)  
 Originator address: 0987590865  
 Type of number (recipient): National (0x02)  
 Numbering plan indicator (recipient): Internet (IP) (0x0e)  
 Recipient address: 2020

Figura 4.11. Parámetros deliver\_sm

A continuación, se muestran los parámetros en respuesta a deliver\_sm en la Figura 4.12.

o.	Time	Source	Destination	Protocol	Length	Info
101394	195.488677	192.168.179.131	192.168.179.1	SMPP	81	SMPP Submit_sm - resp: "ok"
101396	196.116382	192.168.179.131	192.168.179.1	SMPP	107	SMPP Deliver_sm
101397	196.188795	192.168.179.1	192.168.179.131	SMPP	71	SMPP Deliver_sm - resp: "ok"
101401	206.192594	192.168.179.1	192.168.179.131	SMPP	70	SMPP Enquire_link
101403	206.199852	192.168.179.131	192.168.179.1	SMPP	70	SMPP Enquire_link - resp: "ok"

Frame 101397: 71 bytes on wire (568 bits), 71 bytes captured (568 bits) on interface 0  
 Ethernet II, Src: Vmware\_c0:00:08 (00:50:56:c0:00:08), Dst: Vmware\_65:65:1f (00:0c:29:65:65:1f)  
 Internet Protocol Version 4, Src: 192.168.179.1 (192.168.179.1), Dst: 192.168.179.131 (192.168.179.131)  
 Transmission Control Protocol, Src Port: 24307 (24307), Dst Port: 2251 (2251), Seq: 83, Ack: 102, Len: 17  
 Short Message Peer to Peer, Command: Deliver\_sm - resp, Status: "ok", Seq: 1, Len: 17  
 Length: 17  
 Operation: Deliver\_sm - resp (0x80000005)  
 Result: ok (0x00000000)  
 Sequence #: 1

Figura 4.12. Parámetros deliver\_sm\_resp

## PDU Enquire\_link

El PDU enquire\_link y PDU enquire\_link\_resp no presentan el campo cuerpo, es decir que solamente tiene cabecera como se mencionó en la teoría, esto se comprueba observando la Figura 4.13 y Figura 4.14 respectivamente. Al no tener el mensaje PDU un cuerpo y solamente la

cabecera, se hace mas fácil saber el tamaño de ese mensaje PDU, porque cada campo de la cabecera tiene 4 octetos, y la cabecera consta de 4 campos. Por lo tanto, el tamaño del mensaje PDU enquire\_link y su respuesta, tiene 16 octetos.

The image shows a Wireshark packet capture with a filter set to 'smpp'. The packet list pane shows several SMPP packets. Packet 101401 is highlighted, showing an Enquire\_link command. The packet details pane below shows the structure of the Short Message Peer to Peer (SMPP) protocol, including the Command field set to Enquire\_link (0x00000015) and the Sequence number set to 2.

No.	Time	Source	Destination	Protocol	Length	Info
101394	195.488677	192.168.179.131	192.168.179.1	SMPP	81	SMPP Submit_sm - resp: "ok"
101396	196.116382	192.168.179.131	192.168.179.1	SMPP	107	SMPP Deliver_sm
101397	196.188795	192.168.179.1	192.168.179.131	SMPP	71	SMPP Deliver_sm - resp: "ok"
101401	206.192594	192.168.179.1	192.168.179.131	SMPP	70	SMPP Enquire_link
101403	206.199852	192.168.179.131	192.168.179.1	SMPP	70	SMPP Enquire_link - resp: "ok"

Frame 101401: 70 bytes on wire (560 bits), 70 bytes captured (560 bits) on interface 0  
 Ethernet II, Src: Vmware\_c0:00:08 (00:50:56:c0:00:08), Dst: Vmware\_65:65:1f (00:0c:29:65:65:1f)  
 Internet Protocol Version 4, Src: 192.168.179.1 (192.168.179.1), Dst: 192.168.179.131 (192.168.179.131)  
 Transmission Control Protocol, Src Port: 24307 (24307), Dst Port: 2251 (2251), Seq: 100, Ack: 102, Len: 16  
 Short Message Peer to Peer, Command: Enquire\_link, Seq: 2, Len: 16  
 Length: 16  
 Operation: Enquire\_link (0x00000015)  
 Sequence #: 2

Figura 4.13. Parámetros enquire\_link

Los parámetros en respuesta a enquire\_link se muestran en la Figura 4.14.

The image shows a Wireshark packet capture with a filter set to 'smpp'. The packet list pane shows several SMPP packets. Packet 101403 is highlighted, showing an Enquire\_link response. The packet details pane below shows the structure of the Short Message Peer to Peer (SMPP) protocol, including the Command field set to Enquire\_link - resp (0x80000015) and the Result field set to ok (0x00000000).

No.	Time	Source	Destination	Protocol	Length	Info
101394	195.488677	192.168.179.131	192.168.179.1	SMPP	81	SMPP Submit_sm - resp: "ok"
101396	196.116382	192.168.179.131	192.168.179.1	SMPP	107	SMPP Deliver_sm
101397	196.188795	192.168.179.1	192.168.179.131	SMPP	71	SMPP Deliver_sm - resp: "ok"
101401	206.192594	192.168.179.1	192.168.179.131	SMPP	70	SMPP Enquire_link
101403	206.199852	192.168.179.131	192.168.179.1	SMPP	70	SMPP Enquire_link - resp: "ok"

Frame 101403: 70 bytes on wire (560 bits), 70 bytes captured (560 bits) on interface 0  
 Ethernet II, Src: Vmware\_65:65:1f (00:0c:29:65:65:1f), Dst: Vmware\_c0:00:08 (00:50:56:c0:00:08)  
 Internet Protocol Version 4, Src: 192.168.179.131 (192.168.179.131), Dst: 192.168.179.1 (192.168.179.1)  
 Transmission Control Protocol, Src Port: 2251 (2251), Dst Port: 24307 (24307), Seq: 102, Ack: 116, Len: 16  
 Short Message Peer to Peer, Command: Enquire\_link - resp, Status: "ok", Seq: 2, Len: 16  
 Length: 16  
 Operation: Enquire\_link - resp (0x80000015)  
 Result: ok (0x00000000)  
 Sequence #: 2

Figura 4.14. Parámetros enquire\_link\_resp

## **CONCLUSIONES Y RECOMENDACIONES**

Las conclusiones se detallan a continuación:

1. El objetivo principal en la explicación del protocolo SMPP como un servidor (SMSC) se ha detallado de una forma entendible y rápida. Su explicación didáctica proporciona las herramientas básicas que ayudarán al entendimiento del protocolo SMPP. Esta tesis es útil para estudiantes de ingeniería relacionados en el campo de las telecomunicaciones e interesados en el desarrollo de sistemas de envío y recepción de mensajes de texto por medio del protocolo SMPP.
2. Las operadoras celulares manejan mensajes cortos punto a punto mediante su SMSC. Desde un esquema de la red celular, el protocolo SMPP es usado fuera de la red móvil entre un SMSC y un ESME.
3. Este trabajo sólo tiene un alcance didáctico y no comercial, debido a que el SMSC es parte de los equipos dentro de una operadora de telefonía



celular. Para un completo funcionamiento requiere el manejo de otros protocolos de SMS para la recepción de los mensajes internos a la operadora, donde este trabajo solo se enfoca al protocolo SMPP y no cubre lo antes indicado.

4. El protocolo SMPP tiene un gran uso por las empresas telefónicas y proveedores de contenidos. Desde entidades externas que buscan algún negocio en particular, hasta aquellas que lo usan para informar a cierta población.
5. La implementación de protocolos de comunicación con un enfoque didáctico, es viable usando lenguajes de programación que tengan una gran comunidad como soporte, como se lo realizó en este trabajo usando PERL y la librería Net::SMPP.
6. Este trabajo permite que otros estudiantes puedan entender claramente el funcionamiento del protocolo SMPP, debido a que se implementó el uso del protocolo y se muestra en un entorno web de fácil entendimiento el flujo de los mensajes entre el SMSC y el ESME.
7. La herramienta WireShark, es de gran apoyo y utilidad para el entendimiento y análisis de tramas IP, esencial en la docencia para que los estudiantes visualicen los protocolos de comunicación revisados en clases. En el ámbito laboral, es de suma ayuda para el análisis de problemas más complejos donde se requiere revisar a nivel de paquetes el problema.

Las recomendaciones se detallan a continuación:

1. Para un mejor análisis de los mensajes SMPP y su entendimiento con la herramienta WireShark, se puede hacer esta implementación con 3 máquinas virtuales, para que 2 hagan la función de ESME y una de SMSC y así se puede identificar el tráfico de cada uno por la IP fuente.
2. Para una implementación física se puede usar las minicomputadoras “raspberry pi”, debido a su bajo costo y que funcionan con Linux y es totalmente compatible con los requisitos de este trabajo.
3. Para poder realizar este trabajo, se recomienda tener altos conocimientos sobre el sistema operativo Linux, debido a la necesidad de la implementación del entorno web y lenguajes de programación.
4. Para la realización de futuros trabajos similares, se puede usar otros lenguajes de programación como “nodejs” y “python”, que son de mayor facilidad para el programador y también tiene un alto soporte de comunidad y librerías específicas.

## BIBLIOGRAFÍA

- [1] ActiveXperts software, SMPP Introduction [http://www.activexperts.com/sms-component/smpp-specifications/introduction/#smpp\\_overview](http://www.activexperts.com/sms-component/smpp-specifications/introduction/#smpp_overview), fecha de consulta noviembre 2014.
- [2] Solares Juan, Desarrollo de un diccionario de la lengua española en línea accesado vía mensajes cortos (sms) en la red celular GSM de COMCEL (TIGO), [http://biblioteca.usac.edu.gt/tesis/08/08\\_0346\\_EO.pdf](http://biblioteca.usac.edu.gt/tesis/08/08_0346_EO.pdf), fecha de consulta noviembre 2014, pág. 58.
- [3] Computer Información, Especificaciones para el protocolo SMPP, [http://ordenador.wingwit.com/Redes/other-computer-networking/78036.html#.VNltj\\_mG9zY](http://ordenador.wingwit.com/Redes/other-computer-networking/78036.html#.VNltj_mG9zY), fecha de consulta noviembre 2014.
- [4] lasprovincias.es, Acision es reconocido como el número uno del mundo en servicio de mensajería, <http://www.lasprovincias.es/20100209/mas-actualidad/tecnologia/acision-reconocido-como-numero-201002091906.html>, fecha de consulta noviembre 2014.
- [5] Protocolo SMPP, [http://infotelecommil.webcindario.com/manuales/SMPP\\_Intro.pdf](http://infotelecommil.webcindario.com/manuales/SMPP_Intro.pdf), fecha de consulta noviembre 2014.

[6] SMPP Developers Forum, Short Message Peer to Peer Protocol Specification v3.4, <http://www.smstrade.de/pdf/smpp.pdf>, fecha de consulta noviembre 2014.

[7] Logix Mobile, Whatis SMSC?, <http://www.logixmobile.com/faq/show.asp?catid=1&faqid=4>, fecha de consulta diciembre 2014.

[8] Solares Juan, Desarrollo de un diccionario de la lengua española en línea accesado vía mensajes cortos (sms) en la red celular GSM de COMCEL (TIGO), [http://biblioteca.usac.edu.gt/tesis/08/08\\_0346\\_EO.pdf](http://biblioteca.usac.edu.gt/tesis/08/08_0346_EO.pdf), fecha de consulta diciembre 2014, pág. 59.

[9] ActiveXperts software, SMPP SessionDescription, [http://www.activexperts.com/sms-component/smpp-specifications/overview/#smpp\\_session\\_description](http://www.activexperts.com/sms-component/smpp-specifications/overview/#smpp_session_description), fecha de consulta diciembre 2014.

[10] ActiveXperts software, SMPP Protocol Definition, [http://www.activexperts.com/sms-component/smpp-specifications/overview/#smpp\\_protocol\\_definition](http://www.activexperts.com/sms-component/smpp-specifications/overview/#smpp_protocol_definition), fecha de consulta diciembre 2014.

[11] ActiveXperts software, SMPP PDU Definition, <http://www.activexperts.com/sms-component/smpp-specifications/smpp-pdu-definition/>, fecha de consulta diciembre 2014.

[12] ActiveXperts software, SMPP PDU Format – Overview, [http://www.activexperts.com/sms-component/smpp-specifications/pdu-type-format-definitions/#smpp\\_pdu\\_format](http://www.activexperts.com/sms-component/smpp-specifications/pdu-type-format-definitions/#smpp_pdu_format), fecha de consulta diciembre 2014.

[13] Nuruddin Ashr, SMPP, <http://es.slideshare.net/uudashr/smpp-presentation>, fecha de consulta diciembre 2014.

[14] AddPac Technology, SMPP (Short Message Peer to Peer) Protocol, [http://www.google.com.ec/url?sa=t&rct=j&q=&esrc=s&source=web&cd=23&cad=rja&uact=8&ved=0CGgQFjAMOAo&url=http%3A%2F%2Fwww.addpac.com%2Faddpac\\_eng%2Fdown.php%3Ffile%3D663\\_f16.pdf&ei=jo9uVJf2JcmbgwS4mYTYDw&usg=AFQjCNGT31qCB\\_koVoK0JxCD9avvPUuH5g&bvm=bv.80185997,d.eXY](http://www.google.com.ec/url?sa=t&rct=j&q=&esrc=s&source=web&cd=23&cad=rja&uact=8&ved=0CGgQFjAMOAo&url=http%3A%2F%2Fwww.addpac.com%2Faddpac_eng%2Fdown.php%3Ffile%3D663_f16.pdf&ei=jo9uVJf2JcmbgwS4mYTYDw&usg=AFQjCNGT31qCB_koVoK0JxCD9avvPUuH5g&bvm=bv.80185997,d.eXY), fecha de consulta diciembre 2014.

[15] ActiveXperts software, SMPP Error Codes, <http://www.activexperts.com/activsms/sms/smpperrorcodes/>, fecha de consulta diciembre 2014.

[16] LinuxZone, Ubuntu, <http://linuxzone.es/distribuciones-principales/ubuntu/>, fecha de consulta enero 2015.

[17] CPAN, Net::SMPP, <http://search.cpan.org/~sampo/Net-SMPP/SMPP.pm>, fecha de consulta enero 2015.

[18] Sagástegui Lescano Walter, Máquinas virtuales (VMWare, Virtual PC, Sandbox). Qué son y para que sirven. (DV00402A), [http://www.aprenderaprogramar.com/index.php?option=com\\_attachments&task=download&id=139](http://www.aprenderaprogramar.com/index.php?option=com_attachments&task=download&id=139), fecha de consulta febrero 2015.

## ANEXO

### Command\_id

<b>Campo command_id</b>	<b>valor en hexadecimal</b>
generic_nack	0x80000000
bind_receiver	0x00000001
bind_receiver_resp	0x80000001
bind_transmitter	0x00000002
bind_transmitter_resp	0x80000002
query_sm	0x00000003
query_sm_resp	0x80000003
submit_sm	0x00000004
submit_sm_resp	0x80000004
deliver_sm	0x00000005
deliver_sm_resp	0x80000005
unbind	0x00000006
unbind_resp	0x80000006
replace_sm	0x00000007
replace_sm_resp	0x80000007
cancel_sm	0x00000008
cancel_sm_resp	0x80000008
bind_transceiver	0x00000009
bind_transceiver_resp	0x80000009
Reserved	0x0000000A 0x8000000A
outbind	0x0000000B
Reserved	0x0000000C – 0x00000014 0x8000000B – 0x80000014
enquire_link	0x00000015
enquire_link_resp	0x80000015
Reserved	0x00000016 – 0x00000020 0x80000016 – 0x80000020
submit_multi	0x00000021
submit_multi_resp	0x80000021
Reserved	0x00000022 – 0x000000FF 0x80000022 – 0x800000FF
Reserved	0x00000100 0x80000100

Reserved	0x00000101 0x80000101
alert_notification	0x00000102
Reserved	0x80000102
data_sm	0x00000103
data_sm_resp	0x80000103
Reserved for SMPP extension	0x00000104 – 0x0000FFFF 0x80000104 – 0x8000FFFF
Reserved	0x00010000 – 0x000101FF 0x80010000 – 0x800101FF
Reserved for SMSC vendor	0x00010200 – 0x000102FF 0x80010200 – 0x800102FF
Reserved	0x00010300 – 0xFFFFFFFF

### Command\_status

Tipo de Error	Valor en Hex	Descripción
ESME_ROK	0x00000000	No hay error
ESME_RINVMGLEN	0x00000001	Longitud del mensaje inválido
ESME_RINVCMDLEN	0x00000002	Command Length es inválido
ESME_RINVCMDID	0x00000003	Command ID es inválido
ESME_RINVBNDSTS	0x00000004	Bind incorrecto por un comando dado
ESME_RALYBND	0x00000005	ESME ya en estado Bound
ESME_RINVPRTFLG	0x00000006	Bandera de prioridad inválida
ESME_RINVREGDLVFLG	0x00000007	Inválido registro de bandera entregada
ESME_RSYSERR	0x00000008	Error del sistema
Reserved	0x00000009	Reservado
ESME_RINVSRCADR	0x0000000A	Dirección de origen inválido
ESME_RINVDSTADR	0x0000000B	Dirección de destino inválido
ESME_RINVMSGID	0x0000000C	Message ID es inválido
ESME_RBINDFAIL	0x0000000D	Fracaso de Bind
ESME_RINVPASWD	0x0000000E	Clave inválida
ESME_RINVSYSID	0x0000000F	System ID es inválido
Reserved	0x00000010	Reservado
ESME_RCANCELFAIL	0x00000011	Fracaso de Cancel SM
Reserved	0x00000012	Reservado
ESME_RREPLACEFAIL	0x00000013	Fracaso de Replace SM
ESME_RMSGQFUL	0x00000014	Cola de mensajes lleno



ESME_RINVSERTYP	0x00000015	Tipo de servicio inválido
Reserved	0x00000016 - 0x00000032	Reservado
ESME_RINVNUMDESTS	0x00000033	Número de destinos inválido
ESME_RINVDLNAME	0x00000034	Nombre de lista de distribución inválidos
Reserved	0x00000035 - 0x0000003F	Reservado
ESME_RINVDESTFLAG	0x00000040	Bandera de destino inválido (submit_multi)
Reserved	0x00000041	Reservado
ESME_RINVSUBREP	0x00000042	Solicitud de submit con replace inválido
ESME_RINVESMCLASS	0x00000043	Esm_class, campo data inválido
ESME_RCNTSUBDL	0x00000044	Lista de distribución no puede enviarse
ESME_RSUBMITFAIL	0x00000045	Fracaso de submit_sm o submit_multi
Reserved	0x00000046 - 0x00000047	Reservado
ESME_RINVSRCNPI	0x00000048	Dirección de origen TON inválido
ESME_RINVSRCNPI	0x00000049	Dirección de origen NPI inválido
ESME_RINVDSTTON	0x00000050	Dirección de destino TON inválido
ESME_RINVDSTNPI	0x00000051	Dirección de destino NPI inválido
Reserved	0x00000052	Reservado
ESME_RINVSYSTYP	0x00000053	Campo system_type inválido
ESME_RINVREPFLAG	0x00000054	Bandera replace_if_present inválido
ESME_RINVNUMMSGGS	0x00000055	Número de mensajes inválido
Reserved	0x00000056 - 0x00000057	Reservado
ESME_RTHROTTLED	0x00000058	Error Throttling (El ESME ha superado los límites de mensajes permitidos)
Reserved	0x00000059 - 0x00000060	Reservado
ESME_RINVSCHED	0x00000061	Tiempo de entrega programado inválido
ESME_RINVEXPIRY	0x00000062	Período de validez del mensaje inválido (tiempo de expiración)
ESME_RINVDFTMSGID	0x00000063	Mensaje predefinido inválido o no encontrado

ESME_RX_T_APPN	0x00000064	ESME Receiver temporalmente en código de error
ESME_RX_P_APPN	0x00000065	ESME Receiver permanentemente en código de error
ESME_RX_R_APPN	0x00000066	ESME Receiver rechaza mensaje de código de error
ESME_RQUERYFAIL	0x00000067	Fracaso de query_sm
Reserved	0x00000068 - 0x000000BF	Reservado
ESME_RINVOPTPARSTREAM	0x000000C0	Error en la parte opcional del cuerpo PDU
ESME_ROPTPARNOTALLWD	0x000000C1	Parámetros opcionales no permitidos
ESME_RINVPARLEN	0x000000C2	Parámetro Length inválido
ESME_RMISSINGOPTPARAM	0x000000C3	Parámetro opcional esperado que falta
ESME_RINVOPTPARAMVAL	0x000000C4	Valor del parámetro opcional inválido
Reserved	0x000000C5 - 0x000000FD	Reservado
ESME_RDELIVERYFAILURE	0x000000FE	Fallo en la entrega (usado por data_sm_resp)
ESME_RUNKNOWNERR	0x000000FF	Error desconocido
Reservedfor SMPP extension	0x00000100 - 0x000003FF	Reservado para extensión SMPP
Reservedfor SMSC vendor	0x00000400 - 0x000004FF	Reservado para errores determinados por el proveedor SMSC
Reserved	0x00000500 - 0xFFFFFFFF	Reservado

### Submit\_sm y Delivery\_sm

Cuerpo PDU SMPP - Parámetros submit_sm y delivery_sm		
Nombre del Parámetro	Tamaño en octetos	Descripción
service_type	Varía max6	Identifica el servicio de aplicación SMS asociado con el mensaje. Por defecto se pone en NULL en la configuración SMSC.
source_addr_ton	1	TON para dirección origen. Si no se conoce se pone en NULL.

source_addr_npi	1	NPI para dirección origen. Si no se conoce se pone en NULL.
source_addr	Varía max 21	Dirección SME quien origina ese mensaje. Si no se conoce se pone en NULL.
dest_addr_ton	1	TON para destino.
dest_addr_npi	1	NPI para destino.
destination_addr	Varía max 21	Dirección destino del mensaje corto.
esm_class	1	Indica el modo de mensaje y tipo de mensaje.
protocol_id	1	Identificador de protocolo.
priority_flag	1	Designa el nivel de prioridad del mensaje.
schedule_delivery_time	1 o 17	El mensaje corto es programado por la SMSC para la entrega. Se establece en NULL para la inmediata entrega de mensajes. Este campo no es usado por delivery_sm y se debe poner en NULL en tal caso.
validity_period	1 o 17	Indica el período de validez del mensaje. Este campo no es usado por delivery_sm y se debe poner en NULL en tal caso.
registered_delivery	1	Indica si un SMSC delivery receipt o un SME acknowledgement es requerido. Con delivery_sm solo indica si un ESME acknowledgement es requerido.
replace_if_present_flag	1	Bandera que indica si un mensaje enviado debe reemplazar a un mensaje existente. Este campo no es usado por delivery_sm y se debe poner en NULL en tal caso.
data_coding	1	Define el esquema de codificación del mensaje corto.
sm_default_msg_id	1	Indica el mensaje corto al enviar una lista predefinida de mensajes cortos almacenados en la SMSC. Si no se utiliza un mensaje predefinido de la SMSC, se establece en NULL. Este campo no es usado por delivery_sm y se debe poner en NULL en tal caso.
sm_length	1	Longitud en octetos de short_message
short_message	Varía de 0 - 254	Se permite hasta 254 octetos de mensaje corto de datos del usuario.

<b>Cuerpo PDU SMPP - Parámetros submit_sm_resp y delivery_sm_resp</b>		
<b>Nombre del Parámetro</b>	<b>Tamaño en octetos</b>	<b>Descripción</b>
message_id	Max. 65 en submit y solo 1 con delivery.	Este campo contiene el message_id SMSC del mensaje enviado. Este campo no es usado por delivery_sm y se debe poner en NULL en tal caso.

### **Submit\_Multi**

<b>Cuerpo PDU SMPP - Parámetros submit_multi</b>		
<b>Nombre del Parámetro</b>	<b>Tamaño en octetos</b>	<b>Descripción</b>
service_type	Max. 6	Identifica el servicio de aplicación SMS asociado con el mensaje. Por defecto se pone en NULL en la configuración SMSC.
source_addr_ton	1	TON para dirección origen. Si no se conoce se pone en NULL.
source_addr_npi	1	NPI para dirección origen. Si no se conoce se pone en NULL.
source_addr	Max. 21	Dirección SME quien origina ese mensaje. Si no se conoce se pone en NULL.
number_of_dests	1	Números de direcciones destino. Indica el número de estructura de dest_address que se van a seguir.
dest_address	2-24	Contiene una o más (number_of_dests) direcciones SME y/o una lista de distribución.
esm_class	1	Indica el modo de mensaje y tipo de mensaje.
protocol_id	1	Identificador de protocolo.
priority_flag	1	Designa el nivel de prioridad del mensaje.
schedule_delivery_time	1 o 17	El mensaje corto es programado por la SMSC para la entrega. Se establece en NULL para la inmediata entrega de mensajes.
validity_period	1 o 17	Indica el período de validez del mensaje. Se establece en NULL para solicitar el periodo de validez por defecto del SMSC.

registered_delivery	1	Indica si un SMSC delivery receipt o un SME acknowledgement es requerido.
replace_if_present_flag	1	Bandera que indica si un mensaje enviado debe reemplazar a un mensaje existente.
data_coding	1	Define el esquema de codificación del mensaje corto.
sm_default_msg_id	1	Indica el mensaje corto al enviar una lista predefinida de mensajes cortos almacenados en la SMSC. Si no se utiliza un mensaje predefinido de la SMSC, se establece en NULL.
sm_length	1	Longitud en octetos de short_message
short_message	Varía de 0 - 254	Se permite hasta 254 octetos de mensaje corto de datos del usuario.

<b>Cuerpo PDU SMPP - Parámetros submit_multi_resp</b>		
<b>Nombre del Parámetro</b>	<b>Tamaño en octetos</b>	<b>Descripción</b>
message_id	Max. 65	Es el mensaje identificador SMSC del mensaje enviado.
no_unsuccess	1	Es el número de mensajes a direcciones SME destinos que fueron enviados sin éxito al SMSC.
unsuccess_sme	7-27	Contiene una o más (no_unsuccess) direcciones ESME o lista de distribución cual envío fue sin éxito.