



ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL

Facultad de Ingeniería en Electricidad y Computación

“ANÁLISIS E IMPLEMENTACIÓN DE UN DISPOSITIVO VIRTUAL EN EL LENGUAJE ABIERTO PREPROCESADOR DE HIPERTEXTO (PHP) SOBRE LINUX QUE EMULE UN DISPOSITIVO MÓVIL PARA LA GENERACIÓN DE LLAMADAS PREPAGO, POSTPAGO Y EL ENVIÓ Y RECEPCIÓN DE MENSAJES CORTOS (SMS), CON UNA INTERACCIÓN DIRECTA CON UNA CENTRAL DE CONMUTACIÓN MÓVIL (MSC).”

INFORME DE MATERIA DE GRADUACIÓN

PREVIO A LA OBTENCIÓN DEL TÍTULO DE:

**INGENIERO EN ELECTRÓNICA Y TELECOMUNICACIONES E  
INGENIERO EN TELEMÁTICA**

Presentado por:

**Iván Alberto Cedeño Cáceres**

**Rory David Gavilanes Rodríguez**

**GUAYAQUIL – ECUADOR**

**2015**

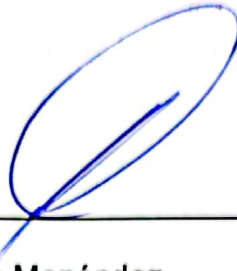
## **AGRADECIMIENTO**

Alcanzar la culminación de este proyecto ha sido gracias al apoyo y confianza de muchas personas que nos han ayudado en todo el proceso de este trabajo, especialmente a nuestros padres que han sido nuestro soporte principal y nos han brindado toda su comprensión en este importante paso de nuestras vidas.

## **DEDICATORIA**

Este logro se lo dedicamos principalmente a nuestros padres, hermanas, sobrinos, y amigos cercanos, ya que han sido nuestro motor fundamental en el transcurso de este proceso y nos han brindado su confianza plena en esta etapa de nuestras vidas.

# TRIBUNAL DE SUSTENTACIÓN



Msc. José Menéndez

PROFESOR DE LA MATERIA DE GRADUACIÓN



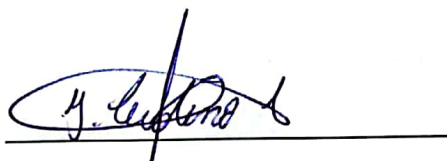
Ing. Marcia Garcés

PROFESOR DELEGADO POR LA UNIDAD ACADÉMICA

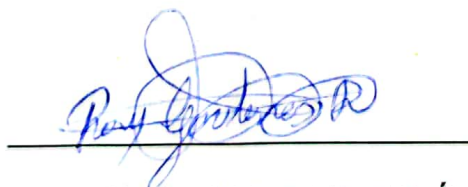
## DECLARACIÓN EXPRESA

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

(Reglamento de Graduación de la ESPOL)

A handwritten signature in black ink, appearing to read 'I. Cedeño', is written over a horizontal line.

IVAN ALBERTO CEDEÑO CÁCERES

A handwritten signature in blue ink, appearing to read 'Rory Gavilanes R', is written over a horizontal line.

RORY DAVID GAVILANES RODRÍGUEZ

# ÍNDICE GENERAL

AGRADECIMIENTO .....	ii
DEDICATORIA .....	iii
TRIBUNAL DE SUSTENTACIÓN .....	iv
DECLARACIÓN EXPRESA .....	v
ÍNDICE GENERAL.....	vi
ÍNDICE DE FIGURAS.....	ix
GLOSARIO .....	xi
INTRODUCCIÓN .....	xiii
CAPÍTULO 1 .....	1
1 ANTECEDENTES Y JUSTIFICACIÓN.....	1
1.1 Antecedentes de simulador de terminal móvil sobre linux. ....	1
1.2 Justificación de Aplicación para SS7 .....	3
1.3 Descripción del Proyecto .....	4
1.3.1 Objetivos Generales.....	5
1.3.2 Objetivos Específicos .....	5
1.4 Alcance .....	5
1.5 Limitaciones.....	6
1.6 Metodología .....	7
CAPÍTULO 2.....	9

2	FUNDAMENTOS TEÓRICOS .....	9
2.1	La Señalización SS7 en una Red Celular .....	9
2.2	Arquitectura de la Red SS7 .....	10
2.3	Capas del Protocolo SS7.....	13
2.4	GSM.....	17
2.5	Short Message Service (SMS).....	29
2.6	Software Netbeans, XAMPP y Google Chrome.....	30
	CAPÍTULO 3.....	32
3	Desarrollo del Proyecto .....	32
3.1	Requerimientos del Proyecto.....	33
3.1.1	Topología de Red.....	34
3.2	Flujo de Mensajes.....	35
3.2.1	Flujo Inicial .....	35
3.2.2	Actualización de Localización .....	39
3.2.3	Llamada Celular .....	43
3.3	Software.....	45
3.3.1	LINUX como Sistema Operativo .....	46
3.3.2	Preprocesador de Hipertexto (PHP) para su Implementación... ..	47
3.4	Implementación del Proyecto.....	48

3.5	Aplicaciones.....	57
3.6	Resumen del Proceso de Implementación del Proyecto.....	57
CAPÍTULO 4.....		60
4	Análisis de Resultados .....	60
4.1	Resultados Obtenidos.....	61
4.2	Interpretación de resultados .....	68
CONCLUSIONES .....		72
RECOMENDACIONES.....		74
BIBLIOGRAFIA.....		76



## ÍNDICE DE FIGURAS

Figura 2.1 Arquitectura de señalización SS7 .....	11
Figura 2.2 Protocolo SS7 vs. Modelo OSI .....	14
Figura 2.3 Red GSM .....	18
Figura 2.4 División de Canales Lógicos .....	23
Figura 2.5 Células en diferentes áreas .....	26
Figura 2.6 Protocolos en Red GSM .....	28
Figura 3.1 Topología de red.....	34
Figura 3.2 Mensaje Señalización Inicialización.....	36
Figura 3.3 Señalización de actualización de localización .....	40
Figura 3.4 Mensaje Señalización Llamada Celular .....	43
Figura 3.5 Código HTML en NETBEANS .....	50
Figura 3.6 Código CSS .....	51
Figura 3.7 Variables usadas en el código HTML .....	51
Figura 3.8 Función para registrar el flujo en código java .....	52
Figura 3.9 Estados de cambios iniciales del celular.....	53
Figura 3.10 Función para iniciar el flujo de mensajes .....	53
Figura 3.11 Función para el traspaso de los números a escoger .....	54
Figura 3.12 Función pedido de actualización de localización .....	55
Figura 3.13 Función para mostrar información .....	56
Figura 3.14 Código de la interfaz del celular.....	56

Figura 4.1 Interfaz gráfica del celular (apagado).....	61
Figura 4.2 Celular encendido (Seleccione número).....	62
Figura 4.3 Operadora Celular .....	63
Figura 4.4 Información de SIM y otros parámetros (#A) .....	64
Figura 4.5 Barra de estado (actualizando).....	65
Figura 4.6 Información después de cambio de localización.....	66
Figura 4.7 Flujo de actualización de localización .....	67
Figura 4.8 Parámetros de cada mensaje enviado.....	68
Figura 4.9 mensaje de autenticación .....	69
Figura 4.10 Mensaje de respuesta de autenticación.....	70
Figura 4.11 Mensaje final.....	70

## GLOSARIO

AUC	Authentication Center
CAMEL	Customized Applications for Mobile Network Enhanced Logic
GSM	Global System for Mobile Communication
HLR	Home Location Register
HTTP	Hyper Text Transfer
INAP	Intelligent Network Application Part
ISDN	Integrated Services Digital Network
ISUP	ISDN User Part
LAPD	Link Access Protocol for D-channel
MAP	Mobile Application Part
MTP-1	Message Transfer Part level 1
MTP-2	Message Transfer Part level 2
MTP-3	Message Transfer Part level 3
OSI	Open Systems Interconnection
PSTN	Public Switched Telephone Network
REL	Release message
RLC	Release Call message
SCP	Signaling Control Point
SCCP	Signaling Connection Control Part
SMS	Short Message Service

SMSC	Short Message Service Center
SMSC	Short Message Service Center
SS7	Signaling System #7
SSP	Signaling Service Point
STP	Signal Transport Point
TCAP	Transaction Capabilities Application Part
TDM	Time Division Multiplexing

# INTRODUCCIÓN

Hoy en día las aplicaciones Web son muy populares y necesarias en todos los ámbitos ya que en el mundo digital en el que vivimos la mayoría de personas tienen acceso a internet, y por ende acceden constantemente ya sea por trabajo, estudio o simplemente para realizar otras actividades.

Debido a esto las aplicaciones web son muy rentables [1], y un sin número de personas se dedican a desarrollarlas para empresas de todo tipo, entre las ventajas de estas aplicaciones tenemos:

- Son livianas para la computadora, ya que no requiere mucho espacio en el disco para ser ejecutada.
- No es necesario su actualización ya que esto se lo hace de lado del servidor.
- Tiene una gran compatibilidad entre plataformas dado que su funcionamiento es en un navegador web.

# CAPÍTULO 1

## 1 ANTECEDENTES Y JUSTIFICACIÓN

### 1.1 Antecedentes de simulador de terminal móvil sobre Linux.

En 1983 Richard Stallman fundó el proyecto GNU, para crear sistemas operativos parecidos a UNIX. Dos años más tarde creó la "Fundación del Software Libre" y escribió la GNU *General Public License* para posibilitar el software libre en el sistema de copyright. Dicho GNU experimentó en esa época un auge muy rápido, que dio como resultado que una variedad de programas fueran escritos a principios de 1990, pero como sistema operativo aún le faltaba el Kernel y aunque era constantemente desarrollado, no disponía de ningún buen Kernel basado en UNIX, por el contrario era un número de

proyectos de software libres que podían ser traducidos en las variantes UNIX mediante el compilador de GNU.

A finales de 1990 Linus Torvalds propone hacer un sistema operativo que se comporte exactamente igual al sistema operativo UNIX, pero que funcione sobre cualquier ordenador compatible PC. Con el desarrollo de internet le añade facilidades al trabajo que realizaba y colaboran entre sí, aportando con programas que vienen con UNIX.

El objetivo principal de Linus era crear un núcleo del sistema operativo que fuera compatible con UNIX y que permita ejecutar programas gratuitos y respeten las licencias GNU. Para crear su núcleo, Linus se basó en Minix, una versión reducida de UNIX desarrollada por Andy Tanenbaum para crear código sobre algún sistema operativo.

A finales de 1991 Linus escribió un pequeño núcleo que tenía lo necesario para leer y escribir ficheros en un disquete creando la versión 0.01. Como no era muy agradable de usar y no hacía gran cosa, no lo anunció. Le puso como nombre Linux, que es un acrónimo en inglés de "Linus UNIX" (el UNIX de Linus). La primera versión oficial de Linux es la 0.02, que ya podía ejecutar dos herramientas básicas de GNU: el intérprete de órdenes (bash) y el compilador de C (gcc).

Tras un constante cambio de versiones y actualizaciones de Linux, en 1992 se creó la versión 1.0 que era más proactivo que versiones anteriores y cumplían con los parámetros que se había plantado en un principio, por lo que se puede

trabajar en un entorno Linux para crear algún tipo de aplicación y que sea accesible a cualquier persona de manera gratis.

El concepto de Aplicación Web tiene varios años, el primer lenguaje de programación en que se desarrollaron las aplicaciones web fue PERL, este fue creado por Larry Wall a fines de los años 80, pero fue en el año 1995 que se puso a disposición el lenguaje PHP con el que se realmente el desarrollo de aplicaciones web despuso, entre las principales aplicaciones que se han desarrollado en este lenguaje tenemos Google, Facebook, Wikipedia, entre otros.

En el año 1998 la gigantesca Google desarrollo su aplicación Web de motor de búsqueda, que por su manera de indexar las páginas web facilitaba enormemente la búsqueda de información a través de la web, desde ese momento Google es uno de los principales innovadores de aplicaciones web en el mundo, entre sus ejemplos tenemos, Google Maps, Google Docs, Gmail, Google +, entre otros.

## **1.2 Justificación de Aplicación para SS7**

En la actualidad el mercado de las telecomunicaciones es tan amplio y ofrece nuevos servicios o aplicaciones que son creadas o actualizadas constantemente, por lo que se requiere de una infraestructura adecuada, además de mecanismos o protocolos que contengan información necesaria para lograr con éxito el establecimiento del servicio a utilizarse.



En nuestro proyecto vamos a realizar el proceso de una llamada y el envío de mensajes de textos desde la estación móvil (MS) hasta la central de conmutación móvil (MSC) dentro del sistema operativo de LINUX, por lo nos enfocamos en el flujo señalización SS7, que no solo nos ofrece unas variedades de uso en telefonía celular sino que nos brinda confiabilidad de la entrega de mensajes y logra un uso eficiente de los recursos a utilizarse dentro de un proceso de telefonía celular.

El proceso de una llamada celular o el envío de un mensaje de texto conllevan un flujo constante de mensajes entre diferentes estaciones o bases de datos que forma parte de la arquitectura celular. Estas infraestructuras se comunican entre sí para realizar el servicio requerido, por lo que es necesario el uso del protocolo SS7 para el respectivo flujo de control de la información que se envía de un nodo a otro.

### **1.3 Descripción del Proyecto**

El proyecto se realiza bajo la arquitectura del protocolo SS7, con el que implementaremos la reproducción de llamadas prepago y postpago desde un dispositivo virtual en el lenguaje abierto Preprocesador de Hipertexto (PHP) que va a simular a una estación móvil (celular) sobre el sistema operativo de LINUX que va a tener una interacción directa con una Central de Conmutación Móvil (MSC).

El esquema en la plataforma de LINUX se realiza mediante la ejecución de varios archivos en formato PHP y HTML que crean *scripts* para que estos simulen diferentes centrales de conmutación móvil y las respectivas estaciones móviles, y se genere una llamada entre ellas además consulta y recarga de saldo.

### **1.3.1 Objetivos Generales**

El principal objetivo del proyecto es implementar el protocolo SS7 en una red celular, que podrá ser visualizada sobre LINUX mediante un dispositivo virtual desarrollado en el lenguaje abierto Preprocesador de Hipertexto (PHP).

### **1.3.2 Objetivos Específicos**

- Describir el funcionamiento y operación del protocolo de acceso.
- Implementar un dispositivo móvil básico virtual capaz de realizar llamadas prepago y postpago.
- Describir los principales flujos de señalización que intervienen del lado del dispositivo final de un abonado de telefonía móvil.

## **1.4 Alcance**

La implementación de nuestro Proyecto tiene como alcance, describir los pasos necesarios para instalar y adecuar, en un servidor Linux, los paquetes

necesarios para poder desarrollar la implementación del protocolo SS7 en una llamada prepago desde la estación móvil hasta la MSC; nuestra aplicación web simulara a una estación móvil en una red celular, este será capaz de hacer llamadas entre celulares, consultar su saldo actual a través del \*282 opción 1, hacer recarga de saldo con el mismo método \*282 pero opción 2 introduciendo un código de tarjeta valida seguido del signo #, actualización de localización cuando se cambie de una MSC a otra e interactúe con los demás proyectos que emplean los protocolos MAP (*Mobile Application Part*), CAMEL (*Customized Applications for Mobile Network Enhanced Logic*), y en general toda la arquitectura SS7 para formar todo en conjunto un servicio de telefonía móvil, como lo son las llamadas entre móviles.

También se presenta una explicación detallada de los diferentes tipos de mensajes de señalización que interviene en la comunicación y funcionamiento de la aplicación.

### **1.5 Limitaciones**

Entre las limitaciones de nuestro proyecto tenemos los siguientes escenarios:

- No se enruta la llamada que se realiza desde un móvil postpago hacia otro móvil cualquiera.
- El proyecto ha sido creado para ser ejecutado en Google Chrome, pues en otros exploradores la visualización resulta alterada.
- No realizará el envío y recepción de mensajes de texto.

- No tendrá base de datos de números guardados.
- No se podrá consultar las últimas llamadas realizadas.
- No se validó el escenario en el que dos interfaces del celular seleccionan el mismo número, por lo que se generaría un conflicto al momento de ejecutar el proyecto.

## **1.6 Metodología**

Este trabajo se enfoca en crear una aplicación acerca del protocolo SS7 desde la estación móvil hacia el centro de conmutación móvil, que se ejecuta sobre LINUX, con la ayuda del lenguaje de programación PHP y HTML, para que corra el programa desde una interfaz visual que se muestra en pantalla. Por estas razones el proyecto queda abierto hacia nuevas modificaciones y aplicaciones que en el futuro puedan servir de referencia hacia nuevos mensajes que se podrían agregar.

Debido a que se ejecuta un orden en la estructura del proyecto se puede observar jerarquías de conceptos que servirán en el entendimiento de hacia dónde se va dirigido, por lo que el primer capítulo se refiere a una descripción de todo el proyecto, así como también los objetivos hacia donde nos enfocamos.

En el segundo capítulo se manejan todos los conceptos necesarios para que se pueda entender todos los procesos que se manejan en el capítulo tres, además conlleva los principios con lo que se maneja el proyecto. En el tercer

capítulo se realiza toda la descripción del proyecto paso a paso, es decir el proceso de cada implementación que se realiza.

En el cuarto capítulo se procede con el análisis de resultado de todo el proyecto, es decir que se procede a describir cada señal que se recibe como respuesta ante un proceso inicial y en general todas las respuestas de las llamadas prepago, y en la actualización de localización. Por último en el capítulo cinco se realizan la argumentación de todas las conclusiones y recomendaciones necesarias para conocer los puntos más importantes del proyecto.

## **CAPÍTULO 2**

### **2 FUNDAMENTOS TEÓRICOS**

#### **2.1 La Señalización SS7 en una Red Celular**

El sistema de señalización N°7 (SS7), define diferentes procedimientos y protocolos, por medio del cual elementos de la Red de Telefonía Pública Conmutada (PSTN) intercambian la información necesaria por medio de un enlace de señalización digital, gracias a esto se logra la comunicación sea inalámbrica (celular) o alámbrica (telefonía fija), el enrutamiento y control; vemos que el sistema de Señalización N°7(SS7) es un elemento clave para la introducción y desarrollo de todos los servicios que nos ofrece la gran red de telecomunicaciones [2]

El sistema de señalización SS7 es un sistema de comunicación máquina a máquina, ya que no requiere en sí una intervención humana para su funcionamiento, debido a que la mayoría de sus procesos son automatizados, entre las principales características de la señalización SS7 podemos recalcar:

- Su flexibilidad ya que puede ser usado en distintos tipos de servicios en la rama de las telecomunicaciones.
- Su confiabilidad ya que en su estructura se puede encontrar funciones que ayudan a eliminar problemas en la red celular de señalización.
- Su Alta capacidad ya que un solo enlace de señalización puede soportar un sin número de troncales.
- Su alta velocidad ya que toma menos de un segundo establecer una llamada.
- Su ahorro de dinero ya que puede ser usado en varios servicios de telecomunicaciones pero utilizando menos hardware que los anteriores sistemas [3]

## **2.2 Arquitectura de la Red SS7**

La red SS7 como ya sabemos es una red por conmutación de paquetes, esta contiene encaminadores de paquetes los cuales son llamados puntos de transferencia de señalización (*STP, Signaling Transfer Point*) y dispositivos terminales que pueden ser conmutadores telefónicos, servidores o bases de

datos, estos dispositivos tienen el nombre de puntos de señalización (*SP*, *Signaling Point*).

Gracias a la arquitectura de la red de señalización SS7, dos conmutadores telefónicos intercambian en todo momento mensajes de señalización, independientemente de los circuitos establecidos entre ellos [4]

La arquitectura de señalización SS7 consta de tres puntos de señalización principales, conectados entre sí por enlaces de señalización.

- SSP (*Service Switching Point*)
- STP (*Signal Transfer Point*)
- SCP (*Service Control Point*)

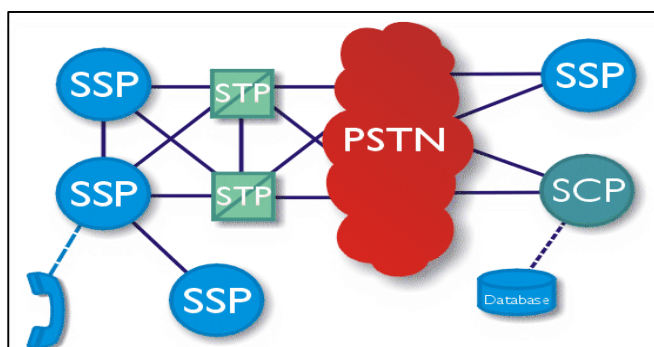


Figura 2.1 Arquitectura de señalización SS7



Los puntos de señalización son los que permiten el acceso a la red SS7, a las bases de datos, también son los que permiten la transferencia de mensajes SS7 a diferentes puntos de señalización.

Como observamos en la figura todos los paquetes se transportan de un SSP a otro SSP usando siempre los servicios que tiene el STP, en la arquitectura SS7 nos podemos dar cuenta que el STP funciona como un Enrutador ya que conmuta y dirige los mensajes que recibe del SSP a su destino correspondiente.

Existen 3 niveles de STP (Nacional, Internacional y Puerta de Enlace o *Gateway*)

- STP Nacional: Este se encuentra dentro de una red nacional y tiene la capacidad de transferir mensajes utilizando el protocolo SS7 nacional. También puede transferir mensajes a una red internacional pero para esto utiliza convertidores de protocolos.
- STP Internacional: Su funcionamiento es similar al STP nacional solo que aplicado a la red internacional, utiliza el protocolo SS7 estándar internacional ITU-T, gracias a esto se permite la conectividad entre diferentes países a pesar de que tengan una estructura de direccionamiento diferente.
- STP Puerta de Enlace (*Gateway*): Usada en redes celulares, soporta protocolos nacionales e internacionales, gracias a esto el STP Gateway es el que realiza la conversión de un protocolo en formato nacional a un

internacional. Es utilizado como punto de acceso a redes internacionales SS7.

Utiliza servicios de examinación por medio de tablas para mantener la seguridad de la red. Este examina los paquetes de entrada y de salida, permitiendo el acceso aquellos que son autorizados, así mantiene un correcto funcionamiento de la red ya que evita conflictos de paquetes inestables [5]

El punto de control de servicio (SCP) es un equipo que funciona como interfaz para acceder a la base de datos, la dirección de una base de datos es un número de un subsistema en cambio la dirección del SCP es un punto de código (*point code*), la base de datos contiene información de los usuarios, sus servicios, enrutamiento de números especiales, validación de tarjetas inteligentes.

La SCP puede realizar conversiones del protocolo de SS7 a X.25, o permitir comunicación directa a la base de datos utilizando primitivas, estas son parte del protocolo TCAP.

### **2.3 Capas del Protocolo SS7**

Como podemos observar el protocolo SS7 lo conforman varios protocolos que utilizan un mecanismo en común para lograr el transporte de la información.

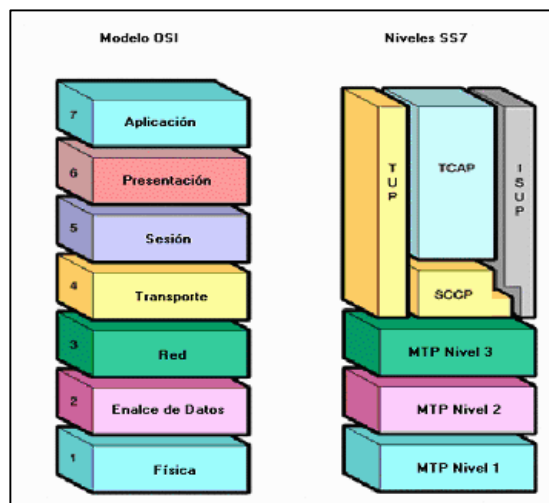


Figura 2.2 Protocolo SS7 vs. Modelo OSI

Las tres primeras capas del protocolo SS7 se la denomina MTP (*Message Transfer Part*), encima de estas tres capas existen varios protocolos que realizan funciones de control de llamada, búsqueda de información de suscriptores entre otras.

A continuación vamos a describir cada una de ellas de una manera resumida comparándolo con el modelo OSI.

### **MTP 1**

Es la capa física como en el modelo OSI, su función es el mantenimiento del enlace físico, además establece la conexión de SP y convierte los mensajes en señales eléctricas.

**MTP 2**

Es la capa de enlace de datos, su función es proveer la transferencia de datos entre SP de una manera segura, revisa cada dato en busca de errores y de ser posible los corrige, también se encarga del control de flujo de los mensajes.

También se encarga de empaquetar los mensajes salientes denominados MSU (*Message Signal Units*), estos paquetes son los que transportan los mensajes SS7 desde las capas superiores.

**MTP 3**

Haciendo comparación al modelo OSI es la capa de Red, esta se encarga del enrutamiento de los mensajes y de la gestión de red es decir control de errores, control de tráfico, etc.

Las capas arriba de MTP3 se las puede denominar usuarias, todas confían en este protocolo para que entregue de manera segura los mensajes enviados.

Debido a que existen diferentes capas encima de este, se ha implementado una manera de para saber a qué capa debe ser entregada la información, cada MSU tiene en su empaquetado un código denominado SIO (*Service Information Octect*) que permite diferenciarlos, ósea que cada capa tiene un SIO diferente.

## **CAPAS SUPERIORES**

En las capas superiores encontramos protocolos encargados del control de llamadas de circuitos como ISUP, TUP y protocolos que brindan servicios no orientados a circuitos como lo es SCCP (*Signaling Connection Control Part*).

### **ISUP**

ISUP (*ISDN User Part*), este protocolo es el encargado del establecimiento de las llamadas telefónicas y de su gestión, abarca las llamadas de voz y de transmisión de datos.

ISUP se deriva de TUP, tienes los principios básicos de este, además permite la integración con redes ISDN.

### **TUP**

TUP (*Telephone User Part*), fue diseñado para el control de establecimiento de llamadas, fue el primer usuario SS7 designado por la ITU-T, TUP no soporta la transmisión de servicios de datos ISDN por lo cual fue reemplazado por ISUP, aunque en algunos países aún se sigue usando este protocolo.

### **SCCP**

Este protocolo utiliza métodos de direccionamiento más avanzados para transmitir la información de señalización a su destino de una manera adecuada y segura.

SCCP es capaz de llegar a la base de datos correcta, mediante la combinación del SSN (*Subsystem number*) y su PC (*Point Code*).

## **TCAP**

TCAP (*Transaction Capabilities Application Part*), está diseñado para hacer consultas a bases de datos y obtener información de ellas, como logra esto, gracias al protocolo SCCP, cada vez que TCAP hace una consulta a una base de datos esta lleva consigo un número que la identifica.

## **2.4 GSM**

GSM (*Global System for Mobile Communications*), es un estándar Europeo, que actualmente es el más usado en casi todos los países, por lo que se considera un estándar global, entre sus características más relevantes están la de compatibilidad entre países y seguridad en el ámbito de radiofrecuencia [6]

### **2.4.1 Arquitectura GSM**

Una red GSM consta de 4 grupos importantes: La estación móvil, el subsistema de la estación móvil, el subsistema de red y conmutación (NSS) y el subsistema de soporte y operación.

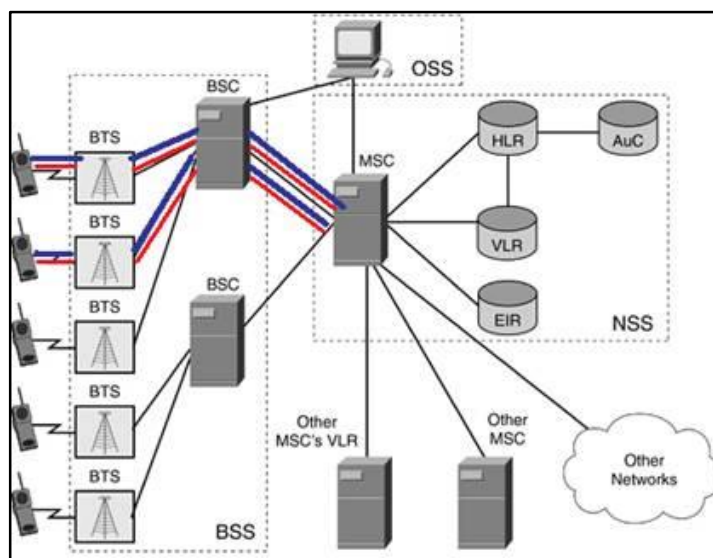


Figura 2.3 Red GSM

A continuación se realiza la descripción de cada grupo del sistema GSM.

**Estación Móvil (*MS, Mobile Station*):** Constituida por el equipo móvil (*ME, Mobile Equipment*) y por el módulo de identidad del suscriptor (*SIM, Subscriber Identity Module*), forman el dispositivo final con el que el usuario podrá integrarse a la red GSM. EL equipo móvil es un dispositivo terminal que va a permitir realizar procesos de comunicaciones ya sea para transmitir o recibir por medio de radio frecuencia. El ME contiene un parámetro único llamado IMEI (numero internacional del equipo móvil) que permite al administrador de red conocer sobre el estado del equipo, es decir si es robado o no.

El SIM es un chip, ósea una tarjeta inteligente que contiene información referente al usuario de una telefónica y con la que se puedan realizar diferentes

servicios de comunicación. Entre los datos más importantes se encuentran el IMSI (Identidad internacional del suscriptor móvil), TMSI (Identidad temporal del suscriptor móvil) y otros parámetros que sirven para dar seguridad al usuario.

**El subsistema de la estación móvil (*Base Station Subsystem, BSS*):** Este subsistema permite al MS conectarse con la red por medio de radiofrecuencia. Se presenta a continuación como está constituida:

**Transceptor de Estación Base (*Base Transceiver Station, BTS*):** Es el que provee la comunicación vía interfaz de radio, a través de grupos de antenas que forman toda la red. Algunas tareas de la BTS son la de cifrado de información, gestión de los canales, modulación y demodulación, entre otros.

**Controlador de Estación Base (*Base Station Controller, BSC*):** Su principal tarea es administrar un grupo de BTS, además de la asignación de canales y asociarse con la MSC en procesos de cualquier servicio requerido por algún abonado de la red.

**El subsistema de red y conmutación (*Network Subsystem and Switching, NSS*):** Es la herramienta principal en el sistema GSM, pues es el encargado de procesar todos los requerimientos en la red. Se constituye por un grupo de base de datos y conmutadores que se presentan a continuación:

**Central de Conmutación Móvil (*Mobile Services Switching Center, MSC*):** Es el encargado del control de la señalización producida en el establecimiento de cualquier tipo de llamada o de otro servicio requerido por el suscriptor.



**Central Principal de Conmutación Móvil (*Gateway MSC, GMSC*):** Su funcionamiento es el necesario para acceder al *roaming* entre centrales de conmutación, pues es el principal eje de conmutación nacional dentro de una red GSM.

**Registro de Localización Base (*Home Local Register, HLR*):** Es una base de datos en la cual se encuentran almacenados información de los abonados de una telefónica en específico, para de esta manera poder administrar y gestionar estos datos. Entre los parámetros que almacena se encuentra el IMSI y el MSISDN que es su número celular en un formato en el que se pueda reconocer desde cualquier parte.

**Registros de Localización de Visitantes (*Visitor Location Register, VLR*):** Es una pequeña base de datos adjunta a cada MSC que se encuentre en una red y en la que administra datos de abonados temporales que al salirse de su cobertura borra lo datos de dicho suscriptor. Uno de los datos que almacena es el TMSI que lo gestiona en un *call setup* o en la actualización de localización.

**Centro de Autenticación (*Authentication Center, AuC*):** Esta base de datos trabaja con el HLR al momento de quererse establecer una llamada o una actualización de localización. En ella almacena parámetros contenidos también en la SIM de un MS como lo es el IMSI y el Ki que es una clave secreta y que servirá en el proceso de autenticación.

**Registro de Identidad de Equipos (*Equipment Identity Register, EIR*):** Esta base de datos contiene el IMEI de los abonados de esa red y lo almacena en tres listas: Lista blanca que son los equipos que pueden accederse sin ningún problema; Lista gris que son los equipos que tiene algún fallo y por último la lista negra que es en la que se tienen los numero de abonados robados.

**El subsistema de soporte y operación (*Operation Subsystem, OSS*):** Es el encargado de dar mantenimiento general de toda la red GSM de una telefónica, con el objetivo de superar alguna brecha o simplemente comprobar el efectivo funcionamiento de la red.

#### **2.4.2 Estructura GSM**

Un área de cobertura es visualizada por medio de células (*cells*) que están compuestas por rangos de frecuencias que pertenecerán a la banda que cada telefónica tiene y que son necesarias que un abonado pueda comunicarse con la red. Cada celda se visualiza con una forma hexagonal y en la que contiene una BTS y es la que proveerá al abonado constante conexión.

Un grupo de 7 células se lo denomina clúster y va a ser administrado y gestionado por algún BSC de esta forma vamos a tener tantos BTS y BSC dependiendo del área de cobertura que se necesite cubrir. Así pues tenemos que un grupo de BSC por ejemplo de una ciudad pequeña va a ser administrado por una MSC, por lo que se van a tener algunas MSC en un país que van a ser administradas por una GMSC, y de esta forma se observa la

jerarquía de la red GSM y la convergencia internacional que tenía al poner comunicarse con la telefónica que administra la GMSC.

### **2.4.3 Canales Lógicos**

La comunicación en la interfaz del aire en la red GSM se maneja por medio de canales físicos los cuales son transmitidos en forma de tramas a respectivas frecuencias dependiendo de la banda del MS, cada trama se subdivide en 8 *time slots*. Cada slot de una trama contendrá su respectivo canal lógico dentro del cual proveerá información necesaria para generar una comunicación entre el MS y la red celular [7]

Los canales lógicos transmiten información necesaria para la señalización de la red, o de tráfico de voz y datos, por lo que se dividen en dos tipos principales de canales:

- Canales de control
- Canales de tráfico

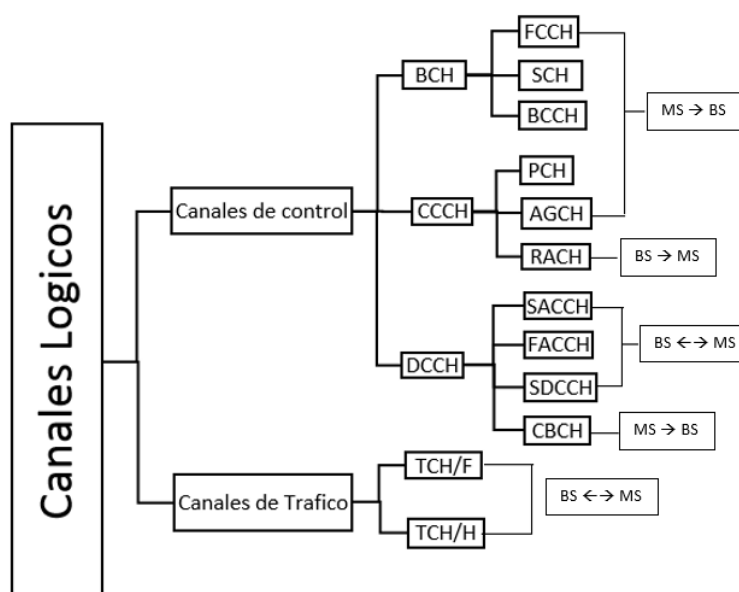


Figura 2.4 División de Canales Lógicos

### Canales de Control

Este canal relaciona toda la señalización necesaria para generar una correcta comunicación entre la red y el MS, es decir controla todo el tráfico dentro de la red celular. Se divide en tres grupos de canales:

- Broadcast Channels (BCH)
- Common Control Channels (CCCH)
- Dedicated Control Channels (DCCH)

### Broadcast Channels (BCH)

El canal de radiodifusión (BCH) transmite información desde el MSC al MS y es el que proporciona información para que el MS pueda sincronizarse con la red; está siempre en estado activo y pueden encontrarse 3 tipos de BCH:

- Frequency Correction Channels (FCCH): La BS trasmite su frecuencia portadora al MS para que se sincronicen por medio del canal BCCH.
- Synchronization Channels (SCH): Su tarea es identificar la BS sintonizada y sincronizar su estructura de trama, además de decodificar la trama en el MS.
- Broadcast Control Channels (BCCH): Identifica diferentes parámetros e información procedente de la BS al MS. Estos parámetros pueden ser el LAI y CI, además de información referente al *paging*, en general es importante para acceder e identificar a la red en la que se está trabajando.

### Common Control Channels (CCCH)

Siempre está en proceso de espera para gestionar requerimientos que quieran acceder a los terminales en la red. Se pueden relacionar 3 canales presentados a continuación:

- Paging Channels (PCH): Permiten a la BS enviar una trama para notificarle a una terminal móvil de una llamada entrante.
- Random Access Channels (RACH): Tiene un enlace ascendente desde el MS a la BS que permita el acceso de una llamada.

- Access Grant Channels (AGCH): Acepta o declina la petición de una llamada entrante al MS desde la central.

#### Dedicated Control Channels (DCCH)

Como su nombre lo indica su función principal es realizar actividades de control específicas entre la red y el móvil. Existen 3 tipos de canales dedicados:

- Slow Associated Control Channel (SACCH): En la BS envía información de la potencia transmitida y del *time advance*. En el enlace ascendente envía información relacionada a *handovers* y potencias de celdas vecinas. Se transmite adjunto a un canal de tráfico, puesto que mantiene constantes actualizaciones de cambios que puedan generarse en el transcurso de una llamada al MS.
- Fast Associated Control Channel (FACCH): Se relaciona rápidamente con el traspaso de un canal de tráfico de una celda a otra, siendo información que se maneja de manera urgente.
- Stand-Alone Dedicated Control Channel (SDCCH): Es un enlace bidireccional entre la BS y el MS, en el que se transmite información asociada al *call setup* y SMS.
- Cell Broadcast Channel (CBCH): Transmite información en vivida en pequeños mensajes desde la BS al MS en forma de radiodifusión general.

## Canales de Trafico

Trasmiten información de voz y datos por lo que se utilizan dos portados correspondientes a un proceso en una red. Se pueden distinguir dos tipos:

- TCH/F: Full Rate. En voz tiene velocidad de 22.8 Kbit/s y en dato (9.6 ,4.8 , 2.6) Kbit/s
- TCH/H: Half Rate. En voz tiene velocidad de 11.4 Kbit/s y en dato (4.8 , 2.6) Kbit/s

### 2.4.4 Handovers

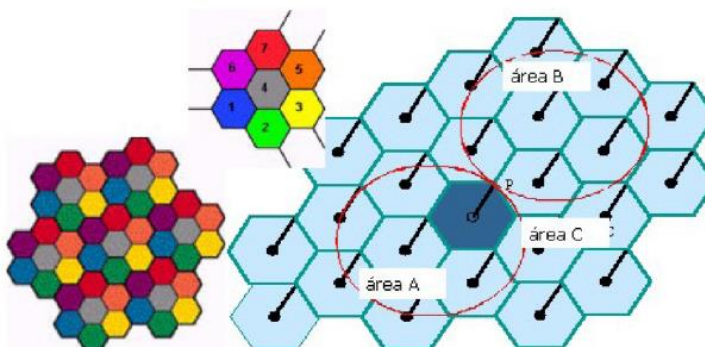


Figura 2.5 Células en diferentes áreas

Este término hace referencia al cambio de una célula a otra de un móvil, ya sea mientras el dispositivo está prestando algún servicio o simplemente si no se lo está usando.

El proceso de *handover* se debe a que el celular siempre va a estar enganchado a la BTS que este más próxima a su área de ubicación y esto va

a ser detectado por medio de la potencia de la antena que percibe el celular, de esta forma si un abonado está en movimiento constante se va a producir este efecto.

Se tienen diferentes tipos de *handovers*, ya que al cambiarse de célula no solo se cambia de BTS, sino que también puede ser de BSC e incluso llegar a cambiarse de MSC, aunque de forma global el concepto es el mismo, es decir, el cambio de células de un abonado celular.

#### **2.4.5 Protocolos de GSM para la Señalización.**

La señalización que se utiliza para el control de cualquier servicio va a ser basada en la señalización SS7 que permite administrar la red de tal forma que solo utilice los recursos necesarios para el establecimiento, y por su estructura de canal común realiza diferentes requerimientos por un mismo canal.

En nuestro caso nos vamos a enfocar en la señalización que se produce desde el MS al MSC, es decir que nos vamos a enfocar en los protocolos que se utilizan hasta la tercera capa.



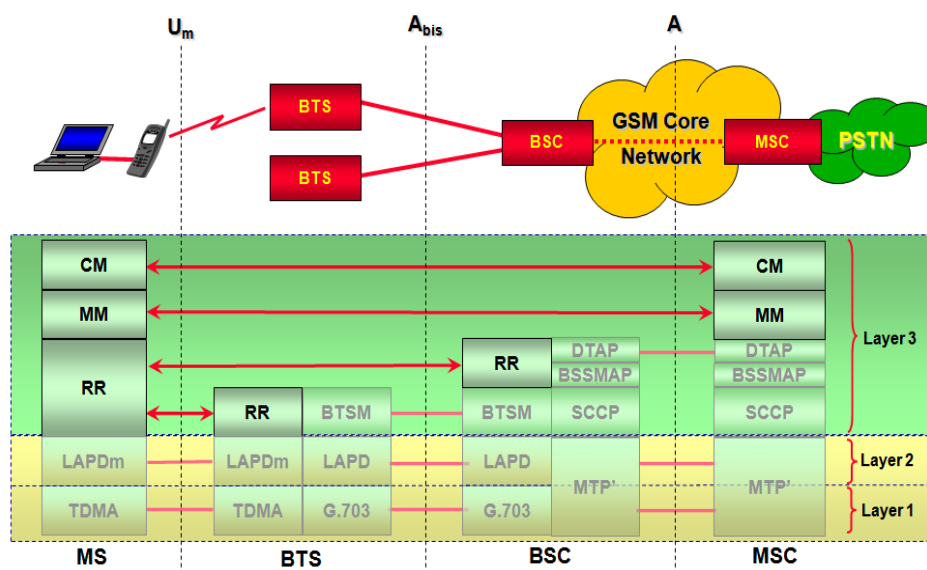


Figura 2.6 Protocolos en Red GSM

La capa física en una red GSM se basa en las transmisiones de información por medio de radiofrecuencia utilizando diferentes técnicas de multiplexación de señales, la más común es la de TDM.

En la capa de enlace de red se maneja el protocolo LAPD (*Link Access Procedures, D channel*) que es el encargado de verificar que las tramas enviadas por radio tengan la secuencia necesaria para que se conecte con la red, y este libre de errores para ser transmitida y empieza el proceso de conexión con la red.

En la capa de red se utilizan diferentes protocolos y en los que nos vamos a enfocar son en los que se relacionan con la señalización SS7.

*The BSS Management Application Part (BSSMAP)* establece parámetros entre la BSS y la MSC para requerimientos de conexión por medio del enlace aire y de esta manera administrar los recursos de este enlace.

*The Direct Transfer Application Part (DTAP)* es el encargado de gestionar el envío de mensajes que se producen en un control de llamadas o en el *mobility management* (MM), por lo que no se interactúa con la BSS.

En esta capa se observan 3 procesos importantes para el establecimiento de una llamada, y estos son:

- *Radio Resource (RR) Management*: Se encarga de administrar y gestionar los canales lógicos necesarios para generar una conexión entre la red y el celular.
- *Mobility Management* (MM): Se establecen mensajes que proporcionan información de *paging*, requerimientos de localización, *handovers*, entre otros que se enfocan en la ubicación del MS.
- *Connection Management* (CM): Es el encargado de enviar mensajes de establecimiento, control y finalización de una llamada.

## **2.5 Short Message Service (SMS)**

El servicio de mensajes cortos o mensajes de textos es una plataforma en la cual una estación móvil o celular es capaz de prestar este servicio el cual permite transportar texto de hasta 140 caracteres desde un MS a otro por medio de una red en la que se incorpora al SMSC (*Short Message Service*

*Center*) que es el centro responsable de proveer el transporte de dicho mensaje desde su punto de partida ósea su generador hasta el MS que va a recibir dicho mensaje.

El protocolo que se utiliza para transportar un SMS es el SMPP (*Short Message Peer to Peer*) que nos permite una gran variedad de compatibilidad con tecnologías móviles como lo es GSM, ya que es un estándar abierto y es el encargado de establecer los parámetros necesarios para transportar el mensaje desde la SMSC hasta las centrales constituidas dentro de la red en la que se está trabajando.

## **2.6 Software Netbeans, XAMPP y Google Chrome.**

**Linux**, es un sistema operativo gratis que se basa en la combinación de un *kernel* o núcleo y una versión de Unix que nos permite modificar su código y de esta manera hacer uso de todas las herramientas que nos brinda su arquitectura.

Entre sus características más importantes se encuentran:

- Permite modificar, y usar de forma gratuita su sistema operativo.
- Brinda múltiples plataformas de programación, ya que es compatible con un extenso número de lenguaje de programación como lo son C, java, php, entre otros que nos permitirán implementar diferentes aplicaciones.
- Nos brinda importantes beneficios de memoria ya que no ocupa mucha memoria.

- Trabaja mediante comandos cualquier configuración por lo cual nos permite reconocer los cambios realizados y estar siempre en constante interacción con su plataforma.

**XAMPP**, es un servidor independiente multiplataforma, de software libre, que consiste principalmente en la base de datos MySQL, el servidor web Apache, y los interpretes de lenguaje PHP y PERL, fue desarrollado por Apache Friends que es grupo de desarrolladores de tecnologías [8]

**NetBeans**, es un software de desarrollo que tiene un entorno muy completo y profesional, contiene muchas funcionalidades que facilitan la programación, fue creado principalmente para la programación en lenguaje java, [9] pero hoy en día está adaptado para diferentes tipos de lenguajes de programación, NetBeans es un producto de libre distribución además es multiplataforma [10].

**Google Chrome**, es un navegador web multiplataforma desarrollado por Google, es de código abierto, es el navegador más usado en el mundo por su estabilidad, confiabilidad y rapidez, fue lanzado por primera vez en el año 2008 con una versión beta, y a finales de ese año ya se lanzó una versión estable para el uso general [11]

## **CAPÍTULO 3**

### **3 Desarrollo del Proyecto**

Los diferentes servicios que se presentan en este proyecto están enfocados principalmente en la información que se transmite desde una entidad a otra, para lo cual se emplea como eje central la pila SS7, con la que podremos tener protocolos de comunicación en sus diferentes etapas, los cuales pueden ser: MTP Level 2, MTP Level 3, SCCP, TCAP, ISUP; además de trabajar con otros protocolos como lo son CAMEL, INAP (*Intelligent Network Application Part*) y MAP. El enfoque que realizaremos será el de los protocolos que brinden servicios de telefonía celular desde la estación móvil hasta la MSC, por lo que se trabajaran con protocolos como lo son MTP2, MTP3, SCCP y TCAP [12]

### **3.1 Requerimientos del Proyecto**

Para realizar la aplicación del celular y la simulación del mismo se requiere de una parte en software en la cual se programaran los diferentes mensajes del protocolo SS7 y por otra parte el hardware necesario para la vinculación de todas las entidades, es decir las conexiones físicas que se utilicen para el respectivo traspaso de información.

El sistema operativo que utilizamos es LINUX ubuntu-14.04.1-LTS ya que es gratis, permitiendo utilizarlo sin ningún tipo de restricción, por esta razón los software que utilizamos son de iguales características que su sistema operativo, y estos software son: XAMPP 5.5.19-0, NetBeans 8.0.2-php, Google Chrome stable-current y otras herramientas que utilizamos para generar la interfaz del celular como lo son el protocolo HTML, CSS y JavaScript.

En la parte de hardware utilizamos dos enrutadores que simulen dos áreas de localización diferente y un conmutador que será el encargado principal de mantener una conexión cableada entre centrales de conmutación y HLR. Esta parte se la realiza para el proceso de actualización de localización del celular. También se utilizan laptops en las cuales se podrá procesar tanto los softwares necesarios para el arranque y programación de la aplicación así como también las instalaciones de los mismos y la convergencia de la red en dichos puntos para que de esta forma se genere el correcto funcionamiento del proyecto y cualquier ejecución que realice lo logre con éxito.

### 3.1.1 Topología de Red

Se creó una topología en la que intervienen tres equipos de telecomunicaciones dos enrutadores y un conmutador, con este se va a querer representar tres entidades fundamentales en este proceso las cuales serán dos MSC y un HLR. En esta topología se establecen conexiones alámbricas e inalámbricas ya que se requiere que el celular se conecte de forma inalámbrica a cualquiera de las MSC mencionadas anteriormente.

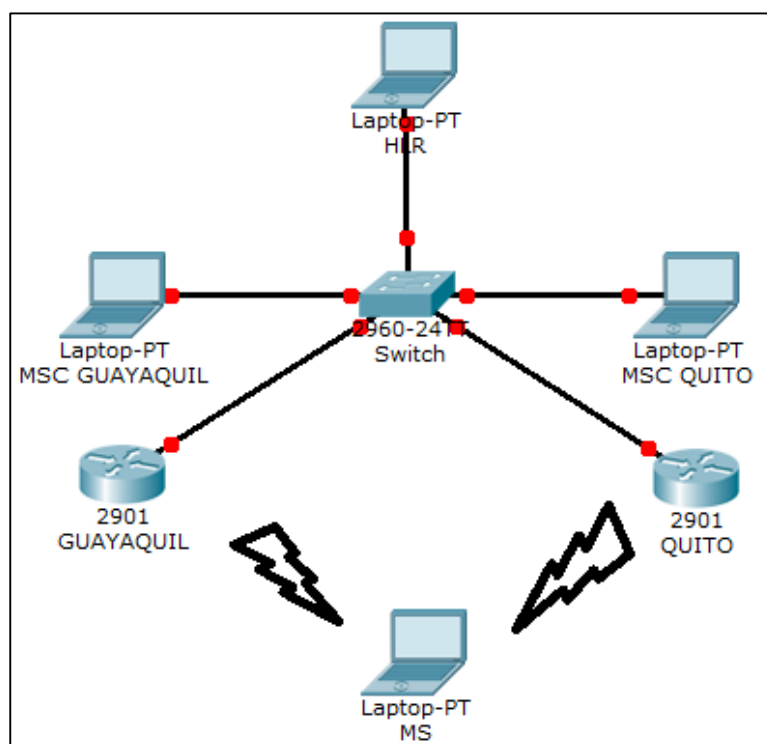


Figura 3.1 Topología de red

## **3.2 Flujo de Mensajes**

### **3.2.1 Flujo Inicial**

Para que el proyecto logre ser convergente con las demás entidades se ha creado un flujo adicional de mensajes para la respectiva inicialización de los escenarios mencionados anteriormente, es decir que los parámetros que contienen cada número celular estén en todas las entidades en la que se procese dicha información que será necesaria para la autenticación e identificación en cualquier proceso posible. El HLR será el que contenga en primera instancia toda la información referente a cada número celular, al igual que en ambas MSC también tendrán información almacenada en base de datos adjunta a ellos denominada VLR, además el HLR tendrá otra base de datos para el proceso de autenticación, dicha base se llama AuC.

Este proceso de inicialización es el que se muestra a continuación:



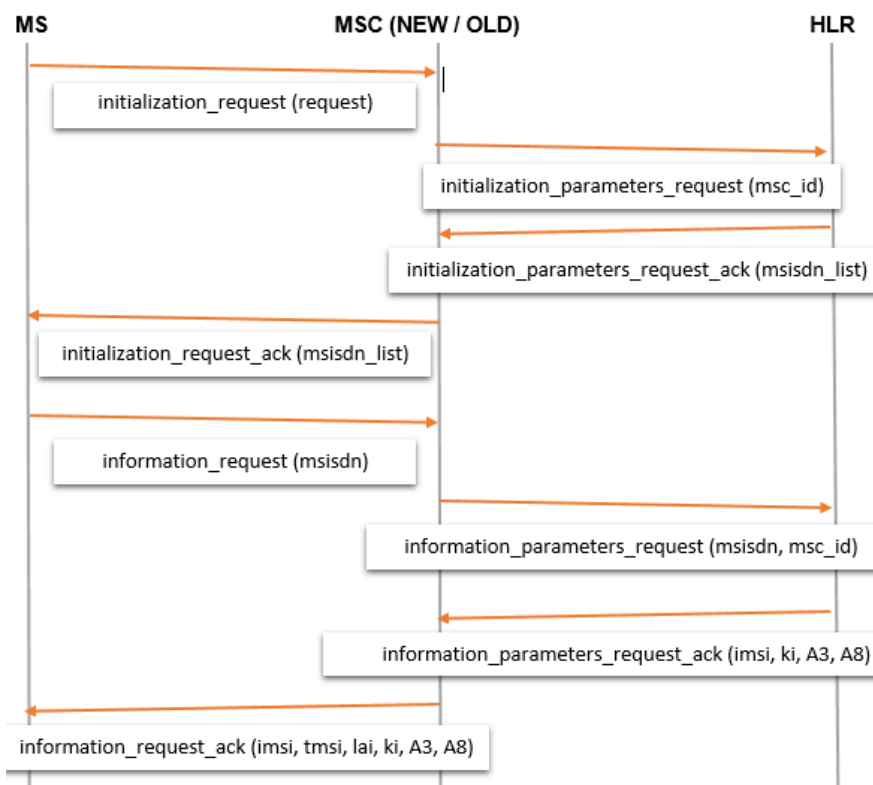


Figura 3.2 Mensaje Señalización Inicialización

### ***INITIALIZATION REQUEST***

Este mensaje se envía desde la MS hacia la MSC en la que se encuentra actualmente que sería la red en la que está conectada y envía el parámetro denominado “*request*” con formato de un bit, que no tiene ninguna relevancia sino que se lo utiliza para seguir con un estándar de comunicación en el cual cada mensaje enviado deberá contener al menos un parámetro a ser procesado, para luego esperar un mensaje que puede ser de acuse de recibido (*acknowledge, ack*) o un mensaje que contenga parámetros que han sido procesados.

Luego el MSC va a enviar un mensaje en el cual le pide al HLR los números de celulares que él tiene almacenados en una tabla de su base de datos, en dicha tabla cada número celular tiene parámetros que identifican a cada estación móvil, como lo son: TMSI, LAI, MSISDN, IMSI, SRES, RAND, KC, MSC\_ID, e información de servicios suplementarios. En otra tabla del HLR se guardaran los parámetros con los que se realizan la autenticación de cada estación móvil, esta tabla se la denomina AuC, y los parámetro que almacenan son: KI, A3 y A8.

En general se han almacenado información de 10 números celulares, los mismos que serán enviados desde el HLR hasta la MSC, para que sea redirigido hacia la estación móvil. Cabe recalcar que cada MSC también consta con información de los números celulares, pero no de todos, ya que esta entidad solo contiene los que se encuentran en su área de localización y se almacenan en el VLR, por esta razón se hizo que cada MSC tengan 5 números de los 10 que se encuentran en el HLR y así distinguirlos de ubicación por medio del MSC\_ID.

El VLR contendrá parámetros de identificación de sus números celulares como lo son el LAI, TMSI, IMSI, y servicios suplementarios, también tendrá identificación de su propia entidad como lo es el MSC\_ID.

***INITIALIZATION REQUEST ACK***

Este mensaje es el que se envía como respuesta al mensaje inicial desde la MSC hacia la estación móvil y contiene como parámetros un arreglo de 10 números celulares en el formato MSISDN, que serán mostrados en la pantalla de la interfaz gráfica del celular, para que el usuario seleccione un número de entre los 10.

***INFORMATION REQUEST***

Una vez que el usuario seleccione un número este será enviado como parámetro del mensaje desde la estación móvil hacia la MSC. La MSC que trabaja como intermediaria para el traspaso de información en este proceso, envía hacia el HLR el número seleccionado para que esta le responda con cada parámetro que tiene dicho número y que serán necesarios para realizar los procesos de identificación y autenticación de la estación móvil.

***INFORMATION REQUEST ACK***

Este mensaje tiene como parámetros, identificadores del número celular seleccionado que son enviados desde la MSC hacia la estación móvil, con lo que culmina este proceso de inicialización en el que se realiza la carga de información hacia la estación móvil y desde ahora será viable realizar cualquier proceso de telefonía celular en el que se emplee el protocolo SS7.

### 3.2.2 Actualización de Localización

El escenario en el que nos enfocamos se explica a continuación es acerca del proceso de actualización de localización de un celular en la cual la principal señalización se basa en el protocolo MAP donde se emplean mecanismos de seguridad para el intercambio de mensajes que se origine entre la estación móvil y la MSC de manera bidireccional.

La actualización de localización se da en un celular por tres motivos:

1. Encendido de un celular
2. Registro por el cambio de área de localización
3. Periódicamente.

Este escenario se basa en el segundo ítem, es decir en el caso de que una estación móvil cambie su área de localización, es decir realiza el proceso que se conoce como *handoff* o *handover* por lo que el celular ejecutara una actualización de localización ya sea por el cambio de BS, MSC, o en general el cambio de una celda a otra. Este cambio de localización del celular se efectúa en el proyecto por medio de redes inalámbricas, ya que contamos con dos enrutadores que representan dos áreas diferentes, además de representar dos MSC distintas, de esta manera el celular tiene la opción de conectarse a cualquiera de la dos redes y por ende cambiarse de MSC en cualquier instante de tiempo y empezar el proceso de actualización de localización.

Al momento que se realice el cambio de red de la computadora donde se encuentra la interfaz gráfica del celular se empieza a hacer uso del protocolo

MAP para el traspaso de mensajes entre el MS, la MSC vieja (previa al cambio de red), la MSC nueva (en la que se encuentra actualmente) y el HLR, para generar la actualización de localización.

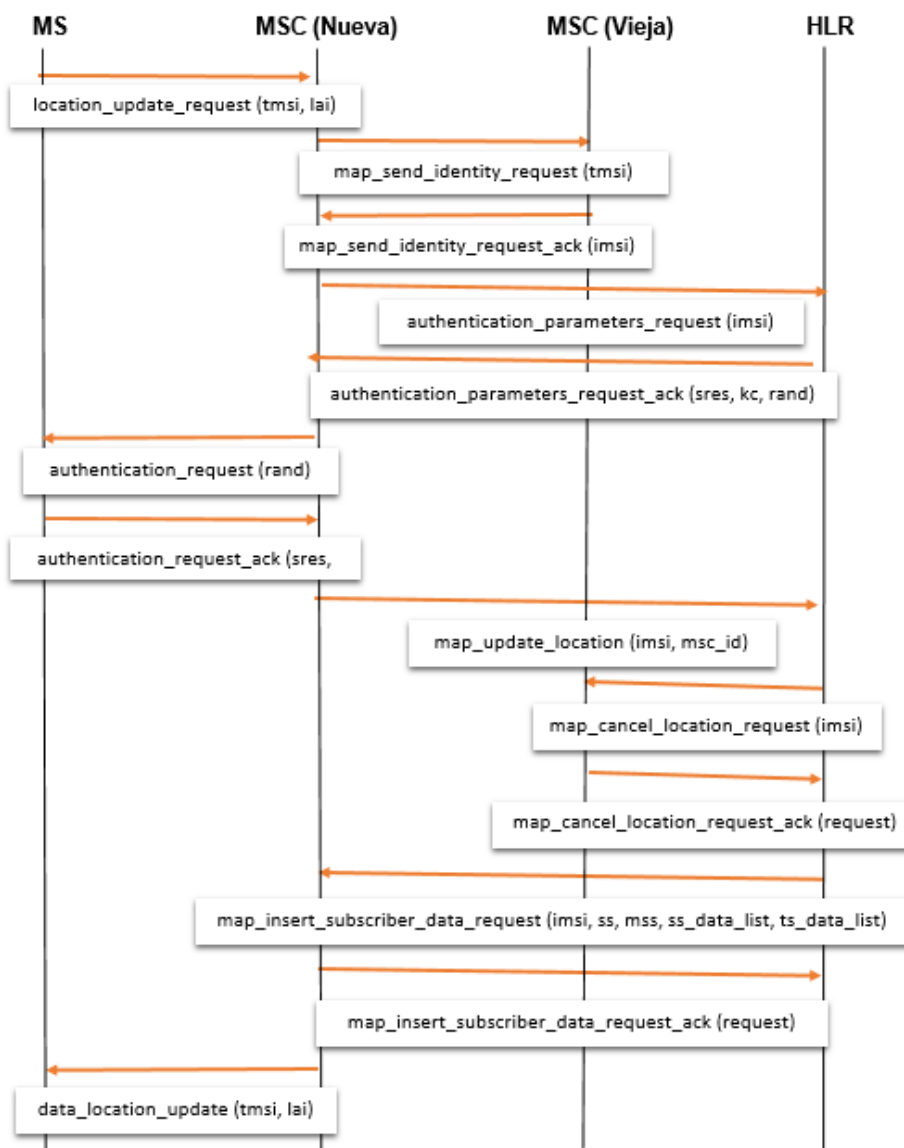


Figura 3.3 Señalización de actualización de localización

### ***LOCATION UPDATE REQUEST***

Este es el mensaje inicial que ocurre desde la estación móvil hacia el MSC cuando se cambia de área de localización, o en nuestro caso el cambio de una MSC hacia otra, por lo que serán nombradas como MSC nueva y MSC vieja. Este mensaje contiene dos parámetros que son el TMSI y el LAI, necesarios para que la nueva MSC compruebe que ese número no se encuentra entre los números que tienen almacenados en el VLR, ya que el TMSI en su formato contiene el NRI que no es más que la MSC\_ID en la que se encontraba anteriormente, es decir que será con el que se verifique que es nuevo en esa área de localización, y se proceda a ejecutar el protocolo MAP, en el cual participa la MSC vieja que brinda información del celular hacia la MSC nueva, luego el HLR procede a realizar la autenticación del celular enviando un mensaje a la MSC nueva con los siguientes parámetros SRES, KC y RAND que son ejecutados en el AuC vinculado al HLR.

### ***AUTHENTICATION REQUEST***

Este es el mensaje que se origina desde la MSC nueva hacia la estación móvil y contiene el parámetro RAND; Con dicho parámetro se ejecuta un cálculo de dos parámetros por medio de dos algoritmos, los mismos que se ejecutan tanto del lado de la AuC como en la estación móvil.

***AUTHENTICATION REQUEST ACK***

Se origina en la estación móvil y lleva en su contenido dos parámetros los cuales son generados por medio de los algoritmos A3 y A8 que darán como resultado al SRES y KC respectivamente, estos parámetros con enviados hacia la MSC nueva para que a su vez sean enviados hacia el HLR y compara estos parámetros con los que se encuentran almacenados en el AuC y confirmar que efectivamente los datos proporcionados en él envió de mensajes pertenecen a la estación móvil en la que se está ejecutando un proceso de telefonía celular.

Luego de que se comprueba que la autenticación es correcta, se procede a enviar el mensaje de actualización de localización hacia el HLR, para que este cambie en su base de datos parámetros referente a dicho celular, pero antes de hacer esto, la HLR ejecuta un proceso de eliminación del número en la MSC vieja, además de intercambiar información desde la MSC vieja hacia la HLR sobre servicios suplementarios de dicho abonado.

Por último el HLR pasa estos parámetros proporcionados por la MSC vieja, hacia la MSC nueva logrando que se actualicen sus datos y por consiguiente el área de localización de dicho abonado.

***LOCATION UPDATE REQUEST ACK***

Este es el último mensaje en este proceso que se envía desde la MSC nueva hacia la estación nueva, con los parámetros TMSI y LAI, para que el celular

los guarde y de esta manera actualice su localización y complete con éxito dicho proceso.

### 3.2.3 Llamada Celular

En el escenario de una llamada se puede observar una gran parte de protocolos de SS7 que se ejecuta en la parte superior de su pila, para que se genere con éxito el proceso de una llamada celular. Entre los protocolos que se utilizan están los de ISUP, INAP, CAMEL y en general los protocolos inferiores necesarios para el establecimiento de toda la pila de protocolos SS7.

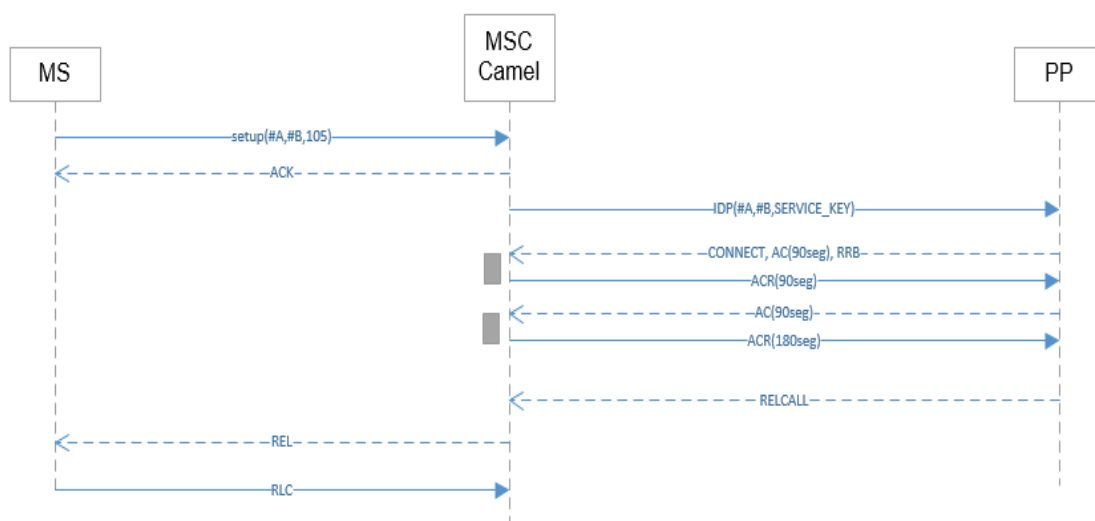


Figura 3.4 Mensaje Señalización Llamada Celular



**SETUP**

Este mensaje es enviado desde el abonado MS\_A hacia la MSC, en el que se envían tres parámetros los cuales son el número del abonado que realiza la llamada (#A), el número del abonado marcado (#B), y por último la categoría de enrutamiento (*Routing Category, RC*) del número A ya que este parámetro indica a la MSC si se trata de un número prepago o pos pago. Luego que este mensaje llega a la MSC se procesan los datos del #A y se verifica que el abonado A sea prepago por medio del protocolo de CAMEL, y en la que intervienen la MSC con la entidad de PREPAGO (PP), esta última contiene todos los abonados prepagos de dicha operadora. Una vez que se verifica en PP que el abonado A tiene saldo se en ruta la llamada hacia el número B mediante el protocolo ISUP.

**ACK**

Este mensaje es enviado desde la MSC hacia el MS\_A e indica que se ha realizado el enrutamiento hasta haber encontrado al MS\_B. Una vez que el abonado B contesta la llamada se envía un mensaje de conexión hacia el PP, para que se genere el respectivo descuento del saldo de su base de datos y se envían mensajes constantes del protocolo CAMEL, verificando si el abonado A aún tiene saldo para seguir conectado con el abonado B; Este proceso se realiza hasta que uno de los dos celulares cuelgue o en nuestro

caso se le acabe el saldo al número A por lo que empieza el proceso de liberación de los dos abonados de la red.

### ***RELEASE***

Una vez que la red termina con la conexión que se había generado se envía este mensaje hacia los dos abonados participantes en la llamada y sirve para que los abonados culminen con cualquier tipo de transferencia de mensajes entre la red y los abonados.

### ***RELEASE COMPLETE***

Este es el último mensaje que se envía desde los dos números hacia la red y simplemente es una respuesta del mensaje anterior para confirmar que se terminó cualquier tipo de transferencia de mensajes.

## **3.3 Software**

Para nuestro proyecto, como ya sabemos un requisito es realizarlo sobre el sistema operativo Linux, en nuestro caso utilizamos la Distribución de LINUX UBUNTU 14.04 LTS, sobre él se instaló NETBEANS como editor de texto para programar nuestra aplicación, adicional se instaló XAMPP que es una aplicación para el desarrollo en entorno PHP, que funciona como servidor independiente, en él se levanta un servidor web en este caso apache, una

base de datos MYSQL y el intérprete de scripts para programar en PHP y PERL.

### 3.3.1 LINUX como Sistema Operativo

Por qué utilizamos LINUX para nuestro proyecto, bueno por muchas razones que vamos a detallar a continuación. Pero primero hablemos un poco de Linux. Linux es un sistema operativo de código abierto y libre distribución, su creador fue Linus Torvalds en el año de 1991, su principal objetivo era impulsar el software de libre distribución, junto con su código fuente para que este sea modificado por cualquier persona en el mundo, este hecho hace que el sistema operativo se expanda y desarrolle.

Entre las características principales de Linux tenemos:

- Es un sistema multitarea, esto quiere decir que Linux es capaz de ejecutar varios programas o aplicaciones al mismo tiempo sin necesidad de detener la ejecución de cada una de las aplicaciones
- Es un sistema operativo multiusuario, esto ayuda a tener varios usuarios en el sistema y ellos a su vez pueden acceder a las aplicaciones y recursos del sistema al mismo tiempo.
- Independencia de dispositivos, Linux permite numerosos tipos de dispositivos gracias a que cada vez que alguien instala un dispositivo en su equipo este se añade a su *kernel* y guarda todas sus configuraciones,

debido a esto se puede decir que Linux posee una gran adaptabilidad y no está limitado como otros sistemas operativos.

- Tiene *shells* programables, un Shell conecta las órdenes del usuario con su *kernel*, que pueden ser programados para adaptarla a las necesidades de cada usuario.
- En comunicaciones, Linux ofrece una mayor flexibilidad para conectarse a cualquier dispositivo, ya que internet se desarrolló en UNIX, y Linux siendo muy similar a este, permite mayores capacidades de navegación.

### **3.3.2 Preprocesador de Hipertexto (PHP) para su Implementación.**

Elegimos PHP como lenguaje de programación gracias a sus beneficios, entre ellos tenemos:

- Lenguaje de programación Multiplataforma
- Está orientado para el diseño Web de contenido dinámico
- Trabaja de lado del servidor por lo que se puede incorporar directamente documentos HTML para ser procesados.
- Su código es invisible al cliente y navegador ya que el servidor se encarga de ejecutarlo y enviarlo al navegador, haciendo esto que la programación PHP sea muy confiable.
- No tiene problemas con la conexión de diferentes motores de bases de datos entre ellos Mysql y PostgreSQL.

- Utiliza diferentes tipos de extensiones que facilita el desarrollo de su programación.

### **3.4 Implementación del Proyecto.**

Para la implementación de nuestro proyecto utilizamos:

1 *Laptop* Toshiba Satellite C55t-B.

Procesador Intel Core I3 Cuarta Generación 1.7 GHz.

750 GB disco Duro.

Memoria RAM 4 GB.

Sistema Operativo Linux, Distribución Ubuntu 14.04 LTS

Para la instalación de Ubuntu, lo que nosotros hicimos fue desfragmentar el disco duro de nuestra laptop ya que sobre el teníamos instalado sistema operativo Windows 8.1, y sobre la partición nueva creada instalar el sistema operativo con el que vamos a trabajar en nuestro proyecto. Primero debemos descargar UBUNTU de la página oficial: <http://www.ubuntu.com/download>, luego instalarlo sobre la partición, siguiendo unos sencillos pasos que encontramos en cualquier blog o página de internet. [13]

Una vez teniendo esto hay que configurar nuestro sistema operativo para desarrollar nuestra aplicación web, que simulara una estación móvil o celular en una red de telefonía. Para ello vamos a programar nuestra aplicación

principalmente en lenguaje PHP, aunque también utilizamos lenguaje JavaScript y HTML para la interacción con los otros programas.

Lo que vamos hacer es instalar XAMPP como se mencionó al principio, también instalaremos como editor de texto Netbeans, y el navegador Google Chrome para la ejecución de nuestro programa.

Para la instalación de estos tres componentes en el ambiente Linux, lo hicimos a través de línea de comandos.

Para XAMPP, primero descargamos de su página oficial (<https://www.apachefriends.org/es/download.html>) el paquete de instalación para Linux. Luego seguimos los pasos de instalación a través de línea de comandos de la siguiente pagina ([http://www.academia.edu/7825418/Ubuntu-guia\\_Instalar\\_Xampp\\_1\\_8.0\\_en\\_Ubuntu\\_12\\_04](http://www.academia.edu/7825418/Ubuntu-guia_Instalar_Xampp_1_8.0_en_Ubuntu_12_04))[14]

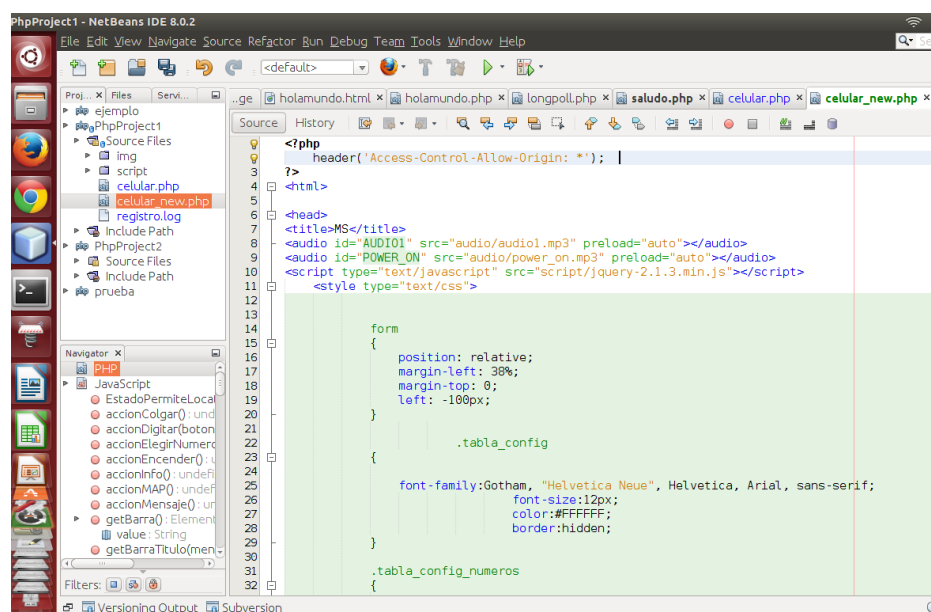
Para Netbeans consultamos el siguiente enlace de instalación en Ubuntu (<https://geofenix.wordpress.com/2014/03/29/instalar-netbeans-8-0-en-ubuntu-13-10/>)[15]

Para google chrome igualmente seguimos estos sencillos pasos detallados en el siguiente enlace (<http://www.howopensource.com/2011/10/install-google-chrome-in-ubuntu-11-10-11-04-10-10-10-04/>)[16]

Luego de hacer todo esto comenzamos con la programación de nuestra aplicación, abrimos NETBEANS y creamos un nuevo proyecto en PHP al que le llamamos PhpProject1.

## CODIGO HTML NETBEANS.

Aquí observamos la primera parte del código hecha en NETBEANS de nuestra Aplicación Web, tenemos la declaración de los audios para el encendido y actualización del celular, además vemos unas declaraciones de la forma de la aplicación donde ponemos su posición, margen, letra, color y estilo de presentación.



```

<?php
header('Access-Control-Allow-Origin: *'); |
?>
<html>
<head>
<title>MS</title>
<audio id="AUDIO1" src="audio/audiol.mp3" preload="auto"></audio>
<audio id="POWER_ON" src="audio/power_on.mp3" preload="auto"></audio>
<script type="text/javascript" src="script/jquery-2.1.3.min.js"></script>
<style type="text/css">

    form
    {
        position: relative;
        margin-left: 38%;
        margin-top: 0;
        left: -100px;
    }

    .tabla_config
    {
        font-family:Gotham, "Helvetica Neue", Helvetica, Arial, sans-serif;
        font-size:12px;
        color:#FFFFFF;
        border:hidden;
    }

    .tabla_config_numeros
    {

```

Figura 3.5 Código HTML en NETBEANS

En la *Figura 3.6* observamos la siguiente parte del código, donde seguimos con algunas declaraciones, en este caso tenemos la de celular, pantalla, barra de notificaciones y de algunos botones, a estas le configuramos su posición, tamaño, estilo de letra, color, imagen etc.

```

52     }
53
54
55     #celular
56     {
57     position: absolute;
58     width: 307px;
59     height: 563px;
60     left: 103px;
61     top: 5px;
62     background-image: url(img/img_phone.png);
63     text-align: center;
64     border-bottom: hidden;
65     border: hidden;
66     }
67
68     #pantalla
69     {
70     position: absolute;
71     border: hidden;
72     background-color: grey;
73     top: 95px;
74     left: 134px;
75     width: 242px;
76     height: 184px;
77     overflow: auto;
78     text-align: center;
79     font-family: "Lucida Grande", "Lucida Sans Unicode", "Lucida Sans", "DejaVu Sans", Ve
80     color: #ffffff;
81     }
82

```

Figura 3.6 Código CSS

Código de programación de los mensajes.

En esta imagen tenemos las declaraciones de variables que vamos utilizar en todas las funciones a programar para nuestra aplicación. Además de la declaración del audio de encendido (POWER\_ON) y actualización (AUDIO1) y de todos los estados en que puede estar el celular.

```

248
249     #btn_digito_2 {
250     font-family: Consolas, "Andale Mono", "Lucida Console", "Lucida Sans Typewriter", Monospace;
251     }
252 </style>
253 <script type="text/javascript">
254
255     var AUDIO1= document.getElementById("AUDIO1");
256     var POWER_ON= document.getElementById("POWER_ON");
257     var fondo_pantalla='img/fondo.png';
258
259     var ESTADO_INICIO = "INICIO";
260     var ESTADO_MENSAJE = "MENSAJE";
261     var ESTADO_APAGADO = "APAGADO";
262     var ESTADO_CARGANDO = "CARGANDO";
263     var ESTADO_ELEGIR_NUMERO = "ELEGIR_NUMERO";
264     var ESTADO_INFO = "INFO";
265
266
267     var estado = ESTADO_APAGADO;
268     var infoNumeroSeleccionado = null;
269     var infoImsi = null;
270     var infoTmsi = null;
271     var infoLai = null;
272     var infoKi = null;
273     var infoA3 = null;
274     var infoA8 = null;
275     var infoOperadora="Sin Conexion";
276

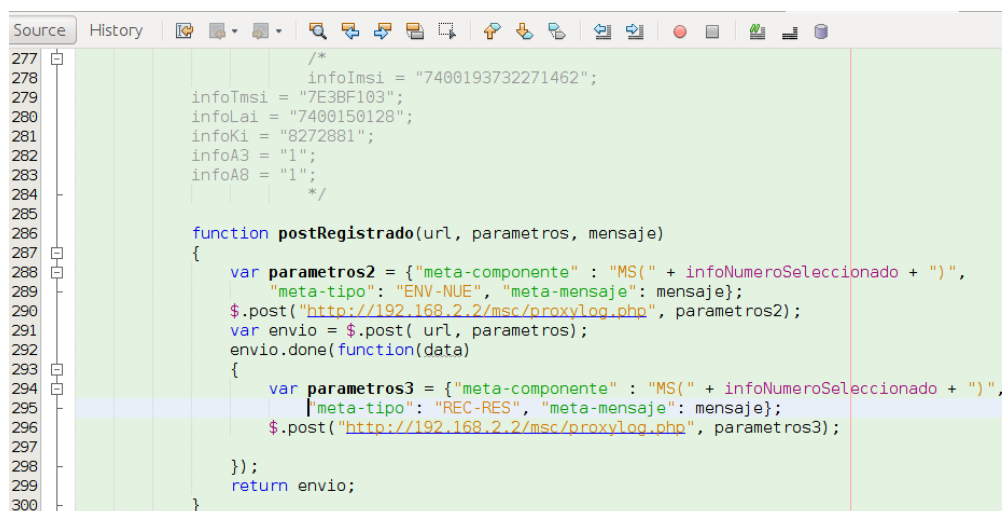
```

Figura 3.7 Variables usadas en el código HTML



En estas imágenes observamos algunas funciones creadas como:

- `postRegistrado` que lo que hace es enviar unos parámetros a través de POST a la MSC para saber si está conectada a uno de ellas.



```

277      /*
278          infoImsi = "7400193732271462";
279          infoTmsi = "7E3BF103";
280          infoLai = "7400150128";
281          infoKi = "8272881";
282          infoA3 = "1";
283          infoA8 = "1";
284      */
285
286      function postRegistrado(url, parametros, mensaje)
287      {
288          var parametros2 = {"meta-componente" : "MS(" + infoNumeroSeleccionado + ")",
289                          "meta-tipo": "ENV-NUE", "meta-mensaje": mensaje};
290          $.post("http://192.168.2.2/msc/proxylog.php", parametros2);
291          var envio = $.post( url, parametros);
292          envio.done(function(data)
293          {
294              var parametros3 = {"meta-componente" : "MS(" + infoNumeroSeleccionado + ")",
295                              "meta-tipo": "REC-RES", "meta-mensaje": mensaje};
296              $.post("http://192.168.2.2/msc/proxylog.php", parametros3);
297          });
298          return envio;
299      }
300  
```

Figura 3.8 Función para registrar el flujo en código java

- `accionEncender`, donde verificamos el estado si es `ESTADOAPAGADO` entonces procedemos a la acción de encender, cambiando el fondo de la pantalla, la barra de notificaciones y llamamos a la función `mensajeInitalizationRequest()`;

```

304     }
306     function accionEncender()
307     {
308         if(estado === ESTADO_APAGADO)
309         {
310             estado = ESTADO_CARGANDO;
311
312             document.getElementById("pantalla").style.background="url("+fondo_pantalla+")";
313             getBarra().style.background="url('img/fondo_barra.png')";
314             mensajeInitializationRequest();
315             //mensajeLocationUpdateRequest();
316             //window.setTimeout(function(){
317             //    mensajeLocationUpdateRequest();
318             //}, 1000);
319
320         }else
321         {
322             estado = ESTADO_APAGADO;
323             getPantalla().style.background= "grey";
324             getPantalla().innerHTML = "";
325             getBarra().style.background= "grey";
326             getBarra().value = "";
327         }
328     }
329 }
330

```

Figura 3.9 Estados de cambios iniciales del celular

- `mensajeInitializationRequest`, esta función que cada vez que es llamada la función de encender, primero reproduce el sonido de encendido, además envía a la MSC a través de POST un *request*, si todo está OK, esta pide los números y los muestra en una tabla en la pantalla para poder elegir uno.

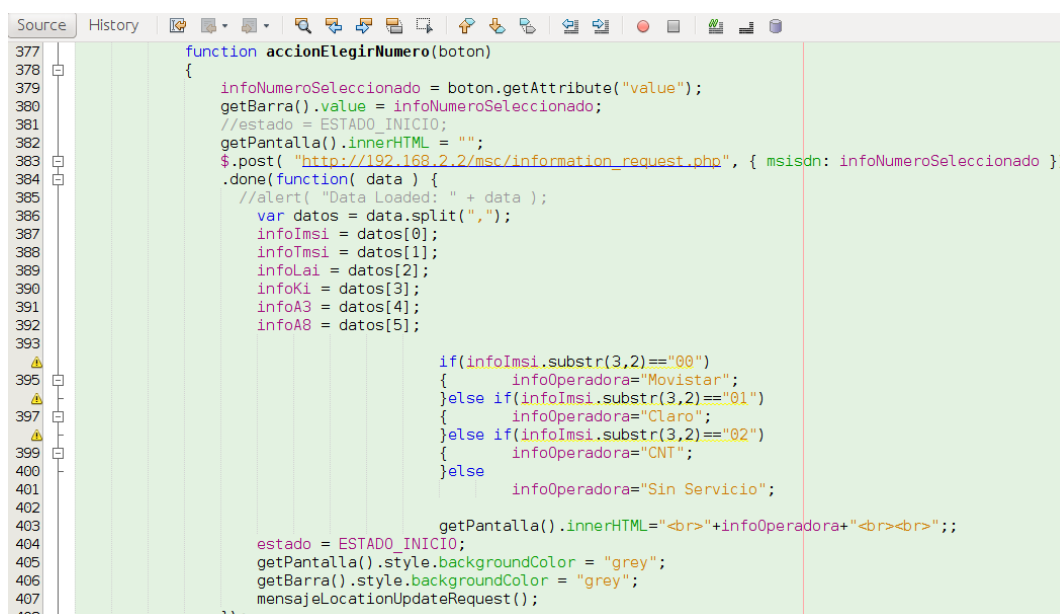
```

331     function mensajeInitializationRequest()
332     {
333         //alert("Solicitando datos...");
334         //getBarra().value = "Iniciando.." + getHora();
335         POWER_ON.play();
336         getBarraTitulo("Iniciando..");
337         getPantalla().innerHTML = "<br>+info0peradora+<br><br>";
338         //mostrarInfo();
339
340         $.post( "http://192.168.2.2/msc/initialization_request.php", { request: "OK" } )
341         .done(function( data ) {
342             estado = ESTADO_ELEGIR_NUMERO;
343             //alert( "Data Loaded: " + data );
344             var numeros = data.split(",");
345             var menuNumeros = "<br>Seleccionar numero:<br><center><table class='tabla_config_
346             for(var i=0; i<numeros.length; i++)
347             {
348                 menuNumeros += "<tr><td><button type='button' id='btn_numeros' onclick='accion
349             }
350             menuNumeros += "</table></center>";
351             getPantalla().innerHTML = menuNumeros;
352         });
353     }
354
355     function getBarraTitulo(mensaje_barra)
356     {
357         var espacios="";
358         getBarra().value = mensaje_barra+espacios+getHora();
359     }
360
361     function getPantalla()

```

Figura 3.10 Función para iniciar el flujo de mensajes

- accionElegirNumero, como su nombre lo dice, aquí se programó la acción para elegir el número que se muestra en pantalla, luego de esto cargamos todas las variables requeridas con los datos del numero seleccionado, como son el Imsi, Tmsi, Kai, etc. También la operadora a la que pertenece y se lo muestra en pantalla.



```

377 function accionElegirNumero(boton)
378 {
379     infoNumeroSeleccionado = boton.getAttribute("value");
380     getBarra().value = infoNumeroSeleccionado;
381     //estado = ESTADO_INICIO;
382     getPantalla().innerHTML = "";
383     $.post( "http://192.168.2.2/msc/information_request.php", { msisdn: infoNumeroSeleccionado } )
384     .done(function( data ) {
385         //alert( "Data Loaded: " + data );
386         var datos = data.split(",");
387         infoImsi = datos[0];
388         infoTmsi = datos[1];
389         infoLai = datos[2];
390         infoK1 = datos[3];
391         infoA3 = datos[4];
392         infoA8 = datos[5];
393
394         if(infoImsi.substr(3,2)=="00")
395         {
396             infoOperadora="Movistar";
397         }else if(infoImsi.substr(3,2)=="01")
398         {
399             infoOperadora="Claro";
400         }else if(infoImsi.substr(3,2)=="02")
401         {
402             infoOperadora="CNT";
403         }else
404         {
405             infoOperadora="Sin Servicio";
406         }
407         getPantalla().innerHTML="<br>" + infoOperadora + "<br><br>";
408         estado = ESTADO_INICIO;
409         getBarra().style.backgroundColor = "grey";
410         getBarra().style.backgroundColor = "grey";
411         mensajeLocationUpdateRequest();
412     });
413 }

```

Figura 3.11 Función para el traspaso de los números a escoger

- mensajeLocationUpdateRequest, esta función sirve para el cambio de localización, está programado para que cuando se cambie de MSC los valores de su nueva localización cambien y cuando esto suceda se reproducirá un audio.

```

426 function mensajeLocationUpdateRequest()
427 {
428     if(EstadoPermiteLocationUpdateRequest())
429     {
430         getBarra().value = "";
431         postRegistrado( "http://192.168.2.2/msc/location_update_request.php",
432             { tmsi: infoTmsi, lai: infoLai }, "location_update_request")
433         .done(function( data ) {
434             if(EstadoPermiteLocationUpdateRequest())
435             {
436                 var datosLocation = data.split(",");
437                 if(datosLocation[0] === "CANCEL")
438                 {
439                     programarLocationUpdateRequest();
440                     return;
441                 }
442                 getBarraTitulo("Actualizando...");
443                 AUDIO1.play();
444                 var rand = datosLocation[0];
445                 var sres = infoK1 + rand;
446                 var kc = rand + infoK1;
447                 postRegistrado("http://192.168.2.2/msc/authentication_request_ack.php",
448                     { sres: sres, kc: kc }, "authentication_request_ack")
449                 .done(function(data) {
450                     if(EstadoPermiteLocationUpdateRequest())
451                     {
452                         var datos = data.split(",");
453                         infoTmsi = datos[0];
454                         infoLai = datos[1];
455                     }
456                 })
457             }
458         })
459     }
460 }

```

Figura 3.12 Función pedido de actualización de localización

- mostrarInfo, esta sirve para mostrar toda la información de cuando seleccionamos el número, en que MSC estamos y las variables que tenemos guardadas, todo esto se muestra en una tabla en pantalla.
- getHora obtiene la hora del sistema y la muestra en la barra de notificaciones.
- accionMap, esta función abre una nueva ventana, accediendo a la bases de datos de la MSC, para observar el flujo de señalización que se está llevando a cabo.

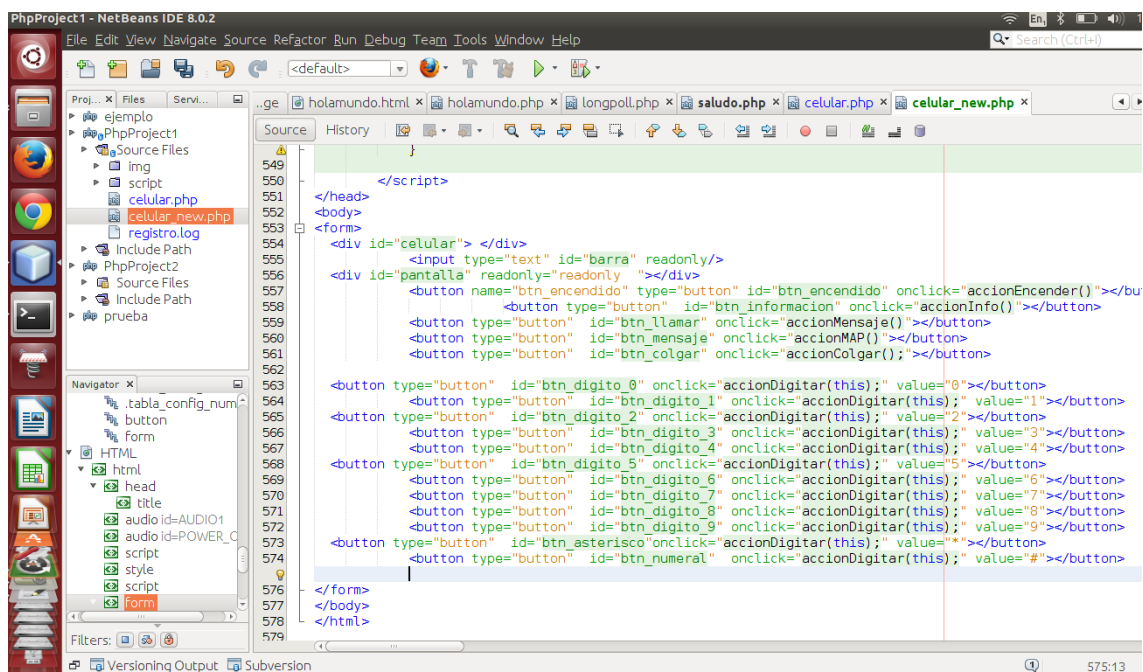
```

519
520     function mostrarInfo()
521     {
522         getBarraTitulo("Informacion ");
523         document.getElementById("pantalla").style.background="black";
524         var msc_id_js=(eval("0x"+infoTms1)).toString(2).substr(-8,8);
525         var name_msc_js="";
526         if(msc_id_js=="00000011")
527         {
528             name_msc_js="GYE";
529         }else if(msc_id_js=="00000100")
530         {
531             name_msc_js="UI0";
532         }else
533         {
534             name_msc_js="";
535         }
536         getPantalla().innerHTML = "<table class='tabla_config' border=1 width=100%><tr><td>MNC:</td></tr></table>";
537     }
538     function getHora()
539     {
540         var hora, d = new Date();
541         hora=d.getHours()+':'+d.getMinutes();
542         return hora;
543     }
544     function accionMAP()
545     {
546         window.open("http://192.168.2.3/html/indexMAP.php", "BASEDATOS",
547             "width=700,height=600,top=20,left=40,scrollbars=NO,titlebar=NO,menubar=YES,\n\
548             toolbar=NO,directories=YES,location=YES,status=NO,resizable=NO")
549     }

```

Figura 3.13 Función para mostrar información

## CODIGO HTML DISEÑO CELULAR Y ACCIONES DE BOTONES



```

549
550     </script>
551 </head>
552 <body>
553 <form>
554 <div id="celular"> </div>
555 <input type="text" id="barra" readonly/>
556 <div id="pantalla" readonly="readonly" ></div>
557 <button name="btn_encendido" type="button" id="btn_encendido" onclick="accionEncender()"></button>
558 <button type="button" id="btn_llamar" onclick="accionMensaje()"></button>
559 <button type="button" id="btn_mensaje" onclick="accionMAP()"></button>
560 <button type="button" id="btn_colgar" onclick="accionColgar()"></button>
561
562 <button type="button" id="btn_digito_0" onclick="accionDigitar(this);" value="0"></button>
563 <button type="button" id="btn_digito_1" onclick="accionDigitar(this);" value="1"></button>
564 <button type="button" id="btn_digito_2" onclick="accionDigitar(this);" value="2"></button>
565 <button type="button" id="btn_digito_3" onclick="accionDigitar(this);" value="3"></button>
566 <button type="button" id="btn_digito_4" onclick="accionDigitar(this);" value="4"></button>
567 <button type="button" id="btn_digito_5" onclick="accionDigitar(this);" value="5"></button>
568 <button type="button" id="btn_digito_6" onclick="accionDigitar(this);" value="6"></button>
569 <button type="button" id="btn_digito_7" onclick="accionDigitar(this);" value="7"></button>
570 <button type="button" id="btn_digito_8" onclick="accionDigitar(this);" value="8"></button>
571 <button type="button" id="btn_digito_9" onclick="accionDigitar(this);" value="9"></button>
572 <button type="button" id="btn_asterisco" onclick="accionDigitar(this);" value="*"></button>
573 <button type="button" id="btn_numeral" onclick="accionDigitar(this);" value="#"></button>
574
575 </form>
576 </body>
577 </html>
578
579

```

Figura 3.14 Código de la interfaz del celular

### **3.5 Aplicaciones**

En este proyecto el principal objetivo es desarrollar una aplicación en la cual se va a simular diferentes procesos que se producen en la telefonía celular y en los que se utilizan los protocolos SS7. Entre los escenarios que se realizarán en el proyecto constan los siguientes: La transferencia de mensajes que se producen desde la Estación Móvil (Mobile Station, MS) al Centro de Conmutación Móvil (Mobile Switching Center, MSC) en una llamada de un móvil a otro.

Otro escenario será el que se realiza en una llamada de servicio a la operadora celular que se la identifica como una llamada de emergencia, y en la que se podrá obtener información del saldo e ingresar una tarjeta prepago. El último escenario que se realiza será el del flujo de actualización de localización que se produce por el cambio del área de localización del celular desde una MSC a otra.

### **3.6 Resumen del Proceso de Implementación del Proyecto**

El proceso de implementación comenzó analizando todos los escenarios en los que la estación móvil interactuaba con la central de conmutación móvil mediante el protocolo SS7 para enfocarse solo en los siguientes servicios: Llamadas entre móviles prepago, llamada de un móvil prepago a la central celular y actualización de localización.

Después de tener toda la información necesaria para reconocer los parámetros y mensajes que se envían en el protocolo SS7, se ejecuta la etapa de instalación de todos los softwares que se utilizan para generar la interfaz gráfica, estos softwares ya han sido mencionados anteriormente y entre ellos tenemos a NETBEANS, XAMPP, Google Chrome, además de otras extensiones. También se instaló como sistema operativo a una plataforma de LINUX llamada UBUNTU y en la cual se trabajó todo el proyecto. Adjuntamente un requisito primordial fue el hardware en el cual se ejecute UBUNTU, para lo cual se utilizó laptops con una RAM de por lo menos 4 GB para que la ejecución del sistema operativo sea el más eficiente y no causen ningún tipo de latencia al momento de implementar el proyecto en dicha plataforma.

Una vez obtenido el hardware y software que se utilizaría en el proyecto, fue necesario ejecutar una topología de red para que el proyecto pueda ejecutarse de manera conjunta con otros grupos de la materia de graduación, por lo que se hizo uso de enrutadores y conmutadores, y se crearon topologías tanto alámbricas como inalámbricas para simular entidades en los procesos de telefonía celular.

Finalmente se realizó el proceso de programación en PHP, HTML, JAVASCRIPT y CSS para generar entidades puntuales que en mi caso fue la estación móvil y en la cual se generaron mecanismos de programación que puedan simular audios y envíos de mensajes del protocolo SS7 mediante

mensajes POST que son manejados en php, dando por finalizado todo el proceso de implementación del proyecto.



## **CAPÍTULO 4**

### **4 Análisis de Resultados**

A lo largo de todo el proceso de implementación se tuvo como idea principal generar una interfaz gráfica de un celular en la cual se puedan ejecutar servicios de telefonía celular como lo son llamadas celulares prepago, postpago y actualización de localización. Los puntos mencionados anteriormente se logró integrar con otros grupos gracias a la arquitectura SS7 y sus diferentes protocolos, en las diferentes entidades de una red celular y de esta forma simular entornos que sean los más parecidos posibles al escenario real de un abonado común. Por otro lado, ya que la parte del acceso físico de una comunicación (canales físicos y lógicos) no podían ser implementados de manera real, en este proyecto fueron implementados en un entorno LINUX y





Figura 4.2 Celular encendido (Seleccione número)

Una vez que se enciende el celular mediante el botón ON/OFF empieza un proceso inicial de selección de un número. Esto se lo realizo por que se debe manejar un entorno en el que todas las entidades contengan parámetros propios de una estación móvil para su identificación y generación de algunos servicios, por esto se creó un proceso inicial y es el que se muestra en la figura 4.2, donde se puede escoger un numero de entre diez de diferentes operadoras. Además se creó una barra de mensajes de estados en la que se puede observar texto referente al proceso de flujo que se realiza.

El proyecto maneja una propuesta en la que cualquier proceso que se realice debe utilizar dos celulares por lo que todo este proceso inicial se repite en ambos celulares o en las interfaces que pueden ser ejecutadas en una misma

computadora o en otra, siempre que se encuentren dentro de la topología de red para poder comunicarse con las entidades de conmutación.



Figura 4.3 Operadora Celular

En esta imagen se puede observar la operadora del número que se ha seleccionado la cual es movistar y lo reconoce ya que la SIM del número contiene un parámetro llamado LAI que en su formato contiene la red celular del país.



Figura 4.4 Información de SIM y otros parámetros (#A)

En la figura 4.4 se puede observar la información de los parámetros que identifican a cada estación móvil y también a parámetros que la ubican dentro de un área de cobertura de una central de la operadora celular. Como observamos el celular A se encuentra dentro del área de cobertura de una central de Quito, esto también se lo visualiza mediante el parámetro LAC que en este caso es 10540; Otros parámetros de localización son el TMSI: 0XHDC008C04 y el LAI: 7400010540

Como el proyecto opera mediante redes inalámbricas se maneja una topología tal que una red simule una central de conmutación móvil (MSC\_GYE) y otra

red simule otra MSC (MSC\_UIO). Entonces resulta obvio que la red en la que se encuentra el celular es la MSC\_UIO.

Siguiendo con el proceso para que se cumpla el cambio de localización se realiza de manera manual la conexión a la red de la MSC\_GYE



Figura 4.5 Barra de estado (actualizando)

Se visualiza en la barra de estado que luego de realizarse el cambio de red, el flujo entra en un proceso de actualización de localización denotado por el seudónimo “actualizando”



Figura 4.6 Información después de cambio de localización

Mediante el botón de información se puede visualizar todos los parámetros referentes a la SIM y se muestran los cambios en los parámetros de área de localización, puesto que se generó un proceso de actualización de localización. Los parámetros procesados son TMSI: 0XHDC008C03, el LAI: 7400016110 y LAC: 16110. Comparando estos parámetros con los que se encuentran en la figura 4.4 se observa que efectivamente se realizó un cambio de localización puesto que sus datos han sido actualizados exitosamente.

## Flujo de mensajes

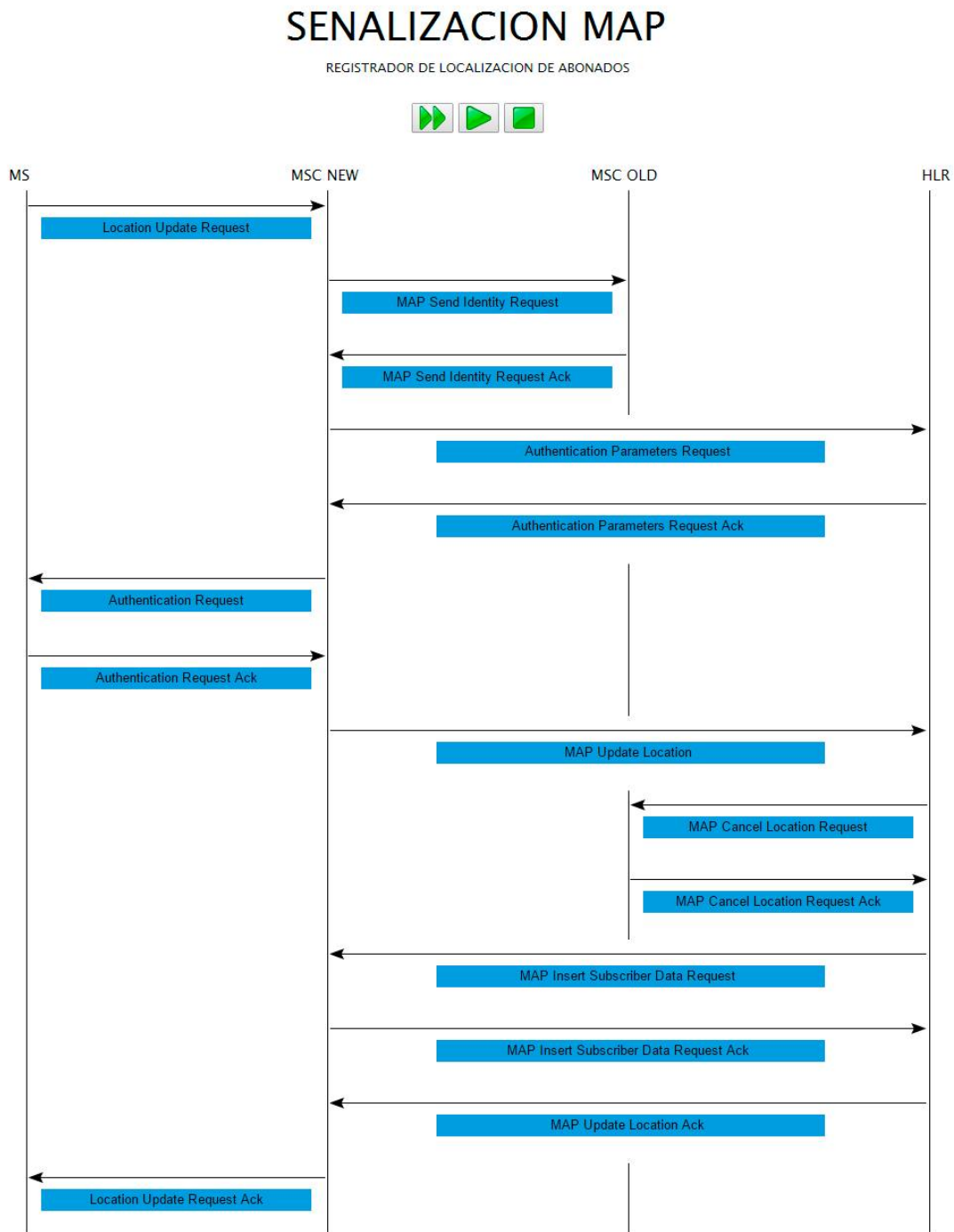


Figura 4.7 Flujo de actualización de localización de nuestra aplicación



La figura 4.7 muestra el flujo de mensajes que nuestra aplicación genera, aquí se observa el intercambio entre entidades en una red celular, utilizando para este proceso el protocolo MAP. Este flujo se lo presenta en otra página web y se la abre mediante el botón mensaje que se encuentra en la interfaz del celular.

Una parte de mostrar el flujo de mensajes es el de poder observar los parámetros que son procesados mediante el protocolo MAP, por eso en el diseño de la interfaz del flujo se pueda dar clic en cada mensaje y mostrar su contenido.

## 4.2 Interpretación de resultados

A través de las interfaces se pueden visualizar de manera más formal todos los resultados, expuesto esto tenemos las capturas de mensajes intercambiados entre la estación móvil y la central de conmutación que muestran por completo la actualización de localización.



Figura 4.8 Parámetros de cada mensaje enviado

En esta imagen se observan los parámetros del primer mensaje llamado “*Location Update Request*” y que contienen el TMSI: DC008C03 y LAI: 7400016110. Además se observa que el mensaje es enviado desde la estación móvil hacia la central de conmutación nueva.

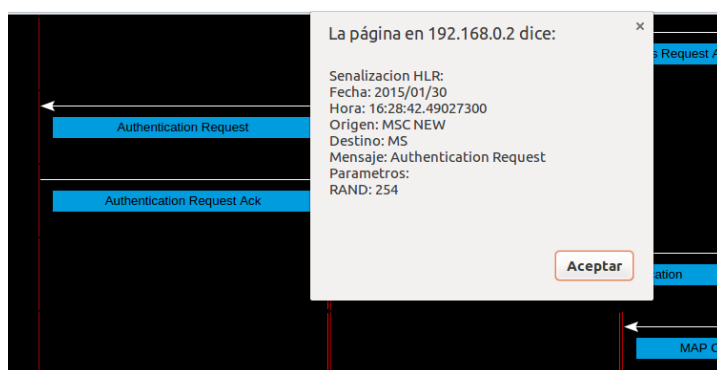


Figura 4.9 mensaje de autenticación

Aquí se entra en la parte de autenticación el celular que se da mediante dos algoritmos el A3 y A8, este mensaje es enviado desde la MSC\_NEW hacia la estación móvil con el parámetro RAND: 254.

Una vez que el celular procesa el parámetro RAND por medio de los algoritmos mencionados anteriormente van a generar dos parámetros el SRES y el KC. Estos parámetros son tanto almacenados en el MS como en el HLR y sirven para dar permiso al celular de conectarse o no con la red de la operadora celular.

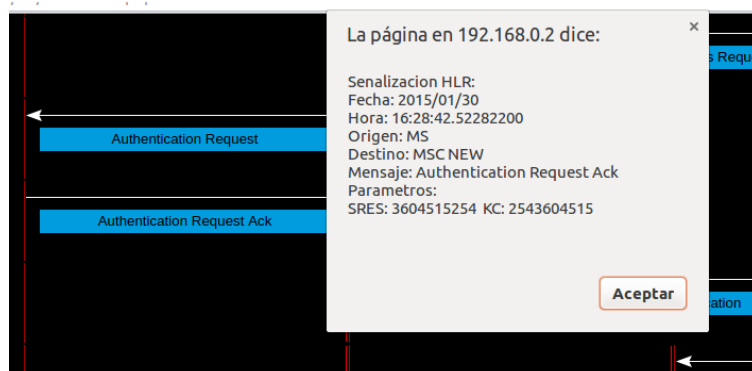


Figura 4.10 Mensaje de respuesta de autenticación

Este mensaje es enviado desde la MS hacia la MSC\_NEW y tiene como parámetros al SRES: 3604515254 y KC: 2543604515. Una vez enviados estos parámetros van a ser comparados en la MSC que deberá decidir si el resultado es correcto o si en caso de no pertenecer a dicha operadora, ser expulsado de la misma.

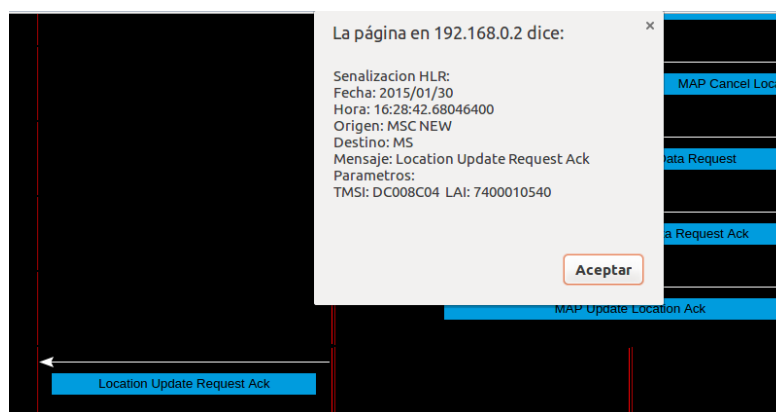


Figura 4.11 Mensaje final

En este mensaje llamado "Location Update Request Ack" se envían los parámetros de TMSI: DC008C04 y LAI: 7400010540 con sus respectivos cambios de área de localización. Estos parámetros en comparación con los de la figura 4.8 han sido cambiados dando como resultado un proceso de actualización de localización exitoso.

## **CONCLUSIONES**

Una vez finalizado la implementación del proyecto y con los resultados obtenidos se pueden realizar las conclusiones que darán un enfoque global de la ejecución de nuestra aplicación.

1. Se logró simular un terminal móvil a través de una aplicación web para realización de llamadas y actualización de localización de un celular, a través de la integración de las entidades que participan en una red celular y con la cual se pudo observar el flujo de señalización que interviene al momento de ejecutar nuestra aplicación.
2. Nuestro proyecto genera una plataforma de comunicación común entre diferentes entidades de una red de telefonía celular que sirve para la ejecución de diferentes servicios y que estos sean manejados de tal forma

que se den prioridades a ciertas transacciones, además de emplear mecanismos que convergen en diferentes aplicaciones, dando como resultado que la transferencia de mensajes sea entendible y pueda ser procesada para respectivo servicios celulares.

3. LINUX como sistema operativo de código abierto nos brindó una gama muy amplia de opciones al momento de realizar la compilación del proyecto en la plataforma de UBUNTU, utilizando como softwares a NetBeans para la respectiva programación en PHP y HTML. XAMPP se lo utilizo para almacenar los *scripts* en servidores para que puedan ser ejecutados en cada entidad y puedan procesar información en los procesos donde se requiera la transferencia de datos.
4. La generación de los servicios celulares como lo son las llamadas entre móviles prepago y postpago, además de la actualización de localización, requirieron de un flujo de señalización implementados en la tecnología GSM que se los acoplo al proyecto de tal manera que solo sea aplicado hacia estructuras que puedan ser simuladas en un entorno de páginas web, por lo que dentro de la pila SS7 solo se ejecutaron mensajes que no empleen la capa física.

## RECOMENDACIONES

En todo el proceso de implementación y ejecución del proyecto surgieron ciertos contratiempos o problemas que fueron resueltos uno a uno, por lo que se redacta a continuación recomendaciones necesarias para una ejecución exitosa o con menos probabilidades de errores.

1. Para el proceso de instalar UBUNTU, en cualquier computadora se recomienda hacerlo sobre una maquina limpia para que no tengan problemas de falta memoria que pueden afectar a la ejecución de la aplicación. En cambio sí tenemos ya una maquina con otro sistema operativo es necesario fragmentar el disco duro para darle espacio donde se ejecuta el sistema operativo de LINUX, luego hay que darle espacio suficiente a ciertas memorias ya sean volátiles o no y así crear una eficiente ejecución de UBUNTU. Además se tuvo que configurar en la BIOS del

computador para que pueda dar la opción de ingresar ya sea al sistema operativo de LINUX o al que se tenía en primera instancia en dicho computador.

2. En la ejecución de NetBeans fue necesario la instalación de extensiones JAVA para que estén puedan ejecutar diferentes programaciones de objeto, además incluir librerías que facilitaron el proceso de programación de todo el proyecto.
3. Es recomendable hacer todas las pruebas posibles de la aplicación para observar todos los fallos o posibles errores de programación que se nos haya pasado por alto.
4. Siempre antes de cada prueba se recomienda realizar *ping* a las direcciones ip de las maquinas que interactúan entre sí, para saber si hay conexión entre ellas, esto nos ahorrara muchos dolores de cabeza, ya que uno está seguro que es un problema de conexión y no de la aplicación en sí.



## BIBLIOGRAFÍA

- [1] lanceTalent, Cuanto cuesta desarrollar una app móvil y como se desarrolla, <http://www.lancetalent.com/blog/cuanto-cuesta-crear-una-app-como-se-desarrolla/>, fecha de consulta enero 2015
- [2] Wikipedia,SS7,[http://es.wikipedia.org/wiki/Sistema\\_de\\_se%C3%B1alizaci%C3%B3n\\_por\\_canal\\_com%C3%BAn\\_n.%C2%BA\\_7](http://es.wikipedia.org/wiki/Sistema_de_se%C3%B1alizaci%C3%B3n_por_canal_com%C3%BAn_n.%C2%BA_7), fecha de consulta noviembre 2014
- [3] Slideshare,SS7,<http://es.slideshare.net/margginathaly/sistema-de-sealizacion-de-canal-comn-ss7>, fecha de consulta noviembre 2014
- [4] UIT, Sistema de Señalización de Red Telefónica, <http://blog.uca.edu.ni/edlacayo/files/2009/09/Sistema-de-Se%C3%B1alizacion-Telefonica.pdf>, fecha de consulta noviembre 2014
- [5] ESPOL, Implementación de protocolo SS7 sobre conexiones entre dos servidores asterik utilizando los equipos SDH del laboratorio de telecomunicaciones, Trabajo, fecha de consulta, noviembre 2014
- [6] A. R. Fernandez, «Despliegue de un Sistema de Telefonía Móvil GSM/GPRS en las Comarcas de Tarragona,» 2007

- [7] Klari Martinez , TODO SOBRE PHP, <http://klarimartinezbenjumea.blogspot.com/2011/04/ventajas-y-desventajas.html>, fecha de consulta enero 2015
- [8] UTEQ, XAMPP, <http://es.slideshare.net/kissees/xampp-25917007>, fecha de consulta noviembre 2014.
- [9] FDI, uso de netbeans, <https://www.fdi.ucm.es/profesor/luis/fp/devtools/-NetBeansUso.html>, fecha de consulta Diciembre 2014
- [10] Wikipedia, Netbeans, <http://es.wikipedia.org/wiki/NetBeans>, fecha de consulta Diciembre 2014
- [11] Wikipedia, Google Chrome, [http://es.wikipedia.org/wiki/Google\\_Chrome](http://es.wikipedia.org/wiki/Google_Chrome), fecha de consulta Diciembre 2014
- [12] Anónimo, Diseño de una red de señalización sobre ip para redes de telefonía móvil GSM y UTMS, Tesis de Grado, 2003
- [13] Wordpress, Linux Ubuntu Instalar Netbeans, <https://geofenix.wordpress.com/2014/03/29/instalar-netbeans-8-0-en-ubuntu-13-10/>, fecha de consulta diciembre 2014
- [14] academia, Ubuntu-guía: Instalar Xampp, [https://www.academia.edu/7825418/Ubuntu-guia\\_Instalar\\_Xampp\\_1\\_8.0\\_en\\_Ubuntu\\_12\\_04](https://www.academia.edu/7825418/Ubuntu-guia_Instalar_Xampp_1_8.0_en_Ubuntu_12_04), fecha de consulta noviembre 2014

- [15] ANONIMO, LINUX UBUNTU, Instalar Netbeans en Ubuntu, <https://geofenix.wordpress.com/2014/03/29/instalar-netbeans-8-0-en-ubuntu-13-10/>, fecha de consulta diciembre 2014
- [16] How Open Source, Install Google Chrome in Ubuntu, <http://www.howopensource.com/2011/10/install-google-chrome-in-ubuntu-11-10-11-04-10-10-10-04>, fecha de consulta Noviembre 2014
- [17] ESPOL, Implementacion de protocolo SS7 sobre conexiones entre dos servidores asterik utilizando los equipos SDH del laboratorio de telecomunicaciones, Trabajo, fecha de consulta, noviembre 2014
- [18] Rafael Barzanallana, Historia de desarrollo de aplicaciones web, <http://www.um.es/docencia/barzana/DIVULGACION/INFORMATICA/Historia-desarrollo-aplicaciones-web.html>, fecha de consulta enero 2015
- [19] ESPOL, Estándar IS-41, Tópico de Graduación 1, fecha de consulta noviembre 2014
- [20] ESPOL, Simulación de la señalización de un usuario móvil y un usuario fijo usando SS7, Proyecto, fecha de consulta diciembre 2014
- [21] Canonical Ltda, Ubuntu Download, <http://www.ubuntu.com/download>, fecha de consulta octubre 2014

[22] Delarze García Denis Michell, Análisis y diseño de un Gateway de múltiples protocolos para mensajería corta, Tesis de Grado, 2005