

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 UN
SISTEMA DE RED INTELIGENTE PREPAGO PARA LOS SERVICIOS DE
VOZ A TRAVÉS DE LA SEÑALIZACIÓN DE APLICACIONES
PERSONALIZADAS PARA REDES MÓVILES DE LÓGICA MEJORADA,
CAMEL.”

INFORME DE MATERIA DE GRADUACIÓN

PREVIO A LA OBTENCIÓN DEL TÍTULO DE:

INGENIERO EN ELECTRÓNICA Y TELECOMUNICACIONES

PRESENTADO POR:

Julio Ricardo Prócel Contreras

Marco Antonio Valdivieso Urbina

GUAYAQUIL – ECUADOR

2015

AGRADECIMIENTO

Agradezco primeramente a Dios por permitirme terminar de una muy excelente manera esta etapa muy importante de mi vida.

A mis padres por confiar en mí y brindarme todo su apoyo, que ha sido muy incondicional de su parte.

A mi hermano que me ha acompañado en todos mis años de estudio siendo muy importante en mi vida.

A Ani que desde el inicio de este proyecto estuvo ayudándome y dándome todo su apoyo hasta el final de este proceso.

Al Ingeniero José Miguel que me supo encaminar y guiar de una excelente manera en este proyecto muy importante de mi carrera.

Julio Prócel Contreras

Al culminar esta etapa académica me siento muy orgulloso y agradecido, este logro alcanzado lo debo a muchas personas que estuvieron constantemente motivando y brindándome todo su apoyo. Por ello quisiera expresar mis agradecimientos a:

Dios por todas sus bendiciones recibidas.

A mis padres y hermano que son mi pilar fundamental, por todo su apoyo y confianza para la culminación de esta etapa universitaria.

A mi esposa, Diana, por brindarme su apoyo incondicional y por estar al cuidado de nuestro hijo mientras terminaba mis estudios universitarios.

A mi tutor Msc. José Miguel Menéndez por toda su paciencia y ayuda para la realización de este proyecto.

Marco Valdivieso

DEDICATORIA

Dedico este logro a mis padres Julio Prócel y Lilia Contreras, por darme la oportunidad de cumplir con toda normalidad esta meta que me fijé desde que entre a ESPOL y así convertirme en el profesional y la persona que soy.

Julio Prócel Contreras

Este logro quiero dedicar a mis padres Marco Valdivieso y Gloria Urbina, por todo su esfuerzo que me han brindado durante muchos años en especial en mi etapa universitaria.

A mi esposa, por su comprensión y apoyo, y de manera muy especial a mi hijo Marco Sebastián por ser el motivo de todo mi esfuerzo y perseverancia.

Marco Valdivieso

TRIBUNAL DE SUSTENTACIÓN

Msc. José Miguel Menéndez

PROFESOR DE LA MATERIA DE GRADUACIÓN

Msc. Washington Medina

PROFESOR DELEGADO POR LA UNIDAD ACADEMICA

DECLARACIÓN EXPRESA

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

(Art. 12 del Reglamento de Graduación de la ESPOL)

Julio Ricardo Prócel Contreras

Marco Antonio Valdivieso Urbina

RESUMEN

La realización de este proyecto tiene como finalidad la creación de un simulador de telefonía celular enfocado en la plataforma de servicios de Prepago, que tiene la función de debitar el saldo disponible de los abonados, según el tiempo de duración de las llamadas que realice sujetándose a los costos que conlleva las llamadas a diferentes operadoras.

El protocolo de señalización de Aplicaciones Personalizadas para Redes Móviles de Lógica Mejorada, CAMEL por sus siglas en inglés, permitirá interactuar con las demás plataformas que posee la red celular para la correcta utilización de la plataforma prepago. En este proyecto Prepago ofrece los servicios como son llamadas al *282 para consultar saldo o realizar una recarga de saldo, de la misma manera como sucede en la vida real por medio de códigos de tarjeta; otro servicio es la realización de llamadas entre la misma operadora; Prepago informará constantemente a la MSC que es la central inteligente de la red celular, el saldo disponible de los abonados y cuando pueden o no realizar llamadas, en el caso de tener saldo el abonado la MSC continuará la señalización con las demás plataformas caso contrario Prepago envía a la MSC a desconectar la llamada .

ÍNDICE GENERAL

AGRADECIMIENTO	II
DEDICATORIA.....	IV
TRIBUNAL DE SUSTENTACIÓN.....	V
DECLARACIÓN EXPRESA.....	VI
RESUMEN	VII
ÍNDICE GENERAL.....	VIII
ABREVIATURAS Y SIMBOLOGÍA.....	XI
ÍNDICE DE FIGURAS.....	XIII
INTRODUCCIÓN	XV
CAPÍTULO 1	1
1 ANTECEDENTES Y JUSTIFICACIÓN.....	1
1.1 Antecedentes de SS7.....	1
1.2 Justificación del uso de un simulador de una llamada celular.....	4
1.3 Descripción del Proyecto	4
1.3.1 Objetivo general.....	5
1.3.2 Objetivos específicos	6
1.4 Alcance	6
1.5 Limitaciones.....	6
1.6 Metodología	7

CAPÍTULO 2.....	9
2 FUNDAMENTOS TEÓRICOS.....	9
2.1 Sistema de Señalización No. 7 (SS7).....	9
2.1.1 Arquitectura de SS7.....	11
2.1.2 Capas del modelo SS7.....	13
2.2 Camel fase 2 y Arquitectura.....	16
2.2.1 GSM-SCF.....	18
2.2.2 MCS / gsmSFF.....	18
2.3 Control de llamadas por Camel fase 2 y tarificación.	18
2.3.1 Control de carga On-line.	19
2.3.2 On-line charging.....	19
2.3.3 Off-line charging.....	19
2.4 Control de duración de una llamada.	20
2.5 Sistema Operativo.	23
2.5.1 Sistemas Operativo Linux	23
2.5.2 Ubuntu.....	24
2.6 Lenguaje abierto PHP.....	25
CAPÍTULO 3.....	27
3 DESARROLLO DEL PROYECTO.....	27
3.1 Requerimientos necesarios para la implementación.	27

3.1.1	Hardware.....	27
3.1.2	Software.....	28
3.2	Estructura básica del funcionamiento de los mensajes prepago.	35
3.2.1	Mensajes que utiliza Prepago para una llamada entre terminales.....	35
3.2.2	Mensajes que utiliza Prepago para una llamada al *282.....	45
3.3	Conectividad entre componentes.	53
3.4	Pasos a tener en cuenta al realizar los scripts.....	55
3.5	Resumen para la implementación del Módulo Prepago.....	56
4	Análisis de Resultados.....	58
4.1	Resultados obtenidos.	58
4.2	Interpretación de resultados esperados.....	59
4.3	Interfaz gráfica del módulo Prepago	59
4.4	Análisis de la topología de conexión empleada.....	67
4.5	Discusión.....	69
4.6	Futuro uso del módulo prepago desarrollado.	70
	CONCLUSIONES	72
	RECOMENDACIONES.....	74
	BIBLIOGRAFÍA.....	76

ABREVIATURAS Y SIMBOLOGÍA

SS7	Sistema de Señalización Número Siete
UITT	Unión Internacional de Telecomunicaciones
PHP	Preprocesador de Híper Texto
ISUP	Parte de Usuario de la Red Digital de Servicios Integrados.
INAP	Parte de aplicación de red inteligente.
MAP	Parte de Aplicación para Móviles.
CAP	Parte de Aplicación Camel
SCCP	Parte de Control de la Señalización de Conexión
CAMEL	Lógica mejorada para aplicaciones móviles personalizadas.
TCAP	Parte de aplicación para capacidades de transacción.
MTP	Parte de transferencia de mensajes.
MTP-1	Parte de transferencia de mensajes nivel 1
MTP-2	Parte de transferencia de mensajes nivel 2
MTP-3	Parte de transferencia de mensajes nivel 3
AIN	Red Inteligente Avanzada
HLR	Registro Local de Ubicación.
IVR	Respuesta de Voz Interactiva
SMSC	Centro de Servicios de Mensajes Cortos

IN	Red Inteligente
OSI	Interconexión de Sistemas Abiertos
REL	Mensaje de Liberación
SSP	Punto de conmutación de servicio
STP	Punto de Transferencia de Señal
SCP	Punto de control de servicio

ÍNDICE DE FIGURAS

Figura 2.1. Componentes de una red SS7 [5].....	11
Figura 2.2. Capas del Modelo SS7 [8]	13
Figura 2.3. Arquitectura de Camel Fase 2 [10]	17
Figura 2.4. Proceso de Control de Llamada [14].....	22
Figura 3.1 Asistente de instalación de Ubuntu [16].....	29
Figura 3.2 Asistente de Ubuntu finalizando la instalación [17].....	30
Figura 3.3 Ventana principal del servidor XAMPP [18]	32
Figura 3.4 Ventana principal de Netbeans IDE [19].....	34
Figura 3.5 Diagrama de Interacción de Objetos. Flujo en una llamada celular, el número llamante queda sin saldo.	39
Figura 3.6 Diagrama de Interacción de Objetos. Flujo en una llamada celular, número llamante corta en cualquier instante.	41
Figura 3.7 Diagrama de Interacción de Objetos. Flujo en una llamada celular, número destino corta en cualquier instante.	42
Figura 3.8 Diagrama de Interacción de Objetos. Flujo en una llamada celular número llamante corta antes que conteste el número destino.....	43
Figura 3.9 Diagrama de Interacción de Objetos. Flujo en una llamada celular número llamante no dispone de saldo.	44
Figura 3.10 Diagrama de Interacción de Objetos. Flujo en una llamada celular número destino no contesta llamada o desvía la llamada.	45
Figura 3.11 Llamada al *282 para consultar saldo disponible.....	49

Figura 3.12 Llamada al *282 a consultar saldo y el usuario corta la llamada.	50
Figura 3.13 Llamada al *282 para ingresar una tarjeta de recarga.	52
Figura 3.14 Comando para alzar servidor Apache.....	54
Figura 3.15 Envío de request correcto del script implementado.	54
Figura 3.16 Envío de request incorrecto del script implementado.	55
Figura 4.1 Panel de Control del módulo Prepago.	60
Figura 4.2 Ventana para visualizar los mensajes de señalización.	61
Figura 4.3 Ventana para visualizar las tarjetas prepago con su respectivo valor y estado.....	62
Figura 4.4 Ventana para visualizar las tarjetas prepago con su respectivo valor y estado.....	63
Figura 4.5 Ventana para visualizar los números que pertenecen al servicio prepago.....	64
Figura 4.6 Ventana para visualizar el número con su respectivo saldo con la opción de modificar el saldo si se desea.	65
Figura 4.7 Ventana para visualizar la tarifa de facturación por minuto de la llamada.	66
Figura 4.8 Ventana para modificar la tarifa de facturación de la llamada.	67
Figura 4.9 Representación de la conexión empleada para la simulación de la red celular.	69

INTRODUCCIÓN

En la actualidad la gran demanda de nuevos servicios por parte de los usuarios ha permitido que las operadoras celulares evolucionen su infraestructura lo que conlleva a la utilización de redes inteligentes, para soportar el gran número de usuarios y servicios, logrando una comunicación global y permitiendo satisfacer los servicios demandados como son: mayor ancho de banda para aplicaciones como video, compras online, video conferencias, llamadas roaming.

Para lograr la comunicación entre la red de la operadora móvil y las Redes Inteligentes se han establecido estándares de comunicación y sistemas de señalización, para que exista compatibilidad entre dichas redes. Uno de estos sistemas de señalización es el Sistema de Señalización número 7 (SS7), el cual es el más usado en redes modernas puesto que su diseño permite soportar varios servicios, dentro del cual existen protocolos de señalización que ayudan a la conexión de las Redes Inteligentes.

El protocolo de señalización a estudiar es el de Aplicaciones Personalizadas para Redes Móviles de Lógica Mejorada, CAMEL por sus siglas en inglés. Una de sus funciones ampliamente usada es la parte de facturación de la llamada y es donde nos enfocaremos, escenario donde un abonado de telefonía móvil realiza una llamada celular bajo el perfil de abonado prepago, es ahí donde este entra en acción permitiendo controlar dicha llamada.

La gran mayoría de las operadoras celulares utilizan CAMEL dentro de su red celular por ende la importancia de estudiar este protocolo de señalización que es muy importante por los servicios que brinda. El sistema de prepago ofrece a los abonados la facilidad de pagar de manera anticipada el uso de los servicios, por lo que se requiere supervisión en tiempo real. Es por esto que prepago provee a la MSC tarifas e información de tiempos para el correcto control de la duración de las llamadas [1].

En el capítulo uno se mencionará los objetivos del proyecto, el alcance y limitaciones del simulador además de la metodología empleada para la realización del mismo. En el capítulo dos se realizará el marco teórico: en donde la literatura permite profundizar los conceptos que encierra al tema y

que son relevantes para el correcto funcionamiento e interacción con las demás partes que conforman nuestro proyecto. Este capítulo es de suma importancia puesto que al profundizar dentro del tema nos encaminaremos de una forma correcta para llegar al cumplimiento de nuestros objetivos.

Después de haber comprendido correctamente el tema y recolectado la información más destacada con la ayuda del marco teórico, se procederá con el desarrollo del capítulo tres, se realizará una guía para personas que deseen implementarla, se indicará los elementos empleados para la programación de la plataforma de Prepago y la importancia que tienen los mensajes de Prepago tanto los que envía como los que recibe, aquellos que permiten la señalización y el intercambio de información entre los diferentes protocolos de red inteligente para lograr con éxito una llamada celular.

Una vez logrado todo lo expuesto anteriormente, se elaborará el informe final de tesis con las conclusiones logradas y las recomendaciones para las futuras investigaciones que se puedan generar con el trabajo realizado.

CAPÍTULO 1

1 ANTECEDENTES Y JUSTIFICACIÓN

1.1 Antecedentes de SS7.

La tecnología celular en sus inicios fue creada únicamente para transmisión y recepción de voz, el aumento de usuarios en las operadoras celulares y a la aparición de nuevos servicios como lo son aquellos que utilizan datos, audio y video, ha permitido que la telefonía celular evolucione junto a la aparición de terminales de usuarios de mayor potencia para poder ofrecer de manera eficaz y segura la información a través de la red celular. Al mejorar la velocidad del tráfico celular y al acelerar las descargas en los terminales de los usuarios

permite que las operadoras dispongan de más usuarios por cada celda disponible.

Toda la red celular se basa en la señalización, se encuentra presente en todos los niveles de las redes de telecomunicaciones y de telefonía, en esta última permite establecer, conectar, desconectar y facturar las llamadas. En otras palabras el papel principal de la señalización ha sido de controlar el tráfico, acceder a las bases de datos y de gestionar la red.

Existen dos tipos de señalización como lo son la señalización de canal asociado (CAS) y señalización por canal común (CCS). Las desventajas de CAS es que no ofrece seguridad, es vulnerable al fraude, existe interferencia en los tonos de señalización. En cambio la señalización por canal común utiliza una ruta solo para la señalización, la troncal de voz solo se utiliza una vez que existe conexión, los tiempos para establecer las llamadas son muy cortos puesto que sus recursos son utilizados de forma más eficiente, estas ventajas mencionadas que posee la señalización por canal común ha permitido llegar hasta la señalización por canal común número 7.

Los sistemas antiguos de señalización utilizaban la misma ruta física tanto para la señalización de control de llamadas como para la llamada en tiempo real, por lo que estos métodos eran ineficientes y actualmente se encuentran siendo sustituido por técnicas de canal común o también conocidos como fuera de banda.

Los sistemas de señalización también han evolucionado con el crecimiento de las redes y de la tecnología celular, hasta llegar al Sistema de Señalización Canal Común Número Siete (SS7), su principal característica en relación a los antiguos sistemas de señalización es que proporciona inteligencia a la red celular, permite la realización de llamadas de una manera más rápida, la organización responsable de su administración es la Unión Internacional de Telecomunicaciones, con sus siglas en ingles UITT [2].

SS7 al ser un protocolo de alto rendimiento y de alta velocidad, permite comunicar grandes cantidades de información al establecerse la llamada, durante y al final de la llamada, esto ha permitido que se desarrollen nuevos servicios relacionados con las llamadas como por ejemplo desvío de llamadas, correo de voz, llamada en conferencia, llamada en espera, filtrado de llamadas, ocupado. La señalización SS7 se utiliza sobre todo para la señalización entre centrales telefónicas.

1.2 Justificación del uso de un simulador de una llamada celular.

Con la realización de un simulador se pretende dar a conocer a las personas interesadas en el tema de las telecomunicaciones especialmente en el área de la red celular, los procedimientos que existen dentro de dicha red para la realización de una llamada exitosa, sea para una llamada prepago o para consultar saldo; estos dos casos mencionados son los únicos que serán abarcados para nuestro simulador.

La gran mayoría conoce como realizar una llamada celular pero no los diferentes protocolos encargados de establecer dicha llamada, he aquí la gran importancia del proyecto, con la ayuda del lenguaje de programación PHP se procederá a programar dichos protocolos con los mensajes más importante que cada uno posee para establecer una correcta señalización y lograr con éxito una comunicación.

1.3 Descripción del Proyecto

Con la importancia por parte de las operadoras celulares por brindar mejores servicios de voz en sus redes inteligentes se procederá a analizar e implementar en el lenguaje abierto Preprocesador de Hipertexto (PHP) sobre Linux un sistema de red inteligente prepago para los servicios de voz a través de la señalización de Aplicaciones Personalizadas para Redes Móviles de Lógica Mejorada, CAMEL.

Consistiendo básicamente en estudiar, implementar y simular cómo funcionan los servicios de voz sobre CAMEL, cuando se realiza una llamada prepago tanto para *onnet (llamada entre abonados de la misma operadora por ejemplo: Claro-Claro)* así como también una llamada al sistema de respuesta de interacción por voz (IVR), *282 ya sea para consulta de crédito o recarga del mismo.

Primero se estudiará la aplicación de Camel, el funcionamiento en sí del sistema, para proceder con la implementación del mismo usando los parámetros necesarios con la finalidad de que se asemeje a la realidad del funcionamiento de Camel. En la simulación se podrá apreciar el funcionamiento del protocolo Camel, analizando las operaciones realizadas por dicho protocolo, mientras se realiza la llamada, así se podrá entender de una forma más didáctica el funcionamiento del mismo mientras existe la comunicación.

1.3.1 Objetivo general.

Describir el funcionamiento y operación del protocolo de señalización CAMEL fase 2.

Describir los servicios que las operadoras móviles ofrecen gracias a sus sistemas prepago basado en CAMEL.

Determinar el alcance del protocolo de señalización CAMEL en una red telefónica.

1.3.2 Objetivos específicos

1. Implementar una plataforma prepago virtual de servicios de voz para abonados móviles de una operadora celular.
2. Determinar el esquema de cobro que utilizan las operadoras móviles para los abonados prepago.

1.4 Alcance

La realización de este proyecto el cual consiste en un simulador de señalización entre los diferentes protocolos de la red celular, ofrecerá una ayuda didáctica para las personas interesadas en el tema de las telecomunicaciones, se observará la importancia de cada protocolo, la interacción entre ellos y los posibles errores que puedan ocurrir para lograr una llamada celular y la correspondiente facturación. Posee una interfaz de usuario muy fácil de manejar y permite ver el desarrollo por los diferentes sistemas que interactúan.

1.5 Limitaciones

El inconveniente que posee un simulador es que imita el funcionamiento de un sistema, pero no reproduce perfectamente con

todas las variables que posee en la vida real debido a que existen aspectos que no se pueden simular.

Los estándares de los protocolos de señalización poseen una gran variedad de atributos, de los cuales solo se ha seleccionado los más importantes para llegar a los objetivos planteados, aquellos que permitieron enviar y recibir información entre los diferentes niveles del protocolo SS7 para realizar con éxito una llamada celular.

El simulador permite realizar llamadas entre la misma operadora, consulta de saldo del abonado e ingreso de tarjetas de recargas; en caso de que el abonado desee realizar una llamada a una operadora diferente se desconectará automáticamente la llamada.

1.6 Metodología

Se utilizará el método científico cualitativo del tipo descriptivo para enfocar el proyecto, este método de investigación permite conocer las características relevantes, analizando detalladamente su interacción. Se resumirá la información de manera cuidadosa y se analizará minuciosamente los resultados, a fin de extraer lo más esencial para contribuir al conocimiento [3].

Con este trabajo se pretende que sea una guía para la enseñanza didáctica para futuras investigaciones en lo que concierne a la señalización en la telefonía celular, se creará un simulador en donde se podrá explicar y apreciar de una forma gráfica el funcionamiento de la señalización celular, centrando la interacción con los diferentes nodos que conforman la red de telefonía celular.

CAPÍTULO 2

2 FUNDAMENTOS TEÓRICOS

2.1 Sistema de Señalización No. 7 (SS7)

SS7 utiliza un canal diferente al de voz, destinado específicamente a la señalización, permitiendo que exista un canal que transporte la voz y otro canal que se encarga de la señalización, ambos funcionan de manera independiente. Por su gran complejidad solo requiere una mínima intervención humana debido a que la mayoría de los procesos son automatizados y no requieren control de un operador [4].

La red de señalización por canal común número 7 es la más utilizada, tiene la particularidad de soportar la señalización de abonados telefónicos analógicos y digitales, su misión es de encaminar la información de control entre los elementos que posee un red de telecomunicación, como por ejemplo las bases de datos, servidores y conmutadores telefónicos, permite que la señalización se lleve a cabo en todo momento, aun cuando no existe una llamada establecida. Al usuario final le ofrece una gran variedad de servicios, entre ellos encontramos el identificador de llamadas, números gratuitos 1800.

SS7 ofrece una gran cantidad de ventajas que se detallan a continuación:

- Un enlace de señalización puede transportar información de muchos canales de voz y de datos.
- Controla la información de telefonía y datos, optimiza al máximo los canales de señalización.
- Ofrece confiabilidad en la transmisión de mensajes.
- Permite la señalización entre los diferentes elementos de la red sin importar tener una comunicación directa.

2.1.1 Arquitectura de SS7.

Para que la arquitectura SS7 sea robusta, la red deberá diseñarse de tal forma que ofrezca un alto grado de redundancia. De esta forma, cualquier problema que pudiera surgir en alguno de los enlaces, no causaría desastres en la red, llegando a ser una arquitectura confiable y rápida.

La Figura 2.1 muestra los componentes básicos de la red SS7 y la manera en que forman una red interconectada. Las principales características de cada elemento serán detalladas a continuación.



Figura 2.1. Componentes de una red SS7 [5].

SSP

Punto de conmutación de servicio (Service Switching Point), es un conmutador de central local que se utiliza para convertir la señalización recibida desde el conmutador de voz, aquí se procesa el tráfico de banda de voz (voz, fax, módem,...) y procede a realizar la

señalización. Un SSP puede originar y terminar mensajes, pero no puede transferirlos. Si un mensaje es recibido con un código que no coincide con el código de la SSP receptor, el mensaje se descarta [6].

STP

Punto de Transferencia de Señal (Signal Transfer Point), es el responsable de la transferencia de los mensajes SS7 con los otros nodos que posee, enruta mensajes a través de la red a sus destinos apropiados. STP usualmente se disponen en pares llamados pares acoplados y su importancia radica en la capacidad de asumir toda la carga en caso de que alguna falle, logrando alta fiabilidad y seguridad [7].

SCP

Punto de Control de Servicio (Service Control Point), concentran la mayor parte de la inteligencia de proceso de la red, administran las bases de datos con información sobre operaciones, mantenimiento y servicios suplementarios.

2.1.2 Capas del modelo SS7.

El protocolo SS7, es un conjunto de servicios y se encuentra dividido en capas y presenta similitudes con el modelo OSI. La Figura 2.2 muestra cuáles son las capas que integran el modelo SS7.

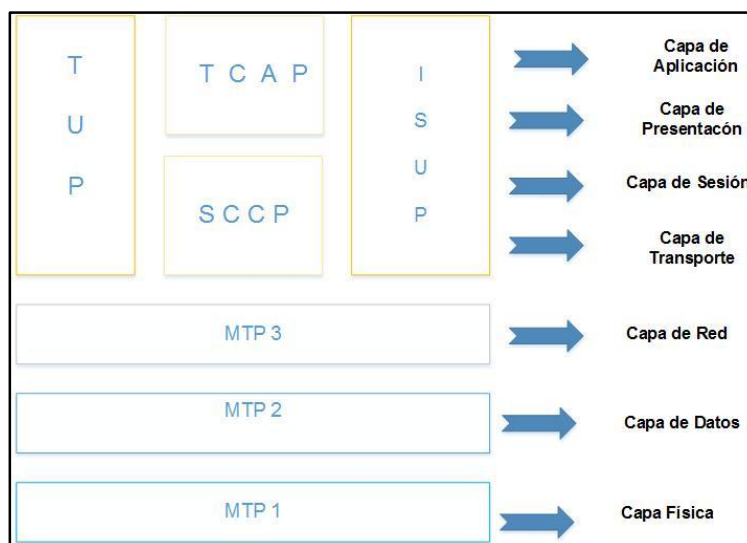


Figura 2.2. Capas del Modelo SS7 [8].

Parte de transferencia de mensajes del nivel 1 (MTP-1)

Los niveles MTP del 1 al 3 cumplen con la función de transportar la información desde su origen hasta su destino. MTP proporciona una transferencia de los mensajes de señalización de manera confiable. MTP-1 es la capa responsable de definir las características físicas y eléctricas en lo que concierne a los enlaces de la red de señalización además se encarga del mantenimiento del enlace físico.

Parte de transferencia de mensajes del nivel 2 (MTP-2)

Proporciona la funcionalidad de la capa de enlace. Se asegura de que los dos puntos extremos de un enlace de señalización fiable puedan intercambiar mensajes de señalización. Incorpora capacidades tales como la comprobación de errores, control de flujo, y el control de secuencia.

Parte de transferencia de mensajes del nivel 3 (MTP-3)

Su función es decidir el camino correcto para que los mensajes puedan ser entregados entre los puntos de señalización a través de la red. Incluye funciones tales como respuestas ante fallos de enlaces, respuesta a situaciones de congestión y bloqueo [9].

Parte de Señalización de Control de Conexión (SCCP)

El protocolo SCCP provee funciones adicionales de enrutamiento, control de flujo, segmentación, orientación a conexión y corrección de errores en el sistema de señalización número 7. En combinación con MTP, el SCCP puede transmitir mensajes que no son relacionados a circuitos. Estos mensajes son utilizados para soportar servicios adicionales como por ejemplo, servicios de llamadas sin costo, portabilidad numérica local y finalización de llamadas para

suscriptores ocupados en redes inteligentes así como movilidad, roaming y servicios de mensajes cortos (SMS) en las redes móviles.

Parte de Aplicación de Capacidad de Transacción (TCAP)

TCAP permite a las aplicaciones comunicarse dentro de la red SS7 utilizando instrucciones enviadas entre aplicaciones. Por ejemplo, cuando un abonado cambia de ubicación VLR en un sistema global para comunicaciones móviles, su HLR se actualiza con la nueva ubicación VLR por medio de un componente UpdateLocation.

Las aplicaciones más comunes son:

- Número gratuito (E800)
- Red Inteligente Avanzada (AIN)
- Aplicación del Protocolo de Red Inteligente (INAP)
- Las aplicaciones personalizables para móvil lógica mejorada (CAMEL)
- Parte de Aplicación Móvil (MAP)

TCAP está estructurado en dos sub-capas una de Componente y otra de Transacción.

Sub-capa componente

Una entidad emisora pide a la entidad receptora la ejecución de una operación; la entidad receptora interpreta la petición, ejecuta la

operación si la interpretación ha sido posible, e informa del resultado positivo o negativo de esta ejecución.

Sub-capa transacción

Este servicio permite a las dos entidades establecer, mantener y terminar los diálogos, e interaccionar entre ellas en el interior de un diálogo establecido intercambiando componentes.

Capa de Aplicaciones.

Los principales servicios que ofrece SS7 serán mencionadas a continuación:

SMSC.- Permite el uso del servicio SMS.

VLR.- Consiste en una base de datos temporal de los registros de los abonados de una operadora telefónica.

HLR.- Contiene la base de datos de los clientes de una operadora celular.

CAMEL.- Establece los servicios de una red inteligente exclusivamente para las redes GSM.

2.2 Camel fase 2 y Arquitectura.

CAMEL fase 2 tiene más funcionalidades para el control y tarificación de la llamada, se puede decir que es una mejora a CAMEL fase 1 sin olvidar que estos dos son protocolos diferentes.

El encargado de solicitar el establecimiento de CAMEL fase 1 o fase 2 es la MSC, y Prepago se encargará de relacionar con el mismo servicio, esto quiere decir, que si la MSC requiere Camel fase 2, prepago no podrá relacionarse con Camel Fase 1.

A continuación la Figura 2.3 muestra la arquitectura de CAMEL fase 2

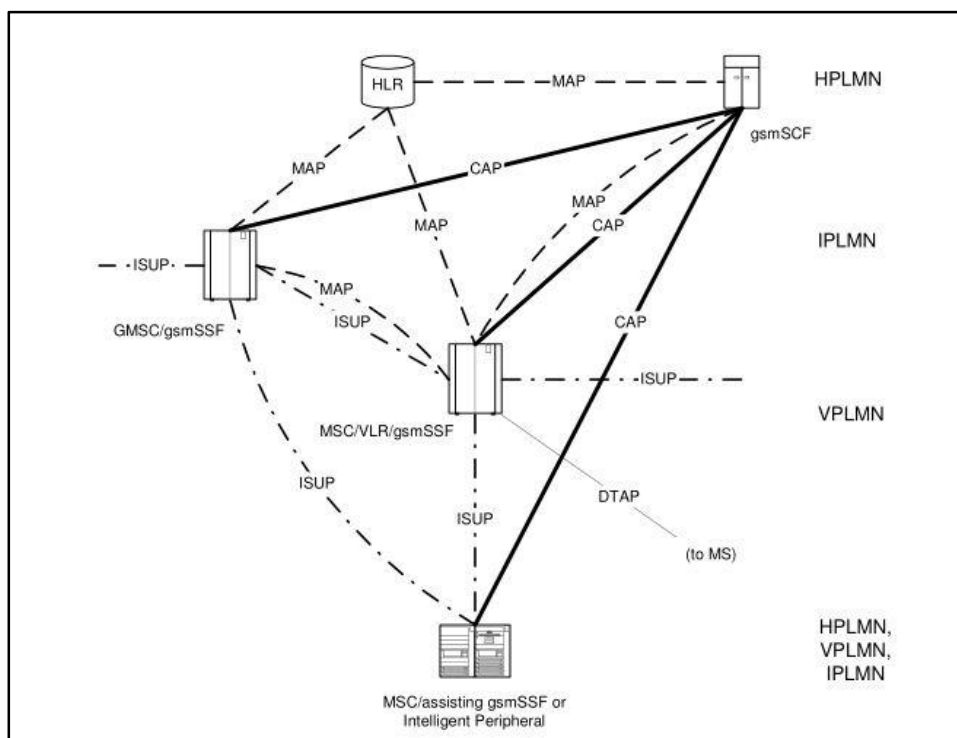


Figura 2.3. Arquitectura de Camel Fase 2 [10].

De esta arquitectura se enfocará en la MSC/gsmSSF y gsmSCF que representa a la parte de prepago en la cual se basa este estudio,

debido que para su comunicación utilizan CAP (Camel Application Part). También se hace referencia a las entidades de la arquitectura CAMEL fase 2 que se utilizará para simular la plataforma de prepago.

2.2.1 GSM-SCF.

El gsmSCF para este estudio se relaciona como prepago. CAMEL fase 2 puede responder a un establecimiento de relación de la MSC/gsmSFF de CAMEL fase 2 sobre una de CAMEL fase 1 si la MSC lo requiere. Aparte de la MSC/gsmSFF, el gsmSCF también puede establecer relaciones de Camel fase 2 con otras entidades como assistinggsmSFF o IntelligentPeriferial [11].

2.2.2 MCS / gsmSFF.

La MSC/gsmSFF de camel fase 2 puede enviar notificaciones suplementarias de servicio al gsmSCF y establecer relación de servicio con gsmSCF con Camel fase 2.

2.3 Control de llamadas por Camel fase 2 y tarifación.

Para el control de la llamada se usará (On-line charging control) que a su vez se relaciona con el control de duración de la llamada.

2.3.1 Control de carga On-line.

Control de carga On-line permite a Prepago controlar y supervisar la duración de la llamada mientras la llamada está en curso, que es lo que no hace Off-line charging [12].

2.3.2 On-line charging.

El servicio de CAMEL determina el nivel de carga de la llamada cuando se establece la llamada. El establecimiento de una llamada sólo se permitirá cuando el abonado tiene crédito suficiente en su cuenta. Cuando se establece la llamada, el servicio de Camel se encarga de monitorear y debitar el crédito de la cuenta del abonado. Si el abonado no posee crédito para continuar con la llamada el servicio de Camel tiene la potestad de terminar la llamada.

2.3.3 Off-line charging.

La tasa de carga de la llamada se determina una vez finalizada la llamada, a través del procesamiento de registros de detalles de llamadas (CDR). El MSC que está sirviendo al suscriptor genera un CDR que contiene todos los detalles de las llamadas pertinentes, como el tiempo, la ubicación, identificación del abonado, duración de la llamada, etc. Estos datos son utilizados para determinar el costo de la llamada y tarificar al abonado [13].

Un error muy común es pensar que prepago trabaja con on-line charging y que postpago trabaja con off-line charging, sin embargo prepago y postpago pueden trabajar con on-line charging. Los abonados de prepago tienen que contar con crédito de llamadas antes de establecer llamadas salientes o recibir llamadas entrantes (en roaming). El abonado puede utilizar el crédito para las llamadas entrantes y salientes. El sistema on-line charging monitorea las llamadas y reduce el crédito restante, de acuerdo con el índice de llamada (porque se puede hacer llamadas onnet y offnet) y duración de la llamada.

2.4 Control de duración de una llamada.

El principal mecanismo de CAMEL, control de duración de la llamada, es el monitoreo durante el periodo de la llamada en el MSC. Cuando se establece una llamada y el prepago ha logrado el control de la llamada, prepago puede dar instrucciones al MSC para controlar la duración de la llamada y enviar un informe de duración de la llamada a prepago después de un período predefinido de tiempo. El monitoreo de llamadas en el MSC comienza tan pronto cuando la llamada ha alcanzado el estado activo.

La llamada se divide en períodos de llamadas. Después de cada período de llamada, el MSC envía un informe a prepago, notificando a prepago sobre la duración transcurrida de la llamada. Prepago puede enviar una nueva solicitud de monitoreo para duración de la llamada a la MSC. Esta secuencia de pedido de monitoreo de duración de la llamada y la generación de informe de carga continúa hasta que una de las partes llamada termina la llamada o el crédito disponible ha alcanzado un nivel mínimo es decir se queda sin saldo para continuar la llamada. En tal momento, prepago terminará la llamada.

Las operaciones (CAP) que se utilizan para este mecanismo son:

- ApplyCharging (ACH): esta es la forma de instrucciones de prepago al MSC para iniciar o continuar el seguimiento de duración de la llamada.
- ApplyChargingReport (ACR): Este es el informe que se envía desde la MSC a prepago al final de un período de llamada o cuando se libera la llamada. Además, cuando se produce configurar fallo de la llamada, como interlocutor llamado ocupado o no contesta, la MSC también envía un informe de carga (si se solicita previamente).

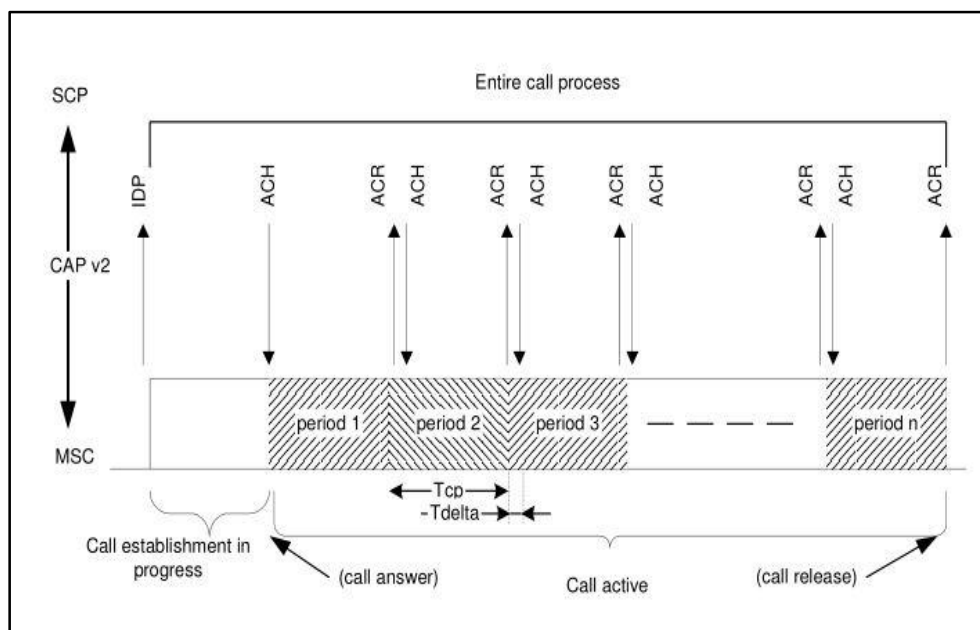


Figura 2.4. Proceso de Control de Llamada [14].

En la Figura 2.4 se puede observar que la MSC envía un mensaje IDP, esto es para que prepago establezca la llamada en caso de tener crédito el abonado. Cuando se ha verificado que el abonado suscriptor posee saldo y es abonado prepago, prepago envía un ACH que es una solicitud de informe de llamada apenas la llamada sea contestada. Luego que el periodo de la llamada se completa la MSC envía un informe de la llamada que es un ACR, seguido de este informe prepago vuelve a enviar un ACH y se espera otro periodo, así hasta que la parte llamante o llamada cuelgue o el abonado llamante se quede sin crédito suficiente para continuar con la llamada.

2.5 Sistema Operativo.

Un sistema operativo es un conjunto de programas que gestionan los recursos de hardware permitiendo la correcta utilización de los recursos disponibles. Las funciones básicas de los sistemas operativos son de administrar los recursos que posee el equipo, coordinar el hardware, organizar los archivos que posee incluso es responsable de la seguridad del sistema. El sistema operativo determina el orden y el tiempo de la ejecución de las aplicaciones antes de dar paso a otra aplicación, gestiona el intercambio de memoria interna entre varias aplicaciones, se ocupa de los dispositivos de hardware conectados, envía mensajes a cada aplicación o al usuario en caso de errores que se hayan producido. Los sistemas operativos más utilizados son Linux, Windows, DOS y Mac.

2.5.1 Sistemas Operativo Linux

Linux es un sistema operativo de libre distribución basado en Unix debido a su kernel o núcleo libre y el sistema GNU. Al ser un sistema operativo libre no es necesario pagar ningún tipo de licencia por el uso del mismo, el sistema brinda al programador su código fuente lo que permite realizar cambios según la necesidad, de igual manera los programas que se ejecutan en este sistema son de libre acceso.

2.5.2 Ubuntu.

Ubuntu es una distribución Linux enfocado a ordenadores de escritorio además de brindar soporte para servidores. Es de gran utilidad para la realización del proyecto por permitir montar un servidor con gran facilidad y sin tener que gastar para ello.

Se lo puede descargar desde la web oficial del sistema operativo, ofrece versiones para 32 y 64 bits. Los requisitos mínimos para instalarlo son los siguientes:

- Procesador x86 a 1Ghz.
- Memoria RAM de 1GB
- Disco Duro de 5 GB.
- Tarjeta gráfica y monitor para una resolución de 800x600.
- Lector de DVD o puerto USB.
- Tarjeta de red.

Las diferencias entre Ubuntu y Windows se detallan a continuación:

- La diferencia que más destaca es que Linux es gratis.
- Los sistemas de software son libres y seguros.
- Linux nos ofrece seguridad, no hay riesgo de virus.
- La velocidad de procesamiento es más rápido.

2.6 Lenguaje abierto PHP.

PHP (Pre Procesador de Hipertexto), es un lenguaje de código abierto de programación para desarrollar páginas web, al ser código abierto permite a los programadores utilizarlo libremente, necesita estar alojado en un servidor que pueda procesar PHP. Toda página web posee tres aspectos importantes el hosting que es un servicio que provee un espacio dentro de Internet para los sitios web, el dominio que es el nombre con el que accedemos a la página y el diseño del contenido.

Es considerado uno de los lenguajes más flexibles, potentes y de alto rendimiento en la actualidad. Por lo que múltiples sitios con gran demanda de tráfico han optado por PHP como tecnología de servidor. PHP se utiliza para generar páginas web dinámicas cuyo contenido puede cambiar en base a los cambios que se realicen en una base de datos o aportaciones de los usuarios. Cada vez que se inicia una página web, el servidor consulta en la base de datos las últimas actualizaciones ingresadas y se muestra en el navegador del visitante una página HTML como si fuera estática [15].

La gran demanda que posee PHP se debe a que su codificación es muy parecido a otros lenguajes de programación como lo es el

lenguaje C, también por su capacidad de ser ejecutado en la mayoría de los sistemas operativos, tales como Linux, Mac OS y Windows y puede interactuar con los servidores de web más populares sin costo alguno, permitiendo crear aplicaciones muy complejas.

CAPÍTULO 3

3 DESARROLLO DEL PROYECTO

3.1 Requerimientos necesarios para la implementación.

Las herramientas básicas que permitieron la realización con éxito del proyecto se detallarán a continuación, podemos clasificarlas en dos partes lo que respecta a Hardware y Software.

3.1.1 Hardware.

Laptop Dual Core Dell.

Permitirá ejecutar los programas necesarios para realizar la programación de la plataforma de estudio, además de mostrar la interfaz gráfica de la interacción con los demás protocolos que permiten el establecimiento de llamadas celulares. Entre las características importantes de la laptop que se utilizó se puede

mencionar las siguientes: posee un procesador Intel Core(TM) Dual Core 1.83 GHz y memoria RAM de 2,00 GB.

Router D-Link.

Este dispositivo electrónico proporciona conectividad a nivel de red, está encargada de encaminar paquetes de datos dentro de una red. El router ofrecerá una IP única para cada máquina permitiendo que los paquetes de datos sean enviados y procesados a un destino específico. La función del router es crear una conexión WLAN que simula a la red celular.

3.1.2 Software.

Ubuntu 14.04.1

Por las características destacadas que posee y las mismas que fueron mencionadas en el capítulo 2, se instaló en la laptop la versión Ubuntu 14.04.1 LTS de 32 bits. Ubuntu presenta diferentes versiones, la importancia de utilizar la última versión disponible radica en el hecho de corrección de errores y nuevos parches de seguridad, convirtiéndola en una versión más estable y de mejor rendimiento.

Los pasos para la correcta instalación del sistema operativo Ubuntu se detallarán a continuación:

Para instalar Ubuntu en la laptop fue necesario reservar espacio en el disco duro, con 15 GB fue suficiente, pero queda a disposición de cada

persona la cantidad a reservar teniendo en cuenta la cantidad de programas que se utilizarán. Para la realización del proyecto solo se instaló Netbeans y el servidor web Xampp para la programación en PHP.

Se debe descargar la versión de Ubuntu que se va a utilizar, en este caso la versión 14.04 LTS, versión de 32 o 64 bits. Una vez que se descargó la imagen iso de Ubuntu se debe grabarla en un DVD o en un pendrive. Insertamos el medio que se va a utilizar y reiniciamos el ordenador.

Cuando se reinicie el ordenador se observará en la pantalla el asistente de instalación que guiará la instalación, este asistente comprobará los requisitos mínimos para la instalación en caso de no cumplir los requerimientos, no se podrá concluir la instalación.



Figura 3.1 Asistente de instalación de Ubuntu [16].

La Figura 3.1 muestra el asistente de instalación solicitando al usuario la partición en donde desea instalar Ubuntu, tener en cuenta que no debe ser la misma partición en donde se encuentra Windows u otro sistema que se posea ya que podría existir daños.

Los últimos pasos son de configuración del sistema como es la región donde nos encontramos y la zona horario; además de la distribución del teclado.

El proceso de instalación termina cuando se observa la Figura 3.2 se procede a retirar el medio instalador y se reinicia el ordenador para empezar a utilizar el nuevo sistema operativo.

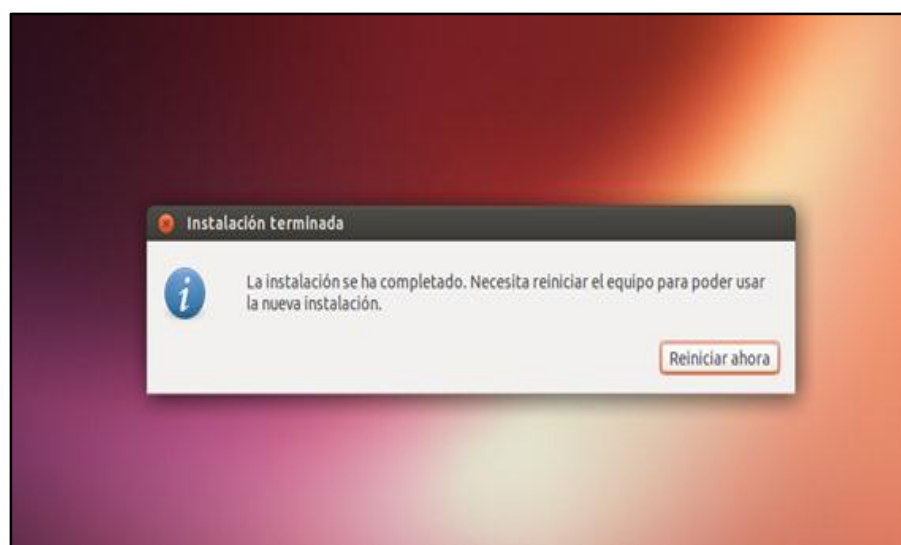


Figura 3.2 Asistente de Ubuntu finalizando la instalación [17].

SERVIDOR XAMPP

Es una plataforma de software libre que se basa en el servidor web Apache, permitiendo el desarrollo web en un servidor local. El servidor XAMPP datos MySQL. Se puede instalar en cualquier sistema operativo. Se procederá a indicar los pasos para montarlo dentro de Ubuntu.

Una vez descargado el instalador se da permiso de instalación por medio del terminal de Ubuntu a través del comando `sudo chmod 755 xampp-linux-installer.run`. Esto hará que en la pantalla del ordenador se inicialice el asistente de instalación, se procede a seguir las instrucciones y marcar las casillas necesarias para lograr la correcta instalación. Una vez finalizada la instalación se abre una página de bienvenida en el navegador de Ubuntu. Se escoge el idioma de nuestra preferencia y pulsar en la sección de status para la verificación que todo esté correcto.

En nuestro explorador se abrirá una pestaña como en la Figura 2.7 mostrando la ventana principal de XAMPP y por ende se culminará la instalación.



Figura 3.3 Ventana principal del servidor XAMPP [18].

Netbeans IDE 8.0.2

Netbeans es un entorno de desarrollo libre que permite la programación de lenguajes como Java, C++, Ruby, php entre otros. Básicamente está desarrollado en Java por lo que necesita la instalación de JDK y JVM para su funcionamiento, las actualizaciones más recientes ya contienen su instalador los requisitos mencionados, razón por la cual se utiliza la versión IDE 8.0.2. Se puede instalar en varios sistemas operativos como son: Linux, Mac Os, Windows.

La importancia de utilizar este IDE es que ofrece acceso a diferentes gestores de base de datos como son Oracle, MySql permitiendo la utilización y modificaciones de bases de datos desde el propio IDE.

Netbeans ofrece la utilización de diversos servidores que son gestionados desde el propio IDE, entre los servidores más conocidos se encuentran Apache, WebLogic, para el proyecto se utiliza Xampp que es una distribución del servidor Apache.

Se debe cumplir los requerimientos mínimos de hardware que son los siguientes:

- Ubuntu 9.10
- Procesador de 800 Mhz Intel Pentium III o equivalente.
- Memoria RAM de 512 MB
- Espacio libre de 650 Mb en el disco duro

Los procedimientos para su correcta instalación se mencionan a continuación:

Se procede a descargar el IDE desde la página oficial de Netbeans, una vez descargado se debe utilizar el terminal de Ubuntu y se escribe los comandos que se detallarán a continuación para abrir el instalador:

Estos comandos abrirán el asistente de instalación, se da click en los botones requeridos para continuar el proceso. Una vez ya finalizada la

instalación tendremos nuestro IDE listo para su utilización. La Figura 2.9 muestra la página principal del IDE de NetBeans.

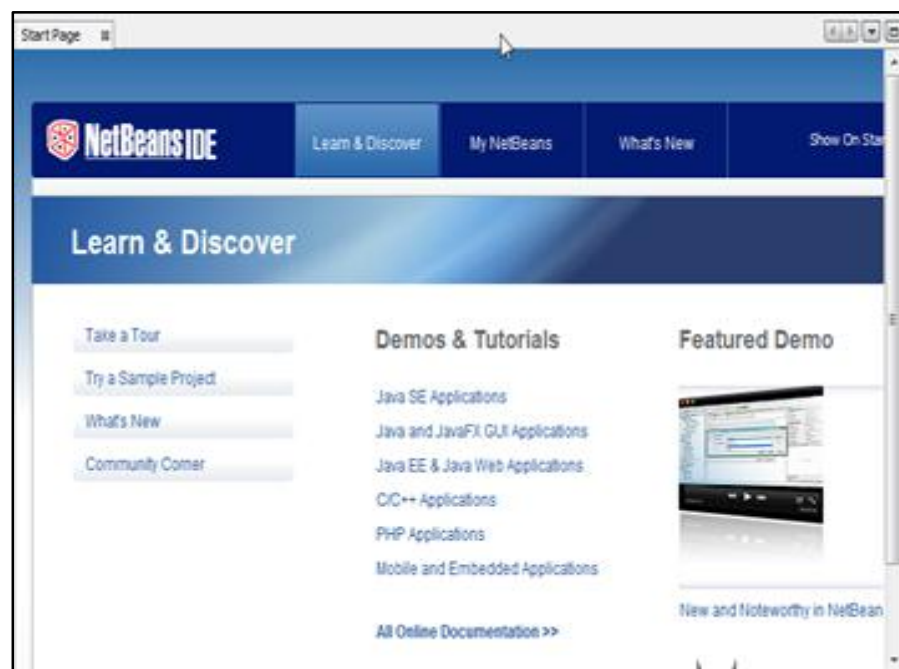


Figura 3.4 Ventana principal de Netbeans IDE [19].

Advance Rest Client

Permite ejecutar los scripts programados, se agrega los valores de los parámetros que recibe el script y retorna una respuesta según la programación de cada script. Es una aplicación de HTTP, realiza sencillas operaciones de solicitud/respuesta. Un cliente establece una conexión con un servidor y envía un mensaje con los datos de la solicitud. El servidor responde con un mensaje que contiene el estado de la operación y su posible resultado.

3.2 Estructura básica del funcionamiento de los mensajes prepago.

Para el desarrollo del módulo Prepago del Sistema Prepago que se desea simular, se analizarán dos escenarios, el primero para llamadas entre terminales analizando todos los casos como son: *On-net* (para este caso claro-claro) u *Off-net* (claro-cnt, claro-movistar) y el segundo escenario cuando el usuario marca *282 para consultar saldo o realizar una recarga. Para los dos escenarios los mensajes que utilizan prepago y la interacción con los demás componentes del sistema son diferentes.

3.2.1 Mensajes que utiliza Prepago para una llamada entre terminales.

Los mensajes que interactúan entre los diferentes componentes y prepago, para la realización de una llamada serán clasificados a continuación.

Mensajes que llegan a Prepago:

Initial Detection Point (IDP).- Este mensaje es enviado por la MSC para solicitar a Prepago instrucciones para completar la llamada. Posee los siguientes parámetros (*opcode = 0*, *calling party number*, *called party number* y *Service Key*) que son número llamante, número llamado y *Service Key* que le dice el servicio de red inteligente que se va a utilizar.

Apply Charging Report (ACR).- Este mensaje es enviado por la MSC hacia Prepago como respuesta al ACH que es enviado por Prepago. Este mensaje se envía una vez que termina el periodo de la llamada, enviado por prepago o cuando el usuario termina la llamada. El ACR enviado es la acumulación de los anteriores; ACR posee los siguientes parámetros (*opcode = 36, timeIfNoTariffSwitch*).

Event Report BCSM (EBCSM).- Este mensaje se utiliza para notificar a Prepago sobre cualquier evento inesperado en la llamada, posee los siguientes parámetros (*opcode = 24, eventTypeBCSM*). Para los eventos que se reporten se utilizará *Odisconnect, tdisconnect, tNoAnswer* que irán dentro de la variable *eventTypeBCSM*.

Mensajes que salen de Prepago.

Connect.- Este mensaje se utiliza para solicitar a la MSC proceda a conectar la llamada, lo identifica *opcode = 20*.

Apply Charging (ACH).- Es el encargado de interactuar con la MSC para cualquier llamada realizada a Prepago, indica el periodo de llamada disponible para el abonado dependiendo del total de minutos disponibles para hablar, el valor enviado siempre será menor a 90 si

el saldo disponible es menor a 90 segundos o igual a 90 si el saldo es mayor igual que 90 segundos. Posee los siguientes parámetros (opcode = 35, ACH (X)).

Request Report BCSM Event (RRB). - Este mensaje se utiliza para solicitar a la MSC notifique cualquier evento durante la llamada a Prepago. Se identifica por el opcode = 23.

Release Call.- Este mensaje es utilizado por Prepago para terminar el proceso de la llamada en cualquier estado de la misma se identifica con opcode = 22.

Dentro de la llamada entre dos terminales se analizarán todos los casos posibles que ocurren en la vida real.

Caso 1: Terminal A realiza la llamada y se queda sin saldo.

En la Figura 3.5 se muestra un diagrama de interacción de objetos, cuando el número A llama a número B y continúa la llamada hasta quedarse sin saldo. En este caso de uso; se va asumir que el número A posee 215 segundos de saldo para hablar; se puede observar que Prepago (PP) recibe un mensaje IDP de la MSC, lo procesa para ver si el número A es prepago, si posee saldo y si el número B se

encuentra en la base de datos de Prepago para poder mandar a conectar la llamada, dado que las condiciones mencionadas se cumplen Prepago responde a la MSC un mensaje *Connect* para empezar el envío de mensajes entre Prepago y MSC, un ACH (90) que le indica que se conecta la llamada por al menos 90 segundos y un RRB que le pide a MSC que le notifique de algún evento que pase del lado de los terminales o de la misma MSC. Transcurridos los 90 segundos con normalidad, Prepago recibe un ACR (90) de parte de MSC notificando que ha transcurrido ese tiempo y preguntándole a Prepago si puede hablar 90 segundos más, Prepago consulta a su base de datos y verifica que puede hablar 90 segundos más y responde a MSC con otro ACH (90). Se vuelve a repetir lo anterior por parte de MSC pero esta vez le envía un ACR (180) a Prepago, ese 180 segundos indica los segundos hablados acumulados y le vuelve a preguntar cuánto tiempo más puede seguir hablando, Prepago consulta en su base de datos y verifica que puede hablar 35 segundos más y le envía esta vez un ACH (35). Pasado los 35 segundos MSC vuelve a enviar un ACR de 215 es decir la acumulación de lo que ha hablado, Prepago vuelve a consultar en su base de datos si puede seguir hablando, como ya no posee saldo este le notifica a MSC que de por terminada la llamada, esto lo hace con un *Release Call* y

procede a actualizar la base de datos con el nuevo saldo que sería cero dólares con cero centavos.

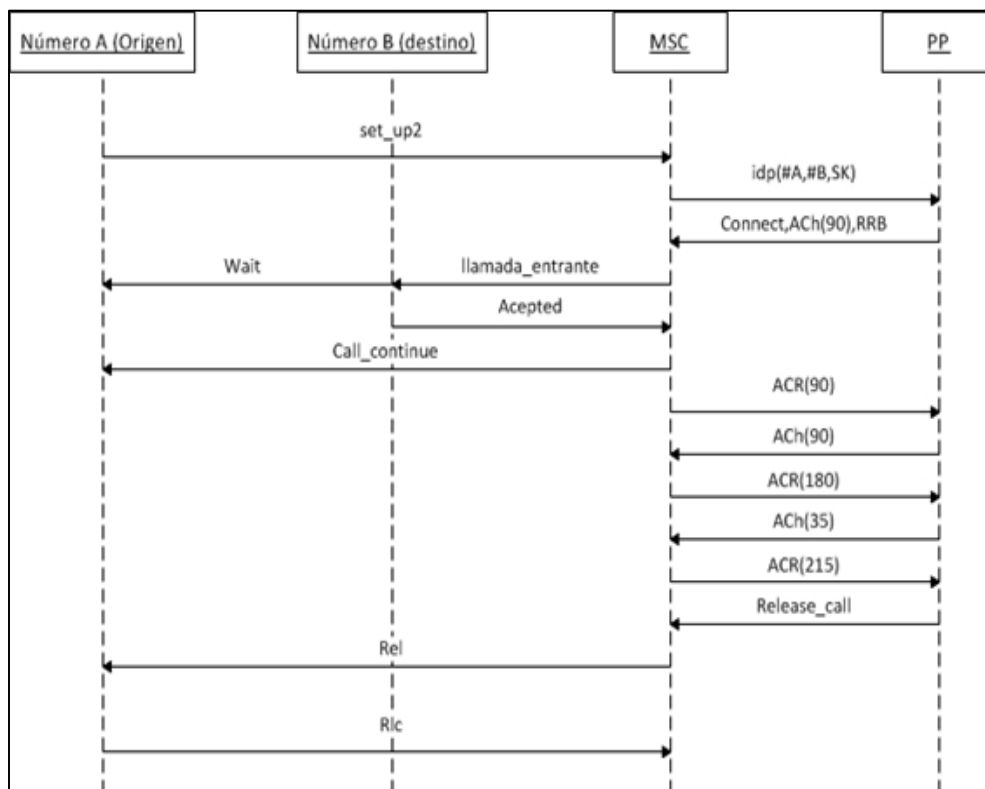


Figura 3.5 Diagrama de Interacción de Objetos. Flujo en una llamada celular, el número llamante queda sin saldo.

Caso 2: Terminal A termina la llamada en cualquier instante de ella

En la Figura 3.6 se muestra el siguiente diagrama de interacción de objetos, el cual, indica para este caso de uso; que el terminal A es el que cuelga la llamada para esto se asumirá que el terminal posee más de 110 segundos de saldo. Como se menciona en el caso anterior

Prepago recibe un mensaje IDP el cual consulta la base de datos para verificar si el número A es prepago, posee saldo y que el número B esté en la base de datos del sistema, como todo se cumple Prepago responde con un mensaje *connect*, con un mensaje ACH (90) y un RRB, después de pasar los 90 segundos Prepago recibe de MSC un mensaje ACR (90) con el tiempo transcurrido y le vuelve a preguntar a la base de datos si posee saldo para seguir hablando. Como el número A posee saldo para hablar 90 segundos más Prepago envía un nuevo ACH (90), y espera por el siguiente ACR; pero en este caso el terminal A cierra la llamada entonces inmediatamente Prepago recibe un ACH de 110 segundos proveniente de MSC; esto significa que de los 90 segundos más que podía seguir hablando solo hablo 20 segundos más; seguido de un mensaje EBCSM con el siguiente parámetro (Odisconnect), el cual nos indica, que la llamada finalizó y que parte de la llamada lo hizo, en este caso la parte llamante. A todo esto Prepago responde a la MSC con un *Release Call* después de haber actualizado la base de datos con el saldo restante.

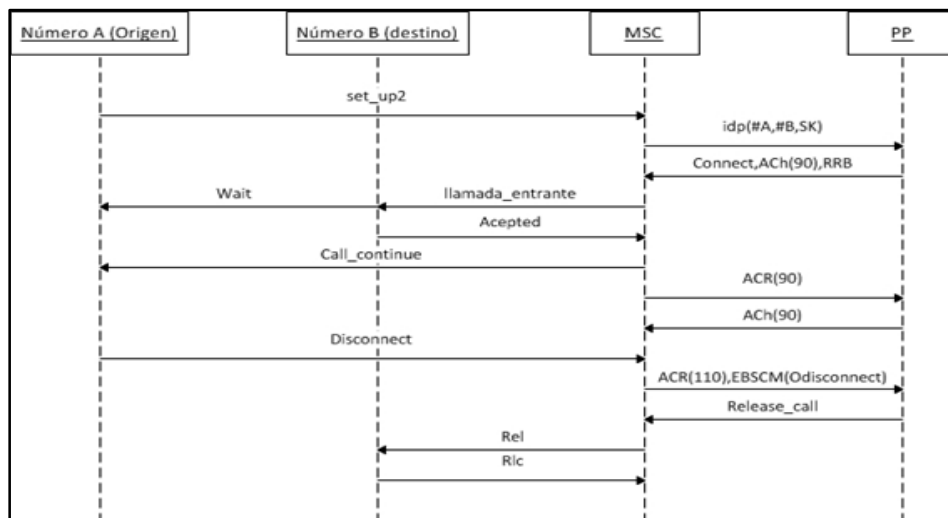


Figura 3.6 Diagrama de Interacción de Objetos. Flujo en una llamada celular, número llamante corta en cualquier instante.

Caso 3: Terminal B termina la llamada en cualquier instante de la misma.

En la Figura 3.7 se muestra el caso de uso, en el que la llamada es finalizada por el Terminal B. En este escenario se asume que el terminal A posee saldo de 75 segundos para realizar la llamada. De la MSC se recibe un mensaje IDP, se procede a verificar lo antes mencionado, Prepago responde con un mensaje *connect*, con un mensaje *ACH(75)* porque en este caso el terminal A posee saldo para hablar solo 75 segundos; si hubiese tenido más de 90 segundos el ACH sería de 90 segundos; y con un RBB. En éste caso el terminal B decide cerrar la llamada a los 50 segundos y la MSC responde a Prepago con un ACR de 50 segundos, seguido de un mensaje EBSCM

como respuesta al RRB, con un Tdisconnect que nos indica que el Terminal B es quien desconectó la llamada. Acto seguido Prepago actualiza la base de datos con el saldo restante y envía un *Release Call* para terminar el proceso.

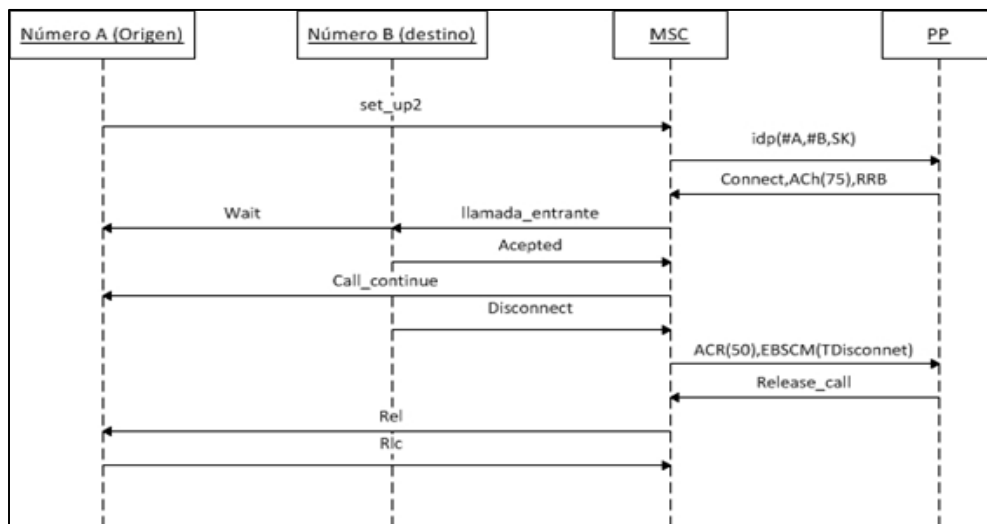


Figura 3.7 Diagrama de Interacción de Objetos. Flujo en una llamada celular, número destino corta en cualquier instante.

Caso 4: Terminal A finaliza la llamada antes de que pueda contestar terminal B.

En el siguiente caso de uso que se muestra en la Figura 3.8, el terminal A cierra la llamada antes de que el terminal B pueda contestar. Se procede de la misma manera en la que se le envía un mensaje IDP a Prepago y éste le responde con un mensaje *Connect*, con un mensaje ACH y un mensaje RRB. Como esta vez terminal A cuelga la llamada sin que se haya consumido el tiempo del ACH, se envía un mensaje EBSCM (Odisconnect) indicando que fue el terminal A es quien

desconectó la llamada y Prepago responde a la MSC con un *Release Call* para terminar el proceso.

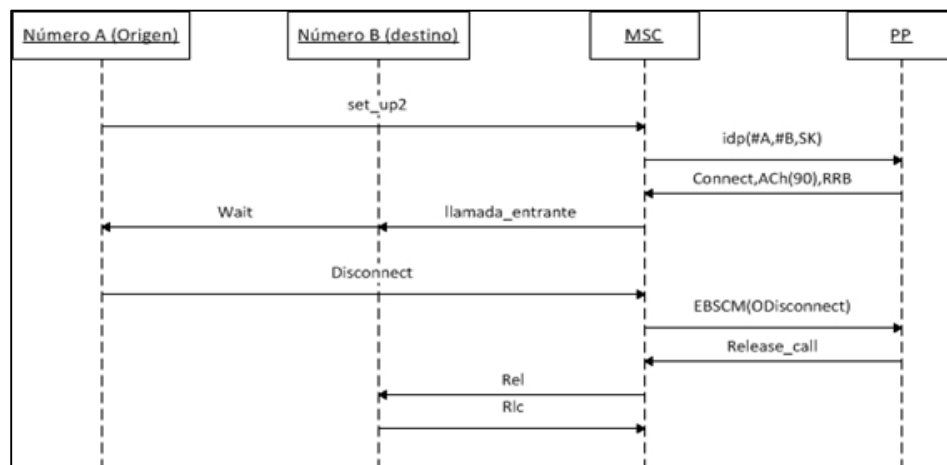


Figura 3.8 Diagrama de Interacción de Objetos. Flujo en una llamada celular número llamante corta antes de conteste el número destino.

Caso 5: Terminal A no posee saldo para realizar la llamada.

La Figura 3.9 detalla el caso de uso cuando el terminal A no posee saldo para realizar una llamada. Cuando Prepago recibe de la MSC el mensaje IDP y este verifica que el número llamante no posee saldo para realizar la llamada, lo que procede a realizar Prepago es mandar un mensaje *Release Call* para terminar el proceso.

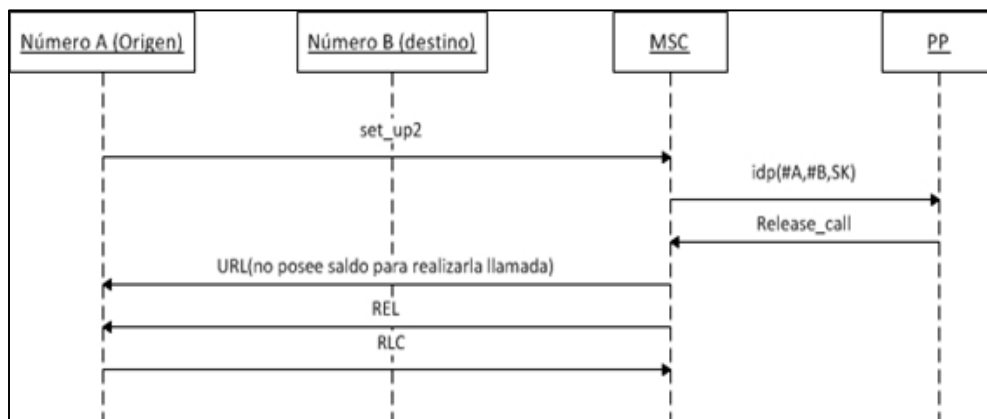


Figura 3.9 Diagrama de Interacción de Objetos. Flujo en una llamada celular número llamante no dispone de saldo.

Caso 6: Terminal B no contesta llamada o desvía la misma.

En la Figura 3.10 se muestra el caso de uso en el que el terminal B no contesta la llamada o decide desviar la misma. De igual manera que en los casos anteriores la MSC envía un mensaje IDP para ver si Prepago puede mandar a conectar la llamada, como respuesta a esto Prepago envía un mensaje *Connect*, ACH y un mensaje RRB; luego Prepago espera por el ACR, pero en este caso no lo recibe, en vez de eso recibe un mensaje EBCSM (tNoAnswer) que nos indica que el terminal B no contestó la llamada o la desvió. Inmediatamente Prepago responde con un mensaje *Release Call* que nos indica que el proceso se ha finalizado.

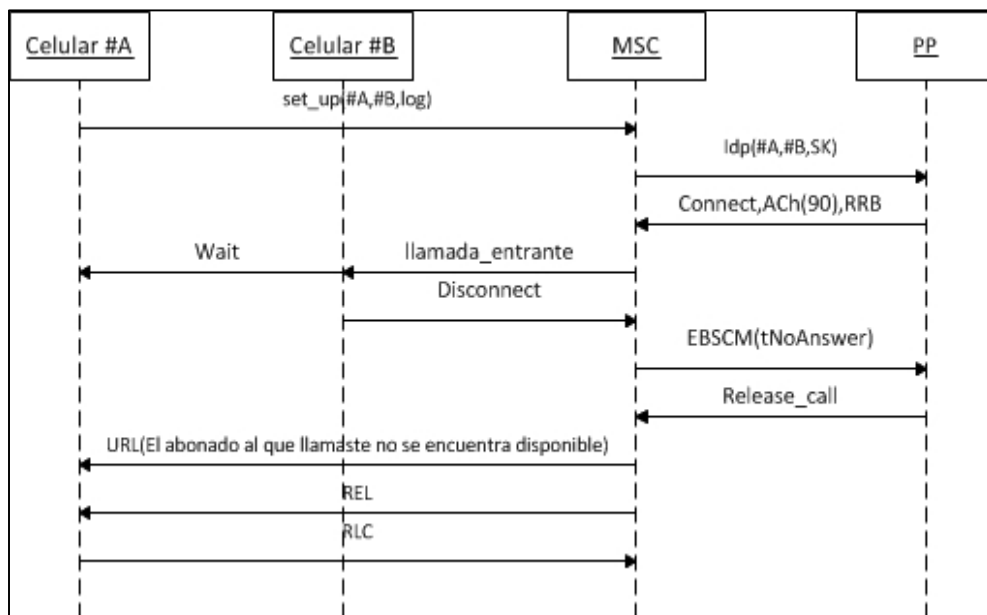


Figura 3.10 Diagrama de Interacción de Objetos. Flujo en una llamada celular número destino no contesta llamada o desvía la llamada.

3.2.2 Mensajes que utiliza Prepago para una llamada al *282.

Dentro de la llamada al *282 se ofrecen dos opciones al usuario, la primera para consultar saldo, la segunda para que ingrese una tarjeta y recargue el saldo disponible. Los mensajes que interactúan entre Prepago y el resto de componentes es decir MSC e IVR en una llamada al *282 son los siguientes:

Mensajes que llegan a Prepago

Initial Detection Point (IDP).- Este mensaje es enviado por la MSC para solicitar a Prepago instrucciones para completar la llamada. Posee los siguientes parámetros (*opcode = 0, Calling Party Number, Called Party Number y Service Key*) que son número llamante, número llamado y *Service Key* que le dice el servicio de red inteligente que se va a utilizar.

Assist Request Instructions (ARI).- Este mensaje se utiliza para ayudar en un procedimiento de asistencia, es decir le pregunta a prepago que *prompt* debe tocar para dicho procedimiento. Se identifica con el *opcode = 16* y en sus datos lleva el *Calling Party Number* es decir el número llamante.

Specialized Resource Report (SRR).- Esta operación se utiliza como respuesta a una operación de "*PlayAnnouncement*" cuando el anuncio es completado y llega a la IVR. Es identificado por el *opcode = 49*.

Mensajes que salen de Prepago

Establish Temporary Connection (ETC).- Este mensaje sirve para mandar a establecer una conexión temporal y decirle a la MSC que

levante un ISUP, es decir cuando se llama al *282, se identifica con el *opcode* = 17.

Play Announcement (PA).- Este mensaje se utiliza para la interacción con un usuario dentro de la banda GSM, se lo identifica con el *opcode* = 47

Se procederá a analizar los diferentes escenarios que existen cuando se realiza una llamada celular al *282.

Caso 1: Llamada al *282 consultando el saldo hasta que se termina de escuchar el audio.

La Figura 3.11 representa el caso de uso cuando el usuario marca el *282, para este escenario se procederá a realizar la consulta de saldo. Prepago recibe por parte de MSC el mensaje IDP el cual posee al *282; prepago analiza el número en su base de datos y sabe que es el número de prepago amigo, es decir, no debe cobrar por realizar dicha llamada; en ese momento, Prepago envía un mensaje ETC el cual le indica a la MSC que proceda a levantar ISUP. Después de que se levanta la conexión ISUP, IVR envía a Prepago un mensaje ARI el cual indica que debe responder con un mensaje PA, por ser el primer mensaje ARI que se recibe por parte de IVR, Prepago sabe que debe

responder con la dirección del *prompt* que dice (Bienvenido a claro, para información de tu cuenta digita 1, para ingresar una tarjeta amigo en tu celular digita 2), esto lo envía dentro del PA en forma de coordenadas, para que, IVR mande a tocar dichos *prompts*, como confirmación de que a IVR le llegó el mensaje PA por parte de Prepago éste envía un mensaje SRR. Luego de que el usuario digita la opción 1; por parte de IVR, Prepago recibe un mensaje AR12, que indica que el usuario está preguntando por su saldo. Prepago procede a realizar la consulta en su base de datos y dentro del mensaje PA, Prepago envía a IVR las coordenadas de los prompts que poseen los audios del saldo (el saldo de su cuenta es, X, dólares, y, Z, centavos), como confirmación a ese PA, Prepago recibe un Mensaje SRR de confirmación de que se ha recibido la información.

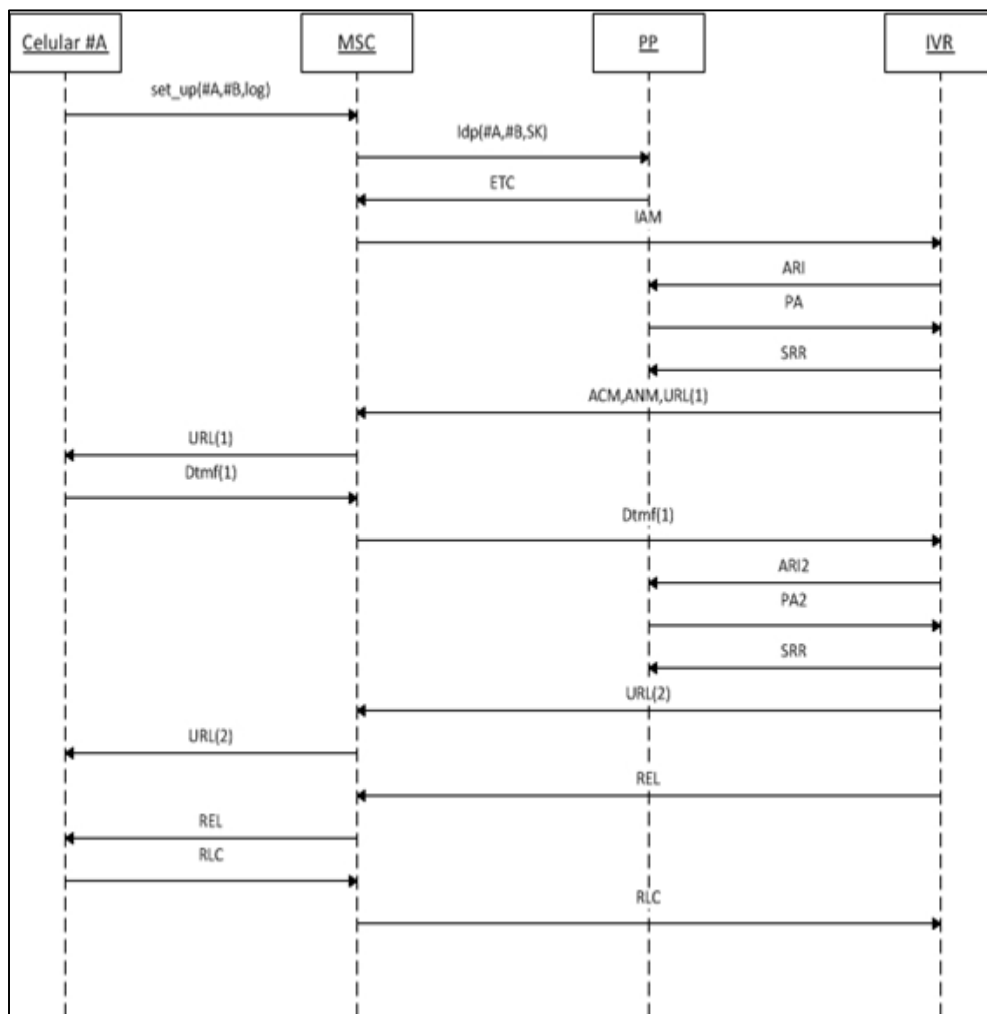


Figura 3.11 Llamada al *282 para consultar saldo disponible.

Caso 2: Llamada al *282 consultando el saldo y usuario cuelga la llamada.

En la Figura 3.12, el procedimiento que se ejecuta en relación al caso anterior, Prepago se comporta de la misma manera.

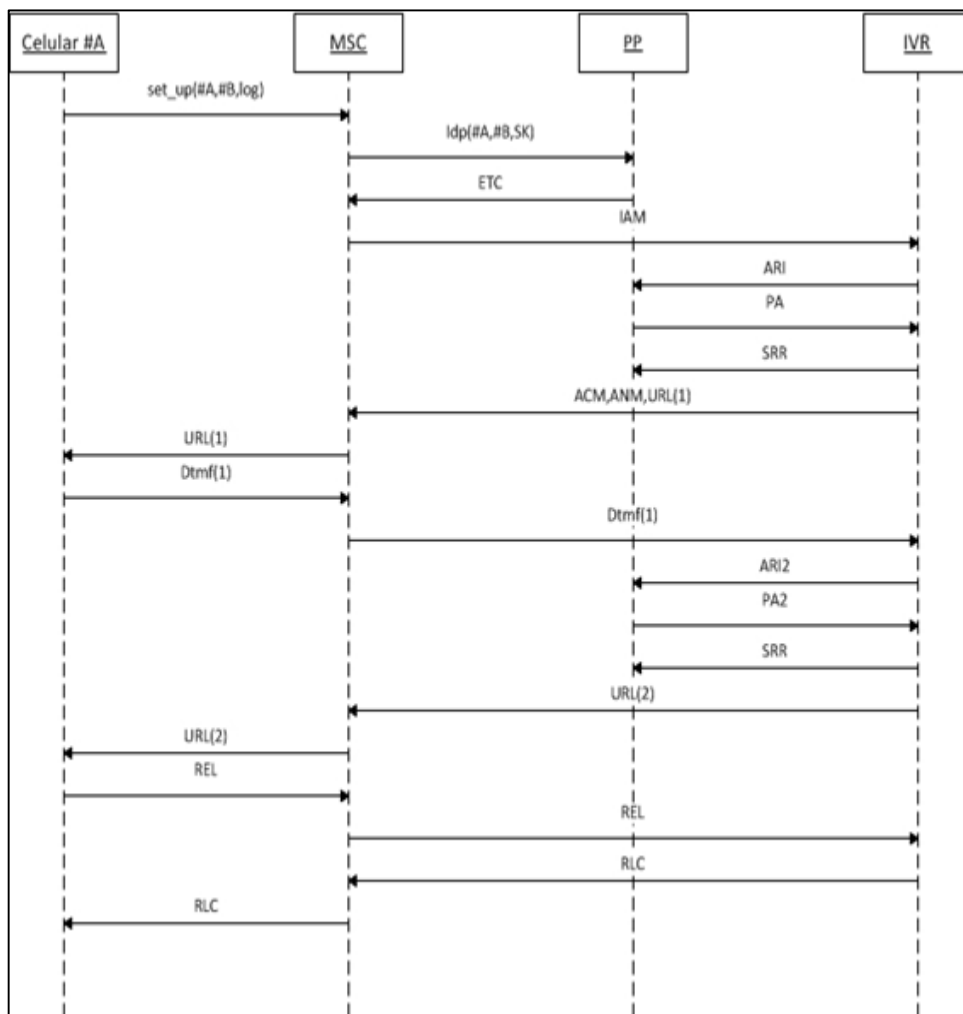


Figura 3.12 Llamada al *282 a consultar saldo y el usuario corta la llamada.

Caso 3: Llamada al *282 para ingresar tarjeta, hasta que se termina de escuchar audio esté correcto o incorrecto el número de la tarjeta.

En la Figura 3.13, se muestra el caso de uso en el que, el usuario digita la opción 2 para realizar una recarga. Para este proceso se repite lo

mismo hasta después de recibir el primer mensaje SSR por parte de IVR a Prepago. Continuando con la secuencia Prepago recibe un *ARI3* por parte de IVR, con esto Prepago sabe que tiene que enviar dentro del mensaje PA3 las coordenadas del *prompt* que indica, que el usuario debe ingresar la el código secreto de la tarjeta seguido del signo #, como respuesta a lo enviado Prepago recibe por parte de IVR un mensaje SRR de confirmación. Seguido de esto hay dos opciones que el usuario digite bien el código de la tarjeta, que el usuario ingrese un código ya usado o un código incorrecto; esta información llega a Prepago por medio de un *ARI4*, con esto Prepago procede a verificar dicho código secreto en la base de datos dependiendo de ese proceso, que puede ser (el código de la tarjeta es incorrecto, el código de la tarjeta ya ha sido activado y si esta correcto el código entonces toca el saldo de su cuenta ha sido acreditado con X dólares) Prepago envía un mensaje PA4 a IVR con las coordenadas de los *prompts* que debe tocar la IVR. Como respuesta al mensaje *PA4*, Prepago recibe un último *SRR* de confirmación y termina el proceso con Prepago.

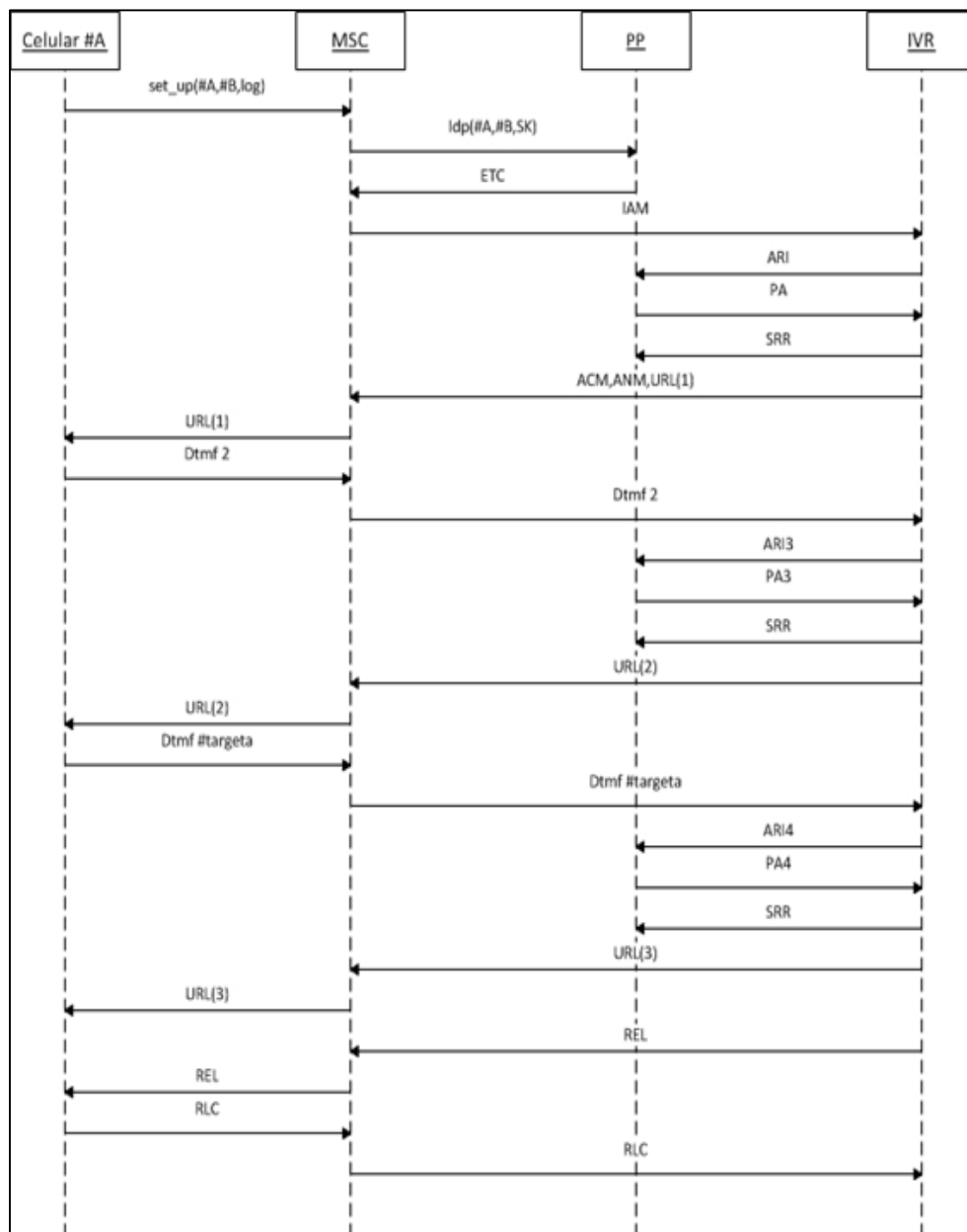


Figura 3.13 Llamada al *282 para ingresar una tarjeta de recarga.

3.3 Conectividad entre componentes.

Como la implementación del módulo Prepago se la realizó en *PHP*, se utilizó el protocolo de conexión HTTP, debido a que se ajusta al envío de mensajes con la figura de cliente - servidor y es el más idóneo para realizar la comunicación entre componentes que conforman el proyecto de simulación de una red prepago.

Al momento de ejecutar el proyecto se debe asegurar que los equipos se conecten es decir, primero deben estar en la misma red, que para nuestro uso es una red *WLAN*, la cual está conformada por un router y algunos componentes que interactúan entre sí; para esto se les asignó direcciones *IP* fijas a los componentes que intervendrán, para que los paquetes que se enviarán lleguen a su destino requerido.

Como se realizarán pruebas y no se estará conectado a internet, usamos xampp un servidor web apache, que simula como si se tuviese salida a internet y así poder probar la conexión entre los equipos. A continuación se muestra la Figura 3.14 donde se levanta el servidor apache para que los componentes se puedan comunicar.

```

julio@julio-HP-Pavilion-dv6000-GA384UA-ABA: ~
julio@julio-HP-Pavilion-dv6000-GA384UA-ABA:~$ sudo /opt/lampp/lampp start
[sudo] password for julio:
Starting XAMPP for Linux 5.6.3-0...
XAMPP: Starting Apache...ok.
XAMPP: Starting MySQL...ok.
XAMPP: Starting ProFTPD...ok.

```

Figura 3.14 Comando para alzar servidor Apache.

Al momento de realizar las pruebas de envío de mensajes se utiliza la siguiente herramienta que nos permite enviar un *request* hacia el *script* invocado que deseamos procesar, luego de esto el *script invocado* envía su respuesta hacia el *script* que lo invocó, a continuación la Figura 3.15 muestra lo antes mencionado.

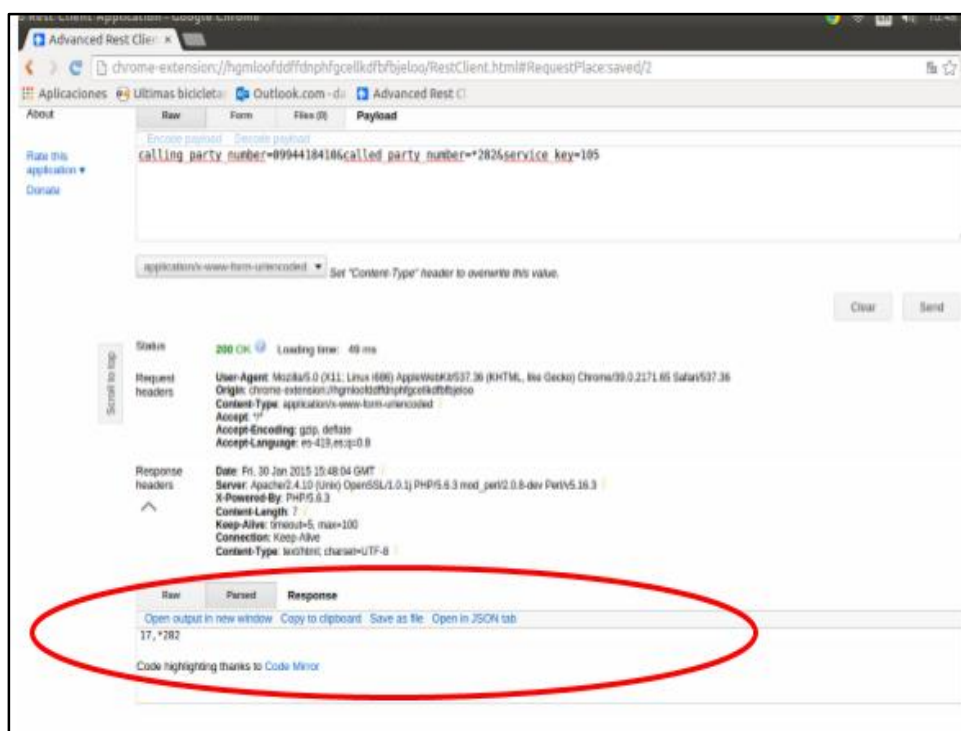


Figura 3.15 Envío de request correcto del script implementado.

Si el *script* invocado no recibe los parámetros correctos no se realizará el envío de la respuesta correctamente porque faltan datos para realizar el procedimiento, en la Figura 3.16 se muestra dicho evento. El *script* notifica que faltó algún parámetro por enviar. Es así como se realizaron las pruebas de conexión y de ejecución del módulo Prepago.

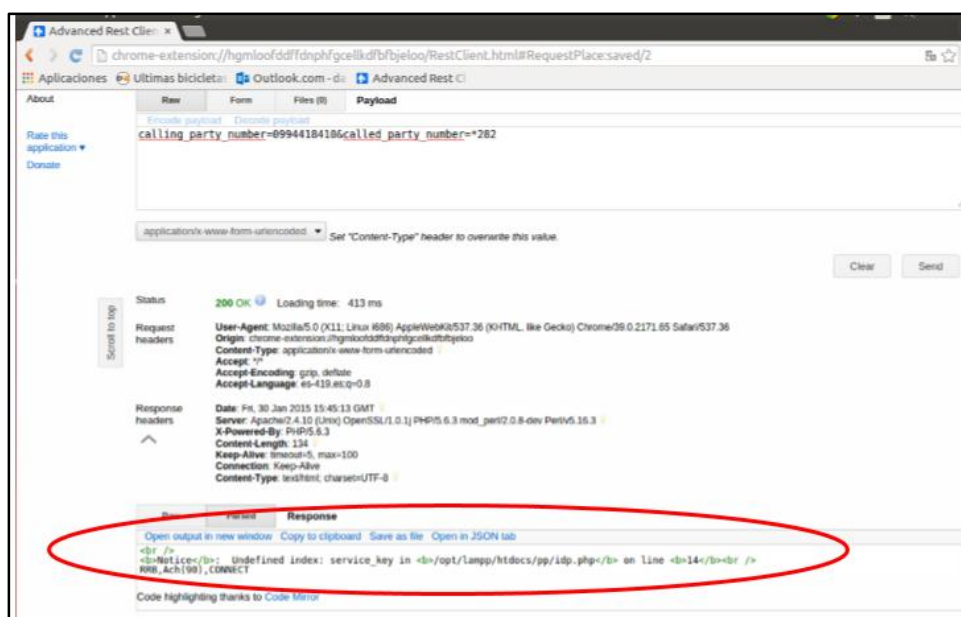


Figura 3.16 Envío de request incorrecto del script implementado.

3.4 Pasos a tener en cuenta al realizar los scripts.

En resumen se deben seguir los siguientes pasos para empezar a realizar los *scripts* que representarán a los mensajes que se envían entre los módulos Prepago, MSC e IVR:

1. Tener instalado Ubuntu como sistema operativo.
2. Tener instalado el IDE NETBEANS para poder implementar los *scripts* PHP.
3. Para realizar las pruebas de conexión se debe tener instalado servidor apache para realizar los *request* y *response* hacia los *scripts* PHP invocados.
4. No olvidar de hacer ping entre las máquinas que se van a utilizar para asegurar que los *request* y *response* de los *scripts* invocados lleguen al destino deseado.
5. Entender la estandarización de los protocolos que interactúan para proceder a realizar la correcta implementación de los *scripts* necesarios para lograr la comunicación entre los módulos que conformarán la simulación de la red prepago celular.

3.5 Resumen para la implementación del Módulo Prepago.

Para el desarrollo del Módulo Prepago se analizó detenidamente su estandarización la cual es muy compleja por tal motivo se utilizó un software llamado Wireshark que consiste en un analizador de protocolos utilizado para el análisis y solución de problemas en redes de comunicaciones. En este software se observa con facilidad los argumentos necesarios de cada mensaje enviado y recibido para

establecer llamadas celulares, solo los argumentos más relevantes de cada mensaje se procederán a programar mediante los scripts.

Una vez ya establecido el ambiente de trabajo y las herramientas elementales para el desarrollo del módulo tanto como software, hardware y los scripts programados que respondan correctamente según las peticiones requeridas, es necesario estar conectado a la red local a través de enlaces inalámbricos o enlaces por cables, cada nodo de la red celular tendrá un IP asignada la cual no cambiará lo que permitirá asegurarnos que los mensajes lleguen al destino correcto. Es necesario realizar pruebas con todos los equipos que interactúan además del módulo Prepago y seguir paso a paso la interfaz gráfica, para detectar rápidamente donde se originan inconvenientes y proceder a su corrección.

CAPÍTULO 4

4 Análisis de Resultados.

4.1 Resultados obtenidos.

Al finalizar la implementación, se obtiene como resultado el módulo Prepago que ofrece las siguientes consultas: el saldo disponible de usuario, la tarifa con la que se debe realizar la facturación después de haber realizado la llamada, verificar si la tarjeta de recarga de saldo existe o ya ha sido activada, los *prompts* que debe mandar a tocar dependiendo la consulta que se realiza a Prepago.

En el módulo Prepago desarrollado, además de las consultas antes mencionadas; se realizan operaciones de débito de saldo, operaciones para calcular el número de segundos que puede hablar el usuario en fracciones de 90 segundos o menos, actualización del saldo cuando el usuario ingresa una tarjeta para recargar su saldo y mandar a finalizar la llamada cuando el abonado se queda sin saldo.

4.2 Interpretación de resultados esperados.

Los resultados obtenidos se ajustan a la realidad, en los diferentes casos antes mencionados en los apartados 3.2.1, 3.2.2 y 4.1. En el módulo Prepago es donde se realizan las mayorías de transacciones y verificaciones de saldos, minutos por hablar, tarjetas de recargas y *prompts* para mandar a reproducir según sea el caso. Como el protocolo de señalización usado es muy abstracto para personas que no entienden del tema, para su interpretación se realizó una interfaz gráfica, donde nos muestra de una forma más amigable los procedimientos que ocurren mientras se ejecuta la llamada y los diferentes *scripts*.

4.3 Interfaz gráfica del módulo Prepago

A continuación se muestra la interfaz gráfica del módulo Prepago implementado, la misma que ayuda a visualizar de una manera más

amigable el flujo de mensajes de señalización que intervienen en el proceso durante la llamada ya sea entre terminales o al *282.



Figura 4.1 Panel de Control del módulo Prepago.

Como se muestra en la Figura 4.1 se puede observar la página principal con su menú muy básico fácil de interpretar por el usuario. Si se le da click en el logo se podrán visualizar los mensaje de los protocolos de señalización usados durante la llamada realizada, tal como se muestra en la Figura 4.2.

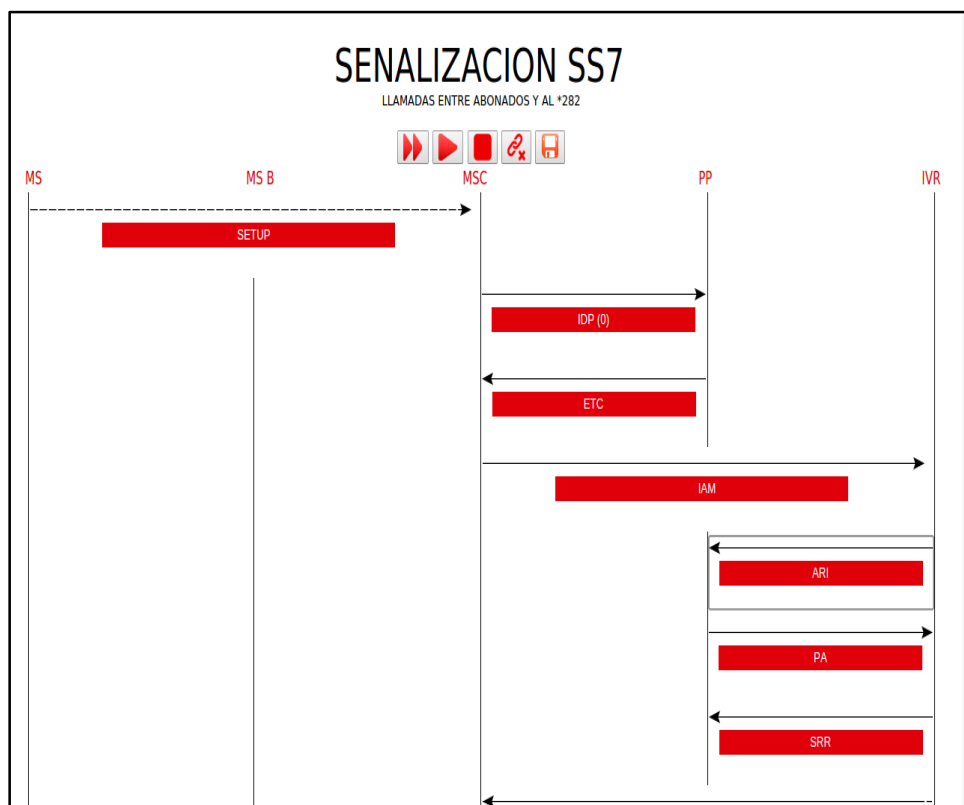


Figura 4.2 Ventana para visualizar los mensajes de señalización.

En esta ventana se pueden observar algunos botones los cuales sirven para dar inicio a la simulación, pausar, avanzar. También se puede observar el botón de guardar, esto permite almacenar la información capturada en un archivo de texto, el cual tendrá toda la información de señalización que ocurrió durante la llamada.

Volviendo a la Figura 4.1 se puede observar tres botones, el primero de ellos permite abrir una ventana; esto se muestra en la Figura 4.3,

donde se puede ver las tarjetas de recarga que existen con su respectivo valor y si están usadas.



Figura 4.3 Ventana para visualizar las tarjetas prepago con su respectivo valor y estado.

Si se selecciona una tarjeta y se da click en buscar aparece una ventana donde se puede modificar el estado de la tarjeta y su valor como se muestra en la Figura 4.4.



Figura 4.4 Ventana para visualizar las tarjetas prepago con su respectivo valor y estado.

El segundo botón que se muestra en la Figura 4.1 permite abrir una ventana donde muestra los abonados que pertenecen al servicio de prepago dándole click en la flecha de expansión como se ve en la Figura 4.5

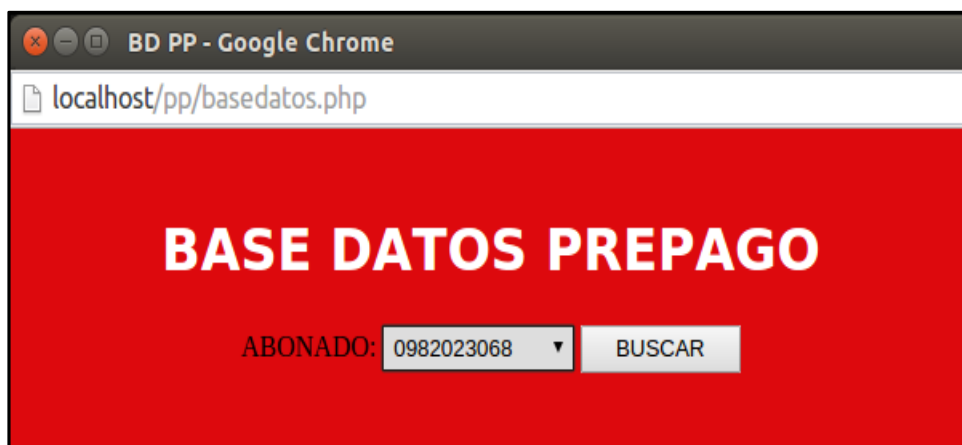


Figura 4.5 Ventana para visualizar los números que pertenecen al servicio prepago.

Al dar click en buscar se abrirá una ventana donde se puede ver el saldo del abonado y modificarlo, este saldo esta en centavos. A continuación se visualiza en la Figura 4.6.



Figura 4.6 Ventana para visualizar el número con su respectivo saldo con la opción de modificar el saldo si se desea.

Para finalizar volviendo a la Figura 4.1 al presionar el último botón se abre una ventana donde se puede conocer la tarifa que se cobra al abonado dependiendo hacia que operadora se realiza la llamada, esto se observa en la Figura 4.7.



Figura 4.7 Ventana para visualizar la tarifa de facturación por minuto de la llamada.

Si se presiona el botón buscar se abrirá una ventana donde se puede modificar la tarifa a cobrar en una llamada si se desea, como se muestra en la Figura 4.8.



Figura 4.8 Ventana para modificar la tarifa de facturación de la llamada.

4.4 Análisis de la topología de conexión empleada.

Para facilitar el desarrollo del proyecto de toda la red celular y en particular el módulo prepago, se utilizaron tres *routers*, dos de ellos representan centrales telefónicas una para Guayaquil y la otra para Quito, y el restante router representa a SS7, la Figura 4.1 indica la

conexión de todos los nodos de la red celular; representados por diferentes laptops que están conectadas por cables de datos y por enlace inalámbrico cada una posee un IP en particular, dentro de cada laptop se encuentra el código fuente ajustándose a la estandarización de cada protocolo empleado.

El módulo del celular a emplearse en la simulación tendrá la opción de conectarse a la central de Guayaquil o Quito y realizar una llamada a un número que pertenezca a su misma central o a una diferente y proceder a realizar los servicios que brinda el simulador que ya han sido mencionados con anterioridad. Por facilidad se desarrolló un clon al módulo IVR lugar donde se encuentran almacenados los *prompts* de audio que el usuario escucha al realizar una llamada o realizar consulta, debido a que la IVR original estaba conectada por cable al router SS7 y por enlace inalámbrico a la central Quito y no se puede acceder a otra central como es la de Guayaquil, la IVR clon está enlazada tanto al router principal que es SS7 como a su vez a la central de Quito y de esta manera se logró simular una llamada ajustándose a la realidad, el abonado escuchará los audios de consulta y llamada sin importar a la central que se encuentra conectado.

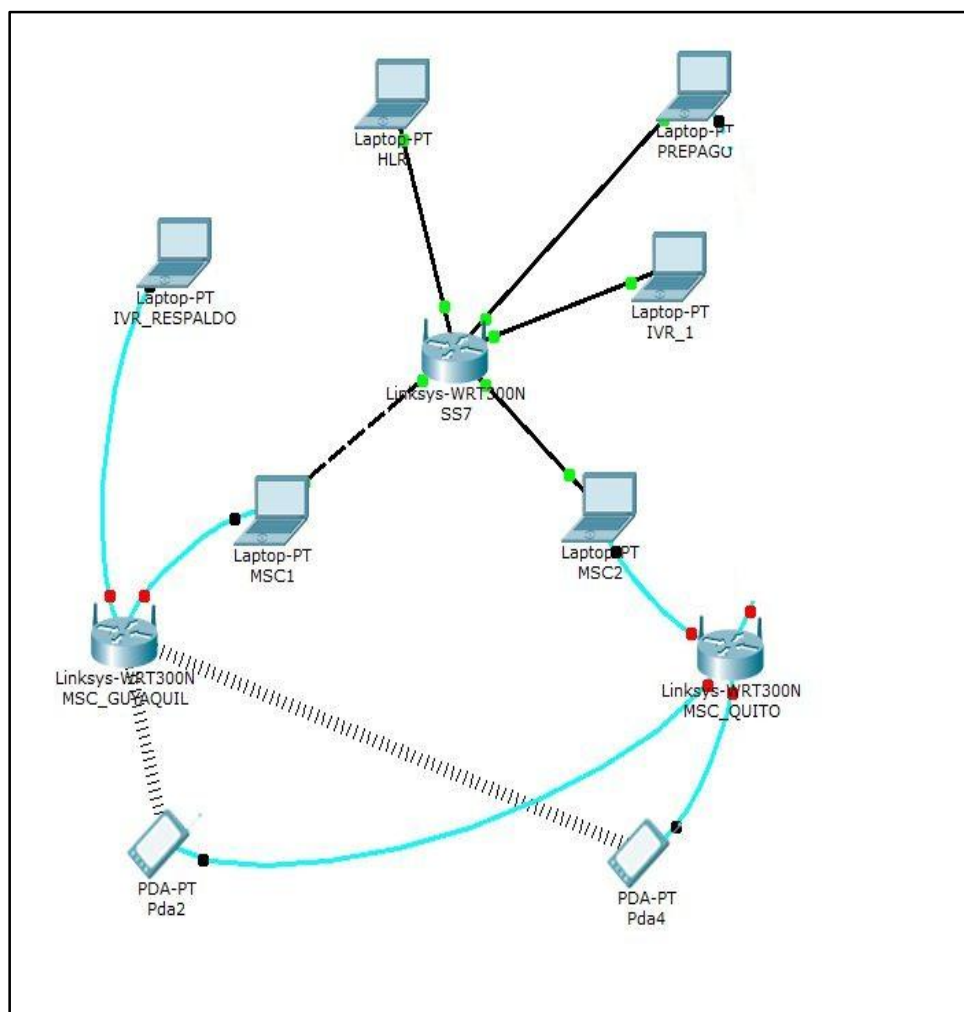


Figura 4.9 Representación de la conexión empleada para la simulación de la red celular.

4.5 Discusión.

El desarrollo del módulo prepago resultó muy complejo debido a la interacción con varios protocolos incluso con celulares, cada protocolo posee una estandarización muy amplia por lo que se tuvo que analizar

detalladamente los mensajes primordiales para cumplir los objetivos planteados y lograr establecer llamadas celulares.

Una vez ya lograda la comunicación y el respectivo flujo de mensajes a través de los nodos de la red celular se optó por implementar una interfaz gráfica entendible para la persona que lo llegara a utilizar, anteriormente el comportamiento de cada nodo se observaba en un terminal de Ubuntu con fondo negro lo que era muy primitivo y confuso; para solucionar este inconveniente se desarrolló una aplicación Java que funcionará con el protocolo HTML logrando una interfaz gráfica amigable.

4.6 Futuro uso del módulo prepago desarrollado.

El uso que tendrá el módulo prepago será netamente didáctico para la explicación en lo que concierne a facturación dentro de una red de telefonía celular, las personas interesadas en este tema dispondrán de un simulador que facilitará la interpretación de manera gráfica de los protocolos que interactúan dentro de la red celular logrando eliminar lo abstracto y complejo que posee dichos protocolos. Al disponer el código fuente del módulo prepago las personas interesadas que lo lleguen a emplear tendrán la opción de ajustar la tarificación de las

llamadas lo que cambiará la facturación del saldo disponible de los abonados.

CONCLUSIONES

Del presente informe se puede inferir la importancia del protocolo de comunicación Camel sobre el módulo Prepago que se implementó, el cual ayuda a entender de mejor manera el funcionamiento del mismo en el mundo de la telefonía celular.

1. La implementación de Camel asegura la idea básica de una red inteligente, que consiste en separar los servicios de conmutación y los de servicio, asegura que los usuarios de la red puedan utilizar todos los beneficios disponibles aun no estando en su red local.
2. El control de una llamada celular dentro de una red inteligente conlleva a la MSC una mayor carga dentro de los procesos que ejecuta, por lo que Camel incorpora operaciones que permitan monitorear las llamadas y procedimientos para actualización de registros de localización.

3. Dentro de la red celular Camel brinda a las operadoras celulares una gran variedad de servicios tales como facturación, control de llamada, consulta de saldo, recargas entre otros.
4. El uso del módulo Prepago permite comprender de una manera más sencilla el funcionamiento de Camel durante la simulación de la llamada, mostrando los mensajes de señalización en una interfaz gráfica la cual es muy entendible para personas que no conocen el procedimiento que ocurre en llamadas entre celulares o mientras el usuario realiza consulta de saldo o recarga del mismo.
5. Al realizar la implementación del módulo prepago en el lenguaje de programación PHP, se tuvo que ajustar la lógica del protocolo Camel a la lógica de comunicación HTTP, la cual maneja el mecanismo de *request-response*, para que exista la interacción correcta entre los diferentes módulos que conforman el sistema de simulación de prepago.
6. En la actualidad existen una gran variedad de softwares que permiten la captura en tiempo real de una llamada celular, muchos de estos no son tan amigables para presentar los datos de señalización obtenidos; utilizando los valores capturados se podría realizar la simulación de la misma llamada obteniendo como resultado un mecanismo más entendible para presentar la señalización estudiada.

RECOMENDACIONES

Las recomendaciones que se mencionan a continuación, ayudan al correcto manejo del módulo implementado, para la realización de pruebas y simulación de uso del sistema prepago.

1. Para el módulo prepago se ajuste a la realidad, es necesario tener presente la estandarización que lo rige, y tratar de simular todos los casos posibles que puedan ocurrir, incluso ayudándonos con softwares y capturas en tiempo en tiempo real de llamadas realizadas en la vida real para estos procesos.
2. Antes de ejecutar la simulación de llamada prepago o llamada de consulta al *282 se debe tener en cuenta que los módulos que conforman el sistema estén conectados a su red respectiva y tengan levantado el apache, el cual

es nuestro servidor web, luego de esto se procede a realizar PING entre componentes para saber que existe conectividad para enviar o recibir información de los módulos con quienes interactúan.

3. Recordar que el módulo Prepago posee una base de datos donde se puede hacer la consulta de si el abonado que está solicitando el servicio es prepago; solo si el número llamante se encuentra en dicha base de datos se procederá con el servicio antes mencionado, ya que el sistema simula una red de servicios prepago.
4. Al momento de iniciar la simulación recordar que por el alcance del proyecto que es netamente didáctico se variaron parámetros no tan relevantes como la cantidad de dígitos utilizados en las tarjetas de recargas, el número de abonados de la operadora, los prompts utilizados entre otros.
5. Para comprobar que el módulo prepago funciona correctamente es importante revisar la base de datos con los saldos de los abonados y verificar que se cumple el débito según los minutos hablados, además que se realice con éxito el aumento del saldo al momento de realizar recargas al número.
6. Recordar que cada máquina tiene una IP fija asignada por lo que es necesario realizar pruebas de conectividad para ello se debe realizar ping entre las máquinas que participan en la simulación.

BIBLIOGRAFÍA

- [1] ProEVA, Introducción a redes inteligentes GSM. Protocolo CAMEL, «www.eva.fing.edu.uy,» Diciembre 2014. [En línea]. Available: https://eva.fing.edu.uy/pluginfile.php/63822/mod_folder/content/0/Introduccion_a_CAMEL.pdf?forcedownload=1.
- [2] Travis Russel, Signaling System #7, McGraw-Hill Telecommunications, Enero 2015.
- [3] Uyles Black, ISDN and SS7 Architectures for Digital Signaling Networks, Prentice Hall Series in Advanced Communications Technologies, Enero 2015.
- [4] Miguel López Lavado y Antonio Barba, Evolución de la plataforma de servicios CAMEL más allá de los sistemas 3G, «www.ewh.ieee.org,» Diciembre 2014. [En línea]. Available: http://www.ewh.ieee.org/reg/9/etrans/ieee/issues/vol04/vol4issue5Sept.2006/4TLA5_02LopezLavado.pdf

- [5] Martyn Shuttleworth, Diferentes métodos de investigación, «Explorable.com,» Enero 2015. [En línea]. Available: <https://explorable.com/es/diferentes-metodos-de-investigacion>.
- [6] Z. Zvonar, Peter Jung, Karl Kammerlander, GSM: Evolution Towards 3rd Generation System, Springer Science & Business Media, Enero 2015.
- [7] Informit, «www.informit.com,» 2014. [En línea]. Available: http://www.informit.com/library/content.aspx?b=Signaling_System_No_7&seqNum=23.
- [8] ETSI, 3GPP TS 29.078 v3.15.0, www.qtc.jp/3GPP/Specs/29078-3f0.pdf, consultado Diciembre 2014.
- [9] Juan Báez y Pérez de Tudela, Investigación Cualitativa, ESIC editorial, fecha de consulta Noviembre 2014.
- [10] Lee Dryburgh y Jeff Hewett, Signaling System No. 7, Cisco Press, fecha de consulta Diciembre 2014.
- [11] Maille Altuve, SISTEMA DE SEÑALIZACIÓN POR CANAL COMÚN N.º 7, <http://ss7uftxdatos.blogspot.com/2013/02/sistema-de-senalizacion-por-canal-comun.html>, Diciembre 2014.
- [12] Russell, Travis: “Signaling System 7,” 2nd edition, McGraw-Hill, 2014.

- [13] Antonio Barba Martí, Gestión de red, Ediciones UPC, fecha de consulta Diciembre 2014.
- [14] Davidson Jonathan, "Voice over IP Fundamentals", Cisco Press, fecha de consulta Diciembre 2014.
- [15] Rogier Noldus, Camel: intelligent Networks for the GSM, GPRS and UMTS Network, WILEY, consultado Noviembre 2014.
- [16] Rogier Noldus, Camel: intelligent Networks for the GSM, GPRS and UMTS Network, WILEY, consultado Noviembre 2014.
- [17] Heurtel Olivier, PHP 5.3: Desarrollar un sitio Web dinámico e interactivo, El Nuevo PHP paso a paso, Ediciones ENI, fecha de consulta enero 2015.
- [18] Rogier Noldus, Camel: intelligent Networks for the GSM, GPRS and UMTS Network, WILEY, consultado Noviembre 2014.
- [19] Heurtel Olivier, PHP 5.3: Desarrollar un sitio Web dinámico e interactivo, El Nuevo PHP paso a paso, Ediciones ENI, fecha de consulta enero 2015.