

**CLIENTE MÓVIL SIP PARA UN SISTEMA IVR BASADO EN LA ARQUITECTURA DE  
SERVICIOS IMS**

**JOSÉ FERNANDO MUÑOZ BERMEO  
XIMENA VELASCO MELO**

**Trabajo de Grado presentado como requisito para optar al Título de  
Ingeniero en Electrónica y Telecomunicaciones**

**Director**

**JAVIER ALEXANDER HURTADO**

**Ingeniero en Electrónica y Telecomunicaciones**

**Universidad del Cauca  
Facultad de Ingeniería Electrónica y Telecomunicaciones  
Departamento de Telemática  
Popayán  
2006**



## TABLA DE CONTENIDO

	Página
<b>INTRODUCCIÓN</b>	7
<b>1.ARQUITECTURA DE SERVICIOS IMS</b>	10
1.1. ARQUITECTURA DE SERVICIOS IMS	10
1.1.1. Definición	10
1.1.2. Características	10
1.1.3. Arquitectura	11
1.1.3.1. Capa de Acceso y Transporte	12
1.1.3.2. Capa de Control de Sesión	13
1.1.3.3. Capa de Aplicaciones	13
1.1.4. Estado actual y evolución	14
1.1.5. Principales tecnologías y protocolos	14
1.1.5.1. SIP	14
1.1.5.2. IPv6	14
1.1.5.3. RTP y RTCP	15
1.1.5.4. COPS	15
1.1.5.5. DIAMETER	15
1.1.5.6. RSVP y DIFFSERV	15
1.1.5.7. MEGACO	15
1.1.5.8. SIGTRAN	16
1.2. PROTOCOLO DE INICIO DE SESIÓN SIP	16
1.2.1. Definición y origen de SIP	16
1.2.2. Funciones	16
1.2.3. Arquitectura	17
1.2.3.1. Entidades	17
1.2.3.1.1. Principales componentes	17
1.2.3.1.2. Servidores SIP: Servidor de Medios	18
<b>2. OPERACIÓN DE IMS EN CONJUNCIÓN CON SIP</b>	20
2.1. IMS Y SIP	20
2.2. CABECERAS DEL PROTOCOLO SIP PARA IMS	20
2.2.1. P-Associated-URI	20
2.2.2. P-Called-Party-ID	21
2.2.3. P-Visited-Network-ID	21
2.2.4. P-Access-Network-Info	21
2.2.5. P-Charging-Function-Addresses	22
2.2.6. P-Charging-Vector	22
2.3. PROCEDIMIENTO DE REGISTRO	23
2.4. ESTABLECIMIENTO DE SESIÓN	25
2.5. FLUJOS DE SESIÓN	26
<b>3. SERVICIOS IVR SOBRE IMS</b>	28
3.1. SISTEMAS IVR	28
3.1.1. IVR (Interactive Voice Response)	28
3.1.2. Conversión de Texto a Habla	29
3.1.2.1. El proceso de la síntesis del habla	30
3.1.3. Reconocimiento Automático del Habla (RAH)	31



3.1.3.1. Tipos de motores de Reconocimiento	32
3.1.3.2. Análisis de la Señal Acústica	32
3.1.3.2.1. Reconocimiento basado en Patrones	33
3.1.3.2.2. Modelos Ocultos de Markov	33
3.2. SERVICIOS IVR EN IMS	34
3.2.1. La Voz Sobre IP (VoIP) en las Redes Móviles	34
3.2.1.1. Establecimiento del camino de habla	35
3.2.1.2. Señalización	36
3.2.1.3. Transporte de los paquetes de habla	37
3.2.2. Servicios Multimedia en IMS	38
3.2.2.1. Arquitectura de la MRF	39
3.2.2.2. Interacciones que involucran al MRFC y al MRFP	40
3.2.2.3. Requerimientos funcionales para la provisión de capacidades multimedia	40
3.2.2.4. Procedimientos del MRFC y MRFP en la provisión de ASR, TTS, y detección DTMF	42
3.2.2.5. Requerimientos de la arquitectura IMS para los servicios de reconocimiento del habla	42
3.2.2.5.1. Técnicas para mejorar el desempeño del reconocimiento de habla	42
3.2.2.5.2. Impactos en la arquitectura IMS por las mejoras en el reconocimiento de habla	43
3.2.3. Servicios IVR en IMS que se adapten a las redes 2.5G actuales	44
<b>4. HERRAMIENTAS DE PROGRAMACIÓN PARA LA CREACIÓN DE CLIENTES MÓVILES SIP Y SERVICIOS IVR CON BASE EN IMS</b>	<b>46</b>
4.1. HERRAMIENTAS PARA EL DESARROLLO DE LOS MÓDULOS QUE COMPONEN LA ARQUITECTURA IMS	46
4.1.1. Entornos de Desarrollo	46
4.1.2. API para la creación de la Lógica del Servicio	46
4.1.3. Servidores de Aplicaciones	47
4.1.4. Servidores de Medios	48
4.1.4.1. Servidores de Medios Comerciales	48
4.1.4.2. Sintetizadores de Voz	48
4.1.4.3. Reconocedores de Voz	49
4.1.5. Librerías de C++ para el protocolo SIP y el protocolo RTP	50
4.1.5.1. Librerías de C++ para el protocolo SIP	50
4.1.5.2. Librerías de C++ para el protocolo RTP	50
4.1.6. Sistemas de gestión de Bases de datos	51
4.1.7. API para la realización de pruebas de señalización SIP	51
4.2. HERRAMIENTAS PARA EL DESARROLLO DEL CLIENTE MÓVIL SIP	52
4.2.1. Entornos de ejecución de aplicaciones para dispositivos móviles	52
4.2.2. Herramientas de programación para dispositivos móviles	52
4.2.3. APIs y Librerías para J2ME	53
<b>5. PROTOTIPO CLIENTE MÓVIL SIP PARA UN SISTEMA IVR BASADO EN LA ARQUITECTURA DE SERVICIOS IMS</b>	<b>54</b>
5.1. ANÁLISIS DE REQUERIMIENTOS Y FORMULACIÓN DE LA ARQUITECTURA GENERAL DEL SISTEMA	54
5.1.1. Cliente Móvil SIP	54
5.1.2. Servidor CSCF	54
5.1.3. Servidor de Aplicaciones SIP	55
5.1.4. MRF	55
5.1.5. HSS	55
5.1.6. Arquitectura General del Sistema	56
5.2. DISEÑO E IMPLEMENTACIÓN DEL SISTEMA	57
5.2.1. Diagrama de Casos de Uso del Sistema	57



5.2.1.1. Casos de Uso del Sistema iniciados por el actor Cliente Móvil	59
5.2.1.2. Casos de Uso del Sistema iniciados por el actor Administrador	60
5.2.2. Diseño de componentes	61
5.2.2.1. Cliente Móvil SIP	61
5.2.2.2. Servidor CSCF y Servidor de Aplicaciones SIP	62
5.2.2.3. MRF	63
5.2.2.4. HSS	64
5.2.3. Diagramas de Secuencia para los Casos de Uso Esenciales del Sistema originados por el actor Cliente Móvil	65
5.2.3.1. Caso de Uso Registrar Usuario	66
5.2.3.2. Caso de Uso Acceder a Correo Unicauca a través de IVR	67
5.2.3.3. Caso de Uso Desconectar Usuario de la red	70
5.2.3.4. Caso de Uso Gestionar Información de Usuario	71
5.2.4. Diagrama de Secuencia para el Caso de Uso Esencial del Sistema originado por el actor Administrador	72
5.3. PRUEBAS	74
5.3.1. Pruebas de Validación del Sistema	74
5.3.2. Pruebas de Desempeño	77
5.3.3. Pruebas de Usabilidad	82
5.3.4. Pruebas de uso de Memoria	83
5.3.5. Pruebas de Ancho de Banda necesario	84
5.4. APORTES	87
<b>6. RECOMENDACIONES PARA EL DESARROLLO DE CLIENTES MÓVILES IVR BASADO EN LA ARQUITECTURA IMS</b>	<b>88</b>
6.1. RECOMENDACIONES PARA EL DESARROLLO DE UN SISTEMA IVR, CON CAPACIDADES DE ASR Y TTS	88
6.1.1. Recomendaciones para el desarrollo de sistemas IVR basados en reconocimiento de habla	88
6.1.2. Otras recomendaciones	97
6.2. RECOMENDACIONES PARA EL DESARROLLO DE LOS MÓDULOS FUNCIONALES DE LA ARQUITECTURA IMS EN COMBINACIÓN CON LA ARQUITECTURA DEL PROTOCOLO SIP	98
6.3. RECOMENDACIONES PARA EL DESARROLLO DEL CLIENTE MÓVIL SIP	101
<b>7. CONCLUSIONES Y TRABAJOS FUTUROS</b>	<b>108</b>
7.1. CONCLUSIONES	108
7.2. TRABAJOS FUTUROS	109
<b>GLOSARIO</b>	<b>110</b>
<b>BIBLIOGRAFÍA</b>	<b>114</b>



## LISTA DE FIGURAS

	Página
Figura 1. Arquitectura de Servicios IMS	12
Figura 2. Procedimiento de registro en IMS	23
Figura 3. Procedimiento de establecimiento de sesión en IMS	25
Figura 4. Proceso de Síntesis del Habla	30
Figura 5. Proceso de Reconocimiento del Habla basado en patrones	33
Figura 6. Pila de protocolos utilizada en VoIP para redes móviles	36
Figura 7. Transporte de SIP y SDP en IP	37
Figura 8. Transporte del habla en IP	37
Figura 9. Arquitectura de la MRF	39
Figura 10. Arquitectura General del Sistema	56
Figura 11. Diagrama de Casos de Uso del Sistema	58
Figura 12. Diseño del componente Cliente Móvil SIP	61
Figura 13. Diseño de los componentes: Servidor CSCF y Servidor de Aplicaciones SIP	62
Figura 14. Diseño del componente MRF	64
Figura 15. Diseño del componente HSS	65
Figura 16. Caso de Uso Registrar Usuario	66
Figura 17. Caso de Uso Acceder a Correo Unicauca a través de IVR	69
Figura 18. Caso de Uso Desconectar Usuario de la red	70
Figura 19. Caso de Uso Gestionar Información de Usuario	71
Figura 20. Caso de Uso Gestionar Información de Usuarios	73
Figura 21. Cliente JAIN SIP de pruebas	74
Figura 22. Consola SIPMethod	75
Figura 23. Consola MRF	76
Figura 24. Consola Lógica de Servicio	76
Figura 25. Rendimiento del PC No.1 sin programas en ejecución	77
Figura 26. Rendimiento del PC No.1 al activar el servidor de base de datos	78
Figura 27. Rendimiento cuando se ejecuta SIPMethod	79
Figura 28. Rendimiento cuando se ejecuta el Cliente Móvil	79
Figura 29. Rendimiento del PC No.2 sin programas en ejecución	80
Figura 30. Rendimiento cuando se ejecuta Verbio TTS/ASR	81
Figura 31. Rendimiento cuando se ejecuta la MRF	82
Figura 32. Uso de memoria inicial del Cliente Móvil	83
Figura 33. Uso de memoria para el registro del Cliente Móvil	83
Figura 34. Uso de memoria para el inicio de sesión del Cliente Móvil	84
Figura 35. Uso de memoria para el intercambio de flujo RTP del Cliente Móvil	84
Figura 36. Mensajes SIP utilizados por la aplicación	85
Figura 37. Datagramas empleados en SIP	86
Figura 38. Datagramas empleados en RTP	86



### LISTA DE TABLAS

	Página
Tabla 1. Caso de Uso Registrar Usuario	59
Tabla 2. Caso de Uso Acceder a Aplicación a través de IVR	59
Tabla 3. Caso de Uso Desconectar Usuario de la red	59
Tabla 4. Caso de Uso Gestionar Información de Usuario	60
Tabla 5. Caso de Uso Gestionar Información de Usuarios	60
Tabla 6. Caso de Uso Iniciar Servicio	60
Tabla 7. Caso de Uso Finalizar Servicio	60
Tabla 8. Pruebas de Usabilidad	82

### LISTA DE ANEXOS

- Anexo A. Arquitectura de servicios IMS, protocolo SIP y servicios IVR sobre IMS
- Anexo B. Entorno de desarrollo y herramientas de programación para clientes móviles SIP
- Anexo C. Herramientas para el desarrollo de los módulos que componen la arquitectura IMS
- Anexo D. Manual de Usuario y de Instalación del Sistema



## INTRODUCCIÓN

El teléfono móvil es una herramienta que facilita la vida de las personas, por sus características inherentes de movilidad. Sin embargo, no pueden o son difíciles de utilizar, por el tamaño de la pantalla en conjunto con un teclado reducido, las cuales constituyen las limitaciones más evidentes para su utilización o manejo por parte de personas con deficiencias audiovisuales, motrices, de avanzada edad, o que simplemente busquen facilidad en la realización de tareas. Todo ello, a pesar de ofrecer cada día mejores capacidades de medios y de comunicación, y de ser cada vez más fácilmente programables. [1]

Motivados por las limitaciones físicas nombradas, se plantearon diversos retos con relación al tratamiento de la información y a la creación de servicios que añadan accesibilidad a los dispositivos móviles, entendiéndose esto último como el hecho de proporcionar flexibilidad dirigida a las necesidades de cada usuario y sus preferencias [2]. Además, para de alguna manera contribuir y dar un primer paso, dentro del marco que se desarrolla este proyecto, para que los campos de accesibilidad software y tecnología móvil converjan [1], ya que existe un esfuerzo importante, de los entes de estandarización a nivel mundial, por conseguir la “accesibilidad para todos” y la integración real y efectiva de lo que se denomina “diseño para todos” en el desarrollo de productos [3]. Es por esto, que en el presente Trabajo de Grado se desarrolló un servicio que añade accesibilidad a los teléfonos móviles para así brindar un primer avance en mejorar la vida de aquellas personas que más necesiten y quieran cambiar la forma como han venido interactuando con estos dispositivos.

Dentro de las tecnologías que permiten mejorar la accesibilidad de los dispositivos móviles, están las tecnologías de habla, entre las cuales se encuentran la conversión de texto a voz o voz sintética y el reconocimiento de voz. La primera, es una tecnología capaz de generar de forma automática la secuencia de sonidos que produciría una persona al leer un texto cualquiera en voz alta. Con la conversión de texto a voz, cualquier información escrita a la que pueda acceder un computador es leída en voz alta, y puede ser enviada a cualquier dispositivo de salida de audio o por ejemplo puede ser almacenada en un archivo para luego ser enviada a través de una red como Internet. Por otra parte, el reconocimiento de voz es una tecnología que supone un paso más en la búsqueda de una interfaz cómoda y potente para la interacción de los usuarios con las máquinas y dispositivos, permitiendo dar ordenes a estos mediante la voz. La tecnología más avanzada en este campo es la conocida como “reconocimiento de lenguaje natural” que consiste básicamente en disponer de un reconocedor capaz de “entender” lo que diga el usuario, sin atenerse a una expresión específica. Se trata de que la persona que emplea el reconocedor pueda expresarse como quiera, con frases cortas o largas, empleando las palabras que desee. [3]

Por otro lado, se hace necesaria la incorporación rápida de estos nuevos servicios a las infraestructuras de telecomunicaciones actuales, y en este sentido se buscó una infraestructura cuya filosofía así lo permitiera. Es en este momento cuando se decidió la utilización de una arquitectura de servicios que brindará acceso a usuarios de dispositivos móviles y que fuera capaz de soportar la creación e implementación de servicios nuevos de manera que estos procesos no requirieran demasiado tiempo, característica propia de redes que basan su funcionamiento en estándares abiertos.

En el camino de búsqueda y estudio de tecnologías actuales con esas características, así mismo como del entorno en el cual estas se enmarcan, se comprendieron las limitaciones del campo en que pretendía incursionar este proyecto debido a inconvenientes de aceptación en el mercado,



pues hoy en día los servicios más empleados por los usuarios móviles siguen siendo los tradicionales de voz, SMS, y MMS, y para un proyecto como este dichos servicios no permiten su desarrollo.

Se encontró entonces, que los servicios de datos siguen en sus fases tempranas en las redes del Sistema Global para Comunicaciones Móviles (Global System for Mobile Communications, GSM) y del Servicio General de Paquetes de Radio (General Packet Radio Service, GPRS), creciendo a paso lento aunque seguro. Sin embargo, los esfuerzos evolutivos del Proyecto Conjunto de Tercera Generación (Third Generation Partnership Project, 3GPP) plasmados en la incorporación del Subsistema Multimedia IP (IP Multimedia Subsystem, IMS) al núcleo de la red móvil de la Tercera generación de sistemas de comunicaciones móviles (3G) del Sistema de Telecomunicaciones Móviles Universales (Universal Mobile Telecommunications System, UMTS), en la Release 5, están ayudando a que se le empiece a dar una mayor importancia y valoración a los servicios de datos en las redes móviles. IMS es un sistema de control de sesión diseñado con tecnologías de Internet adaptadas al mundo móvil, que hace posible la provisión de servicios móviles multimedia sobre conmutación de paquetes. El objetivo económico y comercial de esta iniciativa es doble: aumentar los ingresos medios por abonado y reducir costos. Para lo primero, la estrategia consiste en potenciar el mercado de la conmutación móvil de paquetes, introduciendo capacidades para ofrecer nuevos servicios avanzados a los usuarios finales. Con IMS aparece un nuevo concepto de servicios, radicalmente diferente, innovador y factible al mismo tiempo. Se pretende que el atractivo de estos servicios IMS involucre definitivamente a los usuarios móviles en el mundo de los datos y de esta forma el presente proyecto se basa en una arquitectura que además de permitir su desarrollo rápido, de alguna forma le garantiza su éxito, hablando en términos reales de implementación e incursión en un mediano plazo en el mercado local. [4]

Para la implementación de esta nueva tecnología, y para asegurar la Calidad de Servicio (Quality of Service, QoS), la red debe tener una arquitectura consistente y robusta de servicios, para garantizar, por ejemplo, la alta confiabilidad proporcionada por las redes maduras de telefonía [5]. La obtención de un tipo de arquitectura como estas, con los requerimientos que establece y se encuentra actualmente definiendo, no será de la noche a la mañana y según los analistas la migración a esta nueva red podría durar hasta 6 años más. En este sentido, se buscó un servicio que accedido a través de los sistemas Respuesta de Voz Interactiva (Interactive Voice Response, IVR) y con base a las capacidades actuales de las redes de 2.5G se logrará llevar a cabo. Los servicios que más se adecuan al entorno actual en el que se enmarca este proyecto son los servicios que no tienen tantos requerimientos como los sistemas que brindan capacidades IVR a las redes de acceso y de transporte ya maduras. Por lo anterior, se seleccionó un servicio del tipo Persona a Contenido, en el cual el usuario móvil puede acceder a su correo electrónico de la Universidad, mediante el establecimiento de una sesión de audio que se encamina a lo largo de una red IMS básica.

IMS no solo proporciona soluciones al acceso inalámbrico, sino que plantea una red independiente del acceso, generando con esto un marco estandarizado y abierto para la convergencia de redes. La diversidad de servicios que se ofrezcan, podrán ser accedidos por usuarios a través de múltiples dispositivos, siempre y cuando estos tengan las capacidades de hacerlo. [6]

La adquisición de los fundamentos arquitectónicos y normativos de IMS, y particularmente de las especificaciones de protocolos e interfaces específicas de IMS, así como el conocimiento de soluciones ya disponibles en el mercado, son esenciales para todos los actores (operador de redes o de servicios, proveedor de equipos, desarrolladores de aplicaciones, o sus usuarios) que desean tomar un lugar en el mundo emergente de los servicios sobre el Protocolo de Internet (Internet Protocol, IP).



En el marco de este proyecto, factores como el crecimiento de la penetración celular, la difusión de la banda ancha, la proliferación de soluciones de voz, datos y video, así como la educación de los usuarios en lo que a servicios se refiere, permitirán que Colombia empiece a vivir en un futuro próximo una etapa de convergencia gracias a la utilización de tecnologías como IMS, que aunque se encuentra enmarcada dentro de una red mucho mayor, que aún se encuentra en estandarización, permite el desarrollo e implementaciones adecuadas para la nueva generación de servicios que están surgiendo.

En cuanto a proyectos desarrollados en la Universidad del Cauca cuyas temáticas tratadas son afines a las del presente proyecto se tienen: Proyecto de “Sistemas para la gestión de servicios de Red Inteligente soportados en Internet” [7], el cual dentro de sus temáticas, abarca el área de IVR sobre la plataforma SPIRITS; Proyecto de “Diseño de un Módulo de Control de Sesiones de Usuario basado en la Arquitectura de servicios IMS para el despliegue de aplicaciones y/o servicios en redes de Telefonía Móvil” [8], es un prototipo de CSCF. Y a nivel internacional, en la Universidad de Cataluña se realizó la “Implementación de un servidor Proxy SIP en Java” [9], donde se tratan los temas de VoIP e IMS dando especial énfasis al CSCF.

El presente documento se dividió en 6 capítulos. En el capítulo 1, se realiza una descripción general acerca de la Arquitectura de Servicios IMS y el protocolo SIP (como se soportan los servicios IVR dentro de IMS). En el capítulo 2, se describen la forma como opera IMS en conjunto con SIP, algunas cabeceras del perfil IMS de SIP, y dos procedimientos básicos en el manejo de sesiones. En el capítulo 3, se habla en términos generales acerca de los Sistemas IVR, y se enfatiza en los servicios IVR enmarcados dentro de la Arquitectura de Servicios IMS. En el capítulo 4, se describen las posibles herramientas de programación para la creación de clientes móviles que utilicen el Protocolo de Inicio de Sesión (Session Initiation Protocol, SIP) y servicios IVR, con base a la arquitectura IMS. En el capítulo 5, se detalla el análisis, diseño, implementación y la forma como se validó el Prototipo cliente móvil SIP para un sistema IVR basado en la arquitectura de servicios IMS. En el capítulo 6, se elaboraron una serie de recomendaciones para el desarrollo de clientes móviles con acceso a servicios IVR, basado en la arquitectura IMS. Estas recomendaciones se dividieron de la siguiente forma: Recomendaciones para el desarrollo de un sistema IVR, con capacidades de Reconocimiento Automático de Habla (Automatic Speech Recognition, ASR) y de Conversión de Texto a Habla (Text To Speech, TTS); Recomendaciones para el desarrollo de los módulos funcionales de la arquitectura IMS en combinación con la arquitectura del protocolo SIP, y Recomendaciones para el desarrollo del cliente móvil SIP. Finalmente, en el capítulo 7 se muestran las conclusiones y trabajos futuros.

Adicionalmente a la monografía se tienen 4 anexos, los cuales deberán consultarse para profundizar en varios de los temas más importantes que se estudiaron y utilizaron para el desarrollo del presente proyecto. En el anexo A, se trata en detalle la Arquitectura de servicios IMS, el protocolo SIP y servicios IVR sobre IMS. En el anexo B, se describe el entorno de desarrollo y las herramientas de programación que se utilizaron para la creación del cliente móvil SIP. En el anexo C, se describen las herramientas que se utilizaron para el desarrollo de los módulos que componen la arquitectura IMS. El último anexo, el D, contiene el manual de instalación del sistema.



## 1. ARQUITECTURA DE SERVICIOS IMS

### 1.1. ARQUITECTURA DE SERVICIOS IMS

#### 1.1.1. Definición

El objetivo de IMS es llevar la evolución de las redes de operadores fijos y móviles hacia un escenario de convergencia. Este estándar, desarrollado por el 3GPP y el 3GPP2, se basa en el protocolo de control de sesiones SIP, estandarizado por la IETF. IMS permite la creación de Redes de Próxima Generación (Next Generation Network, NGN), además de nuevas aplicaciones multimedia capaces de suministrar al usuario final una experiencia de comunicaciones integrada, independientemente del tipo de aplicación, método de acceso a la red, e infraestructura de la misma, de modo que los usuarios finales tengan acceso a servicios desde cualquier sitio, en cualquier momento, y sobre múltiples tecnologías de acceso [10].

IMS es un sistema totalmente independiente del acceso, lo que permite un escenario de desarrollo de servicios sobre IP y servicios multimedia para usuarios conectados a través de redes fijas, Líneas de Abonado Digitales Asimétricas (Asymmetric Digital Subscriber Line, ADSL), WLAN, WIMAX, GPRS, WCDMA, CDMA2000, UMTS, o cualquier otro acceso, lo cual posibilita la convergencia, con el consecuente beneficio para operadores (creación de nuevos servicios y sinergias en las inversiones) y usuarios (facilidad de uso y acceso a servicios desde cualquier equipo terminal). [11]

IMS permite y optimiza servicios móviles multimedia en tiempo real y no real. Entre los servicios IMS se pueden incluir el de Presionar Para Hablar para celulares (Push To Talk Over Cellular, PoC), compartir video en tiempo real, juegos interactivos, servicios de mensajería instantánea, mensajería de voz, telefonía de voz y video, y videoconferencia. [12]

#### 1.1.2. Características

Como IMS fue diseñado para una red evolucionada de GSM, hereda ciertas cualidades intrínsecas del mundo móvil, como lo son [4][13]:

- La itinerancia. IMS soporta la itinerancia o “roaming” de tipo nativo, lo que se define como la capacidad del sistema de admitir y dar servicio a abonados de otros operadores que emplean la misma tecnología, y con los cuales se tiene el acuerdo de negocio pertinente.
- La interconexión con redes y servicios heredados. Existen elementos IMS para el interfuncionamiento entre las sesiones multimedia con los componentes de audio y las redes PSTN, GSM o el dominio de conmutación de circuitos de la propia red 3G.
- La seguridad integrada. IMS requiere autenticación de abonado y especifica sus propios mecanismos y arquitectura de seguridad, los cuales son independientes de los propios de UMTS.



- La Calidad de Servicio. El subsistema GPRS de 3G hace uso de la arquitectura de QoS de UMTS, que define una jerarquía de portadoras adecuadas para servicios con diferentes requisitos de QoS. Por lo anterior, se han definido funciones nuevas e interfaces opcionales en GPRS 3G, en el marco de Release 5, que permiten que IMS controle y autorice el uso de recursos del subsistema de transporte GPRS.

Otras características de IMS son las siguientes [14][15]:

- Soporte a servicios de comunicaciones no tiempo real, pseudo tiempo real y tiempo real.
- Conexiones orientadas por la sesión. Varias sesiones y servicios simultáneamente sobre la misma conexión de red.
- Movilidad sin restricciones (movilidad de servicios, movilidad del usuario, y movilidad de la sesión), también permitiendo el control de servicios de la red origen.
- Convergencia fija-móvil de servicios y de la operación de la red.
- Independencia de la red de acceso. Servicios distintos debido al tipo de acceso.
- Interfaces abiertas y estandarizadas para todos los elementos de red.
- Base de datos centralizada para simplificación de la operación.
- Interoperabilidad terminal - red. Variedad de terminales, terminales libres.
- Permite al operador mantener el control y el modelo de negocio actual, ya que las dependencias con la red le ayudan a asegurar su posición. De esta forma, IMS asegura que el operador no sea relegado a ser un simple transportador de datos.
- Al ser un estándar aceptado por la industria, IMS permite un despliegue de servicios multimedia a un mercado masivo.
- La eficiencia en costos ayuda a disminuir el riesgo en el lanzamiento de nuevos servicios.

### **1.1.3. Arquitectura [4][5][13][14]**

La arquitectura de servicios IMS es una colección de funciones lógicas que se dividen en 3 capas: la capa de acceso y transporte, la capa de control, y la capa de aplicaciones. Estas capas y sus componentes básicos se muestran en la Figura 1.

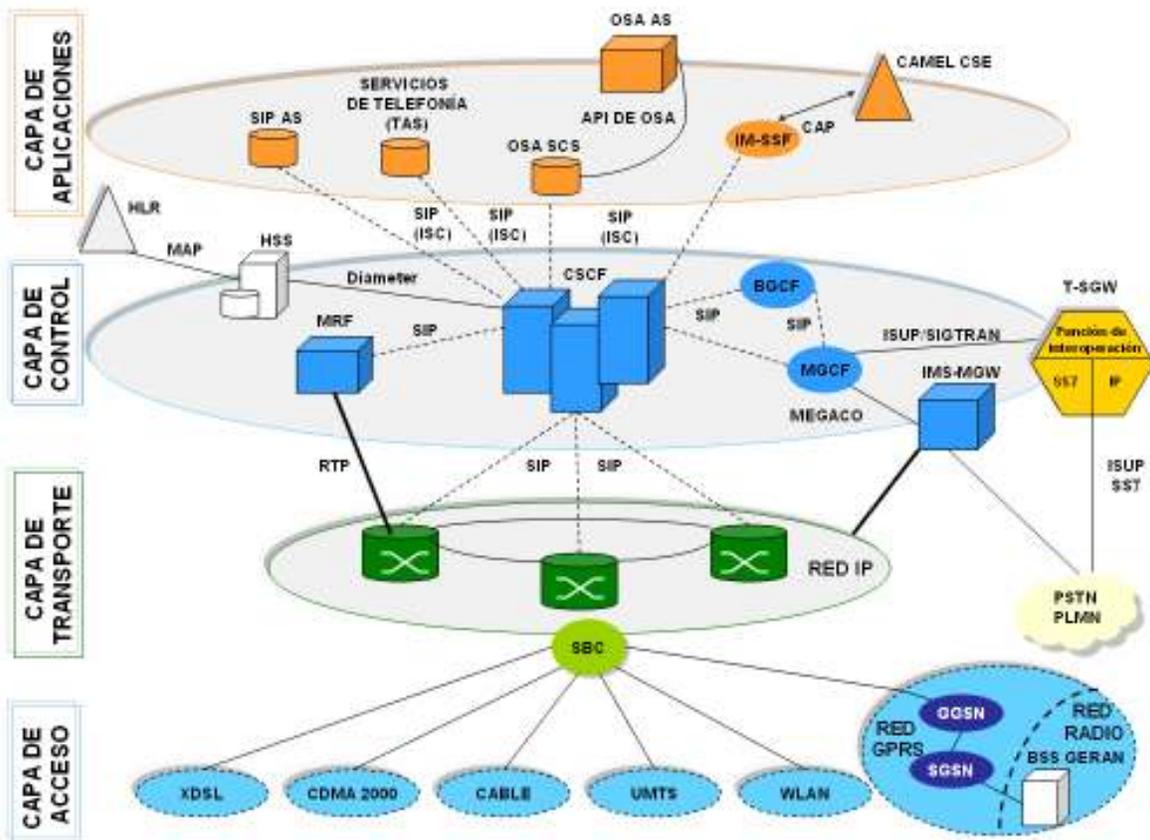


Figura 1: Arquitectura de Servicios IMS

En las siguientes secciones se describen las características más importantes de las entidades que conforman la arquitectura IMS.

### 1.1.3.1. Capa de Acceso y Transporte

Conecta las diferentes redes de acceso con el corazón de la arquitectura IMS, mediante el empleo de pasarelas y servidores de control, los cuales se encargan, por ejemplo, de la conversión de los flujos de Voz sobre IP (Voice over IP, VoIP) al formato de Multiplexación por División de Tiempo (Time Division Multiplexing, TDM) de la Red Telefónica Pública Conmutada (Public Switched Telephone Network, PSTN).

Inicia y termina la señalización SIP mediante la cual se pueden configurar sesiones y proveer los servicios que se soportan utilizando el protocolo de Transporte en Tiempo Real (Realtime Transport Protocol, RTP).



### **1.1.3.2. Capa de Control de Sesión**

Abarca los servidores de control de red para la gestión de las llamadas (establecimiento, modificación y liberación); contiene la Función de Control de la Sesión de Llamada (Call Session Control Function, CSCF), que constituye la entidad funcional clave encargada de proveer el registro de los extremos finales y el enrutamiento de los mensajes de señalización SIP hacia el servidor de aplicación apropiado, además realiza funciones de soporte como aprovisionamiento, tarificación, operación y mantenimiento. La CSCF interopera con la capa de transporte y de acceso para garantizar la calidad de servicio a través de todos los servicios. Además, la CSCF puede desempeñar tres papeles: el papel de Proxy-CSCF (P-CSCF), el papel de Interrogating-CSCF (I-CSCF) y el papel de Serving-CSCF (S-CSCF). El P-CSCF es el primer punto de contacto de un móvil en la red IMS. El S-CSCF es responsable de gestionar la sesión del móvil. La función del I-CSCF es determinar el S-CSCF basado en carga o capacidad de la red.

Esta capa contiene también la base de datos local de suscriptores (Home Subscriber Server, HSS), la cual mantiene un perfil único de servicio por cada usuario final. Dicho perfil almacena toda la información de servicio y preferencias de un usuario en una localización central, lo cual incluye la información actual de registro del usuario final (como por ejemplo la dirección IP), información de itinerancia, servicios de telefonía (como por ejemplo información de reenvío de llamada), información del servicio de mensajería instantánea (por ejemplo lista de contactos), opciones del correo de voz (por ejemplo saludos), etc. Como esta información se encuentra centralizada, las aplicaciones pueden compartirla para crear directorios personales unificados, múltiples tipos de usuarios con la información de presencia, y servicios combinados. También, este orden centralizado simplifica enormemente la administración de los datos de usuario.

Por otro lado, la capa de control de sesión también incluye la Función de Recursos Multimedia (Media Resource Function, MRF), la cual se encarga de los recursos multimedia, y la Función de Control de la Pasarela de Medios (Media Gateway Control Function, MGCF), la cual se encarga de hacer interoperar la señalización SIP con la señalización utilizada por la pasarela de medios (por ejemplo H.248).

### **1.1.3.3. Capa de Aplicaciones**

Contiene los servidores de aplicaciones y de contenido, que son los encargados de proveer la lógica de servicio del usuario final. Tanto IMS como la señalización SIP son lo suficientemente flexibles como para soportar una variedad de servidores de aplicaciones, tanto telefónicas como no telefónicas. Por ejemplo, los estándares de SIP se desarrollaron para servicios de telefonía y para servicios Multimedia IP.

IMS define tres formas diferentes de ofrecer servicios: servicios nativos SIP, servicios tradicionales, y servicios de terceras partes.

Como protocolo para conectar la gran mayoría de los nodos de IMS entre sí y con el resto de los elementos que componen la red móvil, se ha determinado en el 3GPP la utilización del Protocolo de Inicio de Sesión SIP, el cual se tratará más adelante. Por el momento, se puede agregar que IMS utiliza también a SIP para establecer llamadas de multimedia y voz en Internet y se lo eligió por su simplicidad, extensibilidad y su amplia disponibilidad.



#### **1.1.4. Estado actual y evolución [4][16][17]**

IMS está definido en las versiones del sistema 3GPP del Release 5 y 6. En el Release 5, ya finalizado, se definen perfectamente la arquitectura y las entidades funcionales, aunque en las especificaciones de este Release solamente se describe de forma detallada el funcionamiento básico de IMS, dejándole al Release 6 bastantes funcionalidades de este último. Así pues el Release 5 y Release 6 están muy relacionados.

Dentro del marco del Release 5 se plantea la utilización de SIP para el control de sesiones IP multimedia en cuanto a la señalización entre el Equipo de Usuario (User Equipment, UE) e IMS, y también entre entidades funcionales IMS. Igualmente, la especificación del subsistema IMS utiliza SIP para completar llamadas de voz y llamadas multimedia con otras redes (redes telefónicas e Internet).

En 3GPP Release 6 (IMS Fase 2) se completa y amplía el concepto relativo a los servicios IP multimedia.

El Release 7 se encuentra en desarrollo, y en esta se trata la habilitación de velocidades mayores y mejoras en la capacidad, así como el soporte a servicios de tiempo real mejorados tales como la VoIP, los juegos interactivos, y el PoC, entre otros temas.

#### **1.1.5. Principales tecnologías y protocolos [4][13][18]**

##### **1.1.5.1. SIP**

La señalización en IMS se efectúa mediante el protocolo SIP, que la Fuerza de Trabajo en Ingeniería de Internet (Internet Engineering Task Force, IETF) diseñó para la gestión de sesiones multimedia en Internet. El IETF ha ido añadiendo al protocolo básico extensiones y cabeceras privadas para adaptar su uso a las necesidades del entorno móvil, y a las particularidades de UMTS. Por ello, se habla del perfil 3GPP del protocolo SIP, una variante personalizada para la red 3G IMS. SIP aporta las funciones para el registro, establecimiento, liberación y mantenimiento de las sesiones IMS, lo que incluye funciones de enrutado de sesiones e identificación de usuarios y nodos, y también habilita todo tipo de servicios suplementarios.

Más información de SIP se puede encontrar en la RFC3261 [19].

##### **1.1.5.2. IPv6**

IMS se ha definido desde el inicio como una red y un servicio que fundamenta completamente su transporte de red sobre IPv6. El subsistema GPRS 3G que proporciona acceso a dicha red IPv6 modifica sus especificaciones para soportar el transporte de datagramas IPv6 desde el terminal de usuario hasta IMS, así como la configuración y la asignación de direcciones de red. Por otro lado, el terminal IMS ha de soportar la pila de protocolos de IPv6, y posiblemente de IPv4, por ello se han realizado trabajos en la generación de mecanismos para el interfuncionamiento IPv4/IPv6.

Por otro lado, además de las ventajas inherentes a IPv6 como son la QoS y la seguridad integradas, así como la autoconfiguración y el mayor espacio de direccionamiento, el tráfico del



plano de usuario se transfiere directamente entre terminales siguiendo el esquema “peer-to-peer” o “de igual a igual”. Por tanto, IPv6 simplifica el transporte de este modelo de tráfico en las redes IPv4 privadas, como son la mayor parte de las redes GPRS existentes en el mundo.

Más información de IPv6 se puede encontrar en la RFC791 [20].

### **1.1.5.3. RTP y RTCP**

La IETF desarrolló los protocolos RTP/RTCP de envío de flujos multimedia de audio y vídeo basados en el protocolo UDP. Estos protocolos se utilizan en IMS para el transporte y secuenciación de flujos IP multimedia del plano de usuario, RTP para la transmisión de la voz y vídeo sobre IP, y el protocolo RTCP para el control de la transmisión de los paquetes RTP.

Más información de RTP y RTCP se puede encontrar en las RFC1889 [21] y RFC3605 [22], respectivamente.

### **1.1.5.4. COPS**

El protocolo de Servicio de Políticas Común y Abierto (Common Open Policy Service, COPS) se utiliza para el control de los recursos de GPRS, de señalización de sesión y señalización de transporte, mediante el uso de políticas o normas de asignación de los mismos en función de los objetivos de calidad establecidos.

Más información de COPS se puede encontrar en la RFC2748 [23].

### **1.1.5.5. DIAMETER**

Es un protocolo que proporciona un marco de Autenticación, Autorización y Tasación (Authentication, Authorization and Accounting, AAA) para aplicaciones de acceso a redes, movilidad en redes IP e itinerancia. Diameter es compatible con Radius. Se compone de una especificación base más una serie de especificaciones dependientes de la aplicación. Principalmente se emplea como heredero de la Parte de Aplicación Móvil (Mobile Application Part, MAP) para el diálogo con el nodo HSS de IMS.

### **1.1.5.6. RSVP y DIFFSERV**

Los protocolos de Reservación de Recursos (Resource Reservation Protocol, RSVP) y DiffServ los utiliza IMS para asegurar la QoS extremo a extremo, especialmente cuando la conectividad IP requerida se extiende más allá de la red móvil GPRS.

Más información de RSVP y DiffServ se puede encontrar en las RFC2205 [24] y RFC2475 [25], respectivamente.

### **1.1.5.7. MEGACO**

Este protocolo se utiliza en arquitecturas SIP y H.323 para el control de las Pasarelas de Medios que permiten el interfuncionamiento con las redes TDM.



Más información de MEGACO o H.248 se puede encontrar en la RFC 3525 [26].

#### **1.1.5.8. SIGTRAN**

El grupo de trabajo del IETF denominado SIGTRAN, ha definido un conjunto de protocolos y adaptadores de protocolos estándares que permiten el transporte de información del Sistema de Señalización número 7 (Signaling System 7, SS7) sobre redes basadas en el protocolo IP.

### **1.2. PROTOCOLO DE INICIO DE SESIÓN SIP**

#### **1.2.1. Definición y origen de SIP [4][16]**

SIP son las siglas en inglés del Protocolo para el Inicio de Sesión, el cual ha existido desde 1995, y fue ratificado en 1999 bajo el RFC 2543 por la IETF. El protocolo actual está definido en el RFC 3261.

Este es un protocolo de señalización del nivel de aplicación diseñado para el establecimiento, modificación, mantenimiento y terminación de sesiones interactivas sobre redes IP para diferentes tipos de aplicaciones. El inicio de sesión, cambio o término de la misma, son independientes del tipo de medio o aplicación que se use en la llamada; dichas sesiones incluyen elementos multimedia como voz, datos, video y otros formatos, y pueden ser desde una llamada de teléfono IP, una conversación breve a través de un servicio de mensajes instantáneos donde se pueden compartir páginas Web y documentos, Chat o juegos interactivos, hasta una videoconferencia.

SIP aporta las funciones para el registro, enrutado de sesiones, identificación de usuarios y nodos, localización y disponibilidad de usuarios, negociación de las capacidades de comunicación y también habilita todo tipo de servicios suplementarios.

#### **1.2.2. Funciones [13][27]**

- Resolución de Direcciones
- Funciones de Sesión
  - Establecimiento.
  - Determinación de la disponibilidad del punto extremo.
  - Negociación de medios.
  - Modificación de la sesión.
  - Transferencia y terminación de sesiones.
  - Cancelación de sesión.
  - Señalización en llamada.
  - Control de llamada.
  - Configuración de QoS.



Los tipos de Sesiones que puede establecer SIP son los siguientes:

- ✓ Entre dos partes (llamadas telefónicas)
  - ✓ Multiparte (todos pueden participar)
  - ✓ Multicast (uno envía, muchos reciben)
- Funciones no relacionadas con la sesión
    - Movilidad.
    - Transporte de mensajes.
    - Suscripción a eventos.
    - Autenticación y contabilidad.
    - Otras funciones (SIP es extensible).

### 1.2.3. Arquitectura

SIP es un protocolo basado en texto y en la arquitectura cliente-servidor, por lo que todos los procesos se ejecutan mediante el intercambio de mensajes en forma de peticiones y respuestas entre una entidad cliente y otra que actúa como servidor.

#### 1.2.3.1. Entidades

##### 1.2.3.1.1. Principales componentes [28][29][30]

- **Agente de Usuario (User Agent)**

Corresponde a una aplicación software ubicada en la estación terminal. Puede operar como cliente, realizando peticiones SIP, y en este caso se tratará de un Agente de Usuario Cliente (User Agent Client, UAC), o como servidor, generando respuestas, y tratándose en este caso de un Agente de Usuario Servidor (User Agent Server, UAS). En ambos procesos, el agente de usuario es capaz de soportar una comunicación básica, en la cual administra el estado de la llamada.

- **Servidor Proxy (SIP Proxy)**

Actúa como cliente y servidor, y es el responsable primario del encaminamiento de mensajes entre equipos finales. Se encarga de interpretar, reescribir o traducir un mensaje, en caso de ser necesario, antes de enrutarlo hacia su destino final, y retiene la información por cuestiones de contabilidad o facturación.

- **Servidor de Redirección (Redirect Server)**

Se encarga de aceptar peticiones, y a partir de la dirección del destinatario final obtiene la dirección del siguiente elemento de la cadena. Dicha dirección es devuelta al cliente para que éste sea capaz, por sí mismo, de contactar al destinatario final. Al contrario de un servidor Proxy, el Redirect no inicia sus propios mensajes SIP, sólo responde y al contrario de un agente de usuario, no acepta o termina llamadas.



- **Servidor de Registro (Register Server)**

Acepta peticiones de registro por parte de los UA y almacena dicha información para proporcionarla a otros UA. Un usuario puede estar registrado con múltiples dispositivos y un dispositivo puede tener registrado múltiples usuarios, pero cada usuario es responsable de registrar y mantener el registro en sus diferentes dispositivos.

- **Servidor de Localización (Location Server)**

Suministra la información sobre la posible localización del destinatario de la llamada a los servidores de Redirección o Proxy. Es alimentado por el servidor de Registro o por interfaces de aprovisionamiento de usuarios. Consiste en una base de datos y no utiliza SIP para comunicarse con los otros servidores.

La división entre servidores Proxy, Register, Redirect y Location no es física, sino conceptual, con lo cual dichas funciones pueden coexistir en un único servidor, o estar separadas por motivos de escalabilidad, redundancia o rendimiento.

Cuando un usuario o terminal SIP desea establecer una comunicación con otro, envía un mensaje a su servidor Proxy para conocer la dirección física que tiene en ese momento el destinatario de la comunicación. En esta fase, el llamante indicará al servidor Proxy la dirección lógica del usuario llamado, por ejemplo, la dirección de correo electrónico de este último. Con esta información, el servidor Proxy escalará la consulta al servidor de Localización quien resolverá la localización física del usuario llamado.

#### **1.2.3.1.2. Servidores SIP: Servidor de Medios [13]**

Existen varios tipos de servidores SIP:

- **Servidores de aplicación (Application servers):** permiten el control de la ejecución de la aplicación.
- **Servidores de medios (Media servers):** permitiendo el control de medios.
- **Servidores de llamada (Call servers, CSCF):** responsables del control y enrutamiento de llamadas, así como de la invocación de los servicios internos (ejecutados por el CSCF) y externos (ejecutados por servidores de aplicación).
- **Servidores de mensajería (Messaging servers):** responsables del almacenamiento de mensajes cortos, del envío de los mensajes y de la notificación de nuevos mensajes. Este es un tipo particular de servidor de aplicación.

#### **Funcionalidades del servidor de medios**

Las funcionalidades del servidor de medios incluyen el control de medios y de los recursos en sí: Anuncios (audio y vídeo), reconocimiento automático de la voz, generación de información de tasación, IVR, grabación de mensajes vocales, conversión de texto a voz, traducción entre diferentes dispositivos codificadores-decodificadores o “codecs”, gestión de multipartes y de conferencia, replicación del contenido audio y vídeo RTP, interfaces normalizadas y abiertas (SIP, VoiceXML).



### **Interfaces del servidor de medios**

- Control: Protocolo SIP.
- Multimedia: Protocolo RTP para el transporte de voz y vídeo y RTCP para el control del transporte.
- Servidor de archivo externo: Los archivos multimedia (audio, vídeo) son almacenados en el servidor de medios o en un servidor de archivo externo. El protocolo de Transferencia de Hipertexto (Hypertext Transfer Protocol, HTTP) permite leer y almacenar los ficheros en un servidor de archivo externo.
- ASR y TTS: Las funcionalidades ASR y TTS pueden ser provistas por el servidor de medios o por un servidor externo. El protocolo de control del servidor ASR/TTS es el Protocolo de Control de los Recursos Multimedia (Media Resource Control Protocol, MRCP).
- Tasación: Protocolo DIAMETER para recuperar los Registros Detallados de Llamada (Call Detailed Record, CDR) del servidor de medios.
- Gestión de sesión interactiva: Protocolo de Transmisión Continua en Tiempo Real (Real-Time Streaming Protocol, RTSP).
- Gestión: La gestión de los servidores de medios es hecha por interfaces del Protocolo Simple de Administración de Red (Simple Network Management Protocol, SNMP), Interfaces por Línea de Comandos (Command Line Interface, CLI) y HTTP.



## 2. OPERACIÓN DE IMS EN CONJUNCIÓN CON SIP

### 2.1. IMS Y SIP [31]

El 3GPP adopta el protocolo SIP, el cual fue estandarizado inicialmente por la IETF y le hizo los ajustes necesarios para proveer soporte completo a los requerimientos de una red IMS, los cuales son básicamente la definición de varias extensiones. Estas extensiones adaptan el protocolo SIP para su uso dentro de la arquitectura IMS, y son definidas en el estándar TS.24.229 del 3GPP.

SIP ha sido extendido con el fin de soportar numerosos servicios, tales como servicios de presencia, PTT, mensajería instantánea, transferencia de llamada, conferencia, servicios complementarios de telefonía, etc.

Por definición SIP no es un protocolo diseñado para una red o una aplicación específica. Para utilizar SIP, se debe definir un Perfil de uso. Los perfiles de uso trabajan como plantillas, y proporcionan un ambiente variado y flexible según los requisitos particulares.

El perfil de SIP que se creó para IMS es uno de los más importantes en el ámbito de las telecomunicaciones, pues no solo involucra las redes móviles sino toda la industria de las telecomunicaciones en conjunto y según los expertos actualmente es el más apropiado para las NGN.

Para una mejor comprensión de las funciones de las entidades IMS, en las secciones 2.3 y 2.4 se muestran dos de los procedimientos más comunes en IMS: el proceso de registro (necesario para que el abonado pueda acceder a los servicios IP multimedia) y el establecimiento de sesión (que permite iniciar las comunicaciones con otros abonados y con los servicios multimedia).

### 2.2. CABECERAS DEL PROTOCOLO SIP PARA IMS [32][33]

#### 2.2.1. P-Associated-URI [34]

La cabecera *P-Associated-URI* contiene la lista de identidades públicas que el usuario autoriza, es decir, contiene un conjunto de URI's relacionadas con una dirección de registro.

La primera URI en la lista de las identidades públicas del usuario, provista por el HSS al S-CSCF, indica la identidad pública de usuario, que el S-CSCF usará por defecto. Un servidor de registro (Register) contiene información que permite a una URI (dirección de registro) asociarse a cero o más URI's, generalmente todas éstas se asignan a un usuario específico.

Esta extensión de SIP le permite al UAC saber, sobre un registro realizado, cuales son las URI's asociadas, si las hay. Esta cabecera es aplicable en redes SIP, donde el proveedor asigna a un usuario un conjunto de identidades que podrá utilizar, en este caso el proveedor tendrá



conocimiento de todas las identidades de cada usuario, tanto de las que tiene en uso, como las que tiene restringidas.

El servidor de registro inserta la cabecera `P-Associated-URI` en una respuesta `200 OK` de una petición `REGISTER`, y el valor del campo de cabecera se llena con las URI's asociadas a la URI de la dirección de registro. El servidor realiza este procedimiento para un proceso de registro, tanto de "re-registro", como de "de-registro".

### **2.2.2. P-Called-Party-ID**

Esta cabecera es insertada por un servidor Proxy, con el valor de la dirección lógica que registró un usuario, por lo general en una petición `INVITE`, antes de sustituirla con la dirección que va a utilizar para encaminar la petición al usuario. De este modo se asegura que el destino reciba la dirección lógica correspondiente a la petición.

### **2.2.3. P-Visited-Network-ID**

Esta cabecera está diseñada para realizar las funciones relacionadas con la itinerancia de los usuarios, es decir, cuando un usuario se traslada de su red local a una red visitada.

Esta cabecera se utiliza para transmitir el identificador de red visitada desde el Servidor de Registro al Proxy de la red local. Dicho identificador es entendido por el servidor de Registro o Proxy tanto de la red local como de la red visitada. Generalmente, la red local autoriza a un usuario para entrar en una red visitada, siempre y cuando existan acuerdos de itinerancia entre dichas redes.

### **2.2.4. P-Access-Network-Info**

Esta cabecera contiene información sobre la red de acceso que el UA está utilizando, esta información es conocida por el UA y requerida por el Proxy a la hora de proveer los servicios.

El protocolo SIP fue pensado para trabajar independiente de la tecnología de acceso, por lo tanto esta información no es de uso general del protocolo SIP. Además, la información que se transporta en esta cabecera es alterable fácilmente, la protección de esta información depende de la existencia de acuerdos y relaciones de seguridad entre los servidores Proxy que utilizarán el contenido de esta cabecera, y depende también del conocimiento que el UA tenga de esas relaciones. Es por eso que este mecanismo es apropiado solo para ambientes donde existan los elementos de seguridad apropiados.

Cuando un UA envía una solicitud o respuesta SIP al servidor proveedor de servicios, inserta esta cabecera en el mensaje SIP. La cabecera contiene información de la red de acceso que el UA está usando para tener conectividad IP. Generalmente, esta cabecera la ignoran los servidores intermedios que se encuentran entre el UA y el servidor proveedor del servicio. Este último, puede "leer" y usar la información que se encuentra en la cabecera, y luego quitarla del mensaje. Para



borrar la cabecera también deben existir transacciones seguras entre el UA y el servidor que provee los servicios, estas transacciones generalmente soportan mecanismos de seguridad como AKA.

El servidor que provee los servicios al usuario esta ubicado generalmente en su red local y por lo tanto es confiable, entonces se debe borrar la cabecera cuando la señalización SIP es enviada a un servidor ubicado en un dominio poco confiable.

### **2.2.5. P-Charging-Function-Addresses [35]**

Esta cabecera contiene los nombres de los servidores o las direcciones IP de los nodos que reciben la información de facturación. En esta cabecera se introduce una lista de direcciones de una o más entidades de facturación a las cuales un Proxy puede enviar información relacionada con la tarificación. La solución que provee el 3GPP define dos tipos de entidades funcionales de tarificación una es la CCF (Charging Collection Function) y la otra es ECF (Event Charging Function).

- CCF: se utiliza para la facturación de usuarios post-pago. [36]
- ECF: se utiliza para la facturación de usuarios prepago. [36]

La cabecera P-Charging-Function-Addresses no se incluye en los mensajes SIP enviados fuera del dominio de la red, tampoco si el domino no provee una función de facturación.

### **2.2.6. P-Charging-Vector [35]**

Cabecera que proporciona información para poder correlacionar los registros de tarificación generados, por cada una de las entidades de red involucradas en una misma sesión.

Existen tres tipos de información de correlación que se puede transmitir: El valor de la identidad de tarificación de IMS, llamada ICID (IMS Charging Identity), la dirección del servidor Proxy SIP que crea el valor del ID, y el IOI (Inter operator identifiers).

- ICID: se utiliza para correlacionar archivos de tarificación, y debe ser un valor global único.
- IOI: se utiliza para identificar la red que origina y donde finaliza una transacción SIP. Existe un parámetro IOI para cada lado de la transacción, es decir uno para la red origen y otro para la red destino.

El vector de tarificación es definido como un recolector de información de tarificación, el cual se utiliza para trasportar información de facturación, como por ejemplo el valor del ICID.

Más información de estas cabeceras se podrá encontrar en el RFC 3455 del IETF. [32]



### 2.3. PROCEDIMIENTO DE REGISTRO [4]

Antes de utilizar los servicios del dominio IMS, tales como establecer una sesión multimedia o recibir una sesión, el usuario debe registrarse en IMS. Este proceso involucra un P-CSCF de la red visitada si el usuario está en una red visitada o un Proxy Controlador de la Función de Sesión de Llamada (Proxy-CSCF, P-CSCF) de la red nominal si el usuario está en su red.

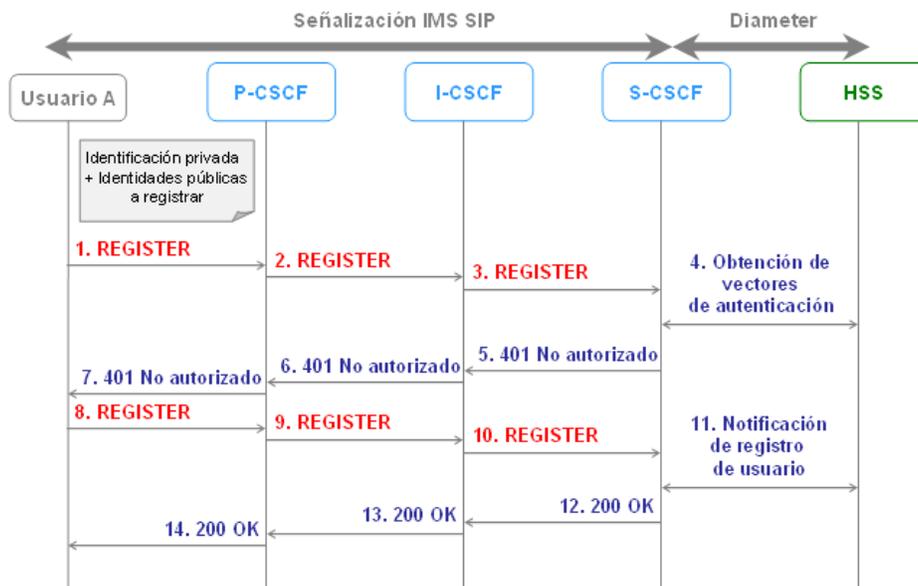


Figura 2: Procedimiento de registro en IMS

El procedimiento de registro en IMS se representa en la Figura 8. Este consta de las siguientes fases:

- En primer lugar, como paso previo para acceder a IMS, el usuario debe registrarse en el sistema. Mediante este proceso se activan las identidades públicas que el usuario desea emplear en sus sesiones multimedia y se establece el S-CSCF que le aportará el servicio. Para llevarlo a cabo, se emplea la señalización SIP y un algoritmo de autorización/autenticación por desafío de usuario a red, y viceversa, que recibe el nombre de Clave de Acuerdo y Autenticación IMS (IMS Authentication and Key Agreement, IMS-AKA).

Más información acerca de IMS-AKA se puede encontrar en 3GPP TSG SA WG3 Security [37] y 3GPP TS 33.102 V7.0.0 [38].

- A continuación, el usuario inicia el proceso enviando un mensaje SIP REGISTER hacia el P-CSCF, que detecta que se trata de un mensaje que no se encuentra protegido por ninguna asociación de seguridad previa; es decir, se trata de un mensaje de registro inicial. En ese mensaje se encuentran la identidad privada del usuario, almacenada en la ISIM, y las identidades públicas que desea registrar para su posterior uso.



En esta fase, el P-CSCF envía el mensaje hacia un I-CSCF, que se encarga de seleccionar un S-CSCF hacia el que reenvía la petición de registro. Cuando el S-CSCF recibe el mensaje, comprueba que no se trata de un usuario que ya este registrado y contacta con el HSS para obtener los vectores de autenticación, necesarios para el algoritmo IMS-AKA.

Posteriormente, para solicitar la autenticación, devuelve hacia el terminal móvil un mensaje SIP 401 "No autorizado", en el que se incluyen ciertos números generados aleatoriamente, así como las claves para el cifrado y protección de la integridad de la señalización IMS.

- Por último, el usuario, en base al mensaje de desafío especificado en la anterior fase, comprueba la identidad de la red IMS y genera un mensaje nuevo SIP del tipo REGISTER. Este segundo mensaje contiene una respuesta formada a partir del algoritmo de autenticación IMS-AKA.

Cuando el mensaje llega al S-CSCF, el usuario es registrado finalmente después de comprobar la veracidad de su identidad. Posteriormente, el S-CSCF indica al HSS que aquel ha registrado al abonado satisfactoriamente y descarga desde allí la suscripción IMS del usuario. El proceso finaliza con el asentimiento SIP 200 OK enviado hacia el terminal móvil.

Gracias al registro IMS (utilizando el protocolo SIP):

- La red IMS notifica al HSS de la localización actual del terminal del usuario y actualiza su perfil.
- El usuario es autenticado antes de poder acceder a los servicios IMS.
- El dominio IMS nominal del usuario selecciona un S-CSCF apropiado responsable de la invocación de sus servicios. Estos últimos se encuentran en servidores de aplicación. El S-CSCF tiene el perfil del usuario (como la Información de Suscripción en el Servidor de Aplicaciones o ASSI) que le permite conocer la lista de servicios a invocar.



## 2.4. ESTABLECIMIENTO DE SESIÓN [4]

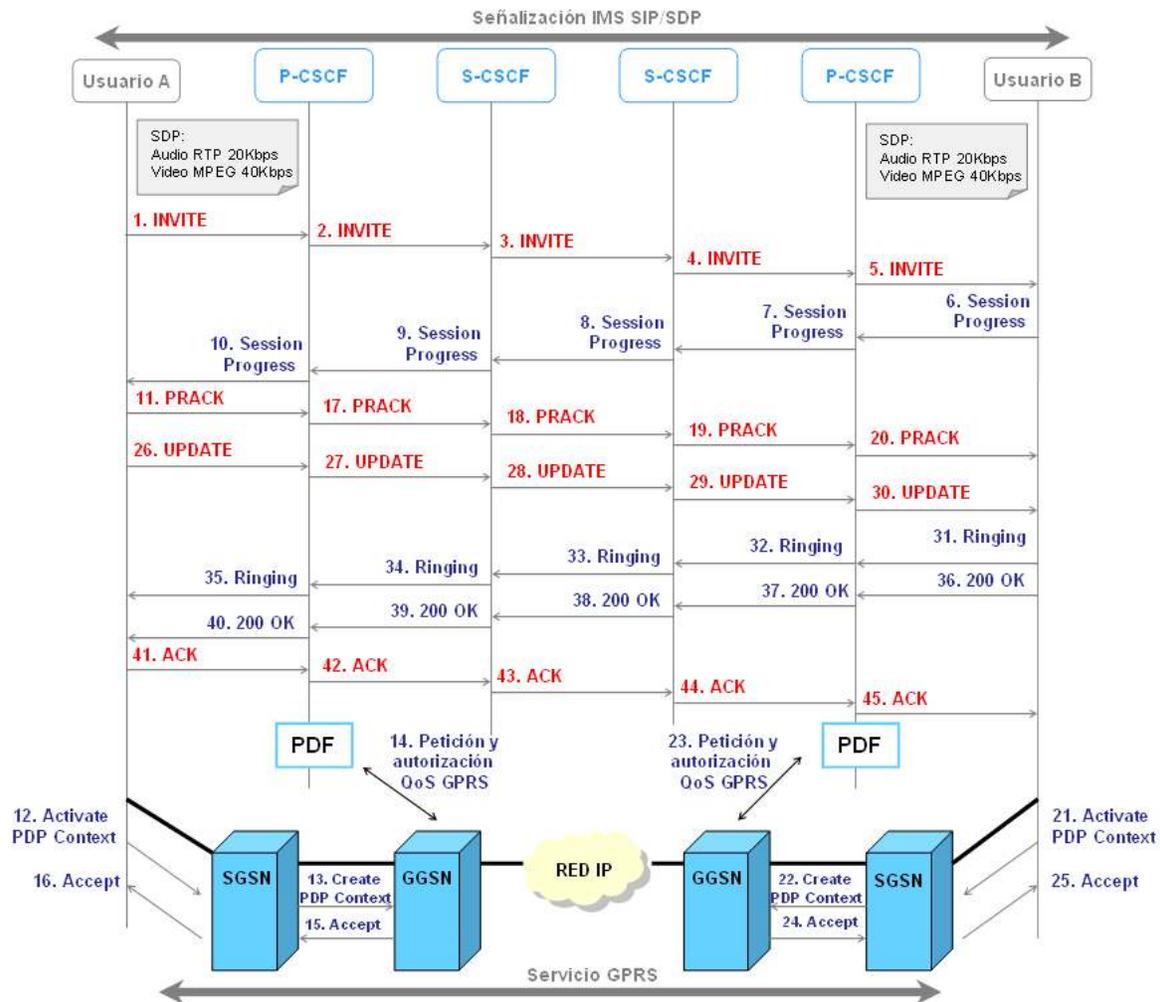


Figura 3: Procedimiento de establecimiento de sesión en IMS

El procedimiento de inicio de sesión se muestra en la Figura 3. Las fases de este proceso son las siguientes:

- En primer lugar, una vez que el usuario ha sido registrado en el subsistema IMS, aquel puede acceder a los servicios IP multimedia que proporciona IMS. De esta manera el usuario, por ejemplo, podría establecer una sesión de videoconferencia con otro de una red diferente. Para ello, utilizará el subsistema IMS para intercambiar información de señalización mediante los protocolos SIP y el Protocolo de Descripción de Sesión (Session Describing Protocol, SDP) con el usuario con el que se quiere comunicar. El objetivo de este intercambio de señalización es el establecimiento de una sesión, mediante la cual se contactará con el nodo destino, se negociarán los parámetros de sesión y se activarán los recursos GPRS necesarios para soportar la sesión multimedia.



Para realizar lo anterior, el usuario origen debe enviar a través de IMS un mensaje SIP `INVITE`, en el que añadirá también el mensaje SDP que describe las capacidades de la sesión que pretende establecer.

En ese mensaje SDP estarán incluidos los medios que quiere transmitir, la tasa binaria a la que se transmitirá cada medio, los protocolos utilizados para la transmisión de los medios, los codecs que se utilizarán, etc. La señalización SIP y SDP llegará al usuario remoto pasando por los nodos IMS de la red origen y destino.

- A continuación, el terminal destino retorna al nodo origen un mensaje SIP de progreso de sesión (`183`), en el que se añade un mensaje SDP con la respuesta del nodo destino al ofrecimiento de los parámetros SDP del nodo origen. Estos parámetros pueden haber sido modificados en función de las capacidades del terminal o las preferencias del usuario.
- Seguidamente el terminal origen envía un mensaje `PRACK` como respuesta al de progreso de sesión, en el que está incluida la oferta SDP final. En este momento es cuando se activan los recursos necesarios a nivel GPRS en la red origen, para soportar los medios que se han negociado. Si se ha habilitado el control de QoS mediante COPS, IMS puede interactuar con el nivel GPRS para autorizar los recursos y la QoS para cada medio.

Por su parte, en la red destino se activan los recursos GPRS, una vez que llega el `PRACK` con el mensaje SDP final. Para simplificar, en la Figura 3 no se ha mostrado el mensaje SIP `200 OK` de respuesta del `PRACK` desde el terminal destino hacia el origen. Cuando éste recibe el mensaje `200 OK`, envía un mensaje SIP `UPDATE` para indicar al destino que ha tenido éxito la operación de activación GPRS.

Es entonces cuando el terminal destino avisa acerca de la llamada que está entrando, a la vez que envía la señalización SIP para indicar al terminal origen que el usuario destino está siendo alertado.

- Por último, cuando el usuario destino descuelga para recibir la sesión de videoconferencia, su terminal envía otro mensaje `200 OK`, con el que se confirma el establecimiento definitivo de la sesión desde el usuario remoto. Si se implementa el control QoS desde IMS, este mensaje activa a su paso, a través de IMS, el plano de transporte GPRS para autorizar la transferencia de paquetes, y empieza el intercambio de tráfico de usuario, compuesto de audio y vídeo, entre los terminales que soportan el servicio GPRS.

## 2.5. FLUJOS DE SESIÓN [6]

Existen dos tipos de flujo, uno es el flujo de mensajes de señalización y otro es el flujo de datos del usuario.

Los mensajes de señalización irán del móvil a través de la Red Terrestre de Acceso Radio UMTS (UMTS Terrestrial Radio Access Network, UTRAN) o el Subsistema de la Estación Base (Base Station Subsystem, BSS) de Acceso Radio GPRS EDGE (GPRS EDGE Radio Access, GERAN), al Nodo de Soporte del Servidor GPRS (Server GPRS Support Node, SGSN) y al Nodo de Soporte de la Pasarela GPRS (Gateway GPRS Support Node, GGSN), para luego ir al CSCF y finalmente a la red destino (a otra red de IMS Release 5, a una red de MGCF con la Pasarela de Medios, o



simplemente hacia la red IP). Es importante distinguir entre qué componentes procesan el mensaje y cuáles lo enrutan. Cuando el móvil envía una petición para establecer un servicio, la petición se envía al S-CSCF (vía el P-CSCF y el I-CSCF). El SGSN y el GGSN realizarán solamente la función de enrutadores. No miran el contenido del mensaje, sólo miran la dirección IP del destino y encaminan el mensaje de acuerdo a ésta.

La información del usuario se transmitirá del móvil, a través del SGSN y el GGSN hacia la red destino (al GGSN de otra red de Release 5, o a la red PSTN pasando por la Pasarela de Medios MGW, o hacia Internet), saltándose la red de CSCFs. Es importante notar que en IMS por tanto se sigue la filosofía de que señalización y datos toman caminos distintos en la red.



### 3. SERVICIOS IVR SOBRE IMS

#### 3.1. SISTEMAS IVR

##### 3.1.1. IVR (Interactive Voice Response)

IVR es una tecnología de automatización de contacto con el usuario, es decir pretende que un operador humano se reemplace por un programa interactivo de alta calidad que provea respuestas a las peticiones de los usuarios, realizadas a través del teclado telefónico o comandos de voz. [39]

Esto difiere de CATI (Entrevista telefónica asistida por computador) donde un entrevistador lee las preguntas e ingresa la respuestas en un computador [40]. Muchos atributos de CATI son compartidos por IVR incluyendo salto lógico automático o ramificación basada en respuestas, entrevistas que pueden ser pausadas y reanudadas en un tiempo posterior, validación de respuestas, control de grabación, etc. Además, un sistema IVR puede ofrecer ventajas considerables tales como: Un sistema IVR puede correr automática y continuamente, los llamantes pueden responder a su propia velocidad; Mejoramiento en la exactitud de la entrevista (cuando se utiliza para ese fin); Reducción de costos y un alto grado de confidencialidad [41].

IVR ha sido utilizado inicialmente en la construcción de “Calls Center” o Centros de Llamada y en compañías grandes para manipular un amplio número de llamadas recibidas en los departamentos de atención al usuario. Estos servicios han sido utilizados en el sistema bancario y en sistemas de reservación. [39]

Los sistemas modernos IVR tienen las siguientes características [42]:

- Facilidad de reconocimiento de voz de llamante.
- Soporte para múltiples lenguajes.
- Capacidad de integración con otras plataformas IVR.

La mayoría de los sistemas utilizan interfaces DTMF donde un usuario marca un número y un computador responde la llamada e inicia el proceso de dialogo. Una vez la sesión es terminada, el sistema IVR desconecta la llamada.

Para 1980 DTMF era el estándar de FACTO para IVR. Sin embargo esta interfaz está siendo reemplazada por una técnica capaz de reconocer la voz con más precisión. En 1980 el ARPA fundó una organización de trabajo en el desarrollo de sistemas de reconocimiento de voz basado en computador. [43]

Aunque las tecnologías de reconocimiento de voz han avanzado notablemente los sistemas basados en DTMF todavía sobreviven debido a que son menos costosos y de desarrollo rápido comparado con los sistemas de voz natural. [44]

La tecnología IVR moderna se soporta en dos tecnologías afines:

- TTS (Síntesis de Voz)
- ASR (Reconocimiento de Voz)



### 3.1.2. Conversión de Texto a Habla [42][43][44]

Permite a los computadores y otros sistemas electrónicos simular respuestas de voz, es decir, convertir textos escritos en locuciones de voz natural.

Las dos características utilizadas para describir la calidad de la síntesis de voz son la *naturalidad* e *inteligibilidad*. La *naturalidad* se refiere a cuánto el resultado se aproxima a la voz de una persona real. La *inteligibilidad* a la facilidad de entender el habla sintetizada. La síntesis ideal debe de ser a la vez natural e inteligible, y cada tecnología intenta conseguir el máximo de ambas.

A continuación se explican de manera general las tecnologías más utilizadas en la conversión de texto a voz.

- **Síntesis por selección de unidades**

La síntesis por selección de unidades utiliza una base de datos de voz grabada (más de una hora de habla grabada). Durante la creación de la base de datos, el habla se segmenta en algunas o todas de las siguientes unidades: fonemas, sílabas, palabras, frases y oraciones. Típicamente, la división en segmentos se realiza usando un reconocedor de voz modificado para forzar su alineamiento con un texto conocido. Después se corrige manualmente, utilizando representaciones como la forma de onda y el espectrograma. Se crea un índice de las unidades en la base de datos basada en parámetros acústicos de la segmentación como la duración, la posición en la sílaba y los fonemas vecinos. En tiempo de ejecución, el objetivo deseado se crea determinando la mejor cadena de candidatos de la base de datos (selección de unidades). Este proceso se logra utilizando fundamentalmente un árbol de decisión ponderado adecuadamente.

La selección de unidades da la máxima naturalidad debido a que no aplica en gran medida el procesamiento digital de la señal al habla grabada, lo que a menudo hace que el sonido grabado suene menos natural, aunque algunos sistemas usan un poco de procesamiento de señal en la concatenación para suavizar las formas de onda. En este sentido, la salida de la mejor selección de unidades es a menudo indistinguible de la voz humana real, especialmente en contextos en los que el sistema ha sido adaptado. Por ejemplo, un sistema de síntesis de voz que se encargue de dar informaciones de vuelos puede ganar en naturalidad si la base de datos fue construida a base de grabaciones de informaciones de vuelos, ya que será más probable que aparezcan unidades apropiadas e incluso cadenas enteras en la base de datos. Sin embargo, la máxima naturalidad a menudo requiere que la base de datos sea muy amplia, llegando en algunos sistemas a los Gigabytes de datos grabados.

- **Síntesis de difonos**

La síntesis de difonos utiliza una base de datos mínima conteniendo todos los difonos que pueden aparecer en un lenguaje dado. El número de difonos depende de la fonotáctica del lenguaje: el español tiene unos 800 difonos, el alemán unos 2500. En la síntesis de difonos, la base de datos contiene un sólo ejemplo de cada difono. En tiempo de ejecución, la prosodia de una oración se impone a estas unidades mínimas mediante el procesamiento digital de la señal, como codificación lineal predictiva, PSOLA o MBROLA.



- **Síntesis específica para un dominio**

La síntesis específica para un dominio concatena palabras y frases grabadas para crear salidas completas. Se utiliza en aplicaciones donde la variedad de textos que el sistema puede producir está limitada a un dominio particular, como anuncios de salidas de trenes o información meteorológica.

Esta tecnología es muy sencilla de implementar, y se ha utilizado comercialmente durante largo tiempo en aparatos como relojes y calculadoras parlantes. La naturalidad de estos sistemas puede ser muy alta porque la variedad de oraciones está limitada y corresponde a la entonación y la prosodia de las grabaciones originales. Sin embargo, al estar limitados a unas ciertas frases y palabras de la base de datos, no son de propósito general y sólo pueden sintetizar la combinación de palabras y frases para las que fueron diseñados.

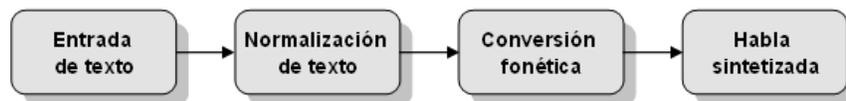
### 3.1.2.1. El proceso de la síntesis del habla [45]

En idiomas como el español y el portugués los sintetizadores permiten localizar las vocales sobre las que recae el acento para evitar ambigüedades.

Otra mejora que ofrecen los sintetizadores de voz es la localización de pausas no marcadas ortográficamente, estas pausas se insertan cuando se detecta que no es posible leer un fragmento demasiado largo sin hacer ninguna pausa intermedia.

Los programas de conversión de texto a voz toman la mayoría de entradas desde un archivo de texto. Mientras se recibe el texto, es necesario convertirlo a Unicode o a otro formato nativo.

Como se observa en la Figura 4, los sintetizadores actuales realizan procesos complementarios para mejorar la calidad de la salida de voz, esta tarea consiste en asignar palabras convencionales a números, abreviaturas o símbolos para facilitar su correspondencia con el lenguaje hablado, a este proceso se le denomina: normalización.



**Figura 4: Proceso de Síntesis del Habla**

El proceso de normalizar texto es pocas veces simple. Los textos están llenos de homógrafos, números y abreviaturas que tienen que ser transformados en una representación fonética. Sin embargo, en lenguas donde la correspondencia entre el texto escrito y su equivalente fonético es poca (por ejemplo el inglés) o ninguna (por ejemplo el mandarín), la creación de estos sistemas se complica.

Muchos sistemas de texto a voz no generan representaciones semánticas de los textos de entradas, ya que los sistemas para hacerlo no son fiables o efectivos. Como resultado, se utilizan varias técnicas heurísticas para estimar la manera correcta de eliminar la ambigüedad de los homógrafos, y entre estas se tienen buscar palabras vecinas y usar estadísticas sobre la frecuencia de aparición de las palabras.



Decidir como convertir números en palabras es otro problema que tienen que solucionar los sintetizadores de voz. Es un desafío en gran parte simple el programar un sistema que convierta números en palabras, como por ejemplo transformar 1325 en "mil trescientos veinticinco". Sin embargo, los números aparecen en diferentes contextos, y mientras 1325 puede ser un número ordinal, "uno tres dos cinco" pueden ser los últimos dígitos de una identificación o "trece veinticinco" un número de teléfono. A menudo un sistema de síntesis de voz puede inferir como expandir un número con base en las palabras o números vecinos y la puntuación, y algunos sistemas proporcionan un sistema para especificar el tipo de contexto si este resulta ambiguo.

El paso siguiente a la normalización del texto es la conversión fonética que se describió anteriormente (síntesis por unidades, dítonos, etc.), hasta llegar obtener habla sintetizada representada por muestras o archivos de audio.

### **3.1.3. Reconocimiento Automático del Habla (RAH) [46]**

Se define como la conversión de una señal acústica (recogida por un micrófono o por un teléfono) en su equivalente escrito. Los sistemas ASR se basan en un modelo de generación de habla estudiando el proceso de comunicación humano, donde el flujo de aire es modulado para producir potencia acústica en un determinado rango de frecuencias.

En primer lugar, se debe distinguir los sistemas de reconocimiento del habla de los sistemas de comprensión del habla.

Se suele considerar que la comprensión del habla es un concepto más amplio, que si bien incluye entre otras partes a un sistema de reconocimiento su objetivo es el de capturar la semántica del mensaje, y de esta forma no solamente transcribirlo en texto sino entenderlo correctamente.

La interfaz de voz es muy prometedora puesto que actualmente la mayoría de la gente sigue tecleando unas 60 palabras por minuto (en el mejor de los casos) cuando podría llegar a pronunciar unas 200 en el mismo tiempo. Sin embargo, las dificultades asociadas con estos sistemas son:

- Cantidad de información en una señal de voz.
- Flujo continuo de voz (concatenación de palabras).
- Variabilidad entre las palabras dichas del mismo locutor (debido a la velocidad, estado de ánimo, etc).
- Variabilidad entre locutores.
- Ruido, calidad del micrófono, etc.

Por otro lado, el respaldo tecnológico que contribuye al avance de esta tecnología está relacionado con los siguientes campos:

- Microprocesadores más poderosos
- Integración de computadoras y telefonía (CTI)
- Crecimiento de aplicaciones multimedia



### 3.1.3.1. Tipos de motores de reconocimiento [47][48]

- 1) *En función del hablante*: según el grupo de locutores que se utilizan en el entrenamiento y evaluación del sistema se pueden distinguir 3 tipos de reconocedores:
  - Reconocedores *monolocutores* y se evalúa para un único hablante.
  - Reconocedores *multilocutor* el entrenamiento y evaluación se realiza sobre un grupo de locutores específico.
  - Motores *independientes del hablante* en los cuales la evaluación del sistema se efectúa sobre un conjunto de locutores diferentes para entrenarlo.
  
- 2) *En función de la manera de hablar y la aplicación*: existen los siguientes 3 tipos:
  - Reconocedores de palabras completas emitidas de forma aislada entre sí.
  - Reconocedores de palabras conectadas, se utilizan las palabras como unidades de reconocimiento, pero éstas pueden ser emitidas secuencialmente y con pausas entre ellas.
  - Reconocedores de voz continuos, los cuales realizan las tareas atendiendo a unidades más pequeñas que la palabra (fonemas, difonos, etc) y sobre frases completas sin necesidad de establecer silencios entre las palabras que las constituyen.
  
- 3) *En función de las distorsiones de la señal*: se pueden tener en cuenta las distorsiones producidas en la señal de voz por distintos factores. En este sentido, se pueden separar los reconocedores en dos tipos:
  - *Reconocedores de habla limpia*, son aquellos entrenados y probados en condiciones de laboratorio.
  - *Reconocedores robustos*, los cuales permiten su uso en ambientes reales donde varios factores complican la tarea (ruido, reverberación, etc.).

### 3.1.3.2. Análisis de la Señal Acústica [47][48]

El proceso de resonancia acústica es el más importante para determinar las propiedades del sonido. La principal estructura resonante, particularmente para vocales, es el tracto vocal, que inicia desde la laringe y se extiende hacia arriba en la faringe la boca y los labios. En algunos sonidos también se incluye la cavidad nasal que hace más complicado el sistema resonante.

Los modos resonantes del tracto vocal son conocidos como formantes y por lo general son numerados desde la frecuencia más baja. Por consiguiente son referidos como  $F_1$ ,  $F_2$ ,  $F_3$ . En general  $F_1$  y  $F_2$  usualmente en el rango de 250hz a 3Khz son las más significativas en la determinación de las propiedades fonéticas de los sonidos del habla. Pero algunos formantes de frecuencia más alta pueden ser importantes para ciertos fonemas

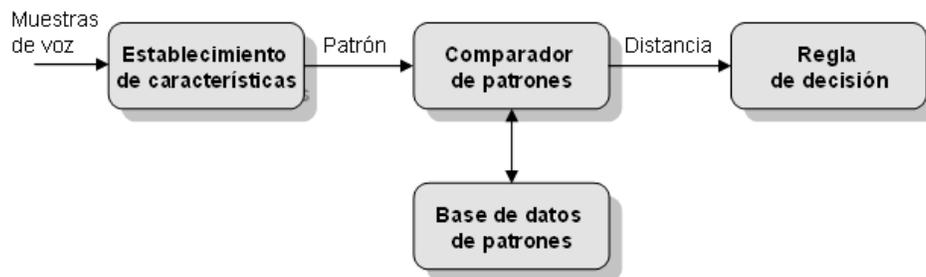
La frecuencia fundamental de la señal de habla se encuentra típicamente en el rango de 50-200Hz, para hombres adultos y cerca de una octava más alta para mujeres.

Uno de los métodos empleados con el fin de determinar las propiedades del habla es el espectro de señal de corto tiempo o función de ventanas de tiempo, equivalente a la magnitud de la



transformada de Fourier multiplicada por una franja de tiempo (unos pocos milisegundos), y el reconocimiento basado en *patrones*, que se verá a continuación.

### 3.1.3.2.1. Reconocimiento basado en Patrones [48]



**Figura 5: Proceso de Reconocimiento del Habla basado en patrones**

Esta teoría ha dado lugar a toda una familia de técnicas de reconocimiento de voz, causando un considerable avance durante las décadas de los 70's y 80's.

El principio, que se puede ver en la Figura 5, consiste en lo siguiente: se almacenan patrones de voz típicos, como modelos de referencia, en un diccionario de palabras candidato. El reconocimiento se lleva a cabo comparando la expresión desconocida con los patrones almacenados y seleccionando el que más se le parece.

Por lo general, se construyen patrones para las palabras completas. La ventaja es que se evitan errores debidos a la segmentación o clasificación de unidades pequeñas que pueden variar acústicamente, como los fonemas. Y la desventaja es que cada palabra requiere de su patrón y el tiempo para preparar y compararlos se vuelve demasiado grande al incrementar el tamaño del vocabulario.

De este método, nació uno que ha sido ampliamente utilizado en el desarrollo de reconocedores de voz, el cual corresponde al de Modelos Ocultos de Markov (MOM).

### 3.1.3.2.2. Modelos Ocultos de Markov [49]

Los modelos ocultos de Markov (MOM) constituyen una de las técnicas que se ha utilizado con más éxito en el reconocimiento automático del habla (RAH).

Principalmente, esta técnica ha permitido modelar adecuadamente la gran variabilidad en el tiempo de la señal de voz. En la terminología del ASR con MOM suele hacerse referencia no sólo a la técnica de los modelos ocultos de Markov, propiamente dicha, sino también a una larga lista de adaptaciones y técnicas asociadas que se fueron incorporando para solucionar el problema de RAH.

Si se considera el proceso de la comunicación oral se podría pensar que para cada texto el locutor activa un sistema y da como salida una determinada emisión sonora. Para comenzar a entender como se aplican los MOM al RAH se debe empezar por imaginar que para cada una de las posibles emisiones se puede encontrar un modelo capaz de imitar al sistema y activado por el locutor. Es decir, un modelo que sea capaz de generar la misma emisión que generó el locutor a



partir del texto que había en su mente. De esta forma, se puede suponer que se cuenta con tantos modelos como posibles emisiones pueda hacer el locutor y, para cada modelo un texto asociado. En caso de que se conozcan perfectamente estos modelos, se podría utilizar el camino inverso para resolver el problema de RAH. Teniendo una determinada emisión del locutor, lo que se haría es buscar cuál de todos los modelos generaría el sonido más parecido al que generó el locutor. Al encontrar el modelo que genera el sonido más parecido a la emisión del locutor entonces también se habrá encontrado el texto, ya que se había dicho que todos los modelos estaban asociados a un determinado texto.

### **3.2. SERVICIOS IVR EN IMS**

Uno de los objetivos del presente Trabajo de Grado es la implementación de un prototipo de cliente SIP para dispositivos móviles con capacidades de acceso a servicios IVR en base a la arquitectura de servicios IMS. Por lo tanto, antes de exponer las características, requerimientos y provisión de servicios IVR soportados por IMS, es necesario conocer y definir el escenario a través del cual los usuarios van a tener acceso a estos. El primer tema que se debe tratar en esta sección es entonces el acceso a los servicios VoIP para usuarios móviles. Una vez identificado y caracterizado el escenario de acceso, se procederá a describir la forma como la arquitectura IMS especifica actualmente la prestación de servicios de VoIP, más específicos, como lo son los servicios de IVR (ASR y TTS), enfocando el estudio en aquellos que se ajustan a las capacidades tecnológicas actuales de las redes de 2.5G.

#### **3.2.1. La Voz Sobre IP (VoIP) en las Redes Móviles [4][18][50]**

VoIP es la tecnología que permite el soporte de comunicaciones de voz en redes IP. Apareció a mediados de los años 90, y probablemente fue en el año 1998 cuando disfrutó de su primer despegue importante. Pero solo hasta el año 2003 se puede considerar el ascenso definitivo de esta tecnología.

La red telefónica basada en la conmutación de circuitos se ha utilizado para la transmisión de la voz durante más de 100 años. Sin embargo, en la actualidad, el crecimiento del tráfico y de las redes de datos es muy superior al de voz. Las redes IP son las redes de paquetes más comúnmente extendidas en todo el mundo, por lo que VoIP se convierte en la solución más atractiva, dado los bajos costos de Internet; enmarcándose además en la convergencia hacia una red multiservicio única.

Los operadores móviles han hecho una clara apuesta de futuro por esta tecnología, como queda reflejado en los últimos Releases 5 y 6 del 3GPP con la introducción del subsistema IMS basado en SIP. Sin embargo, las características de la interfaz aire en comparación con los sistemas de transmisión cableados, por un lado, y la movilidad de usuario que ofrece el sistema, por otro, dificultan la adopción de la tecnología IP y requieren una mayor adecuación técnica para cumplir los requisitos estrictos de temporización y QoS que el servicio clásico de voz impone, además de otros requerimientos adicionales tales como la codificación de voz, interconexión de redes, etc.

A continuación, se describen de manera resumida algunos temas que se deben tener en cuenta al hablar de la VoIP.



- **La QoS en VoIP**

Las redes de paquetes degradan por su naturaleza la calidad de servicio de las comunicaciones de voz, debido a varios factores:

*El retardo.* Los participantes en una conversación telefónica consideran retardos inaceptables, extremo a extremo, aquellos superiores a 200-250ms. Este retardo depende de la carga del sistema.

*El jitter.* Se asocia a la variación del tiempo entre las llegadas de los paquetes pertenecientes a un mismo flujo. La solución natural al jitter es el empleo de un buffer en recepción que amortigüe la varianza del retardo (en el buffer entrarán paquetes con jitter y se sacarán a una tasa constante). Sin embargo, el buffer repercute negativamente en el retardo.

*La pérdida de paquetes.* Debido a la característica temporal de las comunicaciones de voz, la retransmisión de los paquetes no es factible. Las dos técnicas empleadas para recuperación ante pérdidas son:

- Interpolar la información perdida a partir de la conocida. Es una solución acertada cuando las pérdidas son pocas.
- Enviar información redundante. Es menos sensible a las pérdidas de ráfagas, pero aumenta el retardo y el consumo de ancho de banda.

- **Codificación de la voz**

Las premisas de la codificación de la voz son obtener representaciones que sean compactas, que degraden poco la señal original y que requieran poca capacidad de procesamiento. En VoIP se emplean codecs con tasas más de 10 veces inferiores a los 64Kbps de la G.711 (MIC), que es el estándar en las redes de telefonía, todo ello con una calidad de voz bastante aceptable.

La ITU-T normaliza los esquemas de codificación de voz CELP, MP-MLQ, PCM y la Modulación por Impulsos Codificados Diferencial y Adaptativa (Adaptive Differential Pulse Code Modulation, ADPCM) en sus recomendaciones de la serie G. Entre los estándares de codificación más populares para telefonía y voz por paquetes, además del G711, se incluyen: el G.726 (codificación ADPCM a 40, 32, 24, 16 Kbps), el G.729 (describe la compresión CELP que permite que la voz sea codificada en corrientes de 8Kbps), y el G.723.1 (describe una técnica de compresión que se puede utilizar para comprimir voz u otros componentes de la señal de audio de los servicios multimedia a una baja velocidad de bit).

- **Interconexión de redes**

Entre las capacidades que deberá tener una red IP para poder ofrecer servicios de VoIP (Computador a Computador, Computador a Teléfono, Teléfono a Teléfono, etc.) tiene gran importancia la interoperación con la PSTN.

### 3.2.1.1. Establecimiento del camino de habla

Así como en las redes de VoIP fijas, en las móviles la voz digitalizada de cada usuario se transporta en paquetes IP a través de la red y entre los terminales de usuario. El camino que toman estos paquetes a través de la red es el camino del habla. A diferencia del entorno de la conmutación de circuitos, los paquetes pueden tomar rutas diferentes a través del núcleo de la red



IP para llegar a un punto común, en lugar de estar forzados a seguir un circuito específico. Sin embargo, si la red no está congestionada es probable que los paquetes sigan una misma ruta.

Para el establecimiento de un camino para el habla y para sincronizar a los usuarios y sus equipos, se programan funcionalidades de control de llamada tanto en los equipos de usuario como en la red. Dichas funciones de control de llamada se comunican utilizando mensajes de señalización. El control de la llamada, por ejemplo, permite poder pasar al terminal del usuario las direcciones del punto final para poder establecer el camino del habla, la negociación de los recursos, que se necesitan para la llamada, tanto de la red como del equipo de usuario, entre los cuales se tienen los codecs y la calidad de servicio que se requiere.

### 3.2.1.2. Señalización

Los dos protocolos principales de señalización para la VoIP son SIP de la IETF y H.323 de la UIT. Es probable que ambos protocolos convivan a mediano plazo, pero dependiendo del escenario dentro del cual un determinado tipo de servicio se despliegue, se debe decidir qué protocolo es más válido. El presente trabajo se enmarca dentro de un escenario basado en la arquitectura de servicios IMS, por lo cual es SIP el protocolo utilizado para la señalización de los servicios VoIP que se van a estudiar en las secciones posteriores.

En la Figura 6 se muestra la pila de protocolos utilizada para VoIP al emplear SIP como protocolo de señalización y RTP como el protocolo utilizado para la transmisión de los datos de voz en los terminales móviles.



**Figura 6: Pila de protocolos utilizada en VoIP para redes móviles**

El protocolo de señalización SIP se utiliza para el registro y el control de la llamada. En términos generales, el control de la llamada hace referencia a la invitación y sincronización de varios participantes en una llamada. Esto permite a los participantes describir y compartir la información de las características de sus terminales y del camino de habla a utilizar entre ellos. La anterior se conoce como la información de descripción de la sesión, y puede incluir, por ejemplo, el codec a utilizar para el habla y el ancho de banda necesario para los caminos de esta.

SIP incluye mecanismos de confiabilidad que pueden utilizarse cuando el transporte no es seguro, pero estos se pueden omitir si se trata de protocolos de transporte como el Protocolo de Control de la Transmisión (Transmission Control Protocol, TCP) o el Protocolo de Control del Flujo de la Transmisión (Stream Control Transmission Protocol, SCTP). Estos protocolos son transportados sobre IP, la red GPRS, y el dominio IM del núcleo de la red IP como se observa en la Figura 7.

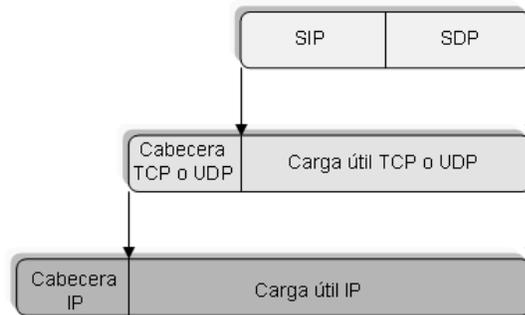


Figura 7: Transporte de SIP y SDP en IP

### 3.2.1.3. Transporte de los paquetes de habla

Con el fin de que el habla se pueda enviar y recibir en paquetes IP, el habla del usuario se muestrea por medio del equipo de él mismo y se codifica para la transmisión (utilizando por ejemplo el codec de voz de multi-tasa adaptativa AMR). Una vez se halla tomado un número de muestras, que usualmente se encuentran entre los 10 y los 40 milisegundos, las muestras codificadas se empaquetan y se envían a la red. El tiempo que toma realizar el empaquetamiento de las muestras de habla, le adiciona un retardo considerable al camino de datos por lo cual se puede necesitar dispositivos de cancelación de eco, ya sea dentro del terminal de usuario o dentro de la red. Sin embargo, sería ineficiente enviar simplemente paquetes de muestras de habla más pequeños, ya que esto incrementaría el ancho de banda necesario, además de los requerimientos de enrutadores IP adicionales, para encaminar más paquetes de habla. Adicional al retardo en la transmisión extremo a extremo, también se encuentra uno al final del camino del habla, donde los paquetes tienen que pasar a través de un buffer para que el habla digitalizada se sincronice antes de que se reproduzca, y este retardo es debido al jitter de los paquetes.

Los paquetes de habla son transportados en paquetes del Protocolo de Datagramas de Usuario (User Datagram Protocol, UDP) e IP por la red GPRS y el núcleo de la red IP, entre los equipos de usuario. Para sincronizar las muestras y controlar la tasa de muestreo, se requiere de un protocolo de entramado. Los protocolos de la IETF para el entramado de la voz y de la multimedia son RTP y RTCP, los cuales son llevados en paquetes UDP/IP. Esto se muestra en la Figura 8.

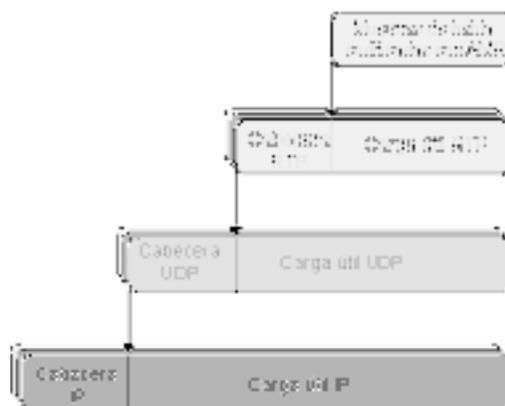


Figura 8: Transporte del habla en IP



Para que el equipo de usuario sea capaz de enviar y recibir paquetes de habla hacia y desde el dominio IMS, este debe activar de manera bidireccional el contexto del Protocolo de Paquetes de Datos (Packet Data Protocol, PDP), entre él mismo y el dominio IMS. De esta forma se podrá asignar el ancho de banda y la calidad de servicio requerida, para el transporte de los paquetes de habla, sobre la UTRAN y la red GPRS. El punto de entrada al dominio IMS va a tener cortafuegos para la seguridad y la prevención de ataques a servicios, los cuales a su vez también pueden controlarse dinámicamente por medio del control de la sesión, esto último con el fin de prevenir el envío o recepción de paquetes de habla antes de que se establezca una sesión. Mediante la desactivación del contexto PDP GPRS adecuado, se puede desconectar el camino de habla entre el equipo de usuario y el dominio IMS.

La red GPRS especifica una clase de servicio conversacional para proporcionar una calidad de servicio mayor que la del "mejor esfuerzo". En esta se priorizan los paquetes de habla por medio de los parámetros: retardo bajo y jitter bajo. De forma semejante, para proporcionar una calidad de servicio alta dentro del dominio IMS se pueden utilizar mecanismos tales como Diffserv o el Protocolo de Reservación de Recursos (Resource Reservation Protocol, RSVP).

El codec por defecto que todos los terminales de la Release 5 deben soportar es el AMR, aunque también se podrían soportar otros codecs. Debido a que la señalización del control de la sesión soporta una operación de traducción libre, la codificación de habla AMR se puede utilizar extremo a extremo entre los terminales de usuario, sin la necesidad de tener que traducir a otro estándar. Sin embargo, con una codificación del habla AMR a 12.2kbps, y un muestreo de habla de 20ms, da como resultado una carga útil de habla RTP de valor igual a la mitad del tamaño de la combinación de los encabezados de los paquetes IP, UDP y RTP. Lo anterior produce un uso del ancho de banda ineficiente, especialmente en el acceso radio que es bastante costoso. Al incrementar el tamaño de las muestras se reduce el problema, pero se incrementa el retardo de los paquetes de habla extremo a extremo, así como se incrementa la probabilidad de la pérdida de paquetes en la interfaz radio. Una solución a lo anterior es llevar a cabo la compresión de encabezados entre el equipo de usuario y la UTRAN, que es donde el ancho de banda es más costoso.

Un beneficio de la utilización de GPRS para llevar los caminos del habla, así como con los de señalización, es que GPRS controla el "handover", o la transferencia de los caminos del habla de una estación base a otra, mientras los usuarios se mueven entre las celdas de radio. Sin embargo, se requiere que se mejoren los procedimientos de "handover" de GPRS, para asegurar la calidad de servicio requerida por la voz cuando ocurre este proceso.

### **3.2.2. Servicios Multimedia en IMS [51][52][53]**

A medida que las arquitecturas de redes de telecomunicaciones han evolucionado, ha habido una tendencia fuerte hacia la centralización de las capacidades de procesamiento multimedia en un nodo lógico y especializado. Este se conoce genéricamente como el Servidor de Medios, y de manera equivalente IMS lo ha llamado la MRF, la cual se descompuso posteriormente en dos subnodos llamados el Controlador de la Función de Recursos Multimedia (Media Resource Function Controller, MRFC) y el Procesador de Funciones de Recursos Multimedia (Media Resource Function Processor, MRFP). El MRFC muestra una interfaz SIP a los otros componentes de la red, y proporciona las funciones de señalización y control, mientras que el MRFP proporciona las capacidades de procesamiento de medios.



Típicamente la MRF es invocada por las aplicaciones que se encuentran corriendo en el Servidor de Aplicaciones (Application Server, AS) y a través del CSCF. La interfaz que el componente MRFC, de la MRF, muestra a la red está basada en SIP. La multimedia termina y se origina desde el MRFP, a través de la interfaz Mb, además el MRFP utiliza RTP para el transporte de la multimedia a través de las redes IP.

En los despliegues modernos, los servicios a gran escala del reconocimiento del habla y de la síntesis de la misma se proporcionan por servidores de habla especializados, los cuales son controlados por un protocolo de la IETF que tiene el nombre de Protocolo de Control de los Recursos Multimedia (Media Resource Control Protocol, MRCP). El MRFC incorporará un cliente MRCP el cual se encarga del control de los servidores de habla distribuidos y el MRFP suministrará y recibirá los medios de los servidores de habla.

A continuación, se describe con más detalle los requerimientos funcionales y el flujo de información que genera los procedimientos entre el MRFC y el MRFP, además de su interacción con las entidades de la arquitectura IMS que necesitan de dichos procesos. El desarrollo de esta temática se basa en los Reportes y Especificaciones Técnicas del 3GPP que abordan los Servicios de Habla dentro de la arquitectura IMS.

### 3.2.2.1. Arquitectura de la MRF

La arquitectura básica de la MRF se puede ver en la Figura 9.

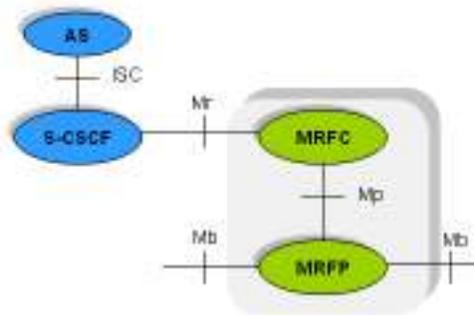


Figura 9: Arquitectura de la MRF

Las tareas del MRFC son las siguientes:

- Controlar los recursos de los flujos multimedia del MRFP.
- Interpretar la información que proviene del AS y del S-CSCF (por ejemplo el identificador de la sesión) y en consecuencia controla al MRFP.
- Generar los CDRs

El punto de referencia Mp le permite al MRFC controlar los recursos de los flujos multimedia que provee el MRFP. El punto de referencia Mp tiene las siguientes propiedades:

- Obedece completamente al estándar H.248 [26]
- Se puede llevar a cabo un trabajo de definición de extensiones sobre la interfaz para una arquitectura abierta.

Las tareas del MRFP son las siguientes:

- Controlar la entidad portadora en el punto de referencia Mb.



- Proporcionar los recursos que va a controlar el MRFC.
- Mezclar los flujos multimedia de entrada (cuando por ejemplo se manejan múltiples partes).
- Originar flujo multimedia (por ejemplo para anuncios multimedia).
- Procesar flujo multimedia (por ejemplo análisis de medios).
- Control de piso o “floor” (por ejemplo la gestión de los derechos de acceso para compartir recursos durante una conferencia).

Las tareas de un Servidor de Aplicaciones con relación a la MRF son: facturación de conferencia y gestión de la información de facturación (momento en el que empezó, duración, lista de participantes).

El protocolo que se utiliza en el punto de referencia Mr es SIP.

### **3.2.2.2. Interacciones que involucran al MRFC y al MRFP**

- **Interacciones entre el UE y el MRFC**

En algunos casos un operador puede querer que el MRFC este disponible directamente para el UE, por ejemplo para soportar sesiones multipartita “de igual a igual” iniciadas por el UE.

- **Interacciones entre el MRFC y el AS basadas en el control del servicio**

Basado en el control del servicio, los recursos del MRFC y el MRFP también se pueden utilizar dentro de una red IMS. En este caso es el AS el que interactúa con el MRFC, y los mensajes de control de la sesión se pasan desde este al MRFC a través del S-CSCF.

### **3.2.2.3. Requerimientos funcionales para la provisión de capacidades multimedia**

IMS no define las aplicaciones o servicios que pueden ofertarse al usuario final, sino la infraestructura y capacidades del servicio que los operadores o proveedores de servicio pueden emplear para construir sus propias aplicaciones y producir su oferta de servicios. En este sentido, IMS no impone límites, son la capacidad de la red de acceso y las características de los terminales las que las fijan.

Las capacidades que la MRF le proporciona a la arquitectura IMS son: reproducción de tonos, reproducción de anuncios, conversión de texto a habla, grabación de audio, detección de Tono Dual Multifrecuencia (Dual Tone Multi Frequency, DTMF), reconocimiento automático de habla, reproducción de multimedia, grabación de multimedia, conferencia de audio, conferencia multimedia, entre otras.

A continuación, se describen los requerimientos funcionales de las capacidades de conversión de texto a habla, reconocimiento automático de habla, y detección DTMF.

- **Conversión de Texto a Habla**

- El MRFC puede estar en capacidad de pedirle al MRFP que reproduzca el texto a una de varias, a múltiples o todas las partes que se encuentren conectadas en una llamada o sesión, ya sea por valor (indicando directamente el texto) o por referencia (a través de una URI).



- El MRFC puede estar en capacidad de pedirle al MRFP que reproduzca un texto en un bucle hasta que le ordene detenerse.
- El MRFC puede estar en capacidad de pedirle al MRFP que reproduzca un texto durante un número fijo de veces.
- El MRFC puede estar en capacidad de pedir detección DTMF durante la reproducción de un texto.
- El MRFC puede estar en capacidad de pedirle al MRFP que termine la reproducción de un texto cuando se detecte un dígito DTMF.
- El MRFC puede estar en capacidad de indicarle al MRFP qué tipo de lenguaje se va a utilizar para reproducir el texto (el tipo de lenguaje hace referencia al lenguaje fuente del texto, por ejemplo puede ser no necesaria una función de traducción en el MRFP).
- El MRFC puede estar en capacidad de pedirle al MRFP que indique cuando un texto se ha reproducido de manera exitosa.
- El MRFP puede estar en capacidad de indicar los casos en que ocurren errores, entre los cuales, se tiene cuando el texto no se reprodujo exitosamente.

En este apartado se realiza una anotación relacionada con la poca claridad que se tiene acerca de cómo el MRFC obtiene el texto y en qué formato se debe encontrar (tamaño, si esta segmentado para que se concatene, etc.), ya que esto depende de la definición de la interfaz Mr.

- **Reconocimiento Automático de Habla**

La función del Reconocimiento Automático de Habla consiste en que un reconocedor procese la voz de entrada del usuario y la compare con unos “datos específicos”, para luego producir un resultado que represente la detección de la misma. En IMS, el MRFP actúa como el reconocedor, y este se encuentra bajo el control del MRFC.

- El MRFC está en capacidad de pedirle al MRFP que inicie el reconocimiento automático del habla.
- El MRFC puede estar en capacidad de pedir la detección DTMF cuando se está ejecutando el ASR.
- El MRFC puede estar en capacidad de pedirle al MRFP que detenga el ASR en el caso en que se detecte un dígito DTMF predefinido.
- El MRFC puede estar en capacidad de indicarle al MRFP el tipo del lenguaje del suscriptor.
- El MRFC puede estar en capacidad de indicarle al MRFP los datos específicos con los cuales se va a comparar la voz de entrada.
- El MRFC puede estar en capacidad de pedirle al MRFP que le indique cuando un proceso de ASR específico se halla culminado exitosamente.
- Cuando un proceso de ASR se culmina exitosamente el MRFP puede notificarle al MRFC el nivel de confidencialidad.
- El MRFP estará en capacidad de indicar los casos de error en los cuales el proceso de ASR no se ejecutó exitosamente.

En este apartado se realizan las siguientes anotaciones:

- Se encuentra para estudio futuro la configuración del dígito DTMF predefinido, ya que puede ser dentro del AS, MRFC o del MRFP.
- La definición de los “datos específicos” es para estudios futuros.
- Se necesita definir el nivel de confidencialidad.



- **Detección DTMF**

- El MRFC debe estar en capacidad de pedirle al MRFP que detecte y reporte los dígitos DTMF.

En este apartado se realiza la anotación de la necesidad de revisar si se requiere detección de tonos DTMF dentro de banda, adicionalmente al informe de los Eventos de Telefonía DTMF RTP. Más adelante, cuando se reporten los eventos de telefonía DTMF, se necesita revisar si se debe reportar su recepción (para qué propósito, donde se valida el evento de telefonía DTMF, duración de un tono DTMF válido).

#### **3.2.2.4. Procedimientos del MRFC y MRFP en la provisión de ASR, TTS, y detección DTMF**

En la especificación técnica donde se trata la descripción de los procedimientos del MRFC y el MRFP para la provisión de capacidades multimedia, aún están por definirse los procedimientos para la conversión de texto a habla, el reconocimiento automático de habla, y la detección DTMF.

#### **3.2.2.5. Requerimientos de la arquitectura IMS para los servicios de reconocimiento del habla**

Los servicios de habla generan requerimientos adicionales a la arquitectura IMS, ya que se deben aplicar una serie de técnicas para mejorar el desempeño en lo que respecta al reconocimiento del habla. Con base en un Reporte Técnico del 3GPP, en esta sección se describe la forma como debe mejorarse el sistema 3GPP existente para proporcionar la señalización que controle estas técnicas.

##### **3.2.2.5.1. Técnicas para mejorar el desempeño del reconocimiento del habla**

Las plataformas de ASR tienden a realizar la siguiente secuencia de operaciones: cancelación de eco, extracción e interpretación. El proceso de cancelación de eco se utiliza para permitir el envío de comandos a la plataforma ASR mientras ésta se encuentra reproduciendo anuncios de voz. (Sin la cancelación de eco la plataforma ASR no podría distinguir entre su voz y la voz del usuario). En el proceso de extracción, se extraen los fonemas de la señal de habla de entrada y se transforman en palabras mediante los modelos acústicos. Luego, el motor de reconocimiento de habla realiza una búsqueda de las palabras pronunciadas en la gramática creada por el Lenguaje Gramatical Específico (Grammar Specific Language, GSL). El proceso de interpretación se utiliza para extraer la interpretación semántica de la secuencia de palabras.

- Supresión adicional de ruido en el terminal. El rendimiento del ASR se puede mejorar si este le pide al Equipo de Usuario (User Equipment, UE) que adicione supresión extra de ruido.
- Deshabilitar la transmisión no continua (Discontinuous Transmission, DTX) para mejorar el desempeño del ASR. En un caso normal, el terminal utiliza la transmisión no continua para reducir las transmisiones en el enlace de subida. Las tramas descriptoras de silencio se envían a una velocidad baja y la Unidad de Adaptación de Tasa y Traducción (Transcoder & Rate Adaptation Unit, TRAU) las utiliza para enviar en el enlace de subida ruido aceptable para mejorar la calidad normal del habla. Una consecuencia de esto es que durante una parte pequeña de la conversación el codec de habla codifica el ruido de fondo durante los periodos de silencio. Esto generará problemas en la plataforma ASR debido a que el codec de habla utilizará los periodos de silencio para estimar el ruido de fondo. Si la unidad ASR pudiera



ordenarle al dispositivo móvil deshabilitar el DTX, entonces el codec de voz siempre se utilizaría para codificar el ruido de fondo, y de esta forma podría permitirse que la unidad ASR realice una estimación más exacta del ruido.

- Indicar la utilización o no del DTX. Es posible que la unidad ASR encuentre como beneficioso el hecho de conocer cuando el DTX esta siendo o no utilizado.
- Indicar a la plataforma ASR el modo del codec. El desempeño del reconocimiento de habla puede ser influenciado por el tipo de codec que se utilice para la compresión en entornos inalámbricos ruidosos. Por lo tanto, resultaría útil proveer a la plataforma ASR una indicación del codec utilizado.
- Ordenar que se proporcione una codificación de alta velocidad para el modo de reconocimiento de habla. El desempeño en el reconocimiento del ASR puede disminuir significativamente por tasas de codificación por debajo de 8-10Kbps. Por lo tanto, es necesario resaltar que durante las sesiones ASR no se considere la optimización de la capacidad.
- Para el modo de banda ancha AMR se necesita la señalización Operación Libre de Tandem (Tandem Free Operation, TFO) en la plataforma ASR. Para mejorar la calidad del habla en llamadas de voz normales el AMR de banda ancha se puede implementar en el terminal. Para llevar la señal de banda ancha a la plataforma ASR es necesario, por ejemplo, algo como el TFO.
- Tener en cuenta los acentos regionales. Las personas de diferentes regiones hablan con diferentes acentos por lo cual pueden sufrir un peor desempeño del ASR. Potencialmente, el HLR (o plataformas CAMEL) podría proporcionar al Centro Móvil de Conmutación Visitado (Visited Mobile Switching Centre, VMSC) la “información del acento regional”, la cual a su vez se pasa a la plataforma ASR para permitir que sus unidades utilicen una gramática específica de acuerdo al acento de una región.

### 3.2.2.5.2. Impactos en la arquitectura IMS por las mejoras en el reconocimiento de habla

La utilización de un codec de Reconocimiento de voz distribuido (Distributed Speech Recognition, DSR) para el reconocimiento del habla puede ocasionar algunos impactos en IMS, los cuales incluyen:

a) Imaginarse el caso en que la plataforma ASR se encuentra ejecutando una aplicación de marcado activado por voz. El móvil necesita utilizar el codec DSR cuando hable con la plataforma ASR, y tendrá que cambiar al codec AMR cuando quiera conectarse con un usuario. El tema importante en este escenario, es la forma como la señalización IMS va a hacer que el codec se cambie entre dichos codecs.

b) Cuando el móvil está utilizando el codec DSR con la plataforma ASR, en el enlace de bajada se tiene que utilizar el codec AMR. Por lo tanto, los parámetros SDP que se incluyen por el móvil, tienen que reflejar estos flujos unidireccionales de medios. Además, es muy probable que esto tenga impacto en muchas de las implementaciones de SIP que esperaban utilizar el mismo codec en ambas direcciones.

El primero de estos temas también aplica si se utiliza el codec AMR para Banda Ancha (Wide Band - Adaptive Multi Rate, WB-AMR) para el reconocimiento de habla y el otro usuario soporta AMR para Banda Estrecha (Narrow Band – Adaptive Multi Rate, NB-AMR) solamente. La utilización del codec NB-AMR para el reconocimiento del habla parece evitar este asunto.

Un servidor ASR puede actuar como un agente de usuario SIP “back-to-back” para la conexión de un usuario con otro, o puede actuar como un agente que transfiere la llamada de un usuario. Así sea que la plataforma ASR este en modo de conexión a un usuario o en el modo de reenvío, el



SDP tiene que soportar la situación de que el enlace de subida utilice un tipo de codec y el enlace de bajada utilice otro tipo de codec.

La manera de modelar un canal asimétrico en SDP es mediante la utilización de los atributos *recvonly* y *sendonly*, lo cual le permite a SIP soportar tanto la simetría como la asimetría en los canales.

Un ejemplo que indique en el SDP de un mensaje INVITE un enlace de subida con DSR y un enlace de bajada con AMR es el siguiente:

```
v=0
o=SESPhone 0 1 IN IP4 10.132.30.33
s=asymmetric ses dsr session
c=IN IP4 10.132.30.33
t=0 0
m=audio 9002 RTP/AVP 97
a=rtpmap:97 AMR/8000
a=recvonly
m=audio 9002 RTP/AVP 101
a=rtpmap:101 dsr-es202212/8000
a=sendonly
```

El anterior ejemplo utiliza IPv4 pero se puede extender fácilmente a IPv6.

### 3.2.3. Servicios IVR en IMS que se adapten a las redes 2.5G actuales

Actualmente, la tecnología de VoIP tiene un grado de desarrollo mucho mayor en las operadoras fijas que en las móviles. Sin embargo, en el camino de evolución hacia lo que se denomina “todo IP” las operadoras móviles han dirigido parte de sus esfuerzos hacia esta tecnología, y un reflejo de ello ha sido la introducción de IMS a los sistemas UMTS, utilizando SIP como protocolo de señalización para el control de las sesiones. El camino de especificación e implementación de las redes 3G de manera completa es aún largo, sin embargo el proceso de estandarización contribuye a la superación de barreras tecnológicas que se presentan debido a la utilización de tecnologías como la VoIP en entornos móviles, donde la principales limitantes son la capacidad y costo actuales del acceso radio, y la falta de garantías para la prestación de servicios con calidad.

El servicio clásico de voz impone requisitos estrictos de temporización, y por lo dicho anteriormente, las características y capacidades actuales de la interfaz aire requieren de una mayor adecuación técnica, que se encuentra en proceso de especificación y estandarización, para llegar a ser un medio de transmisión de datos con buenas prestaciones. Sin embargo, existen aplicaciones de voz empaquetada con requisitos de tiempo menos restrictivos, y cuya puesta en servicio se ha considerado factible en un menor plazo que las llamadas tradicionales con tecnología VoIP. Estos son los casos de los servicios persona a persona donde la voz se transmite a ráfagas a través de un canal semidúplex, o servicios PoC, y los servicios persona a contenido, donde los requerimientos técnicos de acceso son inclusive menores que los primeros ya que la transmisión en tiempo real de la información no es un aspecto obligatorio, y la gestión de la sesión es menos compleja.

Ambos casos permiten un mayor ahorro de recursos de red y admiten una mayor tolerancia al retardo para proporcionar una percepción aceptable por parte del usuario. Esto permitiría su



despliegue sobre la red de datos móvil de forma efectiva, y les permitirá a los operadores introducir y experimentar de forma segura con la voz sobre IP, en un entorno real, a través de un nuevo servicio, sin sustituir ni experimentar con el servicio de llamadas.

Los sistemas IVR proporcionan diversas capacidades de interacción con los usuarios, y dicha interacción está regida por el tipo de servicio que se ofrezca. Los dos tipos de servicios nombrados anteriormente y las diversas aplicaciones que de ellos se desprenden pueden ser proveídas y soportadas por sistemas IVR, sin embargo, por los alcances y objetivos del presente Trabajo de Grado se van a seleccionar los servicios persona a contenido, los cuales permitirán demostrar de una manera básica las funcionalidades ofrecidas por los sistemas ASR y TTS al enmarcarse dentro de la arquitectura IMS.

Entre los servicios persona a contenido, a través de IVR, se pueden encontrar: reservaciones en el área del turismo, servicios de información geográfica y meteorológica, servicios de ayuda a usuarios por parte de empresas, consulta de información personal de usuarios como por ejemplo su correo electrónico o su disponibilidad bancaria, envío de mensajes cortos, etc.



#### **4. HERRAMIENTAS DE PROGRAMACIÓN PARA LA CREACIÓN DE CLIENTES MÓVILES SIP Y SERVICIOS IVR CON BASE EN IMS**

##### **4.1. HERRAMIENTAS PARA EL DESARROLLO DE LOS MÓDULOS QUE COMPONEN LA ARQUITECTURA IMS**

###### **4.1.1. Entornos de Desarrollo**

- **Eclipse [54]**

Eclipse es un Entorno de Desarrollo Integrado (Integrated Development Environment, IDE) multiplataforma y de código abierto, que sirve para crear aplicaciones clientes de cualquier tipo. Actualmente se tiene la versión 3.2.1 del SDK.

Una de sus grandes ventajas es que basa su funcionamiento en Plug-ins o módulos con lo cual es ampliable para que haga prácticamente cualquier cosa, desde edición de XML hasta control de servidores Web como Tomcat, pasando por módulos para otros lenguajes como Perl, php, Shell Script, etc.

- **Netbeans [55]**

Es un IDE multiplataforma con todas las funciones de código abierto y sin costo, para el desarrollo de aplicaciones Java que extienden J2SE, compatibilidad con J2EE 1.4 de segundo nivel, clientes de escritorio, clientes móviles y aplicaciones Web. Actualmente se tiene NetBeans 5.5 con disponibilidad de fuente abierta, como los IDEs anteriores.

- **Borland Jbuilder (Versiones X y 2005) [56]**

Es un entorno de desarrollo robusto a nivel empresarial que cuenta con gran cantidad de asistentes para la creación de proyectos muy diversos. Jbuilder cuenta con un framework de Herramientas Abiertas u “Open Tools” para la creación y adición de componentes, similar al sistema de módulos o “plugins” de Eclipse. Este framework permite personalizar el IDE y aumentar sus funcionalidades.

###### **4.1.2. API para la creación de la Lógica del Servicio**

- **Servlet SIP [57]**

Un Servlet SIP es un componente de aplicación basado en Java el cual realiza el proceso de señalización SIP, y es dirigido por un contenedor de Servlets SIP, quien ejecuta la lógica del servicio y soporta opcionalmente HTTP y la plataforma J2EE. Los Servlets interactúan con clientes SIP mediante el intercambio de mensajes de peticiones y respuestas, a través de un contenedor de servlets.



Esta API soporta la base de SIP definida en el RFC 3261 y también soporta las siguientes extensiones: el entorno de notificación de eventos (RFC 3265) y el método de mensajes para mensajería instantánea (RFC 3428).

La especificación del API pretende permitir a las aplicaciones el control completo sobre la señalización SIP, mientras al mismo tiempo esconde mucho de la complejidad no esencial de SIP, la cual no es relevante para los desarrolladores de aplicaciones.

- **Arquitectura [58]**

- **Contenedor de Servlets**

- Este es un entorno en el cual un servlet puede existir.
- Aquí se carga y se inicializa un servlet.
- Cuando un mensaje SIP llega se encarga de invocar los métodos apropiados.
- El contenedor de SIP Servlet administra recursos como puertos de escucha, los hilos, transacciones y diálogos, el estado de la sesión y los componentes de la aplicación.

- **Servlets**

Son clases con métodos de servicio que se compilan en un archivo SAR (Servlet Archive File)

- **Descriptores de despliegue**

Son archivos basados en XML y que contienen la información de configuración y las reglas para orientar el procesamiento de los mensajes SIP.

#### 4.1.3. Servidores de Aplicaciones

- **Ubiquity SIP Application Server 7.1 [59]**

Este servidor de aplicaciones constituye un entorno de creación de aplicaciones, basado en estándares, el cual permite a los proveedores de servicios el desarrollo, despliegue, y hospedaje de servicios innovadores, basados en aplicaciones convergentes.

El Servidor de Aplicaciones SIP 7.1 permite el desarrollo de aplicaciones para redes de próxima generación, mediante el empleo de una arquitectura de Servlets SIP.

Algunos ejemplos de las aplicaciones que se pueden crear son los siguientes:

- Aplicaciones móviles
- Aplicaciones empresariales
- Aplicaciones de VoIP
- Aplicaciones de conferencia



- **SIPMethod [60]**

SIPMethod es un servidor de aplicaciones SIP que sigue la JSR-116 (API de Servlets SIP), el RFC-3261 (Especificación de SIP) y soporta las extensiones claves de SIP. Esta diseñado para el desarrollo y despliegue de aplicaciones basadas en SIP con la tecnología de Servlets SIP.

El Entorno de Creación de Aplicaciones SIPMethod ACE le permite a los desarrolladores la creación rápida y el despliegue de aplicaciones basadas en Servlets SIP. Las herramientas se construyeron en forma de Plug-ins o módulos de Eclipse.

- **BEA WebLogic Communications Platform [61]**

La Plataforma de Comunicaciones BEA WebLogic consiste en: un servidor de aplicaciones SIP IMS, un servidor SIP WebLogic, la ejecución de políticas poderosas, una plataforma para Servicios Web, y una Gatekeeper de Red BEA WebLogic. Por lo anterior, BEA WebLogic ofrece una plataforma para la creación, entrega y control de servicios IMS.

El servidor de aplicaciones SIP BEA WebLogic soporta completamente la Release 5 del 3GPP, las RFCs de la IETF para SIP necesarias para que este cumpla con los roles de los servidores de red SIP, y las extensiones del 3GPP para el soporte del Control del Servicio IMS (IMS Service Control, ISC).

#### **4.1.4. Servidores de Medios**

##### **4.1.4.1. Servidores de Medios Comerciales**

Entre los Servidores de medios comerciales que se tienen actualmente están: El MRF de la empresa BroadSoft para servicios de voz y video en una red IP [62]; El OpenCall Media Platform Media Resource Function de HP para dar soporte a servicios multimedia interactivos y avanzados [63]; El Brooktrout's SnowShore de IBM, que tiene, entre otras de sus características, el soporte a VoiceXML 2.0, aplicaciones ASR y TTS, y del cual existe un versión demo [64].

##### **4.1.4.2. Sintetizadores de Voz**

- *Sapi 4.0* soporta los motores TTS3000 de Lernout y Hauspie para español, inglés americano, inglés británico, holandés, francés, alemán, italiano, japonés, coreano, ruso y portugués de brasil. [65]
- *Festival* es un sintetizador de voz disponible libremente basado en concatenación de difonos y selección de unidades. Está disponible para español, inglés británico y americano y galés. [66]
- *Flite (Festival-lite)* es una alternativa más pequeña de Festival, diseñado para sistemas empotrados y servidores de gran volumen de trabajo. [67]
- *FreeTTS* escrito enteramente en Java y basado en Flite. [68]
- *MBROLA* es un sistema de concatenación de difonos para unas 25 lenguas. [69]
- *Gnuspeech* es un paquete extensible de texto a voz basado en síntesis por reglas articulatorias en tiempo real. [70]



- *Verbio TTS*. [71]

#### 4.1.4.3. Reconocedores de Voz

Las soluciones más utilizadas son:

- **Dragon NaturallySpeaking SDK, Edition 9 para Cliente o Servidor (Disponible para español) [72]**

Requiere: 1 Giga de espacio en disco duro; tarjeta de sonido de 16 bits compatible con Creative Labs Sound Blaster, solo para Windows. Tiene una precisión alta, buen desempeño y vocabulario extensible (este software incorpora su propio administrador de léxico, llamado Vocabulary Builder). Reconoce voz natural (sin pausas). Permite desarrollo con cualquier lenguaje que soporte Active X, C++, C# y Visual Basic.

- **IBM Via Voice (Disponible para español) [73]**

Es un reconocedor dependiente del usuario, soporta tanto el reconocimiento como la síntesis de voz. Permite N-Best (capacidad de retornar una o más frases probables para que el usuario escoja la correcta), manejo de palabras homónimas e integración con código java ya que este motor utiliza el API de Java para el habla: `jsapi`.

- **L & H Voice Xpress Profesional Versión 4.0 (Sólo para Ingles) [74]**

L & H Voice Xpress Profesional ofrece las características básicas de reconocimiento, es posible dictar en cualquier aplicación, crear macros de voz y controlar las aplicaciones con comandos de voz. El punto fuerte del programa son sus comandos intuitivos con lenguaje natural para Microsoft Word, Excel y el procesador de palabras sencillo: Voice Xpress.

- **Verbio (Atlas) (Disponible en español) [75]**

Basado en modelos ocultos de Markov y redes neuronales. *Verbio* es un conjunto de librerías y utilidades destinadas a conseguir una rápida y sencilla incorporación de herramientas del habla (reconocimiento y síntesis) en aquellas aplicaciones en las que pueda resultar necesario disponer de una interfaz vocal.

Cualquier entorno que disponga de un dispositivo que permita obtener y/o reproducir muestras de audio es susceptible de incorporar las herramientas contenidas en *Verbio*. Es decir, el sistema de reconocimiento requerirá la obtención de las muestras de audio dictadas por el locutor para procesarlas y obtener el resultado de reconocimiento.

Este motor solo acepta audio en formato PCM ley A o  $\mu$ , 8 bits de cuantificación y 16Khz de frecuencia de muestreo. Actualmente solo está disponible para Windows.

- **Sphinx (Solo en Inglés) [76]**

Es un proyecto de código abierto, diseñado en la Universidad Carnegie Mellon. Está basado en las cadenas ocultas de Markov y es dependiente del usuario, lo que requiere un entrenamiento previo.

SphinxII y III están desarrollados en C++, y pueden correr en sistemas Windows y Linux. SphinxIV fue desarrollado en java.



Este motor reconoce audio almacenado en archivos `raw`, con 16 bits de cuantificación y 16Khz de frecuencia de muestreo.

#### 4.1.5. Librerías de C++ para el protocolo SIP y el protocolo RTP

##### 4.1.5.1. Librerías de C++ para el protocolo SIP

- **OSIP [77]**

Es una librería de código abierto, que no implementa la capa de transporte ni la capa de aplicación. Es portable para Windows y Linux. Contiene analizadores sintácticos que convierten los mensajes SIP (basados en texto) en estructuras más fáciles de manipular.

**Características:**

- Cuando se recibe o envía un mensaje SIP se debe registrar en la pila de OSIP para que realice los procesos de transacción.
- Tiene funciones de “*callback*” que se encargan de informar a la aplicación de cualquier evento que afecte a una sesión.
- Los elementos principales de esta librería son el `sip_event`, el `sip_message`.
- OSIP aún no tiene soporte para SIMPLE [78] (mensajería instantánea con SIP) pero es relativamente sencillo agregar esta funcionalidad.

- **Sofia-SIP [79]**

Es una librería de código abierto acorde con el RFC 3261. Está desarrollada especialmente para Linux. Tiene capacidades de presencia y mensajería, incluye la capa de transporte con TCP y UDP en ipv4 e ipv6. Además provee un módulo para mensajes SDP.

- **Yate [80]**

Yate (Yet Another Telephony Engine) es un motor para telefonía de nueva generación, que actualmente se centra en VoIP y PSTN. Es robusto y puede ser extendido fácilmente. Yate está licenciado bajo GPL y está diseñado especialmente para sistemas Linux.

##### 4.1.5.2. Librerías de C++ para el protocolo RTP

- **JrtpLib [81]**

Es una librería orientada a objetos, escrita en C++, basada en el RFC 3550. Soporta RTP bajo UDP, TCP en IPV4 e IPV6. Corre sobre entornos Linux y Windows.



- **GNU ccRTP [82]**

Es una implementación de RTP, de alto rendimiento, flexible y de fácil extensión, basada en el RFC 3555. Soporta unicast, multi-unicast, y multicast.

#### **4.1.6. Sistemas de gestión de Bases de datos**

A continuación se describen brevemente dos de los sistemas gestores de bases de datos, libres, más utilizados.

- **MySQL [83]**

Es un sistema gestor de base de datos extremadamente rápido. Aunque no ofrece las mismas capacidades y funcionalidades que otros, esto lo compensa con un rendimiento excelente que hace de esta la base de datos de elección en aquellas situaciones en las que se necesitan sólo unas capacidades básicas.

- **PostgreSQL [84]**

Sistema gestión de bases de datos relacional, de código abierto, liberado bajo la licencia de Distribución de Software Berkeley (Berkeley Software Distribution, BSD), y corre sobre la mayoría de sistemas operativos, incluyendo Linux, Unix, BeOS y Windows. Ofrece confiabilidad, estabilidad, extensibilidad, está diseñado para entornos con altos volúmenes de datos. Cuenta con herramientas gráficas para la administración y el diseño de Bases de Datos.

#### **4.1.7. API para la realización de pruebas de señalización SIP**

- **JAIN SIP [85]**

Las APIs de Java para Redes Integradas SIP (Java APIs for Integrated Networks SIP, JAIN SIP) es una especificación para transacciones de propósito general para el protocolo SIP y que se basa en interfaces Java. Es rico tanto en semántica como en la definición del protocolo SIP, y se desarrolló como una interfaz estándar para el protocolo que puede ser usado independientemente del nivel de programación y el entorno. JAIN SIP puede ser usado de múltiples formas:

- Como una especificación para plataformas J2SE que permiten el desarrollo de aplicaciones independientes de agentes de usuario, aplicaciones de registro y proxy.
- Como base para la implementación SIP para contenedores de SIP Servlets que permite el desarrollo de agentes de usuario, aplicaciones de registro y proxy dentro de un entorno basado en Servlets.
- Como base para la implementación SIP para contenedores de EJBs (Enterprise JavaBeans) que permiten el desarrollo de agentes de usuario, aplicaciones de registro y proxy en un entorno EJB.



## **4.2. HERRAMIENTAS PARA EL DESARROLLO DEL CLIENTE MÓVIL SIP**

### **4.2.1. Entornos de ejecución de aplicaciones para dispositivos móviles**

Existen distintos entornos de ejecución binaria, los que más se destacan son los siguientes:

- J2ME: Entorno de ejecución de aplicaciones Java para terminales con elevadas restricciones en sus capacidades. Este entorno de ejecución está orientado a aplicaciones desarrolladas en lenguaje Java y se basa en el concepto de la máquina virtual.
- Symbian: Sistema operativo para terminales móviles. Permite la ejecución de aplicaciones desarrolladas en C++ y Java.

Ambos entornos de ejecución proporcionan el API de soporte del Protocolo SIP para dispositivos móviles. Para J2ME se trata de la JSR-180 [86], y para Symbian este API aparece a partir de la segunda edición de la serie 60 basada en Symbian OS v7.0s [87].

### **4.2.2. Herramientas de programación para dispositivos móviles**

- **Eclipse [54]**

EclipseME es un plug-in para Eclipse que facilita la integración de los distintos componentes de desarrollo J2ME con este IDE, para de esta forma aprovechar la gestión de proyectos, herramientas de desarrollo y depuración dentro de emuladores J2ME.

- **Netbeans [55]**

El “Pack” de Movilidad de NetBeans es una herramienta para desarrollar aplicaciones que se ejecutan en teléfonos móviles. Esta herramienta puede ser utilizada para escribir, probar, y depurar aplicaciones para la plataforma Java ME.

- **Borland Jbuilder [56]**

Borland JBuilder ofrece desarrollo para el perfil J2ME para dispositivos móviles.

- **Carbide.j 1.5 [88]**

Carbide.j 1.5 es un paquete de software dirigido a desarrolladores de aplicaciones MIDP y de Perfil Personal. Durante la instalación del Carbide .j también se instala y se configura automáticamente un SDK de Nokia, este es el Nokia Prototype SDK 4.0 for Java™ ME, el cual contiene siete simuladores para MIDP. Este software se puede integrar con Borland JBuilder, NetBeans, Eclipse e IBM WebSphere Studio Device Developer 5.6 y 5.7.



- **Carbide.c++ Express 1.0 [89]**

Carbide.c++ Express es el primero en la familia de entornos de desarrollos basados en Eclipse para el desarrollo de aplicaciones para Symbian OS C++.

Esta herramienta hace que los desarrolladores ingresen fácilmente al mundo del desarrollo de aplicaciones móviles. Esta es una herramienta que no tiene costo y es fácil de instalar.

- **Sun Java Wireless Toolkit 2.5 for CLDC, Beta Release [90]**

El Conjunto de Herramientas Inalámbricas en lenguaje Java de Sun (WTK) (conocido antes como el Conjunto de Herramientas J2ME- Sun Java Wireless Toolkit) es un conjunto de herramientas para la creación de aplicaciones Java que tienen por fin su ejecución en dispositivos que obedezcan la Tecnología Java para la Industria Inalámbrica (Java Technology for the Wireless Industry, JTWI - JSR 185) y a la especificación de la Arquitectura de Servicio Móvil (Mobile Service Architecture, MSA - JSR 248). El WTK 2.5 consta de herramientas para la construcción, diversas utilidades y un simulador.

#### 4.2.3. APIs y Librerías para J2ME

- **MMAPI (Mobile Media API) [91]**

Es un conjunto de utilidades que permiten a los desarrolladores J2ME, incorporar funcionalidad multimedia a sus aplicaciones. Permite reproducción de archivos de audio: wav, amr, mp3, y captura de audio y video. Solo está disponible para MIDP 2.0.

- **java.net.RTP para J2ME [92]**

El soporte al protocolo RTP para J2ME se basa en una adaptación del API java.net.RTP para RTP/RTCP desarrollada en Java (J2SE). Tiene soporte para unicast y multicast. El código del API es modular y fácil de entender.

- **API de SIP para J2ME (JSR-180) [86]**

Permite que dispositivos móviles Java puedan enviar y recibir mensajes SIP ya que el MIDP 2.0 de J2ME incluye soporte para los sockets TCP/IP y los datagramas UDP/IP que utiliza este protocolo para su transporte. Se trata de un API compacto y genérico lo cual permite que se utilice con cualquier perfil J2ME, no solamente con el MIDP.



## **5. PROTOTIPO CLIENTE MÓVIL SIP PARA UN SISTEMA IVR BASADO EN LA ARQUITECTURA DE SERVICIOS IMS**

### **5.1. ANÁLISIS DE REQUERIMIENTOS Y FORMULACIÓN DE LA ARQUITECTURA GENERAL DEL SISTEMA**

De acuerdo a la investigación teórica realizada en los capítulos 1 y 2 a continuación se describen los componentes esenciales con los cuales debe contar un sistema para la provisión de servicios IVR enmarcados dentro de la arquitectura de servicios IMS y cuyo cliente final es un móvil SIP. De esta forma se podrá llegar a la formulación de la Arquitectura General del Sistema.

El objetivo de este capítulo es realizar el análisis, diseño e implementación de la infraestructura adecuada para soportar servicios de IVR enmarcados dentro de la arquitectura IMS, que le permitan al usuario final acceder a estos mediante una aplicación específica. En este sentido, se debe entender que la infraestructura a implementar servirá de base para el soporte de diversas aplicaciones clientes que requieran de las capacidades de un sistema IVR. La aplicación cliente a desarrollar corresponde al acceso de un usuario móvil al correo electrónico de la Universidad del Cauca, por medio de un sistema IVR servido desde una red IMS.

#### **5.1.1. Cliente Móvil SIP**

Es un cliente para dispositivos móviles que debe tener las siguientes características:

- Soporte de un API de envío y recepción de mensajes SIP
- Soporte de un API para la grabación y reproducción de audio
- Soporte de un API para el envío y recepción de flujos RTP

#### **5.1.2. Servidor CSCF**

En la arquitectura IMS la CSCF puede ser de 3 tipos:

- P-CSCF
- S-CSCF
- I-CSCF

Con respecto a la relación entre la arquitectura del protocolo SIP y la arquitectura de servicios IMS se encuentran las siguientes analogías:

- El servidor CSCF de la arquitectura IMS es análogo al conjunto de servidores: Proxy, Register y Redirect de la arquitectura del protocolo SIP.
- El HSS de la arquitectura IMS es análogo al Location Server de la arquitectura SIP.



Por lo anterior, fue necesario implementar un servidor CSCF que conste de los siguientes módulos esenciales:

- P-CSCF (Servidor Proxy SIP): Soporte al encaminamiento de los mensajes SIP
- S-CSCF (Servidor Register): Soporte a la gestión de las sesiones SIP, registro de usuarios, punto de comunicación entre el Servidor de Aplicaciones SIP y la MRF y entre el Servidor de Aplicaciones SIP y el HSS.

El I-CSCF es un nodo opcional en la arquitectura de IMS ya que el sistema se puede configurar para que P-CSCF entre en contacto directo con el S-CSCF, y por esta razón no tuvo en cuenta como un módulo esencial para la implementación del sistema.

### **5.1.3. Servidor de Aplicaciones SIP**

El Servidor de Aplicaciones SIP es análogo al Servidor de Aplicaciones en la arquitectura del protocolo SIP. Este servidor debe tener las siguientes características:

- Soporte a la lógica del servicio correspondiente para permitirle al usuario móvil acceder a su correo electrónico a través de su interacción con el sistema IVR. Para el soporte de la aplicación de acceso al correo electrónico se utilizó una API que permite transacciones con el servidor de correo.

### **5.1.4. MRF**

La Función de Recursos Multimedia de la arquitectura IMS es análoga a los Servidores de Medios de la arquitectura del protocolo SIP, es por esto que solo se denominará este componente como la MRF.

La MRF debe tener las siguientes características:

- Soporte al protocolo SIP
- Soporte para el envío y recepción de flujos RTP
- Soporte al control de recursos
- Soporte al servicio ASR
- Soporte al servicio TTS

### **5.1.5. HSS**

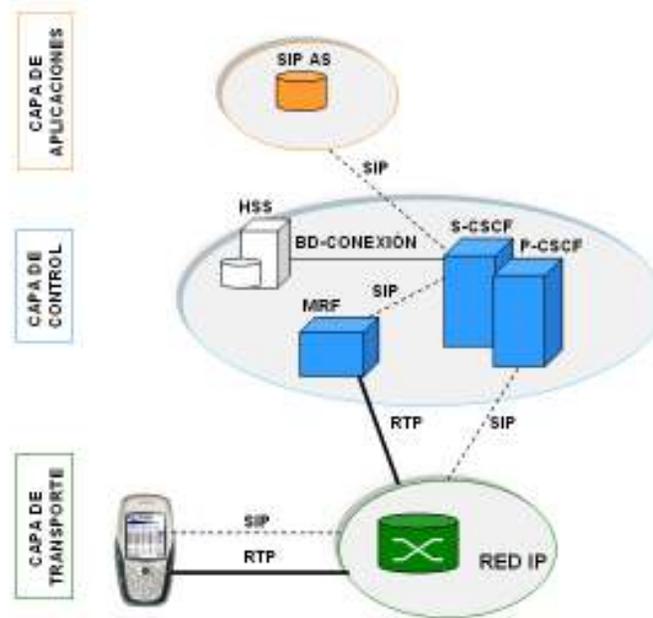
Como se dijo anteriormente, el HSS de la arquitectura IMS es análogo al Servidor Location de la arquitectura del protocolo SIP, es por esto que solo se denominará este componente como el HSS. El HSS debe tener las siguientes características:

- Soporte al almacenamiento de la información personal de los usuarios.
- Soporte a la comunicación con el servidor de aplicaciones.



### 5.1.6. Arquitectura General del Sistema

De acuerdo al análisis de requerimientos que se realizó, en la Figura 10 se formula la arquitectura general del sistema:



**Figura 10: Arquitectura General del Sistema**

La capa de transporte consta de un cliente móvil con capacidad de manejo de flujos de señalización y de datos de audio. A su vez, en esta capa se encuentra la red IP encargada del transporte de dichos flujos hacia módulos específicos de la capa de control. Los flujos de señalización y datos provenientes del móvil tienen como destino los servidores S-CSCF y P-CSCF, y la MRF respectivamente.

La capa de control está constituida por: la MRF, encargada de soportar el sistema IVR y brindar hacia el resto de la infraestructura una interfaz SIP para la comunicación con el Servidor de Aplicaciones a través del S-CSCF, y otra RTP para la recepción y envío de los flujos de audio desde y hacia el móvil; los servidores S-CSCF y P-CSCF encargados de gestionar la sesión y redireccionar mensajes de señalización, respectivamente; y el HSS para el manejo dinámico de la información personal y de sesión del usuario, posee una interfaz (JDBC) mediante la cual el S-CSCF realiza las transacciones de base de datos que se requieren.

En la capa de aplicaciones se tiene el Servidor de Aplicaciones SIP, el cual se encarga de soportar la lógica de servicio del usuario, y su comunicación con la MRF es a través del S-CSCF bajo el protocolo SIP.



## **5.2. DISEÑO E IMPLEMENTACIÓN DEL SISTEMA**

Para la construcción de los componentes del sistema se utilizó el Modelo de Construcción de Soluciones (M.C.S) propuesto por el Ingeniero Carlos Serrano profesor de la FIET [93]. Este modelo se basa en tres macrocomponentes esenciales para la construcción de una solución que puede estar enmarcada en cualquier área: Modelo del proceso del desarrollo, Estructura para la descripción del sistema y Modelo para la organización del talento. En esta sección solo se utilizarán los dos primeros componentes. Dentro del Modelo del proceso del desarrollo se identifican cuatro etapas aplicables a cualquier proyecto: estudio de prefactibilidad, formulación del proyecto, ejecución del proyecto, y validación de la solución. En este capítulo solo se detalla la etapa de ejecución del proyecto, ya que de esta forma se planteó en el anteproyecto. La aplicación de la Estructura para la descripción del sistema se realizó con base a los modelos de Descripción del sistema y de Implementación del sistema.

Una vez realizado el análisis de los componentes esenciales del sistema, se puede entrar a diseñar e implementar cada componente de acuerdo al soporte teórico con el que se cuenta, y herramientas disponibles para su desarrollo. Las bases teóricas obtenidas en los capítulos previos, y las fases de análisis y diseño, constituyen la información que describe el sistema. Entre tanto que la selección de herramientas y su posterior utilización para el desarrollo de los componentes del sistema se basan en el modelo de Implementación.

La selección de las herramientas que permitieron el desarrollo de cada componente tuvo en cuenta principalmente: la disponibilidad de acceso a la herramienta, que contara con una buena documentación de instalación y uso, que facilitara ejemplos de aplicaciones previamente realizadas, que permitiera un desarrollo rápido debido a experiencias previas en su utilización. En lo que respecta a esta última consideración, se decidió utilizar como lenguaje general de programación el lenguaje Java.

En las siguientes secciones se realiza el diseño y se especifican las herramientas que se seleccionaron para el desarrollo de cada componente del sistema, de acuerdo a los requerimientos que se plantearon en la Fase de Análisis y a la Formulación de la Arquitectura General del Sistema. Esto constituirá la etapa de ejecución del proyecto.

### **5.2.1. Diagrama de Casos de Uso del Sistema**

En la Figura 11 se ilustra el Diagrama de Casos de Uso del Sistema.

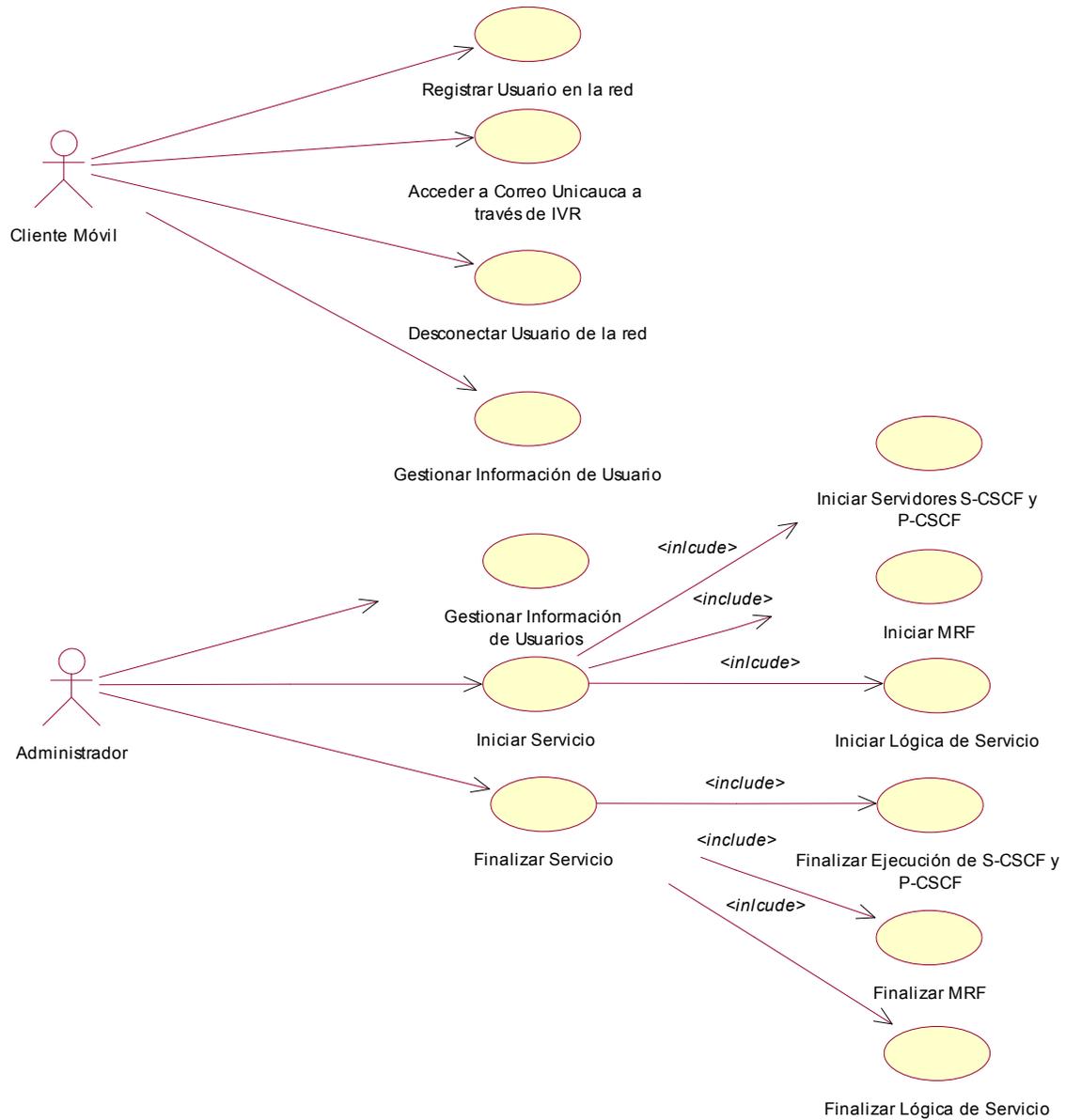


Figura 11: Diagrama de Casos de Uso del Sistema



### 5.2.1.1. Casos de Uso del Sistema iniciados por el actor Cliente Móvil

<b>Registrar Usuario</b>	
<b>Nombre</b>	
<b>Iniciador</b>	Cliente Móvil
<b>Propósito</b>	Registrar usuario dentro de la red IMS
<b>Resumen</b>	El usuario envía un mensaje SIP de registro, con dos parámetros: "login" y "contraseña", al módulo CSCF de la red IMS, y espera una respuesta del mismo, la cual puede ser de confirmación o de rechazo al registro.

**Tabla 1: Caso de Uso Registrar Usuario**

<b>Acceder a Correo Unicauca a través de IVR</b>	
<b>Nombre</b>	
<b>Iniciador</b>	Cliente Móvil
<b>Propósito</b>	Iniciar, llevar a cabo, y terminar una sesión IVR por medio de una aplicación que le permita al usuario acceder a su correo electrónico.
<b>Resumen</b>	La primera operación que debe ejecutar el usuario es obtener los servicios que este tiene registrados dentro de la red IMS. Para esto, el usuario inicia una sesión de voz para consultar los servicios que tiene registrados dentro de la red IMS. Una vez se haya iniciado la sesión, se empieza la transferencia de los datos de voz la cual consta básicamente de: 1. Envío de saludo del sistema al usuario, 2. Envío del sistema al usuario de la información de los servicios que el usuario tiene registrados, 3. Envío del usuario al sistema del comando de voz de la opción seleccionada, luego el sistema se encarga de procesar dicho comando y generar las respuestas adecuadas para: terminar con la sesión, o continuar con la ejecución de la aplicación que el usuario seleccionó. Finalmente el usuario puede terminar la sesión, ya sea mediante un comando de voz, o la selección de la opción salir de un menú del cual se disponga en el dispositivo móvil.

**Tabla 2: Caso de Uso Acceder a Aplicación a través de IVR**

<b>Desconectar Usuario de la red</b>	
<b>Nombre</b>	
<b>Iniciador</b>	Cliente Móvil
<b>Propósito</b>	Desconectar al usuario de la red IMS
<b>Resumen</b>	El usuario envía un mensaje SIP de registro, con dos parámetros: login y contraseña, al módulo CSCF de la red IMS, y espera una respuesta de este módulo. La respuesta puede ser de confirmación o rechazo a la desconexión en la red.

**Tabla 3: Caso de Uso Desconectar Usuario de la red**



Nombre	Gestionar Información de Usuario
Iniciador	Cliente Móvil
Propósito	Permitirle al usuario gestionar su información personal.
Resumen	El usuario accede a su información a través de un navegador Web, y puede realizar la tarea de modificación de la misma.

Tabla 4: Caso de Uso Gestionar Información de Usuario

#### 5.2.1.2. Casos de Uso del Sistema iniciados por el actor Administrador

Nombre	Gestionar Información de Usuarios
Iniciador	Administrador
Propósito	Permitirle al administrador gestionar la información personal y de contactos de todos los usuarios de la red IMS.
Resumen	El administrador accede a la información de los usuarios a través de un navegador Web, y puede realizar la tarea de modificación de la información personal. La modificación incluye, adición, eliminación y cambios en la información.

Tabla 5: Caso de Uso Gestionar Información de Usuarios

Nombre	Iniciar Servicio
Iniciador	Administrador
Propósito	Iniciar la prestación de servicios IVR dentro de la arquitectura IMS.
Resumen	Al solicitar la ejecución de Iniciar servicio, el sistema reservará los recursos requeridos para la prestación de servicios IVR dentro de la arquitectura IMS. Este caso de uso requiere de otros que hacen referencia a la puesta en marcha de los servidores S-CSCF y P-CSCF, la MRF y la lógica del servicio.

Tabla 6: Caso de Uso Iniciar Servicio

Nombre	Finalizar Servicio
Iniciador	Administrador
Propósito	Finalizar la prestación de servicios IVR dentro de la arquitectura IMS.
Resumen	Al solicitar la ejecución de finalizar servicio, el sistema libera los recursos requeridos para la prestación de servicios IVR dentro de la arquitectura IMS. Este caso de uso requiere de otros que hacen referencia a la finalización de la ejecución de los servidores S-CSCF y P-CSCF, la MRF y la lógica del servicio.

Tabla 7: Caso de Uso Finalizar Servicio



## 5.2.2. Diseño de componentes

### 5.2.2.1. Cliente Móvil SIP

Para dar soporte a los requerimientos planteados, el cliente móvil debe estar constituido de la forma como se muestra en la Figura 12:

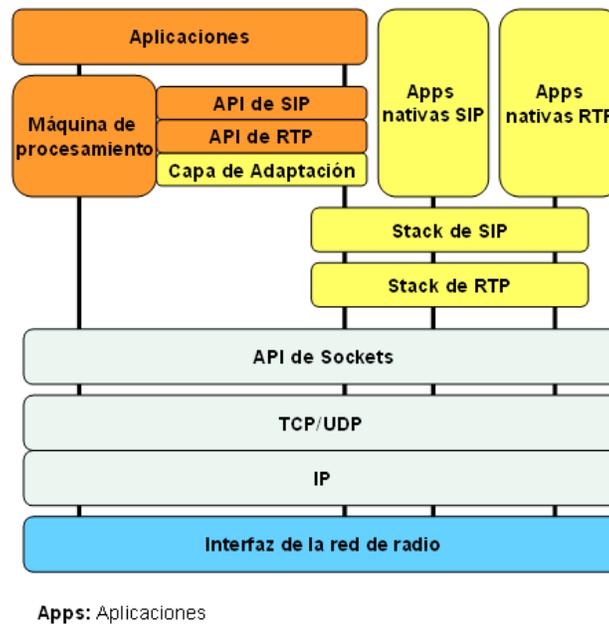


Figura 12: Diseño del componente Cliente Móvil SIP

Las herramientas que se seleccionaron para el desarrollo del cliente móvil SIP son las siguientes:

- **Lenguaje de programación**  
J2ME: Java para dispositivos móviles
- **API de SIP**  
JSR-180: API de SIP para J2ME
- **API de RTP**  
Adaptación de la solución `java.net.RTP` al lenguaje J2ME
- **Máquina de procesamiento**  
Máquina Virtual de Java (Java Virtual Machine, JVM)
- **Soporte a TCP/UDP e IP**  
El MIDP 2.0 incluye soporte para los sockets TCP/IP y los datagramas UDP/IP



▪ **Interfaz de la red de radio**

El alcance del cliente móvil SIP es la etapa de simulación por lo tanto no se tiene en cuenta la implementación de la interfaz radio.

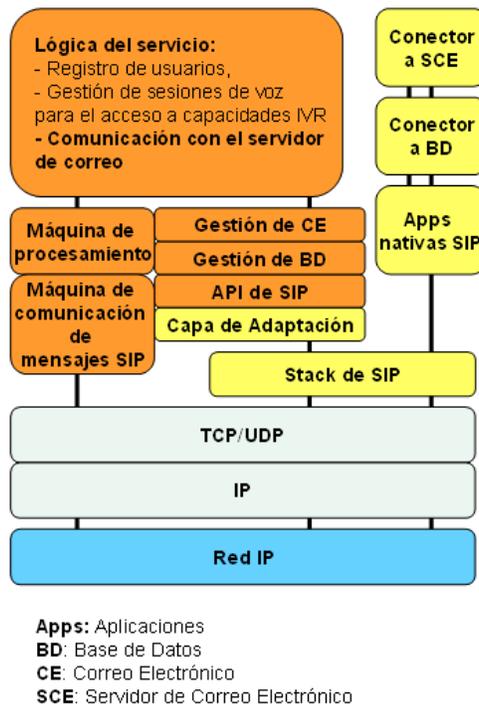
La codificación de audio a utilizar es PCM (G.711), debido a que la codificación AMR no tiene una implementación de referencia.

**5.2.2.2. Servidor CSCF y Servidor de Aplicaciones SIP**

Los módulos necesarios para crear la CSCF son: el P-CSCF y el S-CSCF.

Dado que el S-CSCF es el módulo encargado de comunicar el Servidor de Aplicaciones SIP tanto con la MRF como con el HSS, se decidió diseñar el Servidor de Aplicaciones SIP en conjunto con el S-CSCF, y de esta forma el servidor de aplicaciones deberá también soportar las características requeridas por el módulo S-CSCF. A su vez, el módulo P-CSCF también se puede diseñar en conjunto con los dos anteriores, por lo tanto, el Servidor de Aplicaciones SIP también soporta las características requeridas por el módulo P-CSCF.

De acuerdo a lo anterior, el Servidor de aplicaciones SIP debe estar constituido funcionalmente de la forma como se muestra en la Figura 13:



**Figura 13: Diseño de los componentes: Servidor CSCF y Servidor de Aplicaciones SIP**

Las herramientas que se seleccionaron para el desarrollo del Servidor CSCF y el Servidor de Aplicaciones SIP son las siguientes:

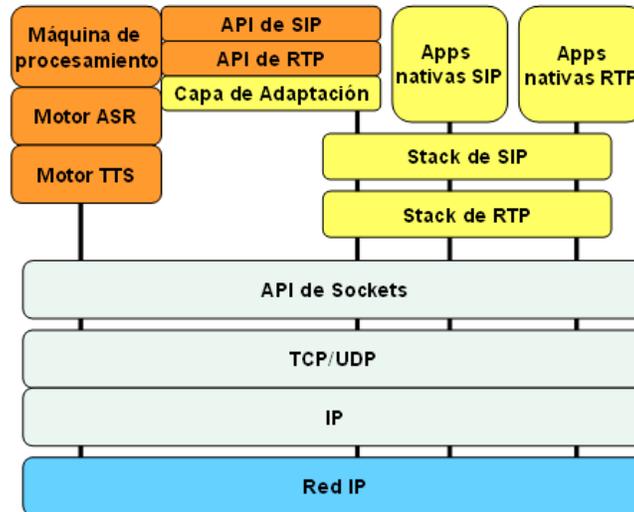


- **Lenguaje de programación**  
Java para entornos de escritorio
- **API de SIP**  
Servlets SIP: API de SIP para el desarrollo de servidores de aplicaciones SIP, posee las clases y permite las extensiones necesarias para soportar funcionalidades de: Proxy, gestión de sesiones SIP.
- **Máquina de procesamiento**  
Máquina Virtual de Java (Java Virtual Machine, JVM)
- **Lógica del servicio**  
El registro de usuarios y la gestión de las sesiones se realizan mediante los Servlets SIP. La gestión de bases de datos se realiza por medio del paquete de Java `java.sql.*`. La comunicación con el servidor de correo se realiza mediante el paquete de Java `javax.mail.*`, esta API permite además de gestionar las transacciones, la realización de la conexión con el servidor de correo electrónico.
- **Contenedor de Servlets SIP**  
En este caso, es necesario especificar que para el desarrollo de aplicaciones con Servlets SIP se debe poseer un contenedor para estos el cual se encargue de las diversas transacciones SIP con otros agentes de usuario. De acuerdo a las herramientas que se investigaron para realizar esta tarea, se eligió un Servidor de Aplicaciones de fácil acceso como el SIPMethod (`SIPMethod_AS_1_5_1_win.exe`), el cual se encarga de albergar los Servlets SIP, y de soportar la comunicación de mensajes SIP. Además, este servidor también es capaz de soportar los Servlets HTTP de Java para realizar la gestión de los servicios.
- **Conector a la Base de Datos**  
Librería `mysql-connector-java-2.0.14-bin.jar`

### 5.2.2.3. MRF

La Función de Recursos Multimedia de la arquitectura IMS es análoga a los Servidores de Medios de la arquitectura del protocolo SIP.

Para dar soporte a los requerimientos planteados, el módulo de la Función de Recursos Multimedia esta constituido de la forma como se muestra en la Figura 14.



Apps: Aplicaciones

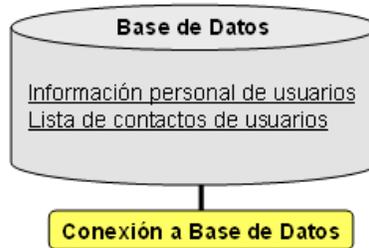
Figura 14: Diseño del componente MRF

Las herramientas que se seleccionaron para el desarrollo de la MRF son las siguientes:

- **Lenguaje de programación**  
C++
- **API de SIP**  
OSIP 3.1: Librería de SIP para C++. Soporte a Sockets TCP/UDP sobre IP.
- **API de RTP**  
JRTPLIB (jrtpplib-3.6.0): Librería de RTP para C++.
- **Máquina de procesamiento**  
Compilador de C++
- **Motor ASR**  
Motor ASR de Verbio
- **Motor TTS**  
Motor TTS de Verbio
- **Adaptación del lenguaje C++ con Verbio**  
Library SDK de Verbio

#### 5.2.2.4. HSS

Para dar soporte a los requerimientos planteados para el módulo HSS, este debe estar constituido como se ilustra en la Figura 15.



**Figura 15: Diseño del componente HSS**

Las herramientas que se seleccionaron para el desarrollo del HSS son las siguientes:

- **Base de Datos**  
MySQL
- **API de SIP**  
JDBC: Conexión con la base de datos.

### 5.2.3. Diagramas de Secuencia para los Casos de Uso Esenciales del Sistema originados por el actor Cliente Móvil



5.2.3.1. Caso de Uso Registrar Usuario

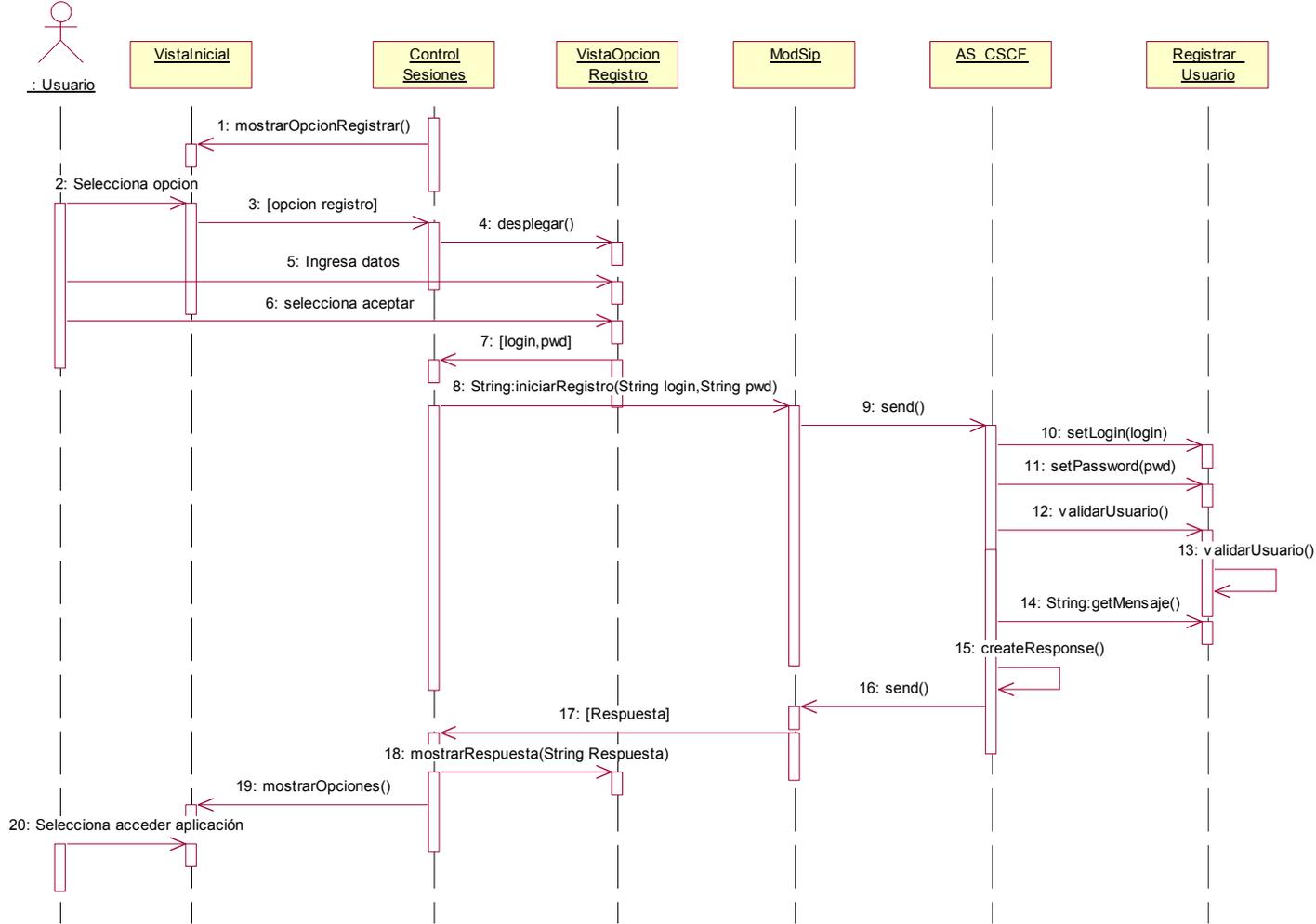
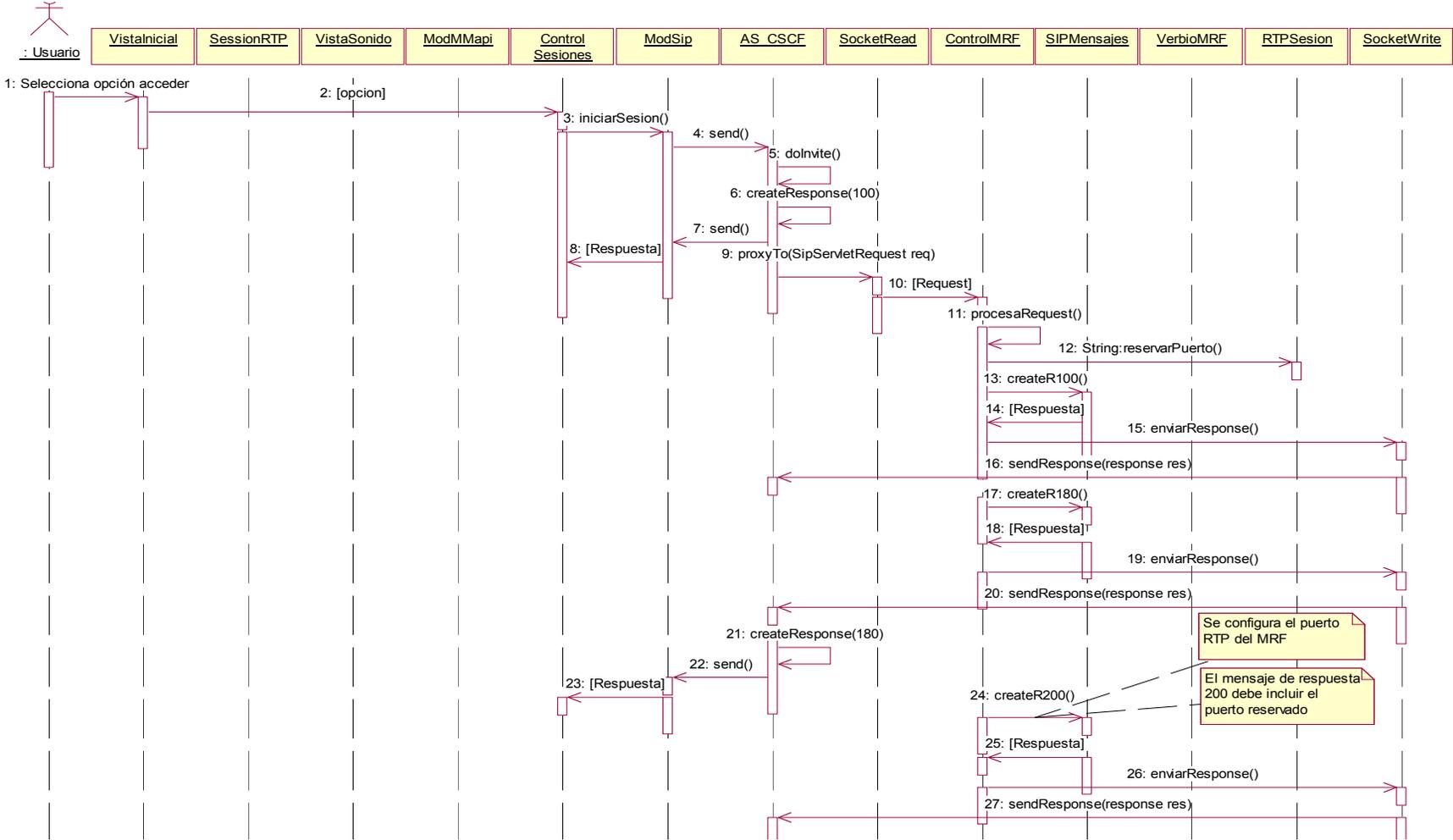
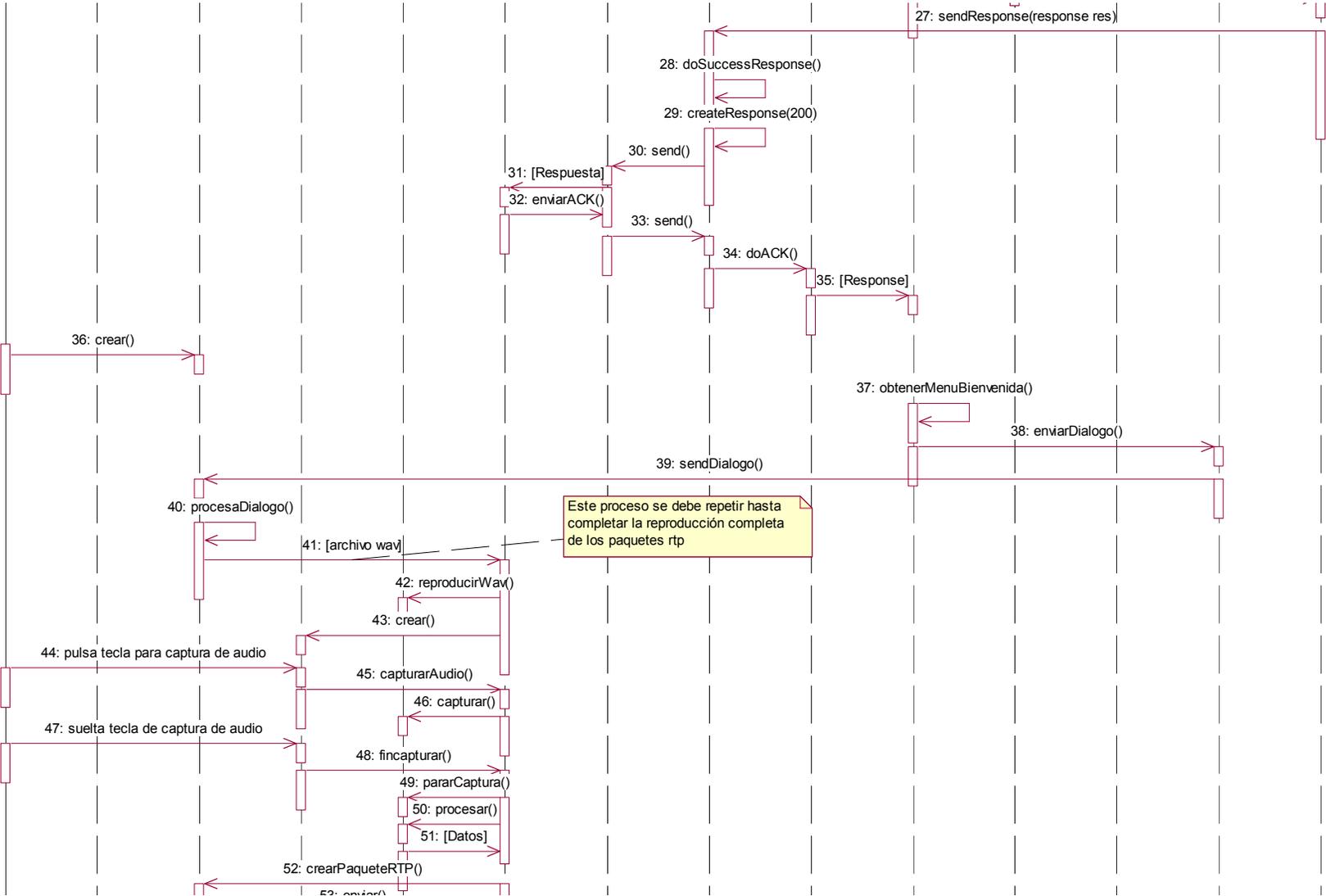


Figura 16: Caso de Uso Registrar Usuario



5.2.3.2. Caso de Uso Acceder a Correo Unicauca a través de IVR





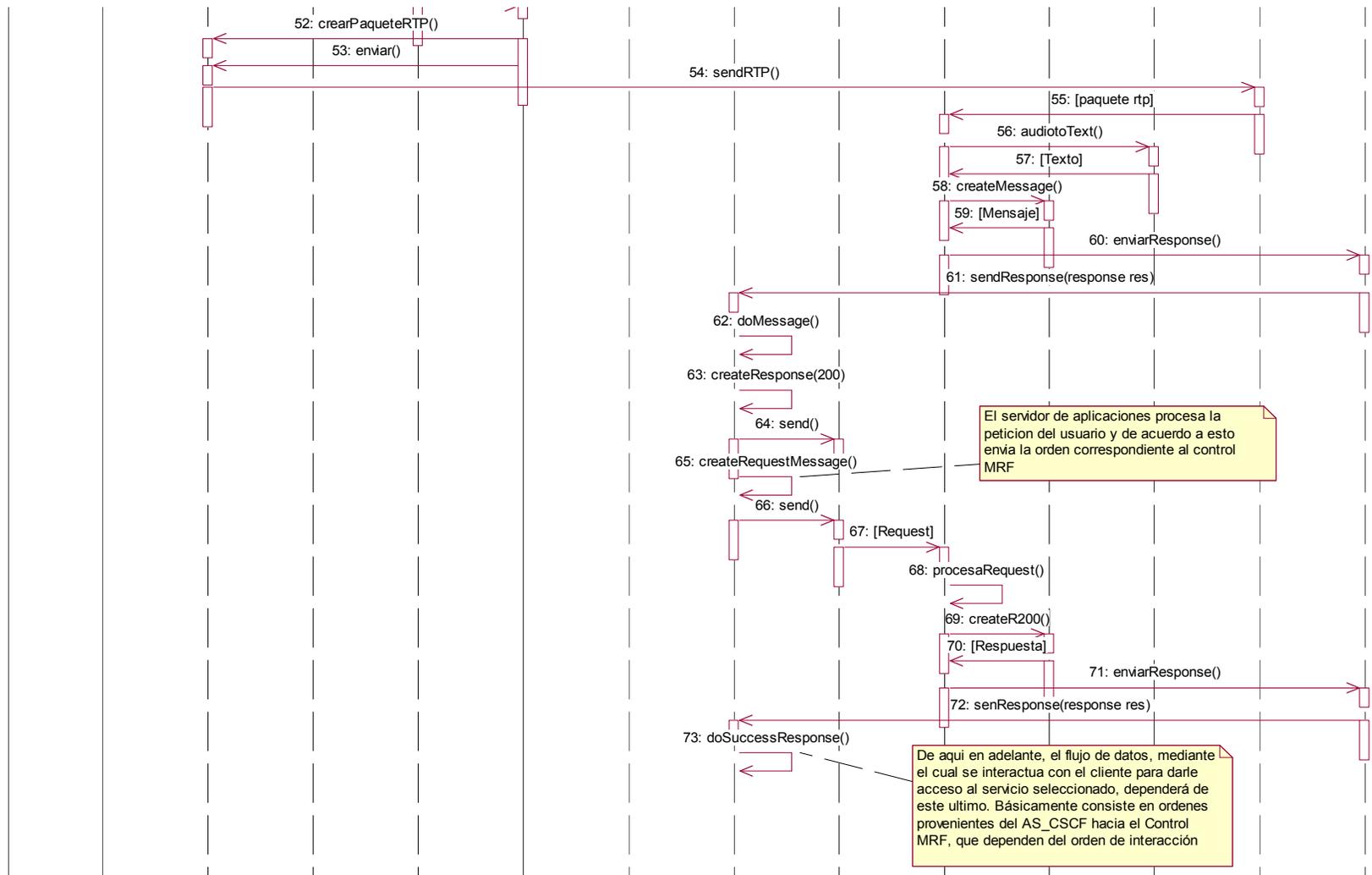


Figura 17: Caso de Uso Acceder a Correo Unicauca a través de IVR



5.2.3.3. Caso de Uso Desconectar Usuario de la red

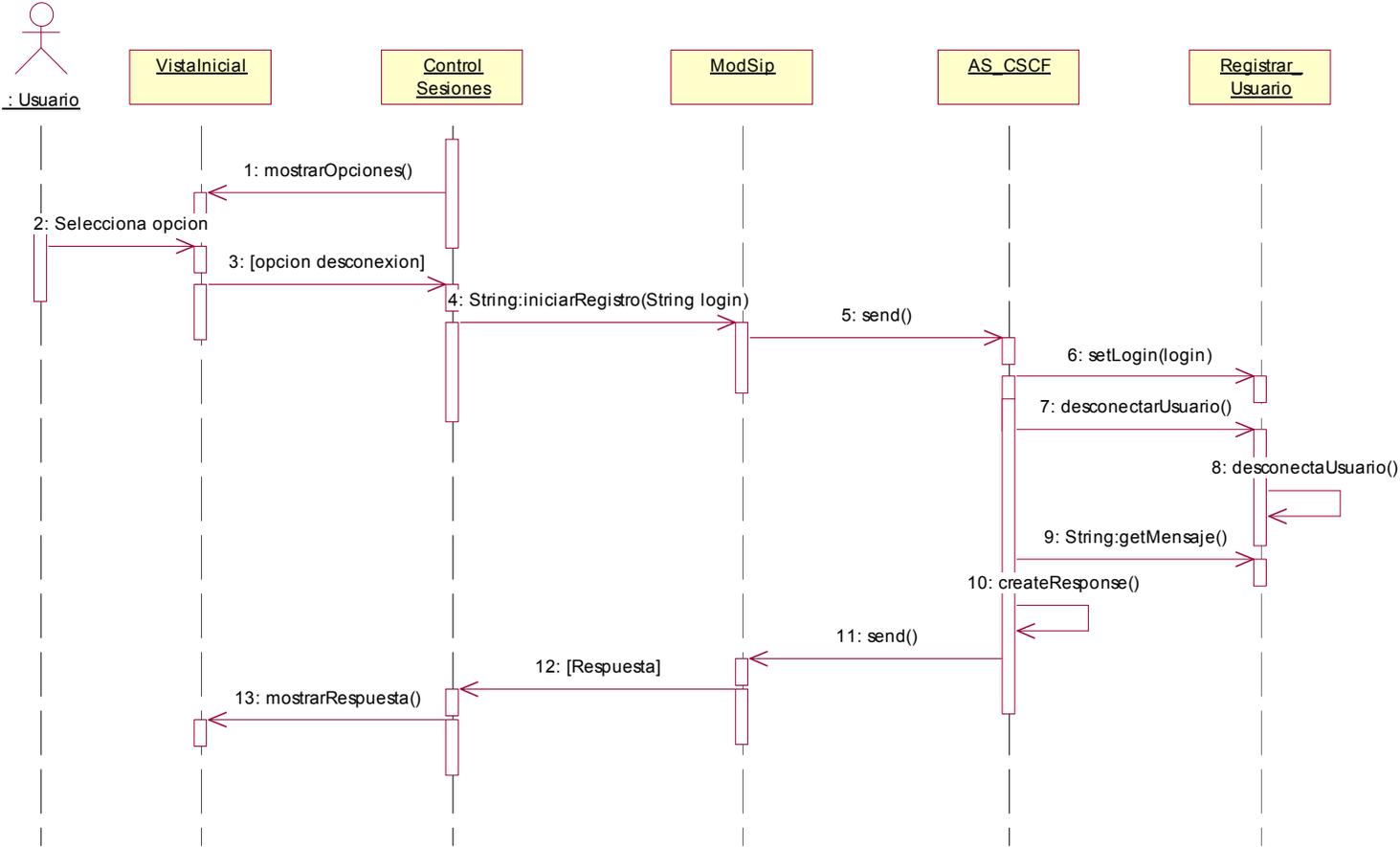


Figura 18: Caso de Uso Desconectar Usuario de la red



5.2.3.4. Caso de Uso Gestionar Información de Usuario

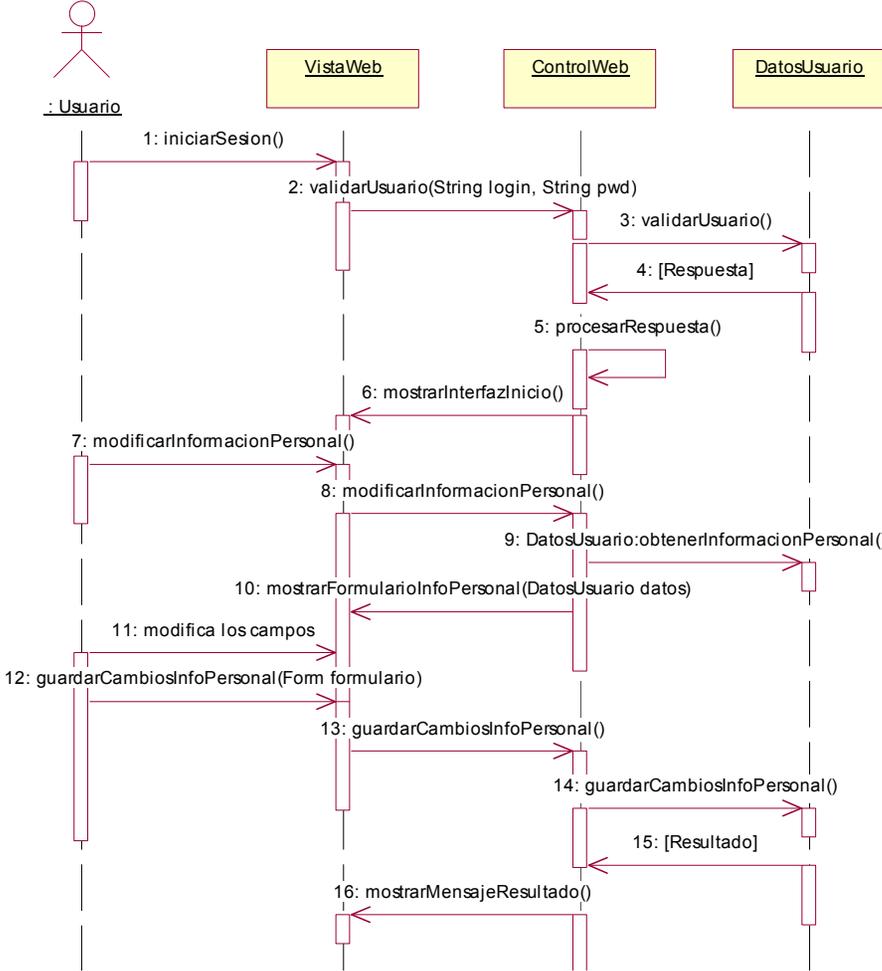
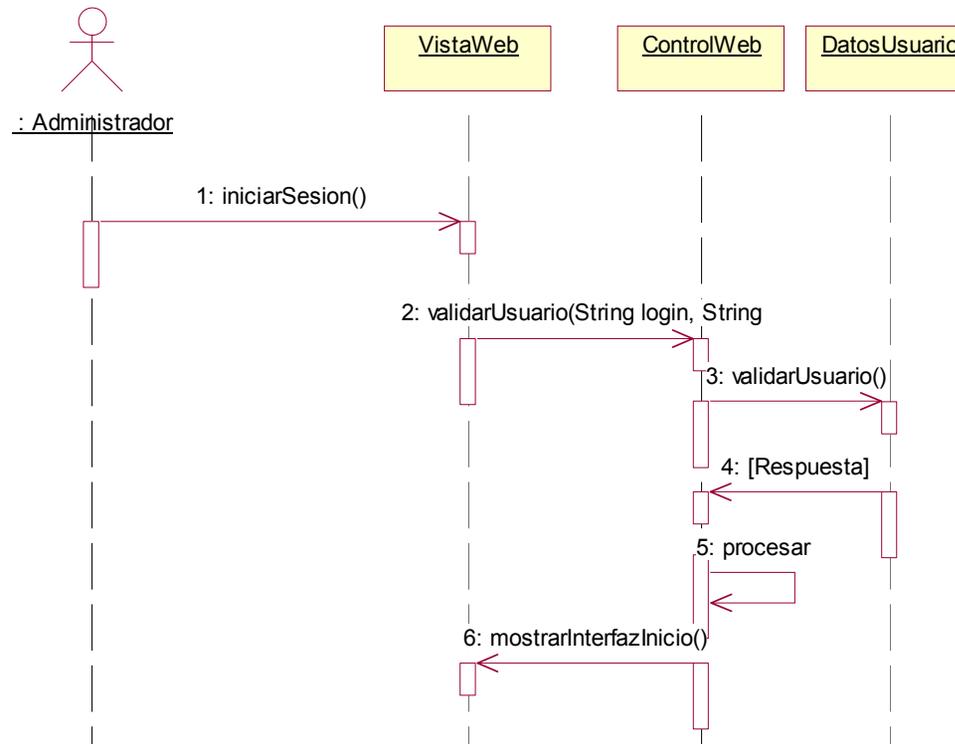


Figura 19: Caso de Uso Gestionar Información de Usuario



### 5.2.4. Diagrama de Secuencia para el Caso de Uso Esencial del Sistema originado por el actor Administrador

- Caso de Uso Gestionar Información de Usuarios



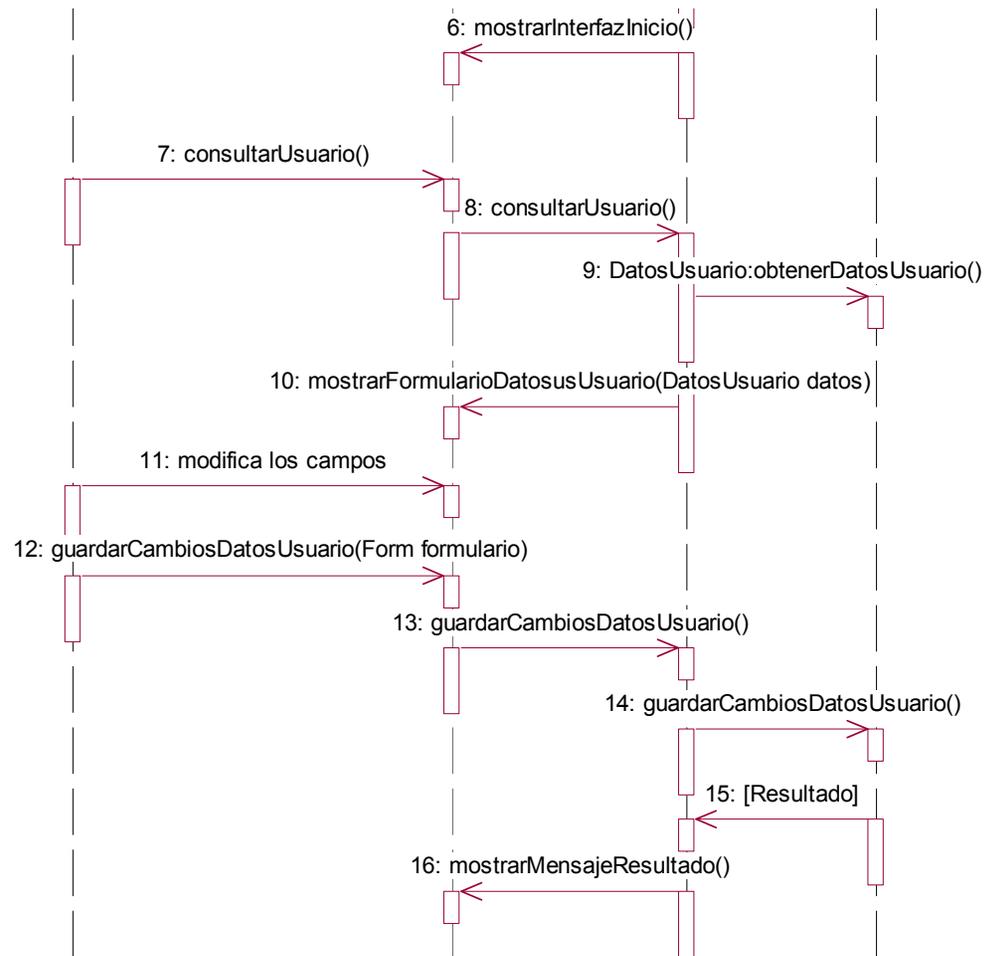


Figura 20: Caso de Uso Gestionar Información de Usuarios



### 5.3. PRUEBAS

#### 5.3.1. Pruebas de Validación del Sistema

Las pruebas de validación del sistema se realizaron mediante evaluaciones individuales de la funcionalidad de cada módulo para luego realizar una integración de componentes casi transparente debido a la utilización de protocolos de comunicación estándar.

Las pruebas relacionadas con el protocolo de señalización SIP se hicieron con la creación de agentes de usuarios basados en JAIN SIP, los cuales podían interactuar con el Cliente Móvil SIP, con el Servidor de Aplicaciones que contenía los Servlets SIP, y con la parte de la MRF que se encargaría de manejar este protocolo. Con dichos agentes de usuario se pudo comprobar la correcta generación y secuenciación de los mensajes SIP, lo que además permitió una integración rápida de la transmisión de señalización entre el Cliente Móvil y el Sistema. En la Figura 21 se puede ver la interfaz de usuario del Cliente JAIN SIP que se desarrolló para dichas pruebas.

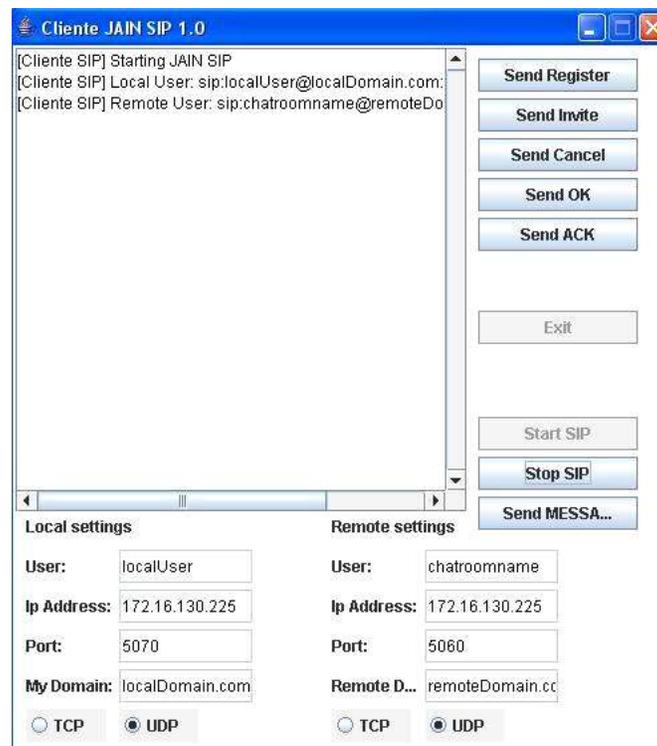


Figura 21: Cliente JAIN SIP de pruebas

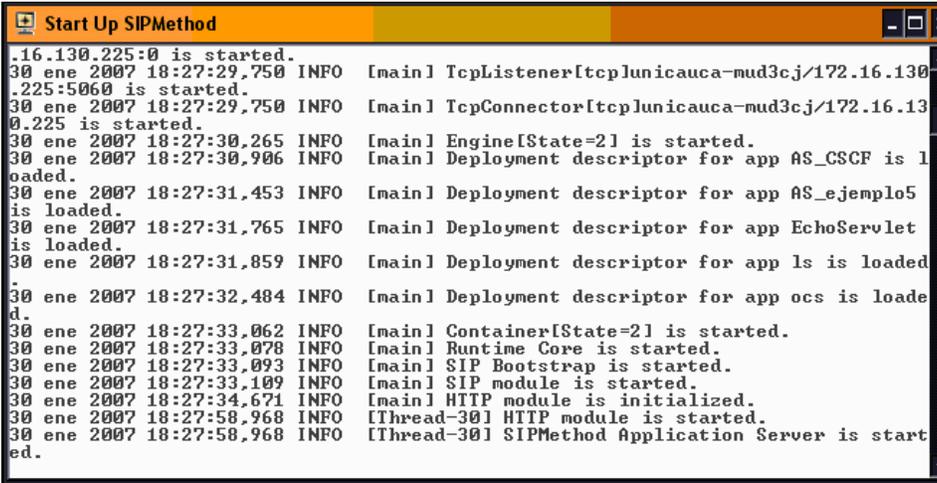
Las pruebas relacionadas con el protocolo de transporte RTP se realizaron mediante la creación de un agente móvil receptor y transmisor de flujos RTP, y en el lado de la MRF las pruebas se realizaron con la ayuda de un cliente de escritorio, desarrollado en C++, receptor y transmisor de flujos RTP. Para la integración de la transmisión de los flujos RTP se hicieron algunos ajustes del lado del Cliente Móvil, en lo que respecta a la creación del paquete RTP. Los resultados obtenidos, luego de estas pruebas, fueron satisfactorios debido a que se facilitó la integración de la



transmisión de los flujos RTP entre el Cliente Móvil y el Sistema IVR, y solo se necesitó de algunos ajustes para la obtención de la funcionalidad RTP.

En cuanto a la validación de los motores ASR y TTS se realizaron aplicaciones en C++ que a través de la Library SDK de Verbio ejecutaran los procesos necesarios para su operación. Para las pruebas del ASR y del TTS se contó respectivamente con archivos de audio grabados, y archivos de texto creados previamente. La funcionalidad de dichas aplicaciones se utilizó posteriormente para realizar la integración con los demás componentes. Los resultados de estos procesos, permitieron comprobar la alta naturalidad, inteligibilidad, y capacidad de reconocimiento de los motores TTS y ASR de Verbio, respectivamente. Además, se logró la obtención de módulos funcionalmente independientes para los procesos de TTS y ASR, y se pudo entender la forma como se genera la gramática para el proceso de ASR.

Para la validación de sistema total, se usaron consolas de información que permitieron comprobar la validez de los flujos de señalización y de datos. El Servidor de Aplicaciones SIPMethod contaba con su propia consola, ver Figura 22, la cual fue utilizada para imprimir los mensajes de señalización SIP que se intercambiaban entre el Cliente Móvil, el S-CSCF, el P-CSCF, la Lógica del Servicio y la MRF. Para la MRF, desarrollada en C++, se utilizó la consola de ejecución correspondiente, ver Figura 23, y en este caso se imprimieron tanto los mensajes de control, como los de la señalización SIP, y los del flujo RTP. Con respecto a la Lógica de Servicio, se desarrolló una consola, ver Figura 24, que permitió observar tanto los mensajes SIP enviados como los recibidos, y se utilizó la consola de ejecución para imprimir los mensajes de control. En el lado del usuario, se imprimieron los mensajes de control, de señalización y de flujo RTP en la consola de ejecución correspondiente.



```
Start Up SIPMethod
.16.130.225:0 is started.
30 ene 2007 18:27:29,750 INFO [main] TcpListener[tcplunicauca-mud3cj/172.16.130
.225:5060 is started.
30 ene 2007 18:27:29,750 INFO [main] TcpConnector[tcplunicauca-mud3cj/172.16.13
0.225 is started.
30 ene 2007 18:27:30,265 INFO [main] Engine[State=2] is started.
30 ene 2007 18:27:30,906 INFO [main] Deployment descriptor for app AS_CSCF is l
oaded.
30 ene 2007 18:27:31,453 INFO [main] Deployment descriptor for app AS_ejemplo5
is loaded.
30 ene 2007 18:27:31,765 INFO [main] Deployment descriptor for app EchoServlet
is loaded.
30 ene 2007 18:27:31,859 INFO [main] Deployment descriptor for app ls is loaded
.
30 ene 2007 18:27:32,484 INFO [main] Deployment descriptor for app ocs is loade
d.
30 ene 2007 18:27:33,062 INFO [main] Container[State=2] is started.
30 ene 2007 18:27:33,078 INFO [main] Runtime Core is started.
30 ene 2007 18:27:33,093 INFO [main] SIP Bootstrap is started.
30 ene 2007 18:27:33,109 INFO [main] SIP module is started.
30 ene 2007 18:27:34,671 INFO [main] HTTP module is initialized.
30 ene 2007 18:27:58,968 INFO [Thread-30] HTTP module is started.
30 ene 2007 18:27:58,968 INFO [Thread-30] SIPMethod Application Server is start
ed.
```

Figura 22: Consola SIPMethod

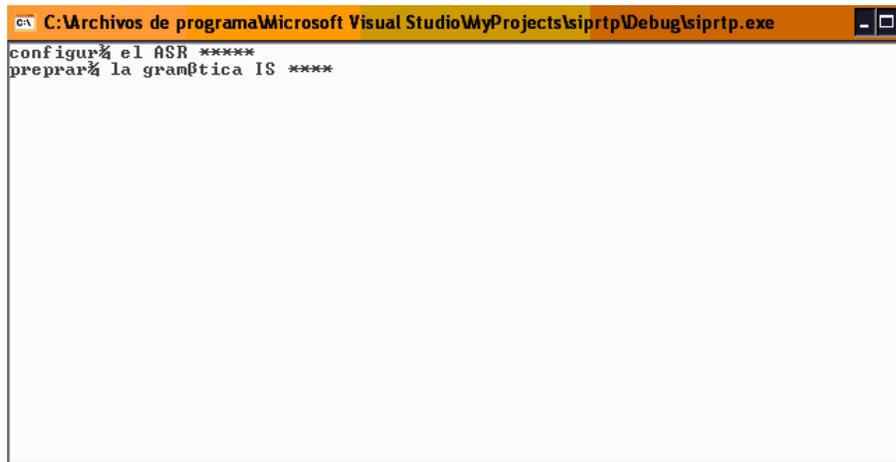


Figura 23: Consola MRF



Figura 24: Consola Lógica de Servicio

Para mayor información acerca de las herramientas, y las tecnologías utilizadas para la creación de las aplicaciones de validación del sistema referirse al Anexo C.



### 5.3.2. Pruebas de Desempeño

A continuación se registra el desempeño de cada uno de los módulos de la aplicación, en términos del consumo de memoria RAM y del uso de la CPU.

Para estas pruebas (que se realizaron en tres pasos: análisis de AS, Cliente Móvil y MRF, por separado) se contó dos computadores con características similares: procesador de 1.83Ghz y 512Mb de RAM.

- **Pruebas en el PC No.1 (AS, Lógica del Servicio y Cliente Móvil)**

En el PC No.1 se ejecuta el AS, la Lógica del Servicio y el Cliente Móvil. Inicialmente, sin programas en ejecución, se tiene un 0% de uso de CPU y 288MB de memoria virtual utilizada, ver Figura 25. Es necesario registrar las condiciones iniciales del PC para tomarlo como referencia durante la prueba de los módulos de la aplicación.

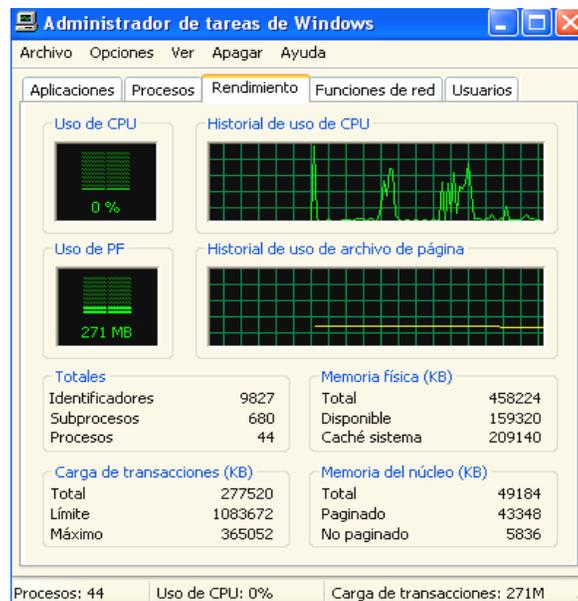


Figura 25: Rendimiento del PC No.1 sin programas en ejecución

- **AS y Lógica del Servicio**

Al activar el motor de base de datos (MySQL), esencial para el servidor de registro, se tiene un incremento en el uso de memoria virtual de 19Mb, sin incremento notable en el uso de la CPU, ver Figura 26.

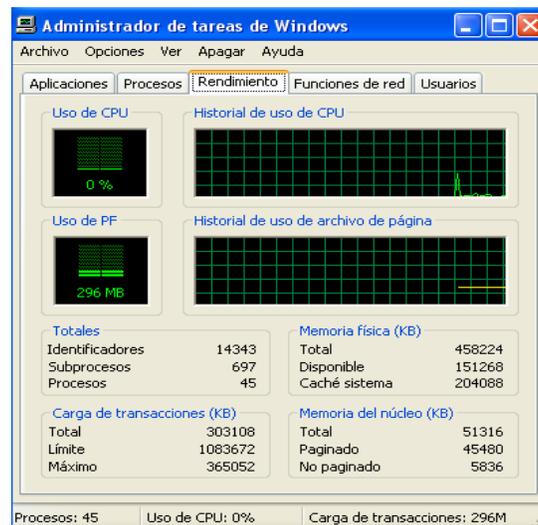


Figura 26: Rendimiento del PC No.1 al activar el servidor de base de datos

Una vez inicializado MySQL, se procede a ejecutar el servidor SIPMethod, que es el contenedor de los Servlets SIP y por tanto es el encargado de soportar la gestión de las sesiones SIP. También, se ejecuta la aplicación encargada de la Lógica del Servicio.

Se pudo concluir que SIPMethod es relativamente liviano para este tipo de aplicaciones, y que el momento de mayor carga del servidor ocurre la primera vez que se consulta el correo del usuario, pues se debe realizar una conexión con MySQL y con el servidor de correo, por medio del módulo de la Lógica del Servicio, ver Figura 27. La conexión con el servidor de correo es a una tasa máxima, de transferencia de datos, de 10Mbps.

Por otro lado, el uso de la CPU se incrementa solamente al cargar los componentes y aplicaciones desplegadas en el SIPMethod, y posteriormente solo se observa un incremento en el consumo de memoria virtual.

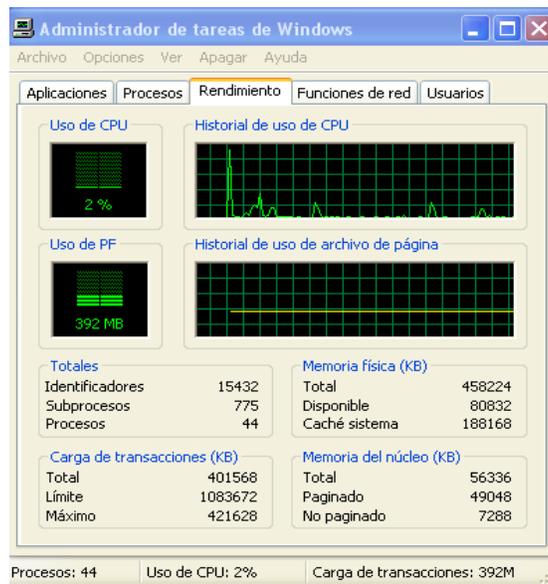


Figura 27: Rendimiento cuando se ejecuta SIPMethod

- **Cliente Móvil**

Con la ejecución del Cliente Móvil se registró un pequeño aumento en el consumo de la CPU, esto se debe a que utiliza varios hilos que corren simultáneamente, ver Figura 28. Sin embargo, se requiere una prueba más detallada que se mostrará luego.

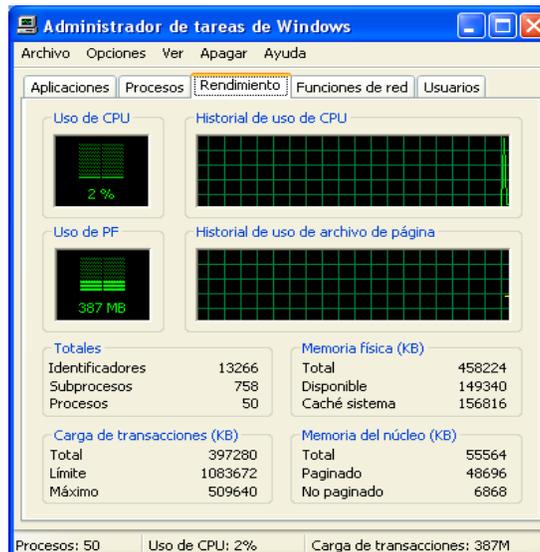


Figura 28: Rendimiento cuando se ejecuta el Cliente Móvil



- Pruebas en el PC No.2 (MRF)
  - MRF

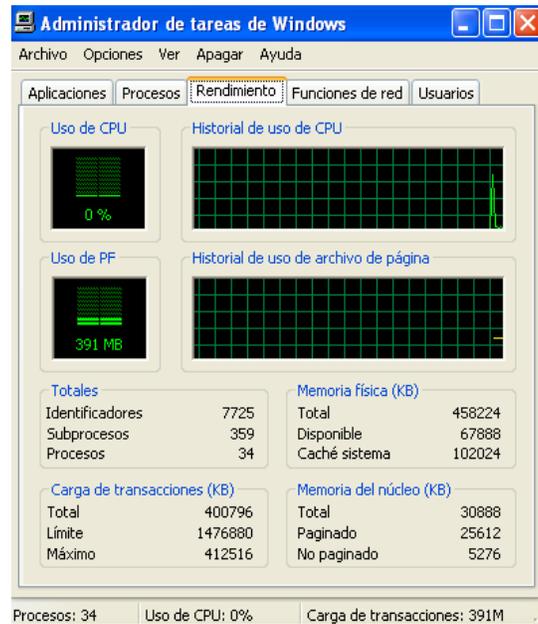
El MRF esta instalado en el PC No.2, y las pruebas se realizaron de la misma manera.

Inicialmente, sin correr el módulo de la MRF, se tiene la Figura 29.



**Figura 29: Rendimiento del PC No.2 sin programas en ejecución**

Ahora, se inicia VerbioTTS y VerbioASR. El rendimiento se registra en la Figura 30.



**Figura 30: Rendimiento cuando se ejecuta Verbio TTS/ASR**

De lo anterior se puede concluir que Verbio incrementa el uso de la CPU, solamente, al cargar los locutores y licencias de reconocimiento, luego el consumo disminuye hasta cero, y aumenta muy ligeramente al enviarle muestras de audio.

Por último, se ejecuta la MRF. Únicamente se registra un incremento en el consumo de memoria virtual y un ligero aumento en el uso de la CPU, al conectarse con Verbio, esto se debe especialmente a que la MRF está desarrollada en C++ que es un lenguaje potente y, al mismo tiempo, liviano. Ver Figura 31.

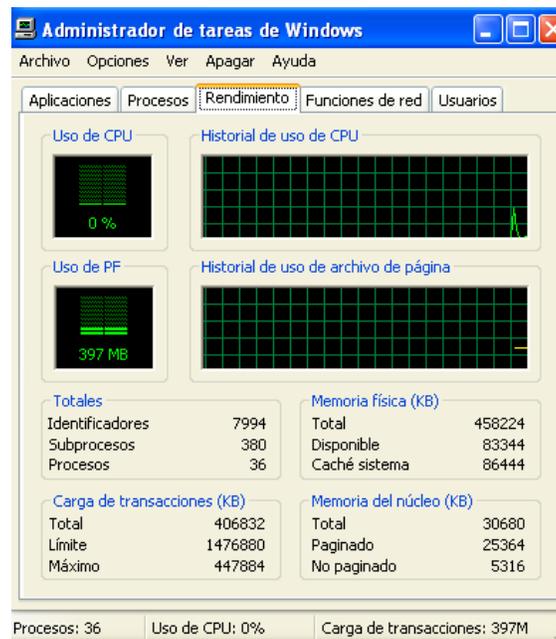


Figura 31: Rendimiento cuando se ejecuta la MRF

### 5.3.3. Pruebas de Usabilidad

Para estas pruebas fue necesario hacer una pequeña encuesta con 3 usuarios que interactuaron con el sistema, la encuesta se presenta en la Tabla 8.

Usuarios	¿Los usuarios pueden utilizar el sistema de forma rápida y sin problemas?	¿Qué factores afectan la satisfacción del usuario?	La navegación ¿es intuitiva? ¿Está diseñada correctamente?
<u>Usuario1</u>	Si	Los retardos en la conexión con el servidor de correo.	Si
<u>Usuario2</u>	Si	La interpretación de siglas, URLs y palabras en inglés de algunos mensajes de correo.	Si
<u>Usuario3</u>	No, hubo un problema con la ejecución de una tarea (comando de voz)	En ocasiones, fallas de interpretación en el reconocimiento del habla.	Si

Tabla 8: Pruebas de Usabilidad



### 5.3.4. Pruebas de uso de Memoria

Esta prueba se realizó, específicamente, en el Cliente Móvil para observar el consumo de memoria de ésta aplicación, y con el fin de estimar los recursos que consumiría en un dispositivo real (para ello se utilizaron las herramientas de medición que incorpora el WTK 2.5, versión Beta).

Inicialmente, para cargar la aplicación se requieren 452Kb, como se muestra en la Figura 32.

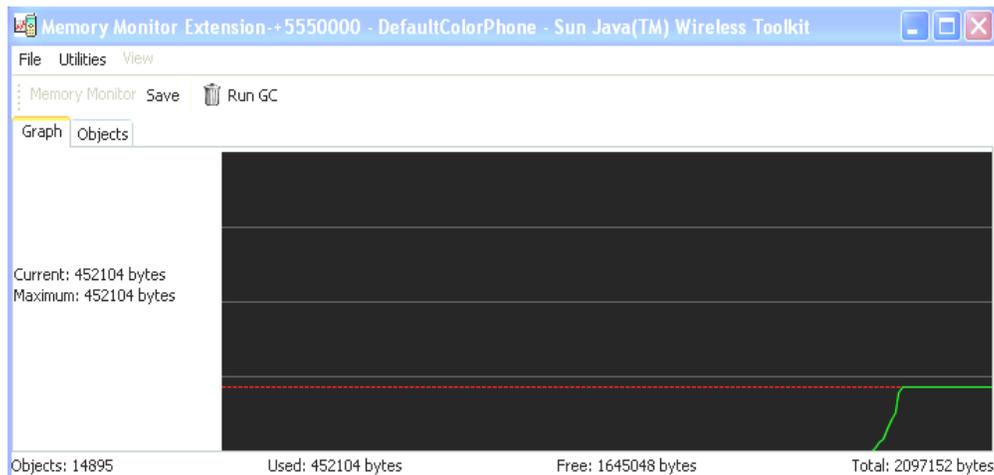


Figura 32: Uso de memoria inicial del Cliente Móvil

Una vez lanzada la aplicación, se procede a realizar el registro del cliente dentro del sistema IMS. Este módulo (Cliente Móvil) consume 117Kb como se observa en Figura 33.

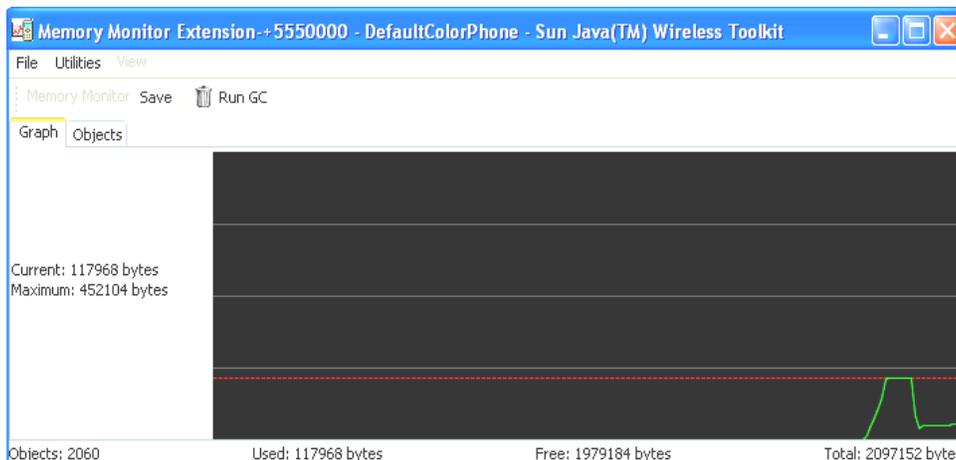
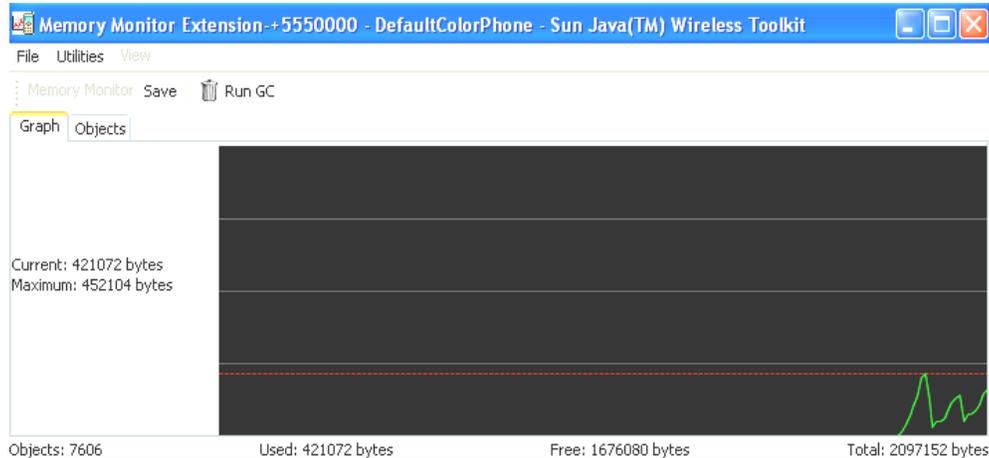


Figura 33: Uso de memoria para el registro del Cliente Móvil

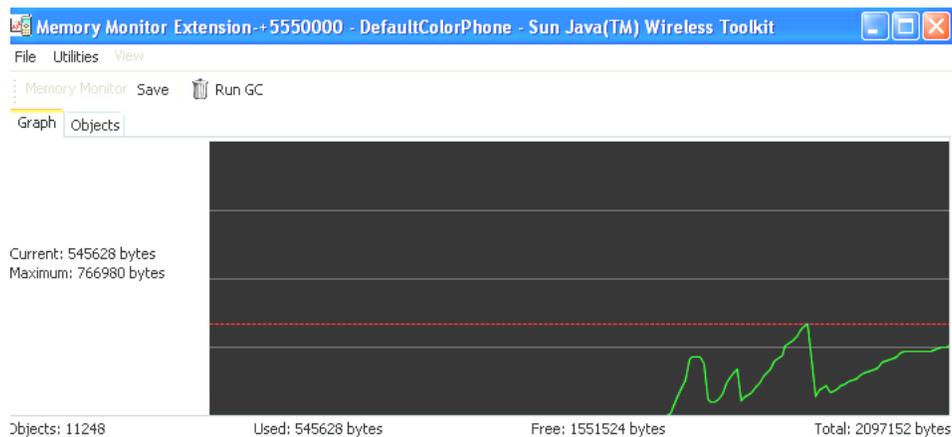


Al iniciar la sesión, se notó un incremento de memoria. Por consiguiente, para iniciar la sesión IVR se necesitan 421Kb.



**Figura 34: Uso de memoria para el inicio de sesión del Cliente Móvil**

Una vez establecida la sesión, se realiza el intercambio de flujo RTP. Al llevar a cabo la prueba, se concluyó que se requieren 545Kb para dicho procedimiento. Por esto, se concluye que este es el momento donde la carga del equipo es más alta, puesto que intervienen 4 procesos: sincronización de paquetes RTP, reproducción de audio, captura de audio y envío de paquetes RTP. Ver Figura 35.



**Figura 35: Uso de memoria para el intercambio de flujo RTP del Cliente Móvil**

### 5.3.5. Pruebas de Ancho de Banda necesario

El intercambio de paquetes se realiza para dos protocolos: SIP y RTP. Para determinar el tamaño de paquetes, se utiliza el monitor de red del WTK.



- SIP

A través de esta herramienta, se puede comprobar que el proceso de SIP se realiza adecuadamente, ver Figura 36.

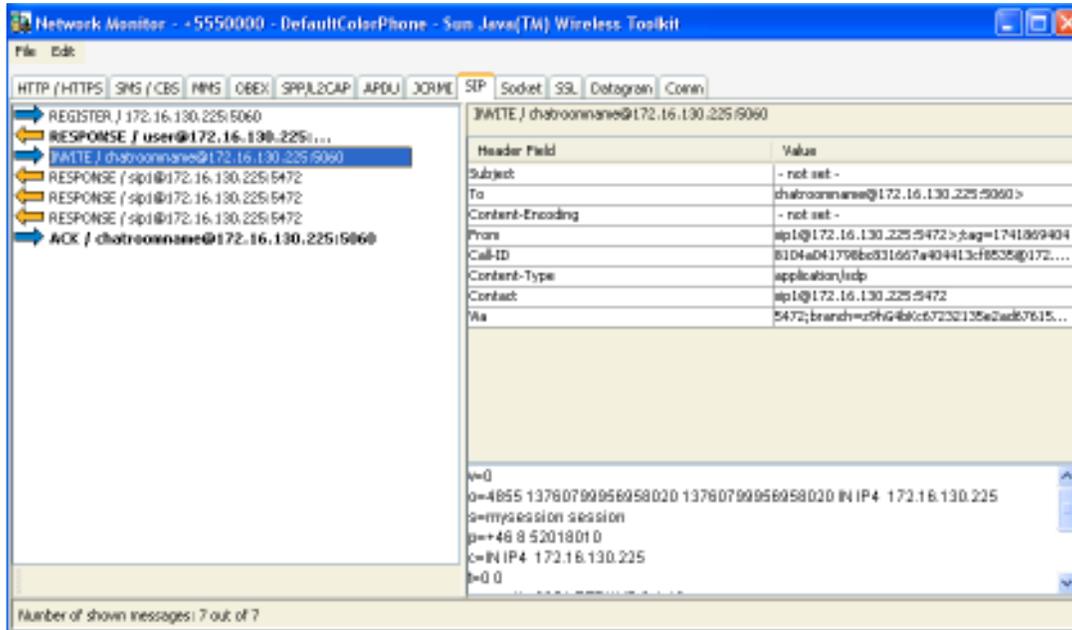


Figura 36: Mensajes SIP utilizados por la aplicación

También, se puede medir el número de bytes involucrados en este proceso, que en este caso es de 1012 para el registro e inicio de sesión, siendo el mensaje INVITE el más grande para transportar.

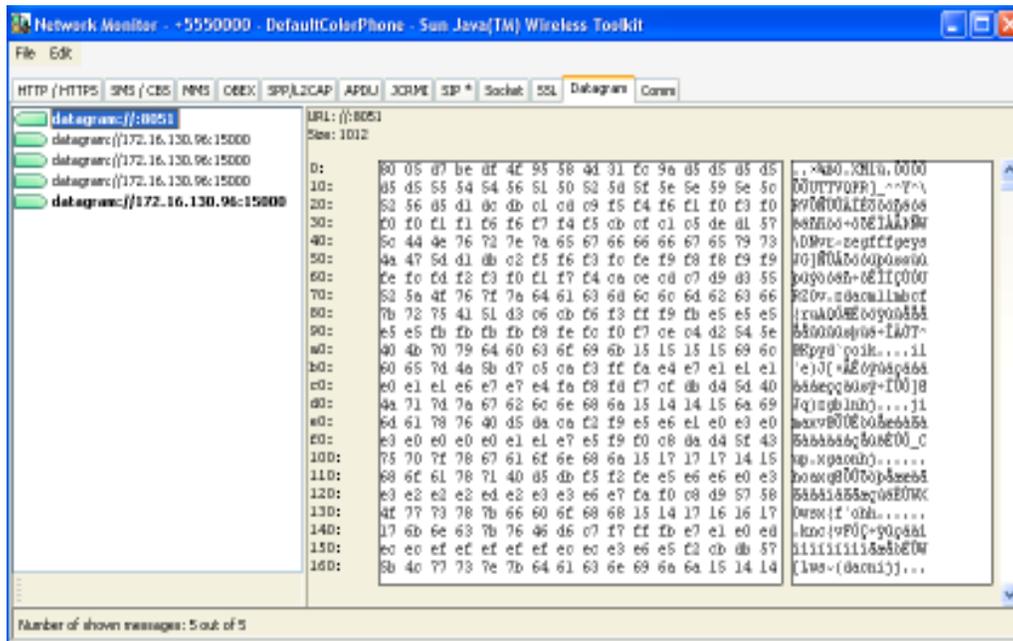


Figura 37: Datagramas empleados en SIP

- RTP

Como se muestra en la Figura 38, el tamaño de los paquetes es de 1012 en 110 ms, luego el ancho de banda utilizado es de 9200bps, esto debido a que se utiliza el codec G.711 (PCM).

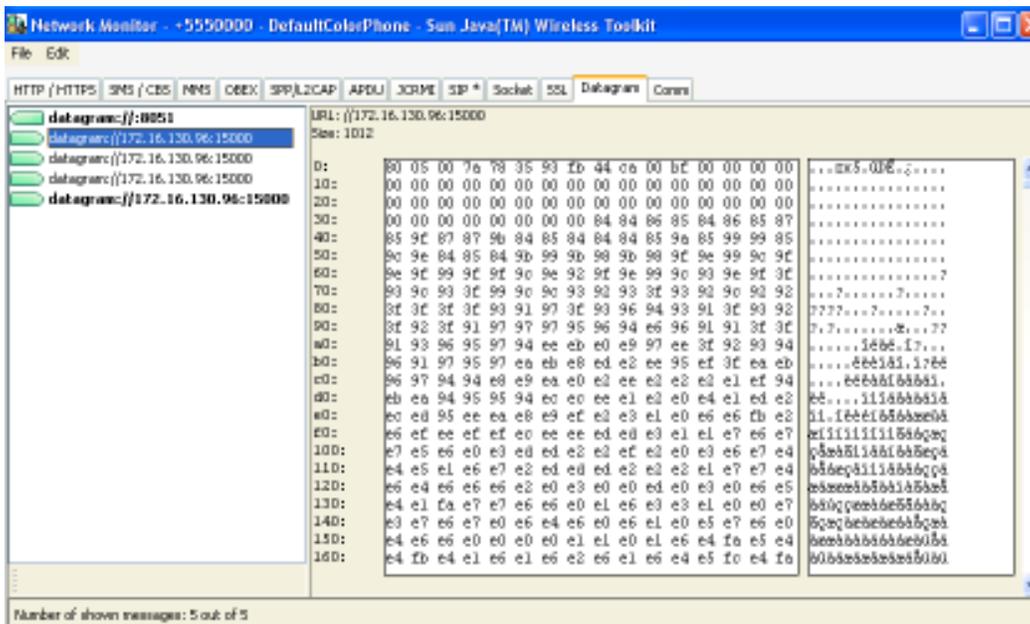


Figura 38: Datagramas empleados en RTP



#### 5.4. APORTES

- API de RTP para J2ME. El protocolo RTP es esencial para la creación de un cliente SIP con capacidades de voz. Sin embargo J2ME no lo incluye en sus APIs (hay algunos desarrollos para RTSP), por tanto fue necesario realizar una implementación de dicho protocolo a través de la adaptación de un API de J2SE
- Codec G.711, PCM lineal, leyes A y  $\mu$ , para J2ME. El MMAPAPI (Mobile Media API) para manejo de audio, en PCM, solo soporta reproducción y captura de audio en PCM lineal. Sin embargo, PCM lineal no es uno de los formatos aceptados para RTP, de modo que se implementó para J2ME un codec PCM (lineal) / PCM (ley A o  $\mu$ ) (que si es aceptado en el RFC), para que los paquetes puedan ser procesados correctamente en otros agentes de usuario o servidores.
- Cliente Móvil con acceso a servicios IVR. Es la integración de la parte SIP con RTP, para lograr un cliente con capacidades de manejar tanto flujos de señalización como flujos de datos.
- Funcionalidad IVR (ASR y TTS) en IMS. Utilizando Verbio se desarrolló un prototipo de MRF que facilita las tareas de IVR. Este MRF consta de una parte que se comunica con el servidor de aplicaciones a través de mensajes SIP, y conforma el MRFC, y los motores ASR y TTS de Verbio que constituyen el MRFP, y cuya comunicación con el sistema es a través de RTP.
- Gestión de la sesión de audio en IMS: registro y desconexión IMS; inicio, cancelación, mantenimiento y terminación de sesiones de audio (Servlets SIP). Utilizando el API de Servlets SIP, se realizó un prototipo de AS/S-CSCF/P-CSCF que soporta la funcionalidad para establecer sesiones de audio, soportadas por el protocolo SIP.
- Recomendaciones sobre el desarrollo de clientes móviles con acceso a servicios IVR. Se registraron algunas sugerencias para mejorar la usabilidad en aplicaciones IVR, estas recomendaciones se recolectaron de fuentes externas y de la experiencia en el proyecto.
- Recomendaciones sobre el desarrollo de servicios IVR en IMS. Basados en la experiencia con el proyecto, se definieron algunos lineamientos sobre el desarrollo de servicios basados en IMS (en especial IVR), los cuales brindan soporte para trabajos futuros que aborden estas temáticas



## **6. RECOMENDACIONES PARA EL DESARROLLO DE CLIENTES MÓVILES IVR BASADO EN LA ARQUITECTURA IMS**

### **6.1. RECOMENDACIONES PARA EL DESARROLLO DE UN SISTEMA IVR, CON CAPACIDADES DE ASR Y TTS**

#### **6.1.1. Recomendaciones para el desarrollo de sistemas IVR basados en reconocimiento de habla**

En el desarrollo de aplicaciones IVR se debe tener en cuenta ciertos aspectos que mejoran la interacción con el usuario final y por consiguiente la accesibilidad de éste a diversos servicios. Con base a las referencias bibliográficas de la [94] a la [101], a continuación se describirá una serie de sugerencias importantes que tienen que ver especialmente con sistemas IVR basados en reconocimiento de habla. Algunas de esas sugerencias, se tuvieron en cuenta para el análisis, diseño e implementación del servicio ASR.

Primero se listan las posibles causas de rechazo y las limitaciones de las interfaces de voz:

Causas de Rechazo:

- Los usuarios se pierden, no encuentran lo que necesitan.
- No entienden cómo deben utilizar el sistema.
- Comportamientos inesperados del sistema.
- Errores continuos en el reconocimiento de la voz provocan fatiga y frustración.

Limitaciones de las interfaces de voz:

- Falta de persistencia: la información se presenta y desaparece.
- Es un proceso lento comparado con otros estilos de interfaz.

Con base a lo anterior se describen las recomendaciones a seguir:

- **Identificación de la Audiencia Objetivo [94][95]**

Es importante conocer al usuario antes de empezar el diseño, con el fin de crear una aplicación de acuerdo con sus necesidades y probar la aplicación tan rápido como sea posible.

Para este propósito, el Diseño Centrado en el Usuario (User Centred Design, UCD) es un acercamiento de diseño donde el usuario final es colocado en el corazón del diseño y del proceso de desarrollo. La clave principal de UCD es integrar a los usuarios que representan el perfil del grupo de usuarios objetivo y un aspecto fundamental de UCD son las tareas de Análisis de Usuario que se explican a continuación.



### ✓ Tareas de Análisis de usuario

Es la identificación y comprensión completa en las tareas que realiza el usuario.

Primero se debe establecer las herramientas que los usuarios utilizan actualmente, el modelo mental con el que desarrollan las tareas y los problemas o inconvenientes que podrían encontrar.

Este análisis puede realizarse haciendo varias suposiciones (basadas en el grupo o audiencia objetivo) e idealmente reuniendo la reacción de usuarios verdaderos y observando como realizan las tareas.

Para identificar la audiencia objetivo es necesario enfocar la investigación para reunir los datos en los siguientes parámetros:

- Tipo de llamantes.
- Grupos de edad de los llamantes.

### • Tipos de llamantes [96]

Cada tipo de llamante tiene alguna característica distinta que afecta la evaluación de una aplicación. En este sentido, estos se enmarcaron dentro de dos categorías amplias: *principiantes* y *experimentados*.

#### ▪ Principiantes

No tienen familiaridad con avisos de síntesis de voz y podrían encontrar dificultad al reconocerlos.

Un problema común asociado con los principiantes es su incapacidad para interactuar apropiadamente con los sistemas para proporcionar una entrada de habla correcta ya que a menudo tienden a utilizar palabras como: “Umm”, “ahh”, mientras proveen la entrada de audio. Esta es una dificultad para que el motor de reconocimiento entienda las palabras pronunciadas.

El siguiente ejemplo de dialogo ilustra la falta de interacción apropiada entre el usuario y un sistema debido a la anomalía en la entrada de un usuario principiante:

**Sistema:** ¿Que tipo de pizza desea?

**Usuario:** Umm...., una pizza simple de queso.

**Sistema:** No entendí lo que usted dijo.

Algunos puntos que se deben considerar mientras se desarrolla una aplicación para principiantes son:

- ✓ Hacer diálogos breves y efectivos.
- ✓ Mantener claridad fonética durante la navegación.
- ✓ Mantener un diseño de gramática simple, incluyendo todas las posibilidades fonéticas de alguna palabra hablada esperada por el usuario.



- ✓ Informar al usuario sobre funciones de acceso universal tales como la salida o la ayuda para que el o ella pueda beneficiarse de estas utilidades.

#### ▪ **Usuarios Experimentados**

Es relativamente más sencillo desarrollar aplicaciones para este tipo de usuarios. Algunos consejos útiles son:

- ✓ Los usuarios experimentados siempre esperan un diseño de dialogo consecuente y avisos de información puntuales.
- ✓ Incluir más palabras claves en la gramática para la navegación y selección rápida para ahorrar tiempo y esfuerzo.
- ✓ Utilizar mensajes de error y ayuda puntual del sistema. Evitar mensajes de ayuda muy elaborados y confusos.
- ✓ Después de haber identificado el tipo de llamante el próximo paso es identificar los grupos de edad de los llamantes y analizar sus rasgos de personalidad.
- ✓ Los llamantes que pertenecen a grupos de edad diferentes tales como niños, adolescentes, etc. tienen algunas características que afectan su interacción con la aplicación.

#### • **Grupos de edad [97]**

Para desarrollar aplicaciones de voz, dirigidas a un grupo particular de audiencia primero se deben recolectar datos considerando el grupo de edad de los llamantes esperados. Luego diseñar diálogos y cuestionamientos basados en estos datos. A continuación, se mencionan algunos.

#### ▪ **Niños**

Los niños siempre esperan algo de entretenimiento. Por ello es útil incluir algunos diálogos amigables y si es posible que los diviertan.

Algunos de los puntos que se deben considerar mientras se desarrolla una aplicación de voz para niños son:

- ✓ Diseñar diálogos atractivos e imaginarios para impresionar a los niños y para ayudarlos a utilizar la aplicación correctamente.
- ✓ En mensajes de error se les dirige en un dialogo informal y amistoso como en el siguiente dialogo:

**Sistema:** “por favor selecciona entre uno y dos”

**Usuario:** “Tres”

**Sistema:** Tu selección no está disponible, selecciona nuevamente.

**Usuario:** dos.

**Sistema:** Eso es correcto, seleccionaste una opción valida.

#### ▪ **Adolescentes**

Para diseñar aplicaciones dirigidas a adolescentes como un almacén de música, etc. se deben tener en cuenta los siguientes puntos:



- ✓ Evitar los diálogos extensos porque los adolescentes siempre quieren que las cosas sucedan rápida y fácilmente.
- ✓ Utilizar palabras claves para terminar la sesión y para facilitar el flujo navegación en cualquier situación al obtener acceso a la aplicación.

#### ▪ **Adultos [98]**

La categoría de adultos se divide en diferentes subgrupos, esto requiere un manejo cuidadoso mientras se desarrolla la aplicación de voz.

Dichos subgrupos recorren los grupos que se denominaron anteriormente: principiantes y experimentados.

Al realizar una aplicación de voz para adultos se deben incluir las siguientes pautas:

- ✓ Mantener diálogos cortos y fáciles de entender para que los usuarios no deban escuchar con cuidado cada vez que acceden a la aplicación. Evitar que los usuarios tengan que memorizar datos relativos a la aplicación, en la medida de lo posible.
- ✓ Otra sugerencia (en la medida de lo posible) es permitir al usuario llamar a la línea de métodos de identificación con el fin de personalizar la aplicación en escenarios específicos. Esto se logra con autenticación que en caso de no hacerlo por voz, se recomienda utilizar un número de identificación personal para dicho objetivo.

#### • **Consideraciones generales [94]**

A parte de las consideraciones sobre la edad, hay algunas consideraciones generales que se deben tener en cuenta:

- ✓ Después del diseño de los cuestionamientos e implementarlos es importante realizar una prueba audible para escucharlos repetidamente. Algunos de los diálogos pueden parecer perfectos en el papel. Pero al escucharlos pueden no ser tan claros y los usuarios podrían tener dificultad al entenderlos.
- ✓ Evitar palabras que no sean fáciles de entender mientras son sintetizadas por el motor de voz (homónimas) en un mismo diálogo.
- ✓ Al interactuar con la aplicación el usuario debe tener la capacidad para terminar la aplicación y terminar la llamada en cualquier etapa de la sesión.
- ✓ Proporcionar la opción para repetir avisos en cada etapa de la aplicación, para evitar la pérdida de información mientras el usuario escucha.
- ✓ Los principios del diseño centrado en el usuario pueden ser aplicables para lograr que las aplicaciones IVR sean más eficientes e intuitivas.
- ✓ Una sola interfaz de audio significa que el usuario solo puede escuchar una sola cosa al tiempo. La interfaz es secuencial. Con interfaces visuales, como ventanas, es posible acentuar el contenido por los tipos de letra y color. El usuario puede revisar cualquier parte de la pantalla en una mirada. Por tanto la manera y el orden en la que la información se brinda es un aspecto muy importante.



▪ **Saludo inicial**

- ✓ Debe haber una bienvenida breve: buenos días, buenas tardes...
- ✓ No repetirlo si en algún momento el usuario vuelve al inicio.
- ✓ Es preciso evitar mensajes promocionales: ponerlos solamente si el usuario los puede encontrar justificados y, sobretodo, deben ser breves permitiendo acceder a la información en aquel mismo momento.
- ✓ Remitir al usuario a otro servicio.

▪ **Menús y Diseño de Diálogos [95]**

Cuando la información es muy grande en todos los cuestionamientos se disminuye el rendimiento y por ello los usuarios tienden a irritarse y a preferir otra alternativa de interacción como la Web o persona-persona.

Los dos puntos clave en el desarrollo de aplicaciones de voz son: el diseño del diálogo y la exactitud en el reconocimiento de habla, asegurándose que la aplicación interprete lo que el usuario dice realmente. Esos dos conceptos son necesarios para proveer una aplicación amigable que las personas deseen utilizar de nuevo.

En el diseño de diálogos hay palabras pequeñas que pueden parecer insignificantes (no funcionales) pero que juegan un papel importante en la continuidad de un diálogo debido a que enlazan de forma natural la conversación, estas palabras son conocidas como marcas de discurso y a continuación se tiene un ejemplo:

**Sistema:** “¿Hacia qué localidad desea realizar el viaje?”.

**Usuario:** “Armenia”

**Sistema:** ¿Qué día desea viajar?

**Usuario:** El próximo martes.

**Sistema:** ¿A qué hora desea viajar?

**Usuario:** Al medio día.

Este dialogo no es natural ni continuo. Es necesario agregar palabras de confirmación que mejoren la conversación:

Ejemplo:

**Sistema:** “¿Hacia qué localidad desea realizar el viaje?”.

**Usuario:** “Armenia”

**Sistema:** Armenia, perfecto, y ¿qué día necesita viajar?

**Usuario:** El próximo martes.

**Sistema:** Está bien, el martes 12 de abril, y ¿a qué hora?

**Usuario:** Al medio día.

En general las personas están más familiarizadas con DTMF que con reconocimiento de voz. Esta familiaridad influencia en cómo los usuarios esperan que la aplicación esté construida. Al ajustarse a un nuevo medio, en este caso reconocimiento de voz, las personas tienden a compararlo con el sistema tradicional. Por ejemplo, en un sistema DTMF un diálogo se estructura de la siguiente forma:

**Sistema:** “Presione 1 para balance de cuenta, 2 para transferir.”



Los usuarios de sistemas ASR esperarían algo como:

**Sistema:** "Para balance de cuenta diga balance de cuenta, para transferir diga transferir".

Sin embargo esto es poco eficiente y la mejor alternativa sería:

**Sistema:** "Por favor elija una opción"  
"Balance de Cuenta"  
"Transferir"

A continuación se enumeran sugerencias relativas al desarrollo de menús:

- ✓ Deben estar provistos de un mensaje de entrada: título del menú que indica al usuario donde ha ido a parar y de qué tratan las opciones que escuchará. Ejemplo: "Información de contactos"
- ✓ Máximo 4 opciones por menú, si se utiliza la respuesta por tonos.
- ✓ Máximo 5 opciones por menú, si se utiliza respuesta por reconocimiento de voz (ya que es improbable que el usuario recuerde más de ese número)
- ✓ Máximo 3 pasos para hacer una tarea.
- ✓ Poner las opciones de navegación al final de todo.
- ✓ Proporcionar sólo la información que se necesita y de la forma más simple posible.
- ✓ Hacer que el sistema trabaje por el usuario, por ejemplo, no pidiendo el mismo dato varias veces y evitando preguntar información que el sistema puede conocer.
- ✓ Ser consistente y no cambiar a menudo las opciones de los menús.
- ✓ Si el servicio tiene alguna restricción, es preciso avisar al usuario primero de todo. Por ejemplo, en un servicio de compra de entradas a alguna actividad de entretenimiento, si se requiere que el usuario sea socio, es preciso avisarle antes de empezar.
- ✓ Ordenar las opciones de las más específicas a las más generales. Por ejemplo:

*Orden incorrecto:*

Noticias  
Deportes  
Economía

*Orden correcto:*

Deportes  
Economía  
Noticias

- ✓ Si existe una opción de paso al operador, debe ser la última.
- ✓ No hay que cortar nunca la comunicación si el usuario se equivoca. Es preciso dar caminos alternativos.
- ✓ Personalizar (si es posible) el contenido de acuerdo con el usuario, esto se logra con autenticación que en caso de no hacerlo por voz, se recomienda utilizar número de identificación personal para dicho objetivo.



#### ▪ Preguntas Sí/No [97]

Se usan para recuperar errores y para confirmar acciones.

Se debe tener presente que en español los sistemas de reconocimiento de voz dan bastantes problemas para reconocer el Sí, por lo que siempre que sea posible, se evitará el uso de preguntas Sí/No.

✓ Tienen que ser preguntas directas que induzcan a decir de forma natural sólo Sí o No:

*Incorrecto:*

Por favor, confirme si su cédula es: XXXXXXXX

*Correcto:*

Cédula: XXXXXXXX. ¿Es correcto?

- ✓ Debe ser una única pregunta, no dos en una.
- ✓ Si además de la pregunta hay texto informativo, poner la pregunta al final de la frase.  
Ejemplo: Para obtener más información sobre las condiciones del contrato, ¿desea que pase la llamada a un agente?
- ✓ No abusar de este tipo de preguntas.
- ✓ Evitar concatenaciones de preguntas Sí/No.
- ✓ Es mejor utilizar expresiones específicas en lugar de preguntas Sí/No.  
Ejemplo: Si quiere cancelar la suscripción diga: dar de baja.

#### ▪ Menú principal

- ✓ El primer menú debe ser el principal a no ser que tenga que pedirse identificación.
- ✓ Máximo 20 segundos de duración total contando el saludo inicial y el menú principal.

#### ▪ Identificación

- ✓ Pedir identificación sólo cuando sea necesaria.
- ✓ Pedir el identificador más fácil de recordar: teléfono, CC, etc.
- ✓ Aceptar que el usuario dé cadenas variables de dígitos sin causarle un error.
- ✓ Gestionar el final de la entrada de datos por medio de un tiempo establecido, no solicitar ninguna acción adicional.

#### ▪ Lenguaje [98]

- ✓ Hablar el idioma del usuario, evitar palabras técnicas o no tan usuales.
- ✓ Las funciones atribuidas a claves particulares y la terminología debe ser consecuente a través de los diálogos.
- ✓ Seleccionar al locutor de la aplicación con detenimiento, el tono, la inflexión la velocidad e incluso el género pueden influir en la actitud hacia la aplicación.



▪ **Navegación [96]**

Desde el comienzo el usuario debe ser informado de cómo navegar en el sistema y cómo obtener ayuda. Si el sistema es intuitivo los mensajes de ayuda no serán necesarios.

Las opciones de navegación recomendables son:

- ✓ Volver atrás
- ✓ Menú principal
- ✓ Repetir
- ✓ Ayuda (sólo si lo necesita la aplicación)

▪ **Mensajes de Error [99]**

La manera en que los errores se tratan es importante. Los llamantes no deben llegar a puntos muertos, es decir situaciones donde no se les informó cómo obtener ayuda o cómo salir o cancelar acciones.

Los mensajes de error se generan por respuestas del usuario que el sistema no entiende o bien cuando el usuario se queda callado. Sin embargo, no se debe culpar nunca al usuario.

Ejemplo: hable más alto, hable más claro...  
Se le debe explicar al usuario lo que ha sucedido.

Ejemplo: Perdone, no le he entendido, inténtelo de nuevo.

Y en estos casos, se debe reformular la pregunta: ampliando la información o proporcionando algún ejemplo.

▪ **Errores de sustitución y falsa aceptación [100]**

Se trata de los errores que puede cometer el usuario al escoger una opción equivocada.

Para prevenir que estos errores puedan ser graves, es preciso que, en función de la criticidad de la aplicación, se pida confirmación y se permita cambiar de opción.

*Confirmación explícita:* cuando se le pide al usuario que confirme la acción que se está apunto de hacer. La aplicación no avanza sin un Sí o un No.

Ejemplo: Por favor, confirme ¿quiere cambiar su tarifa?

*Confirmación implícita:* se avisa al usuario de lo que está haciendo el sistema a partir de lo que éste ha entendido y le permite el cambio de opción. Secuencia de ejemplo:

**Usuario:** Correos nuevos

**Sistema:** Recuperando correos antiguos

**Usuario:** No, los nuevos

**Sistema:** Perdone, ¿Desea pasar a los correos nuevos?

**Usuario:** Sí



**Sistema:** Recuperando correos nuevos

*Confirmación interna:* se aprovechan procesos internos para obtener la confirmación. Por ejemplo, el hecho de pedir una contraseña, ya supone una confirmación.

*Sin confirmación:* cuando el error que se puede producir no es importante. Aun así, el mensaje de entrada del paso siguiente también ayuda al usuario a saber si la opción que ha pedido es la correcta.

▪ **Escalabilidad [101]**

Es imperativo recordar que una aplicación IVR puede crecer en el futuro. Por tanto es necesario planear la estructura del menú, profundidad, claves de selección y terminología para asegurar flexibilidad y añadir nuevos elementos fácilmente.

▪ **Realimentación [96]**

Debe ser breve, pero a su vez proveer toda la información necesaria. Una recomendación es utilizar el famoso lenguaje *Philosopher Grice* que especifica una manera efectiva de realizar conversaciones cooperativas.

Las máximas de Grice son:

- **Máxima de calidad:** Indica que todo lo que se dice debe ser verdadero y que pueda ser sustentado con evidencia.
- **Máxima de cantidad:** Dice que la contribución que realiza un hablante (para propósito de la conversación) debe corresponder con la información requerida, pero ni menos ni más.
- **Máxima de relevancia:** La contribución que realiza un hablante debe ser relevante en el contexto de la conversación.
- **Máxima de manera:** Se debe evitar ambigüedad u oscurecer el sentido de las expresiones.

Es importante separar de la realimentación la ayuda o guía para que solamente la reciba quien la necesita o a quien la solicite en situaciones particulares.

La funcionalidad de la realimentación en sistemas de habla es un área relativamente pequeña.

▪ **Comandos Universales**

Una solución para el problema de “qué debo decir” es el concepto de comandos universales, es decir comandos unificados que estén disponibles en múltiples aplicaciones y que sean familiares para los usuarios.

El uso de comandos universales requiere aprendizaje, sin embargo son útiles porque pueden proveer ayuda, orientación de navegación, corrección de errores e interacción general con el sistema.



La falta de estándares en aplicaciones de habla ha llevado a una situación en la que las aplicaciones trabajan de diferentes maneras y los usuarios necesitan aprender una nueva interfaz cada vez que cambian de aplicación. Sin embargo, los comandos universales son una manera de alcanzar el acceso global para todos los usuarios.

### 6.1.2. Otras recomendaciones

- **Selección del motor ASR, impactos en el resto del sistema.**

El estudio y selección de los diferentes motores ASR que existen es uno de los procesos más importantes y que requiere de una buena parte del tiempo para su realización. Las empresas que crean estas tecnologías y las introducen comercialmente, invierten bastante dinero en su desarrollo y como tal son los precios para el licenciamiento de este tipo de software. Por otro lado, también existen los desarrollos de código libre o licenciamiento gratuito durante un periodo de utilización, dentro de los cuales también se encuentra software apoyado y soportado por grupos desarrolladores muy importantes a nivel mundial.

Para conocer las ventajas y desventajas de un motor ASR se necesita estudiar sus capacidades, la forma en que opera, y llevar a cabo su ejecución. Con respecto a las herramientas de código libre como Sphinx se encuentran requisitos tales como el entrenamiento previo del motor y la instalación de programas adicionales. Para el proceso de entrenamiento y de reconocimiento del habla se necesitan los modelos acústicos y gramaticales del lenguaje a utilizar, que por lo general se encuentran para el inglés, y para el español estos modelos se encuentran en bases de datos que no son de fácil acceso.

En el área de los motores ASR comerciales se pueden encontrar mayores ventajas en el tiempo que requieren para su estudio y puesta en funcionamiento, y la desventaja que más resalta son sus costos. Sin embargo, es posible encontrar herramientas comerciales que ofrezcan una versión gratuita, con la salvedad de que pueden tener restricciones en su tiempo de ejecución y/o de utilización.

Ya que la aceptación de la aplicación depende en gran medida de la precisión del ASR se debe elegir un motor que cumpla con las siguientes características:

- Independiente del usuario.
- Reconocimiento de habla continuo.
- Si es posible, que soporte el acento de los usuarios que van a utilizarlo.

Se debe entonces:

- ✓ Investigar las APIs de integración asociadas con el motor ASR, las cuales permiten la comunicación del motor con otros módulos del sistema.
- ✓ Investigar si el motor ASR está ligado a un entorno de programación específico, esto ocurre generalmente con motores propietarios que en ocasiones solo permiten la compilación de las aplicaciones en un IDE como Visual C++ o IBM WebSphere.



- ✓ Además debido a que el sistema ASR va a realizar una parte pesada del procesamiento es conveniente utilizar un lenguaje liviano y de buenas características como por ejemplo C++ (si está soportado por el motor).
- ✓ También es útil tener en cuenta los formatos de audio que el motor de reconocimiento permite pues algunos de ellos solo reconocen voz en un solo formato (como PCM Ley A o  $\mu$ ).
- ✓ Es muy útil investigar las gramáticas ABNF que manejan varios motores de reconocimiento y se utilizan en general en aplicaciones de voz.
- ✓ Se debe tener en cuenta que el ASR es un módulo que debe funcionar en forma recurrente por tanto es necesario conocer sobre las características de hilos en el lenguaje de programación empleado.
- ✓ De antemano buscar APIs o librerías para SIP y RTP en el lenguaje que maneja el ASR ya que este debe comunicarse con los demás módulos de la arquitectura a través de esos protocolos.

- **Selección del motor TTS**

A la hora de escoger el motor TTS, se deben tener en cuenta las siguientes características:

- ✓ Que tenga una salida de voz natural e inteligible.
- ✓ Que funcione en modo barge-in (permita interrupciones por parte del usuario).
- ✓ Que pueda ser compatible con el ASR seleccionado previamente.

## **6.2. RECOMENDACIONES PARA EL DESARROLLO DE LOS MÓDULOS FUNCIONALES DE LA ARQUITECTURA IMS EN COMBINACIÓN CON LA ARQUITECTURA DEL PROTOCOLO SIP**

- **Conocimiento claro de la arquitectura IMS para el desarrollo de aplicaciones con acceso a recursos multimedia**

Se recomienda realizar un estudio general de la arquitectura de servicios IMS. Los elementos que la componen, los protocolos que utiliza, haciendo énfasis en aquellos que pueden llegar a formar parte fundamental del proyecto y en los cuales se debe profundizar después, los objetivos que han llevado a la creación de esta arquitectura para de esta forma llegar a entender su filosofía, la forma en que se ha ido especificando, su estado y características actuales, las características y limitaciones de la infraestructura que propone para su aplicación en redes pre-IMS como las redes de 2.5G, entre otras temáticas.

Es necesario que durante este estudio se tenga como objetivo la búsqueda del elemento o elementos que se van a encargar de prestar las funcionalidades multimedia para la creación del sistema IVR. Además de los elementos que representarán el punto de entrada de un usuario a la red, ya que por tratarse de una red que busca la convergencia de diversas redes, presenta así



mismo interfaces diferentes. Estos conceptos deben tenerse muy claros antes de iniciar con el diseño e implementación de los módulos de la arquitectura que realmente se necesitan.

Luego de un estudio general, se puede entrar a detallar aquellos módulos que se encontraron como esenciales para el montaje de una infraestructura de red que permita la creación de servicios IVR (ASR y TTS) para usuarios móviles de redes 2.5G. En este caso, se deben obtener las características de cada elemento de red perteneciente a un módulo funcional más complejo, la forma en que operan y se comunican los elementos constituyentes de un módulo, y la forma como cada módulo se relaciona con los demás, entre esto se encuentran las interfaces y los protocolos de comunicación.

Además, se debe tener en cuenta que varias de las especificaciones con las cuales se tendrá que trabajar, se encuentran aun en fase de definición, por lo cual el mayor soporte para el desarrollo de estas temáticas son los borradores que se encuentren publicados actualmente.

- **Conocimiento claro de la arquitectura SIP para la gestión de sesiones de audio**

Se recomienda realizar un estudio detallado de la arquitectura del protocolo SIP. Los elementos que la componen, la estructura y funcionamiento del protocolo, sus capacidades y limitaciones.

Este estudio tiene como objetivo el entendimiento de cómo el protocolo SIP gestiona las sesiones, principalmente las de audio, y de cuales son los elementos principales que se encargan de la misma.

Es importante entender el alcance de SIP, ya que como se verá es un protocolo de señalización capaz de ayudar al establecimiento, modificación y terminación de sesiones multimedia, pero él por si mismo no se hizo para garantizar, por ejemplo, la existencia y distribución adecuada de recursos.

De igual manera, es importante llegar a entender las capacidades y ventajas que tiene este protocolo para la creación de aplicaciones, entre las cuales se pueden nombrar la rapidez en el desarrollo y su extensibilidad para operar en diversos escenarios.

- **Conjunción de las arquitecturas de SIP e IMS: selección de módulos comunes, distribución de servidores y servicios en los módulos.**

Una vez conocidas y entendidas las características de los componentes, y la forma de funcionamiento de las arquitecturas de IMS y SIP se debe buscar la forma en que estas dos tecnologías operan conjuntamente.

Debido a que IMS es una arquitectura de servicios y SIP es un protocolo de señalización que se eligió para trabajar dentro de IMS, se debe comprender que es esta última quien contendrá los módulos funcionales de SIP, sin embargo IMS no es una arquitectura que se basa únicamente en el protocolo SIP por lo cual sus módulos van a resultar más complejos o tendrán características adicionales a los de SIP.



Ahora bien, teóricamente es posible entender el funcionamiento de cada arquitectura por separado, pero cuando se necesita llevar a la práctica una única arquitectura es necesario seleccionar los módulos que son comunes a ambas tecnologías, y definir las características que cada uno de estos tendrán. Es muy importante definir qué módulos son los encargados de servir las funcionalidades requeridas, las cuales van desde el acceso al módulo cliente, pasando por la señalización y el transporte de datos, hasta las capacidades de IVR. En esta definición se puede encontrar módulos diferentes que cumplan aparentemente con la misma funcionalidad, en este caso la decisión depende de los estudios previos realizados. En otras ocasiones, es necesario decidir la conjunción de funcionalidades dentro de un mismo módulo, a pesar de que estas pertenezcan a módulos definidos independientemente, y esto debido a los tiempos de desarrollo estipulados.

- **Conocimiento generalizado de los protocolos a utilizar**

Resulta de vital importancia el estudio generalizado que se haga acerca de los protocolos necesarios para los procesos de señalización (SIP y SDP) y transporte de datos (RTP).

Esta recomendación se hace con el fin de que se disminuya el tiempo de desarrollo a la hora de implementar un determinado protocolo para un lenguaje de programación específico. Por el contenido de esta definición, se podrá prever la necesidad de implementar los protocolos SIP y RTP para diferentes entornos de aplicación.

- **Herramientas para el desarrollo y evaluación de módulos**

Una vez realizada la selección y definición de los módulos, se debe buscar herramientas tales como los entornos de desarrollo, los lenguajes y las APIs de programación, que pueden ser diversos, entre otros elementos, que permitan su desarrollo e implementación.

Los IDE como Eclipse son muy útiles para trabajar con diversos lenguajes, APIs y librerías de programación, y de allí su utilización. Sin embargo, existen módulos muy complejos que se salen de las capacidades y del tiempo destinado para el desarrollo del proyecto, por lo cual deben buscarse contenedores de funcionalidades que cumplan con los requisitos básicos de aquellos módulos. Este es el caso de los Servidores de Aplicaciones SIP los cuales se utilizan para soportar la lógica del servicio creada por los desarrolladores de aplicaciones, ya que de la ejecución y prestación de los servicios de interconexión con otras entidades se encarga el servidor. Además, varias de estas herramientas contenedoras se pueden integrar con el entorno de desarrollo Eclipse, y por lo tanto cuentan con las ventajas y facilidades que este brinda.

Cuando se desarrolla un proyecto de esta magnitud, en las primeras etapas de desarrollo se debe contar con herramientas y/o aplicaciones que permitan la evaluación parcial de los módulos. Para esto, deben crearse una serie de aplicaciones clientes e interfaces que permitan realizar las pruebas requeridas para una validación parcial del módulo. A través de esta metodología se puede llegar a crear módulos lo bastante independientes como para que puedan interactuar con un mayor número de aplicaciones. Teniendo en cuenta la temática de esta sección, es posible que se requiera de clientes SIP que interactúen con el servidor de aplicaciones SIP para poder realizar las pruebas de funcionalidad requeridas. Dichos clientes son diferentes al cliente móvil, y por tanto deberán ser creados con base a otras APIs de programación ya que el entorno de aplicación también cambia.



De acuerdo a lo anterior, también es posible que para realizar las pruebas en el servidor IVR (ASR y TTS) se requiera de un cliente móvil con características básicas, lo cual también es válido ya que cuando se habla de la transmisión de datos solo son pruebas correctas aquellas que se realizan teniendo en cuenta la forma de acceso y las características del usuario cliente, más aún cuando se trabaja con aplicaciones clientes que cuentan con recursos y capacidades limitadas.

- **Soluciones alternas cuando no este completa una especificación**

Dado que los procesos de comunicación entre algunos módulos, y entre los elementos que conforman dichos módulos no se han definido, es necesario plantear soluciones alternas mediante las cuales se proporcione la funcionalidad requerida. Este es el caso de la entidad MRF en la cual aún no se encuentran definidos los procesos de comunicación entre el MRFC y el MRFP para la prestación de servicios de ASR y TTS.

- **Análisis y selección de las aplicaciones que se pueden realizar, antes de iniciar su diseño e implementación. Determinación del alcance del proyecto.**

En este tipo de proyectos es necesario implementar la infraestructura a través de la cual se va a prestar el servicio, y la aplicación cliente básica que va a hacer uso de este último, para luego analizar y definir cuales son el tipo de aplicaciones que se pueden soportar.

Esta recomendación se hace debido a que antes de iniciar el estudio de las tecnologías que involucran el presente proyecto no se conoce a ciencia cierta el alcance y limitaciones que tendrán al tratar de aplicarse al entorno objetivo. En términos más específicos, se puede decir que solo hasta tener la funcionalidad de la arquitectura IMS necesaria para prestar un servicio de IVR a un cliente móvil SIP de una red móvil 2.5G, se podrán definir las aplicaciones que el cliente podrá ejecutar.

En este sentido, se puede ver que se esta haciendo una clara distinción entre lo que significa el servicio de IVR y lo que constituye las aplicaciones de usuario que harán uso de dicho servicio. Estas aplicaciones se rigen por el alcance operacional (ya que se trata de un proyecto que llega hasta la fase de simulación) del proyecto desarrollado.

### **6.3. RECOMENDACIONES PARA EL DESARROLLO DEL CLIENTE MÓVIL SIP**

- **Análisis y planteamiento de requerimientos básicos**

Antes de iniciar el desarrollo del cliente móvil se deben definir las funcionalidades básicas que este debe proporcionar, con el fin de realizar un primer acercamiento a los requerimientos específicos de la aplicación cliente. Este primer acercamiento es debido a que aún no se ha hecho un estudio profundo de las tecnologías que se van a utilizar, no obstante debe arrojar una serie de



requerimientos básicos los cuales en el transcurso de la investigación se irán desglosando o complementando.

El análisis y planteamiento de requerimientos básicos supone un estudio previo y general de las tecnologías que se implementaran en la aplicación cliente y de su alcance. Estas tecnologías corresponden al protocolo SIP, y las capacidades de audio en dispositivos móviles.

Es muy importante identificar cuales de los requerimientos analizados y planteados son obligatorios o pueden resultar opcionales en el momento de cumplir con la especificación de alguna de las tecnologías a soportar, ya que este representa un punto de referencia para la selección de las herramientas de desarrollo.

A pesar de que se trate de un primer acercamiento en la obtención de requerimientos, su resultado debe ser tal que permita realizar una selección posterior y adecuada del lenguaje de programación y/o sistema operativo. En este sentido, cualquier duda con respecto a las tecnologías que la aplicación debe soportar, tiene que ser resuelta ya que puede significar meses de trabajo archivados. Sin embargo, por no tener un conocimiento a fondo de la temática a tratar se deben correr ciertos riesgos que después se pueden convertir en retos de desarrollo.

- **Lenguajes de programación para dispositivos móviles: ventajas y limitaciones, soporte de la funcionalidad básica planteada**

Una vez establecidos los requerimientos básicos se procede a la selección del sistema operativo y/o lenguaje de programación que se van a utilizar para desarrollar el cliente móvil.

Esta selección de herramientas está muy ligada a los requerimientos básicos que se especificaron. Sin embargo, no constituye el único lineamiento a seguir en este proceso. Cuando se empieza a crear una aplicación, en general, se tienen en cuenta otros parámetros relacionados con los aspectos internos y externos del proyecto, y con las personas encargadas del desarrollo del mismo.

Entre los aspectos externos del proyecto se tienen el tiempo presupuestado para la investigación (curvas de aprendizaje), desarrollo (análisis, diseño e implementación) y documentación (proceso que se realiza la mayor parte del tiempo en paralelo con los dos procesos anteriores, pero que también soporta los procesos finales antes de la entrega del sistema) del proyecto.

Entre los aspectos internos del proyecto se tienen la temática que en si se debe soportar, lo cual se describe mediante los requerimientos básicos establecidos, y por otro lado el grupo de dispositivos objetivo en los cuales se van a desarrollar, implementar y probar las aplicaciones. En este último punto se debe tener en cuenta además de los dispositivos, el mercado objetivo, ya que de esto depende en gran medida la validez de un proyecto que tiene como uno de sus objetivos mejorar la accesibilidad para los usuarios de dispositivos móviles. En este sentido, los desarrolladores del presente proyecto se basaron en estadísticas del grado de utilización y aceptación de las tecnologías actuales para dispositivos móviles, y la disponibilidad de las herramientas de desarrollo dirigidas hacia esas tecnologías. Como resultado de esto se optó por dispositivos con capacidades limitadas como lo son la mayoría de dispositivos móviles que se consumen actualmente por usuarios de redes de 2.5G.



Por último, entre los aspectos de las personas encargadas del desarrollo del proyecto se tienen los parámetros que permiten clasificar su conocimiento y experiencia en diversos lenguajes y herramientas de programación, en este caso, para dispositivos móviles. Este tema está muy relacionado con los aspectos externos del proyecto.

De acuerdo a lo anterior, también es necesario definir los aspectos descritos, antes de seleccionar las herramientas de programación a utilizar. Claro está, si dado el caso, y luego de un estudio riguroso de las características y capacidades de las herramientas de programación disponibles, solo un grupo de estas permitieran el desarrollo de la temática, los aspectos de los que se habla no serían relevantes, y los desarrolladores deberían acogerse a su utilización. Sin embargo, existe otra opción, y es la de arriesgarse a la selección de una herramienta (lenguaje y/o sistema operativo) que actualmente no maneje las tecnologías que se necesitan, pero la cual permita la creación de nuevas APIs para el soporte de nuevas aplicaciones.

- **Alcance de implementación del prototipo: simulación o implementación real.**

Una vez seleccionadas las herramientas de programación del cliente móvil, se puede pasar a definir el alcance de la implementación del prototipo. En este punto se deben analizar los recursos con los que se cuenta para el desarrollo, implementación y ejecución, en este caso, del cliente móvil.

Cuando se habla de entornos de desarrollo para dispositivos móviles, estos además de permitir la simulación, permiten la implementación de las aplicaciones en el dispositivo real. Sin embargo, se debe analizar si se cuenta con los recursos HW y software necesarios para llegar a la ejecución de todas funcionalidades creadas, en un dispositivo y entorno reales.

En el entorno que involucra directamente al cliente móvil existen requerimientos principalmente de HW para su ejecución de forma real. El requerimiento más importante y con el que no se pudo contar, ya que se sale de los objetivos del proyecto, hace referencia a la infraestructura de radio de una red 2.5G. En lo que a esto respecta, es necesario aclarar que no se podía plantear la utilización de un acceso radio alternativo, como por ejemplo el acceso a través de redes WiFi, y esto debido a que el proyecto está enmarcado dentro de una arquitectura que plantea y especifica procedimientos a seguir según el tipo de acceso, teniendo en cuenta además que varias de esas especificaciones se encuentran en proceso de estudio y definición.

Reconocer que no se puede llegar a la consecución del requerimiento mencionado, hizo decidir que el proyecto tendría como alcance la fase de simulación del prototipo a crearse. Por otro lado, se puede decir que la decisión de este tipo de cosas hace que la investigación y el desarrollo se enfoquen en los requerimientos reales del proyecto.

- **Conocimiento generalizado de los protocolos a aplicar**

Cuando se trabaja con un tipo de proyecto como el presente, en el cual una misma tecnología debe ser aplicada a diversos entornos es necesario realizar un estudio de dicha tecnología a nivel general. En este caso, con relación al desarrollo del cliente móvil, se debe realizar el estudio de los protocolos SIP y RTP, y de los tipos de formatos de los archivos de audio.



El estudio generalizado de dichas tecnologías, permitirá que cuando se realicen los estudios más específicos de las mismas, aplicadas a los entornos dentro de los cuales deberán enmarcarse, se comprenda la forma como deben modificarse ciertas funciones y procedimientos y, sin embargo, se siga cumpliendo con su filosofía general.

Esta recomendación se hace con el fin de seguir una buena metodología a la hora de abordar una tecnología que será aplicada en áreas con diversas características, donde la mejor forma de comprender un tema es ir de lo general a lo específico. Con esto también se logra la optimización de los tiempos de estudio y de desarrollo de funcionalidades.

En este caso el escenario de aplicación es el de los dispositivos móviles, y, como se sabe, éste tiene limitaciones para el desarrollo de funcionalidades muy complejas. Es por esto, que una vez se comprenda la funcionalidad básica de SIP y RTP, su aplicación particular para este tipo de entornos será realizada más fácilmente. Dicha aplicación puede simplificarse cuando se cuenta con la existencia de APIs que soportan el protocolo para el entorno en el que se está trabajando, sin embargo se puede dar el caso de que no exista el API o que no se encuentren desarrollos que se aproximen a lo que se busca, en el lenguaje de programación que se eligió (de ahí la importancia en la elección de un lenguaje de programación y/o sistema operativo adecuados). Esto último ocurrió con la implementación del protocolo RTP para J2ME, y en este caso se debieron crear las clases que dieran soporte a esta funcionalidad.

Por otro lado, el estudio general de los tipos de formato de audio que soportan los dispositivos móviles, permite, además de conocer sus características, prever que por las limitaciones del dispositivo móvil es posible que se necesite realizar modificaciones del tipo de formato cuando se cambie el escenario de aplicación. Más específicamente, es conveniente investigar sobre la estructura de archivos de audio WAV, puesto que el proceso de captura de audio en ocasiones se realiza grabando archivos de este tipo. Además, se debe tener presente la necesidad de buscar soluciones software que realicen los cambios de formato que se puedan ir requiriendo. En este punto, también se debe entrar a analizar en qué escenario de ejecución se pueden llevar a cabo dichos cambios, es decir, si se realizan a nivel del entorno de red, o si se pueden realizar en el dispositivo móvil.

Si se utiliza J2ME para desarrollar el cliente móvil, y, por ejemplo, el tipo de audio que el MMAPI (el API multimedia para J2ME) va a reproducir son archivos WAV almacenados, entonces la sugerencia es que el sistema IVR envíe paquetes RTP convencionales y el móvil se encargue de recibirlos y armar los archivos WAV para luego reproducirlos, de esta forma se logra un mayor grado de interoperabilidad.

- **Si se está simulando, adecuar los parámetros de simulación a las características reales de implementación. Conocimiento de las características del entorno de aplicación real.**

Para lograr un mayor acercamiento entre la fase de implementación y la de ejecución en entornos y dispositivos reales, se hace necesario conocer las características de mayor relevancia que se especifican en una determinada tecnología.

En este momento se empieza a necesitar de un estudio más profundo de las tecnologías que se deben soportar en el dispositivo, con el fin de realizar una selección adecuada de las



características que se deben tener en cuenta a la hora de realizar el análisis, diseño e implementación de la aplicación que va a resolver una determinada funcionalidad.

Por otro lado, se debe analizar si es posible implementar las características encontradas con las herramientas de programación seleccionadas, y cual es el impacto de éstas en el desarrollo de una aplicación.

Las características más importantes que se deben tener en cuenta para el desarrollo de clientes móviles SIP con capacidades IVR son: capacidad del enlace transmisión de datos, codecs de audio definidos en las especificaciones, flujo de señalización y flujo de datos.

La capacidad del enlace de transmisión hace que se tengan consideraciones en el tamaño de los datos a transmitir, y en este sentido se debe definir la forma como se interactuará con el cliente para poder cumplir con este requisito. Para ello se requiere de una adecuada manipulación del API para el manejo del audio en el dispositivo, y de la utilización de procedimientos efectivos en el desarrollo de sistemas accesibles a los usuarios. Por otro lado, el codec de audio también tiene un papel muy importante en la manipulación del tamaño de los datos, y se deberá decidir si se entra a gestionar este parámetro. Adicionalmente, existen procedimientos planteados para la reducción del tamaño de los datos, y estos van dirigidos hacia la reducción de las cabeceras, ya que el paquete IP final consiste en información redundante que se lleva a través de las diferentes cabeceras. De igual forma, se debe decidir si se entra a gestionar la reducción de cabeceras, y para ello es necesario investigar la existencia de mecanismos estandarizados que lo hagan.

Cuando se trabaja en base a tecnologías que se encuentran estandarizadas, se deben seguir las reglas definidas para su implementación adecuada. Por lo tanto, para el flujo de señalización se deben conocer los mensajes SIP, sus requisitos, la forma de crearse, y el orden en que deben enviarse, según la función que se este ejecutando. Esto mismo se necesita para el flujo de datos, conocer la forma de implementar el protocolo RTP. Lo anterior con el fin de lograr una integración satisfactoria del módulo del cliente móvil con los demás módulos que hacen parte de la infraestructura de red.

- **Herramientas de simulación, y entornos de desarrollo. Herramientas más efectivas.**

Una vez se seleccione el lenguaje y/o sistema operativo con el cual se va a trabajar, se debe determinar cuales herramientas y entornos de desarrollo están disponibles para su utilización, y por otro lado también es importante conocer la accesibilidad de las mismas.

En este sentido, existen entornos de desarrollo y herramientas de simulación para dispositivos móviles soportados por empresas muy comprometidas con sus productos, que ofrecen variadas prestaciones y son de acceso gratuito. Por tal motivo, estos entornos fueron los seleccionados, y entre ellos se pueden nombrar el IDE de Eclipse, Carbide J, WTK2.5.

De acuerdo a las herramientas seleccionadas, se puede decir que se cuenta con un entorno de desarrollo como Eclipse que permite la creación de aplicaciones con tecnologías muy diversas, lo cual genera un aprovechamiento eficiente del tiempo y de los recursos software, ya que se puede acceder a diversas herramientas de desarrollo a través de una interfaz única.

Por medio del Carbide J, el cual se puede agregar como herramienta de Eclipse, se puede trabajar con una serie de SDK dirigidos al desarrollo de aplicaciones para dispositivos móviles. Cada SDK



se caracteriza por estar definido para una determinada familia de dispositivos y cuenta con su propio simulador. En general, los SDKs permiten la creación de aplicaciones, empaquetamiento y simulación de las mismas.

Por otro lado, cuando en el desarrollo de un proyecto como este no se cuenta con la cantidad de computadores necesarios para la creación de una red PC-Red y Servicios-PC, por un lado, la ejecución de varios simuladores a través de mismo entorno de desarrollo no es tan eficiente debido a la demanda de recursos de procesamiento que se presenta, y por otro lado se presentan conflictos con las direcciones IP. Es por esto, que se decidió utilizar otra herramienta para la simulación de las aplicaciones creadas, y esta es el WTK en su versión 2.5. La principal característica de esta herramienta, en dicha versión, es que da soporte a un mayor número de APIs entre las cuales se tiene el API de SIP para J2ME.

Con lo anterior, se quiere dar a entender que es una buena práctica la utilización de herramientas alternativas debido a que se suele perder tiempo muy valioso en la búsqueda de posibles errores en el código de las aplicaciones, y solo es hasta cuando se realizan pruebas con otras herramientas que se descubren las limitaciones de ejecución de las primeras con las que se trabajó. En este sentido, se puede nombrar, por ejemplo, la utilización del WTK 2.5 versión Beta como herramienta de simulación, en lugar de los simuladores que proporciona el Carbide J, y esto debido, entre otras cosas, a características de desempeño como el uso de memoria, tiempos de lanzamiento, y usabilidad.

- **Análisis, diseño, implementación y pruebas del módulo cliente: prever posterior integración con el sistema**

Una vez realizado el análisis de los requerimientos del módulo cliente, seleccionado el lenguaje y las herramientas de programación, se pueden realizar el diseño y la implementación de las funcionalidades con las que contará esta entidad.

Este módulo debe desarrollarse de tal forma que permita su evaluación parcial y de manera individual, ya que por cuestiones de tiempo no se puede asegurar la existencia y funcionamiento del resto del sistema, por lo menos en las primeras etapas de desarrollo. Lo anterior trae consigo la ventaja de crear un sistema modular, cuyos componentes se pueden agregar fácilmente a otros sistemas que necesiten funcionalidades similares. De hecho, esta constituye una de las filosofías de desarrollo del presente trabajo.

Es por esto que las pruebas de los flujos de señalización se realizaron utilizando dos clientes móviles con capacidades SIP y un servidor SIP proporcionado por uno de los SDKs que se instalaron. Por otro lado el flujo de datos también se probó utilizando dos clientes móviles con soporte para RTP.

Las pruebas de validación de la interacción del cliente móvil con el módulo IVR, requieren lógicamente de la existencia de dicha entidad. Y en este caso no existe un procedimiento de validación parcial en una etapa temprana de desarrollo. Sin embargo, se deben tener en cuenta las recomendaciones que se han realizado hasta el momento, para que la futura integración del sistema sea rápida y el resultado cumpla, en este caso, con los objetivos planteados para el módulo del cliente móvil.



- **Reproducción de audio en el cliente móvil**

Es necesario garantizar un grado de continuidad en la reproducción de audio en el cliente móvil, por tanto se debe realizar la sincronización entre la llegada de los paquetes RTP y la reproducción del audio (es decir cuando se termina la reproducción de un archivo y se debe realizar la del siguiente). El MMAPI tiene la clase *Player*, la cual a través de un método informa cuando se termina de reproducir un archivo, sin embargo, este método no es tan confiable puesto que el audio se escucha cortado. Por ende, es recomendable utilizar un temporizador que de acuerdo con la longitud del paquete RTP calcule la duración del archivo WAV y permita enlazar varios archivos de manera que el audio sea reproducido de forma más natural.



## 7. CONCLUSIONES Y TRABAJOS FUTUROS

### 7.1. CONCLUSIONES

- Es posible desarrollar un cliente móvil SIP en el lenguaje J2ME y donde la transmisión de datos de audio se base en el protocolo RTP.
- La utilización de protocolos estandarizados, como SIP y RTP, constituye una de las herramientas más importantes en el desarrollo de aplicaciones en donde estos protocolos tengan lugar, ya que el diseño de los sistemas se puede modularizar a tal grado de obtener componentes individualmente funcionales y fácilmente integrables con otros componentes que manejen el mismo protocolo, y casi independientes del lenguaje de programación en el que se implementaron.
- El conocimiento individual de las arquitecturas de IMS y SIP, en el desarrollo de sistemas que soporten servicios multimedia, permite el diseño e implementación de módulos que soporten las funcionalidades de ambas arquitecturas. En estos procesos se puede dar el caso de encontrar entidades con características comunes para ambas arquitecturas o que deban agregarse para cumplir con los objetivos de operación de cualquiera de ellas. Además, también es posible reunir en un módulo funciones lógicas que hagan parte de otros elementos de la infraestructura del sistema, y esto debido a que la división entre los servidores de ambas arquitecturas no es física, sino conceptual.
- Es posible el desarrollo de aplicaciones de 3G adaptadas a las capacidades de despliegue de las redes de 2.5G. El hecho de crear una aplicación que genera un primer avance en la experimentación de la transmisión de la voz a través de una red de paquetes tanto en la capa de acceso como la de transporte constituye una buena manera de validar las tecnologías que existen actualmente, ya que son pocos los requerimientos que este tipo de desarrollos necesitan y por lo tanto permiten la realización de pruebas, y recomendaciones con respecto a las tecnologías que se han especificado.
- De acuerdo al alcance de este proyecto no fue necesario utilizar las cabeceras SIP para IMS, y basto con el protocolo SIP básico para la obtención de la funcionalidad requerida.
- Actualmente no se cuenta con un buen soporte para aplicaciones ASR (para el idioma español), todavía se necesitan mejoras para poder aplicar las recomendaciones propuestas de accesibilidad de IVR.
- J2ME requiere de mejoras para proveer los servicios que se esperan con ayuda del protocolo SIP.
- Debido a que los terminales móviles no tienen una capacidad de procesamiento y herramientas software comparable a las de un computador, VoIP sobre J2ME todavía no es posible, aunque PoC es una buena aproximación.



## 7.2. TRABAJOS FUTUROS

- Aprovechando las capacidades ASR, un trabajo a largo plazo consiste en implementar mensajería unificada para terminales móviles, de esa manera se explotarían los servicios que se pueden establecer a través de SIP, como VoIP y Mensajería Instantánea.
- Dirigiendo el desarrollo hacia los motores ASR, un trabajo futuro podría consistir en el desarrollo de un Sistema de Reconocimiento Automático del habla para el lenguaje español y de código libre, debido a que los costos de los productos actuales de reconocimiento para este idioma son muy elevados. Este es un trabajo extenso pues abarca áreas tales como gramática, prosodia, etc., sin embargo es muy útil para desarrollar aplicaciones IVR que tienen un alto potencial de crecimiento.
- Realizar la implementación del protocolo DIAMETER para la comunicación entre el HSS y los servidores CSCF.
- Realizar la implementación del protocolo H.248 para la comunicación entre el MRFC y el MRFP.
- Para la creación de las secuencias de interacción del usuario con el sistema IVR sería muy interesante realizar esta implementación utilizando VoiceXML, debido a las capacidades de árboles de decisión que esta brinda.
- Realizar la implementación del algoritmo de autorización/autenticación IMS-AKA.
- Realizar la implementación del cliente móvil para el Sistema Operativo Symbian, y de esa forma aprovechar el mayor número de capacidades que proporciona este entorno de ejecución de aplicaciones para dispositivos móviles.
- Implementación del codec AMR en el dispositivo móvil.
- Realizar el proceso de señalización a través de la red IMS no con el protocolo SIP básico como fue el que se utilizó en el presente trabajo sino con el protocolo SIP con las cabeceras adicionales por el 3GPP para soportar capacidades de la red IMS.
- En relación con el anterior trabajo, también puede surgir la idea de crear una pasarela que convierta los mensajes del protocolo básico SIP al protocolo SIP para IMS, y viceversa.



## GLOSARIO

<b>2.5G</b>	Generación 2.5 de sistemas de comunicaciones móviles
<b>3G</b>	Tercera generación de sistemas de comunicaciones móviles
<b>3GPP</b>	Thrid Generation Partnership Project, Proyecto Conjunto de Tercera Generación
<b>3GPP2</b>	Thrid Generation Partnership Project 2, Proyecto Conjunto de Tercera Generación 2
<b>AAA</b>	Authentication, Authorization and Accounting, Autenticación, Autorización y Tasación
<b>ADPCM</b>	Adaptive Differential Pulse Code Modulation, Modulación por Impulsos Codificados Diferencial y Adaptativa
<b>ADSL</b>	Asymmetric Digital Subscriber Line, Línea de Abonado Digital Asimétrica
<b>AMR</b>	Adaptive Multi Rate, Codificador de voz de multi-tasa adaptativa. Codifica y decodifica las muestras de habla digitalmente.
<b>API</b>	Application Programming Interface, Interfaz de Programación de Aplicaciones
<b>AS</b>	Application Server, Servidor de Aplicaciones
<b>ASR</b>	Automatic Speech Recognition, Reconocimiento Automático de Habla
<b>ASSI</b>	Application Server Subscription Information, Información de Suscripción en el Servidor de Aplicaciones
<b>Back-to-back</b>	Una entidad que puede tener dos roles como el de cliente y el de servidor
<b>BGCF</b>	Breakout Gateway Control Function, Función de control de la Pasarela de Fuga
<b>BSD</b>	Berkeley Software Distribution, Distribución de Software Berkeley
<b>BSS</b>	Base Station Subsystem, Subsistema de la Estación Base
<b>CAMEL</b>	Customized Applications for Mobile Networks Enhanced Logic, Aplicaciones Personalizadas para Redes Móviles de Lógica Mejorada
<b>CAP</b>	CAMEL Application Part, Parte de la Aplicación CAMEL
<b>CDMA2000</b>	Code Division Multiple Access 2000, una tecnología 3G
<b>CDR</b>	Call Detailed Record, Registro Detallado de Llamada
<b>CELP</b>	Codificación que se basa en procedimientos de búsqueda de análisis-por-síntesis, cuantización de vectores con pesos y predicción lineal.
<b>CGI</b>	Common Gateway Interface, Interfaz Común de Pasarela
<b>CLI</b>	Command Line Interface, Interfaz por Línea de Comandos
<b>CN</b>	Core Network, Núcleo de Red
<b>Codec</b>	Dispositivo codificador-decodificador
<b>COPS</b>	Common Open Policy Service, Servicio de Políticas Común y Abierto
<b>CPL</b>	Call Processing Language, Lenguaje de Procesamiento de Llamada
<b>CS</b>	Circuit Switching, Conmutación de Circuitos
<b>CSCF</b>	Call Session Control Function, Función de Control de la Sesión de Llamada
<b>DiffServ</b>	Differentiated Services Internet QoS model, Modelo de Calidad de Servicio en Internet basado en Servicios Diferenciados
<b>Dominio IM</b>	Dominio Multimedia IP
<b>DSR</b>	Distributed Speech Recognition, Reconocimiento de voz distribuido
<b>DTMF</b>	Dual Tone Multi Frequency, Tono Dual Multifrecuencia
<b>DTX</b>	Discontinuous Transmisión, Transmisión no Continúa
<b>E.164</b>	Recomendación que describe el plan internacional de numeración telefónica
<b>EDGE</b>	Enhanced Data Rates for Global Evolution, Tasas de Datos Mejorados para la Evolución Global
<b>FQDN</b>	Fully-Qualified Domain Name, Nombre de Dominio Totalmente Cualificado
<b>Framework</b>	Aplicación especializada



<b>G.711</b>	Técnica de codificación de voz PCM de 64Kbps
<b>GERAN</b>	GPRS EDGE Radio Access, Acceso Radio GPRS EDGE
<b>GGSN</b>	Gateway GPRS Support Node, Nodo de Soporte de la Pasarela GPRS
<b>GPRS</b>	General Packet Radio Service, Servicio General de Paquetes de Radio
<b>GSL</b>	Grammar Specific Language, Lenguaje Gramatical Específico
<b>GSM</b>	Global System for Mobile Communications, Sistema Global para Comunicaciones Móviles
<b>H.248</b>	También conocido como protocolo Megaco
<b>H.323</b>	Estándar de la ITU-T para voz y videoconferencia interactiva en tiempo real, para redes de Área Local e Internet
<b>HSS</b>	Home Subscriber Server, Servidor Local del Suscriptor
<b>HTTP</b>	Hypertext Transfer Protocol, Protocolo de Transferencia de Hipertexto
<b>I-CSCF</b>	CSCF de Interrogación o "Interrogating"
<b>IETF</b>	Internet Engineering Task Force, Fuerza de Trabajo de Ingeniería para Internet
<b>IM</b>	Instant Messaging, Mensajería Instantánea
<b>IMS-AKA</b>	IMS Authentication and Key Agreement, Clave de Acuerdo y Autenticación IMS
<b>IM-SSF</b>	IP Multimedia - Services Switching Function, Función de Conmutación de Servicios - Multimedia IP
<b>IM-SSP</b>	IP Multimedia - Service Switching Point, Punto de Conmutación de Servicios - Multimedia IP
<b>IMS</b>	IP Multimedia Subsystem, Subsistema Multimedia IP
<b>IM-MGW</b>	IP Multi-Media Gateway, Pasarela de Medios Multimedia IP
<b>IP</b>	Internet Protocol, Protocolo Internet
<b>IPv4</b>	Versión 4 del Protocolo Internet
<b>IPv6</b>	Versión 6 del Protocolo Internet
<b>ISC</b>	IP multimedia Service Control, Control del Servicio Multimedia IP
<b>ISDN</b>	Integrated Services Data Network, Red Digital de Servicios Integrados
<b>ISIM</b>	Aplicación lógica que reside en una tarjeta de circuitos integrados
<b>ISUP</b>	ISDN User Part, Parte de Usuario ISDN
<b>IVR</b>	Interactive Voice Response, Respuesta de Voz Interactiva
<b>JAIN</b>	Java APIs for Integrated Networks, APIs de Java para Redes Integradas
<b>JDBC</b>	Java DataBase Connectivity, Conectividad Java a Bases de Datos
<b>JVM</b>	Java Virtual Machine, Máquina Virtual de Java
<b>MAP</b>	Mobile Application Part, Parte de Aplicación Móvil
<b>Mb</b>	Punto de referencia para servicios en redes IPv6
<b>MEGACO</b>	Media Gateway Control, Control de Interfaces de Medios
<b>MGCF</b>	Media Gateway Control Function, Función de Control de la Pasarela de Medios
<b>MGW</b>	Media Gateway, Pasarela de Medios
<b>MIDP</b>	Mobile Information Device Profile. Conjunto de APIs de Java que define el modo en que las aplicaciones de software interactúan con los teléfonos celulares
<b>MIME</b>	Multipurpose Internet Mail Extensions, Extensiones Multipropósito del Correo Electrónico de Internet
<b>Mp</b>	Punto de referencia entre MRFC y MRFP
<b>MP-MLQ</b>	Modo de compresión de VoIP
<b>Mr</b>	Punto de referencia entre CSCF y MRFC
<b>MRCP</b>	Media Resource Control Protocol, Protocolo de Control de los Recursos Multimedia
<b>MRF</b>	Media Resource Function, Función de Recursos Multimedia
<b>MRFC</b>	Media Resource Function Controller, Controlador de la Función de Recursos Multimedia
<b>MRFP</b>	Media Resource Function Processor, Procesador de la Función de Recursos Multimedia
<b>Multicast</b>	Multidifusión



<b>NB-AMR</b>	Narrow Band – Adaptive Multi Rate, AMR para Banda Estrecha
<b>OSA</b>	Open Services Architecture, Arquitectura de Servicios Abiertos
<b>OSA-GW</b>	Open Services Architecture - Gateway, Pasarela de la Arquitectura de Servicios Abiertos
<b>PCM</b>	Pulse Code Modulation, Modulación por Impulsos Codificados
<b>PDP</b>	Packet Data Protocol, Protocolo de Paquetes de Datos
<b>Perl</b>	Practical Extraction and Reporting Language, Lenguaje de Reporte y Extracción Práctica
<b>P-CSCF</b>	Proxy-CSCF, Proxy Controlador de la Función de Sesión de Llamada
<b>Plug-in</b>	Es un módulo opcional que puede ser agregado a un programa
<b>PTT</b>	Push To Talk, Presionar Para Hablar
<b>PoC</b>	PTT para celulares
<b>PSTN</b>	Public Switched Telephone Network, Red Telefónica Pública Conmutada
<b>QoS</b>	Quality of Service, Calidad de Servicio
<b>Radius</b>	Remote Authentication Dialing User Service, Protocolo de Autenticación y autorización para aplicaciones de acceso a la red
<b>Release</b>	Versión o Edición
<b>RFC</b>	Request For Comments, Petición de Comentarios. Conjunto de notas técnicas y organizativas donde se describen los estándares o recomendaciones de Internet
<b>RSVP</b>	Resource Reservation Protocol, Protocolo de Reservación de Recursos
<b>RTC</b>	Red Telefónica Conmutada
<b>RTP</b>	Realtime Transport Protocol, Protocolo de Transporte en Tiempo Real
<b>RTCP</b>	Real Time Control Protocol, Protocolo de Control en Tiempo Real
<b>RTSP</b>	Real-Time Streaming Protocol, Protocolo de Transmisión Continua en Tiempo Real
<b>Script</b>	Tipo de programa que consiste de una serie de instrucciones que serán utilizadas por otra aplicación
<b>SCS</b>	Service Capability Server, Servidor de Capacidad de Servicio
<b>S-CSCF</b>	CSCF de Servicio o “Serving”
<b>SCTP</b>	Stream Control Transmission Protocol, Protocolo de Control del Flujo de la Transmisión
<b>SDP</b>	Session Describing Protocol, Protocolo de Descripción de Sesión
<b>SGSN</b>	Server GPRS Support Node, Nodo de Soporte del Servidor GPRS
<b>SIGTRAN</b>	Signalling Transport, Protocolo de Transporte de Señalización
<b>SIMPLE</b>	SIP for Instant Messaging and Presence Leveraging Extensions, SIP para Mensajería Instantánea y Extensiones para el servicio de Presencia
<b>SIP</b>	Session Initiation Protocol, Protocolo de Inicio de Sesión
<b>SNMP</b>	Simple Network Management Protocol, Protocolo Simple de Administración de Red
<b>SS7</b>	Signaling System 7, Sistema de Señalización No.7
<b>TAS</b>	Telephony Application Server, Servidor de Aplicaciones de Telefonía
<b>TCAP</b>	Transaction Capabilities Application Part, Parte de Aplicación de las Capacidades de Transacción
<b>TCP</b>	Transmission Control Protocol, Protocolo de Control de la Transmisión
<b>TDM</b>	Time Division Multiplexing, Multiplexación por División de Tiempo
<b>TFO</b>	Tandem Free Operation, Operación Libre de Tandem
<b>T-SGW</b>	Trunking Signalling Gateway, Pasarela de Señalización Troncal
<b>TTS</b>	Text To Speech, Conversión de Texto a Habla
<b>UA</b>	User Agent, Agente de Usuario
<b>UAC</b>	User Agent Client, Agente de Usuario Cliente
<b>UAS</b>	User Agent Server, Agente de Usuario Servidor
<b>UCD</b>	User Centred Design, Diseño Centrado en el Usuario
<b>UDP</b>	User Datagram Protocol, Protocolo de Datagramas de Usuario



<b>UE</b>	User Equipment, Equipo de Usuario
<b>UIT</b>	Unión Internacional de Telecomunicaciones
<b>UMTS</b>	Universal Mobile Telecommunications System, Sistema de Telecomunicaciones Móviles Universales
<b>Unicast</b>	Unidifusión
<b>URI</b>	Uniform Resource Identifier, Identificador Uniforme de Recursos
<b>UTRAN</b>	UMTS Terrestrial Radio Access Network, Red Terrestre de Acceso Radio UMTS
<b>VoIP</b>	Voice over IP, Voz sobre IP
<b>WB-AMR</b>	Wide Band - Adaptive Multi Rate, AMR para Banda Ancha
<b>WCDMA</b>	Wideband CDMA, CDMA de Banda Ancha
<b>WIMAX</b>	Worldwide Interoperability for Microwave Access, Interoperabilidad Mundial para Acceso por Microondas
<b>WLAN</b>	Wireless Local Area Network, Red Inalámbrica de Área Local
<b>XML</b>	eXtensible Markup Language, Lenguaje de Mercado Extensible



## BIBLIOGRAFÍA

- [1] **Diego López de Ipiña, Amparo Rodríguez e Iñaki Vázquez.** “Accesibilidad para discapacitados a través de teléfonos y servicios móviles adaptables”. [En línea]. Abril 2005. Disponible en web: <<http://paginaspersonales.deusto.es/ivazquez/ponencias/VazquezJANT2004.pdf>> [Consulta: Abril 14 de 2005]
- [2] **Toni Granollers i Saltiveri.** “MPLu+a. Una Metodología que integra la Ingeniería del Software, la interacción persona-ordenador y la Accesibilidad en el contexto de Equipos de Desarrollo Multidisciplinares”. Departament de Llenguatges i Sistemes Informàtics, Universitat de Lleida. [En línea]. Julio 2004. Disponible en web: <[http://griho.udl.es/publicacions/2004/Tesis\\_Toni/TesiToniGranollers.pdf](http://griho.udl.es/publicacions/2004/Tesis_Toni/TesiToniGranollers.pdf)> [Consulta: Abril de 2005]
- [3] **D. Luis Villarrubia Grande.** “Eliminación de barreras mediante la tecnología del habla”. Telefónica Investigación y Desarrollo, S.A. Unipersonal. [En línea]. Agosto 2004. Disponible en web: <<http://paginaspersonales.deusto.es/ivazquez/ponencias/VazquezJANT2004.pdf>> [Consulta: Agosto de 2006]
- [4] **Telefónica I+D.** “Las Telecomunicaciones y la Movilidad en la Sociedad de la Información”. [En línea]. Febrero 2005. Disponible en web: <<http://www.telefonica.es/sociedaddelainformacion/pdf/publicaciones/movilidad/telecoymovilidad.pdf>> [Consulta: Abril de 2006]
- [5] **Lucent Technologies.** “IP Multimedia Subsystem (IMS) Service Architecture”. [En línea]. Febrero 2005. Disponible en web: <[http://www.lucent.com/livelink/090094038005df2f\\_White\\_paper.pdf](http://www.lucent.com/livelink/090094038005df2f_White_paper.pdf)> [Consulta: Septiembre de 2005]
- [6] **Luis Miralles Pechuán.** “El nuevo sistema multimedia conocido como IMS que adoptarán las redes UMTS”. [En línea]. Trabajo presentado en la asignatura Redes de Ordenadores, Curso 2004-2005, Universidad de Valencia. Disponible en web: <<http://www.uv.es/~montanan/redes/trabajos/IMS.doc>> [Consulta: Octubre de 2005]
- [7] **Pedro José Camacho Ojeda, Isabel Cristina Hernández Pino.** “Sistema para la gestión de servicios de Red Inteligente soportados en Internet”. Trabajo de grado presentado como requisito para optar al título de Ingeniero en Electrónica y Telecomunicaciones, Universidad del Cauca. 2005. [Consulta: Enero de 2006]
- [8] **Gloria Carolina Benavides Cabrera, Maryury Alexandra Muñoz Burbano.** “Diseño de un Módulo de Control de Sesiones de Usuario basado en la Arquitectura de servicios IMS para el despliegue de aplicaciones y/o servicios en redes de Telefonía Móvil”. Trabajo de grado presentado como requisito para optar al título de Ingeniero en Electrónica y Telecomunicaciones, Universidad del Cauca. 2006. [Consulta: Octubre de 2006]
- [9] **Alberto Guirao Villalonga, Toni Oller Arcas.** “Implementación de un servidor proxy SIP en JAVA”. [En línea]. Trabajo de fin de carrera, Universidad de Cataluña. Julio de 2006. Disponible en web: <<http://biblioteca.upc.es/PFC/arxiu/migrats/53972-1.pdf>> [Consulta: Noviembre de 2006]



- [10] **Expocomm Argentina 2005.** *Seminario de Tecnología y Mercado*. [En línea]. Septiembre 2005. Disponible en web: <[http://www.ejkreed.com/extras/img/web\\_727\\_brochurestm.pdf](http://www.ejkreed.com/extras/img/web_727_brochurestm.pdf)> [Consulta: Abril 14 de 2005]
- [11] **Anselmo Carrascosa.** *"Nuevos Servicios Multimedia, Convergencia y Evolución con IMS"*. [En línea]. Agosto-Septiembre 2005. Disponible en web: <<http://www.coit.es/publicaciones/bit/bit152/84-87.pdf>> [Consulta: Enero de 2006]
- [12] **3G Americas.** *"Gateway IP Multimedia Subsystem (IMS) para nuevos e innovadores servicios"*. [En línea]. Julio 30 de 2004. Disponible en web: <[http://www.3gamericas.org/Spanish/News\\_Room/DisplayPressRelease.cfm?id=886&s=SPN](http://www.3gamericas.org/Spanish/News_Room/DisplayPressRelease.cfm?id=886&s=SPN)> [Consulta: Enero de 2006]
- [13] **Simon ZNATY.** *"NGN: Red de Nueva Generación"*. Agosto 2006. Disponible en web: <<http://www.efort.com>> [Consulta: Agosto de 2006]
- [14] **Flávio Tonioli Mariotto.** *"Tendencias en Convergencia Móvil"*. [En línea]. Disponible en web: <<http://www.comunidadmovil.com.co/media/Flavio%20Mariotto2.pdf>> [Consulta: Abril de 2006]
- [15] **Elena Romero.** *"Evolución de las infraestructuras hacia un nuevo entorno. El estándar IMS como elemento clave"*. Customer Solutions & System Integration, Ericsson. [En línea]. Mayo 2005. Disponible en web: <[http://www.ericsson.com/es/documentos/estandar\\_ims.pdf](http://www.ericsson.com/es/documentos/estandar_ims.pdf)> [Consulta: Septiembre de 2005]
- [16] **Manuel Moreno Martín, Manuel Álvarez-Campana Fernández-Corredor, Joan Vinyes Sanz.** *"Propuesta de utilización de SIP como protocolo de señalización en la red de acceso radio de sistemas UMTS"*. [En línea]. Disponible en web: <<http://www.ahciet.net/comun/portales/1000/10002/10007/10299/docs/011-72-80.pdf>> [Consulta: Abril de 2006]
- [17] **3GNewsroom.** *"UMTS ready to meet demands for high speed wireless data services"*. [En línea]. Julio 2006. Disponible en web: <[http://www.3gnewsroom.com/3g\\_news/jul\\_06/news\\_7095.shtml](http://www.3gnewsroom.com/3g_news/jul_06/news_7095.shtml)> [Consulta: Agosto de 2006]
- [18] **María Victoria de Diego Bartolomé, Diego Gallego Pérez, José Antonio López Mora, Alberto Gómez Vicente.** *"UMTS: hacia una red todo IP"*. [En línea]. Enero 2002. Disponible en web: <<http://bibliotecavirtual.clacso.org.ar/ar/libros/peru/grade/arregui/art4.pdf>> [Consulta: Abril de 2006]
- [19] **SIP.** [En línea]. Disponible en web: <<http://www.rfc-editor.org/rfc/rfc3261.txt>>
- [20] **IPv6.** [En línea]. Disponible en web: <<http://www.rfc-editor.org/rfc/rfc791.txt>>
- [21] **RTP.** [En línea]. Disponible en web: <<http://www.rfc-editor.org/rfc/rfc1889.txt>>
- [22] **RTCP.** [En línea]. Disponible en web: <<http://www.rfc-editor.org/rfc/rfc3605.txt>>
- [23] **COPS.** [En línea]. Disponible en web: <<http://www.rfc-editor.org/rfc/rfc2748.txt>>
- [24] **RSVP.** [En línea]. Disponible en web: <<http://www.rfc-editor.org/rfc/rfc2205.txt>>
- [25] **DiffServ.** [En línea]. Disponible en web: <<http://www.rfc-editor.org/rfc/rfc2475.txt>>



- [26] **MEGACO**. [En línea]. Disponible en web: <<http://tools.ietf.org/html/rfc3525>>
- [27] **Diego Andrés Acosta**. “SIP Session Initiation Protocol”. [En línea]. Julio 2002. Disponible en web: <<http://greco.dit.upm.es/~david/TAR/trabajos2002/01-SIP-%20Diego-Acosta.pdf>> [Consulta: Abril de 2006]
- [28] **CIENTEC**. “SIP: hacia la era de las comunicaciones interactivas”. [En línea]. Disponible en web: <<http://www.cientec.com/tendencias/tendencias28.asp>> [Consulta: Septiembre de 2005]
- [29] **Mariano Stokle, NORTEL**. “This is the way (SIP Tutorial V3)”. Disponible en pdf: <Session Initiation Protocol (SIP) Tutorial - Slides.pdf> [Consulta: Septiembre de 2005]
- [30] **Unitronics comunicaciones**. “Arquitecturas en Telefonía IP y Factores de convergencia Voz/Datos”. [En línea]. Disponible en web: <<http://www.comunicaciones.unitronics.es/tecnologia/TelefoniaIP.htm>> [Consulta: Octubre 19 de 2005]
- [31] **RADVISION**. “IMS SIP and Signaling. The RADVISION Perspective” A Technology Overview. White Paper. [En línea]. 2006. Disponible en web: <<http://www.radvision.com/NR/rdonlyres/FC60D840-1FE5-4F82-A6A2-088D2D4AADCB/0/IMSSIPWhitePaper.pdf>> [Consulta: Agosto de 2006]
- [32] **RFC 3455**. “Private Header (P-Header) Extensions to the Session Initiation Protocol (SIP) for the 3rd-Generation Partnership Project (3GPP)”. [En línea]. Enero 2003. Disponible en web: <<http://www.rfc-archive.org/getrfc.php?rfc=3455>> [Consulta: Agosto de 2006]
- [33] **Telefónica**. “Características técnicas de las interfaces de Telefónica España. Interfaz para la conexión de terminales a los servicios de voz sobre IP”. [En línea]. Noviembre 2005. Disponible en web: <[http://www.recursovoip.com/docs/spanish/ITE\\_BA\\_009\\_v1.pdf](http://www.recursovoip.com/docs/spanish/ITE_BA_009_v1.pdf)> [Consulta: Agosto de 2006]
- [34] **3GPP, TSG-CN**. “Missing statements regarding P-Charging-Function-Addresses”. [En línea]. Marzo 2004. Disponible en web: <[http://www.3gpp.org/ftp/tsg\\_cn/tsg\\_cn/TSGN\\_23/Docs/PDF/NP-040039.pdf](http://www.3gpp.org/ftp/tsg_cn/tsg_cn/TSGN_23/Docs/PDF/NP-040039.pdf)> [Consulta: Agosto de 2006]
- [35] **3GPP, TR 45**. “All-IP Core Network Multimedia Domain. IP Multimedia Subsystem – Charging Architecture”. [En línea]. Disponible en web: <[http://ftp.tiaonline.org/tr-45/tr452/Public/PN-3-4935.007\\_X.P0013.007\\_IMS\\_Charging\\_Architecture/PN-3-4935.07%20v033%20IMS%20Charg%20Arch%20Post%20Ballot.doc](http://ftp.tiaonline.org/tr-45/tr452/Public/PN-3-4935.007_X.P0013.007_IMS_Charging_Architecture/PN-3-4935.07%20v033%20IMS%20Charg%20Arch%20Post%20Ballot.doc)> [Consulta: Agosto de 2006]
- [36] **3GPP, TS 32.200**. “3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; Telecommunication management; Charging management; Charging principles. Release 5”. [En línea]. Septiembre 2005. Disponible en web: <[http://www.3gpp.org/ftp/Specs/archive/32\\_series/32.200/32200-590.zip](http://www.3gpp.org/ftp/Specs/archive/32_series/32.200/32200-590.zip)> [Consulta: Agosto de 2006]
- [37] **Günther Horn, Dirk Kröselberg, Klaus Müller**. “IMS authentication and integrity/confidentiality protection”. Siemens AG, Corporate Technology, Competence Centre Security. [En línea]. Noviembre 2000. Disponible en web:



- <[http://www.3gpp.org/ftp/tsg\\_sa/WG3\\_Security/TSGS3\\_15bis\\_Munich/Docs/PDF/S3z000035.pdf](http://www.3gpp.org/ftp/tsg_sa/WG3_Security/TSGS3_15bis_Munich/Docs/PDF/S3z000035.pdf)> [Consulta: Abril de 2006]
- [38] **3GPP**. “3GPP TS 33.102 V7.0.0 (2005-12) 3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; 3G Security; Security architecture (Release 7)”. [En línea]. Diciembre 2005. Disponible en web: <[http://www.arib.or.jp/IMT-2000/V600Dec06/2\\_T63/ARIB-STD-T63/Rel7/33/A33102-700.pdf](http://www.arib.or.jp/IMT-2000/V600Dec06/2_T63/ARIB-STD-T63/Rel7/33/A33102-700.pdf)> [Consulta: Abril de 2006]
- [39] **Dulude Louise**. “Automated telephone answering systems and aging”. Conference Papers – American Association for Public Opinion Research. Mayo 2002. Vol. 21 Issue 3, p171-184. [Consulta: Abril de 2006]
- [40] **Singer Eleanor**. “Does Voice Matter? An Interactive Voice Response (IVR) Experiment”. Behaviour & Information Technology. 2003 Annual Meeting, Nashville, TN. [Consulta: Abril de 2006]
- [41] **Corkrey Ross**. “Interactive Voice Response: Review of studies 1989–2000”. Behavior Research Methods, Instruments, & Computers. Agosto 2002. Vol. 34 Issue 3. [Consulta: Abril de 2006]
- [42] **Kuperstein Michael**. “Computer Speech Rec at Every Call Center”. Communications Convergence. Diciembre 2003. Vol. 11 Issue 12. [Consulta: Abril de 2006]
- [43] **Caton, Michael**. “Banking on speech services”. Agosto 2004. Vol. 21 Issue 35. [Consulta: Abril de 2006]
- [44] **Lenke Nils**. “Supercharging man/machine dialog”. Electronic Engineering Times. Marzo 1999. Issue 1052. [Consulta: Abril de 2006]
- [45] **B.H. Juang**. “Automatic Speech Recognition” – A Brief History of the Technology Development. [En línea]. Disponible en web: <[http://www.caip.rutgers.edu/~lrr/lrr%20papers/353\\_LALI-ASRHistory-final-10-8.pdf](http://www.caip.rutgers.edu/~lrr/lrr%20papers/353_LALI-ASRHistory-final-10-8.pdf)> [Consulta: Abril de 2006]
- [46] **Richard V**. “A segmental speech coder based on a concatenative TTS”. Speech Communication. Septiembre 2002. Vol. 38. [Consulta: Abril de 2006]
- [47] **Baum, David**. “Will your PC talk back by 2001?”. InfoWorld. Mayo 1994. Vol. 16 Issue 20. [Consulta: Abril de 2006]
- [48] **Van Santen**. “Synthesis of prosody using multi-level unit sequences”. Speech Communication. Julio 2005. Vol. 46 Issue 3/4. [Consulta: Abril de 2006]
- [49] **Scharenborg, Odette**. “Early recognition of polysyllabic words in continuous speech”. Computer Speech & Language. Enero 2006. Vol. 21. [Consulta: Abril de 2006]
- [50] **M C Bale**. “Voice and Internet multimedia in UMTS networks”. [En línea]. Enero 2001. Disponible en web: <[http://www.sipcenter.com/sip.nsf/html/WEBB5YH724/\\$FILE/Voice\\_and\\_MM\\_in\\_3G.pdf](http://www.sipcenter.com/sip.nsf/html/WEBB5YH724/$FILE/Voice_and_MM_in_3G.pdf)> [Consulta: Abril de 2006]
- [51] **Voxpilot**. “Interactive Services for 3G and IMS Networks: The Next Wave”. [En línea]. Mayo 2006. Disponible en web: <<http://www.voxpilot.com/pdf/IVVR3G.pdf>> [Consulta: Agosto de 2006]



- [52] **3GPP**. “3GPP TR 23.877 V6.0.0 (2004-03) 3rd Generation Partnership Project; Technical Specification Group Services and System Aspects Architectural aspects of Speech Enabled Services; (Release 6)”. [En línea]. Marzo 2004. Disponible en web:  
<[http://www.3gpp.org/ftp/specs/archive/23\\_series/23.877/23877-600.zip](http://www.3gpp.org/ftp/specs/archive/23_series/23.877/23877-600.zip)> [Consulta: Abril de 2004]
- [53] **3GPP**. “3GPP TS 23.333 V0.3.0 (2006-09) 3rd Generation Partnership Project; Technical Specification Group Core Network and Terminals; Multimedia Resource Function Controller (MRFC) – Multimedia Resource Function Processor (MRFP) Mp interface: Procedures Descriptions (Release 7)”. [En línea]. Septiembre 2006. Disponible en web:  
<<http://www.3gpp.org/ftp/specs/latest-drafts/23333-030.zip>> [Consulta: Octubre de 2006]
- [54] **Eclipse**. Disponible en web: <<http://www.eclipse.org/>>
- [55] **Netbeans**. Disponible en web: <<http://www.netbeans.org/>>
- [56] **Borland**. Disponible en web: <<http://www.borland.com/>>
- [57] **The Java Community Process**. “SIP Servlet API Version 1.0”. [En línea]. Febrero 2003. Disponible en web:  
<<https://sdic1b.sun.com/ECOM/ECOMActionServlet;jsessionid=44E0C3137185115CC66BC66F8D0C7874/>> [Consulta: Agosto de 2006]
- [58] **Dinamicsoft**. “SIP Servlets”. [En línea]. Disponible en web:  
<[http://phoenix.labri.fr/documentation/sip/Documentation/Papers/Programming\\_SIP/Presentation/SIP\\_Servlet/Presentation\\_SIP\\_Servlets.ppt](http://phoenix.labri.fr/documentation/sip/Documentation/Papers/Programming_SIP/Presentation/SIP_Servlet/Presentation_SIP_Servlets.ppt)> [Consulta: Agosto de 2006]
- [59] **Ubiquity Developer Network**. Disponible en web: <<https://developer.ubiquitysoftware.com/>>
- [60] **MICROMETHOD**. SIPMethod Platform. Disponible en web:  
<<http://www.micromethod.com/products/sipmethod.htm>>
- [61] **BEA**. BEA WebLogic Communications Platform. Disponible en web:  
<<http://www.bea.com/framework.jsp?CNT=index.htm&FP=/content/products/wlcom/>>
- [62] **voip-news**. “BroadSoft Announces IMS-ready Media Resource Function Server”. [En línea]. Disponible en web: <<http://www.voip-news.com/press-releases/broadsoft-ims-media-server-032806/>> [Consulta: Noviembre de 2006]
- [63] **dailywireless.org**. “IMS: HP’s Backdoor Play”. [En línea]. Noviembre 2005. Disponible en web:  
<<http://www.dailywireless.org/2005/11/30/ims-hps-backdoor-play/>> [Consulta: Noviembre de 2006]
- [64] **3G.co.uk**. “New Release of Software Media Server to Drive Innovative IMS and 3G Applications”. [En línea]. Marzo 2006. Disponible en web:  
<<http://www.3g.co.uk/PR/March2006/2784.htm>> [Consulta: Noviembre de 2006]
- [65] **Microsoft Agent**. SAPI 4.0 runtime support. [En línea]. Disponible en web:  
<<http://www.microsoft.com/msagent/downloads/user.asp>> [Consulta: Enero de 2006]
- [66] **Universidad de Edinburgo**. The Festival Speech Synthesis System. [En línea]. Disponible en web:  
<<http://www.cstr.ed.ac.uk/projects/festival/>> [Consulta: Agosto de 2006]



- [67] **Carnegie Mellon University.** *Flite: a small, fast run time synthesis engine.* [En línea]. Disponible en web: <<http://www.speech.cs.cmu.edu/flite/>> [Consulta: Agosto de 2006]
- [68] **SourceForge.net.** *FreeTTS.* [En línea]. Disponible en web: <<http://sourceforge.net/projects/freetts/>> [Consulta: Agosto de 2006]
- [69] **MBROLA.** *The MBROLA Project.* [En línea]. Disponible en web: <<http://tcts.fpms.ac.be/synthesis/mbrola.html>> [Consulta: Agosto de 2006]
- [70] **gnu.org.** *gnuspeech.* [En línea]. Disponible en web: <<http://www.gnu.org/software/gnuspeech/>> [Consulta: Agosto de 2006]
- [71] **ATLAS, Verbio TTS.** *Verbio TTS.* [En línea]. Disponible en web: <<http://www.verbio.com/productes.php?id=1>> [Consulta: Agosto de 2006]
- [72] **NUANCE.** *Dragon NaturallySpeaking Edition 9. Versión Actual.* [En línea]. Disponible en web: <<http://www.nuance.com/naturallyspeaking/sdk/>> [Consulta: Enero de 2006]
- [73] **IBM.** *IBM Via Voice. Versión Actual.* [En línea]. Disponible en web: <<http://www-306.ibm.com/software/voice/viavoice/>> [Consulta: Enero de 2006]
- [74] **Silvia Janet Espinosa de la Peña.** "Software de reconocimiento de voz". [En línea]. Disponible en web: <<http://www.itq.edu.mx/vidatec/espacio/aisc/ARTICULOS/ASCC/articuloASCC.htm>> [Consulta: Enero de 2006]
- [75] **ATLAS, Verbio ASR.** *Verbio ASR (Automatic Speech Recognition - Reconocimiento del habla).* [En línea]. Disponible en web: <<http://www.verbio.com/productes.php?id=2>> [Consulta: Enero de 2006]
- [76] **Carnegie Mellon University.** *The CMU Sphinx Group Open Source Speech Recognition Engines.* [En línea]. Disponible en web: <<http://cmusphinx.sourceforge.net/html/cmusphinx.php>> [Consulta: Enero de 2006]
- [77] **gnu.org.** *The GNU oSIP library.* [En línea]. Disponible en web: <<http://www.gnu.org/software/osip/>> [Consulta: Septiembre de 2006]
- [78] **SIMPLE.** [En línea]. Disponible en web: <<http://www.rfc-editor.org/rfc/rfc3428.txt>>
- [79] **SourceForge.net.** *Sofia-SIP.* [En línea]. Disponible en web: <<http://sourceforge.net/projects/sofia-sip>> [Consulta: Septiembre de 2006]
- [80] **Yate.** *Yet Another Telephony Engine.* [En línea]. Disponible en web: <<http://yate.null.ro/pmwiki/>> [Consulta: Septiembre de 2006]
- [81] **Jori Liesenborgs.** *JRTPLIB.* [En línea]. Disponible en web: <<http://research.edm.uhasselt.be/~jori/page/index.php?n=CS.Jrtplib>> [Consulta: Marzo de 2006]
- [82] **gnu.org.** *GNU ccRTP.* [En línea]. Disponible en web: <<http://www.gnu.org/software/ccrtp/>> [Consulta: Marzo de 2006]
- [83] **MySQL AB.** [En línea]. Disponible en web: <<http://www.mysql.com/>>



- [84] **PostgreSQL**. [En línea]. Disponible en web: <<http://www.postgresql.org/>>
- [85] **Phelim O'Doherty – Sun Microsystems, Inc.** “*SIP Specifications and the Java Platforms, the look and feel of SIP*”. [En línea]. Enero 2003. Disponible en web: <<http://www.cs.columbia.edu/sip/Java-SIP-Specifications.pdf>> [Consulta: Septiembre de 2005]
- [86] **Java Community Process, JSR 180 Expert Group.** *JSR-180 SIP API for J2ME™, Version 1.0.1*. [En línea]. Diciembre 2004. Disponible en web: <[http://sw.nokia.com/id/b81fe286-720f-4133-b20a-5d8d21bee228/JSR-180\\_SIP\\_API\\_for\\_J2ME.zip](http://sw.nokia.com/id/b81fe286-720f-4133-b20a-5d8d21bee228/JSR-180_SIP_API_for_J2ME.zip)> [Consulta: Marzo de 2006]
- [87] **Symbian**. Disponible en web: <<http://www.symbian.com/>>
- [88] **Nokia Forum.** *Carbide.j 1.5*. [En línea]. Disponible en web: <<http://forum.nokia.com/info/sw.nokia.com/id/d9f7e9b2-3932-4358-9e8e-aa5cd26be54e.html>> [Consulta: Marzo de 2006]
- [89] **Nokia Forum.** *Carbide.c++ Express v1.0*. [En línea]. Disponible en web: <[http://www.forum.nokia.com/info/sw.nokia.com/id/ad80c54a-76ae-4846-9427-8036ce26e443/Carbide\\_c\\_Express\\_10.html](http://www.forum.nokia.com/info/sw.nokia.com/id/ad80c54a-76ae-4846-9427-8036ce26e443/Carbide_c_Express_10.html)> [Consulta: Marzo de 2006]
- [90] **SDN – Sun Developer Network.** *Sun Java Wireless Toolkit 2.5 for CLDC, Beta 2 Release*. [En línea]. Disponible en web: <[http://java.sun.com/products/sjwtoolkit/download-2\\_5.html](http://java.sun.com/products/sjwtoolkit/download-2_5.html)> [Consulta: Mayo de 2006]
- [91] **SDN – Sun Developer Network.** *Mobile Media API (MMAPI); JSR 135*. [En línea]. Disponible en web: <<http://java.sun.com/products/mmapi/>> [Consulta: Enero de 2006]
- [92] **Universidad de Columbia.** *java.net.RTP*. [En línea]. Disponible en web: <[http://www.cs.columbia.edu/~hgs/teaching/ais/1998/projects/java\\_rtp/report.html](http://www.cs.columbia.edu/~hgs/teaching/ais/1998/projects/java_rtp/report.html)> [Consulta: Marzo de 2006]
- [93] **Serrano, Carlos Enrique.** *Conferencias Modelo Para la Construcción de Soluciones. Universidad del Cauca*. [En línea]. Diciembre 2002. [Consulta: Abril de 2004]
- [94] **Junqua Jean-Claude.** “*Robust Speech Recognition in embedded systems and PC applications*”. Panasonic Technologies, Inc., U. S. A. [Consulta: Abril de 2006]
- [95] **Holmes John.** “*Speech Synthesis and Recognition*”. 2001. ISBN0-203-48468-1. [Consulta: Abril de 2006]
- [96] **Smith, Ronnie; Hipp, Richard.** “*Spoken Natural Language Dialog Systems: A Practical Approach*”. Oxford University Press, Incorporated. [Consulta: Abril de 2006]
- [97] **Frank Gaine, frontend.com.** “*Usable Interactive Voice Response Applications*”. [En línea]. Disponible en web: <<http://infocentre.frontend.com/infocentre/articles/usableivrapplications.html>>. [Consulta: Abril de 2006]
- [98] **Fiona Murphy, frontend.com.** “*Introduction to User Centred Design Process*”. [En línea]. Disponible en web: <<http://infocentre.frontend.com/infocentre/articles/introtouc.html>>. [Consulta: Abril de 2006]



- [99] **Inovdesigns, Inc.** *“Design Guidelines for Interactive Voice Response Systems”*. [En línea]. Disponible en web: <<http://www.inovdesigns.com/learningcenter/ivr.asp>>. [Consulta: Abril de 2006]
- [100] **Dream Tech Software India Inc.** *“VoiceXML 2.0 Developer's Guide”*. McGraw-Hill Professional. 2002. [Consulta: Abril de 2006]
- [101] **Steinbach, Markus.** *“Middle Voice. A comparative study in the syntax-semantics interface of German”*. 2002. [Consulta: Abril de 2006]