

**AMBIENTE DE DESARROLLO PARA LA IMPLEMENTACIÓN DE NUEVOS
SERVICIOS TELEMÁTICOS SOBRE LA RED MULTISERVICIOS DE EMCALI**

DIANA MARCELA GUEVARA BONILLA

**UNIVERSIDAD AUTONOMA DE OCCIDENTE
FACULTAD DE INGENIERIA
DEPARTAMENTO DE AUTOMATICA Y ELECTRÓNICA
PROGRAMA DE INGENIERIA ELECTRÓNICA
SANTIAGO DE CALI
2008**

**AMBIENTE DE DESARROLLO PARA LA IMPLEMENTACIÓN DE NUEVOS
SERVICIOS TELEMÁTICOS SOBRE LA RED MULTISERVICIOS DE EMCALI**

DIANA MARCELA GUEVARA BONILLA

Pasantía para optar al título de Ingeniero Electrónico

**Asesor en EMCALI EICE ESP
EUGENIO CASTRO MEDINA
Ingeniero Electrónico y Telecomunicaciones**

**Director
OSCAR HERNAN MONDRAGON
Ingeniero Electrónico y Telecomunicaciones
Master in wireless systems and related technologies**

**UNIVERSIDAD AUTONOMA DE OCCIDENTE
FACULTAD DE INGENIERIA
DEPARTAMENTO DE AUTOMATICA Y ELECTRÓNICA
PROGRAMA DE INGENIERIA ELECTRÓNICA
SANTIAGO DE CALI
2008**

Nota de aceptación

Aprobado por el Comité de Grado en cumplimiento de los requisitos exigidos por la Universidad Autónoma de Occidente para optar al título de Ingeniera Electrónica.

Ing. ZEYDA SOLARTE

Jurado

Ing. JUAN DIEGO PULGARIN

Jurado

Santiago de Cali, 27 de junio de 2008

CONTENIDO

	Pág.
GLOSARIO	14
RESUMEN	20
INTRODUCCIÓN	21
1. DESCRIPCIÓN GENERAL DEL PROYECTO	22
1.1. MARCO TEÓRICO	22
1.1.1 Evolución de la red clásica hacia las NGN	23
1.1.2 Arquitectura de las NGN	24
1.1.3 Los servicios en las redes NGN.	26
1.2. ANTECEDENTES	27
1.2.1 Servicios	27
1.3. OBJETIVO GENERAL	28
1.4. OBJETIVOS ESPECÍFICOS	28
1.5. JUSTIFICACIÓN	28
2. PLATAFORMA MULTISERVICIOS DE EMCALI	30
2.1. CAPA DE ACCESO	30
2.2. CAPA DE PAQUETES (CORE)	33
2.2.1 Gateway troncales	34
2.2.2 Gateway de señalización	34
2.3. CAPA DE CONTROL	35
2.4. CAPA DE SERVICIOS	37
2.4.1 Capa de aplicaciones	38

2.4.2	Capa de control de servicio	38
2.4.3	Capa de adaptación	38
2.4.4	Capa de recursos	39
2.5.	COMPONENTES DE LA PLATAFORMA SISTEMA ZXUP10	39
2.5.1	Parlay Gateway	39
2.5.2	Servidor de aplicaciones	40
2.5.3	Servidor de medios	41
2.5.4	Servidor TTS (Text To Speech)	41
2.5.5	Servidor Web	41
2.5.6	Consola de mantenimiento OAM	41
3.	ANÁLISIS DE TECNOLOGÍAS PARA LA CREACIÓN DE SERVICIOS SOBRE LAS REDES NGN	43
3.1.	CREACIÓN DE SERVICIOS EN REDES NGN	46
3.1.1	API (Application Programming Interfaces)	46
3.1.2	Lenguajes de scripting	47
3.1.3	SCE (Service Creation Enviroments)	47
3.2.	EVALUACIÓN DE TECNOLOGÍAS PARA LA CREACIÓN DE NUEVOS SERVICIOS	48
3.2.1	Servicios Web	48
3.2.2	SIP Servlets	50
3.2.3	Jain SIP Lite	51
3.2.4	Voice XML (Voice eXtensible Markup Language)	52
3.2.5	CCXML (Call Control eXtensible Markup Language)	54
3.2.6	JCC (Java Call Control)	56

3.2.7 CPL (Call Processing Language)	57
3.2.8 SCML (Service Creation Markup Language)	58
3.2.9 XTML (eXtensible Telephony Markup Language)	59
3.3. OSA/PARLAY	60
3.3.1 Arquitectura OSA/Parlay	61
3.3.2 SCF's (Capacidades de servicio) de OSA/Parlay	63
3.3.3 Middleware en OSA/Parlay	65
3.3.4 API's de OSA/Parlay	65
3.3.5 Definición de los tipos de interfaces	67
3.3.6 Definición de interfaces entre la aplicación y el framework	68
3.3.7 Definición de interfaces entre la aplicación y las SCF's	69
3.3.8 Uso de sesiones	70
3.3.9 Interfaces y sesiones	70
3.3.10 Interfaces de callback	70
3.3.11 Métodos síncronos y asíncronos	70
3.3.12 Parámetros de salida	71
3.4. MECANISMO DE ACCESO A UN SERVICIO DESDE UNA APLICACIÓN	71
4. CONSIDERACIONES PARA EL DESARROLLO E IMPLEMENTACIÓN DE NUEVOS SERVICIOS SOBRE LA PLATAFORMA MULTISERVICIOS DE EMCALI	79
4.1. SISTEMA OPERATIVO AIX	79
4.2. CLUSTER HACMP	80
4.2.1 Componentes físicos de un cluster HACMP	82
4.2.2 Componentes software de un nodo HACMP	83

4.3. PARLAY CLIENT HUB (PCH)	86
4.3.1 Localización de PCH dentro de la plataforma ZXUP10	86
4.3.2 Funciones de PCH	87
4.4. HERRAMIENTAS PARA EL DESARROLLO DE APLICACIONES	88
4.4.1 Configuración de JBuilder	89
4.5. IMPLEMENTACIÓN DEL SERVICIO Y VERIFICACIÓN EN LA PLATAFORMA DE SERVICIOS	92
4.5.1 Descarga del servicio	93
4.5.2 Descarga de la información de un servicio por medio de un script	93
4.5.3 Descarga de anuncios de voz	93
4.5.4 Descarga de la lógica del servicio	94
5. CAPTURA DE REQUERIMIENTOS	95
5.1. SISTEMA DE INFORMACIÓN Y VISUALIZACIÓN DE TRÁFICO VIAL	96
5.2. SERVICIO DE ÚNICO NÚMERO TELEFÓNICO	96
5.3. DIRECTORIO TELEFÓNICO	97
5.4. SERVICIO DE GUÍA EN LA CIUDAD	98
5.5. SERVICIO DE TABLERO ELECTRÓNICO	98
5.6. SERVICIO DE BACKTONES	99
5.7. LOCALIZACIÓN POR MEDIO DE DISPOSITIVOS MOVILES	99
6. CASO DE ESTUDIO	101
6.1. SERVICIO NP (NUMBER PORTABILITY)	101
6.1.1 Funcionamiento del servicio NP	101

6.1.2 Casos de uso	103
6.2. DISEÑO DEL PROGRAMA PARA EL SERVICIO NP	103
6.3. IMPLEMENTACIÓN DEL SERVICIO	105
6.3.1 Paso 1. Incluir librerías en JBuilder	105
6.3.2 Paso 2. Configuración de la clase principal	107
6.3.3 Paso 3. Configuración del archivo pch.properties y Log4j.properties	108
6.3.4 Paso 4. Desarrollo de código del servicio NP	109
6.3.5 Interpretación del código del servicio NP	112
6.4. VERIFICACIÓN DEL SERVICIO	113
7. CONCLUSIONES	117
BIBLIOGRAFÍA	118
ANEXOS	121

LISTA DE FIGURAS

	Pág.
Figura 1. Arquitectura NGN	25
Figura 2. Modelo en capas de la red multiservicios	30
Figura 3. Topología de la red multiservicios	33
Figura 4. Red de acceso y red de paquetes de la red multiservicios	35
Figura 5. Distribución de cargas de los softswitch de colon (SS1) y guabito (SS2)	37
Figura 6. Arquitectura de la plataforma de servicios ZXUP10	38
Figura 7. Estructura física del ZXUP10	39
Figura 8. Parlay Gateway	40
Figura 9. Arquitectura de referencia P1109	45
Figura 10. Servicios Web con respecto a la arquitectura P1109	49
Figura 11. SIP servlet y contenedor SIP servlet	51
Figura 12. Arquitectura básica para un servicio VoiceXML	53
Figura 13. Aspecto de una aplicación CCXML	55
Figura 14. Arquitectura CCXML	56
Figura 15. Arquitectura SCML	59
Figura 16. Arquitectura OSA/Parlay	61
Figura 17. Service Capability Server	62
Figura 18. SCF's de OSA/Parlay y framework	63
Figura 19. OSA/Parlay API	66
Figura 20. Interfaces OSA/Parlay	68

Figura 21. Diagrama de clases acceso al framework desde la aplicación	72
Figura 22. Acceso al framework desde la aplicación	73
Figura 23. selectEncryptionMethod()	73
Figura 24. Método authenticate()	74
Figura 25. Método authenticationSucceeded()	74
Figura 26. Método authenticate() del lado del cliente	74
Figura 27. Autenticación exitosa	74
Figura 28. Método requestAccess()	75
Figura 29. Método obtainInterface()	75
Figura 30. Diagrama de secuencia para autenticación con el framework	76
Figura 31. Método discoveryService()	76
Figura 32. Selección del servicio	77
Figura 33. Autorización para utilizar el servicio	77
Figura 34. Cluster HACMP	83
Figura 35. Modelo en capas de un nodo en un cluster HACMP	84
Figura 36. Localización de PCH dentro de la plataforma ZXUP10	87
Figura 37. Configuración de librerías de la plataforma zxup10 en JBuilder	89
Figura 38. Configuración de la clase principal	90
Figura 39. Archivos de configuración pch.properties y log4.properties	91
Figura 40. Archivo de configuración pch.properties	92
Figura 41. Herramienta ISM para configuración de los servicios	93

Figura 42. Flujo de información en una llamada	94
Figura 43. Flujo de información del servicio NP	102
Figura 44. Definición de casos de uso para el servicio NP	103
Figura 45. Diagrama de clases para el servicio NP	104
Figura 46. Diagrama de secuencia para el servicio NP	105
Figura 47. Configuración de librerías en JBuilder	106
Figura 48. Como incluir las librerías dentro de JBuilder	107
Figura 49. Configuración de la clase principal para el servicio NP	108
Figura 50. Configuración de PCH-OAM	114
Figura 51. Configuración de OAM	115
Figura 52. Flujo de información durante una llamada en el servicio NP	116

LISTA DE ANEXOS

	Pág.
Anexo A. Especificación de requerimientos para las funciones de administración de los servicios	121
Anexo B. Especificación de requerimientos para la definición de nuevos servicios	134

LISTA DE TABLAS

	Pág.
Tabla 1. Cuadro de comparación de tecnologías	78
Tabla 2. Caso de uso para servicio NP	103

GLOSARIO

API (APPLICATION PROGRAMMING INTERFACE): interfaz de programación de aplicaciones es un conjunto de funciones y procedimientos ofrecidos por una biblioteca para ser utilizados por un software para permitirle el acceso a ciertos servicios. Es la interfaz de comunicación entre componente software. El principal propósito de las API's consiste en proporcionar funciones de uso general facilitando a los programadores no programar todo desde el principio.

ATM (ASYNCHRONOUS TRANSFER MODE): modo de transferencia asíncrona. Es una tecnología de red compartida que ofrece poca latencia y fluctuación a anchos de banda muy altos. Tiene una velocidad de transmisión de datos superior a los 155 Mbps. La tecnología ATM es capaz de transferir voz, video y datos a través de redes privadas y públicas. Tiene una arquitectura basada en celdas las cuales tienen una longitud fija de 53 bytes que son distribuidos e 5 bytes para un encabezado ATM y 48 bytes de carga ATM. Las conexiones se realizan a través de circuitos virtuales.

BRI (BASIC RATE INTERFACE): interfaz de acceso básico, es un método de acceso estándar RDSI (Red Digital de Servicios Integrados), provee un grupo multiplexado de canales B y D. Utiliza dos canales de 64 kbps (Canales B), y un canal de 16 kbps (Canal D). Los canales B son utilizados para transmisiones digitales de voz y datos a velocidades relativamente altas, la información es transportada por medio de tramas y el canal D transporta mensajes de señalización tales como establecimiento y corte de la llamada, para el control de llamadas en los canales B. Como la BRI utiliza dos canales B y un canal D, a veces se conoce como 2B+D.

CAMEL (CUSTOMISED APPLICATIONS FOR MOBILE NETWORKS ENHANCED LOGIC): es un protocolo para redes móviles, es una estándar para redes GSM, permite definir servicios sobre estas redes. Su arquitectura está basada en estándares para redes inteligentes.

CHAP: es un protocolo de autenticación. Se utiliza al iniciar un enlace y verifica de forma periódica la identidad del cliente remoto usando un intercambio de información de tres etapas. CHAP se realiza al establecer el enlace inicial y se repite durante el tiempo que dure el enlace. La verificación se basa en un secreto compartido (como una contraseña). Después de completar la fase de establecimiento del enlace, el agente autenticador envía un mensaje de comprobación al usuario, este responde con un valor calculado mediante la función hash (método para generar claves), esta respuesta se basa en la contraseña y el mensaje de comprobación. El autenticador verifica la

respuesta con el resultado de su propio cálculo de la función hash y si el valor coincide se produce la autenticación de lo contrario la conexión se terminará. A intervalos periódicos el autenticador manda de nuevo un mensaje de comprobación, repitiendo el proceso.

CDR (CALL DETAIL RECORD): es un registro generado por un intercambio de teléfono que contiene todos los detalles de las llamadas que pasan a través de él.

CORBA (COMMON OBJECT REQUEST BROKER ARCHITECTURE): es un estándar definido por el OMG (Object Management Group) que permite escribir componentes de software en múltiples lenguajes y ejecutarlos en múltiples computadoras para trabajar juntos. Es una arquitectura estándar para sistemas de objetos distribuidos. Define las API's, el protocolo de comunicaciones y los mecanismos necesarios para permitir la interoperabilidad entre diferentes aplicaciones escritas en diferentes lenguajes y ejecutadas en diferentes plataformas.

CPE (CUSTOMER PREMISES EQUIPMENT): equipo terminal del abonado. Son usados tanto en interiores como en exteriores para originar, encaminar o terminar una comunicación. Estos dispositivos se encuentran localizados del lado del cliente y se encuentran conectados con el canal de comunicaciones del proveedor.

DSL (DIGITAL SUBSCRIBER LINE): línea de Abonado Digital. Es una tecnología de banda ancha que utiliza líneas telefónicas de par trenzado para transportar datos de alto ancho de banda para dar servicio a los suscriptores. El servicio DSL se considera de banda ancha.

DTD (DOCUMENT TYPE DEFINITION): definición de tipo de documento. Es la descripción de la estructura y sintaxis de un documento XML. Describe el formato de los datos para usar un formato común y mantener la consistencia entre todos los documentos que utilicen la misma DTD.

FRAME RELAY: es una técnica de comunicación mediante retransmisión de tramas. La velocidad de transmisión de datos disponible es de por lo menos 4 Mbps, y algunos proveedores ofrecen velocidades mayores. Funciona al nivel de la capa de enlace de datos. No realiza ningún control de flujo ni errores. Las conexiones Frame Relay utilizan circuitos virtuales. Ofrece una conectividad permanente, de ancho de banda mediano que envía tanto tráfico de voz como de datos.

IAD (DISPOSITIVO O EQUIPO DE ACCESO INTEGRADO): es un dispositivo instalado en las premisas del cliente para proveer el acceso a redes de banda ancha e Internet. Agrega múltiples canales de información incluyendo voz y datos a través de un único enlace de acceso compartido hacia el proveedor del servicio.

IDL (INTERFACE DEFINITION LANGUAGE): es un lenguaje de definición de interfaces que se utiliza en sistemas de programación distribuidos (CORBA). Ofrece la sintaxis necesaria para definir los objetos o métodos que se quieren invocar de forma remota. Está diseñado para permitir que los objetos implementados en lenguajes diferentes se invoquen unos a otros, gracias a que define una interfaz neutral para que exista una comunicación entre componentes de software. El IDL es multiplataforma.

IMAP (INTERNET MESSAGE ACCESS PROTOCOL): es un protocolo de red de acceso a mensajes electrónicos almacenados en un servidor. Por medio de este protocolo se puede tener acceso al correo electrónico desde cualquier equipo que tenga una conexión a internet.

LDAP (LIGHTWEIGHT DIRECTORY ACCESS PROTOCOL): es un protocolo de nivel de aplicación que permite el acceso a un servicio de directorio ordenado que contiene toda la información de un entorno de red, es considerado una base de datos en la que se pueden realizar consultas. Usualmente almacena información de login y es utilizado para autenticarse guardando información por ejemplo de datos de contacto del usuario, ubicación de recursos de la red etc.

MGCP (MEDIA GATEWAY CONTROL PROTOCOL): es un protocolo de control de dispositivos, donde un Gateway esclavo (media Gateway), es controlado por un maestro MGC (Media Gateway controller). Es un protocolo basado en redes de telefonía IP. Esta definido por la arquitectura cliente/servidor.

MPLS (MULTIPROTOCOL LABEL SWITCHING): es un mecanismo de transporte de datos estándar creado por la IETF, opera entre las capas de enlace de datos y la capa de red. Fue diseñado para unificar el servicio de transporte de datos para las redes basadas en circuitos y las basadas en paquetes. Puede ser utilizado para transportar voz y paquetes IP. Esta tecnología proporciona circuitos virtuales en las redes IP. MPLS etiqueta los paquetes teniendo en cuenta criterios de prioridad y calidad y la conmutación de paquetes se hace en función de las etiquetas añadidas.

MULTICAST: es un servicio de red en donde el flujo de datos proveniente de una fuente puede ser enviado de forma simultánea a diversos destinatarios. Es utilizado principalmente en aplicaciones multimedia compartidas.

NAS (NETWORK ATTACHED STORAGE): es una tecnología de almacenamiento dedicada a compartir la capacidad de almacenamiento de un computador (servidor), con ordenadores personales o servidores clientes a través de una red. El acceso a los dispositivos NAS se realiza por medio de protocolos de red como TCP/IP. Es muy utilizado en entornos donde se manejan grandes cantidades de datos.

POTS (PLAIN OLD TELEPHONE SERVICE) Y PSTN (PUBLIC SWITCHED TELEPHONE NETWORK): se refieren al servicio telefónico estándar de voz analógico que utiliza hilos de cobre. Está constituido por todos los medios de transmisión y conmutación necesarios para establecer la comunicación entre dos terminales mediante un circuito físico que se establece durante la comunicación y desaparece cuando se ha terminado. Trabaja entre las frecuencias de 300 y 3400 Hz y tiene un ancho de banda de 56 Kbps. El costo depende de la distancia y de la duración de la conexión. La diferencia entre las dos es que POTS se asocia con la red NGN de EMCALI y PSTN con el servicio antiguo de telefonía.

PPP (POINT TO POINT PROTOCOL): protocolo punto a punto. Es el protocolo de preferencia para las conexiones WAN conmutadas. Permite establecer una comunicación a nivel de enlace de datos entre dos computadoras. Generalmente se utiliza para establecer la conexión a Internet con un proveedor de servicios a través de un modem telefónico. Es utilizado en conexiones de banda ancha, facilita la autenticación por medio de claves de acceso y la asignación dinámica de la IP ya que a cada cliente le asigna una dirección y la conserva hasta que termina la conexión por PPP.

PRI (PRIMARY RATE INTERFACE): interfaz de acceso primario, es un método de acceso estándar RDSI (Red Digital de Servicios Integrados), también provee un grupo multiplexado de canales B y D, los canales B utilizados para voz y datos y el canal D utilizado para señalización. En América del Norte y Japón, las PRI ofrecen 23 canales B de 64 Kbps y un canal D de 64 Kbps para un total de velocidad de transmisión de 1,544 Mbps. En Europa y en gran parte del resto del mundo ofrecen 30 canales B y un canal D ofreciendo una velocidad de transmisión de hasta 2,048Mbps. Las PRI's permiten que se realicen conferencias de video y conexiones de datos de banda ancha.

PROTOCOLO SIP (SESSION INITIATION PROTOCOL): es un protocolo que opera en el nivel de aplicación y es empleado para realizar el control de llamadas multimedia y servicios telefónicos avanzados. Se encarga de establecer modificar y terminar sesiones multimedia o llamadas con uno o más participantes. Es un protocolo basado en http, empleado en Internet. Esta basado en una arquitectura cliente/servidor en donde el intercambio de mensajes se realiza en forma de peticiones y respuestas entre una entidad cliente y otra que funciona como servidor. Los usuarios se identifican por medio de direcciones similares a las de correo electrónico y disponen de gran movilidad ya que la red se encarga de localizarlos cuando se establece una comunicación entre ellos.

RISC (REDUCED INSTRUCTION SET COMPUTER): computadora con conjunto de instrucciones reducido. Es un tipo de arquitectura de computadores que promueve conjuntos pequeños y simples de instrucciones que pueden tomar poco tiempo para ejecutarse.

SDH (JERARQUÍA DIGITAL SINCRONA): es una norma para el transporte de datos en telecomunicaciones formulado por la Unión de Telecomunicaciones Internacional (ITU), el transporte de datos se realiza por medio de tramas y cada una va encapsulada en una especie de contenedor en donde se le añaden cabeceras de control que permiten identificar el contenido. La trama de SDH es el STM-1 con una velocidad de 155 Mbps. Utiliza como medio de transmisión la fibra óptica y permite el incremento en la flexibilidad y el ancho de banda para la transmisión de los datos.

SMPP (SHORT MESSAGE PEER TO PEER PROTOCOL): es un protocolo estándar utilizado para enviar SMS. Define el conjunto de operaciones para enviar mensajes. El protocolo se basa en el intercambio petición/respuesta, el intercambio de datos es de forma asíncrona o síncrona.

SMTP (SIMPLE MAIL TRANSFER PROTOCOL): protocolo simple de transferencia de correo. Es un protocolo basado en texto utilizado para intercambiar mensajes de correo electrónico entre computadoras o distintos dispositivos. Se basa en el modelo cliente / servidor.

SOAP (SIMPLE OBJECT ACCESS PROTOCOL): es un protocolo que define como dos objetos en diferentes procesos pueden comunicarse por medio del intercambio de datos XML. Este protocolo es utilizado en servicios Web y funciona sobre cualquier protocolo de internet. No se encuentra asociado a ningún lenguaje ni a ningún protocolo de transporte.

TDM (MULTIPLEXACIÓN POR DIVISIÓN DE TIEMPO): es un tipo de multiplexación digital en la cual dos o mas señales o cadenas de bits se transfieren al parecer simultáneamente como subcanales en un solo canal de comunicación pero físicamente están turnándose en el canal. El dominio del tiempo es dividido en varios periodos de longitudes fijas uno para cada subcanal.

VOLUMEN LOGICO: unidad básica para la base de datos, registro de recuperación y agrupaciones de almacenamiento. Por ejemplo un volumen lógico puede ser un archivo estándar del sistema de archivos. Cada volumen se identifica mediante un identificador de volumen exclusivo.

WSDL (WEB SERVICES DESCRIPTION LANGUAGE): es un formato XML que se utiliza para describir servicios Web. Describe la interfaz pública a los servicios Web, describe la forma de comunicación, requisitos del protocolo y los formatos de los mensajes necesarios para interactuar con los servicios. Por ejemplo un programa cliente que se conecta a un servicio web puede leer el WSDL para determinar las funciones disponibles en el servidor.

XPON (REDES PASIVAS ÓPTICAS): como su nombre lo indica son redes de nueva generación basadas en fibra óptica. Actualmente se destacan los estándares GPON (Gigabit capable Passive Optical Network) y EPON (Ethernet Passive Optical Network), utilizadas para el despliegue de nuevas

redes de acceso de alto rendimiento, permitiendo la continuidad desde la red troncal hasta el usuario. Pueden cubrir distancias hasta de 20 kilómetros, minimiza la cantidad de fibra a instalar pues utiliza la topología point to multipoint, ofrecen mayor ancho de banda y permiten la difusión tanto de video digital como analógico.

WDM (WAVELENGTH DIVISION MULTIPLEXING): es una tecnología que multiplexa varias señales sobre una sola fibra óptica mediante portadoras ópticas de diferente longitud de onda usando luz procedente de un laser o un LED.

RESUMEN

La plataforma multiservicios instalada en las Empresas Municipales de Cali (EMCALI), permite el despliegue de una gran cantidad de servicios tanto para las redes fijas como móviles, gracias a que cuenta con la arquitectura OSA/Parlay, en donde los servicios pueden ser desarrollados por terceras personas preocupándose solamente por la aplicación pues no es necesario que tengan conocimiento de la infraestructura de las redes de EMCALI. Esto trae como ventaja el aumento del número de desarrolladores y por tanto de la diversidad y cantidad de servicios que EMCALI podrá ofrecer a sus usuarios.

EMCALI no ha utilizado la tecnología OSA/Parlay para el despliegue de los servicios que actualmente presta; por lo cual, el objetivo de este proyecto es entender el concepto de OSA/Parlay, el funcionamiento y la estructura de la plataforma multiservicios instalada en EMCALI con el fin de iniciar el camino de la verdadera explotación de las capacidades de la plataforma de EMCALI.

Este proyecto inicia realizando un recorrido por la actual red NGN instalada en EMCALI, como está conformada, la arquitectura y las diferentes redes con las que cuenta para prestar servicios. También se hace una descripción detallada de la plataforma multiservicios, su arquitectura y definición de cada uno de los servicios y dispositivos con los que cuenta. Luego se analizan las diferentes tecnologías existentes para el desarrollo de servicios, teniendo en cuenta algunos puntos para su evaluación, los que permitirán establecer sus diferencias, ventajas, desventajas, su funcionamiento, en donde aplican mejor y hasta qué punto han sido desarrolladas.

Finalmente, se realiza una descripción detallada de las herramientas a utilizar para el desarrollo de los servicios en la plataforma multiservicios de EMCALI, siendo una guía para el futuro desarrollo de servicios sobre la plataforma. También se hace una captura de requerimientos acerca de algunos de los servicios que EMCALI puede ofrecer a sus usuarios utilizando los recursos con los que cuenta la plataforma multiservicios instalada actualmente.

INTRODUCCIÓN

Actualmente el proceso evolutivo de las telecomunicaciones con el avance tecnológico, los nuevos dispositivos de comunicaciones, el crecimiento de las tecnologías de la información, la aparición de técnicas, protocolos, la masificación de Internet y los servicios que se pueden adquirir por medio de él, han creado la necesidad en los proveedores de las redes de comunicación de buscar nuevas formas de brindar mejores servicios a sus usuarios.

Ante las nuevas exigencias del mercado de las telecomunicaciones los proveedores de las redes y de los servicios de telefonía e internet han tenido que evolucionar modificando la infraestructura de sus redes, para lograr ofrecer a sus usuarios nuevos servicios que incluyan el envío de voz, datos, video, al mismo tiempo. Esto ha llevado a que las redes sean construidas para integrar todo tipo de servicios en una única infraestructura de red bajo el protocolo de internet IP, que son las redes que actualmente conocemos como NGN (Redes de Nueva Generación).

Las empresas Municipales de Cali (EMCALI), han entrado dentro de este proceso evolutivo y han modificado sus redes para poder prestar a los usuarios una gran cantidad de servicios incluyendo los ya existentes que estén acordes a las nuevas exigencias tecnológicas. EMCALI ha adquirido nuevos dispositivos para la implementación de su red NGN, aquí también se encuentra incluida una nueva plataforma de servicios que permite la implementación de una amplia variedad de nuevos servicios como accesos de banda ancha, servicios de conexión a internet, servicios de voz a través de líneas POTS, voz sobre IP, telefonía sobre IP y demás servicios como: servicios de red inteligente, correo de voz, mensajería unificada, servicios prepago y televisión sobre IP.

Esta nueva plataforma multiservicios trabaja con la arquitectura OSA/Parlay que permite el desarrollo de nuevos servicios utilizando aplicaciones de otros proveedores, que acceden a los recursos de la red a través de un conjunto de interfaces estandarizadas y abiertas. EMCALI pretende desplegar una amplia variedad de servicios sobre su red multiservicios y en el presente proyecto se definirán todas las herramientas necesarias para que el desarrollador pueda desplegar nuevos servicios sobre esta plataforma tomando como base la documentación incluida en este proyecto.

Toda la información obtenida durante el desarrollo de este proyecto será la base, para iniciar un segundo proyecto donde se llevará a cabo las etapas de análisis y diseño de un modelo general para la creación de servicios para luego terminar con la implementación de un prototipo funcional de un servicio sobre la red multiservicios de EMCALI.

1. DESCRIPCIÓN GENERAL DEL PROYECTO

La definición de un ambiente de desarrollo para la implementación de nuevos servicios sobre la plataforma multiservicios de EMCALI es un proyecto que pretende definir las herramientas y obtener la documentación necesaria para el desarrollo de nuevos servicios. Durante el desarrollo de este capítulo se pretende brindar una visión global del enfoque del proyecto y hacia donde va dirigido.

1.1. MARCO TEÓRICO

El proceso evolutivo del sector de las telecomunicaciones ha provocado cambios en el modelo de negocio de muchos operadores y ha modificado de manera radical el modelo de provisión de servicios. Se ha pasado de un modelo vertical, en el cual la red y los servicios aparecen estrechamente ligados, a un modelo horizontal en el que se propone una independencia absoluta entre ambos y una única solución de red común a todos ellos¹.

El proceso de evolución en las redes tradicionales hacia las redes de nueva generación se ha dado fundamentalmente motivado por una serie de factores como son:

- La necesidad de reducir los costos respecto a los modelos tradicionales.
- La necesidad de compartir infraestructuras entre distintas unidades de negocio
- La preponderancia cada vez mayor del modelo Internet
- La necesidad de establecer la convergencia y compatibilidad entre las distintas redes.
- La necesidad de acelerar el proceso de creación y puesta en funcionamiento de las aplicaciones y servicios.
- La necesidad de simplificar y unificar la gestión, la operación y el mantenimiento de los servicios.

Pretender agrupar en una única infraestructura de red las distintas alternativas, existentes o por venir, implica que dicha infraestructura debe responder a unos criterios de diseño estrictos que aseguren su funcionamiento con los niveles de calidad, capacidad, fiabilidad y disponibilidad requeridos por los servicios que soportará dicha red. Además se debe tener en cuenta que los niveles actuales

¹. I+D. Las telecomunicaciones y la movilidad en la sociedad de la información [en línea]. Madrid: División de relaciones corporativas y comunicación de telefónica I+D, 2005. [Consultado en enero 30 de 2008]. Disponible en internet: http://www.telefónica.es/sociedaddelainformacion/html/publicaciones_movilidad.shtml

de calidad, fiabilidad y disponibilidad de determinados servicios, como es el caso de los servicios de voz, son muy altos. La consecuencia de ello es que los clientes han desarrollado una percepción subjetiva de la calidad muy elevada, a la cual se han habituado a lo largo de años de uso, lo que supondrá un importante reto para cualquier solución basada en la NGN.

Los servicios tradicionales, verán disminuir de forma paulatina su importancia a favor de nuevos servicios, muchos de ellos aún desconocidos y, por tanto, de difícil caracterización en el momento de diseñar una red.

La actual red telefónica está evolucionando para adaptarse a los servicios multimedia, constituyendo la base de NGN. Gran parte del desarrollo y provisión de los servicios finales partirá de los Operadores Públicos de Red, soportados por servicios básicos desarrollados sobre interfaces abiertas.

Por otro lado, existe una clara separación entre clientes empresariales y residenciales, ya que sus objetivos y motivaciones son distintos. Mientras que para el grupo de clientes empresariales el principal atractivo de las NGN puede ser los servicios tradicionales (como los servicios de voz, las redes privadas virtuales, etc.) a costos moderados, para el sector residencial, por el contrario, el principal atractivo será mejorar los actuales servicios, manteniendo una estructura de costos bajos, y ampliar la oferta de servicios de entretenimiento.

1.1.1. Evolución de la red clásica hacia las NGN. En las redes clásicas, los servicios estaban estrechamente ligados a la infraestructura de red, de hecho, se consideraban partes indivisibles. Los servicios se integraban de forma vertical. Debido a ello, el desarrollo de la infraestructura de red se adaptó muy bien a los servicios para los que fue diseñada, pero tenía un alto grado de ineficiencia y complejidad que la hacía poco flexible al desarrollo y despliegue de nuevos servicios².

En estas redes, el equipamiento es complejo, de elevado coste y de difícil y costosa explotación. La calidad de servicio se resuelve mediante la asignación y reserva de recursos específicos de red. No soporta de forma nativa las técnicas de distribución basadas en la tecnología multicast, lo cual redundará en un incremento de la complejidad y coste del despliegue de servicios masivos de distribución de contenidos.

La libre competencia en el sector de las telecomunicaciones, motivó que se intentara ampliar el abanico de servicios que cada operador podía ofrecer a sus clientes sobre las infraestructuras existentes en cada caso. De esta forma, las redes se vieron en la necesidad de dar soporte a servicios para los que inicialmente no habían sido diseñadas, apareciendo los primeros síntomas de un problema de fondo: la incapacidad de las redes existentes para dar soporte de forma óptima a toda esta serie de nuevos servicios.

² *Ibíd.*, p. 200

Comenzó de esta forma la búsqueda de soluciones mejor adaptadas al nuevo escenario. En paralelo, se producía una evolución tecnológica en el mundo de las redes de datos, motivada, en gran medida, por una creciente necesidad de comunicaciones en entornos empresariales. Las primeras soluciones se desarrollaron en torno al estándar de comunicaciones ATM (Asynchronous Transfer Mode), aunque fue rápidamente absorbido, por las soluciones nativas IP/Ethernet, una vez que éstas alcanzaron los niveles de velocidad y funcionalidad que habían hecho atractivas las soluciones basadas en ATM.

Finalmente aparece un factor que le dio el impulso definitivo al cambio: El acelerado desarrollo de internet a nivel mundial. Conforme Internet se ampliaba y su uso se normalizaba en gran parte de los entornos tanto empresariales como residenciales, aparecieron corrientes de opinión que apostaban por una solución común basada en las redes IP, que como ya se ha dicho es conocida como All-IP. Sin embargo, las soluciones IP tradicionales presentaban carencias importantes que las hacían poco adecuadas: estaban aún basadas en equipos con serias limitaciones en su capacidad, no existía una solución adecuada de calidad de servicio y los aspectos de seguridad estaban deficientemente tratados.

En este contexto es donde aparece y se desarrolla el concepto NGN, planteándose como la solución que permitirá llevar a cabo las propuestas del modelo All-IP (Todo sobre IP) de forma adecuada. Se presenta, por tanto, como una solución para la convergencia de las redes con interfaces de alta velocidad, con seguridad y calidad garantizadas y que facilita el despliegue de los servicios, tanto actuales como futuros. El objetivo fundamental para los operadores será optimizar las inversiones y asegurar unos rápidos retornos de las inversiones.

1.1.2. Arquitectura de las NGN. Para la ETSI, NGN es un concepto para la definición y despliegue de redes, con una separación formal entre diferentes capas y planos con interfaces abiertos, que ofrece a los proveedores de servicios una plataforma sobre la que sea posible evolucionar paso a paso para crear, desplegar y gestionar servicios innovadores.

NGN es una red funcional multiservicios, basada en tecnología IP, producto de la evolución de las actuales redes IP, con la posibilidad de ofrecer servicios diferenciados y acordes a la calidad de servicio demandada por las aplicaciones de cliente.

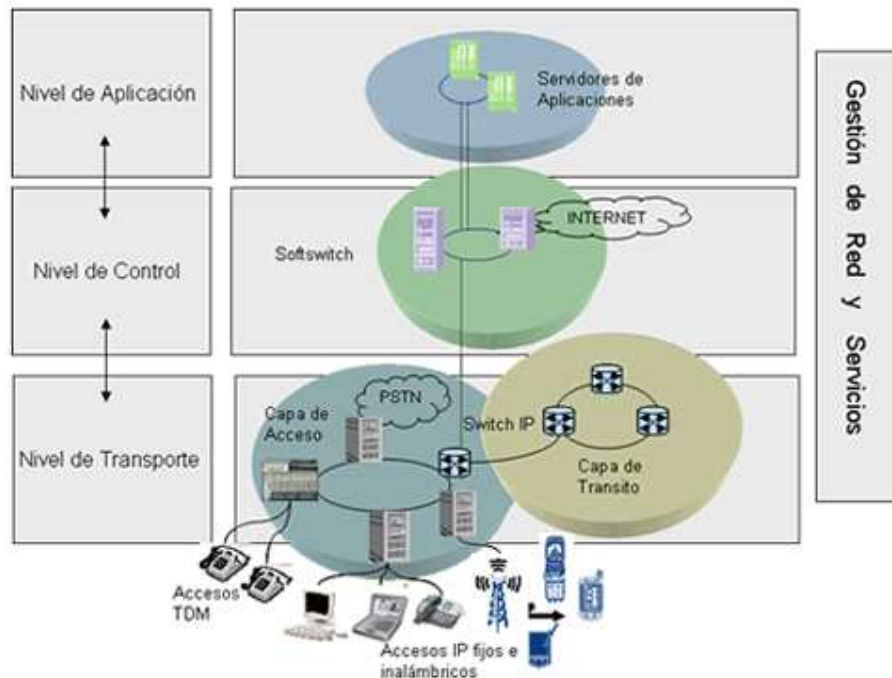
La NGN tiene las siguientes características³:

- La convergencia de los servicios de voz (suministrados en red fija y móvil), vídeo y datos se hace sobre la misma infraestructura de red. La infraestructura de transporte y comunicación debe ser de datos.

³ *Ibíd.*, p. 205.

- La red de conmutación de paquetes (datagramas) debe ser IPv4/IPv6. Soporte de MPLS (MultiProtocol Label Switch) para servicios de ingeniería de tráfico (TE), redes privadas (VPN), etc.
- Soporte de políticas de Calidad de Servicio (QoS). Para el caso de los servicios de voz, el nivel de calidad debe ser al menos como la existente en la red clásica
- Soporte nativo de Multicast
- Alta escalabilidad, disponibilidad, fiabilidad y seguridad.

Figura 1. Arquitectura NGN



Fuente: Migración a NGN [en línea]. Bogotá D.C: Centro de Investigación de las Telecomunicaciones CINTEL, 2003. [Consultado 07 de febrero de 2008]. Disponible en Internet: <http://www.cintel.org.co:8080/cintel/opencms/cintel/inicio/lineas/linea/migracion.html>

Los elementos indispensables con que debe contar toda implementación NGN son:

- Los sistemas de transmisión deben ser de última generación y basados en tecnologías ópticas WDM (Wavelength Division Multiplexing)
- Los elementos de conmutación son de tipo Gigabit Switch-Router (GSR) o Terabit Switch-Router (TSR), conformando una red IPv4/IPv6 con soporte de MPLS

- Se dispone de una política de calidad de servicio (QoS) efectiva y totalmente operativa
- Se dispone de una política de seguridad tanto a nivel de red como de cliente
- Estructura de red escalable que permita evoluciones futuras de forma gradual
- Se incorporan técnicas eficaces, en el entorno de equipo y sistema, que aseguren una fiabilidad y disponibilidad adecuadas

1.1.3. Los servicios en las redes NGN. Los servicios para redes de siguiente generación deben proporcionar entornos seguros para su creación, despliegue y modificación, de igual forma, para su desarrollo se deben considerar aspectos como la interoperabilidad, la convergencia, la movilidad, la calidad del servicio, la seguridad y se enfatiza en la necesidad de adaptarse a los usos de los usuarios.

Para el desarrollo de servicios en redes de siguiente generación se hace necesario tener en cuenta los siguientes términos:

- Personalización: Cuando se desarrolla un servicio es importante saber para quien se ofrece de forma que se puedan ofrecer soluciones particulares para las necesidades de los diferentes usuarios
- Independencia del terminal y de la red de acceso: Es importante que los servicios que se ofrezcan a los usuarios puedan ser accedidos desde cualquier terminal que se utilice teniendo en cuenta que en un futuro las redes de telecomunicaciones estarán compuestas por la interconexión de múltiples tecnologías
- Cortos espacios de desarrollo: Es necesario lanzar al mercado servicios que puedan ser modificados para adaptarlos a las preferencias de los usuarios incorporándoles nuevas funcionalidades en cortos periodos de tiempo
- Universalidad: Este término indica que un servicio desarrollado pueda ser aplicado a todo tipo de entornos con independencia de la red
- Inteligencia: Es necesario que los servicios se adapten a las preferencias de los usuarios
- Ubicuidad: Los servicios pueden ser accedidos desde cualquier tipo de dispositivo por lo cual las redes que permiten la interconexión de estos dispositivos hace que los entornos de desarrollo de los servicios que quieran adaptarse a ellos requieran un diseño cuidadoso

Los entornos tradicionales de desarrollo de servicios para redes no se adaptan bien a los términos mencionados anteriormente, por tanto se hace necesario diseñar redes con interfaces abiertos y estándares que permitan el rápido y flexible desarrollo de servicios.

1.2. ANTECEDENTES

Las Empresas Municipales de Cali (EMCALI), en un proceso de reestructuración en las telecomunicaciones, y colocándose a la par de las empresas de telefonía no solo en Colombia sino a nivel mundial, implemento su propia plataforma de red multiservicios la cual integró los proyectos de sustitución de 95.230 líneas entre analógicas y digitales por la tecnología IP y la ampliación de líneas de telefonía básica para ofertar 44,800 líneas nuevas, las cuales permitirán atender la demanda insatisfecha actual y la que aparecerá con el desarrollo de nuevas viviendas de acuerdo al crecimiento que se esta presentando en el área de influencia de EMCALI⁴.

La empresa estatal china ZTE Corporation, ganó la licitación para el montaje de la red multiservicios de telecomunicaciones de EMCALI, esta empresa es también la encargada de implementar los servicios que presta actualmente EMCALI, pero para ello no se está utilizando la tecnología OSA/Parlay, y EMCALI no cuenta con la documentación necesaria para que otros proveedores de servicios puedan implementar nuevos servicios sobre la plataforma.

1.2.1. Servicios. La plataforma multiservicios de EMCALI, soporta la arquitectura OSA/Parlay para la construcción de servicios, por lo cual estos servicios o aplicaciones pueden ser desarrollados por terceras partes y pueden ser independientes de la red. En el año 2007, dos estudiantes de la universidad Autónoma de Occidente iniciaron un proyecto para el diseño de un servicio telemático para la red NGN de EMCALI, usando la tecnología OSA/Parlay, el cual consiste en una línea de opinión en donde los usuarios podrán opinar acerca de diferentes temas de interés. Este servicio también tenía como funciones principales:

- Escuchar y grabar opiniones acerca de los temas planteado
- Los usuarios podrán votar por las opiniones ya grabadas
- Generar estadísticas respecto a la duración y horario de las llamadas, ubicación geográfica de los usuarios y el comportamiento de estos frente a las opiniones que ya han sido registradas.

⁴ Página web de EMCALI [en línea]. Santiago de Cali: EMCALI. 2007. [Consultado 05 de febrero de 2008]. Disponible en Internet: <http://www.emcali.com.co/vsmPC/bin/smRenderFS.php?PHPSESSID=d6d190dc0a09addbf7967e1cc3ced2a&cerror=&xnode=5>

La implementación de este servicio sobre la plataforma multiservicios de EMCALI no fue posible, ya que no se tenía la documentación necesaria sobre esta, por lo cual, el servicio fue probado en un simulador de la plataforma de Ericsson NGR (Network Resource Gateway), que también soporta la arquitectura OSA/Parlay y que permite simular y probar servicios, pero la plataforma de Ericsson presentaba algunos inconvenientes al momento de la simulación ya que la versión utilizada no soportaba la captura de audio y no fue posible la grabación de las opiniones de los usuarios.

1.3. OBJETIVO GENERAL

Definir el ambiente de desarrollo para la implementación de nuevos servicios telemáticos sobre la red multiservicios de EMCALI.

1.4. OBJETIVOS ESPECÍFICOS

- Realizar la captura de los requerimientos para la construcción de nuevos servicios telemáticos en la red multiservicios de EMCALI
- Analizar y documentar la arquitectura NGN actual de EMCALI
- Analizar las diferentes tecnologías para la creación de servicios de nueva generación, y elegir la más adecuada para la arquitectura NGN de EMCALI.

1.5. JUSTIFICACIÓN

Los operadores de línea fija afrontan una intensa competencia por parte de los operadores de telecomunicaciones inalámbricas, los proveedores de redes de televisión por cable y los grandes proveedores de contenido Internet que cuentan con marcas muy conocidas y mucho dinero para invertir. La búsqueda de nuevas fuentes de ingresos mediante los cada vez más populares paquetes agregados de "tres en uno" o "cuatro en uno" con IPTV, llamadas locales y acceso ultrarrápido a Internet en banda ancha, ha acelerado la tendencia a acercar las redes de fibra a los hogares y oficinas. Por otra parte, los operadores móviles se encuentran mejorando sus redes para ofrecer conectividad sin interfaces en favor de aplicaciones intensivas en anchura de banda como la televisión móvil⁵.

⁵ COHEN, Tracy; HERNEDEZ, Janet. El camino hacia las redes de próxima generación-pioneros [en línea]. Estados Unidos: Unión Internacional de Telecomunicaciones, 2008. [Consultado 05 de febrero de 2008]. Disponible en Internet: <http://www.itu.int/itunews/manager/display.asp?lang=es&year=2007&issue=02&ipage=next-generation2&ext=html#top>

Junto con el acceso a Internet a velocidades de transmisión más elevadas que las que hace posible la tecnología ADSL, las NGN facilitan la prestación de una gama muy completa de servicios públicos. Por este motivo, las instancias decisorias y los reguladores no se preguntan ya si deberían fomentar esta incesante evolución, sino cómo acelerarla.

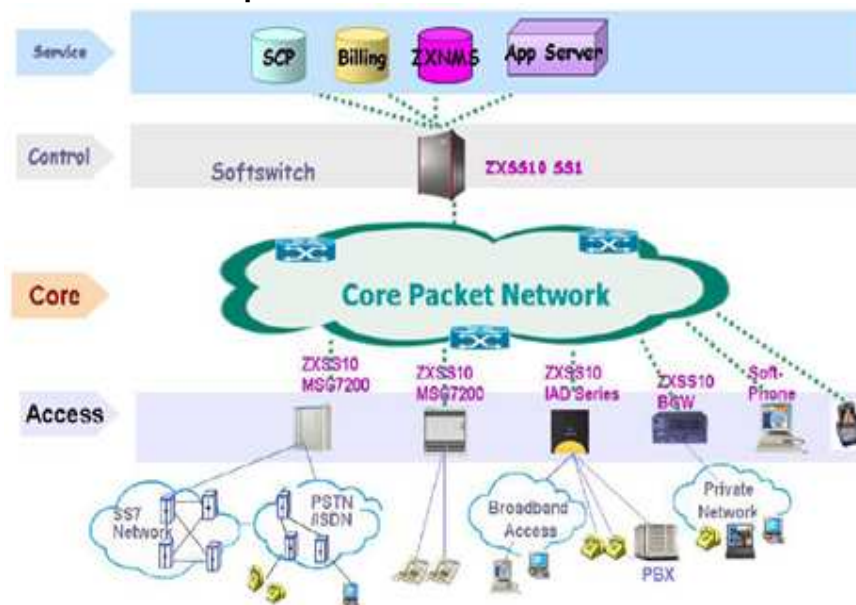
EMCALI, cuenta con la infraestructura física necesaria para prestar servicios de voz, datos y video, con un alto grado de calidad, además cuenta con una plataforma de servicios lo que le permite posicionarse como una de las primeras empresas en Latinoamérica con este tipo de infraestructura pero que no está siendo explotada con respecto a la tecnología OSA/Parlay, siendo necesario entonces obtener la documentación necesaria sobre las características de esta plataforma, su funcionamiento, como esta implementada, como se pueden montar los servicios, que servicios se pueden ofrecer a través de ella, que elementos la conforman, como se puede acceder a ella, lo que permitirá en un futuro el desarrollo de una cantidad de servicios acordes a las necesidades prácticamente de cada usuario.

2. PLATAFORMA MULTISERVICIOS DE EMCALI

Las Empresas Municipales de Cali (EMCALI), han migrado hacia las redes de nueva generación, adaptando su infraestructura y adquiriendo nuevos equipos en favor de ofrecer nuevos y mejores servicios con mayor calidad, de acuerdo, a las necesidades de los habitantes de la ciudad de Cali.

La red multiservicios de EMCALI, físicamente esta dividida en cuatro capas, donde se encuentran una serie de dispositivos los cuales cumplen funciones específicas dentro de cada una, siendo importante entonces, conocer de forma descriptiva las especificaciones técnicas y funcionales de estos dispositivos para obtener un entendimiento completo acerca del funcionamiento de esta nueva plataforma multiservicios.

Figura 2. Modelo en capas de la red multiservicios



Fuente: EMCALI EICE ESP. Descripción de la Red Existente. Santiago de Cali, 2008. 1 archivo de computador.

2.1. CAPA DE ACCESO

Es la capa inferior dentro del modelo de capas de la plataforma multiservicios, y tal como su nombre lo indica es la que permite a todos los usuarios (línea POTS, PSTN, banda ancha etc.), acceder a la red, es decir, es el punto en el que cada usuario se conecta a la red y puede obtener los servicios que son ofrecidos a través de esta.

La red de acceso está conformada por seis anillos dentro de los cuales se encuentra switches de nivel 2/3, con funcionalidades MPLS, conectados por medio de dos hilos de fibra óptica monomodo. Esta red de acceso está conformada por troncales de 10GbE a través del anillo de fibra óptica y se conecta a la red de paquetes por troncales de 10GbE igualmente.

Los switches utilizados para esta red de acceso son los T64G, los cuales cuentan con características como: son switches de alta capacidad, pueden realizar procesos de forma paralela, son modulares, soportan una gran variedad de interfaces 10 GE, FE entre otras, tienen funcionalidades MPLS, NAT, multicast, control de ancho de banda, trabajan en nivel 2/3 y 4 por lo cual son utilizados para niveles de acceso y core en las redes, son carrier-class, ofrecen escalabilidad, soportan IPv4 e IPv6 .

Estos switches conforman 16 nodos de acceso los cuales son el punto de convergencia de todo el tráfico de datos y voz para luego ser transportado hacia la red de paquetes.

A esta red de acceso se conectan las UAM's (Unidades de Acceso Multiservicios), conocidas también como Gateway de acceso, las cuales conectan a los clientes a través de la red de cobre, y de conexión con las centrales TDM a través de la red de acceso SDH con la red de paquetes por medio de la red de acceso enviando el tráfico en paquetes.

Las UAM's se encuentran ubicadas en el nodo de conmutación o en las premisas del cliente. También proveen los puertos para las líneas POTS, PBX, servicios de banda ancha, interfaces XPON (redes ópticas pasivas), puertos BRI y PRIS de RDSI, interfaces E1 y Ethernet hacia el lado del cliente e interfaces de paquetes (eléctricas u ópticas) hacia el lado de la red de acceso.(Tomado de documento red existente).

La UAM puede ser equipada con módulos de diferentes tipos de servicios como módulos para puertos POTS, módulos para banda ancha (xDSL) y módulos para puertos BRI o PRI.

Las UAM's tienen las siguientes funciones básicas⁶:

- Convierte el tráfico TDM proveniente de los clientes y conectado en las interfaces TDM (POTS, E1'S, BRIS, PRIS), vía par de cobre, fibra u otro medio, en tráfico de paquetes y lo entrega al switch de la red de acceso y la función inversa de convertir el tráfico de paquetes, proveniente del switch, en tráfico TDM y entregarlo a los clientes
- Convierte el tráfico ATM de los puertos XDSL, proveniente del modem del cliente, en tráfico de paquetes para ser entregado al switch de la red de

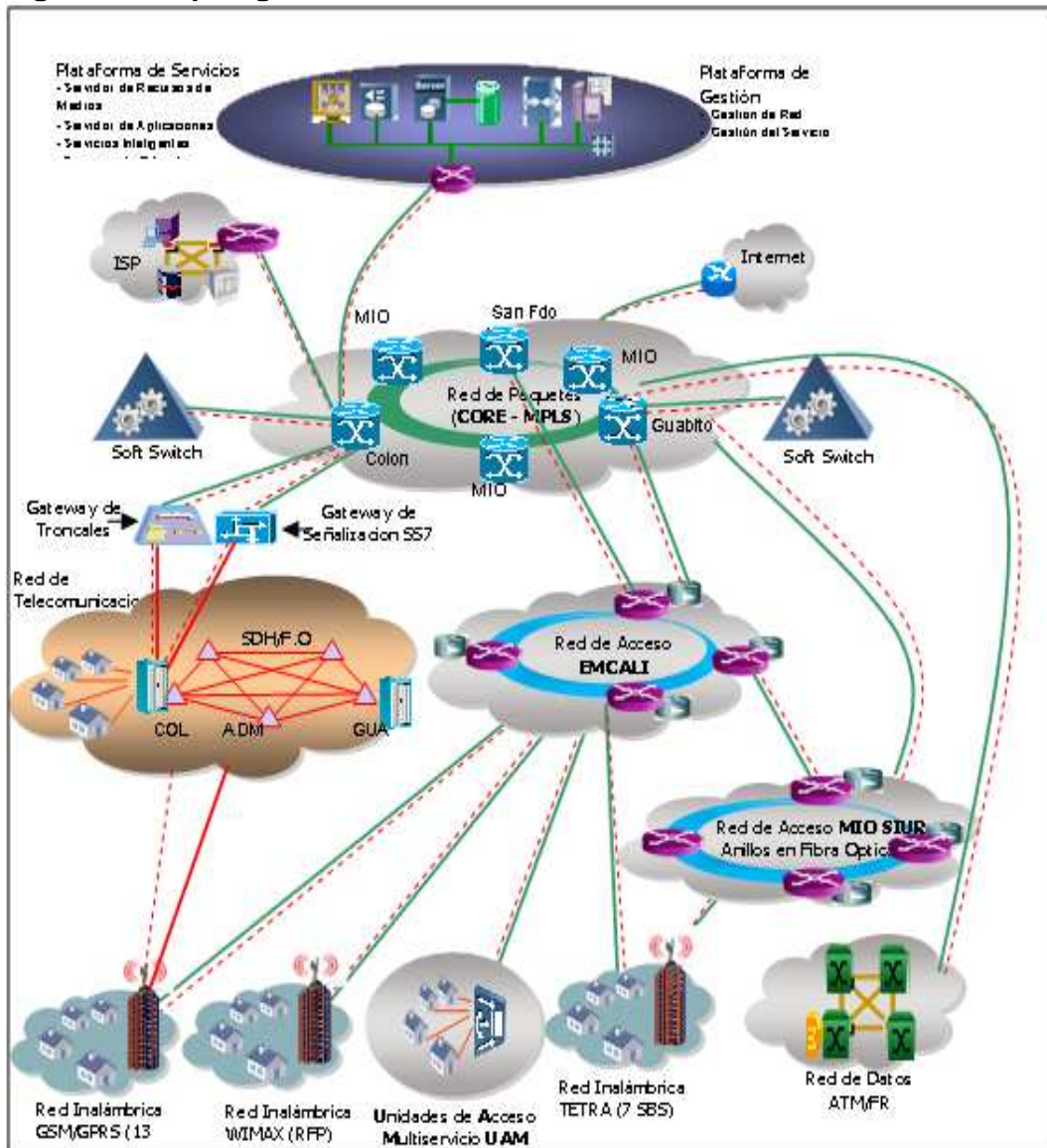
⁶ EMCALI EICE ESP. Descripción de la Red Existente. Santiago de Cali, 2008. 1 archivo de computador.

acceso y la función inversa de convertir el tráfico de paquetes, proveniente del switch de la red de acceso, en tráfico ATM hacia el modem del cliente

- Convierte el tráfico XPON proveniente de los clientes en tráfico de paquetes y lo entrega al switch de la red de acceso y la función inversa de convertir el tráfico de paquetes, proveniente del switch en tráfico XPON y entregarlo a los clientes
- Las UAM's adaptan las interfaces Ethernet del lado de los clientes a las interfaces Ethernet del lado del switch de la red de acceso.

En la red Multiservicios de EMCALI se cuenta con 230.000 líneas POTS y 132.000 puertos XDSL, los cuales están conectados a las UAM's quienes tienen la funcionalidad de Gateway de mediación enviando el tráfico paquetizado para ser transportado desde al red de acceso hacia la red de paquetes.

Figura 3. Topología de la red multiservicios



Fuente: EMCALI EICE ESP. Descripción de la Red Existente. Santiago de Cali, 2008. 1 archivo de computador.

2.2. CAPA DE PAQUETES (CORE)

La capa de núcleo, principal o core es la encargada de desviar el tráfico lo más rápido posible hacia los servicios que han sido solicitados. Es decir, es la capa que conmuta los paquetes, teniendo en cuenta factores como la velocidad, la latencia y los protocolos que permitan converger rápidamente hacia el servicio invocado.

Esta capa tiene conexión con la red acceso por medio de cinco nodos con switches T160G MPLS, ubicados en Colón, San Fernando, Guabito, Versalles y Parcelaciones, conectados entre sí por enlaces de fibra óptica, por medio de troncales de 10GbE transportando de manera eficiente los paquetes.

El switch T160G tiene funciones de switch/router de alta capacidad, trabaja en capa 2/3 y 4, es modular lo que permite que pueda realizar múltiples procesos, soporta una variedad de interfaces de 10GE, FE, POS, es carrier-class, realiza diversas funciones con IPv4,Pv6,MPLS,NAT, QoS y multicast.

Dentro de esta capa se encuentran dos componentes importantes como son los gateway troncales y los gateway de señalización.

2.2.1. Gateway troncales (TG). Son los encargados de convertir la información TDM en información de paquetes y viceversa suministrando las interfaces E1 y Ethernet para integrar la red PSTN existente con el core de la red multiservicios. Estos gateway troncales han sido ubicados en Colon y guabito, manejando el tráfico de voz e Internet.

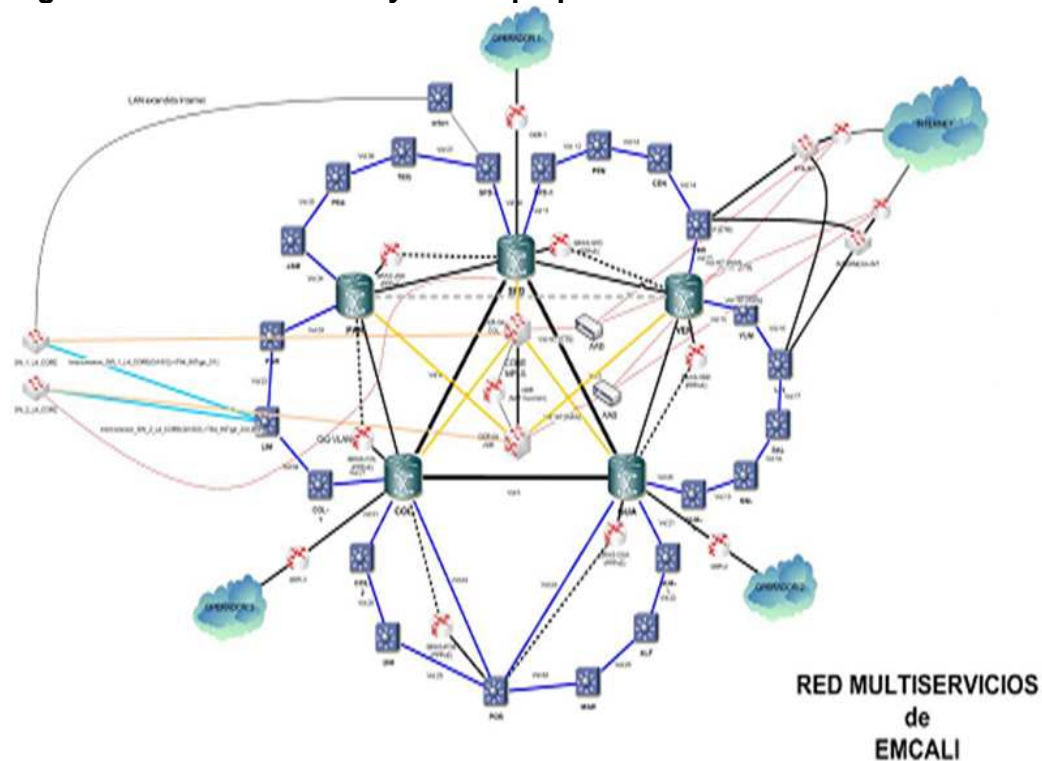
Los gateway utilizados para esta plataforma multiservicios son los ZXSS10 MSG7200 (TG) los cuales están en capacidad de armar y desarmar paquetes de voz para poderlos transmitir a las centrales telefónicas de la red PSTN o a la red de paquetes.

2.2.2. Gateway de señalización (SG). Tienen la función de traducir la señalización CSN (Circuit Switched Network), conocida como SS7 de la red PSTN existente en señalización de paquetes y viceversa para transmitirlos a través de ambas.

Los gateway de señalización implementados en la red NGN son los ZXSS10 MSG7200 (SG), los cuales funcionan al borde de la red SS7 y la red de paquetes en el modo de circuito conmutado, recibe y envía mensajes SS7 de la red de conmutación de circuitos por medio de E1 y adopta los protocolos de adaptación de capas y transporte como SIP o H.323, realizando la interconexión entre las dos redes internas de señalización. Este gateway también tiene la función de proporcionar rutas de transmisión para la voz.

La conexión entre la red de acceso por medio de las UAM's hacia la red de paquetes se ilustra en la siguiente figura:

Figura 4. Red de acceso y red de paquetes de la red multiservicios



Fuente: EMCALI EICE ESP. Descripción de la Red Existente. Santiago de Cali, 2008. 1 archivo de computador.

2.3. CAPA DE CONTROL

Como su nombre lo indica en esta capa se realiza el control de todas las operaciones que se lleven a cabo dentro de la plataforma multiservicios y del control igualmente de todos los dispositivos que hacen parte de esta. Asegura el funcionamiento entre las demás capas obteniendo así los mecanismos necesarios para la provisión de los servicios.

Dentro de esta capa reside el softswitch el cual es el elemento central de la red el cual realiza las mismas funciones de una red de conmutación de circuitos permitiendo conectar abonados, interconectar centrales telefónicas y ofrecer servicios de telefonía. El softswitch implementado en la plataforma multiservicios de EMCALI es el ZXSS10 SS1b desempeñando funciones tales como:

- Control de llamadas
- Control y enrutamiento de llamadas y sesiones
- Adaptación de protocolos para interconectar usuarios de diferentes redes
- Provisión de servicios

- Control de los gateway de troncales y de señalización
- Responsable de la tasación del tráfico de voz, de Internet y de todos los usuarios de la red multiservicios
- Administración del ancho de banda y de los recursos (recursos de troncales, servidores de medios, recursos del sistema de procesamiento etc.) del sistema en general, según la demanda de los diferentes servicios
- Establecimiento de canales con diferentes anchos de banda de acuerdo al estado de la red de datos
- Establecimiento de circuitos virtuales requeridos por un patrón de tráfico predeterminado o para el establecimiento de una nueva llamada
- Permitir ofrecer servicios de telefonía clase 5, carrier-class y servicios multimedia
- Control de tráfico IP proveniente de las líneas POTS (a través del modulo NAS del gateway de troncales), de los puertos XDSL de la red NGN o de la red PSTN
- Control de los servicios de Internet
- Control sobre las UAM's, gateway troncales, gateway de señalización y la plataforma de servicios

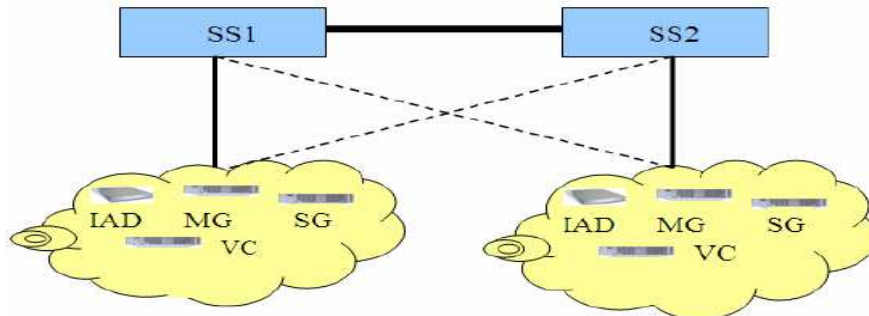
Este softswitch posee redundancia a nivel de hardware y software, al ser un dispositivo modular todas sus tarjetas (tarjetas de control de servicios, de procesamiento de servicios, de interfaz de red etc.), son intercambiables en funcionamiento. Cuando alguna tarjeta tiene un problema, la tarjeta de backup toma su lugar inmediatamente, guarda un registro de las llamadas y los servicios no son interrumpidos.

A nivel de software el diseño también es redundante, ya que este equipo puede ser configurado en modo de solución redundante, para resolver problemas de seguridad y confiabilidad de red.

En el diseño de la plataforma multiservicios se tiene un softswitch en colon y otro en guabito siendo este último el que proporciona el sistema de redundancia para el softswitch de colon como protección contra desastres. Normalmente los dos softswitches controlan los grupos de usuario y los respectivos gateways en modo balanceado de carga, es decir, en el softswitch de colon el 50% es usado para datos de usuario (información de usuarios, información de troncales, etc.) y el otro 50% es usado para datos de backup donde se encuentran almacenados los datos de usuario del softswitch ubicado en guabito el cual opera de igual forma que el de colon teniendo en datos de backup los datos de usuario del softswitch de colon.

Cuando alguno de los dos softswitch encuentra algún problema, los gateways y unidades de red controlados por este, se vuelven a registrar en el softswitch de backup, de esta forma se pueden seguir prestando los servicios normalmente a los usuarios, conservando también, el estado en que se encuentre algún servicio en el momento en que se presentó la falla. De esta forma cualquiera de los dos softswitches de la red multiservicios puede trabajar con el 100% de la carga de la red. Esta distribución de cargas de los softswitch es ilustrada en la siguiente figura.

Figura 5. Distribución de cargas de los softswitch de colon (SS1) y quabito (SS2)



Fuente: ZTE. Propuesta técnica NGN de ZTE para EMCALI. Santiago de Cali, 2005. 1 archivo de computador.

2.4. CAPA DE SERVICIOS

Esta última capa es la que contiene cada uno de los servicios que serán utilizados por los usuarios. Estos servicios pueden ser desarrollados por terceras partes en diferentes plataformas sin importar la infraestructura de la red.

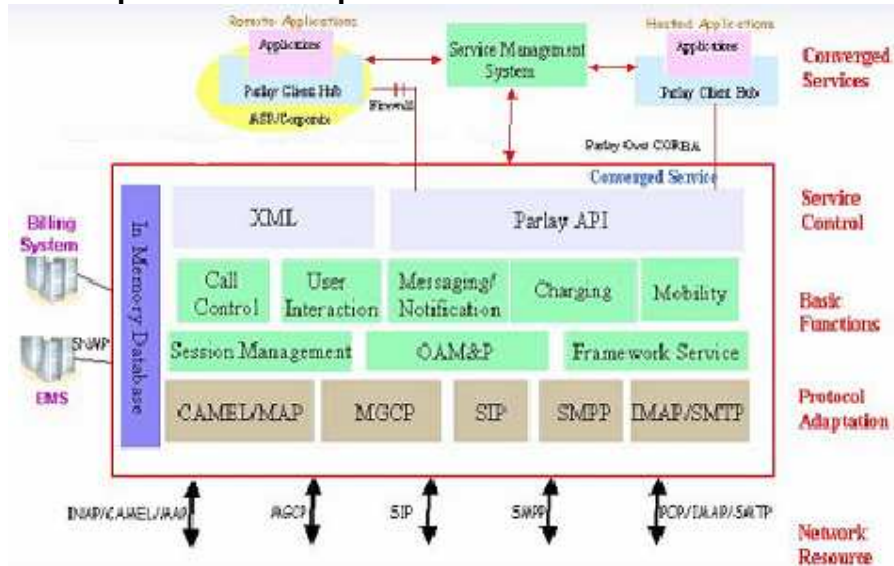
La capa de servicios está conformada por la plataforma de servicios ZXUP10, el cual es un servidor de aplicaciones convergente, modular orientado hacia NGN y adopta la arquitectura OSA/Parlay.

El sistema ZXUP10 tiene como principales características:

- Está diseñada para trabajar tanto con la red existente como con la red NGN
- Provee interfaces estándar y abiertas lo cual permite el desarrollo de servicios por terceras partes
- Soporta múltiples protocolos como INAP, CAMEL.WIP, SIP, MGCP y SMPP
- Las aplicaciones pueden ser ejecutadas desde diferentes servidores de aplicaciones reduciendo posibles fallas en el sistema

- Proporciona encapsulamiento y encapsulamiento de los recursos de la red

Figura 6. Arquitectura de la plataforma de servicios ZXUP10



Fuente: ZTE. Propuesta técnica plataforma de servicios. Santiago de Cali, 2005. 1 archivo de computador.

El ZXUP10 está dividido en cuatro capas cada una con una función específica:

2.4.1. Capa de aplicación. Dentro de esta capa se encuentra el servidor de aplicaciones generando los diferentes servicios por medio del Parlay API. Esta capa realiza el registro y autenticación con el framework de Parlay y obtiene el derecho de usar un determinado módulo SCF (Feature Capability Service) del servidor Parlay de acuerdo a los requerimientos del servicio invocado para de esta forma crear la lógica de la aplicación.

2.4.2. Capa de control de servicios. Esta capa implementa el Parlay API. Es aquí donde se encuentran todos los módulos SCF que pueden ser invocados por las aplicaciones y donde se registran con el framework, de esta forma una aplicación sabe si puede obtener el servicio requerido desde el servidor Parlay. Dentro de estas SCF's encontramos la SCF de control de llamadas, SCF UI (User Interaction), SCF de mensajes, SCF de tasación, SCF de movilidad entre otras.

2.4.3. Capa de adaptación. Esta es la capa de interfaz de protocolos la cual controla la capa de control de servicios y la capa de recursos de la red. Es la encargada de seleccionar el protocolo adecuado de acuerdo al tipo de red y envía la solicitud de API (donde se detallan los requerimientos) hacia una

entidad de recursos de red específica y la respuesta es enviada a la SCF correspondiente.

El servidor Parlay realiza la adaptación de protocolos como CAMEL, SIP, MGCP, IMAP, SMTP y SMPP.

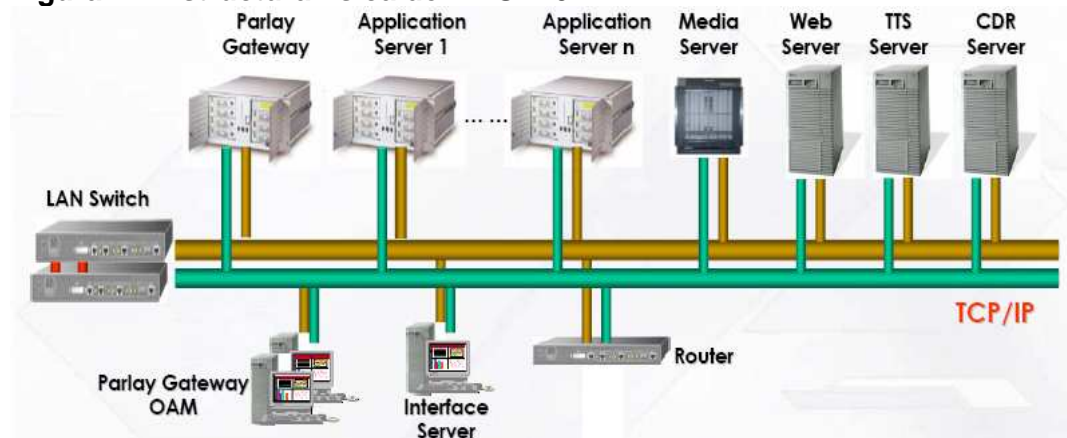
2.4.4. Capa de recursos. Esta compuesta por cada uno de los recursos de la red, conformado por elementos como el softswitch, el servidor de medio, el servidor de correo electrónico, el centro de mensajería corta entre otros.

La capa de aplicación asigna la solicitud mediante la interfaz de Parlay al módulo SCF correspondiente en la capa de control de servicio. Después la SCF utiliza una interfaz para enviar la solicitud a la capa de adaptación de la red para la conversión de los parámetros del protocolo. Finalmente, la solicitud se envía a la capa de recursos de red.

2.5. COMPONENTES DE LA PLATAFORMA ZXUP10

Dentro de la estructura física de la plataforma ZXUP10 se encuentran elementos como el Parlay Gateway, el servidor de medios, el servidor de aplicaciones, el administrador de operaciones, la consola de mantenimiento (OAM), y el gateway de señalización.

Figura 7. Estructura física del ZXUP10



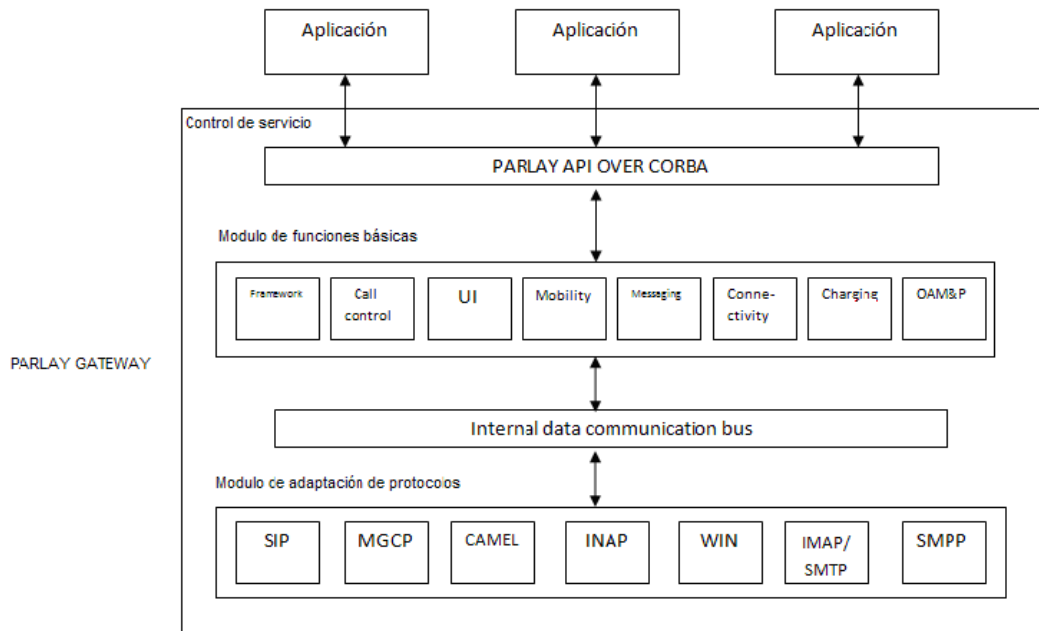
Fuente: ZTE. Plataforma Unificada de Servicios ZXUP10. Santiago de Cali, 2005. 1 archivo de computador.

2.5.1. Parlay Gateway. Es el núcleo del sistema ZXUP10, el cual encapsula varios niveles de la red de recursos, provee las interfaces API's abiertas de acuerdo al protocolo Parlay y ofrece una plataforma para el desarrollo de

servicios por terceras partes. Es el enlace entre las aplicaciones y los elementos de la red. Dentro del Parlay gateway se encuentra el framework. El Parlay gateway está compuesto por tres módulos funcionales: el módulo de adaptación de protocolos, módulo de funciones básicas y el módulo de control de servicio.

- **Modulo de adaptación de protocolos:** Este modulo, es la misma capa de adaptación nombrada anteriormente dentro de la estructura de la UP10, la cual tiene como función seleccionar el protocolo adecuado de acuerdo al tipo de red
- **Modulo de funciones básicas:** Este modulo es la capa de control y gestión del Parlay gateway y es igualmente la capa de control de la estructura de la UP10, es aquí donde se encuentra el framework y las SCF's
- **Modulo de control de servicio:** Este modulo implementa las API's de Parlay, relacionadas con los servicios de telecomunicaciones y recibe las invocaciones de aplicaciones de terceras partes permitiendo a estas aplicaciones comunicarse con las otras capas de la red a través de las API's de Parlay.

Figura 8. Parlay Gateway



Fuente: ZTE. Plataforma Unificada de Servicios ZXUP10. Santiago de Cali, 2005. 1 archivo de computador.

2.5.2. Servidor de aplicaciones. Todos los servicios se encuentran dentro del servidor de aplicaciones. Este es el responsable del mantenimiento lógico y la ejecución de los servicios. Es aquí donde corren los servicios desarrollados

por diferentes fabricantes. Los proveedores de las aplicaciones obtienen el acceso a los módulos del parlay gateway a través del framework después de ser autenticados.

Varios servidores de aplicaciones pueden ser conectados al mismo parlay gateway y aunque utilizan los mismos recursos de este, sus servicios son ejecutados de forma independiente, por lo cual si un servidor de aplicaciones falla no afectara a los demás. Tanto el parlay gateway como el servidor de aplicaciones pueden estar en diferentes servidores o integrados en uno solo.

El servidor de aplicaciones es un elemento independiente del softswitch en la capa de control permitiendo que los servicios puedan ser separados o independientes, facilitando esto de igual forma la introducción de nuevos servicios desarrollados por terceros.

2.5.3. Servidor de medios. Es un dispositivo independiente en el sistema UP10, el cual suministra recursos de medios dedicados en el sistema de softswitch. El servidor de medios tiene funciones como: recolección y decodificación de señales DTMF (Dual Tone Multi-Frecuency), generación y envío de tonos de señal, anuncios, conferencia y conversión entre los diferentes esquemas de decodificación, funciones de comunicación y de procesamiento de medios en los servicios básicos y funciones de gestión y mantenimiento.

2.5.4. Servidor TTS (Text To Speech). El servidor TTS realiza la conversión de texto a voz sintética por medios automáticos, donde a partir del texto se genera una voz artificial generando un sonido idéntico al de una persona leyendo en voz alta. Esta conversión es enviada al servidor de medios para luego enviarla a los usuarios.

2.5.5. Servidor web. Este servidor permite desplegar páginas web las cuales brindan información relacionada con los servicios. El usuario puede realizar consultas acerca de las características de los servicios tales como facturación, estado del servicio, contenidos del servicio etc.

2.5.6. Consola de mantenimiento (OAM). La consola OAM es utilizada para la gestión y mantenimiento de la plataforma UP10. Tiene implementadas funciones tales como: configuración y modificación de datos, alarmas, mediciones, consultas de información, gestión de operación, mantenimiento de interfaces y de aplicaciones de terceras partes.

Aunque para la desarrollador de las aplicaciones o el proveedor del servicio no es tan importante tener un conocimiento detallado de la infraestructura de la red, es necesario tener una visión general del tipo de redes con las que cuenta

EMCALI, por las cuales presta servicios a sus usuarios, como está conformada y como se puede acceder a los servicios. De esta forma el desarrollador o proveedor del servicio podrá proponer servicios enfocados al tipo de usuarios con los que cuenta EMCALI, refiriéndose en este punto a si utilizan la red GSM, PSTN, Banda ancha, WI-MAX, ATM entre otras. También es importante conocer como se relacionan cada una de las capas de la red multiservicios pues estas van a ser las encargadas de llevar a cada uno de los usuarios el servicio solicitado.

La plataforma ZXUP10 es donde se implementan los servicios, conocer su infraestructura interna si es de vital importancia para el desarrollador del servicio pues es con esta con quien va a interactuar directamente al momento de poner en marcha su aplicación. Tener un previo conocimiento acerca de los elementos que la componen, como utilizar cada uno y su función permite al desarrollador una forma más óptima y rápida de montar los servicios.

El desarrollador debe tener en cuenta todo el entorno de la plataforma multiservicios en lo que se refiere al montaje de los servicios y donde van a correr. Para el objetivo de este proyecto que es la definición del ambiente de desarrollo para la implementación de nuevos servicios sobre la red multiservicios de EMCALI, es un gran avance conocer la infraestructura física de la plataforma pues permite dar conocer con que elementos se cuenta, donde se pueden montar los servicios, en donde se encuentran los módulos funcionales que hace cada uno y en que tipo de aplicaciones se pueden usar ya que los servicios actualmente no están siendo desarrollados utilizando esta plataforma ni la arquitectura OSA/Parlay.

3. ANÁLISIS DE TECNOLOGÍAS PARA LA CREACIÓN DE SERVICIOS SOBRE LAS REDES NGN

Las redes NGN (Redes de nueva generación), permiten el desarrollo de nuevos servicios tomando en cuenta las necesidades del cliente, teniendo como base fundamental la disminución de costos operacionales y de infraestructura para los operadores de redes, lo cual permitirá aumentar la oferta de servicios para los diferentes tipos de clientes.

El desarrollo de aplicaciones en Internet es muy cerrado y parecido al desarrollo de aplicaciones NGN por lo cual es necesario que los desarrolladores se adapten a ambientes de desarrollo dinámicos como java o XML en donde contarán con muchas opciones en cuanto a interfaces de programación.

Para la comparación de las tecnologías a utilizar en el desarrollo de servicios se deben tener en cuenta factores como: capacidades de soporte de la red, arquitectura, abstracción de interfaces, tipo de interfaces (lenguajes de descripción), capacidad para el desarrollo de servicios por terceras partes, facilidad de uso, soporte, madurez, y futuras pruebas.

El primer factor se basa en las características de la red, en la abstracción de su infraestructura para que pueda ser usada por las aplicaciones desarrolladas y explotar sus funcionalidades.

Para definir las diferentes capacidades de la red se ha tomado como base la arquitectura OSA/Parlay que ha definido un conjunto de capacidades las cuales son mencionadas a continuación:

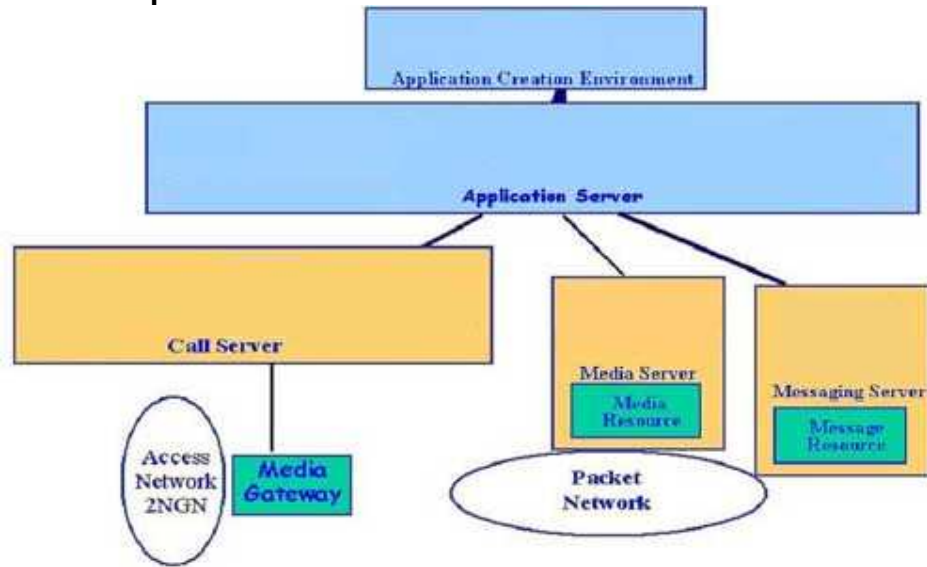
- **Framework:** parte de la arquitectura OSA/Parlay la cual provee las interfaces necesarias para acceder al los servicios de forma segura
- **GCC (Generic Call control):** Es el conjunto de procedimientos que se requieren para establecer una llamada entre dos usuarios
- **MPCC (Multi-Party Call Control):** Permite establecer una llamada entre múltiples usuarios
- **MMCC (Multi-Media Call Control):** Cumple la misma función de MPCC pero multimedia
- **CCC (Coferencing Call Control):** Igualmente trabaja como MPCC pero maneja sesiones de conferencia y subconferencias

- TPCC (Third Party Call Control): Permite a una aplicación crear o terminar una llamada entre dos usuarios
- GUIN (Generic User Interaction): Permite al llamante interactuar con un sistema de información, solo debe digitar el número correspondiente a la acción que requiere
- UL (User Location): es utilizado para lograr la ubicación geográfica de u usuario
- US (User Status): Define la disponibilidad del usuario, define eventos y notificaciones de cambios en el estado de comunicación del usuario
- DASC (Data Access Session Control): Provee el control y establecimiento de sesiones de datos
- Messaging
- TC (Terminal Capabilities): Permite definir las características del equipo del usuario para adaptar el tipo de información que es enviada según el equipo final
- UP (User Profile): Maneja la información de los perfiles de usuario, que servicios utiliza, cuales son sus preferencias, que aplicaciones puede utilizar

Nota: Algunas de estas capacidades serán definidas con mayor detalle mas adelante.

El Segundo criterio se enfoca en la arquitectura en cuanto a que módulos cubre una tecnología teniendo en cuenta la categorización propuesta en el proyecto P1109 como arquitectura de referencia.

Figura 9. Arquitectura de referencia P1109



Esta arquitectura en capas define las diferencias entre tecnologías, dependiendo de sus características. La capa de servidor de aplicaciones incluye tecnologías usadas para ejecutar servicios, las cuales son programadas con herramientas representadas por la capa de entorno de creación de aplicaciones (Application Creation Environment).

El nivel de Call Server incluye tecnologías de manejo de enrutamiento y entrega de llamadas de voz, el nivel de media Server representa las tecnologías que incluyen comunicaciones multimedia, el nivel de servidor de mensajería esta en favor de entidades de manejo de mensajes y comunicaciones asíncronas y por ultimo el nivel de media gateway representa las tecnologías relacionadas con las redes.

El tercer criterio es la evaluación de interfaces ofrecidas por las tecnologías para los desarrolladores. La evaluación de interfaces esta definida por tres aspectos: un nivel de abstracción (AIL), el tipo de interfaz (KOI), y el lenguaje de descripción de interfaces (IDL).

Con respecto al nivel de abstracción, una interfaz abstracta oculta los detalles técnicos de la tecnología hacia el desarrollador, para ganar más portabilidad y facilidad de uso. Una interfaz de nivel medio oculta parte de los detalles de la tecnología, pero requiere algún nivel de conocimiento por parte del desarrollador. Una interfaz de nivel bajo, provee detalles de acceso de la tecnología correspondiente, tiene menos portabilidad y códigos mas largos de difícil entendimiento.

El tipo de interfaz debe describir el método de comunicación por el cual la tecnología en cuestión expone las capacidades de la red hacia sistemas externos. Esto incluye las siguientes categorías:

- API
- Interfaz basado en protocolo
- Lenguaje de scripting

El tipo de interfaz define el lenguaje usado para definir su API, esto puede ser utilizando lenguajes como java, c++, middleware basado en OMG IDL en corba, WSDL en servicios web, o definición de datos basado en XML DTD, CPL o XML.

Otro criterio es la programabilidad la cual describe la flexibilidad de la tecnología en cuanto a las aplicaciones desarrolladas por terceras. Los servicios que proveen las terceras partes describen la calificación de la tecnología.

El siguiente criterio es la facilidad de uso. Esto puede medirse en función de:

- Los conocimientos necesarios por el desarrollador, es decir cuanto conocimiento y experiencia es requerido por la tecnología
- Tiempo de servicio: que tan rápido se desarrollan e implementan servicios con determinada tecnología
- Potencia: el alcance que se puede lograr usando una tecnología determinada

Otro tema importante para evaluar es la industria y el soporte, las cuales miden la disponibilidad y la madurez de la tecnología, mostrando que tanto es soportada en la industria y el nivel de madurez comparado con otros estándares.

Finalmente el criterio de evaluación de la tecnología debe identificar el futuro para esta, describiendo que tan bien responderá ante las nuevas tecnologías emergentes en la industria.

3.1. CREACIÓN DE SERVICIOS EN REDES (NGN)

El enfoque de la creación de servicios en Redes de Nueva Generación (NGN) puede resumirse en tres categorías: basado en APIs programables, basado en lenguajes de scripting o en entornos gráficos de creación de servicios (SCE).

3.1.1. API (Application Programming Interfaces). Una gran variedad y tipos de API's están emergiendo en los productos de las redes de siguiente generación (NGN), proporcionando diferentes niveles de funcionalidad.

Algunos son estándares conformados (JAIN, SIP-servlet) y otros no. Esto puede presentar confusión en la comunidad de desarrolladores cuando tienen que escoger cuál API utilizar para algún servicio en particular.

Por otro lado, los desarrolladores pueden escoger libremente las herramientas que desean usar, vinculando los componentes que ofrecen las APIs para lograr la conectividad con los protocolos de las redes NGN.

Algunos sistemas proveen varios niveles de API's (abstracto, medio y bajo), lo cual brinda a los desarrolladores la flexibilidad de escoger el nivel de abstracción más apropiado para una determinada aplicación (nivel bajo para controlar todos los protocolos y detalles específicos de la red y nivel alto para ocultar algunos detalles de la red).

3.1.2. Lenguajes de Scripting. Los lenguajes de scripting son lenguajes livianos, personalizables e interpretados, apropiados para el desarrollo rápido de aplicaciones, ya que proveen conexiones entre componentes ya existentes. Estas características permiten que sean usados para realizar código o modificar aplicaciones al mismo tiempo, e interactuar con programas en ejecución.

Los lenguajes de scripting están representados por archivos basados en XML, el comportamiento del servicio puede ser modificado mientras esta corriendo, haciendo como una reconfiguración dinámica. Por su calidad y sus características estos lenguajes pueden ser utilizados en el campo de la programación de aplicaciones. Típicamente los scripts son creados, editados, y validados usando un editor convencional o como un resultado de las técnicas de transformación aplicadas.

Por ejemplo el Service Creation Markup Language (SCML) y el Call Processing Language (CPL), utilizan lenguajes de scripting ya que permiten conectar los componentes existentes con una API particular dependiendo del contenido del archivo del script. Esto permite obtener prototipos y desarrollos de aplicaciones rápidamente. También pueden presentarse casos en los que el mismo usuario pueda editar las características de un servicio y personalizarlo.

3.1.3. SCE (Service Creation Enviroments). Usando entornos gráficos de creación de servicios se logra un rápido desarrollo de servicios, aun si se tiene poco conocimiento sobre los protocolos de red. La lógica del servicio es diseñada usando diagramas de flujo de componentes básicos ofrecidos por el SCE. Estos componentes ofrecen conectividad con diferentes protocolos, utilizando código externo (generalmente Java y C++), y proporcionando conectividad con bases de datos.

Cada uno de las instancias de los componentes deben ser programadas, pero el aprendizaje de SCE es rápido y sencillo y libera a muchos desarrolladores de algunos aspectos técnicos.

Dependiendo del SCE escogido por el desarrollador, éste puede tener más o menos control sobre los campos que están relacionados directamente con los protocolos que se requieren para el funcionamiento del servicio que se está creando. Puede tener más control pero necesita un profundo conocimiento de aspectos técnicos y puede tener menos control y el desarrollo de aplicaciones es más fácil pero no es seguro que pueda ser escalable o un servicio más complejo.

3.2. EVALUACIÓN DE TECNOLOGÍAS PARA LA CREACIÓN DE SERVICIOS

A continuación se realizará una descripción de cada una de las tecnologías más importantes utilizadas para el desarrollo de servicios en redes NGN teniendo en cuenta tanto los criterios de evaluación mencionados anteriormente como el enfoque para la creación de servicios en redes NGN.

3.2.1. Servicios web. El principal objetivo de la arquitectura de servicios Web es la realización de una red interoperable entre servicios, en donde por medio de un conjunto de protocolos y estándares se pueden intercambiar datos entre aplicaciones. Esto es muy importante ya que permite interactuar con aplicaciones desarrolladas por terceras partes y permite exportar servicios por un operador de red o un proveedor de servicios.

Los servicios Web pueden exportar sus servicios por medio de interfaces WSDL (Web Services Definition Language), el cual es el lenguaje de interfaz pública para los servicios Web, permite establecer un acuerdo entre un servicio y un cliente en cuanto a detalles de transporte de mensajes y su contenido (sintaxis y mecanismos de transporte), esta basado en XML (Extensible Markup Language) como una descripción de los requisitos funcionales necesarios para establecer una comunicación con los servicios Web. Los servicios se comunican a través del protocolo SOAP (Simple Object Access Protocol), sobre el cual se establece el intercambio de datos.

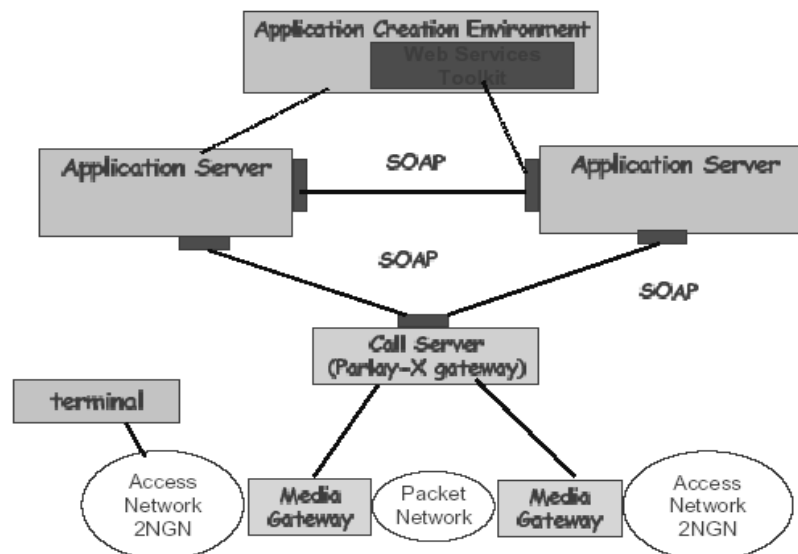
El descubrimiento y registro de los servicios son implementados en el registro UDDI (Universal Discovery, Description and Integration), utilizado para publicar información de los servicios Web, y comprobar que servicios están disponibles. XML (Extensible Markup Language) es utilizado como formato estándar para los datos que se vayan a intercambiar por medio de los mensajes SOAP, que se basa en los actuales protocolos de Internet como HTTP. La implementación de servicios Web necesita que las API's sean traducidas en WSDL y el servidor de aplicaciones donde son desplegados los servicios Web deben traducir los mensajes SOAP a las interfaces (java).

Dentro de los servicios Web no están definidas las capacidades de la red establecidas por la arquitectura OSA/Parlay, ya que su principal objetivo es la

interoperabilidad entre aplicaciones de software para servicios Web independientemente de sus propiedades y de las plataformas en que se instalen. Otras características relevantes de estos servicios es que fomentan los estándares y protocolos basados en texto, facilitando el acceso a su contenido y su entendimiento, también permite que servicios desarrollados por diferentes compañías en diferentes sitios geográficos y en plataformas de distintos fabricantes puedan fusionarse fácilmente para proveer servicios integrados.

Con respecto a la arquitectura P1109, los servicios corren sobre diferentes servidores de aplicaciones, los cuales pueden comunicarse utilizando mensajes SOAP. Otro método para implementar servicios puede estar conformado por un Call Server utilizando interfaces WSDL que ofrecen conectividad a la red con los servicios de terceras partes corriendo en un servidor de aplicaciones externo.

Figura 10. Servicios Web con respecto a la arquitectura P1109



Fuente: FALCARIN Paolo; LICCIARDI, Carlos Alberto. Analysis of NGN service creation technologies [en línea]. Torino: Politécnico di Torino. [Consultado 12 de marzo de 2008]. Disponible en Internet: <http://softeng.polito.it/falcarin/docs/Annals03.pdf>

Los servicios Web son una buena solución para integrar aplicaciones heterogéneas aportando independencia entre la aplicación que usa el servicio Web y el propio servicio, lo cual es muy importante ya que la tendencia de construir aplicaciones a partir de componentes distribuidos es cada vez mas utilizada. A pesar de ser una buena solución esta tecnología aun no ha madurado para su amplia implementación ya que aun presenta algunos inconvenientes como:

- No hay un estándar definido para las transacciones XML y su grado de desarrollo no puede compararse con estándares abiertos de computación distribuida como CORBA.
- Su rendimiento es bajo, este es un inconveniente presentado por el uso de un formato basado en texto
- No hay manera de establecer parámetros y QoS para manejar la tolerancia a fallos y la alta disponibilidad del servicio.

3.2.2. SIP servlets. SIP (Session Initiation Protocol), es un protocolo que permite establecer, modificar y terminar sesiones IP multimedia. Soporta diferentes servicios como telefonía IP, mensajería instantánea, transferencia de llamada entre otros.

Las aplicaciones SIP son programas que utilizan un SIP servlet el cual es un componente de aplicación basado en java que contiene un conjunto de librerías utilizadas para la creación de servicios sobre redes IP.

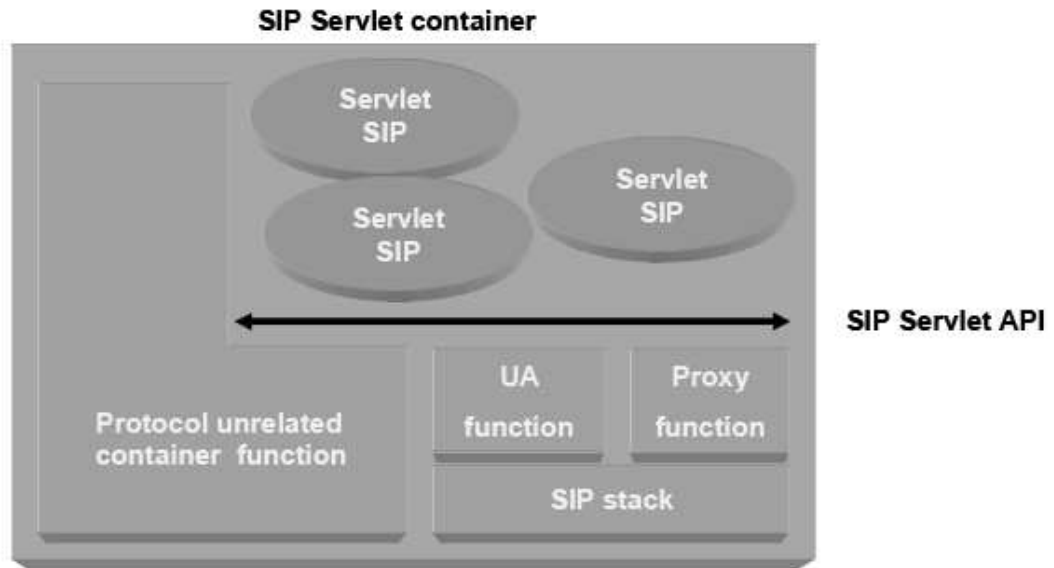
Un SIP servlet es gestionado durante su tiempo de vida por un contenedor de SIP servlet el cual hace parte de un servidor de aplicaciones que proporciona los servicios de red a través de los cuales se reciben y envían las peticiones y las respuestas. Aquí son alojadas las aplicaciones y el contenedor decide cual invocar y en qué orden ya que puede recibir varias peticiones al mismo tiempo.

Las especificaciones del SIP servlet también tiene el objetivo de estandarizar los siguientes aspectos del contenedor servlet: arquitectura, modelo de seguridad, el formato de datos (XML, DTD), formato de archivos basado en .jar (similar al formato de archivo WAR utilizado por servlets HTTP).

SIP servlet es adecuado para servicios desarrollados por terceras partes. Se podría decir que el desarrollo de servicios por terceros es bastante simple ya que son vistos como librerías de java.

Teniendo en cuenta la referencia P1109 el contenedor SIP servlet se encuentra en el nivel de servidor de aplicación, y como es una nueva API estándar es ideal para el desarrollo de servicios por terceras partes. También tiene funciones SIP como agente usuario la cual es una aplicación sobre un equipo de usuario que emite y recibe peticiones SIP a través de un software instalado en el equipo que puede ser un PC, un teléfono IP, etc, y funciones como Proxy donde recibe solicitudes de clientes que el mismo trata o encamina hacia otros servidores después de haber realizado modificaciones sobre estas solicitudes.

Figura 11. SIP servlet y contenedor SIP servlet



Fuente: FALCARIN Paolo; LICCIARDI, Carlos Alberto. Analysis of NGN service creation technologies [en línea]. Torino: Politécnico di Torino. [Consultado 12 de marzo de 2008]. Disponible en Internet: <http://softeng.polito.it/falcarin/docs/Annals03.pdf>

Las especificaciones del API SIP servlet en su primera versión esta aun en etapa de revisión (no se ha dado a conocer), se ha previsto una segunda versión pero esta debe normalizar aspectos del contenedor SIP servlet.

3.2.3. Jain SIP lite. El API de JAIN SIP Lite es una aplicación java encaminada solamente a aplicaciones de agente usuario. El objetivo de la JAIN SIP Lite API es mejorar los esfuerzos realizados por JAIN-SIP y SIP Servlets API que requieren de un programador experto. JAIN-SIP Lite ofrece una API de alto nivel capaz de manejar algunos aspectos técnicos de SIP transparentes al programador, lo que permite el desarrollo de aplicaciones que tengan a SIP como protocolo principal sin que el desarrollador tengan un profundo conocimiento acerca de este, facilitando un rápido desarrollo de aplicaciones del tipo agente usuario con API's muy fáciles de usar.

La principal diferencia comparado con SIP servlets es que JAIN SIP Lite no necesariamente aborda el desarrollo de aplicaciones en un servidor de aplicaciones y no tiene una función de Proxy como soporte de la plataforma.

Con respecto a la arquitectura P1109 el API de JAIN SIP LITE puede residir en el servidor de aplicaciones o en las aplicaciones de escritorio. Como API estándar es ideal para el desarrollo de servicios por terceras partes y sus especificaciones son frecuentemente estandarizadas con el JPC. Aunque aun se encuentran en revisión y no están disponibles hacia el publico.

3.2.4. Voice XML (Voice eXtensible Markup Language). Ha sido definido como una tecnología que permite al usuario interactuar con el servidor Web a través del reconocimiento de voz, explotando las características del servidor de medios.

VoiceXML es un lenguaje de etiquetas basado en XML que permite describir servicios de voz con independencia de la aplicación en la que corran. De esta manera no es necesario conocer detalles específicos de una plataforma para entender el funcionamiento del sistema de diálogo⁷.

De igual forma VoiceXML permite crear diálogos con los que se puede interactuar escuchando comandos hablados, controlables a través de entradas de voz, donde VoiceXML se encarga de convertir el habla en texto. Requiere reconocimientos de speech para identificar el patrón de un determinado usuario.

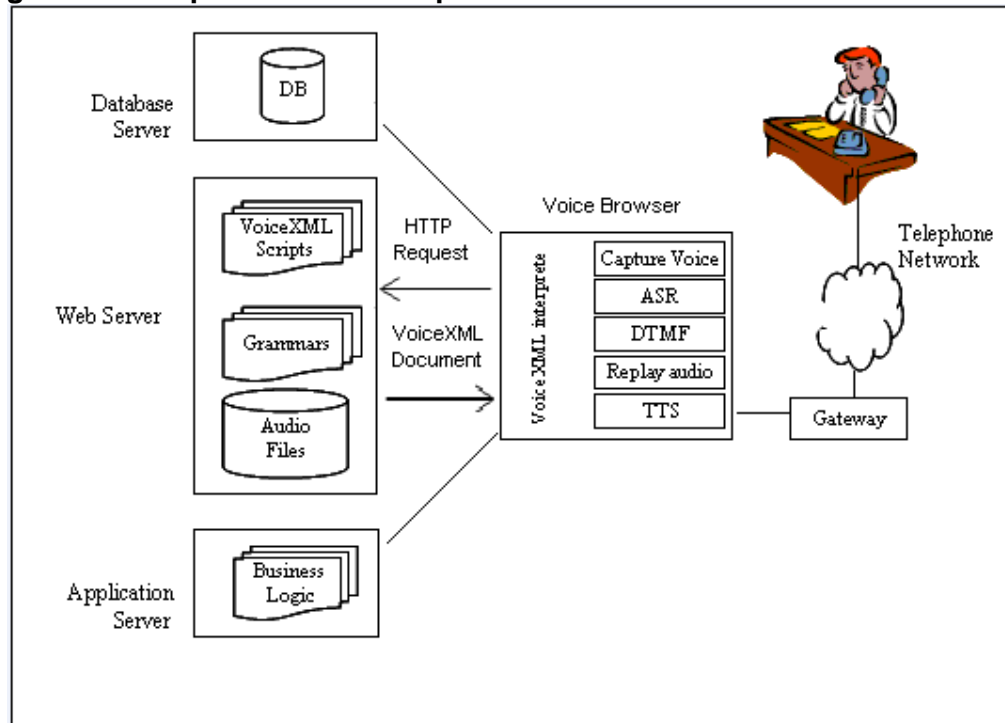
Las herramientas construidas y adaptadas para soportar el estándar comparten un formato de definición de sistemas de diálogo que permite la portabilidad y transferencia de datos entre aplicaciones heterogéneas. Además este tipo de herramientas tienden a facilitar enormemente el trabajo del desarrollador, que no necesita conocer los detalles de implementación del sistema que está construyendo.

La arquitectura interna de VoiceXML esta conformada por dos partes:

- **Voice browser:** es el software que permite hacer una secuencia de dos diálogos entre el usuario y el sistema. Esta hecho de un núcleo interprete voice XML, integrando componentes software para texto y audio para la producción, grabación y reconocimiento de voz. Es aquí donde el código es interpretado y mostrado. Todos los teléfonos pueden ser usados para marcar hacia el buscador de voz
- **Web Server:** aquí es donde residen las páginas de aplicación, estas pueden ser VoiceXML, php, asp, jsp. Para de forma dinámica crear paginas VoiceXML.

⁷ GARCIA ALVAREZ, Víctor. Estándar VoiceXML [en línea]. Asturias: Universidad de Oviedo, 2004. [Consultado 28 de abril de 2008]. Disponible en Internet: <http://www.di.uniovi.es/~cueva/asignaturas/doctorado/2006/trabajos/VoiceXML.pdf>

Figura 12. Arquitectura básica para un servicio VoiceXML



Fuente: GARCIA YAGUE, Adolfo. VoiceXML, CCXML y SALT Convergencia voz y aplicaciones [en línea]. Sevilla: Centro de investigaciones científicas Isla de la Cartuja, 2004. [Consultado en abril 25 de 2008]. Disponible en internet: <http://80.34.206.133/netica.com/convergencia-voz-app-v1.2.pdf>

Un usuario marca hacia un número en particular correspondiente al voice browser el cual envía una petición HTTP a un determinado servidor Web. Luego el mismo voice browser es el encargado de recibir un documento (texto) VoiceXML desde el servidor Web a través del intérprete VoiceXML, el cual lo pasa a voz para que los mensajes puedan ser escuchados por el usuario, permitiéndole de igual forma a este último acceder y manipular datos por medio del servidor de base de datos e invocar las aplicaciones que se encuentran en el servidor de aplicaciones.

VoiceXML es un estándar ampliamente aceptado, por medio del cual las aplicaciones de voz son desarrolladas en Internet. Esta representado en el nivel de servidor de medios en la arquitectura P1109, y la interfaz entre el servidor de aplicaciones y el servidor de medios no ha sido totalmente definida. Es posible que SIP y VoiceXML sean la interfaz entre la próxima generación de servidores de aplicaciones y los servidores de medios. Estos pueden ser utilizados juntos para iniciar o terminar sesiones, no solo señalización y control sino también contenido de sesiones.

Una de las desventajas de VoiceXML es que implementa funciones de telefonía, pero el conjunto de funciones es demasiado reducido y su

funcionalidad muy limitada. Por ello los sistemas desarrollados con VoiceXML no consiguen alcanzar la complejidad necesaria para algunos flujos de telefonía y operaciones de voz (tomado texto de voicexml.pdf).

VoiceXML es utilizado de manera conjunta con otros estándares, proporciona una base sólida para la definición de sistemas de voz. Las constantes revisiones y ampliaciones del estándar aseguran su continuidad y progresiva incorporación en las herramientas para la construcción de aplicaciones de voz.

3.2.5. CCXML (Call Control eXtensible Markup Language). Es un lenguaje que adopta una arquitectura basada en la red para ejercer el control de llamada (gestión de llamadas, proceso de eventos y conferencias) en una aplicación telefónica. Utiliza XML como sintaxis, lenguajes de script y puede emplear varias definiciones para el control de llamada entre ellas JAIN CALL CONTROL

CCXML es un lenguaje guiado por eventos, se comporta como una maquina de estados, lo que le permite manejar eficazmente los eventos telefónicos y lanzar las aplicaciones voiceXML. Con respecto a VoiceXML, la tecnología CCXML ha sido diseñada para ser integrada y complementar a VoiceXML, ya que este ultimo no soporta algunas características como el manejo de varias conferencias multimedia, asignarle a cada llamada su propio voice interpreter, la flexibilidad ya que VoiceXML solo se enfoca en la interacción vocal de la aplicación pero han sido integradas para trabajar juntas precisamente para proveer la interacción de diálogos de voz con el usuario.

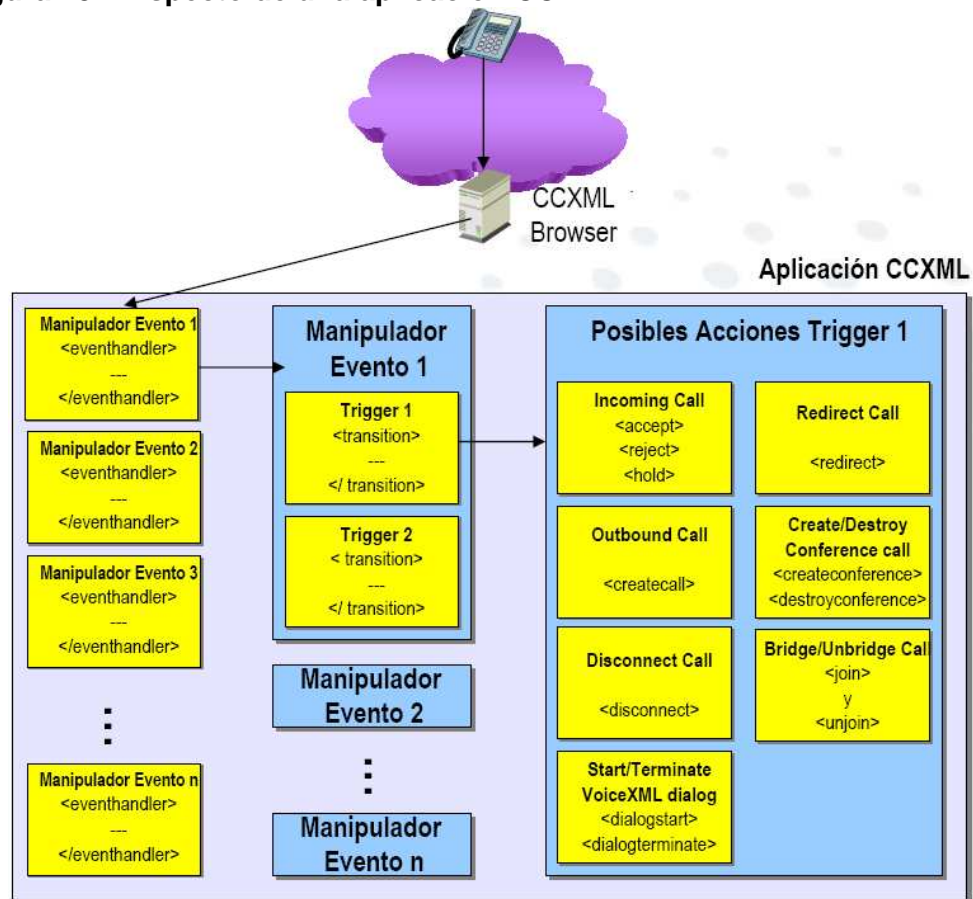
Entre las características más importante de CCXML son la facilidad de uso, la flexibilidad como ya se había mencionado anteriormente y la capacidad de procesar aplicaciones complejas.

Con respecto a las capacidades, CCXML tiene como objetivo proveer las características de control de llamada multimedia que incluye:

- Llamadas salientes: Realiza y controla las llamadas salientes, iniciando un dialogo VoiceXML cuando la conexión es establecida
- Control de llamadas múltiples: Controla todas las llamadas e inicia un dialogo VoiceXML para cada una
- Whisper transfer: Envía un mensaje al destinatario antes de establecer la conexión con el llamante
- Conferencia: Permite a múltiples participantes unirse a una llamada telefónica
- Gestión de llamadas: Por medio de un agente de soporte son gestionadas las llamadas de los usuarios

- Manipulación de eventos: Manejo de eventos asíncronos que vienen desde la infraestructura telefónica y el interprete VoiceXML
- Sesión de inicio y finalización del interprete voiceXML: inicia o detiene una sesión VoiceXML en cualquier momento
- Sesión de inicio del interprete CCXML: Durante la ejecución de una aplicación, otra nueva puede ser puesta en marcha
- Enrutamiento: Encamina una llamada a la siguiente extensión disponible

Figura 13. Aspecto de una aplicación CCXML

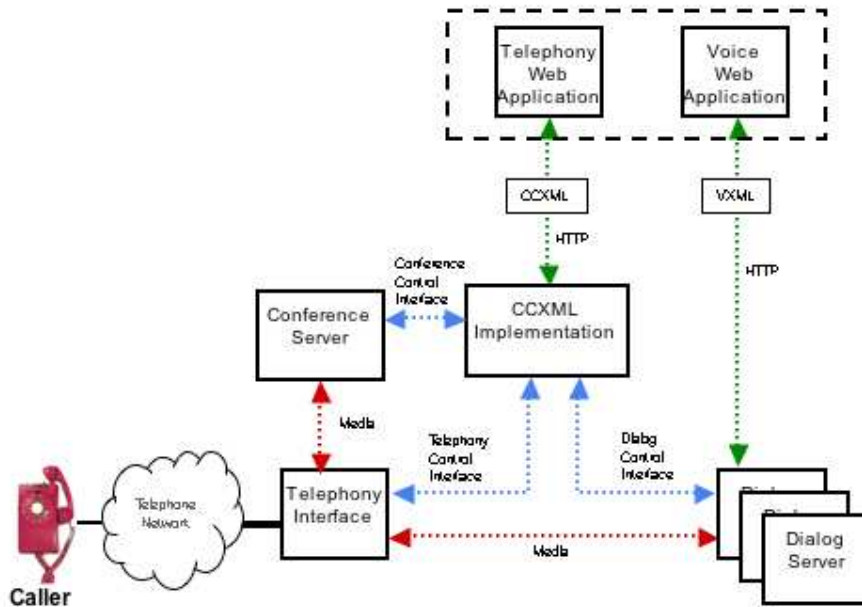


Fuente: GARCIA YAGUE, Adolfo. VoiceXML, CCXML y SALT Convergencia voz y aplicaciones [en línea]. Sevilla: Centro de investigaciones científicas Isla de la Cartuja, 2004. [Consultado en abril 25 de 2008]. Disponible en internet: <http://80.34.206.133/netica/com/convergencia-voz-app-v1.2.pdf>

En la arquitectura de referencia P1109 el intérprete CCXML encaja bien en el Call Server y el script CCXML puede ser alojado o dinámicamente generado en el servidor de aplicaciones. La interfaz entre el CCXML call Server y el

servidor de aplicaciones puede ser http y de igual forma entre el call Server y el servidor de medios VoiceXML.

Figura 14. Arquitectura CCXML



Fuente: AUBURN, R.J. Voice Browser Call Control: CCXML 1.0 [en línea]. Estados Unidos, 2004. [Consultado 28 de abril de 2008]. Disponible en Internet: <http://www.voxeo.com/>

En cuanto al desarrollo de aplicaciones por terceros CCXML sigue en gran medida el enfoque propuesto por VoiceXML, pero aun se deben resolver algunos puntos en cuanto a relaciones de confianza y autenticación.

3.2.6. JCC (Java Call Control). JCC (Java Call Control) es un API, que provee una interfaz para un modelo de control de llamada genérico (GCC), esta interfaz es utilizada para implementar todos los procesos propios de un protocolo de señalización como crear sesiones, modificar, cancelar, cerrar, etc. Las características (SCF's) pueden ser invocadas o descargadas durante la configuración de sesiones. Por medio de JCC los desarrolladores pueden hacer aplicaciones para ser ejecutadas desde cualquier plataforma que soporte la API, permitiendo a los proveedores de servicios ofrecer mayor cantidad de servicios de forma rápida y eficiente, desarrollándolos ellos mismos, por outsourcing o por terceras partes.

JCC incluye las facilidades para observación, inicialización, respuesta, proceso, manejo de llamadas, permite también invocar aplicaciones y retornar resultados durante el desarrollo de las llamadas. Una llamada puede incluir sesiones multimedia, tripartita, etc, sobre la red. El API es usada para implementar una gran variedad de aplicaciones de voz y de datos como: originar y terminar llamadas, VPN (Voice Virtual Private Network), traducción de número de

teléfono, activación de marcación de voz, clic para marcar, conferencias entre otros.

JCC esta encaminado hacia redes convergentes, y esta previsto para ser implementado con softswitches, call agents, proporcionando una interfaz abstracta para las aplicaciones. Puede ser utilizado para manejar sesiones en redes PSTN, redes de paquetes IP o ATM, redes inalámbricas o una combinación de ambas, sin afectar el desarrollo de los servicios usando la API.

Con respecto a la arquitectura P1109 JCC puede ser implementado de dos formas: la implementación JCC puede ser incluida como una librería en el servidor de aplicaciones a fin de dar conectividad a la red con el call Server o puede ser efectuado directamente con el nivel media gateway.

3.2.7. CPL (Call Processing Language). CPL es un lenguaje de script basado en XML que permite manejar la entrada y salida de llamadas en redes NGN, utilizando SIP como mecanismo de señalización para el establecimiento de conexiones, es decir, CPL define servicios que capturen eventos de llamadas y los manejen de forma adecuada.

Para la implementación de servicios, CPL puede correr sobre un servidor Proxy SIP y los scripts CPL creados por usuarios finales pueden ser cargados en servidores SIP para configuración de llamadas en un entorno seguro.

CPL es liviano, eficiente, fácil de implementar, escalable porque es posible adicionar funciones personalizadas de manera que los script existentes continúen trabajando. Es un lenguaje de alto nivel de abstracción, estandarizado que permite crear servicios personalizados por lo clientes donde pueden fácilmente escribir y editar sus aplicaciones. Como CPL es basado en XML, el tipo de interfaz ofrecida por el nivel de aplicación son los mismos lenguajes de script.

CPL puede ser utilizado para implementar servicios en diferentes escenarios, usando scripts creados por el usuario final y montados en un servidor, usando scripts creados por los administradores del servidor en nombre de los usuarios, o usando scripts creados por aplicaciones Web que los transforman en CPL. El modelo de creación de aplicaciones propuesto por CPL es independiente del sistema operativo y del lenguaje de programación utilizado para la ejecución de la aplicación. Otra ventaja es que CPL es un lenguaje que ha madurado rápidamente y esta especificado en su totalidad

Algunas limitaciones que presenta CPL son en cuanto al procesamiento de llamadas ya que CPL no puede originar llamadas hacia dos o más usuarios porque es activado solo a través de llamadas relacionadas con eventos y no puede ser usado para crear scripts complejos. Esto es, porque la estructura CPL esta definida sobre DTD (Declaración tipo de documento), que es uno de los lenguajes utilizados para definir y validar la estructura de archivos XML,

pero no es tan flexible como el esquema XML, que es mas extensible y mas fácil de aprender. Otra desventaja es que CPL no propone un entorno distribuido en donde la aplicación y el elemento controlado pueda residir en nodos diferentes, el usuario o el proveedor del servicio deben cargar las aplicaciones en el mismo servidor por el que esta pasando la señalización telefónica, e interactuar sobre ella.

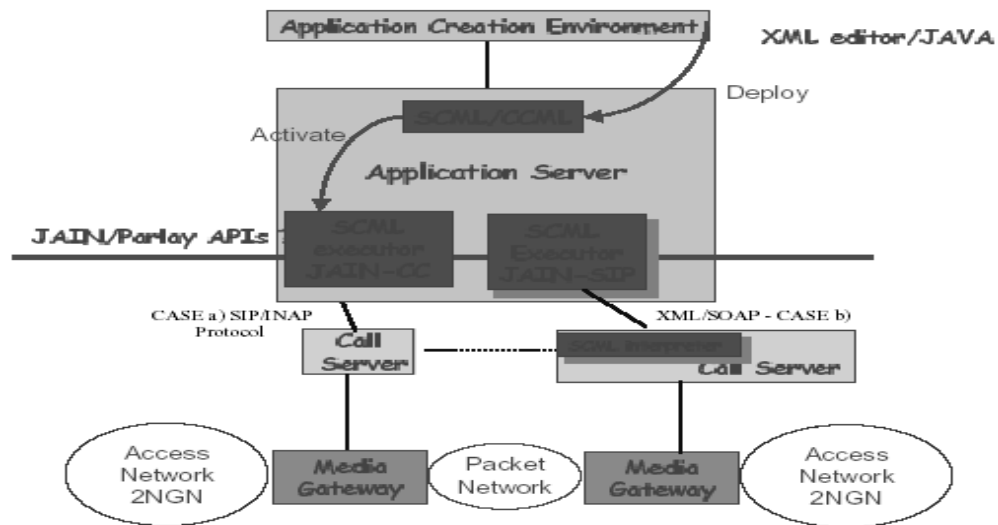
3.2.8. SCML (Service Creation Markup Language). SCML es un lenguaje de script basado en XML schema. Maneja múltiples SCF's como UI (User Interaction), MMCC (Multimedia Call Control), GCC (Generis Call Control), las cuales deben ser provistas por el proveedor del servicio y usados por el desarrollador del servicio, aunque el trabajo se ha enfocado mas hacia la SCF de control de llamada.

Con respecto a la arquitectura los scripts SCML pueden correr sobre el Call Server o sobre un servidor de aplicaciones, no es necesario que el servidor donde se ejecuten los servicios sea el nodo de la red donde se maneje la señalización, aunque esto lleva a la utilización de protocolos de transporte de notificaciones y eventos entre el servidor que contiene la aplicación y el nodo de señalización de la red.

En la fase de creación del servicio y el entorno de creación de la aplicación el script es creado usando herramientas XML (XML, XML editors, XSLT), editores de texto o convirtiendo programas en java y C++. El entorno de creación de la aplicación forma parte de la fase de despliegue donde el script SCML es validado y la sintaxis es guardada en un repositorio. El script será activado descargándolo en el servidor de aplicaciones para ejecutarlo.

El servidor de aplicaciones puede enviar mensajes XML alternadamente (por medio del protocolo SOAP), a otro procesador XML, colocado en el servidor de llamadas haciendo la interacción entre estos dos servidores por medio de un lenguaje independiente (Ver figura 15 caso a). En el caso b como se puede apreciar en la figura 15 el enfoque tiene mejor desempeño y puede ser mas recomendado para interacciones inter-dominio, puede ser mas apropiado para cuando el servidor de aplicaciones y de llamada pertenecen a dos dominios diferentes esto en el caso de proveedores de servicios por terceras partes.

Figura 15. Arquitectura SCML



Fuente: FALCARIN Paolo; LICCIARDI, Carlos Alberto. Analysis of NGN service creation technologies [en línea]. Torino: Politécnico di Torino. [Consultado 12 de marzo de 2008]. Disponible en Internet: <http://softeng.polito.it/falcarin/docs/Annals03.pdf>

La abstracción de la interfaz puede ser considerada una API de alto nivel. Basada en la API JCC, estandarizada por JAIN y por lo tanto es independiente del protocolo. Esto elimina complejidad en la red y permite el manejo básico de eventos para procesar una llamada. Esta tecnología es ideal para programadores que utilizan XML con DTD, las aplicaciones son desarrolladas en corto tiempo.

Una desventaja de SCML es que no tiene un buen mecanismo de seguridad debe depender de los mecanismos desarrollados por OSA/Parlay (interfaces con el framework), por lo cual las API's de OSA/Parlay están siendo especificadas por el grupo JAIN. Con respecto a la tecnología CPL es más potente y flexible.

3.2.9. XHTML (eXtensible Telephony Markup Language). XHTML es otro lenguaje de script basado en XML, orientado a crear servicios de telefonía y multimedia para redes NGN, que se adapta a cualquier red sin apoyarse en un protocolo de señalización específico. Esto no significa que la programación de servicios sea independiente de la red sino que XHTML define un marco genérico para la creación de servicios para cada tipo de red y define los eventos propios de cada tecnología mediante plug-ins.

XHTML está basado en eventos donde la definición del marco genérico se enfoca en la asociación de funciones a los eventos que se quieran tratar en la red. Cuando un servicio registra una serie de eventos se ejecuta la función asociada en una instancia de la aplicación. En XHTML las instancias son definidas como sesiones, y dentro de una aplicación se presentan tantas

sesiones como eventos se van produciendo. Cuando se presenta un evento se ejecuta la primera acción de la función asociada, dependiendo del resultado de esa ejecución se invoca una segunda acción siguiendo de esta forma hasta finalizar el tratamiento de la función. Cada instancia guarda su propio estado de forma independiente permitiendo que un servicio sea aplicado para varios usuarios al mismo tiempo y de forma independiente.

En cuanto a la creación de aplicaciones con XHTML el uso de XML permite tener un entorno de creación de servicios sencillo, la creación de servicios es en cortos espacios de tiempo, son fáciles de depurar y pueden integrarse fácilmente con otras herramientas.

El desarrollo de servicios por terceras parte es limitado ya que XHTML depende de la red y siempre se debe contar con la aprobación del operador para la instalación de las aplicaciones en el servidor que es ante todo un elemento de la red. Una aplicación XHTML debe correr sobre un servidor de aplicaciones que pueda soportar uno o más protocolos de señalización. Si un servidor de aplicaciones corre con una infraestructura muy limitada los desarrolladores de servicios pueden no obtener ganancias.

3.3. OSA/PARLAY:

OSA / Parlay define una arquitectura que permite la interoperabilidad entre las aplicaciones de las tecnologías de la información (TI) y características de las telecomunicaciones en las redes, a través de una interfaz abierta estandarizada. OSA/Parlay es una API (Interfaz de Programación de Aplicaciones) que permite la creación rápida de servicios de telecomunicaciones.

OSA (Open Service Architecture), es una arquitectura flexible que hace parte de la tercera generación de redes de telecomunicaciones para servicios móviles desarrollada por el 3GPP (3rd Generation Partnership Program). Parlay es el grupo que busca crear los estándares necesarios que permita la unión de las TI con las telecomunicaciones.

Las API's de OSA/Parlay están definidas por el Parlay group, el cual, es una organización sin ánimo de lucro conformada por 65 empresas del área de las telecomunicaciones e industrias de las tecnologías de información. Algunas compañías miembros del Parlay Group son Alcatel, British Telecom, Ericsson, Fujitsu, HP, IBM, Lucent, NTT, Siemens, Telecom Italia entre otras.

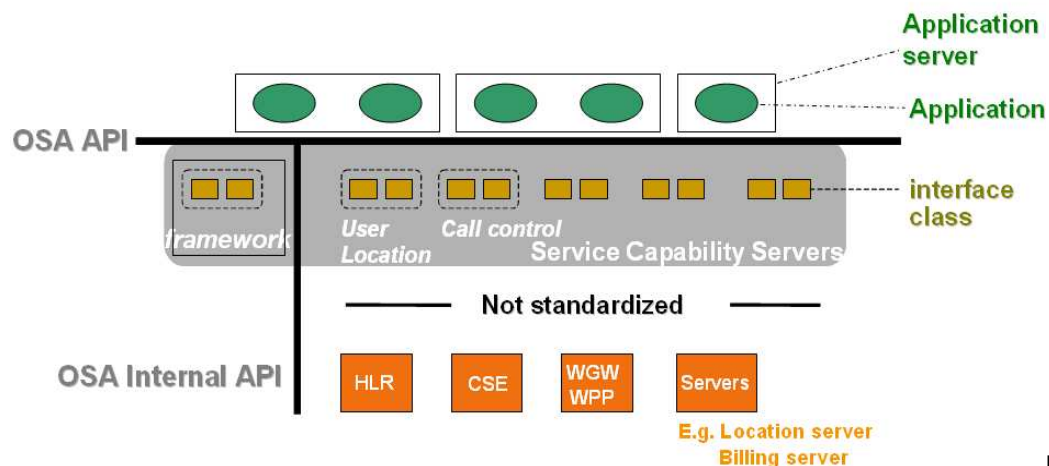
OSA/Parlay es una tecnología independiente y fue diseñada para ser usada por redes móviles, redes fijas y redes de nueva generación (NGN), basadas en el protocolo IP. Esta tecnología se caracteriza principalmente por permitir a los operadores y aplicaciones de terceros acceder a las funcionalidades de la red por medio del conjunto de interfaces abiertas estandarizadas. Para la creación

de aplicaciones los desarrolladores pueden utilizar diferentes lenguajes de programación como java y C++.

OSA/Parlay se basa en estándares abiertos como CORBA, IDL, Java, UML y servicios Web (SOAP, XML y WSDL).

3.3.1. Arquitectura OSA/Parlay.

Figura 16. Arquitectura OSA/Parlay



Fuente: POPOFF, Jeff. Parlay/OSA 101 [en línea]. Boston: The Parlay Group, 2005. [Consultado 23 de febrero de 2008]. Disponible en Internet: <http://www.parlay.org>

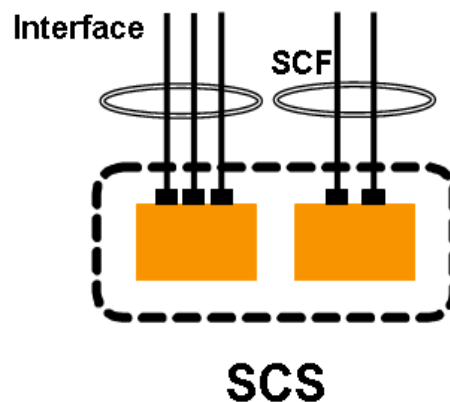
La arquitectura OSA/Parlay está definida por los siguientes componentes:

- Servidor de aplicaciones:** Sobre este servidor son ejecutadas todas las aplicaciones, puede estar conformado por uno o más hosts. En algunos documentos se refieren al servidor de aplicaciones como Parlay client o Parlay client Proxy. Sobre estos servidores también corren los servicios desarrollados por terceras partes. Esto es una gran ventaja ya que el proveedor del servicio utiliza las mismas interfaces para sus propios servicios y los servicios desarrollados por terceras partes
- Parlay gateway:** Este elemento del modelo OSA/Parlay permite conectar las aplicaciones utilizando las API's de Parlay con los elementos de la red. El parlay gateway se encuentra bajo el control del operador o del proveedor del servicio y es el punto por el cual pasan todos los procesos que se vayan a llevar a cabo con OSA/Parlay. Gracias al Parlay gateway las aplicaciones son independientes de los protocolos utilizados por el proveedor de la red, y las redes pueden ser modificadas sin afectar el funcionamiento de las aplicaciones

y los servicios. Dentro del Parlay gateway se encuentran los servicios y el framework

- **Servicios:** El parlay gateway esta conformado por varios SCS (Service Capability Server), los cuales son entidades funcionales que proveen interfaces OSA/Parlay a través de las aplicaciones. Cada SCS es visto por las aplicaciones como uno o varios SCF's (Service Capability Features, capacidades de servicio), que son abstracciones de las funcionalidades ofrecidas por la red e interactúan con los elementos de la red como SSP, HLR, CSE, servidores de localización entre otros. Estos SCF's reciben también el nombre de servicios y se puede acceder a ellos por medio de las API's de Parlay.

Figura 17. Service Capability Server

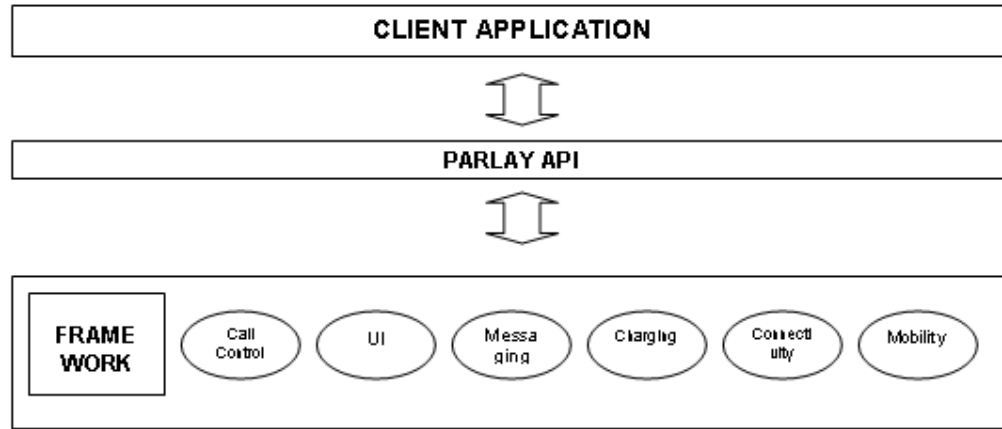


Fuente: POPOFF, Jeff. Parlay/OSA 101 [en línea]. Boston: The Parlay Group, 2005. [Consultado 23 de febrero de 2008]. Disponible en Internet: <http://www.parlay.org>

Cada SCF o varios pueden ser utilizados por una aplicación de acuerdo a la función del servicio que se vaya a implementar.

3.3.2. SCF's (Capacidades de servicio)de OSA/Parlay

Figura 18. SCF's de OSA/Parlay y framework



- **Modulo de control de llamadas:** Este modulo es el encargado de controlar las llamadas. Permite al servidor de aplicaciones establecer las llamadas y efectuar el enrutamiento en la red. Permite el establecimiento de llamada tripartita. Suministra la posibilidad de control de llamadas multimedia y llamada de conferencia.

El modulo de control de llamada se divide en:

- Servicio de control de llamadas genérico (GCCS): suministra la API de servicio de control de llamada básico. Permite originar llamadas desde la red y el encaminamiento de estas a través de la red. De igual forma permite la gestión de llamadas
- Servicio de control de llamada de múltiples abonados (MPCCS): suministra la API de control de llamadas tripartita. Sus interfaces incluyen gestión de control de llamadas de múltiples abonados e interfaz de llamada de múltiples abonados
- Servicio de control de llamada de multimedia (MMCCS): suministra la API de control para realizar llamadas multimedia. Sus interfaces incluyen gestión de control de llamadas multimedia, interfaz de llamada multimedia e interfaz de flujo de multimedia
- Servicio de control de llamadas de conferencia: este servicio realiza el control de llamadas de conferencia en donde una aplicación puede manejar subconferencias dentro de una conferencia

- **Modulo Interacción con el usuario (UI):** Este modulo posee interfaces las cuales permiten la interacción con los usuarios finales. Tiene como función el envío o recolección de información relacionada con la llamada para un usuario involucrado en la llamada. Permite el envío de mensajes cortos (SMS) hacia usuarios involucrados o no en la llamada. También permite a los usuarios colocar tonos, recibir y grabar números
 - **Modulo de mensajería:** Este modulo es utilizado para el envío, almacenamiento y recepción de mensajes. Puede procesar mensajes de voz y emails. Cada suscriptor posee una carpeta en donde puede almacenar los mensajes
 - **Modulo de tasación:** Esta SCF se utiliza para la tasación basada en el uso de las aplicaciones, permite crear y gestionar la tasación e igualmente permite manejar el ciclo de vida de las cuentas de suscriptor
 - **Modulo de gestión de conectividad:** Este modulo es utilizado para gestionar la red del operador. Contienen las API's entre el operador y el proveedor de la red las cuales permiten establecer parámetros de calidad de servicio. Estas API's proveen información como paquetes perdidos, retardos en la entrega de paquetes y características del tráfico en la red
 - **Modulo de movilidad:** Este modulo contiene el servicio de localización de usuario (UL) el cual permite proporcionar servicios de localización geográfica y el estado de los usuarios (US) de teléfono fijo, IP y abonados móviles. Por medio del servicio Camel de Localización de Usuario (ULC) proporciona información de ubicación geográfica basada en información de la red específica. Dentro de este modulo también se encuentra el servicio de Emergencia de Localización de Usuario (ULE) donde la red puede ubicar a quien realiza la llamada automáticamente y enviar el resultado a la aplicación
 - **Modulo de gestión de cuentas:** Este modulo es utilizado para implementar la notificación de tasación y solicitar la gestión de cuentas del usuario, proporcionando todos los métodos necesarios para el monitoreo de cuentas de usuario
 - **Framework:** Este modulo provee las interfaces de control y seguridad de acceso a las interfaces de servicios. También provee funciones para soportar el incremento de nuevas interfaces de servicios. Este modulo tiene como funciones también suministrar aplicaciones para el descubrimiento y autenticación de servicio, gestión de acuerdo de servicio, gestión de integridad y notificación de eventos
- Autenticación: Todas las aplicaciones deben pasar primero por la interfaz de autenticación, esto debe hacerse antes de poder utilizar otra interfaz
- Autorización: la aplicación es autorizada de acceder al SCF. Antes de este paso la aplicación debe estar previamente autenticada.

- Descubrimiento del framework y del SCF: Después de la autenticación las interfaces pueden obtener las interfaces disponibles del framework y usar la interfaz de descubrimiento para obtener información acerca del SCF solicitado
- Establecimiento de un acuerdo de servicio: Antes que cualquier aplicación pueda acceder a una SCF se debe establecer un acuerdo de servicio donde se negocian ciertos parámetros para el uso de la aplicación
- Acceso al SCF: El framework debe proveer control de acceso hacia el SCF o servicio de datos desde un método de cualquier aplicación.

Todas las aplicaciones y servicios que necesiten usar las API's de OSA/Parlay primero deben estar registrados en el framework, ya que las aplicaciones no pueden acceder directamente a los servicios.

Una de las principales desventajas de OSA/Parlay con respecto a las SCF's es que a medida que se van añadiendo nuevas capacidades de servicio con las nuevas versiones de la especificación, las nuevas funcionalidades añadidas no podrán ser utilizadas por aplicaciones que utilicen las versiones anteriores. Esto es debido a que la definición de nuevas capacidades no se hace teniendo en cuenta el refinamiento de capacidades genéricas sino que cada especificación es totalmente nueva.

3.3.3. Middleware en OSA/Parlay. OSA/Parlay separa las aplicaciones de las capacidades de servicio por medio de interfaces abiertos para ello necesita de una tecnología middleware y OSA/Parlay cuenta con dos opciones CORBA y servicios Web.

CORBA fue la primera tecnología adoptada por OSA/Parlay por su fiabilidad y prestaciones en las redes de telecomunicaciones.

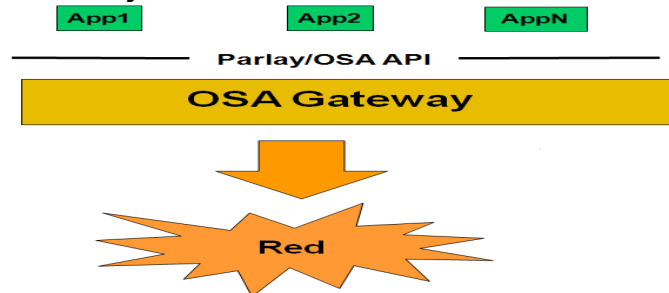
Luego, en el 2002 aparece otra alternativa. Los servicios web, que permiten la realización de una red de servicios interoperable que se centra en la reutilización de dichos servicios, como se explico anteriormente los servicios web se basan en HTTP para invocaciones remotas, utiliza WSDL para la definición de interfaces y el protocolo SOAP para el transporte de información, la única diferencia es que en OSA/Parlay no necesita el uso de UDDI ya que la misma plataforma se encarga de realizar las funciones de registro y búsqueda.

3.3.4. API's de OSA/Parlay. Las especificaciones de OSA definen una arquitectura que permite a los desarrolladores de aplicaciones de servicios hacer uso de las funcionalidades de la red a través de las interfaces abiertas estandarizadas (OSA API's). Esto quiere decir que los servicios desarrollados por terceras partes que corren en servidores de aplicaciones externos, donde las aplicaciones son independientes de la red pueden ser ofrecidos por los operadores a través de un Gateway. Para proveer mayor seguridad y control

en cuanto al acceso de las capacidades de la red del proveedor de servicios, el Parlay Gateway tiene implementado el framework.

El API de Parlay sirve como interfaz entre el parlay Gateway y el servidor de aplicaciones.

Figura 19. OSA/Parlay API



Fuente: CAICEDO RENDON, Oscar Mauricio. Parlay/OSA [en línea]. Popayán: Universidad del Cauca, 2003. [Consultado 08 de febrero de 2008]. Disponible en Internet: <http://wapcolombia.unicauca.edu.co/documentos/osaparlay.ppt>

El propósito de las API's de OSA/Parlay es ocultar la complejidad de la red, sus protocolos e implementación específica desde el punto de vista de las aplicaciones. Esto significa que las aplicaciones no deben tener en cuenta la infraestructura de la red ya que el SCS interactúa directamente con la aplicación con el fin de proveer las SCF's necesarias.

El uso de APIs estandarizados basados en tecnología abierta. Aporta las siguientes ventajas:

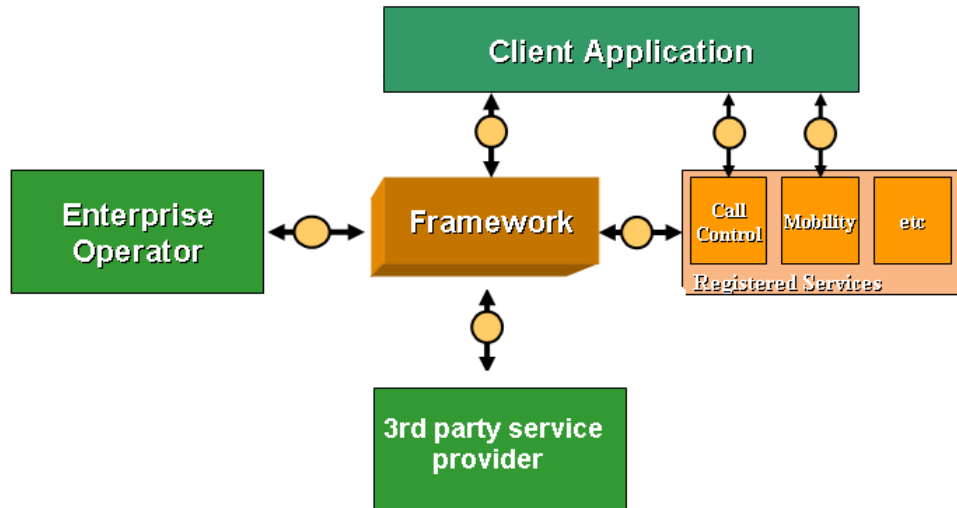
- Acorta el tiempo de puesta en el mercado de nuevas soluciones por el nivel de abstracción y el uso de tecnología abierta
- Las Aplicaciones pueden ser desarrolladas por terceras partes lo que implica una ventaja al poder contar con la creatividad, e ideas de servicios nuevos e innovares que puedan proponer otras personas
- Las aplicaciones pueden ser independientes de las redes (multiacceso/multiservicio)
- Elimina la necesidad que los programadores aprendan muchos protocolos de telecomunicaciones, disminuyendo costos
- Permite la creación de aplicaciones que pueden funcionar a través de múltiples redes

Existen una serie de reglas generales que fueron seguidas por el diseño de las API's de OSA/Parlay que deben ser usadas y son definidas a continuación.

3.3.5. Definición de los tipos de interfaces.

- Interfaces del framework hacia las aplicaciones: Son utilizadas para iniciar sesiones de acceso a los servicios, por medio de mecanismos básicos como autenticación, autorización, descubrimiento, selección del servicio y acceso al servicio permiten hacer uso de los SCF's. Estas interfaces se encuentran en el framework y son usadas por la aplicación. Pueden ser identificadas por medio del prefijo "Ip<name>"
- Interfaces de callback que corren del lado de la aplicación hacia el framework: Estas interfaces necesitan ser implementadas por una aplicación (cliente). Los nombres de estas interfaces tienen el prefijo "IpApp<name>"
- Interfaces del framework que son utilizadas por los servicios: por medio de estas interfaces los servicios pueden ser registrados en el framework. El prefijo utilizado para identificar estas interfaces es "IpFw<name>"
- Interfaces de un servicio que son utilizados por el framework: El nombre de este tipo de interfaz de servicio tiene el prefijo "IPSvc<name>"
- Interfaces del lado de las aplicaciones y de los SCF's que son compartidas: estas últimas pueden ser servicios individuales los cuales son solicitados por el cliente permitiendo la ejecución de aplicaciones desarrolladas por terceros a través de la interfaz. El nombre de este tipo de interfaz tiene el prefijo "IpClient"
- Interfaz entre el framework y el operador: Provee mecanismo básicos que permiten administrar la aplicación del cliente, por medio de esta interfaz los operadores pueden suscribir los servicios
- Interfaz entre el framework y los proveedores de servicios por terceras partes: esta interfaz permite que estos proveedores proporcionen servicios Parlay

Figura 20. Interfaces OSA/Parlay



Fuente: CAICEDO RENDON, Oscar Mauricio. Parlay/OSA [en línea]. Popayán: Universidad del Cauca, 2003. [Consultado 08 de febrero de 2008]. Disponible en Internet: <http://wapcolombia.unicauca.edu.co/documentos/osaparlay.ppt>

3.3.6. Definición de interfaces entre la aplicación y el framework. Estas interfaces son necesarias para las aplicaciones para obtener el acceso a los servicios, por medio de los procesos de autenticación, autorización, descubrimiento del servicio. Algunas de estas interfaces se describen a continuación permitiendo conocer en que parte del proceso de autenticación se utilizan y qué función cumplen.

- **IpClientAPILevelAuthentication:** Es la interfaz utilizada para iniciar el proceso de autenticación entre la aplicación y el framework. Contiene los métodos: `authenticate()`: utilizado por el framework para autenticar al cliente, `abortuthentication()`: usado como su nombre lo indica para abortar el proceso de autenticación entre el framework y la aplicación y `authenticationsucceded()`: utilizado por el framework para informar al cliente que se llevo a cabo el proceso de autenticación
- **IpClientAccess:** Esta interfaz es ofrecida por el cliente al framework para permitirles interactuar durante la sesión de acceso. Contiene el método `terminateAccess()`, utilizado para terminar la sesión de acceso entre el cliente y el framework. Por consiguiente el proceso de autenticación no es exitoso
- **IpAppServiceAgreementManagement:** Esta interfaz es utilizada por el framework para establecer una acuerdo de servicio con la aplicación
- **IpAppFaultManager:** Esta interfaz es utilizada para informar a la aplicación de eventos que afectan la integridad del framework, del servicio o de la aplicación del cliente

- IpAppHeartBeatMgmt: Esta interfaz permite iniciar un pulso para supervisar la aplicación del cliente desde el framework
- IpAppHeartBeat: Esta interfaz es lo contrario de la anterior el pulso es enviado desde el framework a la aplicación
- IpAppLoadManager: Permite manejar las solicitudes, reportes y respuestas desde el framework como estadísticas, notificaciones entre otros
- IpAppOAM: La interfaz de aplicación del cliente es usada por el framework para consultar e intercambiar con la aplicación datos como fecha y hora para propósitos de sincronización
- IpAppEventNotification: Esta interfaz es utilizada por los servicios para informar a la aplicación de eventos relacionados con estos.

3.3.7. Definición de interfaces entre la aplicación y las SCF's.

- Para GCC (Generic Call Control) se utilizan las interfaces IpAppCallControlManager, que provee funciones para el manejo del servicio GCC. El programador puede utilizar esta interfaz para funciones de control, crear objetos y permitir o no la notificación de eventos y la interfaz IpAppCall, utilizada por el desarrollador para manipular solicitudes de llamadas, respuesta y reportes de estado
- Para MPCC (MultiParty Call Control) se utilizan dos interfaces la primera IpAppMultiPartyCallControlManager, es una interfaz del lado de la aplicación que le permite administrar las funciones del servicio MPCC y la segunda interfaz es IpAppMultiMediaCall, la cual es una interfaz del lado de la aplicación que permite manipular eventos y reportes del servicio MPCC
- Para tasación se utilizan las interfaces IpAppChargingManager, es una interfaz del lado de la aplicación que permite manejar las funciones del servicio de tasación y la interfaz IpAppCahrgingSession, la cual es una interfaz de aplicación que debe ser implementada por la aplicación del cliente para manipular las llamadas por medio de sesiones de tasación
- Para Interacción con el usuario UI se utilizan las interfaces IpAppUIManager, que provee las funciones necesarias para el manejo del servicio de interacción con el usuario UI, indicando a la aplicación cuando una instancia ha terminado o notificando eventos y la interfaz IpAppUI, implementada por el desarrollador de la aplicación del cliente y es utilizada para manipular solicitudes, respuestas y reportes del servicio UI.

3.3.8. Uso de sesiones. Una sesión es una serie de interacciones entre dos puntos de comunicación que ocurren durante el lapso de la conexión. Un ejemplo de estas interacciones son todas las operaciones para crear, controlar y terminar una llamada. Una sesión se identifica por medio de un ID que es único dentro de las instancias del servicio y puede estar relacionado con números de sesiones utilizados en la red.

3.3.9. Interfaces y sesiones. Algunas interfaces tienen una relación uno a uno con una sesión. Para cada sesión hay una interfaz. Todos los métodos invocados en una interfaz operan sobre la misma sesión. Estas interfaces no utilizan ID's para las sesiones.

Otras interfaces pueden representar múltiples sesiones. La aplicación puede crear una instancia por sesión o puede crear varias sesiones por instancia (para disminuir el uso de recursos). Cuando un método de una interfaz es invocado la sesión requiere un ID para identificar la sesión para la cual aplica.

3.3.10. Interfaces de callback. Algunas interfaces de OSA/Parlay requieren de una aplicación para registrar una interfaz de callback. Esta interfaz reside sobre la aplicación del cliente y es usada por el servidor de servicios para reportar eventos, resultados y errores. Una interfaz debe registrar su interfaz de callback tan pronto como la interfaz del servidor es creada.

3.3.11. Métodos síncronos y asíncronos. Dos tipos de métodos existen en las interfaces OSA/Parlay. Cuando un método no requiere el SCS para contactar otros nodos en la red es implementado como un método síncrono. Cuando el método retorna el resultado de la operación, es suministrado a la aplicación. Cuando ocurre un error una excepción es lanzada. Algunos ejemplos de métodos síncronos, son métodos para recuperar datos que están disponibles en el SCS y métodos que crean un objeto.

En otros casos cuando un método necesita al SCS para contactar otros nodos en la red. Puede presentarse un retraso entre el momento en que el mensaje es enviado hasta que la respuesta es recibida o un error es detectado. Para prevenir que la aplicación sea bloqueada o que la aplicación tenga que adivinar donde hay un problema en el SCS se utilizan los métodos asíncronos.

Un método asíncrono de una interfaz puede ser reconocido ya que su nombre termina con "Req" (de petición), y en la interfaz de retorno incluye dos métodos el "Res" (de resultado), y el "Err" (de error). Cuando no hay errores el método "Res" es invocado cuando el resultado esta disponible, y en el caso contrario es decir cuando se presentan errores el método "Err" es invocado y una excepción es lanzada permitiendo la detección de fallos en el SCS.

3.3.12. Parámetros de salida. Los métodos utilizados en las interfaces de OSA/Parlay solo tienen parámetros de entrada. Un resultado solo puede ser reportado por un valor de retorno. Si se necesita que múltiples valores sean retornados se requiere un tipo de dato que consista en una secuencia de valores. Un valor de este tipo de dato es retornado por un método. Este procedimiento se hace de esta forma ya que muchos middleware no son capaces de manejar múltiples parámetros de salida.

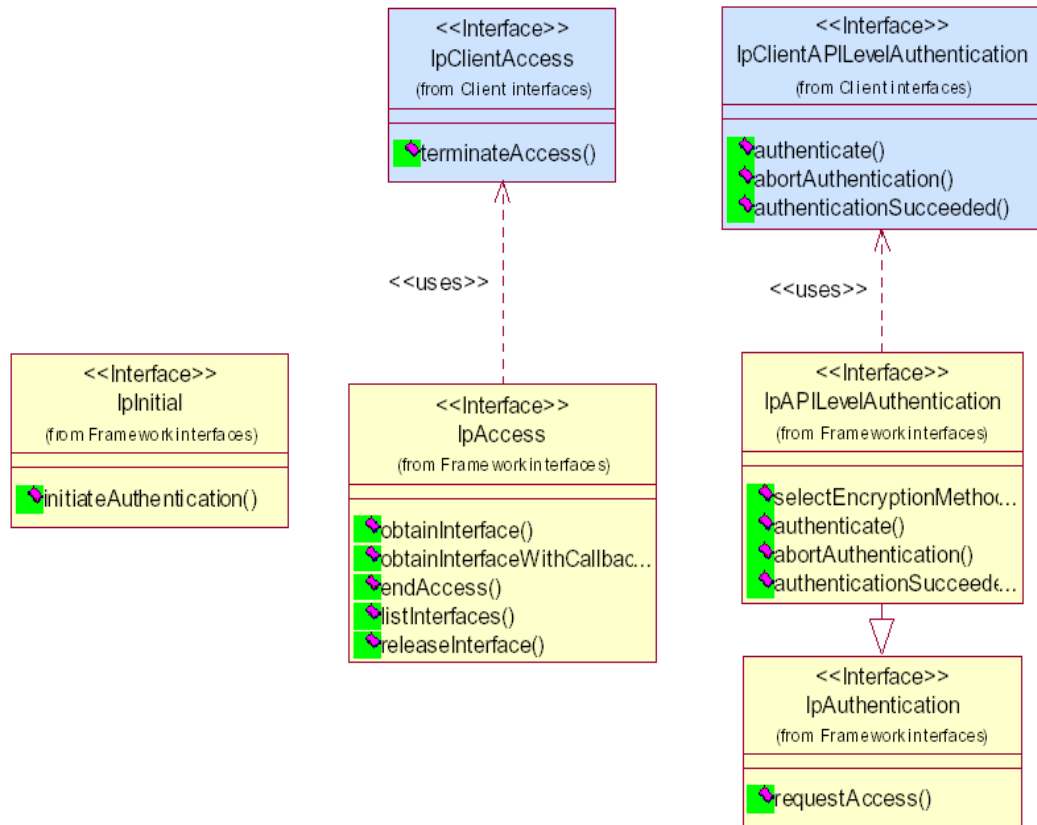
Con respecto a la arquitectura P1109 OSA/Parlay ha hecho un gran esfuerzo por estandarizar las interfaces entre el nivel del servidor de aplicaciones y todos los niveles subyacentes (Call Server, media Server, messaging Server). Actualmente se están desarrollando rápidamente servidores especializados para aplicaciones OSA/Parlay y su difusión ha sido exitosa.

Las API's de OSA/Parlay proveen un nivel medio de abstracción. Muchos de los productos de OSA/Parlay están disponibles, como Parlay gateway, framework, servidores de aplicaciones, muchas de estas ofertadas a redes móviles. Por las facilidades que ofrece OSA/Parlay actualmente muchas empresas están desarrollando nuevos servicios utilizando esta tecnología.

3.4. MECANISMO DE ACCESO A UN SERVICIO DESDE UNA APLICACIÓN

Todo el proceso de registro de las aplicaciones en el framework para utilizar los servicios es igual para cualquier tipo de aplicación.

Figura 21. Diagrama de clases acceso al framework desde la aplicación



Fuente: ZTE. UP10 Parlay API specification. Santiago de Cali, 2005. 1 archivo de computador.

El proceso por el cual la aplicación del cliente tiene acceso al cliente ha sido dividido en tres etapas, cada una soportada por un interfaz diferente del framework

- Contacto inicial con el framework
- Autenticación con el framework
- Acceso al framework y a las SCF's

Para iniciar el contacto inicial con el framework, como se muestra en la fig. se utiliza la interfaz IpInitial que es la interfaz inicial del framework usada por el cliente para comenzar la autenticación, por medio del método initiateAuthentication(), invocado por el cliente el cual pregunta por el uso de un método de autenticación específico.

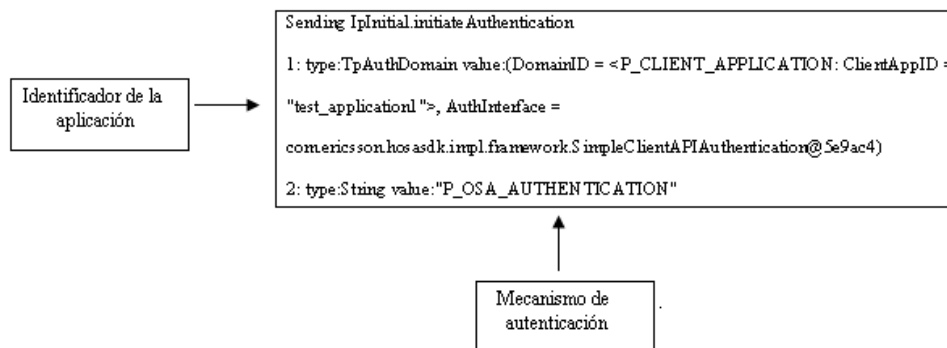
Dentro de este método se definen dos parámetros uno para identificar el dominio del cliente con el framework (domainID) y otro para proveer una referencia para la interfaz de autenticación del framework (authInterface).

El parámetro domainID puede ser un identificador cualquiera para la aplicación del cliente por ejemplo TPClientAppID, o para un operador se utiliza TpEntOpID, para un servicio existente registrado se utiliza TpServiceId o para un proveedor de servicio se usa TpServiceSupplierID. Si el framework no reconoce el domainID, retorna un código de error P_INVALID_DOMAIN_ID.

El parámetro authInterface define el tipo de mecanismo de autenticación solicitado por el cliente. La autenticación del nivel OSA API es el mecanismo de autenticación por defecto (P_OSA_AUTHENTICATION).

Al iniciar una aplicación el primer paso es acceder al framework realizando el procedimiento que se muestra en la siguiente figura.

Figura 22. Acceso al framework desde la aplicación



Después de acceder al framework se inicia el proceso de autenticación por medio de la interfaz IpAPILevelAuthentication, que es una interfaz implementada por el framework usada por el cliente para realizar su parte dentro del proceso de autenticación. Este proceso es necesario para utilizar cualquier interfaz soportada por el framework.

Dentro de esta interfaz se encuentran los métodos necesarios para llevar a cabo totalmente el proceso de autenticación.

Se inicia el proceso de autenticación donde el cliente primero invoca el método selectEncryptionMethod() que se encuentra en el framework, el cual permite enviar un algoritmo de autenticación basado en CHAP. El framework decide el método para encriptar que se va a utilizar.

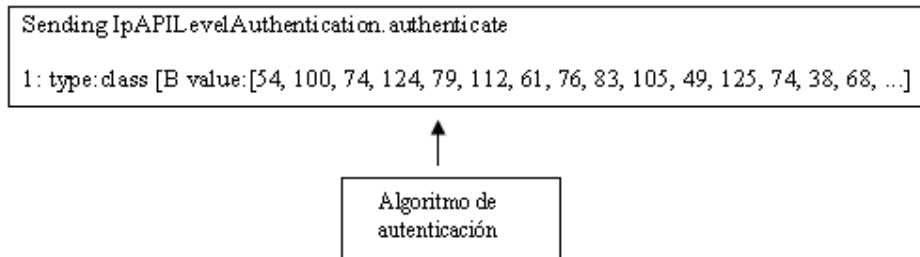
Figura 23. selectEncryptionMethod()

```

Sending IpAPILevelAuthentication.selectEncryptionMethod
1: type:String value:"P_NONE"
  
```

Después de seleccionar el método para encriptar, la aplicación se autentica con el framework por medio del método `authenticate()`. La información de autenticación es encriptada usando el mecanismo seleccionado en el método `selectEncryptionMethod()`.

Figura 24. Método `authenticate()`



A través del método `authenticationSucceeded()`, el cliente informa al framework que ha terminado el intento de autenticación.

Figura 25. Método `authenticationSucceeded()`

Sending IpAPILevelAuthentication.authenticationSucceeded

El framework autentica el cliente por medio del método `authenticate()`, dentro de la interfaz `IpClientAPILevelAuthentication` que se encuentra en el cliente, en donde responde al algoritmo de autenticación.

Figura 26. Método `authenticate()` del lado del cliente

Received IpClientAPILevelAuthentication.authenticate
1: type: class [B value: [21, -83, -9, -73, 39, -118, -110, -94, -120, -17, 35, -92, 42, -107, -81, ...]]

Después el framework informa al cliente que la autenticación ha terminado utilizando el método `authenticationSucceeded()`.

Figura 27. Autenticación exitosa

Received SimpleClientAPIAuthentication.authenticationSucceeded

Una vez el cliente ha sido autenticado por el framework, el cliente puede invocar el método `requestAccess()`, en la interfaz `IPAuthentication` o `IpAPILevelAuthentication`. Este método le permite al cliente solicitar el tipo de

interfaz de acceso que requiere del framework. Si la solicitud es P_OSA_ACCESS entonces una referencia es devuelta a la interfaz IpAccess. Los operadores pueden definir sus propias interfaces de acceso para satisfacer los requerimientos del cliente para diferentes tipos de acceso.

Figura 28. Método requestAccess()

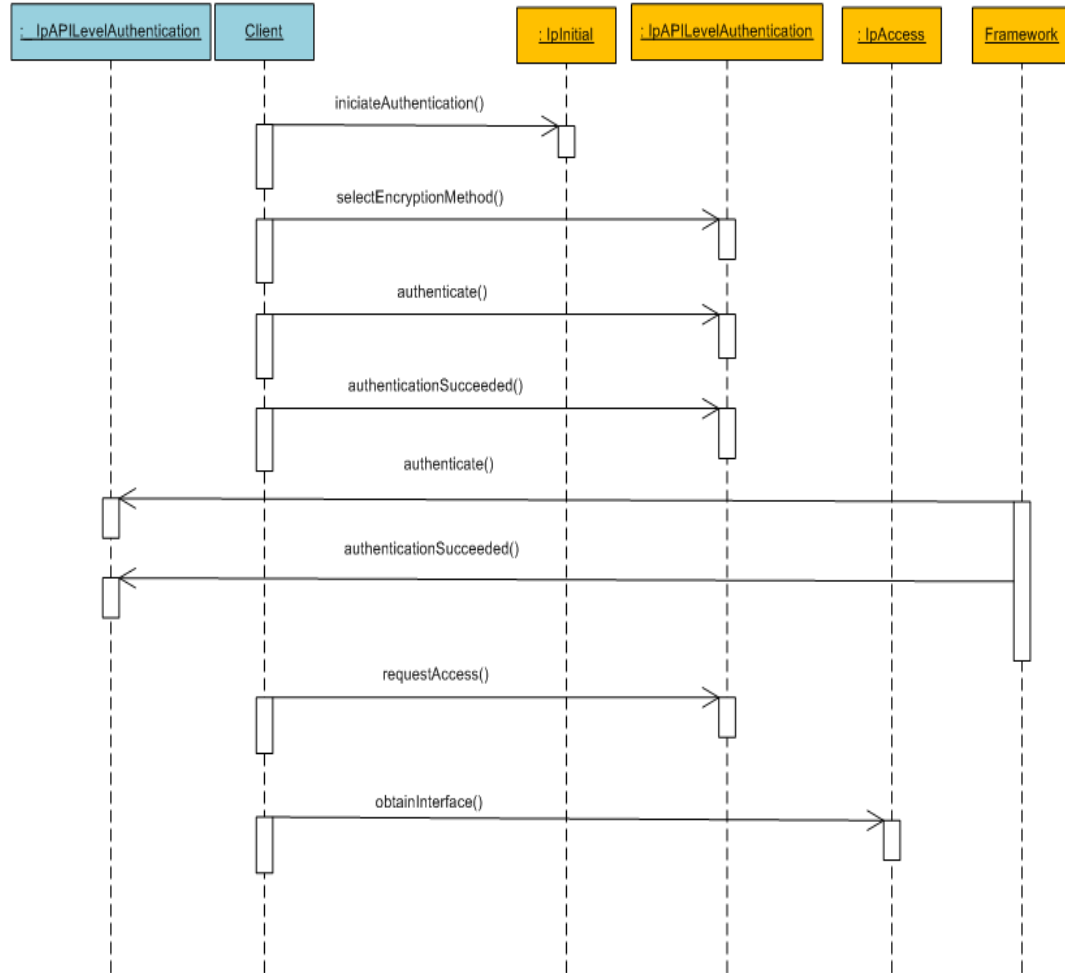
```
Sending IpAPILevelAuthentication.requestAccess  
1: type:String value:"P_OSA_ACCESS"
```

El cliente invoca el método obtainInterface(), en la interfaz Ipaccess en el framework, para obtener una interfaz de descubrimiento del servicio.

Figura 29. Método obtainInterface()

```
Sending IpAccess.obtainInterface  
1: type:String value:"P_DISCOVERY"
```

Figura 30. Diagrama de secuencia para autenticación con el framework



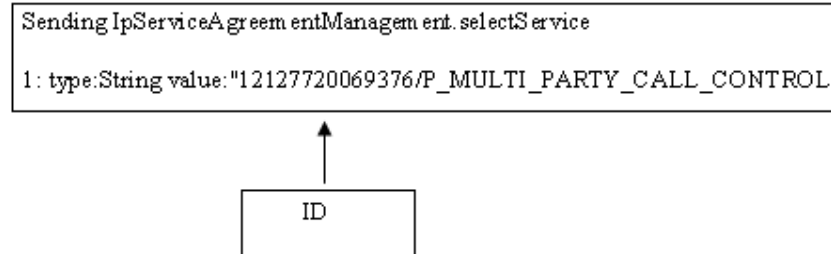
En este punto finaliza la autenticación con el framework, el paso siguiente es acceder a los SCF's. Este proceso se inicia por medio de la interfaz IpServiceDiscovery, en donde como su nombre lo indica es la interfaz de descubrimiento del servicio. Dentro de esta interfaz se encuentra el método discoveryService(), por el cual la aplicación del cliente describe las propiedades del servicio que está buscando.

Figura 31. Método discoveryService()



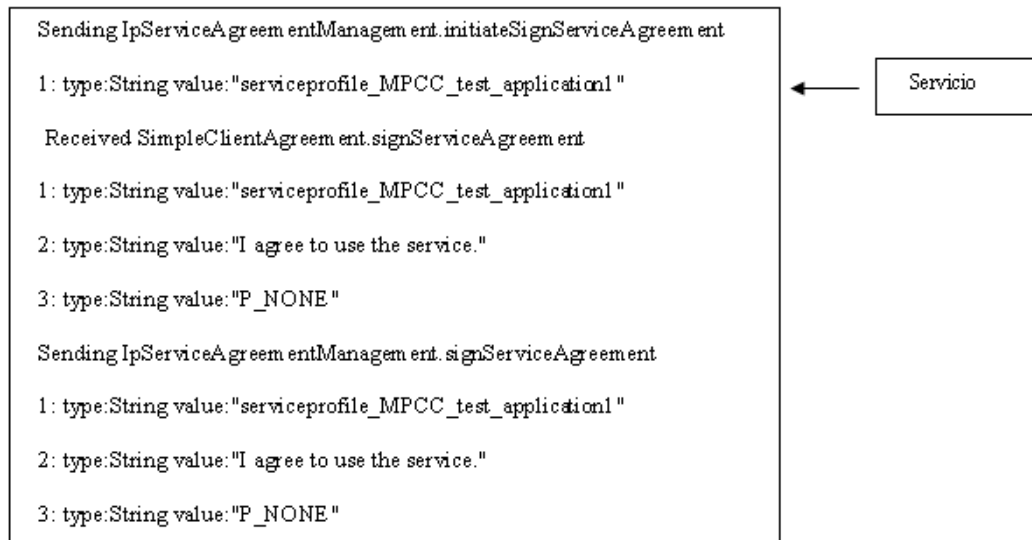
Para seleccionar el servicio que la aplicación necesita es necesario utilizar la interfaz IpServiceAgreementManagement en donde por medio del método selectService(), la aplicación identifica el servicio, por medio de un ID.

Figura 32. Selección del servicio



El cliente inicia la búsqueda del servicio por medio del método initiateSignServiceAgreement(), dentro de la interfaz IPServiceAgreementManagement, con el nombre de la aplicación, y por último el framework autoriza a la aplicación del cliente a utilizar el servicio.

Figura 33. Autorización para utilizar el servicio



Nota: El proceso de autenticación aquí mostrado se realizó utilizando el simulador de Ericsson NGR (Network Resource Gateway), teniendo en cuenta que este proceso es igual para cualquier plataforma donde sea ejecutado.

En la siguiente tabla se hace la comparación entre todas las tecnologías para el desarrollo de servicios mencionadas anteriormente teniendo en cuenta los criterios empleados para su análisis.

Tabla 1. Cuadro de comparación de tecnologías

TECNOLOGÍAS	CAPACIDADES DE LA RED	EVALUACION DE INTERFACES			PROGRAMABILIDAD POR TERCERAS PARTES	FACIL USO	INDUSTRIA
		NIVEL DE ABSTRACCION	TIPO DE INTERFACE	LENGUAJE DE DESCRIPCION DE INTERFACES			
SERVICIOS WEB	No están definidas	Nivel abstracto	XML	WSDL	Aplica	Aplica	No se ha establecido totalmente
SIP SERVLETS	CC	Nivel Bajo	Java	---	Aplica	Aplica	La primera versión en revisión
JAIN SIP LITE	CC	Nivel Alto	Java	---	Aplica	Aplica	Se encuentra en revisión
VOICE XML	No están definidas	Nivel Alto	XML	DTD	Aplica	Aplica	Funcionalidad limitada (voz).
CCXML	CC	Nivel Alto	XML Script	DTD	Aplica	Aplica	Complementa a Voice XML (conferencias multimedia)
JCC	GCC UI	Nivel Bajo	Java	---	Aplica		Redes PSTN, ATM, IP
CPL	CC	Nivel Alto	XML	DTD	Aplica	Aplica	permite crear servicios personalizados
SCML	UI, MMCC, GCC	Nivel Alto	XML (java)	XML con DTD	Aplica	Aplica	No tiene buen mecanismo de seguridad
XTML	No están definidas	Nivel Alto	XTML	---	Limitado	Aplica	Propietario
OSA/PARLAY	FW, CC, UI, MSS, etc.	Nivel Bajo	Java C++	IDL, WSDL	Aplica	Aplica	Tecnología ya establecida

4. CONSIDERACIONES PARA EL DESARROLLO E IMPLEMENTACIÓN DE NUEVOS SERVICIOS SOBRE LA PLATAFORMA MULTISERVICIOS DE EMCALI

Este proyecto busca definir el ambiente de desarrollo para la implementación de nuevos servicios sobre la red multiservicios de EMCALI, en donde es la principal tarea es ubicar al desarrollador de nuevas aplicaciones en el entorno con el que va a interactuar, como acceder a los servicios (SCF's), su ubicación, que cambios o configuraciones debe hacer para montarlos y que herramienta puede utilizar para el desarrollo de servicios.

En este capítulo también se presenta al desarrollador la estructura física de la plataforma de servicios y todos los ítems a tener en cuenta para desarrollar las aplicaciones e implementarlas en la plataforma. Empezamos entonces por definir la estructura física del parlay gateway y el sistema operativo utilizado para acceder a él.

4.1. SISTEMA OPERATIVO AIX

Todos los servicios (SCF's), se encuentran dentro del parlay gateway como ya ha sido mencionado anteriormente, y es necesario conocer bajo que entorno trabaja, y como el desarrollador puede interactuar con el ya que el parlay gateway es el núcleo de la plataforma de servicios y en él se encuentra la información que toda aplicación necesita para ser montada en la red con éxito.

El parlay gateway trabaja bajo el sistema operativo AIX, la versión utilizada en la plataforma de EMCALI es la 5.2.y el acceso a el se hace por medio de este sistema operativo.

AIX (Advanced Interactive eXecutive) es un sistema operativo Linux abierto desarrollado por IBM, que permite ejecutar las aplicaciones deseadas en cualquier hardware y servidor IBM Linux. Utiliza procesadores de 32 y 64 bits. Dispone de un excelente asistente para realizar todo tipo de tareas denominado SMIT (System Management Interface), aunque las tareas también pueden ser ejecutadas desde la línea de comando, generalmente no se hace con las ordenes convencionales de UNIX por lo cual SMIT es una herramienta importante para el administrador.

AIX es un entorno de trabajo fiable, muy estable y tiene un ODM (Object Data Manager, base de datos de información del sistema). Es un sistema operativo totalmente funcional, brinda a los usuarios un ambiente multitarea y multiusuario que permite la ejecución de todo tipo de aplicaciones. Entre sus

principales ventajas esta el uso de menús para las funciones de administración del sistema y el soporte a volúmenes lógicos.

El sistema AIX es de arquitectura abierta y sobresale por contar con una alta tecnología y una extraordinaria relación precio/rendimiento. Esto gracias a que utiliza la arquitectura POWER en donde se optimiza el desempeño del sistema mediante el uso de la tecnología RISC, que implica que en menos de un ciclo de reloj se pueda ejecutar una instrucción que pertenezca a un conjunto reducido y optimizado de instrucciones.

Una de las principales características de AIX es la interoperabilidad pues ofrece varias alternativas de conectividad dentro de los sistemas abiertos, desde redes locales LAN (token_ring y ethernet), redes WAN y puede soportar dispositivos periféricos. AIX cuenta con la capacidad de interoperar con sistema no-IBM en ambientes de red mediante el uso de diferentes dispositivos periféricos.

Este sistema operativo es escalable ya que esta diseñado para soportar los cambios que sean necesarios lo que le permite tener un alto poder de procesamiento. Puede ser utilizado con tecnologías y plataformas hardware diferentes ya que utiliza compatibilidad binaria.

AIX permite manejar muchos datos y muchos usuarios eficientemente. Permite también adicionar nuevos dispositivos y mayor espacio de memoria cuando sea necesario, permite realizar copias de seguridad de los datos teniendo un mayor control de los sistemas.

4.2. CLUSTER HACMP

Dentro de la plataforma multiservicios de EMCALI, el parlay gateway también posee redundancia tanto a nivel de hardware como de software. Con el fin de brindar mayor rendimiento y disponibilidad en el procesamiento de información, la plataforma de EMCALI cuenta con el cluster HACMP (High Availability Cluster Multi-Processing) implementado para los dos parlay gateway con los que cuenta la plataforma.

La característica alta disponibilidad (High Availability), garantiza la provisión continua del servicio para las aplicaciones de los clientes. Esto significa que cualquier falla que se presente en algún componente sea hardware, software o del sistema de gestión no cause daños en la disponibilidad de la aplicación y los datos que contiene hacia el usuario final. Todos los componente hardware se encuentran duplicados, es decir, tienen un respaldo teniendo equipos de características iguales o similares controlados por un componente software especializado.

HACMP también provee un componente multi-proceso. Esta característica viene del hecho que un cluster esta formado por múltiples recursos hardware y software y HACMP se encarga de manejarlos para proveer mayor funcionalidad y mejor utilización de los recursos. En un cluster multi-proceso corren múltiples aplicaciones a través de un número de nodos que comparten la información.

El cluster HACMP con que cuenta la plataforma de EMCALI fue desarrollado por IBM y actualmente se trabaja con la versión 5.2, en donde todas las aplicaciones están bajo el control de HACMP asegurando la disponibilidad de todos los procesos de los clientes así un componente falle, moviendo las aplicaciones que están siendo utilizados en ese momento, a otro nodo en el Cluster.

HACMP trabaja de la mano con el sistema operativo AIX y dependiendo de la versión de este último debe ser instalada la versión del cluster. HACMP toma ventajas de las características de AIX ya que este incluye una buena disponibilidad de los datos en aplicaciones críticas. HACMP provee escalabilidad tanto horizontal como vertical.

Entre las características de AIX que mejoran la disponibilidad del sistema se encuentran:

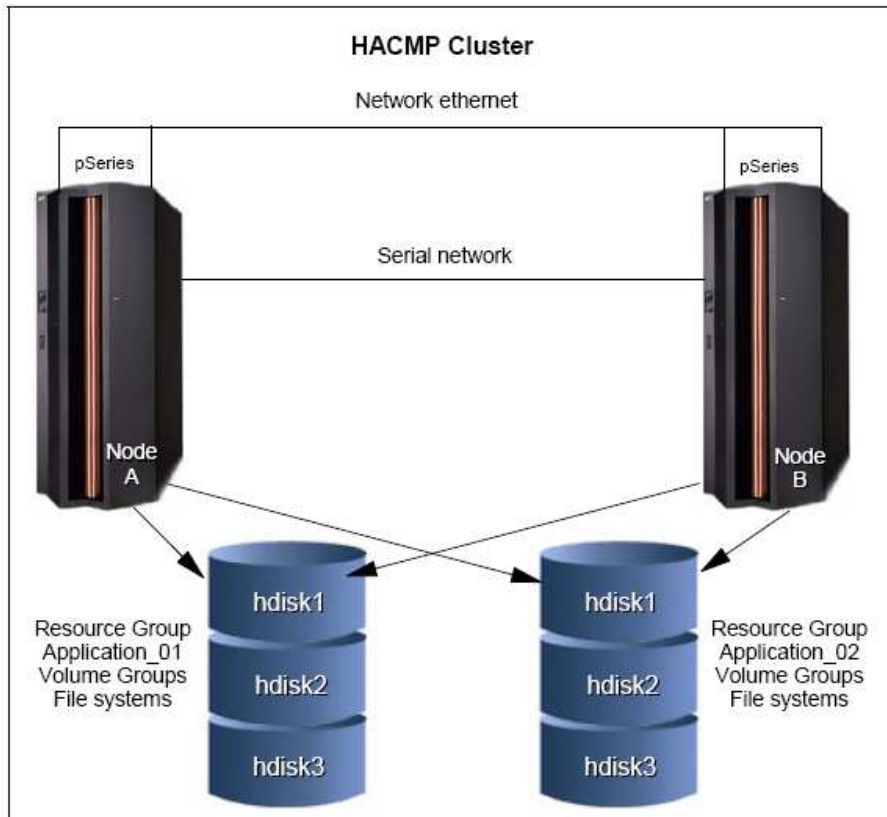
- AIX mantiene la integridad estructural de los archivos del sistema usando log's que guardan la información y en el caso que el sistema se reinicie los archivos del sistema inician en su ultimo estado por medio del Journaled Failed System (JFS)
- Previene la perdida de datos ya que mantiene hasta tres copias de datos guardadas en discos diferentes para que se pueda acceder a ellos en caso de que alguno de los discos falle
- El sistema de control de recursos de AIX (SRC), monitorea y controla todos los procesos, el SRC puede detectar cuando un proceso termina de forma incorrecta, pasa mensajes a un programa de notificación y reestablece un porceso desde un backup
- AIX permite la detección y notificación de errores, en donde los métodos para la notificación de errores se configuran en HACMP, permitiéndole reaccionar ante los fallos y recuperarse

Es decir el sistema operativo AIX le permite a HACMP un control sobre los datos y procesos que se estén llevando a cabo en un momento determinado, garantizado un buen funcionamiento y la no perdida de información ante cualquier fallo.

4.2.1. Componentes físicos de un cluster HACMP. El cluster HACMP provee un entorno de alta disponibilidad para proveer un conjunto de recursos esenciales para un proceso no interrumpido. Dentro de los componentes físicos de un cluster HACMP se encuentran:

- **Nodos:** Forman parte del núcleo del cluster HACMP. Un nodo es un proceso en donde corren AIX, el software HACMP y el software de la aplicación. Los nodos son identificados por un único nombre y pueden tener su propio conjunto de recursos, archivos de sistema, redes, direcciones de redes y aplicaciones. El software HACMP soporta desde 2 a 32 nodos. Un nodo en el cluster HACMP tiene diferentes capas de componentes de software los cuales serán explicados mas adelante
- **Dispositivo de disco externo compartido:** Cada nodo tiene uno o mas dispositivos de disco externo compartido, estos son discos físicamente conectados a múltiples nodos. Cada nodo en el cluster HACMP tiene un disco interno donde almacena el sistema operativo, pero estos discos no son compartidos
- **Redes:** El software HACMP esta diseñado para trabajar con co redes basadas en TCP/IP. Los nodos en un cluster HACMP usan las redes para permitir a los clientes acceder a los nodos de los cluster's y para acceder de forma serial a los datos
- **Interfaces:** El cluster HACMP define dos tipos de redes de comunicación, uno basado en interfaces de comunicación basadas en TCP/IP en donde se pueden conectar dos o más nodos servidores y opcionalmente permite a los clientes acceder a los nodos del cluster usando el protocolo TCP/IP. El otro tipo es por medio de dispositivos de comunicación sin utilizar el protocolo TCP/IP, estos dispositivos proveen una conexión punto a punto entre dos nodos de un cluster para el control de mensajes y del tráfico
- **Cientes:** Es un equipo que puede acceder a los nodos en el cluster a través de una red LAN. Los clientes ejecutan las aplicaciones que se encuentran corriendo en el servidor de aplicaciones en un nodo del cluster. HACMP tiene una herramienta que le permite obtener el estado del cluster y se encuentra en `/usr/es/bin/cluster/clstat`. Esta herramienta reporta el estado de los componentes del cluster los nodos, las interfaces y los recursos de cada nodo.

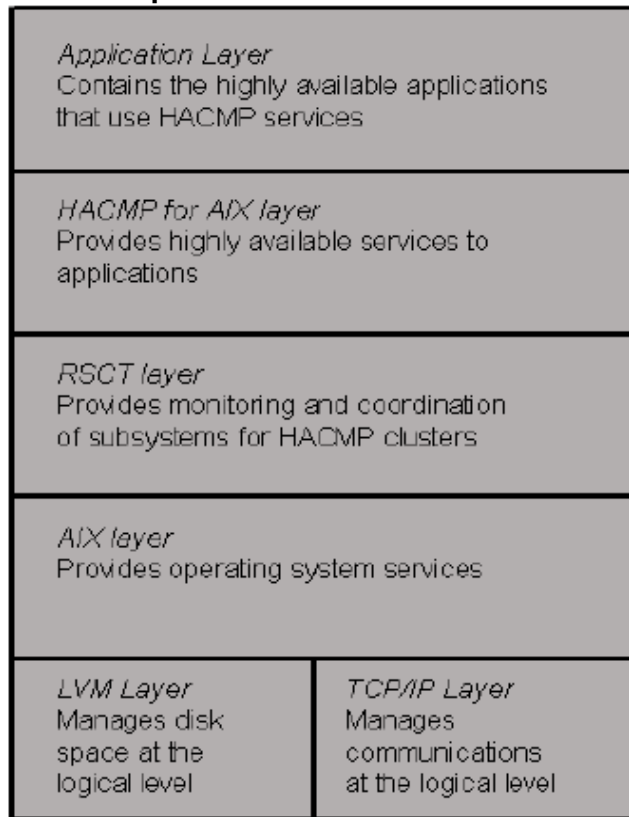
Figura 34. Cluster HACMP



Fuente: PRELEC, Dussan; RAYMON Ken. Implementing High Availability Cluster Multi-Processing (HACMP) cookbook [en línea]. Estados Unidos: International Business Machines Corporation, 2005. [Consultado 17 de junio de 2008]. Disponible en internet: http://www-03.ibm.com/systems/p/support/techdocs/clusters_aix.html

4.2.2. Componentes software de un nodo HACMP. Los componentes software dentro de un nodo en un cluster HACMP están divididos en capas como se muestra en la siguiente figura

Figura 35. Modelo en capas de un nodo en un cluster HACMP



Fuente: High Availability Cluster Multi-Processing for AIX Concepts and facilities guide [en línea]. Estados Unidos: International Business Machines Corporation, 2004. [Consultado 17 de junio de 2008]. Disponible en internet: <http://publib.boulder.ibm.com/infocenter/clresctr/vxrx/index.jsp?topic=/com.ibm.cluster.hacmp.doc/hacmpbooks.html>

- **Capa de aplicación:** Esta capa contiene los códigos de las aplicaciones (programas, demonios, etc), los datos de configuración de las aplicaciones. Cualquier aplicación tiene alta disponibilidad a través de los servicios que provee HACMP para AIX
- **Capa de HACMP para AIX:** Es la capa que contiene el software que reconoce los cambios dentro de un cluster y coordina el uso de las características de AIX para crear entornos de alta disponibilidad. Dentro de esta capa se encuentra el código HACMP (librerías, comandos, scripts, etc.) y los archivos de registro (log's) HACMP
- **Capa RSCT (IBM Reliable Scalable Cluster Technology):** El RSCT permite monitorear los miembros de un nodo, las interfaces de la red y las interfaces de comunicación. Monitorea notificaciones de eventos, sincronización y coordinación de mensajes. Dentro de esta capa se encuentra el código RSCT, los archivos de configuración y la topología.

- **Capa AIX:** Dentro de esta capa se encuentra el software que provee el soporte para el cluster HACMP, incluyendo la capa del gestor de volúmenes lógicos (LVM), la cual maneja los datos almacenados en el nivel lógico (discos) y la capa TCP/IP la cual provee el soporte de las comunicaciones para un cluster HACMP.

Dentro de los componentes software de un cluster HACMP se encuentran:

- **Gestor del cluster:** El cual es un demonio que corre en cada nodo del cluster y es el que responde a los eventos no planeados que pueden estar representados por fallos o eventos iniciados por los usuarios
- **Subsistema de comunicación segura en el cluster:** El cluster utiliza el demonio de comunicación en el cluster que corre sobre cada nodo para la comunicación entre ellos para garantizar una comunicación fiable
- **RSCT:** Monitorea el estado de los dispositivos de la red
- **Cluster SMUX Peer y Programas de monitoreo SNMP:** Un cluster HACMP es dinámico y puede someterse a varias transiciones en su estado a los largo del tiempo, cada cambio puede afectar la composición del cluster sobre todo cuando los servicios los proveen los nodos del cluster. Para solucionar este problema HACMP cuenta con el protocolo SNMP el cual permite soportar y manejar todas las aplicaciones del cliente. También se cuenta con una MIB HACMP la cual mantiene la información del cluster
- **Programa de información del cluster (Clinfo):** El demonio Clinfo corre sobre la maquina del cliente o sobre un nodo del cluster, permite obtener información acerca del estado del cluster, nodos y redes
- **Servidor de alta disponibilidad (NFS):** Es el que contiene un backup para recuperar el estado de una actividad que se este llevando a cabo en cado que se presente un fallo
- **Acceso a los discos externos compartidos:** El software HACMP soporta dos modos de acceso a los discos compartidos el modo no concurrente en donde solo un nodo tiene acceso al disco externo compartido en un momento determinado y el modo concurrente donde se permiten accesos simultáneos a los discos externos
- **Administrador de los recursos concurrentes:** Provee el acceso a los discos compartidos simultáneamente en un cluster de alta disponibilidad permitiendo adaptarse a las necesidades de una empresa.

El cluster HACMP implementado en EMCALI tiene funcionando un solo nodo y el otro se encuentra en espera, en caso que en algún momento el otro nodo falle, dentro de cada nodo se encuentran el servidor de aplicaciones, el parlay gateway y la base de datos de oracle.

Nota: Para mas información acerca del cluster HACMP referirse al documento High Availability Cluster Multi-Processing for AIX Concepts and facilities guide

4.3. PARLAY CLIENT HUB (PCH)

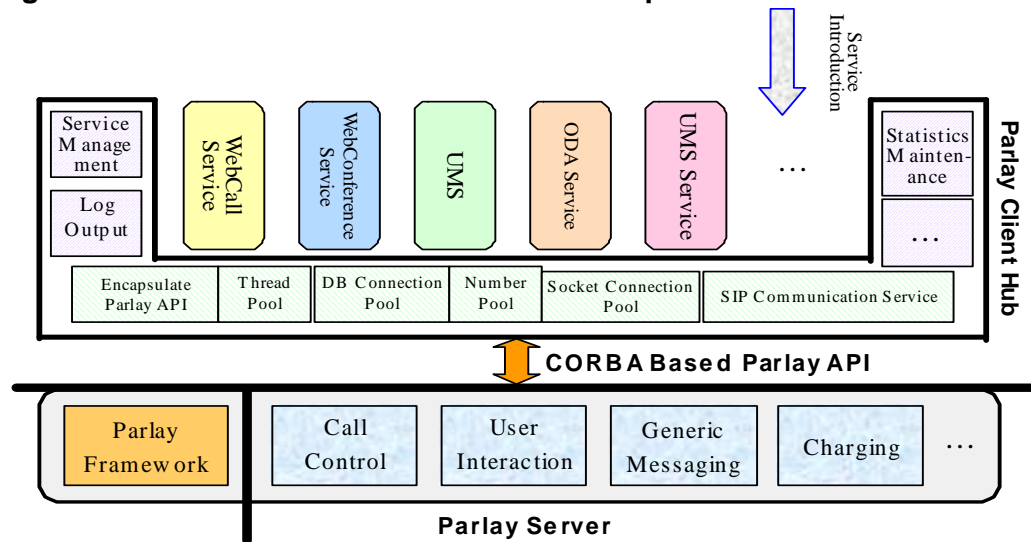
Las aplicaciones del cliente son implementadas en los servidores de aplicaciones, y para acceder a cada uno de los servicios (SCF's) no necesitan estar directamente conectados al parlay Gateway en donde se encuentran ubicados.

La comunicación entre el parlay Gateway y las aplicaciones desarrolladas por terceras partes se lleva a cabo por medio de interfaces, la plataforma ZXUP10 instalada en EMCALI cuenta con dos formas de comunicación entre dichas partes. La primera es por medio de CORBA basado en las API's de parlay y la segunda es por medio de un conjunto de API's basadas en java utilizando PCH (Parlay Client Hub). Gracias a PCH las aplicaciones no necesitan conectarse directamente con el parlay gateway sino a través de él.

PCH es quien provee las interfaces de comunicación entre las aplicaciones y el parlay Gateway, en donde las interfaces de PCH deben estar incluidas dentro de la lógica del servicio teniendo en cuenta que métodos son definidos para cada interfaz. Parlay Client Hub provee servicios desarrollados con una simple y segura plataforma encapsulando las interfaces Parlay entre los servidores de aplicaciones y las SCF's. Provee interfaces para todas las SCF's.

4.3.1. Localización de PCH dentro de la plataforma ZXUP10. Como se muestra en la figura 35, dentro de la arquitectura de la plataforma ZXUP10 el PCH se encuentra por debajo del nivel de aplicaciones y por encima del nivel de los servicios (SCF's). Las especificaciones del estándar Parlay (autenticación, búsqueda de interfaces de servicios, protocolos entre otros), se cumplen entre el PCH y el framework y entre PCH y el servidor. PCH utiliza CORBA para invocar y retornar interfaces. La intención de PCH es lograr un desarrollo adecuado de las aplicaciones, unificando el procesamiento de interfaces entre las aplicaciones y el framework y entre las aplicaciones y los SCF's para permitir a los desarrolladores enfocarse solamente en la lógica de la aplicación. Logrando que dentro de las aplicaciones del cliente estén incluidas las interfaces y los métodos correspondientes para invocar cualquier interfaz disponible que provee PCH se puede establecer la interacción con el servidor parlay.

Figura 36. Localización de PCH dentro de la plataforma ZXUP10



Fuente: ZTE. Introduction to Parlay Client Hub (PCH). Santiago de Cali, 2008. 1 archivo de computador.

4.3.2. Funciones de PCH. PCH provee algunas funciones que pueden ser usadas por las aplicaciones del cliente.

- **Manejo de sesiones.** PCH administra las sesiones de todas las aplicaciones, incluyendo creación de sesiones, actualización del estado de las sesiones y eliminación de sesiones. A través de sesiones PCH recibe servicios de retorno, y distribuye servicios (SCF's) a diferentes aplicaciones a través de la gestión o administración de aplicaciones. Adicionalmente las aplicaciones del cliente utilizan sesiones para invocar API's que proveen las SCF's
- **Manejo de aplicaciones.** PCH provee servicios comunes para diferentes aplicaciones. PCH puede controlar las funciones de una aplicación configurando la información de esta. Por ejemplo puede controlar los servicios utilizados por cada aplicación, la localización de los archivos de configuración de la aplicación y la información que contiene. PCH también puede realizar operaciones de carga y descarga para las aplicaciones cuando el sistema esta operando normalmente
- **Pool de hilos.** Para aumentar la eficiencia de operación. PCH inicia un nuevo hilo para el procesamiento de cada servicio, este pool de hilos es configurable para aumentar la capacidad de procesamiento del sistema. El número de hilos es configurado de acuerdo a las condiciones del sistema
- **Conexión a la base de datos.** Las aplicaciones del cliente necesitan consultar bases de datos durante el tiempo de operación del servicio. Para aumentar la velocidad de conexión PCH permite que la conexión a la base de

datos sea configurable. El mínimo y el máximo de conexiones pueden ser configuradas de acuerdo a las condiciones reales de operación

- **Pool de números de terminal.** Normalmente el softswitch tendrá un número de segmento, Cuando se usan terminales soft, las aplicaciones necesitan seleccionar un número para registrarlo en el softswitch
- **La función log.** PCH provee la función de log para las aplicaciones y para la posición del servicio los cuales deben ser configurados encapsulados en un archivo log4j. Los archivos log son archivos que guardan el historial del servicio.

Gracia a PCH las aplicaciones pueden acceder a los servicios ubicados en el parlay gateway sin acceder directamente a el, y utilizando el manejo de sesiones. Las aplicaciones deben interactuar con el parlay gateway en modo asíncrono y lo mas importante es que cada una de las interfaces y sus métodos deben estar incluidas dentro de la lógica del servicio lo que facilita el trabajo de los desarrolladores pues no deben hacer tareas adicionales para el llamado o solicitud de las SCF's.

4.4. HERRAMIENTAS PARA EL DESARROLLO DE APLICACIONES

PCH es implementado a través de java y todas las aplicaciones desarrolladas deben incluir las librerías necesarias para utilizar las interfaces OSA/Parlay y el PCH.

Las librerías se encuentran en el Parlay Gateway como archivos con formato .jar. Estos últimos, son archivos comprimidos que contienen las clases (.class), que ha generado la compilación del programa.

La ubicación de estas librerías dentro del parlay Gateway, se encuentran en `home/zxin10/clienthub/lib>*.jar`.

Estas librerías deben ser descargadas del parlay Gateway para poder incluirlas dentro del programa que se va a utilizar para el desarrollo de la aplicación, esto se hace por medio del comando `import home/zxin10/clienthub/lib>*.jar`.

Para definir el entorno para el desarrollo de los servicios se utiliza java. Una herramienta que permite el desarrollo rápido de aplicaciones, que es de fácil configuración y se adapta de forma sencilla a la plataforma de EMCALI y es recomendada por la empresa ZTE para el desarrollo de las aplicaciones es JBuilder.

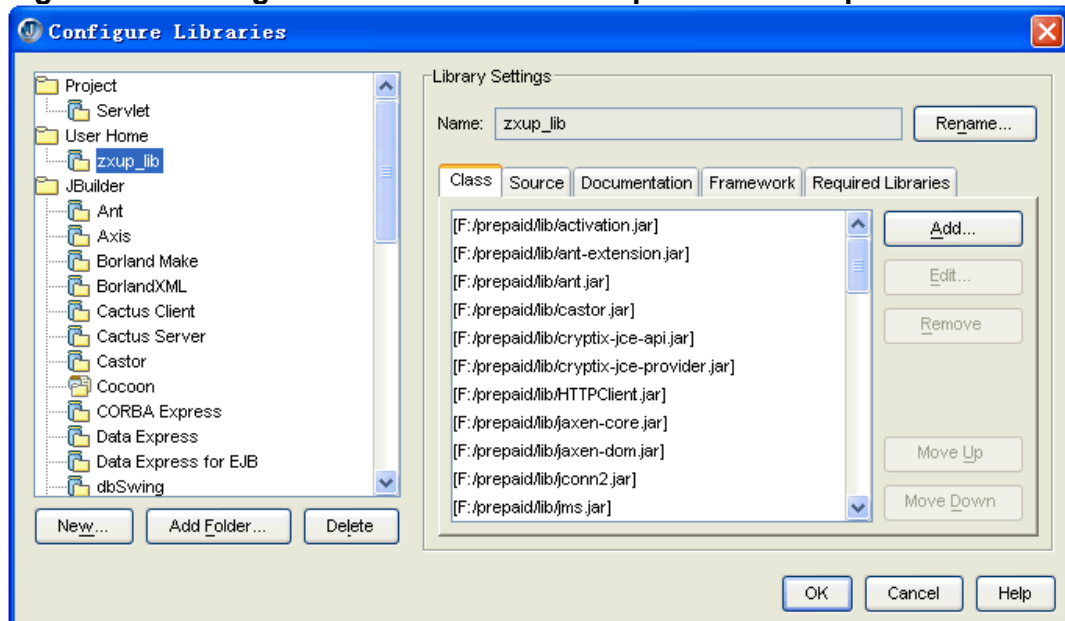
JBuilder es un entorno para desarrollar aplicaciones de forma rápida y fiable para java basado en eclipse. Permite el desarrollo de aplicaciones robustas,

servicios web, aplicaciones empresariales e industriales y puede trabajar también con versiones anteriores.

La última versión, JBuilder 2008, tiene 3 ediciones: Enterprise que es la versión más completa, profesional y Turbo esta última es la versión libre. Está disponible para Windows. Linux y macos X y pertenece a la compañía de software americana CodeGear.

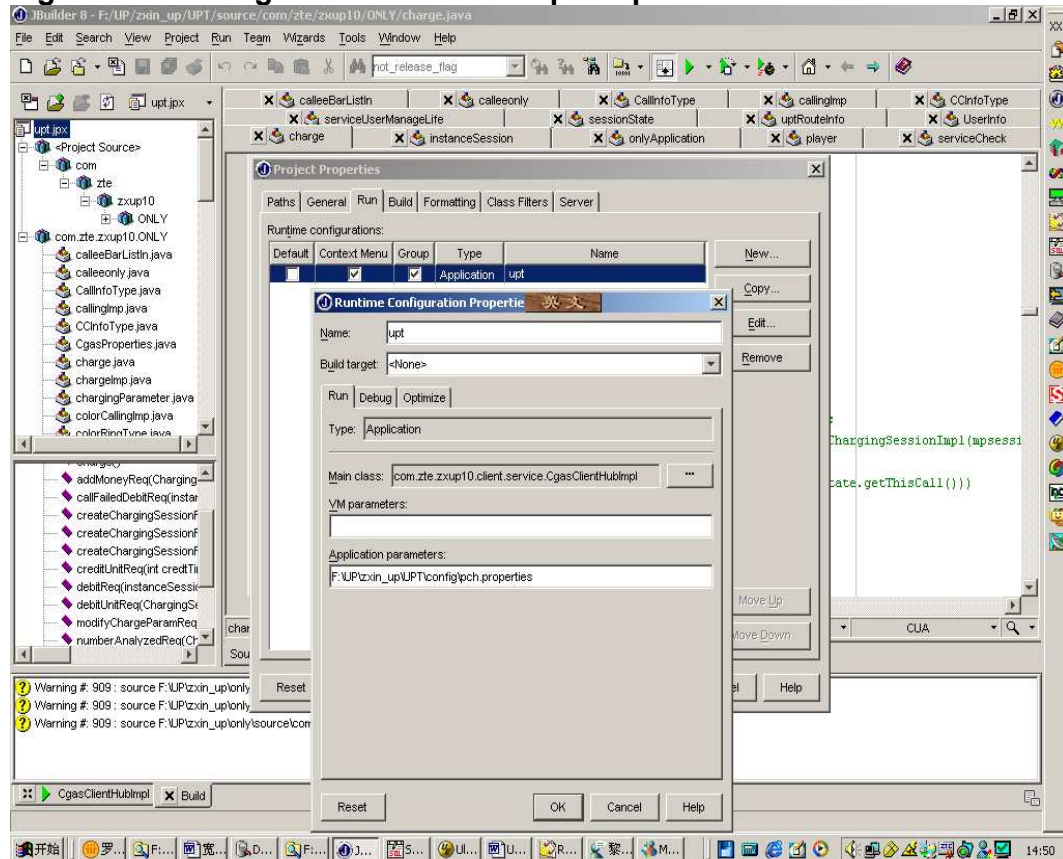
4.4.1. Configuración de JBuilder. Para la configuración del programa JBuilder primero deben ser descargadas las librerías BES lib, tool lib y pch.lib del parlay Gateway que se encuentran en home/zxin10/clienthub/lib>*.jar y configurarlas dentro de JBuilder.

Figura 37. Configuración de librerías de la plataforma zxup10 en JBuilder



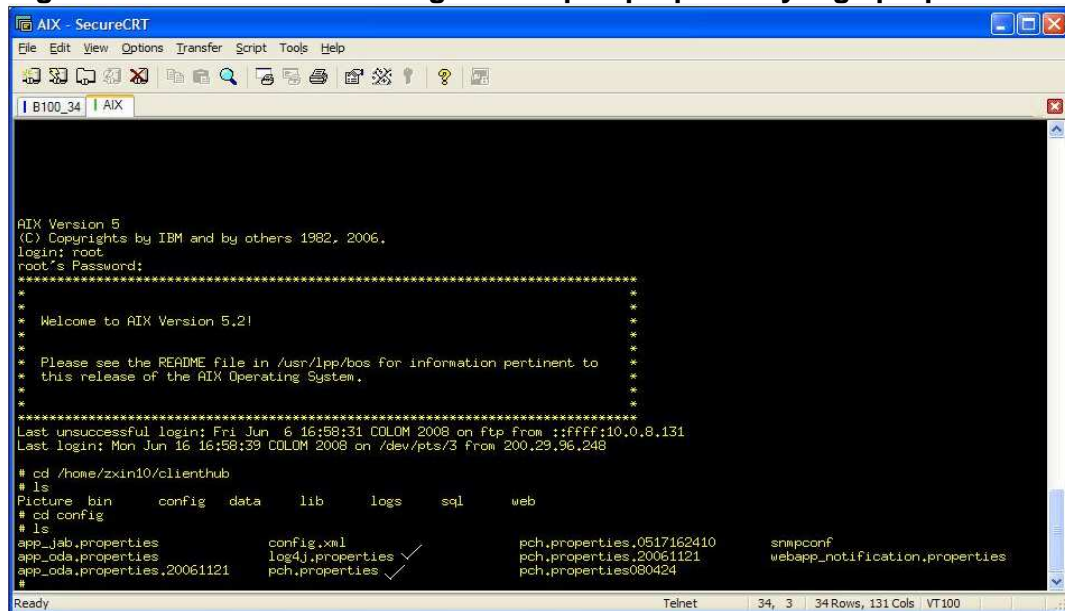
La configuración de la clase principal para ejecutar el servicio está incluida en com.zte.zxup10.client.service.CgasClientHubImpl y también debe ser configurada en JBuilder, cada aplicación tiene una clase principal diferente.

Figura 38. Configuración de la clase principal



Dentro del parlay gateway también debe hacerse una serie de modificaciones para ingresar las propiedades del servicio que se desea implementar. Los archivos de configuración de un servicio dentro del parlay Gateway son pch.properties donde se configura la inicialización del servicio y log4.properties donde se configuran los log's (registros) de nivel y de posición del servicio y PCH. El archivo de configuración log4j.properties se encuentra localizado en el directorio classes del servicio. Estos dos archivos se encuentran en /home/zxin10/clienthub.

Figura 39. Archivos de configuración pch.properties y log4.properties



El archivo de configuración para el arranque del servicio es pch.properties dentro del parlay gateway. La información contenida en este archivo incluye:

JDBC_DRIVER=com.sybase.jdbc2.jdbc.SybDriver #pch //Driver de la base de datos configurada

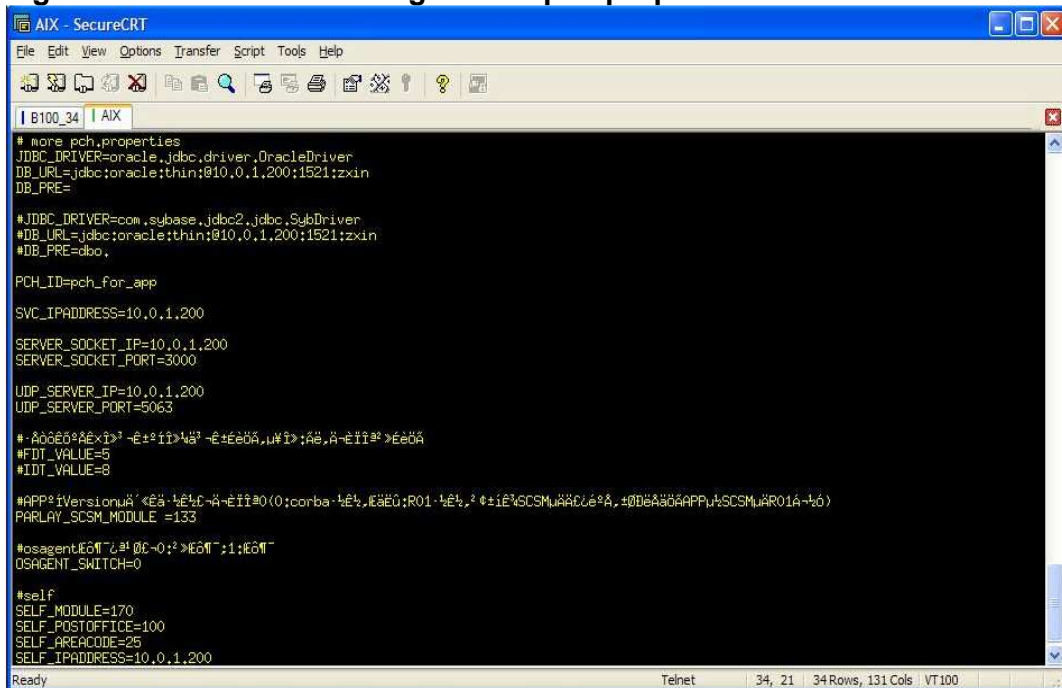
DB_URL=jdbc:sybase:Tds:10.40.58.76:4100 #pch //URL de la base de datos configurada

PCH_ID=pch_for_app # //pch_id utilizado en el servicio

SVC_IPADDRESS=10.40.90.193 #dirección IP del parlay gw

Para el caso de la plataforma multiservicios de EMCALI este archivo esta configurado como se muestra a continuación:

Figura 40. Archivo de configuración pch.properties



```
# more pch.properties
JDBC_DRIVER=oracle.jdbc.driver.OracleDriver
DB_URL=jdbc:oracle:thin:@10.0.1.200:1521:zxin
DB_PRE=

#JDBC_DRIVER=com.sybase.jdbc2.jdbc.SybDriver
#DB_URL=jdbc:oracle:thin:@10.0.1.200:1521:zxin
#DB_PRE=dbo,

PCH_ID=pch_for_app

SVC_IPADDRESS=10.0.1.200

SERVER_SOCKET_IP=10.0.1.200
SERVER_SOCKET_PORT=3000

UDP_SERVER_IP=10.0.1.200
UDP_SERVER_PORT=5063

#-ÀöGÉ0²Æ×i³-è±ii²¼è³-è±èèöâ,u#i²:Äè,Ä-èii²>èèöâ
#FDT_VALUE=5
#IDT_VALUE=8

#APP²iVersionuÄ²<èâ-èè¼c-Ä-èii²#(0:conba-èè¼.EèèÜ;R01-èè¼²²±ièXSCSM,ÄÄDÜè²Ä,±0DBÄöâAPP,ÄSCSM,ÄR01Ä-èö)
PARLAY_SCSM_MODULE =133

#osagentèèö¼²@è-0:²>èèö¼²;1:èèö¼²
OSAGENT_SWITCH=0

#self
SELF_MODULE=170
SELF_POSTOFFICE=100
SELF_AREACODE=25
SELF_IPADDRESS=10.0.1.200
```

En la anterior grafica podemos observar que el driver utilizado para la configuración de la base de datos es oracle.jdbc.driver.OracleDriver. La URL de la base de datos configurada es jdbc:oracle:thin:@10.0.1.200:1521:zxin y la dirección IP del Parlay Gateway es 10.0.1.200.

La forma como se puede acceder al parlay Gateway de forma remota es por medio de telnet a la dirección IP del parlay Gateway 10.0.1.200 y con una identificación por parte de quien desea ingresar.

4.5. IMPLEMENTACIÓN DEL SERVICIO Y VERIFICACIÓN EN LA PLATAFORMA DE SERVICIOS

Como fue nombrado anteriormente la consola OAM es la encargada de la gestion de los servicios instalados en la plataforma de servicios, por lo tanto es aquí donde se lleva a cabo todo el proceso de montaje del servicio para empezar su funcionamiento y para poder ser utilizado por los usuarios.

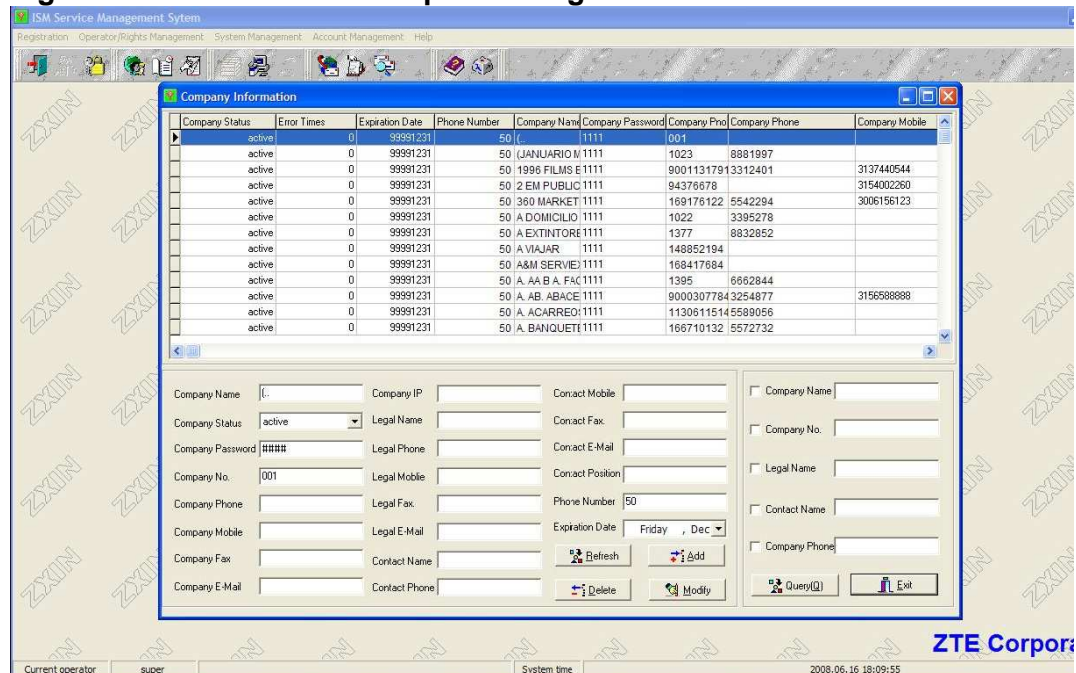
4.5.1. Descarga del servicio. Toda la información del servicio es descargada a través de PCH-OAM y OAM. PCH-OAM es donde se configura el acceso al servicio. Dentro de esta información está incluido:

- Nombre del servicio, archivo de configuración y clase de inicialización
- Las SCF's que necesitan ser registradas para el servicio, en este caso para tener acceso a los servicios requeridos por MPCC

Dentro de la información configurada a través de OAM se encuentra la identificación de la sesión asignada para Call Control (CC).

La configuración de los servicios se realiza a través de la herramienta ISM como se muestra a continuación.

Figura 41. Herramienta ISM para configuración de los servicios



4.5.2. Descarga de la información de un servicio por medio de un script. Todas las operaciones pueden ser escritas en un script en un base de datos y descargar la información del servicio corriendo el script. Esto puede simplificar las operaciones. Por ejemplo el script pchoam_innet.sql puede ser utilizado para implementar operaciones.

4.5.3. Descarga de anuncios de voz. Los archivos de servicios de anuncios pueden ser guardados en el servidor de medios. Generalmente los archivos

de anuncios son guardados en /mailstore/mp/ann. La extensión para el archivo de voz debe ser .wav.

4.5.4. Descarga de la lógica del servicio. Generalmente la lógica del servicio es guardada en el archivo home/zxin10/clienthub/lib. Debe asegurarse que la clase de inicialización para el servicio sea configurada en home/zxin10/clienthub/data/runtime.env. el archivo de configuración del servicio generalmente esta ubicado en home/zxin10/clienthub/config

Por medio de la herramienta mtrace es posible verificar el flujo normal de una llamada como se muestra a continuación.

Figura 42. Flujo de información en una llamada

序号	会话ID	事件ID	事件名称	消息内容	时间
0	60003	MGCP	TRACE	Send to MS:AUEP 7 ms/*/*@[202.202.91.5] MGCP 1.0 ...	20
0	60003	MGCP	TRACE	Res from MS:200 7 Audit endpoint is done	20
12	0	SIP	TRACE	rev OPTIONS_MTD_SS_LINKTEST !!!!	20
13	10002	SIP	TRACE	SIPMSG IN:INVITE sip:013813907286@202.202.91.193;...	20
13	10001	SIP	TRACE	SIPMSG OUT:SIP/2.0 100 TryingVia: SIP/2.0/UDP 202.20...	20
6144	60003	CC	TRACE	Receive ADRESSEVENTNOTIFY: src=[0105160004], de...	20
1	60003	CC	TRACE	Call IpAppMultiPartyCallControlManager.reportNotificatio...	20
1	60003	Charg...	TRACE	createChargingSession rcv from app, user = '01381390...	20
1	60003	Charg...	TRACE	createChargingSession ok	20
1	60003	Charg...	TRACE	NUM_ANALYSIS rcv from app, req units: 0 0 0 0	20
0	1	60003	Charg...	send JF_NUM_ANALYSIS_REQ to jf pro, [013813907286]	20
1	1	60003	Charg...	NUM_ANALYSIS send to app, 25	20
2	1	60003	Charg...	release rcv from app	20
3	0	60003	Charg...	account '013813907286' ref count =0	20
4	0	60003	Charg...	remove account '013813907286'	20
5	1	60003	CC	MultiPartyCall::createCallLeg	20
6	1	60003	CC	MultiPartyCallLeg::eventReportReq, LegID=0x102	20
7	1	60003	CC	Send ReportRequest to Adapter_B3, LegID=0x102	20
8	1	60003	CC	MultiPartyCallLeg::setChargePlan, LegID=0x102	20
9	1	60003	CC	MultiPartyCallLeg::routeReq, LegID=0x102	20
0	0	60003	CC	Route P_CALLLEG_ATTACH_IMPLICITLY, callLegID=0x102	20
1	1	60003	CC	CallLeg::route, tryingChangeBridged =true, LegID=0x102	20
2	1	60003	CC	Send Route to Adapter_B3, LegID=0x102	20
3	14	10001	SIP	SIPMSG OUT:INVITE sip:02513813907286@202.202.91.193;...	20
4	14	10002	SIP	SIPMSG IN:SIP/2.0 100 TryingVia: SIP/2.0/UDP 202.202.91...	20
5	14	10002	SIP	SIPMSG IN:SIP/2.0 180 RingingVia: SIP/2.0/UDP 202.202...	20
6	1	60003	CC	Receive MSG_FROM_ADAPTER_183: LegID=0x102	20
7	0	60003	CC	Send 180 PRACK:0x102	20
8	1	60003	CC	TRACE Send PRACK to SIP_peer_state=0, LegID=0x102	20
9	0	60003	CC	forwardedMessageFromLeg, legSessionID= 0x102	20
10	1	60003	CC	TRACE Call IpAppCallLeg.eventReportRes, callEventType=9,LegID=0x101	20
1	1	60003	CC	TRACE Send Continue to SIP_peer_state=16, LegID=0x101	20
2	14	10001	SIP	SIPMSG OUT:PRACK sip:02513813907286@202.202.91.193;...	20
3	13	10001	SIP	SIPMSG OUT:SIP/2.0 180 RingingVia: SIP/2.0/UDP 202.202.91...	20
4	14	10002	SIP	SIPMSG IN:SIP/2.0 200 OKVia: SIP/2.0/UDP 202.202.91.193;...	20
5	1	60003	CC	TRACE Receive MSG_FROM_ADAPTER_183PRACKRES: LegID=0x102	20
6	15	10001	SIP	SIPMSG OUT:OPTIONS sip:1234567890@202.202.91.55;...	20
16	10001	SIP	TRACE	SIPMSG OUT:OPTIONS sip:1234567890@202.202.91.1;u...	20
8	15	10002	SIP	SIPMSG IN:SIP/2.0 200 OKVia: SIP/2.0/UDP 202.202.91.193;...	20
9	16	10002	SIP	SIPMSG IN:SIP/2.0 200 OKVia: SIP/2.0/UDP 202.202.91.193;...	20
0	14	10002	SIP	SIPMSG IN:SIP/2.0 200 OKVia: SIP/2.0/UDP 202.202.91.193;...	20
1	14	62002	SIP	TRACE EventReportRes	20
2	1	60003	CC	TRACE Receive EVENTREPORTRES: eventType=10, LegID=0x102	20
3	0	60003	CC	TRACE rcv P_CALL_EVENT_ANSWER, and SDPOFBridgedLeg no...	20
4	14	10001	SIP	SIPMSG OUT:ACK sip:02513813907286@202.202.91.55 ...	20
5	0	60003	CC	TRACE forwardedMessageFromLeg, legSessionID= 0x102	20
6	1	60003	CC	TRACE Call IpAppCallLeg.eventReportRes, callEventType=10,LegID=0x101	20
7	1	60003	CC	TRACE Send Continue to SIP_peer_state=17, LegID=0x101	20
8	1	60003	CC	TRACE send 200(Active), tryingChangeBridged =true, LegID=0x...	20
9	13	10001	SIP	SIPMSG OUT:SIP/2.0 200 OKVia: SIP/2.0/UDP 202.202.91.193;...	20
0	13	10002	SIP	SIPMSG IN:ACK sip:13813907286@202.202.91.193 SIP/2.0/UDP 202.202.91.193;...	20
1	14	10002	SIP	SIPMSG IN:BYE sip:0105160004@202.202.91.193 SIP/2.0/UDP 202.202.91.193;...	20
2	14	62002	SIP	TRACE EventReportRes	20
3	1	60003	CC	TRACE Receive EVENTREPORTRES: eventType=11, LegID=0x102	20
4	0	60003	CC	TRACE forwardedMessageFromLeg, legSessionID= 0x102	20
5	1	60003	CC	TRACE Call IpAppCallLeg.eventReportRes, callEventType=11,LegID=0x101	20
6	0	60003	CC	TRACE bridgeLeg(callee) release event, ignore !, legSessionID=...	20
7	0	60003	CC	TRACE Leg release, legSessionID= 0x102	20
8	1	60003	CC	TRACE Send Release to Adapter_B3, LegID=0x102	20
9	1	60003	CC	TRACE MultiPartyCall::release	20
0	1	60003	CC	TRACE Send Release to Adapter_B3, LegID=0x101	20
1	1	60003	CC	TRACE Call IpAppMultiPartyCall.callEnded	20
2	13	10001	SIP	SIPMSG OUT:BYE sip:0105160004@202.202.91.55 SIP/2.0/UDP 202.202.91.55;...	20
3	13	10002	SIP	SIPMSG IN:SIP/2.0 200 OKVia: SIP/2.0/UDP 202.202.91.193;...	20
4	2	60003	Charg...	TRACE createChargingSession rcv from app, user = '01381390...	20
5	2	60003	Charg...	TRACE createChargingSession ok	20
6	2	60003	Charg...	TRACE DirectDebitUnit rcv from app, req units: 18 0 0 0	20
7	2	60003	Charg...	TRACE send to jf pro, InType=1, InAmount=18, OutType=2	20
8	2	60003	Charg...	TRACE jf ok: InType=1, InAmount=18, OutType=2, OutAmount=17	20
9	2	60003	Charg...	TRACE DirectDebitUnit send to app, debited units: 18 0 0 0, pay...	20
0	0	60003	Charg...	TRACE send to cdr pro	20
1	2	60003	Charg...	TRACE release rcv from app	20
2	0	60003	Charg...	TRACE account '013813907286' ref count =0	20
3	0	60003	Charg...	TRACE remove account '013813907286'	20

5. CAPTURA DE REQUERIMIENTOS

La plataforma multiservicios de EMCALI es una herramienta muy potente en cuanto a la generación de nuevos servicios para la gran cantidad de usuarios que posee EMCALI dentro de la ciudad de Cali y las ciudades vecinas. Esta plataforma permite el desarrollo de un gran número de servicios, de diferentes tipos, para satisfacer las necesidades de los usuarios que utilizan las redes de EMCALI, siendo esto también un gran atractivo para los desarrolladores de aplicaciones.

Por lo anterior se hace necesario entonces la captura de requerimientos de los servicios que EMCALI desea implementar sobre la plataforma multiservicios, en donde a continuación se proponen una serie de servicios y por medio de una reunión con el asesor de EMCALI, se establecieron las características principales y las funciones de cada uno de los servicios.

Empezamos primero por identificar la forma en que se van a manejar los servicios y los perfiles de cada uno de los usuarios. Establecemos entonces que es un administrador de servicios el encargado de controlarlos, es el encargado de autorizar a los usuarios para utilizar los diferentes servicios y es el encargado de monitorearlos para la identificación de posibles fallas.

Se definió que las tareas que debe realizar el administrador de los servicios deben ser:

- El administrador debe identificarse por medio de un nombre de usuario y contraseña para poder realizar tareas como administrador
- El administrador debe controlar los servicios dentro de esta tarea se encuentran activar los servicios, deshabilitarlos, eliminarlos o modificarlos
- El administrador debe manejar los perfiles de usuario realizando actividades como ingresar usuarios, deshabilitarlos, eliminarlos o modificarlos
- El administrador debe monitorear los servicios incluyendo la consulta del estado de los servicios y la realización de simulaciones de los mismos

La definición de los casos de uso y su definición se encuentra en el anexo 1.

El segundo paso es la definición y captura de los requerimientos de los servicios que EMCALI considera necesarios implementar en la plataforma. Dentro de la reunión al asesor de EMCALI se le presentó una lista de posibles servicios y sus funciones, de aquí se tomaron los que se consideraron más importantes, se discutieron y se definieron las tareas que cada servicio debe realizar.

A continuación son definidos cada uno de los servicios definidos dentro de la reunión y sus funciones.

5.1. SISTEMA DE INFORMACIÓN Y VISUALIZACIÓN DE TRÁFICO VIAL

Este servicio permite que el usuario a través de un computador utilizando wimax o la red cableada pueda consultar y visualizar el tráfico en las principales vías de la ciudad por medio de cámaras instaladas en las vías. Las imágenes son en tiempo real permitiéndole al usuario tomar las rutas con menos tráfico sobre todo en las horas de mayor flujo vehicular.

Los requerimientos para esta aplicación son:

- El servicio debe permitir la visualización de las vías por medio de las cámaras instaladas en la ciudad
- El servicio debe trabajar con Internet
- El servicio debe permitir escoger a través de un menú una zona de la ciudad (norte, sur, oriente, occidente)
- El servicio debe permitir escoger a través de un menú que vía desea visualizar según la zona
- El servicio debe operar en tiempo real
- El servicio debe permitir al usuario la visualización del tráfico de la vía seleccionada
- El servicio debe permitir visualizar el tráfico en la vía durante un periodo determinado
- El servicio debe permitir hacer varias consultas

5.2. SERVICIO DE ÚNICO NÚMERO TELEFÓNICO

Este servicio permite que el usuario pueda tener un único número telefónico en diferentes ubicaciones y en cualquier tipo de red (casa, oficina, etc). La llamada se transfiere de un de un sitio a otro hasta obtener una respuesta.

Los requerimientos para esta aplicación son:

- El usuario puede utilizar el servicio desde cualquier tipo de red (inalámbrica o fija)

- El usuario debe registrar un número
- El usuario debe registrar los números de los lugares a los que será transferida la llamada
- El servicio debe permitir que los números sean guardados
- El servicio debe enrutar las llamadas a otras ubicaciones a través de un solo número
- La llamada se transfiere de un sitio a otro después de un tiempo de no tener respuesta o la línea se encuentra ocupada
- Si después de hacer todas las transferencias no hay respuesta la llamada es cancelada

5.3. DIRECTORIO TELEFÓNICO

Este servicio funciona como un directorio telefónico en donde el llamante puede decir el nombre de la persona a quien va a llamar y el sistema automáticamente hace la llamada, permitiendo tener un servicio de agenda telefónica.

Los requerimientos para esta aplicación son:

- El servicio debe permitir grabar un nombre en la agenda cuando el usuario dice el nombre
- El servicio debe permitir guardar el número cuando el usuario lo digite
- El servicio debe permitir el acceso a una agenda por medio de una clave asignada al usuario
- El servicio debe realizar la llamada cuando el usuario dice el nombre de la persona que desea llamar
- El servicio debe reconocer el nombre y buscarlo en la agenda
- Al encontrar la información de a quién se va a llamar el servicio debe permitir que automáticamente se haga la llamada
- El servicio debe permitir que el usuario pueda repetir un nombre o cambiarlo en caso que no sea encontrado

- El servicio debe permitir que el usuario borre algún número de la agenda
- El servicio debe permitir que el usuario modifique un número de la agenda

5.4. SERVICIO DE GUÍA EN LA CIUDAD

Este servicio permite al usuario visualizar la ruta para llegar a diferentes sitios en la ciudad, ingresando el sitio donde se encuentra inicialmente y a donde quiere ir a través de un GPS y la información de la ubicación será enviada Al usuario través de la red GPRS de EMCALI la cual tiene cubrimiento en toda la ciudad.

Los requerimientos para este servicio son:

- El servicio debe trabajar con la red GPRS
- El sistema debe ofrecer una guía al usuario para llegar de un sitio a otro dentro de la ciudad
- El servicio debe permitir al usuario visualizar las diferentes opciones de rutas para llegar a un sitio determinado
- El servicio debe permitir visualizar las rutas en un mapa de la ciudad
- El servicio debe permitir al usuario ingresar el sitio de origen y el sitio destino
- El servicio debe permitir que el usuario haga varias consultas
- El servicio debe permitir actualizaciones en las rutas y el mapa de la ciudad

5.5. SERVICIO DE TABLERO ELECTRÓNICO

Este servicio permite iniciar una video conferencia en donde un usuario puede crear un tablero y escribir sobre él y todos los miembros de la conferencia lo pueden ver e igualmente escribir sobre el. Esta aplicación es ideal para gestiones empresariales.

Los requerimientos para este servicio son:

- El servicio debe permitir a un usuario iniciar una conferencia y asignarle un código

- El servicio debe permitir que los usuarios ingresen a la conferencia por medio del código asignado
- El servicio debe permitir que cualquiera de los miembros pueda crear un tablero electrónico
- El servicio debe permitir que todos los miembros de la conferencia puedan visualizar el tablero
- El servicio debe permitir que cualquiera de los miembros de la conferencia escriban sobre el tablero o lo borre
- El servicio debe permitir que cualquier miembro de la conferencia oculte el tablero
- El servicio debe permitir que cualquier miembro de la conferencia guarde la información que se encuentra en el tablero

5.6. SERVICIO DE BACKTONES

Este servicio permite que los usuarios de las diferentes redes puedan configurar backtones. El usuario puede establecer escuchar el backtone cuando realiza llamadas a otros números o en el momento que reciba una llamada el llamante escuchara el backtone mientras el usuario contesta.

Los requerimientos para este servicio son:

- El servicio debe permitir que el usuario escoja entre varias opciones de backtones
- El servicio debe permitir al usuario descargar un backtone
- El servicio debe permitir al usuario escoger si desea escuchar el backtone mientras llama a otros números
- El servicio debe permitir al usuario escoger si desea que los llamantes escuchen el backtone mientras responde la llamada
- El servicio debe permitir guardar los cambios hechos por el usuario

5.7. LOCALIZACIÓN POR MEDIO DE DISPOSITIVOS MOVILES

Este servicio permite al usuario o empresa localizar los vehículos de su empresa utilizando dispositivos móviles. Pede consultar la localización exacta,

puede enviar mensajes desde la central a la empresa, se puede activar una alarma en caso que el dispositivo salga del perímetro autorizado.

Los requerimientos para este servicio son:

- El servicio debe permitir la ubicación de sus recursos (camiones, dispositivos móviles) o del personal de la empresa en cualquier momento
- El servicio debe alertar a quien controle los recursos cuando alguno se salga del perímetro autorizado
- El servicio debe trabajar en tiempo real
- El servicio debe permitir visualizar todos sus recursos al mismo tiempo
- El servicio debe permitir ingresar nuevos dispositivos por una persona autorizada
- El servicio debe permitir borrar dispositivos por una persona autorizada

Nota: Dentro del anexo 2 se especifican los casos de uso y requerimientos funcionales para los servicios que según los asesores de EMCALI son los más relevantes y en un futuro estarían interesados en implementar. Estos servicios son el sistema de información y visualización de tráfico vial, el servicio de guía en la ciudad, el servicio de backtones y el servicio de localización por medio de dispositivos móviles.

6. CASO DE ESTUDIO

Dentro de este capítulo por medio de un ejemplo será explicada la forma en que debe diseñarse, desarrollarse e implementarse un servicio en la plataforma de servicios instalada en EMCALI.

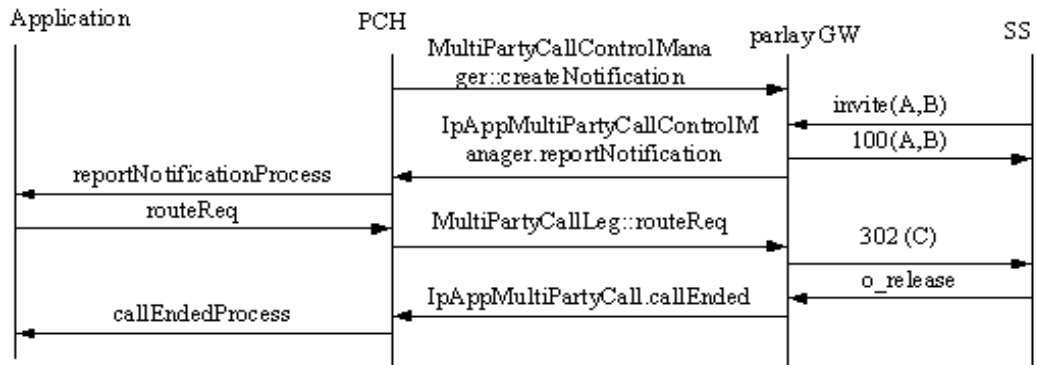
6.1. SERVICIO NP (NUMBER PORTABILITY)

El servicio NP (Number Portability, número portable), es un simple servicio con el que los usuarios pueden seguir recibiendo llamadas con un número antiguo en el caso que su número físico haya sido cambiado, es decir cuando el usuario es llamado a su antiguo número la llamada es transferida al nuevo número asignado.

6.1.1. Funcionamiento del servicio NP. El servicio es implementado a través de una transferencia de llamada. Cuando un usuario marca un número NP, el servicio consulta la base de datos de la aplicación y toma el número actual para realizar la conexión. A través de una transferencia de llamada, el servicio recibe una notificación de evento, enruta la llamada y libera la llamada sin generar CDR.

El servicio utiliza los módulos MPCC (MultiParty Call Control) o GCC (Generic Call Control), por lo que el flujo de información es relativamente fácil, como se muestra en el siguiente diagrama de secuencia.

Figura 43. Flujo de información del servicio NP



SS: Softswitch
 Parlay GW: Parlay gateway
 PCH: Parlay client Hub

Fuente: ZTE. NP Service Analysis. Santiago de Cali, 2008. 1 archivo de computador.

Cuando se inicia PCH, registra una llamada de notificación de un evento (MultiPartyCallControlManager::createNotification), al parlay Gateway por una aplicación. Por medio de esta interfaz son manejados todos los servicios MPCC y el método createNotification(), permite generar notificaciones de eventos que se pueden enviar a la aplicación, este es primer paso para que una aplicación tenga notificación de los diferentes eventos en la red. Luego el parlay Gateway invoca en PCH la interfaz IpAppMultiPartyCallControlManager.reportNotification() donde la aplicación es informada de los eventos por medio del método reportNotification(). Después PCH invoca la implementación del servicio a través de la interfaz reportNotificationProcess creando una sesión, pues la interacción entre el servicio y PCH está basada en el uso de sesiones.

El servicio analiza la información de los parámetros de entrada de la interfaz reportNotificationProcess del PCH y obtiene los números del llamante y el número solicitado. El número actual se obtiene desde una base de datos donde se hace una relación de acuerdo al número marcado por el llamante. Luego la aplicación invoca en el PCH routeReq a través de la transferencia de la llamada. PCH invoca en el parlay Gateway MultiPartyCallLeg::routeReq, por medio de esta interfaz y el método routeReq() puede realizar la conexión de la llamada. Asociando esta llamada con una dirección. Después la llamada es transferida.

Para terminar el parlay Gateway hace una llamada de retorno al PCH por medio de IpAppMultiPartyCall.callEnded indicando a la aplicación que la llamada ha terminado en la red. Luego PCH hace una llamada de retorno al servicio por medio de la interfaz callEndedProcess con la cual el servicio libera el medio y la sesión.

6.1.2. Casos de uso. A continuación se definen cada uno de los caso de uso para el servicio NP para el caso del usuario.

- CU #1: Ingresar usuario
- CU #2: Ingresar número anterior
- CU #3: Ingresar número nuevo

Figura 44. Definición de casos de uso para el servicio NP

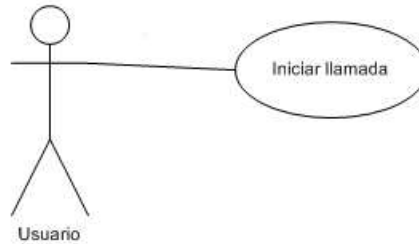


Tabla 2. Caso de uso para servicio NP

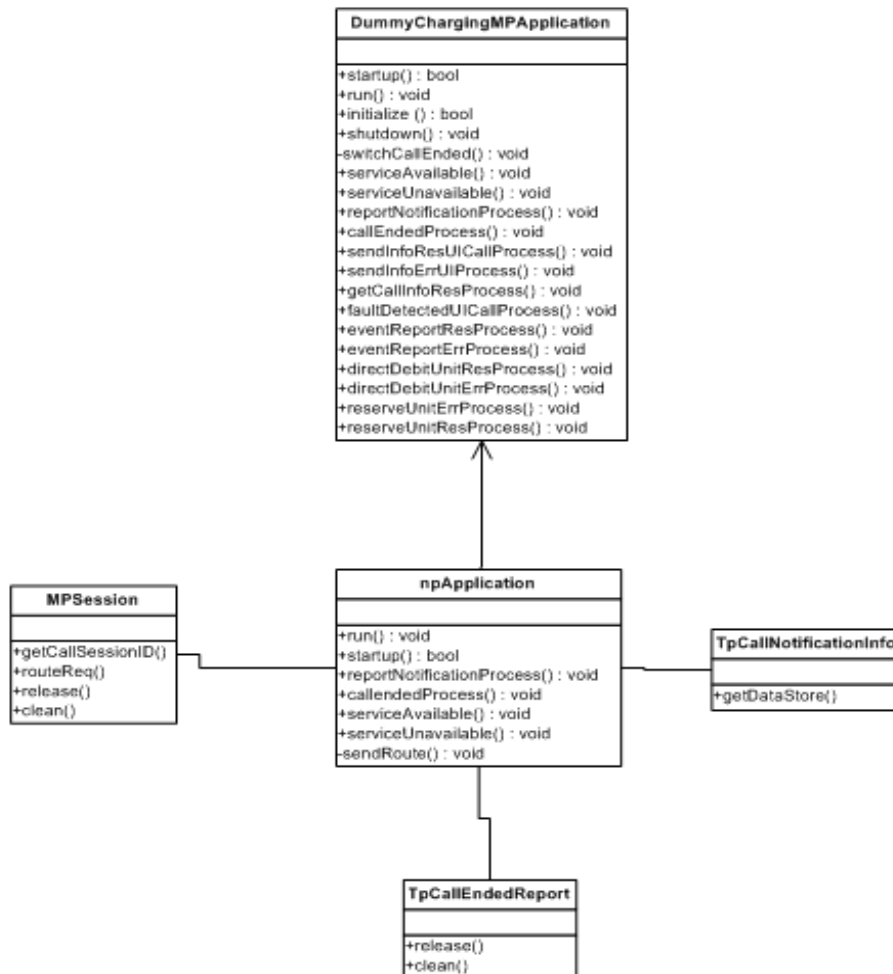
CU #1	Iniciar llamada	
Actor	Usuario	
Precondición	El usuario debe conocer el número del usuario del servicio	
Descripción	Este caso de uso describe como el usuario debe iniciar una llamada	
Flujo de eventos	<ul style="list-style-type: none"> • Este caso de uso inicia cuando un usuario (llamante), inicia una llamada • El sistema debe comparar el número marcado por el llamante en una base de datos y relacionarlo con un número nuevo • El sistema transfiere la llamada al número nuevo 	
Postcondición	El usuario inicia la conversación	
Excepciones	Situación	Respuesta
	En el caso que el número marcado no tenga ningún número relacionado en la base de datos	El sistema transfiere la llamada al número marcado

6.2. DISEÑO DEL PROGRAMA PARA EL SERVICIO NP

El servicio NP tiene una clase principal que es npApplication la cual contiene los metodos necesarios para este servicio, inicializa el servicio, lo detiene, tiene todas las interfaces de PCH incluidas y extiende de la clase DummychargingMPApplication. Por medio de los métodos MPSession, TPCallNotificationInf y TpCallEndedReport se realizan las operaciones de

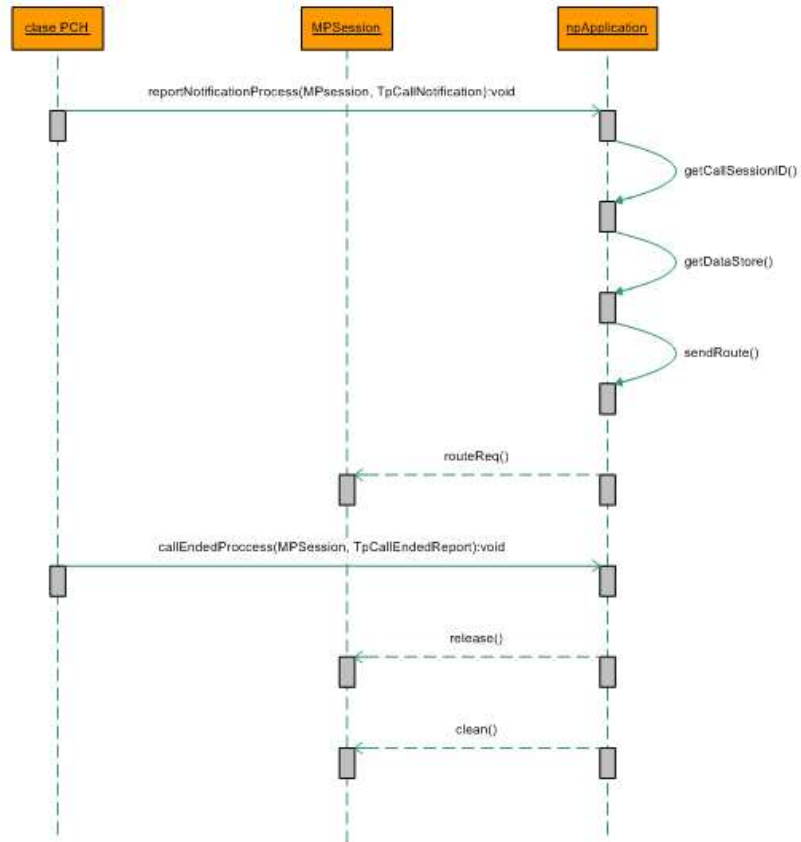
establecimiento de la llamada, asociación del número antiguo con el número nuevo en una base de datos y reportes de fin de llamada.

Figura 45. Diagrama de clases para el servicio NP



El servicio NP según el diagrama de secuencia mostrado en la figura 47, funciona de la siguiente forma: Cuando una llamada accede al servicio por medio de reportNotificationProcess, el servicio crea una instancia de la llamada y se inicia una sesión en el PCH a la cual se le asigna un ID con el método getCallSessionID, el número marcado es buscado dentro de una base de datos para verificar si tiene algún número nuevo asociado por medio de getDataSource, si no tiene ningún número asociado la llamada es transferida al número marcado. Si tiene un número nuevo asociado la llamada es transferida a dicho número, enrutandola a través de sendRoute. Luego el servicio es notificado que la llamada termino por medio de callEndedProcess, el servicio termina la llamada con release y la sesion del PCH con clean.

Figura 46. Diagrama de secuencia para el servicio NP



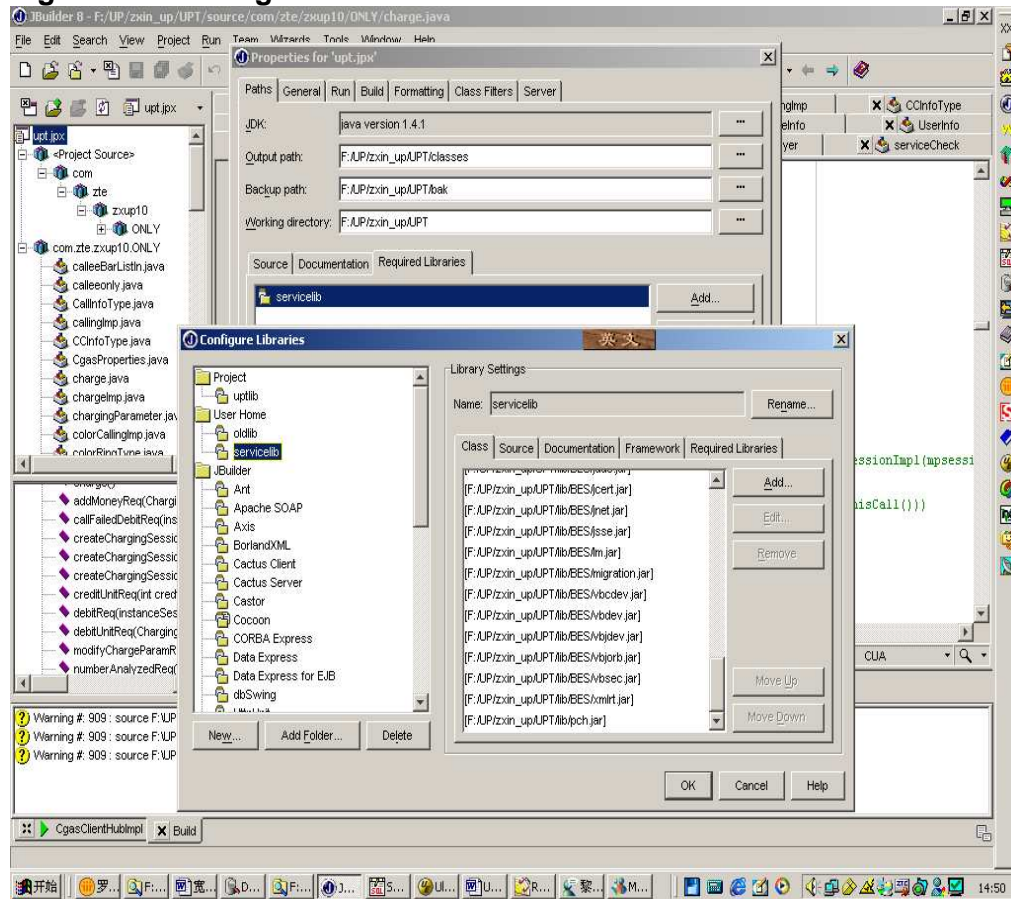
6.3. IMPLEMENTACIÓN DEL SERVICIO

Para describir como se debe implementar el servicio se van a explicar una serie de pasos donde se configuran cada una de las herramientas necesarias para llevarla a cabo.

6.3.1. Paso 1. Incluir librerías en JBuilder. El desarrollo de este servicio se hizo utilizando el programa JBuilder, a continuación se describe la forma en que debe ser configurado.

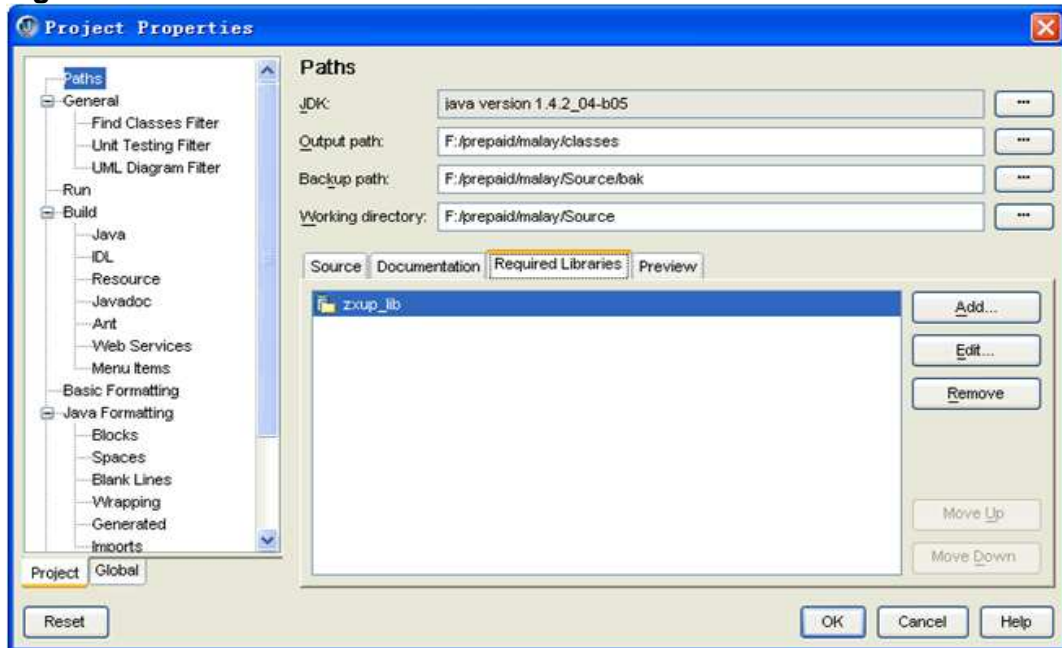
Todos los proyectos deben incluir las librerías BES lib, tool lib y pch.lib como se muestra en la siguiente figura.

Figura 47. Configuración de librerías en JBuilder



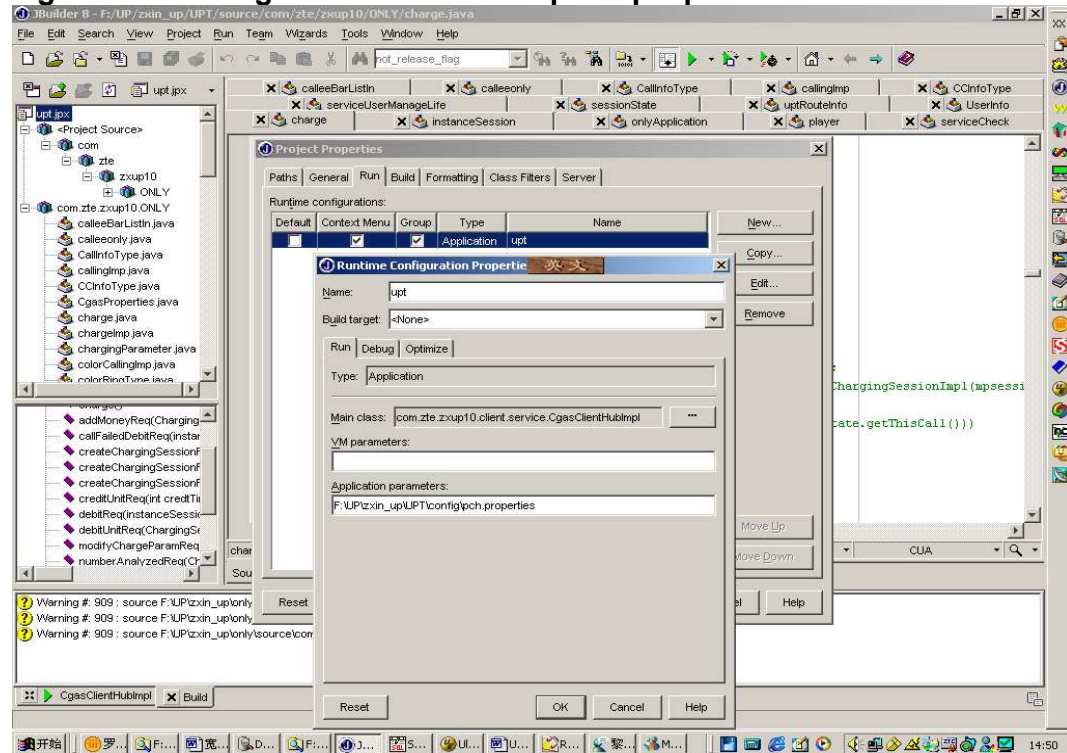
Nota: estas librerías deben ser descargadas desde el parlay Gateway desde la ruta `home/zxin10/clienthub/lib>*.jar` y luego incluirlas dentro del programa JBuilder.

Figura 48. Como incluir las librerías dentro de JBuilder



6.3.2. Paso 2. Configuración de la clase principal. Dentro de JBuilder también se debe configurar la clase principal para ejecutar el servicio incluida en `com.zte.zxup10.client.service.CgasClientHibImpl`, en donde debe especificarse el nombre del servicio para este ejemplo se escribe de la siguiente forma `com.zte.zxup10.client.npApplication.CgasClientHibImpl`.

Figura 49. Configuración de la clase principal para el servicio NP



6.3.3. Paso 3. Configuración del archivo pch.properties y log4j.properties.

El archivo de configuración para el arranque del servicio es pch.properties. La información contenida en este archivo incluye:

JDBC_DRIVER=com.sybase.jdbc2.jdbc.SybDriver #pch //Driver de la base de datos configurada

DB_URL=jdbc:sybase:Tds:10.40.58.76:4100 #pch //URL de la base de datos configurada

PCH_ID=pch_for_app # //pch_id utilizado en el servicio

SVC_IPADDRESS=10.0.1.200 #dirección IP del parlay gw

El archivo de configuración es log4j.properties el cual se encuentra localizado en el directorio del servicio clases debe ser modificado, y para el caso del servicio NP (Number Portability), se debe adicionar la siguiente información en el archivo de configuración:

#NP

log4j.logger.NPLog=INFO, NPLog, NPA # Use the INFO level for debugging

log4j.appender.NPLog=org.apache.log4j.RollingFileAppender

```

log4j.appender.NPLog.MaximumFileSize=5242880
log4j.appender.NPLog.MaxBackupIndex=9
log4j.appender.NPLog.File=f:/servicelog/NPLog # Log file position
log4j.appender.NPLog.layout=com.zte.zxup10.util.logger.logFormatter
log4j.appender.NPA1=org.apache.log4j.ConsoleAppender
log4j.appender.NPA1.layout=com.zte.zxup10.util.logger.logFormatter

```

6.3.4. Paso 4. Desarrollo de código del servicio NP. El servicio no necesita guardar ningún dato, por lo cual la estructura del programa es muy sencilla. Este servicio utiliza la clase `npApplication.java` para su implementación.

El servicio utiliza los siguientes métodos:

- `ReportNotificationProcess()`: Es la llamada de notificación de eventos desde el PCH, los parámetros de entrada incluyen la sesión con PCH, y variables que tienen que ver con información de dirección de la llamada. Las llamadas se inician en este punto. En este método el programa solicita y obtiene el número actual desde la base de datos de acuerdo a la dirección de la llamada e invoca `routerreq` en el PCH para enrutar y hacer la transferencia de la llamada.
- `ServiceAvailable()`: Notifica cuando el servicio empieza o es activado
- `serviceAvailbale()`: Notifica cuando el servicio es desactivado
- `CallEndedProcess()`: Es la llamada de retorno del PCH cuando la llamada es terminada, en donde el servicio libera la sesión y el medio

A continuación se puede observar el código para el servicio NP y su interpretación:

```

package com.zte.zxup10.np;
import java.util.*;
import java.sql.*;
import org.csapi.cc.*;
import org.csapi.ui.*;
import com.zte.zxup10.client.*;
import com.zte.zxup10.util.logger.*;
import org.csapi.*;

public class npApplication
extends DummyChargingMPApplication
implements Runnable
{

```

```

//Inicialización

public boolean startup()
{
    _instance = new npApplication();
    (new Thread(_instance)).start();
    ...
}

public void run() //complete el inicio del servicio

{
    ...
}
//PCH MPregistra la llamada de notificación de un evento. PCH crea una
sesion.

public void reportNotificationProcess(MPSession thisSession,
                                     TpCallNotificationInf anotificationinfo)

{
//Le asigna una identificación (ID) a la sesión

    String pchSessionID = String.valueOf(thisSession.getCallSessionID());

//Dirección de la llamada

    String origAddr =
        anotificationinfo.CallNotificationReportScope.OriginatingAddress.
        AddrString;

//Dirección llamada, el número NP marcado

    String destAddr =
        anotificationinfo.CallNotificationReportScope.DestinationAddress.
        AddrString;
try
{

String rediret = getDataStore(destAddr); // Número de destino obtenido desde
la BD

if(rediret == null || rediret.length()==0)
rediret = destAddr;
sendRoute(thisSession,rediret); // Enrutamiento

}

```

```

catch (Exception e)
{
return;
}
}

//Enrutamiento a través de la transferencia de la llamada

private void sendRoute(MPSession session, String redirectedNum)
{
    String sessionId = String.valueOf(session.getCallSessionID());

    try
    {
        ...
        //Invoca una SCF a través de una sesión, se inicia la conexión de la
        llamada por medio de routeReq a través del parámetro targetAddress

        session.routeReq(0, targetAddress, originatingAddress, infoSet,
            ex_ConProp);

    }

    catch (Exception ex3)
    {
    try
    {
        session.release(); //Termina una sesion irregular

session.clean(); //Clean the session
    }
    }
}

...
//PCH termina la llamada. Libera y borra la sesión.
public void callEndedProcess(MPSession t_Session,
    org.csapi.cc.TpCallEndedReport t_Report)

{

    String pchSessionID = String.valueOf(t_Session.getCallSessionID());
    t_Session.release();
    t_Session.clean();
}

```

```

    }

    //Activación del servicio

    public void serviceAvailable(String serviveType)
    {
        ...
    }

    //Desactivación del servicio

    public void serviceUnavailable(String serviveType)
    {
        Service_Available = false;
        ...
    }
}

```

6.3.5. Interpretación del código del servicio NP

- **Clase**

```

public class npApplication
extends DummyChargingMPApplication
implements Runnable

```

Para el caso de la aplicación el nombre de la clase es npApplication y es la que permite la inicialización de la llamada y el control de flujo del servicio.

- **Métodos**

Startup

```

public boolean startup()

```

Con el metodo startup el servicio es inicializado

run

```

public void run()

```

El método run active el servicio permitiendo que empiece a correr para este caso activa el servicio NP.

reportNotificationProcess


```
public void reportNotificationProcess(MPSession thisSession,  
TpCallNotificationInfo anotificationinfo)
```

PCH registra las llamadas de notificación de eventos y crea una sesión

Utiliza los parámetros thisSession para crear la sesión y anotificationinfo para notificación de eventos.

callEndedProcess

```
public void callEndedProcess(MPSession t_Session,  
org.csapi.cc.TpCallEndedReport t_Report)
```

PCH reporta cuando la llamada ha terminado. El servicio se completa y genera un CDR

Utiliza los parámetros t_session y t_report

serviceAvailable

```
public void serviceAvailable(String serviveType)
```

Al inicializar el servicio PCH notifica que la activación del servicio ha sido exitosa y que se ha obtenido la SCF de forma satisfactoria

Utiliza el parámetro serviveType para indicar el tipo de servicio

serviceUnavailable

```
public void serviceUnavailable(String serviveType)
```

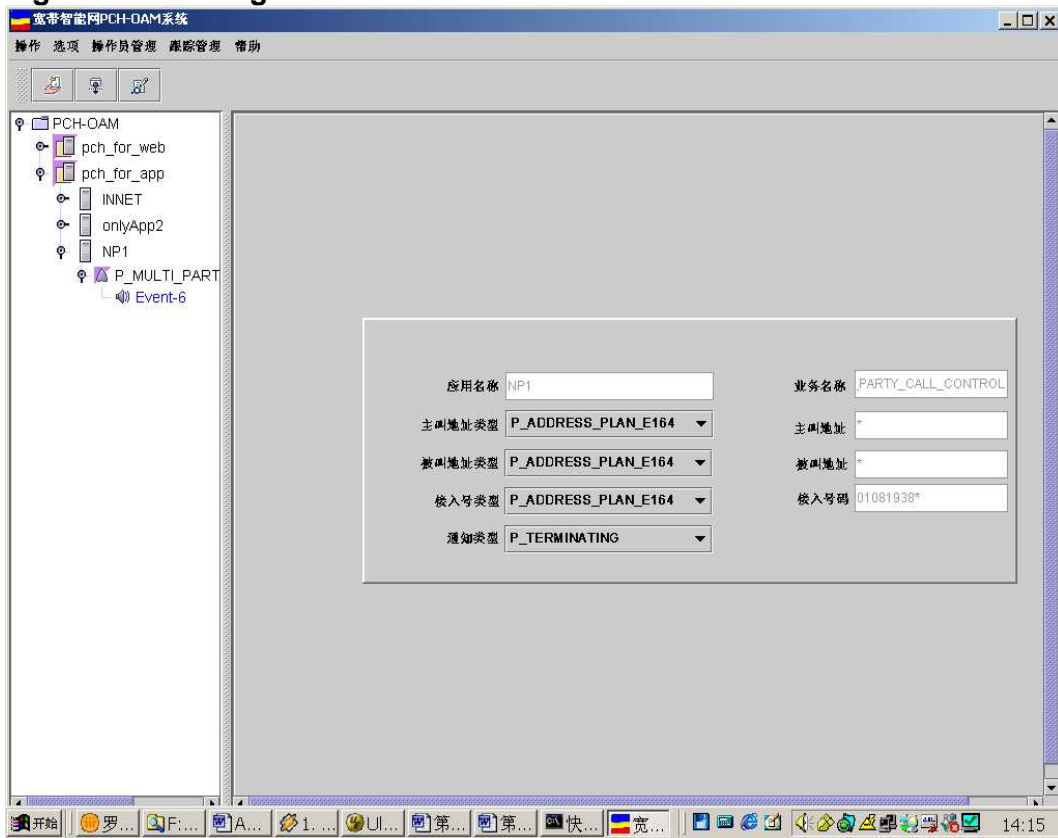
Una notificación es enviada cuando la SCF es inválida

Utiliza el parámetro serviveType para indicar el tipo de servicio

6.4. VERIFICACIÓN DEL SERVICIO

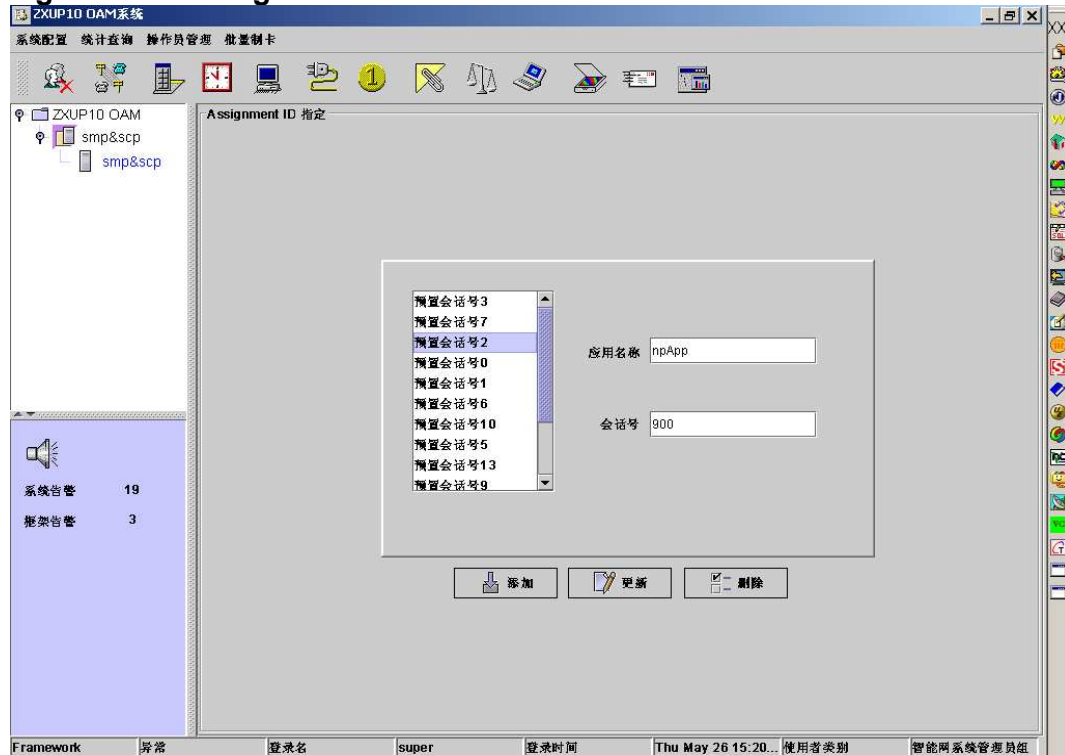
Se configura el acceso al servicio NP a través de PCH-OAM. El servicio NP utiliza solamente MPCC. El servicio es verificado a través de un número de segmento que para este caso es 01081938. En la siguiente figura se muestra la configuración.

Figura 50. Configuración de PCH-OAM



En la anterior grafica podemos observar que se configura el nombre del servicio, que utiliza MPCC y el número del segmento para verificar el servicio. También se asigna un número de sesión Call Control (CC) para el servicio NP en la consola OAM.

Figura 51. Configuración de OAM



Se configura la correspondencia entre el número NP del usuario y el número actual en la base de datos del servicio.

Cuando PCH es inicializado de igual forma los archivos de registro (log's) y herramientas de control de flujo también son activados.

El flujo normal de datos es: el llamante marca un número NP. La llamada es transferida al número actual. La llamada termina normalmente cuando una de las dos partes cuelga. A continuación en la siguiente grafica se muestra los mensajes del flujo normal de información por medio de mtrace.

Figura 52. Flujo de información durante una llamada en el servicio NP

The screenshot shows the Zxin10 3.50 message monitoring interface. The main window displays a log of events with columns for sequence number, session ID, event type, event code, event name, message content, and time. The log shows a sequence of SIP and network events related to a call, including options, notifications, and route changes.

序号	会...	事...	事件...	事件...	消息内容	时间
0	3425	0	SIP	TRACE	rev OPTIONS_MTD ,SS_LINKTEST !!!!	2005-05-23 10:15
1	0	60003	CC	TRACE	ConfCallControlManager::ConfCallControlManager(AppID=odaApp_ljf)	2005-05-23 10:15
2	0	60003	CC	TRACE	MultiPartyCallControlManager::MultiPartyCallControlManager(AppID=odaApp_ljf)	2005-05-23 10:15
3	0	60003	CC	TRACE	MultiPartyCallControlManager::createNotification	2005-05-23 10:15
4	0	60003	CC	TRACE	MultiPartyCallControlManager::createNotification	2005-05-23 10:15
5	3425	10001	SIP	TRACE	SIPMSG OUT:OPTIONS sip:1234567890@202.202.91.99;user=phone SIP/2.0Via: SIP/2.0/UDP 202.20...	2005-05-23 10:15
6	3425	10001	SIP	TRACE	SIPMSG OUT:OPTIONS sip:1234567890@202.202.91.99;user=phone SIP/2.0Via: SIP/2.0/UDP 202.20...	2005-05-23 10:15
7	3425	10001	SIP	TRACE	SIPMSG OUT:OPTIONS sip:1234567890@202.202.91.99;user=phone SIP/2.0Via: SIP/2.0/UDP 202.20...	2005-05-23 10:15
8	3426	10002	SIP	TRACE	SIPMSG IN:INVITE sip:0108193700@202.202.91.193;user=phone;sc=200 SIP/2.0Via: SIP/2.0/UDP 20...	2005-05-23 10:15
9	27917	60003	CC	TRACE	Receive ADDRESSEVENTNOTIFY: src=[0105160004], dest=[0108193700]	2005-05-23 10:15
10	3426	10001	SIP	TRACE	SIPMSG OUT:SIP/2.0 100 TryingVia: SIP/2.0/UDP 202.202.91.55:5060;branch=4af40425.0Via: SIP/2....	2005-05-23 10:15
11	6	60003	CC	TRACE	Call IpAppMultiPartyCallControlManager.reportNotification, AssignmentID=3014	2005-05-23 10:15
12	6	60003	CC	TRACE	MultiPartyCallLeg::routeReq, LegID=0x601	2005-05-23 10:15
13	6	60003	CC	TRACE	CallLeg::route, tryingChangeBridged =false, LegID=0x601	2005-05-23 10:15
14	6	60003	CC	TRACE	Send Route to Adapter_83, LegID=0x601	2005-05-23 10:15
15	3426	10001	SIP	TRACE	SIPMSG OUT:SIP/2.0 302 Moved TemporarilyVia: SIP/2.0/UDP 202.202.91.55:5060;branch=4af40425...	2005-05-23 10:15
16	3426	62002	SIP	TRACE	EventReportRes	2005-05-23 10:15
17	6	60003	CC	TRACE	Receive EVENTREPORTRES: eventType=6, LegID=0x601	2005-05-23 10:15
18	0	60003	CC	TRACE	Leg release, legSessionID= 0x601	2005-05-23 10:15
19	6	60003	CC	TRACE	Send Release to Adapter_83, LegID=0x601	2005-05-23 10:15
20	6	60003	CC	TRACE	Call IpAppMultiPartyCall.callEnded	2005-05-23 10:15
21	3426	10002	SIP	TRACE	SIPMSG IN:ACK sip:0108193700@202.202.91.193;user=phone;sc=200 SIP/2.0Via: SIP/2.0/UDP 202.2...	2005-05-23 10:15
22	3425	10001	SIP	TRACE	SIPMSG OUT:OPTIONS sip:1234567890@202.202.91.99;user=phone SIP/2.0Via: SIP/2.0/UDP 202.20...	2005-05-23 10:16

At the bottom of the window, there is a status bar with the text: 就绪 消息数:0接收流:0 [SLP未注] [CSCSM未注] [SIM未注] [CORB未注] [SIU未注] [Inap EVE] [Inap TCAP] [Cap EVENT未注] [Cap TCAP未注] [SMIT未注] [RADIUS未注] [SCS未注]

7. CONCLUSIONES

Gracias a la tecnología OSA/Parlay el despliegue de servicios puede hacerse de forma más rápida y sencilla, lo que permite una amplia variedad de servicios, un amplio campo para los desarrolladores de aplicaciones y una disminución en cuanto a costos para los operadores, pues la gran cantidad de servicios que se pueden ofrecer a través de las redes fijas y móviles permitirá escoger de un extenso mercado de nuevas soluciones que prácticamente se acomoden a las necesidades de cada uno de los usuarios.

La tecnología OSA/Parlay ha tenido un alto grado de aceptación, gracias a que ha madurado rápidamente y hay una gran tendencia a desarrollar servicios con esta tecnología en la industria.

Existen diferentes tipos de tecnologías para el desarrollo de servicios, la mayoría de ellas dedicadas a un tipo de servicio específico, lo que marca la gran diferencia con la tecnología OSA/Parlay pues esta última permite el desarrollo de muchos tipos de servicios, para voz, texto, audio, video, movilidad, servicios web entre otros y funciona para redes fijas y móviles por lo cual es una tecnología muy completa y segura.

PCH es quien provee todas las funciones necesarias para la comunicación entre la aplicación y el parlay Gateway de tal forma que las aplicaciones no necesitan conectarse directamente con el parlay Gateway ya que proporciona interfaces más sencillas para que puedan ser usadas por el programador. PCH controla las funciones de la aplicación, maneja los log's (registros) del servicio y tiene conexión directa con la base de datos en las aplicaciones que la requieran, es decir, todas las operaciones referentes a la aplicación y el Parlay Gateway se realizan a través de él sin tener acceso directo a la red.

Para el desarrollo de los servicios el programador debe tener claro el tipo de aplicación o el tipo de servicio que desea desarrollar pues cada una de las SCF's en el Parlay/Gateway aplica para diferentes funciones y sus métodos deben ser llamados a través de PCH el cual ofrece las interfaces de comunicación entre las aplicaciones y el Parlay Gateway. Estas interfaces deben estar incluidas dentro de la lógica del servicio por lo tanto es importante saber antes hacia donde está orientado el servicio e incluir las interfaces necesarias.

Este proyecto de investigación sirvió para definir y establecer las herramientas y consideraciones que se deben tener en cuenta para el desarrollo e implementación de los servicios en la plataforma multiservicios de EMCALI, siendo un avance importante para iniciar el desarrollo de servicios pues es una guía para el desarrollador que define todo el proceso a seguir para ejecutar un servicio.

BIBLIOGRAFÍA

AUBURN, RJ. Voice Browser Call Control: CCXML 1.0 [en línea]. Estados Unidos, 2004. [Consultado 28 de abril de 2008]. Disponible en Internet: <http://www.voxeo.com/>

CAICEDO RENDON, Oscar Mauricio. Parlay/OSA [en línea]. Popayán: Universidad del Cauca, 2003. [Consultado 08 de febrero de 2008]. Disponible en Internet: <http://wapcolombia.unicauca.edu.co/documentos/osaparlay.ppt>

CARMONA VITOLA, Marcela. Diseño e implementación de servicios telemáticos sobre la red inteligente NGN de EMCALI. Santiago de Cali, 2007. 256 p. Trabajo de grado (Ingeniera Informática). Universidad Autónoma de Occidente. Facultad de Ingenierías

COHEN, Tracy; HERNEDEZ, Janet. El camino hacia las redes de próxima generación-pioneros [en línea]. Estados Unidos: Unión Internacional de Telecomunicaciones, 2008. [Consultado 05 de febrero de 2008]. Disponible en Internet: <http://www.itu.int/itunews/manager/display.asp?lang=es&year=2007&issue=02&ipage=next-generation2&ext=html#top>

EMCALI EICE ESP. Descripción de la Red Existente. Santiago de Cali, 2008. 1 archivo de computador.

FALCARIN Paolo; LICCIARDI, Carlos Alberto. Analysis of NGN service creation technologies [en línea]. Torino: Politécnico di Torino. [Consultado 12 de marzo de 2008]. Disponible en Internet: <http://softeng.polito.it/falcarin/docs/Annals03.pdf>

GARCIA ALVAREZ, Víctor. Estándar VoiceXML [en línea]. Asturias: Universidad de Oviedo, 2004. [Consultado 28 de abril de 2008]. Disponible en Internet: <http://www.di.uniovi.es/~cueva/asignaturas/doctorado/2006/trabajos/VoiceXML.pdf>

GARCIA YAGUE, Adolfo. VoiceXML, CCXML y SALT Convergencia voz y aplicaciones [en línea]. Sevilla: Centro de investigaciones científicas Isla de la Cartuja, 2004. [Consultado en abril 25 de 2008]. Disponible en internet: <http://80.34.206.133/netica/com/convergencia-voz-app-v1.2.pdf>

Guía del usuario del sistema: Sistema operativo y dispositivos [en línea]. California: International Business Machines Corporation, 2002. [Consultado 17 de junio de 2008]. Disponible en internet:

High Availability Cluster Multi-Processing for AIX Concepts and facilities guide [en línea]. Estados Unidos: International Business Machines Corporation, 2004. [Consultado 17 de junio de 2008]. Disponible en internet: <http://publib.boulder.ibm.com/infocenter/clresctr/vxrx/index.jsp?topic=/com.ibm.ccluster.hacmp.doc/hacmpbooks.html>

Migración a NGN [en línea]. Bogotá D.C: Centro de Investigación de las Telecomunicaciones CINTEL, 2003. [Consultado 07 de febrero de 2008]. Disponible en Internet: <http://www.cintel.org.co:8080/cintel/opencms/cintel/inicio/lineas/linea/migracion.html>

MUÑOZ ORGANERO, Mario. SCMM: Metamodelo para la creación de aplicaciones en redes de siguiente generación. Madrid, 2003. 276 p. Tesis doctoral (Ingeniería telemática). Universidad Carlos III de Madrid. Departamento de ingeniería telemática

Página web de EMCALI [en línea]. Santiago de Cali. EMCALI. 2007. [Consultado 05 de febrero de 2008]. Disponible en Internet: <http://www.emcali.com.co/vsmPC/bin/smRenderFS.php?PHPSESSID=d6d190dc0a09adbf7967e1cc3cbed2a&cerror=&xnode=5>

POPOFF, Jeff. Parlay/OSA 101 [en línea]. Boston: The Parlay Group, 2005. [Consultado 23 de febrero de 2008]. Disponible en Internet: <http://www.parlay.org>

PRELEC, Dussan; RAYMON Ken. Implementing High Availability Cluster Multi-Processing (HACMP) cookbook [en línea]. Estados Unidos: International Business Machines Corporation, 2005. [Consultado 17 de junio de 2008]. Disponible en internet: http://www-03.ibm.com/systems/p/support/techdocs/clusters_aix.html

Telefónica I+D. Las telecomunicaciones y la movilidad en la sociedad de la información [en línea]. Madrid: División de relaciones corporativas y comunicación de telefónica I+D, 2005. [Consultado en enero 30 de 2008]. Disponible en internet: http://www.telefónica.es/sociedaddelainformacion/html/publicaciones_movilidad.shtml

ZTE. Introduction to Parlay Client Hub (PCH). Santiago de Cali, 2008. 1 archivo de computador.

_____. NP Service Analysis. Santiago de Cali, 2008. 1 archivo de computador

_____. Plataforma Unificada de Servicios ZXUP10. Santiago de Cali, 2005. 1 archivo de computador

_____. Propuesta técnica NGN de ZTE para EMCALI. Santiago de Cali, 2005. 1 archivo de computador

_____. Propuesta técnica plataforma de servicios. Santiago de Cali, 2005. 1 archivo de computador

_____. UP10 Parlay API specification. Santiago de Cali, 2005. 1 archivo de computador

_____. Receiving-End access Code interpretation. Santiago de Cali, 2005. 1 archivo de computador

_____. Service Deployment and verification. Santiago de Cali, 2005. 1 archivo de computador

ANEXOS

Anexo A. Especificación de requerimientos para las funciones de administración de los servicios

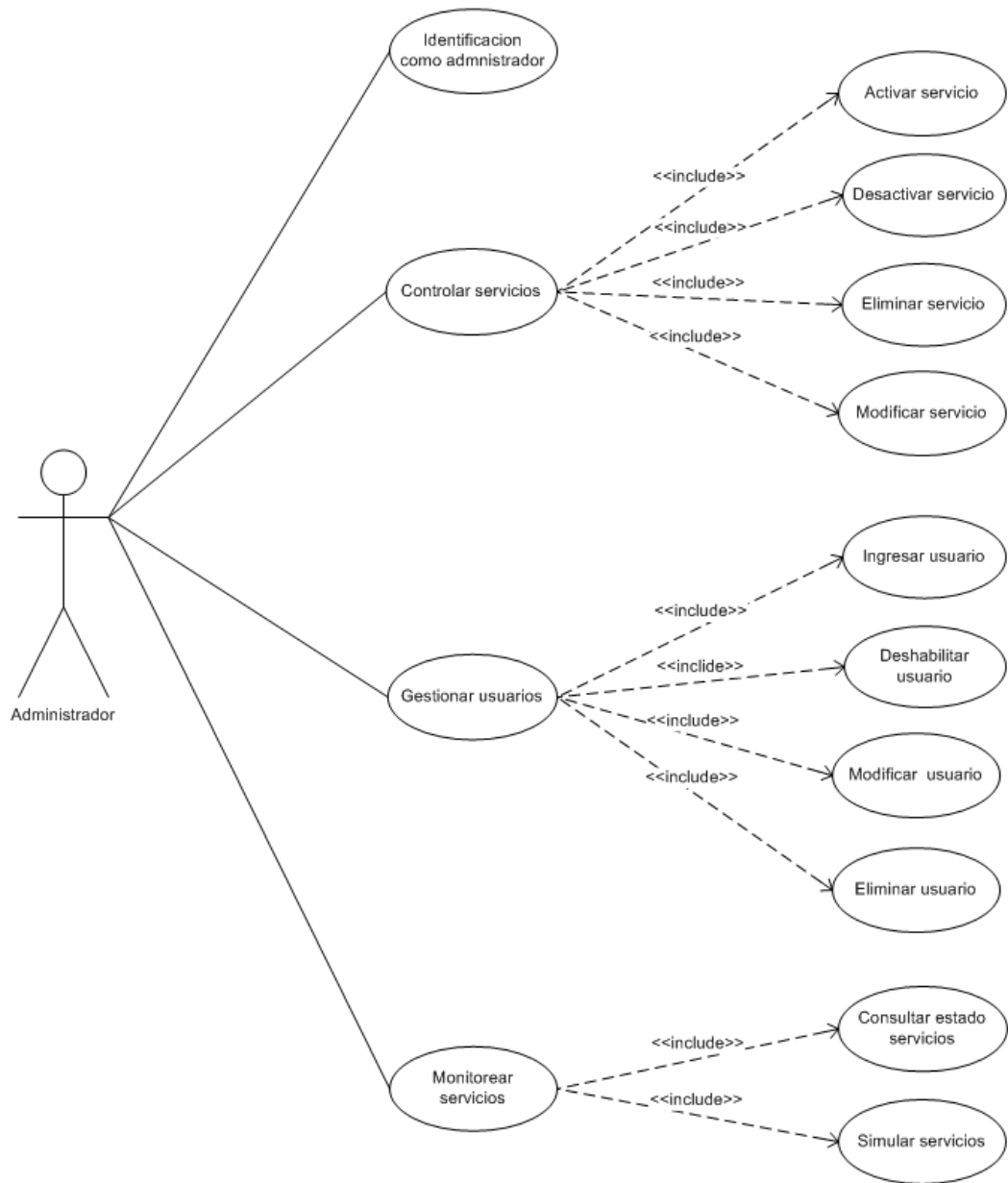
DEFINICIÓN DEL ACTOR

Actor	Administrador
Descripción	Es el actor principal para la administración de los servicios, es quién los va a configurar y va a ejercer control y gestión sobre ellos, de igual forma, realiza el control sobre los usuarios de un determinado servicio es quien les asigna privilegios y restricciones en cuanto al uso estos.

LISTA DE CASOS DE USO

- CU #1:** Identificación como administrador
- CU #2:** Controlar servicios
- CU #3:** Activar servicio
- CU #4:** Desactivar servicio
- CU #5:** Eliminar servicio
- CU #6:** Modificar servicio
- CU# 7:** Gestionar usuarios
- CU #8:** Ingresar usuario
- CU #9:** Deshabilitar usuario
- CU #10:** Modificar usuario
- CU #11:** Eliminar usuario
- CU #12:** Monitorear servicios
- CU #13:** Consultar estado de los servicios
- CU #14:** Simular servicios

DIAGRAMA DE CASOS DE USO



DESCRIPCIÓN DE LOS CASOS DE USO

CU #1	Identificación como administrador	
Actor	Administrador	
Precondición	El administrador debe tener asignado un usuario y una contraseña para tener privilegios como administrador	
Descripción	El administrador para ingresar al sistema y poder realizar modificaciones sobre los servicios debe validarse como administrador en el sistema	
Flujo de eventos	<ul style="list-style-type: none"> • El caso de uso inicia cuando el administrador del sistema selecciona la opción ingresar como administrador • El sistema solicita al administrador un nombre de usuario y una contraseña • El administrador ingresa los datos • El sistema valida los datos ingresados • El sistema indica que los datos ingresados son correctos • El administrador puede controlar los servicios, los perfiles de usuarios y monitorear los servicios 	
Postcondición	Si los datos ingresados son validos puede realizar las tareas de administrador	
Excepciones	Situación	Respuesta
	En el caso que la identificación del administrador no sea valida	El sistema debe responder con un mensaje de error y no permite el ingreso al sistema

CU #2	Controlar servicios	
Actor	Administrador	
Precondición	El nombre de usuario y contraseña del administrador deben ser validos	
Descripción	A partir del momento en que el administrador ha sido validado puede realizar diferentes tareas entre las que se encuentra el control de los servicios, montarlos en la red, modificarlos, deshabilitarlos o eliminarlos.	
Flujo de eventos	<ul style="list-style-type: none"> • El caso de uso inicia cuando el administrador del sistema selecciona la opción controlar servicios • El administrador selecciona de una lista la acción que desea ejecutar sobre un servicio, ya sea activarlo, desactivarlo, modificarlo o eliminarlo • El administrador debe escoger el servicio a modificar de una lista o debe buscar el servicio en caso que desee activar uno nuevo • El administrador realiza una de las opciones de control de los servicios • El administrador guarda los cambios o el nuevo estado del servicio 	

Postcondición	Los cambios hechos sobre los servicios quedan guardados dentro de su configuración y pueden modificar su funcionamiento	
Excepciones	Situación	Respuesta
	Al final de la configuración del servicio no se presiona la opción guardar cambios	El servicio no será alterado y su funcionamiento seguirá igual

CU #3	Activar servicio	
Actor	Administrador	
Precondición	El servicio debe estar previamente simulado y probado para ser activado	
Descripción	Este caso de uso describe como el administrador del sistema activa un servicio y lo pone en funcionamiento dentro de la red	
Flujo de eventos	<ul style="list-style-type: none"> • Este caso de uso inicia cuando el administrador de la red selecciona la opción activar servicio • El sistema debe mostrar una pantalla donde debe indicar el nombre del servicio y su ubicación • El administrador debe registrar el servicio en la consola de manejo de los servicios OAM • El administrador oprime la opción de activar servicio • El sistema pregunta si realmente se quiere activar el servicio • El administrador confirma o no si desea activar el servicio • El sistema guarda el estado del servicio • El sistema muestra un aviso de servicio correctamente activado • El servicio es puesto en marcha 	
Postcondición	Los usuarios pueden acceder a los nuevos servicios activados e interactuar con ellos	
Excepciones	Situación	Respuesta
	Al final de la configuración del servicio no se presiona la opción guardar cambios	El servicio no será alterado y su funcionamiento seguirá igual

CU #4	Desactivar servicio	
Actor	Administrador	
Precondición	El servicio debe estar activo	
Descripción	Este caso de uso describe como el administrador del sistema desactiva un servicio y restringe su uso	
Flujo de eventos	<ul style="list-style-type: none"> • Este caso de uso inicia cuando el administrador del sistema elige la opción desactivar servicio • El sistema debe mostrar una lista con los servicios que se encuentran 	

	activos <ul style="list-style-type: none"> • El administrador debe escoger uno de los servicios de la lista • El sistema debe mostrar información referente al servicio seleccionado • El administrador oprime la opción desactivar servicio • El sistema debe de mostrar un aviso que confirme que realmente se quiere desactivar el servicio • El administrador confirma o no si desea desactivar el servicio • El sistema guarda el nuevo estado del servicio • El servicio es desactivado • El sistema muestra un aviso de servicio correctamente desactivado 	
Postcondición	Los usuarios no pueden acceder al servicio desactivado.	
Excepciones	Situación	Respuesta
	Al final de la configuración del servicio no se presiona la opción guardar cambios	El servicio no será alterado y su funcionamiento seguirá igual

CU #5	Eliminar servicio
Actor	Administrador
Precondición	El servicio debe estar activo
Descripción	Este caso de uso describe como el administrador del sistema elimina un servicio activo en el sistema
Flujo de eventos	<ul style="list-style-type: none"> • Este caso de uso se inicia cuando el administrador del sistema escoge la opción eliminar servicio • El sistema debe mostrar una lista con los servicios que se encuentran activos • El administrador debe escoger uno de los servicios de la lista • El sistema debe mostrar información referente al servicio seleccionado • El administrador oprime la opción eliminar servicio • El sistema debe de mostrar un aviso que confirme que realmente se quiere eliminar el servicio • El administrador confirma o no si desea eliminar el servicio • El sistema guarda el nuevo estado del servicio • El servicio es eliminado

	<ul style="list-style-type: none"> El sistema muestra un aviso de servicio correctamente eliminado 	
Postcondición	Los usuarios no pueden acceder al servicio eliminado.	
Excepciones	Situación	Respuesta
	Al final de la configuración del servicio no se presiona la opción guardar cambios Si se desea recuperar o restablecer el servicio	El servicio no será alterado y su funcionamiento seguirá igual El sistema mostrará un mensaje de error en donde se notifica que el servicio ha sido eliminado

CU #6	Modificar servicio	
Actor	Administrador	
Precondición	El servicio debe estar activo	
Descripción	Este caso de uso describe como el administrador del sistema modifica un servicio un servicio activo en el sistema	
Flujo de eventos	<ul style="list-style-type: none"> Este caso de uso inicia cuando el administrador elige la opción modificar servicio El sistema debe mostrar una lista con los servicios que se encuentran activos El administrador debe escoger uno de los servicios de la lista El sistema debe mostrar información referente al servicio seleccionado El sistema debe mostrar una pantalla donde se deben especificar los parámetro del servicio a modificar El administrador realiza las modificaciones El administrador oprime la opción modificar servicio El sistema debe de mostrar un aviso que confirme que realmente se quiere hacer las modificaciones el servicio El administrador confirma o no si desea hacer las modificaciones el servicio El sistema guarda las modificaciones del servicio El servicio es modificado El sistema muestra un aviso de modificación exitosa 	
Postcondición	El servicio sigue su funcionamiento con los cambios realizados	
Excepciones	Situación	Respuesta
	Si alguna modificación no es correcta Al final de la configuración del servicio no se presiona la opción guardar cambios	El sistema debe indicar al administrador en que parámetro ha cometido algún error El servicio no será alterado y su funcionamiento seguirá igual

CU #7	Gestionar usuarios	
Actor	Administrador	
Precondición	El nombre de usuario y contraseña del administrador deben ser validos	
Descripción	A partir del momento en que el administrador ha sido validado puede realizar diferentes tareas entre las que se encuentra manejar los perfiles de usuario, permitiéndole ingresar los usuarios, desactivarlos, modificarlos o eliminarlos	
Flujo de eventos	<ul style="list-style-type: none"> • El caso de uso inicia cuando el administrador del sistema selecciona la opción manejar perfiles de usuario • El sistema solicita al administrador los datos de usuario • El administrador ingresa los datos del usuario • El administrador selecciona de una lista la acción que desea ejecutar sobre un usuario, ya sea activarlo, desactivarlo, modificarlo o eliminarlo • El administrador realiza una de las opciones de control de los servicios • El administrador guarda los cambios o el nuevo estado del servicio 	
Postcondición	Todos los cambios realizados sobre los perfiles de los usuario quedan grabados	
Excepciones	Situación	Respuesta
	Al final de la configuración del perfil de usuario no se presiona la opción guardar cambios	El perfil del usuario no será alterado y su funcionamiento seguirá igual

CU #8	Ingresar usuario	
Actor	Administrador	
Precondición	Se deben tener los datos necesarios para la creación de los perfiles de usuario	
Descripción	Este caso de uso describe como el administrador del sistema puede ingresar un usuario al sistema para que pueda hacer uso de los diferentes servicios	
Flujo de eventos	<ul style="list-style-type: none"> • El caso de uso inicia cuando el administrador del sistema dentro de la opción manejar perfiles de usuario selecciona la opción ingresar usuario • El sistema solicita al administrador los datos de usuario • El administrador ingresa los datos del usuario • El administrador oprime la opción guardar los datos de usuario • El sistema en una pantalla pregunta si realmente se desea ingresar el perfil de usuario • El administrador confirma o no si desea ingresar el perfil del usuario • El sistema verifica que la información ingresada del usuario sea 	

	<p>correcta</p> <ul style="list-style-type: none"> • El sistema guarda la información del usuario • El sistema muestra en una pantalla que el usuario ha sido ingresado de forma correcta 	
Postcondición	El usuario puede obtener los permisos para utilizar un servicio o varios	
Excepciones	Situación	Respuesta
	Al final de la configuración de ingresar usuario no se presiona la opción guardar cambios	El perfil del usuario no podrá ser ingresado y no podrá utilizar algún servicio

CU #9	Deshabilitar usuario	
Actor	Administrador	
Precondición	El perfil del usuario debe estar activo	
Descripción	Este caso de uso describe como el administrador del sistema puede deshabilitar un usuario activo en el sistema	
Flujo de eventos	<ul style="list-style-type: none"> • El caso de uso inicia cuando el administrador del sistema dentro de la opción manejar perfiles de usuario selecciona la opción deshabilitar usuario • El sistema solicita al administrador los datos de usuario • El administrador ingresa los datos del usuario • El administrador oprime la opción deshabilitar usuario • El sistema muestra en los datos del usuario a deshabilitar • El sistema en una pantalla pregunta si realmente se desea deshabilitar el perfil de usuario • El administrador confirma o no si desea deshabilitar el perfil del usuario • El sistema verifica que la información ingresada del usuario sea correcta • El sistema guarda de los cambios realizados • El sistema muestra en una pantalla que el usuario ha sido deshabilitado de forma correcta 	
Postcondición	El usuario no puede utilizar ningún servicio	
Excepciones	Situación	Respuesta
	Al final de la configuración de deshabilitar usuario no se presiona la opción guardar cambios	El perfil del usuario no podrá ser deshabilitado y los cambios no serán activados

CU #10	Modificar usuario	
Actor	Administrador	
Precondición	El perfil del usuario debe estar activo	
Descripción	Este caso de uso describe como el administrador del sistema puede modificar el perfil de un usuario activo en el sistema	
Flujo de eventos	<ul style="list-style-type: none"> • El caso de uso inicia cuando el administrador del sistema dentro de la opción manejar perfiles de usuario selecciona la opción modificar perfil de usuario • El sistema solicita al administrador los datos de usuario • El administrador ingresa los datos del usuario • El administrador oprime la opción modificar usuario • El sistema muestra los datos de usuario que se pueden modificar • El administrador modifica los datos • El administrador oprime guardar los cambios realizados • El sistema en una pantalla pregunta si realmente se desea modificar el perfil de usuario • El administrador confirma o no si desea modificar el perfil del usuario • El sistema verifica que la información ingresada del usuario sea correcta • El sistema guarda de los cambios realizados • El sistema muestra en una pantalla que el usuario ha sido modificado de forma correcta 	
Postcondición	La información de usuario ha sido modificada	
Excepciones	Situación	Respuesta
	Al final de la configuración de modificar perfil de usuario no se presiona la opción guardar cambios Si alguna modificación no es correcta	El perfil del usuario no podrá ser modificado y los cambios no serán activados El sistema debe indicar al administrador en que parámetro ha cometido algún error

CU #11	Eliminar usuario	
Actor	Administrador	
Precondición	El perfil del usuario debe estar activo	
Descripción	Este caso de uso describe como el administrador del sistema puede eliminar el perfil de un usuario activo en el sistema	
Flujo de eventos	<ul style="list-style-type: none"> • El caso de uso inicia cuando el administrador del sistema dentro de la opción manejar perfiles de usuario selecciona la opción eliminar perfil 	

	de usuario <ul style="list-style-type: none"> • El sistema solicita al administrador los datos de usuario • El administrador ingresa los datos del usuario • El administrador oprime la opción eliminar usuario • El sistema muestra los datos de usuario que se va a eliminar • El sistema en una pantalla pregunta si realmente se desea eliminar el perfil de usuario • El administrador confirma o no si desea eliminar el perfil del usuario • El sistema verifica que la información ingresada del usuario sea correcta • El sistema guarda de los cambios realizados • El sistema muestra en una pantalla que el usuario ha sido eliminado de forma correcta 	
Postcondición	El usuario no puede utilizar ningún servicio	
Excepciones	Situación	Respuesta
	Al final de la configuración de eliminar perfil de usuario no se presiona la opción guardar cambios Si se desea recuperar o restablecer el perfil del usuario	El perfil del usuario no podrá ser eliminado y los cambios no serán activados El sistema mostrará un mensaje de error en donde se notifica que el perfil de usuario ha sido eliminado

CU #12	Monitorear servicios
Actor	Administrador
Precondición	El nombre de usuario y contraseña del administrador deben ser validos
Descripción	Este caso de uso describe como el administrador del sistema puede monitorear los servicios, consultar en qué estado se encuentran y realizar simulaciones de los servicios antes de ponerlos en funcionamiento o para probarlos ante alguna modificación realizada.
Flujo de eventos	<ul style="list-style-type: none"> • El caso de uso inicia cuando el administrador del sistema selecciona la opción monitorear servicios • El sistema permite escoger entre dos opciones la consulta del estado del servicio y simular servicios • El administrador escoge una de las dos opciones • El administrador puede visualizar estado de los servicios o hacer simulaciones
Postcondición	El usuario solo puede visualizar el estado de los servicios

Excepciones	Situación	Respuesta

CU #13	Consultar estado de los servicios	
Actor	Administrador	
Precondición	Los servicios deben tener un sistema de gestión el cual permita obtener información del funcionamiento y estado de los servicio	
Descripción	Este caso de uso describe como el administrador del sistema puede consultar el estado de los servicios, si se encuentra activo, inactivo, o presenta fallas.	
Flujo de eventos	<ul style="list-style-type: none"> • El caso de uso inicia cuando el administrador del sistema dentro de la opción monitorear servicios selecciona la opción consultar estado de los servicios • El sistema en una pantalla muestra todos los servicios • El administrador debe escoger de la lista el servicio que va a ser monitoreado • El administrador puede visualizar las variables del servicio y las fallas • El administrador puede visualizar estadísticas de uso del servicio • El administrador puede visualizar el comportamiento del servicio • El administrador tiene la opción de salir en cualquier momento de la opción de monitorear servicios 	
Postcondición	El usuario solo puede visualizar el estado de los servicios	
Excepciones	Situación	Respuesta

CU #14	Simular servicios	
Actor	Administrador	
Precondición	El sistema debe contar con una plataforma para realizar simulaciones de aplicaciones	
Descripción	Este caso de uso describe como el administrador del sistema puede realizar simulaciones de servicios antes de activarlos en el sistema	
Flujo de eventos	<ul style="list-style-type: none"> • El caso de uso inicia cuando el administrador del sistema dentro de la opción monitorear servicios selecciona la opción simular servicios • El administrador debe descargar el servicio en la plataforma • El administrador debe escoger el servicio 	

	<ul style="list-style-type: none"> • El administrador debe oprimir la opción simular servicio • El administrador puede visualizar el comportamiento del servicio como si estuviera en condiciones reales • El administrador tiene la opción de terminar la simulación en cualquier momento 	
Postcondición	El usuario solo puede visualizar el comportamiento del servicio	
Excepciones	Situación	Respuesta

Anexo B. Especificación de requerimientos para la definición de nuevos servicios

DEFINICIÓN DE ACTORES

Actor	Administrador
Descripción	Es el actor principal para la administración de los servicios, es quién los va a configurar y va a ejercer control y gestión sobre ellos, de igual forma, realiza el control sobre los usuarios de un determinado servicio es quien les asigna privilegios y restricciones en cuanto al uso estos.

Actor	Ingeniero NOC (Network Operations Center)
Descripción	Es el actor encargado de administrar un servicio determinado dentro de una entidad, solo tiene ciertos privilegios para controlar y gestionar este servicio.

Actor	Usuario
Descripción	Son los actores que utilizan los servicios, son quienes han adquirido un servicio con EMCALI.

REQUERIMIENTOS PARA SISTEMA DE INFORMACIÓN Y VISUALIZACIÓN DE TRÁFICO VIAL

LISTA DE REQUERIMIENTOS FUNCIONALES

RF #1: El servicio debe permitir la visualización de las vías por medio de las cámaras instaladas en la ciudad

RF #2: El servicio debe trabajar con Internet

RF #3: El servicio debe permitir escoger a través de un menú una zona de la ciudad (norte, sur, oriente, occidente)

RF #4: El servicio debe permitir escoger a través de un menú que vía desea visualizar según la zona

RF #5: El servicio debe operar en tiempo real

RF #6: El servicio debe permitir al usuario la visualización del tráfico de la vía seleccionada

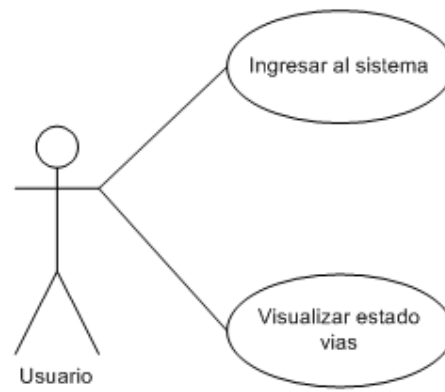
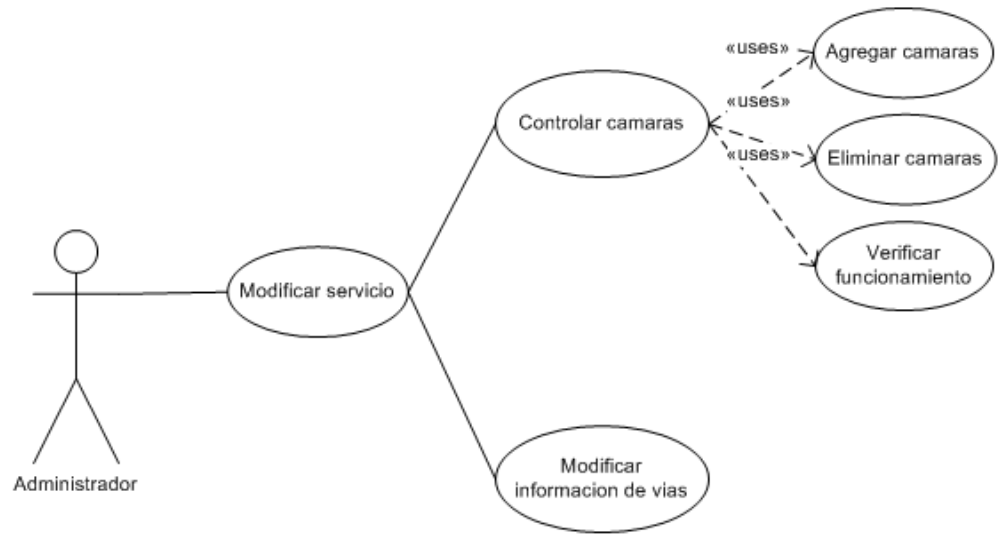
RF #7: El servicio debe permitir visualizar el tráfico en la vía durante un periodo determinado

RF #8: El servicio debe permitir hacer varias consultas

LISTA DE CASOS DE USO

- CU #1:** Modificar servicio
- CU #2:** Controlar cámaras
- CU #3:** Agregar cámaras
- CU #4:** Eliminar cámaras
- CU #5:** Verificar funcionamiento
- CU #6:** Modificar información de vías
- CU #7:** Ingresar al sistema
- CU #8:** Visualizar estado de las vías

DIAGRAMA DE CASOS DE USO



DESCRIPCIÓN DE LOS CASOS DE USO

CU #1	Modificar servicio	
Actor	Administrador	
Precondición	El administrador debe estar validado en el sistema con un nombre de usuario y contraseña	
Descripción	Este caso de uso describe como el administrador de servicios puede modificar el servicio controlando las cámaras o modificando la información de las vías	
Flujo de eventos	<ul style="list-style-type: none"> • El caso de uso inicia cuando el administrador del sistema selecciona la opción modificar servicio • El sistema pregunta el servicio que desea modificar • El administrador debe indicar el servicio de información y visualización de tráfico vial • El sistema solicita al administrador si desea controlar las cámaras o modificar la información de las vías • El administrador selecciona una de las dos opciones 	
Postcondición	El administrador modifica el servicio	
Excepciones	Situación	Respuesta

CU #2	Controlar cámaras	
Actor	Administrador	
Precondición	Deben existir cámaras instaladas en las principales vías de la ciudad	
Descripción	Este caso de uso describe como el administrador puede controlar las cámaras instaladas en las diferentes vías de la ciudad	
Flujo de eventos	<ul style="list-style-type: none"> • El caso de uso inicia cuando el administrador dentro de la opción modificar servicio selecciona controlar cámaras. • El sistema despliega en una lista las acciones que el administrador puede realizar, agregar cámara, eliminar cámara o verificar el funcionamiento de una cámara • El administrador selecciona la acción que desea realizar 	
Postcondición	El administrador puede controlar las cámaras	
Excepciones	Situación	Respuesta

CU #3	Agregar cámaras	
Actor	Administrador	
Precondición	Deben existir cámaras instaladas en las principales vías de la ciudad	
Descripción	Este caso de uso describe como el administrador puede agregar nuevas cámaras ubicadas en diferentes sitios de la ciudad	
Flujo de eventos	<ul style="list-style-type: none"> • Este caso de uso inicia cuando dentro de la opción controlar cámaras el administrador selecciona agregar cámaras • El sistema solicita información de la cámara como ubicación y un número de identificación • El administrador digita información de la cámara • El sistema pregunta si el administrador desea guardar la información • El administrador confirma o no si desea guardar la información • El sistema guarda la información ingresada 	
Postcondición	El usuario puede visualizar otras vías	
Excepciones	Situación	Respuesta
	En el caso que los datos ingresados por el administrador no sean guardados	El sistema no agrega las nuevas cámaras

CU #4	Eliminar cámaras	
Actor	Administrador	
Precondición	Deben existir cámaras instaladas en las principales vías de la ciudad	
Descripción	Este caso de uso como el administrador puede eliminar o desactivar cámaras ubicadas en algunas vías de la ciudad	
Flujo de eventos	<ul style="list-style-type: none"> • Este caso de uso inicia cuando dentro de la opción controlar cámaras el administrador selecciona eliminar cámaras • El sistema solicita información de la cámara como ubicación y un número de identificación • El administrador digita información de la cámara • El sistema pregunta si el administrador si desea eliminar la cámara seleccionada • El administrador confirma o no si desea eliminar la cámara seleccionada • El sistema pregunta si se desean guardar los cambios realizados • El administrador confirma o no si desea guardar la información • El sistema guarda la información 	

Postcondición	Se habilitan nuevas cámaras para la visualización de mas vías	
Excepciones	Situación	Respuesta
	En el caso que el administrador no guarde los cambios	El sistema seguirá operando con el mismo número de cámaras

CU #5	Verificar funcionamiento	
Actor	Administrador	
Precondición	Deben existir cámaras instaladas en las vías de la ciudad	
Descripción	Este caso de uso como el administrador puede verificar el funcionamiento de las cámaras instaladas en la ciudad	
Flujo de eventos	<ul style="list-style-type: none"> • Este caso de uso inicia cuando dentro de la opción controlar cámaras el administrador selecciona verificar funcionamiento • El sistema solicita información de la cámara como ubicación y un número de identificación • El administrador digita información de la cámara • El sistema en una pantalla muestra al administrador el estado (activo o inactivo) en que se encuentra la cámara y si presenta alguna falla 	
Postcondición	El administrador puede observar el funcionamiento de las cámaras	
Excepciones	Situación	Respuesta
	En el caso que una cámara tenga fallas	El administrador solo puede visualizar el estado de las cámaras no puede realizar ninguna modificación

CU #6	Modificar información de las vías	
Actor	Administrador	
Precondición	Deben existir cámaras instaladas en las principales vías de la ciudad	
Descripción	Este caso de uso como el administrador puede modificar información de las vías como direcciones o nombres	
Flujo de eventos	<ul style="list-style-type: none"> • Este caso de uso inicia cuando el administrador selecciona la opción modificar información de las vías • El sistema solicita información de la cámara como ubicación y un número de identificación • El administrador digita información de la cámara • El sistema muestra la información de la vía a la cual pertenece la cámara como dirección o nombre • El sistema pregunta si se desea modificar la información de la vía 	

	<ul style="list-style-type: none"> • El administrador confirma o no si desea modificar la información • El administrador realiza los cambios necesarios • El sistema pregunta si desea guardar los cambios realizados • El administrador confirma o no si desea guardar los cambios • El sistema guarda la información 	
Postcondición	La información acerca de las vías es modificada	
Excepciones	Situación	Respuesta
	En el caso que el administrador no guarde los cambios	El sistema seguirá operando con la misma información acerca de las vías

CU #7	Ingresar al sistema	
Actor	Usuario	
Precondición	El usuario debe tener asignado un nombre de usuario y una contraseña para acceder al servicio	
Descripción	Este caso de uso describe como el usuario debe identificarse para utilizar cualquier servicio	
Flujo de eventos	<ul style="list-style-type: none"> • Este caso de uso inicia cuando el sistema solicita al usuario ingresar un nombre de usuario y una contraseña • El usuario digita el nombre y contraseña • El sistema valida los datos ingresados • El sistema indica que los datos ingresados son correctos • El usuario puede empezar a utilizar el servicio 	
Postcondición	El usuario ingresa al servicio	
Excepciones	Situación	Respuesta
	En el caso que los datos ingresados por el usuario no sean correctos	El sistema debe pedir de nuevo el ingreso de los datos

CU #8	Visualizar estado de las vías	
Actor	Usuario	
Precondición	El usuario debe haber validado sus datos de usuario y contraseña en el sistema	
Descripción	Este caso de uso describe como un usuario puede visualizar el tráfico en una vía seleccionada	
Flujo de eventos	<ul style="list-style-type: none"> • Este caso de uso inicia cuando el usuario selecciona la opción visualizar estado de las vías • El sistema en una lista muestra las diferentes vías que pueden ser visualizadas • El usuario escoge que vía desea visualizar • El sistema muestra en una pantalla el tráfico en la vía en ese momento 	
Postcondición	El usuario puede visualizar el estado del tráfico en una vía en un momento determinado	
Excepciones	Situación	Respuesta
	El usuario puede realizar varias consultas	El sistema vuelve a mostrar la lista con las vías que pueden ser vi

REQUERIMIENTOS PARA SERVICIO DE BACKTONES

LISTA DE REQUERIMIENTOS FUNCIONALES

RF #1: El servicio debe permitir que el usuario escoja entre varias opciones de backtones

RF #2: El servicio debe permitir al usuario descargar un backtone

RF #3: El servicio debe permitir al usuario escoger si desea escuchar el backtone mientras llama a otros números

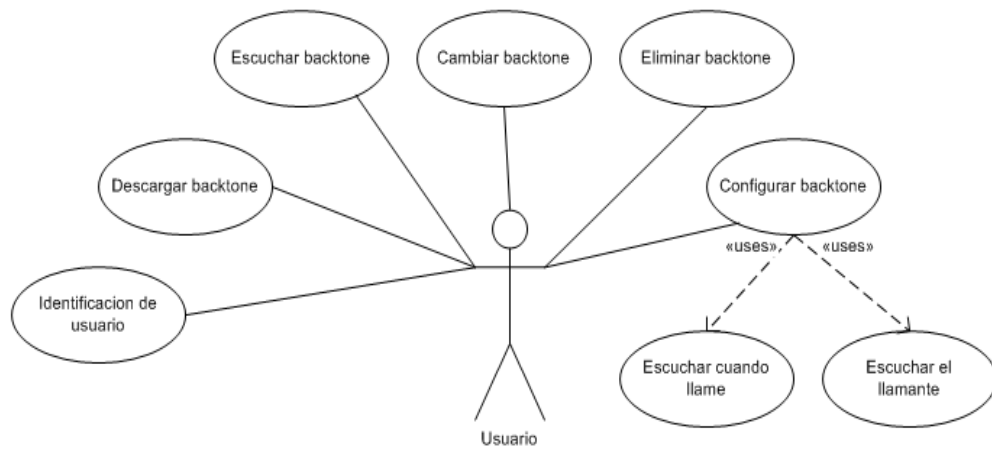
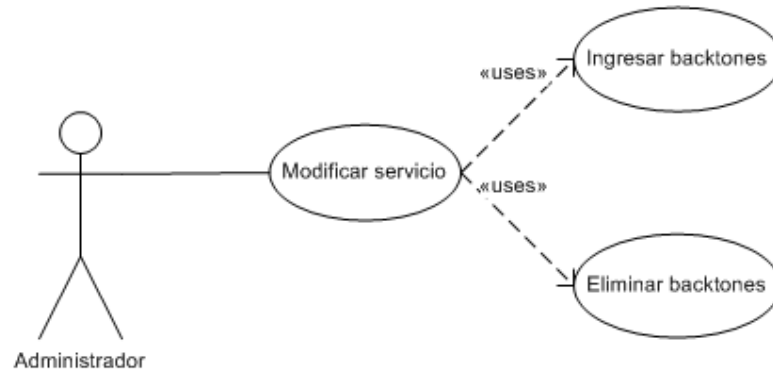
RF #4: El servicio debe permitir al usuario escoger si desea que los llamantes escuchen el backtone mientras responde la llamada

RF #5: El servicio debe permitir guardar los cambios hechos por el usuario

LISTA DE CASOS DE USO

- CU #1:** Modificar servicio
- CU #2:** Ingresar backtones
- CU #3:** Eliminar backtones
- CU #4:** Identificación del usuario
- CU #5:** Descargar backtone
- CU #6:** Cambiar backtone
- CU #7:** Eliminar backtone
- CU #8:** Configurar backtone
- CU #9:** Escuchar cuando realice llamadas
- CU #10:** Escuchar backtone el llamante
- CU #11:** Escuchar backtone

DIAGRAMA DE CASOS DE USO



DESCRIPCIÓN DE LOS CASOS DE USO

CU #1	Modificar servicio	
Actor	Administrador	
Precondición	El administrador debe haber validado sus datos de usuario y contraseña en el sistema	
Descripción	Este caso de uso describe como el administrador del sistema puede modificar el servicio	
Flujo de eventos	<ul style="list-style-type: none"> • El caso de uso inicia cuando el administrador del sistema selecciona la opción modificar servicio • El sistema pregunta el servicio que desea modificar • El administrador debe indicar el servicio de backtones • El sistema solicita al administrador si desea ingresar backtones o eliminar backtones • El administrador selecciona una de las dos opciones 	
Postcondición	El administrador modifica el servicio	
Excepciones	Situación	Respuesta

CU #2	Ingresar backtones	
Actor	Administrador	
Precondición	Deben existir diferentes tipos de backtones	
Descripción	Este caso de uso describe como el administrador del sistema ingresa los diferentes backtones	
Flujo de eventos	<ul style="list-style-type: none"> • Este caso de uso inicia cuando el administrador selecciona la opción ingresar backtones • El sistema solicita al administrador ingresar datos del backtone • El administrador ingresa los datos del backtone • El sistema pregunta si desea ingresar el backtone • El administrador confirma o no si desea ingresar el backtone • El sistema pregunta si se desean guardar los cambios realizados • El sistema guarda la información ingresada • El sistema muestra una pantalla de ingreso de backtone de forma correcta 	

Postcondición	Los usuarios pueden descargar los backtone ingresados	
Excepciones	Situación	Respuesta
	Al final de la configuración del backtone no se presiona la opción guardar cambios	El backtone no será ingresado

CU #3	Eliminar backtones	
Actor	Administrador	
Precondición	Deben existir diferentes tipos de backtones	
Descripción	Este caso de uso describe como el administrador del sistema elimina los backtones	
Flujo de eventos	<ul style="list-style-type: none"> • Este caso de uso inicia cuando el administrador selecciona la opción eliminar backtones • El sistema muestra una lista con los backtones que los usuarios pueden descargar • El administrador selecciona uno de los backtone • El sistema muestra la información referente al backtone • El sistema pregunta si desea eliminar el backtone • El administrador confirma o no si desea eliminar el backtone • El sistema pregunta si se desean guardar los cambios realizados • El sistema guarda los cambios realizados • El sistema muestra una pantalla que la eliminación del backtone ha sido exitosa 	
Postcondición	El backtone ya no se puede descargar	
Excepciones	Situación	Respuesta
	Al final de la configuración del backtone no se presiona la opción guardar cambios	El backtone no será eliminado

CU #4	Identificación de usuario	
Actor	Usuario	
Precondición	El usuario debe tener asignado un nombre de usuario y una contraseña para acceder al servicio	
Descripción	Este caso de uso describe como el usuario debe identificarse para utilizar el servicio	
Flujo de eventos	<ul style="list-style-type: none"> • Este caso de uso inicia cuando el sistema solicita al usuario ingresar un nombre de usuario y una contraseña • El usuario digita el nombre y contraseña 	

	<ul style="list-style-type: none"> • El sistema valida los datos ingresados • El sistema indica que los datos ingresados son correctos • El usuario puede empezar a utilizar el servicio 	
Postcondición	El usuario puede utilizar el servicio	
Excepciones	Situación	Respuesta
	En el caso que los datos ingresados por el usuario no sean correctos	El sistema debe pedir de nuevo el ingreso de los datos

CU #5	Descargar backtone	
Actor	Usuario	
Precondición	Deben existir diferentes tipos de backtones	
Descripción	Este caso de uso describe como el usuario puede escoger un backtone y descargarlo	
Flujo de eventos	<ul style="list-style-type: none"> • Este caso de uso inicia cuando el usuario selecciona la opción descargar backtone • El sistema presenta una lista de los backtones disponibles para descargar • El usuario escoge uno de los backtones • El sistema pregunta si desea descargar el backtone seleccionado • El usuario confirma o no si desea descargar el backtone • El sistema pregunta si desea instalar el backtone en su línea telefónica • El usuario confirma o no si desea instalarlo • El sistema pregunta si desea guardar la configuración • El sistema guarda la información • El sistema instala el backtone en la línea telefónica del usuario 	
Postcondición	El usuario puede utilizar el backtone	
Excepciones	Situación	Respuesta
	Si el usuario no guarda la configuración del backtone	El sistema no hace la instalación del backtone en la línea telefónica

CU #6	Cambiar backtone	
Actor	Usuario	
Precondición	El usuario debe tener un backtone instalado en su línea telefónica	
Descripción	Este caso de uso describe como el usuario puede cambiar el backtone instalado en la línea telefónica	

Flujo de eventos	<ul style="list-style-type: none"> • Este caso de uso inicia cuando el usuario selecciona la opción cambiar backtone • El sistema presenta una lista de los backtones disponibles para descargar • El usuario escoge uno de los backtones • El sistema pregunta si desea cambiar el backtone actual por el seleccionado • El usuario confirma o no si desea cambiar el backtone • El sistema pregunta si desea descargar el backtone seleccionado • El usuario confirma o no si desea descargar el backtone • El sistema pregunta si desea instalar el nuevo backtone en su línea telefónica • El usuario confirma o no si desea instalarlo • El sistema pregunta si desea guardar la configuración • El sistema guarda la información • El sistema instala el nuevo backtone en la línea telefónica del usuario 	
Postcondición	El usuario puede utilizar el nuevo backtone descargado	
Excepciones	Situación	Respuesta
	Si el usuario no guarda la configuración del backtone	El sistema no cambia el backtone antes instalado

CU #7	Eliminar backtone
Actor	Usuario
Precondición	El usuario debe tener un backtone instalado en su línea telefónica
Descripción	Este caso de uso describe como el usuario puede eliminar un backtone instalado en la línea telefónica
Flujo de eventos	<ul style="list-style-type: none"> • Este caso de uso inicia cuando el usuario selecciona la opción eliminar backtone • El sistema presenta la información del backtone actualmente instalado • El sistema pregunta si desea eliminar el backtone • El usuario confirma o no si desea eliminar el backtone • El sistema elimina el backtone • El sistema guarda la nueva configuración • El sistema confirma que el backtone ha sido eliminado

Postcondición	El usuario no tiene backtone instalado en su línea telefónica	
Excepciones	Situación	Respuesta
	Si el usuario no guarda la configuración del backtone	El sistema no elimina el backtone y continua con el anterior backtone

CU #8	Configurar backtone	
Actor	Usuario	
Precondición	El usuario debe tener un backtone instalado en su línea telefónica	
Descripción	Este caso de uso describe como el usuario puede configurar un backtone	
Flujo de eventos	<ul style="list-style-type: none"> • Este caso de uso inicia cuando el usuario selecciona la configurar backtone • El sistema presenta la información del backtone actualmente instalado • El sistema presenta dos opciones que el llamante escuche el backtone o escuchar el backtone cuando el usuario llame • El usuario escoge una de las opciones • El sistema pregunta si desea guardar la configuración • El usuario confirma o no si desea guardar la configuración • El sistema guarda la nueva configuración • El sistema confirma que la configuración ha sido realizada con éxito 	
Postcondición	El backtone es escuchado por el llamante o por el usuario	
Excepciones	Situación	Respuesta
	Si el usuario no guarda la configuración	El sistema por defecto lo configura para que sea escuchado por el llamante

CU #9	Escuchar backtone cuando realice llamadas	
Actor	Usuario	
Precondición	El usuario debe tener un backtone instalado en su línea telefónica	
Descripción	Este caso de uso describe como el usuario puede configurar un backtone para que sea escuchado por el usuario cuando realice llamadas a otros teléfonos	
Flujo de eventos	<ul style="list-style-type: none"> • Este caso de uso inicia cuando el usuario dentro de configurar backtone elige la opción escuchar cuando realice llamadas • El sistema pregunta si desea guardar la configuración • El usuario confirma o no si desea guardar la configuración • El sistema guarda los cambios realizados 	

	<ul style="list-style-type: none"> El sistema muestra en una pantalla que la configuración fue realizada con éxito 	
Postcondición	El backtone es escuchado cuando el usuario realice llamadas a otros teléfonos	
Excepciones	Situación	Respuesta
	Si el usuario no guarda la configuración	El sistema por defecto lo configura para que sea escuchado por el llamante

CU #10	Escuchar backtone el llamante	
Actor	Usuario	
Precondición	El usuario debe tener un backtone instalado en su línea telefónica y cuando se quiera cambiar del modo escuchar cuando realice llamada	
Descripción	Este caso de uso describe como el usuario puede configurar un backtone para que sea escuchado por cualquier llamante cuando quiera comunicarse con el usuario	
Flujo de eventos	<ul style="list-style-type: none"> Este caso de uso inicia cuando el usuario dentro de configurar backtone elige la opción escuchar backtone el llamante El sistema pregunta si desea guardar la configuración El usuario confirma o no si desea guardar la configuración El sistema guarda los cambios realizados El sistema muestra en una pantalla que la configuración fue realizada con éxito 	
Postcondición	El backtone es escuchado por el llamante desee comunicarse con el usuario del servicio	
Excepciones	Situación	Respuesta
	Si el usuario no guarda la configuración	El sistema sigue en el modo de escuchar backtone cuando realice llamadas

CU #11	Escuchar backtone	
Actor	Usuario	
Precondición	El usuario debe tener asignado un nombre de usuario y una contraseña para acceder al servicio	
Descripción	Este caso de uso describe como el usuario puede escuchar un backtone antes de descargarlo	
Flujo de eventos	<ul style="list-style-type: none"> Este caso de uso inicia cuando el usuario elige la opción escuchar backtone El sistema muestra una lista de los backtones activos en el momento El usuario elige uno de los backtone de la lista El sistema pregunta si desea escuchar el backtone seleccionado 	

	<ul style="list-style-type: none"> • El usuario confirma o no si desea escuchar el backtone • El sistema reproduce el backtone 	
Postcondición	El backtone es escuchado por el usuario antes de descargarlo	
Excepciones	Situación	Respuesta

REQUERIMIENTOS PARA SERVICIO DE LOCALIZACIÓN POR MEDIO DE DISPOSITIVOS MOVILES

LISTA DE REQUERIMIENTOS FUNCIONALES

RF #1: El servicio debe permitir ubicar los vehículos de una empresa como (camiones o carros) en cualquier momento dentro de un mapa de la ciudad

RF #2: El servicio debe alertar al encargado de administrar los vehículos cuando alguno sobrepase el límite autorizado

RF #3: El servicio debe trabajar en tiempo real

RF #4: El servicio debe permitir visualizar todos los vehículos de la empresa al mismo tiempo

RF #5: El servicio debe permitir visualizar solo uno de los vehículos de la empresa

RF #6: El servicio debe trabajar con la red GSM/GPRS

RF #7: El servicio debe permitir ingresar nuevos dispositivos móviles para nuevos vehículos

RF #8: El servicio debe permitir deshabilitar un dispositivo móvil por una persona autorizada

RF #9: El servicio debe notificar por un mensaje de texto a quien utiliza el vehículo que se encuentra fuera del perímetro autorizado

RF #10: El servicio debe permitir enviar reportes de la ubicación del vehículo con una frecuencia determinada por el usuario

RF #11: El servicio debe permitir enviar reportes de la velocidad del vehículo con una frecuencia determinada por el usuario

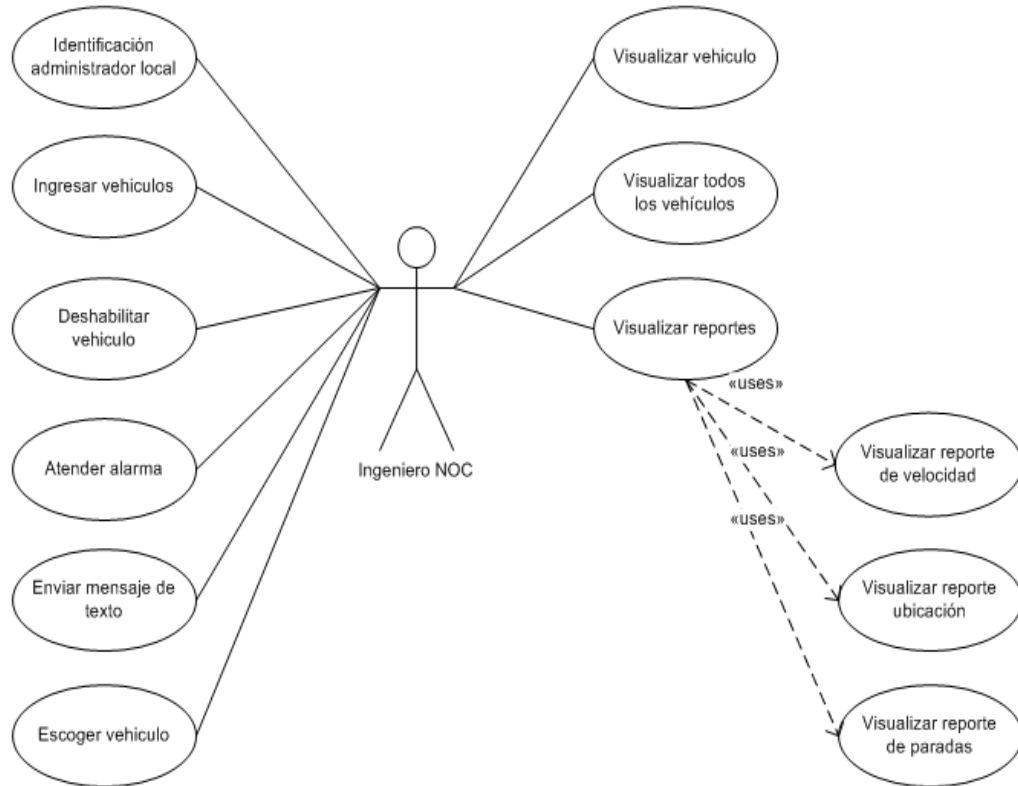
RF #12: El servicio debe permitir enviar reporte de cuantas veces se detuvo el vehículo

RF #13: El servicio debe permitir notificar si el vehículo ha sido robado al accionar un botón de pánico

LISTA DE CASOS DE USO

- CU #1:** Identificación del administrador local
- CU #2:** Ingresar vehículos
- CU #3:** Deshabilitar vehículo
- CU #4:** Atender alarma
- CU #5:** Enviar mensaje de texto
- CU #6:** Escoger vehículo
- CU #7:** Visualizar vehículo
- CU #8:** Visualizar todos los vehículos
- CU #9:** Visualizar reportes
- CU #10:** Visualizar reporte de velocidad
- CU #11:** Visualizar reporte ubicación
- CU #12:** Visualizar reporte de paradas

DIAGRAMA DE CASOS DE USO



DESCRIPCIÓN DE LOS CASOS DE USO

CU #1	Identificación administrador local	
Actor	Ingeniero NOC	
Precondición	El administrador local debe tener asignado un nombre de usuario y una contraseña	
Descripción	Este caso de uso describe como el administrador local debe identificarse para utilizar el servicio	
Flujo de eventos	<ul style="list-style-type: none"> • Este caso de uso inicia cuando el sistema solicita al administrador local ingresar un nombre de usuario y una contraseña • El administrador digita el nombre y contraseña • El sistema valida los datos ingresados • El sistema indica que los datos ingresados son correctos • El administrador puede empezar a utilizar el servicio 	
Postcondición	El usuario puede utilizar el servicio	
Excepciones	Situación	Respuesta
	En el caso que los datos ingresados por el usuario no sean correctos	El sistema debe mostrar un mensaje de error y pedir de nuevo el ingreso de los datos

CU #2	Ingresar vehículos	
Actor	Ingeniero NOC	
Precondición	El nombre de usuario y contraseña del administrador deben ser validos	
Descripción	Este caso de uso describe como el administrador local puede ingresar nuevos vehículos para ser localizados	
Flujo de eventos	<ul style="list-style-type: none"> • Este caso de uso inicia cuando el administrador selecciona la opción ingresar vehículos • El sistema solicita los datos del vehículo • El administrador ingresa los datos • El sistema valida los datos • El sistema le asigna un número de identificación al vehículo • El sistema pregunta al administrador si desea guardar el nuevo ingreso • El administrador confirma o no el ingreso • El sistema pregunta si se desea guardar la información • El administrador confirma o no si desea guardar la información • El sistema muestra en una pantalla que el ingreso del nuevo vehículo 	

	ha sido exitoso	
Postcondición	El vehículo puede ser identificado en el sistema	
Excepciones	Situación	Respuesta
	En el caso que los datos no sean ingresados correctamente	El sistema volverá a solicitar los datos que no sean correctos

CU #3	Deshabilitar vehículo	
Actor	Ingeniero NOC	
Precondición	El nombre de usuario y contraseña del administrador deben ser validos	
Descripción	Este caso de uso describe como el administrador local deshabilitar un vehículo	
Flujo de eventos	<ul style="list-style-type: none"> • Este caso de uso inicia cuando el administrador selecciona la opción deshabilitar vehículo • El sistema solicita los datos del vehículo • El administrador ingresa los datos • El sistema valida los datos • El sistema pregunta al administrador si desea deshabilitar el vehículo • El administrador confirma o no deshabilitar el vehículo • El sistema pregunta si desea guardar los cambios realizados • El administrador confirma o no los cambios • El sistema guarda los cambios realizados • El sistema muestra en una pantalla que el vehículo ha sido deshabilitado 	
Postcondición	El vehículo ya no puede ser localizado	
Excepciones	Situación	Respuesta
	En el caso que los datos no sean ingresados correctamente	El sistema volverá a solicitar los datos que no sean correctos

CU #4	Atender alarma	
Actor	Ingeniero NOC	
Precondición	Un dispositivo debe encontrarse por fuera del perímetro autorizado	
Descripción	Este caso de uso describe como el administrador local debe atender una alarma en el caso que un vehículo se encuentre fuera del área autorizada	
Flujo de eventos	<ul style="list-style-type: none"> • Este caso de uso inicia cuando el sistema genera una alarma avisando que un vehículo se encuentra por fuera del perímetro 	

	<ul style="list-style-type: none"> El sistema muestra en una pantalla la información del dispositivo El sistema muestra a que vehículo pertenece 	
Postcondición	El administrador solo puede visualizar la información	
Excepciones	Situación	Respuesta

CU #5	Enviar mensaje de texto	
Actor	Ingeniero NOC	
Precondición	Una alarma debe ser generada	
Descripción	Este caso de uso describe como el administrador local envía un mensaje de texto al encargado del vehículo para notificar que se encuentra por fuera del perímetro	
Flujo de eventos	<ul style="list-style-type: none"> Este caso de uso inicia cuando se genera la alarma El sistema tiene la información del encargado del vehículo El administrador selecciona enviar mensaje de texto El sistema confirma a que número se debe enviar el mensaje El administrador confirma o no el número El sistema pregunta si se desea enviar el mensaje El administrador confirma o no si desea enviar el mensaje El sistema envía el mensaje El sistema guarda información del mensaje (eje. A qué hora fue enviado) 	
Postcondición	El encargado del vehículo es notificado	
Excepciones	Situación	Respuesta
	En el caso que el mensaje no pueda ser enviado	El sistema debe mostrar un mensaje que el envío no pudo realizarse

CU #6	Escoger vehículo	
Actor	Ingeniero NOC	
Precondición	El nombre de usuario y contraseña del administrador deben ser validos	
Descripción	Este caso de uso describe como el administrador local puede escoger un vehículo determinado y obtener información referente a el	
Flujo de eventos	<ul style="list-style-type: none"> Este caso de uso inicia cuando el administrador selecciona la opción escoger vehículo El sistema muestra una lista de las placas de los vehículos con sus 	

	respectivos dispositivos <ul style="list-style-type: none"> • El administrador escoge un vehículo de la lista • El sistema muestra toda la información referente a ese vehículo (Eje. Placa, dispositivo, dueño, conductor) 	
Postcondición	El administrador puede observar toda la información de un vehículo	
Excepciones	Situación	Respuesta

CU #7	Visualizar vehículo	
Actor	Ingeniero NOC	
Precondición	El nombre de usuario y contraseña del administrador deben ser validos	
Descripción	Este caso de uso describe como el administrador local puede visualizar la ubicación de un vehículo en un momento determinado	
Flujo de eventos	<ul style="list-style-type: none"> • Este caso de uso inicia cuando el administrador selecciona la opción visualizar vehículo • El sistema muestra en una pantalla una lista de todos vehículos • El administrador escoge el vehículo que desea visualizar • El sistema muestra en un mapa la ruta que ha hecho el vehículo y en donde se encuentra en el momento de la consulta 	
Postcondición	El administrador puede visualizar la ubicación de un vehículo	
Excepciones	Situación	Respuesta
	<p>En el caso que el vehículo no pueda ser observado</p> <p>La ubicación en un momento determinado depende del tiempo de actualización que haya escogido quien adquiere el servicio</p>	El sistema mostrara un mensaje de información no disponible en el momento intentar después

CU #8	Visualizar todos los vehículos	
Actor	Ingeniero NOC	
Precondición	El nombre de usuario y contraseña del administrador deben ser validos	
Descripción	Este caso de uso describe como el administrador local puede visualizar las ubicaciones de todos los vehículos de una empresa en un momento determinado	
Flujo de eventos	<ul style="list-style-type: none"> • Este caso de uso inicia cuando el administrador selecciona la opción visualizar todos los vehículos • El sistema muestra en un mapa la ruta que ha hecho cada el vehículo y 	

	<p>en donde se encuentra en el momento de la consulta al mismo tiempo</p> <ul style="list-style-type: none"> • El sistema permite identificar cada vehículo • El sistema diferencia cada ruta 	
Postcondición	El administrador puede visualizar la ubicación de todos los vehículos de una empresa	
Excepciones	Situación	Respuesta
	<p>En el caso que los vehículos no puedan ser observado</p> <p>La ubicación en un momento determinado depende del tiempo de actualización que haya escogido quien adquiere el servicio</p>	El sistema mostrara un mensaje de información no disponible en el momento intentar después

CU #9	Visualizar reportes	
Actor	Ingeniero NOC	
Precondición	El nombre de usuario y contraseña del administrador deben ser validos	
Descripción	Este caso de uso describe como el administrador local puede visualizar diferentes reportes generados por la información de cada vehículo según el tiempo de actualización escogido	
Flujo de eventos	<ul style="list-style-type: none"> • Este caso de uso inicia cuando el administrador selecciona la opción visualizar reportes • El sistema muestra en un pantalla la lista de vehículos • El administrador escoge uno de los vehículos de la lista • El sistema muestra una lista de reportes que puede visualizar el administrador reporte de ubicación, de velocidad y de paradas • El administrador escoge el reporte que desea visualizar • El sistema muestra el reporte generado por el vehículo durante el día, cada tiempo que se hace la actualización de datos 	
Postcondición	El administrador puede visualizar los reportes generados por cada vehículo	
Excepciones	Situación	Respuesta
	<p>En el caso que el reporte no pueda ser visualizado</p> <p>Los reportes dependen del tiempo de actualización que haya escogido quien adquiere el servicio</p>	El sistema mostrara un mensaje de información no disponible en el momento intentar después

CU #10	Visualizar reporte de velocidad	
Actor	Ingeniero NOC	
Precondición	El nombre de usuario y contraseña del administrador deben ser validos	
Descripción	Este caso de uso describe como el administrador local puede visualizar el reporte de velocidad generado por cada vehículo según el tiempo de actualización escogido	
Flujo de eventos	<ul style="list-style-type: none"> • Este caso de uso inicia cuando el administrador dentro de visualizar reportes selecciona la opción visualizar reporte de velocidad • El sistema muestra en un pantalla la lista de vehículos • El administrador escoge uno de los vehículos de la lista • El sistema muestra en una tabla el reporte de velocidad generado por el vehículo durante el día, cada tiempo que se hace la actualización de datos 	
Postcondición	El administrador puede visualizar el reporte de velocidad generado por cada vehículo	
Excepciones	Situación	Respuesta
	<p>En el caso que el reporte no pueda ser visualizado</p> <p>Los reportes dependen del tiempo de actualización que haya escogido quien adquiere el servicio</p>	El sistema mostrara un mensaje de información no disponible en el momento intentar después

CU #11	Visualizar reporte de ubicación	
Actor	Ingeniero NOC	
Precondición	El nombre de usuario y contraseña del administrador deben ser validos	
Descripción	Este caso de uso describe como el administrador local puede visualizar el reporte de ubicación generado por cada vehículo según el tiempo de actualización escogido	
Flujo de eventos	<ul style="list-style-type: none"> • Este caso de uso inicia cuando el administrador dentro de visualizar reportes selecciona la opción visualizar reporte de ubicación • El sistema muestra en un pantalla la lista de vehículos • El administrador escoge uno de los vehículos de la lista • El sistema muestra en una tabla el reporte de los diferentes lugares donde ha estado el vehículo durante el día, cada tiempo que se hace la actualización de datos 	
Postcondición	El administrador puede visualizar el reporte de ubicación de cada vehículo	
Excepciones	Situación	Respuesta
	En el caso que el reporte no pueda ser visualizado	El sistema mostrara un mensaje de información no disponible en el momento intentar después

	Los reportes dependen del tiempo de actualización que haya escogido quien adquiere el servicio	
--	--	--

CU #12	Visualizar reporte de paradas	
Actor	Ingeniero NOC	
Precondición	El nombre de usuario y contraseña del administrador deben ser validos	
Descripción	Este caso de uso describe como el administrador local puede visualizar el reporte de paradas generado por cada vehículo según el tiempo de actualización escogido	
Flujo de eventos	<ul style="list-style-type: none"> • Este caso de uso inicia cuando el administrador dentro de visualizar reportes selecciona la opción visualizar reporte de paradas • El sistema muestra en un pantalla la lista de vehículos • El administrador escoge uno de los vehículos de la lista • El sistema muestra en una tabla el reporte de los diferentes lugares donde ha parado el vehículo durante el día, cada tiempo que se hace la actualización de datos 	
Postcondición	El administrador puede visualizar el reporte de paradas de cada vehículo	
Excepciones	Situación	Respuesta
	En el caso que el reporte no pueda ser visualizado	El sistema mostrara un mensaje de información no disponible en el momento intentar después
	Los reportes dependen del tiempo de actualización que haya escogido quien adquiere el servicio	

REQUERIMIENTOS PARA SERVICIO DE GUÍA EN LA CIUDAD

LISTA DE REQUERIMIENTOS FUNCIONALES

RF #1: El servicio debe trabajar con la red GPRS

RF #2: El sistema debe ofrecer una guía al usuario para llegar de un sitio a otro dentro de la ciudad

RF #3: El servicio debe permitir al usuario visualizar las diferentes opciones de rutas para llegar a un sitio determinado

RF #4: El servicio debe permitir visualizar las rutas en un mapa de la ciudad

RF #5: El servicio debe permitir al usuario ingresar el sitio de origen y el sitio destino

RF #6: El servicio debe permitir que el usuario haga varias consultas

RF #7: El servicio debe permitir actualizaciones en las rutas y el mapa de la ciudad

LISTA DE CASOS DE USO

CU #1: Modificar servicio

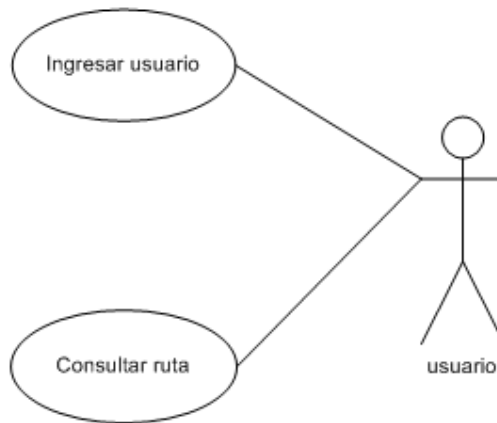
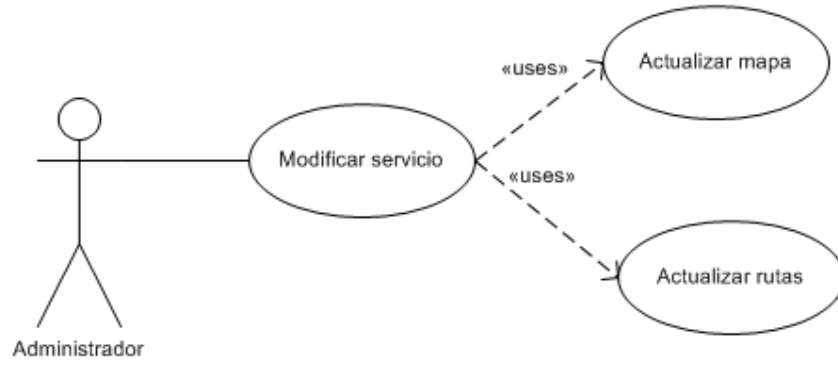
CU #2: Actualizar mapa

CU #3: Actualizar rutas

CU #4: Ingresar usuario

CU #5: Consultar ruta

DIAGRAMA DE CASOS DE USO



DESCRIPCIÓN DE LOS CASOS DE USO

CU #1	Modificar servicio	
Actor	Administrador	
Precondición	El servicio debe tener configurado un mapa de la ciudad y las rutas de un lugar a otro	
Descripción	Este caso de uso describe como el administrador de servicios puede modificar el servicio controlando las cámaras o modificando la información de las vías	
Flujo de eventos	<ul style="list-style-type: none"> • El caso de uso inicia cuando el administrador del sistema selecciona la opción modificar servicio • El sistema pregunta el servicio que desea modificar • El administrador debe indicar el servicio de guía en la ciudad • El sistema solicita al administrador si desea actualizar mapa o actualizar rutas • El administrador selecciona una de las dos opciones 	
Postcondición	El administrador modifica el servicio	
Excepciones	Situación	Respuesta

CU #2	Actualizar mapa	
Actor	Administrador	
Precondición	El servicio debe tener configurado un mapa de la ciudad y las rutas de un lugar a otro	
Descripción	Este caso de uso describe como el administrador del servicio puede actualizar un mapa de la ciudad	
Flujo de eventos	<ul style="list-style-type: none"> • Este caso de uso inicia cuando el administrador dentro de modificar el servicio escoge la opción actualizar mapa • El sistema pregunta si se desea actualizar el mapa anterior • El administrador confirma o no si desea actualizar el mapa anterior • El sistema solicita al usuario subir el nuevo mapa al sistema • El usuario actualiza el mapa • El sistema pregunta si desea guardar los cambios realizados • El administrador confirma o no si desea guardar los cambios • El sistema guarda los cambios • El sistema indica en una pantalla que la actualización ha sido exitosa 	
Postcondición	El mapa de la ciudad es actualizado	

	Situación	Respuesta
Excepciones	En el caso que el administrado no guarde los cambios	El sistema sigue mostrando el mapa anterior

CU #3	Actualizar rutas	
Actor	Administrador	
Precondición	El servicio debe tener configurado un mapa de la ciudad y las rutas de un lugar a otro	
Descripción	Este caso de uso describe como el administrador del servicio puede actualizar las rutas dentro del mapa de la ciudad	
Flujo de eventos	<ul style="list-style-type: none"> • Este caso de uso inicia cuando el administrador dentro de modificar el servicio escoge la opción actualizar rutas • El sistema muestra el mapa de la ciudad • El sistema pregunta que ruta se desea actualizar o ingresar • El administrador escoge la ruta que se va a actualizar o ingresar • El sistema pregunta si desea guardar los cambios realizados • El administrador confirma o no si desea guardar los cambios • El sistema guarda los cambios • El sistema indica en una pantalla que la actualización ha sido exitosa 	
Postcondición	Las rutas son actualizadas	
	Situación	Respuesta
Excepciones	En el caso que el administrado no guarde los cambios	El sistema sigue mostrando en el mapa las rutas anteriores

CU #4	Ingresar usuario	
Actor	Usuario	
Precondición	El usuario debe tener asignado un nombre de usuario y una contraseña para acceder al servicio	
Descripción	Este caso de uso describe como el usuario debe identificarse para utilizar el servicio	
Flujo de eventos	<ul style="list-style-type: none"> • Este caso de uso inicia cuando el sistema solicita al usuario ingresar un nombre de usuario y una contraseña • El usuario digita el nombre y contraseña • El sistema valida los datos ingresados • El sistema indica que los datos ingresados son correctos • El usuario puede empezar a utilizar el servicio 	
Postcondición	El usuario puede utilizar el servicio	

	Situación	Respuesta
Excepciones	En el caso que los datos ingresados por el usuario no sean correctos	El sistema debe pedir de nuevo el ingreso de los datos

CU #5	Consultar ruta	
Actor	Usuario	
Precondición	El nombre de usuario y contraseña deben ser validos	
Descripción	Este caso de uso describe como el usuario puede consultar la ruta que desea visualizar	
Flujo de eventos	<ul style="list-style-type: none"> • Este caso de uso inicia cuando usuario selecciona la opción consultar ruta • El sistema solicita al usuario el sitio donde se encuentra ubicado inicialmente • El usuario ingresa la dirección del sitio de origen • El sistema procesa la información ingresada • El sistema muestra en una pantalla el sitio de origen ingresado • El sistema solicita al usuario el sitio donde a dónde quiere llegar • El usuario ingresa la dirección del sitio destino • El sistema procesa la información ingresada • El sistema muestra en una pantalla el sitio de origen y el sitio destino ingresado • El sistema solicita confirmar la información • El usuario confirma o no que la información sea correcta • El sistema procesa la información • El sistema muestra en una pantalla la ruta establecida 	
Postcondición	El sistema muestra la ruta de un sitio a otro	
	Situación	Respuesta
Excepciones	En el caso que los datos ingresados por el usuario no sean correctos	El sistema debe pedir de nuevo el ingreso de los datos

