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

Integración de un Sistema de Comunicaciones Unificadas con Redes Sociales

Trabajo Especial de Grado
presentado ante la Ilustre
Universidad Central de Venezuela
por los Bachilleres:

Francisco J. Mora M.
C.I.: 20826830
mora_francisco123@hotmail.com

Diego E. Oliveros De F.
C.I.: 24041185
Oliverosdiego17@gmail.com

Para optar al título de Licenciado en Computación

Tutores**:** Prof. Dedaniel Urribarri y Prof. Eric Gamess

Caracas, Abril 2017

Universidad Central de Venezuela

Facultad de Ciencias
Escuela de Computación
Laboratorio de Comunicación y Redes

ACTA DE VEREDICTO

Quienes suscriben, miembros del jurado designado por el Consejo de Escuela de Computación, para examinar el Trabajo Especial de Grado presentado por los Bachilleres Francisco Mora C.I. V-20.826.830 y Diego Oliveros C.I. V-24.041.185, con el título "**Integración de un Sistema de Comunicaciones Unificadas con Redes Sociales**", a los fines de cumplir con el requisito legal para optar al título de Licenciado en Computación, dejan constancia de lo siguiente:

Leído el nombrado trabajo por cada uno de los miembros del jurado, éste fijó el día jueves 30 de marzo de 2017, para que sus autores lo defendieran en forma pública, en la Sala PAIII de la Escuela de Computación, mediante una exposición oral de su contenido, luego de la cual respondieron satisfactoriamente a las preguntas que les fueron formuladas por el jurado, todo ello conforme a lo dispuesto en la Ley de Universidades y demás normativas vigentes de la Universidad Central de Venezuela.

Finalizada la defensa pública del Trabajo Especial de Grado, el jurado decidió aprobarlo.

Es de aclarar que el Profesor Eric Gamess se encuentra de permiso fuera del país. Por esta razón, estuvo vía videoconferencia, de común acuerdo con los demás miembros del jurado. Por ende, el Profesor Robinson Rivas, director de la Escuela de Computación, firma el presente documento en su lugar.

En fe de lo cual se levanta el presente acta en Caracas a los 30 días del mes de marzo del año 2017, dejando constancia que el Profesor Dedaniel Urribarrí actuó como coordinador del jurado.

Prof. Dedaniel Urribarri
Tutor

Prof Robinson Rivas

Prof. Eric Gamess
Tutor

Prof. Roger Bello
Jurado Principal

Prof. Antonio Russoniello
Jurado Principal

Agradecimientos

Agradezco a Dios, por haberme dado la fortaleza y sabiduría necesaria para realizar y llevar a feliz conclusión esta investigación.

A toda mi familia, en especial a mi madre María Esperanza Monroy Lancheros, mi padre Francisco Antonio Mora García. Gracias a todos por estar pendientes y apoyarme en cada uno de los obstáculos que aparecieron en mi camino.

A la Universidad Central de Venezuela y a cada uno de los profesores que allí laboran, los cuales han brindado su tiempo para sembrar cada granito de conocimiento posible en mi ser y asi lograr vencer las sombras.

A mis tutores, el profesor Dedaniel Urribarri y el profesor Eric Gamess, que mediante su desinteresado y valioso esfuerzo, guiaron esta investigación y permitieron la culminación de este trabajo.

A todos mis amigos cercanos, que estuvieron al pendiente de todos mis avances y trabas, buscando así, apoyarme si estaba a su alcance. En especial a mi amigo Robert Arrieche, gracias por tu colaboración desinteresada y útil en las distintas etapas de la realización de este trabajo.

A todos mis compañeros de promoción, que de una u otra manera me prestaron su colaboración para llegar a la conclusión de esta etapa.

A todos, Gracias...

Francisco J. Mora M.

Agradecimientos

Dedico este trabajo y todos sus frutos a mi familia y amigos que estuvieron a lo largo de esta investigación, a lo largo de esta nueva etapa universitaria.

Agradezco a todos los que estuvieron involucrados en este trabajo especial de grado y su aporte, los que me conocen saben que soy un hombre de pocas palabras y prefiero demostrar mi agradecimiento a cada uno.

Gracias aquellos amigos que estuvieron pendiente de la evolución de esta investigación y ver todo el esfuerzo que desempeñe para que pudiera lograrse.

Gracias a Todos.

Diego E. Oliveros D. F.

Resumen

TÍTULO
Integración de un Sistema de Comunicaciones Unificadas con Redes Sociales

AUTORES
Francisco J. Mora M.
Diego E. Oliveros De F.

TUTORES
Prof. Dedaniel Urribarri y Prof. Eric Gamess.

Resumen

El crecimiento exponencial de la raza humana y las diferencias étnicas-culturales, han hecho que la comunicación entre distintos individuos sea de primera necesidad, simples acciones como ponerse al día, compartir ideas, llegar a un acuerdo, sirvieron de base al rápido avance de las tecnologías destinadas al mejoramiento de distintos procesos inmersos en las comunicaciones, generando así distintas herramientas que permiten la interacción, sea en ámbito casual o laboral.

Las organizaciones están compuestas por talento humano, esto motiva a que las comunicaciones entre cada integrante debe ser lo más precisa y confiable, para un correcto funcionamiento interno, por esta razón, se utilizan mecanismos que permitan mantener y/o mejorar la eficiencia de las operaciones.

Los puntos descritos anteriormente, se reducen en soluciones actualmente conocidas en todo el mundo, Servidores de Comunicaciones Unificadas y Redes Sociales, en primera instancia, éstos son manejados en planos paralelos y su funcionalidad conjunta en el área de gestión para atención al cliente no ha sido explotada. Por esta razón, este Trabajo Especial de Grado propone la Integración de un Servidor de Comunicaciones Unificadas con la Red Social Twitter.

Dicha integración se realizará construyendo 2 módulos, el primero, el módulo de procesamiento, se encargará de procesar los mensajes generados por los usuarios desde Twitter, y el segundo, el módulo social, será una extensión del servidor de comunicaciones unificadas y se encargará de la recepción y distribución de los mensajes ya procesados para su posterior atención mediante respuestas personalizadas, auto respuestas, o llamadas por parte de los agentes humanos asociados al módulo de Call Center.

Una vez se realice la implementación completa de la solución, se contarán con distintos escenarios de pruebas, los cuales serán generados desde la red social Twitter y gestionables desde la consola de Agente.

Palabras Claves: Comunicación, SIP, Asterisk, Elastix, Twitter.

Tabla de Contenido

Índice de Figuras

Índice de Tablas

1. Introducción

En la actualidad la importancia que tienen las comunicaciones las ha llevado a ser parte estratégica de las empresas. Todos los que actualmente trabajan en este mundo globalizado, ocupan parte de su jornada laboral en ver correos electrónicos, realizar llamadas, utilizar la mensajería instantánea para comunicarse con colaboradores o clientes, e incluso participar de videoconferencias. Por ello, los sistemas para las comunicaciones empresariales tienen un gran reto por delante, que es poder ofrecer a las empresas las herramientas necesarias para mejorar su forma de hacer negocios.

La telefonía IP es una tecnología que se implementa sobre la ya existente red de datos. Esta tecnología, desde finales de los noventas se encuentra en el mercado, pero no ha sido hasta hace poco que se ha generalizado gracias, principalmente, a la mejora y estandarización de los sistemas de control de la calidad de la voz y a la universalización del servicio de Internet. La telefonía IP es capaz de crear un ambiente eficiente en una organización con múltiples características, y permite integrar otras locaciones para centralizar las comunicaciones de una empresa y llevarlas a niveles globales.

Por otro lado, las redes sociales permiten una interacción rápida, efectiva y sencilla entre una gran cantidad de personas. El enfoque de las empresas en el uso de las redes sociales no se puede limitar a la publicación de productos y promociones, sino que debe apuntar a la resolución de dudas de los clientes, a solucionar problemas y a mantener este contacto directo por medio del Internet.

A continuación se da una breve descripción del contenido de los mismos, lo cual se pretende sea un medio que facilite al lector la comprensión de la distribución del contenido del documento:

Capítulo 1: Contiene el razonamiento detrás de la ubicación del problema y la definición de los objetivos que permitirán tener la solución a dichos problemas

Capítulo 2: Tiene como finalidad dar una introducción al lector sobre los procesos de comunicación y funcionamientos de la transmisión de voz sobre la red de datos, además del establecimiento de la comunicación a través del protocolo SIP.

Capítulo 3: Este capítulo se enfoca en el servidor de comunicaciones unificadas basado en Asterisk, su historia y características.

Capítulo 4: Este capítulo explica brevemente el concepto de redes sociales tanto a nivel general como técnico para poder integrarlo con diferentes sistemas a través de sus interfaces.

Capítulo 6: La experiencia recabada en trabajos afines, sirve de apoyo para la creación de nuevas teorías y nuevos productos, por ello que este capítulo hace referencia a algunos trabajos de interés que posean una relación directa con la investigación realizada.

Capítulo 7: Describe los procedimientos que se plantean para la integración de las redes sociales con arquitectura del servidor de comunicaciones unificados.

Capítulo8: Se describe las diferentes tecnologías usadas a lo largo del desarrollo de la solución, explicando las ventajas y desventajas de dichos software.

Capítulo 9: Contiene el diseño de la integración realizada, la descripción de las actividades realizadas para lograr dicho objetivo y el resultado de la misma.

2. El Problema

A lo largo de este capítulo se llevara a cabo la descripción y planteamiento del problema observado en las centrales telefónicas antiguas, la justificación por la cual los problemaspueden ser solucionados a través de la integración de servicios de redes sociales para que sea aún más versátil los sistemas telefónicos basados en VoIP. Para definir una propuesta bien elaborada, es importante definir los objetivos de esta como se podrá ver a lo largo de este capítulo.

2.1 Planteamiento del Problema

Mediante el estudio realizado previamente, se evidencia la creciente limitación que presentan las centrales telefónicas antiguas. Por ejemplo, este antiguo sistema telefónico presenta serios problemas de escalabilidad, y se dificulta mucho agregar nuevas líneas telefónicas. Además, presenta poca flexibilidad al momento de querer desarrollar aplicaciones específicas que cumplan con las necesidades de una empresa particular. Por estas razones, se ha observado a nivel mundial la migración del sistema telefónico empresarial hacia VoIP.

El uso de la tecnología ha aumentado en forma significativa en todos los ámbitos. En particular, las tecnologías de información se hacen ubicuas en todos los procesos llevados en la sociedad. La tendencia colectiva incrementó de manera sustancial el uso de las redes sociales, lo que conlleva a la mayoría de las personas a cambiar sus hábitos en lo personal como en lo profesional.

Sin embargo, aun cuando las redes sociales están tomando mucha fuerza, han estado muy poco presentes en los servicios de comunicación unificados, en particular en las siguientes áreas: gestión de procesos, atención al cliente, mercadeo y publicidad.Estas áreas pueden ser atendidas, agilizadas y mejoradas integrando los servicios de comunicación unificado con las redes sociales. En la actualidad,prácticamente son inexistentes las integracionesentre centrales telefónicas de última generación y redes sociales, por lo tanto, no hay evidencia sustancial de las bondades que brindarían estas herramientas trabajando en conjunto, ampliando de esta manera los servicios que ofrecen en la actualidad las centrales telefónicas de última generación.

2.2 Justificación del Problema

Se ha determinado que las organizaciones que utilizan los beneficios que ofrecen las centrales telefónicas de última generación, han tenido que adaptar su gestión general de procesos a los servicios actuales que éstas poseen, en vez de adaptar la misma a lo que requiere la empresa. Mediante la investigación realizada, se evidencia que uno de los servicios con mayor utilización en el área de gestión al cliente son las llamadas telefónicas, ya que los clientes prefieren una comunicación directa con el personal encargado.

Por los problemas mencionados, se da la necesidad de crear una solución que integre las redes sociales con los servicios de comunicación unificados en pro de ampliar, optimizar y agilizar los procesos que prestan las centrales telefónicas de última generación,

complementando los servicios que presta la organización en cuanto a la gestión de clientes.

2.3 Objetivos

A continuación se describen los objetivos que se desean lograr con la realización del trabajo propuesto.

2.3.1 Objetivo General

Ampliar las capacidades de un servidor de comunicaciones unificadas basado en Asterisk realizando una integración con la red social Twitter.

2.3.2 Objetivos Específicos

- Generar un módulo que permita la comunicación entre un servidor de comunicaciones unificadas basada en Asterisk y la red social Twitter.
- Establecer una comunicación efectiva usando un API de Twitter con el servidor de comunicaciones unificadas basada en Asterisk y sus protocolos de comunicación e integración.
- Clasificar los datos obtenidos de la red social para su correspondiente uso en el servidor de comunicaciones unificadas.
- Gestionar procesos internos en el servidor de comunicación unificado dado el requerimiento procesado por el integrador de redes sociales.
- Generar estadísticas de los procesos internos del servidor de comunicación unificado relacionados con el uso del integrador de redes sociales para mejorar la gestión e indicadores de atención de las organizaciones.

2.4 Alcances

Integrar el conjunto de funcionalidades del servidor de comunicaciones unificadas de última generación con las bondades asociadas a la gestión de atención al cliente que puede brindar la red social Twitter, mediante la extensión de un módulo que se encargue del procesamiento y enrutamiento de los mensajes obtenidos desde el API de Twitter hacia el Call Center para su correspondiente atención por acción humana o automática.

3. Estándar VoIP

La transmisión de voz y la transmisión de datos erandos conceptos que se consideraban teóricamente diferentes.Con la revolución del Internet y los avances tecnológicos en redes de datos se ha establecido que dichos conceptos pueden convivir sobre una misma red, lo cual representa un gran ahorro para quienes utilizan una red de datos en entornos como los hogares, instituciones, oficinas, sucursales, proveedores y clientes dentro de un entorno empresarial.

3.1 Definición VoIP

VoIP (Voice over Internet Protocol) o “Voz sobre Protocolo de Internet" es un método medianteel cual se toman señales de audio analógicas que se transforman en datos digitales que pueden ser transmitidos a través de Internet hacia una dirección IP determinada[1].El pasodelatelefoníatradicionalalatelefonía IP hapermitidodarunsalto cualitativoenla tecnología de comunicación.La telefonía IP facilita el envío de la voz y el video a través de las redes IP e Internet. Sin embargo,la vigencia de los conceptos básicos de la telefonía tradicional le otorga una gran importancia a su conocimiento y estudio.

Es muy importante diferenciar entre VoIP y Telefonía IP:

- VoIP: es el conjunto de normas, dispositivos, protocolos, en definitiva la tecnología que permite comunicar voz sobre el protocolo IP.
- Telefonía IP: hace referencia a comunicaciones telefónicas realizadas a través de redes TCP/IP. A diferencia de PSTN(Public Switched Telephone Network), que se compone de señales analógicas y digitales a través de una red con conmutación de circuitos, la telefonía IP utiliza conmutación de paquetes.

3.2 Funcionamiento

La voz sobre IP convierte las señales de voz estándar en paquetes de datos que son transportados a través de redes de datos en lugar de líneas telefónicas tradicionales. La evolución de la transmisión conmutada por circuitos a la transmisión basada en paquetes toma el tráfico de una PSTNy lo coloca en redes IP. Las señales de voz se encapsulan en paquetes IP que pueden transportarse por diversas tecnologías como Ethernet, Frame Relay, ATM o SONET.

3.2.1 Señalización

La señalización es la forma en que se comunican los dispositivos en la red, llevando a cabo un proceso de configuración, activación y coordinación de diversos componentes necesarios para hacer posible una llamada.Cuando se realiza una llamada telefónica por IP, la voz se digitalizay se envía en paquetes de datos IP. Estos paquetes se envían a través de Internet a la persona con la que se está hablando. Cuando alcanzan su destino, son ensamblados de nuevo y convertidos en la señal de voz original.

Los procedimientos de señalización de voz sobre IP juegan un papel muy importante en la red, ya que establecen, mantienen, finalizan, administran y proveen Calidad de Servicio (QoS) en la telefonía por paquetes. La señalización es utilizada para controlar y administrar la conexión entre dos puntos, ofreciendo funciones como supervisión,

marcado, llamadas y retorno de tonos de progreso. Los estándares normalmente utilizados para la señalización de VoIP se conocen como H.323, SIP (Session Initiation Protocol), MGCP (Media Gateway Control Protocol) e IAX (Inter-Asterisk eXchange). En la Sección 2.5 se realizará un estudio de SIP, el estándar más utilizado en la señalización de VoIP.

3.2.2 Codificador y Decodificador

La comunicación de voz es analógica, mientras que la red de datos es digital. El proceso de convertir ondas analógicas a información digital y viceversa se hace con un codificador-decodificador (códec). Los códec de voz son modelos matemáticos usados digitalmente para codificar y comprimir la información de audio analógica. Muchos de estos modelos matemáticos se basan en la capacidad que tiene el cerebro humano de obtener la información a pesar de que esta sea incompleta. Comúnmente conocidos como “codecs”, “speech coders” o “voice coders”, son parte fundamental en el manejo y uso de la tecnología VoIP, y se encargan de codificar-decodificar la voz.

El proceso de conversión de una señal analógica a digital es complejo.La mayoría de las conversiones se basan en la modulación codificada mediante pulsos (Pulse Code Modulation, PCM) o variaciones de ella.La función principal de un códec es codificar las muestras de la conversación del usuario en tramas mediante un código de modulación de pulso, de manera que la conversación logre las ventajas que presenta un sistema digital, como pueden ser: mejorar la señal en presencia de errores en la comunicación y disminuir la inestabilidad en las redes y transmisiones ruidosas. En el receptor, las tramas son decodificadas para obtener las muestras de conversación de PCM y después convertidas en forma de onda.

Los códec se clasifican en tres tipos: códec en forma de onda, Vocoder y códec híbrido. Cada uno de ellos maneja diferentes técnicas como: PCM, DPCM (Differential Pulse Code Modulation), ADPCM (Adaptive Differential Pulse Code Modulation), LPC (Linear Prediction Coding), Codificacion Multipulso (MPC), entre otros[2].La razón por la cual existen tantas variantes en la compresión se debe a años de investigación y una meta en común: mayor calidad, mayor eficiencia en el algoritmo y menor retardo en la compresión. En la Tabla 3.1 se puede observar las técnicas usadas por los distintos códec y el ancho de banda requerido para la transmisión de cada uno.

Año	Norma	Técnica	Velocidad (kbps)
1972	G.711	PCM	64
1984	G.721	ADPCM	32
1988	G.722	ADPCM	48, 56 y 64
1992	G.728	LD-CELP	16
1995	G.729	CS-CELP	8
1995	G.723.1	MPC-MLQ	5.3 y 6.4

Tabla 3.1: Técnicas de Comprensión de Voz

3.2.3 Establecimiento y Control de Transmisión

La conexión de una llamada es realizada por dos destinos finales estableciendo o abriendo una sesión de comunicación entre ambos. En una red PSTN, los conmutadores públicos o privados se conectan a un canal digital o analógico a través de la red para completar una llamada.En una implementación VoIP, esta conexión viene dada por

ráfagas de datos multimedia transportadas en tiempo real (audio, video o ambos). Esta conexión es el canal de la portadora y representa el contenido de la voz o video que está siendo transportado. Cuando una comunicación es finalizada, la sesión IP es liberada y los recursos de red opcionales son liberados.

Los protocolos que se utilizan para realizar las transmisiones de los paquetes de voz, video o datos en VoIP se ven reflejados en la Figura 3.1,donde se muestra el modelo de capas de la familia de protocolos de Internet relacionados con VoIP. En el caso de la voz, esta puede transportarse mediante tres caminos: sobre IP,sobre UDP/IP o sobreRTP/UDP/IP.

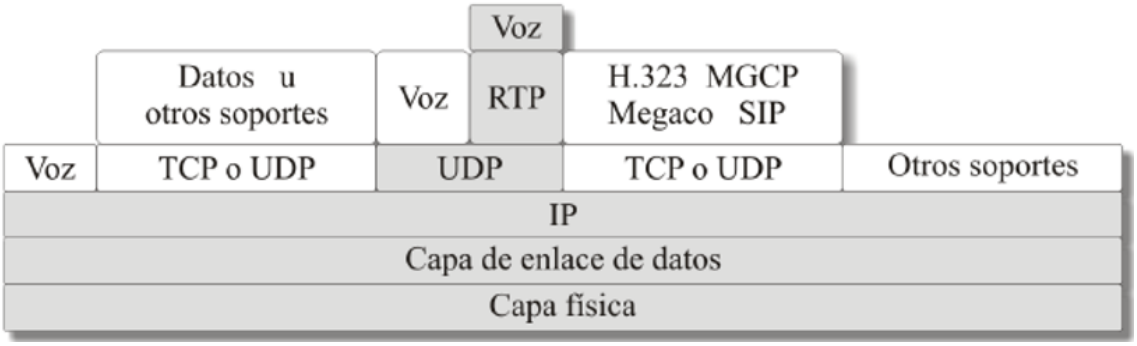

Figura 3.1: Conjunto de Protocolos de VoIP

Transmisión de Voz sobre IP: Transmitir directamente la voz sobre IP significa encapsular el tráfico de voz en la carga útil correspondiente al paquete IP.

Transmisión de Voz sobre UDP:UDP es un protocolo importante para operaciones VoIP debido a que se encarga de administrar y controlar los números de puertos en Internet entre computadores y aplicaciones, e identifica a estas últimas como la capa de aplicación que se ejecuta sobre UDP.La cabecera del datagrama UDP contiene los números de puerto, origen y destino, requeridos para la ejecución apropiada de los protocolos de la capa de aplicación. El número de puerto es concatenado con la dirección IP para formar el socket. Dicha dirección tiene que ser única en Internet y un par de sockets identifica a cada punto final de la conexión (end-point connection).

Aunque el mapeo de puertos destinado a procesos de capas superiores puede ser manejado como asunto interno en un host, Internet publica los números de puertos reservados para procesos de más alto nivel de uso frecuente conocidos como puertos bien conocidos (well-known ports). Los sockets identifican las sesiones entre las aplicaciones finales.

Otro punto a considerar con respecto al uso de UDP es que algunos protocolos de señalización basados en VoIP no pueden funcionar correctamente sin el uso de puertos. Por ejemplo, una de las funciones de SIP (Session Initial Protocol) es soportar el paso de números de puertos entre aplicaciones, los cuales se utilizan durante la llamada telefónica de paquetes (packet telephone call). Finalmente, resulta apropiada la utilización de puertos y por elloUDP es parte del camino de VoIP

Transmisión de Voz sobre RTP:RTP está diseñado para soportar tráfico de voz y video en tiempo real, brinda soporte a las aplicaciones unicast y multicast, proporciona servicios

que incluyen la identificación del tipo de la carga útil (por ejemplo, el tipo de tráfico de audio como G.723 o G.729), numeración de secuencia, muestra de tiempo (timestamping) y monitoreo de entrega.

Existen aplicaciones que usualmente ejecutan RTP sobre UDP para realizar la suma de comprobación (checksum) y la multiplexación de puertos UDP. RTP soporta transferencia de datos para múltiples destinos usando multicast. Mediante los números de secuencia, el receptor puede reconstruir la secuencia de paquetes del remitente, además RTP proporciona funciones útiles a los usuarios de VoIP.

3.3 Arquitectura de Red VoIP

Los componentes que conforman una plataforma de red VoIP, a pesar de poseer ciertas diferencias con respecto a los de la red PSTN (Public Switched Telephone Network), realizan funcionalidades similares, permitiendo así la ejecución de todas las tareas realizadas por una red PSTN. La Figura 3.2muestrauna arquitectura general de los elementos que conforman una red VoIP.

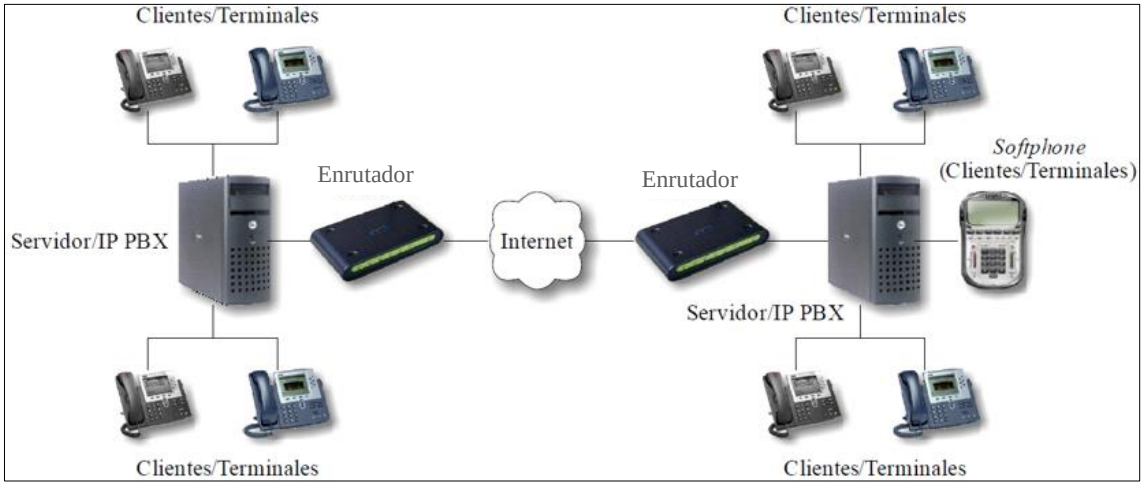

Figura 3.2: Elementos de la Tecnología VoIP

3.3.1 Central Telefónica IP PBX

PBX (Private Branch Exchange) es el corazón del sistema telefónico de VoIP. Es el encargado de controlar un gran númerode operaciones de base de datos en tiempo real, las cuales incluyen validación de usuarios, distribución de utilidades, enrutamiento, administración general del servicio, carga de clientes, control del servicio, registro de usuarios y servicios de directorio. Este elemento es de vital importancia, ya que sin él no se puede realizar la administración del registro y el establecimiento de las llamadas VoIP.

El servidor de procesamiento de llamadas usualmente está basado en software y puede ser implementado como un servidor independiente, como parte de un servidor o como una granja de servidores con funcionalidades distribuidas. IP PBX también puede estar basado en una plataforma de enrutador o como un dispositivo dedicado.

3.3.2 Teléfonos IP

Se encargan de establecer y terminar las llamadas de voz; codificar, empaquetar y transmitir la información de salida que se genera por el micrófono del usuario; recibir,

decodificar y reproducir la información de voz de entrada a través de las bocinas o audífonos del usuario. El teléfono puede presentarse como:

- Hardphones: estos teléfonos a primera vista se ven como los teléfonos convencionales, con un auricular, una base y cables. Sin embargo, los teléfonos IP, en lugar de tener un conector RJ-11 para la conexión a las líneas de teléfono convencional, vienen con un conector RJ-45 para conectarse directamente al punto de la red de datos y tienen todo el hardware y el software necesario para manejar correctamente las llamadas VoIP. También existen teléfonos IP inalámbricos con WiFique permiten llamadas VoIP a personas que utilicen este tipo de teléfonos, siempre que exista conectividad inalámbrica.
- Softphones: esta es la manera más fácil de utilizar VoIP. Todo lo que se necesita es un micrófono, parlantes, una tarjeta de sonido y un programa de Teléfono Virtual (Softphone), además de una conexión a la red o Internet. A excepciónde los costos porel servicio de Internet, usualmente no existe cargo alguno por este tipo de comunicaciones VoIP entre computador y computador, sin importar las distancias.

3.3.3 Gateways

Un gatewayes un dispositivo de red que permite que las llamadas salientes generadas por la central tradicional se conviertan a IPo viceversa.El gateway es un elemento esencial en la mayoría de las redes, pues su misión es la de enlazar la red VoIP con la red PSTN o RDSI (Red Digital de Servicios Integrados). Se puede considerar el gateway como una caja que por un lado tiene un interfaz LAN y por el otro dispone de una o varias de las siguientes interfaces:

- FXO: para conexión a extensiones de centrales o a la red telefónica básica.
- FXS: para conexión a enlaces de centrales o a teléfonos analógicos.
- E&M: para conexión específica a centrales.
- BRI: acceso básico RDSI (2B+D).
- PRI: acceso primario RDSI (30B+D).

3.3.4 Adaptadores Analógicos

Un ATA (Analog Telephone Adaptor) permite conectar teléfonos comunes no digitales (de los que se usan en la telefonía convencional) a un computador o a una red para utilizarlos con VoIP.Esta toma la señal analógica de los teléfonos convencionales y la convierte en datos digitales listos para ser transmitidos a través de la red de datos o Internet.

3.3.5 Gatekeeper

Es un elemento opcional en la red, pero cuando está presente, todos los demás elementos que contacten dicha red deben hacer uso de él. La función del gatekeeper es la de gestión y control de los recursos de la red, de manera que no se produzcan situaciones de saturación de la misma. El gatekeeper contempla las siguientes funciones:

- Servicio de traducción de direcciones (DNS):permite usar nombres en lugar de direcciones IP.
- Conversión de dirección (NAT): traducción de una dirección del alias a la dirección de transporte, al usar la tabla de traducción que se actualiza con los mensajes del registro.

- Control de Admisión: el gatekeeper puede conceder o negar el acceso basado en la autorización de la llamada, las direcciones fuente y destino o algunos otros criterios.
- Señalización de llamada: el gatekeeper puede ordenar, aprender y conocer los puntos finales para conectar la llamada.
- Autorización de llamadas: el gatekeeper junto con el gateway puede restringir las llamadas a ciertos números dentro de la red, hacer la marcación más versátil, por ejemplo, en casos de llamadas de emergencias.

3.3.6 Calidad de Servicio

Calidad de servicio es la capacidad que posee una red de ofrecer un mejor trato a un tráfico en particular. Esto se refleja en el rendimiento promedio visto por los usuarios de una red. Se mide en base a diferentes aspectos de la misma, tales como el jitter, el retardo, el ancho de banda, la tasa de errores, entre otros.

La comunicación de voz sobre IP debe tener características de tiempo real. Desafortunadamente, TCP/IP no siempre puede garantizar este tipo de particularidad, de modo que es imprescindible introducir algunas políticas que puedan manejar el flujo de paquetes en todos los enrutadores que deban intercambiar paquetes. Algunos de los elementos que se deben tomar en cuenta para mejorar la calidad de servicio en la telefonía IP son:

- Retardo: cuando se diseñan redes que transportan voz en paquetes es importante entender todos los posibles causales de retardos, para mantener la red en un estado aceptable. La calidad de la voz es función de muchos factores, como lo son: los algoritmos de compresión, los errores, las pérdidas de tramas, la cancelación del eco y los retardos.
- Jitter: se define técnicamente como la variación en el tiempo dela llegada de los paquetes, causada por congestión de red, pérdida de sincronización o por las diferentes rutas seguidas por los paquetes para llegar al destino. Las comunicaciones en tiempo real son especialmente sensibles a este efecto.
- Pérdida de paquetes: el porcentaje de pérdida de paquetes que pueda presentar una red depende básicamente del proveedor de servicio (ISP) que esté proporcionando el enlace.
- Consumo de ancho de banda: lograr transportar voz de alta calidad telefónica sobre IP en tiempo real no es una tarea nada fácil de realizar, ya que tal labor requiere manejo de las capacidades de la red que permita el control del tráfico, protocolos de tiempo real y anchos de banda “dedicados” durante el tiempo que tome la realización de la llamada.

Día a día las limitaciones en los servicios de voz basados en IP están siendo superadas, gracias a las mejoras en los algoritmos de compresión que permiten la optimización de la utilización del ancho de banda y el gran desarrollo de los actuales protocolos de enrutamiento.Dichos protocolos son capaces de tener en consideración el retardo por cada uno de los caminos posibles que puede tomar el paquete, para así determinar la mejor ruta que puede seguir.Además, puede proveer reservas de ancho de banda mientras dura la conversación y dar preferencia al procesamiento de los paquetes dentro

de los límites del enrutador, de manera que aquellos de alta prioridad son procesados primero.

3.4 Ventajas y Desventajas de VoIP

VoIP siendo una tecnología nueva está orientada a mejorar las comunicaciones. Su implementación logra numerosas ventajas sobre las tecnologías anteriores. Sin embargo,al ser una tecnología en proceso de madurez, también posee algunas desventajas. A continuación, se describen algunas de las ventajas y desventajas que se pueden encontrar en VoIP.

3.4.1 Ventajas

La tecnología VoIP presenta las siguientes ventajas[3]:

- Proporciona un número de teléfono único: con este número el usuario puede comunicarse desde cualquier lugar donde haya una conexión a Internet.
- Ahorro en llamadas de larga distancia: una llamada de PC a PC o entre sucursales conectadas mediante la red IP corporativa no representa costo alguno. El único costo que se incluye es el de los servicios de Internet.
- Llamadas a teléfonos fijos y celulares: la comunicación de VoIP no se limita a llamadas de PC a PC con conexión a Internet, sino que permite comunicarse a teléfonos de cualquier parte del mundo, ya sean fijos o celulares, con un ahorro aproximado del 80% sobre el costo de la llamada.
- Mensajería unificada y correos de voz: en situaciones en que no se pueda hacer uso de un teléfono, se pueden consultar los mensajes de voz mediante una conexión a Internet y si es necesario, se puede realizar la llamada al número telefónico del mensaje correspondiente mediante un softphone.
- Variedad de aplicaciones: existe una gran variedad de aplicaciones para VoIP que incluyen fax y multiconferencia.
- Trabajo a distancia (teletrabajo): es un nuevo concepto destinado a las bolsas de trabajo, en el que las personas pueden realizar su trabajo desde casa mediante una conexión IP con la red de la empresa.
- Servicios XoIP: hace referencia a un nuevo conjunto de servicios sobre IP, donde la “X” puede ser voz, video, datos, fax, etc.
- Supresión de silencios: la actividad de detección de voz (Voice Activity Detection, VAD) es un factor importante encargado de eliminar el envío de paquetes de silencio sobre la red IP y de detectar sólo los tiempos de conversación para que puedan ser enviados como paquetes de datos, con lo cual se libera ancho de banda.

3.4.2 Desventajas

La tecnología VoIP presenta las siguientes desventajas:

- Retardo o latencia: tiempo necesario para comprimir, transmitir y descomprimir los paquetes de voz; debe tener un rango de 120 a 150 ms para que la conversación tenga coherencia.
- Eco: por lo regular se debe al mal acoplamiento de impedancias.
- Pérdida de paquetes de voz: este fenómeno provoca pérdida de coherencia en la conversación, ya que el destino no cuenta con todos los paquetes de voz enviados desde el origen.
- Priorización de paquetes en la red IP: en una red IP interna se pueden emplear técnicas de priorización de paquetes como son: CQ (Custom Queuing), que asigna

un porcentaje del ancho de banda disponible; PQ (Priority Queuing), que establece prioridad en las colas; WFQ (Weight Fair Queuing), que asigna prioridad al tráfico de menos carga; y DiffServ,que se basa en la división del tráfico en diferentes clases, y en la asignación de prioridades a estos agregados.

3.5 Protocolo de Establecimiento de Sesión (SIP)

SIP (Session Initiation Protocol) es un protocolo de señalización simple utilizado para telefonía y videoconferencia por Internet. Basado en SMTP (Simple Mail Transfer Protocol) y HTTP (Hypertext Transfer Protocol), fue desarrollado por el IETF (Internet Engineering Task Force) con la intención de ser el estándar para la iniciación, modificación y finalización de sesiones interactivas de usuario, donde intervienen elementos multimedia como el video, voz, mensajería instantánea, juegos online y realidad virtual. SIP está definido en el RFC 3261[4].

3.5.1 Definición de SIP

SIP es un protocolo de la capa de aplicación independiente de los protocolos de paquetes subyacentes (TCP, UDP, ATM, X.25). SIP está basado en una arquitectura cliente servidor en la cual los clientes inician las llamadas y los servidores la responden. Es un protocolo abierto basado en estándares, ampliamente soportado y no es dependiente de un solo fabricante de equipos.

SIP puede establecer sesiones de dos partes (llamadas ordinarias), de múltiples partes (en donde todos pueden oír y hablar) y de multidifusión (un emisor, muchos receptores). Las sesiones pueden contener audio, video o datos.

3.5.2 Características Principales

El protocolo SIP posee cuatro características que lo hacen muy recomendable para cumplir la función de señalización:

- Localización del usuario: SIP posee la capacidad de poder conocer en todo momento la localización de los usuarios. De esta manera no importa en qué lugar se encuentre un determinado usuario.
- Negociación de los parámetros: posibilidad de negociar los parámetros necesarios para la comunicación como lo son los puertos para el tráfico SIP, así como el tráfico multimedia, direcciones IP para el tráfico multimedia, códec, etc.
- Disponibilidad del usuario: SIP permite determinar si un determinado usuario está disponible o no para establecer una comunicación.
- Gestión de la comunicación: permite la modificación, transferencia y finalización de la sesión activa. Además, informa del estado de la comunicación que se encuentra en progreso.

SIP requiere de otros protocolos para tener un sistema completo, tales como:

- Real-time Transport Protocol (RTP) (RFC 1889)[5] para transportar datos en tiempo real y proveer calidad de servicio (QoS).
- Real-Time Streaming Protocol (RTSP) (RFC 2326)[6] para controlar envío de datos multimedia.
- Session Description Protocol (SDP) (RFC 2327)[7]es el protocolo empleado para describir una sesión multimedia, que consiste en un conjunto de flujos de medios (audio, vídeo o datos), que existen durante un determinado tiempo. Los paquetes

SDP contienen (entre otros campos) información acerca del ancho de banda, los protocolos de transporte empleados, los códec utilizados en la sesión, y la dirección de contacto del iniciador de la sesión.

3.5.3 Arquitectura SIP

Para una comunicación con el protocolo SIP es necesaria la intervención de varios elementos, donde cada uno desempeña su función. Los elementos de la comunicación definidos en el RFC 3261[4] son:

- Agente de Usuario: es una aplicación con arquitectura cliente/servidor que se utiliza para iniciar y terminar las sesiones. El UAC (User Agent Client) se encarga de realizar peticiones SIP, mientras que el UAS (User Agent Server) notifica al usuario cuando se recibe una petición y responde a dicha petición dependiendo de la acción tomada por el usuario.
- El Servidor de Redirecciones: acepta una petición SIP y envía una respuesta al cliente que contiene las direcciones de los servidores con los que debe contactar el cliente.
- El Servidor Proxy: contiene funciones de servidor y cliente, y actúa como un intermediario que realiza peticiones en nombre de otros clientes. Para ello interpreta la cabecera del mensaje y la reescribe identificando al proxy como el que inicia la solicitud, recibe la respuesta del destinatario y se la reenvía al cliente.
- Un Servidor de Registro: almacena o actualiza en una base de datos la información de contacto del usuario que realiza la petición.
- Un B2BUA (Back to Back User Agent): es una entidad que recibe una petición INVITE y la procesa como un UAS. Para determinar la respuesta a la petición, actúa como un UAC que determina cómo responder a la petición y cómo realizar llamadas salientes. A diferencia de un proxy, un B2BUA debe mantener el estado de la llamada y participar activamente en ella, enviando peticiones y respuestas. Un B2BUA tiene un mayor control de la llamada que un proxy.

3.5.4 Mensajes SIP

Los componentes SIP se conectan intercambiando mensajes SIP. Los tipos de mensajes utilizados son peticiones y respuesta.Las peticiones son mensajes SIP enviados desde un cliente a un servidor con el propósito de invocar una operación en particular. En la Tabla 3.2 se muestra las distintas peticiones SIP definidas en el RFC 3261[4]:

Petición SIP	Descripción
INVITE	Permite invitar un usuario o servicio para participar en una sesión o para modificar parámetros en una sesión ya existente.
ACK	Confirma el establecimiento de una sesión. Esta petición es enviada por el usuario origen que envió la petición INVITE para hacer saber al usuario destino que su respuesta OK ha sido recibida. Es el momento en que ambos pueden empezar a enviar tráfico multimedia.
OPTION	Solicita información sobre las capacidades de un servidor. Un UAC puede enviar peticiones OPTIONs a un UAS para solicitar cierta información sobre este.
BYE	Indica la terminación de la sesión establecida anteriormente con INVITE.

CANCEL	Cancela una petición previamente enviada por un cliente. Específicamente, se pide a los UASs cesar el procesamiento de la petición y generar una respuesta de error a esa petición. CANCEL no tiene efecto sobre peticiones en las que el UAS ya ha dado una respuesta definitiva, es por ello que es útil para cancelar peticiones en las que el servidor tarda mucho tiempo en responder. INVITE es un buen ejemplo de este tipo de peticiones.
REGISTER	Registra al User Agent. Un UAC envía peticiones REGISTER a un servidor para informar de la posición actual en la que se encuentra en un momento determinado. Esto hace posible que el UAC pueda ser localizado haciendo uso de su misma dirección user@dominio sin importar donde el UAC se encuentre físicamente.

Tabla 3.2: Peticiones SIP

Las respuestas son mensajes SIP enviados desde un servidor a un cliente para indicar el estado de una petición enviada desde el cliente al servidor. Cada petición SIP lleva asociada una respuesta enumerada con un código que la identifica. Estos códigos van desde el identificador 100 hasta el identificador 699, y se disponen en grupos de respuestas tales como: 1xx, 2xx, 3xx, 4xx, 5xx y 6xx. En la Tabla 3.3 se muestra las distintas respuestas SIP definidas en el RFC 3261[4]:

Tipo de Respuesta	Identificador	Significado
Informan del estado provisional de la comunicación	100	Trying – Intentando
	180	Ringing – Sonando
	181	Call Is Being Forwarded – Llamada Está Siendo Transferida
	182	Queued – Encolada
	183	Session Progress – Llamada en Progreso
Informan del éxito de la comunicación	200	OK – OK
	202	Accepted – Aceptada
Informan del reenvío necesario de la petición SIP	300	Multiple Choices – Múltiples Opciones
	301	Moved Permanently – Movido Permanentemente
	302	Moved Temporarily – Movido Temporalmente
	305	Use Proxy – Usar Proxy
	380	Alternative Service – Servicio Alternativo
Informan de errores del cliente	400	Bad Request – Mala Petición
	401	Unauthorized – No Autorizado
	402	Payment Required – Se Requiere Pago
	403	Forbidden – Prohibido
	404	Not Found – No Encontrado
	405	Method Not Allowed – Método no Permitido
	406	Not Acceptable – No es Aceptable
	407	Proxy Authentication Required – Se Requiere Autenticación
	408	Request Timeout – Tiempo Agotado para la Petición
	410	Gone – Ya no Existe

	413	Request Entity Too Large – Petición Demasiado Grande
	414	Request URI Too Long – URI Demasiado Largo
	415	Unsupported Media Type – Tipo de Medio no Soportado
	416	Unsupported URI Scheme – Esquema URI no Soportado
	420	Bad Extension – Extensión Incorrecta
	421	Extension Required – Se Requiere Extensión
	423	Interval Too Brief – Intervalo Demasiado Corto
	480	Temporarily Unavailable – No Disponible Temporalmente
	481	Call/Transaction Does Not Exist – No Existe la Llamada/Transacción
	482	Loop Detected – Bucle Detectado
	483	Too Many Hops – Demasiados Saltos
	484	Address Incomplete – Dirección Incompleta
	485	Ambiguous – Ambiguo
	486	Busy Here – Ocupado
	487	Request Terminated – Petición Terminada
	488	Not Acceptable Here – No Aceptable en este Momento o Aquí
	491	Request Pending – Petición Pendiente
	493	Undecipherable – Indescifrable
Informan de errores del servidor	500	Server Internal Error – Error Interno del Servidor
	501	Not Implemented – No Implementado
	502	Bad Gateway – Pasarela (Gateway) Equivocada
	503	Service Unavailable – Servicio no Disponible
	504	Server Time-out – Tiempo Agotado en el Servidor
	505	Version Not Supported – Versión no Soportada
	513	Message Too Large – Mensaje Demasiado Largo
Informan de errores generales	600	Busy Everywhere – Ocupado en todos Sitios
	603	Decline – Rechazado
	604	Does Not Exist Anywhere – No Existe en nigún Sitio
	606	Not Acceptable – No Aceptable

Tabla 3.3: Respuesta del Protocolo SIP

3.5.5 Direccionamiento SIP

El protocolo es similar a HTTP por la forma en que funciona y a SMTP en la forma en la que se especifican las direcciones SIP. Las direcciones SIP identifican a un usuario de un determinado dominio. A estas direcciones SIP habitualmente se les llama URI (Uniform Resource Identifier). Ejemplos de URIs SIP:

sip:compusoft@worknet.com
sip:claudia@192.168.10.1
sip:178293@gateway.worknet.com

Dependiendo del tipo de “User Agent”, estos también pueden utilizar otros tipos de URIs como HTTP o mailto:

http://www.worknet.com
mailto:compusoft@worknet.com

3.5.6 Encabezado SIP

SIP se basa en un modelo de transacción del tipo solicitud-respuesta. En la Figura 3.3, la transacción comienza con una solicitud INVITE enviada al número telefónico de Bob. Cada solicitud contiene ciertos campos de encabezado que se denominan atributos y llevan información adicional. En este caso incluyen un identificador de la llamada, dirección destino, dirección origen e información sobre el tipo de sesión que se desea establecer.

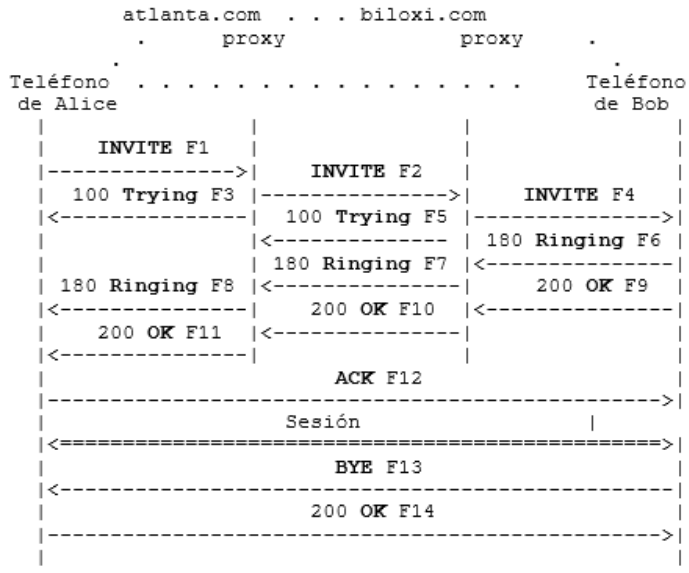

Figura 3.3: Establecimiento de Sesión SIP

A partir de este ejemplo se toma un mensaje INVITE mostrando su contenido como se muestra en la Figura 3.4.

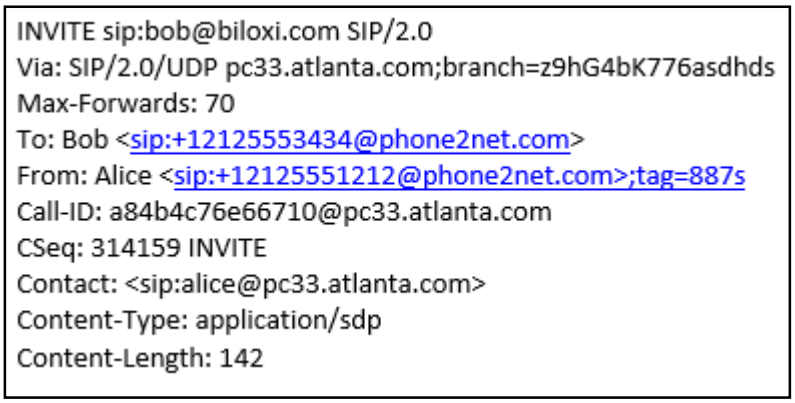

Figura 3.4: Detalle del Mensaje INVITE

Los campos del encabezado del mensaje INVITE de la Figura 2.4 se describen a continuación:

- Via: contiene la dirección (pc33.atlanta.com) de Alice ybranch identifica la transacción.
- Max-Forwards: limita el número de saltos que pueden realizarse para llegar al destino.
- To: contiene el nombre del destino (Bob) y su dirección de URI(sip:+12125553434 @phone2net.com).
- From: contiene el nombre del origen (Alice) y su dirección de URI(sip:+12125551212@phone2net.com). Para propósitos de identificación contiene una etiqueta aleatoria (tag) (887s) agregada por el equipo telefónico de origen.
- Call-ID: contiene un identificador único global para esta llamada, generado por la combinación de un número aleatorio y el dominio de origen. La combinación To tag, From tag y Call-ID definen completamente una relación SIP extremo a extremo (peer-to-peer) y se denomina diálogo.
- CSeq o Command Sequence: contiene un número entero y un método.El número CSeq se incrementa en cada nueva solicitud de un diálogo y es un número secuencial tradicional.
- Contact: contiene una dirección SIP o SIPS URI que representa una ruta directa para localizar al origen (Alice).
- Content-Type: contiene una descripción del contenido del mensaje (no mostrado).
- Content-Length: contiene la cantidad de bytes del contenido del mensaje.

Las respuestas en SIP usan un código de tres dígitos, seguido por una frase de descripción. Las respuestas contienen los mismos campos de “To”, “From”, “Call-ID”, “CSeq” y “branch” del parámetro “Via” del método INVITE.

3.5.7 Establecimiento de la Comunicación

En una llamada SIP hay varias transacciones SIP. Una transacción SIP se realiza mediante un intercambio de mensajes entre un cliente y un servidor.

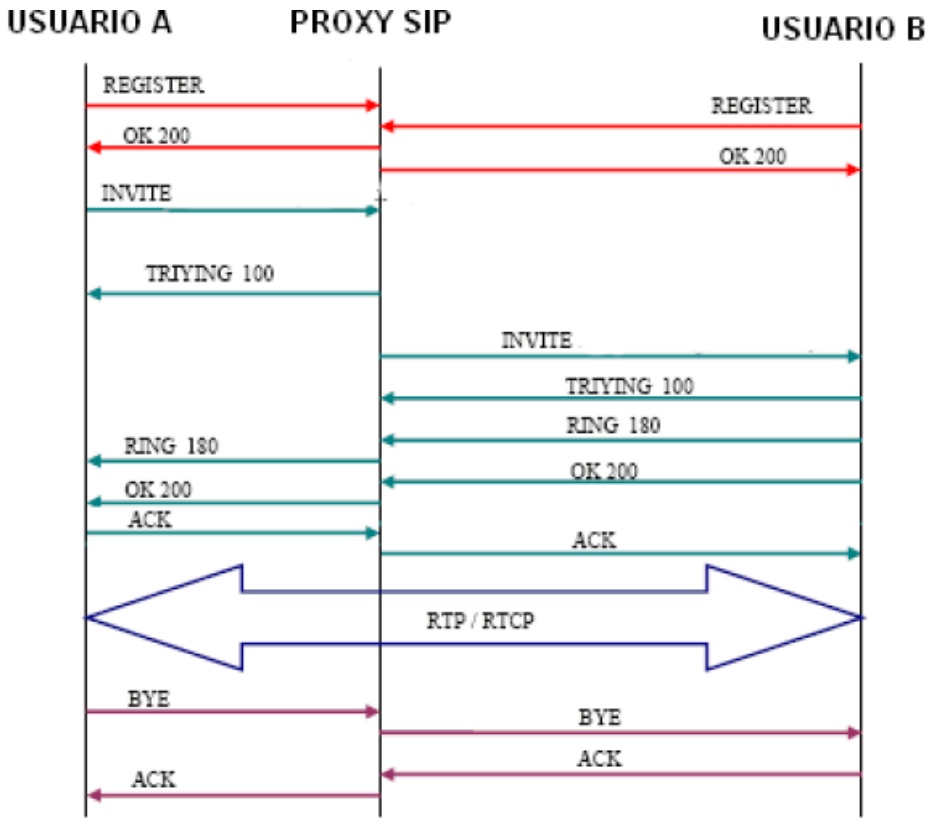

Figura 3.5: Comunicación entre Dos Usuarios SIP

En la Figura 3.5, las dos primeras transacciones corresponden al registro de los usuarios. Los usuarios deben registrarse para poder ser encontrados por otros usuarios, y en este caso, los terminales envían una petición REGISTER donde los campos “From” y “To” corresponden al usuario registrado. El servidor Proxy, que actúa como “Register”, consulta si el usuario puede ser autenticado y envía un mensaje de OK en caso positivo.

La siguiente transacción corresponde a un establecimiento de sesión. Esta sesión consiste en una petición INVITE del usuario al Proxy. Inmediatamente, el Proxy envía un “TRYING 100” para detener las retransmisiones y reenvía la petición al usuario B. El usuario B envía un “Ringing 180” cuando el teléfono empieza a sonar y también es reenviado por el Proxy hacia el usuario A. Por último, el “OK 200” corresponde a aceptar la llamada.En este momento la llamada está establecida. Pasa a funcionar el protocolo de transporte RTP con los parámetros (puertos, direcciones, codecs, etc.) establecidos en la negociación mediante el protocolo SDP.

La última transacción corresponde a una finalización de sesión. Esta finalización se lleva a cabo con una única petición BYE enviada al Proxy, y posteriormente reenviada al usuario B. Este usuario contesta con un “OK 200” para confirmar que se ha recibido el mensaje final correctamente.

3.6 Telefonía Tradicional

Los sistemas de telefonía tradicional están guiados por un sistema muy simple pero ineficiente denominado conmutación de circuitos,ésta ha sido usada por las operadoras tradicionales por más de 100 años. En este sistema cuando una llamada es realizada la

conexión es mantenida durante todo el tiempo que dure la comunicación y este tipo de comunicaciones es denominada “circuito” porque la conexión ésta realizada entre 2 puntos hacia ambas direcciones. Estos son los fundamentos del sistema de telefonía convencional.

Con el auge en los últimos años de la telefonía IP, la telefonía convencional analógica y digital va desapareciendo. Las facilidades de integración con muchos servicios ya existentes en las comunicaciones informáticas han permitido la aceptación y rápidaadopción de la telefonía IP, lográndose la incorporación de funcionalidades que con la telefonía convencional es imposible lograrlos.

Esto fue pasando de ser una simple PBX a todo un sistema de comunicaciones unificadas, en el cual además del servicio telefónico que se conoce ofrece otros como la integración de los mensajes de voz, libreta de contactos y faxes. Permite ademáscolaboración por medio de salas de conferencia dondepueden interactuar remotamente los participantes ycon esto se logra que la comunicación nunca se pierda sin importar donde se encuentre el usuario.

La telefonía IP aunque su plataforma de comunicación sea la red de datos y su funcionamiento es sobre la red IP, tiene como base y se rige por los mismo principios de la telefonía convencional, es por esto la importancia de conocer los fundamentos de la telefonía y así comprender porque muchas cosas se siguen haciendo igual desde que se inventó el teléfono.

4. Asterisk y Elastix

Aunque VoIP puede definirse de forma abreviada como una tecnología que aprovecha elprotocolo TCP/IP para ofrecer conversaciones de voz, lo cierto es que es mucho más que esto. VoIP puede ser usada para reemplazar la telefonía tradicional en un entorno empresarial,
enunpequeñonegociooencasa,osimplementeparaañadirventajasaunsistemade telefonía tradicional. Estosupusootrogranimpulsoala VoIPyprovocóquealdíadehoyexistan muchas solucionesquehacenusodeestatecnología como por ejemplo Asterisk y Elastix.

4.1 Introducción a Asterisk

Asterisk [3] es una plataforma de software de dominio público para el desarrollo de centrales telefónicas (PBXs), también es considerada como una aplicación de respuesta interactiva de voz (Interactive Voice Response (IVR)) para tecnología híbrida de TDM y VoIP. De forma no oficial, es posible que Asterisk sea considerado como la pieza de software disponible más poderosa, flexible y extensible para la integración de las diferentes tecnologías en el ámbito de telecomunicaciones. Su nombre viene dado a partir del símbolo asterisco (*), el cual es utilizado como un comodín en diferentes sistemas operativos para hacer referencia a cualquier tipo de archivo. Siguiendo este norte, Asterisk fue diseñado para interconectar cualquier pieza telefónica de hardware o software, con cualquier aplicación telefónica, de forma sencilla y consistente.

Normalmente, los productos telefónicos son diseñados para satisfacer una necesidad técnica específica en una red. Sin embargo, muchas aplicaciones utilizadas por la telefonía comparten gran parte de la tecnología. Asterisk toma ventaja de esa sinergia para crear un ambiente único que puede ser moldeado para encajar en cualquier aplicación en particular, o conjunto de aplicaciones.

Asterisk provee de todas aquellas funcionalidades que ofrece una central telefónica tradicional y a su vez, otro repertorio de herramientas adicionales que son de gran ayuda. Esta PBX de software implementa la tecnología de VoIP bajo diversos protocolos y a la vez puede ínter-operar con casi todos los estándares existentes basados en dispositivos telefónicos con la utilización de hardware económico.

4.1.1 Origen de Asterisk

La mente innovadora tras la creación de Asterisk fue el ingeniero Mark Spencer, que posteriormente se convertirá en el fundador de la empresa Digium, Inc. Esta organización tiene como principal objetivo continuar con el desarrollo y patrocinio de Asterisk, aprovisionando de igual forma una gran variedad de productos de hardware telefónico de altas y bajas prestaciones, así como servicios profesionales netamente relacionados al soporte de la PBX.

Asterisk tuvo su debut frente a la comunidad en septiembre de 2004. Dado que éste es un proyecto de código abierto, algunos segmentos importantes del mismo han sido provisto por desarrolladores que prestaron su colaboración desde diferentes partes del mundo. Otro punto importante, es que debido a la alta cantidad de participantes que apoyan este trabajo, la detección y corrección de errores se vuelve un procesosencillo y rápido, permitiendo así la optimización y crecimiento constante del mismo.

4.2 Funcionalidades de Asterisk

La meta primordial de Asterisk es proveer soluciones tecnológicas en el área de telefonía, buscando de esta manera, entregar una variada gama de funcionalidades potentes y a su vez flexibles. Se ofrecen tantas funcionalidades clásicas como cualquier PBX tradicional brindaría, y a su vez funcionalidades avanzadas que tendrían un alto costo en sistemas propietarios. Uno de los puntos fuertes a recalcar de Asterisk es que fue diseñado con la capacidad de ínter-operar tanto con estándares de sistemas telefónicos tradicionales, así como sistemas basados en VoIP. En la Tabla 3.1 se detallarán las funciones que se consideran de mayor importancia[8]:

Básicas	Avanzadas
Transferencias: Permite transferir llamadas de una extensión a otra. • Atendida: Se acepta la llamada. • Directa: Sin consentimiento.	**Buzón de voz**: Para situaciones en las que el usuario no pueda atender una llamada, será transferida al sistema para su posterior escucha. Usa correo para avisar sobre el mensaje.
Desvíos: Transferencia automática de llamadas entrantes hacia un número determinado dada cierta condición.	**Operadora IVR**: Aplicación que permite interacción con el usuario a través de diferentes directorios programados.
Capturas: Se obtiene la llamada activa de otra extensión en una distinta usando *8 por defecto.	**Distribución automática de llamadas**: Aplicación que permite distribuir las llamadas en un grupo específico de agentes.
Conferencia múltiple: Conectar múltiples usuarios a una misma llamada si es soportado por el modelo del terminal.	**Sala de Conferencias**: Sistema diseñado para la puesta en marcha de reuniones dentro de la empresa, accesible tanto interna como externamente.
Llamada directa a extensión: Permite enrutar llamadas directamente de un DDI/DID hacia una extensión.	**Gestor de llamadas por horario**: Permite definir el destino de las llamadas basándose en el horario en el que se realiza la misma.
Grupos de llamadas: Las llamadas entrantes serán dirigidas a un grupo, donde múltiples extensiones sonarán a la vez dada la política del negocio.	**Extensiones DISA**: Permite configurar post marcaciones para ejecutar de manera rápida llamadas a nuevos destinos una vez comunicado con la PBX.
***DND* - Do Not Disturb**: Opción de no molestar para impedir las llamadas entrantes a la extensión asignada.	**Callback y Retrollamada**: Módulos para ejecutar acciones en base a llamadas perdidas.

Tabla 4.1: Funcionalidades de Asterisk

4.2.1 Llamadas

Ahora se listan de forma general las funcionalidades de llamadas [9] presentes en el módulo features de Asterisk en forma predeterminada:

- Redirección de llamadas
- Llamadas en espera
- No molestar (DND)
- Servicios
- Discado rápido
- Voicemail
- Colas
- Lista negra
- Transferir una llamada
- Estacionamiento de llamada
- Grabaciones

4.2.2 Escalabilidad

La capacidad que posee Asterisk de escalar viene limitada por la premisa “¿qué tanto se puede crecer sin perder la estabilidad del sistema?” [10]. En primera instancia,se necesita escalar básicamente cuando el servidor Asterisk no cubre las necesidades que se deben prestar. Como primera opción para solucionar esto, se tiene la compra de una central telefónica más robusta y potente, pero eso no quiere decir que se depende obligatoriamente de una marca u organización para cumplir los objetivos, dicho esto, existen 2 acciones que solucionan el escenario recién mencionado, pero cada una posee su norte.

- Escala Vertical: Se basa en aumentar las prestaciones en el ámbito de hardware del servidor ya existente para que aumente la carga que puede soportar. Además se buscan externalizar servicios y disminuir las cargas en el uso de protocolos costosos en uso del CPU.
- Escala Horizontal: A diferencia del escalado vertical, este se centra en la adición de servidores para que trabajen en conjunto y sirvan peticiones de manera paralela bajo un balanceador de carga dado que Asterisk no puede ser instalado en un clúster[11].

Los elementos presentes en una PBX que deben ser evaluados para elevar sus prestaciones son los siguientes:

- Extensiones: Cuando se habla sobre el aumento del número de extensiones, Asterisk puede manejar una gran cantidad de estas fácilmente, pero, a medida que estas crecen, la recreación de los archivos de configuración ylas recargas del servicio que se ejecutan con cada cambio en la central acarrean más tiempo de procesador, por lo que después de cierta cantidad el sistema se ve inhibido. Además otros factores importantes que influyen en las extensiones son sus configuraciones en cuanto al códec y protocolo de comunicación usados por la misma[12].
- Líneas Telefónicas: Debido a la gran cantidad de usuarios finales que pueden alcanzarse mediante la telefonía convencional, Asterisk tiene la capacidad de usar

líneas telefónicas para expandir su rango de comunicación efectiva desde la organización hacia el exterior aprovechando la infraestructura de las redes PSTN existentes. Para realizar dicha conexión existen diferentes métodos, tales como líneas POTS, ISDN, T1/E1 hasta conexiones VoIP, cada uno con sus ventajas y desventajas frente a los otros, pero básicamente el uso de cualquiera de estas tecnologías será restringida por las necesidades de la PBX [11].

- Niveles de Servicio: Cada funcionalidad adjunta a Asterisk puede evaluarse como un servicio autónomo, ya que cada uno de estos expanden las funcionalidades que ofrece la central PBX en diferentes ámbitos. Dada la premisa anterior, se pueden hacer crecer de manera "ilimitada" las funcionalidades de Asterisk por así decirlo, debido a que este fue diseñado para ser la base de todo un conjunto de módulos aislados, que a final de cuentas, permitirán al núcleo el enrutamiento de llamadas entre nodos finales.

Dejando de lado el hardware, el núcleo de Asterisk ha mejorado con cada versión que se libera,aumentando las prestaciones totales que manejaráel servidor, como se puede observar con los siguientes valores estadísticos:

- Asterisk 1.2 maneja hasta 220 llamadasSIP concurrentes[13].
- Asterisk 1.4 trabaja con el doble de llamadas (440 llamadas SIP aproximadamente)
- Asterisk 1.6 uso de HASH mejora comportamiento de llamadas SIP de 1-4 a 3-4.
- Asterisk 11 mejora el manejo de llamadas SIP bajo UDP 2-3 veces más rápido que la versión 1.8.

4.2.3 Códec

Los códec, palabra que viene de los términos codificar/decodificar, son aquellas funciones algorítmicas que se encargan de convertir señales de voz analógicas a versiones digitales codificadas.Algunos códec comprimen la voz para un menor ancho de banda[13]. Dependiendo de la versión, Asterisk soporta los siguientes formatos de códec en su core:

- G.711 u-law (Canadá, Estados Unidos y Japón) y a-law (resto del mundo): Códec fundamental de la telefonía tradicional que transmite palabras de 8 bits a una velocidad de 8000 veces por segundo.
- G.722: Códec estándar desarrollado por la ITU-T que mejora notablemente la calidad de audio usando el mismo espacio que el G.711.
- G.723.1: Códec propietario especializado para voz, que posee la capacidad de transmitir a 5.3 ó 6.3kbps.
- G.726: Códec sucesor del G.721-G.723, conocido como el códec comprimido original, trabajando con modulaciones de 16,24 y 32kbps. Asterisk soporta trabajar solamente con 32kbps. Conocido también como ADPCM.
- G.726: Códec estándar creado por la ITU-T que permite la transmisión de voz a 4 diferentes velocidades, tales como 16, 24,32 y 40kbps.
- G.729: Códec propietario que es sumamente conocido por la gran calidad de sonido y poco ancho de banda que utiliza. Debido a lo anteriormente mencionado, requiere gran cantidad de esfuerzo del CPU.

- GSM: Códec preferido para los usuarios de Asterisk debido a la calidad de audio producida y el ancho de banda utilizado (13kbps) pero principalmente debido a que no necesita licencia.
- iLBC: Códec que trabaja en base a baja calidad de audio y a su vez con bajo consumo de ancho de banda, especialmente atractivo para enlaces de red con altos niveles de pérdida de data.
- LPC10: Códec diseñado para voz basado en la codificación predictiva lineal, éste opera a 2.4kbps.
- Speex: Códec de tasa de bit variable que trabaja desde 2.15 hasta 22.4kbps dependiendo de las necesidades requeridas y las condiciones cambiantes de la red.
- MP3: Códec optimizado para música, especialmente usado en Asterisk para entregar los archivos de Música de Espera a los usuarios finales.

En la Tabla 3.2 se presenta la cantidad de llamadas concurrentes que se puede efectuar en una troncal con los códec más reconocidos[14].

Códec	Velocidad	T1 - Llamadas Concurrentes	Notas
u-law	64 kbps	24	
G.723.1	5.3/6.3 kbps	289/243	Licenciado
G.726	16/24/32/40 kbps	96/64/48/38	
G.729	8 kbps	192	Licenciado
GSM	13 kbps	118	
iLBC	15 kbps	102	
LPC-10	2.5 kbps	614	
Speex	2.15 - 44.2 kbps	714 hasta 34	

Tabla 4.2: Comparación de Códec Usados en Telefonía VoIP

A nivel de video soporta:

- H.261
- H.263
- H.263+
- H.264

A nivel de imagen soporta:

- JPEG
- PNG

4.2.4 Soporte de Protocolos y Plataformas

Asterisk tiene la capacidad de trabajar con una amplia gama de protocolos basados en TDM para el manejo y la transmisión de voz sobre las redes telefónicas tradicionales. Además, es importante recalcar que fue diseñado con la capacidad de interoperar con los

estándares de señalización definidos por excelencia en los sistemas telefónicos tanto de Estados Unidos como de Europa, lo cual beneficia la inversión en nuevas tecnologías de voz y datos ya que existe retro compatibilidad.

El uso de protocolos diseñados para VoIP permiten llevar a un siguiente nivel la transmisión de información, dando como ejemplo IAX2, el cual mejora en gran ámbito el envío de tráfico en cuanto a voz, datos, video y señalización, todo esto, para usar de manera óptima los recursos de red[3].

El proyecto fue desarrollado sobre el sistema operativo GNU/Linux. Debido a los excelentes resultados que ha dado, se han generado compilados para distintas distros, tales como PPC, BSD, Solaris, Mac OS X Jaguar y Windows. Debido a que Digium trabaja sobre Linux, se ofrece un mayor soporte y estabilidad de Asterisk en estas distribuciones.

4.3 Arquitectura de Asterisk

La arquitectura de Asterisk fue concebida de manera simplista. Esencialmente actúa como un ente mediador de diferentes módulos con funcionalidades específicas que pueden activarse y desactivarse a voluntad. De esta forma, puede conectar diferentes tecnologías y aplicaciones telefónicas, creando así un ambiente consistente a las necesidades que deben satisfacerse. El núcleo de Asterisk contiene 5 motoresque interpretan un rol importante y a su vez crítico en la forma que opera el software, ya que en términos generales, la funcionalidad de Asterisk se basa en el de una central telefónica PBX, y por ende debe abstraerse de las diferentes configuraciones y tecnologías que participan a la hora de enrutar una llamada.

Internamente Asterisk se divide en las siguientes 4 APIs [14]:

- Canales: Se encarga de controlar el tipo de conexión por la cual llega el cliente.
- Aplicaciones: Permite el correcto funcionamiento de diferentes módulos de tarea.
- Códec: Carga de módulos y códec que trabajaran sobre los distintos audios.
- Formato de Ficheros: Control del almacenaje de datos en el sistema de archivos.

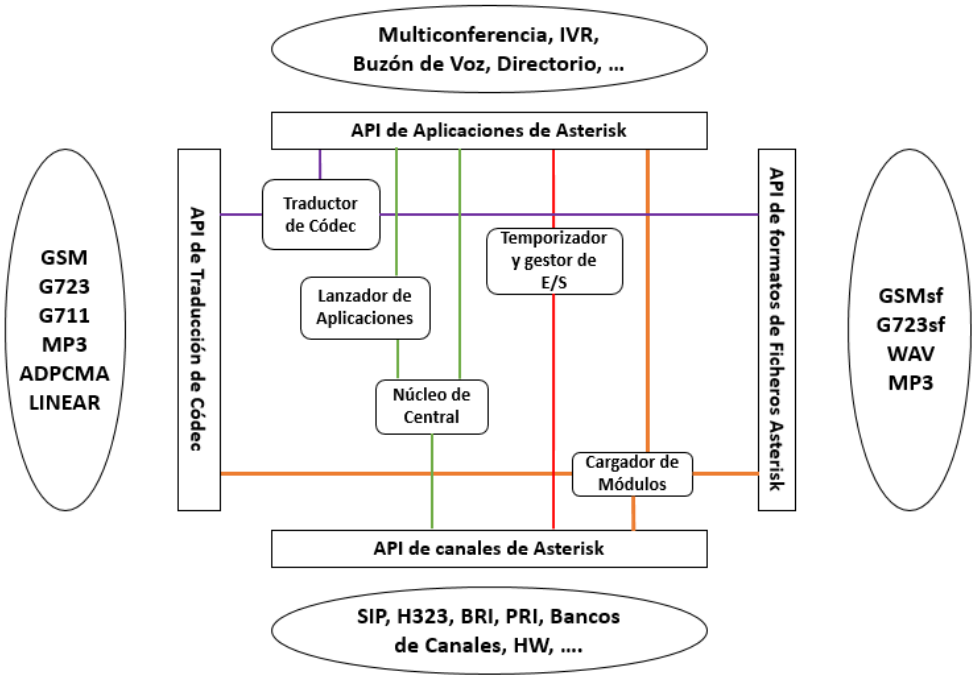

Figura 4.1: Arquitectura de API's de Asterisk

Con la utilización de estos API's como se muestra en la Figura 4.1, Asterisk logra abstracción completa de sus funciones core como servidor PBX de la variedad de tecnologías y funcionalidades existentes o en desarrollo en el área de telefonía. Finalmente, cargar todas las aplicaciones necesarias bajo un módulo, permite al administrador diseñar la mejor trayectoria, bajo el modelo de negocios, para las llamadas entrantes en el sistema PBX, así como modificar las rutas de las mismas para darles sentido y puedan ser redirigidas.

4.3.1 Módulos

Como se menciona previamente, Asterisk se compone de un núcleo centralizado que se encarga de unir diferentes módulos con tareas específicas. La ventaja al ser administrador de la central telefónica, es que se pueden cargar módulos a conveniencia para aumentar de esta forma las capacidades del sistema [14]. Cada módulo, visto en forma lógica, suele ser un archivo ubicado en el directorio por defecto para los módulos de Asterisk, por lo que cada vez que éste inicia carga dichos archivos, aumentando así las prestaciones existentes. Hay diferentes tipos de módulos, tal como se muestran en la Figura 4.2.

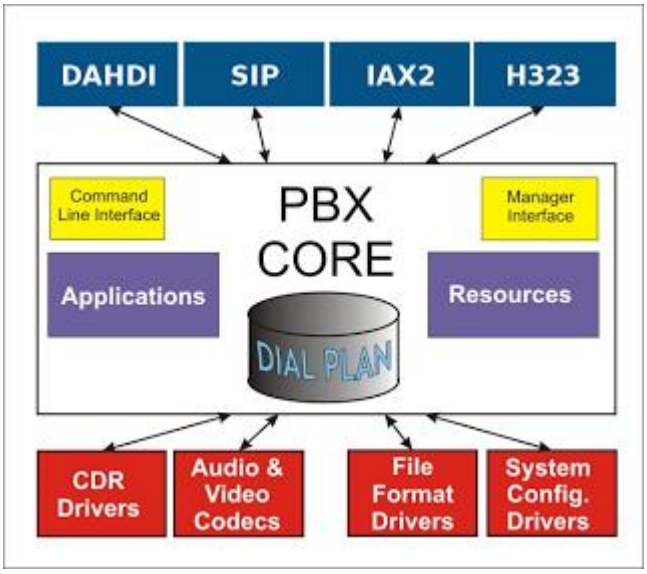

Figura 4.2: Visión Modular de Asterisk

Cada módulo cumple una función específica que mejora las funcionalidades de la central, éstos serán detallados a continuación:

- Canales de drivers: los canales de drivers permiten la comunicación entre el núcleo de Asterisk y los dispositivos externos al mismo. Se encargan principalmente de traducir señalizaciones y protocolos particulares hacia el core. En la Figura 3.2, en la parte superior se indica los canales de drivers.
- Aplicaciones dialplan: las aplicaciones proporcionan funcionalidad al sistema. Una aplicación tiene la capacidad de responder a una llamada, reproducir un sonido del sistema, colgar una llamada, y así sucesivamente.
- Funciones dialplan: las funciones se utilizan para recuperar o establecer diferentes parámetros en una llamada. Por ejemplo, una función puede ser utilizada para establecer el identificador de llamadas en una llamada saliente.
- Recursos: este módulo proporciona recursos para Asterisk. Los ejemplos más comunes incluyen música en espera y estacionamiento de llamadas.
- Códec: módulo para la codificación o decodificación de audio o vídeo. Por lo general se utilizan códec para codificar la data a transmitir a fin de que se necesite menos ancho de banda.
- Drivers de formato de archivo: utilizado para guardar los medios en el disco en un formato de archivo en particular, y para convertir los archivos a flujos de datos multimedia en la red.
- Drivers registro detallado de llamadas (CDR): se encargan de escribir los registros de llamadas en un disco o base de datos.
- Drivers log de eventos de llamadas (CEL): guarda similitud con el CDR, pero maneja más detalles sobre lo que sucedió dentro de Asterisk durante una llamada en particular.
- Drivers bridge: proporcionan diferentes métodos de bridging en los diferentes canales de llamada entre diferentes participantes.

4.3.2 Plan de Discado

Esencialmente, el plan de discado o “dialplan” es un lenguaje de scripting específico y la forma principal de denotar el comportamiento de Asterisk. Se encarga de unir todos y cada uno de los módulos, permitiendo de esta forma enrutar y manipular llamadas en términos de programación [3].

El dialplan, que viene a ser el esqueleto o base de la forma en que trabaja Asterisk, se encuentra en el archivo de configuración “extensions.conf”, el cual puede ser modificado y cargado sin interrumpir el servicio que presta la PBX.

4.3.3 Interfaces

Existen diferentes interfaces que permiten expandir los servicios base que ofrece la central telefónica, se explicarán de manera simple las siguientes [13][3]:

- Asterisk Gateway Interface (AGI): interfaz que se utiliza principalmente para añadir nuevas características a Asterisk mediante el uso de diferentes lenguajes de programación, tales como Perl, PHP, C, Pascal, entre otros. Su función es conectar el dialplan de Asterisk con un programa externo que busque manipular un canal en el plan de discado.
- Asterisk Manager Interface (AMI): interfaz basada en el modelo cliente-servidor a través de TCP (puerto por defecto el 5038), usada para controlar la PBX, ejecutar comandos Asterisk, originar llamadas, revisar estado del buzón, monitorear canales y colas. En otras palabras, permite a programas de usuario conectarse a una instancia de Asterisk de manera tal que le permita enviar comandos o leer eventos asociados a Asterisk usando un flujo TCP/IP.
- Asterisk Calendaring: Interfaz genérica en desarrollo que se basa en el módulo “res_calendar”. Se busca que sea capaz de interoperar con diferentes tecnologías de calendario y sea capaz de soportar la lectura y escritura de eventos y a su vez la notificación de eventos venideros pendientes a través del dialplan de Asterisk.
- Asterisk Call Files: interfaz con la capacidad de iniciar llamadas fuera de los procedimientos comunes, tales como el dialplan y la AMI. Para completar esto, se debe proveer información en cómo realizar la llamada y qué hacer cuando ésta es contestada.
- Simple Network Management Protocol Support: interfaz integrada de forma predefinida en Asterisk que permite el acceso, control y procesamiento rudimentario de tramas SNMP. Usado especialmente por programas de integración para realizar seguimiento del estado de un cliente telefónico dentro de Asterisk y tratarlo en base a reglas personalizables.
- Asterisk RESTful Interface (ARI): interfaz asíncrona creada por los vacíos que se hallaron al momento de usar en ciertos escenarios AGI y AMI. Permite a los desarrolladores crear aplicaciones para la comunicación en Asterisk exponiendo objetos primitivos de Asterisk tales como canales, bridges, endpoints, entre otros, mediante una interfaz REST intuitiva. En la Figura 4.3 se puede observar la visión general de las interfaces.

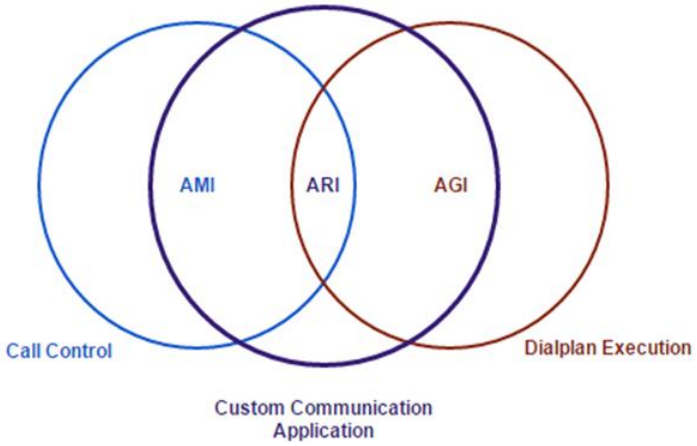

Figura 4.3: Visión General de las Interfaces de Asterisk

4.4 Evolución de Interfaces

Actualmente la interfaz ARI es el nuevo desarrollo en Asterisk para que sea fácil de integrar módulos dentro de él. Cuando Asterisk fue creado en 1999, su diseño estaba enfocado en ser una PBX stand-alone que pudiera configurarse vía archivos de configuración estáticos. Aunque este nivel de configuración es suficiente para muchas aplicaciones, para algunos dominios, es mucho más preferible manipular Asterisk a través de un sistema externo. Por lo tanto se añadieron dos APIs que son Asterisk Gateway Interface (AMI) y Asterisk Manager Interface. Estos dos tienen diferentes propósitos:

- AGI es análogo al CGI en Apache. AGI proporciona una interfaz entre el plan de discado de Asterisk y un programa externo que quiere manipular un canal en el plan de discado. La interfaz es sincrónico, las acciones tomadas en un canal de un bloque de AGI y no regresan hasta que se complete la acción.
- AMI proporciona un mecanismo para controlar donde los canales se ejecutan en el plan de discado. A diferencia de AGI, AMI es una interfaz asíncrona, por eventos. En su mayor parte, AMI no proporciona mecanismos para controlar la ejecución de canal, más bien que proporciona información sobre el estado de los canales y donde los canales están ejecutando.

Ambas interfaces son de gran alcance y ha abierto una amplia gama de posibilidades de integración. El uso de AGI, la ejecución del plan de discado remoto podría estar habilitado, lo que permitió a los desarrolladores controlar los canales en Asterisk usando PHP, Python, Java y otros lenguajes. UsandoAMI, el estado de Asterisk podria mostrarse en pantalla, iniciar llamadas y ubicar canales controlados. El uso de ambas API juntas, podría crear una complejidad en la creación de aplicaciones usando Asterisk como motor para desarrollar.

Sin embargo, mientras que el AMI es bueno en el control de llamadas y AGI es bueno permitiendo a un proceso remoto ejecutar aplicaciones dentro del plan de discado, ninguna de estas API fueron diseñadas para permitir que un desarrollador construya su propia aplicación de comunicaciones personalizadas. Fundamentalmente, ni el API expone los tipos de primitivas de comunicación que existen en Asterisk necesario para construir fácilmente una aplicación y ni ese era su propósito.

Por lo tanto, la comunidad de desarrolladores de Asterisk se propuso construir un mejor API que haría más fácil construir aplicaciones de comunicaciones personalizadas. ARI fue el resultado de este esfuerzo.

4.4.1 Asterisk RESTful Interface (ARI)

ARI permite a los desarrolladores construir aplicaciones de comunicaciones personalizadas en el idioma de su elección. ARI expone las primitivas en Asterisk que normalmente están reservadas para los módulos de C (canales, puentes, puntos finales, medios de comunicación, etc) a través de una interfaz REST intuitiva. ARI transmite el estado de los objetos siendo controlado por el usuario a través de eventos JSON. Al entregar el control de los bloques de construcción fundamentales en Asterisk a todos los desarrolladores, independientemente del lenguaje de programación, Asterisk se convierte en un motor de la comunicación, con la lógica de negocio de cómo las cosas deben comunicar diferido a la aplicación mediante Asterisk.

ARI no es un sustituto de AMI o AGI. Más bien, es una API complementaria:

- AGI permite controlar la ejecución de plan de marcado de un canal.
- AMI permite gestionar las llamadas a un alto nivel.
- ARI permite reemplazar las aplicaciones del plan de discado con su propia aplicación de comunicaciones personalizadas.

4.5 Introducción a Elastix

Elastix es una plataforma de software opensource que tiene como objetivo el incorporar en una única solución todos los medios y alternativas de comunicación existentes en el ámbito empresarial [3].Su funcionalidad está basada en el uso de cuatro programas de software base muy importantes:

- Asterisk
- HylaFAX
- Openfire
- Postfix

los cuales se encargan de brindar funciones de PBX, fax, mensajería instantánea y correo electrónico, respectivamente. Elastix corre sobre el sistema operativo CentOS, actualmente su versión comercial con mayor masificación es Elastix 2.4.0 mientras que en desarrollo se encuentra la versión estable 4.0.0.

4.6 Historia de Elastix

Elastix es desarrollado por la empresa ecuatoriana PaloSanto Solutions y liberado al público en marzo del 2006. Inicialmente fue concebido como una interfaz para mostrar el registro de llamadas en Asterisk y no como una solución de comunicación unificada.

A medida que la demanda de clientes de Asterisk aumenta, la organización PaloSanto Solutions se vio en la obligación de estandarizar su instalación para la implementación de telefonía IP. Con este objetivo se decide elaborar una solución que instale Asterisk partiendo de la instalación previa del sistema operativo sobre un servidor y a su vez agregue diferentes funcionalidades como una interfaz gráfica para administración y configuraciones básicas. Inicialmente deciden lanzar la solución bajo la licencia GPLv2

debido a que fue implementada en varios clientes con mucho éxito, y esto mantuvo el ideal de PaloSanto de apuntar al open source como estructura principal de negocios.

Cuando todo inicio, el número de descargas no fue lo que se esperaba, pero luego de trabajo ensayo y error en busca de mejorar, PaloSanto Solutions lanza una nueva versión incluyendo algunas mejoras lo cual motiva el aumento lineal de personas que deciden empezar a colaborar con el proyecto. Para el año 2007, la comunidad de Elastix crece, lo que se refleja en el incremento del número de descargas y usuarios alrededor del mundo. Se han incorporado varios socios tecnológicos que colaboran en la compatibilidad del software con diferente hardware de telefonía.

En 2010, se lanza Elastix 2.0 incorporando Asterisk 1.6, un módulo de Addons, su propio panel de operador (Elastix Operator Panel), Faxing basado en web, entre otras mejoras y desarrollos. Actualmente Elastix se distribuye como imagen ISO y puede ser descargada desde la página oficial www.elastix.org o desde la página del proyecto en SourceForge.

4.7 Arquitectura de Elastix

La telefonía es el medio convencional que ha liderado durante mucho tiempo las comunicaciones, diferentes organizaciones y clientes enfocan sus necesidades en el simple hecho de establecer un servicio telefónico en su organización lo que presta a confusión en cuanto a distribuciones de comunicaciones unificadas con equipos de hardware destinados a ser centrales telefónicas.

Lo previamente explicado nos devuelve al contexto de que Elastix no se encarga solamente de proveer telefonía, sino que además, integra otros medios de comunicación para hacer eficiente y productivo el entorno de trabajo. Al tener una integración de diferentes sistemas de comunicaciones se logra una mejor productividad laboral en diferentes aspectos como ahorro de tiempo y papel, facilidad en acceso a la información compartida, entre otros. En la Figura 4.4 se muestra las diferentes capas de comunicaciones presentes en el esquema general de Elastix.

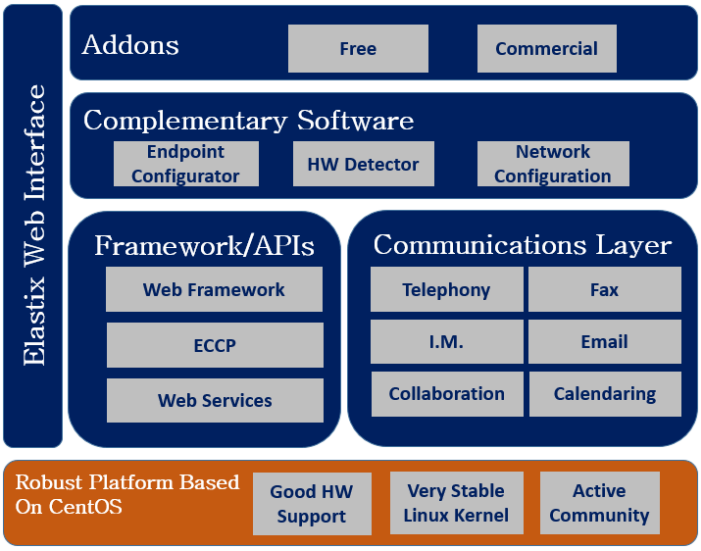

Figura 4.4: Arquitectura de Elastix

4.8 Características de Elastix

Elastix posee múltiples características y funcionalidades relacionadas a los servicios que presta, estas son mejoradas o agregadas a medida que se desarrollan nuevas versiones. A continuación, se listan características y funcionalidades de Elastix [3]:

4.8.1 General

- Ayuda en línea embebida.
- Administración centralizada de actualizaciones.
- Monitor de recursos del sistema.
- Soporte para backup/restore a través de web.
- Configurador de parámetros de red.
- Soporte para temas o skins.
- Control de apagado/re-encendido de la central vía web.
- Soporte para configuración de fechas en el servidor, horas y zonas horarias.
- Control de acceso a la interfaz, basado en ACLs.

4.8.2 Módulo PBX

- Grabación de llamadas.
- Centro de conferencias con salas virtuales.
- Correo de voz.
- Soporte para protocolos SIP e IAX, entre otros.
- Códecs soportados: ADPCM, G.711 (a-law &u-law), G.722, G.723.1 (pass through), G.726, G.728, G.729, GSM, iLBC (opcional) entre otros.
- IVR configurable y flexible.
- Soporte para interfaces análogas como FXS/FXO (PSTN/POTS).
- Soporte para sintetización de voz.
- Soporte para interfaces digitales E1/T1/J1 a través de los protocolos PRI/BRI/R2.
- Herramienta para la creación de extensiones por lote.
- Identificación de llamadas.
- Cancelador de eco integrado.
- Troncalización.
- Provee teléfonos vía web.
- Rutas entrantes y salientes con configuración por coincidencia de patrones de marcado.
- Soporte para videófonos.
- Soporte para follow-me.
- Interfaz de detección de hardware.
- Soporte para grupos de timbrado.
- Servidor DHCP para asignación dinámica de IP.
- Soporte para paging e intercom.
- Panel de operador basado en web.
- Soporte para condiciones de tiempo.
- Reporte de detalle de llamadas (CDR).
- Soporte para DISA (Direct Inward System Access).
- Tarifación con reporte de consumo por destino.
- Soporte para callback.

- Reportes de uso de canales.
- Soporte para interfaces tipo bluetooth a través de teléfonos celulares(chan_mobile).
- Soporte para colas de llamadas.

4.8.3 Módulo Fax

- Servidor fax basado en HylaFAX.
- Personalización de faxes-a-email.
- Visor de faxes integrado con PDFs descargables.
- Control de acceso para clientes de fax.
- Aplicación fax-a-email.
- Puede ser integrada con Winprint HylaFAX.

4.8.4 Módulo de Colaboración

- Calendario integrado con PBX con soporte para recordatorios de voz.
- Libreta telefónica (Phonebook) con capacidad click to call.
- Dos productos CRM integrados a la interfaz (vtigerCR and SugarCRM).
- Web Conference.
- Nuevas características en el Módulo Calendario.

4.8.5 Módulo Mensajería Instantánea

- Servidor de mensajería instantánea basado en openfire.
- Reporte de sesiones de usuarios.
- Inicio de llamadas desde cliente de mensajería.
- Soporte Jabber.
- Servidor de mensajería es configurable desde web.
- Soporte de plugins.
- Soporta grupos de usuarios.
- Soporte LDAP.
- Soporta conexión a otras redes de mensajería como MSN, Yahoo Messenger, GTalk, ICQ.
- Soporta conexiones server-to-server para compartir usuarios.

4.8.6 Módulo Email

- Servidor de Email con soporte multidominio.
- Soporte para cuotas.
- Administración centralizada vía web.
- Soporte antispam.
- Interfaz de configuración de relay.
- Basado en Postfix para un alto volumen de correos.
- Cliente de email basado en web.
- Módulo de SMTP remoto.
- Administración de listas de emails.

4.9 Elastix Call Center Protocol

Se trata de un protocolo de texto basado en XML y especializado para Call Centers. El objetivo es que las aplicaciones “cliente” utilicen este protocolo como protocolo único,

evitando el uso de otras tecnologías como Web Services o AMI (Asterisk Manager Interface).

Una de las ventajas del protocolo es el desempeño debido a que el esquema disponible hasta hoy era basado en el concepto de “polling”, lo cual generaba consultas constantes al servidor y un desperdicio innecesario de recursos del servidor Elastix. El nuevo protocolo soporta la comunicación de eventos asincrónicos, lo cual elimina la necesidad del “polling” y ofrece la posibilidad de una solución escalable.

Este protocolo provee un API de comunicaciones disponible a través de un puerto TCP al cual aplicaciones 'cliente' se pueden conectar con el objetode poder comunicarse con el marcador predictivo que posee Elastix,permitiendo que terceros desarrollen sus propias consolas de agente u otro tipo de aplicaciones 'cliente'.

La actual consola de agente, también desarrollada por PaloSanto Solutions e incluida en el addon “Call Center”, continuará siendo mantenida pero no se añadirán nuevas características. El objetivo es que nuevas y mejores consolas de agente reemplacen a la actual, abriendo un abanico de posibilidades a los usuarios. La mayor parte del esfuerzo de desarrollo de ahora en adelante será dedicado a la interfaz de administración de campañas y al marcador predictivo.

5. Redes Sociales

El fenómeno de las redes sociales ha revolucionado los conceptos de relacionespersonales y del entretenimiento. En ellas se busca mantener el contacto con las personas, crear nuevos vínculos o incluso encontrar oportunidades laborales.

Cada día, millones de personas en todo el mundo utilizan las redes sociales como Facebook y Twitter para expresar sus opiniones. Sus ideas suelen incluir reflexionessobre varias experiencias. Esto representa una valiosa oportunidad para las empresas para conocer a sus clientes de una manera más detallada, además de llegar a una mayor cantidad de personas que no se podría hacer de manera tradicional.

5.1 Antecedentes Históricos

Trazar la historia de las redes sociales no es una tarea fácil, pues su origen es difuso y su evolución acelerada. No existe consenso sobre cuál fue la primera red social, existen diferentes puntos de vista al respecto.

En 1994 se fundó GeoCities, la primera red social que se asemeja a las que conocemos hoy en día. Tenía una estructura muy dinámica y apuntaba a que los usuarios crearan sus propias páginas de Internet y las alojaran en diferentes “barrios”, donde podían así interactuar con otras personas que también se encontraban registradas en el sitio[15][16].

Se dice que Randy Conrads fue el verdadero pionero del servicio mediante la creación del sitio web Classmates, el cual consistía en una red social que brindaba la posibilidad de que las personas de todo el mundo pudieran recuperar o continuar manteniendo contacto con sus antiguos amigos, ya fueran compañeros de colegio, de la universidad o de distintos ámbitos laborales en medio de un mundo totalmente globalizado. Obviamente, esto fue posible gracias a la llegada de la denominada Web 2.0, que al final se trata de un sistema que posee una clara orientación social.

Pocos años después, más precisamente en el 2002, se dio la explosión y el nacimiento de redes sociales que van desde Friendster (2002) y Myspace (2003) hasta Facebook (2004) y Twitter (2006).

Teniendo en cuenta que la historia de las redes sociales no es muy larga, es un hecho que ha abarcado a miles de millones de personas en el mundo. El número de usuarios registrados de Facebook es de 1.650 millones, Youtube cuenta con 1 billón de usuarios, Google Plus cuenta con 540 millones de usuarios, Twitter cuenta con 500 millones de usuarios, Instagram con 300 millones y LinkedIn cuenta con 400 millones de usuarios activos en 2014. Aunque las estadísticas no son muy precisas, ya que se toman en cuenta usuarios registrados y no los usuarios activos o cuentas falsas, se debe tener presente la cantidad de personas que hacen uso de las redes sociales hoy en día.

5.2 Uso Principal de las Redes Sociales

Desde un principio, las redes sociales fueron usadas para conectarse entre personas y al pasar los años fueron convirtiéndose en un sitio donde las empresas pueden interactuar con los clientes y extender sus productos.

Según la clasificación realizada por Del Moral [17], se destacan cuatro usos principales:

- Mantenimiento de amistades: significa seguir en contacto con amigos, colegas o excompañeros de trabajo, conocidos de verano, etc., quienes, de no ser por estos servicios, irían perdiendo relación como ha ocurrido desde tiempos inmemoriales previos al auge de las redes sociales.
- Nueva creación de amistades: Si bien las redes mantienen el contacto entre personas que se conocen, cada una de las personas que participa, relaciona de una forma u otra, a su contactos con segundas o terceras personas, que pueden a su vez interactuar y conocerse. Así el amigo de un amigo, puede llegar a ser contacto y posteriormente amigo de un tercero. Esto converge en la teoría de "Seis grados de separación" de Frigyes Karinthy, el cual apuntaba que no requeriríamos contactar con más de seis personas para encontrar a alguien siguiendo sus redes de amigos y conocidos. Lo que equivale a afirmar que cualesquiera dos personas del planeta están vinculadas, sin saberlo y sólo habría que recurrir a seis "pasos intermedios" para establecer dicha relación.
- Entretenimiento: Aunque las redes sociales sirven para interactuar y acrecentar las relaciones, también hay un perfil de usuarios de éstas que las usa como portal de entretenimiento, sin más pretensiones. Exploran las actualizaciones del estado de algunos usuarios, se ponen al día sobre vidas ajenas, descubren los nuevos colegas de antiguos compañeros de clase, etc. Es el recurso de observar lo que acontece sin ser visto.
- Gestión interna de organizaciones empresariales: sin duda, este uso está circunscrito a empresas dentro de cuya estructura se crean redes sociales privadas para agilizar trámites, comunicaciones, conferencias, informes o se crean otras redes simplemente para poder estar en contacto con profesionales del sector, tanto a nivel laboral como personal.

5.3 Enfoque de las Empresas

Es importante para las empresas saber que las redes sociales no deberían usarse meramente como plataforma publicitaria, sino como un medio para socializar, un lugar de encuentro y de intercambio de experiencias. De hecho, utilizar las redes sociales principalmente como plataforma de ventas, puede generar una percepción negativa de la marca, dada la expectativa social, y no comercial, que supone el término "red social".

Sin embargo, a pesar de no tener un propósito exclusivamente publicitario, son muchos los beneficios que puede generar una correcta estrategia digital en redes sociales, y estos dependerán no solo de la estrategia como tal, sino también de otros factores externos como el sector de la empresa, la actividad comercial, el tamaño de la compañía, la comunidad objetivo y por último, pero no menos importante, el nivel de compromiso de la empresa con la implementación de un plan de marketing digital. Algunos de estos son:

- Posicionamiento: actualmente, a la hora de localizar contenidos, los buscadores tienen muy en cuenta aquellosque se generan en blogs y redes sociales, por lo que su posición ayuda de manera significativa a la visibilidad de la marca.
- Diferenciación: desarrollo de contenidos novedosos, emocionales y propios generan una imagen fresca y conectan a los usuarios y clientes potenciales con la marca.

- Rentabilidad: la implementación de estrategias en redes sociales constituyen una inversión pequeña, nada comparable al costo de los medios tradicionales y masivos.

5.4 APIs de Redes Sociales

Las redes sociales hoy en día poseen información de los usuarios que resultabastante útil para fines específicos como promocionar marcas y servicios. Actualmente muchas aplicaciones están siendo diseñadas de tal forma que tengan la capacidad de establecer una conexión con el API (Application Programming Interface) de dichas redes y obtener datos relevantes para sus funciones.

Para establecer dicha conexión, se debe seguir un proceso de autenticación y autorización de permisos mediante la implementación del protocolo “oAuth” (Open Authentication). Este protocolo permite que un usuario conceda a un tercero el acceso a sus datos, sin tener que proporcionarle su usuario y contraseña.

En dicho proceso y al momento en que el usuario concede permisos a la aplicación, la red social proporciona un “token” que deberá ser guardado por dicha aplicación para poder realizar peticiones en nombre del usuario, tales como leer información personal, intereses, contactos o publicar nueva información.

La interacción entre las redes sociales y la aplicación es realizada mediante peticiones enviadas sobre el protocolo HTTPS (Hypertext Transfer Protocol Secure). Dependiendo de la acción que se desee realizar pueden usarse peticiones de tipo GET, POST,PUT o DELETE. Estas peticiones son análogas a las acciones de leer, escribir, editar y eliminar respectivamente.

Todas las peticiones deben incluir el “token” de acceso por medio del cual se valida la petición,aceptándose o rechazándose de acuerdo con los permisos otorgados por el usuario. Es recomendable que el “token” de acceso sea enviado dentro del encabezado de las peticiones, aunque también puede ser enviado como parámetro en la URL, ya que es protegido por el mismo protocolo HTTPS.

5.4.1 Twitter

Twitter es una de las redes sociales que por su naturaleza no tiene muchas restricciones de privacidad, es decir, la gran mayoría de usuarios tiene un perfil público y por lo tanto también sus “tweets”. Por esta razón Twitter permite el acceso a toda esta información a través de su API [18].

Una de las restricciones con las que se encuentran los desarrolladores es que existen límites de peticiones en un periodo de tiempo, es decir, existen intervalos de 15 minutos en donde se puede hacer un número máximo de peticiones. Cabe mencionar que estos límites son por usuario, no por aplicación, y de esta forma cada usuario es controlado de forma independiente. Dependiendo del tipo de recurso que sea solicitado, existen dos tipos de restricciones principales: 15 peticiones cada 15 minutos y 180 peticiones cada 15 minutos. Adicionalmente, no se permite recuperar información histórica, es decir, si se ejecuta una búsqueda solo es posible obtener información generada siete días atrás como máximo.

En caso de exceder el número máximo de peticiones en un periodo de tiempo, se obtiene un código de respuesta en donde se provee información acerca del recurso restringido temporalmente y el tiempo de espera para que el recurso se encuentre nuevamente disponible. Otra particularidad de Twitter, es que cuenta con streaming, lo que permite obtener información en tiempo real sin restringir el número de peticiones por un periodo de tiempo. La versión actual del API de Twitter es la 1.1.

5.4.2 Facebook

En el caso de Facebook, en vista de todas las restricciones de privacidad que protegen la información del usuario, no es posible acceder a datos protegidos a menos que ambos usuarios tengan una relación de “amistad” en la red social. En caso contrario solo puede obtenerse acceso sin restricción al nombre y al género de los usuarios.Sin embargo, si un usuario da consentimiento explícito a que un tercero acceda a su información de perfil, correo electrónico, gustos, intereses, amigos, eventos, datos de amigos, entre otros más, entonces la aplicación prácticamente no tendrá restricciones para ese usuario en particular [19].

El acceso a los mensajes públicos en Facebookse obtiene mediante su “Open Graph”, que soporta también peticiones utilizando el “Facebook Query Language”. Facebook no tiene documentado el número máximo de peticiones en un periodo de tiempo o si las restricciones son por usuario, aplicación, dirección IP o la combinación de estos. Sin embargo, si continuamente se envían peticiones hacia un recurso específico, con el tiempo se aprecia que se empieza a obtener menos información acerca del mismo.

Facebook cuenta con actualizaciones en tiempo real, lo que permitesubscribirse a ciertos recursos como el muro de un usuario y recibir notificaciones si hay cambios. Esta funcionalidad evita realizar peticiones de forma periódica. Actualmente el Open Graph se encuentra en la versión 2.0 y de acuerdo a la documentación oficial de la misma, ya no será posible acceder a los mensajes públicos que actualmente están disponibles en su versión 1.0.

5.4.3 Google Plus

Google Plus ha luchado para atraer más público desde su lanzamiento, aunque es popular entre los usuarios técnicos. Además Google ha tomado su tiempo parala liberación de una API. De ello se desprende la estructura estándar de todas las APIs de Google, que no siempre es la más clara o más reducida. Se proporciona funcionalidad completa para compartir enlaces, fotos y otros contenidos, así como los SDK para iOS (para la autenticación) [20].

5.5 Mensajería Instantánea

La mensajería instantánea o IM (Instant Messaging) es un servicio de comunicación en tiempo real entre dispositivos como computadores, tabletas, celulares, etc.

Para la mensajería instantánea se emplea un cliente IM que suele mostrar los usuarios conectados y su estado de disponibilidad (de una lista de usuarios que previamente agregó el propietario de la cuenta de mensajería, también llamada lista de contactos). A los usuarios conectados se les puede enviar mensajes de texto, gráficos, sonidos, animaciones, archivos y videos. Algunos permiten compartir recursos y juegos entre

usuarios, y también asociar la cuenta del mensajero a otros servicios como weblogs y servicio de emails.

Los programas de mensajería instantánea son tan populares como el uso de las redes sociales, ya que permite el contacto entre personas de una manera sencilla y rápida en cualquier lugar. Algunos de los programas más populares en mensajería instantánea son:

- Skype: es un software para realizar llamadas o videollamadas sobre Internet (VoIP) y también permite el intercambio de archivos. El código y el protocolo de comunicación usados en Skype no son abiertos, pero los usuarios pueden descargar gratuitamente la aplicación.
- WhatsApp: es una aplicación que ganó popularidad rápidamente y que fue adquirida por Facebook en 2014.
- Google Hangouts: este es un servicio muy completo de comunicaciones que incluye un componente de mensajería instantánea. Incluye al servicio conocido como gtalk (Google Talk) y lo que fue en algún momento la mensajería de Google Plus.

El reinado de clientes como ICQ (se refiere a la frase “I seek you”), MSN Messenger (que fue remplazadopor Skype), Yahoo Messenger o el propio Google Talk (reemplazado por Hangouts) puede haber llegado a su fin.

5.5.1 Protocolos de Mensajería Instantánea

En la mensajería “antigua”, que no se incorporaba necesariamente a los teléfonos móviles, varios servicios se pusieron de acuerdo en utilizar el XMPP (Extensible Messaging and Presence Protocol), también conocido como “Jabber”, que permite la interoperabilidad entre diferentes plataformas de mensajería. De esta manera, fue posible conectarse a múltiples servicios desde una sola aplicación o cliente. Así no era necesario recordar qué aplicación usa qué amigo para poder enviarle un mensaje, ya que tienes todas las cuentas unificadas en un solo lugar vía Trillian, Pidgin u otra opción.

Hubo otros protocolos como “Oscar”,el protocolo oficial del programa de mensajería instantánea de AOL (America Online) y AIM (America-On-Line Instant Messenger). Este es un desarrollo propietario y no ofrece ninguna documentación ni da acceso a sus fuentes. Por esto, muchos diseñadores de programa con soporte para múltiples plataformas de mensajería han tenido que recurrir a la ingeniería inversa para conocer su forma de actuar y adaptar sus programas para hacerlos compatibles con AIM.

Actualmente no es posible unificar los contactos de WhatsApp, Viber, Line, Kik, Facebook Messenger, BBM, iMessage, Skype o Hangouts en un solo lugar, y nada apunta a que esta situación vaya a cambiar en el futuro cercano.

La irrupción de los móviles es la que ha cambiado el peso en la balanza y ha desalojado a XMPP como estándar en la práctica, tanto por razones técnicas como por motivos “estratégicos” o políticos de las compañías, que buscan encerrar al usuario en un ecosistema controlado. Luego de que Google abandonara el protocolo XMPP al cerrar Gtalk y crear Hangout, la Electronic Frontier Foundation (EFF) indicó que: “Estos cambios representan un giro desde protocolos abiertos a propietarios, y un claro retroceso para muchos usuarios” [21].

6. Trabajos Relacionados

En este capítulo se presentan una serie de trabajos relacionados con la integración de sistemas de comunicación unificado con las redes sociales. Estos trabajos abarcan desde el uso de redes sociales como el de mensajería instantánea en las centrales de última generación basados en Asterisk.

Para realizar algún desarrollo dentro de un servidor de comunicaciones unificadas basado en Asterisk es necesario conocer y manejar el AMI (Asterisk Manager Interface) y el AGI (Asterisk Gateway Interface), estos son canales que permiten la comunicación con software de terceros además de ejecutar procesos internos para manejar el plan de discado, monitorear, construcción de aplicaciones de URA (Unattended Response Automatic), entre otros.

6.1 Integración de Elastix con Gtalk

El trabajo realizado por Hugo Gaibor [22] tiene como objetivo configurar el servicio de mensajería instantánea Gtalk en Elastix para poder comunicarse a través de él. Para realizar la integración hubo una investigación previa sobre el soporte de Elastix con algunos protocolos y servicios, y se descubrió que la versión de Asterisk que viene con Elastix 2.2 trae un soporte de Gtalk compilado que permite una comunicación sencilla con el servicio.

En el proceso de configuración se tuvo que modificar el archivo del protocolo XMPP, también llamado Jabber, en donde se creaba un usuario haciendo uso de una cuenta de Gmail. Luego se debía crear el contexto del usuario creado en el archivo de extensiones personalizadas permitiendo crear una serie de pasos al momento de llamar a la extensión. Por último se agregó al plan de discado y se hicieron pruebas para comprobar el buen funcionamiento del usuario.

Una de las pruebas realizadas por el autor fue ingresar a una de las cuentas de Gmail que fueron configuradas y pudo observar que el otro usuario que se configuró se encontraba conectado, lo que significa que el usuario fue registrado de manera satisfactoria ya que la manera de ver a un usuario conectado es que él haya ingresado a su cuenta Google.

Con el usuario registrado satisfactoriamente, se realizó una llamada desde Gtalk a una extensión que se encontraba en el plan de discado. En la Figura 6.1 se muestra la llamada que se está recibiendo por parte del usuario de Google.

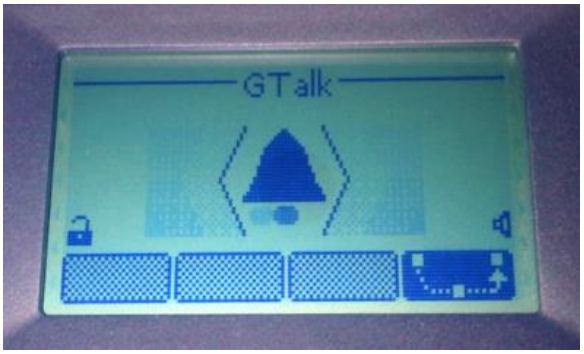

Figura 6.1: Llamada Recibida desde una Cuenta de Google

El resultado obtenido por Hugo Gaibor fue satisfactorio al poder llamar con una extensión a la cuenta de Gmail asociada. El procedimiento para realizar la integración fue sencillo ya que la versión de Elastix que fue usado traía embebido un soporte de Gtalk.

6.2 Publicación en Twitter mediante ASR

Para poder realizar publicaciones en Twitter es necesario usar tanto el API de Twitter como el AGI de Asterisk y así realizar la conexión. Asterisk puede trabajar con varios lenguajes de programación para poder crear funciones personalizadas.

El trabajo realizado por Scoot Smith [23] trata de poder escribir un tweet en una cuenta personal de Twitter, llamando a través de una extensión y comentar lo que se quiera escribir haciendo uso de un ASR (Automatic Speech Recognition) sin costo alguno.Para comenzar se necesita una biblioteca para poder trabajar con PHP a través de AGI, llamada PHP-AGI. Para este trabajo se utilizó la versión 2.20; es posible que para versiones superiores no funcione el script que se realizó. Por otro lado, se necesita un script en PHP, específicamente creado para “operar” con Twitter.

También se requiere un conversor de WAV (Waveform Audio Format)a FLAC (Free Lossless Audio Codec). La diferencia es que WAV es un audio digital que no tiene compresión de datos, mientras que FLAC permite que el audio digital sea comprimido de tal manera que el tamaño del archivo se reduce sin que se pierda información. La conversión es necesaria, ya que se usa un servicio de ASR gratuito de Google que solo recibe formatos FLAC. Para poder pasar de WAV a FLAC se usa el programa“SoX”, una herramienta que sirve para convertir y editar archivos de audio, añadir efectos y otras funcionalidades avanzadas de manipulación de sonido desde el terminal [24].

Por último se necesita crear y autorizar una aplicación en Twitter para que el script pueda funcionar.Para la aplicación se deben dar permisos de lectura, escritura y acceso a los mensajes directos para poder manipular la cuenta. Las informaciones más importantes que se deben obtener de la aplicación son: el consumer key, el consumer secret, el access token y el access token secret.

Cuando se tiene todos los requerimientos anteriores es posible realizar el script que se ejecutará desde la aplicación AGI en el plan de discado. Luego se crea una extensión que graba lo dicho por el usuario y que ejecuta el script anterior. Cuando el usuario realice la llamada, hable y finalice, será posible entrar a la cuenta de Twitter configurada y observar el tweet recién creado con el mensaje dicho por él.

6.3 Conclusión de Trabajos Relacionados

Una vez realizada la investigación respectiva, solamente se pudo encontrar los trabajos relaciones expuestos anteriormente, ya que no existe un estudio detallado acerca de la integración de redes sociales con sistemas de comunicaciones unificados. Además, dichas investigaciones fueron realizadas por la motivación de los autores de juntar la tecnología de VoIP con las redes sociales que usan en el día a día, para así tener otra vía de comunicación.

Después de haber mencionado los trabajos anteriores, se puede apreciar que para el caso de centrales telefónicas de última generación, como es el caso de Asterisk, se tiene una amplia gama de integraciones a través de protocolos y herramientas prediseñados para tal fin, como lo son el AGI que permite añadir funcionalidades a Asterisk con

diferentes lenguajes de programación permitiendo una mayor libertad en la integración de nuevas funcionalidades, el AMI que permite a un programa cliente conectarse a una instancia de Asterisk permitiendo ejecutar comandos o leer eventos sobre TCP/IP y por último también se encuentran archivos de configuración donde se tienen definidos otros protocolos para la comunicación entre diferentes módulos de Asterisk.

7. Marco Metodológico

En este capítulo se define la metodología que permitirá cubrir los objetivos y problemas que fueron expuestos en el Capítulo 2. Por lo tanto, se hace necesario definir un esquema de trabajo que permita desarrollar la solución de forma estructurada y planificada, con la finalidad de integrar la red social Twitter en un servidor de comunicaciones unificadas basado en Asterisk.

7.1 Adaptación de la Metodología de Desarrollo

A continuación se definen los lineamientos necesarios para realizar la integración de la red social Twitter en un servidor de comunicaciones unificadas basado en Asterisk, los cuales, serán realizados de manera ordenada para permitir alcanzar las metas establecidas. Lineamientos:

- Diseño general de la implementación.
- Implementación de ambientes virtualizados.
- Esquematización y diseño de los módulos.
- Verificación, demostración y análisis de resultados.

7.2 Diseño General de la Implementación

Como prerrequisito para implementar la solución de integración, se debe realizar el diseño de la arquitectura de la misma, tomando en cuenta diferentes factores como el ambiente en la que será implementada, hasta los componentes inmersos que formarán parte de la arquitectura. El proceso de creación permite definir:

- **Arquitectura Planteada:**La observación de los componentes que estarán integrándose en el desarrollo de la arquitectura permite definir las mejores estrategias para encontrar una solución independiente y escalable en los componentes involucrados.
- **Requerimientos de la Arquitectura:** Los requerimientos que se desprenden del planteamiento anterior, pueden ser a nivel de red, hardware, ambientes de instalación e incluso requerimientos de recursos y tráfico de red, asegurando así el buen funcionamiento de las distintas herramientas participantes en la solución final.
- **Requerimientos Funcionales**: Luego de establecer la arquitectura, se definen los roles que cumplirán cada uno de los módulos y la forma en la que se comunican.

7.3 Implementación de Ambientes Virtualizados

Una vez realizado el análisis de requerimientos de la solución, por limitaciones físicas de aprovisionamiento y hardware, se opta por la utilización de ambientes virtualizados en un servidor que implementa el hipervisor Xen Project. Este permitirá la creación de distintas máquinas virtuales para la instalación y configuración de las distintas herramientas requeridas por la solución. La concesión de este tipo de ambiente permite destacar la independencia inter módulos deseada.

7.4 Esquematización y Diseño de los Módulos

Entre los objetivos de este trabajo se hace referencia a las funciones de los módulos que estarán involucrados en la arquitectura, dichas funcionalidades son definidas por las

tareas específicas que ha de cumplir cada uno para mantener la independencia de los recursos involucrados, definiendo así, el flujo de comunicación que a final de cuentas permitirá dar respuesta al servicio. Los módulos son:

- **Módulo de Procesamiento**: el objetivo de este módulo es habilitar una comunicación efectiva con la red social Twitter mediante la utilización de su API, tanto para obtener datos como enviarlos de vuelta. De igual forma, permite clasificar los datos obtenidos mediante una fase de procesamiento para que funcionen como insumo del módulo social que existe dentro del servidor de comunicaciones unificadas. Para que la operación sea independiente y se involucre cualquier sistema, se implementan servicios web basados en REST, facilitando así la comunicación y reduciendo los recursos a nivel de cómputo.
- **Módulo Social**: este módulo existe en el servidor de comunicaciones unificadas y, a través de una capa de servicios, se encargará del manejo de los distintos flujos de trabajo que se generan con los mensajes recibidos desde el módulo de procesamiento. De igual forma, contará con persistencia para gestionar las funcionalidades internas de atención al cliente y manejar procesos que involucran a los agentes del módulo de Call Center.

La intercomunicación pasa a ser un punto importante, ya que el módulo de procesamiento necesita información del negocio presente en el servidor de comunicaciones unificadas, con el fin de poder procesar los mensajes de manera correcta para su posterior tratamiento.

7.5 Verificación y Análisis de Resultados

Basado en la arquitectura que se desea implementar, es importante verificar el comportamiento de los módulos que se mencionan a lo largo de este capítulo, por esto, es requerido el análisis de resultados en los siguientes aspectos:

- **Tiempos de Respuesta:**en el proceso de atención al cliente, es importante que los tiempos de respuestas sean aceptables, este análisis depende también de los servicios presentes en los módulos involucrados en la arquitectura.
- **Pruebas de Estrés:** la idea de separar los módulos es no sobrecargar el servidor con las solicitudes que se realizan hacia la red social, el procesamiento y las llamadas dentro del servidor de comunicaciones unificadas. Es importante realizar pruebas en distintos escenarios y flujos de mensajes, que coloquen a prueba los módulos desde el punto de vista de tiempo de respuesta.

8. Marco Tecnológico

En el desarrollo de la integración del servidor de comunicaciones unificadas con las redes sociales se utilizaron diferentes herramientas de software de código abierto disponibles en la actualidad. En estecapítulo se describe cada una de ellas, para dar una idea general de su funcionamiento y ventajas.

8.1 REST

Se refiere a un estilo de arquitectura de la web denominado REpresentational State Transfer. Un modelo basado en las interacciones comunes en las aplicaciones web promedio. Jakl [25] define REST como un conjunto coordinado de restricciones de arquitectura que intenta minimizar la latencia y cantidad de comunicación en la red mientras maximiza la independencia y escalabilidad de los componentes implementados. REST permite la reutilización, sustitución dinámica de componentes y procesamiento de acciones por intermediarios, de modo de cumplir con las necesidades de los sistemas distribuidos de hipermedia.

8.2 Ruby on Rails

Rails es un framework para el desarrollo de aplicaciones web escritas en el lenguaje de programación Ruby, software libre por naturaleza y basadas en el patrón de diseño Modelo Vista Controlador (MVC). Está diseñado para hacer que la programación de aplicaciones web sea más fácil. Permite escribir un buen código evitando las repeticiones y favoreciendo la convención sobre la configuración [29].

Ruby on Rails utiliza como elementos fundamentales las gemas. Una gema es la manera en que Ruby permite distribuir programas, módulos o librerías que extienden funcionalidad, casi siempre específica, y que hacen más fácil el flujo de desarrollo.

8.3 Protocolo AMQP

El estándar AMQP(Advanced Message Queuing Protocol) es un protocolo de estándar abierto existente en distintos sistemas de comunicaciones, específicamente a nivel de la capa de aplicación. Permite crear una interoperabilidad funcional entre diferentes clientes y servidores intermedios de mensajería (middleware).

AMQP es un protocolo binario con una amplia gama de características modernas que lo hacen efectivo en su tarea, tales como:

- Multi-canal
- Negociación
- Asíncrono
- Seguro
- Portable
- Neutral
- Eficiente.

Define tanto el protocolo de red como los servicios que se proveen desde el servidor; este último se encarga de ofrecer distintas capacidades de mensajería denominadas Modelo AMQ, el cual consiste en la definición de diferentes elementos y reglas que permiten la

intercomunicación entre componentes, además del enrutamiento y almacenamiento de mensajes dentro del sistema de mensajería.

Un punto importante es que se encarga de toda la comunicación entre los clientes que inicien una comunicación con el servidor, interactuando con el modelo AMQ que este último implementa.

La semántica asociada al modelo AMQ es establecida de forma explícita en el servidor, dado que se busca interoperabilidad en cualquier servidor que implemente este modelo. Existen tres (3) componentes principales que definen los elementos y las reglas de interconexión entre los mismos para proveer la funcionalidad deseada, se listan a continuación:

- Exchange (Distribuidor): Se encarga de recibir los mensajes de las aplicaciones productoras y enrutarlos a una cola de mensajes basado en un criterio arbitrario, que usualmente son propiedades de los mensajes o contenido del mismo.
- Message Queue (Cola de mensajes): Almacena los mensajes hasta que puedan ser procesados por una o múltiples aplicaciones definidas como consumidor.
- Binding (Enlace): Define la relación existente entre las colas de mensajes y los exchanges, a su vez proveen el parámetro necesario para enrutar el mensaje.

8.4 RabbitMQ

Para el encolado y manejo intermedio de los mensajes entre los módulos desarrollados, se utiliza el bróker de mensajería RabbitMQ, el cual, es un software de negociación de mensajes de código abierto desarrollado por Rabbit Technologies Ltd., usando el lenguaje de programación Erlang. Este implementa el modelo Productor-Consumidor y utiliza el protocolo AMQP (Advanced Message Queuing Protocol) para el intercambio de mensajes [27].

Dado que RabbitMQ implementa AMQP, dispone de los mismos componentes para el manejo de mensajes, los cuales son:

- Colas: Entidades que almacenan los mensajes que se reciben de los productores. Poseen diferentes configuraciones en base a parámetros tales como nombre, persistencia, TTL, entre otros.
- Exchanges: Entidades AMQP que definen el algoritmo de enrutamiento de los mensajes hacia las colas. Existen 4 tipos Direct, Fanout, Topic y Headers.
- Enlaces/Bindings: Relaciones establecidas entre las colas y los exchanges, pueden poseer parámetros conocidos como Routing Key.

Además, provee las siguientes características:

- Servidor de intercambio.
- Validación, transformación y enrutamiento de mensajes.
- Ejecución distribuida y conmutación ante errores gracias al framework OTP (Open Telecom Platform).
- Soporte de múltiples protocolos de mensaje (HTTP, XMPP y STOMP).
- Variedad en bibliotecas cliente y conectores para distintos lenguajes de programación.
- Plugins para extender las funcionalidades actuales del sistema.

- UI para gestión de prestaciones tales como colas, exchanges, mensajes, usuarios, entre otros.
- Logs de sistema para facilitar la administración y depuración de errores.

8.4 Silex

Silex es un micro-framework desarrollado con el lenguaje de programación PHP, utiliza los mismos principios que Symfony y Pimple. Su creación fue inspirada en el micro-framework Sinatra de Ruby. Se implementó con la meta de exponer una API intuitiva, concisa y extensible, que es fácilmente acoplable a otros sistemas que sigan las especificaciones y códigos que rigen el protocolo HTTP. Se utiliza para exponer los servicios web que serán brindados por el módulo social hacia el módulo de procesamiento. El funcionamiento básico de Silex consiste en definir controladores y asociarlos con rutas, todo en un solo paso [26].

8.5 JSON

JSON (JavaScript Object Notation) es un formato ligero de intercambio de datos. Leerlo y escribirlo es simple para humanos, mientras que para las máquinas es simple interpretarlo y generarlo. JSON es un formato de texto que es completamente independiente del lenguaje pero utiliza convenciones que son ampliamente conocidas por los programadores de la familia de lenguajes C, incluyendo C, C++, C#, Java, JavaScript, Perl, Python, y muchos otros. Estas propiedades hacen que JSON sea un lenguaje ideal para el intercambio de datos [28].

Este formato está constituido por dos estructuras:

- Una colección de pares de nombre/valor: en varios lenguajes de programación esto es conocido como un objeto, registro, estructura, diccionario, tabla hash, lista de claves o un arreglo asociativo.
- Una lista ordenada de valores: En la mayoría de los lenguajes, esto se implementa como arreglos, vectores, listas o secuencias.

Estas son estructuras universales; virtualmente todos los lenguajes de programación las soportan de una forma u otra. Es razonable que un formato de intercambio de datos que es independiente del lenguaje de programación se base en estas estructuras.

9. Integración del servidor de comunicaciones unificadas con Twitter

En este capítulo se presentan las herramientas de software que fueron utilizadas para la integración de las redes sociales con el servidor de comunicaciones unificadas. También se explica el diseño de la arquitectura del sistema, el alcance propuesto para las funcionalidades y el flujo de la aplicación final. Finalmente, se expone de forma detallada los pasos que se siguieron para la implementación del sistema.

9.1 Diseño General de la Solución

En la Sección 7.1, se comenta acerca de la importancia de determinar las consideraciones para el diseño de la arquitectura a implementar, y para cumplir los objetivos se crea el diseño mostrado en la Figura 9.1.

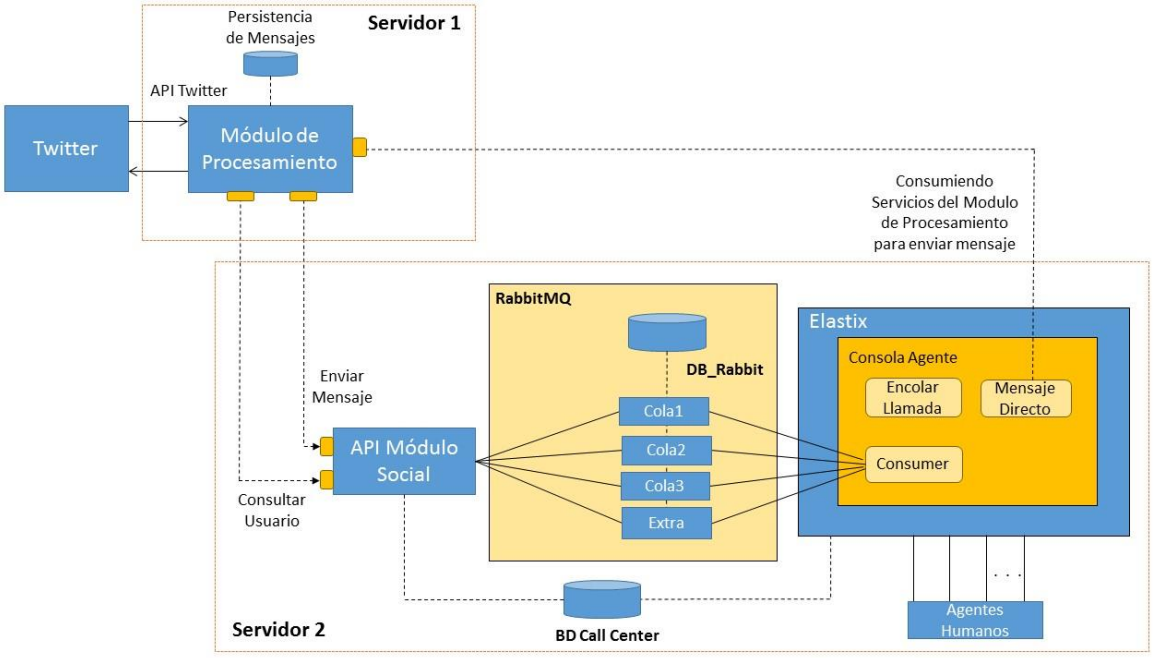

Figura 9.1: Arquitectura General de la Solución

La Figura 9.1 refleja la arquitectura funcional implementada. En dicha arquitectura se separan los módulos en distintos servidores y los elementos inmersos en ella. A continuación se listan los mismos:

- **Twitter**: componente de software intermediario en las peticiones de los usuarios, encargado de recibir las solicitudes y a su vez, publicar las respuestas asociadas a ellas. La interacción con el API de Twitter se basa en contar con una cuenta Twitter con permisos para que una aplicación pueda leer, escribir y acceder a los mensajes directos. Los mensajes directos forman el dato de intercambio a proveer por este componente, debido a la privacidad asociada al mismo; comunicación bidireccional exclusiva entre el emisor y el receptor, además de no tener límite de caracteres en la composición de los mensajes.

- **Módulo de Procesamiento**: encargado de usar los servicios del API de Twitter para obtener y enviar mensajes directos. En el proceso de clasificación de los mensajes, es necesario mantener la persistencia de los datos debido al flujo asíncrono que existe en la entrega de los mismos hacia el módulo social, de tal forma que se pueda etiquetar los estados que tienen el mensaje en el proceso de resolver la incidencia.
- **Módulo Social**: componente responsable de las operaciones sobre los mensajes recibidos por el módulo de procesamiento, además de brindar herramientas de apoyo a la gestión de los centros de contacto en lo referente a su vínculo con las redes sociales. Su estructura se enmarca dentro del ecosistema del servidor de comunicaciones unificadas, donde mantiene dependencia con el resto de componentes que lo integran. El Módulo Social en su ámbito operativo mantiene un manejo de colas de atención y esquema de distribución de mensajes hacia los agentes, permitiendo que éstos apoyen su gestión con la generación de llamadas, envío y recepción de mensajes directos, administración de campañas salientes a redes sociales, entre otros.

9.2 Implementación de Ambientes Virtualizados

Como fue comentado en la Sección 7.3, se opta por realizar el diseño anteriormente mostrado en un ambiente virtualizado. Para lograr esto, se cuenta con un servidor, con las especificaciones descritas en la Tabla 9.1, donde es montado un hipervisor llamado Xen Project v4.6.

Componente	Descripción
Sistema	ThinkServer TS140 Lenovo
Procesador	Intel(R) Xeon(R) CPU E3-1225 v3 @ 3.20GHz. 4 Nucleos
Memoria	16 GB, 4 slots con 4 Tarjetas de 4 GBs Cada una
Disco Duro	1 TB Western Digital WD10EZEX-00B • 1 GB Partición /boot • 80 GB Partición RAID Autodetect (Ubicación de Dominio 0) • 850 GB Partición RAID Autodetect 1 TB Western Digital WD10EZEX-08M • 1 GB Partición /boot • 80 GB Partición RAID Autodetect (Ubicación de Dominio 0) • 850 GB Partición RAID Autodetect
Interfaz de Red	eth0: Ethernet Connection I217-LM 1Gbit/s

Tabla 9.1: Especificaciones del Servidor Huésped de la Arquitectura

Xen Project es un hipervisor tipo 1 que permite la virtualización de distintas instancias de máquinas de manera paralela en una misma máquina (en este caso, el servidor provisto por la empresa). Para lograr esta autonomía de distintas instancias corriendo distintos sistemas operativos gestionados por una misma máquina, Xen divide la gestión de su sistema en distintos dominios:

- Dominio 0**:** el dominio de control principal y el que se encarga de la gestión del resto de los dominios y las máquinas virtuales creadas en estos
- Otros Dominios**:** En estos dominios es donde se encuentran las máquinas virtuales siendo gestionadas por los mismos. Dichos dominios tienen restricciones de acceso y conocimientos a la información del sistema principal y acceso a los controladores del sistema. Todas las máquinas de la arquitectura se encuentran gestionadas por el Dominio U.

Xen Project maneja la técnica de bridging que consiste en la capacidad del sistema operativo encontrado en el Dominio 0 para la creación de interfaces “bridge” (o interfaces puentes) contra interfaces físicas o loopbacks. Así, las distintas interfaces virtuales creadas por Xen en el Dominio 0 que se mapean con las interfaces de las máquinas virtuales en el Dominio U, pueden comunicarse entre sí mediante el switch virtual generado por la interfaz bridge creada anteriormente, esto permite tener en un solo servidor varias máquinas independientes entre sí.

9.3 Diseño del Módulo de Procesamiento

Este módulo, como se ha explicado anteriormente, permite comunicarse y procesar los mensajes a través del API de Twitter. En la Figura 9.2 se observan las funciones específicas que contiene el módulo.

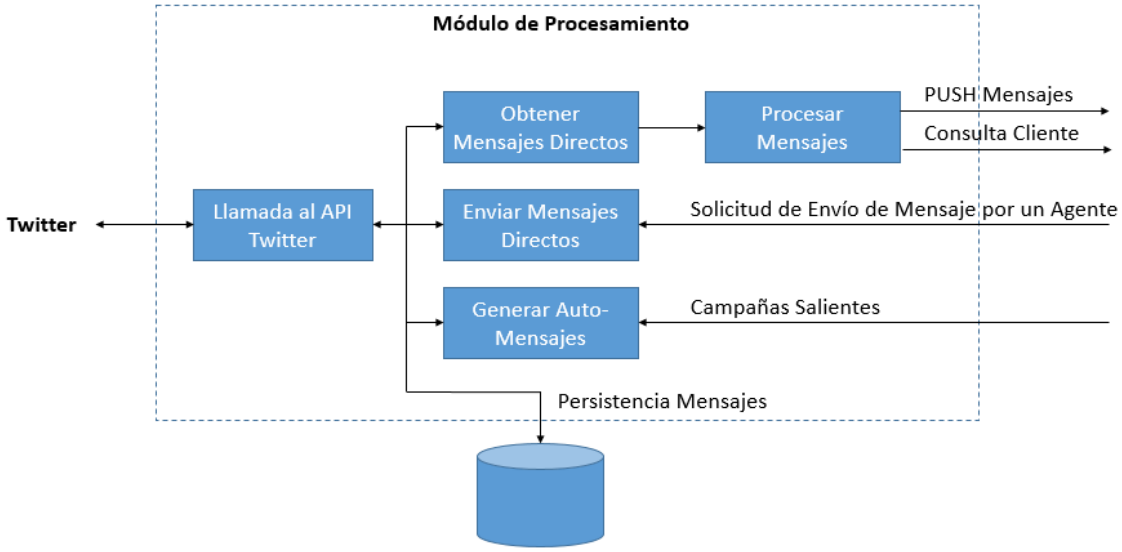

Figura 9.2: Arquitectura Funcional del Módulo de Procesamiento

Este módulo se desarrolla usando el framework Rails por las facilidades y ventajas que ofrece al momento de la programación, también se usa la arquitectura REST y el formato JSON para la transferencia de los datos por su simplicidad y rapidez de procesamiento, además que el API de Twitter también maneja estas tecnologías. A continuación se explica el funcionamiento de los componentes del módulo mostrado en la Figura 9.2.

9.3.1 Llamada al API Twitter

Este componente se encarga de configurar los datos de la cuenta asociada a Twitter para poder realizar las llamadas en nombre del dueño de la cuenta, para ello hace uso de una

gema de Rails llamada “Twitter”. Para realizar la configuración es necesario obtener de la aplicación de Twitter unos valores secretos y únicos llamados: consumer key, consumer secret, access token y access token secret. En la Figura 9.3 se muestra un ejemplo de la configuración haciendo uso de la gema.

```
@twitter_user = Twitter::REST::Client.new do |config|
    config.consumer_key = "clw8ffgmTKLzq9p4CN0BhqBQm"
    config.consumer_secret = "4GD2wO1InKShad4EjlYvYxguItcAaDFi2hZvXYOIejvDks8bok"
    config.access_token = "752602791940612096-kXyygSb3FN9BSDAmCY38rAmgF34BdmK"
    config.access_token_secret = "b3zz9t2kK2igAxKuJ0AyuVHX08meQnJ5meyRSER492Wi9"
end
```

Figura 9.3: Configuración de la Cuenta Twitter en Ruby On Rails

Luego de la configuración y gracias a la gema usada, se puede realizar tweets, enviar y obtener mensajes directos a través de directivas sencillas como:

- **create_direct_message**(<ID_or_UserName>,<Message>): permite enviar un mensaje directo sin límite de caracteres a un usuario colocando su ID o su nombre de usuario.
- **direct_messages**: obtiene todos los mensajes directos de la cuenta configurada.
- **update**(<tweet>): permite actualizar el timeline de la cuenta haciendo un tweet de 140 caracteres.

Estas directivas van a ser usadas en los demás componentes que se ha mostrado en la Figura 9.2.

9.3.2 Obtener Mensajes Directos y Procesar Mensajes

La función de estos componentes es usar la directiva de Twitter para obtener todos los mensajes directos y realizar la clasificación de los mensajes que se obtienen a través de una estructura establecida en los mensajes directos. La clasificación se realiza si un mensaje es por una duda, una queja o una felicitación, y este tiene una estructura dependiendo de esta clasificación como se muestra en la Figura 9.4, donde se observan cuatro estructuras que los usuarios deben usar que se explican a continuación:

```
#id_negocio #id_asociado #tipo_solicitud #servicio_asociado #nro_contacto
Descripción de la Queja

#tipo_solicitud #servicio_asociado #nro_contacto Descripción de la Duda

#tipo_solicitud #servicio_asociado Descripción de la Duda

#tipo_solicitud #servicio_asociado Descripción de la Felicitación
```

Figura 9.4: Estructura de los Mensajes que se Reciben de Twitter

- **Mensaje de Queja**: Son mensajes donde el usuario tiene algún problema con el servicio contratado, lo que implica que es un cliente de la organización. Por lo antes mencionado, en el mensaje se envía un identificador del negocio y uno asociado al contacto (cédula de identidad en este caso). Se debe indicar el tipo de solicitud que en este caso es de queja, el servicio que tiene asociado el cliente, un número de contacto y la descripción breve de su problema.

- **Mensaje de Duda**:Son mensajes en donde el usuario, sin importar si es cliente o no de la organización, pueda enviar sus dudas con respecto a alguno de los servicios que esta ofrece. En este caso existen dos estructuras para las dudas: con teléfono o sin teléfono. De esta manera el agente encargado de responder la solicitud sabe que el cliente prefiere que le respondan mediante un mensaje directo (estructura sin teléfono) o mediante una llamada (estructura con teléfono).
- **Mensaje de Felicitación**: Es un mensaje donde el usuario puede dejar comentarios positivos a la organización por la atención recibida o por la calidad del servicio contratado.

Cuando la estructura de los mensajes no se cumple, se envía una respuesta al usuario indicando que el mensaje no se pudo procesar por un error en el formato. En caso de que sea correcta la estructura y el tipo de mensaje es de queja, se necesita verificar que el usuario se encuentre registrado en la organización, para ello se consulta un servicio web provisto por el módulo social que permite averiguar si un usuario existe en la base de datos.

Luego de pasar todas las verificaciones se construye el mensaje que va a ser enviado al módulo social, para ello se utiliza un servicio web provisto por el módulo socialque permite enviar el mensaje en forma de JSON para que el módulo pueda asignar la incidencia a un agente. La estructura del mensaje que se envía se muestra en la Figura 9.5.


```
{
   id: "47787543878",
   type_msg: "0",
  ▾ user: {
       user_twitter_id: "77783829",
       user_id: "45444878",
       business_id: "0000920",
       phone: "04248890054"
   },
  ▾ message: {
       service: "Central Telefonica",
       content: "No puedo realizar llamadas hacia el exterior",
       date: "2016-01-01 12:02:03"
   }
}
```


Figura 9.5: JSON Enviado al Módulo Social

Los mensajes que se obtienen se guardan en una base de datos y se mantienen las siguientes etiquetas: **waiting**, **delivered**, **finished**. Estas etiquetas corresponden al proceso de recepción del mensaje mediante el API, envío al módulo social para su atención y la respuesta efectiva del agente a la incidencia del usuario.Es importante mantener guardados los mensajes para tener un respaldo en caso de fallar la comunicación y también para generar estadísticas y medir el tiempo de respuesta de los agentes a las incidencias.

9.3.3 Enviar Mensajes Directos

Es un servicio web para ser usado por el agente que permite enviar la respuesta de la incidencia por mensaje directo al usuario. El componente recibe un JSON como se muestra en la Figura 9.6.

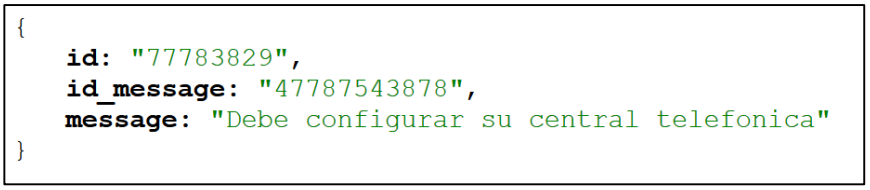

Figura 9.6: JSON que Recibe el Servicio "Enviar Mensaje Directo"

Al momento de utilizar la directiva para enviar un mensaje directo, éste devuelve los mismos datos que son usados durante la llamada a la directiva de obtener mensajes directos, por lo que de igual manera que en el componente que los procesa, se guarda el mensaje enviado para tener constancia de que se ha respondido al usuario.

9.3.4 Generar Auto Mensajes

El componente de generación de auto mensajespermite trabajar con mensajes automáticos y campañas salientes, éste se encuentra dividido en tres partes:

- **Tweets Informativos**: Son tweets que se generan cada cierto tiempo para informar a los clientes sobre la estructura de los mensajes que deben ser enviado en los mensajes directos o cualquier otro tipo de información de la organización para que puedan contactarlos.
- **Campañas Salientes por Tweets**: Es un servicio web que permite que un agente envíe una campaña por lotes para promocionar los productos o servicios de la organización a través de tweets. El formato JSON que recibe se muestra en la Figura 9.7.

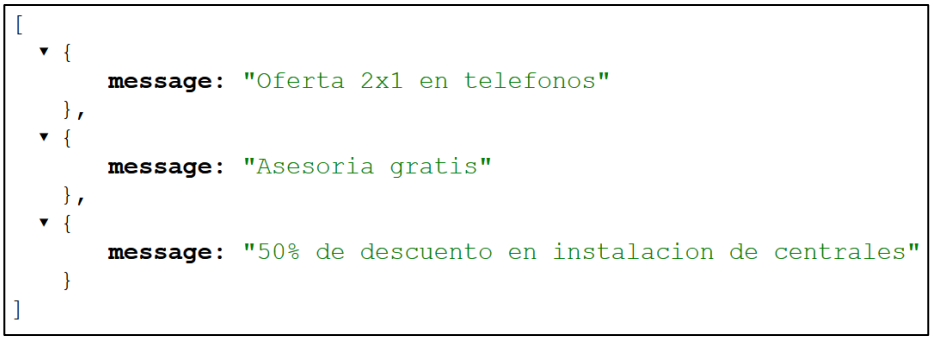

Figura 9.7: JSON que Recibe el Servicio "Campaña Saliente por Tweets"

- **Campañas Salientes por Mensaje Directo**: Es un servicio web que permite que un agente envíe una campaña por lotes para ofrecerle una promoción o nuevos productos a un cliente en específico, la diferencia con el servicio anterior es que en este ocurre por casos especiales en los usuarios, ya sea por el producto que tengan o por el tiempo que tienen usando los productos de la organización. En la Figura 9.8 se muestra un ejemplo del formato aceptado en este servicio.

```
[
  ▾ {
        id: "9989273174",
        message: "Ya son 4 años juntos, recibe gratis tu obsequio"
    },
  ▾ {
        id: "23314784",
        message: "Te ofrecemos un 2x1 en productos telefonicos"
    },
  ▾ {
        id: "6679421244",
        message: "50% en todos nuestros productos"
    }
]
```


Figura 9.8: JSON que Recibe el Servicio Web “Campaña Saliente por Mensaje Directo”

9.4 Diseño del Módulo Social

El módulo social permite la gestión y distribución de los mensajes que se reciben desde el módulo de procesamiento a lo largo del existente sistema interno del servidor de comunicaciones unificadas, específicamente, hasta un agente humano autenticado en el módulo de Call Center. En la Figura 9.9 se observa las estructuras definidas para las funciones específicas que contiene el módulo.

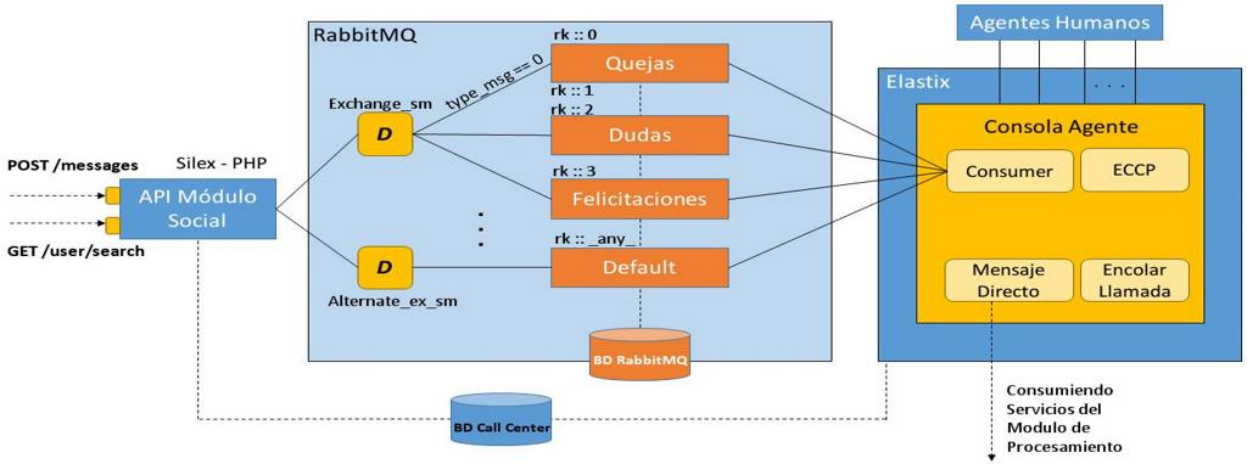

Figura 9.9: Arquitectura Detallada del Módulo Social

Para efectuar una comunicación desde el módulo de procesamiento hacia el core del servidor de comunicaciones unificadas, se opta por manejar un modelo productor-consumidor, en el cual, se utilizaron diferentes tecnologías que permitieron habilitar el pase de mensajes, previamente procesados bajo cierta lógica de negocio desde el módulo de procesamiento, hacia la consola del agente, la cual, es una interfaz del módulo de Call Center que brinda de manera gratuita la solución del servidor de comunicaciones unificadas. A continuación, se explican dichas tecnologías que permitieron construir el módulo social.

9.4.1 API Módulo Social

Dado el acercamiento propuesto para la comunicación entre los módulos, se implementa un API con el microframework PHP Silex, para que funcione como intermediario entre el

módulo de procesamiento y las colas de atención manejadas por el broker de mensajería RabbitMQ, el cual, se encarga de proveer los servicios requeridos para completar el flujo de trabajo de cada mensaje.

Se utiliza una arquitectura dividida en capas, aprovechando las bondades que este método de desarrollo ofrece para unir sistemas abiertos. Se mantiene la visión de separar cada grueso de la funcionalidad en una capa distinta de manera tal que cada modificación no afecte directamente el resto del código que generalmente implica una lógica de negocio distinta. Bajo dicha premisa, la funcionalidad que se busca realzar es mantener un único punto de entrada para cualquier productor de mensajes para el sistema, en pocas palabras, se espera que el generador de datos no será solamente Twitter sino cualquier otro medio tal como sería otra red social, mensajes de texto, entre otros.

Se exponen dos servicios web REST importantes:

- messages: servicio accedido a través de un request POST que permite encolar en RabbitMQ un mensaje previamente procesado y parseado en formato JSON.
- /user/search: servicio accedido a través de un request GET que permite verificar la existencia de un cliente dentro del módulo de Call Center, enviando 2 identificadores del mismo (cédula y contrato).

9.4.2 Colas de Atención

Las colas de atención gestionadas por el módulo Social se soportan en estructuras adicionales a las existentes en Asterisk, dada la particularidad de los atributos que son almacenados en ellas que distan de los manejados en el ámbito de telefonía, es decir, almacenan mensajes procesados por el módulo de procesamiento que serán consumidos por los agentes disponibles. Para esto, se aprovechan las bondades del broker de mensajería RabbitMQ para hacer enrutamiento, persistencia y manejo de los mensajes que son entregados mediante el API del módulo social.

Se utilizan dos (2) enrutadores que hacen la entrega de los mensajes a cuatro (4) distintas estructuras de colas que poseen configuraciones que habilitan funcionalidades específicas de persistencia (durables) y modo de trabajo (ACK waiting). Cada una de estas colas representa un flujo distinto sobre el sistema completo, ya que cada mensaje posee información puntual, es decir, cada una recibe datos específicos dependiendo del tipo de mensaje en el cual fue categorizado por el módulo de procesamiento, dada la solicitud que generó el cliente.

El tipo de mensaje hace correspondencia uno a uno con el routing key que posee cada cola dentro de Rabbit, que en este caso, es un entero del 0 al 3, representando así las quejas, dudas (por llamada o mensajes directos) y las felicitaciones. De igual forma existe una cola extra manejada por un enrutador alterno, el cual se implementó por motivos de futura expansión y para cubrir la pérdida de cualquier mensaje de usuario que sea enrutado con un key route distinto al conocido dentro de la solución.

9.4.3 Módulo de Call Center

El módulo de Call Center del servidor de comunicaciones unificadas está diseñado para manejar campañas de llamadas entrantes y salientes, permitiendo la interacción entre agentes y suscriptores de servicio telefónico. El módulo incluye un marcador predictivo que se encarga de llamar de manera automática a los números “objetivo”. Además incluye

dos componentes importantes: una consola de agente (basada en web) y una interfaz de administración de llamadas [30]. La versión actual incluye soporte para el protocolo ECCP, mencionado en el Capítulo 4, que permite que la operación del módulo sea eficiente y agrega la posibilidad de integrar aplicaciones de terceros.

El módulo del Call Center está implementado alrededor del soporte de colas de Asterisk. El diseño base asume que cada una de las colas alberga al menos a un agente. Cada campaña utiliza exactamente una cola y un agente puede pertenecer a varias colas. Actualmente se reconocen dos modos principales de funcionamiento: Campañas Salientes y Campañas Entrantes.

Para efecto de la integración con redes sociales, los mensajes que se encuentran en las distintas colas de Rabbit corresponden a las campañas entrantes y las campañas salientes se utilizarán al momento de agendar una llamada. En este esquema, el consumidor se encuentra dentro del ambiente del servidor de comunicaciones unificadas, específicamente en la consola del agente, donde se desarrolló un panel donde han de aparecer todas las solicitudes asociadas a Twitter y los usuarios afiliados a la organización.

9.4.4 Inicialización del Ambiente para el Desarrollo

Antes de que se empezara el desarrollo del panel en la consola del agente, se configuraron ciertos elementos para poder iniciar la operación de Call Center siguiendo pasos recomendados por la comunidad de Elastix [31]:

- Creación de grupos de agentes: para utilizar la consola, el agente u operador tiene que ingresar en servidor de comunicaciones unificadas con unas credenciales creadas con anticipación. Por efectos de seguridad y control, es importante crear un grupo que tenga acceso limitado a las funcionalidades que presenta el servidor, por ejemplo: crear una extensión, modificar un archivo o alterar una cola.
- Creación de usuarios: como se menciona en el punto anterior, los agentes u operadores deben tener un usuario creado en servidor de comunicaciones unificadas para poder ingresar a la interfaz y poder utilizar la consola de agente. Una vez que se ha creado el grupo con acceso restringido, se procede a crear los usuarios correspondientes.
- Creación de extensión: cada agente debe tener asignada una extensión durante la operación, por lo cual se crearán tantas extensiones como Agentes operen en una campaña. El módulo de call center transfiere al agente todas las llamadas en una campaña a su extensión si este se encuentra disponible. Para facilitar la operación del Call Center, estas extensiones son de tipo SIP y estarán registradas mediante el softphone Zoiper.
- Creación de agentes: la creación de agentes se diferencia de la creación de usuarios ya que estos son generados dentro del módulo de Call Center, y será mediante estos, que se hará uso de la consola de agente.
- Creación de colas: es necesario crear una cola que haga correspondencia a la campaña entrante que maneja la atención de incidencias, ya que sin una de estas, el agente no puede ingresar a la interfaz de la consola.

Luego de realizar la configuración, se revisó el correcto funcionamiento del módulo de Call Center, ingresando en la consola con un agente y llamando directamente a la cola de

atención designada con un cliente registrado en Zoiper. Se observó que el agente se mantiene en un estado ocioso mientras no recibe llamadas. En el momento que la central le asigna la llamada al agente, se cambia el estado a ocupado. Por otro lado, si el agente se encuentra en un break, la central no le asigna llamadas.

9.4.5 Campaña Saliente

La campaña saliente es un concepto importante para la solución de integración. Se basan en un conjunto de llamadas realizadas por el servidor de comunicaciones unificadas a un listado de abonados previamente incluidos en el sistema, las cuales son asignadas a un agente que se encuentre conectado y disponible durante la duración de la misma [30].

Los contactos son proveídos al sistema con un archivo CSV que contiene los números telefónicos a marcar. Al activarse la campaña, el sistema marca los números telefónicos del listado, verificando el número de agentes libres registrados que se encuentren en la cola.

Esta aproximación trabaja uniendo a la cola asignada a la campaña, el número del abonado al que se quiere contactar, por lo cual, si la llamada es exitosa, entra a la cola y es asignada al siguiente agente libre. El servidor registra en base de datos información de los agentes y las llamadas conectadas a estos, la cual servirá para varios propósitos de consulta y estadísticas. Al terminar la llamada, el agente pasa al estado ocioso, indicando así que está nuevamente disponible para recibir llamadas.

9.4.6 Campaña Entrante

Una campaña entrante es aquella que recibe llamadas de clientes o abonados de manera directa. Por lo tanto, no es necesario tener un listado de números o clientes y la predicción de llamadas no funciona en esta modalidad. Al igual que en las campañas salientes, es necesario definir una cola, en la cual se incluyen los agentes que trabajarán con cada usuario en esta campaña [30].

Se debe asegurar que los abonados que llamen, sean dirigidos a la cola configurada para la campaña, esto se relaciona generalmente al uso de un IVR que dirija al cliente hacia la cola, o con la asignación de un número directo para la misma.

Al igual que en las campañas salientes, el sistema registra en base de datos al agente que le fue asignada la llamada. Esta información servirá para varios propósitos de consulta y estadísticas. Aunque no existe la necesidad de un listado para la generación de llamadas, se pueden proveer datos sobre números específicos para que los agentes tengan información del cliente disponible en la consola.

9.4.7 Breaks

Los breaks permiten que un agente salga de operación y permanezca en espera como no disponible. En este estado el dialer no le asignará llamadas al agente, sean estas por una campaña entrante o saliente. Se utilizan de igual forma en el área de reportes, para determinar la eficiencia de cada agente en el ámbito de atención de llamadas [30].

La integración realizada se orienta a la utilización de los breaks para separar los procesos de gestión existentes en la consola del agente, de tal manera que se puedan diferenciar los flujos de las llamadas, con los mensajes asíncronos que habilita el módulo social. Para lo antes mencionado, se crea en la base de datos del servidor un nuevo break llamado

Social, que será el que indica cuando un agente está atendiendo mensajes de las redes sociales, para que así, este no sea interrumpido por alguna llamada mientras culmina su gestión.

9.4.8 Consola de Agente

La consola de agente administra la interacción del agente con el sistema. Esta es una herramienta basada en Web y actualmente funciona con el ECCP. La consola recibe eventos asociados a una llamada y toda la información del cliente mediante el protocolo ECCP. Por medio de la consola, el agente obtiene información de la operación (número telefónico atendido, información de contacto, scripts de atención, entre otros) [30]. La consola de agente que se encuentra por defecto al instalar el módulo en servidor de comunicaciones unificadas solo posee el panel de llamada, tal como se muestra en la Figura 9.10.

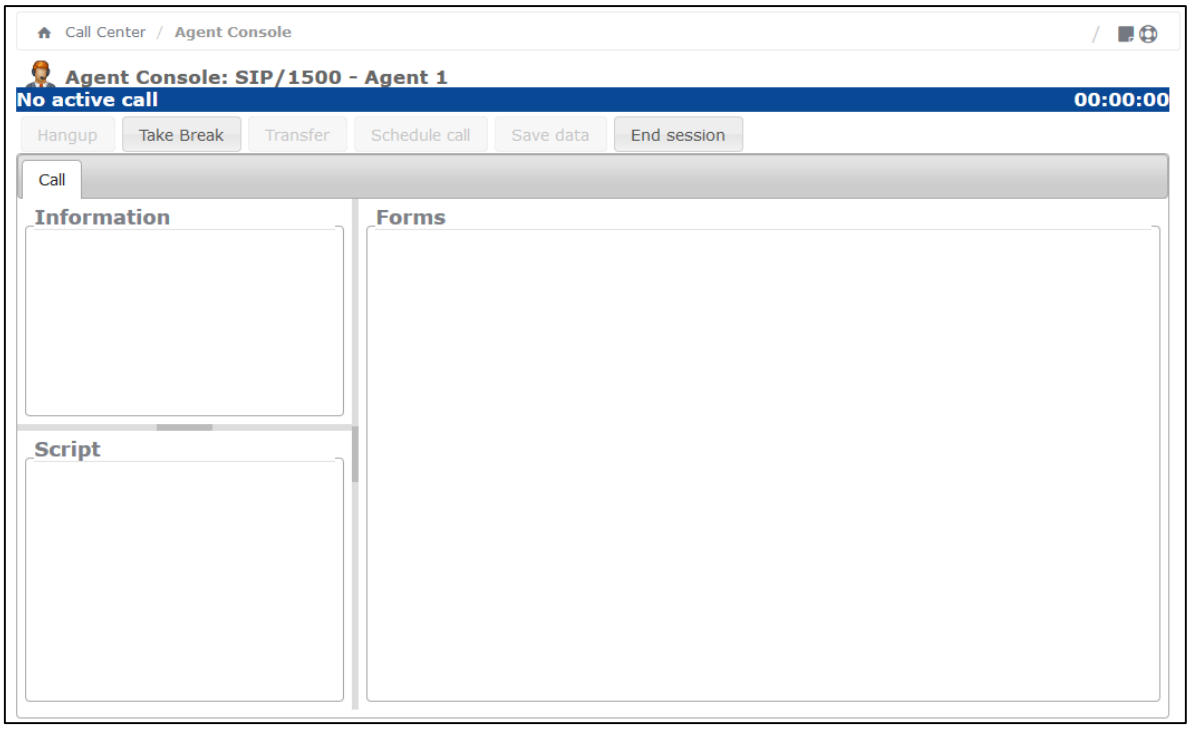

Figura 9.10: Consola de Agente por Defecto

Actualmente la versión 4 de Elastix permite a los desarrolladores insertar paneles personalizados y así, agregar funcionalidades a la consola según las necesidades de la organización. Existe un modelo a seguir para estandarizar el API de estos paneles personalizados con el objeto de minimizar los cambios a realizar en los archivos centrales de la consola de agente [31]:

Organización de archivos: Para agregar un nuevo panel a la consola, se necesita crear una estructura de directorios específicos para que el módulo pueda visualizarlo. Dentro del directorio se debe incluir el archivo “index.php”, el cual contiene los puntos de entrada para implementar la funcionalidad deseada. Además se deben incluir los siguientes directorios dentro del panel, de ser estos necesarios:

- “js”: funciones javascript que deben incluirse para implementar el panel. Todo archivo dentro de este directorio se incluirá al momento de cargarse el módulo.
- “lang”: traducciones de idioma para los textos a usar en el panel. El formato de archivo de traducción es idéntico al de los archivos usados por los módulos ordinarios.
- “tpl”: son archivos de plantillas a usar para el panel.
- Otros directorios u archivos: son ignorados por el código aunque pueden ser incluidos explícitamente.

Funciones y clases: para evitar colisiones entre paneles, el archivo index.php del panel debe de definir una clase cuyo nombre se deriva del nombre de directorio del panel, con la primera letra puesta en mayúscula. Por ejemplo, si el directorio se llama "registro", se espera una clase "Panel_Registro", para efectos de esta implementación se utiliza el sufijo “Social”. Dentro de la clase se pueden definir las funciones siguientes:

- “templateContent”: la función en caso de existir, debe devolver el título y el contenido del panel, y tiene que existir para que el panel aparezca en la consola. En la Figura 9.11 se puede observar la definición que se realizó, en este caso se trabajó con las traducciones del panel y la carga del contenido.

```
static function templateContent($module_name, $smarty, $local_templates_dir, $oPaloConsola, $estado){
    $smarty->assign(array(
        'ID_CLIENTE' => _tr('ID Number'),
        'NOMBRE_CLIENTE' => _tr('Name'),
        'APELLIDO_CLIENTE' => _tr('Last name'),
        'TELEFONO_CLIENTE' => _tr('Phone'),
        'ID_NEGOCIO' => _tr('Business ID'),
        'ID_MENSAJE' => _tr('ID Number'),
        'TELEFONO_MENSAJE' => _tr('Phone'),
        'SERVICIO_MENSAJE' => _tr('Service'),
        'CONTENIDO_MENSAJE' => _tr('Message'),
        'FECHA_MENSAJE' => _tr('Date'),
        'COLA_MENSAJE' => _tr('Queue'),
        'ENVIO_BOTON' => _tr('Send DM'),
        'LLAMADA_BOTON' => _tr('Enable Schedule Call'),
        'CONSUMIR_BOTON' => _tr('Enable Social Media'),
        'NO_CONSUMIR_BOTON' => _tr('Disable Social Media'),
        'INFO_CLIENTE' => _tr("Client's Data"),
        'MENSAJE_CLIENTE' => _tr("Client's Message"),
        'ENVIO_MODAL' => _tr('Send Direct Message'),
        'PLANTILLA_MODAL' => _tr('Default Templates'),
        'OPCION_1_MODAL' => _tr('None'),
        'CONTENIDO_MODAL' => _tr('Message'),
        'BOTON_MODAL' => _tr('Send'),
        'CANCELAR_MODAL' => _tr('Cancel'),
        'ENCOLAR_BOTON' => _tr('Queue Message') ));
    $content = $smarty->fetch("$local_templates_dir/default.tpl");
    return array('title' => "Social", 'content' => $content);
}
```

Figura 9.11: Función del Contenido del Panel

- “handleJSON_ACCION”: para cada operación definida por el panel que requiera acceso al servidor, se puede definir una función llamada handleJSON_ACCION, dónde ACCION se usa para discriminar la operación entre las múltiples existentes. En la Figura 9.12 se muestran las funciones definidas para el panel Social.

```
function handleJSON_devolverMensaje($module_name, $smarty, $local_templates_dir, $oPaloConsola, $estado)
function handleJSON_buscarBreak($module_name, $smarty, $local_templates_dir, $oPaloConsola, $estado)
function handleJSON_agendarLlamada($module_name, $smarty, $local_templates_dir, $oPaloConsola, $estado)
function handleJSON_agregarCliente($module_name, $smarty, $local_templates_dir, $oPaloConsola, $estado)
function handleJSON_buscarCliente($module_name, $smarty, $local_templates_dir, $oPaloConsola, $estado)
function handleJSON_buscarSaliente($module_name, $smarty, $local_templates_dir, $oPaloConsola, $estado)
function handleJSON_mensajeDirecto($module_name, $smarty, $local_templates_dir, $oPaloConsola, $estado)
function handleJSON_tomarMensaje($module_name, $smarty, $local_templates_dir, $oPaloConsola, $estado)
```

Figura 9.12: Funciones Definidas en el Panel Social

- Llamada del Javascript: para construir una petición web que invoque la acción, se debe especificar un parámetro GET o POST llamado "action" cuyo valor sea "nombrepanel_ACCION". En la Figura 9.13 se muestra un ejemplo de una llamada en el javascript a la función del panel social en donde se define el módulo que se está trabajando, el “rawmode” se envía para que no se envuelva el resultado de la función en el HTML de los menús del servidor de comunicaciones unificadas, el parámetro “action” es donde se indica la función que se va a llamar, en este caso se llama a la función “devolverMensaje” del panel social y por último, luego de definir los parámetros principales, se envían los datos para que sean usados dentro de la función.

```
$.post('index.php?menu=' + module_name + '&rawmode=yes', {
    menu:       module_name,
    rawmode:    'yes',
    action:     'social_devolverMensaje',
    data:   $mensaje_json,
},
function (respuesta) {
    if(respuesta['code'] == "Success"){
        mostrar_mensaje_info(respuesta['desc']);
    }else{
        mostrar_mensaje_error(respuesta['desc']);
    }
$mensaje_json = null;
}, 'json')
.fail(function() {
    mostrar_mensaje_error('Failed to connect to server to run request!');
});
```

Figura 9.13: Llamada Javascript a las Funciones del Panel Social

9.4.9 Funcionamiento del Panel Social

Dada la naturaleza de la consola del agente para proveer atención al cliente mediante llamadas telefónicas, en el desarrollo del panel social, se habilitaron diferentes tipos de respuestas de las cuales harán uso los agentes para solucionar las peticiones que obtengan desde la red social, en este caso, se manejan las respuestas a través de mensajes directos al usuario, o se planifica una llamada hacia el mismo.

Debido a que el escenario planteado maneja agentes con diferentes set de habilidades para cada tipo de solicitud, se plantea la creación de campañas específicas solamente para la atención del módulo social, buscando de esta forma no solapar los flujos que existen actualmente en la consola del agente.

En el funcionamiento normal de la consola del agente, para poder iniciar sesión, el agente debe estar asociado a una cola, en la cual estarán las llamadas de los abonados,

permitiendo así realizar labores de atención. En el módulo social, el enfoque para la autenticación del agente en la consola se mantiene, solo que el modelo de trabajo cambia debido a que los mensajes de cada cliente serán gestionados en el broker RabbitMQ y no en las colas internas del servidor de comunicaciones unificadas.

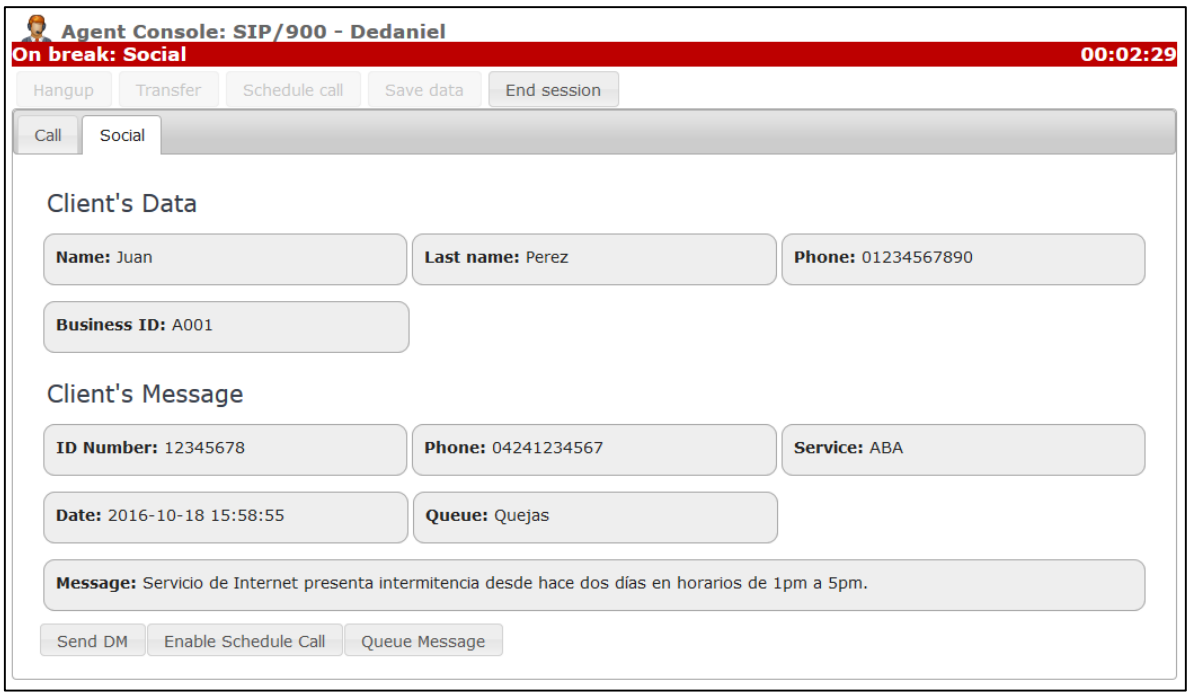

Figura 9.14: Consola del Agente en el Panel Social

Una vez el agente inicia sesión y activa el panel social, se presentan 2 secciones en la interfaz tal como se observa en la Figura 9.14. En la primera sección titulada “Client’s Data”, se despliega toda la información que posee el negocio acerca del contacto que genera la solicitud, sólo si éste existe en la base de datos del servidor. La segunda sección, “Client’s Message”, refleja los datos procesados del mensaje que se obtienen desde la red social a través del módulo de procesamiento, de tal forma que el agente pueda colocarse en contexto en cuanto a la solicitud obtenida y pueda decidir la forma en la que atenderá la incidencia.

En el panel social se tiene el botón “Enable Social Media”, este habilita el proceso de consumo de mensajes hacia RabbitMQ, activarlo desencadena una serie de flujos que se explican a continuación:

1. Instanciación de consumidor de RabbitMQ.
2. Obtención de mensaje desde las colas suscritas por el consumidor.
3. Despliegue de información del mensaje en la consola del agente (Panel social).
4. Cambio de displays de botones en la interfaz.
5. Timers de atención para cada instancia de consolas de agentes activas.

Al desplegarse el mensaje en el panel luego de activar el modo social, se activan botones con las funciones de respuesta mediante mensaje directo, agendar llamada y reencolar mensaje. Dependiendo del tipo de mensaje obtenido, se mostrará un conjunto de botones específicos, dado que cada mensaje puede contener información distinta del usuario, es

decir, para el alcance actual, existen mensajes de usuarios no conocidos, por lo que solo se posee su username en la red social, por lo tanto, no hay forma de contactarlo mediante una llamada telefónica.

El primer botón, “Send DM”, permite desplegar un modal, tal como se observa en la Figura 9.15, donde el agente puede personalizar su respuesta a voluntad dependiendo del requerimiento a tratar. La acción del modal consume directamente el servicio web “Send Direct Message” alojado en el módulo de procesamiento, sirviendo éste de puente, para entregar el mensaje del agente a la red social.

Figura 9.15: Modal de Envío de Mensaje Directo

El funcionamiento concebido para el segundo botón, “Enable Schedule Call”, es activar el agendado de llamadas por defecto que tiene la consola del agente, dejando el trabajo de encolado y asignación de la llamada al dialer, el cual conectará la llamada en el momento que el agente se encuentre en estado disponible, es decir, fuera del modo social. Se realiza una pequeña modificación al modal de agendado del servidor para solo obtener el número a marcar y el nombre del contacto, tal como se observa en la Figura 9.16.

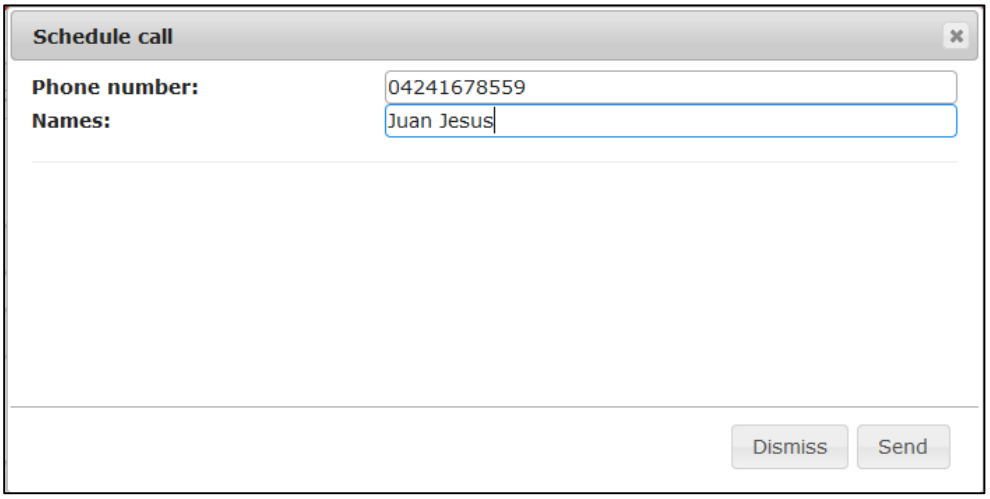

Figura 9.16: Modal de Agendar Llamada

El tercer botón, “Queue Message”, tal como su nombre lo indica, permite al agente devolver el mensaje al broker para que sea atendido posteriormente. Su funcionalidad

está basada en el criterio de que el agente es talento humano y por ello no se puede garantizar que siempre pueda responder el mensaje, sea ya por que no tenga la experticia para ello, sea un problema mayor y deba ser transferido a otro agente con un conjunto diferente de habilidades o deba retirarse por alguna emergencia que deba atender.

Por otro lado, para evitar que un agente esté de manera ociosa con un mensaje, se implementa el uso de dos temporizadores, el primerocumplirá las funciones de alertar al agente que su tiempo para atender la incidencia está próximo a expirar y el segundo activa la acción de retirar el mensaje de la consola para volverlo a encolar al broker de mensajería. El modelo planteado estipula que el temporizador de alerta hacia el agente se activa cuatro minutos después que se obtiene un mensaje en el panel social, y el temporizador de reencolado se activa un minuto después que expira el de alerta, es decir, 5 minutos después de obtener el mensaje.

El modo de operación del panel social es cíclico, al activar el modo social se consume un mensaje de la cola de atención de Rabbit, se atiende el mensaje por el agente y finalmente se da un tiempo intermedio de acción antes de que se vuelva a consumir un mensaje. El tiempo intermedio del que se habla previamente, es un temporizador que cumple la función de permitir la siguiente acción del agente que puede ser, continuar atendiendo incidencias o deshabilitar el modo social.Si al momento de consumir un mensaje, las colas de atención se encuentran vacías, se realiza un segundo intento luego de esperar una cantidad de tiempo definida, si al hacer el segundo consumo las colas continúan sin mensajes, se desactiva el flujo del módulo social.

9.4.10 Estadísticas

Para calcular la productividad de los agentes participantes en las campañas, el servidor de comunicaciones unificadas se apoya en registros contenidos en tablas existentes en la base de datos call_center, las cuales mantienen información de las llamadas conectadas y los breaks tomados por los agentes. El modo de operación del panel social implementa el uso de los breaks, buscando que el agente no sea interrumpido por alguna llamada (entrante/saliente) en caso que esté resolviendo alguna incidencia obtenida desde la red social. Por lo tanto, para el cálculo de estadísticas del módulo social se verifica el tiempo utilizado por los agentes en el break social y a su vez la proporción de mensajes respondidos haciendo uso del servicio web existente en el módulo social.

9.5 Pruebas y Análisis de Resultados

La siguiente sección tiene como objetivo describir el conjunto de pruebas asociadas a rendimiento y carga, que se aplicaron para determinar el nivel de afectación del módulo Social sobre el servidor de comunicaciones. Se busca de esta forma, evaluar y determinar si la implantación de este módulo sobre un ambiente en producción, es factible en cuanto a los tiempos de respuesta que maneja el servidor para no afectar la atención al cliente.

Para realizar las pruebas, se utilizó el framework para pruebas PHPUnit, en este, se ajustan los valores según el escenario de cada prueba dependiendo del requerimiento que se quiere evaluar. En la tabla 9.2 se explicarán cada una de las pruebas realizadas.

ID	Nombre	Descripción
1	Procesamiento de Mensajes Directos	Determinar el impacto que causa la aplicación cuando se procesan los mensajes directos obtenidos en el MP desde el API de Twitter.
2	Envió de Mensajes Directos	Determinar el comportamiento de la aplicación al momento de generar un mensaje directo desde la consola del agente.
3	Obtener Mensaje Directos	Determinar el promedio de tiempo y el impacto que genera obtener el conjunto de los mensajes directos enviados a la cuenta asociada de la aplicación.

Tabla 9.2: Descripción de las Pruebas

9.5.1 Configuración del Módulo de Procesamiento

El servidor donde se instaló Ruby on Rails tiene su configuración por defecto y este viene con un servidor web predeterminado llamado WEBrick. El servidor web viene en una librería estándar que está instalada en todas las máquinas que tienen Ruby, la mayoría de frameworks como Rails y Rack utilizan WEBrick como un servidor web de desarrollo por defecto.

Aunque WEBrick debería estar bien para el desarrollo, no fue diseñado para manejar una alta carga de trabajo concurrente que una aplicación Ruby debe servir en producción. De forma predeterminada WEBrick es un hilo único, de proceso único. Esto significa que si dos solicitudes entran al mismo tiempo, la segunda debe esperar a que la primera termine.

Si bien el servidor es Single Thread, las pruebas se enfocan en el proceso de comunicación de Twitter con el módulo de procesamiento, por lo que los problemas de concurrencia no tienen gran impacto en el tiempo para realizar una petición al API.

9.5.2 Tiempo Procesando Mensajes Directos

La finalidad de esta prueba fue determinar los distintos tiempos de procesamiento que pueden obtenerse desde que un mensaje es obtenido por el módulo de procesamiento hasta que es enviado a su respectiva cola de atención.

- Resultado Esperado: Debido al tipo de petición realizada y la infraestructura de red poseída, se estipulaba un tiempo máximo de dos mil milisegundos (2000 ms) para el procesamiento efectivo del mensaje.
- Datos Utilizados: Se generaron cincuenta (50) mensajes desde la red social hacia las diferentes colas de atención, específicamente veinte (20) asociados a la cola de quejas, veinte (20) a la de dudas y diez (10) a la de felicitaciones.

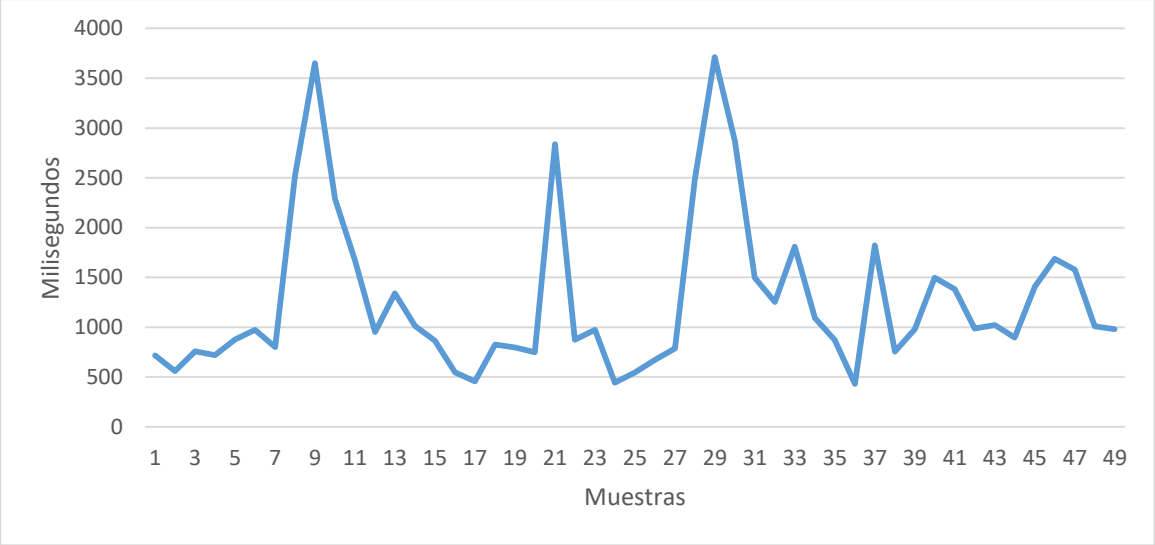

Figura 9.17: Tiempo de Respuesta al Procesar Mensajes Directos

En la Figura 9.17 se observan algunos picos relevantes al momento de procesar los mensajes, a pesar de ello, el promedio en el tiempo de procesamiento se mantiene debajo de los dos mil milisegundos (2000 ms). Dichos comportamientos pueden ser provocados por diferentes variables, tales como:

- Tráfico en la red al momento de obtener los mensajes de Twitter.
- Los picos altos pueden ser provocados por recibir una gran cantidad de mensajes directos que no han sido procesados por el módulo.
- Los picos bajos pueden ser provocados porque gran parte de los mensajes obtenidos han sido procesados previamente por el módulo.

9.5.3 Tiempo Enviando Mensaje Directo

Con esta prueba se buscó determinar el tiempo efectivo para consumir el servicio web presente en el módulo de procesamiento, el cual se encarga de enviar mensajes directos a usuarios de Twitter, específicamente aquellos a los que se les ha respondido desde una instancia de la consola del agente.

- Resultado Esperado: Para consumir el servicio de mensajes directos se esperaba un promedio de mil quinientos milisegundos (1500ms), considerado de esta forma ya que el consumo era directo contra el módulo de procesamiento y por ende no se utilizaban las colas de atención.
- Datos Utilizados: Se generaron respuestas desde la consola del agente a los cincuenta (50) mensajes obtenidos desde las colas de atención, específicamente aquellos que fueron generados en la prueba anterior.

Figura 9.18: Tiempo de Respuesta al Enviar Mensaje Directo

Se considera que los picos observados en la Figura 9.18 están relacionados directamente con el tráfico en la red existente al momento de ejecutar las mismas, debido a que de por si no existe un arduo procesamiento del servicio web expuesto por el módulo de procesamiento, que pudiera afectar de forma significativa el tiempo de respuesta.

Por otro lado, se buscó extender el alcance inicial de esta prueba, generando el envío iterativo de cuarenta (40) mensajes en un período menor a quince (15) minutos, con la finalidad de verificar el límite teórico que ofrece la documentación de Twitter acerca de su API y las peticiones que este acepta en un intervalo de tiempo definido.

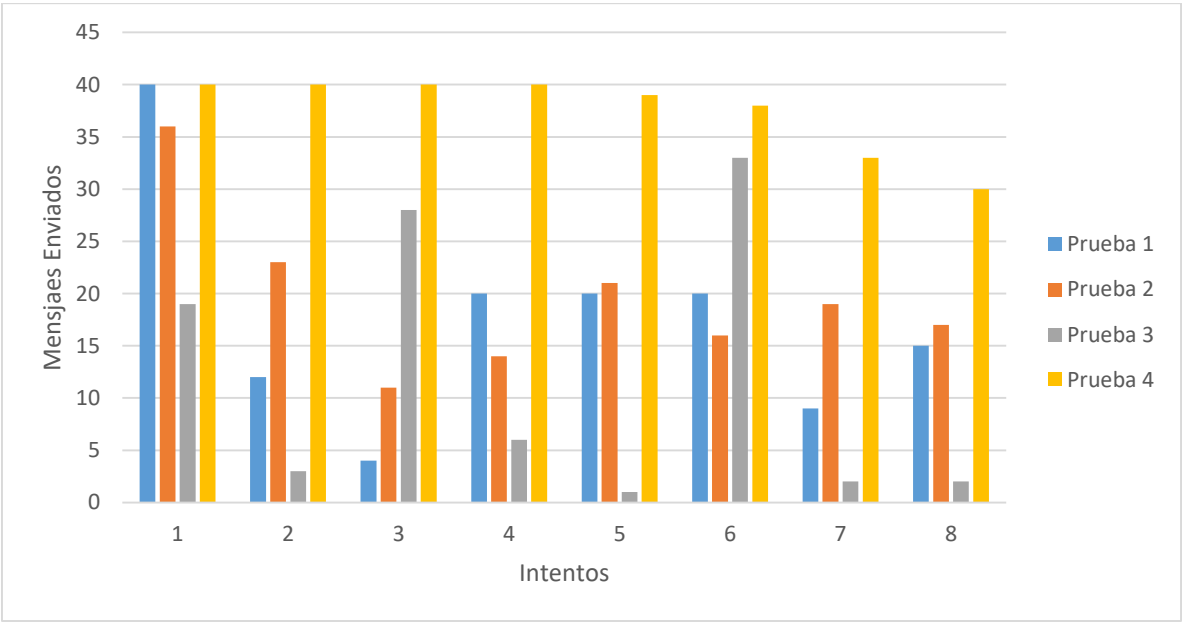

Figura 9.19: Prueba de Estrés Enviando Mensajes Directos Masivos

La regla establecida por Twitter es obtener como máximo quince (15) mensajes en un intervalo de quince (15) minutos, en pocas palabras, una petición por minuto. Por lo que la idea principal fue verificar que pasaba con aquellas peticiones generadas luego de exceder el umbral establecido.

En el escenario planteado, se realizó el envío de mil doscientos ochenta (1280) mensajes en total, dividido en cuatro sets de pruebas variando en la cantidad de mensajes exitosos. En las primeras tres series de pruebas, el contenido de los mensajes fueron variando muy poco y el sistema de Twitter en algunos casos para proteger los usuarios de spam no completa la acción, es por ello que se ve altibajos en la Figura 9.19.

En la última serie, los mensajes enviados no tuvieron una estructura definida, variando mucho más que las pruebas anteriores y esto se vio reflejado en el resultado ya que se obtuvo una gran cantidad de mensajes enviados. Tras las pruebas se pudo sobrepasar el límite teórico de Twitter y se observa que en ocasiones los mensajes pueden ser bloqueados aunque tengan la estructura correcta.

9.5.4 Obtener Mensajes Directos

El objetivo de esta prueba fue determinar los distintos tiempos de respuesta que pueden generarse en el Módulo de Procesamiento al momento de enviar una petición hacia la red social Twitter, específicamente para obtener todos los mensajes directos asociados a un usuario específico.

- Resultado Esperado: Por el tipo de petición realizada y el desconocimiento de la infraestructura de red existente hacia los servidores de Twitter, la cual es variante debido al servicio de Internet, se estipuló un tiempo máximo de mil milisegundos (1000 ms) para el procesamiento efectivo del mensaje.
- Datos Utilizados: Se generaron cincuenta (50) peticiones desde el Módulo de Procesamiento hacia la red social Twitter para obtener los mensajes directos asociados a la cuenta @SocialMedia_App.

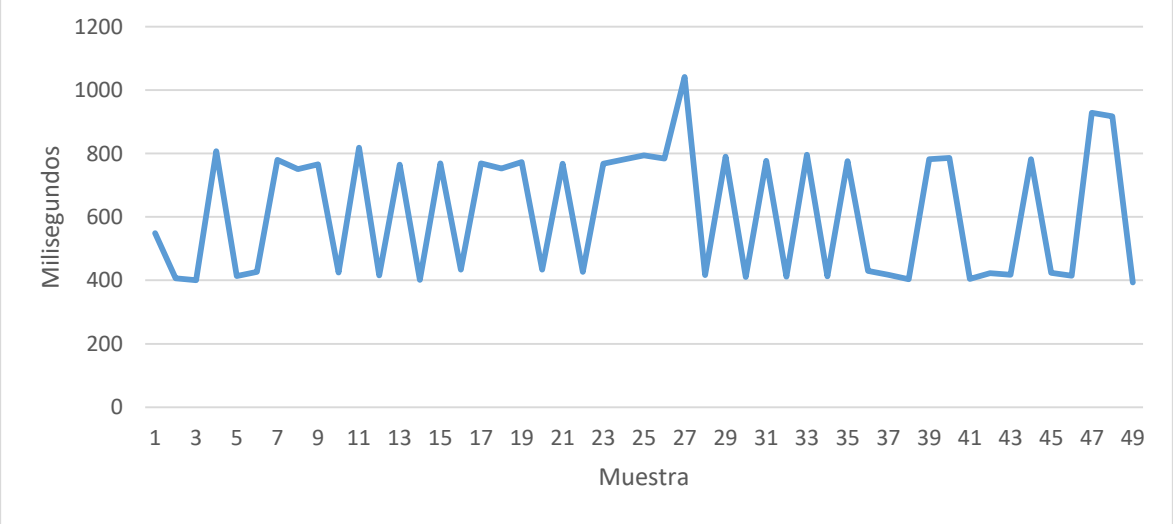

Figura 9.20: Tiempo en la Petición Hacia Twitter

Como se observa en la Figura 9.20, el valor promedio obtenidopara la obtención de los mensajes fue ochocientos milisegundos (800 ms), la variación observada en la gráfica puede ser referente a lo mencionado anteriormente, la ruta que siguen los paquetes para llegar al destino. Influye la velocidad del enlace y la localización de los servidores de aplicación con respecto a los servidores de Twitter.

Al igual que la prueba anterior, se extiende el objetivo de esta prueba buscando sobrepasar el límite definido por la API de Twitter en cuanto a peticiones de búsquedas definidas en la misma.

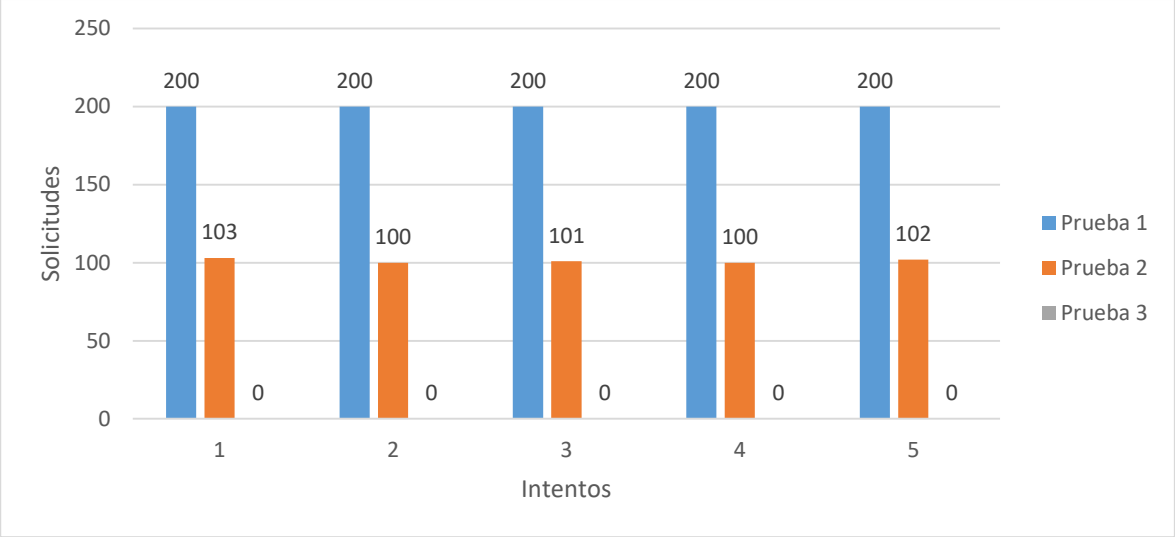

Figura 9.21: Prueba de Estrés Obteniendo Mensajes Directos

Según la documentación, el API de Twitter permite un máximo de ciento ochenta (180) peticiones asociadas a búsqueda de contenido en un intervalo de quince (15) minutos, valor que se asemeja al de creación de contenido utilizado en la prueba anterior.

Se realizan doscientas (200) peticiones de mensajes directos asociados a la cuenta @SocialMedia_App, y como se esperaba, dados los resultados del caso anterior, la búsqueda de mensajes no fue congruente, permitiendo la obtención de doscientos (200) mensajes en un primer intento, los cuales fueron variando en las peticiones subsecuentes, como se observa en la Figura 9.21.

Cada intento que se realizó fue en un intervalo de quince (15) minutos y según los resultados, se puede decir que se puede realizar aproximadamente trecientas (300) solicitudes de obtención de mensaje directo a Twitter antes de recibir el error “To Many Request”. Se sobrepasa una vez más el límite teórico de Twitter, aunque en comparación con otros casos de pruebas, se reciben resultados similares en cada intento.

10. Conclusiones

La evolución en el uso de la tecnología ha aumentado exponencialmente con el auge del servicio de Internet, esto ha venido dejando atrás algunos mecanismos que facilitaban los procesos de comunicación entre diferentes personas. Por lo antes descrito, en el trabajo de investigación se evaluó la factibilidad de integrar el sistema de comunicaciones unificadas basado en Asterisk con la red social Twitter, con el fin de aprovechar aquellos elementos de mejora que puede otorgar la utilización de redes sociales en los procesos internos de gestión al cliente en una organización.

Se desarrolló un módulo con el framework Ruby on Rails que permite la captura y el procesamiento de los mensajes de cada uno de los usuarios conectados en la red social Twitter en un formato definido y hacia una cuenta en específico. De igual forma se habilitaron servicios web para que funcione como puente para enviar mensajes hacia la red social, es decir, permite la comunicación bidireccional entre la red social Twitter y el servidor de comunicaciones unificadas.

También se estudiaron las diversas soluciones de software libres existentes en el mercado para la gestión de mensajes entre distintos sistemas, llegando así a lo que es el broker RabbitMQ, con el cual se diseñó un apartado de colas de atención, basado en los diferentes tipos de mensajes que es capaz de interpretar el módulo de procesamiento, para que funcionará como generador de data para el servidor de comunicaciones unificadas, específicamente del módulo de Call Center.

En el servidor de comunicaciones unificadas, se optó por extender las funcionalidades del módulo de Call Center, específicamente la consola del agente, mediante el desarrollo de un panel que permite la gestión de los mensajes obtenidos desde la cola de atención por los diferentes agentes humanos presentes en la organización con la tarea de resolver problemas, todo esto con la finalidad de ampliar los diferentes canales existentes para la atención del cliente y evolucionar de un Call Center a un Contact Center.

Por último se diseñaron diferentes set de pruebas a las cuales fue sometido el sistema para determinar su rendimiento y el comportamiento que adquiere por cada uno de sus flujos, específicamente desde que un mensaje es obtenido por la red social, hasta que es entregado a un agente humano en la instancia de consola de agente a la cual está asignado.

En base a lo dicho anteriormente se puede concluir que los objetivos específicos propuestos al comienzo de la investigación y el objetivo general del presente trabajo fueron cumplidos, obteniendo así la integración de un sistema de comunicaciones unificadas con la red social Twitter.

10.1 Limitaciones

Las limitaciones del trabajo son:

- La principal limitación de este trabajo se derivó en la cantidad de solicitudes definidas por el API de la red social Twitter.
- Velocidad de Internet y tráfico en la red interna en el ambiente del servidor.

10.2 Trabajos Futuros

Se proponen los siguientes trabajos a futuro:

- Transformar en un backoffice el módulo de procesamiento para que sea una interfaz sencilla y configurable para cualquier organización.
- Realizar un módulo de redes sociales en donde se puedan realizar configuraciones dentro del servidor de comunicaciones unificadas permitiendo gestionar campañas salientes mediante mensajes directos o tweets a varios clientes utilizando plantillas configurables.
- Agendar llamadas bajo un horario predilecto y no de forma instantánea.
- Generar plantillas prediseñadas para ser usadas por el agente al momento de enviar un mensaje directo.
- Generar estadísticas en una interfaz sencilla en cuanto a incidencias atendidas y el tiempo de respuesta a la misma.
- Agregar inteligencia artificial al procesamiento de los mensajes, entrenando al sistema a reconocer palabras o frases negativas. Esto serviría tanto al momento de recibir los mensajes como al enviarlos.
- Realizar procesos de minería de datos que permita conocer a los clientes de la organización y personalizar las campañas salientes enviadas a cada uno de los usuarios.

Bibliografía

[1] Telefonía Voz IP. http://www.telefoniavozip.com.

[2] N. Anaya, Fundamentos de Telefonía IP e Introducción a Asterisk/Elastix, Enero 2013.

[3] ElastixTech, VoIP - Telefonía IP. http://elastixtech.com.

[4] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley y E. Schooler. SIP: Session Initiation Protocol. RFC 3261. Junio 2002.

[5] H. Schulzrinne, S. Casner, R. Frederick y V. Jacobson. RTP: A Transport Protocol for Real-Time Applications. RFC 1889. Enero 1996.

[6] H. Schulzrinne, A.Rao y R. Lanphier. Real Time Streaming Protocol (RTSP). RFC 2326. Abril 1998.

[7] M. Handley y V. Jacobson. SDP: Session Description Protocol. RFC 2327. Abril 1998.

[8] Quarea. http://www.quarea.com/es/asterisk_funcionalidades_basicas_avanzadas.

[9] E. Viegas y C. Facundo, Minestron.it. http://www.minestron.it/asterisk-desconsolado/funciones-de-llamada.

[10] J. Bonilla, Escalado y Balanceo de Carga de Sistemas SIP. http://es.slideshare.net/elastixorg/escalado-y-balanceo-de-carga-de-sistemas-sip.

[11] D. Merel, B. Dempster y D. Gomilion, Asterisk 1.6 - Build Feature-rich Telephony Systems with Asterisk, Packt Publishing, 2009.

[12] Free PBX Community. http://community.freepbx.org/t/recommended-maximum-number-of-extensions/4941.

[13] VoIP Wiki. http://www.voip-info.org/wiki.

[14] R. Bryant, L. Madsen y J. V. Meggelen, The Definitive Guide, O'Reilly, Noviembre 2013.

[15] G. Sedano, Historia de las Redes Sociales. http://www.webespacio.com.

[16] Breve Historia de las Redes Sociales. http://www.marketingdirecto.com.

[17] J. Del Moral, Redes Sociales ¿Moda o nuevo Paradigma?, Madrid: Asociación de Usuarios de Internet, 2005.

[18] Twitter, Twitter Developers. https://dev.twitter.com.

[19] Facebook, Facebook Developers. https://developers.facebook.com.

[20] Google, API Google. https://developers.google.com.

[21] S. Cony, Guerra por los Mensajes. https://www.fayerwayer.com/2013/05/la-guerra-por-los-mensajes.

[22] H. Gaibor, Integración de Elastix con Gtalk. http://www.elastix.com/integration-of-elastix-with-gtalk.

[23] S. Smith, Publicación en Twitter Mediante ASR. http://www.10000horas.com/2011/12/08/un-agi-legendario-publicacion-en-twitter-mediante-asr.

[24] G. Sied, 15 Ejemplos de Manipular Archivos de Audio. http://www.sied.com.ar/2014/02/15-ejemplos-impresionantes-de-manipular-archivos-de-audio-utilizando-intercambio-sano-sox.html.

[25] M. Jakl, Representational State Transfer, 2005.

[26] F. Potencier y I. Wiedler, Silex. http://librosweb.es/libro/silex.

[27] RabbitMQ. https://www.rabbitmq.com.

[28] Introduccion a JSON. http://www.json.org/json-es.html.

[29] Ruby on Rails. http://rubyonrails.org.

[30] P. Estrella, J. Bustos y A. Muñoz. Implementando Call Center con Elastix. 2013.

[31] A. Santos. Elastix Manual para Desarrolladores. 2012.