



# ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL

Facultad de Ingeniería en Electricidad y Computación

“ANÁLISIS E IMPLEMENTACIÓN DE UN SISTEMA DE RESPUESTA INTERACTIVA (IVR), VIRTUAL EN EL LENGUAJE ABIERTO PREPROCESADOR DE HIPERTEXTO (PHP) SOBRE LINUX PARA EL VOCEO DE AUDIOS QUE INDIQUE EL SALDO DE LOS ABONADOS PREPAGOS ALOJADOS EN UNA RED INTELIGENTE A TRAVÉS DEL PROTOCOLO DE SEÑALIZACIÓN PARTE DE USUARIO DE RED DIGITAL DE SERVICIOS INTEGRADOS (ISUP).”

## **INFORME DE MATERIA DE GRADUACIÓN**

PREVIO A LA OBTENCIÓN DEL TÍTULO DE:

## **INGENIERO EN ELECTRÓNICA Y TELECOMUNICACIONES**

Presentado por:

**RAÚL ANTONIO CAMAS OVIEDO**

Guayaquil - Ecuador

2015

## **AGRADECIMIENTO**

Encontrarme en esta etapa, la cual ha sido muy difícil a la cual llegar, no pudo ser posible de no ser por mi familia, en especial mis padres, quienes jamás me retiraron su apoyo en el transcurso de la realización de este logro. Al apoyo y sabia orientación del Ing. José Miguel Menéndez. Y a Dios, por sus bendiciones constantes, que hacen todo posible.

Raúl Camas Oviedo

## **DEDICATORIA**

Este trabajo está dedicado a mis padres, esos seres humanos maravillosos que son el pilar de la formación que he tenido, a la ESPOL, institución que ha hecho posible este gran logro, a mis compañeros que han sido de apoyo grande en este trabajo, en especial a mis compañeros de seminario.

Raúl Camas Oviedo

## **TRIBUNAL DE SUSTENTACIÓN**

---

Ing. José Miguel Menéndez, MSc.  
Profesor de la Materia de Graduación

---

Ing. Ronald Ponguillo I., Mg.  
Profesor Delegado por la Unidad Académica

## **DECLARACIÓN EXPRESA**

"La responsabilidad del contenido de este Informe de Materia de Graduación, 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)

---

Raúl Camas Oviedo

## RESUMEN

El presente trabajo se encargará de implementar parte de un sistema de señalización de Canal Común, llamado el Sistema de Señalización Número 7 (SS7, por sus siglas en inglés), que hacen posible la facturación de las llamadas así como el intercambio de mensajes de texto, más específicamente, la implementación del Sistema Interactivo de Voz (IVR) para consultas de saldos y promociones por parte de abonados de una red. Dicha interfaz se desarrollará bajo el protocolo de la Red Digital de Servicios Integrados (ISDN) Parte de Usuario (ISUP) implementada en lenguaje PHP bajo Linux, con el objetivo de brindar una interfaz de consulta comprensible y cómoda para los abonados, y puedan realizar sus solicitudes de una manera eficiente.

Se busca validar cada uno de los tipos de mensaje que ocurren en una transacción de ISUP, junto con un mensaje automatizado por cada mensaje recibido. Se creará un software capaz de simular satisfactoriamente dicha transacción y el sistema de consulta para abonados.

## ÍNDICE GENERAL

AGRADECIMIENTO .....	ii
DEDICATORIA .....	iii
TRIBUNAL DE SUSTENTACIÓN .....	iv
DECLARACIÓN EXPRESA .....	v
RESUMEN .....	vi
ÍNDICE GENERAL.....	vii
ÍNDICE DE FIGURAS.....	x
GLOSARIO .....	xii
INTRODUCCIÓN .....	xv
<b>CAPÍTULO 1: DESCRIPCIÓN GENERAL DEL PROYECTO</b>	
1.1 ANTECEDENTES.....	1
1.2 JUSTIFICACIÓN DEL PROYECTO.....	3
1.3 DESCRIPCIÓN DEL PROYECTO.....	4
1.4 OBJETIVOS DEL PROYECTO.....	6
1.4.1 Objetivo General .....	6
1.4.2 Objetivos Específicos .....	6
1.5 ALCANCE DEL PROYECTO.....	6
1.6 LIMITACIONES DEL PROYECTO.....	7
1.7 METODOLOGÍA .....	8
<b>CAPÍTULO 2: FUNDAMENTACIÓN TEÓRICA</b>	
2.1 INTRODUCCIÓN A SS7.....	10

2.1.1	Reseña .....	11
2.1.2	Identificación de puntos de señalización .....	12
2.1.3	Tipos de Nodos .....	13
2.1.4	Modos de Señalización .....	14
2.1.5	Estructura .....	15
2.2	DEFINICIÓN DE ISUP .....	18
2.2.1	Transacción Básica .....	19
2.2.2	Transacción con terminal automático .....	23
2.2.3	Estructura de los mensajes .....	24
2.2.4	Mensajes básicos de la llamada.....	25
2.2.5	Otros mensajes .....	29
2.3	ACERCA DEL SISTEMA IVR .....	29
2.3.1	Definición.....	30
2.3.2	Tonos DTMF .....	31
2.4	ACERCA DEL SISTEMA OPERATIVO LINUX.....	33
2.4.1	Distribución de Linux .....	34
2.4.2	La distribución Ubuntu.....	35
<b>CAPÍTULO 3: DESARROLLO DEL PROYECTO</b>		
3.1	ANÁLISIS.....	37
3.2	PREPARATIVOS PARA LA IMPLEMENTACIÓN.....	38
3.2.1	Software .....	38
3.2.2	Hardware.....	40



3.3 IMPLEMENTACIÓN.....	42
3.3.1 Instalación de Ubuntu.....	43
3.3.2 Instalación de XAMPP.....	44
3.3.3 Instalación del IDE NetBeans.....	46
3.4 DISEÑO GENERAL.....	50
3.4.1 Archivos de utilidades.....	50
3.4.2 Programación de mensajes de señalización.....	53
3.4.3 Ingreso de teclado.....	55
3.5 SIMULACIÓN.....	56
3.6 POSIBLES APLICACIONES.....	58
CAPÍTULO 4: ANÁLISIS DE RESULTADOS	
4.1 DESCRIPCIÓN GENERAL.....	60
4.2 INTERPRETACIÓN DE RESULTADOS.....	61
4.2.1 Registro de transacción.....	61
4.2.2 Interfaz de señalización.....	64
4.2.3 Interfaz de señales.....	65
4.3 DISCUSIÓN FINAL.....	67
CONCLUSIONES.....	69
RECOMENDACIONES.....	71
BIBLIOGRAFÍA.....	73
ANEXOS.....	76

## ÍNDICE DE FIGURAS

Figura 2.1 Zonas de Códigos de Punto .....	13
Figura 2.2 Tipos de nodos de SS7.....	13
Figura 2.3 Modos de señalización .....	15
Figura 2.4 Estructura de SS7.....	17
Figura 2.5 Llamada ordinaria completada .....	20
Figura 2.6 Liberación de llamada normal.....	22
Figura 2.7 Transacción entre SIP e ISUP con mensaje CON.....	23
Figura 2.8 Estructura de un mensaje de ISUP.....	25
Figura 2.9 Estructura básica de un mensaje IAM .....	26
Figura 2.10 Estructura de un mensaje ACM.....	28
Figura 2.11 Estructura de un mensaje REL .....	29
Figura 2.12 Funcionamiento de un sistema IVR .....	30
Figura 2.13 Marcado por tono a un sistema IVR.....	31
Figura 2.14 Tabla de frecuencias DTMF.....	32
Figura 2.15 Núcleo Linux versión 3.0 en arranque .....	34
Figura 2.16 Escritorio de Ubuntu .....	36
Figura 3.1 Sitio principal del servidor XAMPP.....	39
Figura 3.2 Interfaz básica de NetBeans.....	40
Figura 3.3 Máquina donde se implementa la IVR .....	41
Figura 3.4 Ejemplo de una topología con tres de las máquinas de la red.....	41

Figura 3.5 Instalador de Ubuntu.....	44
Figura 3.6 Terminal de Ubuntu .....	45
Figura 3.7 Instalación de XAMPP .....	46
Figura 3.8 Instalación de NetBeans .....	47
Figura 3.9 Interfaz de Assembla .....	48
Figura 3.10 Actualización en desarrollo .....	50
Figura 3.11 Funciones de conversión .....	51
Figura 3.12 Ejemplo de función de envío de mensaje .....	51
Figura 3.13 Obtención de IP a través de una función .....	52
Figura 3.14 Código de la función utilitaria Parámetros .....	52
Figura 3.15 Funciones de registro de mensajes .....	53
Figura 3.16 Código del mensaje IAM (Envío a PP por INAP) .....	54
Figura 3.17 Código del mensaje IAM (Envío de ACM a MSC) .....	54
Figura 3.18 Envío de REL para colgado de llamada.....	55
Figura 3.19 Código de los DTMF .....	55
Figura 3.20 Simulador de señalización SS7 .....	57
Figura 4.1 Prueba de envío de mensaje a la parte de INAP .....	62
Figura 4.2 Prueba de reserva de circuito .....	63
Figura 4.3 Señalización de ISUP en la interfaz de la aplicación web .....	65
Figura 4.4 Interfaz del mensaje IAM con su respectiva trama .....	66
Figura 4.5 Interfaces de los mensajes ACM y ANM .....	66
Figura 4.6 Interfaces de los mensajes REL y RLC .....	66

Figura A.1 Flujo para el caso de consulta de saldos (Opción 1).....	76
Figura A.2 Flujo para el caso de recargas (Opción 2).....	77

## GLOSARIO

ACM	Address Complete Message
API	Application Programming Interface
ANM	Answer Message
BCI	Backward Call Indicator
CAMEL	Customized Applications for Mobile Networks Enhanced Logic
CAS	Channel Associated Signaling
CIC	Circuit Identification Code
CON	Connect Message
CCIS	Common Channel Interoffice Signaling
CCS	Common Channel Signaling
CPG	Calling Progress
DPC	Destination Point Code
DTMF	Dual-Tone Multi-Frequency
GNU	GNU's Not Unix!
HTTP	Hypertext Transfer Protocol
FCI	Forward Call Indicator
IAM	Initial Address Message
IDE	Integrated Development Environment
INAP	Intelligent Network Application Part
IP	Internet Protocol

ISDN	Integrated Services Digital Network
ISUP	ISDN User Part
ITU	International Telecommunication Union
IVR	Interactive Voice Response
MSC	Mobile Switching Centre
MTP	Message Transfer Part
MTP1	Message Transfer Part Level 1
MTP2	Message Transfer Part Level 2
MTP3	Message Transfer Part Level 3
MySQL	My Structured Query Language
NOC	Nature Of Connection Indicators
OSI	Open Systems Interconnection
OPC	Origin Point Code
PC	Point Code
PHP	Hypertext Preprocessor
REL	Release
REST	Representational State Part
RLC	Release Complete
SAM	Subsequent Address Message
SCCP	Signaling Connection Control Part
SCP	Service Control Point
SS6	Signaling System No. 6

SS7	Signaling System No. 7
SSP	Service Switching Point
STP	Signal Transfer Point
TMR	Transmission Medium Requirement
TUP	Telephone User Part
URL	Uniform Resource Locator
XAMPP	X – Apache – MySQL – PHP – Perl
XML	Extensible Markup Language

## INTRODUCCIÓN

Como finalidad principal, este proyecto pretende explicar y demostrar el funcionamiento exitoso del Sistema Interactivo de Respuesta (IVR por sus siglas en inglés) a través del protocolo de Red de Servicios Integrados de Parte de Usuario (o ISUP, por sus siglas en inglés). Para explicar dicho funcionamiento es necesario presentar cada uno de dichos mensajes que pueden ocurrir en esa transacción, y sus diversas interacciones con otros protocolos e impacto en el sistema generalizado. Más precisamente, se estudiará la transacción entre este sistema y una central móvil. Se mostrará el proceso de consulta a la base de datos implementada por el sistema prepago, de donde se obtendrá la información necesaria para dar una respuesta al abonado, del mismo modo con otros tipos de consulta solicitados por el mismo.

Los protocolos correspondientes al Sistema de Señalización por Canal Común Número 7 (SS7), entre ellos ISUP, son sujeto de un extenso estudio por su complejidad, pero al mismo tiempo efectividad y confiabilidad, por lo que son partes fundamentales de las redes utilizadas en las operadoras celulares de hoy en día, y dicho estudio es requerido para lograr plantear el desarrollo de un simulador de red en equipos compatibles.



# **CAPÍTULO 1**

## **1. DESCRIPCIÓN GENERAL DEL PROYECTO**

### **1.1 Antecedentes**

Los sistemas de señalización del nivel de complejidad de SS7 suelen ser bastante complejos de comprender para las personas que apenas se encuentren iniciando algún estudio en cuanto a sistemas de telefonía móvil o comunicaciones de tipo inalámbrico. Un estudiante, persona individual o empresas en busca de capacitar a su personal en este tipo de sistemas han buscado soluciones módicas y cómodas para brindar una explicación clara y comprensible del manejo de los equipos de la red, su funcionamiento, y lograr apreciar el intercambio de mensajes entre los mismos.

Desde los inicios del uso del sistema, se ha buscado simular el comportamiento de las redes de señalización y troncales que le permitan a los desarrolladores examinar la capacidad de la red, su supervivencia, y el análisis de las posibles introducciones de nuevos servicios. Los desarrolladores durante los años ochenta e inicios de los noventa tenían como objetivo principal predecir de manera precisa el rendimiento de la red y comportamiento para un ambiente en constante cambio, y para las grandes redes como las utilizadas en los sistemas de telefonía pública [1]

Para una mejor flexibilidad de desarrollo, estos se trabajan en código abierto. Algunas personas se refieren como tal al *software libre*, ya que el término evoca más a la idea de libertad que “abierto”. Software libre es aquel que respeta la libertad de los usuarios y la comunidad. En otras palabras, significa que los usuarios tienen la libertad para ejecutar, copiar, distribuir, estudiar, modificar y mejorar el software. Se trata de una cuestión de libertad sin embargo no del precio del mismo [2]

Debido a la naturaleza de sistemas operativos bajo Linux de ser de código abierto, se empezaron a desarrollar tales aplicaciones con mejores herramientas diseñadas para ser trabajadas bajo sus distribuciones. Un ejemplo de esto es el proyecto de SS7 Abierto (*OpenSS7*) iniciado en 1996 para ofrecer un sistema SS7 compatible con los sistemas operativos libres como UNIX. Los resultados mejoraron ya que era posible redistribuir el código fuente y no contaban con la molesta limitación de licencias para software,

dando lugar a su personalización y más hábil desarrollo de herramientas de apoyo [3]

## **1.2 Justificación del proyecto**

Con la creciente afluencia de los sistemas de telecomunicaciones, en especial la telefonía celular, cada día aumentan más el número de abonados suscritos a estos tipos de servicio, por lo que surge la necesidad de crear un sistema automatizado de consultas de servicios, saldos y promociones, parte de la séptima capa del modelo OSI, interactuando directamente con el usuario a través del teclado del equipo por medio de Multi-Frecuencias de Tono-Dual (DTMF, por sus siglas en inglés).

Es por eso que se considera necesario el desarrollo de aplicaciones de sistemas IVR sobre el protocolo ISUP, permitiendo expandir el alcance del desarrollo de este tipo de aplicaciones dentro de sistemas de señalización más extensos. El lenguaje PHP facilita el desarrollo de aplicaciones en la Web, debido a su flexibilidad y potencia para desarrollo. Mayor parte de los servidores web de licencia libre tienen capacidad para interpretar lenguajes de script, entre ellos PHP.

### 1.3 Descripción del Proyecto

En el Ecuador, las operadoras de telefonía móvil, Claro, Movistar y la Corporación Nacional de Telecomunicaciones todas constan de un sistema automatizado de respuesta de voz interactiva. Este trabaja sobre protocolos dentro del Sistema de Señalización Número 7 (SS7) comunicándose con una estación móvil para poder acceder a la base de datos del servicio a consultar, que en la presente, será un sistema de Prepago. El IVR y la central se comunicarán mediante una transacción ocurrente en el protocolo de ISUP, de donde se establecerán las solicitudes iniciales: El abonado que desea realizar la consulta, tras acceder a la IVR recibe el mensaje de que debe escoger una de las opciones disponibles marcando un número asignado del teclado de su equipo, este se detecta en el sistema analizando la frecuencia recibida de la matriz DTMF. Dependiendo de la solicitud realizada, el sistema consultará la base de datos de Prepago a través de mensajes del protocolo de Lógica Mejorada de Aplicaciones Personalizadas para Redes Móviles (o CAMEL, por sus siglas en inglés), recibiendo la información y procesándola de la base de datos para así recibir un vector de coordenadas de la matriz DTMF, denotando el archivo de audio a reproducir por el sistema para acceso del abonado.

El proyecto se enfocará en la transacción ISUP entre la central y el sistema a desarrollar, el cual se logrará mediante una aplicación escrita en lenguaje PHP, sobre un sistema operativo "Open Source", en el presente caso, se trabajará sobre el sistema operativo de código abierto Linux. En el transcurso del

documento será detallado el uso del servidor XAMPP (**X**, para cualquiera de los diferentes sistemas operativos, **A**pache, **M**ySQL, **P**HP, **P**erl) que se utilizará para implementar la simulación, el cual consiste en un servidor web (Apache) y la base de datos de MySQL (Lenguaje de Consulta Estructurado por sus siglas en inglés) para la interpretación del lenguaje PHP. Se ha escogido este servidor por poseer licencia de carácter libre, eliminando la preocupación por el costo de alguna licencia de software.

El IVR será implementado de manera virtual sobre Linux mediante el lenguaje Web PHP. Su código puede fácilmente ser utilizado junto con código HTML y debe ser procesado por un servidor web para mostrar la página creada, que trabajará con un teclado virtual que simulará una marcación por “*Touch-Tone*” (en español, tono por toque) para solicitar al usuario que presione el botón virtual correspondiente a las consultas o acciones disponibles.

Se trabajará en conjunto con otras aplicaciones correspondientes a los protocolos acompañantes a ISUP, más específicamente con las aplicaciones de CAMEL e INAP para accesos a bases de datos e información de tarificación.

## **1.4 Objetivos del Proyecto**

### **1.4.1 Objetivo General**

Presentar la documentación completa y detallada de una IVR virtual, explicando su implementación y dando las pautas para el desarrollo de aplicaciones interactivas que puedan operar en el protocolo ISUP.

### **1.4.2 Objetivos Específicos**

- Describir el funcionamiento y operación del protocolo ISUP.
- Implementar un IVR que procese los mensajes del protocolo de señalización ISUP para los servicios de consulta de saldos en una red telefónica móvil.
- Describir los principales flujos de señalización que intervienen en una comunicación ISUP y determinar el alcance en una red de telefonía móvil.

## **1.5 Alcance del Proyecto**

El alcance de implementar una IVR virtual es dar una descripción clara del respectivo proceso, del código necesario para su implementación, y de explicar acerca de los protocolos relacionados, su rol en el sistema SS7 y el análisis de la compatibilidad en la comunicación entre protocolos, mediante el

análisis de la estructura de los mensajes a utilizar dentro de la transacción y la cantidad de bits empleada por cada campo.

Un detalle que se tomará mucho en cuenta será la respectiva base de datos a crearse para la IVR virtual, la cual contendrá toda la información de abonados suscritos. Esta llega al sistema mediante una comunicación con el sistema de Prepago, el cual maneja directamente la mencionada base de datos.

### **1.6 Limitaciones del Proyecto**

Es necesario dar a notar que a pesar de la flexibilidad del lenguaje utilizado y la complejidad del sistema en un escenario real, hay ciertos aspectos que difieren del verdadero sistema. Uno de ellos es el ingreso de un código secreto para recargas de saldo, el mismo que sería de menor tamaño al del ambiente comercializado por las operadoras por motivos de tiempo de desarrollo.

Para operar como es debido, es necesario que la máquina donde se encuentre implementada esté conectada a una red con las demás máquinas, y verificarse bien el estado de dicha conexión. Sin embargo la seguridad de dicha red es baja, y para ejecutar pruebas generales simulando llamadas se requieren todas las máquinas anexas a la red, correspondientes al celular y a las máquinas que manejen los demás protocolos del sistema. Este es un protocolo donde no se transporta la voz mediante portadoras, tales como E1, en su lugar se simula un canal que se reserva en la comunicación a través de los mensajes de respuesta y confirmación del protocolo ISUP.

La máquina donde se desarrolla el proyecto posee una arquitectura de 64 bits, por lo que la implementación del mismo tendrá ello como requisito, además de poseer el sistema operativo indicado, en el presente caso, Ubuntu LTS 14.04 de arquitectura x64.

### **1.7 Metodología**

Uno de los objetivos principales que se han planteado para este proyecto es dar a conocer de un modo metódico de implementar y ejecutar el proyecto, el mismo que debe servir de guía para implementaciones parecidas y de investigaciones que se realicen en el futuro en torno a los protocolos del sistema SS7 y sus respectivos mensajes e interacciones.

El primer capítulo tratará acerca de los antecedentes de los sistemas de señalización, IVR, y se analizará el alcance y lo que justifica al desarrollo del presente proyecto, se dará a conocer además la metodología propuesta: la implementación a través del lenguaje de Preprocesador de Hipertexto (PHP).

El segundo capítulo constará de la fundamentación teórica del sistema de respuesta, de la señalización de canal común y del protocolo ISUP, introduciendo conceptos básicos, explicaciones detalladas de la transacción del protocolo, dando a conocer cada tipo de mensaje y su influencia en la comunicación entre los llamantes.

El tercer capítulo detallará todo lo relacionado con el desarrollo del proyecto. Paso a paso se hará explicación de la implementación del sistema, dando a



conocer su respectivo código, los comandos que se utilizarán en el mismo y las respectivas descripciones de cada parte del sistema.

Por último, en el cuarto capítulo se efectuará un análisis de los posibles resultados del proyecto, con sus simulaciones y otras pruebas. Conclusiones y recomendaciones se enlistarán al final del presente.

## **CAPÍTULO 2**

### **2. FUNDAMENTACIÓN TEÓRICA**

#### **2.1 Introducción a SS7**

SS7, Sistema de Señalización por Canal Común No. 7, es un conjunto de protocolos que cumplen una función muy importante en las redes telefónicas: el establecimiento y la respectiva finalización de llamadas. Una red de Señalización por canal común está conformada por nodos de conmutación y los respectivos procesos que se interconectan por enlaces de transmisión. Este sistema sucedió a lo que fueron los sistemas de señalización por banda, antes de evolucionar con el paso de los años a lo que se conoce hoy.

### 2.1.1 Reseña

Previo a los inicios de la señalización por canal común, se trabajaba con señalizaciones “en banda” para todas las redes telefónicas. Estas requerían de un troncal para poder transportarse de un extremo hacia el otro. En la Señalización de Canal Asociado (o CAS por sus siglas en inglés), la información de señalización para una troncal era llevada por la troncal propiamente. En la Señalización de Canal Común (o CCS por sus siglas en inglés) un enlace de señalización común lleva los mensajes de señalización para un número de troncales. CCS fue desarrollado para la tercera generación introducida en los años sesenta. Existen muchas razones para el cambio entre CAS y CCS, una de las principales sostenía que era más barato implementar la señalización de canal común, debido a que se utilizaban un número relativamente menor de enlaces de señalización [4]

La primera generación de señalización por canal común fue introducida en los años setenta. Vino en 2 versiones Señalización de canal común entre oficinas (CCIS, por sus siglas en inglés), definido por Sistemas Bell, y el Sistema de Señalización 6 (SS6). A CCIS y SS6 lo siguió el Sistema de Señalización No. 7 (SS7). El objetivo de SS7 era básicamente desarrollar un sistema de señalización que pudiera ser usado mundialmente, incrementando el alcance de SS6 y CCIS.

Los sistemas de voz interactiva (IVR por sus siglas en inglés) empezaron en la década de los treinta como un proyecto de los Laboratorios Bell, con la

finalidad de sintetizar la voz. Su desarrollo continuó y ya en los años setenta, tras haberse introducido los primeros teléfonos con marcación por DTMF (Multi-Frecuencia de Tono-Dual) años atrás, ya muchos centros de llamadas utilizaban sistemas automatizados de respuesta, a pesar de que la tecnología para ello en esa época era considerablemente costosa. En los años noventa esta tecnología migró a la telefonía informática, de donde se implementó el modelo actual del sistema IVR. Ya en los años dos mil volvió a migrar al estándar VXML con aplicaciones de código propietario, volviéndose mucho más barato de implementar [5]

Los beneficios principales de la utilización del protocolo de Red Digital de Servicios Integrados Parte de Usuario (ISUP) son la estandarización de intercambio de mensajes e incrementado ancho de banda de señalización. ISUP posee diversas variantes especificadas por la ITU-T, que varían en características como tipos de mensaje, parámetro... [6]

### **2.1.2 Identificación de puntos de señalización**

Cada punto de señalización de una red SS7 se identifica por códigos de punto (PC por sus siglas en inglés). Con un formato de 14 bits, indican las zonas donde se opera, así como también el grupo que se forma y se enviará por el canal.

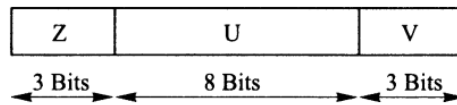


Figura 2.1 Zonas de Códigos de Punto

Donde:

Z: identifica seis zonas geográficas del mundo (7 en decimal para Latinoamérica)

U: representa un grupo de puntos de señalización en una red.

V: identifica un centro de conmutación particular internacional

### 2.1.3 Tipos de nodos

Existen 3 tipos de nodos en una red de señalización SS7:

- Punto de transferencia de señal (STP)
- Punto de conmutación de servicio (SSP)
- Punto de control de servicio (SCP)



Figura 2.2 Tipos de nodos de SS7

**STP:** STP es un switch de paquetes donde el tráfico de red entre puntos de señalización podrían ser encaminadas. Una STP encamina cada mensaje

entrante a una dirección de señalización saliente basada en información de ruteo dentro del mensaje SS7.

**SSP:** SSP es un switch que se encarga de la inicialización, duración, terminación y redirección de llamadas. Una SSP envía mensajes de señalización a otros SSP para manejar o configurar circuitos de voz necesarios para ejecutar una llamada.

**SCP:** SCP es una base de datos centralizada que determina la ruta de una llamada tras ser notificado por la SSP a través de un mensaje. La SCP responde a la SSP enviándole la información del número al cual se marcó (Número B) [7]

En resumen, decimos que STP se encarga de transferir los mensajes SS7 entre nodos; SSP es un conmutador de voz que incorpora funcionalidades de SS7; y SCP se considera como una interfaz que interactúa entre las bases de datos de las telecomunicaciones y la red SS7 propiamente.

#### **2.1.4 Modos de señalización**

Este término se refiere a la relación que existe entre dos nodos SS7. Pueden ser asociados o quasi-asociados. El modo asociado consiste en la conexión directa entre dos SSP, origen y destino; mientras que en el modo quasi-

asociado los mensajes SS7 deben pasar por un nodo intermedio (por ejemplo, STP).

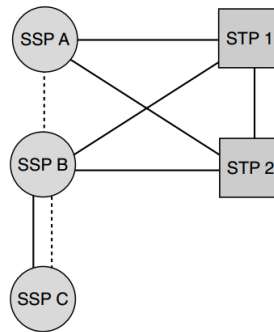


Figura 2.3 Modos de señalización

### 2.1.5 Estructura

Fuertemente bajo la influencia del modelo de Interconexión de Sistemas Abiertos (OSI, por sus siglas en inglés), la red SS7 está constituida por varios protocolos, los cuales se agrupan en cuatro niveles principales, que vendrán a ser equivalentes a las capas del modelo OSI. Dichos niveles son:

- Nivel 1: Nivel físico
- Nivel 2: Nivel de enlace de datos
- Nivel 3: Nivel de red
- Nivel 4: Nivel de partes de usuario

Los tres primeros niveles de la red son constituidos por un solo protocolo, el protocolo de Parte de Transferencia de Mensajes (MTP por sus siglas en

inglés), que es el principal responsable del tráfico de los mensajes de la red SS7 y provee los servicios de transferencia de mensajes para los usuarios. Este protocolo se constituye de tres niveles, cada uno formando parte de los tres primeros niveles de la arquitectura de SS7. MTP de niveles 1, 2 y 3 correspondientes a los niveles Físico, Enlace de Datos y Red, respectivamente.

**MTP1:** MTP de nivel 1 es la parte física de la arquitectura del protocolo. Esta define las características eléctricas del Enlace de Señalización Digital.

**MTP2:** MTP de nivel 2 es el encargado del enlace de datos principal, y establece transmisiones precisas sobre enlaces de señalización. En este nivel se implementa el control de flujo de mensajes, validaciones de secuencia y corrección de errores.

**MTP3:** En el nivel 3 de MTP, se establece el enrutamiento de los mensajes entre puntos de señalización en el sistema. Cuando un mensaje se topa con un enlace inválido o fallido, este es redirigido a un enlace con tráfico menor. En este proceso, no se pierden estos mensajes ni hay la necesidad de efectuar una reorganización de los mismos [8]



El cuarto nivel de SS7, llamado el nivel de Parte de Usuario está constituido por diversos protocolos. En la figura 2.4 podemos observar la estructura propiamente de SS7, detallada a continuación:

- Parte de usuario de telefonía (TUP)
- Parte de usuario RSDI (ISUP)
- Parte control de conexión de señalización (SCCP)
- Capacidades de transacción (TC)
- Entidades de aplicación (AE)
- Elementos de servicio de aplicación (ASE)

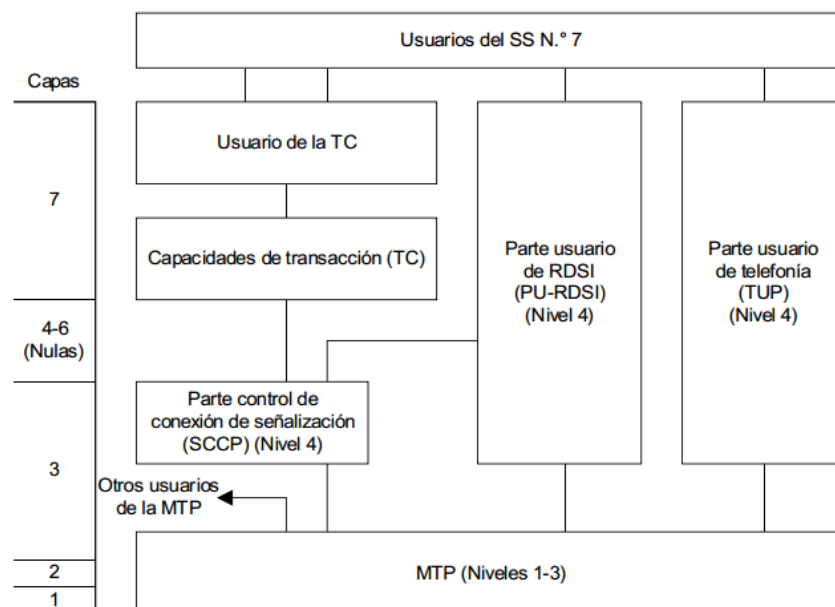


Figura 2.4 Estructura de SS7

En la capa siguiente tenemos al protocolo ISUP, encargado del control de la llamada, esto es, es el responsable de establecer y terminar las troncales usadas en una red celular. Es el reemplazo de TUP, ya que él no soportaba la transmisión de datos o circuitos digitales.

Así mismo, en esta capa se encuentra SCCP, el cual proporciona funciones adicionales a la parte de transferencia de mensajes con objeto de prestar servicios de red sin conexión y servicios de red con conexión. Lo propio ISUP, los cuales se encuentran en la misma capa que SCCP y comprenden las funciones de señalización requeridas para proporcionar servicios para aplicaciones vocales y no vocales en la RDSI.

## **2.2 Definición de ISUP**

El protocolo de Parte de Usuario de la Red Digital de Servicios Integrados (ISUP, por sus siglas en inglés) es uno de los protocolos principales empleados por el Sistema de Señalización Número 7 (SS7) cuya función principal es configurar, definir los parámetros y administrar los circuitos de una red telefónica. Estos circuitos son los encargados del transporte de datos de llamada y de la voz, a través de toda la red pública telefónica.

ISUP es utilizado tanto para llamadas que son parte de la Red Integrada de Servicios como para llamadas que no son parte de dicha red. ISUP es basado en el protocolo telefónico de parte de usuario (TUP) el cual también forma parte del cuarto nivel de SS7, el nivel de Partes de Usuario, podría decirse que ISUP

es básicamente un TUP compatible con redes inteligentes de servicios integrados [9]

Una transacción básica de ISUP contiene algunos mensajes los cuales transfieren parámetros e información detallada de cada parte de la llamada, el número A y número B, junto con la información que define el circuito de la llamada, una transacción puede suceder incluso entre la central y sistemas automatizados para acceso del usuario, como por ejemplo una IVR, para solicitar información de la base de datos de un sistema de prepago. Se estudiará con detalle cada paso de una llamada en ISUP, enlistando los parámetros necesarios para una transacción exitosa.

### **2.2.1 Transacción básica**

La señalización de ISUP ocurre durante llamadas cualquiera, sean o no parte de una Red Inteligente de servicios. En una llamada básica ocurren intercambios de mensajes de notificación de inicialización, los cuales deben ser contestados con un aviso de respuesta. Estos mensajes contienen datos a ser entregados al destino, cada uno abarcando una determinada cantidad de bits, estos pueden enviarse ya sea en un “flujo” de cifras binarias o en una sola cadena hexadecimal [10]



**2. Confirmación de dirección enlazada:** El SSP de destino procede a comprobar que los dígitos registrados coincidan con el de su terminal, y entonces notifica al número A si la línea se encuentra libre y disponible para realizar la llamada. De estarlo, el SSP de destino envía un *Mensaje de Dirección Completa (ACM)* al SSP de origen y de ese modo se reserva el circuito de llamada. Un STP encamina el mensaje al origen para establecer el canal de voz. En estos momentos el llamante escucha un tono que le indica que el llamado debe contestar, mientras que el terminal de destino emite el tono de llamada.

**3. Confirmación de respuesta:** Si la persona que es llamada levanta el terminal durante el tiempo límite del tono, este cesa y el SSP retorna al origen un *Mensaje de Respuesta (ANM)* a través de su correspondiente STP. Una vez se confirme se trata del circuito reservado, la llamada toma a lugar, empezando también la facturación de la misma.

**4. Desconexión del circuito:** Tras haberse dado la llamada, dependiendo de cuál de las dos partes es la que cuelgue la llamada, el SSP correspondiente a quien cuelgue envía a la otra parte un *Mensaje de Liberación (REL)* el cual es encaminado por la correspondiente STP. En caso de que se trate de una línea ocupada, siempre es el SSP de destino quien envía el mensaje.

**5. Confirmación de desconexión:** Cuando el mensaje REL es recibido, el switch en el destino desconecta el circuito de la línea, dejándolo en un estado de suspensión al mismo tiempo que responde con un *Mensaje de Liberación Completada* (RLC) hacia el origen. Esto culmina la facturación de la llamada, liberando el circuito de voz y dejándolo disponible para más llamadas.

En los casos de llamadas a terminales automatizados, como es el caso de una IVR, el mensaje IAM es respondido en cambio por un *Mensaje de Conexión* (CON), debido que se trata de un sistema que responde de inmediato. Este reemplaza el rol de los mensajes ACM y ANM.

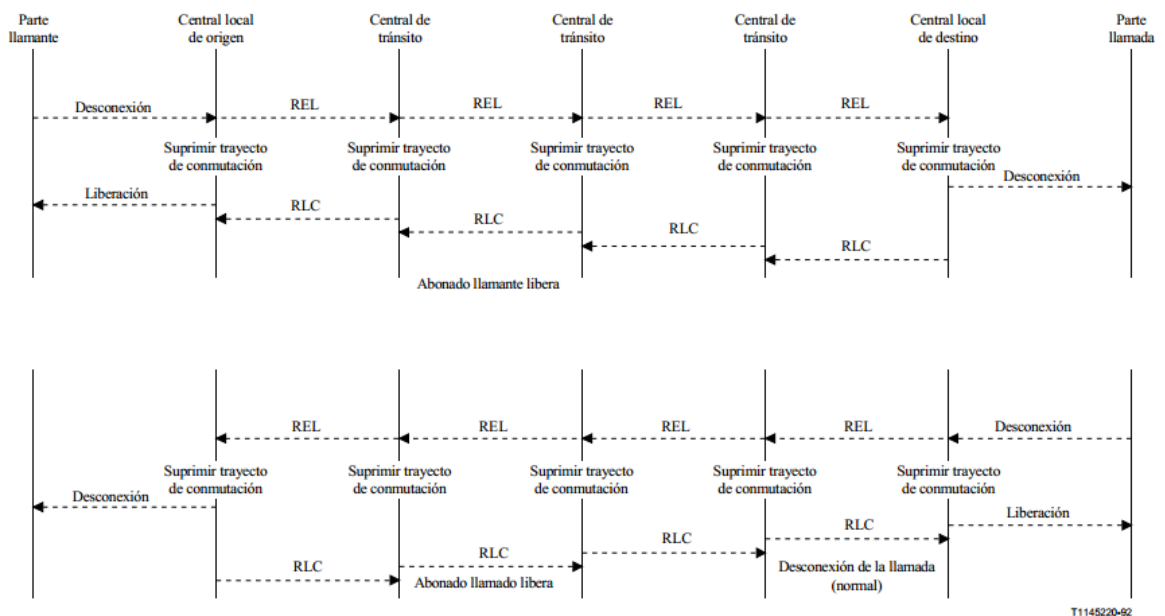


Figura 2.6 Liberación de llamada normal

### 2.2.2 Transacción con terminal automático

En ciertos casos, la llamada suele ser a un terminal asignado por la correspondiente operadora para servicios como, consultas, promociones, conexión a centros de servicio, entre otras. Estos tipos de llamada tienen un flujo con algunas diferencias al flujo convencional de mensajes de una llamada normal.

En un caso real, ciertos mensajes son reemplazados por otros tipos de mensaje del protocolo ya que su contenido en cuanto a parámetros es similar, estos tipos de mensajes poseen un parámetro especial denominado *Número Conectado*, que funciona como un identificador del número conectado a la llamada.

Nótese que estas transacciones se realizan en relación al protocolo SIP (*Protocolo de Iniciación de Sesión*), entre los cuales sucede la transacción con el mensaje CON [11]

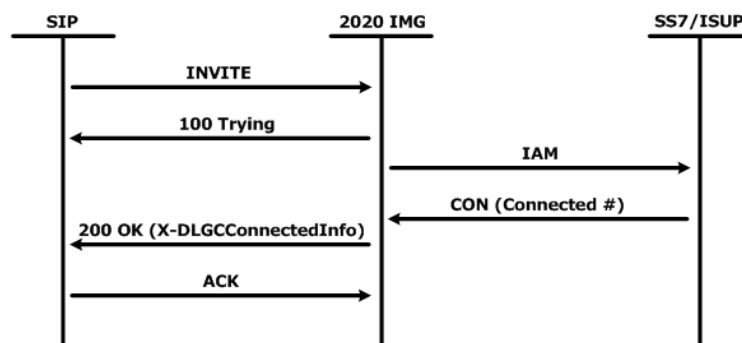


Figura 2.7 Transacción entre SIP e ISUP con mensaje CON

### 2.2.3 Estructura de los mensajes

Cada uno de los mensajes que se envían o reciben en ISUP cuentan con un determinado número de parámetros, estos incluyen identificadores del tipo de mensaje a enviar, información del circuito, entre otros.

Todo mensaje de ISUP contiene en su formato la siguiente información predeterminada [12]:

- **Código de Identificación del Circuito (CIC):** Identifica el circuito de voz relacionado al mensaje.
- **Tipo de Mensaje:** Contiene una cifra que determina el tipo de mensaje ISUP relacionado.
- **Parte Fija Obligatoria:** Constan de todos los parámetros de mensaje de una longitud fija.
- **Parte Variable Obligatoria:** Contiene parámetros de longitud variable, sus variables serían longitud del parámetro y su contenido.
- **Parte Opcional:** Campos opcionales que pueden incluirse, como nombre del parámetro y su longitud. Estos no son obligatorios [13]



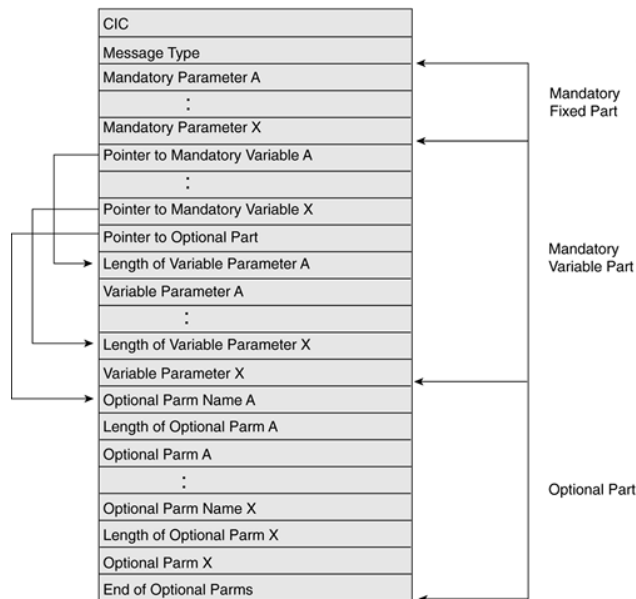


Figura 2.8 Estructura de un mensaje de ISUP

#### 2.2.4 Mensajes básicos de la llamada

La sección a introducirse contiene los parámetros obligatorios en cada uno de los mensajes básicos de una transacción revisados en la sección anterior, junto con un diagrama que muestra dichos parámetros. Dichos parámetros se encuentran establecidos y debidamente documentados en la recomendación Q.763 de la ITU-T.

#### Mensaje IAM

Un IAM contiene la información suficiente necesitada para establecer la llamada. Es el mensaje más grande puesto a que es compatible con varios parámetros opcionales que pueden ser incluidos. IAM es un mensaje de Tipo 1.

Message Type (IAM)
Nature of Connection Indicators <ul style="list-style-type: none"> <li>• <i>Satellite Ind.</i></li> <li>• <i>Continuity Ind.</i></li> <li>• <i>Echo Control Device Ind.</i></li> </ul>
Forward Call Indicators <ul style="list-style-type: none"> <li>• <i>Nat/Int'l Call Ind.</i></li> <li>• <i>End-to-End Method Ind.</i></li> <li>• <i>Interworking Ind.</i></li> <li>• <i>End-to-End Information Ind.</i></li> <li>• <i>ISDN User Part Ind.</i></li> <li>• <i>ISDN User Part Preference Ind.</i></li> <li>• <i>ISDN Access Ind.</i></li> <li>• <i>SCCP Method Ind.</i></li> <li>• <i>Ported Number Translation Ind.</i></li> <li>• <i>Query On Release Attempt Ind.</i></li> </ul>
Calling Party's Category
Transmission Medium Requirement (ITU Networks)
User Service Info (ANSI Networks)
Called Party Number
Optional Params

Figura 2.9 Estructura básica de un mensaje IAM

Como se puede observar, IAM contiene dos grupos de parámetros obligatorios, correspondientes a los parámetros fijos y variables.

- **NOC (Naturaleza de Indicadores de Conexión):** Estos campos contienen información acerca de la conexión correspondiente al lado de quien reciba la llamada. Entre ellos se incluye un indicador de continuidad, que indica si existe una continuidad fija durante la conexión del circuito para la llamada.

- **FCI (Indicadores de Llamada Entrante):** Estos indicadores son los que dan a conocer las preferencias de la configuración de la llamada y el estado en el que se encuentre el circuito, denotando las condiciones en las que se dará una llamada

- **Categoría de la Parte Llamante:** Es la categoría en donde está clasificada la parte llamante, ya sea un abonado, operador, teléfono público...
- **TMR (Requerimiento del Medio de Transmisión):** Aquí se detallan los requerimientos necesarios para realizar la llamada, como frecuencia del audio, tamaño del canal...
- **USI (Información de Servicio de Usuario):** Es el equivalente a TMR para los circuitos bajo el estándar ANSI (Instituto Nacional Americano de Estándares), ya que TMR es restringido a los circuitos de estándares de la ITU-T.
- **Número de la Parte Llamada:** Es el número de destino a donde se está realizando la llamada.

### **Mensaje ACM**

ACM se envía por el nodo de destino para notificar la satisfactoria conectividad de un circuito de voz ya reservado, tras haber recibido antes un IAM. ACM también señala información sobre el estado de la llamada a través de indicadores de llamada hacia atrás (BCI por sus siglas en inglés), que son contrarios a los parámetros FCI contenidos en el IAM. ACM no es un mensaje necesariamente obligatorio, es posible un envío directo de un ANM, llamado "respuesta rápida". ACM es un mensaje de tipo 6.

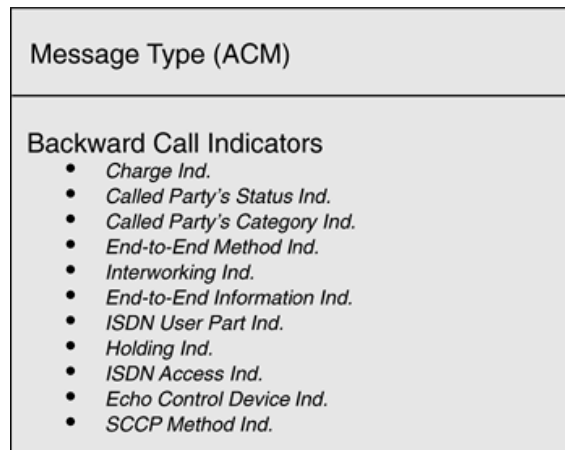


Figura 2.10 Estructura de un mensaje ACM

### **Mensaje ANM**

Un ANM se envía una vez la persona llamada descuelgue el terminal para responder a la llamada. Este mensaje contiene ciertos parámetros opcionales, sin embargo no contiene parámetros obligatorios a más de su tipo de mensajes.

ANM es de tipo 9.

### **Mensaje REL**

Un REL notifica la liberación del circuito. Este tipo de mensaje puede ser enviado en cualquier dirección del circuito, ya sea del llamante a quien reciba o viceversa. Contiene parámetros obligatorios llamados Indicadores de Causa, que denotan el motivo de la liberación de la llamada. REL es un mensaje de tipo 12.



Figura 2.11 Estructura de un mensaje REL

### Mensaje RLC

RLC actúa como el acuse de recibo de un REL. Una vez se ha receptado, el circuito entra en estado de libre. RLC es un mensaje de tipo 16.

#### 2.2.5 Otros mensajes

**SAM:** El *Mensaje de Dirección Subsecuente* SAM está disponible en redes con soporte de marcado por superposición, el cual es un caso especial donde el IAM no contiene por completo el número al que se llama, con un SAM conteniendo los dígitos faltantes.

**CPG:** Un mensaje de *Progreso de Llamada* CPG contiene información opcional del estado y progreso de la llamada.

### 2.3 Acerca del Sistema IVR

El sistema automatizado IVR es bastante común en sistemas celulares que se manejan por señalización, los cuales ofrecen un sinnúmero de servicios a abonados u otros tipos de suscriptores. La sección presente se encarga de dar a conocer más acerca de este tipo de sistemas, y ofrecer un detalle más teórico sobre su funcionamiento.

### 2.3.1 Definición

IVR, de las siglas Respuesta de voz interactiva en inglés, hace referencia a aplicaciones interactivas entre los dispositivos móviles, teniendo como entrada tonos o voz según sea el caso de implementación. Es capaz de interactuar con el ser humano a través de grabaciones de voz, obteniendo información del mismo para cumplir a cabalidad ciertos procesos [14]

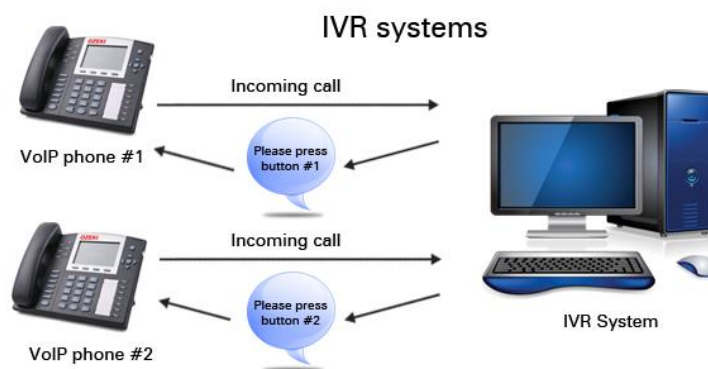


Figura 2.12 Funcionamiento de un IVR

### ¿Cómo funciona?

El móvil realiza una llamada un número de consulta proporcionado por la operadora. El sistema IVR recibe ese número y en respuesta al usuario envía grabaciones de una base de datos disponible para la transacción. El usuario recibe esta información y procede a responder con las teclas de marcado necesarias para continuar el proceso correspondiente, en nuestro caso la

consulta de saldos de su línea telefónica. En una configuración básica de IVR, se realizan dos procesos básicos:

- **Captura de datos**

Se utilizan los tonos de marcado DTMF, presionando las teclas del teléfono.

- **Entrega de información**

Hay varias formas de realizar este proceso, pero la más usada es a partir de grabaciones de voz, alojadas en una base de datos propia de IVR, con un formato definido a manera de vectores. Cada vector posee una secuencia de grabaciones, que al ser programadas una detrás de otra, podremos obtener una frase total que será entendida de manera eficiente por el usuario.

### 2.3.2 Tonos DTMF

Como se mencionaba anteriormente, para la captura de datos en una IVR se utilizan los tonos de marcado DTMF. No es otra cosa que los tonos que se obtienen al presionar las teclas de nuestro teléfono. Éstas se enviarán por el respectivo canal de voz previamente seleccionado en el proceso anteriormente descrito.

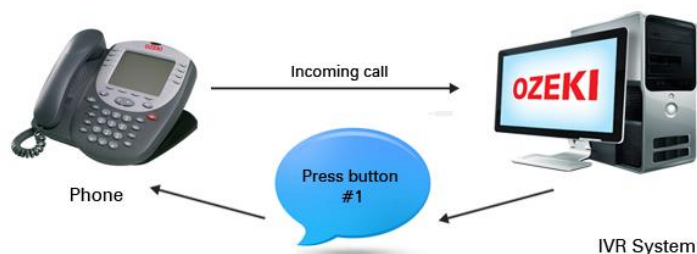


Figura 2.13 Marcado por teclado a un sistema IVR

Una señal DTMF consiste en la suma de dos tonos, uno de un grupo bajo y el otro de un grupo alto. Las frecuencias de los tonos se eligieron muy cuidadosamente, para no tener problemas con sus armónicos y evitar el deterioro de la señal por la intermodulación que existe entre ellos. Para cada tecla existe una combinación de frecuencias, detallada en la siguiente tabla [15]

Frequency (Hz)	1209	1336	1477	1633
697	1	2	3	A
770	4	5	6	B
852	7	8	9	C
941	*	0	#	D

Figura 2.14 Tabla de frecuencias DTMF

Según ITU, en la recomendación Q.24, las especificaciones para la detección DTMF son las siguientes:

**Tolerancia a la frecuencia:** Un símbolo válido DTMF debe tener una desviación en frecuencia dentro del 1.5% de tolerancia. Los símbolos con una desviación en frecuencia mayor al 3.5% deberán ser rechazados.

**Duración de la señal:** Un símbolo DTMF con una duración de 40ms debe ser considerado válido. La duración de la señal no debe ser menor de 23ms.

**Atenuación de la señal:** El detector debe trabajar con una relación señal-ruido (SNR 43) de 15db y en el peor caso con una atenuación de 26dB.



**Interrupción de la señal:** Una señal DTMF válida interrumpida por 10ms o menos no debe ser detectada como dos símbolos distintos.

**Pausa en la señal:** Una señal DTMF válida separada por una pausa de tiempo de al menos 40ms debe ser detectada como dos símbolos distintos.

**Fase:** El detector debe operar con un máximo de 8dB en fase normal y 4dB en fase invertida.

## **2.4 Acerca del Sistema Operativo Linux**

Linux es un sistema operativo de código abierto, por lo que su desarrollo constante se debe a las colaboraciones de varios programadores que aportan con el proyecto, abarcando hasta a alrededor de 1000 desarrolladores contribuyendo con cada lanzamiento de núcleo del sistema. Este núcleo puede editarse y redistribuirse a gusto de quien haya trabajado en él. Hasta el día de hoy numerosas distribuciones se han liberado a los usuarios y ahora poseen una gama amplia de distribuciones de núcleo a escoger.

```

[ 0.953693] serio: i8042 KBD port at 0x60,0x64 irq 1
[ 0.954816] serio: i8042 AUX port at 0x60,0x64 irq 12
[ 0.956069] mousedev: PS/2 mouse device common for all mice
[ 0.957743] input: AT Translated Set 2 keyboard as /devices/platform/i8042/ser
rio0/input/input0
[ 0.960144] rtc_cmos rtc_cmos: rtc core: registered rtc_cmos as rtc0
[ 0.961230] rtc0: alarms up to one day, 114 bytes nvram
[ 0.962307] cpuidle: using governor ladder
[ 0.963320] cpuidle: using governor menu
[ 0.964366] TCP cubic registered
[ 0.965316] NET: Registered protocol family 10
[ 0.967271] Mobile IPv6
[ 0.970938] NET: Registered protocol family 17
[ 0.972209] Registering the dns_resolver key type
[ 0.973334] Using IPI No-Shortcut mode
[ 0.974557] registered taskstats version 1
[ 0.976637] rtc_cmos rtc_cmos: setting system clock to 2011-09-09 20:32:52 UT
C (1315600372)
[ 0.982005] Initializing network drop monitor service
[ 0.983351] Freeing unused kernel memory: 404k freed
[ 0.984627] Write protecting the kernel text: 2768k
[ 0.985825] Write protecting the kernel read-only data: 1068k
[ 0.986935] NX-protecting the kernel data: 3376k
Loading, please wait...

```

Figura 2.15 Núcleo Linux versión 3.0 en arranque

### 2.4.1 Distribución de Linux

Cuando se habla de una distribución, se habla de conjuntos de aplicaciones agrupadas que pueden contener software propietario, utilizado para fácilmente instalar Linux en un ordenador. Toda distribución está basada en el núcleo Linux, y tienen en común su código y herramientas con el proyecto GNU, el cual empezó como el desarrollo de un sistema operativo basado en Unix. Es más, varios componentes de Linux como su interpretador de código, fueron producidos por el proyecto GNU [16]

Entre algunas de las distribuciones más conocidas se encuentran:

- **Debian:** Distribución creada alrededor de 1993, al mismo tiempo que varias utilidades del proyecto GNU. La intención de la distribución era originalmente enfocarse en software de libre distribución, el cual pone a disposición de sus usuarios a través de Internet.

- **SUSE:** Distribución creada en 1994, fue popular por su fácil instalación y sencilla interfaz gráfica, que le facilitaba bastante al usuario tareas básicas. Este programa de instalación, llamado YaST2, permite administrar las redes, actualizaciones de software y controles de usuario.
- **Slax:** Distribución basada en Slackware, una distribución de Linux creada en 1993, la cual también fue la base para SUSE. Slax posee facilidades para personalización a través de paquetes, que pueden contener software adicional o crear nuevos módulos de Slackware para facilitar instalaciones.

#### **2.4.2 La distribución Ubuntu**

Ubuntu es una distribución de Linux basada en Debian, cuya interfaz de escritorio utilizada es Unity. Una instalación de Ubuntu incluye de por sí varios paquetes de software libre disponibles, en especial LibreOffice, que consta de una suite con procesador de palabras, hoja de cálculo y editor de presentaciones, el cliente ThunderBird de Mozilla con su navegador Firefox, entre otros programas. Ubuntu cuenta con un centro de descargas de software donde se encuentra disponible software libre previamente incluido en una instalación normal.



Figura 2.16 Escritorio de Ubuntu

## **CAPÍTULO 3**

### **3. DESARROLLO DEL PROYECTO**

#### **3.1 Análisis**

Un sistema de señalización SS7 consiste principalmente en diversos equipos o componentes en donde se efectuará la respectiva señalización que den lugar a una satisfactoria comunicación entre las partes involucradas. Para simular este sistema, es necesario comprender la estructura de cada máquina a utilizarse, junto con las herramientas necesarias para llevar a cabo la simulación. Cada una de las máquinas será configurada para ser parte de una pequeña red en donde se mostrará el comportamiento de un sistema de señalización, estas poseerán una interfaz donde se podrá apreciar el

intercambio de mensajes y sus respectivos parámetros en tiempo real. Todos los cambios realizados y archivos se actualizarán y estarán disponibles para cada equipo de esta pequeña red.

## **3.2 Preparativos para la implementación**

### **3.2.1 Software**

**XAMPP** es un servidor web el cual es libre e independiente de alguna plataforma. Se trata de una distribución sin costo alguno del servidor web **Apache**, mismo que es de código abierto y trabaja sobre HTTP (Protocolo de Transferencia de Hipertexto por sus siglas en inglés), y también contiene la base de datos MySQL (Mi Lenguaje Estructurado de Consulta por sus siglas en inglés), y es capaz de interpretar “scripts” de los lenguajes de programación PHP y Perl. XAMPP hasta el día de hoy se encuentra disponible para los principales sistemas operativos en el mercado, incluyendo a Solaris de Oracle [17]

Bajo el servidor XAMPP se implementará la red donde se conectarán las máquinas, las cuales utilizan como sistema operativo la distribución Ubuntu descrita en el capítulo anterior.

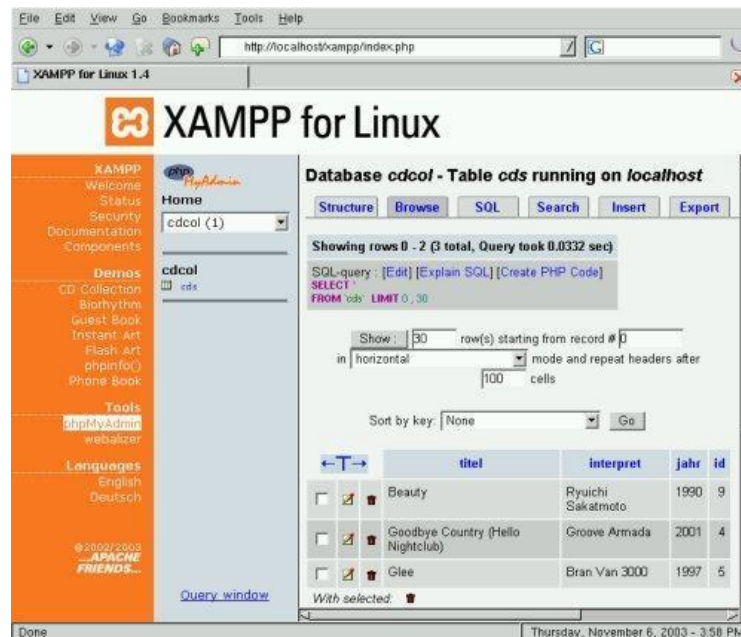


Figura 3.1 Sitio principal del servidor XAMPP

**NetBeans** es un Entorno de Desarrollo Integrado (IDE por sus siglas en inglés) cuyo propósito principal es el desarrollo en el lenguaje Java, concebido en conjunto por Sun Microsystems y Oracle. Es un proyecto fundado en el año 2000 y que hoy en día cuenta con casi 100 socios alrededor del mundo, junto con una creciente comunidad de usuarios.

En Noviembre de 2014, se introdujo la versión más reciente, 8.0.2, la cual será utilizada para el desarrollo del código. Cuenta con los respectivos paquetes para los lenguajes de programación soportados, como el paquete para C y C++, y el paquete de PHP que soporta la versión 5 del mismo.

NetBeans es compatible con versionado de Apache, donde será acoplado al proyecto para constantes actualizaciones del código fuente, que podrán ser

vistas por toda la red de trabajo. Con *versionado* se refiere al sistema de control de cambios de código del proyecto.

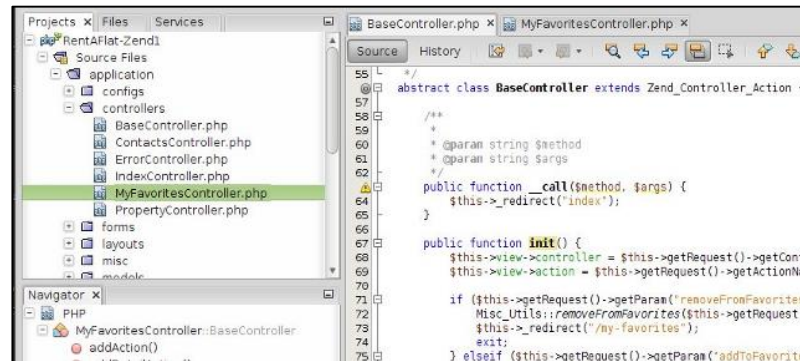


Figura 3.2 Interfaz básica de NetBeans

### 3.2.2 Hardware

Dentro de la red a simular, el componente correspondiente a IVR será implementado en una máquina, esta interactuará con otras dos máquinas mediante dos protocolos distintos. IVR ejecuta transacciones con la central MSC mediante ISUP, y con la base de datos del sistema prepago a través del protocolo Parte de Aplicación de Red Inteligente (INAP, por sus siglas en inglés). La máquina donde se simulará una IVR cuenta con las siguientes especificaciones:

- Modelo: Dell Inspiron 14R 5420
- Procesador: Inter Core i5-3210M 2.5 GHz
- RAM: 6.00 GB HDD: 500 GB
- Sistemas operativos: Windows 8.1 Pro. Ubuntu LTS 14.04





Figura 3.3 Máquina donde se implementa la IVR

La máquina se encuentra conectada a una red inalámbrica de área local (WLAN) personalizada utilizando un pequeño enrutador inalámbrico marca D-Link, de donde se conectan los equipos.

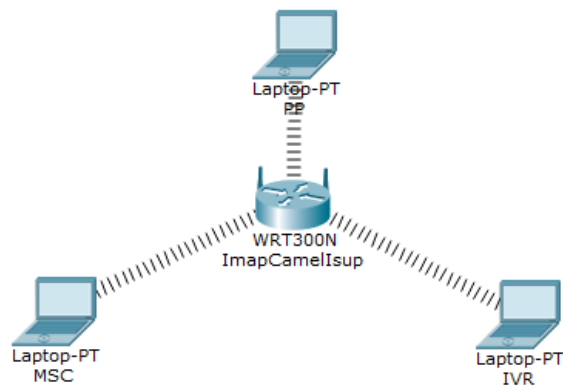


Figura 3.4 Ejemplo de una topología con tres de las máquinas en una red inalámbrica.

Para la integración con la red de actualización de localidad, esta máquina se conecta físicamente al enrutador a través de un cable Ethernet, mientras que

se debe conectar inalámbricamente a uno de los enrutadores de las centrales del sistema, ya sea la correspondiente a las ciudades de Quito o Guayaquil.

### 3.3 Implementación

En un computador que ya posee un sistema operativo, ya sea Windows o Mac OS, es posible crear una nueva partición de disco para instalar el sistema Ubuntu en ella. Para ello es necesario crear un disco de instalación del sistema operativo, ya sea en un CD/DVD o en una unidad USB.

Para crear una unidad USB de instalación de Ubuntu, puede descargar el software *Universal USB Installer* que se provee en el sitio web [www.pendrivelinux.com](http://www.pendrivelinux.com), e instalarlo siguiendo las sencillas indicaciones del asistente. Se descarga también la imagen en formato .iso del sistema operativo, disponible en el sitio web del mismo ([www.ubuntu.com/download/desktop](http://www.ubuntu.com/download/desktop)) ya sea para equipos de arquitecturas de 32 o de 64 bits, con el sitio web sugiriendo la versión recomendada.

1. Abrir el Universal USB Installer, y de la lista desplegable de la distribución a instalar, se selecciona Ubuntu.
2. Se da clic en "Browse" y se selecciona la imagen (.iso) de Ubuntu descargada.
3. De tener la USB a utilizar archivos importantes, respaldarlos ya que el proceso formateará la unidad para usarse como disco de arranque.

4. Tras insertarse ya la USB formateada, seleccionarla en el desplegable de seleccionar unidad y dar clic en “Create”.

### **3.3.1 Instalación de Ubuntu**

De instalar Ubuntu en algún computador con algún otro sistema operativo, se debe asignar una nueva unidad existente para ello, o bien crear una nueva partición del disco duro de ser la única unidad en el computador.

El instalador de Ubuntu permite realizar esta operación de un modo sencillo. También es posible crear manualmente una partición de disco desde las herramientas proporcionadas por el sistema operativo respectivo.

Para instalar Ubuntu deben seguirse los siguientes pasos:

1. Encender la computadora con la unidad USB de instalación conectada, tras aparecer la pantalla del BIOS entrar y seleccionar arrancar desde la USB. La ubicación de tal opción depende de la BIOS individual de cada equipo.
2. El instalador de Ubuntu aparecerá en pantalla, pedirá que se seleccione un idioma y a continuación se da clic en instalar Ubuntu.
3. Avanzar por las instrucciones del asistente en pantalla, cuando se solicite el modo de instalación, se selecciona “Instalar junto con Windows (7/8/8.1)”, y a continuación se mostrará un selector de espacio en el disco duro que se asignará a Ubuntu (de no haberse particionado manualmente).

4. Seleccionar la zona horaria del reloj, idioma del teclado, e insertar los datos del usuario, contraseñas y otros.
5. La instalación se completará después de un tiempo, tras finalizar se solicitará un reinicio del equipo.



Figura 3.5 Instalador de Ubuntu

Tras reiniciar el equipo, ya aparecerá un menú en donde se selecciona el sistema operativo a utilizar.

### 3.3.2 Instalación de XAMPP

El paquete de XAMPP para Linux puede ser descargado desde su propia página web (<https://www.apachefriends.org/download.html>), antes de ello se deber verificar que el archivo a descargar sea para la respectiva versión de Linux, de 32 o de 64 bits.

Tras descargar este paquete, se debe ejecutar desde el terminal de Ubuntu, para asignar los respectivos permisos de ejecución al paquete, ya que necesita acceso al directorio raíz.

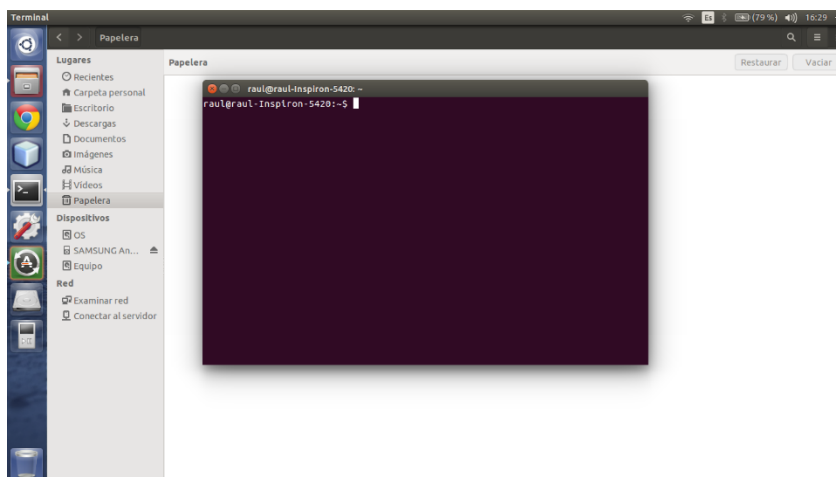


Figura 3.6 Terminal de Ubuntu

En Ubuntu se abre el terminal de comandos, al cual se puede acceder desde el panel de aplicaciones. Desde ahí se instalará XAMPP ingresando el código a continuación, con el nombre del archivo correspondiente descargado.

```
chmod 755 xampp-linux-x64-x.x.xx-installer.run  
sudo ./xampp-linux-x64-x.x.xx-installer.run
```

Dependiendo del sistema operativo, x64 corresponde a sistemas de 64 bits, mientras que x86 corresponde a 32 bits, la sección "x.x.xx" representa el número de versión de la aplicación descargada, y debe reemplazarse debidamente. Los comandos "sudo" solicitan acceso al directorio raíz del sistema operativo, y solicitan el ingreso de la contraseña de administrador de

poseer una. Al ingresar este comando aparecerá el asistente de instalación de XAMPP, siguiendo los pasos del asistente y asegurándose de que este se instale en el directorio /opt/lampp.



Figura 3.7 Instalación de XAMPP.

Tras completarse la instalación, ahora es posible levantar el servidor Apache ingresando al terminal y digitando el código a continuación:

```
sudo /opt/lampp/lampp start
```

Desde el navegador web, se digita la dirección `http://localhost`. XAMPP se mostrará de haberse completado exitosamente la instalación.

### 3.3.3 Instalación del IDE NetBeans

Para instalar NetBeans, primero se debe descargar e instalar el Kit de Desarrollo de Java para Linux desde su página oficial [18] Se recuerda tomar en cuenta la arquitectura del equipo para elegir correctamente la versión a descargar (32 o 64 bits). Tras descargar dicho kit, este debe ser reubicado en

la misma carpeta donde está localizado el servidor (/opt/lampp/) dentro de la carpeta "opt". Para ello se extrae el paquete .zip en la misma carpeta de descargas y se traslada la carpeta mediante el comando a continuación:

```
sudo mv /home/usuario/descargas/java1.8.0 /opt
```

Se procede entonces a ejecutar el instalador de NetBeans, obtenible desde su sitio oficial (<https://netbeans.org/downloads/>) con el comando siguiente:

```
sudo sh /home/usuario/descargas/netbeans-8.0.2-linux.sh
```

Se deben seguir las instrucciones del instalador, el cual puede ser la versión completa o simplemente la versión con soporte para PHP versión 5.

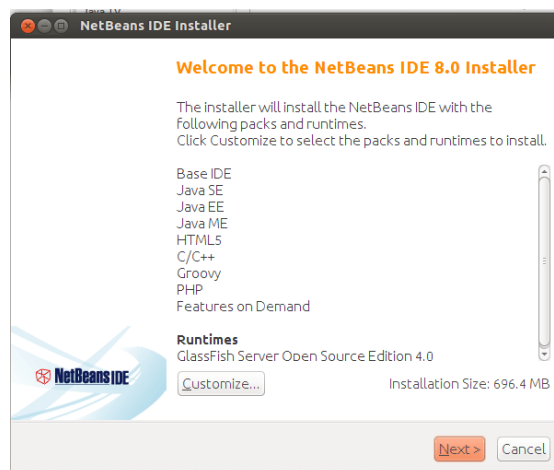


Figura 3.8 Instalación de NetBeans

Tras su instalación, NetBeans aparecerá en el panel de aplicaciones de Ubuntu, en la sección de aplicaciones instaladas. Se recomienda instalar una versión con plugins tanto de Java como de PHP 5.

Una vez asignada cada máquina una dirección IP con la que trabajará dentro de esta red, se registran los equipos y sus respectivos usuarios en una red de trabajo donde sea posible compartir dentro de toda la red los cambios realizados en los archivos del proyecto en tiempo real y poder descargar archivos de apoyo dentro de un repositorio.

**Página recomendada:** Assembla es un sitio web de administración de tareas y código en la nube. En este sitio web es posible crear un grupo de trabajo con su respectivo repositorio para almacenar código fuente, además de herramientas de administración del mismo. Assembla es compatible con el versionado de Apache (llamado SVN), sistema que habilita versionado inmediato de código fuente, permitiendo comparar cambios realizados con versiones anteriores del mismo.

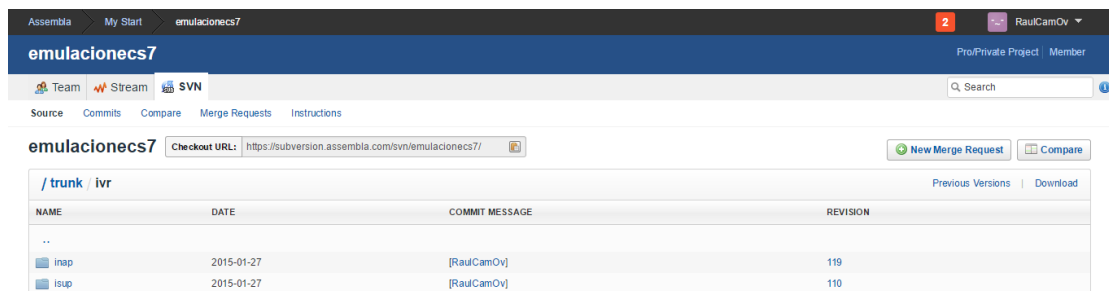


Figura 3.9 Interfaz de Assembla

Para descargar o subir una nueva versión del proyecto de NetBeans en donde se esté trabajando, se realizan los pasos a continuación:



1. Tras crear un proyecto, tomar nota de la carpeta física del computador en donde el proyecto se crea.
2. En la barra de menú de NetBeans, dirigirse a Team -> Subversion -> Import Into...
3. Aparecerá una ventana en donde se solicita al usuario colocar la URL (Localizador Uniforme de Recurso por sus siglas en inglés) del versionado a utilizar. El mismo se encuentra en la sección de SVN (Subversión) de Assembla en la sección "Checkout URL", mostrado en la imagen superior.
4. Se ingresa el usuario y contraseña de la cuenta de la página donde se encuentre el repositorio. Para evitar solicitud constante de estos datos, se coloca visto en el recuadro "Grabar usuario/contraseña".
5. La siguiente ventana solicita ingresar la dirección de la carpeta del repositorio, la cual debe elegirse dentro del mismo, y la carpeta local que se vinculará al proyecto.

Para subir nuevas versiones del código ejecutado con sus cambios, se siguen los pasos a seguir:

1. Se da clic derecho sobre el nombre del proyecto, y se elige Subversion -> Update -> Update to HEAD
2. Luego de realizarse la actualización, se vuelve a dar clic derecho y se elige Subversion -> Commit. En el cuadro que aparece, se da clic en "Commit".

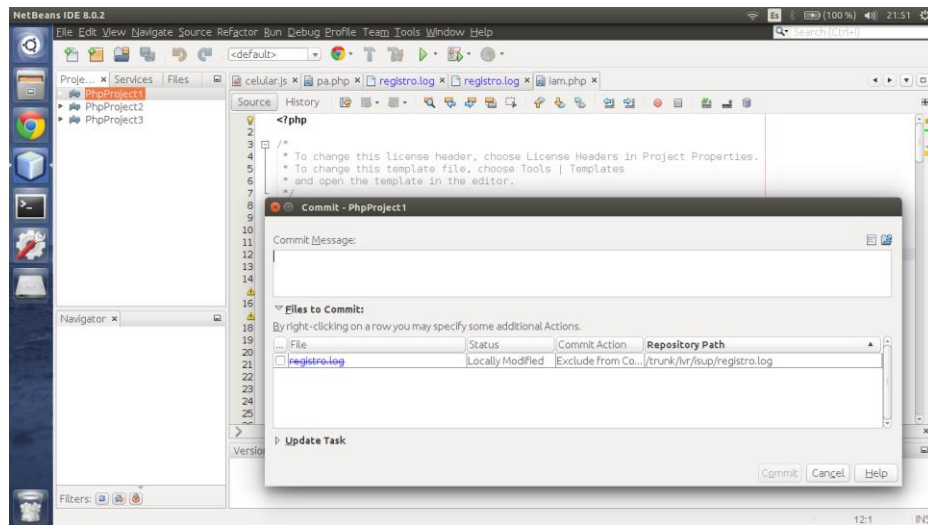


Figura 3.10 Actualización en desarrollo

### 3.4 Diseño general

Dentro del IVR, aquellos mensajes que ingresan y se deben programar para ser procesados son los correspondientes a IAM, REL y RLC, para validar los casos de finalización por ambas partes (MSC o el propio IVR).

#### 3.4.1 Archivos de utilidades

Cada mensaje se programa en lenguaje PHP en el IDE NetBeans ya instalado, con su respectiva cabecera y funciones especiales. El proyecto posee una carpeta especial de archivos de “utilidades”, los cuales contienen las sentencias con los archivos a continuación:

**“conversiones.php”**: Archivo creado para realizar conversiones de números, con funciones para convertir cadenas binarias o decimales a hexadecimales. Se basan en la función *base\_convert*.

```
function cadena_binaria_a_hexadecimal($bits)
{
    return base_convert($bits, 2, 16);
}

function cadena_decimal_a_hexadecimal($decimal)
{
    return base_convert($decimal, 10, 16);
}
```

Figura 3.11 Funciones de conversión

**“enviar\_mensaje.php”**: Este archivo controla el envío de mensajes nuevos desde el IVR y sus respectivos acuses de recibo, declarando la función en sí, y el modo de registrar dichos envíos en el archivo de registro creado por el programa.

```
function enviar_mensaje($url, $fields)
{
    $fields_string = "";
    //url-ify the data for the POST
    foreach($fields as $key=>$value) { $fields_string .= $key.'='.$value.'&'; }
    rtrim($fields_string, '&');

    //open connection
    $ch = curl_init();

    //set the url, number of POST vars, POST data
    curl_setopt($ch,CURLOPT_URL, $url);
    curl_setopt($ch,CURLOPT_POST, count($fields));
    curl_setopt($ch,CURLOPT_POSTFIELDS, $fields_string);
    curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);
    //execute post
    $result = curl_exec($ch);

    //close connection
    curl_close($ch);
    return $result;
}
```

Figura 3.12 Ejemplo de función de envío de mensaje

**“ips.php”**: Este archivo captura las direcciones IP (Protocolo de Internet) de cada máquina al momento de necesitarse su declaración.

```
function get_parametro_ip_ivr_isup() {
    return "192.168.0.113";
}
```

Figura 3.13 Obtención de una IP a través de una función.

**“parametros.php”**: Captura los parámetros de los mensajes alojados en la base de datos del proyecto.

```
function get_parametro_msc_id()
{
    $conexion = mysql_connect("localhost", "root");
    if(!$conexion)
    {
        die("No se puede conectar a la base de datos");
    }
    mysql_select_db("vlr");
    $resultado = mysql_query("select * from parametros where nombre = 'msc_id'");
    echo mysql_error();
    $fila = mysql_fetch_assoc($resultado);
    /*
    if ($fila)
    {
        echo "Hola mundo, soy la msc ". $fila["valor"];
    }
    */

    $msc_id = $fila["valor"];
    mysql_close();
    return $msc_id;
}
```

Figura 3.14 Código de función utilitaria Parámetros.

**“registrar.php”**: En este archivo se declara el registro de intercambios en un archivo .log creado para el proyecto (llamado propiamente “registro”) y se declaran los registros de todo mensaje que ingrese o salga, a aparecer en el archivo de registro, con su respectiva fecha y hora de ingreso.

```

function registrar_recibido_nuevo($mensaje, $parametros)
{
    registrar_central(getComponente(), "REC-NUE", $mensaje, $parametros);
    registrar_completo("REC NUE : ");
    registrar(presentar_array(array("mensaje" => $mensaje, "parametros" => $parametros)));
}

function registrar_envio_nuevo($destinatario, $ip, $mensaje, $parametros)
{
    registrar_central(getComponente(), "ENV-NUE", $mensaje, $parametros);
    registrar_completo("ENV NUE : ");
    registrar(presentar_array(array("destinatario"=>$destinatario, "ip" => $ip, "mensaje" => $mensaje,

```

Figura 3.15 Funciones de registro de mensajes.

Esta carpeta de utilidades debe estar dentro de la carpeta del proyecto principal, y cada archivo debe poderse declarar desde cualquier mensaje programado en el proyecto principal a través de la sentencia *“include”*.

Ciertas funciones, como conversión a dígito, se han incluido en un archivo por separado denominado *“funciones”*, estas también se deben declarar a través de *“include”*.

### 3.4.2 Programación de mensajes de señalización

**IAM:** El IAM es recibido desde la MSC (Centro de Conmutación Móvil por sus siglas en inglés) como una trama hexadecimal con su respectivo CIC y números de llamante y llamado. Se registra la trama y se convierte la trama a binaria para realizar la sustracción de los parámetros de interés para la transacción (se utiliza la función *substr*). Luego de extraer cada dígito de los dos números involucrados, estos se concatenan en una nueva variable a ser enviada a través de nuevos mensajes declarados para la parte de INAP.

```

$iam_process = enviar_nuevo("ivr/inap", get_parametro_ip_ivr_inap(), "iam", array("cic"=> cadena_binaria
$conexion = mysql_connect("localhost", "root") or die("No se puede conectar a la base de datos");
mysql_select_db("ivr_isup");
mysql_query("delete from estado_llamada where remitente = '$calling_party_number'"); // por si acaso hay
mysql_query("insert into estado_llamada (remitente, estado, cadena_dtmf) values ('$calling_party_number'

```

Figura 3.16 Código del mensaje IAM (Envío a PP por INAP).

Los mensajes ACM y ANM son enviados a MSC con la sentencia *enviar\_nuevo* declarada en *enviar\_mensaje.php*. Se crea un lazo *for* donde se declara la dirección URL que direcciona a los archivos de audio alojados en IVR, donde se involucran la IP de la máquina y el nombre de la pista recibido desde Prepago a través de INAP.

```

$acm_process = enviar_nuevo("msc/isup", get_parametro_ip_msc_isup(), "acm", array("trama_acm" => "00001
//////////////////////////////////////////////////INTERFAZ//////////////////////////////////////
$SS7_ack=registrarSS7("IVR (ISUP)", "MSC (ISUP)", "ACM", array("trama_acm" => "000015060${cicHexadecimal}
//////////////////////////////////////////////////INTERFAZ//////////////////////////////////////
$anm_process = enviar_nuevo("msc/isup", get_parametro_ip_msc_isup(), "anm", array("trama_anm" => "00090
//////////////////////////////////////////////////INTERFAZ//////////////////////////////////////
$SS7_ack=registrarSS7("IVR (ISUP)", "MSC (ISUP)", "ANM", array("trama_anm" => "00090${cicHexadecimal}10
//////////////////////////////////////////////////INTERFAZ//////////////////////////////////////

```

Figura 3.17 Código del mensaje IAM (Envío de ACM a MSC)

**Timeout.php:** Se define como *timeout* al tiempo transcurrido tras la reproducción del audio en el que el IVR debe cerrar la llamada automáticamente. Cuando se recibe el *timeout* de parte de MSC significa que el audio terminó de reproducirse, y se debe devolver un mensaje REL para culminar la llamada. Se programa también el recibimiento de un RLC que confirme la liberación de la llamada.

```

registrar_recibido_nuevo("timeout", $_POST);

$calling_party_number = $_POST["calling_party_number"];
enviar_nuevo("msc/isup", get_parametro_ip_msc_isup(), "rel", array("calling_party_number"=>$calling_party_number));
////////////////////////////////////// INTERFAZ ////////////////////////////////////////
$SS7_ack=registrarSS7("IVR (ISUP)", "MSC (ISUP)", "REL", array("OPC"=>"2","DPC"=>"1","TYPE_OF_MESSAGE"=>"REL"));
$SS7_ack=registrarSS7("MSC (ISUP)", "IVR (ISUP)", "RLC ", array("OPC"=>"1","DPC"=>"2","TYPE_OF_MESSAGE"=>"REL"));
$SS7_ack=registrarSS7("MSC (ISUP)", "MS", "HUNG UP", array("0"=>"RELEASE"));
////////////////////////////////////// INTERFAZ ////////////////////////////////////////
enviar_respuesta(array("REL"));

```

Figura 3.18 Envío de REL para colgado de llamada.

### 3.4.3 Ingreso de teclado

**DTMF.php:** En este archivo se crea un diagrama de estados a través de la sentencia *switch*, donde se declaran los casos para el ingreso de teclas 1 o 2. INAP procesa este ingreso para solicitar un servicio a Prepago. De no cumplirse ninguno de los dos casos, el audio debe reproducirse nuevamente por lo que se declara el lazo de las pistas similar al del mensaje IAM. De llegarse al estado de liberación de la llamada (REL), se envía el mensaje, o de lo contrario se vuelve a solicitar el ingreso de teclas.

```

$estado = $fila["estado"];
$cadenaDTMF = $fila["cadena_dtmf"];
$pistas = null;
//$rel = false;
switch($estado)
{
    case "282":
        if ($tecla==1)
        {
            $pistas= enviar_nuevo("ivr/inap", get_parametro_ip_ivr_inap(), "dtmf1",array("tecla"=>$tecla));
            $estado = "rel";
            //$estado = "282_1";
        }
        else if($tecla == 2)
        {
            $pistas= enviar_nuevo("ivr/inap", get_parametro_ip_ivr_inap(), "dtmf2",array("tecla"=>$tecla));
            $estado = "282_2";
        }
        else
        {
            // enviar_respuesta(array("reproducir otra vez url 1,2,3,14"));
            $pistas = array(102, 103, 2);
        }
        break;
}

```

Figura 3.19 Código de los DTMF

### 3.5 Simulación

La máquina utilizada fue asignada con la dirección IP 192.168.0.113 para conexión alámbrica al enrutador, y 192.168.2.113 para la conexión inalámbrica que le permite a las centrales reproducir audios al llegar al equipo necesario. El sistema completo utiliza dos centrales, una para la ciudad de Quito y otra para Guayaquil, cada una posee un duplicado de IVR para reproducir audios correctamente al realizarse llamadas al \*282 desde cualquier ciudad. El código desarrollado permite al sistema procesar el mensaje de inicialización de la transacción (IAM) y devolver los correspondientes mensajes de confirmación (ACM y ANM) y las direcciones de la red para acceder a los archivos de audio a ser reproducidos por el simulador. IVR contiene alojado en su sistema de archivos los audios correspondientes al servicio de consulta de saldos y recargas.

Esta simulación será mostrada de una mejor manera a través de una interfaz gráfica amigable al usuario, la cual es fácil de manejar y consiste en mostrar la señalización completa de todos los protocolos que se encuentren trabajando en el intercambio de mensaje. Eso incluye los correspondientes a ISUP ya descritos a lo largo del presente documento.



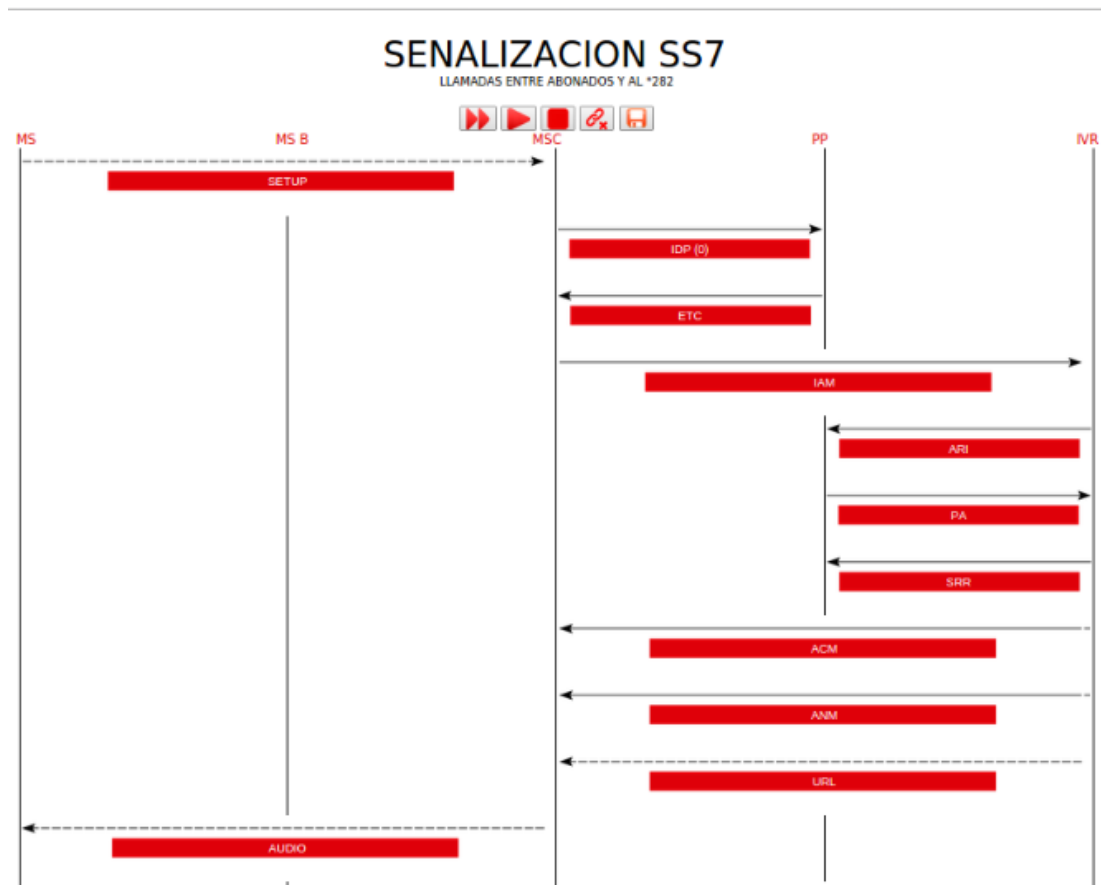


Figura 3.20 Simulador de señalización SS7

Este es muy útil para detectar errores en el momento de realizar pruebas. IVR emplea intercambio ISUP para devolver las direcciones correspondientes a los audios a reproducir, y se encarga de terminar la llamada de no receptarse ingreso alguno del teclado. Como se puede apreciar en la figura previa, Se muestra el envío de las respectivas direcciones de red para reproducir los audios alojados en el IVR, junto con los mensajes de señalización de la simulación.

Entre las situaciones a simular, se muestran las situaciones en que:

- El abonado llama al \*282, pero no realiza acción alguna, por lo que transcurrido un tiempo, la llamada es colgada por el IVR.
- El abonado decide terminar la llamada, por lo que MSC manda a colgar la llamada.
- El abonado consulta su saldo al sistema (Opción 1)
- El abonado realiza una recarga con una tarjeta (Opción 2)
- El abonado utiliza un código de tarjeta no válido
- El abonado utiliza un código de tarjeta ya utilizado.

Los códigos de tarjeta son manejados dentro de la base de datos de la máquina de prepago. IVR únicamente maneja la base de datos de los estados sobre los cuales se manejan las opciones disponibles del servicio, codificadas junto a la validación del marcado por tono.

### **3.6 Posibles aplicaciones**

El simulador está diseñado para mostrar a nuevos usuarios, quienes apenas inicien estudios de redes celulares, como opera un sistema de señalización SS7, ya sea durante seminarios o cursos de postgrado donde se especialicen en redes de telefonía móvil. Estos cursos usualmente tratan bastante acerca de todos los componentes que son parte de la red de señalización, detallando los mensajes y parámetros de cada uno de sus protocolos, por lo que podrían

requerir de un simulador de ellos como software de apoyo. El mismo podría también ser aprovechado por los propios estudiantes para ellos mismos explorar mejor el sistema y sacarle provecho a dichos conocimientos para posibles proyectos académicos o personales [19]

Otra posible aplicación en el campo profesional requeriría de este simulador para hacer pruebas virtuales de un sistema antes de su debida instalación física, ya sea por una operadora que recién inicia negocios o una nueva red implementada por una operadora en ejercicio.

## **CAPÍTULO 4**

### **4. ANÁLISIS DE RESULTADOS**

#### **4.1 Descripción general**

El IVR virtual, con el objetivo de pasar una conexión de voz, es capaz de comunicarse con su respectivo Centro de Conmutación Móvil intercambiado mensajes correspondientes al protocolo de ISUP. El equipo es capaz de conectarse a una central y podrá direccionar los audios alojados en el mismo, correspondientes al servicio de consulta y recarga de saldo a través de un *voucher*, el cual se accede al marcar el número \*282.

De no poseer directamente una máquina con Linux, es posible implementar el proyecto desde una máquina virtual a través de software virtualizador como Virtual Box o VMWare, siguiendo los pasos descritos en el capítulo anterior, sin embargo no es recomendable ya que podría haber problemas al capturar las especificaciones de red de dicha máquina, además de que Ubuntu propiamente podría tener problemas con permisos de los archivos dentro de la máquina virtual sino se tiene un gran dominio de Linux.

El presente capítulo evaluará los resultados y su pertinente interpretación, y se comentará en posibles errores que podrían ocurrir en la implementación física. Empezando por la topología, donde las direcciones IP de los equipos involucrados deben ser correctamente asignadas.

## **4.2 Interpretación de resultados**

### **4.2.1 Registro de transacción**

Dentro de Ubuntu, se abre una ventana de terminal, desde donde es posible ejecutar un comando de revisión del archivo de registro (.log) creado en el proyecto. En el archivo se visualiza paso a paso el intercambio de mensajes con los otros componentes de la red., a mostrarse paso a paso en el terminal.

```
tail -f /opt/lampp/htdocs/ivr/isup/registro.log
```

El registro que se ha programado en PHP debe indicar si el mensaje se recepta o se envía, usando los indicadores “ENV NUE (Envío nuevo)”, “REC NUE (Recibido nuevo)”, y el envío y recepción de acuses de respuesta. También se

indica en los mensajes su destinatario, sus parámetros y se muestra la IP de la máquina involucrada.

```

2015/02/08 22:43:58.06148500 - INFO      :calling party category 00001010
2015/02/08 22:43:58.06153200 - INFO      :SCCP Indicator 0
2015/02/08 22:43:58.06158200 - INFO      :Originating access indicator 00
2015/02/08 22:43:58.06162900 - INFO      :Call Type 0
2015/02/08 22:43:58.06167900 - INFO      :Message Type 00000001
2015/02/08 22:43:58.06172500 - INFO      :CIC 000000000101
2015/02/08 22:43:58.06177600 - INFO      :SLS 0001
2015/02/08 22:43:58.06183300 - INFO      :OPC 00000000000001
2015/02/08 22:43:58.06191600 - INFO      :DPC 00000000000001

2015/02/08 22:43:58.06419700 - ENV NUE :
'destinatario' : 'ivr/inap',
'ip' : '192.168.0.113',
'mensaje' : 'iam',
'parametros' :
(
  'cic' : '5',
  'calling_party_number' : '0988272881',
  'called_party_number' : '*282',
)
2015/02/08 22:43:58.21804900 - REC RES :
'ack' :
(
  0 : '101',
  1 : '102',
  2 : '103',
  3 : '2',
)

```

Figura 4.1 Prueba de envío de mensaje a la parte de INAP, ambos protocolos trabajan en el mismo equipo

Dentro del registro es posible reportar los errores ocurrientes en el intercambio de mensajes, como algún parámetro no mostrado, o que se haya enviado de manera errónea.

```

2015/02/08 22:43:58.27966600 - ENV NUE :
'destinatario' : 'msc/isup',
'ip' : '192.168.0.4',
'mensaje' : 'acm',
'parametros' :
(
  'trama_acm' : '00001506000510004001',
  'cic' : '5',
  'message_type' : '6',
  'opc' : '1',
  'dpc' : '1',
  'sls' : '1',
  'no_charge' : '1',
  'called_party' : '1',
  'subs_ordinario' : '1',
)
2015/02/08 22:43:58.28907600 - REC RES :
'ack' :
(
  0 : 'OK',
)
2015/02/08 22:43:58.29088100 - ENV NUE :
'destinatario' : 'msc/isup',
'ip' : '192.168.0.4',
'mensaje' : 'ann',
'parametros' :
(
  'trama_ann' : '0009000510004001',
  'cic' : '5',
  'message_type' : '9',
  'opc' : '1',
  'dpc' : '1',
  'sls' : '1',
)
2015/02/08 22:43:58.30009900 - REC RES :
'ack' :
(
  0 : 'OK',
)

```

Figura 4.2 Prueba de reserva de circuito, mostrando mensajes ACN y ANM.

En el registro se muestra completa la dirección apuntada hacia los archivos de audio necesarios para la transacción, mostrando su respectiva dirección en la red, que al mismo tiempo el equipo correspondiente al celular debe recibir para reproducirlo.

#### **4.2.2 Interfaz de señalización**

La interfaz ha sido diseñada para mostrar la señalización que cursa entre las partes presentes del sistema. La simulación en cualquiera de los escenarios inicia con la marcación del \*282 desde la máquina del celular, desde tras solicitarse un ingreso, el usuario marca ya sea la opción 1 o 2, dependiendo de qué es lo que se desee hacer. Tras escucharse el respectivo audio, e ingresarse por teclado la opción correspondiente, al usuario se solicita digite el código de tarjeta seguido del signo # para el caso de opción 2. En cada caso la simulación de la señalización será mostrada en la interfaz.

Estas se muestran con una flecha de línea recta continua apuntando a la dirección del mensaje, junto con un recuadro que muestra el nombre del mensaje respectivo. Toda la señalización correspondiente al protocolo ISUP se encuentra presente en la interfaz, incluyendo el envío y la recepción del marcado desde el teclado y el direccionamiento al audio del equipo.



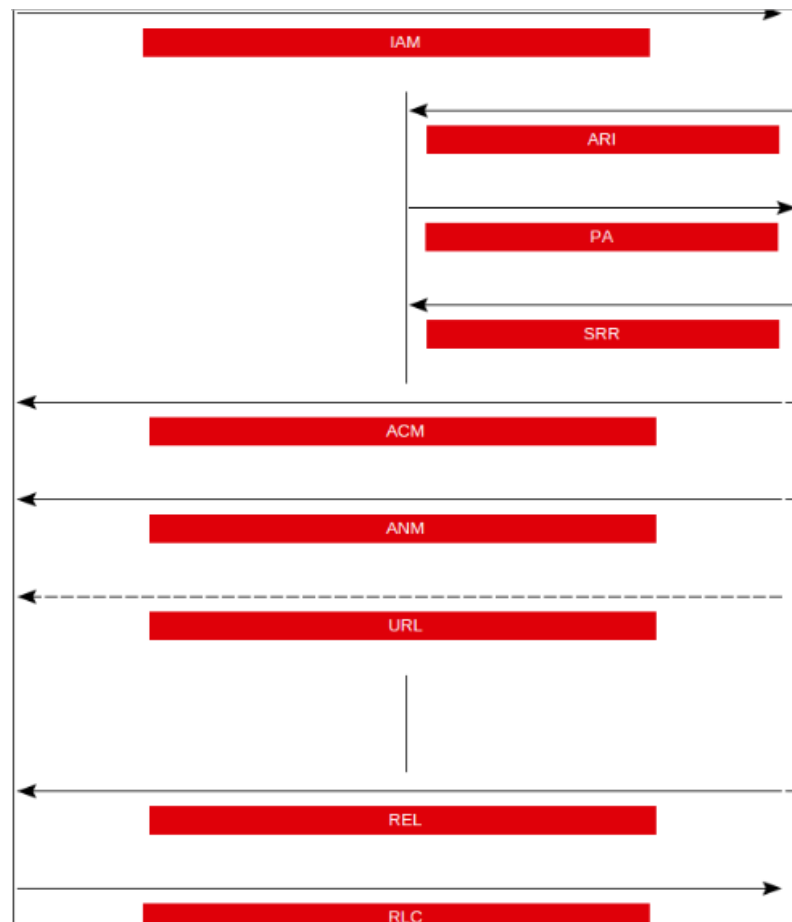


Figura 4.3 Señalización de ISUP en la interfaz de la aplicación web.

### 4.2.3 Interfaz de mensajes

Si se opta por dar un clic sobre el nombre de cada uno de los mensajes, un recuadro pequeño se muestra con los parámetros del mensaje. Para ISUP, lo que se mostrará es la respectiva trama hexadecimal para los mensajes de IAM, ACM y ANM, los cuales contienen información tal como el Identificador de Circuito (CIC), códigos de punto de origen y destino (OPC y DPC), y en el caso del IAM, los números de llamante y llamado (\*282 en estos casos).



Figura 4.4 Interfaz del mensaje IAM con su respectiva trama

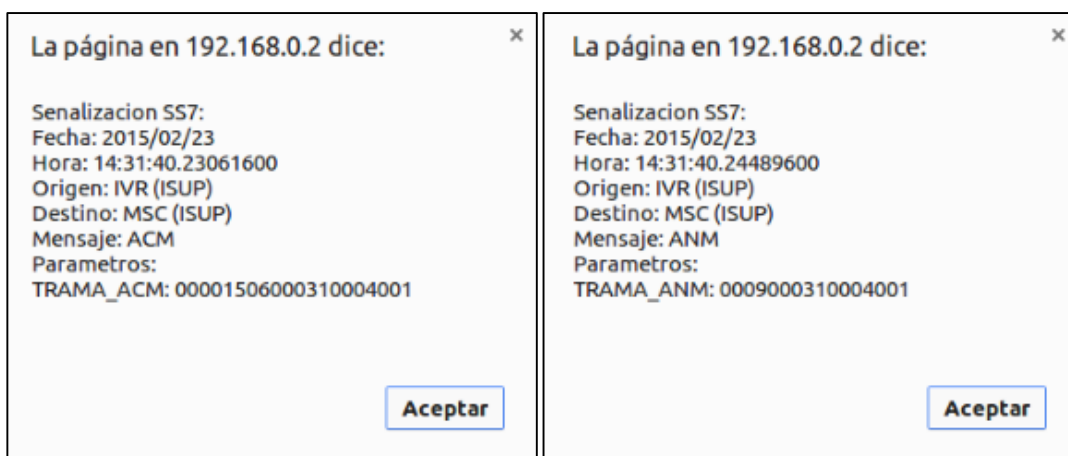


Figura 4.5 Interfaces de los mensajes ACM y ANM

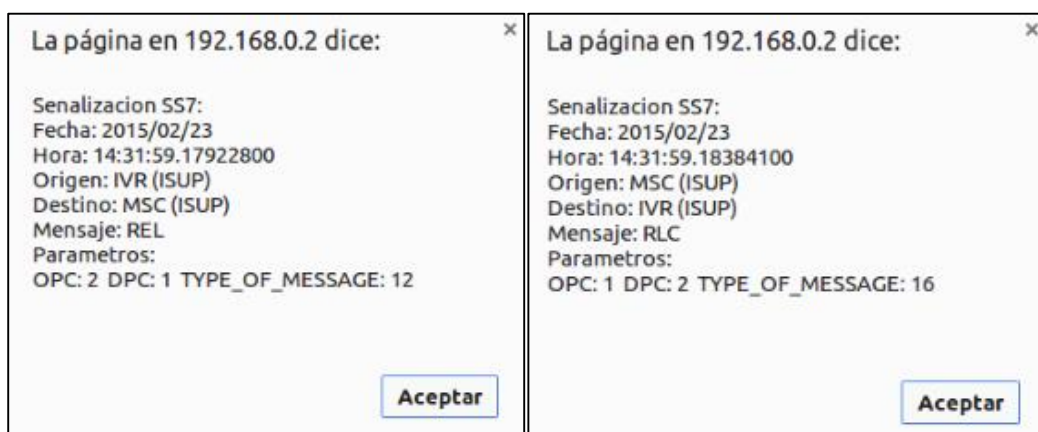


Figura 4.6 Interfaces de los mensajes REL y RLC

### 4.3 Discusión final

Se deben tener bien claros cuáles son los posibles conflictos a suceder dentro de la simulación de la transacción. Antes de tratar de operar la aplicación web, se considera necesario tener conocimientos acerca de los protocolos involucrados directamente con el IVR, principalmente el protocolo INAP, que se comunica con el sistema de Prepago para solicitar información de su base de datos alojada. No estaría de más mencionar la operación de la aplicación del celular, desde donde se realizan los ingresos por teclado para el IVR y desde donde se accede al mismo, proyectos que van de la mano y por lo tanto se necesita conocimiento de su funcionamiento para mejor comprensión del comportamiento del equipo y la aplicación.

Como se indicó en el apartado anterior, se podría tratar de errores físicos, pero en gran parte de las pruebas se trataban de errores mínimos que suelen pasar desapercibido en gran parte de dichas pruebas. En las primeras pruebas, era necesario que todas las máquinas de la red levanten el servidor Apache con el cual se acceda a la aplicación programada, sin embargo cuando uno o más de los equipos no se encontraban dentro del servidor, se retornaba un mensaje de error al tratar de llegar hacia algún archivo dentro de la máquina involucrada. Otros errores comunes son el de no cambiar correctamente el nombre de un archivo a llamar, o escribirlo incorrectamente, resultando en un error de “archivo no encontrado” al momento de buscar el archivo declarado para su

reproducción en el equipo de celular, también se necesita verificar que este bien escrito el directorio, lo que resultaría en un error similar.

Finalmente, y algo que debería verificarse cada vez que se ingresan nuevos archivos al directorio del proyecto, es su respectiva asignación de permisos.

Desde Linux, se opera a través de súper usuario (*root*) para dar privilegios al directorio del proyecto.

## **CONCLUSIONES**

El proyecto concluye con varios aspectos de mucho interés para personas que tengan algún dominio del tema, donde se describe el funcionamiento e información extensa del modo de operar de una red móvil, de donde se puede obtener material que ayude a quienes deseen realizar un estudio del tema a una mejor comprensión del mismo:

1. La documentación a concluir contiene un extensivo análisis de la operación de un sistema IVR, junto con el cual se explica con detalle en forma de manual el desarrollo de una aplicación que simule su comportamiento utilizando mensajes del protocolo ISUP, utilizando las pertinentes herramientas informáticas como lo son el lenguaje PHP y el Sistema Operativo Linux, donde se ha construido un ambiente de trabajo amigable

para pasar al desarrollo de la aplicación, la cual se detallan paso a paso en el presente documento.

2. Aunque en el presente proyecto se utilizó con la finalidad de comunicarse con un sistema virtualizado, por lo que únicamente pasan archivos de audio por el circuito de voz, ISUP opera de modo correcto para establecer la conexión de voz con un IVR, el cual brinda los servicios de modo eficaz y satisfactorio.
3. El sistema IVR se implementó con éxito dentro de una red telefónica virtual, funcionando bajo los protocolos establecidos. Este se ha conectado a una red correspondiente al servicio de una ciudad (Quito o Guayaquil) y utiliza mensajes de ISUP para comunicarse con su respectiva Central Móvil, desde donde llegan las solicitudes del usuario. Este IVR brinda los servicios de consulta de saldos y recarga de los mismos.
4. El sistema emplea todos los mensajes de señalización acordes a la estandarización ISUP, en todos los casos pertinentes se envían y reciben los mensajes correspondientes a cada caso estudiado, los cuales también son mostrados e identificados en la interfaz. Este protocolo es capaz de establecer con éxito cualquier tipo de llamada siendo de vital importancia para cualquier red móvil de telefonía.

## RECOMENDACIONES

Las recomendaciones a puntualizar se enfocan en la correcta instalación del equipo junto con los demás componentes del sistema completo, y a ciertas observaciones de diseño:

1. Se recomienda tener un conocimiento intermedio de Linux para su correcto manejo, principalmente comandos de asignación de permisos de directorios, instalaciones de software y ejecutables, en especial en la distribución de Ubuntu, sobre donde se ha trabajado el proyecto.
2. Tener un conocimiento avanzado del lenguaje PHP para manipular sin problemas los ficheros de cada mensaje, poder detectar rápidamente algún conflicto que se genere con algún archivo y resolverlo de igual manera.

3. Al trabajar dentro de la red correspondiente al sistema, siempre descargar cambios realizados de archivos relacionados al IVR a través de la herramienta *Update* de la subversión en NetBeans, y también subir los cambios hechos en el mismo equipo para la disponibilidad de los otros equipos a través de un *Commit*.
4. No solo limitarse a los respaldos en la nube hechos al repositorio, también es altamente recomendable realizar más respaldos del código fuente de todos los mensajes y utilidades ya sea en alguna memoria física o bien en otros almacenamientos en nube, dese el caso de que se pase por alto algún respaldo en el repositorio, o si los cambios realizados para una prueba no resulten efectivos y se altere el funcionamiento del sistema que haya sido funcional en versiones anteriores.
5. Tener sumo cuidado con la conectividad en la topología. Primero realizando una prueba física, viendo si todos los enrutadores o switches se encuentran encendidos y funcionales, y las IP estén asignadas correctamente. Luego ejecutando pruebas de conectividad a través de un *ping* en el terminal de comandos.
6. Conocer a fondo la operación de cada mensaje de ISUP como IAM, REL, RLC, ACM y ANM, cada uno sus respectivas tramas, se recomienda estudiar las recomendaciones de la ITU acerca de este protocolo y sus mensajes.



## BIBLIOGRAFÍA

- [1] Simulation of SS7 Common Channel Signaling, IEEE Communications Magazine, «<http://dl.acm.org/citation.cfm?id=2294268>», fecha de consulta Enero 2015.
- [2] ¿Qué es el Software Libre? Proyecto GNU. Free Software Foundation. «<https://www.gnu.org/philosophy/free-sw.es.html>», fecha de consulta Febrero 2015.
- [3] OpenSS7 Project. Status. «<http://www.openss7.org/status.html>», fecha de consulta Febrero 2015.
- [4] Clase de Signaling System No. 7. Reseña Histórica. EVA. «[https://eva.fing.edu.uy/pluginfile.php/67113/mod\\_resource/content/2/SS7\\_2014%20%28presentacion%29.pdf](https://eva.fing.edu.uy/pluginfile.php/67113/mod_resource/content/2/SS7_2014%20%28presentacion%29.pdf)», fecha de consulta Diciembre 2014.
- [5] Análisis y entorno de las Telecomunicaciones parte 2: Historia Tendencias y Estadísticas IVR, «<http://edgarhernandez317.blogspot.com/p/historia-ivr-lasivrs-comenzaron-ser.html>», fecha de consulta Diciembre 2014.
- [6] Inform IT. Signaling System No. 7. Chapter 8. ISDN User Part (ISUP), «[http://www.informit.com/library/content.aspx?b=Signaling\\_System\\_No\\_7&seqNum=57](http://www.informit.com/library/content.aspx?b=Signaling_System_No_7&seqNum=57)», fecha de consulta Diciembre 2014.

- [7] Performance Technologies. Tutorial on Signaling System 7 (SS7), «[http://web.uct.ac.za/depts/commnetwork/eee5026/note/eee526-06-250ss7\\_tutorial\\_091503v2.pdf](http://web.uct.ac.za/depts/commnetwork/eee5026/note/eee526-06-250ss7_tutorial_091503v2.pdf)», fecha de consulta Diciembre 2014.
- [8] Message Transfer Part Level 3. The Wireshark Wiki, «<http://wiki.wireshark.org/MTP3>», fecha de consulta Diciembre 2014.
- [9] Pradip Bhuyan's SS7 Tutorial. The SS7 Protocol Stack, «[http://www.angelfire.com/indie/pradip\\_bhuyan/ss7/SS7.html](http://www.angelfire.com/indie/pradip_bhuyan/ss7/SS7.html)», fecha de consulta Diciembre 2014.
- [10] SS7 Overview. «<http://www.techfest.com/networking/wan/ss7.htm>», fecha de consulta Diciembre 2014.
- [11] SS7 Protocol Suite. «<http://www.protocols.com/pbook/ss7.htm>», fecha de consulta Diciembre 2014.
- [12] Interwork Connected Number between SIP and SS7, «[https://www.dialogic.com/webhelp/BorderNet2020/1.1.0/WebHelp/iw\\_connected\\_number\\_sip\\_ss7.htm](https://www.dialogic.com/webhelp/BorderNet2020/1.1.0/WebHelp/iw_connected_number_sip_ss7.htm)», fecha de consulta Diciembre 2014.
- [13] Inform IT. ISUP Message Format, «[http://www.informit.com/library/content.aspx?b=Signaling\\_System\\_No\\_7&seqNum=66](http://www.informit.com/library/content.aspx?b=Signaling_System_No_7&seqNum=66)», fecha de consulta Diciembre 2014.
- [14] What is IVR? Interactive Voice Response. «<http://www.aspect.com/glossary/what-is-ivr-interactive-voice-response>», fecha de consulta Diciembre 2014.

- [15] DTMF (Dual-Tone Multi-Frequency). «<http://www.mediacollege.com/audio/tone/dtmf.html>», fecha de consulta Enero 2015.
- [16] Todo sobre Linux, Technoblogy, «<http://tecnoblogy.wordpress.com/todo-sobre-linux/>», fecha de consulta Diciembre 2014.
- [17] Apache Friends, XAMPP. «<https://www.apachefriends.org/index.html>», fecha de consulta Enero 2015.
- [18] Java SE Development Kit, Oracle Corporation, «<http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html>», fecha de consulta Enero 2015.
- [19] Curso Introducción al Sistema SS7. Serprotel. «<http://www.serprotel.com.ar/areas/core/122.html>», fecha de consulta Febrero 2015.

# ANEXOS

## Señalización ISUP

A continuación se muestra un flujo de señalización de ISUP. Nótese que mientras en el flujo se muestran dos representaciones para IVR, una para el protocolo ISUP y otra para INAP (debido a que se trabajaron por separado), los dos protocolos son utilizados por el mismo IVR (un solo equipo para IVR).

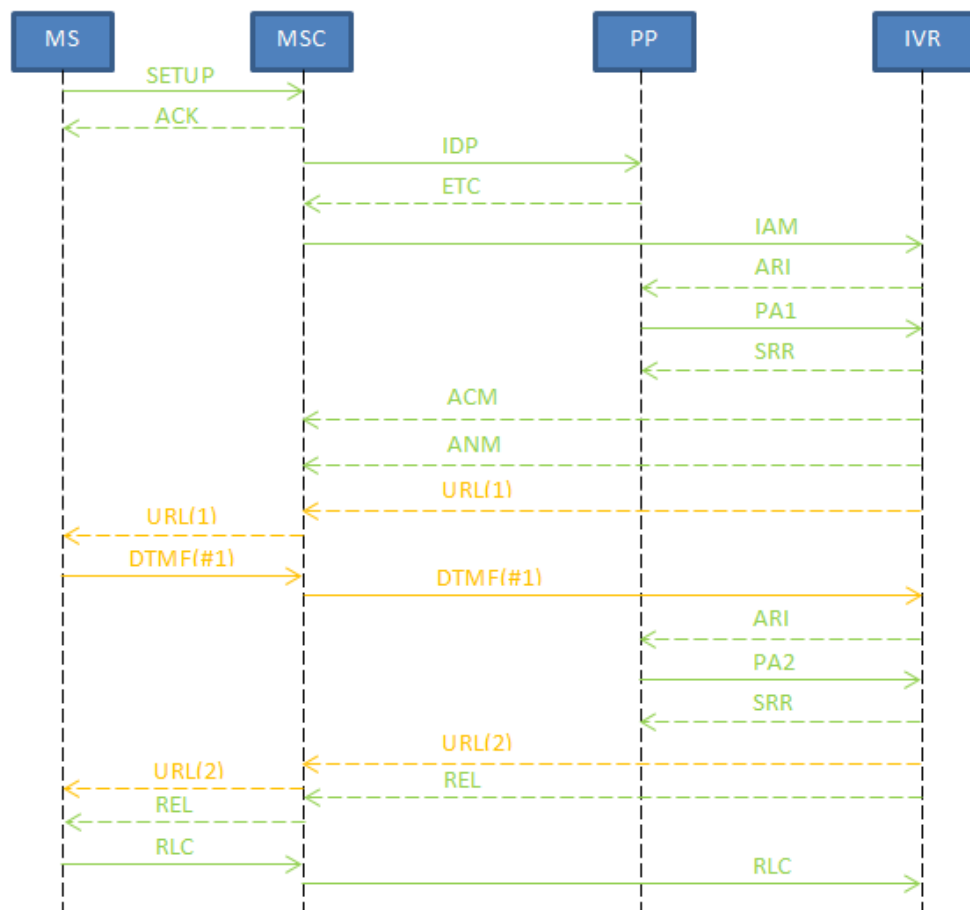


Figura A.1 Flujo para el caso de consulta de saldos (Opción 1).

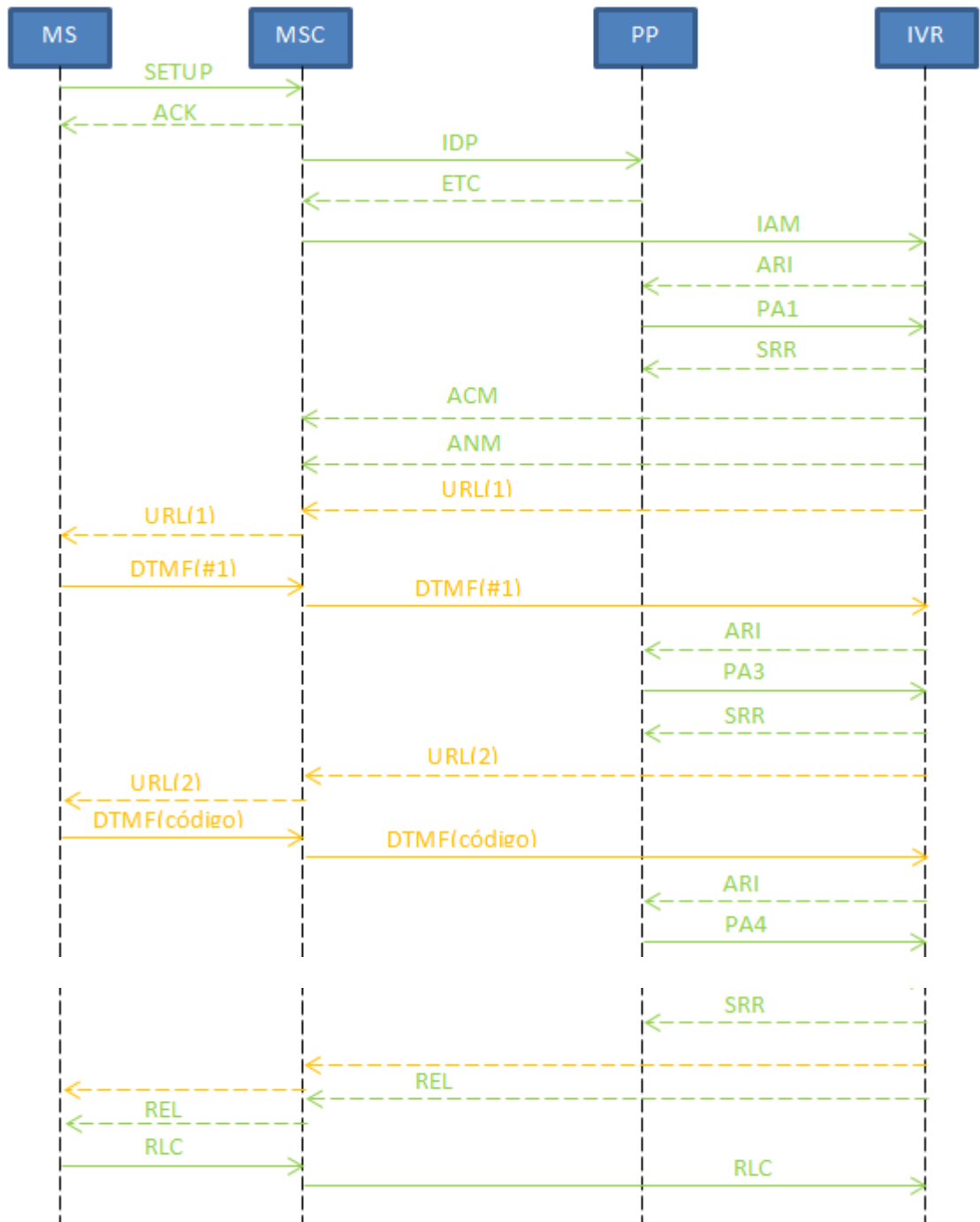


Figura A.2 Flujo para el caso de recargas (Opción 2).