



ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL

Facultad de Ingeniería en Electricidad y Computación

“ANÁLISIS E IMPLEMENTACIÓN EN EL LENGUAJE ABIERTO
PREPROCESADOR DE HIPERTEXTO (PHP) SOBRE LINUX DE UNA
CENTRAL DE CONMUTACIÓN MÓVIL (MSC) VIRTUAL PARA LA
INTERACCIÓN CON UN SISTEMA PREPAGO A TRAVÉS DE
SEÑALIZACIÓN DE APLICACIONES PERSONALIZADAS PARA REDES
MÓVILES DE LÓGICA MEJORADA (CAMEL) PARA LOS SERVICIOS DE
VOZ DE ABONADOS PREPAGOS.”

INFORME DE MATERIA DE GRADUACIÓN

Previo a la obtención del Título de:

INGENIERO ELECTRÓNICO Y TELECOMUNICACIONES

Presentado por:

Kléber Andrés Mora Guevara

Guayaquil – Ecuador

2015

AGRADECIMIENTO

Estos años de aprendizaje y crecimiento personal son frutos de dedicación y sacrificio, pero sobretodo fueron capaces de realizarse por todas las bendiciones de Dios derramadas sobre mí; la más grande, haberme permitido contar con el apoyo de mis padres y su tan abnegado amor que me permitió alcanzar los objetivos deseados y a los cuales manifiesto mi más profunda gratitud.

DEDICATORIA

Este trabajo no hubiese sido posible con solamente un aporte individual, el tiempo implementado aquí son momentos ausentes de mi familia y amigos, los cuales de cierto modo aportaron en el objetivo y a los cuales dedico este logro.

Pero en especial se lo ofrezco a mis padres: Eduardo Mora Wilmot, por su apoyo moral y monetario en este largo camino y a Cecilia Guevara Salinas (+), que a pesar de su ausencia física, su memoria y sus enseñanzas siempre fueron un pilar fundamental en mi carrera universitaria.

TRIBUNAL DE SUSTENTACIÓN



Ing. José Miguel Menéndez, MG.

PROFESOR DE LA MATERIA DE
GRADUACIÓN



Ing. Washington Medina, MG.

PROFESOR DELEGADO POR LA
UNIDAD ACADÉMICA

RESUMEN

El siguiente informe recopila, de forma amplia y detallada, la investigación concerniente a una Central de Conmutación Móvil virtual (MSC por sus siglas en inglés), de una operadora de servicios móviles; la cual trabaja e interactúa dentro de una Red de Telefonía Celular con otros componentes.

Su diseño e implementación se realiza sobre la plataforma del Sistema Operativo LINUX, con lenguaje de Preprocesador de Hipertexto (PHP), el cual brinda facilidades para crear el emulador de la Central y dar lugar a las líneas de códigos y otras aplicaciones como bases de datos y conmutación de archivos de los demás elementos.

Esta MSC opera bajo el protocolo de Aplicaciones a Medida para la Red Móvil de Lógica Mejorada (CAMEL por sus siglas en inglés), que se ha utilizado para el envío y recepción de requerimientos del Sistema de Señalización SS7 en la parte de la comunicación con el sistema Prepago (PP), en lo que respecta a llamadas entre abonados o servicios de consulta.

ÍNDICE GENERAL

AGRADECIMIENTO	II
DEDICATORIA	III
TRIBUNAL DE SUSTENTACIÓN	IV
DECLARACIÓN EXPRESA	V
RESUMEN	VI
ÍNDICE GENERAL	VII
ABREVIATURAS Y SIMBOLOGÍA	X
ÍNDICE DE FIGURAS	XIII
INTRODUCCIÓN	XVI
CAPÍTULO 1	1
1 DESCRIPCIÓN GENERAL	1
1.1 Antecedentes del SS7	1
1.2 Justificación del SS7	3
1.3 Descripción del Proyecto	4
1.3.1 Objetivo General	5
1.3.2 Objetivos Específicos	5
1.4 Alcance	6
1.5 Limitaciones	7
1.6 Metodología	8
CAPÍTULO 2	9

2	FUNDAMENTOS TEÓRICOS	9
2.1	SS7	9
2.1.1	Arquitectura de SS7	9
2.2	Sistema de Señalización SS7	11
2.2.1	Capas del Modelo SS7	12
2.3	Central de Conmutación Móvil (MSC)	14
2.3.1	Partes de un Subsistema de Conmutación de Red	15
2.4	PROTOCOLO CAMEL	17
2.4.1	CAMEL Fase 1	17
2.4.2	CAMEL Fase 2	18
2.4.3	Nodos de llamadas de CAMEL Fase 2	19
2.5	SISTEMAS OPERATIVO	21
2.5.1	LINUX	21
2.5.2	Lenguaje PHP	22
2.6	Software <i>Wireshark</i>	23
	CAPÍTULO 3	25
3	DESCRIPCIÓN DEL PROYECTO	25
3.1	Software del Sistema	25
3.2	Hardware del Sistema	26
3.3	Topología de la Red	27
3.4	Requisitos para la Implementación	28

3.4.1	Ubuntu.....	28
3.4.2	XAMPP (APACHE).....	35
3.4.3	NETBEANS	38
3.4.4	Acerca de POSTMAN.	40
3.4.5	PHPMYADMIN	41
3.5	Funcionamiento del Emulador.....	42
3.5.1	Mensajes del protocolo CAMEL que operan sobre la MSC	42
3.5.2	Análisis del Funcionamiento	53
3.5.3	Mensajes de Programación y líneas de código del Emulador MSC	57
3.6	Síntesis de Implementación y Funcionamiento	62
	CAPÍTULO 4.....	64
4	ESTADO FINAL DEL PROYECTO.....	64
4.1	Resultados Obtenidos	64
4.2	Análisis de Resultados	67
4.2.1	Topología de la Implementación Total del Proyecto.....	67
4.2.2	Interfaz Gráfica de Módulos Implementados	71
4.3	Discusión	76
	CONCLUSIONES Y RECOMENDACIONES.....	78
	BIBLIOGRAFÍA	82

ABREVIATURAS Y SIMBOLOGÍA

ACR	Reporte de Aplicación de Carga
ANSI	Instituto Americano de Normas Nacionales
BCSM	Modelo Básico de Estado de Llamada
BTS	Estación Base del Transceptor
CAMEL	Aplicaciones a Medida para la Red Móvil de Lógica Mejorada
CAP	Parte de Aplicaciones CAMEL
CH	Canal
CON	Mensaje Conectar
DPC	Destino de Código de Punto
ETC	Conexión de Estabilización Temporal
ERBCSM	Reporte de Evento BCSM
GSM	Sistema Global Para Comunicaciones Móviles
GPRS	Paquete General de Servicio de Radio
HLR	Registros de Usuario Local
HTTP	Transferencia de Hipertexto
IDP	Punto de Detección Inicial
IN	Red Inteligente
INAP	Parte de Aplicaciones de Red Inteligente
ISDN	Servicios Integrados de la Red Digital
ISUP	Parte de Usuario ISDN

ITU-T	Unión Internacional de Telecomunicaciones –Sector de las Telecomunicaciones
MAP	Parte de Aplicación Móvil
MSISDN	Estación Móvil de Servicios Integrados de una Red Celular
MTP-1	Parte de Transferencia de mensaje Nivel 1
MTP-2	Parte de Transferencia de mensaje Nivel 2
MTP-3	Parte de Transferencia de mensaje Nivel 3
OMAP	Operaciones, Mantenimiento y Administración de Piezas
OPC	Código de Punto de Origen
PP	Sistema Prepago
PSTN	Red de Telefonía Pública Conmutada
REL	Mensaje de Liberación
RLC	Llamada Liberada
SCF	Función de Servicios de Conmutación
SCN	Red de Conmutación de Circuitos
SCP	Punto de Control de Señalización
SCCP	Señalización de Control de Conexión de Pieza
SCTP	Transaccionales Capacidades de Aplicación de la Parte
SMS	Servicio de Mensajes Cortos
SRF	Función de Recursos Especializados
SSF	Función de Conmutación de Servicios

SS7	Sistema de Señalización por Canal Común No.7
SSP	Punto de Señalización de Servicio
STP	Punto de Señalización de Transporte
TCAP	Parte de Aplicación de Capacidades de Transacciones
UMTS	Sistema Universal de Telecomunicaciones Móviles
VLR	Registro de Usuarios Visitantes

ÍNDICE DE FIGURAS

Figura 2.1 Estructura de Red SS7	10
Figura 2.2 Capas de Modelo SS7	12
Figura 2.3 Elementos de Interconexión de una MSC.....	15
Figura 2.4 Arquitectura de CAMEL FASE 1	18
Figura 2.5 Arquitectura de CAMEL Fase 2	19
Figura 2.6 Modelo de Escritorio en Software LINUX.....	22
Figura 2.7 Captura de mensaje IDP.....	23
Figura 2.8 Captura de registro de segundos hablados	24
Figura 2.9 Captura de cierre de llamadas por falta de saldo	24
Figura 2.10 Desconexión por parte de número llamante	24
Figura 3.1 Máquina utilizada para implementar la MSC	26
Figura 3.2 Conexión de Red para la señalización.....	28
Figura 3.3 Ventana de selección de idioma [17]	30
Figura 3.4 Ventana de descarga [17].....	31
Figura 3.5 Ventana de tipo de instalación [17].....	31
Figura 3.6 Ventana de partición de disco [17].....	32
Figura 3.7 Tamaño de nuevo disco [17].....	33
Figura 3.8 Distribución del teclado [17].....	33
Figura 3.9 Configuración personal [17]	34
Figura 3.10 Ventana de acceso al sistema [17]	34
Figura 3.11 Comando de instalación de XAMPP [18].....	35

Figura 3.12 Comando de ejecución de XAMPP [18].....	36
Figura 3.13 Asistente de instalación [18]	36
Figura 3.14 Selección de componentes [18].....	36
Figura 3.15 Ventana de Instalación [18]	37
Figura 3.16 Ventana de Inicio de XAMPP [18].....	37
Figura 3.17 Ejecución de Servidores [18]	38
Figura 3.18 Descarga de Java [19]	39
Figura 3.19 Selección de tecnologías compatibles [19]	39
Figura 3.20 Comandos de ejecución de NETBEANS [19]	40
Figura 3.21 Ventana de servidor POSTMAN [20]	40
Figura 3.22 Ventana Principal del servicios PHPMyAdmin [21]	41
Figura 3.23 Atributos del mensaje IDP	43
Figura 3.24 Atributos del mensaje AChR.....	46
Figura 3.25 Atributos del mensaje ERBCSM	47
Figura 3.26 Mensaje de Set Up	48
Figura 3.27 Mensajes con el Sistema Prepago.....	50
Figura 3.28 Mensaje con Abonado B.....	51
Figura 3.29 Mensaje ETC	52
Figura 3.30 Caso típico de llamada entre abonados.....	54
Figura 3.31 Consulta de saldo o activación de tarjeta prepago	56
Figura 3.32 Primer mensaje del emulador MSC	58
Figura 3.33 Primera parte de código Set_Up2.....	58

Figura 3.34 Segunda parte de código Set_Up2.....	59
Figura 3.35 Tercera parte del código Set_Up2	59
Figura 3.36 Cuarta parte del código Set_Up2.....	60
Figura 3.37 Líneas de código de Set_Up3.....	61
Figura 3.38 Código programado de segundos acumulados en llamada	61
Figura 3.39 Reporte de evento de llamada por parte de la MSC	62
Figura 4.1 Fichero registro.log	65
Figura 4.2 Mensaje recibido por la MSC proveniente del abonado.....	65
Figura 4.3 Mensaje de requerimiento al número de servicio *282	66
Figura 4.4 Mensaje que recibe la MSC de autorización de llamadas	66
Figura 4.5 Mensajes ACR y EBCSM hacia el sistema prepago.....	67
Figura 4.6 Routers para la conexión inalámbrica.....	68
Figura 4.7 Switch con los puertos LAN utilizados	69
Figura 4.8 Topología Final de Red.....	70
Figura 4.9 Interfaz gráfica de la Central.....	71
Figura 4.10 Presentación de Interfaz	72
Figura 4.11 Captura 1 Tramo de llamada	73
Figura 4.12 Captura 2 Tramo de llamada	74
Figura 4.13 Captura 3 Tramo de llamada	74
Figura 4.14 Parámetros vistos desde la aplicación.....	75

INTRODUCCIÓN

En el transcurso de los años, el ser humano se ha visto en la imperiosa necesidad de construir métodos y canales de comunicación cada vez más eficientes, que alberguen una amalgama de prioridades, cubriendo así todo tipo de necesidades en las que se han tomado en cuenta factores como las grandes distancias, calidad en la comunicación o el tiempo de duración.

Con el avance en el campo tecnológico se ha podido desarrollar las tan nombradas y utilizadas Redes Inteligentes (IN por sus siglas en inglés) para poder cubrir la demanda que existe en las Telecomunicaciones, con estándares globales y sistemas de señalización.

En esta área entra en acción el Sistema de Señalización por canal común No. 7 (SS7), que es muy utilizado hoy en día en las Redes Telefónicas, por los diferentes servicios que presta y por su lógica de comunicación para la fácil conexión de Redes Inteligentes entre sí.

CAPÍTULO 1

1 DESCRIPCIÓN GENERAL

1.1 Antecedentes del SS7

El Sistema de Señalización por canal común no. 7 o SS7 (por sus siglas en inglés), fue desarrollado por el Sector de Normalización de Telecomunicaciones (UIT-T), que es uno de los tres sectores de la Unión Internacional de Telecomunicaciones (UIT), y las grandes compañías telefónicas desde el año 1980[1]

La idea de introducir un nuevo sistema de señalización surge de la necesidad de sustituir a sus sistemas precedentes tales como el SS5 o SS6, también podríamos acotar al R1 y al R2, aunque en el caso de R1 y R2 se sigue utilizando en algunas Centrales en pocos países, pero con los nodos de nueva generación, su uso se ha vuelto casi obsoleto.

Es importante hacer mención al proceso evolutivo que han tenido las redes de telefonía desde sus inicios, como veremos a continuación:

Desde finales del siglo XIX ya existía lo que sería la primera red de telefonía, que consistía en líneas telefónicas conectadas a un Conmutador (Central). Solo podía haber comunicación entre las líneas que había en un mismo Conmutador.

En 1934 entra al mercado el sistema de señalización SS1, de bajo nivel y monofrecuente, solo podía utilizarse la banda de 500 o 1000 Hz, con un rango de 20 Hz para llamadas.

El SS2 fue desarrollado en 1938 con una señalización de servicio direccional semiautomático igual que el SS3 que lo reemplazaría en 1954.

En el mismo año que apareciera el SS3, entra a sucederlo el SS4 que sería adoptado como el sistema más utilizado por toda Europa en el tráfico internacional. Esta señalización ya contaba con servicio direccional automático y semiautomático.

El SS5, con sus variaciones de R1 y R2, aparecerían en años posteriores y su uso sería fundamental para las conexiones internacionales por cable y por satélite, brindando mayores facilidades que las anteriores. Estos tipos de señalización se denominan “señalización dentro de banda”; donde la información del establecimiento de la llamada se producía en canales portadores (líneas telefónicas) pero había problemas de seguridad, ya que hubo personas que pudieron manejar la señalización de red para uso y conveniencia propia.

El SS6 es diseñado en 1977 y se convierte en el primer sistema de señalización por canal común, que sería reemplazado por el SS7 3 años después. Se los conoce así porque usan un canal de señalización separado de los canales portadores de datos de usuario; esto requiere un canal exclusivamente para la señalización, pero por la rapidez de transmisión del SS7, existe disponibilidad de canales frecuentemente[2]

1.2 Justificación del SS7

Los sistemas de señalización se refieren al intercambio de información entre los componentes de llamadas, los cuales se requieren para entregar y mantener servicio. El SS7 mejora la comunicación entre dichos componentes porque provee una estructura universal para señalización de redes de telefonía, mensajería, interconexión y mantenimiento de redes. La información es transportada en forma de mensajes. Además se ocupa del establecimiento de una llamada, intercambio de información de usuario, enrutamiento de llamada, estructuras de abonados diferentes, y soporta servicios de IN.

Es muy importante porque soporta la señalización de abonados telefónicos analógicos y digitales; y puede aplicarse a todas las redes de servicios nacionales e internacionales.

El SS7 posee una capacidad de mejoramiento y en un futuro se estará frente a un escenario diferente de comunicaciones por el papel

estratégico de la señalización, en lo que se refiere a la transmisión de la información por el incremento en el intercambio de tráfico de señalización[3]

1.3 Descripción del Proyecto

Este proyecto consiste en el diseño e implementación de CAMEL, que opera en la capa de aplicación del Sistema de Señalización por canal común No.7. El propósito base de los sistemas de señalización es crear lenguajes técnicos para el intercambio de información entre 2 terminales de una red telefónica, pudiendo ser esta información: establecimiento y finalización de llamadas, la tarificación del sistema prepago.

Lo que se busca en este documento es repasar todos los aspectos del tópico y desarrollarlos de una manera didáctica y entendible para lectores interesados en Aplicaciones SS7.

El protocolo que utilizará este proyecto será Aplicaciones Personalizadas para la Red Móvil de Lógica Mejorada, en su fase 2, que permitirá interactuar a una Central de Conmutación Móvil (MSC) virtual, mediante mensajes de este protocolo, con servicios de voz de abonados prepago. CAMEL fase 2 es una extensión de la IN, y define los servicios básicos del control de llamadas, la detección de los números prepagos y su inmediata autorización para establecer la llamada; por lo cual se busca

que se mantenga el tráfico de mensajes de señalización entre el MSC virtual con el sistema prepago durante la interacción de los abonados.

Además, se trata de indicar los beneficios que obtienen las operadoras de telefonía móvil al trabajar con el protocolo CAMEL, como el servicio *Roaming* que permite mantener una llamada en una red visitante pidiendo información a su red local.

Todo esto se hará sobre el lenguaje PHP en la plataforma Linux, que es un software robusto, confiable y que permite simplicidad y mejor forma de implementación de nuestro módulo MSC.

1.3.1 Objetivo General

La investigación y desarrollo de este proyecto tiene como meta a conseguir: la implementación de una Central de Conmutación Móvil (MSC) virtual en lenguaje Linux, para la interacción con un sistema prepago a través de señalización de Aplicaciones Personalizadas, para los servicios de voz de abonados prepagos.

1.3.2 Objetivos Específicos

- Determinar el alcance del protocolo de señalización CAMEL en una Red telefónica.
- Describir la funcionabilidad del protocolo de señalización CAMEL fase 2 de una MSC.
- Detallar los servicios que ofrecen las operadoras móviles gracias a sus sistemas prepago basados en CAMEL.

- Determinar el alcance de una MSC en una operadora celular.

1.4 Alcance

La investigación e implementación de una Central de Conmutación Móvil tiene como propósito el manejo y utilización de mensajes observados en el protocolo CAMEL, que sirven para que dicha MSC desarrolle un enlace de comunicación con el sistema prepago en tiempo real; ya que se espera que sea capaz de discernir entre ciertos escenarios que se producen en una llamada telefónica. Dichos mensajes serán correctamente detallados con el fin de proporcionar una idea de la transferencia de información.

El sistema operativo Ubuntu, que trabaja sobre la plataforma Linux, es el encargado de emular de forma correcta la Función de Conmutación de Servicios (SSF por sus siglas en inglés) de una Central, para lograr interactuar con la Función de Control de Servicios (SCF) de la PP, con el fin de controlar lo consumido durante una llamada mediante señalización y su posterior cierre al término de tiempo disponible de su cuenta, notificaciones a las terminales celulares de conexión o desvío de llamadas, activación de códigos de tarjetas o consulta de disponibilidad de saldo

Además se busca simular todos los casos gracias a la programación de líneas de códigos y conmutación de archivos de los demás elementos de la Red de Telefonía implementada.

1.5 Limitaciones

A la hora de simular una llamada entre abonados, la interconexión de los celulares con la Central de Conmutación Móvil se limita a cierto tipo de notificaciones; ya que los terminales no se comunican de una manera directa con la MSC, si no que la información pasa por otros elementos de Red como Estaciones Bases o Estaciones Transceptoras; por esto los datos no se envían ni reciben completamente.

La MSC contiene muchos servicios y funciones que incluye a algunos tipos de protocolos, pero al remitirnos a CAMEL, solo se podrá invocar las más importantes en el proceso de una conexión entre la Central y un sistema Prepago, que permitan el flujo de información de la capa de Aplicación en la arquitectura SS7, para la llamada en proceso[4]

A la hora de la unión de los Módulos para el funcionamiento final del proyecto, la distancia entre ellos supone una restricción, porque la Red implementada requiere conexiones por puertos LAN, esto quiere decir que necesitan estar a centímetros del *switch*; y además comunicación vía inalámbrica, por lo que los *routers* no se pueden encontrar a más de 10 metros de separación.

1.6 Metodología

El enfoque que se busca dar a este proyecto es de tipo didáctico, basado en la Metodología de la Investigación, que referencia al conjunto de procedimientos racionales utilizados para alcanzar una gama de objetivos que rigen una investigación científica [5]. Bajo esta premisa se quiere indicar la forma en que una MSC debe trabajar correctamente para conectarse a los servicios prepagos mediante la señalización SS7 para CAMEL.

Guiándonos en el temario del presente trabajo, en el segundo capítulo se desarrolla los Fundamentos Teóricos donde hablamos de la función SSF de una MSC y hacemos mención del lenguaje abierto PHP sobre Linux.

En el capítulo siguiente se adentra a la constitución del proyecto propiamente dicho, con sus requisitos y la implementación de la MSC en el lenguaje, y su operación e interacción con el sistema de abonados de voz para prepago.

En el cuarto capítulo se procede al análisis de los datos obtenidos y su debida explicación de forma explícita, refiriéndose a toda la expectativa en lo que se trata de implementación.

Una vez realizado el análisis de los resultados obtenidos, se procede a incluir en el proyecto conclusiones a las que podemos llegar y acotaciones que sirvan en futuros trabajos con este tipo de señalización.

CAPÍTULO 2

2 FUNDAMENTOS TEÓRICOS

2.1 SS7

SS7 son las siglas de sistema de señalización No. 7, que es un lenguaje técnico que se encarga de la comunicación entre dos terminales móviles desde el establecimiento hasta la finalización de la llamada y es la más utilizada por las redes telefónicas mundiales en la actualidad.

2.1.1 Arquitectura de SS7

El SS7 ofrece distintos tipos de servicios desde que se establece una llamada telefónica, como el enrutamiento de la misma, el intercambio de información de los usuarios y la estructuración de los abonados, de igual forma en el envío de mensajes; por lo tanto

requiere que la red tenga un alto grado de redundancia, para el soporte de los canales de señalización.

En la figura mostrada a continuación se detalla los componentes principales en la Arquitectura de SS7 [6]

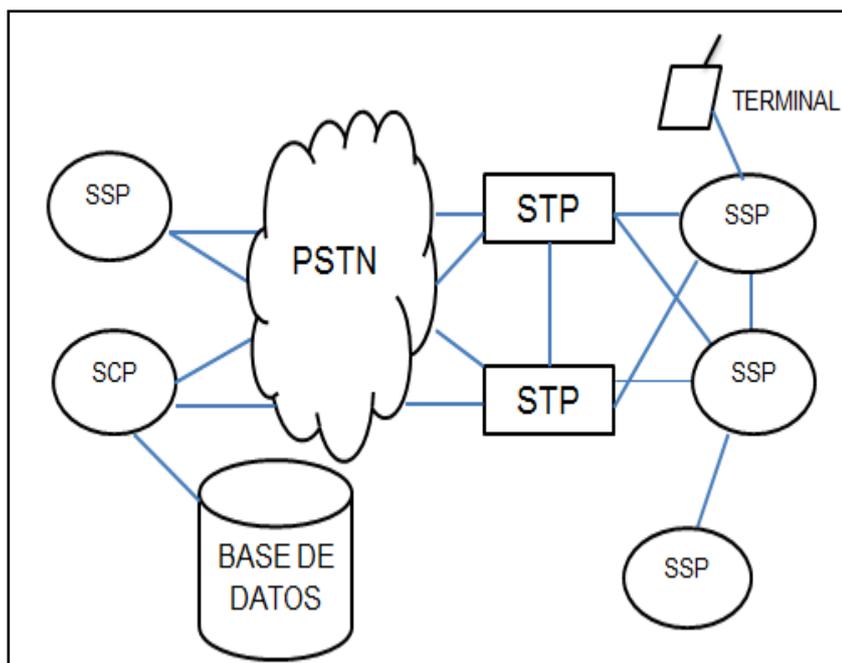


Figura 2.1 Estructura de Red SS7

En la figura 2.1 se muestran los elementos físicos que participan en una IN con señalización por canal común No. 7.

SSP (*Service Switching Point*): Este Punto de Conmutación de Servicio es la parte física donde comienza la conexión con la Red Inteligente y su ubicación es en la Central de Conmutación. De aquí se envía la respectiva información requerida de la llamada al SCP.

SCP (*Signal Control Point*): Aquí es donde se localiza la lógica del servicio, que sirve para identificar las características del perfil del usuario solicitante. Autoriza el establecimiento de conexión mediante algoritmos en tiempo real de las funciones del servicio, y finaliza la llamada según eventos notificados por el SSP.

STP (*Signal Transfer Point*): Este punto de la señalización sirve como una especie de puente, ya que al recibir el mensaje lo redirige hacia otro punto de enlace en La Arquitectura de Red; en este punto no se origina ni termina el mensaje[7]

2.2 Sistema de Señalización SS7

Un sistema de señalización provee la idea de un conjunto de funciones y protocolos que son establecidas para la generación de información e interacción de una Red de Telefonía Pública Conmutada y es utilizada, en mayor porcentaje, por las Redes Telefónicas mundiales. Su estandarización es global, definido por la Unión Internacional de Telecomunicaciones (ITU-T) y facilita altamente el manejo de la comunicación desde la aparición de la IN, y tiene como principal objetivo la conexión y finalización de las llamadas telefónicas[8]

El SS7 brinda un tipo de señalización por canal común, que maneja la información por enlaces diferentes a los que se maneja la voz en los enlaces de usuarios en las troncales Telefónicas. Los protocolos

funcionan en una Red, independiente a los circuitos de voz, de señalización constituida por puntos y canales sobre los cuales se conmutan los distintos mensajes de los protocolos existentes en la señalización.

Entre las facultades de este Sistema por Canal Común, es que se transporta información de la señalización de muchos canales de voz por un solo canal de datos que es directamente utilizado para este fin. Además proporciona una alta confiabilidad en la transmisión de datos, menos inconvenientes con el retardo de llamadas, más seguridad en el envío de mensajes y optimización de los canales de señalización.

2.2.1 Capas del Modelo SS7

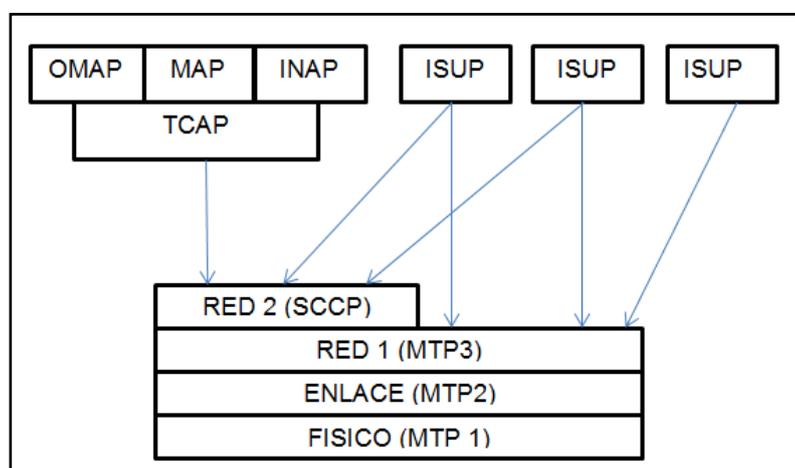


Figura 2.2 Capas de Modelo SS7

En la figura 2.2 se observan los distintos niveles del modelo SS7, que posee gran similitud con las capas del modelo OSI.

MTP Nivel 1

Este nivel se asemeja a la capa Física en el modelo OSI, y su función es el enlace de señalización, que son dos canales de transmisión digital entre dos puntos de señalización que llevan datos SS7.

MTP Nivel 2

Este nivel se considera la capa de Enlace en el modelo OSI, y se encarga del control para la estabilización de comunicación mediante mensajes de señalización.

MTP Nivel 3

Este nivel tiene las características de capa de Red en el modelo OSI, y es el encargado de la interfaz del Protocolo de Transferencia de Mensaje (MTP por sus siglas en ingles), y del re enrutamiento de los mensajes en la arquitectura de señalización.

Nivel 4: Capa de Aplicaciones

Transaccionales Capacidades de Aplicación de la Parte (TCAP), facilita la conexión a una base de datos externa.

Operaciones, Mantenimiento y Administración de Piezas (OMAP), utiliza las aplicaciones y funciones de control TCAP, a través de la red.

Aplicación Móvil de la Parte (MAP por sus siglas en inglés), se utiliza para compartir información sobre los abonados celulares entre distintas redes.

Señalización de Control de Conexión de Pieza (SCCP), es necesaria para el enrutamiento de mensajes TCAP a su base de datos correspondiente.

La Parte del Usuario de la Red Digital de Servicios Integrados (ISUP), apoya el llamamiento de telefonía básica de conexión o desconexión entre los puntos de señalización[9]

2.3 Central de Conmutación Móvil (MSC)

Una Central de Conmutación Móvil es un elemento de las Redes de Comunicaciones Móviles (GSM por sus siglas en inglés). Una MSC es una compleja central telefónica que se la considera el cerebro de la Red Celular, la cual proporciona conmutación de llamadas (establecimiento, enrutamiento y finalización), accesos en movilidad y servicios de Red GSM, entre los cuales se permite interactuar a los usuarios en lo que es transmisiones de voz y datos así como mensajes, para aquellos que se encuentran dentro del área del servicio; además de permitir la interconexión con otras MSC y PSTN.

Las Centrales de Conmutación Móvil forman parte del Subsistema de Conmutación de Red (NSS por sus siglas en inglés). En la figura 2.3 se muestra conexiones de la MSC con ciertos elementos de una NSS[10]

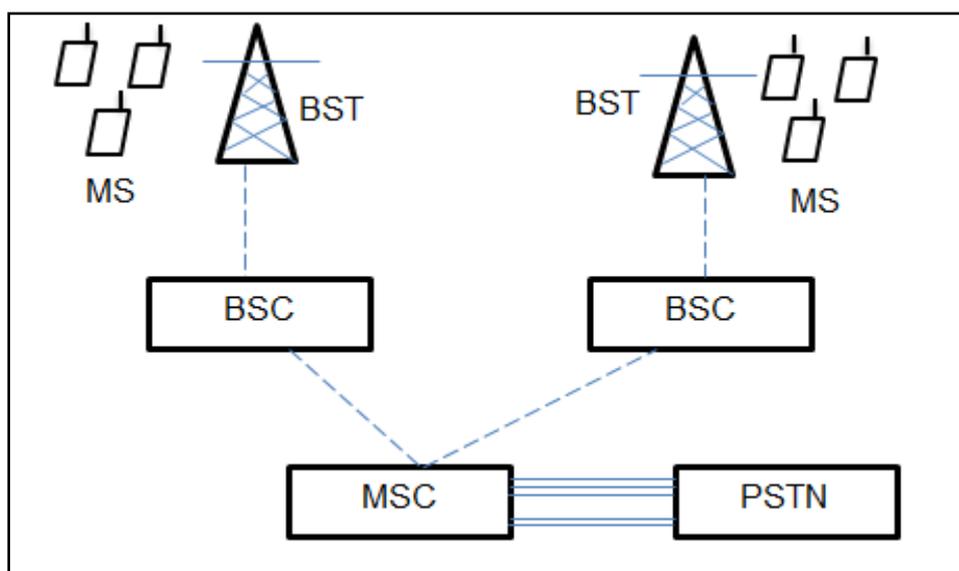


Figura 2.3 Elementos de Interconexión de una MSC

2.3.1 Partes de un Subsistema de Conmutación de Red

Estaciones Móviles de Usuarios (MS por sus siglas en inglés).

Provee servicios de voz, datos e imágenes. Cada estación móvil actúa transmitiendo o recibiendo la información.

Estación Radio Base (BTS)

Esta encargada de establecer el enlace radioeléctrico entre la MS y la Estación de Control del Servicio durante la comunicación.

Cada Estación Base calcula el número de móviles en un área de cobertura para proporcionar el servicio determinado.

Estación de Control Base (BSC)

Realiza funciones de gestión mantenimiento del servicio. Tiene la tarea de asignar estaciones base en un sector dentro del área de cobertura a las estaciones móviles que se mantienen en desplazamiento.

Además la MSC se conecta con otras redes como la PSTN, mientras las estaciones base se conectan a ella.

Otros elementos de la red GSM conectados a la MSC son el Registro de Usuarios Locales (HLR por sus siglas en inglés) y el Registro de Usuarios Visitantes (VLR por sus siglas en inglés) [11].

HLR

Es una base de datos que almacena la posición del usuario dentro de la red, además de sus características, entre ellas los servicios o el tipo terminal que es. Cada número telefónico está registrado en un HLR único.

VLR

Almacena información relativa a los abonados visitantes, esto significa que momentáneamente se crea un registro de un abonado en otra MSC.

2.4 PROTOCOLO CAMEL

Las Aplicaciones Personalizadas para Redes Móviles de Lógica Mejoradas, es un protocolo que encierra normas y funciones diseñadas para el control de servicios específicos dentro de una red GSM o UMTS; teniendo como idea principal la separación de los enlaces de conmutación y de servicios en una Red Inteligente.

CAMEL es desarrollado para ofrecer distintos servicios que no corresponden a los servicios estándares de un usuario dentro de una red celular, tales como servicios de tarificación prepago, servicios de datos, multimedia o de voz, traslación de números. Todos estos servicios de Red Inteligente se permiten y son llevados a cabo cuando el abonado está dentro de su propia red o cuando esta itinerante (*roaming*) fuera de su red [12]

2.4.1 CAMEL Fase 1

La Fase 1 representa una similitud a las IN, solamente que para identificarla como red GSM, se le antecede el prefijo gsm a las funciones de esta Fase. CAMEL Fase 1 está definida para servicios de llamadas básicas, aunque tiene como capacidades adicionales bloquear llamadas o establecer un modelo de estado básico de llamada (BCSM por sus siglas en inglés). Además

proporciona otras capacidades como definir eventos que suceden en el transcurso de una llamada como la conexión o desconexión, o si el abonado se encuentra ocupado o no responde; servicios que proporciona la entidad Función de Control de Servicio (gsmSCF) gracias a la comunicación con la gsmSSF y los registros de usuarios, como se aprecia en la figura 2.4.

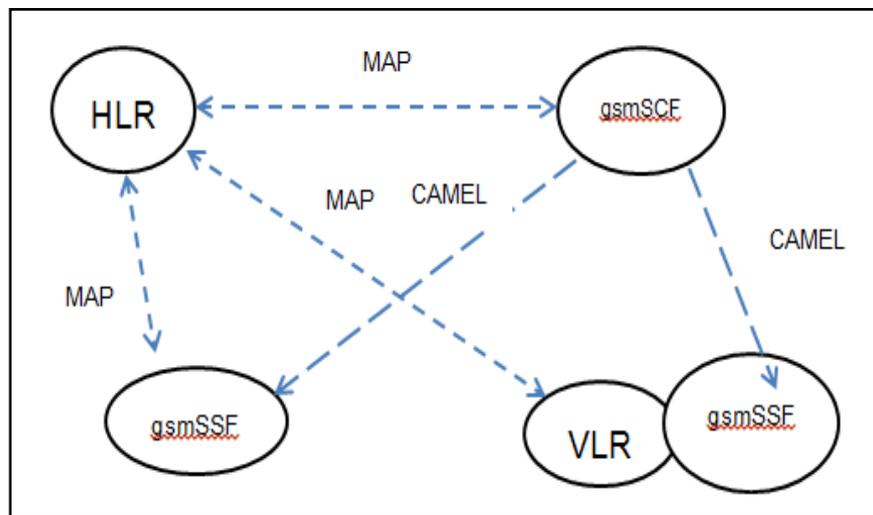


Figura 2.4 Arquitectura de CAMEL FASE 1

2.4.2 CAMEL Fase 2

La arquitectura de CAMEL Fase 2 es la más utilizada en las operadoras celulares a nivel mundial y está basada en CAMEL Fase 1. Aquí se permite la comunicación CAMEL entre las redes local y visitante para verificar la información del usuario en la entidad gsmSCF, así solo se transfiere señalización entre estas dos redes. Adicionalmente, en esta Fase, se incorpora la entidad

Función de Recursos Especializados (gsmSRF por sus siglas en inglés), que permite el reconocimiento de voz en la interacción con el abonado.

La Fase 2 también incluye otros servicios como puntos adicionales de detección de eventos, control en la duración de cada llamada y asesoramiento al abonado o la información de las entidades de servicios en los registros de llamadas, toda esta información llega primero a la Central que es la que tiene comunicación directa con el móvil, como se indica en la figura 2.5 [13]

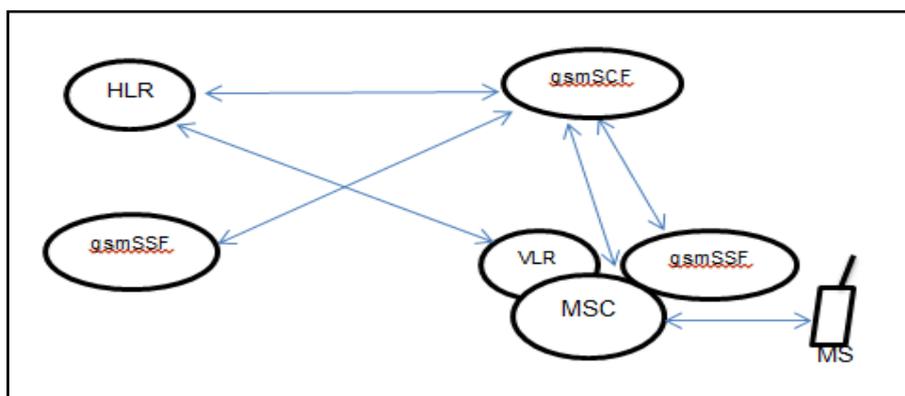


Figura 2.5 Arquitectura de CAMEL Fase 2

2.4.3 Nodos de Llamadas de CAMEL Fase 2

SSP/SSF (Service Switching Point/Function): Este es el punto de enlace entre la Red y la IN. Básicamente se encuentra dentro de la MSC que empieza la señalización. El SSP es responsable de la comunicación que existe entre la Central y el Control de Servicios,

ordena mantener la llamada e indica la sistema prepago que realice la tarificación del servicio.

SCP/SCF (*Service Control Point/Function*): Este es el nodo donde se localiza la lógica de los servicios, por lo tanto se encarga de tramitar las operaciones del protocolo CAMEL que sostienen una llamada dentro de la IN, además de manejar los datos de un usuario, para aprobar el servicio requerido.

SDP/SDF (*Service Data Point/Function*): Aquí se procede a cumplir la función de almacenamiento de la información, esto sucede cuando las operadoras proveen más servicios a usuarios que los requieran.

SRF (*Specialized Resource Function*): Esta es una de las mejoras que se integraron a CAMEL fase 2; toma a cargo las tareas de interacción de los terminales móviles, ejm.: reconocimientos de voz.

SMP/SMF (*Service Management Point/Function*): La entidad encargada del manejo de los datos de los abonados, como actualizaciones y correcciones en los servicios suministrados a ellos.

SCE (*Service Creation Environment Function*): Esta función crea establecimientos de servicios de una manera sencilla, como filtros de llamadas.

CCF (*Call Control Function*): Se produce gracias a CAMEL, que maneja el curso de la llamada hasta alguna orden de la MSC con respecto a los servicios [14]

2.5 SISTEMAS OPERATIVO

Es el software básico en una computadora y puede ser definido como un programa o conjunto de programas de un sistema informático que gestiona los recursos de hardware y provee servicios para la ejecución de los programas de aplicación, además de administrar los recursos de la máquina y organizar archivos y directorios en dispositivos de almacenamiento.

2.5.1 LINUX

LINUX es un Sistema Operativo con características similares a DOS o Windows. Es decir, Linux es el software necesario para que un ordenador permita la utilización de programas como editores de texto, navegadores de Internet, etc. Linux puede ser usado y presentado mediante una interfaz gráfico al igual que otros sistemas operativos, pero también puede trabajar mediante línea de comandos como DOS [15]

Dos características que diferencian a Linux del resto de software es que posee licencia libre, esto significa que no se tiene que

pagar a ninguna casa desarrolladora de software por su uso; la segunda es que el SO cuenta con código fuente. Además posee un gran número de librerías y aplicaciones que hacen posible el desarrollo de proyectos [16]. La MSC accede a dichas aplicaciones desde el escritorio, mostrado en la figura 2.6

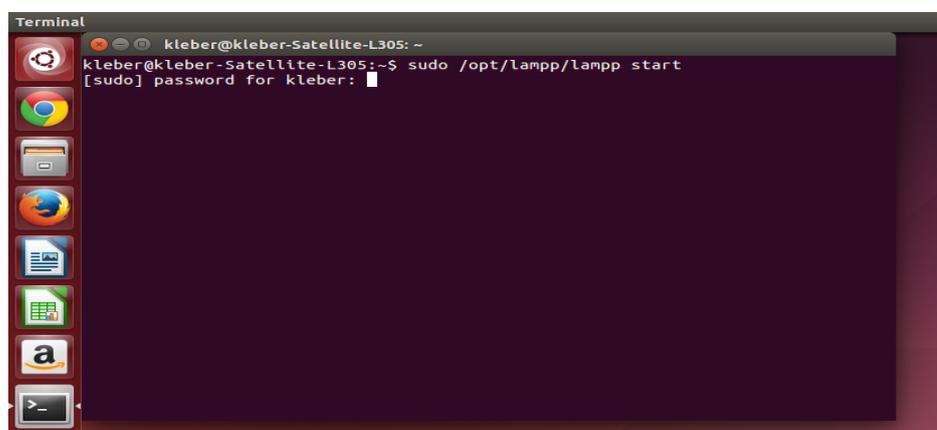


Figura 2.6 Modelo de Escritorio en Software LINUX

2.5.2 Lenguaje PHP

El Procesador de Hipertexto PHP (acrónimo recursivo de "PHP: *Hypertext Preprocessor*") es un lenguaje de programación diseñado para producir sitios web dinámicos. PHP es utilizado en aplicaciones del lado del servidor, aunque puede ser usado también desde una interfaz de línea de comandos o como aplicación de escritorio.

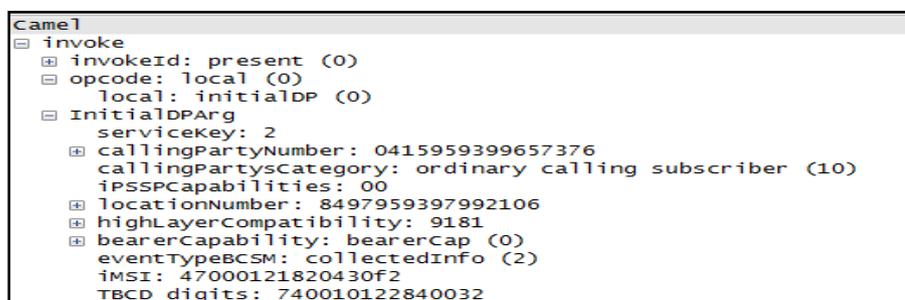
Este lenguaje de programación es el utilizado en este proyecto por sus cualidades de robustez, facilidades a la hora de programar y por la factibilidad de emular muchos de los escenarios y protocolos de CAMEL en la Central de Conmutación Móvil.

2.6 Software *Wireshark*

Wireshark es un analizador de protocolos utilizado para realizar estudios y solucionar problemas en redes de comunicaciones, para desarrollo de software y protocolos y como una herramienta didáctica. Cuenta con todas las características estándar de un analizador de protocolos. Posee una interfaz gráfica para el detalle de mensajes y filtros que permiten remitirse a los protocolos requeridos [22]

A continuación capturas de llamadas en tiempo real de una operadora móvil y visualización de mensajes CAMEL, gracias a este software, que serán detallados en el funcionamiento del proyecto en el capítulo tres:

En la figura 2.7 se observa el mensaje IDP que inicializa la conexión entre abonados.



```
Camel
├─ invoke
│   ├── invokeId: present (0)
│   └─ opcode: local (0)
│       └─ local: initialDP (0)
│           └─ InitialDPArg
│               └─ serviceKey: 2
│                   ├── callingPartyNumber: 0415959399657376
│                   ├── callingPartysCategory: ordinary calling subscriber (10)
│                   ├── ipSSPCapabilities: 00
│                   ├── locationNumber: 8497959397992106
│                   ├── highLayerCompatibility: 9181
│                   ├── bearerCapability: bearerCap (0)
│                   ├── eventTypeBCSM: collectedInfo (2)
│                   ├── IMSI: 47000121820430f2
│                   └─ TBCD digits: 740010122840032
```

Figura 2.7 Captura de mensaje IDP

En la figura 2.8 se muestra el caso en que el usuario cuenta con suficiente saldo disponible (AChR=270seg.)

```

Camel
├─ invoke
│  ├─ invokeId: present (0)
│  └─ opcode: local (0)
│     └─ local: applyChargingReport (36)
│        └─ ApplyChargingReportArg: a00ba003810101a10480020a8c
│           └─ CAMEL-CAMEL_CallResult: timeDurationChargingResult (0)
│              └─ timeDurationChargingResult
│                 └─ partyToCharge: receivingSideID (1)
│                    └─ receivingSideID: 01
│                       └─ timeInformation: timeIfNoTariffSwitch (0)
│                          └─ timeIfNoTariffSwitch: 2700

```

Figura 2.8 Captura de registro de segundos hablados

En la figura 2.9 el usuario se le termina el saldo y la PP pide a la MSC cierre de llamada

```

Camel
├─ invoke
│  ├─ invokeId: present (0)
│  └─ opcode: local (0)
│     └─ local: applyChargingReport (36)
│        └─ ApplyChargingReportArg: a00ea003810101a10480020088820100
│           └─ CAMEL-CAMEL_CallResult: timeDurationChargingResult (0)
│              └─ timeDurationChargingResult
│                 └─ partyToCharge: receivingSideID (1)
│                    └─ timeInformation: timeIfNoTariffSwitch (0)
│                       └─ timeIfNoTariffSwitch: 136
│                          └─ legActive: False

```

Figura 2.9 Captura de cierre de llamadas por falta de saldo

En la figura 2.10 se visualiza que el número llamante termina la comunicación (*eventTypeBCSM: oDisconnect*)

```

Camel
├─ invoke
│  ├─ invokeId: present (0)
│  └─ opcode: local (0)
│     └─ local: eventReportBCSM (24)
│        └─ EventReportBCSMArg
│           └─ eventTypeBCSM: oDisconnect (9)
│              └─ eventSpecificInformationBCSM: oDisconnectSpecificInfo (7)
│                 └─ oDisconnectSpecificInfo
│                    └─ releaseCause: 8090
│                       └─ legID: receivingSideID (1)

```

Figura 2.10 Desconexión por parte de número llamante

CAPÍTULO 3

3 DESCRIPCIÓN DEL PROYECTO

3.1 Software del Sistema

Para el diseño e implementación de este tipo de protocolos de un sistema de señalización, se requiere un sistema operativo que su manejo sea “amigable e intuitivo”, y a su vez que sea robusto y confiable para dar soporte a todos los procesos y funciones de una Central de Conmutación Móvil; por eso se elige a UBUNTU que es una plataforma de Linux, para aprovechar las bondades que brinda al usuario su última versión 14.04 con un sistema operativo de 32 bytes, compatible con el computador.

Otras aplicaciones ligadas a este tipo de software libre son:

- XAMPP (APACHE)
- NETBEANS
- POSTMAN

- MY ADMIN

3.2 Hardware del Sistema

Para la parte del funcionamiento físico del proyecto, se ha utilizado una computadora Toshiba, modelo Satélite L305, en donde se ha instalado los software y servidores libres requeridos, y programado la lógica de los mensajes del protocolo que respectan a CAMEL; también en esta máquina se puede encontrar el emulador de la Central de Conmutación Móvil, de donde va a intervenir en la comunicación con otros elementos, módulo que se muestra en la figura 3.1.



Figura 3.1 Máquina utilizada para implementar la MSC

Las características físicas de este modelo de Toshiba son:

- Procesador Intel Pentium Dual CPU T3400.
- Velocidad de CPU de 2.17 GHz.
- Memoria RAM de 2 GB.
- Sistema Operativo de 32 bits.

Además la MSC participa en la primera parte de la implementación con un *router* central que le asigna una ip estática; y en la conformación total con el resto de los componentes se une primero a un *router* inalámbricamente que simula una Estación Base, necesaria para la localización del abonado A, y luego de forma alámbrica con el anterior *router*.

3.3 Topología de la Red

En este proyecto se busca una comunicación directa entre los protocolos CAMEL, ISUP e INAP; los componentes que poseen este tipo de comunicación se conectan mediante un *router*. Estos componentes son: la primera MSC con el protocolo CAMEL, la segunda MSC que maneja el protocolo ISUP, PP que también maneja mensajes de CAMEL e INAP y los dos emuladores IVR que se comunica con la MSC y el PP mediante los protocolos ISUP e INAP respectivamente, y los celulares siempre se comunicarán a los servicios de la Red mediante la MSC que soporta el protocolo CAMEL; todo esto se puede observar en la figura 3.2

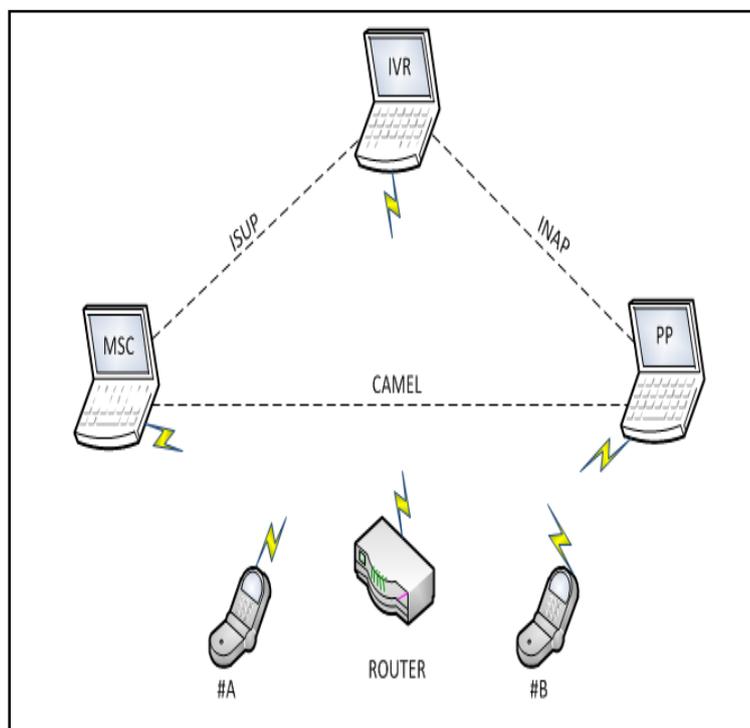


Figura 3.2 Conexión de Red para la señalización

3.4 Requisitos para la Implementación

3.4.1 Ubuntu

Ubuntu es un sistema operativo desarrollado por la empresa británica Canonical Ltd., basado en Linux y reconocido a nivel mundial porque es ideal para los usuarios promedio, ya sea en el hogar o en el ámbito educativo o profesional debido a su facilidad de manejo e instalación.

Ubuntu es un paquete de software que trabajan en código abierto y siempre se distribuye como software libre. Brinda servicios diferentes como correos o procesadores de texto o para páginas web e incluso herramientas muy útiles para programación.

En el contexto de la programación es que Ubuntu se vuelve vital, ya que soporta el diseño de una Central de Conmutación Móvil que emula las funciones y mensajes en el protocolo CAMEL para esta implementación.

Pasos de la Instalación de Ubuntu

Para el debido proceso de instalación de este software, se verifica en la maquina si se cuenta con un sistema operativo de 32 o 64 bits, y según eso se busca el indicado para cada computadora. Además, se requiere un dispositivo USB para quemar la imagen ISO de todas las funciones de este sistema operativo desde su página oficial en internet; se crea un respaldo de los documentos de la unidad USB, ya que al momento de descargar Ubuntu, este programa formatea dicho dispositivo para completar la descarga.

El programa consulta a la computadora si cuenta con el espacio suficiente dentro del disco duro para la instalación de los archivos de raíz y el correcto arranque del sistema operativo. Si este es el caso, se realiza una partición del disco destinada para el sistema Linux [17]

Al momento de introducir el pen drive, se debe reiniciar el computador para configurar correctamente el BIOS y que arranque

desde este dispositivo, así se da origen a una serie de indicaciones mostradas a continuación:

1. Cuando el programa pregunta sobre el idioma a elegir, esto se ve en la figura 3.3.

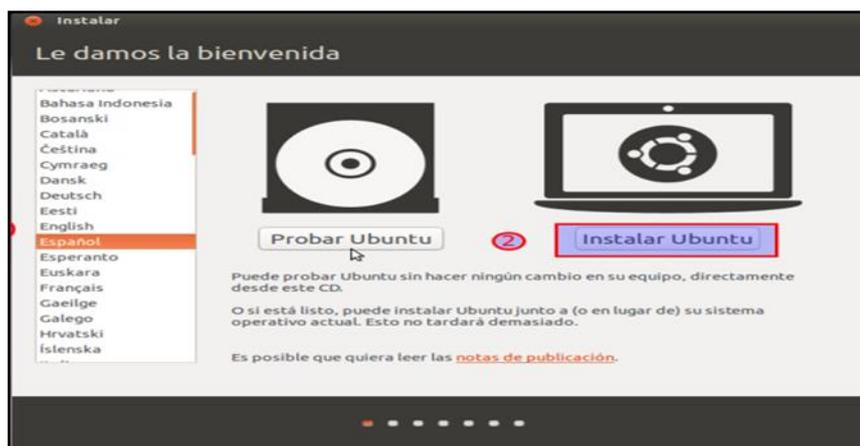


Figura 3.3 Ventana de selección de idioma [17]

2. En la figura 3.4 se prepara la instalación del programa. Se pueden elegir los casilleros “descargar actualizaciones” y “descargar software de terceros”, en este momento o al final de la instalación.



Figura 3.4 Ventana de descarga [17]

3. En la figura 3.5 se aprecia las opciones de instalación. Aquí se selecciona el ícono “más opciones”, para no borrar el sistema operativo ya existente en el disco duro.

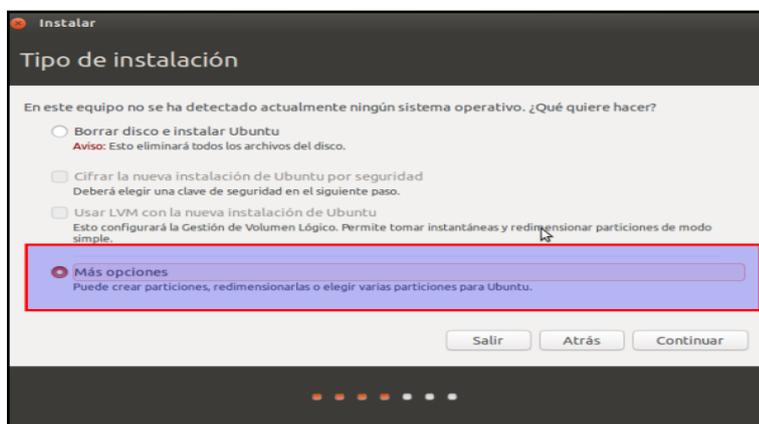


Figura 3.5 Ventana de tipo de instalación [17]

4. En la figura 3.6 existen las particiones del Disco. Aquí se procede a utilizar el disco previamente particionado, para la instalación de Ubuntu.

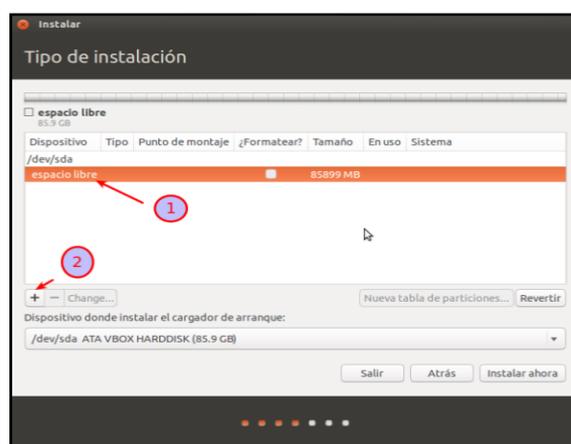


Figura 3.6 Ventana de partición de disco [17]

A continuación se crea el espacio para el GRUB, que es la ventana de arranque del sistema operativo cuando se inicializa la máquina.

Ya en la parte final de la instalación, se señala como queda la partición para el nuevo sistema operativo en el disco duro y el tamaño que alcanza, como se observa la figura 3.7.

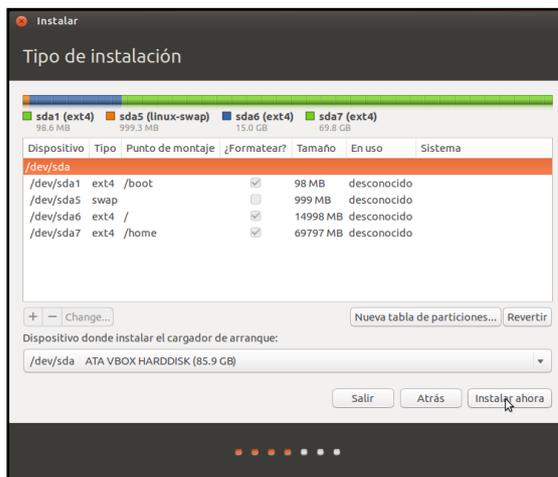


Figura 3.7 Tamaño de nuevo disco [17]

5. En la figura 3.8 aparece la configuración de la Zona Horaria y distribución del Teclado.

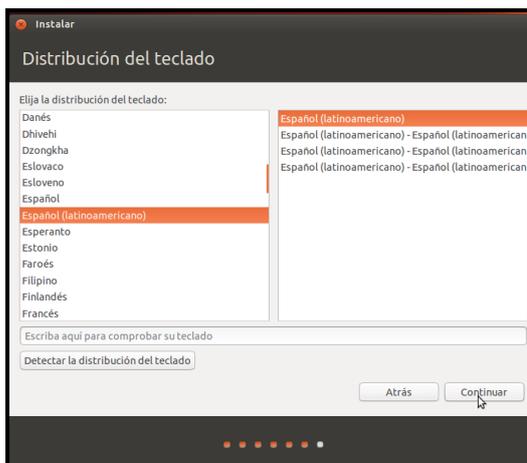


Figura 3.8 Distribución del teclado [17]

6. Cifrar la Carpeta. En la figura 3.9 se establece la configuración personal del usuario y su contraseña, para no tener inconvenientes al momento de acceder a UBUNTU.



Figura 3.9 Configuración personal [17]

7. Reiniciar ahora. En la figura 3.10 se procede a quitar el dispositivo USB y se elige la opción reiniciar. Al final de la reiniciación, usted ya podrá acceder al sistema UBUNTU sin problemas.



Figura 3.10 Ventana de acceso al sistema [17]

3.4.2 XAMPP (APACHE)

XAMPP es una plataforma de servidor, utilizada para desarrollar aplicaciones en lenguajes PHP y PERL, con bases de datos MySQL y servidor web Apache.

Actúa como servidor libre y bajo licencia GNU. Su nombre se debe a las iniciales X (cualquier sistema operativo), APACHE, MySQL, PHP y PERL.

XAMPP es una plataforma de servidor, utilizada para desarrollar aplicaciones en lenguajes PHP y PERL, con bases de datos MySQL y servidor web Apache.

Actúa como servidor libre y bajo licencia GNU. Su nombre se debe a las iniciales X (cualquier sistema operativo), APACHE, MySQL, PHP y PERL.

Pasos para la Instalación de XAMPP

Se descarga XAMPP de su página oficial en Internet, luego hay que dirigirse a la carpeta DESCARGAS, se procede a copiar el archivo y pegarlo en “carpeta personal.” [18]

Se procede a la instalación con las siguientes líneas de comando:

1. Comando que otorga los permisos de ejecución en la figura

```
sudo chmod +x xampp-linux-x64-1.8.3-4-installer.run
```

Figura 3.11 Comando de instalación de XAMPP [18]

2. Para ejecutar la instalacion de XAMPP en la figura 3.12.

```
sudo ./xampp-linux-x64-1.8.3-4-installer.run
```

Figura 3.12 Comando de ejecución de XAMPP [18]

3. En la figura 3.13 aparece el asistente de instalacion y se elige siguiente.



Figura 3.13 Asistente de instalación [18]

Se seleccionan las dos opciones que aparecen y nuevamente siguiente como se aprecia en la figura 3.14.



Figura 3.14 Selección de componentes [18]

4. Siguiendo en la nueva ventana y luego aparece “Listo para la instalación”, y se da clic en siguiente como se ve en la figura 3.15, y se espera a que termine la instalación.

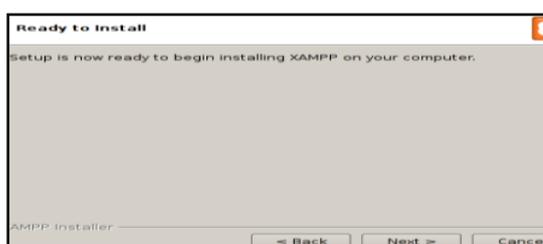


Figura 3.15 Ventana de Instalación [18]

5. A continuación se verifica que “Iniciar XAMPP” este señalada y se elige finalizar. En este proceso intermedio se abrirá una ventana y se elegirá la opción en español. Se inicia “Sistema de Gestión Gráfico de XAMPP”, como se muestra la figura 3.16.



Figura 3.16 Ventana de Inicio de XAMPP [18]

En la figura 3.17 se verifica la ejecución de los servidores y al finalizar se procede a cerrar esta ventana. Aquí está el servidor APACHE, con el que vamos a trabajar.

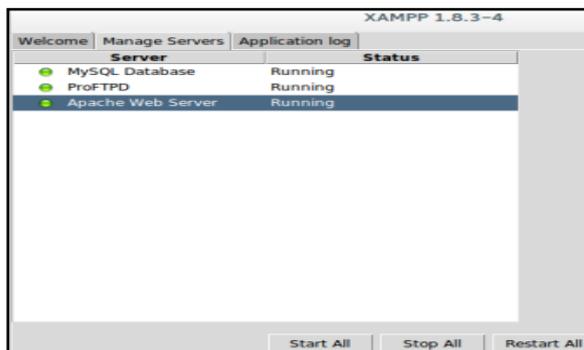


Figura 3.17 Ejecución de Servidores [18]

Así queda correctamente instalada XAMPP en el computador

3.4.3 NETBEANS

Es un entorno de desarrollo integrado creado inicialmente para programar en Java, pero con el paso del tiempo fue incluyendo otros lenguajes, ahora es posible programar para C++, PHP y otros lenguajes. Algunas de sus características son:

- Soporte de extensiones.
- Trabaja con diferentes lenguajes.
- Multiplataforma.
- Resaltado de código.

Pasos de instalación de NETBEANS

Antes que todo se procede a descargar el kit de Java como se aprecia en la figura 3.18, para que puedan ejecutarse los programas de *Netbeans* que trabajan sobre éste [19]. Una vez

hecho esto, se dirige a la carpeta donde se encuentra el servidor (opt/lampp/).



Figura 3.18 Descarga de Java [19]

1. En la página principal de *Netbeans* en internet se accede al programa y se elige los diferentes lenguajes y elementos que se presentan como se muestra en la figura 3.19.

Tecnologías compatibles *	NetBeans IDE Descargar Bundles				
	Java SE	Java EE	C / C ++	HTML5 y PHP	Todo
NetBeans Platform SDK	•	•			•
Java SE	•	•			•
Java FX	•				•
Java EE		•			•
Java ME					•
HTML5				•	•
Java Card™ 3 Conectado		•			—
C / C ++			•		•
Groovy					•
PHP				•	•
Servidores liados					
GlassFish Servidor Open Source Edition 4.0		•			•
Apache Tomcat 8.0.3		•			•
	Download Libre, 89 MB	Download Libre, 191 MB	Download Libre, 62 MB	Download Libre, 63 MB	Download Libre, 203 MB

Figura 3.19 Selección de tecnologías compatibles [19]

2. Una vez descargado el archivo, se lo ejecuta en una terminal con los siguientes comandos establecidos en la figura 3.20:

```

sudo chmod +x netbeans-8.0.2-php-linux.sh
sudo ./netbeans-8.0.2-php-linux.sh

```

Figura 3.20 Comandos de ejecución de NETBEANS [19]

3. El programa queda instalado correctamente para ser usado en las aplicaciones o programaciones que se necesiten.

3.4.4 Acerca de POSTMAN.

Es una aplicación web que realiza pruebas de Interfaces de Aplicación Programable (API por sus siglas en inglés), como se observa en la figura 3.21. Permite mandar mensajes con distintos parámetros y así verificar el envío y recepción entre diferentes dispositivos [20]

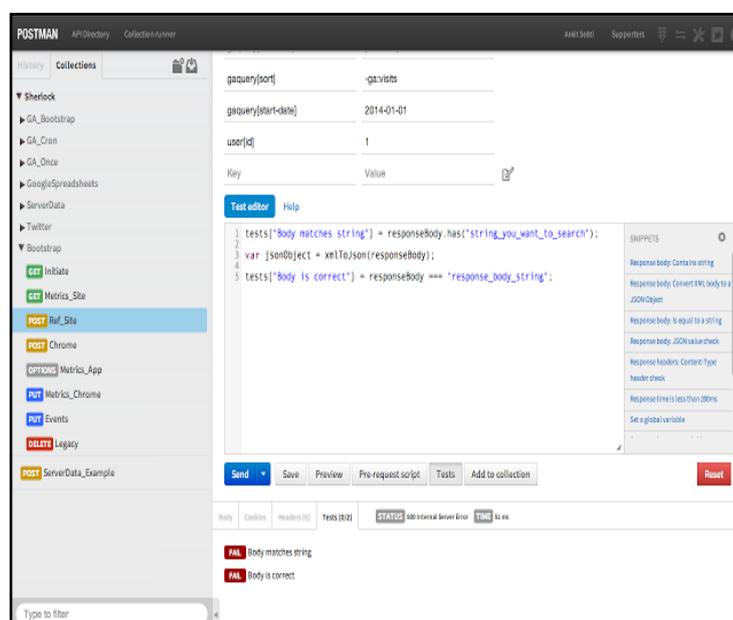


Figura 3.21 Ventana de servidor POSTMAN [20]

3.4.5 PHPMYADMIN

Este servidor facilita algunos tipos de tareas sobre este lenguaje de programación. Es capaz de Administrar la parte de MySQL que permite la creación o eliminación de bases de datos de usuarios como se nota en la figura 3.22, muy importante para la administración de saldos o identificación de números telefónicos; además del acceso a tablas que contienen los parámetros ya establecidos en cada tipo de mensajes [21]

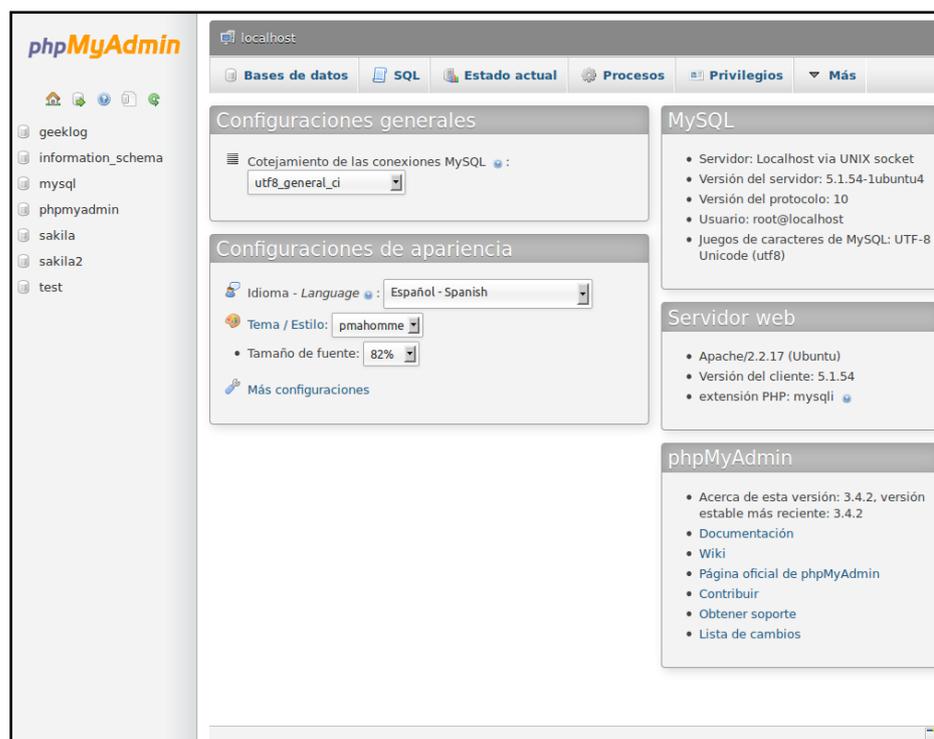


Figura 3.22 Ventana Principal del servidores *PHPMYAdmin* [21]

3.5 Funcionamiento del Emulador

En esta parte del capítulo 3 se busca dar una detallada explicación sobre el diseño de la Central de Conmutación Móvil virtual que interactúa con el sistema prepago, gracias a la comunicación con los mensajes CAMEL, dentro de una IN. Se indicará los mensajes a utilizar y la forma en que se programan las líneas de código gracias a la aplicación NETBEANS, que permiten la creación de la lógica que maneja la Central virtual para procesar los requisitos de señalización de los otros módulos dentro de los distintos casos de llamadas.

3.5.1 Mensajes del protocolo CAMEL que operan sobre la MSC

Son los que se presentarán a continuación.

Mensajes que envía la MSC

IDP (*Initial Detection Point*)→(0)

Este mensaje es enviado por la MSC, después de que se detecta la operación *Setup* ordenada por el usuario, para comunicarse con la PP y a su vez para obtener instrucciones con el propósito completar la llamada.

Estructura de un IDP: El mensaje IDP maneja varios atributos, como se muestra en la figura 3.23, detallados a continuación, pero solo utilizaremos los resaltados en color azul.

InitialDP	
Opcode-initialDP	
InitialDP Arg	
serviceKey	[0]ServiceKey,
calledPartyNumber	[2]CalledPartyNumber {bound} OPTIONAL,
callingPartyNumber	[3]CallingPartyNumber {bound} OPTIONAL,
callingPartyCategory	[5]CallingPartysCategory OPTIONAL,
cGEncountered	[7]CGEncountered OPTIONAL,
iPSSPCapabilities	[8]IPSSPCapabilities {bound} OPTIONAL,
locationNumber	[10]LocationNumber {bound} OPTIONAL,
originalCalledPartyID	[12]OriginalCalledPartyID {bound} OPTIONAL,
extensions	[15]Extensions {bound} OPTIONAL,
highLayerCompatibility	[23]HighLayerCompatibility OPTIONAL,
additionalCallingPartyNumber	[25]AdditionalCallingPartyNumber {bound} OPTIONAL,
bearerCapability	[27]BearerCapability {bound} OPTIONAL,
eventTypeBCSM	[28]EventTypeBCSM OPTIONAL,
redirectingPartyID	[29]RedirectingPartyID {bound} OPTIONAL,
redirectionInformation	[30]RedirectionInformation OPTIONAL,
cause	[17]Cause {bound} OPTIONAL,
serviceInteractionIndicatorsTwo	[32]ServiceInteractionIndicatorsTwo OPTIONAL,
carrier	[37]Carrier {bound} OPTIONAL,
cug-Index	[45]CUG-Index OPTIONAL,
cug-Interlock	[46]CUG-Interlock OPTIONAL,
cug-OutgoingAccess	[47]NULL OPTIONAL,
iMSI	[50]MSI OPTIONAL,

Figura 3.23 Atributos del mensaje IDP

Service Key(2):

Este parámetro identifica la lógica de servicio del SCF (PP) que debe atender esta petición del SSF (MSC).

Calling Party Number (Numero de Parte Llamante):

Es una persona que(o dispositivo que) inicia una llamada telefónica a través de la Red Telefónica Pública Conmutada (PSTN), generalmente marcando un número de teléfono. Este parámetro permite al gsmSCF (PP) ajustar el parámetro de número de parte llamante utilizado en la red. Se utiliza para mostrar información de la parte que envía a la parte receptora.

Called Party Number (Numero de Parte Llamada):

La parte llamada(en algunos contextos llamados el "Número B") es una persona que(o dispositivo que) responde a una llamada telefónica.

El mensaje IDP contiene dos atributos: *Operation Code-IDP* y *IDP-Argument* como se muestra en la parte superior.

El primero, *Opcode-IDP*, siempre tendrá un valor de 0 decimal para indicarnos que el mensaje es un IDP.

El segundo, que es un *IDPArg* posee dos atributos que son: *CallingPartyNumber* y *CalledPartyNumber*, de estos atributos solo usaremos *aChBillingChargingCharacteristics* que posee *TimeDurationCharging* (Duración de la Carga de Tiempo).

Apply Charging Report→(36)

Esta operación es utilizada por el MSC para reportar carga de la información relacionada la PP como había solicitado el PP mediante la operación "*ApplyCharging*" o Carga de Aplicación.

El MSC recibe un mensaje *Apply Charging* de la PP, que es un tipo de autorización señalando que el abonado si cuenta con saldo para realizar la llamada; luego el MSC devuelve el mensaje *Apply Charging Report* con la cantidad de segundos hablados. El *Apply Charging* que nos envía la PP puede ser de 90 segundos o menos, dependiendo el saldo que posea. Cada vez que enviemos un *Apply Charging Report*, se debe mandar el total de lo que el abonado lleva conectado a la llamada en segundos.

Si el abonado ya no cuenta con saldo, inmediatamente envía un *Disconnect*, que es un atributo del mensaje *Event ReportBCSM*.

Estructura del *Apply Charging Report*: El AChR presenta la siguiente estructura, pero solo los atributos a utilizar serán los resaltados con color azul, detallados en la figura 3.24:

ApplyChargingReport	
Opcode-applyChargingReport	
ApplyChargingReportArg	
CallResult {PARAMETERS-BOUND : bound} ::= CHOICE {	
timeDurationChargingResult [0]SEQUENCE {	
timeInformation [1]TimeInformation{	
timeIfNoTariffSwitch [0]TimeIfNoTariffSwitch,	
timeIfTariffSwitch [1]TimeIfTariffSwitch	
partyToCharge [0]ReceivingSideID,	
callActive [2]BOOLEAN DEFAULT TRUE,	
callReleasedAtTcpxpiry [3]NULL OPTIONAL,	

Figura 3.24 Atributos del mensaje AChR

Call Result:

Este parámetro proporciona a la PP la información relacionada con la carga solicitada previamente mediante la operación ApplyCharging. Contiene al parámetro *Time Duration Charging Result*.

Time Information: Es el que proporciona la duración de la llamada. Es una selección de los siguientes parámetros:

TimeIfNoTariffSwitch: Este parámetro estará presente desde la primera operación *Apply Charging*, hasta la desconexión de la llamada llevando el total del tiempo hablado en segundos. Si el usuario se queda sin saldo disponible, se envía un comando "*legActive*" con el valor "*false*" (*legActive: false*).

TimeIfTariffSwitch.- Esto es un servicio de post pago.

Event Report Basic Call State Model→(24)

Esta operación se utiliza para notificar al PP de un evento relacionado con la llamada; pueden ser escenarios tales como: cierre de llamada de la parte llamante o que se quedó sin disponibilidad de saldo, o algún otro tipo de información relacionado con la llamada en proceso, previamente solicitados por el PP en una operación ***RequestReportBCSMEvent***.

Estructura del *Event Report BCSM*: Se presenta la siguiente estructura, con los atributos resaltados con azul en la figura 3.25, que utilizaremos a continuación:

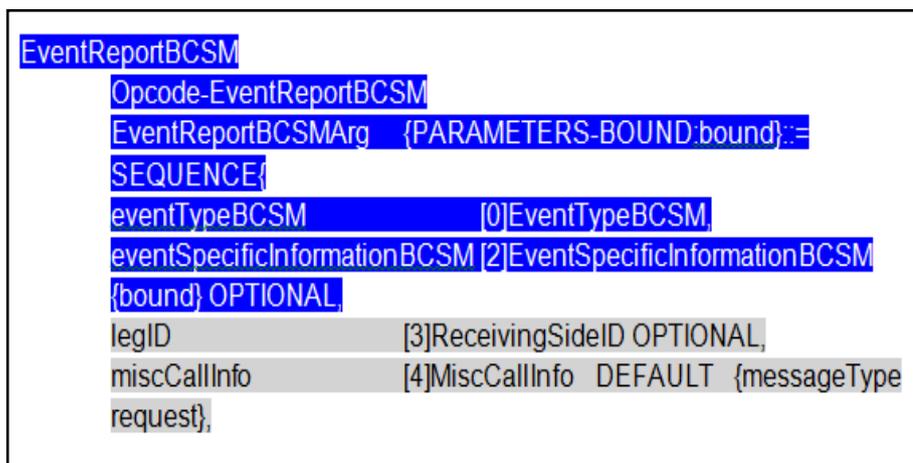


Figura 3.25 Atributos del mensaje ERBCSM

Event Type BCSM: oDisconnect.

Este parámetro muestra el tipo de evento que se reporta (El abonado A desconecta la llamada).

Event Type BCSM: tDisconnect.

El abonado B desconecta la llamada.

Event Type tNoAnswer.

El abonado B rechaza o no contesta a la solicitud de llamada.

Mensajes que recibe la MSC

La Central participa en este proyecto con comunicación directa con los celulares y el sistema prepago. A continuación se menciona algunos de los mensajes más importantes que son enviados hacia la MSC desde estos componentes de la Red y que permiten que sea posible la señalización. Además con estos diagramas y su explicación, se espera dar una idea clara en lo que pueden ser variantes de los dos grandes escenarios de llamadas que se mostrarán en el siguiente subcapítulo.

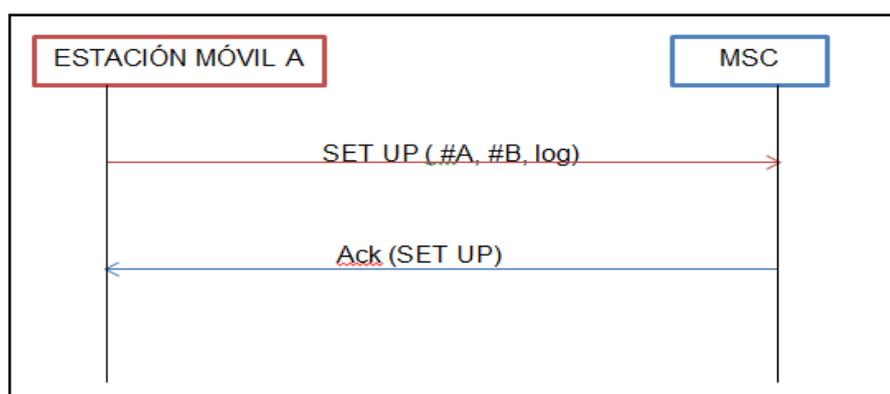
SET UP

Figura 3.26 Mensaje de Set Up

Este mensaje es el comienzo de la comunicación con la Central de Conmutación Móvil. El usuario que desea establecer la llamada, manda automáticamente el Set Up con la respectiva información, cuando marca al número que quiere llamar y es en ese momento que la MSC devuelve un mensaje de confirmación del requerimiento.

Los tres campos que van dentro del Set Up tienen el siguiente formato:

Parámetros:

#A: Formato Dec. 14 dígitos.

#B: Formato Dec. 14 dígitos.

Log: Formato Dec. 3 dígitos.

Los números telefónicos poseen la siguiente estructura para su correcta identificación.

MSISDN=00+CC+NDC+Ki=14 dígitos.

00= *International Code* (2 decimales).

CC= *Code Country* (3 decimales) <Ecuador: 593>.

NDC=*National Destination Code* (2 Decimales) <93, 95, 96, 97>.

Ki=*Individual Subscriber Authentication Key*(7 decimales)
(cualquier combinación).

La lógica del servicio (log) puede variar de la siguiente manera:

100= Llamadas en general

105= Autocontrol (*282, *611, 911, entre otros).

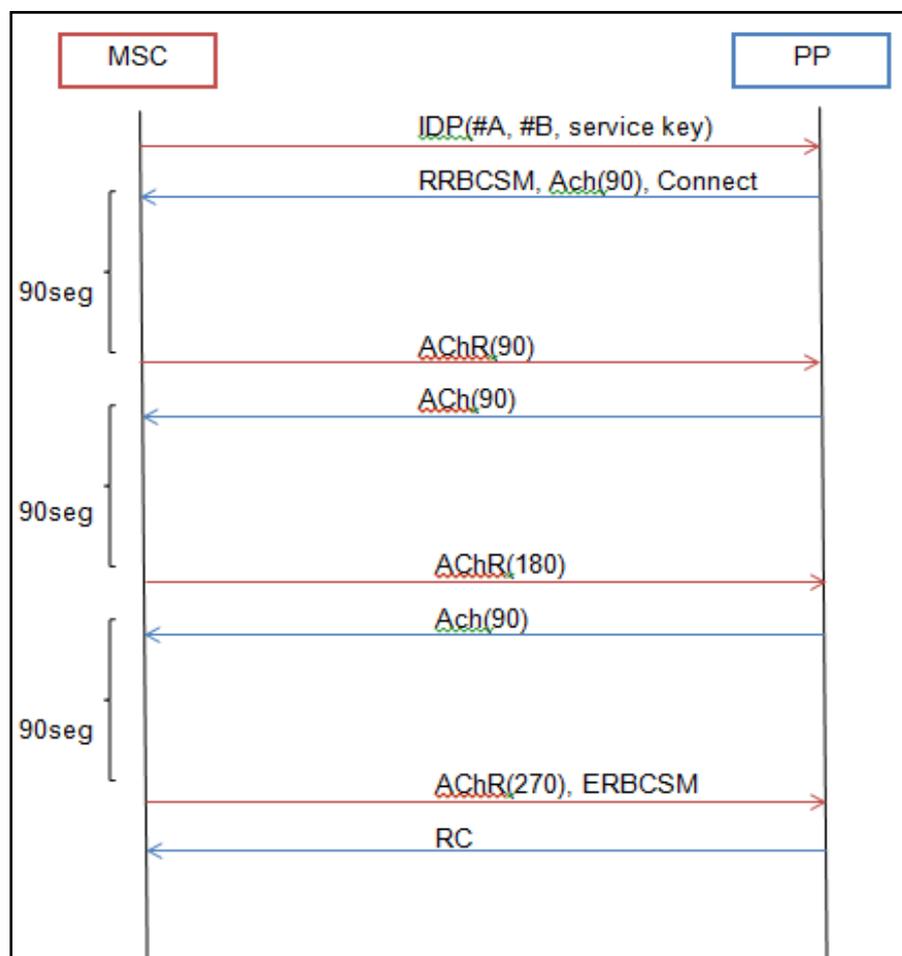


Figura 3.27 Mensajes con el Sistema Prepago

CONNECT

La MSC recibe este mensaje una vez que el sistema prepago ha confirmado que el abonado A cuenta con saldo suficiente para conectar la llamada.

ACh

Una vez que la PP recibe el mensaje IDP, verifica en su base de datos cuanto saldo posee el número llamante, procede a transformarlo en segundos y enviar la notificación a la Central, que controlará el tiempo permitido para el usuario.

RRBCSM

Lo recibe la central desde el sistema prepago para que se le notifique de cualquier evento que suceda con respecto a la llamada.

Release Call

La PP le envía este mensaje a la MSC para el cierre de la comunicación, cuando el móvil que realiza la llamada se queda sin saldo disponible.

INCOMING CALL



Figura 3.28 Mensaje con Abonado B

Una vez que la plataforma prepago busca la información del abonado A en su base de datos y comprueba que cuenta con saldo disponible para establecer la llamada, la PP envía el mensaje *CONNECT* a la MSC y esta a su vez envía el requerimiento *INCOMING CALL* al número B, para que este conteste la llamada.

ETC



Figura 3.29 Mensaje ETC

La Conexión Temporal de Establecimiento (ETC por sus siglas en inglés), es un mensaje enviado por PP hacia la MSC, cuando este detecta que el abonado desea establecer conexión con el *282.

Los campos que pertenecen a este mensaje son:

El número 17 que es la identificación de un ETC y el *282 que indica que se quiere comunicar con este servicio de Red.

3.5.2 Análisis del Funcionamiento

Este proyecto cuenta con dos amplios escenarios sobre los cuales trabaja, que son los siguientes: llamada entre abonados efectuada por un usuario prepago y llamada de un usuario prepago al *282. Al mismo tiempo, estos dos tipos de posibilidades dentro de una Red de Telefonía pueden contener un universo de casos debido a que los eventos en los que se cierra la comunicación, se cuelga antes de una consulta de saldo o un requerimiento de recarga, etc., pueden ser causados por diferentes motivos y en diferentes momentos en los que se está produciendo la señalización.

A continuación se busca detallar la participación de una Central de Conmutación Móvil, encargada de trabajar con el protocolo CAMEL, en la interacción con los diferentes componentes de la Red Celular. Tratando de abarcar gran parte de dichos casos, se hace una exposición en la cual se explique lo concerniente a la MSC y su participación en este tipo de servicios.

En el capítulo dos se hace el desglose de los mensajes con cada uno de sus atributos; ahora se mostrará su intervención en los escenarios antes establecidos.

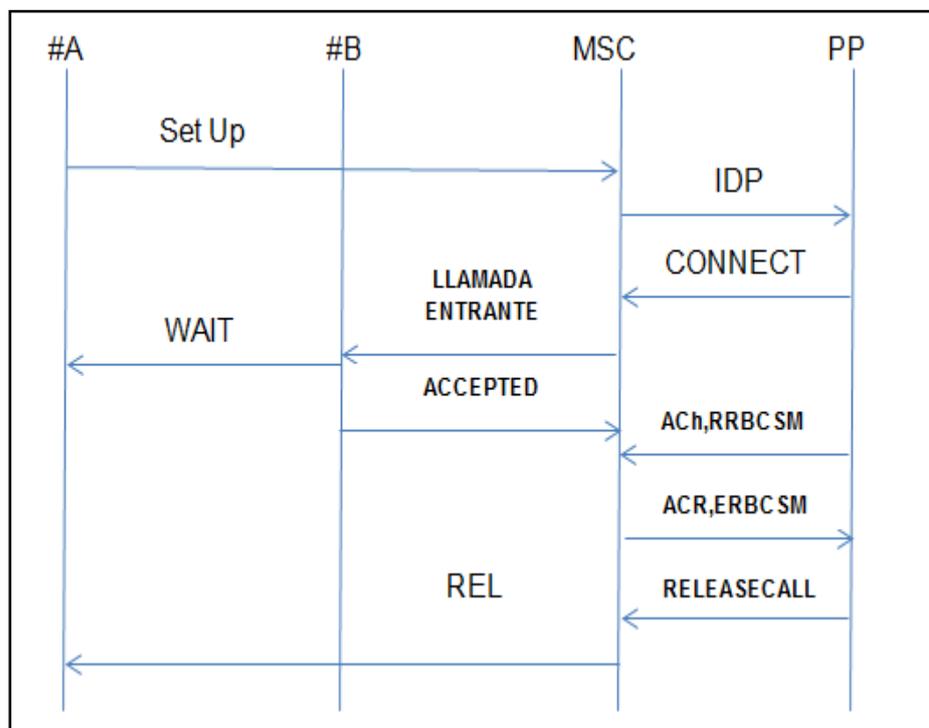


Figura 3.30 Caso típico de llamada entre abonados

La figura 3.26 permite ver cuando el abonado A desea establecer una conexión con el numero B, éste le envía un requerimiento de llamada a la MSC, que es el IDP con los parámetros #A, #B, log; la MSC convierte estos parámetros en *CallingNumber Party*, *Called Number Party* y *Service Key*, y se los transfiere a la PP la cual contiene una base de datos que examina distintos aspectos como si el usuario A posee un plan prepago con la operadora y si cuenta con saldo suficiente para establecer comunicación. La PP devuelve a la Central los campos:

Connect: Que le indica que efectivamente si puede establecer la llamada.

ACh: Este atributo es entregado a la MSC como autorización de mantener la conversación y son lapsos de 90 segundos o menos, dependiendo de la cuenta del número llamante.

RRBSCM: Con esto se espera cualquier notificación de evento con respecto a la llamada.

Inmediatamente la MSC envía dos mensajes que son: llamada entrante al número B y *wait* al A; el número llamado comunica a la MSC cuando a respondido con *Accepted* y la Central da permiso para que se mantenga la conexión con un *Call Continue*.

En esta parte de la señalización la MSC comunica a la PP sobre los segundos hablados en lapsos de 90 y de forma acumulativa en cada reporte hasta el cierre de llamada; dependiendo de si el usuario A o B cuelga, la central nuevamente crea un mensaje hacia el Sistema Prepago, el ERBSCM donde se indica el campo *oDisconnect* (la fuente cierra) o *tDisconnect* (el destino cierra); con esto la PP transmite un *Release Call* a la MSC, y esta a su vez a los terminales.

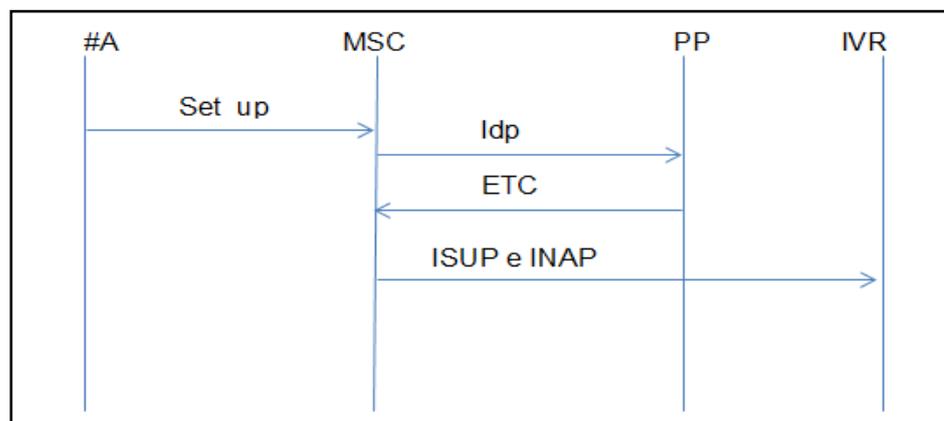


Figura 3.31 Consulta de saldo o activación de tarjeta prepago

En la figura 3.27 se indica el escenario cuando el usuario desea saber el monto en dólares con el que cuenta o necesita hacer una recarga para disponibilidad de sus distintos servicios; intenta establecer conexión con el *282, de este modo envía el mensaje *set_up* a la MSC pero esta vez con este número de consulta como abonado B. La Central procesa el IDP hacia la PP con la lógica de servicio correspondiente y ésta al verificar el destino de la llamada, le contesta con un ETC que indica que el número llamante desea acceder a servicios de la operadora celular; a partir de este momento de la señalización entran a participar elementos con distintos protocolos hasta el final de la llamada.

3.5.3 Mensajes de Programación y líneas de código del Emulador MSC

A continuación se muestra formas y procedimientos que se han utilizado para la programación de los mensajes del protocolo CAMEL, que soporta la Central de Conmutación Móvil.

Antes de comenzar a trabajar en cualquier ámbito de Linux, se debe levantar sesión en APACHE con los comandos anteriormente mencionados. Posteriormente se utiliza NETBEANS con su lenguaje PHP, es aquí la parte fundamental del software, ya que gracias a este servidor se puede programar las líneas de código para el correcto funcionamiento de proyecto.

Para poder utilizar los distintos mensajes, se crean archivos en .php, que permite interactuar con resto de componentes del trabajo implementado ya que son guardados en una lógica igual; así al final se conmutan para el intercambio de comunicación en los escenarios de las llamadas.

En cada .php se recibe un mensaje de otro elemento y siempre se tiene que devolver algo a dicho elemento, ya sea como un nuevo mensaje con parámetros o una simple confirmación de que se pudo establecer un enlace entre ellos; para este fin se utiliza las líneas enviado nuevo y recibido nuevo.

Set_Up



```

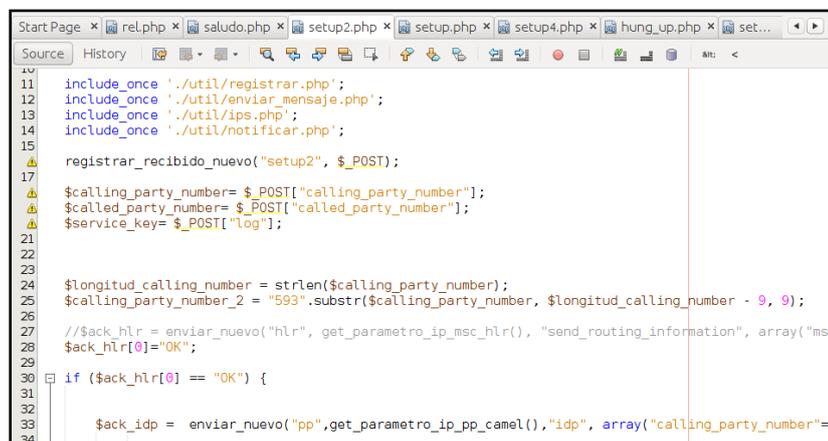
1 k?php
2 header('Access-Control-Allow-Origin: *');
3
4 include_once './util/registrarse.php';
5 include_once './util/enviar_mensaje.php';
6
7 registrar_recibido_nuevo("setup", $_POST);
8
9
10 enviar_respuesta(array("ok"));
11
12
13

```

Figura 3.32 Primer mensaje del emulador MSC

La figura 3.29 es el primer enlace con el celular. Esto se produce en el momento que recibe el requerimiento de llamada y la Central devuelve una confirmación.

Set_Up2



```

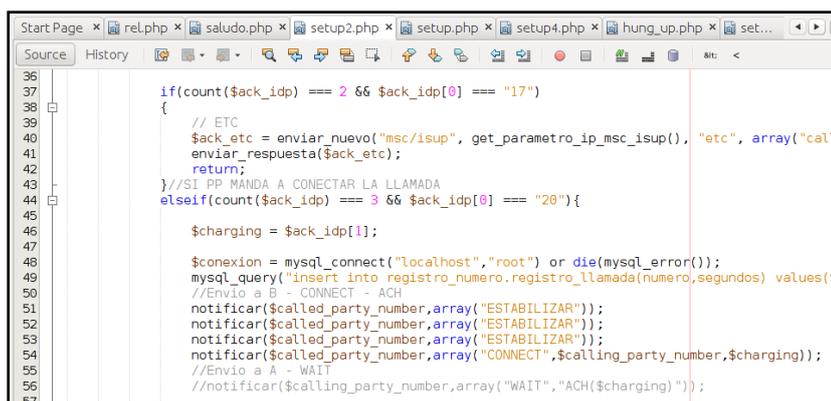
11 include_once './util/registrarse.php';
12 include_once './util/enviar_mensaje.php';
13 include_once './util/ips.php';
14 include_once './util/notificar.php';
15
16 registrar_recibido_nuevo("setup2", $_POST);
17
18 $calling_party_number= $_POST["calling_party_number"];
19 $called_party_number= $_POST["called_party_number"];
20 $service_key= $_POST["log"];
21
22
23
24 $longitud_calling_number = strlen($calling_party_number);
25 $calling_party_number_2 = "593".substr($calling_party_number, $longitud_calling_number - 9, 9);
26
27 // $ack_hlr = enviar_nuevo("hlr", get_parametro_ip_msc_hlr(), "send_routing_information", array("msi
28 $ack_hlr[0]="OK";
29
30 if ($ack_hlr[0] == "OK") {
31
32
33     $ack_idp = enviar_nuevo("pp", get_parametro_ip_pp_camel(), "idp", array("calling_party_number"=>
34

```

Figura 3.33 Primera parte de código Set_Up2

En la figura 3.30 la MSC recibe dos campos que son los números que intervienen en la conexión y un campo que contiene la lógica

del servicio y con el comando *POST* se procede a renombrarlos como *calling party number*, *called party number* y *service key* y se los transfiere a la PP.



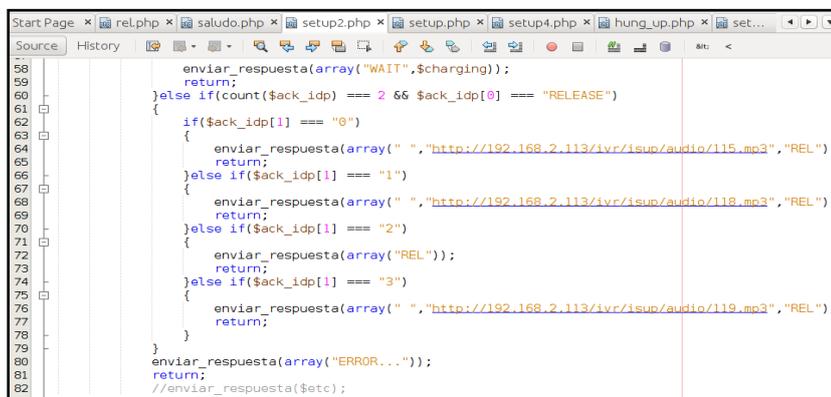
```

36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
    if(count($sack_idp) === 2 && $sack_idp[0] === "17")
    {
        // ETC
        $sack_etc = enviar_nuevo("msc/isup", get_parametro_ip_msc_isup(), "etc", array("call
        enviar_respuesta($sack_etc);
        return;
    } //SI PP MANDA A CONECTAR LA LLAMADA
    elseif(count($sack_idp) === 3 && $sack_idp[0] === "20"){
        $charging = $sack_idp[1];
        $conexion = mysql_connect("localhost", "root") or die(mysql_error());
        mysql_query("insert into registro_numero.registro_llamada(numero,segundos) values($s
        //Envio a B - CONNECT - ACH
        notificar($called_party_number,array("ESTABILIZAR"));
        notificar($called_party_number,array("ESTABILIZAR"));
        notificar($called_party_number,array("ESTABILIZAR"));
        notificar($called_party_number,array("CONNECT",$calling_party_number,$charging));
        //Envio a A - WAIT
        //notificar($calling_party_number,array("WAIT","ACH($charging)"));

```

Figura 3.34 Segunda parte de código Set_Up2

En la figura 3.31 el comando *count* se procede a leer el mensaje proveniente de la PP y el lazo *if* condiciona que si la llamada es al *282 se transfiera el mensaje a la MSC con protocolo ISUP.



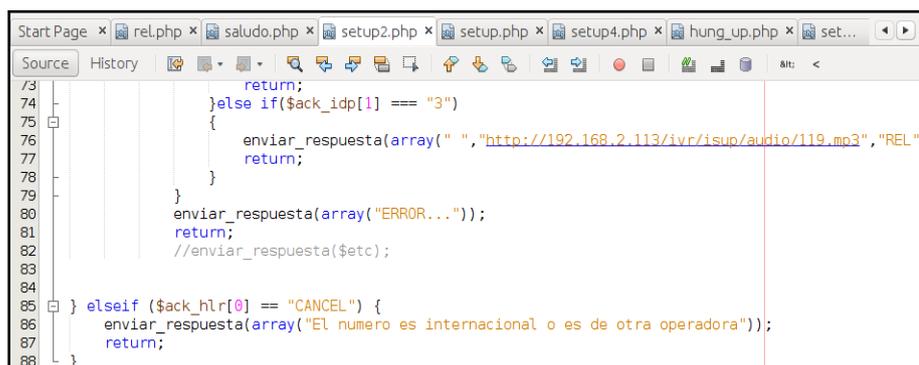
```

58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
    enviar_respuesta(array("WAIT",$charging));
    return;
} else if(count($sack_idp) === 2 && $sack_idp[0] === "RELEASE")
{
    if($sack_idp[1] === "0")
    {
        enviar_respuesta(array(" ", "http://192.168.2.113/ivr/isup/audio/115.mp3", "REL"));
        return;
    } else if($sack_idp[1] === "1")
    {
        enviar_respuesta(array(" ", "http://192.168.2.113/ivr/isup/audio/118.mp3", "REL"));
        return;
    } else if($sack_idp[1] === "2")
    {
        enviar_respuesta(array("REL"));
        return;
    } else if($sack_idp[1] === "3")
    {
        enviar_respuesta(array(" ", "http://192.168.2.113/ivr/isup/audio/119.mp3", "REL"));
        return;
    }
}
    enviar_respuesta(array("ERROR..."));
    return;
//enviar_respuesta($etc);

```

Figura 3.35 Tercera parte del código Set_Up2

La figura 3.32 indica cómo se busca programar las líneas de código para, si es el caso de llamada para consulta de saldo, se proceda a invocar los *prompts* que participan en la comunicación con el *282

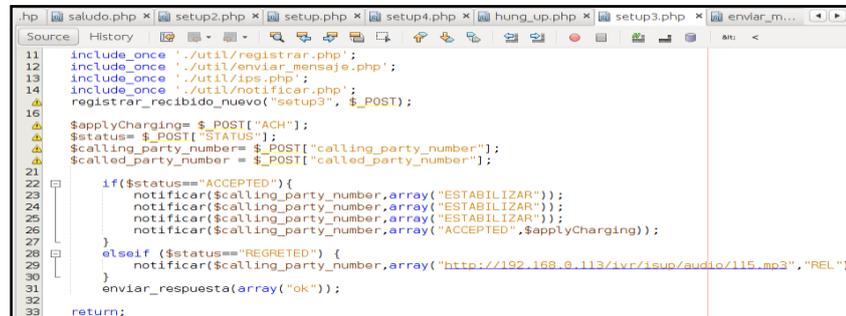


```
73  
74     return;  
75 }else if($ack_idp[1] === "3")  
76 {  
77     enviar_respuesta(array(" ", "http://192.168.2.113/ivr/isup/audio/119.mp3", "REL")  
78     return;  
79 }  
80     enviar_respuesta(array("ERROR..."));  
81     return;  
82     //enviar_respuesta($etc);  
83  
84  
85 } elseif ($ack_hlr[0] == "CANCEL") {  
86     enviar_respuesta(array("El numero es internacional o es de otra operadora"));  
87     return;  
88 }
```

Figura 3.36 Cuarta parte del código Set_Up2

En la figura 3.33 se realizan validaciones para comprobar si el teléfono que realiza la llamada es un abonado prepago, si el usuario B es de otra operadora o es un número incorrecto, bajo el comando *array*.

Set_Up3



```

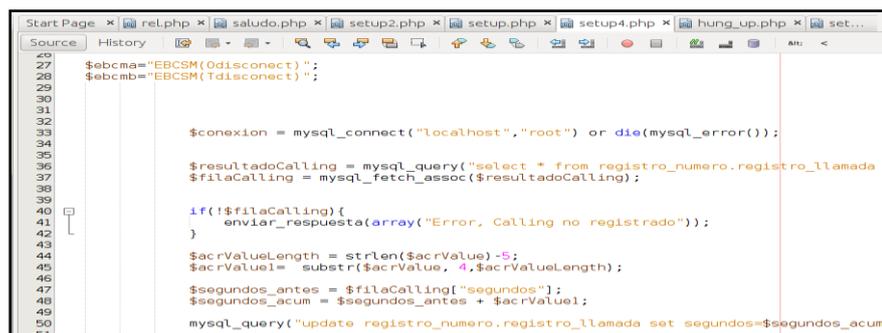
11 include_once './util/registras.php';
12 include_once './util/Enviar_mensaje.php';
13 include_once './util/ips.php';
14 include_once './util/notificar.php';
15 registrar_recibido_nuevo("setup3", $_POST);
16
17 $applyCharging= $_POST["ACH"];
18 $status= $_POST["STATUS"];
19 $calling_party_number= $_POST["calling_party_number"];
20 $called_party_number = $_POST["called_party_number"];
21
22 if($status=="ACCEPTED"){
23     notificar($calling_party_number,array("ESTABILIZAR"));
24     notificar($calling_party_number,array("ESTABILIZAR"));
25     notificar($calling_party_number,array("ESTABILIZAR"));
26     notificar($calling_party_number,array("ACCEPTED",$applyCharging));
27 }
28 elseif ($status=="REGRETED") {
29     notificar($calling_party_number,array("http://192.168.0.113/ivr/isup/audio/115.mp3","REL")
30 }
31 enviar_respuesta(array("ok"));
32
33 return;

```

Figura 3.37 Líneas de código de Set_Up3

Si se establece la llamada, aquí se recibe el parámetro ACh con el que se procede a hacer la acumulación de los segundos hablados, como se nota en la figura 3.34 Si el abonado no cuenta con saldo se le envía un prompt diciendo que por favor active una tarjeta prepago.

Set_Up4



```

49 $ebcma="EBCSM(Odisconnect)";
50 $ebcmb="EBCSM(Tdisconnect)";
51
52
53 $conexion = mysql_connect("localhost","root") or die(mysql_error());
54
55 $resultadoCalling = mysql_query("select * from registro_numero.registro_llamada");
56 $filaCalling = mysql_fetch_assoc($resultadoCalling);
57
58 if(!$filaCalling){
59     enviar_respuesta(array("Error, Calling no registrado"));
60 }
61
62 $acrValueLength = strlen($acrValue)-5;
63 $acrValue1= substr($acrValue, 4,$acrValueLength);
64
65 $segundos_antes = $filaCalling["segundos"];
66 $segundos_acum = $segundos_antes + $acrValue1;
67
68 mysql_query("update registro_numero.registro_llamada set segundos=$segundos_acum");
69

```

Figura 3.38 Código programado de segundos acumulados en llamada

En la figura 3.35 básicamente se busca programar la respuesta AChR que contiene en la MSC los segundos hablados, y se los concatena gracias al comando `$acrValueLength`, para poder comunicarle a la PP en el transcurso de la llamada.

```

54
55
56     if(isset($_POST["DISCONNECT"])){
57
58         $target = $_POST["DISCONNECT"];
59         if($acrValue!=null){
60             if($target == "A"){
61                 $respuestaPP = enviar_nuevo("pp",get_parametro_ip_pp_camel(),"acr", array(
62                     //disconnect_b());
63
64                 enviar_nuevo("msc/isup", get_parametro_ip_msc_isup(), "rel", array("calling_party_number,array("REL"));
65                 //Se borra el registro de la tabla numero_registro
66                 mysql_query("delete from registro_numero_registro_llamada where numero=$call");
67                 enviar_respuesta(array("OK"));
68                 return;
69             }
70             elseif($target == "B"){
71                 $respuestaPP = enviar_nuevo("pp",get_parametro_ip_pp_camel(),"acr", array(
72                     //disconnect_a());
73                 enviar_nuevo("msc/isup", get_parametro_ip_msc_isup(), "rel", array("calling_party_number,array("REL"));
74                 //Se borra el registro de la tabla numero_registro
75                 mysql_query("delete from registro_numero_registro_llamada where numero=$call");
76                 enviar_respuesta(array("OK"));
77                 return;
78             }
79         }
80     }

```

Figura 3.39 Reporte de evento de llamada por parte de la MSC

La figura 3.36 muestra como el comando `$respuestaPP` se comunica al sistema prepago de los diferentes eventos en una llamada y además se solicita actualización en la base de datos sobre el abonado enviando el mensaje `acr`, con sus respectivos parámetros .

3.6 Síntesis de Implementación y Funcionamiento

Es necesario conocer algunos aspectos sobre la plataforma Linux y todas las Aplicaciones con las que cuenta, para la correcta implementación de este software en cada máquina que participa en este proyecto. El software del computador es un SO de 32 bits y tiene que ser compatible con la versión de Ubuntu con la que trabajamos que es Ubuntu 14.04, además de necesitar otros programas como APACHE para el arranque

de todas las funciones en Linux, Netbeans que sirve para programar las líneas de código y PHPMYADMIN que maneja las bases de datos que necesita para los diferentes números telefónicos a utilizar.

A lo largo del capítulo tres además se requiere explicar como será la implementación de los mensajes CAMEL que soporta la MSC para participar en los distintos casos de llamadas; aquí se muestra detalladamente la señalización en lo que respecta a la Función de Conmutación de Servicios, vista anteriormente en el capítulo dos; y por último la programación realizada en base al alcance que tiene la Central en los escenarios simulados.

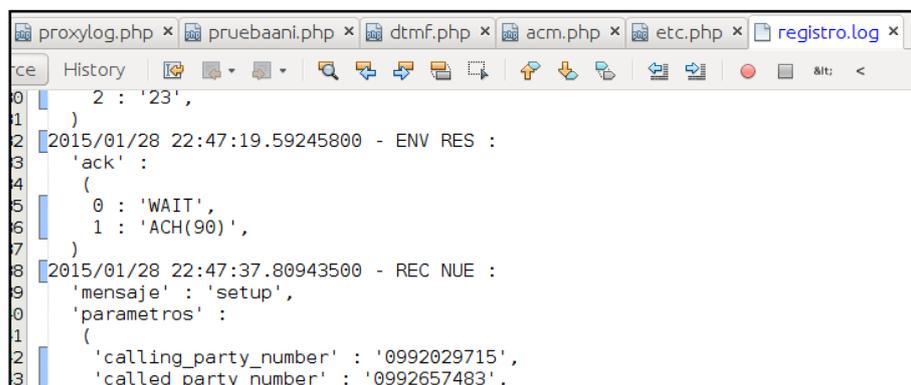
CAPÍTULO 4

4 ESTADO FINAL DEL PROYECTO

4.1 Resultados Obtenidos

En esta parte del proyecto es necesario mostrar lo que ha arrojado el funcionamiento de los emuladores ya implementados; así que al finalizar se obtiene un fichero creado en *NETBEANS*, al cual se denomina registro.log que contiene todas las pruebas realizadas desde la unión de los módulos hasta la realización correcta de todos los escenarios de llamadas entre dos abonados o al *282.

En la figura 4.1 se presenta el archivo donde están los intentos de plasmar los distintos casos requeridos en este trabajo.



```

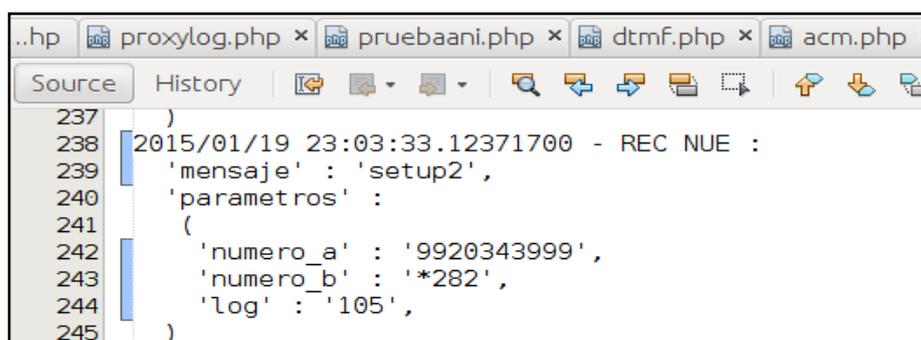
proxylog.php x pruebaani.php x dtmf.php x acm.php x etc.php x registro.log x
History
0 | 2 : '23',
1 | )
2 | [2015/01/28 22:47:19.59245800 - ENV RES :
3 | 'ack' :
4 | (
5 | 0 : 'WAIT',
6 | 1 : 'ACH(90)',
7 | )
8 | [2015/01/28 22:47:37.80943500 - REC NUE :
9 | 'mensaje' : 'setup',
10 | 'parametros' :
11 | (
12 | 'calling_party_number' : '0992029715',
13 | 'called_party_number' : '0992657483',

```

Figura 4.1 Fichero registro.log

Este registro fue de mucha utilidad, ya que de este modo se pudo comprobar que el simulador estaba cumpliendo con las líneas de código previamente creadas y fijadas en el capítulo 3.

A continuación se puede visualizar algunas imágenes del envío y recibo, en tiempo real, de los mensajes de la Central de Conmutación Móvil.



```

..hp proxylog.php x pruebaani.php x dtmf.php x acm.php x
Source History
237 | )
238 | [2015/01/19 23:03:33.12371700 - REC NUE :
239 | 'mensaje' : 'setup2',
240 | 'parametros' :
241 | (
242 | 'numero_a' : '9920343999',
243 | 'numero_b' : '*282',
244 | 'log' : '105',
245 | )

```

Figura 4.2 Mensaje recibido por la MSC proveniente del abonado

En la figura 4.2 se aprecia el recibimiento por parte de la MSC de los campos provenientes del abonado, cuando inició el pedido de requerimiento de llamada.

```

245 )
246 [2015/01/19 23:03:33.12407100 - ENV NUE :
247   'destinatario' : 'pp',
248   'mensaje' : 'idp',
249   'parametros' :
250   (
251     'calling_party_number' : '9920343999',
252     'called_party_number' : '*282',
253     'service_key' : '105',
254   )
255 [2015/01/19 23:03:33.18481300 - REC RES :
256   'ack' :
257   (

```

Figura 4.3 Mensaje de requerimiento al número de servicio *282

El mensaje IDP se envió con sus tres parámetros: número llamante, número llamado y *service key* como se observa en la figura 4.3; y el PP confirmó el recibimiento de los datos en su emulador.

```

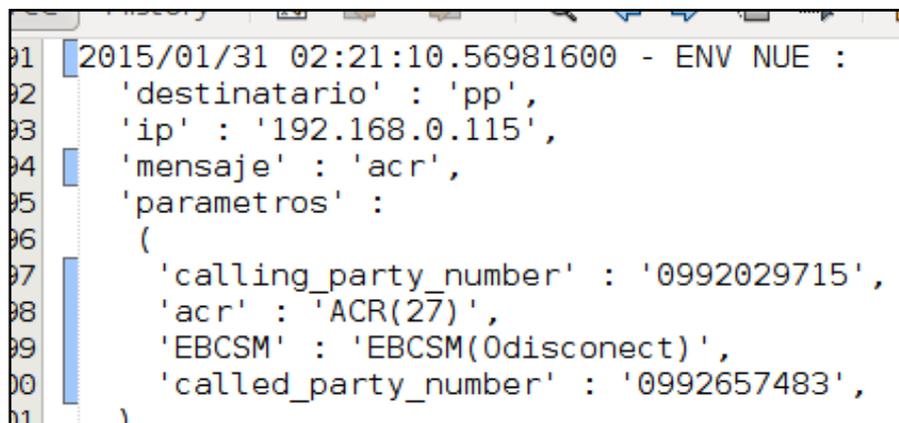
1 [2015/02/07 19:00:58.65532700 - ENV NUE :
2   'destinatario' : 'pp',
3   'ip' : '192.168.0.115',
4   'mensaje' : 'idp',
5   'parametros' :
6   (
7     'calling_party_number' : '0993400399',
8     'called_party_number' : '0987004644',
9     'service_key' : '100',
10  )
11 [2015/02/07 19:00:58.67547600 - REC RES :
12   'ack' :
13   (
14     0 : '20',
15     1 : 'ACH(90)',
16     2 : '23',
17   )

```

Figura 4.4 Mensaje que recibe la MSC de autorización de llamadas

En la llamada entre dos abonados, la Central entregó los campos a la PP y esta hizo el análisis correspondiente para aceptar el requerimiento, devolviéndole a la MSC la autorización de conectar y cadenas de

segundos permitidos según el saldo con el que se cuenta como se muestra en la figura 4.4.



```
91 2015/01/31 02:21:10.56981600 - ENV NUE :
92   'destinatario' : 'pp',
93   'ip' : '192.168.0.115',
94   'mensaje' : 'acr',
95   'parametros' :
96   (
97     'calling_party_number' : '0992029715',
98     'acr' : 'ACR(27)',
99     'EBCSM' : 'EBCSM(0disconnect)',
100    'called_party_number' : '0992657483',
101  )
```

Figura 4.5 Mensajes ACR y EBCSM hacia el sistema prepago

En la figura 4.5 se puede observar como en los casos de llamadas la MSC notificó, de la manera esperada, a la PP sobre los segundos acumulados que utilizó el usuario A durante toda la comunicación y el reporte del evento por el que concluyó la conexión.

4.2 Análisis de Resultados

Después de la implementación y posterior simulación, se indican los aspectos más relevantes de los resultados obtenidos.

4.2.1 Topología de la Implementación Total del Proyecto

A la hora de la presentación final del proyecto se busca incorporar las dos MSC con protocolo MAP que se encargan de simular la

localización de un abonado que se puede encontrar en uno de los sitios establecidos (Quito o Guayaquil), y además el HLR, que posee una base de datos con la información de distintos números. Adicionalmente a esto, las dos MSC MAP se enlazan con un *router* que emula a una Estación Base (BS por sus siglas en inglés).

Se juntan en una sola maquina las MSC con protocolos CAMEL e ISUP y se le asigna el otro *router* (Guayaquil) de forma inalámbrica, que sirve como Estación Base. Esta MSC unificada ahora procede a comunicarse de forma alámbrica con el *switch* central mediante una IP fija. Las dos MSC con protocolo MAP también se juntan en un solo computador y operan con el otro *router* (Quito) por *wifi*; esta MSC a su vez se conecta también de forma inalámbrica con el HLR para la verificación de datos del usuario.



Figura 4.6 Routers para la conexión inalámbrica

Adicionalmente todos los elementos se conectan por cable a un *switch* central que es el encargado de recrear una red común a todos los dispositivos.



Figura 4.7 *Switch* con los puertos LAN utilizados

En la figura 4.8 se aprecia la red correctamente implementada, con los módulos, *routers*, *switch* y su debida conexión anteriormente explicada.

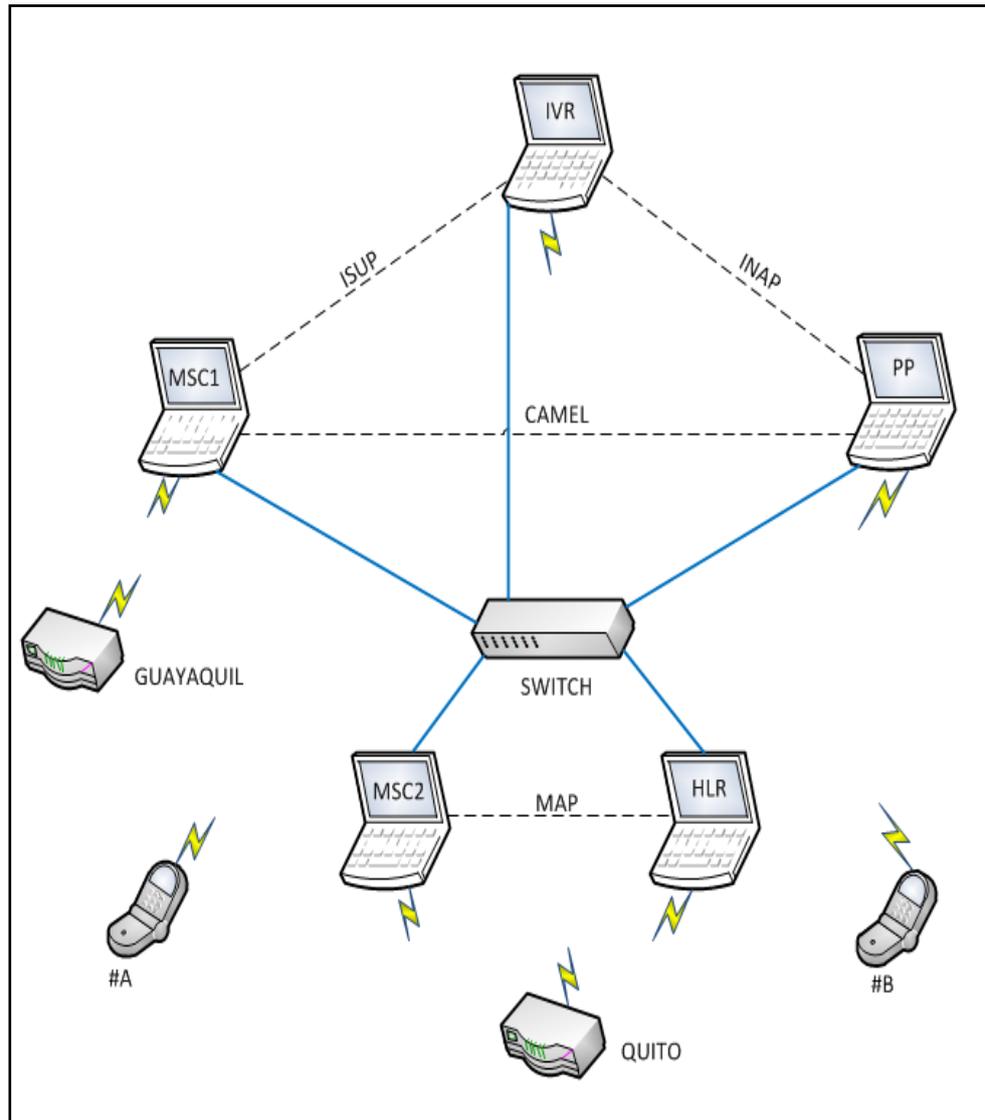


Figura 4.8 Topología Final de Red

4.2.2 Interfaz Gráfica de Módulos Implementados



Figura 4.9 Interfaz gráfica de la Central

En la figura 4.9 se aprecia la imagen con la que se representa al módulo de la Central de Conmutación Móvil dentro de la interfaz desarrollada.



Figura 4.10 Presentación de Interfaz

Desde el menú de inicio en el panel de control, se tiene acceso a la señalización completa, como se puede ver en la figura 4.10, en lo que respecta a la Red de Telefonía simulada.

Estas interfaces fueron creadas para obtener información más detallada de cada uno de los mensajes de manera fácil y con la interacción de los componentes, ya que se visualiza los parámetros enviados y recibidos y la hora en la que se realizó la llamada.

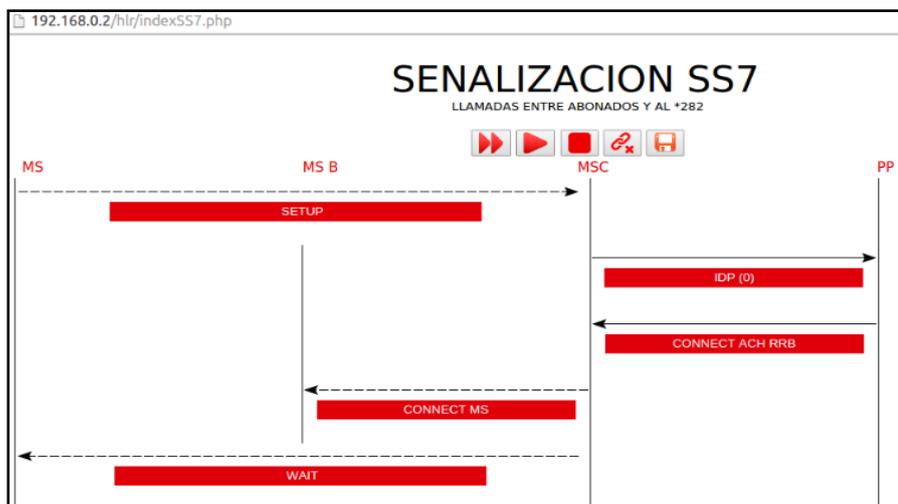


Figura 4.11 Captura 1 Tramo de llamada

En la figura 4.11 se observan partes iniciales de llamadas, realizadas por la implementación final del proyecto y obtenidas gracias a la interfaz previamente desarrollada.



Figura 4.12 Captura 2 Tramo de llamada

Momento en el que el usuario B acepta la llamada y se produce la tarificación en tiempo real con notificaciones, entre la Central y el sistema prepago como se indica en la figura 4.12.

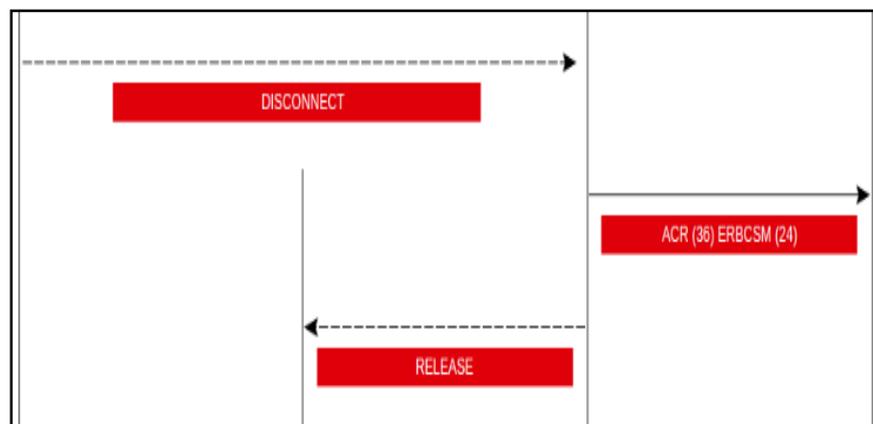


Figura 4.13 Captura 3 Tramo de llamada

En la figura 4.13 se muestra cómo el número llamante procede al cierre de una llamada y esto conlleva a que la MSC notifique la acumulación de segundos y el evento por el cual finalizó la comunicación.

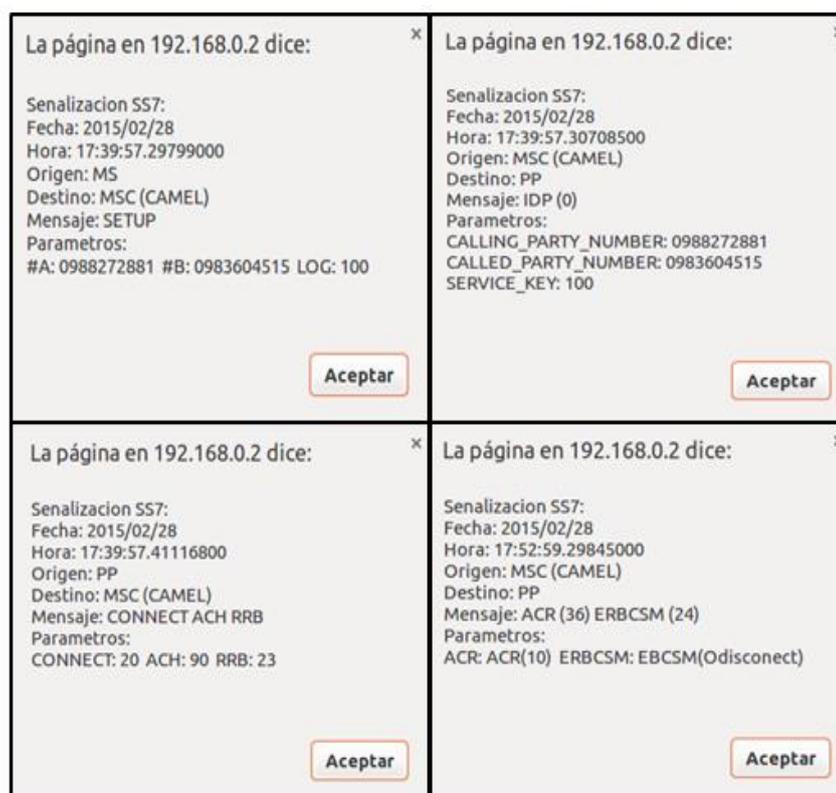


Figura 4.14 Parámetros vistos desde la aplicación

Al hacer *click* izquierdo sobre los mensajes enviados y recibidos por la central, automáticamente aparece una ventana que muestra los parámetros y la hora en que se produjeron. Estos parámetros son los que manejan los mensajes de señalización, explicados anteriormente en el capítulo 2; los cuales muestran los atributos tanto enviados como recibidos, que hacen posibles los casos mencionados durante todo el proyecto.

4.3 Discusión

El trabajo de implementar Aplicaciones SS7 con que trabajan las operadoras móviles es muy amplio, y sujeto a varios problemas en su consecución final, de los cuales se habla a continuación:

Una característica importante fue crear dos Redes inalámbricas para simular que el abonado A podía encontrarse en diferentes lugares geográficos (Quito y Guayaquil) y al encender su teléfono, iniciar una búsqueda y autenticación de los datos del terminal, para así poder realizar una conexión con cualquiera de las dos MSC, lo cual se consiguió con el éxito esperado. Pero a su vez, el abonado A se encuentra en una base de datos, la cual se creó además con números postpagos y de otras operadoras, así que al utilizarlos como números llamantes, tanto a los postpagos como a los abonados de Movistar y CNT se les realiza un desvío de llamada de la misma manera, mostrando una misma señalización para estos casos planteados, no se lo hizo de una manera diferenciada.

Otro aspecto fue de que a la hora de recrear la Central de Conmutación Móvil sobre la plataforma de Linux, el software soportó de manera adecuada, la programación de líneas de código donde se busca señalar las características que tiene la MSC como bases de datos para acumulación de tiempo hablado o activación de ciertos *prompts* que tienen que ver con error en la marcación de números no habilitados en la

Central, pero la comunicación con los celulares no es directamente con la MSC, ya que los terminales manejan otro tipo de protocolos que le permiten comunicarse primero con las Estaciones Base o las Estaciones Transceptoras, así que se buscó dar facilidades a la hora de programación y buscar los mensajes que si manejan los usuarios con los sistemas prepagos pero en forma de notificaciones o confirmaciones, para validar la comunicación.

También se presentaron fallos en el envío y recepción de los mensajes a lo largo de las pruebas de comunicación con todos los elementos de la Red, ya que la comunicación se manejaba únicamente con un *router* que se utilizaba para todos los componentes y se producía conflictos de ip; así que se tuvo que crear dos Redes inalámbricas y en un *switch* una Red LAN, para los diferentes tramos de la señalización que se producía al conmutar todos los archivos para los distintos casos de llamadas.

CONCLUSIONES Y RECOMENDACIONES

Conclusiones

El análisis de lo obtenido, es de alta utilidad para personas que se interesen en recrear los escenarios de este proyecto orientado a una Red de Telefonía Celular, en el área de Telecomunicaciones.

1. CAMEL, como Aplicación de Lógica Mejorada en su fase 2, se considera altamente útil en una Red de Telefonía Celular, al facilitar la comunicación entre nodos de llamadas como las funciones SSF y SCF, que sirven para facturar en tiempo real los servicios de voz y que se encuentran en la MSC y el sistema prepago respectivamente; importantes en el desarrollo del aplicativo de los módulos.
2. El protocolo CAMEL contiene en su funcionabilidad mensajes de señalización que van directamente relacionados a escenarios de llamadas entre dos usuarios, brindando facilidad en su interacción y con el sistema prepago; ya que dentro de sus atributos se

encuentran especificados eventos que van desde la conexión hasta el cierre de la misma, notificando las causas y si ha sido por parte el número llamante o el número llamado o confirmaciones de saldo que posee el usuario y si puede establecer comunicación.

3. La simulación de la MSC cuenta con cierto grado de complejidad, puesto que al usarse en una implementación con varios elementos de una IN; su funcionamiento individual depende de todo el conjunto de módulos encargados de la señalización en las llamadas, necesitándose una adecuada topología de red.
4. La Central de Conmutación Móvil virtuales una de las partes fundamentales en una Red Celular, porque procede a realizar la interpretación de requerimientos de los abonados que maneja otro tipo de protocolos, dando paso a la comunicación con el resto de las máquinas.
5. Al utilizar CAMEL en la implementación de la MSC, se ha obtenido una gran similitud a los parámetros y atributos que se manejan en un flujo de señalización real, pero el emulador no puede recrear el escenario total de las funciones de este protocolo como redireccionamientos o llamadas en espera; por eso se ha buscado introducir las más esenciales a la hora de la interacción, tales como tarificación y eventos de llamadas.

RECOMENDACIONES

Las recomendaciones mencionadas a continuación se proyectan para un debido proceso de diseño e implementación, y para proporcionar un alcance adecuado de este tipo de aplicaciones.

1. Al momento de proyectar en cada módulo la interfaz gráfica, se debe tener en consideración que se guarda toda la señalización de los escenarios de llamadas, por lo tanto cuando se presente el flujo de mensajes, hay que recurrir al botón *stop* y luego *play* de nuestro menú de inicio; de esta manera no habrá confusión entre el caso que se está presentando con los anteriores.
2. Cuando se busca hacer un fichero para crear la interfaz gráfica a partir de la programación desarrollada en *NETBEANS*, se debe verificar que no se repitan nombres de los mensajes tomados, ya que en la señalización presentada en cada computador, habrá conflicto de mensajes repetidos y no aparecerán sus parámetros correspondientes.
3. Es necesario manejar mucha información concerniente a la plataforma *LINUX* y su lenguaje de Hipertexto *PHP*, ya que se encuentran muchas herramientas útiles para la creación de las líneas de códigos, además de ficheros o bases de datos para la implementación; pero se requiere permisos correspondientes, como

la función *chmod* para la ejecución de archivos dentro de este software de aplicación.

4. Se debe crear módulos de simulación para los subelementos de una Red Pública que interaccionan directamente con el celular, como las Estaciones Bases o Transceptoras, que además están relacionados a una Central de Conmutación Móvil; ya que en éstos se manejan distintos tipos de protocolos desde que el requerimiento sale del número llamante hasta que llegan a la MSC, por lo cual resulta difícil manejar toda la información únicamente en la Central.
5. Si se espera contar con un mayor alcance en lo que respecta a conexión desde y hacia cualquier teléfono móvil, se sugiere crear redes paralelas que simulen el funcionamiento de otras Operadoras; específicamente por la validación que realizan las Centrales de cada compañía móvil a los números llamantes, así no se corre riesgos de que el programa elaborado deje de funcionar de manera correcta o arroje señalización que no debería.

BIBLIOGRAFÍA

- [1]Prezi, Tecnología de telefonía. Sistema de Señalización No. 7,SS7, <https://prezi.com/h7t8cp9zvzbz/tecnologia-de-telefoniasistema-de-senalizacion-no-7-ss7/>, fecha de consulta Noviembre 2014.
- [2] Uzcátegui José Antonio, Arquitectura y Red Telefónica, <http://arquitectura-y-senalizacion-ss7.webnode.com.ve/news/primer-blog/>, fecha de consulta Noviembre 2014.
- [3] Escuela Superior Politécnica del Litoral, Año 2010, pag 15, Implementación del protocolo SS7 sobre conexiones de servidores Asterisk.
- [4]Signaling System No. 7 (SS7/C7):Protocol, Architecture, and Services, Cisco System 2003, fecha de consulta Noviembre 2014.
- [5] Hernández Roberto, Metodología de la Investigación, Quinta Edición, fecha de consulta Noviembre 2014.
- [6] Signaling System No. 7 (SS7/C7):Protocol, Architecture, and Services, Cisco System 2003, fecha de consulta Diciembre 2014.
- [7]SSP STP SCP, Telecom Space,www.telecomspace.com/forum/SS7/sspstpscp, Fecha de consulta, Diciembre 2014.
- [8] Dialogic, SS7 Protocols,Publication Date: June 2008, fecha de consulta Diciembre 2014.

- [9] Signaling System No. 7 (SS7/C7):Protocol, Architecture, and Services, Cisco System 2003, fecha de consulta Diciembre 2014.
- [10] Universidad del Cauca, Sistemas de Conmutacion, pag. dtm.unicauca.edu.co/pregrado/conmutacion/transp/5-TelefoniaMovil.pdf, fecha de consulta Diciembre 2014
- [11] NSS,Scribd, <https://es.scribd.com/doc/66244372/3/Subsistema-de-Conmutacion-de-Red-NSS>, fecha de consulta Diciembre 2014.
- [12] Noldus Rogier, Intelligent Networks for the GSM, GPRS and UMTS Network, fecha de consulta Enero 2015
- [13] IEEE LATIN AMERICA TRANSACTIONS, Evolución de la plataforma de servicios CAMEL más allá de los sistemas 3G, http://www.ewh.ieee.org/reg/9/etrans/ieee/issues/vol04/vol4issue5Sept.2006/4TLA5_02LopezLavado.pdf, fecha de consulta Enero 2015
- [14] Noldus Rogier, Intelligent Networks for the GSM, GPRS and UMTS Network, fecha de consulta Enero 2015
- [15]Ciberaula, Todo sobre GNU/LINUX, http://linux.ciberaula.com/articulo/que_es_linux/, fecha de consulta Enero 2015
- [16] Linux, Mindomo, <https://www.mindomo.com/es/mindmap/linux-97226adf56a94ded966f6892056e5907>, fecha de consulta, Enero 2015

- [17] Pasos para instalar Ubuntu, LWP, www.lawebdelprogramador.com/foros/linux/1458115-Pasos-para-instalar-ubuntu-14.04-LTS.html, fecha de consulta Enero 2015
- [18] XAMPP 1.8.3-4 en Ubuntu Linux 14.04 LTS, ConGPL, <https://congpl.wordpress.com/2014/08/21/como-instalar-xampp-1-8-3-4-en-ubuntu-linux-14-04-lts/>, fecha de consulta Enero 2015
- draft-hou-sigtran-tua-00, <http://www.openss7.org/internet-drafts/draft-hou-sigtran-tua-00.txt>, fecha de consulta Enero 2013
- [19] Usuario Ubuntu, como instalar Netbeans, <http://usuarioubuntu.com/como-instalar-netbeans-en-ubuntu-14-04-y-14-10/>, fecha de consulta Enero 2015
- [20] Postman, Postman-REST Client, <https://chrome.google.com/webstore/detail/postman-rest-client-packa/fhbjgbiflinjbdggehcdcbncdddop>, Fecha de consulta Enero 2015
- [21] PHPMYADMIN, Sourceforge, <http://sourceforge.net/projects/phpmyadmin/>, fecha de consulta Enero 2015.
- [22] WIRESHARK,Acrylic, <https://www.acrylicwifi.com/blog/capturar-traffic-wifi-con-wireshark-en-windows/>, fecha de consulta Enero 2015.