



Universidad Central de Venezuela
Facultad de Ciencias
Escuela de Computación
Laboratorio de Comunicación y Redes

SAT-ISUPSS7

Sistema Analizador de Tramas de la capa ISUP del protocolo de Señalización No. 7 (SS7)

Trabajo Especial de Grado
presentado ante la ilustre
Universidad Central de Venezuela
por el Bachiller:

Ricardo Gutiérrez
C.I: V-15.507.078
E-mail: ricardo.ucv@gmail.com

para optar al título de
Licenciado en Computación
Tutor Académico: Profa. Ana Morales

Caracas, Mayo 2014

Resumen

En la implementación de servicios de telefonía las redes se diseñan buscando centralizar la transmisión los datos por troncales que conectan los nodos pertenecientes a la red. Al soportar comunicaciones a gran escala (nacionales e internacionales) se debe asegurar el fiel cumplimiento de las normas establecidas en los acuerdos de servicio (SLA) regulados por entes gubernamentales. Para ello se definen umbrales que miden la calidad del servicio prestado con indicadores críticos de desempeño (KPI) que son monitoreados constantemente para asegurar el cumplimiento de la norma.

Por lo consiguiente sería de utilidad para la operación y mantenimiento de redes telefónicas, contar con una aplicación que permita analizar las trazas de control del protocolo de señalización 7 (SS7) capturadas entre diversas centrales telefónicas. El análisis del tráfico SS7 permite solventar inconvenientes o diferencias que puedan surgir en la operación de centrales telefónicas con gran volumen de tráfico. La idea principal es realizar el análisis de mensajes de la capa ISUP definidos para SS7 con el fin de observar algún comportamiento irregular que permita detectar errores, u oportunidades de mejora en la implementación de las redes de telefonía.

El sistema se basa en una aplicación web que permita analizar los archivos de texto plano emitidos por la central y mostrar los resultados agrupando las trazas de llamadas válidas e inválidas. Como parte del diseño se incluye un diagrama de mensajes ISUP que permita describir las transiciones válidas en una llamada telefónica gestionada con SS7. Se ha desarrollado un módulo que permite guardar los datos analizados y los resultados obtenidos para la generación de reportes con la idea de obtener información valiosa para aplicar mejoras en la red, ampliación de servicios y solución de problemas.

Palabras Claves

Trazas, ISUP, SS7, Redes de Telefonía, SLA

Abstract

In the telephone services setup, networks are designed looking to centralize data transmission through trunks that connects core network nodes. Talking about communications at national level (even international), the carriers shall ensure the commitment of government regulations and *Service Level Agreement* (SLA) established with customers for each product. To do that several thresholds are defined to make measures of *Quality of Service* (QoS) through *Key Performance Indicators*, KPI that are monitored constantly to ensure the regulation fulfillment.

Accordingly that will be so useful to *Operation & Maintenance* (O&M) of *Public Switched Telephone Network*, PSTN count with an application that allows to analyze control traces of *Signaling System No. 7*, SS7 obtained from trunk traffic between centrals. The SS7 traffic analysis allow to resolve issues or differences that could arise in the O&M of high traffic nodes. The main idea is to do message analysis in the *ISDN User Part*, ISUP layer defined for SS7 to observe irregular behavior and be able to make error detection or see improvement opportunities in the PSTN development.

The system is based on an web application that allow to process plain text files obtained from trunk traffic sniffing to show analysis' results grouping traffic traces by good and bad calls respectively. As vital contribution is included a state diagram for ISUP messages that allow to describe valid transitions for phone call made over SS7. Into the web application an Input Module is developed to storage input files analyzed and to reproduce the results obtained by this analysis, this features in order to generate value information to apply network optimization, service extension and issues resolution.

Key Words

Traces, ISUP, SS7, PSTN, SLA

Agradecimientos

Gracias:

A Dios por su gran misericordia y bondad, por darme la dicha de estudiar la hermosa Ciencia de la Computación y por todos los precursores que han acumulado el conocimiento y ejemplos hasta nuestros días.

A mi amada **Cristina Campos**, por ser la fuente de inspiración para nunca abandonar hasta en los momentos más difíciles. A ti **Madre** va todo este esfuerzo y superación como tributo por todo tu maravilloso apoyo y confianza, creíste en mí más que yo mismo al inscribirme en la prueba interna de la Facultad de Ciencias.

A mi amado **Edicberto Gutiérrez**, por complacerme con un *Atari 2600* y despertar en mí la pasión por los video juegos y sus teorías; gracias **Padre** por tan maravilloso regalo.

A todos y cada uno de mis profesores por ser pilares de mi formación, por cada hora de conocimiento dado, a todos ellos por su dedicación y pasión por la enseñanza. En especial a mi Tutora la Profa. Ana Morales por tanta paciencia y dedicación en todo este aprendizaje.

A la Casa que Vence las Sombras, a mi querida **UCV**; por darme la dicha de venir a este mundo y recibirme en su Hospital al nacer, por hacerme Caraquista en su Estadio y por hacerme hombre de Ciencias en su Facultad.

A toda mi familia, amigos y seres queridos por su apoyo.

A la Comunidad de Software Libre Internacional, a los desarrolladores y entusiastas por hacer de este mundo un lugar mejor para aprender y vivir.

Dedicatoria

Dedicado a mi Madre y Padre por tanto amor, cariño y formación de vida; por regalarme la curiosidad como virtud y la pasión por el aprendizaje. A ellos por enseñarme a pensar y valorar, por dejarme crecer con la experiencia, por ser incondicionales en el fracaso y alegres en cada triunfo. Dedicado a ustedes mis padres por darme la vida y el amor necesario para llegar hasta aquí.

Dedicado a Cristina y Valentina; mis gemelas hermosas, el motor de mi existencia. A ustedes mis hijas les dedico este *Trabajo Especial de Grado*, TEG con todo el amor del mundo; que sirva como ejemplo de constancia y fortaleza, de determinación a siempre luchar por cada emprendimiento y nunca abandonar en la faena.

Índice general

Resumen	III
Abstract	IV
Agradecimientos	V
Dedicatoria	VI
Índice general	7
Índice de cuadros	10
Índice de figuras	11
1. Introducción	13
2. Motivación, Objetivos y Alcance	15
2.1. Planteamiento del Problema	15
2.2. Justificación	16
2.3. Objetivos	17
2.3.1. Objetivo general	17
2.3.2. Objetivos Específicos	17
2.4. Alcance	18
3. Marco Teórico	19
3.1. Historia de los Sistemas de Señalización	19
3.2. Tipos de Señalización	22
3.2.1. Señalización por canal asociado	22
3.2.2. Señalización por canal común	23
3.3. Puntos de Señalización	24
3.3.1. Enlaces y Rutas de Señalización	25
3.3.2. Modos de Señalización	25
3.4. Sistema de Señalización No. 7 (SS7)	27

3.4.1.	Estructura de SS7	27
3.4.2.	Red de Señalización	29
3.4.3.	Tipos de enlace de señalización	29
3.4.4.	Implementación de SS7	31
3.4.5.	Parte de transferencia de mensajes (<i>Message Transfer Part</i> , MTP)	31
3.4.6.	Fomato y Codigo de Mensajes	33
3.4.7.	Direccionamiento de las unidades de señalización	34
3.4.8.	Funciones	35
3.4.9.	Calidad de Señalización de la MPT (Parte Transferencia del Mensaje)	36
3.4.10.	Partes de Usuario (<i>User Part</i> , UP)	37
4.	Método de Desarrollo	50
4.1.	Tipos de Investigación	50
4.2.	Método de Desarrollo	51
5.	Marco Aplicativo	53
5.1.	Análisis General del Sistema	53
5.2.	Determinación de Requerimientos	55
5.3.	Diseño del Analizador de Tramas	56
5.3.1.	Arquitectura Cliente-Servidor	57
5.3.2.	Descripción del Modelo de Base de Datos	59
5.4.	Tecnologías Empleadas	62
5.4.1.	<i>Frameworks</i>	62
5.4.2.	Tecnologías del lado del Cliente	64
5.4.3.	Framework para el Diseño Web - UIKIT	66
5.4.4.	Entorno de Desarrollo	67
5.5.	Servidor de Aplicación	69
5.5.1.	Servidor Web	69
5.5.2.	Instalación	72
5.5.3.	Archivos de Configuración	72
5.5.4.	Ruby on Rails	75
5.5.5.	MySQL	76
5.6.	Implementación del Analizador de Tramas	77
6.	Pruebas de Rendimiento y Confiabilidad	82
6.1.	Escenarios de pruebas	82
6.1.1.	Definición de los Escenarios	82
6.1.2.	Análisis de Trazas para cada Escenario	83
6.1.3.	Generación de Reportes	89
	Conclusiones	92
6.1.4.	Aportes	92
6.1.5.	Limitaciones y Trabajos Futuros	93

ÍNDICE GENERAL	9
Acrónimos	94
Bibliografía	102

Índice de cuadros

3.1. Sistemas de señalización[1] (*Obsoleto y **Puede ser usado en interconexión de PCs y electromecánicas)	21
3.2. Indicadores de longitud	34

Índice de figuras

3.1. Historia de los Sistemas de Señalización [5]	20
3.2. Sistema de Señalización por Canal Asociado [5]	23
3.3. Sistema de Señalización por Canal Común [5]	24
3.4. Representación del Modo Asociado de Señalización [5]	25
3.5. Representación del Modo Quasi-Asociado de Señalización [5]	26
3.6. Pila de Protocolos SS7 [15]	28
3.7. Estructura de SS7 [15]	30
3.8. Estructura de SS7 [15]	32
3.9. Trama MSU [15]	32
3.10. Trama LSSU [15]	32
3.11. Trama FISU [15]	33
3.12. Formula Erlangs [15]	37
3.13. Mensaje ISUP [15]	38
3.14. Diagrama de bloques funcionales del SPRC [16]	42
3.15. Diagrama de transición de estado para CPC – Llegada (CPCI) [16]	43
3.16. Diagrama de transición de estado para CPC – Salida (CPCO) [16]	44
3.17. Estructura de SCCP [15]	45
3.18. Estructura de TC [15]	48
5.1. Flujo de mensajes ISUP	54
5.2. Captura de Tramas ISUP	54
5.3. Arquitectura de la Aplicación Web[17]	57
5.4. Arquitectura Cliente-Servidor [15]	57
5.5. Modelo Entidad/Relación	59
5.6. Componentes de una aplicación web basada en Ruby on Rails	63
5.7. UIKIT - <i>Framework</i> para el diseño web	66
5.8. Página principal de SATISUP-SS7	67
5.9. Página de resultados SS7-ISA	67
5.10. Entorno de Desarrollo	69
5.11. Sublime	70
5.12. MySQL Workbench	70
5.13. Terminal	71

5.14. Servidor de Aplicación	71
5.15. Máquina de Estados de SAT-ISUPSS7	81
6.1. Revisión del Escenario #1	84
6.2. Validación del Escenario #1	85
6.3. Revisión del Escenario #2	86
6.4. Revisión del Escenario #3 - Llamadas válidas	87
6.5. Revisión del Escenario #3 - Llamadas inválidas	88
6.6. Revisión del Escenario #4 - Prueba de desempeño	89
6.7. Resultados en PDF	90
6.8. Resultados en XLS	91

Capítulo 1

Introducción

El núcleo de negocios de las operadoras telefónicas a nivel nacional está basado en su mayor parte en los servicios de voz y datos, siendo la voz prioridad por el tráfico que debe manejarse para establecer la comunicación entre dos abonados. La implementación de servicios de telefonía las redes se diseñan buscando centralizar la transmisión los datos por troncales que conectan los nodos pertenecientes a la red. Al soportar comunicaciones a gran escala (nacionales e internacionales) se debe asegurar el fiel cumplimiento de las normas establecidas en los acuerdos de servicio (SLA) y las normas dictadas por entes gubernamentales.

Las mediciones del servicio se realizan a nivel interno en la operación para garantizar el cumplimiento de las regulaciones gubernamentales, y a nivel externo para la medir el consumo de cada abonado para su posterior facturación. Estas mediciones deben ser congruentes para evitar inconvenientes o diferencias entre el servicio prestado y el servicio facturado, ya que eso repercute directamente en la percepción del servicio del usuario final.

Para que un servicio llegue a mercados de consumo masivo el usuario final debe observar que la relación precio/valor está acorde con el servicio prestado, es aquí cuando la precisión de los sistemas de medición es crucial para la correcta elaboración de los registros de llamadas (*Call Detail Record*, CDR), ya que éstos serán el insumo para los entes administrativos que intervienen en el proceso de facturación y cobro al cliente.

Un factor importante que puede afectar los procesos de facturación son los errores de implementación que pueden surgir en los protocolos utilizados para la señalización de las llamadas; por ejemplo, si un usuario finaliza una llamada y la señal no es interpretada correctamente por las centrales telefónicas involucradas, entonces es muy probable que el canal quede abierto generando cargos en la facturación del usuario final.

Por lo antes expuesto sería de gran utilidad para la operación y mantenimiento de redes telefónicas, contar con una aplicación que permita analizar el tráfico capturados en troncales de interconexión. Este análisis permitiría solventar inconvenientes o diferencias que puedan surgir en las mediciones del servicio.

Con la idea de cubrir la necesidad antes descrita, se plantea como objetivo del TEG el desarrollo de un sistema para el análisis de tráfico en centrales telefónicas que implementan el sistema de señalización 7 (SS7) con la finalidad de detectar llamadas válidas e inválidas, partiendo del intercambio de mensajes en la capa ISUP con la finalidad de aplicar mejoras en la red, ampliación de servicios y solución de problemas.

Posterior al Análisis y Diseño del sistema se establecen los entornos de desarrollo y producción para la implementación de *SAT-ISUPSS7: Sistema Analizador de Tramas de la capa ISUP del Protocolo de Señalización No. 7 (SS7)* (SAT-ISUPSS7) teniendo como resultado la definición de una máquina de estados determinística que permite realizar el análisis de las tramas ISUP y su implementación en una Aplicación Web que muestra los resultados obtenidos en el análisis con información de gran utilidad para la O&M de la red.

La estructura del TEG comprende los siguientes capítulos con una breve descripción para cada caso:

- Motivación, Objetivos y Alcance
- Marco Teórico
- Método de Desarrollo
- Marco Aplicativo
- Pruebas de Rendimiento y Confiabilidad
- Conclusiones

Capítulo 2

Motivación, Objetivos y Alcance

2.1. Planteamiento del Problema

En la implementación de servicios de telefonía, las redes se diseñan buscando centralizar las interconexiones usadas en la transmisión los datos de cada llamada por grandes troncales que conectan los nodos pertenecientes a la red. En muchas ocasiones estos nodos pueden comunicar redes de distintas operadoras e inclusive nodos nacionales con nodos internacionales, para esta tarea se implementan centrales telefónicas de comunicaciones a gran escala donde cada una posee especificaciones para su manejo y control que vienen dadas por el fabricante.

Al soportar sistemas de comunicaciones a gran escala, los proveedores deben asegurar a las operadoras telefónicas el fiel cumplimiento de las condiciones establecidas en los acuerdos de servicio (SLA) firmados en contrato. Dentro de los SLA un punto de gran importancia es la gestión, manejo y control del equipo de telecomunicaciones instalado; para ello se definen umbrales que miden la calidad del servicio prestado y en el caso de las llamadas telefónicas, se establecen indicadores críticos de desempeño (KPI) que son monitoreados constantemente para asegurar que el servicio prestado al cliente final cumple con los niveles de servicio esperados.

Así como se mide la calidad de un servicio a través de los KPI de un gssla, también se mide el consumo del servicio prestado al cliente final. Dentro de la administración de los servicios de voz esto ocurre en el proceso de facturación que se ya que incide de forma directa en la percepción del servicio del cliente final. Para que el servicio llegue a mercados de consumo masivo el usuario final debe observar que la relación precio/valor está acorde con el servicio prestado, es aquí cuando la precisión de los sistemas de medición es crucial para la correcta elaboración de los registros de llamadas (CDR), ya que éstos serán el insumo para los entes administrativos que intervienen en el proceso de facturación y cobro al cliente.

Un inconveniente muy común en la operación de centrales telefónicas públicas surge cuando existen diferencias importantes entre el servicio prestado y los CDR. Esto ocurre en gran medida por incongruencias en la medición y registro de las llamadas realizadas entre centrales de múltiples

fabricantes o proveedores que pueden llegar a pertenecer a diferentes operadoras de telecomunicaciones.

Otro factor importante que puede afectar los procesos de facturación son los errores de implementación que pueden surgir en los protocolos utilizados para la señalización de las llamadas; por ejemplo, si un usuario finaliza una llamada y la señal no es interpretada correctamente por las centrales telefónicas involucradas, entonces es muy probable que el canal quede abierto generando cargos importantes en la facturación del usuario final.

Para minimizar los errores de medición, cada servicio debe seguir estándares internacionales para lograr la interconexión de elementos de red de diversos fabricantes, los protocolos de red utilizados están prescritos en la mayoría de los casos por estados y transiciones que definen la implementación del servicio. El conjunto de estados y transiciones de un protocolo es lo que se conoce como máquina de estados, ya que define un estado inicial con transiciones hasta y entre estados intermedios y uno o más estados finales que permiten determinar el comportamiento de las comunicaciones en un servicio prestado.

En gran medida, estos estados representan las banderas lógicas a seguir en los procesos de facturación de llamadas, por esta razón es fundamental realizar las auditorías sobre las señales de control que permitan determinar con certeza, la cantidad de llamadas que siguen cumplen con el las transiciones definidas en el protocolo de señalización y las llamadas inválidas que representan inconvenientes en la implementación del servicio. Al tratarse de centrales telefónicas el volumen de datos es muy grande, esto puede hacer que el proceso de auditoría sea una tarea muy difícil de lograr periódicamente si no se cuenta con sistemas que permitan automatizar el análisis de los datos.

Gran parte de las operadoras telefónicas utilizan SS7 como protocolo de señalización y gestión de las centrales telefónicas, tanto para llamadas locales como de larga distancia. La interconexión entre diferentes operadoras se realiza bajo los mismos parámetros con la finalidad de realizar auditorías periódicas, que permitan determinar el nivel de confiabilidad del servicio y de la medición tomada para los procesos de facturación.

Por lo antes descrito se formula la siguiente pregunta, ¿sería de utilidad para la operación y mantenimiento de las redes de telefonía contar con una aplicación que permita analizar las trazas de control y señalización (SS7) capturadas en las troncales que transportan las llamadas realizadas entre diversas centrales telefónicas?

2.2. Justificación

Con el análisis automatizado del tráfico SS7, se permite solventar de forma rápida posibles inconvenientes o diferencias que puedan surgir por la operación de centrales telefónicas con gran volumen de tráfico. La idea principal es realizar el análisis de las transiciones definidas para SS7 con el fin de

observar algún comportamiento irregular que permita detectar errores u oportunidades de mejora en la implementación de los componentes que conforman la central telefónica.

De esta forma, se daría la posibilidad de realizar auditorías periódicas basadas en un muestreo obtenido por trazas que permita determinar con cierto grado de confianza, la calidad de las comunicaciones realizadas a través de las centrales telefónicas y las troncales de interconexión.

Al minimizar los errores de configuración e implementación de las centrales telefónicas, los KPIs del servicio serían mejorados y por consecuencia el proceso de facturación sería más preciso, lo que se traduce en una mejora sustancial en la percepción del servicio recibido por el usuario final.

2.3. Objetivos

2.3.1. Objetivo general

Desarrollar un sistema para el análisis de tráfico en centrales telefónicas que implementan el sistema de señalización 7 (SS7) con la finalidad de detectar llamadas válidas e inválidas, partiendo del intercambio de mensajes en la capa ISUP con la finalidad de aplicar mejoras en la red, ampliación de servicios y solución de problemas.

2.3.2. Objetivos Específicos

- Estudiar y analizar la máquina de estados del protocolo SS7.
- Desarrollo de los módulos de análisis de trazas, almacenamiento y clasificación de los datos procesados.
- Analizar las trazas generadas por el tráfico de llamadas en centrales telefónicas con señalización SS7.
- Diseñar una aplicación web que permita mostrar los resultados del análisis de los datos.
- Realizar pruebas del sistema para la integración de módulos desarrollados.
- Obtener resultados y evaluar el desempeño del sistema.
- Generar un diagrama de mensajes de señalización que permita describir los estados y transiciones válidas una llamada telefónica gestionada con SS7.

2.4. Alcance

Este sistema se plantea con el objetivo de desarrollar una herramienta que permita analizar el volumen de datos generado a partir del tráfico de llamadas en centrales telefónicas. En este caso de estudio se plantea el análisis del tráfico de llamadas entre enlaces troncales con el fin de detectar posibles errores de implementación en el protocolo SS7.

De igual forma se plantea la elaboración de un módulo que permita guardar el detalle de los datos analizados así como los resultados obtenidos para la generación de reportes. Esto con la idea de obtener información valiosa para aplicar mejoras en la red, ampliación de servicios, solución de problemas, etc.

Para la elaboración del trabajo especial de grado se propone la implementación del sistema antes descrito en el cual se pueda detectar las desviaciones ocurridas en el tráfico de llamadas por troncales telefónicas que utilizan el protocolo SS7 como mecanismo de control y señalización, así como el desarrollo de un diagrama de estados basado en los mensajes ISUP definidos en SS7 para la gestión de llamadas telefónicas.

Capítulo 3

Marco Teórico

Existen dos componentes esenciales en toda llamada telefónica, el primero es la voz de los interlocutores que viaja a través del medio; el segundo es la información de control que permite establecer conexiones y realizar el direccionamiento de las llamadas al destino requerido. Esta información de control se conoce como señalización telefónica y está relacionada con la creación de estándares que permitan el funcionamiento adecuado de los servicios de telefonía.

Estos estándares son conocidos como protocolos de Señalización 7 (SS7) y está representado por un conjunto de protocolos que describen los procedimientos usados para establecer comunicaciones entre las centrales de las redes telefónicas. Estos protocolos han sido creados por varios entes al rededor del mundo bajo las pautas de la *International Telecommunication Union - Telecommunication*, ITU-T.

3.1. Historia de los Sistemas de Señalización

Tratando de regular los estándares de señalización a nivel mundial la ITU-T recomendó sistemas de señalización con el fin de ser usados en las comunicaciones internacionales. Los diferentes estándares de señalización se presentan en la *Figura 3.1*:

SSN°1. (1934): Usado para algunos servicios manuales bidireccionales, es un sistema de bajo nivel, es monofrecuente con valor de 500 o 1000 Hz, y para la selección de llamada 20 Hz incluidos en los valores monofrecuentes.

SSN°2. (1938): creado para ser usado en servicios semiautomáticos. Contiene 2 frecuencias ubicadas en 600 y 750 Hz con selección decádica de impulsos.

SSN°3. (1954): se lo emplea para el servicio direccional semiautomático. Trabaja en 2280 Hz tanto para señales de línea como de registro.

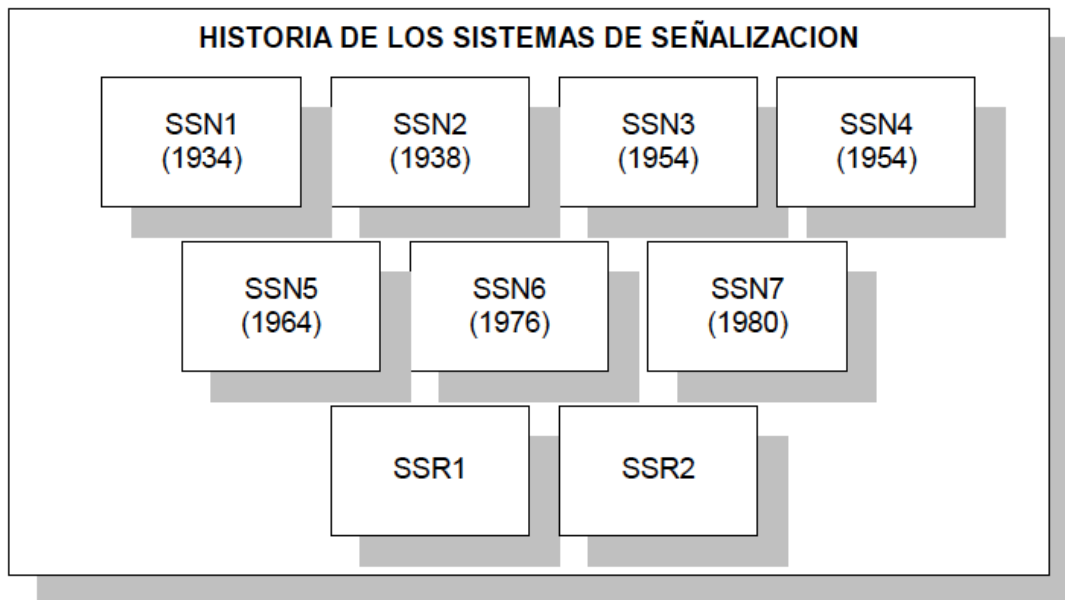


Figura 3.1: Historia de los Sistemas de Señalización [5]

SSN°4. (1954): es el sistema más empleado para tráfico internacional en Europa para redes analógicas, emplea dos frecuencias (2040 Hz y 2400 Hz) y se utiliza en el servicio direccional automático o semiautomático para señales de línea o registros.

SSN°5. (1964): se emplea en conexiones internacionales por cable y por satélite, la señal tiene un período de duración fijo y determinado. El código de impulsos consiste en emitir las señales con un tiempo de duración determinado y sin esperar confirmación.

SSN°6. (1976): usado en servicios dúplex en un canal de 4 hilos mediante un *Modulator - Demodulator*, Modem de datos con modulación en 4 fases para la transmisión digital. La trama consta de 8 ciclos de bloques y cada bloque se divide en 12 unidades de señal. El encabezado tiene la siguiente distribución de bits: 5 bits para la alineación, 4 bits de información, 11 bits de dirección y 8 bits de protección para control de errores.

SSN°7. (1980): utilizado en servicios semiautomáticos y automáticos en líneas bidireccionales. Trabaja para la señalización de línea en las frecuencias de 2400 y 2600 Hz y para señalización de registro en las frecuencias de 700, 900, 1100, 1300, 1500 y 1700 Hz. Es el sistema de señalización más completo en el que se manejan mensajes en un canal común. A diferencia de las anteriores señalizaciones se muestra un modelo de 4 capas.

SSR1: Es apropiado para tráfico semiautomático y automático, se utiliza con código de impulsos para emitir las señales con un tiempo de duración determinado y sin esperar confirmación.

SSR2: Cada paso de mensaje espera la confirmación por el canal de retorno para cortar la señal de ida. Esto implica que la señalización por secuencia obligada requiere de mayor tiempo y una duración no determinada y es utilizado para secuencias obligadas. Trabaja fuera de banda con las frecuencias de 3825 Hz para Europa y 3700 Hz para USA. La voz y la señalización viajan en el mismo canal pero en distinta banda, por ello no se interfieren y se puede efectuar el proceso de cómputo del tiempo de comunicación mediante señales “hacia atrás”. [1]

La siguiente tabla muestra los sistemas de señalización que se tienen al momento en la industria de las telecomunicaciones:

Característica	Sistema de Señalización No.								
	3*	4	5	6	7	R1		R2	
						A	B	A	B
Señalización dentro de banda	X	X	X			X		X	
Señalización fuera de banda								X	
Señalización por canal común				X	X				
Señalización de 2 frecuencias análogas		X	X						
Señalización multifrecuencial análogas	X					X	X	X	X
Señalización digital				X	X		X	X	
Conveniente para operación satelital				X	X		X		X
Usada para operar entre centrales <i>Service Control Point</i> , SCP			X**	X	X	X**	X**	X**	X**

Cuadro 3.1: Sistemas de señalización [1] (*Obsoleto y **Puede ser usado en interconexión de PCs y electromecánicas).

La señalización es una herramienta para establecer canales de control, de igual manera permite gestionar la red para la realización de llamadas mediante la transferencia de información en las diferentes redes de comunicaciones que se encuentran enlazadas. Está constituida por un conjunto de estándares que permiten la operatividad de las redes entre diferentes entidades, éstas brindan transporte e intercambio de información entre el abonado y los conmutadores, entre los conmutadores entre sí y entre los conmutadores y el centro de gestión de red, permitiendo de esta forma una optimización hacia y desde el abonado final.

Entre la gran variedad de funciones destacables que se pueden extraer de la señalización tenemos:

- I. Comunicación audible con el abonado que incluye la señal de línea libre, el tono de marcar, el tono de llamada, la señal de ocupado, entre otras.
- II. Transmisión del número marcado a las centrales de conmutación que intentarán establecer la conexión.
- III. Transmisión de información entre conmutadores indicando que una llamada dada no se puede establecer.

- IV. Transmisión de información entre conmutadores indicando que una llamada ha finalizado y que la ruta puede desconectarse.
- V. Generación de la señal que hace que el teléfono suene.
- VI. Transmisión de información con fines de tasación.
- VII. Transmisión de información indicando el estado de los equipos, las líneas para emplear en el encaminamiento, mantenimiento y diagnóstico de fallos.
- VIII. Control de tiempos de conmutación.

La señalización es una solución ideal para la multiplicación de circuitos en el crecimiento de las redes de comunicaciones, sobre todo en los sistemas de Modulación por Impulsos Codificados (*Pulse Code Modulation*, PCM). Para poder transmitir la señalización estos sistemas utilizan un intervalo de tiempo o canal dedicado.

3.2. Tipos de Señalización

Existen dos tipos de señalización:

- Señalización por canal asociado.
- Señalización por canal común.

3.2.1. Señalización por canal asociado

La *Channel-Associated Signaling*, CAS constituye la señalización de control tradicional en redes de conmutación de circuitos, esta señalización se la realiza a través de la propia línea telefónica o intracanal utilizando el mismo canal para las señales de control y voz. Esta señalización comienza en el abonado origen y sigue la misma ruta que la llamada.

Existen dos formas de señalización intracanal:

- I. **Intrabanda:** se usa la misma banda de frecuencias que las señales de voz que se transmiten, con la ventaja de que las señales de control tienen las mismas propiedades electromagnéticas que las señales de voz, pudiendo llegar a los mismos lugares que éstas. Sus principales desventajas son:
 - a) La velocidad de transferencia de información se encuentra limitada.
 - b) Un canal de voz en uso sólo puede ser utilizado por las señales de control cuando no hay señales de voz en el circuito.
 - c) El retardo existente desde que un abonado introduce una dirección (marca el número) hasta que la conexión se establece.

- II. **Fuera de banda:** se aprovecha el hecho de que las señales de voz no utilizan todo el ancho de banda de 4 KHz asignado y se hace uso de una banda de señalización estrecha e independiente para el envío de señales de control tal y como se muestra en la *Figura 3.2*. La ventaja de este esquema es que estas señales se pueden enviar también cuando no hay voz en el canal lo que permite la supervisión y control continuo de la llamada. Sus principales desventajas son:
- El pequeño ancho de banda.
 - Es difícil transmitir a tiempo los mensajes de control.

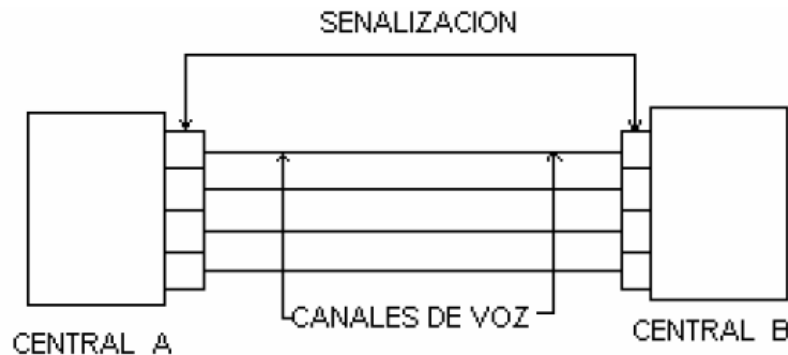


Figura 3.2: Sistema de Señalización por Canal Asociado [5]

Los sistemas que presentan CAS son por ejemplo: lazo desconectado, "robbedbit", CCITT No. 5, CCITT R2 y multifrecuencia (MF) para acceder al discado.

3.2.2. Señalización por canal común

Las señales de control se transmiten por rutas independientes de los canales de voz, dando así lugar a la señalización por canal común tal y como se muestra en la *Figura 3.3*. Los enlaces de datos originados por la diferencia obtenida al separar los datos de control y la voz constituyen los enlaces de señalización.

La Señalización por canal común (*Common-Channel Signaling, CCS*) permite controlar un gran número de circuitos en los canales de voz, las señales de control son un flujo de mensajes que se transfieren en forma bidireccional. Éstos se agrupan en unidades de señales y se le añade información de direccionamiento, identificación del circuito e información para control de errores. La manipulación de los datos de lado y lado permite la gestión de una red distribuida de computadores que se especializan en el transporte de mensajes cortos de control.

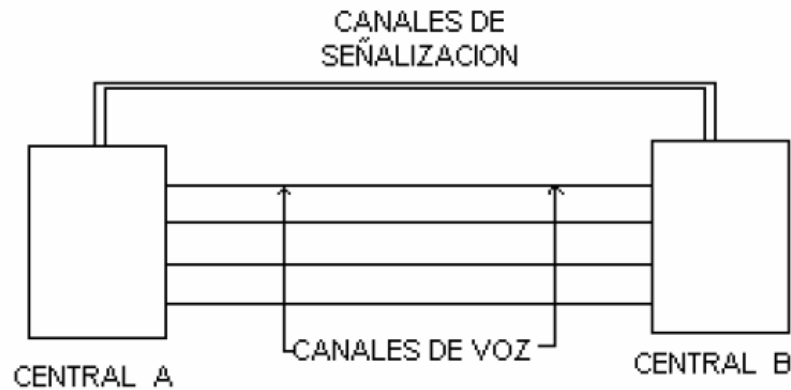


Figura 3.3: Sistema de Señalización por Canal Común [5]

3.3. Puntos de Señalización

Se ha definido una nomenclatura que permite de manera lógica y estructural establecer los nombres de los enlaces que existen en una red de señalización, se pueden nombrar los siguientes:

- **Punto de Origen (*Originating Point, OP*):** es aquel nodo donde se inicia o se genera un mensaje de señalización. En las redes de telecomunicaciones, estos son en primera instancia las centrales telefónicas.
- **Punto de Destino (*Destination Point, DP*):** es aquel nodo a donde se destina el mensaje. Son constituidos en primera instancia por las centrales telefónicas.
- **Punto de Transferencia (*Signalling Transfer Point, STP*):** es aquel nodo donde se recibe un enlace y se transfiere por otro hacia otro punto de señalización. En un punto de transferencia de señalización no se efectúa ninguna operación de conmutación de los mensajes de control. Un STP puede estar integrado en un punto terminal de señalización (una central telefónica) o bien constituir un nodo propio dentro de la red de señalización. En una red de señalización puede haber, según el tamaño de la red, uno o varios niveles de puntos de transferencia de señalización. Todos los puntos de señalización se identifican con un código (como parte del plan de numeración correspondiente) pudiendo así ser diseccionados discrecionalmente en un mensaje de señalización.
- **Punto de Conmutación del Servicio (*Service Switching Point, SSP*):** un nodo SSP envía mensajes de señalización a otro SSP para iniciar, administrar y finalizar los requerimientos de circuitos de voz que completan una llamada. Un SSP puede enviar también un mensaje de pregunta a una base de datos centralizada para determinar enlutar una llamada.
- **Punto de Control del Servicio (SCP):** son bases de datos que proporcionan información necesaria para ejecutar el procesamiento de cada llamada.

3.3.1. Enlaces y Rutas de Señalización

Un enlace de señalización consiste en un canal bidireccional que puede emplearse como un enlace de transmisión. Por razones de redundancia suele haber entre dos nodos más de un enlace de señalización, de esta manera; para el caso de falla de uno de los enlaces se han implementado funciones de SS7 que se encargan de desviar el tráfico de señalización hacia rutas alternativas libres de fallas. El enrutamiento del enlace de señalización entre dos nodos de puede variar y normalmente se agrupan en una troncal de señalización.

Una ruta de señalización es un camino por el que viaja la señalización a un destino determinado, las rutas de señalización que tienen el mismo destino se llaman Grupo de Rutas y pueden contar hasta con 4 rutas a las que se les asigna una prioridad. Las rutas con la misma prioridad operan realizando balanceo de carga.

3.3.2. Modos de Señalización

En la CCS hay dos modos de señalización que hacen referencia a la asociación entre el enlace de voz y el enlace de control, estos son:

- **Modo de Señalización Asociado:** es aquel en el que los mensajes relativos a la comunicación siguen un camino paralelo a los de voz, esto sucede entre dos puntos de señalización (*Signalling Point*, SP) adyacentes.

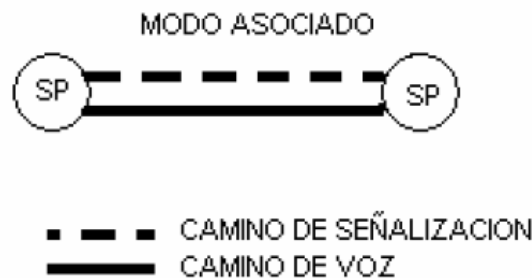


Figura 3.4: Representación del Modo Asociado de Señalización [5]

- **Modo de Señalización Quasi-asociado:** los mensajes pertenecientes a una comunicación se transmiten a través de uno o más SP utilizando para ello dos o más rutas de señalización, este tipo de modo es más potente pero más complejo y bajo esta modalidad se hace crecer la red añadiendo varios STP. En este caso no existe una asignación o correspondencia entre los canales de control y los grupos de enlace y como consecuencia existen dos redes separadas con enlaces entre ellas. Con esta configuración de red se pueden establecer puntos centrales

de control y puede haber un punto que haga de nodo central con una visión global del estado de la red.

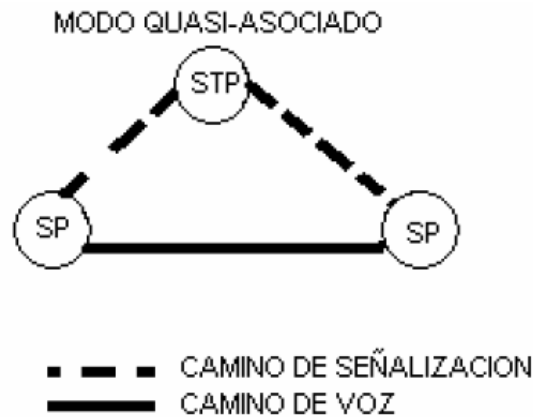


Figura 3.5: Representación del Modo Quasi-Asociado de Señalización [5]

Las señales de control en la señalización por canal común son menos susceptibles a las interferencias entre la señal de voz y de control, esto se debe a que se transfieren en forma directa los mensajes desde el procesador hacia el punto de señalización correspondiente sin agrupar con los canales de voz.

Ventajas de CCS frente a CAS

Las principales ventajas que se encuentran entre la señalización CCS y CAS son:

- Existe menor retardo en el establecimiento de una llamada.
- Número de mensajes prácticamente ilimitados.
- Flexibilidad para nuevos servicios.
- Encaminamiento alternativo.
- Corrección de errores mediante retransmisión de tramas.
- Se cuenta con un menor número de hardware.

Debido a la demanda creciente en las redes de comunicaciones, los estándares adoptados por la ITU-T para el manejo de señalización a nivel mundial han permitido regular las estructuras de las redes de señalización para poder realizar interconexiones entre las diferentes portadoras y *carriers*. Ésto con la idea de unificar las redes de señalización existentes a través de estándares de comunicaciones propuestos hasta llegar a uno de los más utilizados en telecomunicaciones como es el SS7.

3.4. Sistema de Señalización No. 7 (SS7)

El SS7 es un estándar global de las telecomunicaciones definido por la ITU-T que permite el control, la administración y la gestión de redes de telecomunicaciones mediante la implementación de la señalización para lograr la interconexión entre troncales, portadoras y *carriers*. El SS7 se determina fundamentalmente por las características de la red de comunicaciones a la que va a dar servicio, es un sistema de señalización robusto que otorga el control interno y la inteligencia esenciales a una Red Digital de Servicios Integrados (*Integrated Services Digital Network*, ISDN).

SS7 se elaboró a partir de la evolución de los centros de conmutación con programas ya almacenados (*Stored Program Control*, SPC) para circuitos digitales. Los enlaces de señalización junto con las centrales SPC forman la red de señalización lógica por conmutación de paquetes. SS7 es una arquitectura de protocolos de señalización completa donde las unidades de control son los mensajes transportados sobre paquetes. Dicha red está diseñada para trabajar sobre canales de 56 o 64 (Kbps).

Las principales características del SS7 son:

- Arquitectura de protocolos estructura en cuatro niveles.
- Sistema de Señalización por Canal común
- Los nodos y enlaces de señalización son una red de conmutación de paquetes independiente, con un plan de direccionamiento distinto y definido por ITU-T.

3.4.1. Estructura de SS7

La estructura de SS7 se basa en el Modelo OSI y presenta cuatro capas para optimizar el tiempo de procesamiento. El objetivo principal de esta implementación es garantizar y resguardar al máximo la información de señalización para que no se pierda el mensaje que se procesa en los centros de conmutación, aún cuando existan problemas en la red de transporte.

La *Figura 3.6* muestra los niveles de la pila de protocolos utilizados en SS7, entre ellos se pueden describir los siguientes:

- **Capa física (*Message Transfer Part Level 1*, MTP-1):** constituye un enlace dedicado a la señalización y acceso de la red, presenta cierta similitud con la capa física del Modelo OSI (*Open Systems Interconnection Model*, OSI). Utiliza canales DS (*Digital Signal 0*, DS-0) y transporta la data de señalización a tasas de 56 kbps or 64 kbps.[15]
- **Enlace de Señalización (*Message Transfer Part Level 2*, MTP-2):** permite la interacción de los mensajes de señalización estableciendo una unidad de mensajes. Tiene como principales funciones la delimitación de tramas utilizando banderas, el control y corrección de errores y el manejo de la unidad de secuencia de la señal.[15]

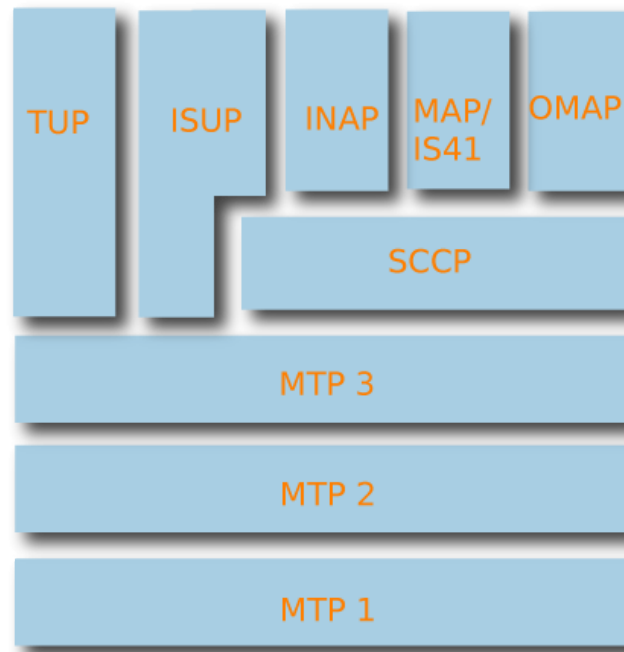


Figura 3.6: Pila de Protocolos SS7 [15]

- **Red de Señalización (*Message Transfer Part Level 3, MTP-3*):** extiende las funcionalidades de la capa anterior para proveer funcionalidades de capa de red. Esto asegura que el mensaje pueda ser entregado entre SP a través de la red SS7 sin importar si estos están conectados directamente. Entre sus principales funcionalidades están el direccionamiento, enrutamiento alterno y control de tráfico.[15]
- **Control de la Conexión de Señalización (*Signaling Connection Control Part, SCCP*):** provee dos de las principales funciones que no están presentes en los niveles de la Parte de Transporte de Mensajes (MTP). La primera es la señalización de aplicaciones a un SP, MTP sólo puede recibir y enviar mensajes de un nodo como un todo sin manejar software de aplicaciones dentro del nodo. Mientras que MTP administra la red para los mensajes que permiten establecer llamadas, otros mensajes son utilizados por aplicaciones (como sub-sistemas) dentro de un nodo; como ejemplo se puede destacar el procesamiento de las llamadas gratuitas (números 800), servicios de señalización para área local, etc. que son manejados por SCCP.[15]
- **Usuario ISDN:** define los mensajes y el protocolo usado en el establecimiento y cierre de llamadas de voz y datos sobre la ISDN y la gestión de la red troncal de la que dependen. A pesar de su nombre, la Parte de Usuario ISDN (ISUP) es usada tanto para redes ISDN como para aquellas que no lo son, en la versión Norteamericana los mensajes ISUP utilizan únicamente MTP para transportar los mensajes entre los nodos involucrados.[15]
- **Capacidades de transacción para la Aplicación (*Transaction Capabilities Appli-***

ation Part, TCAP): define los mensajes y el protocolo usado para comunicar aplicaciones desplegadas como sub-sistemas dentro de los nodos; éste es usado para servicios de base de base de datos como tarjetas prepagadas, número 800, etc.[15]

- **Operaciones, Mantenimiento y Administración (*Operations, Maintenance, and Administration Part, OMAP*)**: define los mensajes y protocolos diseñados para asistir a los administradores de la red SS7, incluye mensajes que usan tanto MTP como SCCP para el enrutamiento.[15]

3.4.2. Red de Señalización

El SS7 utiliza la señalización por canal común y puede transportar los mensajes de señalización necesarios para un gran número de canales útiles. Los enlaces de señalización SS7 enlazan entre sí los nodos en una red de telecomunicaciones. Los puntos y enlaces de señalización conforman una red de señalización autónoma, superpuesta a la red de canales útiles. Un *time-slot* sobre un enlace de señalización T1 o E1, se usa para transmitir los mensajes de SS7 y las aplicaciones tienen la flexibilidad para definir cualquiera de los 24 o 31 *time-slot* como un canal de señalización.

Algunos ejemplos de aplicaciones soportadas por SS7 son: PSTN, ISDN para voz y datos, redes de bases de datos, servicios móviles, administración de operaciones y mantenimiento de redes. En la implementación de redes con SS7 podemos mencionar las siguientes funcionalidades:

- Iniciación básica de una llamada, administración, tarifación y terminación.
- Servicios de valor agregado para la telefonía tales como la llamada en espera, identificador de llamadas, bloqueos por número, etc.
- Administración de red Manejo de la congestión y prioridades para calidad de servicio.
- Servicios inalámbricos de telefonía celular, *roaming* inalámbrico y suscripción de servicios.
- Portabilidad del número local (*Local Number Portability, LNP*).
- Gestión a nivel de base de datos entre elementos de red (*Network Elements, NE*).
- Mecanismos de seguridad y administración de la red de telecomunicaciones.

3.4.3. Tipos de enlace de señalización

La estructura de la red SS7 sigue diferentes tipos de conexión entre SPs. Estos enlaces son organizados lógicamente por tipos desde la A a la F, acorde a su uso y todos son idénticos (56 o 64 kbps sobre enlace de datos bidireccionales). La *Figura 3.7* muestra los tipos de enlaces utilizados en SS7 descritos a continuación:

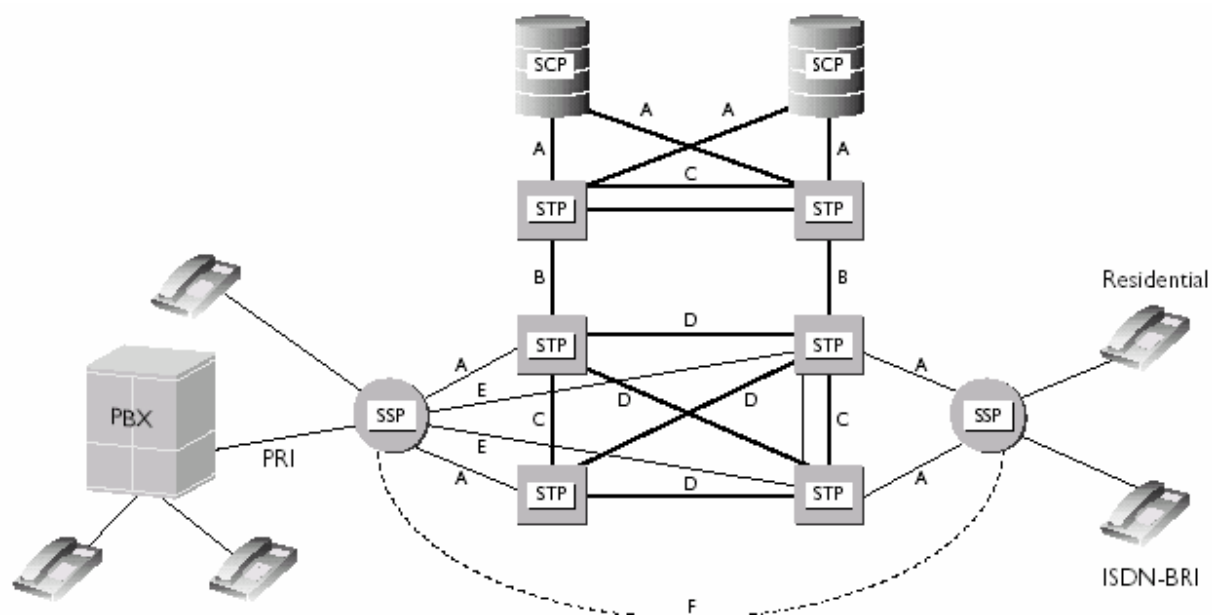


Figura 3.7: Estructura de SS7 [15]

- **ENLACE A:** conecta un punto de señalización origen con un STP y sólo los mensajes originados desde o destinados al punto de señalización final son transmitidos sobre este enlace.
- **ENLACE B:** este tipo de enlace realiza conexiones entre STPs. La distinción entre enlaces B y D es arbitraria y por esta razón los enlaces podrían referirse a B/D.
- **ENLACE C:** es un enlace cruzado que conecta STPs y es usado para mejorar la confiabilidad de la red de señalización. Éste es usado cuando un STP no tiene otra ruta libre para un punto de señalización destino durante la transmisión de datos con un enlace caído.
- **ENLACE D:** es un enlace diagonal que conecta pares de STPs en diferentes niveles de jerarquía. Los STPs secundarios que no pertenecen a la misma red son conectados sobre éste tipo de enlace.
- **ENLACE E:** es un enlace extendido que conecta un SSP a un STP alternativo para proveer un camino de señalización alternativo. No son muy utilizados por su alto costo y un bajo beneficio de un alto margen de degradación.
- **ENLACE F:** es una asociación de enlaces que conecta dos puntos terminales de señalización, por ejemplo; un SSP con un SCP. Estos enlaces no son usados en redes con STP porque ellos desvían la seguridad proveniente de los STPs. En redes sin STP, los enlaces F son directamente conectados a los puntos de señalización.

3.4.4. Implementación de SS7

La implementación de SS7 se compone de dos partes fundamentales para realizar las tareas de señalización:

- **Parte de transferencia de mensajes (MTP):** está formado por tres niveles teniendo una similitud con las características de las primeras tres capas del Modelo OSI, constituye un medio de transporte de mensajes independiente de las partes de usuario. Las funciones de la MTP se llevan a cabo por medio del protocolo del nivel que proporciona servicios a los niveles superiores. La interfaz de servicio con las partes de usuario se describen por medio de primitivas y parámetros, los usuarios de la MTP son la SCCP, parte de usuario de teléfono (*Telephone User Part*, TUP), parte de usuario de datos (*Data User Part*, DUP) e ISUP.
- **Partes de usuario con tareas específicas (UP):** constituyen el nivel 4 del sistema SS7 y controla el establecimiento y la disolución de comunicaciones por canales, la gestión de servicios de valor agregado y las funciones de gestión y mantenimiento de canales. Comprenden en cada caso las funciones, protocolos y codificaciones necesarias para la señalización con el sistema SS7.

3.4.5. Parte de transferencia de mensajes (MTP)

Comprende las recomendaciones de la Q.701 a Q.707 de la ITU-T y permiten a las partes de usuario transmitir mensajes entre sí, independientemente del contenido de cada mensaje. Una UP mediante la MTP envía mensajes que han de transmitirse a otra parte de usuario. La parte de transferencia de mensajes es responsable y debe garantizar que los mensajes emitidos desde un UP lleguen a otro de manera fiable con corrección de errores, chequeo de secuencia y sin pérdidas o duplicidad; tal y como se muestra en la *Figura 3.8*.

Unidades de señalización (*Signaling Unit*, SU)

La MTP transporta mensajes y unidades de señalización de longitud variable. Las unidades de señalización contienen además de la información de señalización e información de control de transferencia, para asegurar el funcionamiento adecuado del enlace de señalización.

Hay tres tipos distintos de unidades de señalización, que se distinguen entre sí por el Indicador de Longitud (*Length Indicator*, LI) que forman parte de las SU, estos son:

- **SU de Mensaje (*Message Signaling Unit*, MSU):** contiene de forma general la información de la señalización ya que interactúa con la MTP y la UP, transporta mensajes de usuarios así como mensajes de gestión de la red. La *Figura 3.9* muestra la trama MSU
- **Unidades de Señalización de Estado del Enlace (*Link Status Signal Unit*, LSSU):** Contienen información para la operación del enlace de señalización.

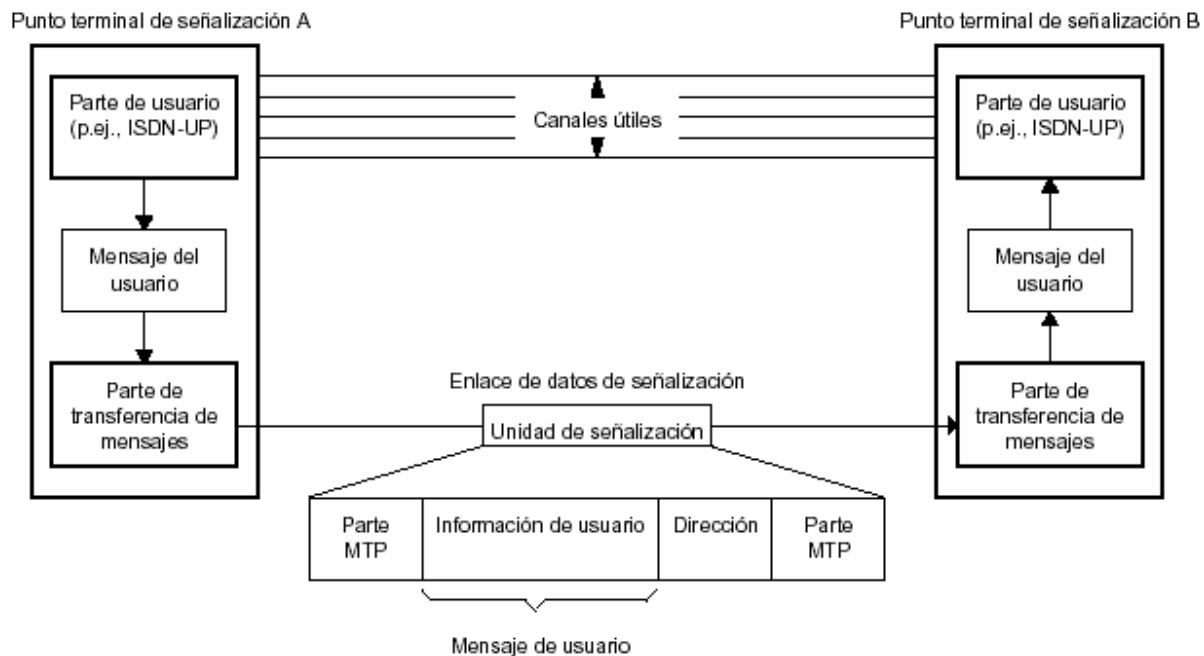


Figura 3.8: Estructura de SS7 [15]

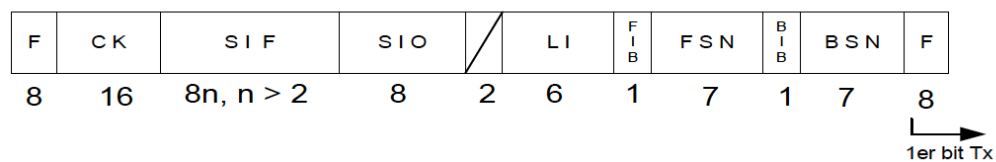


Figura 3.9: Trama MSU [15]

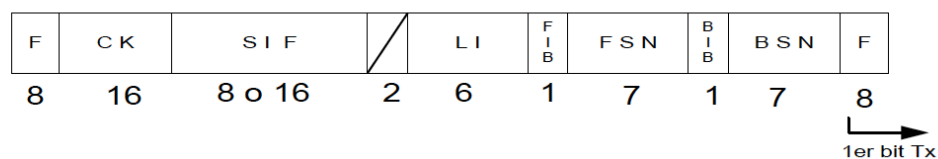


Figura 3.10: Trama LSSU [15]

- Unidades de Señalización de Relleno (*Fill-In Signal Unit, FISU*):** Mantienen el flujo de acuses de recibo y la supervisión de calidad cuando en una de las dos direcciones del enlace de señalización no exista ningún mensaje de usuario disponible para transmisión ya sea MSU o LSSU.

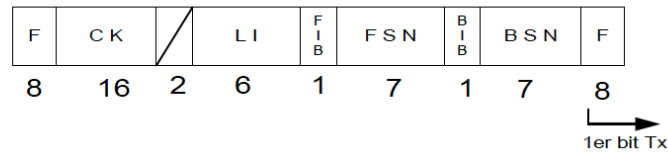


Figura 3.11: Trama FISU [15]

3.4.6. Formato y Código de Mensajes

- **Bandera F (01111110):** Indican el comienzo de una nueva unidad de señalización y el fin de la anterior. Las unidades de señalización tienen longitudes diferentes; sin embargo, cuando está sobrecargado el enlace de señalización pueden transmitirse varias banderas secuencialmente. La bandera se emplea también para la sincronización, las falsas banderas son removidas antes de la transmisión del mensaje adicionando un cero después de cinco números unos.

- **Número Secuencial hacia atrás (*Backward Sequence Number*, BSN):** El número secuencial hacia atrás sirve como portador del acuse de recibo dentro del tratamiento de errores, y a su vez indica la última unidad de señalización recibida.

- **Bit indicador hacia atrás (*Backward Indicator Bit*, BIB):** Usa el procedimiento básico de corrección de errores. El Bit indicador hacia atrás sirve para la corrección de unidades de señalización defectuosas junto con el BSN, de igual manera para el control del número de secuencia.

- **Número secuencial hacia adelante (*Forward Sequence Number*, FSN):** Contiene el número secuencial de la unidad de señalización, el cual se utiliza en el lado de recepción para supervisar la secuencia correcta de las unidades de señalización y como protección contra errores de transmisión. Se dispone de los números 0 a 127.

- **Bit indicador hacia adelante (*Forward Indicator Bit*, FIB):** Se emplea para la corrección de errores y sirve para indicar si una unidad de señalización se transmite por primera vez o se está repitiendo.

- **Indicador de longitud (LI):** Se utiliza para indicar la unidad de señalización. LI distingue tres tipos de unidades de señalización:

- **Octeto de información de servicio (*Service Information Octet*, SIO):** Pertenece a la MSU, se divide en dos subcampos:
 - **Indicador de servicios (*Service Indicator*, SI),** indica el usuario receptor de la señal. La MTP lo usa para distribuir los mensajes al correcto usuario. Se utiliza excepto el 1 y 0 todos los códigos.

LI	Unidad de Señalización
= 0	FISU
= 1 ó 2	LSSU
>2	MSU

Cuadro 3.2: Indicadores de longitud

- **Campo de Subservicio (*Subservice Field, SSF*)**, indica el tipo de red de señalización al que pertenece la MSU (tráfico nacional o internacional) y la prioridad de los mensajes, esto se aplica en condiciones de congestión.
- **Campo de información de señalización (*Signaling Information Field, SIF*)**: El campo de información de señalización solo se presenta en las unidades de señalización de mensaje y contiene, junto con las direcciones, el mensaje de usuario propiamente dicho. La longitud máxima del mensaje de usuario es de 272 octeto, el formato y la codificación del mensaje de usuario se definen por separado para cada parte de usuario. SIF en la MSU es un campo variable.

El código y formato del SIF depende del usuario, y consta de:

- Información de usuario TUP o ISUP
- Etiqueta de enrutamiento con tres campos:
 - Código de punto de destino (*Destination Point Code, DPC*)
 - Código de punto de origen (*Originating Point Code, OPC*)
 - Selección del enlace de señalización (*Signalling Link Selection, SLS*)
- Circuito de voz sobre el que se señala (*Call Instance Code, CIC*)
- **Bits de prueba (*Check Bits, CK*)**: Se generan en el lado de transmisión; en el lado de recepción la parte de transmisión de mensajes puede establecer mediante los bits de prueba, si la unidad de señalización ha sido transmitida sin errores. Según el resultado de la prueba se acusará recibo de la unidad de señalización como libre de errores o defectuosa.
- **Campo de estado (*State Field, SF*)**: En LSSU indica el estado de enlace de señalización y contiene indicadores de estado referentes al enlace de señalización para la sincronización de los sentidos de transmisión y recepción.

3.4.7. Direccionamiento de las unidades de señalización

En el campo de información de señalización SIF se transporta la dirección de las unidades de señalización de mensaje. Esta dirección se divide en:

- Código de punto de destino (DPC)

- Código de punto de origen (OPC)
- Campo de selección del enlace de señalización (SLS)

La red de señalización tiene y asigna para cada punto un código conforme a un plano de numeración, este código de numeración es utilizado por la MTP para el encaminamiento de mensajes. El DPC, indica el SP a donde va el mensaje en la unidad de señalización de mensaje. El OPC lleva la información del SP del que procede el mensaje. La ruta o camino de señalización indica el contenido del campo de selección de enlace de señalización y el campo de selección de enlaces de señalización sirve así para la distribución de la carga (load sharing) en los enlaces de señalización existentes entre dos puntos de señalización.

La información adicional sobre el direccionamiento se lleva en el octeto de información de servicio (SIO). Con el indicador de servicio, la MTP del destino sabe e identifica la parte de usuario a la que va destinado el mensaje. Del indicador de red se desprende, por ejemplo, si se trata de un mensaje para tráfico nacional o internacional. Las unidades de señalización de estado y las de relleno no requieren alguna dirección, porque se intercambian únicamente entre los niveles 2 de MTP adyacentes.

3.4.8. Funciones

La MTP se encarga de la transmisión y recepción de los mensajes, a su vez de la corrección de errores de transmisión, de la gestión de red de señalización y de la sincronización. Todas estas funciones las distribuye en los niveles llamados funcionales 1, 2 y 3.

- **Nivel 1:** Función del Enlace de Datos de Señalización

Este nivel define las características físicas, eléctricas y características funcionales del enlace de señalización de igual manera los equipos de acceso. Define interfaces físicas DS1 (1.544Mbps), E1 (2.048Mbps), V.35 (64kbps), DS0 (64kbps) y DS0A (56kbps). Este nivel constituye el portador para un enlace de señalización. En redes digitales como enlaces de datos de señalización se usa canales de 64 kbps canales, en redes analógicas a través de módems una velocidad binaria de 4,8kbps.

El nivel de enlace de datos de señalización haciendo una comparativa con el modelo OSI constituye la Capa 1, es decir la Capa física.

- **Nivel 2:** Función de Enlace de Señalización

Define las funciones y procedimientos para asegurar que los mensajes se transmitan en un enlace de señalización de manera correcta. Las funciones que se emplea en este nivel son:

- Delimitación de las unidades de señalización mediante banderas
- Supresión de banderas superfluas
- Detección de errores mediante bits de control

- Corrección de errores mediante repetición de unidades de señalización
- Supervisión de la proporción de errores en el enlace de señalización
- Restablecimiento del servicio libre de errores

En general se realiza control de errores, validación de la secuencia de mensajes y corrección de errores. Es importante destacar que cuando un error ocurre sobre un mensaje de señalización, el mensaje se retransmite.

■ Nivel 3: Función de Red de Señalización

Define las funciones de transporte y procedimientos que son comunes e independientes de los enlaces de señalización individuales. Se proporciona mensajes de enrutamiento entre los SP en la red SS7. Se re-enruta el tráfico entre el enlace que falla y SP, se controla de igual manera el tráfico cuando ocurre congestión. De manera general, el nivel 3 (red de señalización) define la interacción entre los distintos enlaces de señalización, abarca dos tareas:

- Tratamiento de mensajes
- Gestión de la red de señalización

Las diversas funciones del nivel 3 interactúan entre sí, así como con funciones de los otros niveles y con las funciones correspondientes de otros puntos de señalización.

3.4.9. Calidad de Señalización de la MPT (Parte Transferencia del Mensaje)

Según la Rec. E.713 de la ITU-T, para determinar el número total de señales en el plano de control en un período de llamada debemos sumar el número de señales originadas por las tentativas de llamadas tratadas en el plano de usuario asociado durante el período de referencia.

■ Parámetros

Para caracterizar el tráfico de llamadas debemos tomar en consideración que existen diversos aspectos que influyen como son:

- La propia generación del control de llamada
- El número de mensajes de señalización que se utilicen y la longitud de los mismos determinan la carga de tráfico de un enlace de señalización. Recordando que para las aplicaciones digitales la velocidad de transferencia de la información es 64 Kbit/seg.

Se adopta el valor de 0.4 Erlangs como la carga de tráfico mínima que se debe garantizar en un enlace de señalización, sin degradación de la calidad del servicio en condiciones anormales de operación.

$$E = \frac{N \cdot K \cdot L \cdot n \cdot (1 + \frac{Z}{100})}{2 \cdot T \cdot V} \text{ (erlangs)}$$

Figura 3.12: Formula Erlangs [15]

3.4.10. Partes de Usuario (UP)

El SS7 permite adaptarse a los requerimientos más diversos, permite también tener una visión de necesidades futuras. Mediante cada parte de usuario se fija las funciones que se van a emplear del MTP a disposición del tipo de usuario. Las partes de usuario especificadas por la ITU-T son:

- Parte de usuario de telefonía (TUP)
- Parte de usuario de la red digital de servicios integrados (ISUP)
- Parte de control de la conexión de señalización (SCCP)
- Parte de aplicación de las capacidades de transacción (*Transaction Capabilities*, TC)

Parte de usuario de telefonía (TUP)

Define el control de llamadas internacionales las funciones de señalización para el inicio y fin de una llamada básica. TUP fue una implementación más temprana de SS7, no se utiliza en para aplicaciones de tipos de datos.

Parte de usuario de la red digital de servicios integrados (ISUP)

Recomendaciones Q.761 a Q.767 de la ITU-T definen los protocolos y procedimientos usados para iniciar, administrar y finalizar el tráfico de circuitos que llevan llamadas de voz y datos entre SSPs sobre la PSTN. Las llamadas que se originan y concluyen en el mismo *switch* no usan señalización ISUP.

Los procedimientos de gestión utilizan mensajes ISUP para controlar el flujo de las llamadas, cada procedimiento tiene definido un diagrama de transición que indica los estados de una llamada y los mensajes que puede procesar en dicho estado. Bajo estas definiciones se establece el protocolo que permite realizar conexiones entre centrales telefónicas con señales de control mediante mensajes ISUP.

En entornos de producción con gran volumen de llamadas, estos mensajes pueden generarse de manera irregular teniendo como consecuencia serios problemas a nivel de circuitos en las centrales telefónicas, de igual forma en los procesos de tasación de la llamada ya que su comportamiento es completamente irregular.

El análisis de estos mensajes y la detección de escenarios irregulares es la motivación principal de este trabajo de investigación, el desarrollo de la aplicación asociada se basa en una máquina de estados que busca englobar las transiciones y estados que son observados en llamadas regulares, de esta forma; ante cualquier comportamiento irregular el sistema es capaz de detectar las llamadas que requieren revisión.

Estructura de los mensajes ISUP

En un mensaje ISUP, el SIF contiene la siguiente etiqueta de ruteo: 14 bit (*American National Standards Institute, ANSI*) o 12 bits (ITU-T) en el código de identificación del circuito CIC.

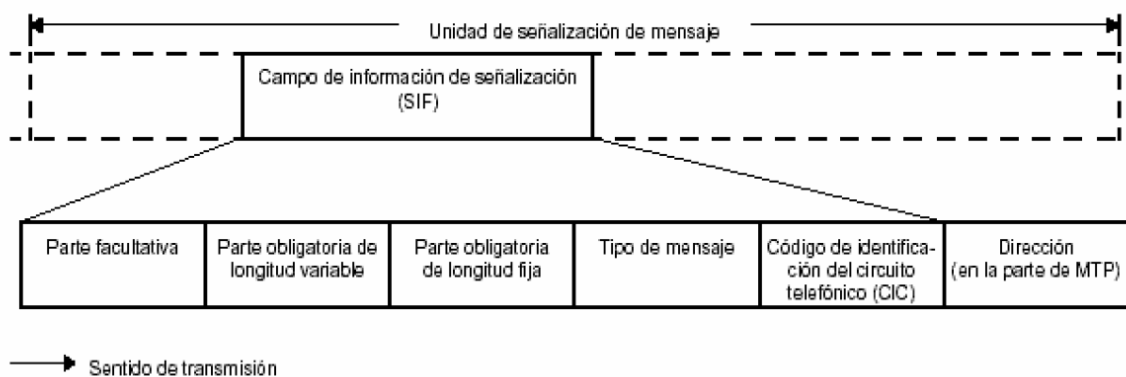


Figura 3.13: Mensaje ISUP [15]

- **La dirección**, consta del DPC, OPC y selección de enlace de señalización.
- **El código de identificación de circuito telefónico (CIC)**, asigna el mensaje a un determinado canal útil, a cada canal útil se le asigna permanentemente un código de identificación de circuito.
- **El tipo de mensaje**, define la función y el formato de un mensaje UP. Entre otros tipos de mensaje que se emiten tenemos:
 - **Mensaje de Dirección Inicial (*Initial Address Message, IAM*)**
Contiene la información para el inicio de una llamada, es el primer mensaje que se envía a la central destino durante el establecimiento de la comunicación. Se utiliza para ocupar un canal útil y contiene toda la información necesaria para el direccionamiento hasta la central de destino.
 - **Mensaje de Dirección Subsecuente (*Subsequent Address Message, SAM*)**
Transporta las cifras marcadas aún no contenidas en el IAM.

- **Mensaje Completo de Dirección (*Address Complete Message, ACM*)**
Indica a la central emisora que se alcanzó la central de destino. La parte receptora escucha el timbre del teléfono sobre el tráfico de voz generada por el switch destino.
- **Mensaje de Respuesta (*Answer Message, ANM*)**
Cuando el receptor contesta, el switch destino concluye la emisión de tonos de timbrado y envía un ANM al switch origen. El switch original inicia la facturación, luego verifica que la línea del usuario que llama se conecte al tráfico reservado.
- **Mensaje de Progreso de Llamada (*Call Progress Message, CPG*)**
Es enviado en cualquier dirección durante la fase de puesta en marcha o activa de la llamada, lo que indica que un evento, el cual es de importancia, y debe ser transmitido a la de acceso de origen o de terminación, se ha producido.
- **Mensaje de Respuesta de Circuito de Grupo (*Circuit Group Response Message, CQR*)**
Enviado en respuesta a un mensaje de grupo de consulta de Circuito / CIC para indicar el estado de todos los circuitos / CICs en un rango particular.
- **Mensaje de Reconocimiento de restablecimiento de Circuito de Grupo (*Circuit Group Reset Acknowledgement Message, GRA*)**
Es enviado en respuesta a un mensaje de Restablecimiento de circuito / CIC Grupo indicando que el grupo requerido de circuitos / CICs se ha restablecido. El mensaje también indica el estado de mantenimiento de bloqueo de cada circuito / CIC.
- **Mensaje de Confusión (*Confusion Message, CFN*)**
Es enviado en respuesta a cualquier mensaje (distinto de un mensaje de confusión), si el nodo no reconoce el mensaje o detecta una parte del mensaje como siendo no reconocido.
- **Mensaje de Conexión (*Connect Message, CON*)**
Es enviado en dirección final indicando que todas las señales de dirección necesarios para el encaminamiento de la llamada a el abonado llamado ha sido recibida y que la llamada ha sido contestada.
- **Mensaje de Continuidad (*Continuity Message, COT*)**
Es enviado en la dirección inicial que indica si hay o no continuidad en el circuito anterior o anteriores, así como del circuito seleccionado para el siguiente intercambio, incluyendo la verificación de la ruta de comunicación a través del intercambio con el grado especificado de fiabilidad.
- **Mensaje de Rechazo de servicio (*Facility Reject Message, FRJ*)**
Es enviado en respuesta a un mensaje de solicitud de servicio para indicar que la solicitud de establecimiento ha sido rechazada.
- **Mensaje de Información (*Information Message, INF*)**
Es enviado para transmitir información en asociación con una llamada, que puede haber sido solicitado en un mensaje de solicitud de información.

- **Mensaje de Solicitud Información (*Information Request Message, INR*)**
Es enviado por un nodo para solicitar información en asociación con una llamada.
 - **Mensaje Información de Usuario a Usuario (*User to User Information Message, USR*)**
Se utiliza para el transporte de señalización de usuario a usuario independiente de los mensajes de control de llamada.
 - **Mensaje de Transferencia Inicial (*Forward Transfer Message, FOT*)**
Un mensaje enviado en la dirección hinicial en semi llamadas automáticas cuando el operador central internacional de salida quiere la ayuda de un operador en la central internacional. El mensaje normalmente servirá para traer un operador de asistencia en la llamada si la llamada se configura automáticamente en el intercambio.
 - **Mensajes de Resumen (*Resume, RES*) o Suspensión (*Suspend, SUS*)**
RES es enviada en uno u otro sentido indicando que la llamada o el abonado llamado, después de haber sido suspendido, se vuelve a conectar. SUS se envía en cualquier dirección indicando que la llamada o el abonado llamado ha desconectado temporalmente.
 - **Mensaje de Liberación (*Release Message, REL*)**
Inicia la disolución de la comunicación por el canal útil e indica la causa de la disolución. Las comunicaciones por canales útiles establecidas sin éxito también se liberan con REL, que en este caso contiene también la información de por qué no tuvo éxito el establecimiento de la comunicación.
Existen 3 tipos de causa de disolución de llamada:
 - CAUSE = 16, Cuando tanto el emisor como el receptor concluyen la llamada.
 - = 17, Cuando la línea del abonado destino esta ocupada.
 - = 34, Cuando el canal no esta habilitado.
 - **Mensaje de Liberación Completa (*Release Complete Message, RLC*)**
Con el RLC se indica el fin de la interconexión de un canal útil y se confirma la recepción del REL. Una vez transmitido o recibido el RLC, el canal útil se encuentra liberado y disponible para establecer una nueva comunicación. Se indica el fin de la llamada y de la tarificación.
 - **Mensaje de Bloqueo (*Blocking Message, BLO*)** Se utiliza para bloquear un canal útil.
 - **Mensaje de Desbloqueo (*UnBlocking Message, UBL*)** Se utiliza para desbloquear un canal útil.
- **La parte obligatoria de longitud fija** del mensaje ISUP contiene los parámetros que son indispensables para un cierto tipo de mensajes y que tienen una longitud fija.
 - **La parte obligatoria de longitud variable** del mensaje ISUP contiene parámetros de longitud variable.
 - **La parte facultativa**, especifica cuáles parámetros pueden transmitirse en esta parte del mensaje, éstos pueden ser parámetros de longitud fija o variable.

Control de procedimiento de señalización

Existen una serie de procedimientos y primitivas definidas en la implementación de SS7 para el control del flujo de una llamada con mensajes ISUP, todos definidos por la ITU-T en las recomendaciones Q.764 del protocolo.[16]

El procedimiento de señalización de la parte usuario UP está dividido en bloques funcionales principales descritos a continuación y mostrados en la *Figura 3.14*:

- **Control de procedimiento de señalización (*Signalling Procedure Control (SPRC)*):** proporciona los procedimientos para el envío de mensajes de la UP al nivel 3 (SCCP o MTP) para su distribución hacia otros bloques funcionales de la UP.
- **Control de tratamiento de llamada (*Call Processing Control (CPC)*):** proporciona los procedimientos de control de la llamada para prestar el servicio básico con conmutación de circuitos, según la petición del usuario.
- **Control de supervisión de circuito (*Circuit Supervision Control (CSC)*):** proporciona los procedimientos para el control de supervisión del circuito para fines de mantenimiento y para la recuperación tras situaciones anormales.

Como se puede observar en la *Figura 3.14*, el control del procesamiento de llamadas se realiza en unidades de control diferentes para llamadas entrantes o salientes respectivamente, los mensajes ISUP funcionan como eventos que disparan transiciones entre estados definidos en el protocolo para establecer comunicaciones. Las *Figuras 3.15 3.16* muestran el diagrama de transición de estados para una llamada entrante y saliente respectivamente.

A partir de estas unidades de control se pueden definir algoritmos que basados en los mensajes ISUP que determinen si una llamada tiene un comportamiento irregular para detectar fallos en la implementación del un servicio. Este trabajo de investigación se fundamenta en las definiciones antes descritas para el diseño e implementación de una máquina de estados global (5.15) que a partir de la lectura de tramas ISUP pueda detectar patrones irregulares en el flujo de mensajes de una llamada.

Parte de control de la conexión de señalización:

Posee una función de direccionamiento propia, proporciona funciones adicionales a la MTP para soporte de servicios de redes orientadas y no orientadas a conexión, y lo vinculado con la Traslación del Título Global (*Global Title Translation, GTT*): número que se traslada a un DPC y a un subsistema numérico para mejorar el ruteo y la liberación del *Open Settlement Protocol, OSP*, y de igual manera el código de punto destino (se transmite al MTP que posteriormente efectuará el enrutamiento). Esta parte de control hace referencia a las Recomendaciones Q.711 a Q.716 de la ITU-T. [16]

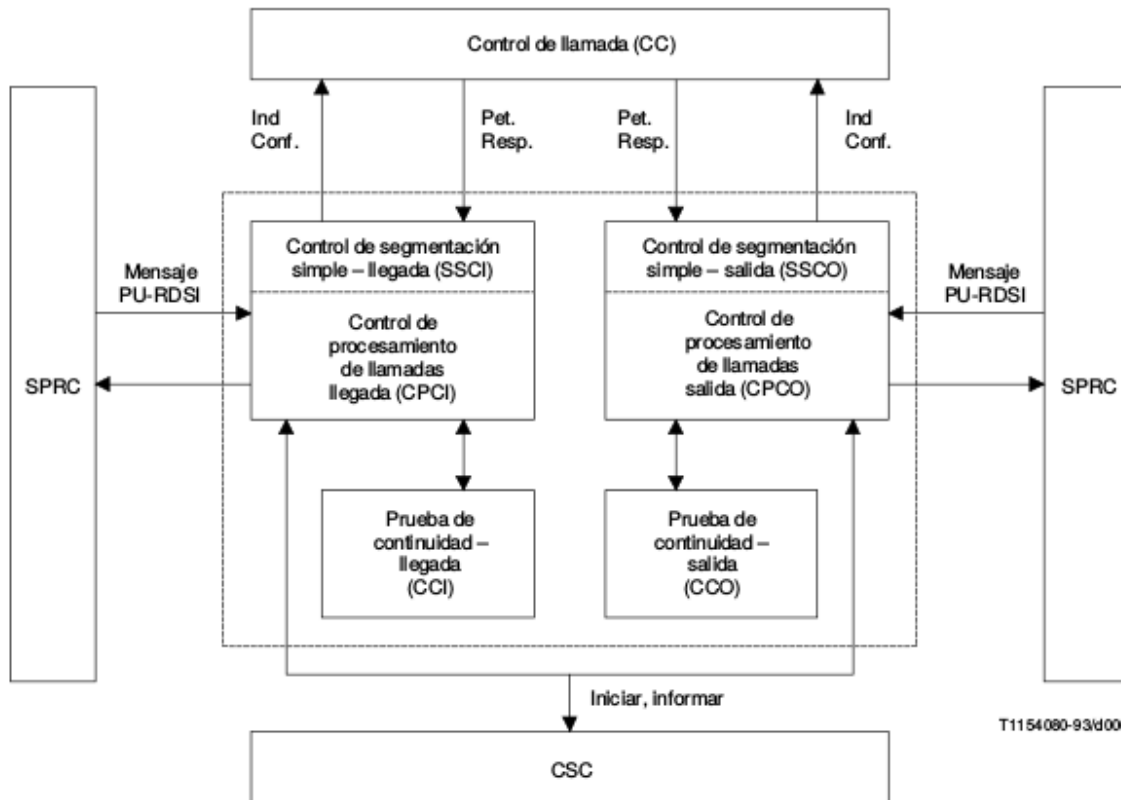


Figura 3.14: Diagrama de bloques funcionales del SPRC [16]

La Parte de control de la conexión de señalización (SCCP) proporciona subsistemas numerados permitiendo que los mensajes se direccionen hacia aplicaciones específicas o subsistemas en puntos específicos de señalización. Se utiliza como la capa de transporte para servicios de TCAP y visto desde la MTP es un usuario con indicador de servicio propio, La combinación de la SCCP con la MTP se denomina parte de servicio de red (*Network Service Part, NSP*).

■ **Estructura del mensaje SCCP** Se encuentra estructurado de la siguiente manera:

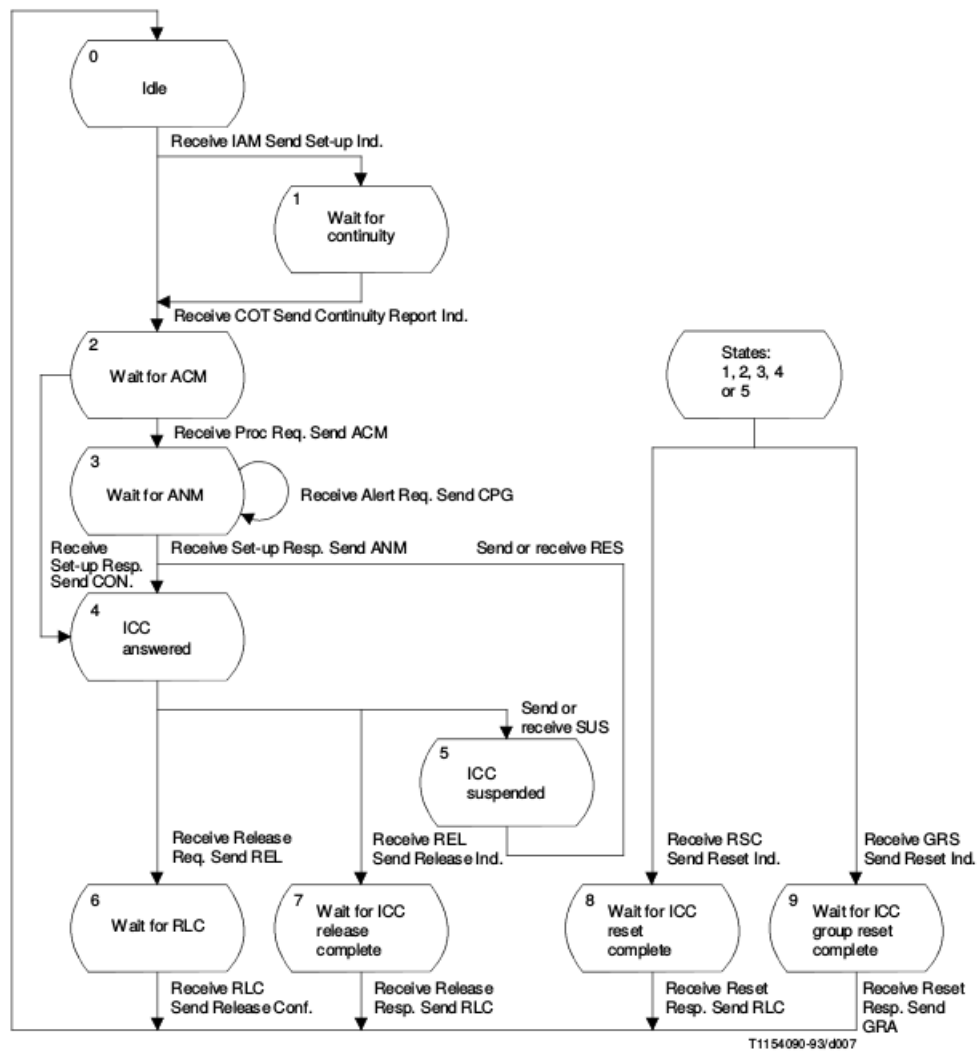
- **Dirección**, especificada en la parte de transferencia de mensajes.
- **Tipo de mensaje**, define la función y el formato del mensaje SCCP.

Dependiendo del tipo de transporte de mensajes se usa:

Para enlaces orientados a no conexión:

- **Datos de Unidad (*Unit Data, UDT*)**

Con el mensaje UDT se envían mensajes SCCP a un destino; se usa con las clases



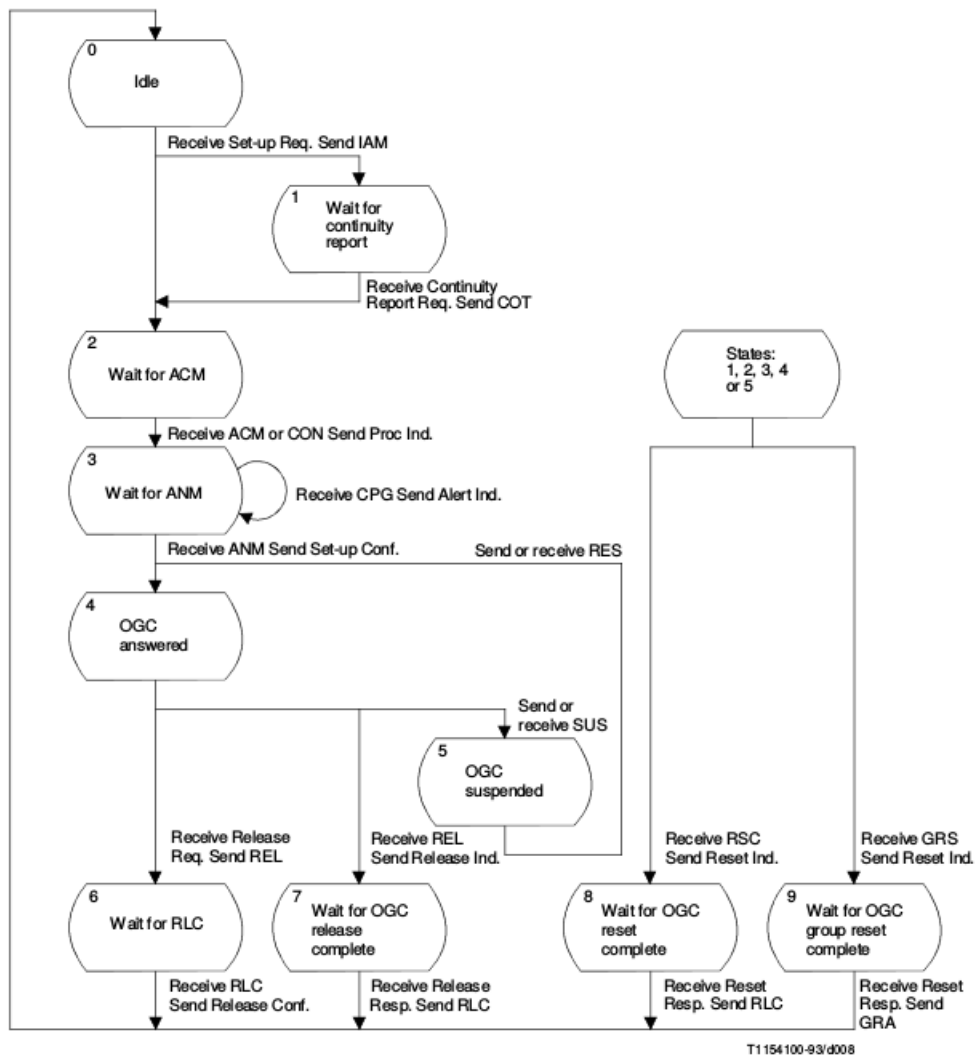
NOTAS

- 1 Un SGM o UMT recibido en cualquier estado desde SPRC, se envía a SSCI en una indicación SGP o UMT.
- 2 Una petición SGP o UMT recibida en cualquier estado desde SSCI, se envía a SPRC en un SGM o UMT.

Figura 3.15: Diagrama de transición de estado para CPC – Llegada (CPCI) [16]

de protocolo 0 y 1.

- o **Servicio de datos de unidad (*Unit Data Service, UDTS*)**
Se utiliza para indicar a la SCCP que esta transmitiendo que no se pudo enviarse el mensaje; se usa con las clases de protocolo 0 y 1.
- o **Datos de unidad ampliados (*Extended Unit Data, XUDT*)**
Se transmite información de señalización en un modo sin conexión; se permiten parámetros opcionales (para la segmentación).
- o **Servicio de datos de unidad ampliado (*Extended Unit Data Service,***



NOTAS

- 1 Un SGM o UMT recibido en cualquier estado desde SPRC, se envía a SSCO en una indicación SGP o UMT.
- 2 Una petición SGP o UMT recibida en cualquier estado desde SSCO, se envía a SPRC en un SGM o UMT.

Figura 3.16: Diagrama de transición de estado para CPC – Salida (CPCO) [16]

XUDTS)

Permite indicar al punto de origen de señalización que XUDT no pueda alcanzar el destino.

Para el transporte de mensajes orientados a conexión:

- Petición de conexión (*Connection Request, CR*)

Este mensaje indica al punto destino de la señalización que debe establecerse un enlace de señalización virtual, puede transmitirse como mensaje propio o conjuntamente con otro mensaje según la clase de protocolo que se use.

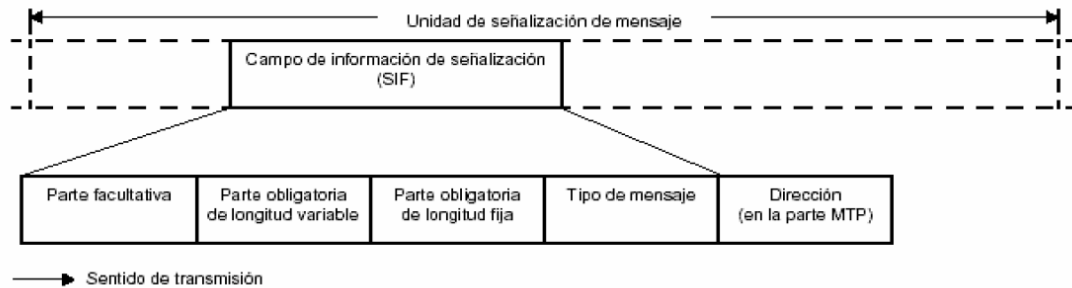


Figura 3.17: Estructura de SCCP [15]

- Confirmación de conexión (*Connection Confirmation, CC*)
Se confirma el establecimiento del enlace de señalización virtual del destino.

Mensajes que se emplean en la disolución del enlace de señalización orientado a conexión:

- Disuelto (*Released, RLSD*)
Indica el inicio de la disolución del enlace de señalización, puede ser enviado desde ambos lados.
- Disolución completa (RLC)
Indica que recibió el mensaje de disolución del enlace de señalización.

Tipos de mensajes del transporte de mensajes:

- Forma de datos 1 (*Data Form 1, DT1*)
Utilizado en la clase de protocolos 2, después de establecer una conexión de señalización, con el mensaje DT1 se pueden transmitir mensajes SCCP en ambas direcciones.
 - Forma de datos 2 (*Data Form 2, DT2*)
Empleado en la clase de protocolos 3, después de establecer una conexión de señalización y con la confirmación de la recepción de los mensajes por parte del lado destino. Se le utiliza en ambas direcciones.
- **Parte obligatoria de longitud fija**, del mensaje SCCP conlleva los parámetros que no se pueden omitir en determinados tipos de mensajes y cuya longitud debe ser fija. Por ejemplo: la referencia local, la clase de protocolo empleada para el transporte de mensajes.
 - **Parte obligatoria de longitud variable**, tiene parámetros de longitud variable. Por ejemplo: el número telefónico del abonado llamado, el código del usuario SCCP (ISUP, TCAP, etc.)
 - **Parte facultativa del mensaje SCCP** lleva todos los tipos de parámetros que pueden

presentar los diferentes tipos de mensajes, puede ser de longitud variable y también fija. Ejemplo: el número telefónico del abonado emisor, mensajes de usuario a ser transportados, etc.

Clases de Protocolos

Para cada tipo de transporte de mensajes, la SCCP ofrece dos clases de protocolo:

Para el transporte de mensajes orientado a no conexión, se tiene las clases de protocolos 0 y 1

- **Clase de protocolo 0**

Los mensajes SCCP se transportan de manera individual e independientemente unos de otros por la parte de transferencia de mensajes.

- **Clase de protocolo 1**

Los mensajes SCCP se transportan en una secuencia definida por el usuario.

Para el transporte de mensajes orientado a conexión, se tiene las clases de protocolos 2 y 3.

- **Clase de protocolo 2**

Esta clase de protocolo garantiza una correcta secuencia de mensajes, entre los puntos finales de la SCCP se transmiten recíprocamente los puntos de origen propios. Se asigna además el enlace de señalización correspondiente, posteriormente se efectúa el intercambio de mensajes.

- **Clase de protocolo 3**

Realiza las mismas funciones que la clase de protocolo 2, incluyendo de manera adicional el apoyo de control de errores.

Procedimiento de Señalización

Cuando se trabaja con la ISUP se tiene el siguiente procedimiento de establecimiento del enlace de señalización. La SCCP de un usuario origen, recibe la petición de establecimiento de enlace de señalización, envía un mensaje CR que incluye: la referencia local e indicaciones sobre la clase de protocolo a la SCCP del punto terminal de señalización destino.

Para el **transporte del mensaje CR**, se cuenta con dos métodos en la clase de protocolos 2:

- **Método normal**

Se envía al mensaje CR como mensaje autónomo al punto de señalización opuesto. En la clase de protocolo 3 se utiliza solo este método.

- **Método incorporado**

En este método no se determina el código del punto de destino, el mensaje CR se integra en otro mensaje de la ISUP.

Cuando se recibe un mensaje CR, tanto la SCCP del lado de recepción como la de transmisión poseen una referencia local, la misma que se envía recíprocamente a ambos lados junto con el código de señalización del lado de recepción, originado en el mensaje CC. Obteniendo que las dos centrales conocen sus respectivos números de código y la referencia local, y se pueden direccionar mensajes para el proceso directamente al punto de señalización opuesto.

Capacidades de transacción TC

Abarca las recomendaciones de la Q.771 a Q.775 de la ITU-T. Las TC son un protocolo de señalización de aplicaciones (capa 7 de OSI) y se comunican directamente con la SCCP. Usan el transporte de mensajes en enlaces orientados a no conexión.

Provee los medios para establecer la comunicación entre dos SPs relativa a conexión de no-circuitos, y apoyan el intercambio de información entre usuarios de distintos nodos de red del sistema SS7. En ISUP existen servicios adicionales de ISDN como (*Call Completion to a Busy Subscriber, CCBS*), (*Completion of Calls on No Reply, CCNR*).

TCAP soporta los cambios relativos de no-circuitos entre aplicaciones a través de la red SS7 usando los servicios orientados a no conexión del SCCP como un transporte. Las preguntas y respuestas entre SSP y SCP son llevadas en los mensajes TCAP. TC puede constar de uno o varios componentes de mensajes individuales, los que pueden ser: Una llamada para una acción a realizar por el usuario TCAP en el nodo de red distante, una interrogación de datos o de estado, o la respuesta a una llamada de acción o a una interrogación.

El usuario TC manda los componentes de mensajes a las TC individualmente, esto se identifica con un mismo código de diálogo, el usuario inicia el envío con una solicitud especial. las TC proceden a agrupar en un mensaje global todos los componentes de mensajes que tengan el mismo indicador de diálogo y los transfiere al SCCP para su transporte al destino deseado.

El usuario TC utiliza dos maneras para la transferencia de mensajes:

- **Diálogo no estructurado**

El usuario transfiere a las TC componentes individuales de mensajes, estos a su vez, transmiten al destino deseado de uno en uno o en grupo. En este caso no es posible una correlación entre el componente de mensaje transmitido y la respuesta.

- **Diálogo estructurado**

El usuario establece un diálogo con el usuario TC remoto y se intercambia mensajes durante el diálogo. Aquí es posible una correlación directa entre el componente de mensaje y la respuesta.

Estructura del mensaje TC

La estructura del mensaje TC es la siguiente:

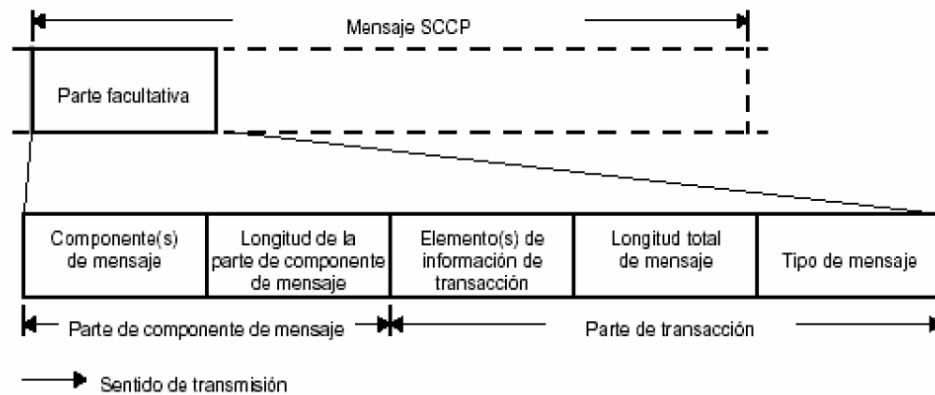


Figura 3.18: Estructura de TC [15]

- **Tipo de mensaje:** Especifica la función del mensaje que tiene el usuario TC, encontrando los siguientes:
 - **Begin:** Empleado en el diálogo estructurado para comenzar un diálogo.
 - **Continue:** Permite intercambiar tareas, mensajes y datos durante el diálogo actual.
 - **End:** Permite concluir un diálogo, se emplea en el diálogo estructurado.
 - **Abort:** Permite interrumpir un diálogo después de un error.
- **La longitud total del mensaje:** Abarca la cantidad de octetos del mensaje.
- **Elemento(s) de información de transacción:** Existen solamente en el diálogo estructurado, éstos pueden contener:
 - **Indicador de transacción:** Las TC de origen asignan el indicador al diálogo y lo envían a las TC destino, así las TC puedan asignar los mensajes entrantes al diálogo respectivo.
 - **Causa de interrupción:** Por ejemplo: un tipo de mensaje o un indicador de transacción desconocidos.
- **Componente(s) de mensaje:** Posee uno o varios componentes de mensaje, tiene una estructura fija que comprende los siguientes campos:
 - **Tipo de componente**
 - *Invoke:* Invoca o realiza una acción.
 - *Return Result:* transmite un resultado.
 - *Return Error:* informa un error en la terminación de una acción.

- *Reject*: rechaza un componente de mensaje.
- **Longitud de Componente**: Contiene la cantidad de octetos de los componentes de mensaje.
- **Elemento de información**:
 - *Etiqueta "Invoke"*: hace referencia a asignar resultados a la invocación de acción correcta.
 - *Operationscode*: indica el tipo de acción que ha de efectuarse.
 - *Código de error*: indica la razón por la cual no puede ejecutarse una acción invocada.
 - *Código de problema*: indica la razón por la cual se rechaza un componente de mensaje.
 - *Parámetro*: contiene información de usuario suplementaria.

Procedimiento de señalización

Existen dos procedimientos de señalización, el diálogo no estructurado y el diálogo estructurado.

El **diálogo no estructurado**. El usuario TC envía a las TC destino los componentes de mensaje que se van a transmitir con una petición de tipo unidireccional, esta petición contiene la información de direccionamiento y la identificación del indicador de diálogo. Esto se identifica con un indicador de diálogo común. Las TC envían este mensaje a la SCCP junto con la información de direccionamiento correspondientes para el transporte al destino deseado. Las TC del lado de recepción reciben los mensajes unidireccionales y los envía al usuario TC direccionado.

El **diálogo estructurado** permite a un usuario TC entablar un diálogo, intercambiar mensajes y terminar el diálogo con otro usuario TC. El usuario TC envía para un nuevo dialogo la petición de begin a otro usuario TC que contiene la información de direccionamiento, el indicador de diálogo de los componentes de mensaje que deben intercambiarse durante el diálogo y el indicador de transacción que sirve para asignar los componentes de mensaje al diálogo correspondiente. Esto se envía desde las TC a la SCCP y esta lo transmite al destino deseado. Las TC del destino reciben el mensaje de comienzo y comunican al usuario TC direccionado el inicio de un nuevo diálogo.

El diálogo puede continuar por ambos usuarios TC. Así, el usuario TCC respectivo envía a las TC un mensaje CONTINUE.

Para finalizar el diálogo se puede:

- Los usuarios determinan previamente el fin del diálogo: no se envía mensaje de fin.
- Un usuario TC se ve forzado a interrumpir el diálogo y envía un mensaje de fin al otro extremo del diálogo.
- Un usuario TC rompe el diálogo debido a un error y envía un mensaje de aborto al TC destino.

Capítulo 4

Método de Desarrollo

En éste capítulo inicialmente se describirán los tipos de investigación definidos según Hernández, Fernández y Baptista (1997)[8], con la finalidad de determinar dentro de cuál de ellas se enmarca el desarrollo del trabajo presentado, seguidamente se presentará el método de desarrollo a seguir, describiendo cada una de sus fases y los resultados obtenidos.

4.1. Tipos de Investigación

Existen varios tipos de investigación en las cuales se pueden clasificar los trabajos o interrogantes de carácter científico en función de sus objetivos y razón de ser. Entre los tipos de investigación se pueden describir brevemente los siguientes:

- **Investigación Exploratoria:** tienen por objeto esencial familiarizarnos con un tópico desconocido o poco estudiado o novedoso. Esta clase de investigaciones sirven para desarrollar métodos a utilizar en estudios más profundos.[8]
- **Investigación Descriptiva:** sirven para analizar cómo es y se manifiesta un fenómeno y sus componentes, un ejemplo de este tipo de estudio puede ser el comportamiento de una población ante el aprovechamiento de un recurso compartido.[8]
- **Investigación Correlacional:** estos estudios pretenden ver cómo se relacionan o vinculan diversos fenómenos entre sí (o si no se relacionan).[8]
- **Investigación Explicativa:** los estudios explicativos buscan encontrar las razones o causas que provocan ciertos fenómenos, en lo cotidiano se puede observar que este tipo de investigación es empleada por organismos de seguridad al investigar un suceso en determinada población.[8]
- **Investigación Proyectiva:** también conocida como proyecto factible; consiste en la investigación, elaboración y desarrollo de una propuesta de un modelo operativo viable para solucionar problemas, requerimientos o necesidades de organizaciones o grupos sociales. La

propuesta que lo define puede referirse a la formulación de políticas, programas, tecnologías, métodos o procesos, que sólo tienen sentido en el ámbito de sus necesidades.[21]

Un proyecto factible, como su nombre lo indica, tiene un propósito de utilización inmediata, la ejecución de la propuesta está orientado a resolver un problema planteado o a satisfacer las necesidades en una institución.

De las definiciones anteriores se deduce que, un proyecto factible consiste en un conjunto de actividades vinculadas entre sí, cuya ejecución permitirá el logro de objetivos previamente definidos en atención a las necesidades que pueda tener una institución o un grupo social en un momento determinado. Es decir, la finalidad del proyecto factible radica en el diseño de una propuesta de acción dirigida a resolver un problema o necesidad previamente detectada en el medio.[21]

De las definiciones anteriormente presentadas se puede concluir que el desarrollo del presente Trabajo Especial de Grado (TEG) se enmarca dentro de un tipo de investigación de Proyecto Factible, debido a que en éste trabajo se desarrolla un Sistema que permitirá determinar llamadas telefónicas irregulares en centrales telefónicas que implementan el Sistema de Señalización 7 (SS7), dando con ello solución a la problemática presentada en el *Capítulo 1*. En la siguiente sección se describirá el método de desarrollo empleado para la implementación del Sistema propuesto.

4.2. Método de Desarrollo

En ésta sección se propone implementar un método de desarrollo compuesto por un conjunto de fases las cuáles fueron ejecutadas con la finalidad de llevar a cabo el desarrollo del proyecto. A continuación se describen las fases que componen el método:

Fase I - Estudio de la situación actual y determinación de requerimientos: en esta fase se realiza el levantamiento de información con la idea de tener una visión holística del problema a resolver, con la finalidad de llevar a cabo la determinación de requerimientos del Sistema, estudiar y analizar el SS7, y sus procedimientos de control y señalización para la capa ISUP.

Fase II - Análisis de las estructuras de datos: en esta fase se realiza el análisis de las capturas de tráfico contenidas en archivos de texto plano, cada entrada contiene información relevante sobre las llamadas y por ende los elementos de red que interactúan en la comunicación. Por esta razón el desarrollo de estructuras de datos es fundamental para realizar la carga del archivo en la base de datos y su posterior análisis.

Fase III - Diseño del Analizador de Tramas: esta fase se enfoca en las funcionalidades requeridas para el procesamiento de las trazas y despliegue de los resultados. Para el análisis y procesamiento de las entradas se realiza el diseño de la máquina de estados del sistema. Para la aplicación se diseñan interfaces que permiten la subida del archivo al servidor web, su procesamiento y almacenamiento en base de datos y el diseño de las vistas, y plantillas para mostrar los

resultados con el fin de generar un entregable para el usuario final.

Fase IV - Implementación de la interfaz web: en esta fase se realiza el desarrollo de las vistas que obtendrá el usuario en la aplicación, se hace especial énfasis en la integración con los controladores de cada modelo de datos buscando mantener la capa de presentación libre de código asociado a la lógica del sistema.

Fase V - Pruebas de rendimiento y confiabilidad: en esta fase se realizaran pruebas para medir el desempeño de la aplicación con distintos archivos de entrada para cada escenario con la finalidad de validar los resultados obtenidos y su confiabilidad.

Capítulo 5

Marco Aplicativo

En este capítulo se describe el análisis general del sistema y el diseño de la solución. A nivel de aplicación serán definidos los requerimientos funcionales y no funcionales, el modelo de la base de datos y la arquitectura del sistema.

También se describirá el proceso de desarrollo y la implementación del Analizador de Trazas ISUP de SS7 explicando las tecnologías empleadas y los componentes del entorno de desarrollo. A nivel de implementación se hará reseña de los elementos que se conjugan para el despliegue, la instalación de dichos componentes, sus archivos de configuración y las características del servidor que ejecuta la Aplicación Web junto con el desarrollo de la metodología descrita en el *Capítulo 4.2*.

5.1. Análisis General del Sistema

Con el fin de tener una idea global de la solución a implementar se realiza un análisis detallado de las trazas capturadas en centrales telefónicas que presentan problemas en la interconexión. Los datos están en archivos de texto plano que contienen información asociada a la señalización generada cuando el Suscriptor A levanta el teléfono para comunicarse con el Suscriptor B.

El proceso para establecer una llamada está sujeto al intercambio de mensajes ISUP de acuerdo a los eventos ocurridos en la comunicación. El intercambio de estos mensajes es mostrado en el diagrama de secuencia de la *Figura 5.1* asociado a la comunicación entre dos Suscriptores.

Esta secuencia de mensajes ISUP describen el proceso regular para establecer una llamada bajo el protocolo definido en el estándar de SS7. Al momento que un *Abonado A* levanta el teléfono para marcar se envía un mensaje *IAM* que le permite reservar un circuito para establecer la comunicación con el *Abonado B*. Cuando el teléfono del *Abonado B* repica se envía un mensaje de vuelta para indicar que se espera respuesta, si la llamada es contestada se envía un mensaje *ANN* que establece la llamada entre ambos abonados.

Por cada evento dentro de la comunicación se generan mensajes de señalización que permiten controlar el flujo de la llamada e implementar el servicio a gran escala. Los elementos a analizar

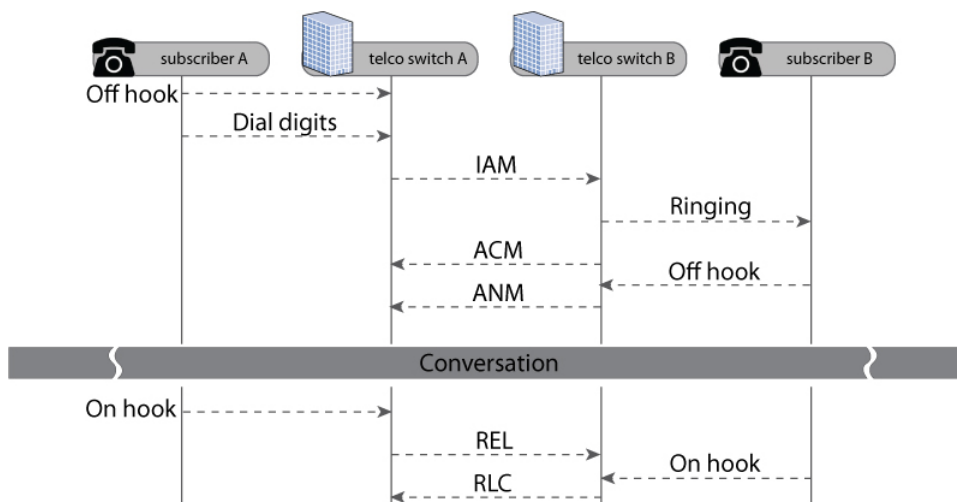


Figura 5.1: Flujo de mensajes ISUP

están representados en entradas de archivos de texto que contienen los siguientes campos: *Número Serial*, *Número de enlace*, *Dirección*, *Tiempo Absoluto*, *Time*, *H1H0*, *CIC/SLC*, *OPC*, *DPC* y la *Traza del Mensaje*. Con esta premisa se realiza la búsqueda de patrones irregulares en los archivos de tramas capturadas, mostrados en la *Figura 5.2*. [2]

Line	Serial Number	Link number	Direction	Absolute time	Time	H1H0	CIC/SLC	OPC	DPC	Message traced
1	00000001	24	Send	Fri 2012-04-13 14:29:00	61835	REL	0000056	3A2D	1C41	0D 05 41 5C 8B
2	00000002	24	Send	Fri 2012-04-13 14:29:00	61843	IAM	0000099	3A2D	1C41	25 05 41 5C 8B
3	00000003	24	Receive	Fri 2012-04-13 14:29:01	61896	RLC	0000056	1C41	3A2D	09 05 2D 7A 10
4	00000004	24	Send	Fri 2012-04-13 14:29:01	61925	COT	0000099	3A2D	1C41	09 05 41 5C 8B
5	00000005	24	Receive	Fri 2012-04-13 14:29:01	61947	ANM	0000086	1C41	3A2D	09 05 2D 7A 10
6	00000006	24	Receive	Fri 2012-04-13 14:29:02	61967	ACM	0000044	1C41	3A2D	0B 05 2D 7A 10

Figura 5.2: Captura de Tramas ISUP

En el proceso de carga del archivo de entrada se toman los datos de las trazas para convertirlos en los objetos asociados al análisis de la llamada (canal, trama, llamada), siguiendo el modelo descrito en el diseño de la base de datos mencionado en el (*Capítulo 5.3.2.*)

Para realizar la carga del archivo de trazas se debe realizar una manipulación de los datos de tal forma que el contenido sea explícito y genere información valiosa para el diagnóstico del problema. El procesamiento de las tramas se lleva a cabo definiendo el canal y llamada asociada en el siguiente orden:

1. Asignar la traza al canal asociado.

2. Asignar el mensaje ISUP a la llamada en caso de que el canal tenga una llamada abierta, en caso contrario se crea una nueva llamada con el mensaje definido en la traza.
3. Validar que los mensajes subsiguientes contengan una secuencia de intercambio regular hasta la llegada de un mensaje ISUP con H0H1 igual a RLC para cerrar la llamada.
4. Repetir los pasos #1 y #2 para la asignación de trazas.
5. Repetir el proceso desde el paso #4 para analizar otra llamada hasta completar el análisis de las trazas para todos los canales.

Con cada mensaje ISUP analizado se actualiza el estado de una llamada hasta que esta es cerrada por un mensaje de liberación del canal, es la condición de parada por defecto ya que permite que el canal sea utilizado por otro suscriptor para establecer una llamada nueva en el mismo canal. En función de los mensajes asociados a una llamada se define si es válida mientras que cumpla con transiciones válidas, en caso contrario se marca como irregular y las tramas son guardadas hasta la liberación del canal.

Para realizar el análisis de las trazas se define una máquina de estados (5.15) que contiene un conjunto de transiciones para cada mensaje ISUP, con cada traza viene un mensaje asociado y en función de éste se realizan cambios de estado, que dependiendo de las transiciones se puede determinar si una llamada es válida o no.

5.2. Determinación de Requerimientos

Requerimientos Funcionales

- El usuario debe poder cargar un archivo de tramas no mayor a 20Mb para el análisis de datos.
- Los datos de las tramas deben permanecer en base de datos para futuras referencias.
- El sistema debe poseer un analizador de tramas que permita estructurar las entradas por canal, llamada y archivo de entrada.
- El sistema debe mostrar en sus resultados la cantidad de llamadas completas e incompletas con las tramas asociadas.
- Los resultados del análisis deben ser representados gráficamente para ver la distribución de los mensajes ISUP.
- El usuario debe poder exportar los resultados del análisis.

Requerimientos No Funcionales

- El usuario debe poder iniciar y cerrar sesión en el sistema.
- La arquitectura del sistema debe implementarse usando herramientas de Software Libre.
- El analizador de tramas debe ser escalable y extensible para la carga de trazas con volúmenes de datos manejados en entornos de producción.
- Se debe generar un diagrama con la máquina de estados implementada por el analizador de trazas.

5.3. Diseño del Analizador de Tramas

El Analizador de Tramas ISUP SS7 es un sistema que se plantea a raíz del muestreo de tramas en centrales telefónicas para el análisis y resolución de fallas, su diseño está basado en la arquitectura Cliente/Servidor partiendo de la posibilidad de que un usuario pueda cargar archivos de trazas para realizar el análisis del tráfico de señalización capturado.

El sistema está diseñado para correr en un navegador web con la idea de brindar el servicio sin la necesidad de instalar herramientas *Stand-Alone* o algún cliente privativo de plataforma, su esencia se basa en herramientas de Software Libre que permiten conjugar una Aplicación Web con un diseño fresco y minimalista.

En el despliegue de los resultados se muestran las trazas agrupadas en dos segmentos, el de llamadas completadas y el de llamadas incompletas; en ambos casos se agrupan las trazas por llamadas realizadas en cada canal del enlace. Este orden le permite al usuario realizar un análisis detallado de una llamada en específico mediante la observación de sus trazas de señalización en el orden que éstas son generadas.

También se incluye como parte de los resultados una gráfica que muestra la distribución de los mensajes ISUP para llamadas completas e incompletas, con esta información el usuario podría determinar un fallo específico en la red por el incremento irregular en la generación de un mensaje ISUP en particular. En el diseño del sistema se incluye la posibilidad de exportar la información mostrada en el despliegue de los resultados en archivos PDF ó XLS.

Tomando en cuenta que el Analizador de Tramas ISUP SS7 es una herramienta *Operations Support Systems*, OSS, su diseño está basado en una máquina de estados cuyas transiciones son realizadas en función de los mensajes ISUP cargados desde el archivo de entrada para su posterior análisis.

La arquitectura de software planteada para la implementación sigue el modelo Cliente/Servidor en el cuál interactúan servidores de aplicación y base de datos para generar la solución que soporta la aplicación web.

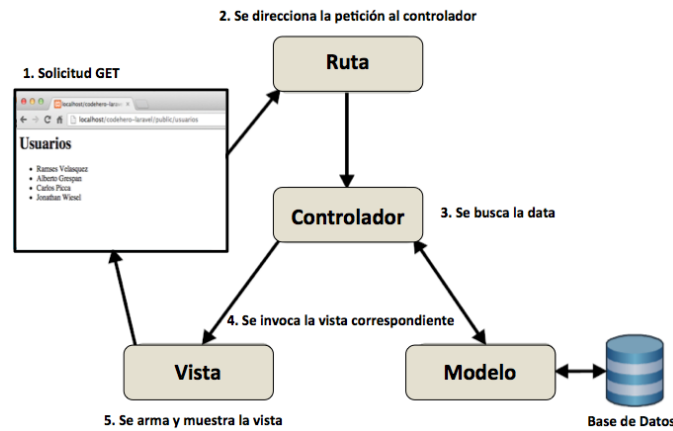


Figura 5.3: Arquitectura de la Aplicación Web[17]

5.3.1. Arquitectura Cliente-Servidor

Se puede definir como una arquitectura distribuida que permite a los usuarios finales obtener acceso a la información en forma transparente aún en entornos multiplataforma.

En esta arquitectura, el cliente envía un mensaje solicitando un determinado servicio a un servidor (hace una petición) y este envía uno o varios mensajes con la respuesta (provee el servicio), como se observa en la *Figura 5.4*. En un sistema distribuido cada máquina que provee un servicio puede cumplir el rol de servidor para algunas tareas y también el rol de cliente para otras.[12]

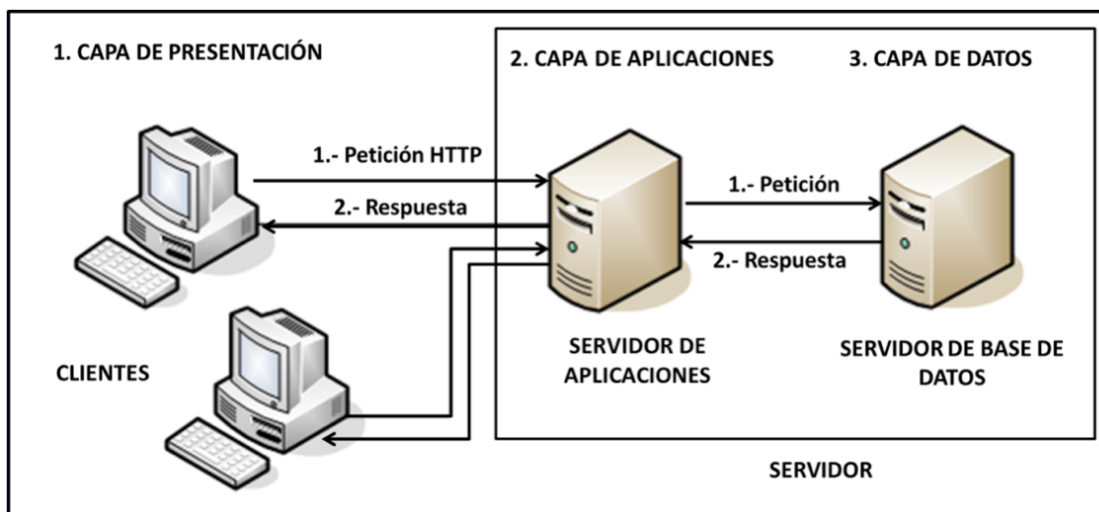


Figura 5.4: Arquitectura Cliente-Servidor [15]

Cliente

Es el proceso que permite al usuario formular los requerimientos y enviarlos al servidor, también se le conoce con el término *front-end*. El cliente habitualmente maneja todas las funciones relacionadas con la manipulación y despliegue de datos, por lo que están desarrollados sobre plataformas que permiten construir interfaces de usuario, además de acceder a los servicios distribuidos en cualquier parte de una red.[12]

Las funciones que lleva a cabo el cliente se resumen en los siguientes puntos:

- Administra la interfaz de usuario.
- Captura y valida los datos de entrada.
- Procesa la lógica de la aplicación y hacer validaciones locales.
- Genera requerimientos de bases de datos.
- Recibe resultados del servidor.
- Despliega resultados.

Servidor de Aplicaciones

El servidor de aplicaciones es el proceso encargado de atender a múltiples clientes que hacen peticiones de algún recurso gestionado. El servidor se encarga de responder a estas peticiones de forma adecuada, entregando como resultado una serie de paquetes de una aplicación web o datos en específico de acuerdo al protocolo manejado en la comunicación.[12]

Las funciones que lleva a cabo el proceso servidor se resumen en los siguientes puntos:

- Acepta peticiones a recursos provenientes del Cliente.
- Procesa los requerimientos de base de datos.
- Prepara los datos para responder peticiones del Cliente.
- Procesa la lógica de la aplicación y realiza validaciones del modelo nivel de bases de datos.

Servidor de Base de Datos

Los Servidores de Base de Datos son *Sistemas Manejadores de Bases de Datos* (SMBD) con una colección de programas cuyo objetivo es servir de interfaz entre la base de datos, el usuario y las aplicaciones. Se componen de un lenguaje de definición de datos, de un lenguaje de manipulación de datos y de un lenguaje de consulta. Un SMBD permite definir los datos a distintos niveles de abstracción y manipularlos, garantizando la seguridad e integridad de los mismos.

Un SMBD permite:

- **Definir una Base de Datos:** especificar tipos, estructuras y restricciones en los datos.
- **Construir la Base de Datos:** guardar los datos en algún medio controlado por el mismo SMBD.
- **Manipular la Base de Datos:** realizar consultas, actualizarla y generar informes.

5.3.2. Descripción del Modelo de Base de Datos

El modelo de datos es fundamental en la lógica del analizador de tramas ya que está diseñado para guardar las posibles transiciones de estado que pueden ser registrados en una llamada telefónica. La *Figura 5.5* muestra el modelo *Entidad - Relación*, ER de base de datos:

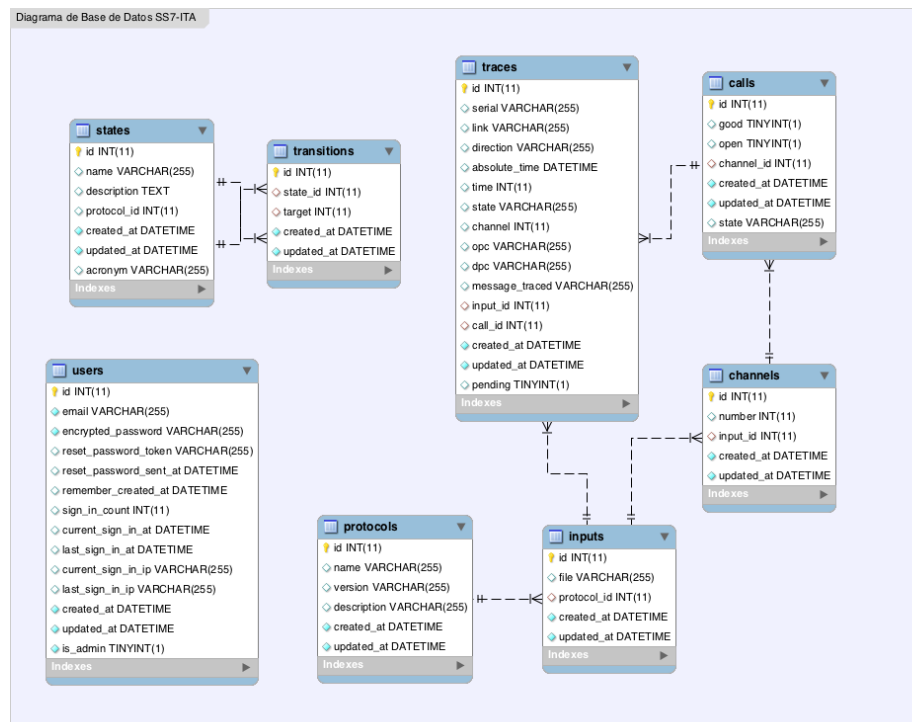


Figura 5.5: Modelo Entidad/Relación

A continuación una breve descripción de cada clase:

- **Users:** esta tabla contiene los datos del usuario e información requerida para el mantenimiento de su sesión, a continuación se describen sus campos:
 - *email:* funciona como nombre de usuario y buzón para las notificaciones por correo en caso que sean implementadas para análisis de datos que corran como tareas programadas.

- *encrypted_password*: contiene la contraseña del usuario cifrada.
 - *reset_password_token*: es una secuencia de caracteres que funcionan como clave primaria para recuperar la contraseña del usuario por correo electrónico en caso de que éste la olvide.
 - *reset_password_sent*: guarda la fecha del último correo enviado para recuperar la contraseña del usuario en el sistema.
 - *remember_created_at*: guarda la fecha en la que el usuario selecciona la opción de ser recordado por el sistema al inicio de la sesión.
 - *sign_in_count*: es un contador de sesiones establecidas en el sistema.
 - *current_sign_in_at*: contiene la hora de la última sesión abierta en el sistema.
 - *last_sign_in_at*: contiene la hora de la última sesión establecida en el sistema.
 - *current_sign_in_ip*: contiene la dirección IP del cliente para la sesión actual.
 - *last_sign_in_ip*: contiene la dirección IP del cliente de la última sesión abierta en el sistema.
 - *created_at, updated_at*: contiene etiquetas de tiempo para guardar la fecha de creación y última actualización de los datos de la tabla.
 - *is_admin*: es una variable binaria para indicar si el usuario es administrador o no.
- **States:** esta tabla contiene los mensajes ISUP de SS7 descritos en la recomendación de la ITU-T para SS7, a continuación se describen sus campos:
- *name*: contiene el nombre del mensaje ISUP.
 - *acronym*: contiene el acrónimo del mensaje ISUP muy utilizado en el análisis de las trazas.
 - *description*: contiene la descripción del mensaje ISUP.
 - *protocol_id*: contiene la clave foránea del protocolo de las trazas. Esta variable fue incluida en el modelo para hacerlo escalar pensando en la posibilidad de analizar tramas de otros protocolos de comunicaciones.
 - *created_at, updated_at*: contiene etiquetas de tiempo para guardar la fecha de creación y última actualización de los datos de la tabla.
- **Transitions:** contiene la lógica de la aplicación al albergar las posibles transiciones de cambio de estados que se pueden realizar entre mensajes ISUP. A continuación se describen sus campos:
- *state_id*: contiene la clave foránea del estado inicial.
 - *target*: contiene la clave foránea del estado final.
 - *acronym*: contiene el acrónimo del mensaje ISUP muy utilizado en el análisis de las trazas.

- *created_at, updated_at*: contiene etiquetas de tiempo para guardar la fecha de creación y última actualización de los datos de la tabla.
- **Protocols**: contiene los datos de los protocolos que puede analizar el sistema, a continuación se describen sus campos:
 - *name*: contiene el nombre del protocolo.
 - *version*: contiene la versión del protocolo.
 - *description*: contiene una breve descripción del protocolo.
 - *created_at, updated_at*: contiene etiquetas de tiempo para guardar la fecha de creación y última actualización de los datos de la tabla.
- **Inputs**: contiene los datos de los archivos cargados al sistema, a continuación se describen sus campos:
 - *file*: contiene el nombre del archivo de entrada.
 - *protocol_id*: contiene la clave foránea del protocolo.
 - *created_at, updated_at*: contiene etiquetas de tiempo para guardar la fecha de creación y última actualización de los datos de la tabla.
- **Channels**: contiene los datos de los canales recolectados de las trazas, a continuación se describen sus campos:
 - *number*: contiene el número del canal empleado.
 - *input_id*: contiene la clave foránea del registro del archivo de entrada.
 - *created_at, updated_at*: contiene etiquetas de tiempo para guardar la fecha de creación y última actualización de los datos de la tabla.
- **Calls**: contiene los datos de las llamadas conformadas en el análisis de las trazas, a continuación se describen sus campos:
 - *good*: es una variable binaria para indicar si la llamada tiene un comportamiento normal o no.
 - *open*: es una variable binaria para indicar si la llamada puede recibir más mensajes ISUP o no.
 - *channel_id*: contiene la clave foránea del registro del canal empleado.
 - *created_at, updated_at*: contiene etiquetas de tiempo para guardar la fecha de creación y última actualización de los datos de la tabla.
 - *state*: contiene el estado de la llamada en función de las transiciones definidas para los mensajes ISUP.

- **Traces:** contiene los datos de las llamadas conformadas en el análisis del las trazas, a continuación se describen sus campos:
 - *serial:* contiene el número del mensaje ISUP.
 - *link:* contiene el número del enlace empleado.
 - *direction:* contiene la dirección en la que va el mensaje ISUP, puede ser *send* o *receive*.
 - *absolute_time:* contiene etiquetas de tiempo para guardar la fecha de creación del mensaje.
 - *time:* es una variable de tiempo que contiene la hora relativa del mensaje en tiempo de ejecución.
 - *state:* contiene el acrónimo del mensaje ISUP contenido en la traza.
 - *channel:* contiene el número del canal empleado.
 - *opc:* contiene el código del punto de origen.
 - *dpc:* contiene el código del punto de destino.
 - *message_traced:* contiene la traza del mensaje con hexadecimal.
 - *input_id:* contiene la clave foránea del registro del archivo de entrada.
 - *call_id:* contiene la clave foránea de la llamada asociada a la traza.
 - *created_at, updated_at:* contiene etiquetas de tiempo para guardar la fecha de creación y última actualización de los datos de la tabla.
 - *pending:* es una variable binaria para indicar si la traza está pendiente por procesar en análisis de datos a dos pasadas.

5.4. Tecnologías Empleadas

Para el desarrollo del sistema se han utilizado diversas tecnologías de software libre que comprenden lenguajes de programación, lenguajes de marcado de texto, servidores de base de datos, servidores de aplicación, controlador de versiones y *frameworks* para desarrollo web. Todas estas herramientas han sido integradas bajo el sistema operativo Linux para generar entorno de desarrollo que permita realizar el despliegue de la aplicación.

5.4.1. Frameworks

La selección del lenguaje de programación es una decisión muy importante en la implementación de un proyecto y son muchas variables que se toman en cuenta para la elección del mismo, entre éstas se pueden mencionar:

- El tipo de aplicación a desarrollar (*scripts*, gráficos, multimedia, juegos).

- Los requerimientos de desempeño y disponibilidad.
- Ejecución en tiempo real o manejo de volúmenes de datos, etc..

Con base en el diseño y de **SS7-ITA**, era requerido un *framework* para el desarrollo web que contara con soporte en la comunidad de software libre y a su vez un amplio catálogo de librerías que pudieran ser integradas en demanda para el desarrollo de cada funcionalidad. De igual forma se tomó en cuenta la curva de aprendizaje implícita en el lenguaje y el nivel de complejidad requerido para la configuración de las librerías e instalación del entorno de desarrollo.

Dadas estas premisas se propone **Ruby on Rails (RoR)**[3] como *framework* para desarrollo web basado en la arquitectura **Modelo Vista Controlador (MVC)** y una gran cantidad de librerías soportadas por la comunidad de desarrolladores, cuenta con un gestor de paquetes llamado RubyGems[13] dedicado a proveer un formato estándar y auto-contenido (*gem*) que permite distribuir programas y librerías escritas en **Ruby**[9] bajo la filosofía del software libre.

Los componentes que interactúan en el despliegue de una aplicación web basada en **RoR** están muy bien definidos y descritos en su documentación[14] y manuales de implementación[4], estos son mostrados en la *Figura 5.6*:



Figura 5.6: Componentes de una aplicación web basada en Ruby on Rails

Como se ha mencionado anteriormente, RoR sigue la arquitectura MVC pero esto es aplicado fundamentalmente del lado del servidor de aplicaciones ya que el cliente o navegador web sólo se

comunica con el servidor bajo una petición de URL predefinida en las rutas de la aplicación. Este modelo en ocasiones no es suficiente para cumplir con los requerimientos de un sistema ya que en muchos casos la forma en la que se presenta la información al usuario puede representar el éxito de un sistema por usabilidad.

Teniendo en cuenta estos escenarios; RoR ofrece métodos, librerías y gemas que son ejecutadas del lado del cliente para brindar una mejor experiencia al usuario y por ende una mejor penetración de mercado dada las bondades provistas a los desarrolladores para la implementación de aplicaciones.

Siguiendo este esquema, se puede observar en la *Figura 5.6* que existen dos componentes de software que se ejecutan del lado del cliente, estos son las plantillas o *layouts* y los ayudantes o *helpers* por sus nombres en inglés. Las plantillas contienen el maquetado del diseño web y la distribución de los elementos a renderizar por el navegador, los *helpers* actúan como métodos que proporcionan a la vista cierta lógica de presentación para mejorar el despliegue de los resultados de una petición web.

En el *Layout* se incluyen otras tecnologías como lo son el lenguaje de marcado de hiper texto (*Hyper Text Markup Language* (HTML)), las hojas de estilo (*Cascading Style Sheets*, CSS3) y el lenguaje de programación *JavaScript* encargados de enriquecer el diseño de las vistas del sistema, estos elementos junto con los *Helpers* hacen de RoR un *framework* de desarrollo web muy versátil ya que le facilita tanto a desarrolladores como diseñadores la creación de interfaces web innovadoras y atractivas para el usuario.

5.4.2. Tecnologías del lado del Cliente

Existen una gran cantidad de páginas dinámicas donde el procesamiento de los efectos y funcionalidades es soportado por el navegador. Las tecnologías del lado del cliente, son aquellas que se ejecutan en el navegador del usuario y proveen dichas funcionalidades a las vistas; a continuación se describen brevemente algunas de estas tecnologías:

- **HTML:** es un estándar reconocido en todo el mundo y cuyas normas define un organismo sin fines de lucro llamado *Wide Web Consortium* (W3C). Es el lenguaje de marcado predominante para la construcción de documentos de hipertexto, que son independientes de la plataforma.

La última versión de este lenguaje (HTML5) cual proporciona una plataforma con la que se pueden desarrollar aplicaciones web parecidas a las aplicaciones de escritorio. Incluye novedades significativas en diversos ámbitos y supone mejoras en áreas que eran delegadas a otras tecnologías del lado del cliente, entre las mejoras significativas de HTML 5 se pueden mencionar:

- **Estructura del cuerpo:** HTML 5 permite agrupar cabecera, pie, navegadores, etc., en nuevas etiquetas que representan cada una de las secciones de una página.

- **Etiquetas para contenido específico:** Hasta ahora se utiliza una única etiqueta para incorporar diversos tipos de contenido enriquecido, como animaciones Flash o vídeo. Ahora es posible usar etiquetas específicas para cada tipo de contenido en particular, como audio, vídeo, etc.
 - **Canvas:** es un nuevo componente que permite dibujar, por medio de las funciones de un *Application Software Interface*, API en la página, todo tipo de formas que podrán estar animadas y responder a interacción del usuario.
 - **Geolocalización:** Las páginas web se pueden localizar geográficamente por medio de una API.
 - **Nuevas APIs para interfaz de usuario:** temas tan utilizados como el (")drag and drop(") (arrastrar y soltar) en las interfaces de usuario de los programas convencionales, son incorporados al HTML 5 por medio de una API.
 - **Fin de las etiquetas de presentación:** todas las etiquetas que tienen que ver con la presentación del documento son eliminadas. La responsabilidad de definir el aspecto de una página web está a cargo únicamente del CSS.
- **CSS:** es un lenguaje formal usado para definir la presentación de un documento estructurado escrito en HTML o XML (y por extensión en XHTML). Presenta numerosas ventajas, ya que obliga a crear documentos bien definidos mejorando la accesibilidad y reduciendo la complejidad de su mantenimiento. La última versión de este lenguaje facilita el trabajo a los desarrolladores, ya que incorpora nuevos mecanismos que proveen un mayor control sobre el estilo con el que se muestran los elementos de las páginas.
 - **JavaScript:** es un lenguaje de programación interpretado que no requiere compilación y se puede ejecutar directamente en el navegador sin necesidad de procesos intermedios. Provee acceso a los elementos del *Document Object Model* (DOM) a través de métodos, lo que permite el desarrollo de aplicaciones dinámicas mediante la modificación del contenido en tiempo real a través del manejo de eventos.
 - **JQuery:** es una librería de *JavaScript* que permite simplificar la manera de interactuar con los documentos HTML, manipular el árbol DOM, manejar eventos, desarrollar animaciones y agregar interacción con la tecnología AJAX a páginas web. *JQueryUser Interface (jQueryUI)* es una extensión de la librería JQuery, que le añade un conjunto de widgets y eventos para la creación de aplicaciones web.

5.4.3. Framework para el Diseño Web - UIKIT

Usando CSS y JavaScript se pueden crear nuevos *frameworks* con fines específicos, ese es el caso de *UIKIT* cuya concepción ha sido orientada a la creación de interfaces web rápidas y poderosas ofreciendo una colección completa de componentes HTML, CSS y *JavaScript* fácil de usar, fácil de personalizar y extensible, de igual forma bajo licencias *Open Source*.

Este *framework* de código abierto mejora considerablemente la vista del diseño web de SAT-ISUPSS7 ya que emplea una serie de componentes que van desde el uso de barras de navegación hasta *canvas* con efectos 3D. La fortaleza del *framework* está definida por el desarrollador ya que se puede adaptar a las necesidades específicas del proyecto.

En la página web de los desarrolladores de *UIKIT* [22] se encuentra un listado de los componentes disponibles para el diseño web junto con instrucciones detalladas para su implementación, también es posible personalizar el tema que trae la herramienta por defecto; tal y como se muestra en la *Figura 5.7*:

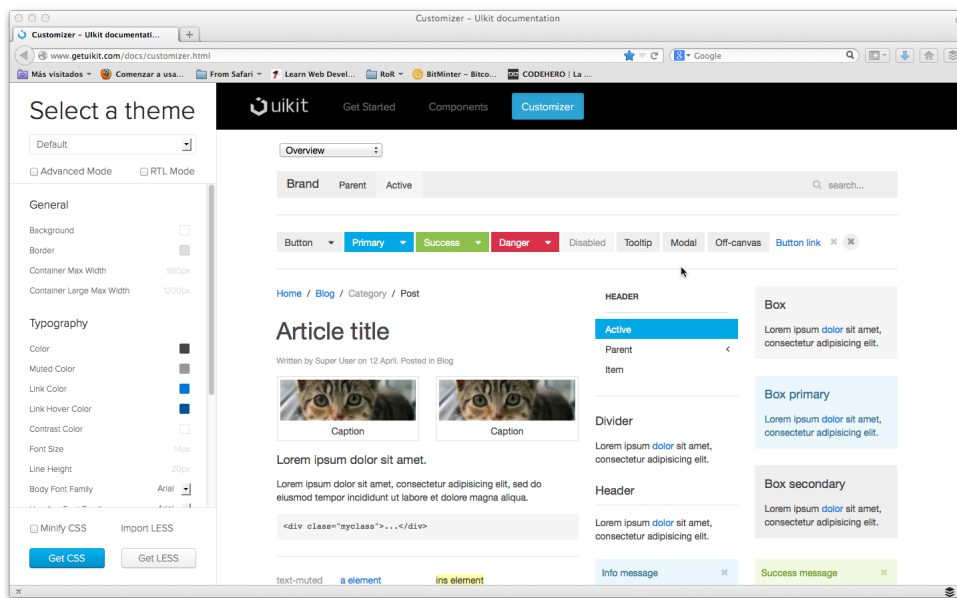


Figura 5.7: *UIKIT* - *Framework* para el diseño web

Esta herramienta fue utilizada para crear el tema de *glsstisup* ajustando elementos y opciones que definen el estilo de la aplicación, luego de esto se descarga un archivo CSS que es posteriormente es cargado en el *layout* de *SS7-ISA*.

Una vez es obtenida la hoja de estilos CSS se procede con la asignación de las clases que dan forma a los elementos HTML para de esta manera plasmar un diseño funcional que es aplicado desde la página principal (*Figura 5.8*) hasta la página que muestra los resultados *Figura 5.9* del análisis de

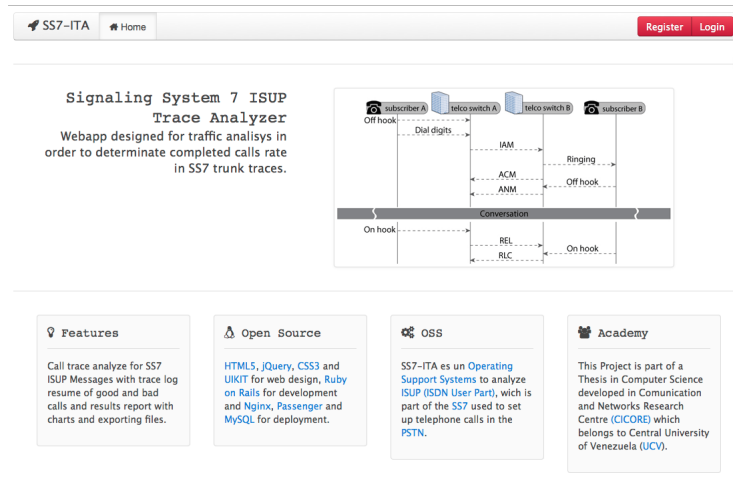


Figura 5.8: Página principal de SATISUP-SS7

trazas SS7 a lo largo de la aplicación.

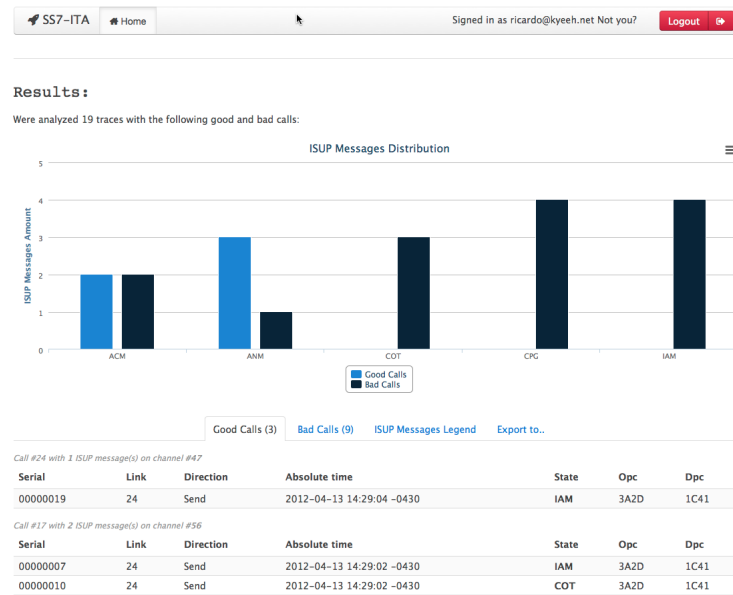


Figura 5.9: Página de resultados SS7-ISA

5.4.4. Entorno de Desarrollo

La integración de las herramientas escogidas para el entorno de desarrollo (*Integrated Development Environment (IDE)*) son un factor fundamental en la planificación de un proyecto, lo ideal es

balancear el dominio que se tiene sobre la herramienta y las funcionalidades o bondades que ésta ofrece al equipo de trabajo.

Un ejemplo claro es la consola o terminal de Linux que prácticamente puede cumplir con las funciones de un IDE pero en la mayoría de los casos no ofrece un entorno amigable al desarrollador, salvo que éste tenga amplio dominio de la ejecución de comandos en *Shell*.

Para el desarrollo de aplicaciones con RoR hay gran cantidad de opciones de IDEs, se pueden encontrar programas tales como *Rubymine* que incluyen gran cantidad de módulos para soportar control de versiones, administración de interpretes y gemas, *frameworks* para pruebas de software y *debugging*, terminal embebido, etc... Estas aplicaciones normalmente son de carácter comercial y vienen con costos asociados a licencias de uso.

Una alternativa a los IDEs comerciales son los entornos de desarrollo con herramientas de Software Libre que soportan las mismas características para el programador sólo que en aplicaciones por separado; entre ellas se pueden mencionar editores de texto, clientes de base de datos, *debuggers*, terminal, clientes para control de versiones, etc..

Todas estas herramientas son aplicaciones que corren de forma independiente pero pueden ser integradas mediante archivos de configuración o comandos por consola. La *Figura 5.10* muestra las herramientas utilizadas en diferentes capas de acuerdo a su funcionalidad:

A continuación se procede a describir cada una de las capas y las herramientas que la conforman:

- **Sistema Operativo:** se opta por Linux con las herramientas GNU asociadas a la distribución[6], a partir de este punto se inicia la instalación y configuración del resto de las aplicaciones.
- **IDE:** es representado por el conjunto de aplicaciones asociadas al desarrollo de la aplicación entre las que se pueden mencionar:
 - **Sublime:** es un editor de texto muy poderoso que soporta diversos lenguajes de programación[7], implementa comandos para casi todas sus funcionalidades buscando incrementar la productividad, mantiene los archivos asociados a un proyecto incluso sin haber guardado cambios, incluye una consola para la instalación de *plugins* permitiendo extender las funcionalidades del editor, es totalmente customizable mediante archivos de configuración; su interfaz gráfica es mostrada en la *Figura 5.11* con código fuente de la aplicación.
 - **MySQL Workbench:** es un cliente MySQL[10] con interfaz para la administración del servidor de base de datos, desarrollo SQL y modelado de datos con ingeniería inversa con una interfaz gráfica muy amigable mostrada en la *Figura 5.12*.
 - **Terminal:** es la interfaz de comandos del sistema y permite administrar cualquier tarea del sistema operativo, a través de esta herramienta se realiza la instalación de los servidores de aplicación, base de datos y control de versiones; también permite ejecutar



Figura 5.10: Entorno de Desarrollo

tareas asociadas al desarrollo web incluidas en RoR como parte de las funcionalidades del *framework*, administrar las variables de entorno y ejecutar comandos para el control de versiones; tal y como se muestra en la *Figura 5.13*.

5.5. Servidor de Aplicación

En esta sección se describirán los componentes del servidor de aplicaciones mencionados en la *Figura 5.14* junto con la descripción del procedimiento de instalación y configuración mostrando instrucciones detalladas para su implementación.

5.5.1. Servidor Web

El Servidor Web está compuesto por dos piezas de software (NGINX, Phusion Passenger) que permiten el despliegue de las páginas web y la interpretación de la lógica de la aplicación, estos componentes se conjugan para desplegar un Servidor Web que responde a las peticiones de contenido estático y dinámico de acuerdo a las funcionalidades del sistema.

```

1 class Call < ActiveRecord::Base
2   attr_accessible :good, :open
3   belongs_to :channel
4
5   has_many :traces
6
7   state_machine :state, :initial => :Nueva do
8     before_transition :Nueva => any => :init
9     #after_transition any - [:Nueva, :Completada] => any, :do => :check_new
10    after_transition any => [:Completa, :Incompleta], :do => :close
11
12    event :iam do
13      transition :Nueva => :Abierta
14      transition any - [:Nueva, :Completa, :Incompleta] => :Irregular
15    end
16
17    event :acc do
18      transition :Abierta => :Replicando # Viene de un IAM
19      transition :Conectando => :Replicando # Viene de un COT
20      #transition any - [:Abierta, :Conectando] => :Irregular
21    end
22
23    event :am do
24      transition :Replicando => :Establecida # Viene de un ACH
25      transition :Restablecida => :Establecida # Viene de un RES
26      #transition any - [:Replicando, :Restablecida] => :Irregular
27    end
28
29    event :cot do
30      transition :Abierta => :Conectando # Viene de un IAM
31      #transition any - :Abierta => :Irregular
32    end
33
34    event :cpg do
35      transition :Replicando => :EnProgreso # Viene de un ACH
36      transition :EnProgreso => :same # Viene de un CPG
37      transition :Establecida => :EnProgreso # Viene de un AMM
38      #transition any - [:Replicando, :EnProgreso, :Establecida] => :Irregular
39    end
40
41    event :rsc do
42      transition :Abierta => :Reset # Viene de un IAM
43      transition :Conectando => :Reset # Viene de un COT
44      transition :Replicando => :Reset # Viene de un ACH
45      transition :EnProgreso => :Reset # Viene de un CPG
46      transition :Establecida => :Reset # Viene de un AMM
47      transition :Suspendida => :Reset # Viene de un SUS

```

Figura 5.11: Sublime

id	serial	link	direction	absolute_time	time	state	channel	opc	dpc
1	00000001	24	Send	2012-04-13...	61835	REL	56	3A2D	1C41
2	00000002	24	Send	2012-04-13...	61843	IAM	99	3A2D	1C41
3	00000003	24	Receive	2012-04-13...	61896	RLC	56	1C41	3A2D
4	00000004	24	Send	2012-04-13...	61925	COT	99	3A2D	1C41
5	00000005	24	Receive	2012-04-13...	61947	ANM	86	1C41	3A2D
6	00000006	24	Receive	2012-04-13...	61967	ACM	44	1C41	3A2D
7	00000007	24	Send	2012-04-13...	61971	IAM	56	3A2D	1C41
8	00000008	24	Send	2012-04-13...	62004	CPG	72	3A2D	1C41
9	00000009	24	Send	2012-04-13...	62048	REL	47	3A2D	1C41
10	00000010	24	Send	2012-04-13...	62053	CDT	56	3A2D	1C41
11	00000011	24	Send	2012-04-13...	62112	REL	56	3A2D	1C41
12	00000012	24	Receive	2012-04-13...	62119	ACM	107	1C41	3A2D
13	00000013	24	Receive	2012-04-13...	62139	REL	72	1C41	3A2D
14	00000014	24	Receive	2012-04-13...	62142	RLC	47	1C41	3A2D

Figura 5.12: MySQL Workbench

NGINX (pronunciado en inglés “engine X”) es un servidor web/proxy inverso ligero de alto rendimiento y un proxy para protocolos de correo electrónico (IMAP/POP3), es software libre y de código abierto, licenciado bajo la Licencia BSD simplificada. Es multiplataforma, por lo que corre en sistemas tipo Unix (GNU/Linux, BSD, Solaris, Mac OS X, etc.) y Windows y es usado por una larga lista de sitios web conocidos tales como *WordPress*, *Hulu*, *GitHub*, *Ohloh*, *SourceFor-*

```

kyeeh@011v1a1:~/Projects/UCV/SandT/SS7-AnaLizer/twa$ gem env
RubyGems Environment:
  - RUBYGEMS VERSION: 2.0.2
  - RUBY VERSION: 2.0.0 (2012-05-14 patchlevel 135) [x86_64-darwin12.0.0]
  - INSTALLATION DIRECTORY: /Users/kyeeh/.rvm/gems/ruby-2.0.0-p195@twa
  - RUBY EXECUTABLE: /Users/kyeeh/.rvm/rubies/ruby-2.0.0-p195/bin/ruby
  - EXECUTABLE DIRECTORY: /Users/kyeeh/.rvm/gems/ruby-2.0.0-p195@twa/bin
  - RUBYGEMS PLATFORMS:
    - ruby
    - x86_64-darwin-12
  - GEM PATHS:
    - /Users/kyeeh/.rvm/gems/ruby-2.0.0-p195@twa
    - /Users/kyeeh/.rvm/gems/ruby-2.0.0-p195@twa
  - GEM CONFIGURATION:
    - update_sources => true
    - verbose => true
    - backtrace => false
    - bulk_threshold => 1000
  - REMOTE SOURCES:
    - https://rubygems.org/
kyeeh@011v1a1:~/Projects/UCV/SandT/SS7-AnaLizer/twa$ git status
# On branch state_machine
# Changes not staged for commit:
#   (use "git add <file>..." to update what will be committed)
#   (use "git checkout -- <file>..." to discard changes in working directory)
#
#   modified: ../Latex/acronimos.tex
#   modified: ../Latex/biblioseminar.bib
#   modified: ../Latex/caps-desarrollo.aux
#   modified: ../Latex/caps-desarrollo.tex
#   modified: ../Latex/caps-resultados.aux
#   modified: ../Latex/conclusion.aux
#   modified: ../Latex/main.acn
#   modified: ../Latex/main.aux
#   modified: ../Latex/main.bbl
#   modified: ../Latex/main.bib
#   modified: ../Latex/main.gls
#   modified: ../Latex/main.idx
#   modified: ../Latex/main.ilg
#   modified: ../Latex/main.ind
#   modified: ../Latex/main.ist
#   modified: ../Latex/main.lof
#   modified: ../Latex/main.log
#   modified: ../Latex/main.pdf
#   modified: ../Latex/main.synctex.gz
#   modified: ../Latex/main.toc
#
# Untracked files:
#   (use "git add <file>..." to include in what will be committed)
#
#   ../Ejemplos/
#   ../Normativa/
#   ../Z1180813_8333_tesis_ATISS7.pdf
#   ../Entregables/19182013_8136_tesis_S57.pdf
#   ../Entregables/19182013_1813_tesis_S57.pdf

```

Figura 5.13: Terminal

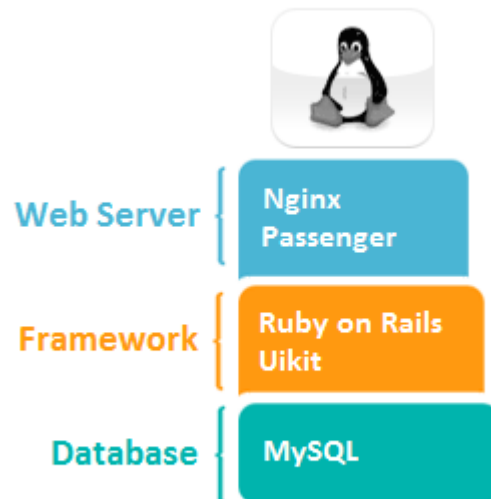


Figura 5.14: Servidor de Aplicación

ge, *TorrentReactor* y partes de *Facebook* (como el servidor de descarga de archivos zip pesados).[18]

Phusion Passenger es un servidor de aplicaciones web escritas en Ruby, Python, Node.js y Me-

teor cuyo objetivo es hacer el despliegue mucho mas simple y sencillo mediante la administración de los procesos y recursos del lado del servidor. Su núcleo esta escrito en C++ lo que lo hace rápido y confiable, tiene arquitectura ZERO-COPY [20] con diseño multihilo y multiproceso. [11]

5.5.2. Instalación

La instalación del servidor se realiza mediante la consola de *Linux (Shell)* y una serie de comandos que permiten la descarga, configuración y compilación del código fuente requerido junto con las librerías incluidas en las dependencias.

Para iniciar el proceso se utiliza una herramienta llamada *Ruby Version Manager (RVM)* que permite la administración de las diferentes versiones de glsror requeridas para cada proyecto, para el caso de SAT-ISUPSS7 se ejecutan los comandos descritos a continuación:

```
1 # RVM installation with Ruby, RubyGems and Rails
2 \curl -sSL https://get.rvm.io | bash
3
4 # NGINX Web Server with Passenger Module
5 passenger-install-nginx-module
```

5.5.3. Archivos de Configuración

A continuación se describen los archivos de configuración y las funcionalidades implementadas en la puesta en producción de SAT-ISUPSS7 en el dominio *twinlight.net*.

Dado que un servidor se pueden ejecutar múltiples aplicaciones, se define un subdominio para la aplicación (*ss7a.twinlight.net*) creando un virtual *host* que responda las peticiones web en NGINX, a continuación se muestran los archivos de configuración asociados.

- Configuración por defecto de *NGINX* con el modulo de *Passenger* activo y tamaño archivo de subida máximo de 20Mb:

```
1 server {
2     user www-data;
3     worker_processes 1;
4
5     events {
6         worker_connections 1024;
7     }
8
9     http {
```

```
10 passenger_root /usr/local/rvm/gems/ruby-2.0.0-p195/gems/passenger
    -4.0.5;
11 passenger_ruby /usr/local/rvm/wrappers/ruby-2.0.0-p195/ruby;
12
13 include mime.types;
14 default_type application/octet-stream;
15
16 log_format gzip '$remote_addr - $remote_user [$time_local] '
17     '$request' $status $bytes_sent '
18     '$http_referer' '$http_user_agent' '$gzip_ratio';
19
20 access_log logs/access.log gzip buffer=512k;
21
22 server_names_hash_bucket_size 64;
23 sendfile on;
24 keepalive_timeout 65;
25
26 gzip on;
27 client_max_body_size 20M;
28
29 ##
30 # Virtual Host Configs
31 ##
32 #include /opt/nginx/conf/*.conf;
33 include /opt/nginx/conf/sites-enabled/*;
34 }
```

- Configuración del dominio *twinlight.net*:

```
1 server {
2     listen 80 default_server; # e.g., listen
    192.168.1.1:80;
3     server_name www.twinlight.net twinlight.net; # e.g., server_name
    source.example.com;
4
5     #charset koi8-r;
6
7     access_log logs/host.access.log gzip buffer=512k;
8
9     location / {
10        root html;
11        index index.php index.html index.htm;
12    }
13
14    error_page 404 /404.html;
```

```

15
16 # redirect server error pages to the static page /50x.html
17 #
18 error_page 500 502 503 504 /50x.html;
19 location = /50x.html {
20     root html;
21 }
22
23 # pass the PHP scripts to FastCGI server listening on
24     127.0.0.1:9000
25 location ~ /\.php$ {
26     fastcgi_pass unix:/var/run/php5-fpm.sock;
27     fastcgi_index index.php;
28     fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name;
29     include fastcgi_params;
30 }

```

- Configuración del subdominio *ss7a.twinlight.net* con *Passenger* activo y entorno de producción por defecto:

```

1 server {
2     listen 80;                # e.g., listen 192.168.1.1:80;
3     server_name ss7a.twinlight.net;    # e.g., server_name source.
4         example.com;
5     passenger_enabled on;
6     passenger_friendly_error_pages on;
7     #passenger_user www-data;
8     rails_env production;
9     root /var/rails/sandt/SS7-Analizer/twa/public;
10
11     location /data {
12         root /var/rails/sandt/data;
13         autoindex on;
14     }
15
16 # individual nginx logs for this ss7a vhost
17 access_log /opt/nginx/logs/ss7a_access.log;
18 error_log /opt/nginx/logs/ss7a_error.log;
19 }

```

5.5.4. Ruby on Rails

Ruby on Rails es el *framework* para desarrollo web de código abierto utilizado en la implementación de SAT-ISUPSS7. Esta escrito en el lenguaje de programación *Ruby* siguiendo el paradigma de la arquitectura MVC, busca combinar la simplicidad basada en convenciones con la posibilidad de desarrollar aplicaciones del mundo real escribiendo menos código que con otros *frameworks* y con un mínimo de configuración.

El lenguaje de programación *Ruby* permite la meta-programación, de la cual *Rails* hace uso, lo que resulta en una sintaxis que muchos de sus usuarios encuentran muy legible. *Rails* se distribuye a través de *RubyGems*, que es el formato oficial de paquete y canal de distribución de bibliotecas y aplicaciones *Ruby*.^[19]

Para la implementación de todas las funcionalidades de SAT-ISUPSS7 se usan una serie de librerías que son cargadas a través del archivo *Gemfile*, este archivo contiene un listado de las gemas a utilizar con la versión para algunos casos tal y como se muestra a continuación:

Archivo *Gemfile* utilizado en SAT-ISUPSS7:

```
1  source 'https://rubygems.org'
2
3  gem 'rails'
4  # Bundle edge Rails instead:
5  # gem 'rails', :git => 'git://github.com/rails/rails.git'
6
7  gem 'mysql2'
8  gem 'passenger'
9  gem 'devise'
10 gem 'carrierwave'
11 gem 'will_paginate', '~> 3.0'
12 gem 'lazy_high_charts'
13 gem 'spreadsheet'
14 gem 'state_machine'
15 gem 'ruby-graphviz', :require => 'graphviz' # Optional: only required
    for graphing
16 gem 'prawn', :git => 'https://github.com/prawnpdf/prawn.git'
17 gem 'prawn_rails'
18
19
20 # Gems used only for assets and not required
21 # in production environments by default.
22 group :assets do
23   gem 'sass-rails', '~> 3.2.3'
24   gem 'coffee-rails', '~> 3.2.1'
25   gem 'uglifier', '>= 1.0.3'
26 end
```

```
27
28   group :development do
29     gem "better_errors"
30   end
31
32   gem 'jquery-rails'
```

Estas librerías son instaladas usando el comando *bundle install* que busca conectar con el repositorio (*source <https://rubygems.org>*) para descargar e instalar las gemas (librerías) listadas en el archivo de acuerdo a su versión. Una vez es configurado el entorno la aplicación es ejecutada satisfactoriamente.

El diseño de las vistas es soportado por un *framework CSS* que ofrece herramientas para generar una hoja de estilos compatible con RoR, este archivo es cargado como la plantilla de la aplicación para aplicar los estilos deseados a cada funcionalidad del sistema.

5.5.5. MySQL

El servidor de Base de Datos utilizado es MySQL con el cliente PhpMyAdmin para su administración, la configuración de la base de datos y los detalles de la conexión están descritos en el archivo *database.yml* de la aplicación, su contenido se muestra a continuación:

Archivo de configuración de la Base de Datos utilizada en SAT-ISUPSS7:

```
1   # MySQL.  Versions 4.1 and 5.0 are recommended.
2   #
3   # Install the MYSQL driver
4   #   gem install mysql2
5   #
6   # Ensure the MySQL gem is defined in your Gemfile
7   #   gem 'mysql2'
8   #
9   # And be sure to use new-style password hashing:
10  #   http://dev.mysql.com/doc/refman/5.0/en/old-client.html
11  development:
12    adapter: mysql2
13    encoding: utf8
14    reconnect: false
15    database: twa_development
16    pool: 5
17    username: twarobot
18    password: "123456"
19    host: localhost
20
```

```
21 # Warning: The database defined as "test" will be erased and
22 # re-generated from your development database when you run "rake".
23 # Do not set this db to the same as development or production.
24 test:
25   adapter: mysql2
26   encoding: utf8
27   reconnect: false
28   database: twa_test
29   pool: 5
30   username: twarobot
31   password: "123456"
32   host: localhost
33
34 production:
35   adapter: mysql2
36   encoding: utf8
37   reconnect: false
38   database: twa_production
39   pool: 5
40   username: twarobot
41   password: "123456"
42   host: localhost
```

Como puede observarse, se define el adaptador usado por RoR para conectar a BD (MySQL en este caso), la codificación de los datos, el nombre de la base de datos, la cantidad de conexiones concurrentes, el nombre del usuario y su contraseña y el nombre del *host* que alberga al servidor.

5.6. Implementación del Analizador de Tramas

El desarrollo del sistema se inicia con la carga de los datos para su procesamiento, este proceso implica el diseño de estructuras que permiten modelar la data proveniente del archivo de trazas. El archivo es cargado por completo para luego ser analizado como objetos que son guardados en base de datos. De esta forma es posible analizar cada llamada con sus respectivos mensajes ISUP para detectar patrones irregulares que no cumplen transiciones válidas.

Para la implementación del analizador de tramas se ha desarrollado una máquina de estados que contiene transiciones definidas para cada mensaje ISUP, de este modo el sistema va determinando con cada traza analizada el estado de una llamada hasta que son analizadas todas las trazas correspondientes.

En la maquina de estados se define un estado inicial *Nueva* (llamada), once estados intermedios y dos estados terminales (*Completa*) e *Incompleta*) que determinan la validez de una llamada;

los catorce estados son descritos a continuación:

- **Nueva:** es el estado de inicialización de una llamada al momento de ser analizada por el sistema.
- **Abierta:** indica el inicio de una llamada con un mensaje ISUP del tipo IAM.
- **Conectando:** indica el progreso de la conexión del Suscriptor A con el Suscriptor B en el establecimiento de las comunicaciones.
- **Repicando:** el teléfono del Suscriptor B empieza a sonar indicando la llamada entrante.
- **Establecida:** indica que la llamada ha sido establecida entre ambos suscriptores.
- **En Progreso:** indica que la llamada ha sido establecida y está en progreso.
- **Suspendida:** indica que la llamada ha sido suspendida y o puesta en espera.
- **Restablecida:** indica que la llamada ha sido restablecida luego de ser puesta en espera.
- **Desconectando:** indica que la llamada ha iniciado el proceso de desconexión el cual puede ser de forma regular o irregular.
- **Completa:** indica que la llamada ha sido desconectada exitosamente de forma regular.
- **Incompleta:** indica que la llamada ha sido desconectada exitosamente de forma irregular.
- **Reset:** indica que la llamada ha tenido problemas de conexión y ha iniciado el proceso de re-inicialización del canal para proceder con la desconexión de forma irregular.
- **Irregular:** indica que la llamada ha tenido problemas de señalización en el cual se capturan trazas hasta completar la desconexión, este estado es fundamental para la revisión de las llamadas irregulares.
- **Probando OOS:** indica que la llamada ha tenido problemas de conexión y ha iniciado el proceso de pruebas para determinar si el canal está fuera de servicio.

Las transiciones están asociadas a los mensajes ISUP3.13 de SS7 para cada uno de los estados definidos anteriormente, las transiciones son descritas a partir del estado origen a continuación:

- **Nueva:** hay una transición hacia el **estado Abierta** y es activada por un mensaje *IAM* que indica el inicio de una llamada.

- **Abierta:** con el mensaje *COT* la llamada pasa al estado **Conectando**; con el mensaje *ACM* la llamada alcanzó al destino y está repicando, pasa al estado **Conectando**; con el mensaje *REL* se solicita la liberación del canal y se espera por confirmación para el cierre en el estado **Desconectando**; con el mensaje *RSC* se solicita la inicialización del canal en caso de problemas en el enlace y la llamada pasa al estado **Reset**; con el mensaje *IAM* la llamada pasa a ser inválida porque ya fue inicializada y finalmente el mensaje *SLTM* indica que se realiza una prueba de OSS en el estado **ProbandoOOS**.
- **Conectando:** con el mensaje *REL* se solicita la liberación del canal y se espera por confirmación para el cierre en el estado **Desconectando**; con el mensaje *RSC* se solicita la inicialización del canal en caso de problemas en el enlace y la llamada pasa al estado **Reset**; con el mensaje *ACM* la llamada alcanzó al destino y está repicando, pasa al estado **Conectando**; el resto de los mensajes que puedan llegar generan una transición a **Irregular**, exceptuando el mensaje *SLTM* indica que se realiza una prueba de OSS en el estado **ProbandoOOS**.
- **Repicando:** con el mensaje *REL* se solicita la liberación del canal y se espera por confirmación para el cierre en el estado **Desconectando**; con el mensaje *ANM* la llamada pasa a **Establecida**; con el mensaje *RSC* se solicita la inicialización del canal en caso de problemas en el enlace y la llamada pasa al estado **Reset**; con el mensaje *CPG* se pasa la llamada al estado **En Progreso** y con el resto de los mensajes la llamada se hace **Irregular**, exceptuando el mensaje *SLTM* indica que se realiza una prueba de OSS en el estado **ProbandoOOS**.
- **Establecida:** con el mensaje *REL* se solicita la liberación del canal y se espera por confirmación para el cierre en el estado **Desconectando**; con el mensaje *SUS* se solicita poner en espera la llamada para mantener el canal mientras se atiende otra llamada para pasar al estado de **Suspendida**; con el mensaje *RSC* se solicita la inicialización del canal en caso de problemas en el enlace y la llamada pasa al estado **Reset**; con el mensaje *CPG* se pasa la llamada al estado **En Progreso** y con el resto de los mensajes la llamada se hace **Irregular**, exceptuando el mensaje *SLTM* indica que se realiza una prueba de OSS en el estado **ProbandoOOS**.
- **En Progreso:** con el mensaje *CPG* se mantiene la llamada actual en el mismo estado reiniciando los temporizadores de control; con el mensaje *RSC* se solicita la inicialización del canal en caso de problemas en el enlace y la llamada pasa al estado **Reset**; con el mensaje *REL* se solicita la liberación del canal y se espera por confirmación para el cierre en el estado **Desconectando** y con el resto de los mensajes la llamada se hace **Irregular**, exceptuando el mensaje *SLTM* indica que se realiza una prueba de OSS en el estado **ProbandoOOS**.
- **Suspendida:** con el mensaje *REL* se solicita la liberación del canal y se espera por confirmación para el cierre en el estado **Desconectando**; con el mensaje *RSC* se solicita la inicialización del canal en caso de problemas en el enlace y la llamada pasa al estado **Reset**; con el mensaje *RES* se solicita la restauración de la llamada en espera para pasar al estado de **Restablecida** y con el resto de los mensajes la llamada se hace **Irregular**, exceptuando el mensaje *SLTM* indica que se realiza una prueba de OSS en el estado **ProbandoOOS**.

- **Restablecida:** con el mensaje *ANM* la llamada pasa a **Establecida** y con el resto de los mensajes la llamada se hace **Irregular**, exceptuando el mensaje *SLTM* indica que se realiza una prueba de OSS en el estado **ProbandoOOS**.
- **Desconectando:** con el mensaje *REL* se vuelve a solicitar la liberación del canal y se espera por confirmación para el cierre manteniendo en el mismo estado; con el mensaje *RLC* se confirma la liberación del canal y el fin de la llamada para pasar a los estados finales **Completa** o **Incompleta** si la llamada viene con tramas irregulares y con el resto de los mensajes la llamada se hace **Irregular**, exceptuando el mensaje *SLTM* indica que se realiza una prueba de OSS en el estado **ProbandoOOS**.
- **Completa:** es un estado terminal sin transiciones y permite validar una llamada como regular.
- **Incompleta:** es un estado terminal sin transiciones y permite validar una llamada como irregular.
- **Reset:** con el mensaje *REL* se solicita la liberación del canal y se espera por confirmación para el cierre en el estado **Desconectando** y con el resto de los mensajes la llamada se hace **Irregular**.
- **Irregular:** en este estado se capturan todas las trazas con mensajes irregulares manteniendo el mismo estado, exceptuando el mensaje *SLTM* indica que se realiza una prueba de OSS en el estado **ProbandoOOS**.
- **Probando OOS:** el mensaje *SLTM* indica que se realiza una prueba de OSS para mantener el mismo estado, las pruebas finalizan con un mensaje *SLTA* para finalizar las pruebas y pasar la llamada al estado **Completa**.

La *Figura 5.15* muestra como éstos mensajes activan las transiciones entre los estados descritos en el Capítulo II, Sección XIII (3.13) permitiendo determinar si una llamada ha sido finalizada regular o irregularmente.

En el análisis las llamadas son inicializadas en la máquina de estados y luego de procesar los mensajes ISUP se marcan banderas lógicas que indican el estado de la llamada luego de ser analizada. Esta información es resumida por la aplicación web que muestra el resultado del análisis agrupando las trazas por llamadas regulares e irregulares, junto con un gráfico de barras con la distribución de los mensajes ISUP procesados.

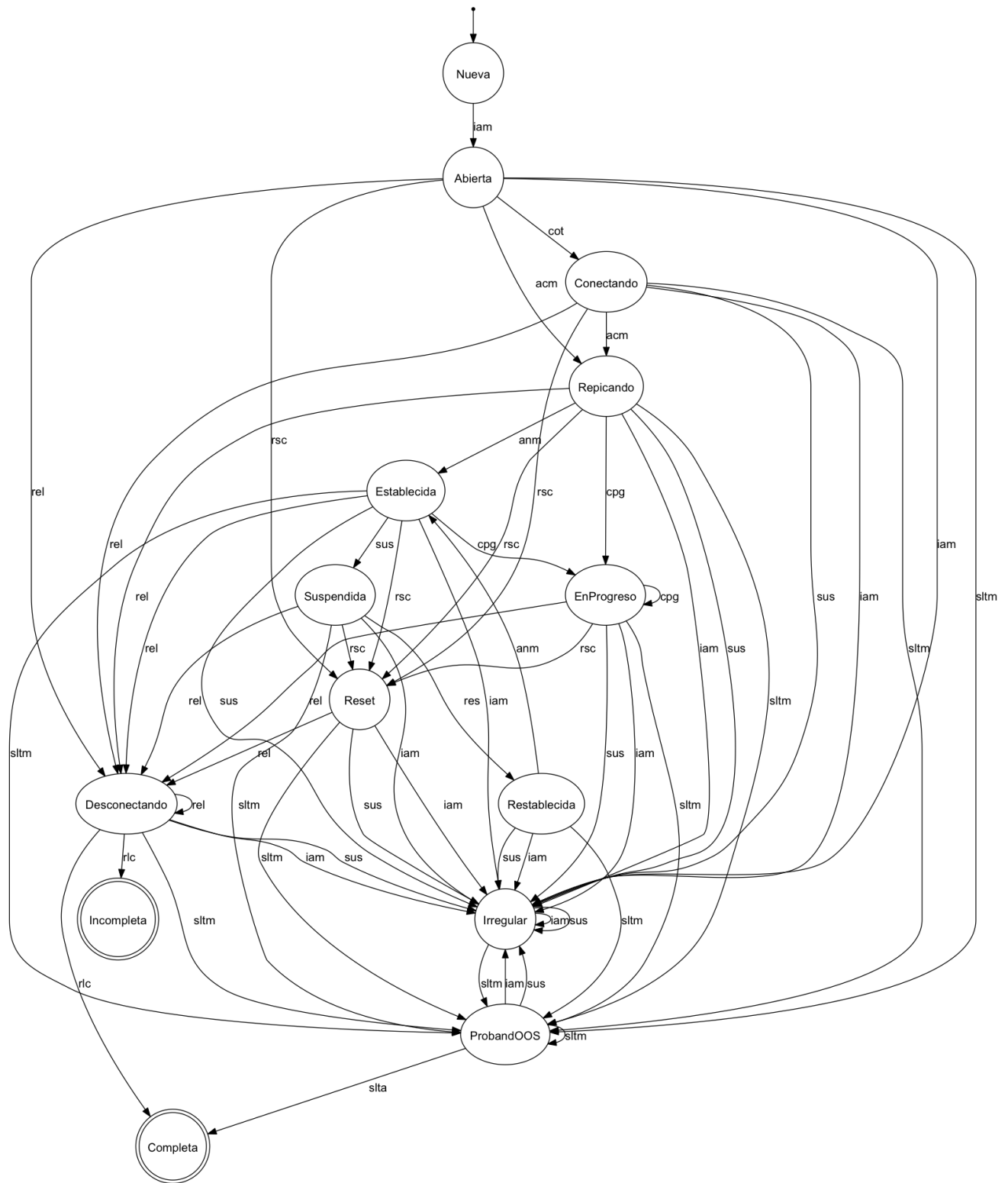


Figura 5.15: Máquina de Estados de SAT-ISUPSS7

Capítulo 6

Pruebas de Rendimiento y Confiabilidad

En este capítulo se muestra el análisis realizado sobre los archivos de trazas y los resultados obtenidos para cada caso de estudio. Se describen a continuación los archivos de entrada para cada escenario de prueba, el resultado esperado y el resultado obtenido en la aplicación web.

6.1. Escenarios de pruebas

Con la idea de validar los resultados obtenidos al analizar un archivo de trazas ISUP, se preparan muestras cuyas trazas tienen un resultado predefinido, es decir; se busca validar que los mensajes contenidos en el archivo de trazas reproduzcan un escenario en particular que está asociado a un resultado conocido.

6.1.1. Definición de los Escenarios

Partiendo de esto se preparan cinco archivos de entradas diferentes para reproducir los siguientes escenarios:

1. **Validación de resultados por llamadas regulares:** en este escenario se prepara un archivo de entrada con mensajes ISUP válidos para obtener después del análisis la agrupación de las trazas en al menos **10 llamadas válidas** con la gráfica de distribución de mensajes. En este caso el valor de las llamadas incompletas debería ser igual a cero validando así la lógica empleada en el algoritmo de análisis.
2. **Validación de resultados por llamadas irregulares:** en este escenario se prepara un archivo de entrada con mensajes ISUP válidos y no válidos para obtener después del análisis, la agrupación de las trazas en al menos **10 llamadas inválidas** con la gráfica de distribución de mensajes. En este caso el valor de las llamadas completas debería ser igual a cero validando la lógica empleada en el algoritmo de análisis.

3. **Validación de resultados con un volumen de dos mil trazas:** en este escenario se preparan dos archivos de entradas con mensajes ISUP provenientes de una muestra real con una cantidad de **2000** trazas aproximadamente, cada archivo contendrá llamadas completas e incompletas para validar los resultados de los escenarios #1 y #2.
4. **Prueba de Rendimiento con un volumen de diez mil trazas:** en este escenario se prepara un archivo de entrada con mensajes ISUP provenientes de una muestra real con una cantidad de **10000** trazas para evaluar el rendimiento del sistema en la generación de resultados.

6.1.2. Análisis de Trazas para cada Escenario

A continuación se describirán los hallazgos encontrados en el análisis de las trazas para cada escenario, se mostrarán los resultados de la corrida inicial y los posibles correctivos realizados en la implementación para obtener la salida esperada de los casos base. (Escenarios #1 y #2)

Los archivos de trazas utilizados para las pruebas y análisis de resultados provienen de tráfico real, han sido generados por la versión de línea de comandos del SAT-ISUPSS7 ampliamente probada y validada en entornos de producción con análisis de trazas que superan los 2 millones de registros.

Esta versión del sistema tiene como función generar dos archivos de texto, uno para llamadas válidas y otro para llamadas inválidas, éstos serán utilizados como entradas para la validación de los escenarios base de la aplicación web.

Escenario #1 - Validación de resultados por llamadas regulares

Este escenario comprende el primer caso base que permite validar el funcionamiento de la aplicación, para ello se realiza la carga del archivo de trazas en el sistema que contiene **diez llamadas válidas o completas**.

Como se puede observar en la *Figura 6.1*, el sistema ha detectado dos llamadas incompletas en un archivo que recrea un **escenario con diez llamadas válidas**, al momento de verificar las trazas de esas dos llamadas se observa que el mensaje *CPG* es un factor común. De acuerdo a esto se procede a validar las transiciones de ese mensaje para verificar si han sido definidas de forma correcta en la máquina de estados.

En este caso se nota la falta de una transición que permita mantener la llamada válida para el caso de obtener mensajes *CPG* consecutivos, para solucionar el inconveniente se hace el correctivo a nivel de sistema y se vuelven a realizar las pruebas para validar los cambios.

Los resultados de la *Figura 6.2* son los esperados de acuerdo al archivo de entrada, ahora se tienen diez llamadas completadas para un archivo que recrea este escenario. Al comparar los resultados

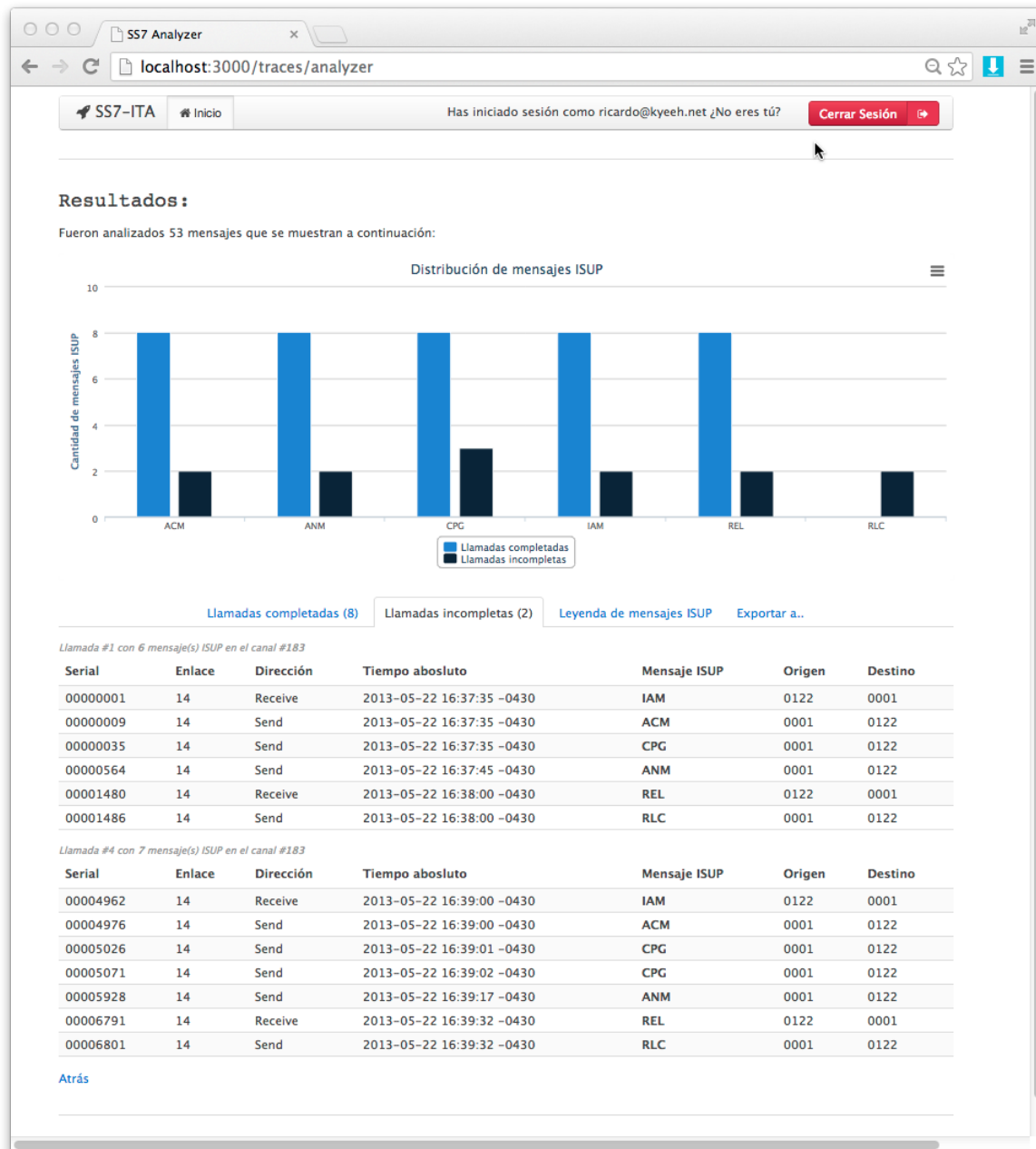


Figura 6.1: Revisión del Escenario #1

se observa que al definir las transiciones necesarias, el SAT-ISUPSS7 funciona correctamente para la validación de llamadas completas.

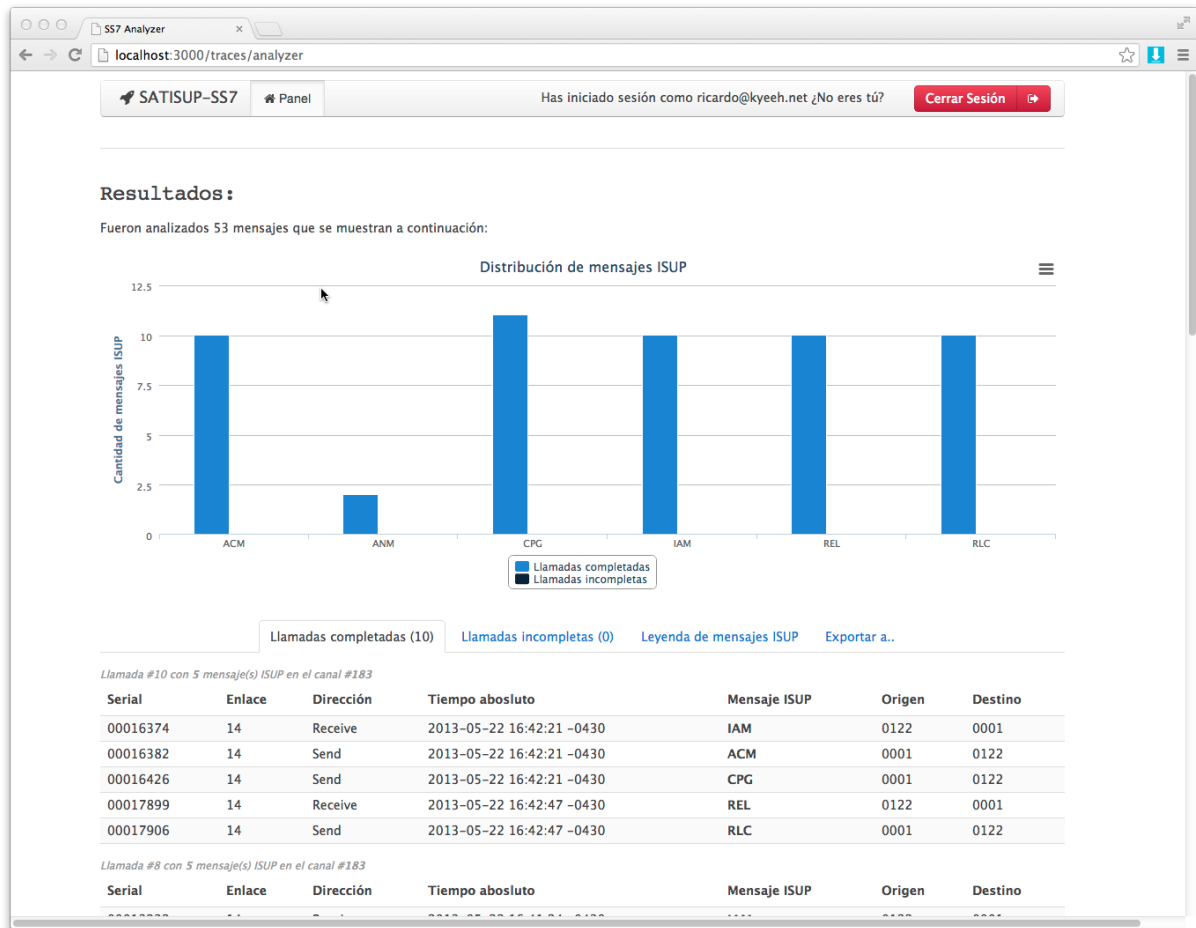


Figura 6.2: Validación del Escenario #1

Escenario #2 - Validación de resultados por llamadas irregulares

Este escenario comprende el segundo caso base que permite validar el funcionamiento de la aplicación, para ello se realiza la carga del archivo de trazas en el sistema que contiene **diez llamadas inválidas o incompletas**.

Como se puede observar en la *Figura 6.3*, el sistema ha detectado diez llamadas incompletas en un archivo que recrea un **escenario con diez llamadas inválidas**, al momento de verificar las trazas se observa que los resultados son los esperados de acuerdo al archivo de entrada. De esta forma se puede corroborar que el SAT-ISUPSS7 funciona correctamente para la validación de llamadas incompletas.

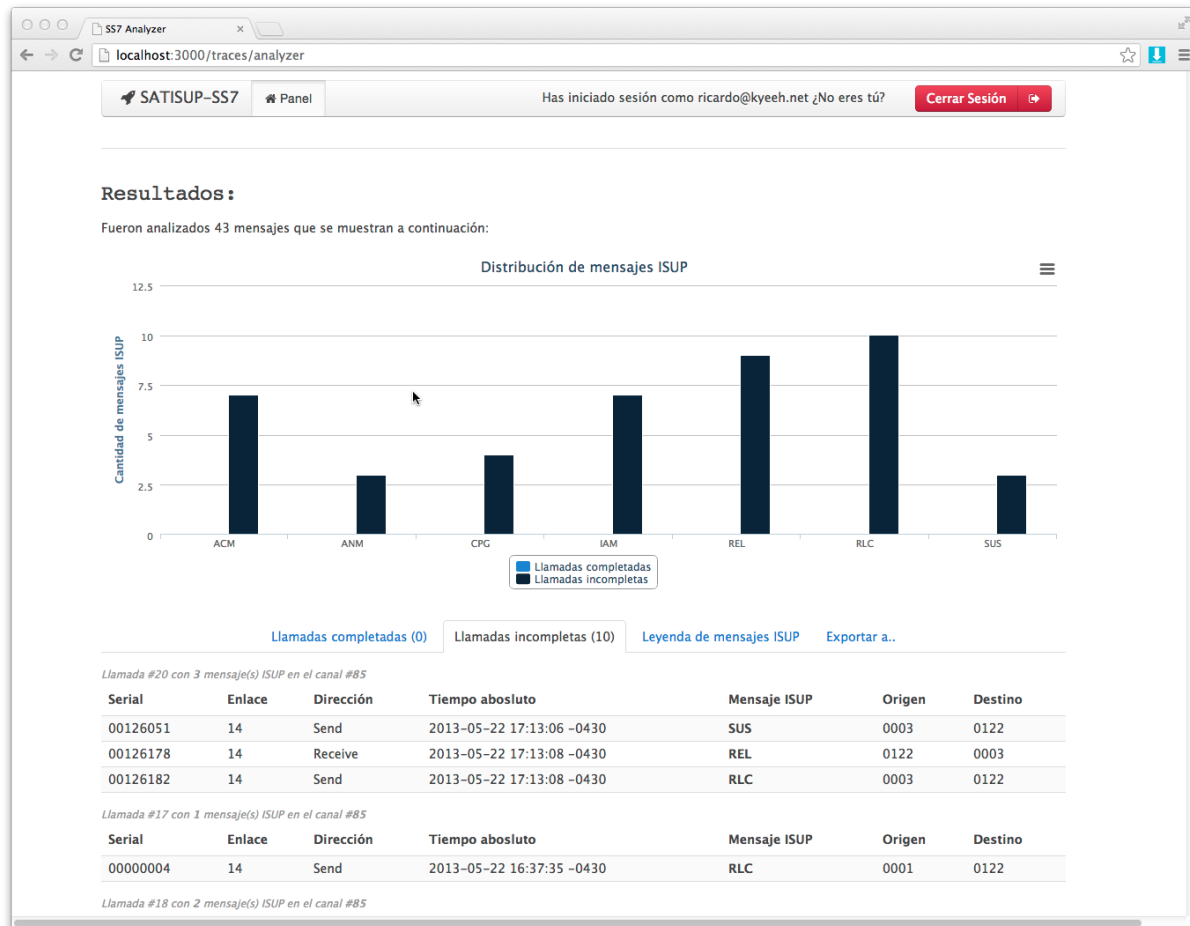


Figura 6.3: Revisión del Escenario #2

Escenario #3 - Validación de resultados con un volumen de datos de 2000 registros

Este escenario comprende una prueba de rendimiento, desempeño y validación de los resultados obtenidos en los escenarios #1 y #2, en este caso se realizará la carga de un archivo de entrada con aproximadamente 2000 registros a ser procesados en un tiempo prudencial.

Las mediciones de desempeño serán tomadas a través de la aplicación con un tiempo inicial que se toma a partir de la carga del archivo en el servidor, y un tiempo final tomado al momento de mostrar los resultados en el navegador.

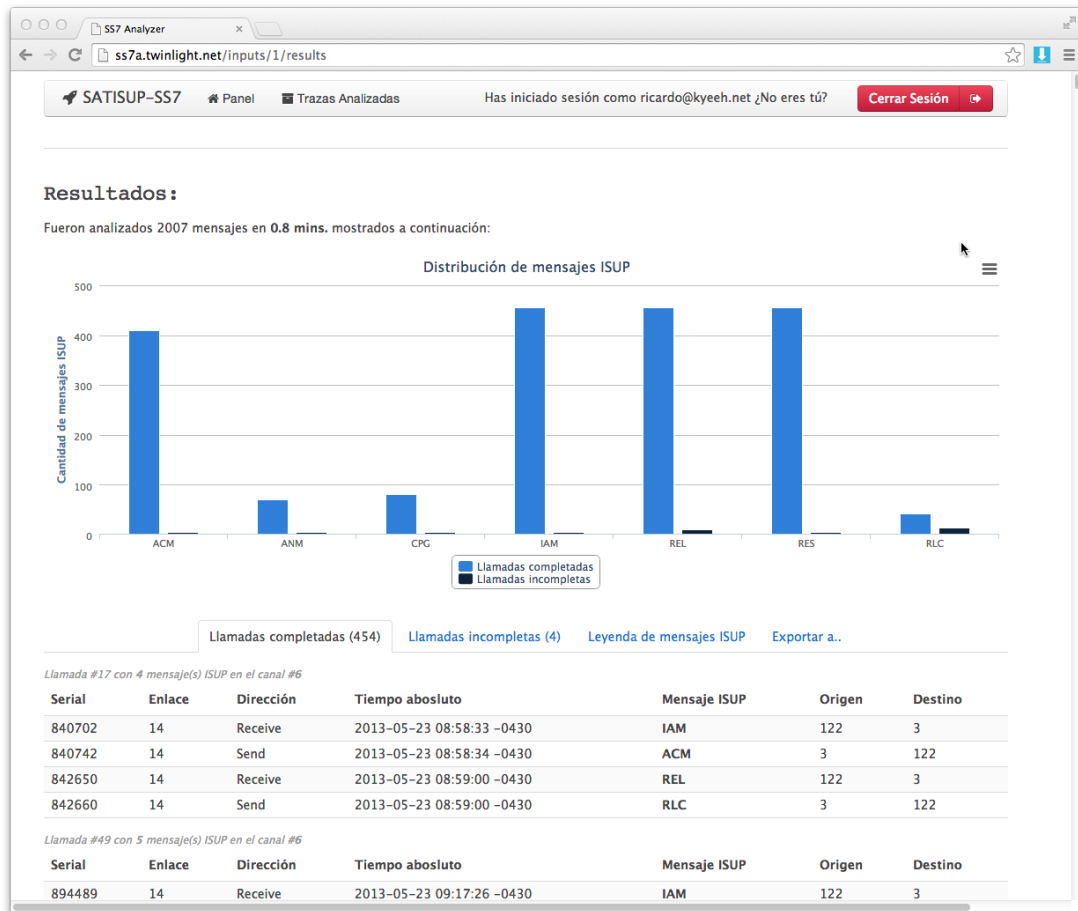


Figura 6.4: Revisión del Escenario #3 - Llamadas válidas

La *Figura 6.4* muestra los resultados emitidos por SAT-ISUPSS7 para un archivo con **sólo** llamadas válidas, aunque en este caso fueron detectadas cuatro llamadas incompletas que cumplen con las transiciones regulares definidas por la ITU-T[5].

En este caso el sistema marca las llamadas como incompletas porque su comportamiento puede ser irregular dentro de la implementación del servicio y puede que estas llamadas requieran una revisión posterior para verificar su naturaleza.

Posteriormente se repite la validación del escenario #3 para un archivo con llamadas inválidas, la *Figura 6.5* muestra los resultados emitidos por SAT-ISUPSS7 para este caso:

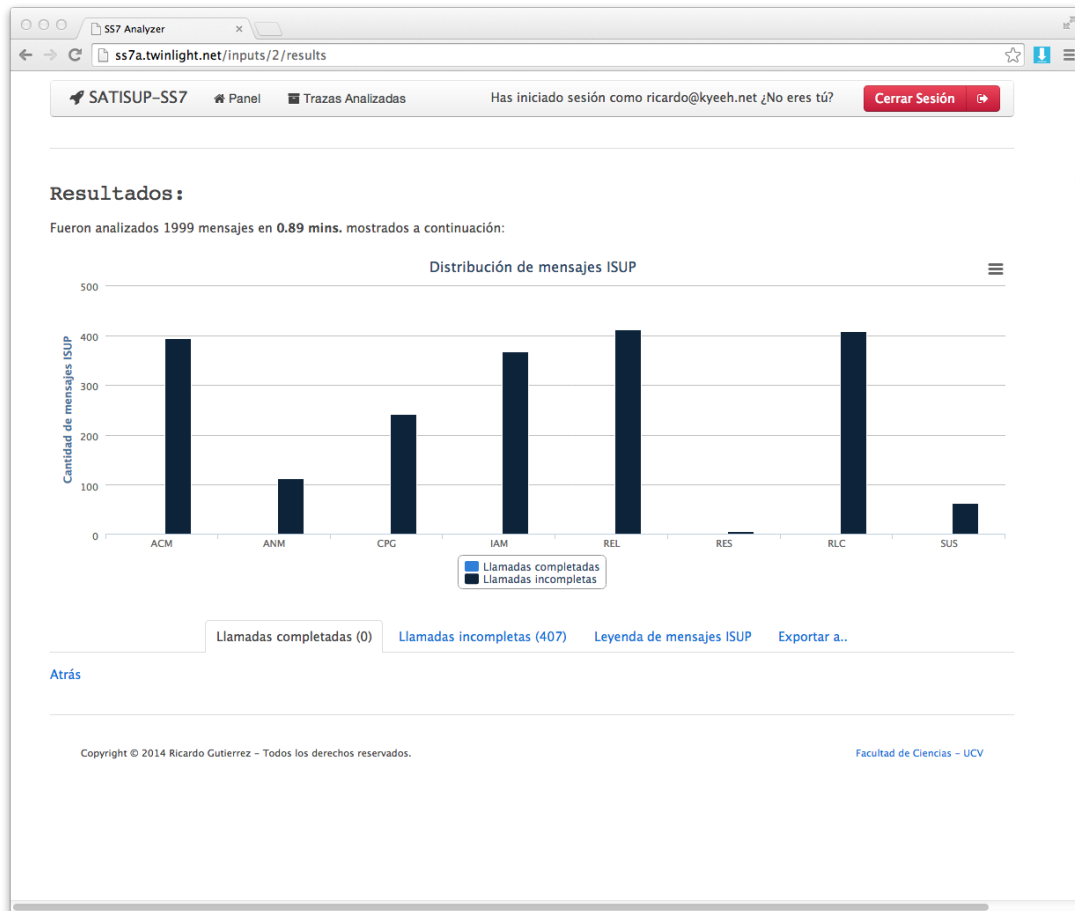


Figura 6.5: Revisión del Escenario #3 - Llamadas inválidas

En este caso los resultados obtenidos corresponden en su totalidad con el archivo de trazas analizado y en ambos casos se realizó en análisis de las trazas en menos de un minuto.

Escenario #4 - Prueba de desempeño con un volumen de datos de 10000 registros

Este escenario comprende una prueba de rendimiento y desempeño del sistema con un archivo que contiene llamadas completas e incompletas con una longitud de 10000 tramas de tráfico real a ser analizadas. La *Figura 6.6* muestra los resultados emitidos por SAT-ISUPSS7 para este caso:

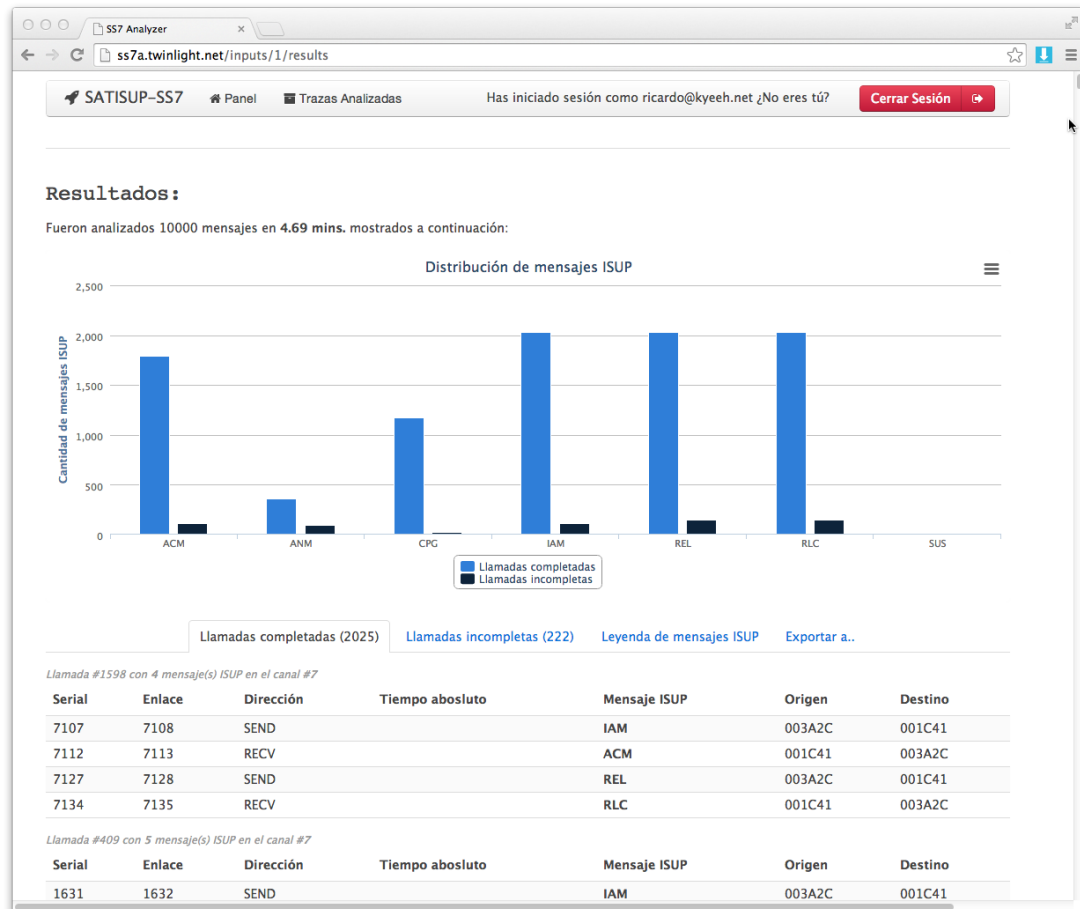


Figura 6.6: Revisión del Escenario #4 - Prueba de desempeño

En este caso los resultados muestran llamadas completas (2025) e incompletas (222) para un archivo con trazas de tráfico real, la agrupación de llamadas incompletas permite la ubicación de la causa raíz del problema haciendo mucho más rápida y eficiente la depuración del servicio. Los resultados fueron obtenidos en un tiempo de 4mins. y 42seg.

6.1.3. Generación de Reportes

Los resultados generados por SAT-ISUPSS7 pueden ser exportados a archivos PDF y XLS con la idea de hacer exportable la solución, el contenido emula la presentación web haciendo un resumen del análisis mostrando las trazas analizadas por agrupación de llamadas completas e incompletas.

El archivo PDF permite generar resultados en un formato de sólo lectura tal y como se muestra en la *Figura 6.7*:

**Sistema Analizador de Tramas de la capa ISUP del Protocolo de Señalización No.7
SATISUP-SS7 | ss7a.twinlight.net**

RESUMEN	
Archivo analizado: real10k.txt	Cantidad de trazas analizadas: 10000
Llamadas válidas: 2025	Llamadas inválidas: 222

Descripción de llamadas válidas:

Llamada #1 con 4 mensajes ISUP en el canal #7:

Serial	Enlace	Dirección	Tiempo Absoluto	Mensaje ISUP	Origen	Destino
7107	7108	SEND		IAM	003A2C	001C41
7112	7113	RECV		ACM	001C41	003A2C
7127	7128	SEND		REL	003A2C	001C41
7134	7135	RECV		RLC	001C41	003A2C

Llamada #2 con 5 mensajes ISUP en el canal #7:

Serial	Enlace	Dirección	Tiempo Absoluto	Mensaje ISUP	Origen	Destino
1631	1632	SEND		IAM	003A2C	001C41
1635	1636	RECV		ACM	001C41	003A2C
1709	1710	RECV		ANM	001C41	003A2C
2365	2366	SEND		REL	003A2C	001C41
2370	2371	RECV		RLC	001C41	003A2C

Llamada #3 con 4 mensajes ISUP en el canal #7:

Serial	Enlace	Dirección	Tiempo Absoluto	Mensaje ISUP	Origen	Destino
9672	9673	SEND		IAM	003A2C	001C41
9700	9701	RECV		ACM	001C41	003A2C
9740	9741	SEND		REL	003A2C	001C41
9746	9747	RECV		RLC	001C41	003A2C

Llamada #4 con 5 mensajes ISUP en el canal #7:

Serial	Enlace	Dirección	Tiempo Absoluto	Mensaje ISUP	Origen	Destino
7899	7900	SEND		IAM	003A2C	001C41
7918	7919	RECV		ACM	001C41	003A2C

Figura 6.7: Resultados en PDF

El archivo XLS permite generar resultados en un formato de lectura y escritura orientado a los trabajos de análisis de trazas para la depuración de llamadas, la salida es mostrada en la *Figura 6.8*:

Sistema Analizador de Tramas de la capa ISUP del Protocolo de Señalización No.7 (SATISUP-SS7) | ss7a.twinlight.net

RESUMEN

Archivo analizado: real10k.txt
 Cantidad de trazas analizadas: 10000
 Llamadas válidas: 2025
 Llamadas inválidas: 222

Descripción de llamadas válidas:

Llamada #1 con 4 mensajes ISUP en el canal #:

Serial	Enlace	Dirección	Tiempo Absoluto	Mensaje ISUP	Origen	Destino
7107	7108	SEND		IAM	003A2C	001C41
7112	7113	RECV		ACM	001C41	003A2C
7127	7128	SEND		REL	003A2C	001C41
7134	7135	RECV		RLC	001C41	003A2C

Llamada #2 con 5 mensajes ISUP en el canal #:

Serial	Enlace	Dirección	Tiempo Absoluto	Mensaje ISUP	Origen	Destino
1631	1632	SEND		IAM	003A2C	001C41
1635	1636	RECV		ACM	001C41	003A2C
1709	1710	RECV		ANM	001C41	003A2C
2365	2366	SEND		REL	003A2C	001C41
2370	2371	RECV		RLC	001C41	003A2C

Llamada #3 con 4 mensajes ISUP en el canal #:

Serial	Enlace	Dirección	Tiempo Absoluto	Mensaje ISUP	Origen	Destino
9672	9673	SEND		IAM	003A2C	001C41
9700	9701	RECV		ACM	001C41	003A2C
9740	9741	SEND		REL	003A2C	001C41
9746	9747	RECV		RLC	001C41	003A2C

Llamada #4 con 5 mensajes ISUP en el canal #:

Serial	Enlace	Dirección	Tiempo Absoluto	Mensaje ISUP	Origen	Destino
7899	7900	SEND		IAM	003A2C	001C41
7918	7919	RECV		ACM	001C41	003A2C
7951	7952	RECV		ANM	001C41	003A2C

Figura 6.8: Resultados en XLS

Conclusiones

Con el desarrollo de este TEG se ha estudiado y analizado los diagramas de transición de estados para los mensajes ISUP de SS7 en sus diversas primitivas y procedimientos, teniendo como foco principal el flujo de mensajes para el control de una llamada telefónica.

A partir de esta investigación se realiza la inferencia o deducción de una máquina de estados global que permite definir las bases del algoritmo de análisis de tramas ISUP capturadas en troncales telefónicas con tráfico real haciendo abstracción de los mensajes de control del canal para enfocar esfuerzos en el flujo de mensajes de control de la llamada telefónica.

El desarrollo del sistema contempla distintos métodos que permiten la carga y análisis de tramas en una sola pasada para una captura de tráfico determinada, permitiendo agrupar de las tramas ISUP por llamada telefónica haciendo uso de la información de control implícita.

El análisis de estos datos proporciona información muy valiosa para la detección de anomalías en la implementación de un servicio de llamadas telefónicas con SS7, se ha desplegado una Aplicación Web (SAT-ISUPSS7) que permite mostrar los resultados obtenidos agrupando trazas por llamadas válidas e inválidas permitiendo al usuario final el análisis específicos en escenarios puntuales donde haya sido detectado un comportamiento irregular.

Cabe destacar que las capturas de mensajes ISUP pueden contener un gran volumen de datos ya que la información es obtenida por la captura de tráfico en troncales que interconectan centrales telefónicas. Teniendo esto en consideración, se ha desarrollado un sistema que permite realizar el análisis de los datos de entrada en un tiempo prudencial evitando así la acumulación de errores y largos tiempos de espera producto del trabajo manual que pudiera realizarse para obtener los resultados deseados.

6.1.4. Aportes

A partir del estudio y análisis de tramas ISUP y los diagramas de estado definidos para SS7 se ha determinado una máquina de estados que permite hacer abstracción de los mensajes de control del enlace para enfocar la solución en el flujo de mensajes de control de una llamada telefónica.

Esta abstracción ha permitido diseñar algoritmos que permiten determinar a partir de un archivo de entrada, la cantidad de llamadas válidas e inválidas que han sido realizadas en un tiempo determinado, esto a partir de la captura de tráfico de mensajes ISUP.

Los resultados son mostrados vía web con la agrupación de tramas por llamada efectuada junto con la distribución de los mensajes ISUP analizados; esta información puede ser descargada de la aplicación web mediante archivos PDF o XLS. De igual forma se pueden recuperar los resultados de los archivos de tramas previamente analizados ya que la aplicación lleva el registro de esos datos por cada cuenta de usuario.

6.1.5. Limitaciones y Trabajos Futuros

Como limitaciones de la aplicación web desarrollada en este TEG se pueden mencionar los siguientes aspectos:

- Los archivos de entrada son de texto plano y en ocasiones el volumen de datos es muy grande, sería de gran utilidad agregar compatibilidad para procesar archivos comprimidos. Eso mejoraría de gran forma los tiempos de ejecución ya que minimiza el tamaño de los archivos subidos al servidor.
- El algoritmo de análisis está enfocado en el rendimiento de la aplicación, por esta razón las tramas son analizadas con la lectura del archivo de forma secuencial. Esto permite minimizar los tiempos de ejecución pero a la vez dificulta la validación de los datos de entrada, por esta razón en ocasiones se pueden obtener comportamientos irregulares en la carga de los datos del archivo de entrada.

Como trabajos futuros se pueden realizar los siguientes desarrollos:

- Ampliar las capacidades de la máquina de estados para el análisis de todas las primitivas sin limitar la aplicación al flujo de mensajes isup para el control de llamadas.
- Definir máquinas de estados para cada componente de la arquitectura de SS7 con el fin de hacer análisis de tráfico de toda la pila de protocolos.
- Agregar soporte para otros protocolos con la idea de crear una herramienta que permita analizar el tráfico de otros servicios u aplicaciones.

Acrónimos

ACM

Address Complete Message. 37

ANM

Answer Message. 37

ANSI

American National Standards Institute. 36

API

Application Software Interface. 62

BIB

Backward Indicator Bit. 31

BLO

Blocking Message. 38

BSN

Backward Sequence Number. 31

CAS

Channel-Associated Signaling. 20, 21, 24

CC

Connection Confirmation. 43, 45

CCBS

Call Completion to a Busy Subscriber. 45

CCNR

Completion of Calls on No Reply. 45

CCS

Common-Channel Signaling. 21, 23, 24

CDR

Call Detail Record. 13

CFN

Confusion Message. 37

CIC

Call Instance Code. 32, 36, 37

CK

Check Bits. 32

CON

Connect Message. 37

COT

Continuity Message. 37

CPC

call processing control. 39

CPG

Call Progress Message. 37, 82

CQR

Circuit Group Response Message. 37

CR

Connection Request. 42, 44, 45

CSC

Circuit Supervision Control. 39

CSS

Cascading Style Sheets. 61–63

DOM

Document Object Model. 62

DP

Destination Point. 22

DPC

Destination Point Code. 32, 33, 36, 39

DS-0

Digital Signal 0. 25

DT1

Data Form 1. 43

DT2

Data Form 2. 43

DUP

Data User Part. 29

ER

Entidad - Relación. 49, 55

FIB

Forward Indicator Bit. 31

FISU

Fill-In Signal Unit. 30, 32

FOT

Forward Transfer Message. 38

FRJ

Facility Reject Message. 37

FSN

Forward Sequence Number. 31

GRA

Circuit Group Reset Acknowledgement Message. 37

GTT

Global Title Translation. 39

HTML

Hyper Text Markup Language. 61, 63

IAM

Initial Address Message. 36, 77

IDE

Integrated Development Environment. 64–66

INF

Information Message. 37

INR

Information Request Message. 38

ISDN

Integrated Services Digital Network. 25–27, 45

ISUP

ISDN User Part. 11, 16, 26, 29, 32, 35, 36, 38, 39, 43–45, 49–52, 54, 56–59, 74–77, 79, 81, 82, 91, 92

ITU-T

International Telecommunication Union - Telecommunication. 17, 24, 25, 29, 34–36, 39, 45, 56, 86

KPI

Key Performance Indicators. 13

LI

Length Indicator. 29, 31, 32

LNP

Local Number Portability. 27

LSSU

Link Status Signal Unit. 29, 32

Modem

Modulator - Demodulator. 18

MSU

Message Signaling Unit. 29, 31, 32

MTP

Message Transfer Part. 8, 26, 27, 29, 31, 33, 39, 40

MTP-1

Message Transfer Part Level 1. 25

MTP-2

Message Transfer Part Level 2. 25

MTP-3

Message Transfer Part Level 3. 26

MVC

Modelo Vista Controlador. 60, 71

NE

Network Elements. 27

NSP

Network Service Part. 40

OMAP

Operations, Maintenance, and Administration Part. 27

OP

Originating Point. 22

OPC

Originating Point Code. 32, 33, 36

OSI

Open Systems Interconnection Model. 25, 29, 33, 45

OSP

Open Settlement Protocol. 39

OSS

Operations Support Systems. 49, 54, 77–79

PCM

Pulse Code Modulation. 20

PSTN

Public Switched Telephone Network. 27, 35

REL

Release Message. 38

RES

Resume. 38

RLC

Release Complete Message. 38, 43, 76

RLSD

Released. 43

RoR

Ruby on Rails. 60, 61, 65, 66, 73, 74

RVM

Ruby Version Manager. 69

SAM

Subsequent Address Message. 36

SAT-ISUPSS7

SAT-ISUPSS7: Sistema Analizador de Tramas de la capa ISUP del Protocolo de Señalización No. 7 (SS7). 63, 69, 71–73, 82–84, 86–88, 91

SCCP

Signaling Connection Control Part. 26, 27, 29, 35, 39–41, 43–45, 47

SCP

Service Control Point. 19, 22, 28, 45

SF

State Field. 32

SI

Service Indicator. 31

SIF

Signaling Information Field. 32, 36

SIO

Service Information Octet. 31, 33

SLA

Service Level Agreement. 13

SLS

Signalling Link Selection. 32, 33

SMBD

Sistemas Manejadores de Bases de Datos. 54

SP

Signalling Point. 23, 26, 27, 33, 45

SPC

Stored Program Control. 25

SPRC

Signalling Procedure Control. 39

SS7

Signaling System No. 7. 7, 14–17, 23–27, 29, 35, 39, 45, 49, 51, 54, 56, 59, 63, 75, 77, 91, 92

SSF

Subservice Field. 32

SSP

Service Switching Point. 22, 28, 35, 45

STP

Signalling Transfer Point. 22, 23, 28

SU

Signaling Unit. 29

SUS

Suspend. 38

TC

Transaction Capabilities. 35, 45–47

TCAP

Transaction Capabilities Application Part. 26, 40, 43, 45

TEG

Trabajo Especial de Grado. 49, 91, 92

TUP

Telephone User Part. 29, 32, 35

UBL

UnBlocking Message. 38

UDT

Unid Data. 40

UDTS

Unid Data Service. 41

UP

User Part. 8, 29, 35, 36, 39

USR

User to User Information Message. 38

W3C

Wide Web Consortium. 61

XUDT

Extended Unid Data. 41, 42

XUDTS

Extended Unid Data Service. 41

Bibliografía

- [1] C.U.W., *Sistema de señalización 7 por canal común*, (2000).
- [2] Network Working Group, *Integrated services digital network (isdn) user part (isup) to session initiation protocol (sip) mapping*, <http://tools.ietf.org/html/rfc3398>, December 2002.
- [3] David Heinemeier Hansson, *Ruby on rails - framework para el desarrollo web*, 2013.
- [4] ———, *Rails guides*, <http://guides.rubyonrails.org/>, 2014.
- [5] ITU-T, *International telephone connections and circuits – general recommendations on the transmission quality for an entire international telephone connection*, Tech. report, INTERNATIONAL TELECOMMUNICATION UNION, May 2003.
- [6] Canonical Ltd., *Ubuntu server edition*, <http://www.ubuntu.com/server>, Octubre 2013.
- [7] Sublime HQ Pty Ltd, *Editor de textos sublime*, <http://www.sublimetext.com/>, 2013.
- [8] Dr. Carlos Fernández Collado y Dra. Pilar Baptista Lucio M. en C. Roberto Hernández Sampieri, *Metodología de la investigación*, no. ISBN 968-422-931-3, MCGRAW-HILL, Enero 1997.
- [9] Ruby Community Members, *Lenguaje de programación ruby*, <https://www.ruby-lang.org/>, Octubre 2014.
- [10] Oracle, *Mysql workbench*, <http://www.mysql.com/products/workbench/>, 2014.
- [11] Phusion, *Phusion passenger*, <https://www.phusionpassenger.com>, Febrero 2014.
- [12] Prof. Eugenio Scalise Prof. Jossie Zambrano, *Aplicaciones en internet*, nd 2012-02 ed., no. ISSN 1316-6239, Centro de Investigación de Ingeniería de Software y Sistemas (ISYS), Universidad Central de Venezuela, Facultad de Ciencias, Escuela de Computación, Marzo 2012.
- [13] Nick Quaranto, *Rubygems*, <http://rubygems.org/>, 2013.
- [14] Rubyonrails.org, *Documentación oficial de ruby on rails*, <http://rubyonrails.org/documentation>, 2014.

-
- [15] Telecomspace.com, *Signalling system 7 (ss7)*, <http://www.telecomspace.com/ss7.html>, May 2012.
- [16] UNIÓN INTERNACIONAL DE TELECOMUNICACIONES, *Sistema de señalización 7 – anexo h: Diagramas de transición de estado*, **SERIE Q: CONMUTACIÓN Y SEÑALIZACIÓN** (1995), no. Q.764.
- [17] Ramses Velasquez, *Arquitectura web*, <http://codehero.co/laravel-4-desde-cero-estructura-del-proyecto/>, Enero 2013.
- [18] Wikipedia.org, *Nginx*, <http://es.wikipedia.org/wiki/nginx>, Febrero 2014.
- [19] ———, *Ruby on rails - framework para el desarrollo web*, Febrero 2014.
- [20] ———, *Zero-copy*, <http://es.wikipedia.org/wiki/zero-copy/>, Febrero 2014.
- [21] Formulación y Evaluación de Proyectos Grupo Blogspot, *Proyecto factible*, <http://proyectofactible6.blogspot.com/>, Junio 2011.
- [22] YOOthemeTM, *Uikit - framework for web design*, <http://www.getuikit.com/>, 2013.