



**Universidad de Mendoza**  
**Facultad de Ingeniería**  
**Maestría en Teleinformática**

**Autorización de Prestaciones Médicas**

**Lic. Alejandro Fernández Herrero**

**Directores de Tesis:**

**Dr. Ing. Cristina Párraga**

**Mg. Ing. Osvaldo Marianetti**

**2009**

## **Agradecimientos**

Quiero agradecerle muy especialmente a Marita mi esposa por su continuo aliento, dedicación, apoyo y amor. A mis tres hijos, por ser mis fuerzas movilizadoras.

A toda mi Familia, en especial mis padres Atilio y Nélida por haberme educado como persona con sólidos valores éticos y morales. A mis hermanos, porque son fuente de apoyo. A mis 12 sobrinos porque viéndolos a ellos tengo confianza en las nuevas generaciones.

A Círculo Médico de Mendoza como Institución que permitió desarrollarme profesionalmente y humanamente.

Al Dr. Ignacio Berríos, Presidente de Círculo Médico de Mendoza por haber apoyado mis proyectos y brindarme siempre su visión complementaria.

Al Dr. Ing. Carlos García Garino, por los momentos compartidos que proyectaron en mi toda su fuerza y empuje.

Al Dr. Richard Branham, Director del Centro de Cómputos de CRICyT, quien fuera mi primer jefe y me enseñó que los problemas se solucionan estudiando los manuales de los sistemas o equipos.

A mi compañero en el CRICyT, Dr. Jaime Prilusky, porque me enseñó que todo se podía resolver, en una edad de mi vida donde siempre llegaba a un callejón sin salida.

Al movimiento de software libre, que me permitió contar con productos y herramientas para mi desarrollo profesional y animarme a hacer lo propio.

A mis dos conductores de tesis Cristina y Osvaldo por su apoyo, presencia y dedicación desde siempre.

## Índice de contenido

1. Introducción General .....	8
2. Marco Conceptual.....	12
2.1. Automatas Telefónicos.....	12
2.2. Fenómenos de convergencia.....	12
2.3. Mensajería Unificada (UM).....	13
2.4. Sistemas Ubicuos.....	13
2.5. Implementación.....	14
3. Caso de Estudio.....	15
3.1. Las Obras Sociales.....	15
3.1.1. Planteo del Problema.....	15
3.1.2. Solución Propuesta.....	16
3.2. Impacto de la solución propuesta.....	16
4. Telefonía IP.....	18
4.1. Definición.....	18
4.2. Funcionamiento.....	18
4.3. Elementos de una red SIP.....	18
4.3.1. Protocolo SIP.....	18
4.3.1.1. Importancia del protocolo SIP.....	20
4.3.2. User Agents (UA).....	21
4.3.3. SIP Proxy Server.....	21
4.3.4. SIP Redirect Server.....	22
4.3.5. Mecánica de una llamada y establecimiento de comunicación.....	22
4.3.6. SIP Registrar Server.....	23
4.3.7. Mecánica de una llamada compleja.....	23
4.3.8. Back to back User Agent - B2BUA.....	25
4.3.9. Media Gateway.....	26
4.4. Flujo Multimedial (Media Streams).....	27
4.4.1. Flujo (Streaming).....	28
4.4.2. Protocolo RTP .....	28
4.4.3. Protocolo RTCP.....	29
4.4.4. Protocolo IAX2.....	29
4.4.5. Capas del Protocolo.....	30
5. Empresas de Telefonía IP.....	32
5.1. Introducción.....	32
5.2. Presencia Local en Oficina Virtual con Ventaja Real.....	32
5.3. Servicios.....	33
5.4. Servicios básicos y gratuitos.....	33
5.5. Servicios avanzados.....	34
5.6. Numeración local DID.....	34
5.7. Aplicaciones.....	35
6. Central Telefónica Asterisk.....	36

6.1. Introducción.....	36
6.2. Plataforma de desarrollo .....	36
6.3. Características de Asterisk .....	37
6.4. Plan de Discado (Dial Plan) .....	40
6.4.1. Contextos .....	40
6.4.2. Extensiones .....	40
6.4.3. Prioridades .....	41
6.4.4. Aplicaciones .....	41
6.5. Respuesta Interactiva de Voz (IVR) .....	42
6.5.1. Servicios.....	42
6.6. Asterisk Gateway Interface – Interfaz de conexión con Asterisk.....	43
6.6.1. Fundamentos de una comunicación con AGI.....	43
6.6.2. API's para programar con AGI.....	44
6.7. Asterisk Manager Interface (AMI) .....	44
7. Autómatas Telefónicos.....	45
7.1. ¿Qué es un Autómata Telefónico?.....	45
7.2. Forma de interacción.....	45
7.3. Usos frecuentes.....	46
7.4. Tecnologías.....	47
7.4.1. Voz pre-grabada.....	47
7.4.2. Voz Sintetizada.....	47
7.4.3. Otros considerandos tecnológicos.....	48
7.5. Críticas.....	49
8. Diseño de un AT.....	50
8.1. Introducción.....	50
8.2. Pautas de Diseño.....	50
8.2.1. Sea sucinto pero no sacrifique significado.....	50
8.2.2. Priorice las tareas importantes.....	50
8.2.3. Enfoque su aplicación.....	50
8.2.4. El sonido tiene que ser lo más natural.....	51
8.2.5. Evite largos silencios.....	51
8.2.6. Construya una maqueta.....	51
8.2.7. Utilice normas.....	51
8.2.8. Utilice una voz profesional.....	51
8.2.9. Evite usar publicidad.....	52
8.2.10. AT adaptativos e inteligentes naturales.....	52
8.2.11. Nunca se confíe en el menú.....	52
8.3. Normas de Diseño.....	52
9. Lenguajes de Programación.....	53
9.1. Introducción.....	53
9.2. Lenguajes más utilizados.....	53
9.3. Generadores de AT.....	54
10. Desarrollo de un AT.....	55

10.1. Introducción.....	55
10.2. Ruby.....	55
10.2.1. Características sobresalientes.....	55
10.3. NetBeans.....	57
10.3.1. Características destacables.....	57
10.3.2. Control de versiones.....	57
10.4. MySQL.....	59
10.4.1. Principales usos.....	59
10.4.2. Redundancia.....	60
10.4.2.1. Redundancia Master/Slave.....	60
10.4.2.2. Redundancia Multi-Master.....	60
10.4.2.3. Redundancia utilizando Clusters.....	61
11. Autorización de Prestaciones Médicas.....	63
11.1. Introducción.....	63
11.2. Autómata finito.....	63
11.2.1. Autómata finito determinístico.....	63
11.2.2. Autómata finito no determinístico.....	64
11.2.3. Fundamento Matemático.....	64
11.3. Autómata telefónico como una máquina de estados finito.....	65
11.4. Lógica de Negocio.....	66
11.4.1. Inicio del autómata.....	67
11.4.2. Proceso.....	71
11.4.3. Finalización.....	75
12. Conclusiones.....	77
12.1. Investigaciones y desarrollos futuros.....	77
13. Bibliografía.....	79

## Índice de Figuras

Inicio de sesión y flujo multimedial.....	23
Conexiones entre dos agentes de usuarios.....	25
Operador de telefonía IP.....	27
Distribución de los protocolos en capas.....	31
Diferencia entre versiones.....	58
Eliminar y adicionar código.....	58
Distribución de almacenamiento y clientes bajo clusters.....	62
Teléfono softphone.....	70

## **Preámbulo**

(Abstract)

Mediante un exhaustivo estudio técnico, el presente trabajo analiza, a partir del caso del de las Obras Sociales que brindan el servicio de autorizar prestaciones médicas, las tecnologías necesarias para poder diseñar, desarrollar y finalmente implementar un Automata Telefónico. Este es definido como un dispositivo que contiene la electrónica y el software necesario para contestar una llamada y poder brindar información a un potencial usuario o ingresar información provista por el usuario, siendo el usuario la persona que llama a un determinado número telefónico.

Además, a lo largo de la tesis, se realiza un análisis de las características y la importancia de la telefonía IP, se describe el Media Gateway y la Central telefónica Asterisk. Posteriormente se hace una caracterización del Automata telefónico, se especifican sus componentes, se describen las tecnologías más importantes, se los diferencia de los Sistemas de Respuesta Interactiva de Voz y se establecen las bases para el diseño de un Automata telefónico exitoso.

La nueva tecnología, utilizada por los Automatas Telefónicos, se enmascara detrás del teléfono. Nuevos dispositivos entran a formar parte de un escenario donde le quedan totalmente ocultos al usuario. Dentro de estos dispositivos tenemos a los que permiten gestionar las llamadas a diversos destinos, es decir, Centrales Telefónicas IP, dispositivos que permitan unir diversas tecnologías, por ejemplo la red de telefonía básica (PSTN) con la red digital de datos (Internet), es decir, Gateways Telefónicos, dispositivos que permitan transformar un tradicional teléfono analógico en un teléfono apto para poder hablar en forma directa por medio de la red digital de datos, es decir un Adaptador Telefónico Analógico (ATA), un dispositivo que permita almacenar grandes volúmenes de datos o información para luego poder ser recuperados para su procesamiento, es decir, bases de datos (BD), aplicaciones que le dan la "inteligencia" al sistema implementando diversas reglas de negocio, es decir autómatas telefónicos. Aún existen otros dispositivos menores que se unen a todos estos como repetidores, teléfonos IP, etc.

El objetivo es dar nuevos servicios, utilizando tecnología de punta, siendo provista por el tradicional teléfono, sirviendo éste de vector para recuperar o introducir información.

## 1. Introducción General

Partiendo del estudio de un caso en particular, el de las Obras Sociales que brindan el servicio de autorizar prestaciones médicas, la presente tesis tiene por objetivo analizar las tecnologías necesarias para poder diseñar, desarrollar y finalmente implementar un Automata Telefónico.

A partir de este objetivo general, iremos cumpliendo con los siguientes objetivos específicos:

- Diferenciar entre los Sistemas de Respuesta Interactiva de Voz (IVR) y Automatas Telefónicos.
- Analizar el impacto en mercados específicos de la utilización de los Automatas Telefónicos
- Analizar los componentes necesarios para la implementación de Automatas Telefónicos.
- Comparar diversas tecnologías para el desarrollo de Automatas Telefónicos.
- Especificar los componentes de un Automata Telefónico.
- Describir las tecnologías más importantes.
- Establecer la importancia de las empresas proveedoras de telefonía IP.
- Establecer bases para diseños exitosos de Automatas Telefónicos.
- Plantear la estrategia de vinculación entre las partes involucradas.

Los gerentes de las empresas siempre tratan de agregar valor a sus procesos, en especial si estos procesos están orientados a servicios. La tecnología es el hilo conductor, pero son muy cautelosos a la hora de implementar una nueva tecnología que puede costar mucho dinero y en especial pérdida de tiempo o fundamentalmente de imagen. Lo que analizan es:

- El grado de aceptación de la tecnología. Es decir, que porcentaje de su cartera se unirá a esta tecnología. Generalmente esto es un porcentaje del total de clientes.
- El tiempo de aceptación. Es decir, la cantidad de tiempo que sus clientes van a adoptar la nueva tecnología. Esto puede estar influenciado por la obligatoriedad, rifa de regalos y cualquier otra estrategia de adhesión.

- La práctica que le puede resultar a los clientes la utilización de la tecnología.

Todos quieren poder brindar mayor información a sus clientes, y en tal sentido, los autómatas telefónicos son un excelente medio para poder relacionarse con los clientes. Es la posibilidad de poder interactuar, no solamente poder brindar información sino poder captar datos del cliente. La implementación temprana de una tecnología puede ser un elemento diferenciador incalculable, pero la aceptación de las tecnologías llamadas disruptivas es en factor clave a la hora de iniciar un proyecto.

Ya no quedan dudas de que estamos inmersos en la *Era de la Información*. Existen diversos medios (vectores) para obtener información, quizás los navegadores (web browsers) son la vedette de todos ellos. Por medio de ellos podemos incluso correr aplicativos que nos permiten introducir u obtener información de una empresa. No obstante para ciertas actividades puntuales, el tradicional teléfono de tonos pareciera ser muchísimo más adecuado que tecnologías más sofisticadas. Su ubicuidad y costo reducido lo hacen un excelente medio para obtener y guardar información.

Los navegadores (Web Browsers) parecieran ser la herramienta por excelencia para poder realizar transacciones utilizando un proceso que tiene una determinada lógica de negocios y que almacena los datos en una BD. No obstante veremos como los autómatas telefónicos han podido desplazar a otros tipos de tecnologías, debido a su facilidad de uso, bajo costo y ubicuidad.

Podemos decir que se ha creado un nuevo nicho dentro del mercado para los Autómatas Telefónicos, cada vez es mayor la cantidad de empresas que los usan. Algunas aplicaciones donde se están utilizando:

- Sistema de proveedores
- Sistema de turnos para cualquier servicio

Es común que cuando llamamos a una empresa nos atienda un dispositivo llamado comúnmente "Pre-Atendedor Telefónico". Éste nos da la bienvenida y nos presenta un conjunto de opciones que por medio de la interacción de las teclas de nuestro teléfono nos va guiando a otro sub-menú, hasta que finalmente nos comunica con una persona o sector de la empresa determinada. Los Atendedores Telefónicos es una expresión que proviene del mundo telefónico. Una evolución de estos Atendedores Telefónicos lo constituyen los Sistemas de Respuesta Interactiva de Voz (IVR) por sus siglas inglesas. Estos sistemas generalmente corren sobre una Central Telefónica Analógica (CT) y tienen la posibilidad de poder

dejar un mensaje de voz a una persona que no responde. Tienen características similares a los Atendedores Telefónicos pero con mayor grado de prestaciones. Los IVR también corren sobre Centrales Telefónicas IP, y como éstas corren generalmente dentro de una computadora tienen aún mayor grado de plasticidad en sus servicios.

Sin embargo, cualquiera de los dos modelos presentados tiene un alcance limitado y restringido a la institución que los alberga. Las Centrales Telefónicas basadas en protocolo IP, corren sobre computadoras y ofrecen al programador un conjunto de interfaces de aplicación (API – Application Program Interface) que permiten desarrollar aplicaciones con todo su potencial. A estas aplicaciones se las denomina Automatas Telefónicos.

La finalidad de los autómatas telefónicos es la de poder interactuar con una persona para proveerle información o ingresar datos. En ambos casos la ubicuidad también es una finalidad de esta tecnología, no importa el tiempo o lugar, tiene que tener acceso a ella. El acceso al teléfono, sea éste fijo o móvil, es un condicionante, que cada día se hace más débil. Analizando la fortaleza, es un medio que tiende cada vez a masificarse y a bajar sus costos.

El ingreso de datos es una fortaleza que distribuye la fuerza de carga de datos en infinidad de bocas, en su gran mayoría, datos validados, listos para ser procesados, pero ingresados por un tercero ajeno a la Empresa.

Todo esto da lugar a transacciones económicas, o sea transacciones que tienen valor tanto para la Empresa como para el Cliente. Ambos están jugando en la economía global. Negocios que se potencian por usar estas herramientas. Se puede ser más o menos reacio a la incorporación de éstas herramientas, pero no se puede ser necio. Ignorar esto hoy es no tener una visión de futuro.

Para verdaderamente comprender el valor de una tecnología, debemos considerar la capacidad u oportunidad de poder realizar un negocio. Todos tenemos casos de tecnologías de menor calidad que se imponen a otras de mayor calidad. Es todo una cuestión de mercado. Por esto es fundamental interrogar al mercado antes de desarrollar e implementar costosos sistemas.

Creemos que es posible desarrollar autómatas telefónicos que brinden una nueva gama de servicios a costos extremadamente competitivos. El tradicional teléfono es potenciado por nuevas tecnologías de comunicación que se enmascaran detrás de él. Así, queda oculta toda la nueva tecnología, el usuario o cliente, sigue viendo su tradicional teléfono, el que le da “seguridad”, lo que conoce, no puede

rechazar la nueva tecnología porque para él es conocida.

## 2. Marco Conceptual

### 2.1. Automatas Telefónicos

La palabra autómata viene del griego *automatos* y significa espontáneo o con movimiento propio. Una de las acepciones de éste vocablo nos dice: “*Máquina que imita la figura y los movimientos de un ser animado*”, ésta es precisamente la definición que se aplicará en el presente trabajo. Obviamente el “ser animado” al que esta haciendo referencia ésta definición es una persona, es decir, si no tuviera al autómata telefónico tendría que ser reemplazado por una persona que conteste la llamada y que a su vez provea información o ingrese datos que le están proveyendo desde el otro extremo de la comunicación.

De manera entonces, que Autómata Telefónico se define como un dispositivo electrónico y software capaz de contestar una llamada y poder brindar información a un potencial usuario o ingresar datos provistos por el usuario, siendo el usuario la persona que llama a un determinado número telefónico. Obviamente que tanto la información entregada o los datos registrados son obtenidos desde una Base de Datos (BD).

### 2.2. Fenómenos de convergencia

Desde hace tiempo se viene hablando de los fenómenos de convergencia aplicados a datos, audio, voz, telefonía y TV. Mucho se ha hecho, pero aún resta un camino muy largo y con grandes inversiones. La idea es sencilla, utilizar una única red de transmisión donde toda manifestación física pueda ser transformada a bits y de esta forma poder ser transportada por una red digital que hasta ahora se piensa solamente en Internet, donde IP juega el rol más importante, es por esto que también se conoce el tema como *Convergencia IP*. Pero aún resta la posibilidad de transmitir grandes caudales de datos, la posibilidad de realizar unicast o broadcast y fundamentalmente poder conmutarse de dispositivo. Por ejemplo estar viendo una película en un televisor de plasma de 38 pulgadas recibiendo la señal por medio de IPTV y que pueda terminar de verla desde el teléfono móvil mientras viaja en un taxi a su trabajo. O que un sólo número sirva para contactarse con una persona, independientemente del lugar y del dispositivo que esté usando. Pero todo esto es impensable sin la implementación de redes nuevas y esto, como ya se dijo, implica inversiones millonarias.

Sin embargo, yendo al tema de los AT, podemos utilizar este fenómeno de convergencia, de esta forma podemos servirnos tanto de

redes de telefonía básica (PSTN) como redes de datos (Internet). Utilizando tramos de una y otra red. A esto podríamos decir que es una convergencia por resultados o híbrida ya que no todo se cursa por una única red como se propone en los fenómenos de convergencia puros.

### **2.3. Mensajería Unificada (UM)**

Como resultado del fenómeno de convergencia tenemos a la Mensajería Unificada (UM). Es la integración de diferentes flujos de comunicación, tales como, e-mail, SMS, Fax, voz, video, etc., en un único (o unificado) almacén de mensajes. Cada uno de estos flujos proviene de distintos dispositivos y es recuperado por aplicativos, que en la mayoría de los casos abarcan el tratamiento de diversos medios.

El objetivo de todo esto es la posibilidad de mejorar cómo los individuos, grupos y empresas interactúan y llevan a cabo sus tareas. Las empresas deberían revisar sus actuales sistemas y procesos de comunicación, para luego implementar pruebas pilotos mejorando así, los procesos de comunicación. Una cosa importante a destacar es que las comunicaciones unificadas son un proceso, por lo tanto esto no se obtiene de una caja que la pueda proveer un distribuidor. En este sentido es la empresa quién toma conciencia de la mejora que puede tener al implementar soluciones de comunicación unificadas. Trabajar con el personal, implementar de a poco las diferentes tecnologías conducentes. En el Open Source existen todos los elementos para poder implementar mensajería unificada, por ejemplo comenzar con el mail, telefonía y fax.

Aún queda mucho por trabajar en este sentido, existen problemas a la hora de implementar soluciones, pero lo que no cabe la menor duda es que este proceso lo van a transitar todas las empresas en los próximos años.

### **2.4. Sistemas Ubicuos**

El vocablo ubicuo significa “*que está en todas partes*” o “*que se encuentra en cualquier parte del espacio*”. La ubicuidad de un autómata telefónico esta dado por la capacidad, a nivel de diseño, de poder ser accedido desde cualquier parte, sin importar lo geográficamente distante que nos encontremos de él. Pero a su vez, y ésta es una de las premisas del diseño, que el costo del acceso sea el menor posible.

De esta forma yo puedo acceder a un autómata telefónico desde Madrid con el costo de una llamada local o puedo ingresar al mismo desde Buenos Aires también con el costo de una llamada local. Pueden darse casos por restricciones de disponibilidad técnica, que el costo sea más elevado que el de una llamada local, pero siempre será el menor

costo posible, como sería el caso de una llamada interurbana o de media distancia para acceder a la red.

Esto nos permite entonces que los Automatas Telefónicos puedan dar cobertura a grandes extensiones geográficas como una provincia, una región un país o el mundo. La restricción geográfica estará dado mas por razones de negocios que por razones técnicas.

## **2.5. Implementación**

Para poder cumplir con los fenómenos explicitados anteriormente los Automatas Telefónicos hacen uso de tecnologías que le sirven de base para poder desarrollarse. De esta manera queda claramente identificado el front-office que es lo que el usuario percibe (teléfono como tal), escucha (los mensajes provenientes del AT) y las teclas (medio para interactuar) y por otro lado el back-office que es todo lo que está en la trastienda, es decir, todo lo que el usuario no percibe pero que tiene que estar debidamente implementado y configurado para que una transacción pueda llevarse a cabo. Para esto, más adelante veremos, telefonía IP, las centrales telefónicas IP, los Automatas Telefónicos y la programación de los AT.

## **3. Caso de Estudio**

### **3.1. Las Obras Sociales**

Analizaremos un caso real donde demostraremos que la utilización de un AT es mucho más adecuada que la implementación de cualquier otra tecnología, este caso es el ámbito de las Obras Sociales.

Las Obras Sociales (OS) son instituciones dedicadas a brindar cobertura médica a sus afiliados. Los afiliados son el medio financiador del sistema, ya que ellos, generalmente por medio de descuentos en sus nóminas de sueldos, realizan el sostén económico y financiero del sistema.

#### **3.1.1. Planteo del Problema**

En las Obras Sociales, los afiliados consumen prestaciones de salud, tales como una consulta médica, una radiografía o un análisis bioquímico. A los encargados de proveer este servicio se los denomina Prestadores de Servicios de Salud. Están formados por médicos, bioquímicos, fonoaudiólogos, etc. Por no ser incumbente para este estudio no veremos las Instituciones donde se realizan internaciones de afiliados. Es decir, este sistema dará servicio, por una cuestión de diseño y simplificación, a los afiliados que se mueven en ambientes ambulatorios o consultorios externos.

Para que el afiliado pueda consumir una prestación, concurre a su OS y solicita una consulta médica o viene con un pedido de estudios recetados por un profesional. La OS le emite un bono y posiblemente le cobre un importe que se denomina coseguro, copago o ambos. Con el pedido del profesional más el bono emitido por la OS, el afiliado se dirige al Prestador. Le entrega ambos papeles y consume su prestación. Luego con el bono recibido por el afiliado el Prestador gestiona su cobro ante la OS. Generalmente éste último trámite lo realiza por medio de una entidad intermedia que le recibe los bonos de todas las OS y le gestiona el cobro de todas las prestaciones, estas empresa gestoras, actúan “por cuenta y orden” del tercero al cual le realizan los trámites de cobro. Círculo Médico de Mendoza, es una entidad que brinda este tipo de servicios.

Nos enfocaremos en el hecho de que el afiliado tuvo que concurrir a la OS para que le autorizaran y emitieran el bono de su prestación. Por otro lado el Prestador tuvo que concurrir a una entidad intermedia para que le gestionaran su cobro.

### **3.1.2. Solución Propuesta**

La solución propuesta es eliminar estos trámites tan problemáticos para el afiliado y prestador ya que tiene que trasladarse a la sede de la OS y esperar que le emitan el bono, para luego concurrir a su prestador. Por otro lado tiene que concurrir a la empresa y cambiar su bono por dinero o promesa de pago.

Con el enfoque que se propone, el afiliado concurre directamente al prestador para realizar la consulta médica o realizarse un estudio, el prestador tiene que llamar a un número telefónico (puede ser de cobro revertido) y transaccionar con el Automata Telefónico, donde le solicitará: tipo de profesión, número de matrícula, PIN, prestación a autorizar y número de afiliado. El AT tiene la lógica de negocio para llevar adelante controles tales como: pertinencia entre especialidad del profesional y práctica, consumo del afiliado, calculo del coseguro, etc. y luego el AT le informa al prestador la decisión tomada, puede ser otorgarle el código de autorización de la prestación o negarle la prestación por alguna situación especial (afiliado dado de baja, no pertinencia de la prestación, consumo excesivo, etc.).

En el caso que el prestador haya obtenido el código de autorización, esta práctica esta autorizada y podrá ser cobrada por dicho profesional. Se puede implementar que el pago de dicha prestación se realice en forma automática a las 24hs de realizada la prestación, evitando de esta forma, la empresa intermedia de gestión de cobranza.

Ahora veamos un escenario mucho más complejo. Supongamos que la sede de la OS se encuentra en Reconquista – Santa Fe (tal un caso que me tocó analizar), ésta ciudad da cobertura a pueblos y ciudades que se encuentran geográficamente a 200km a la redonda. Para que el afiliado pueda concurrir a un prestador del pueblo de, por ejemplo, Villa Ocampo tiene que trasladarse a Reconquista para obtener una orden de atención para luego ir al prestador que se encuentra en su pueblo. Para evitar esto la OS tendría que poner una oficina en Villa Ocampo, pero el costo de esta oficina (alquiler, personal, seguros, costos fijos, etc.) le es prohibitivo. A su vez en Villa Ocampo no existen muchos prestadores que tengan acceso a Internet para poder correr una aplicación Web. De manera que la única solución viable es la instalación de un AT que permita la autorización de la prestación por vía telefónica.

### **3.2. Impacto de la solución propuesta**

La solución a la problemática de autorización de prestaciones

médicas por medio de un AT, resulta ser la más adecuada. La reducción de costos y el incremento de servicios es tangible. Por lo tanto cada una de estas ciudades o pueblos tiene que realizar una llamada interurbana para poder acceder al AT que se encuentra en la ciudad cabecera donde se encuentra el Centro de Llamadas.

Si fuese una Obra Social con cobertura nacional, ¿cómo se puede hacer para evitar tener Centros de Llamada en cada zona?. Una forma sería contratar un servicio telefónico de cobro revertido (0-800) nacional y que todas las llamadas llegasen a un único centro autorizador. Esta solución tiene el inconveniente del elevado costo telefónico, porque se pasa de estar involucrados con costos locales o interurbanos para pasar a esquemas de costo de llamadas de larga distancia.

Para evitar esto, en los centros de cabecera se instalan centrales telefónicas IP. De esta manera arriba la llamada interurbana por medio analógico y se encamina dicha llamada al centro nacional de autorización por Internet. En el siguiente capítulo veremos la telefonía IP, que nos permitirá implementar este servicio.

## **4. Telefonía IP**

### **4.1. Definición**

La telefonía IP es una tecnología que ya lleva varios años con nosotros. Se aplica a la transmisión de la voz en paquetes de datos IP que fluyen a través de redes de datos en lugar de las redes telefónicas convencionales. A esto también se lo conoce como VoIP (Voz sobre IP).

### **4.2. Funcionamiento**

Una llamada se puede originar desde un dispositivo de hardware que puede ser una PC (computadora personal) por medio de un teléfono por software (soft phone), un teléfono analógico unido a un ATA (adaptador telefónico analógico), o un teléfono IP. Esta llamada se envía hacia un servidor que señala y encamina la misma hacia gateways que se encuentran en los distintos destinos.

Podemos tener llamadas puramente IP, en este caso la llamada se origina y termina en un dispositivo IP o podemos tener llamadas híbridas, en este caso utilizamos la red pública de telefonía conmutada (PSTN) para poder ingresar al sistema telefónico IP, poder salir desde el sistema telefónico IP hacia la PSTN o poder realizar ambas cosas a la vez. En este último caso se utiliza a la telefonía IP como un troncal, es decir, la llamada comienza en la PSTN y termina en la PSTN.

La pregunta que surge es ¿porqué no usar todo el tramo por medio de la PSTN?, es sencillo el inicio pueda estar en un lugar y el destino puede estar geográficamente a miles de kilómetros y en ese caso se abonan dos llamadas locales, la del lugar de inicio y la del lugar de destino.

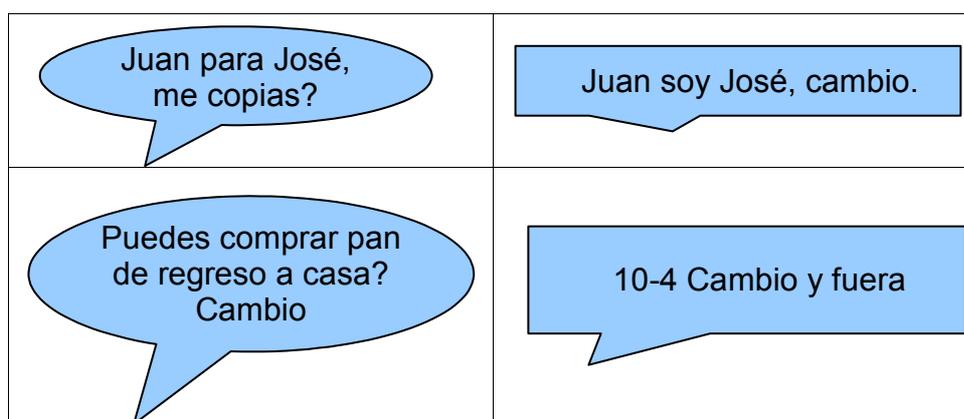
### **4.3. Elementos de una red SIP**

Ya tenemos una idea de cual es el funcionamiento, ahora nos dedicaremos a ver los distintos elementos que hacen posible que la telefonía IP se pueda llevar a cabo.

#### **4.3.1. Protocolo SIP**

Este protocolo ha pasado a ser el protocolo por defecto en

aplicaciones de telefonía IP, desplazando al histórico H.323. SIP, significa Protocolo de Inicio de Sesión, es un protocolo liviano, extensible, solicitud/respuesta para iniciar sesiones de comunicación entre extremos. SIP esta inspirado conceptualmente en los protocolos HTTP y SMTP, si bien su aplicación es diferente. Podríamos comparar al mensaje SIP, a una conversación mantenida bajo Banda Cuidadana (CB) y sus códigos 10-xx.



En este caso el mensaje real, esta circundado con mensajes de negociación.

El SDP (Session Description Protocol) se ubica dentro de la carga útil del protocolo SIP, se emplea para establecer las características de la sesión, entre otras cosas para comunicar los parámetros de sesión como la codificación empleada, tipo de flujo a intercambiar, etc..

¿Cuáles son los beneficios de SIP? Se usa generalmente para negociar una llamada entre dos puntos. Se negocia el medio (texto, voz, vídeo, otros), el transporte (posiblemente RTP) y la codificación a emplear en la comunicación (codec). Una vez que la negociación es exitosa los dos extremos usan el método seleccionado para hablar unos con otros, independientemente de SIP. Una vez que se termina la "llamada" , SIP toma el control para finalizar la llamada. Por lo tanto, SIP se utiliza para mecanismo de señalización. Un extremo, en la jerga SIP, se denomina agente de usuario (user agent), este podría ser un soft phone, cliente de mensajería instantánea, teléfono IP o aún un teléfono celular.

A simple vista, el SIP parece muy sencillo, y realmente lo es. Pero detrás de su sencillez se oculta su gran fortaleza, que es la de ser extensible, esto le ha permitido ser utilizado en muchas aplicaciones por medio de extensiones que el mismo protocolo permite. Pensemos en el protocolo HTTP, la definición en si misma es muy pequeña, pero la forma

de utilización es ilimitada.

#### 4.3.1.1. Importancia del protocolo SIP

Hoy en día se está empezando a afirmar que lo que fue HTTP para la Web, lo hará SIP para las telecomunicaciones.

SIP tiene grandes repercusiones en la industria de las telecomunicaciones. Las empresas de celulares han decidido normalizar sobre SIP todas las aplicaciones futuras. Los distribuidores de VoIP, telefonía IP, mensajería instantánea (como el Microsoft MSN Messenger), están todos normalizados sobre SIP.

El direccionamiento empleado en SIP coincide con una URL y por lo tanto es muy familiar para el usuario acostumbrado a usar Internet. Direcciones típicas podrían ser:

- sip: manuel.otero@empresa.com.ar
- sip: 542614281868@qubitsa.com.ar:user=alejandro
- sip: operadora@qubitsa.com.ar

Si ponemos en la barra de dirección de un navegador sip:manuel.otero@empresa.com.ar. El navegador nos abrirá el aplicativo (UA) que tengamos definido en nuestro sistema para administrar el protocolo SIP, por ejemplo el softphone X-lite.

Sin embargo ya existen protocolos de señalización como así también tecnologías de comunicación peer-to-peer. ¿Qué ventajas ofrece SIP sobre ellos?:

- **Estabilidad:** El protocolo está en uso desde hace años
- **Velocidad:** Este pequeño protocolo basado en UDP es extremadamente eficiente
- **Flexibilidad:** Este protocolo basado en texto es fácilmente extensible.
- **Seguridad:** Están disponibles distintos métodos de cifrado (SSL, S/MIME).
- **Normalización:** Con todas las empresas de telecomunicaciones moviéndose hacia él, ha comenzado a ser un estándar. Está en proceso de normalización por el

IETF (<http://www.ietf.org/rfc/rfc3261.txt>).

### 4.3.2. User Agents (UA)

Son los extremos SIP que inician y responden a solicitudes al comienzo de una comunicación. Siempre los dos agentes en los extremos son pares (peers), por lo tanto se establece una comunicación par-a-par, sin embargo por una cuestión de claridad se los diferencia como UAC para el agente cliente (o que inicia la llamada) y UAS para el agente servidor (o que responde a la llamada). Los UA pueden comunicarse unos con otros en forma directa, sin embargo en escenarios reales aparecen dispositivos intermedios como servidores proxys, servidores de redirección, servidores de registros y adaptadores de medios.

### 4.3.3. SIP Proxy Server

Es una entidad intermediaria que actúa como servidor y cliente con el fin de hacer peticiones a nombre de otros clientes. Un proxy server desempeña, sobre todo, el papel del encaminamiento, que significa que su trabajo es asegurarse de que la petición sea enviada a otra entidad "más cercana" al usuario apuntado. Los proxys son también útiles para hacer cumplir las políticas (por ejemplo, cerciorarse que un usuario pueda hacer una llamada). Un proxy interpreta, y, en caso de necesidad, reescribe partes específicas de un mensaje antes de enviarlo.

Los proxys SIP son los elementos que encaminan peticiones SIP a los UAS y respuestas SIP a los UAC. Una petición puede atravesar varios proxys en su camino hacia un UAS. Cada uno tomará decisiones de enrutamiento, modificando la petición antes de enviarlo al elemento siguiente. Las respuestas se encaminarán a través del mismo sistema de proxys atravesados por la petición pero en el orden reverso.

Con los registros SRV del Servidor de Nombres de Dominio (DNS), se fija un proxy SIP a un dominio específico, para permitir que en las URLs se pueda llamar a dominios, en lugar de trabajar con un proxy específico. Ejemplo de una dirección sería `manuel.otero@empresa.com.ar`.

Existen dos tipos de Proxy Servers: Statefull Proxy y Stateless Proxy.

- Statefull Proxy: mantienen el estado de las transacciones durante el procesamiento de las peticiones. Permite división de una petición en varias (forking), con la finalidad de la localización en paralelo de la llamada y obtener la mejor respuesta para enviarla al

usuario que realizó la llamada.

- Stateless Proxy: no mantienen el estado de las transacciones durante el procesamiento de las peticiones, únicamente reenvían mensajes.

#### **4.3.4. SIP Redirect Server**

Se trata de un servidor que acepta solicitudes SIP, y retorna una respuesta que contiene una lista de las ubicaciones reales de un usuario en particular, al cual se desea contactar. Este servidor recibe las solicitudes y busca al destinatario en la base de datos de localización creada por el servidor Registrar. El emisor de la solicitud extrae luego la lista de destinos y envía otra petición directamente a ellas. Al contrario de un servidor Proxy, los servidores Redirect no pasan las solicitudes a otros servidores. Además no emiten solicitudes SIP ni aceptan llamadas SIP. En el caso de la devolución de una llamada, el Proxy Server tiene la capacidad de traducir el número del destinatario en el mensaje SIP recibido, en un número de reenvío de llamada y encaminar la llamada a este nuevo destino. Todo esto ocurre de forma totalmente transparente para el cliente origen; para el mismo servicio, el Redirect Server devuelve el nuevo número (número de reenvío) al cliente de origen quien se encarga de establecer una llamada hacia este nuevo destino.

#### **4.3.5. Mecánica de una llamada y establecimiento de comunicación**

La Figura 1 nos muestra el trabajo que realiza el protocolo SIP para poder establecer una comunicación. El usuario A realiza una llamada al usuario B. El SIP Proxy recibe la petición de comunicación por medio de un paquete INVITE, en dicho paquete se encuentra la dirección del usuario B, con el cual se desea contactar. El SIP Proxy reenvía dicha petición al usuario B, también utilizando un paquete de tipo INVITE. Inmediatamente el SIP Proxy le envía un paquete al usuario A indicándole que está tratando de contactarse. Cuando el teléfono del usuario B comienza a sonar, le envía un paquete al SIP Proxy indicándole que ha comenzado a sonar, éste le reenvía dicha situación al usuario A. De esta manera el teléfono del usuario A comienza a dar tono de llamado. Cuando finalmente el usuario B atiende le envía un paquete OK al SIP Proxy, a su vez se lo reenvía al usuario A, el usuario A envía la aceptación de la llamada al SIP Proxy por medio del paquete ACK, el SIP Proxy se lo reenvía al usuario B. En este punto el protocolo SIP logró su cometido (inició la sesión) y sale de escena. Ambos extremos tienen la dirección para poder establecer una comunicación directa entre ellos (comunicación P2P), la realizan por medio del protocolo multimedial RTP

(Real time Transport Protocol). Este protocolo permitirá el envío y recepción del audio, manejo de codec, manejo de video etc..

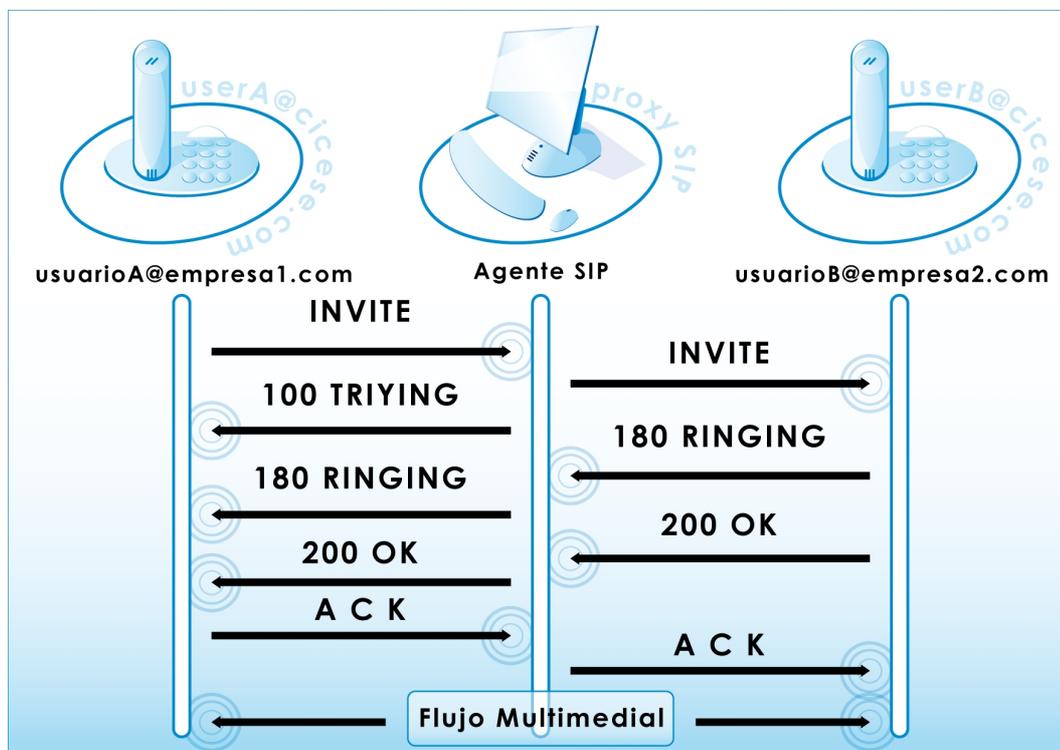


Figura 1: Inicio de sesión y flujo multimedial

#### 4.3.6. SIP Registrar Server

El SIP Register Server es un servidor que acepta solicitudes SIP REGISTER. SIP dispone de la función de registro de los usuarios. El usuario indica por medio de un mensaje REGISTER emitido al Registrar, la dirección donde es localizable (dirección IP). El "Registrar" actualiza una base de datos de localización. El registrador es una función asociada a un Proxy Server o a un Redirect Server. Un mismo usuario puede registrarse desde distintos UAs SIP, en este caso, la llamada le será entregada sobre el conjunto de estos UAs.

#### 4.3.7. Mecánica de una llamada compleja

La Figura 2 muestra los distintos pasos que se llevan a cabo en una

comunicación SIP hasta que finalmente los UAs pueden comunicarse entre ellos. Este es el típico caso en el cual los usuarios pertenecen a dominios distintos, por lo tanto aparecen dos servidores SIP Proxy uno que sirve a un dominio y el otro que sirve al otro dominio. Así supongamos que el usuario sip:1234@empresa1.com.ar llama al usuario sip:2341@empresa2.com.ar es muy probable que existan dos servidores SIP Proxy distintos uno para cada dominio. Esto es similar a si en lugar de direcciones SIP fueran direcciones de correo electrónico, existiría un MTA (Mail Transport Agent) que le da servicio a cada uno de los dominios.

Veremos cada uno de los pasos que se llevan a cabo por medio de protocolo SIP.

1. El UA le pregunta a el SIP Proxy que le da servicio.
2. El SIP Proxy pregunta a un SIP Redirect por las dudas que se haya movido a otro lugar el destinatario.
3. El SIP Redirect responde y le dice que le pregunte a otro SIP Proxy.
4. El SIP Proxy pregunta a otro SIP Proxy redirigido en el paso 2.
5. Este Proxy le pregunta al servicio de localización que le provea la dirección del SIP Proxy donde se encuentra el usuario destino
6. El servicio de localización le responde la dirección solicitada.
7. El SIP Proxy le pregunta a otro SIP Proxy que tiene la función de Registrar a los Uas.
8. El SIP Registrar le pregunta al UA si está en condiciones de poder recibir una llamada.
9. El UA le responde su estado, supongamos que está libre para recibir una llamada.
10. Se propaga la respuesta al UA que inicio el pedido
11. Ídem anterior
12. Ídem anterior
13. El UA origen le envía un reconocimiento al UA destino. Ahora conoce su dirección IP.
14. El protocolo SIP termina su trabajo y comienza a funcionar el protocolo RTP entre los dos pares.

Importante: En los dos casos hemos visto que el flujo multimedial se lleva a cabo entre los UAs, sin embargo pueden existir situaciones donde sea deseable que el flujo multimedial pase por un SIP Proxy, por ejemplo para

realizar transcodificación, o para efectos de billing. En este último caso los UAs pueden avisar al SIP proxy que se desconectó la llamada. En este caso se dicen los los SIP Proxy trabajan bajo control de estado (statefull).

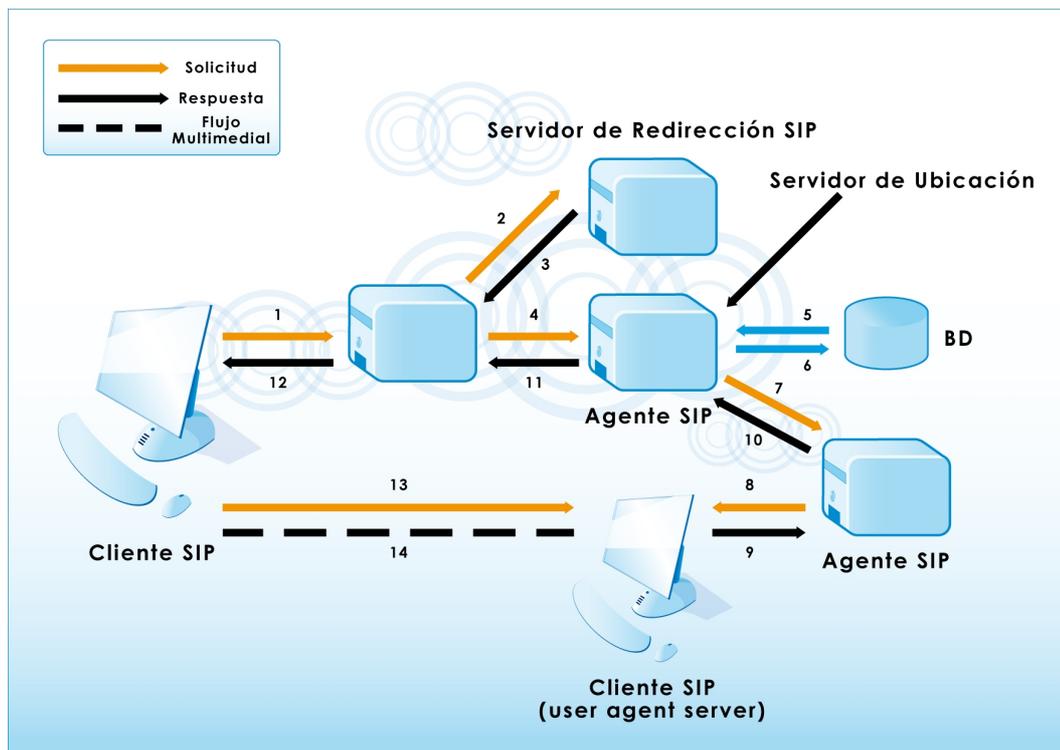


Figura 2: Conexión entre dos Agentes de Usuario

#### 4.3.8. Back to back User Agent - B2BUA

Un B2BUA es una entidad lógica que recibe una solicitud, la cual es procesada como un UAS. Para determinar como se debería responder a la solicitud, actúa como un UAC generando solicitudes.

Se ubica entre ambos extremos de una llamada telefónica. Divide a ésta en dos sesiones y media toda la señalización SIP entre ambos extremos, desde el comienzo hasta el fin de la llamada. Esto permite hacer un seguimiento de la llamada de principio a fin, permitiendo ofrecer funciones adicionales, tales como:

- Grabar la comunicación

- Administrar la llamada (facturación, desconexión, transferencia, etc.)
- Traducir codec entre los extremos de una llamada, esto permite que un UA trabaje con un codec y otro extremo trabaje con otro codec.
- Ocultar la información de la red entre los UAs (direcciones privadas, topología, etc.)

#### **4.3.9. Media Gateway**

El Media Gateway es un elemento auxiliar de una red SIP. Tiene la gran ventaja de poder realizar la unión entre una red de telefonía básica (PSTN), red digital de servicios integrados (RDSI) y una red IP, haciendo las veces de adaptación entre sus medios respectivos. Ejemplo de esto es una central telefónica IP o un ATA. En la Figura 3 se puede observar un Media Gateway, definido como “Gateway Voip”, el cual incluye a la PSTN formando parte de la red VoIP.

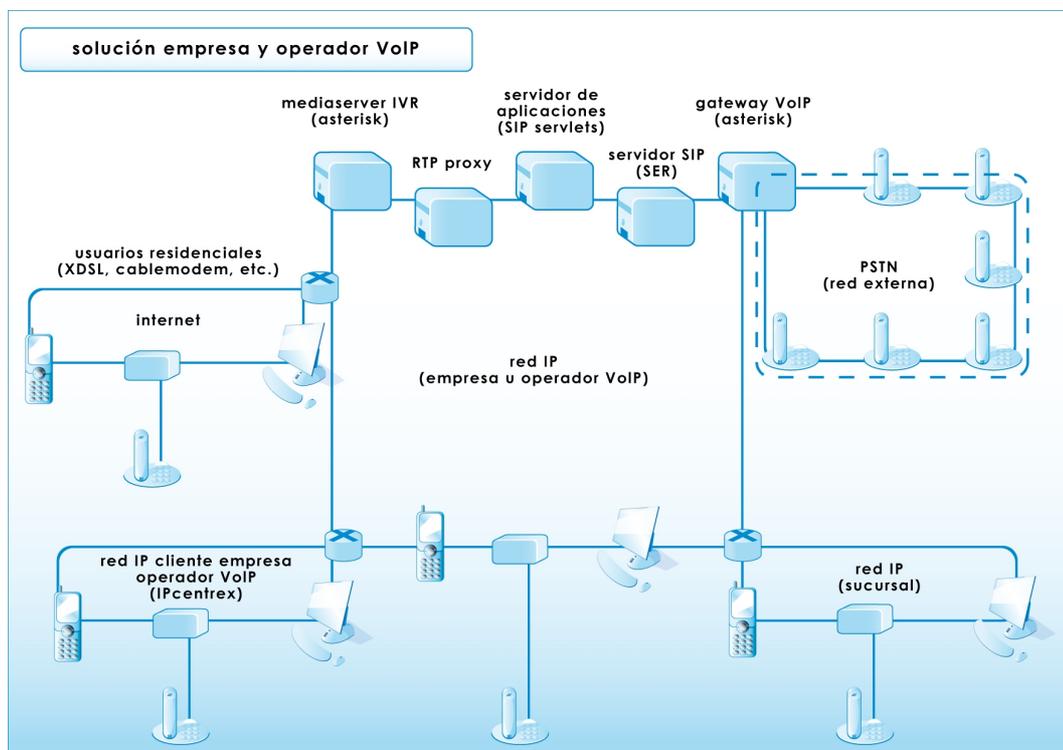


Figura 3: Operador de telefonía IP

#### 4.4. Flujo Multimedial (Media Streams)

Un vez que el protocolo SIP logra establecer la comunicación entre los extremos, comienza a llevarse a cabo la “conversación” propiamente dicha. SIP se utiliza exclusivamente para establecer (iniciar) la sesión (de ahí su nombre). A partir de ese momento se establecen flujos de datos entre ambos extremos (punto-a-punto). Ésto se lleva a cabo por medio del protocolo RTP (Real time Transport Protocol) y RTCP (Real time Transport Control Protocol), encargados del transporte de los flujos multimediales. Para nuestro caso el audio proveniente de la digitalización de la voz o el transporte de los tonos multifrecuencia correspondientes a las teclas del teléfono (DTMF).

El media streaming es un flujo multimedial que permanentemente arriba a un dispositivo, detrás de él existe un aplicativo encargado de capturar, procesar y entregar dicho flujo al usuario, ya sea en formato de video, audio o ambos. En algunos casos el aplicativo que gestiona el flujo

multimedial, puede almacenar parte de dicho flujo antes de entregarlo al usuario para poder absorber variaciones en la velocidad de transmisión(cache). Esto es responsabilidad de la aplicación que procesa el flujo multimedial.

#### **4.4.1. Flujo (Streaming)**

Este término se refiere a la posibilidad de poder ver u oír un flujo de datos sin la necesidad de tener que descargar primero todo el archivo. Esta técnica comienza en el año 1995 con la aparición del RealAudio, y sentó las bases de todos los servicios de valor añadido “bajo demanda”.

Uno de los elementos principales de esta tecnología son los códecs, que son pequeños programas (o plugins) incrustados (embeded) dentro del programa principal, que se encargan de descifrar o interpretar el contenido de un determinado flujo multimedial.

#### **4.4.2. Protocolo RTP**

El protocolo RTP significa protocolo de transporte en tiempo real (Real-time Transport Protocol), es un protocolo de capa de aplicación y sirve para poder transportar el flujo multimedial. Es un protocolo extremadamente adaptable a todo tipo de situación. Uno de los campos del encabezado describe el tipo de carga útil, el mismo se utiliza como índice en una tabla de perfiles de media que describe el formato de carga útil. Los mapeos de carga útil para audio y vídeo están especificados en el RFC 1890.

Debido a que es un protocolo que se transporta por medio de UDP, tiene adosado una estampilla de tiempo (timestamp) para poder llevar el orden de los paquetes.

Para el caso de aplicaciones donde hay más de dos entidades involucradas (multidifusión) el protocolo tiene un mecanismo para poder identificar cual es la fuente del flujo desde donde proviene. El control de errores, tasa de transmisión, etc. son responsabilidad de la aplicación que genera el tráfico multimedia.

Sin embargo, el protocolo RTP no realiza todas las funciones, para esto descansa en otro protocolo que realice esto. Por ejemplo:

- No garantiza el envío ni que los paquetes lleguen fuera de orden
- No proporciona mecanismos para el envío a tiempo.
- No garantiza la calidad de servicio

#### **4.4.3. Protocolo RTCP**

Este se encarga de verificar la calidad del servicio y de proporcionar información acerca de los participantes en una sesión de intercambio de datos. RTCP no está diseñado para soportar todas las necesidades de comunicación de una aplicación, solo las básicas. La principal función de RTCP es proporcionar una retroalimentación útil para mantener una calidad de distribución adecuada, por ejemplo, se puede usar para controlar un mecanismo de adaptación de codificación que responda a las condiciones de la red. Los paquetes de RTCP se envían de modo que el tráfico en la red no aumente linealmente con el número de agentes participantes en la sesión, es decir el intervalo de envío se ajusta de acuerdo al tráfico.

Este protocolo recoge estadísticas de la conexión y también información como por ejemplo bytes enviados, paquetes enviados, paquetes perdidos o jitter entre otros. Una aplicación puede usar esta información para incrementar la calidad de servicio (QoS), ya sea limitando el flujo o usando un códec de compresión más baja (velocidad adaptativa). RTCP por sí mismo no ofrece ninguna clase de cifrado de flujo o de autenticación.

Una de las funciones del protocolo SIP es negociar el puerto por el cual van a fluir los datos multimediales. De tal forma que la aplicación abre dos puertos uno par para el flujo de paquetes RTP y otro impar y seguido del anterior, para el flujo de paquetes de control.

#### **4.4.4. Protocolo IAX2**

Como se ha visto con los dos protocolos anteriores se abren dos puertos por cada sesión multimedial. Supongamos una Central Telefónica IP que tiene 400 llamadas cursándose en paralelo, la pérdida de rendimiento puede llegar a ser más que apreciable, en especial la pérdida de ancho de banda producto de los encabezados de cada datagrama de cada comunicación. El otro gran inconveniente que tiene RTP es la imposibilidad de atravesar un firewall que implementa NAT (Network

Address Translation). Mark Spencer, el creador de la Central Telefónica Asterisk, se percató de estos problemas y diseñó un único protocolo que no tuviese los problemas de SIP y RTP y lo utilizó en su diseño.

IAX implementa la señalización y los datos por el mismo flujo (in-band), es un protocolo binario en lugar de texto plano. No realiza negociación de puertos, para ello utiliza un único flujo de datos sobre UDP (generalmente bajo el puerto 4569) para comunicarse entre los extremos. Las respuestas se envían desde donde ellas vinieron en lugar de tener que negociar una dirección IP no normalizada, esta dirección IP se encuentra dentro del encabezado IP en lugar de la carga útil del protocolo.

Soporta trunking y multicanalización sobre un único enlace. Esto significa que un único datagrama IP puede entregar información para más de una llamada, reduciendo la sobrecarga IP sin crear latencias adicionales.

Mark Spencer, lo diseñó para poder unir sus centrales IP, de ahí su nombre Inter Asterisk Exchange, pronto se fue adoptando como un protocolo alternativo para VoIP, hoy en día vienen teléfonos IP y softphone con soporte para éste protocolo como una alternativa a SIP-RTP. Es un protocolo más rígido que SIP-RTP ya que las extensiones se tienen que agregar en la norma.

Para dar una idea de la eficiencia de IAX, puede duplicar la cantidad de llamadas que se cursan en un enlace de 1Mb utilizando códecs G.729.

#### **4.4.5. Capas del Protocolo**

Hemos visto que los protocolos SIP, SDP, RTP y RTCP son protocolos de capa de aplicación. La Figura IV muestra un esquema y ubicación en las distintas capas.

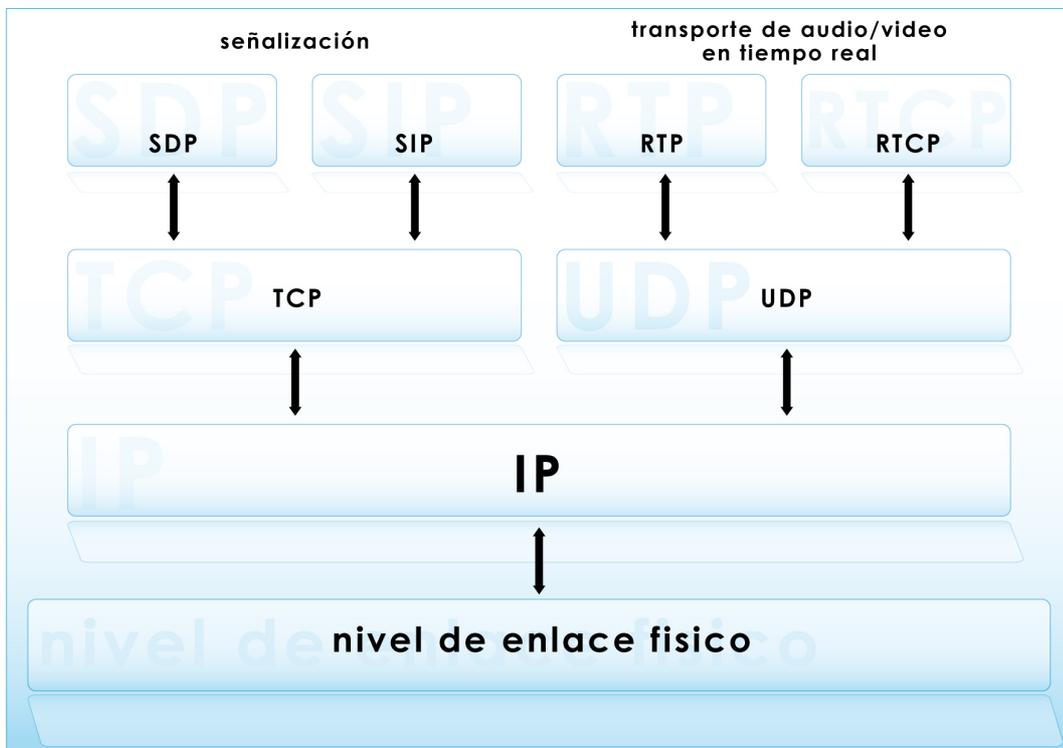


Figura 4: Distribución de los protocolos en capas

## 5. Empresas de Telefonía IP

### 5.1. Introducción

Desde hace ya unos años, han explotado la cantidad de empresas que ofrecen telefonía IP en el mundo y en especial en Argentina. El gran argumento para su penetración comercial es la drástica reducción de costos en llamadas nacionales y con precios irrisorios en las internacionales comparados con la telefonía tradicional.

De esta forma están ingresando a grandes empresas que consumen comunicaciones nacionales y/o internacionales. Pero se da un fenómeno, en la parte residencial también está teniendo una gran aceptación debido a las llamadas internacionales, ya que, por este medio pueden estar en contacto con sus parientes por una fracción del costo anterior.

### 5.2. Presencia Local en Oficina Virtual con Ventaja Real

Se puede tener *Presencia Local*, por medio de un número telefónico perteneciente a la PSTN (Red Telefónica Pública) de Barcelona, España. Ese será el número telefónico local de nuestra empresa que atenderá las llamadas de todos nuestros clientes en Europa. Pero en realidad esa llamada hará sonar un teléfono en nuestra oficina de Mendoza, por lo tanto estamos teniendo una *Oficina Virtual* en Europa. A su vez si nosotros tuviéramos que realizar una llamada a algún país de Europa podemos salir por dicho número. Todos esto nos provee de una *Ventaja Real* y tangible. Con beneficios no solamente financieros sino presencia local, lo cual nos da una ventaja competitiva y mejor imagen comercial.

Por ejemplo, una pequeña productora de artesanías de Jujuy puede tener un número local en Buenos Aires donde pueda recibir los pedidos de los clientes, un hotel marplatense puede tener numeración local en varias ciudades de la Argentina y una agencia de turismo argentina puede tener número local en Santiago de Chile, San Pablo y Miami. Las opciones son infinitas, los beneficios muy concretos.

### **5.3. Servicios**

La primera empresa en masificar este servicio fue Skype. Si bien es una plataforma cerrada y propietaria, el modelo de negocio ha sido exitoso. No obstante hoy, debido a la competencia con la verdadera telefonía IP, está tratando de mezclarse por medio de pasarelas que unan ambas tecnologías.

En Argentina existen gran cantidad de proveedores de telefonía IP como Camundanet, LlamadaIP, Call-IP, IPTel, SmarTel, etc. todas ofrecen más o menos los mismos servicios.

### **5.4. Servicios básicos y gratuitos**

Si queremos empezar a experimentar el servicio de telefonía IP, se puede comenzar con un servicio básico y que hasta el momento es gratuito. Al subscribirse a Camundanet.com o IPTel.org, uno tiene la posibilidad de obtener una extensión de su Central Telefónica IP. Por lo tanto con un softphone o un teléfono IP nos podemos registrar a dicho interno. Con esto una persona que esté suscrita al mismo servicio con la misma empresa nos podrá llamar con el número del interno provisto. No importa si este registro es llevado a cabo por una persona que se encuentra a la vuelta de nuestra casa o en Japón. Otra posibilidad, es llamar desde otra red poniendo el nombre de dominio de la empresa a la cual estamos suscriptos. Por ejemplo supongamos que nos suscribimos a Camundanet.com y nos da el número de extensión 545789, otro usuario que también esté en la misma red nos llamará utilizando dicho número, pero si se encuentra registrado con otro proveedor, por ejemplo IPTel.org deberá llamarnos como 545789@camundanet.com.

Lo interesante de esto es que el lugar físico desde donde se suscribe a la extensión, no importa. Puedo estar en mi hogar y suscribirme por medio de un ATA (Analog Telephon Adapter) y un teléfono analógico, luego puedo ir a mi trabajo y suscribirme por medio de un teléfono IP o puedo viajar a Buenos Aires y suscribirme por medio de un softphone desde la notebook conectada en el hotel. En cualquier caso el número telefónico viaja conmigo, sería una especie de número itinerante ("roaming"). Con esto se consigue la ubicuidad, donde yo estoy está mi teléfono y el número es siempre el mismo. Lo fundamental el costo de la llamada es la misma.

## 5.5. Servicios avanzados

Se puede ir a un esquema más avanzado que el anterior. Para ello se puede contratar un número telefónico perteneciente a la PSTN (Red Telefónica Pública) de un determinado lugar, pero en lugar de contratarla con la Empresa Telefónica local, se contrata con una empresa de telefonía IP. De esta forma tengo un número de la PSTN unido a un número de extensión de una central telefónica IP. Así cuando alguien llame al número perteneciente a la PSTN, sonará el teléfono IP con el cual se ha registrado, sea éste un softphone, un teléfono IP o un autómata telefónico.

Por supuesto que al tener un número telefónico de la PSTN, también se puede usar para salir por dicho número. Los esquemas de utilización de esto siempre son de tipo prepago. La gran ventaja de estas empresas es que tienen convenios con empresas en todo el mundo, realizándose verdaderos “clusters” de empresas potenciando aún más su oferta. <sup>1</sup>

## 5.6. Numeración local DID

Cuando se contrata una numeración local en una ciudad de cualquier país con una empresa de Telefonía IP, se obtiene un DID.

DID (Direct Inward Dialing – Discado Entrante Directo), es un servicio ofrecido por las compañías telefónicas para usar con los sistemas de central telefónica de los clientes (PBX), en donde la compañía telefónica (telco) asigna un rango de números asociados con una o más líneas telefónicas. Esto permite que una empresa contrate 10 líneas telefónicas pero obtenga 40 números de marcado.

De esta forma se asignan números personales para algunos empleados, encaminando la llamada entrante al interno de dicha persona. Es lo que vulgarmente se conoce como “número telefónico directo”. Obviamente que la cantidad máxima de llamadas simultáneas que se pueden cursar son 10, en este caso.

Bajo este mismo esquema, las empresas de telefonía IP, contratan con las telcos un conjunto de, por ejemplo, 100 líneas telefónicas, pero

---

<sup>1</sup> Para un listado completo de empresas a nivel mundial puede ser contactado el siguiente sitio:  
<http://www.sipbroker.com/sipbroker/action/providerWhitePages>

con 200 números o más. De esta forma las incumbentes IP pueden sobrevender (overbooking), líneas abaratando con esto el tramo de telefonía fija que ellas tienen.

## **5.7. Aplicaciones**

Por medio de una central IP nos podemos registrar a un interno provisto por una empresa de telefonía IP. En este caso podríamos decir que nuestra central esta funcionando como un teléfono IP común, pero con una gran ventaja, cuando ingrese una llamada a dicha extensión podré administrarla con la central. Esto permite encaminar la llamada a la PSTN local, a un celular, a otra extensión o a una aplicación. Esto último es lo más interesante porque se puede disparar una aplicación que se encargue de gestionar la llamada. Estas aplicaciones son los Automatas Telefónicos.

Bajo éste esquema se desarrollará el Automata Telefónico motivo de esta tesis.

## **6. Central Telefónica Asterisk**

### **6.1. Introducción**

En el Capítulo 4 observamos que uno de los elementos de la telefonía IP, es el Media Gateway. Una central telefónica es una especialización de medio de enlace ya que nos permite unir la red de telefonía básica con la red IP. Otro Media Gateway es un encaminador SIP (SER).

Para nuestra aplicación utilizaremos una central telefónica Asterisk.

### **6.2. Plataforma de desarrollo**

Asterisk es un software de central telefónica IP basado en código abierto. Corre sobre sistemas operativos Linux. Asterisk representa una forma nueva de poder acercarse al mundo de la telefonía IP. Introduce el concepto de código abierto en un campo que tradicionalmente era 100% propietario.

Las dos principales ventajas de Asterisk son:

- Reducción de costos (el software es gratis)
- Rápido desarrollo. Miles de programadores en todo el mundo contribuyen al desarrollo continuo, permitiendo que Asterisk mejore todos los días.

Con Asterisk es posible construir un sistema de telefonía de altas prestaciones a una fracción del costo del de una plataforma tradicional. Puede operar perfectamente dentro de un entorno de telefonía IP, así como con teléfonos y líneas analógicas. Este modelo de telefonía permitirá cambios dramáticos en el mercado de centrales telefónicas privadas (PBX). Los integradores de tecnología están ofreciendo soluciones que compiten con sistemas propietarios de grandes fabricantes, como Alcatel o Siemens. Este nuevo enfoque representará un problema a las empresas que inviertan en costosos sistemas telefónicos.

Asterisk posee muchas funcionalidades que crecen día a día.

Además de las prestaciones básicas (ruteo de llamadas, reenvío, música en espera), Asterisk puede funcionar como un gateway de comunicaciones integradas. Por ejemplo, tiene la funcionalidad de digitalizar y enviar por email los mensajes de voicemail. También es posible brindar soluciones IVR (Interactive Voice Response) y mucho más.

Existen distintos tipos de usuarios de Asterisk, desde pequeñas empresas con unos pocos internos hasta call centers. Puede funcionar junto con una PBX tradicional. Esto protege la inversión original y al mismo tiempo brinda todas las nuevas funciones de las centrales IP.

La Central Telefónica Asterisk se implementa en una PC, por lo tanto Asterisk es todo software, que utiliza el hardware de la PC como su soporte electrónico y acepta tarjetas de comunicación para poder hacer la unión con la PSTN. Estas tarjetas vienen de diversos tamaños y tecnologías pero fundamentalmente se dividen en dos: las analógicas (con 1, 4, 12 o 24 puertos FXS o FXO) y las que aceptan una trama digital E1 o T1 (con 1, 2 y 4 puertos).

Existen empresas que han desarrollado PABX, tomando como base a Asterisk, implementando tarjetas para telefonía analógica o trama digital y desarrollando interfaces de administración amigables para el usuario. En Argentina una de esas empresas es Eolix Technologies que comercializa un producto desarrollado por ellos, las centrales telefónicas Eolix. Se han vendido un sin número de centrales y han exportado a varios países de latinoamérica.

### 6.3. Características de Asterisk

Cuando se ingresa al mundo de Asterisk, lo primero que se percibe es la cantidad de prestaciones que tiene esta central telefónica. No existe ningún producto comercial tan completo, de ahí su gran difusión. En el siguiente cuadro se muestran sus distintas funciones:

Receptor de Alarmas	Sistema de monitoreo de alarmas	Listas Negras	La central corta si detecta que están llamando desde un determinado número
Autenticación	Diversos mecanismos de identificación y autenticación de usuarios	Transferencia no supervisada	Pulsando una secuencia de transferencia, transferirá la llamada a otro número sin esperar

			que éste conteste.
Preatendedor	Establecer un menú con opciones para los usuarios o esperar por la operadora	Transferencia supervisada	Pulsando una secuencia de transferencia, se podrá comunicar con otro interno mientras queda en espera la llamada que está cursando. Si cuelga quedan conectadas las dos llamadas.
Registro de llamada detallados (CDR)	Registra todas las llamadas realizadas. Útil para tarificación o auditoría	Desvío de llamada si la extensión está ocupada	
Desvío de llamada si la extensión no responde		Desvío variable de llamada	Puede elegir un interno al cual desviar la llamada. *21*<número>, #21# borra el desvío
Supervisión de llamadas	Sólo de voz, voz y datos, entrenamiento o conferencia realizado por un supervisor	Estacionar llamadas	Estacionar una llamada en un estacionamiento de llamadas para luego recuperarla desde otro teléfono
Encolado de llamadas	Asterisk define diversas políticas y estrategias para la administración de colas en los agentes (operadores)	Grabación de llamadas	Grabación de llamadas perteneciente a un interno o grupo de internos.
Recuperación de llamadas	Recuperar llamadas estacionadas	Encaminamiento de llamadas DID	Direccionamiento de llamadas entrantes en función al número marcado
Encaminamiento de llamadas ANI	Direccionamiento de llamadas entrantes en función al prefijo geográfico	Escucha de llamadas	Escuchar, en forma cíclica, un conjunto de canales con llamadas en curso.
Llamada en espera	Al estar hablando cuando llega una segunda llamada podrá atender ésta y dejar a la actual en espera. Si no es atendida irá al voicemail o se encaminará a otra extensión.	Identificación del llamante	Mostrar el número telefónico de la persona que llama
Bloqueo de llamante	Inhabilitar una llamada en función al número del cual proviene.	Identificación del llamante durante la	Aparecerá en el display del teléfono el número de la persona que llama.

		llamada en espera	
Tarjetas de llamadas	Controlar el saldo de un usuario antes y durante la llamada.	Puente telefónico	Unión con diversas tecnologías
Acceso directo al sistema interno (Direct Inward System Access (DISA))	Permite que una persona externa ingrese al sistema y pueda realizar llamadas como si estuviese conectado a la PBX.	Marcado por nombre	Ingresando las tres últimas letras del apellido le dará un menú con posibles nombres. Directory().
Tonos de llamada distintivos	Distintos tonos de llamada en función al CID entrante	Función No Molestar	Las llamadas entrantes se encaminan al Voicemail, operador u otro interno.
ENUM	Mapeo de números telefónicos de la RTB a VoIP	Recepción y transmisión de Fax	Posibilidad de recibir y enviar Fax utilizando Asterisk
Lógica de extensiones flexible	Adicionar lógica a las extensiones	Respuesta vocal interactiva (IVR)	Posibilidad que un usuario pueda navegar una aplicación por medio de los dtmf.
Música en diversas operaciones	Posibilidad de interpretar música en diversas operaciones y situaciones sin dispositivos externos.	Macros	Definir subrutinas que realizan un determinado trabajo
Marcado telefónico predictiva	Es un sistema que toma un conjunto de números, los marca y se los entrega a un agente.	Privacidad	Si el sitio remoto no tiene CID, le pide que le ingrese el número desde donde está llamado.
Protocolo de establecimiento abierto (Open Settlement Protocol)	Soporte de tarificación para voz y fax por VoIP.	Conversión de protocolos	Posibilidad de que los extremos trabajen con distintos codecs.
Captura de llamada remota	Capturar una llamada que suena en otra extensión	Soporte de oficina remota	Posibilidad de suscribir teléfonos a la PABX desde el exterior.
Extensiones itinerantes (Roaming)	Posibilidad de darle un número y contraseña para conectarse desde cualquier parte.	Encaminamiento en función de la identificación del llamante	

Mensajería SMS	Posibilidad de enviar mensajes SMS. Necesita de un gateway SMS	Deletreo / Habla	Deletrear o leer un mensaje
Texto a Voz (TTS)	Posibilidad de leerle un texto a una persona que llama. Por ejemplo un mensaje guardado en una BD.	Buzón de Voz	Posibilidad de definir un voicemail para cada usuario del sistema
Pasarelas VoIP	Puede actuar como un puente entre teléfonos IP y la PSTN.		

## 6.4. Plan de Discado (Dial Plan)

El plan de discado es el corazón de Asterisk. Es donde se define como se administrarán las llamadas. Consiste en una lista de instrucciones o pasos que Asterisk debe seguir. Estas instrucciones se disparan a partir de dígitos ingresados en un canal o aplicación. Podemos dividir el plan de discado en cuatro partes:

- Contextos
- Extensiones
- Prioridades
- Aplicaciones

### 6.4.1. Contextos

Tiene un papel muy importante en la seguridad y organización. También definen alcances y permite separar diferentes partes del plan de discado. Los contextos están ligados directamente a los canales. Cuando ingresa una llamada por un canal, ésta es procesada dentro de un contexto. De manera que en un contexto me permite definir como voy a gestionar la llamada cuando ingrese a la central. Un contexto puede heredar definiciones de otro contexto. Esto se hace para no tener que volver a definir algo que ya se ha definido.

## 6.4.2. Extensiones

Dentro de cada contexto, se definen las extensiones, las cuales podríamos tomarlas como los internos a los cuales se puede llamar. Cada extensión dispara un evento o conjunto de eventos. Veamos la siguiente definición:

```
exten=>4523,1,Dial(SIP/4523,20)
exten=>4523,2,voicemail(u4523)
exten=>4523,3,hangup()
exten=>4523,101,voicemail(b4523)
```

Una instrucción `exten=>`, describe cual es el próximo paso para la llamada. El 4523, es el conjunto de dígitos que se recibió por el canal (número llamado). Los números 1, 2, 3 y 101, son las prioridades que determinan el orden de ejecución de los comandos. En nuestro caso, discando 4523 hará sonar un teléfono SIP registrado bajo el número 4523, si no es contestado dentro de los 20 segundos, ejecutará la prioridad 2, en este caso le dirá que deje un mensaje en el correo de voz, cuando cuelgue pasará a la prioridad 3, que es desconectar el canal. Si en el paso uno la extensión estuviese ocupada, le suma 100 a la prioridad y salta a dicha prioridad, en este caso 101 y ejecuta correo de voz, pero indicando que estaba ocupada la línea (busy).

## 6.4.3. Prioridades

Son pasos numerados en la ejecución de cada extensión. Cada prioridad ejecuta una función específica. Generalmente estos números de prioridad comienzan en una y se van incrementando de uno en uno, aunque, como ya vimos, esto no es requerido.

## 6.4.4. Aplicaciones

Es la parte fundamental dentro de Asterisk. Es lo que lleva a cabo el trabajo. Asterisk trae un conjunto de aplicaciones ya definidas divididas en distintas categorías, entre las que se encuentran: comandos generales, facturación, administración de llamadas, presentación del llamante, ADSI, manejo de BD, integración con aplicaciones de terceros, control de flujo, administración de los timeouts, manejo de variables, ejecutar sonidos, grabar conversaciones, comandos SIP, comandos ZAP,

correo de voz, administración de colas, etc.

## **6.5. Respuesta Interactiva de Voz (IVR)**

Como ya vimos anteriormente, el plan de discado es un excelente medio para configurar la administración de las llamadas. Es una herramienta extremadamente flexible y potente. No obstante, si aún se necesita mayor grado de flexibilidad la única forma de llevar a cabo esto es por medio de un IVR.

Consiste en un sistema telefónico que es capaz de recibir una llamada e interactuar con el humano a través de grabaciones de voz, utilizando como medio de interacción los DTMF que les envía cada tecla presionada por la persona con la cual interactúa. Es un sistema de respuesta interactiva, orientado a entregar y/o capturar información automatizada a través del teléfono IP o tradicional, permitiendo el acceso a los servicios de información y operaciones autorizadas, las 24 horas del día.

### **6.5.1. Servicios**

El IVR es la típica máquina que nos responde con una voz grabada cuando llamamos a una central o que nos recibe en la banca telefónica o cualquier servicio que comience con “fondo ...”. Según las opciones que el usuario le ingresa lo deriva a un centro de atención telefónica o a otra central telefónica.

Es muy utilizado en grandes empresas para extender el horario de atención las 24 horas del día, por ejemplo dar información a los clientes o proveedores sobre ciertas informaciones o poder ingresar información por parte de la persona que llama.

También son muy utilizados en los Contact Center para poder interactuar con el cliente, solicitarle cierta información y en función a la información ingresada derivarlo al operador apropiado para resolver su problema.

Como se trata de un programa, paralelamente a todo lo anterior, puede grabar información en una BD o recuperar datos de una BD y puede informar el estado de avance de un reclamo, o cualquier aplicación que se nos ocurra implementar.

## 6.6. Asterisk Gateway Interface – Interfaz de conexión con Asterisk

El Asterisk Gateway Interface (AGI) es una interfaz para extender aún más las funcionalidades que tiene Asterisk, por medio de diferentes lenguajes de programación, tales como: Perl, Python, C, Ruby, Java, Bourne Shell, etc.

Estos programas externos interactúan con órdenes propias del PD más las funcionalidades propias de AGI. Los programas AGI, se utilizan para implementar lógica avanzada, comunicarse con BD y acceder a otros recursos externos, permitiendo de esta forma realizar tareas que de otra forma serían imposibles de realizar.

### 6.6.1. Fundamentos de una comunicación con AGI

En lugar de contar Asterisk con una API desarrollada en uno o varios lenguajes de programación, se comunica con AGI por medio de los descriptores de archivos conocidos como STDIN, STDOUT y STDERR. De la misma forma que CGI (Common Gateway Interface) lo hace con un Web Server.

Generalmente para iniciar un guión AGI, se hace usando un comando dentro del PD, su nombre es agi ( ):

```
exten => 123,1,Answer()  
exten => 123,2,AGI (agi-test.rb)
```

Esto significa que si arriba una llamada a la extensión 123, se ejecutará un programa cuyo nombre es agi-test.rb. El directorio donde se encuentra estos programas están bajo: /var/lib/asterisk/agi-bin. Una vez que comienza la ejecución recibe por medio de la STDIN un conjunto de información que es muy importante para el desarrollo posterior del programa, entre otras le envía:

- tipo de canal: si la llamada proviene de un canal Sip o un canal Zap
- nombre del programa

- número desde donde se llama
- interno desde donde se llama
- etc.

Esta información llega en un formato <atributo>:<valor>, de manera que generalmente se guarda como configuración inicial de la llamada. Por ejemplo el CID (Caller ID) se puede utilizar para verificar una lista negra. Un ejemplo de esto es: `agi_channel: Zap/1-1`, en este caso el atributo es `agi_channel` y su valor es `Zap/1-1`.

### **6.6.2. API's para programar con AGI**

Si bien desde el programa se puede trabajar directamente con la STDIN y la STDOUT (por el propio diseño de AGI), es poco práctico realizar esto. Para resolver este problema, se han desarrollado un sin número de API's en diversos lenguajes que facilitan al programador su trabajo. Existen API's implementadas para AGI en más de 8 lenguajes de programación y a su vez en algunos lenguajes existen hasta 5 diferentes bibliotecas de funciones o métodos. La mayoría de ellas están desarrolladas orientadas a objetos, de manera que uno crea el objeto al comienzo del programa y a partir de ese punto puede utilizar todos los métodos del objeto.

### **6.7. Asterisk Manager Interface (AMI)**

La interfaz de administración nos permite poder relacionarnos con Asterisk y realizar tareas administrativas por medio de su interfaz. Es una interfaz conceptualmente similar a AGI, pero su incumbencia es exclusivamente la de administración. Esto ha permitido a terceros desarrollar decenas de aplicaciones que pueden administrar Asterisk. Liberando al equipo que desarrolla Asterisk el desarrollo de estos aplicativos.

## 7. Autómatas Telefónicos

### 7.1. ¿Qué es un Autómata Telefónico?

Como planteábamos en el marco conceptual, la palabra autómatas viene del griego *automatos* y significa espontáneo o con movimiento propio. Una de las acepciones de éste vocablo nos dice: “*Máquina que imita la figura y los movimientos de un ser animado*”, ésta es precisamente la definición que se aplicará en el presente trabajo. Obviamente el “ser animado” al que esta haciendo referencia ésta definición es una persona, es decir, si no tuviera al autómatas telefónico tendría que ser reemplazado por una persona que contestara la llamada y que a su vez provea información o ingrese datos que le están solicitando desde el otro extremo de la comunicación.

De manera que Autómata Telefónico se define como un dispositivo electrónico y de software capaz de contestar una llamada y poder brindar información a un potencial usuario o ingresar información provista por el usuario, siendo el usuario la persona que llama a un determinado número telefónico. Obviamente que el programa utiliza una Base de Datos para poder leer o registrar información.

### 7.2. Forma de interacción

Un usuario disca un número telefónico que es respondido por un AT. Se ejecuta la aplicación que está relacionada con el número marcado. Como parte de la aplicación, le informa las opciones disponibles por medio de archivos de audio pre-grabados o dinámicamente accediendo a texto y convirtiéndolo a audio (TTS).

El usuario interactúa con el sistema por medio de dos métodos bien diferenciados:

- Utilización de tonos (DTMF – Dual Tone Multi Frequency) generados al presionar un botón del teclado telefónico.
- Reconocimiento de voz, generalmente esto se hace con vocablos

cortos.

De las dos tecnologías la más difundida es la interacción por medio de tonos generados por el teléfono cuando el usuario presiona una tecla del teclado telefónico.

### **7.3. Usos frecuentes**

Los AT se usan para dar servicios a altos volúmenes de llamados, reducir el costo y mejorar la relación con proveedores y/o clientes. Ejemplos típicos son el: fono banking, televoto, solicitar información o registrar pedidos. La lista es tan larga como aplicaciones se deseen implementar.

Grandes empresas utilizan los servicios de AT para extender las horas de atención. Para identificar y segmentar a los usuarios que llaman. La posibilidad de poder identificar a un usuario permite la personalización de los servicios de acuerdo al perfil del cliente. También permite la elección de servicios automatizados. Permite cosas como: “Espere en línea, escoja un servicio automático o le llamaremos a la brevedad”, en éste último caso se le puede solicitar información del porque de su llamado o datos personales. El uso de CTI (Computer Telephone Integration) le permite a los AT poder buscar en una BD e identificar al llamante. Esto es preciso en el 80% de los casos y totalmente determinista cuando se utiliza un teléfono móvil, ya que se puede identificar directamente a la persona que llama. Cuando esto no es posible se utilizan otros métodos como ingresar un PIN y si el sistema requiere de mayor seguridad se le puede adicionar el pedido de una contraseña. Los servicios de información por número llamante permiten la personalización de la aplicación.

Los AT se pueden utilizar como pretendedores a una PABX reemplazando o ayudando al operador. Esto se puede utilizar en hospitales y grandes negocios para reducir el tiempo de espera de las llamadas entrantes. Funciones adicionales permiten ubicar a personas dentro del hospital y transferir la llamada directamente a dicha persona.

En el mundo del entretenimiento se utilizan los AT para realizar votaciones sobre juegos determinados, pronóstico climático, etc.

El campo de las encuestas ha encontrado una buena aplicación de los AT, puede realizar llamadas a un conjunto de números obtenidos en forma aleatoria o dirigida de una BD y presentar lo que se denomina Formularios Audibles, con lo cual el encuestado puede responder

presionando teclas en su teléfono. Se pueden generar grandes volúmenes de información con costos bajísimos.

## 7.4. Tecnologías

### 7.4.1. Voz pre-grabada

Existen dos formas de relacionarse con el usuario que llama, uno es a través de este medio, mensajes pre-grabados. En función de la información que tiene que transmitir puede concatenar varios archivos de audio. Por ejemplo, para informarle el saldo de una cuenta bancaria, el sistema le dice: “ Su saldo es de doscientos pesos con treinta centavos”, si bien el usuario esto lo percibe como un único sonido, en realidad puede estar formado por la unión de varios archivos de sonido, incluso para los números existen rutinas similares como las que existen para convertir número a texto, pero en este caso es número a audio (number2speech). En este caso todos los archivos de audio están pregrabados.

### 7.4.2. Voz Sintetizada

Existe la posibilidad de no tener grabado ningún archivo de audio, sino a medida que el sistema necesita “pronunciar” un texto, éste es pasado por un programa conocido como texto a voz (TTS – Text To Speech).

Durante los últimos años se ha convertido en una verdadera ciencia dentro de la parte lingüística produciéndose avances más que notorios. Desde las primeras voces sintetizadas que nos hacen recordar a las voces latosas de los robots a las actuales voces, es increíble el avance que se ha tenido. Por esta razón, los autómatas telefónicos cada vez más se están apoyando en estos nuevos programas de sintetización de la voz. De esta manera los mensajes son guardados como texto en la BD y se pronuncian en el momento de necesitarlo. Esto permite tener AT multilingües.

Esto permite una mayor plasticidad al momento de diseñar los AT, incluso en función de quien llame se les puede dar acentos distintos, como colombiano, chileno o argentino e incluso, obviamente, distintos idiomas, inglés, francés, italiano, etc.

En el mundo open source existe un producto llamado **Festival**, es

un sistema de síntesis de voz de propósito general para múltiples lenguajes desarrollado originalmente por el Centro de Investigación de Tecnologías del Lenguaje de la Universidad de Edimburgo. La Universidad Carnegie Mellon así como otros centros de enseñanza han realizado contribuciones substanciales al proyecto<sup>2</sup>.

### 7.4.3. Otros considerandos tecnológicos

Otras tecnologías incluyen la posibilidad de reproducir información dinámica tales como correos electrónicos, reportes, estado del tiempo, etc.

Un AT se puede utilizar de diferentes maneras:

1. Equipamiento instalado en el cliente
2. Equipamiento instalado en las centrales de la PSTN
3. Proveedor de servicios de aplicaciones (ASP)

Muchas aplicaciones emplean esta tecnología, como los bancos para: ingresar pedidos, identificación y redirección del llamante, consulta de saldos, reservas de pasajes.

Los sistemas de tratamiento de correos electrónicos, son totalmente diferentes ya que son una comunicación persona-a-persona, en cambio los AT son comunicaciones máquina-a-persona. Los AT, se pueden diseñar de manera interactiva, de forma tal que luego de ingresada una información le pregunte si el mensaje que acaba de grabar lo desea escuchar, editar, enviar o eliminar.

Los Distribuidores Automáticos de Llamadas (ACD), a menudo son el primer punto de contacto cuando se llama a grandes empresas. Utiliza almacenamientos en BD para dar distintas bienvenidas o pasar anuncios, generalmente redirige una llamada sin hacer ninguna consulta. Un AT puede pasar anuncios en función de quien llama, solicitar datos de entrada. Esta información se puede usar para redirigir la llamada a un conjunto de personas entrenadas para atender dicha llamada (skillset).

Los AT se utilizan para atender las llamadas entrantes en un centro de llamadas (call center), identificando las necesidades de la persona que llama. La información se obtiene del llamante, vgr. número de cuenta. Puede responder preguntas simples como saldos o información pre-

---

<sup>2</sup> Actualmente el proyecto puede ser consultado en <http://www.festvox.org>.

grabada, todo esto sin la intervención de un operador. Los números de cuenta se comparan con el CID por cuestiones de seguridad y se le pueden hacer preguntas adicionales si la llamada no proviene de un determinado número.

La lógica a seguir en una llamada administrada por un AT puede ser implementada de varias maneras. En algunos casos se implementa utilizando lenguajes de programación o scripting, últimamente se están desarrollando medios más focalizados tales como VoiceXML, SALT o T-XML. Existen incluso SDK (Software Development Kit) diseñados para esta tarea, donde una persona sin habilidades de programación pueda implementar sencillos AT, no obstante el tiempo ha demostrado que ante complejas respuestas solamente se pueden implementar con una buena lógica de programación, desarrollados por programadores avezados.

## **7.5. Críticas**

A menudo se critica a los AT por no ser útiles o ser difíciles de usar. En ambos casos se debe a un pobre diseño y a no tener en cuenta las necesidades de los usuarios. Un AT correctamente diseñado debería prestar un excelente servicio a los usuarios con un mínimo de tiempo y complejidad. En la fase de diseño se tienen que cumplir determinadas normas para optimizar los resultados.

## **8. Diseño de un AT**

### **8.1. Introducción**

Antes de comenzar a desarrollar un AT, tenemos que pasar por la fase del diseño. Existen un conjunto de pautas relacionadas con el buen arte para el diseño de los AT que nos permitirán que nuestro aplicativo sea de mayor utilidad y que el grado de rechazo, normal que se tiene contra ellos, sea el menor posible.

### **8.2. Pautas de Diseño**

#### **8.2.1. Sea sucinto pero no sacrifique significado**

La mejor transacción llevada a cabo por un AT es la más corta. Antes de comenzar a grabar los menús de su aplicación, escriba cada sentencia y analicela al mayor grado de detalle. La existencia de cada palabra debe ser justificada. Cada sentencia debe utilizar la frase más corta posible sin perder el sentido.

#### **8.2.2. Priorice las tareas importantes**

Ponga primero las tareas importantes, de forma tal que un usuario no tenga que navegar demasiado por los menús antes de escuchar lo que desea. Lo más importante puede también medirse en frecuencia. Debe colocarse al inicio la opción más frecuente.

#### **8.2.3. Enfoque su aplicación**

Existe la tendencia a hacer una aplicación que contemple la mayor cantidad de opciones. Esto genera en el llamante confusión, para evitar ésto se deberían desarrollar aplicaciones cortas y bien enfocadas por temas o incumbencias.

#### **8.2.4. El sonido tiene que ser lo más natural**

Las opciones dentro del menú tienen que ser expresadas de forma natural, sin grandes exageraciones y a todas darles la misma importancia. Grabe todo un menú de una sola vez ya que es mucho más claro y sin variaciones en el tono, timbre e intensidad. Evite ruidos, señales de baja intensidad. Utilice locutores profesionales.

#### **8.2.5. Evite largos silencios**

Una separación de más de 3 segundos puede hacer pensar a un usuario que el sistema no funciona. Por otro lado todo sin separación es lo más molesto para el oído y en especial para entender con claridad. El paso correcto para cada situación es fundamental.

#### **8.2.6. Construya una maqueta**

No espere a que el sistema esté terminado para poner en producción su sistema, al contrario primero desarrolle una maqueta navegable y que un conjunto de usuarios lo utilice y le de su opinión sobre el aplicativo.

#### **8.2.7. Utilice normas**

Existen muchos estudios sobre la forma de diseñar los Automatas Telefónicos para que sean más útiles, claros y comprensibles. Estos estudios se han resumido en varias normas tales como la HFES 200.4 o la ISO/IEC 13714. Antes de comenzar con el diseño de un Automata Telefónico no deje de leer ambas.

#### **8.2.8. Utilice una voz profesional**

En un AT, el audio lo es todo. Mala calidad de grabación, ruido de fondo y una voz no entrenada pueden crear una mala impresión en la mente del usuario y negarse a utilizar el aplicativo. Utilice una voz

profesional, grabe los archivos de audio en un estudio de grabación.

### **8.2.9. Evite usar publicidad**

La cantidad de personas que entran a estos sitios puede ser muy grande y uno puede tentarse a hacer publicidad. Es la peor opción que puede tomar y el peor lugar para hacer publicidad. Lo pondrá al usuario de mal humor. Evítelo, existen canales más adecuados para esto.

### **8.2.10. AT adaptativos e inteligentes naturales**

No hay nada más molesto que navegar un AT pasando de una opción a otra sólo para que el sistema le diga que el horario de atención es hasta las 17 horas. Si tiene departamentos que cierran a distintas horas se deberá adaptar a esta situación para no llegar al punto de decirle al usuario que el departamento está cerrado, luego de haber presionado varias opciones.

### **8.2.11. Nunca se confíe en el menú**

La mayoría de los usuarios se puede confundir dentro de la aplicación, por lo tanto una buena salida es poner un menú de ayuda o aclaración para cuando el usuario no sabe que hacer. Si bien siempre son pocos los que pueden llegar a utilizar estas ayudas, siempre son necesarias. Nunca utilice la excusa “¡Pero si el menú los guía el menú, todo es muy comprensible!”.

## **8.3. Normas de Diseño**

Existe un conjunto de normas que todos utilizan a la hora de diseñar un AT. Éste nos dan la idea del buen arte y buenas prácticas de diseño para evitar el rechazo de los usuarios al sistema.

Nunca las pase por alto, escoja la norma que más se adecue a sus necesidades.

## 9. Lenguajes de Programación

### 9.1. Introducción

La lógica de un AT se implementa por medio de un lenguaje de programación. En este sentido existen gran cantidad de lenguajes que se pueden utilizar y también scripting. Lo que uno tiene que tener en cuenta es que exista una API para relacionarse con Asterisk.

### 9.2. Lenguajes más utilizados

La siguiente es una tabla con los lenguajes y los nombres de las bibliotecas para realizar desarrollo de AT.

Lenguaje	Bibliotecas
Java	OrderlyCalls Asterisk-java Brazil Web Framework
Pascal	TpasAGI
Perl	Asterisk Perl Library Asterisk::config callback daemon
PHP	Asterisk AGI PHP phpAGI AGIPhp5 PHPASTLIB AsteriskPHP
Python	Py-Asterisk

	Python AGI PyAstre StarPy
Ruby	<a href="#">Adhearsion</a> Batphone ruby-agi RAGI RAMI Telegraph
C	CAGI Quivr AgiAddOns
.NET	Mono-Tone nAsterisk

### 9.3. Generadores de AT

Siempre existen empresas que pregonan que el desarrollo de software es un proceso que se puede automatizar utilizando para ello generadores de código. Esto es realmente cierto para pequeñas aplicaciones, pero cuando la lógica de negocio detrás de un AT es mediana, se justifica el desarrollo del AT en cualquier lenguaje de programación.

Con una herramienta de desarrollo, el límite lo impone la misma herramienta, en cambio el límite de diseño con un lenguaje de programación lo impone la imaginación del analista o programador.

## 10. Desarrollo de un AT

### 10.1. Introducción

Recordemos que un Autómata Telefónico no es más que un programa especializado. La central telefónica IP le provee servicios y soporte, lo que aporta el AT es la lógica de negocio. En este tipo de aplicación la interacción con una BD es fundamental. Para desarrollar nuestra aplicación utilizamos el lenguaje Ruby, como ambiente integrado de desarrollo (IDE) se ha escogido Netbeans y la Base de Datos es MySQL.

### 10.2. Ruby

Ruby es un lenguaje de programación orientado a objetos, dinámico y reflexivo. Está inspirado por la sintaxis de Perl y Smalltalk y también comparte algunos rasgos de Python, Lisp, Dylan y CLU. Es un lenguaje de scripting de un sólo paso. Su implementación pertenece al Open Source y está escrito en C.

El lenguaje fue desarrollado por Yukihiro Matsumoto, comenzó en el año 1993 y lo liberó en 1995. Cada año se suman muchísimos programadores a sus filas, se estima que es el 10<sup>mo</sup> lenguaje más utilizado.

#### 10.2.1. Características sobresalientes

En Ruby, todo es un objeto. Se le puede asignar propiedades y acciones a toda información y al código. La programación orientada a objetos llama a las propiedades *variables de instancia* y las acciones son conocidas como *métodos*. La orientación a objetos pura de Ruby se suele demostrar con un simple código que aplica una acción a un número.

```
5.times { print "Nos *encanta* Ruby -- ¡es fuera de serie!" }
```

En muchos lenguajes, los números y otros tipos primitivos no son objetos. Ruby sigue la influencia del lenguaje Smalltalk pudiendo poner

métodos y variables de instancia a todos sus tipos de datos. Esto facilita el uso de Ruby, porque las reglas que se aplican a los objetos son aplicables a todo Ruby.

Ruby es considerado un lenguaje flexible, ya que permite a sus usuarios alterarlo libremente. Las partes esenciales de Ruby pueden ser quitadas o redefinidas a placer. Se puede agregar funcionalidad a partes ya existentes. Ruby intenta no restringir al desarrollador.

Por ejemplo, la suma se realiza con el operador suma (+). Pero si prefiere usar la palabra *sumar*, puedes agregar un método llamado *sumar* a la clase *Numeric* que viene incorporada.

```
class Numeric
  def sumar(x)
    self.+(x)
  end
end
y = 5.sumar 6
# ahora y vale 11
```

Los operadores Ruby son simples conveniencias sintácticas para los métodos. Son altamente flexibles para ser redefinidos.

Ruby tiene un conjunto de otras funcionalidades entre las que se encuentran las siguientes:

- manejo de excepciones, como Java y Python, para facilitar el manejo de errores.
- un verdadero mark-and-sweep garbage collector para todos los objetos de Ruby. No es necesario mantener contadores de referencias en bibliotecas externas.
- escribir extensiones en C para Ruby es más fácil que hacer lo mismo para Perl o Python, con una API muy elegante para utilizar Ruby desde C. Esto incluye llamadas para embeber Ruby en otros programas, y así usarlo como lenguaje de scripting. También está disponible una interfaz SWIG.
- puede cargar bibliotecas de extensión dinámicamente si lo permite el sistema operativo.
- tiene manejo de hilos (threading) independiente del sistema

operativo. De esta forma, tienes soporte multi-hilo en todas las plataformas en las que corre Ruby, sin importar si el sistema operativo lo soporta o no, ¡incluso en MS-DOS!

- Ruby es fácilmente portable: se desarrolla mayoritariamente en GNU/Linux, pero corre en varios tipos de UNIX, Mac OS X, Windows 95/98/Me/NT/2000/XP, DOS, BeOS, OS/2, etc.

### **10.3. NetBeans**

NetBeans es un IDE (Integrated Development Environment) con características sobresalientes. En primer lugar esta herramienta, de calidad superior, es open-source. Está íntegramente desarrollada en Java, lo que da un altísimo grado de portabilidad, de esta manera, se puede correr bajo Linux, Windows, Mac OS X y Solaris.

#### **10.3.1. Características destacables**

Es el IDE más intuitivo, según análisis de los especialistas. Indica errores de sentencias incompletas a medida que ingresa el código, comparable al rasgo IntelliSense que posee el Visual Studio de Microsoft. El inconveniente a solucionar es que cuando encuentra un error resalta toda la línea en lugar del lugar donde ha encontrado el error. Tiene integrado en su plataforma un servidor Apache Tomcat, lo que permite probar las aplicaciones Web sin tener que salirse de la herramienta. Cuenta con explorador integrado de BD.

No obstante, el rasgo más sobresaliente es la posibilidad de poder integrar códigos de terceras partes programadas especialmente para NetBeans. Esto se lleva a cabo por medio de programas que se insertan en tiempo de ejecución (plugins). Existen más de cien plugins que se pueden utilizar dentro de NetBeans.

#### **10.3.2. Control de versiones**

Una característica deseable en todo proyecto de desarrollo de software, es el control de versiones, esto nos permite controlar las modificaciones realizadas a nuestro desarrollo, poder ver qué se ha modificado, eliminado o agregado. Poder reconstruir un soft a una determinada fecha o versión. En NetBeans, esto es un punto extremadamente cuidado. La Figura 5 nos muestra las modificaciones

realizadas entre nuestro programa local y la última versión guardada (1.3). El color celeste muestra la sección de código modificada, el color verde muestra el código agregado y el color rosado el código eliminado. Además de informarnos nos permite poder deshacer alguna de estas acciones y volverla al estado anterior.

Comparación entre versión actual y anterior:

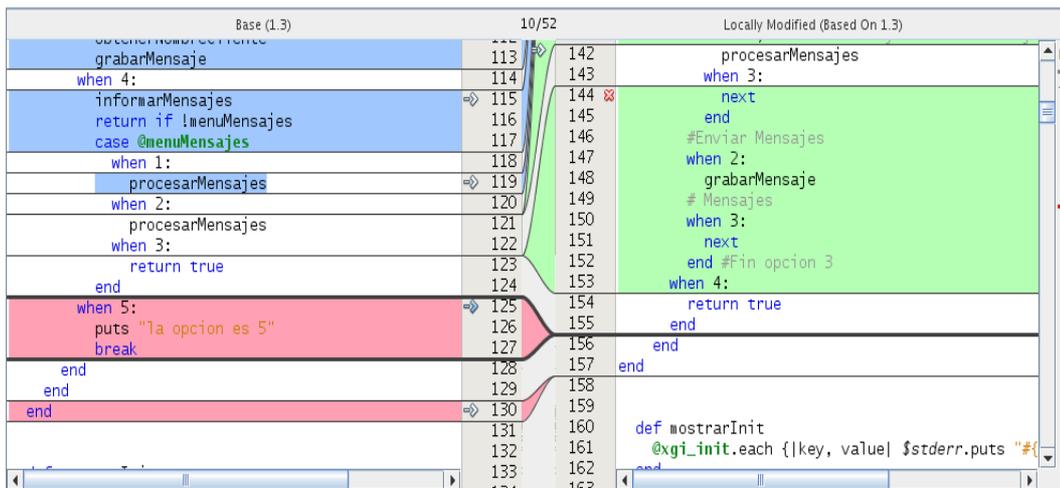


Figura 6: Diferencia entre versiones

La Figura 6, nos muestra cómo eliminamos la palabra *puts*, dejamos la palabra *sql* y agregamos el código que se encuentra en verde en la segunda columna.



Figura 7: Eliminar y adicionar código

Esto es una herramienta gráfica de extremada utilidad ya que nos permite saber los cambios que se han realizado en nuestro software.

## **10.4. MySQL**

MySQL es un sistema de gestión de base de datos relacional, multihilo y multiusuario. Posee más de seis millones de usuarios. Tiene un esquema de licenciamiento dual. Por un lado se ofrece bajo la GNU GPL, pero, empresas que quieran incorporarlo en productos propietarios pueden adquirir una licencia que les permita ese uso. Está desarrollado en su mayor parte en ANSI C.

A diferencia de proyectos como Apache, donde el software es desarrollado por una comunidad pública, y el copyright del código está en poder del autor individual, MySQL es propiedad y está patrocinado por una empresa privada, que posee el copyright de la mayor parte del código.

### **10.4.1. Principales usos**

Inicialmente carecía de rasgos considerados esenciales para las BDR, tales como integridad referencial y transacciones. Pero justamente esto era lo que hacía que su velocidad no tuviese comparación, era extremadamente rápida y por ser Open Source fue aceptada rápidamente para proyectos Web.

Poco a poco los rasgos de los que carecía se fueron incorporando. Entre las características disponibles en las últimas versiones se pueden destacar:

- Amplio subconjunto del lenguaje SQL. Se incorporan extensiones propietarias para facilitar algunas tareas.
- Disponibilidad en gran cantidad de plataformas y sistemas.
- Diferentes opciones de almacenamiento según si se desea velocidad en las operaciones o el mayor número de operaciones disponibles.
- Transacciones y claves foráneas.
- Conectividad segura.
- Replica.

- Búsqueda e indexación en campos de texto.

Posee API para todos los lenguajes de programación más conocidos. Esto le permite ser incorporado en la mayoría de los proyectos en forma nativa. Se puede acceder también por vía ODBC, pero no se aconseja, debido a la lentitud de este protocolo.

## **10.4.2. Redundancia**

Los sistemas de alta disponibilidad necesitan algún medio de redundancia, es decir, cuando un medio de almacenamiento falla, inmediatamente se conmuta a un sistema que esté sincronizado con el anterior y pueda así continuar trabajando como si nada hubiese ocurrido. En este sentido existen varios esquemas de redundancia dentro de MySQL.

### **10.4.2.1. Redundancia Master/Slave**

En este tipo de redundancia o replica, existe un único Maestro y varios Esclavos. Las consultas a la BD se pueden hacer tanto al maestro como a los esclavos ya que no se modifican los datos de la base. En cambio, cuando se modifican datos de la base únicamente se tienen que llevar a cabo sobre el maestro. Los esclavos se actualizan a partir de la información de la bitácora que va dejando el maestro luego de realizarse una transacción. Este tipo de esquema siempre es una operación asíncrona, de esta forma existe una demora en propagarse las modificaciones de datos realizadas en el maestro. Cuando el maestro se cae no existen mecanismos automáticos para realizar la conmutación.

Lo que generalmente se hace en estos casos es modificar un DNS para que se utilice otro maestro en otra dirección IP. Sobre esta nueva dirección IP se desactiva MySQL como esclavo y se lo levanta como maestro. Esta operación puede tomar algunos minutos y puede llegar a ser una operación manual o por medio de un guión (script) perteneciente al Sistema Operativo.

### **10.4.2.2. Redundancia Multi-Master**

Este tipo de redundancia tuvo mucho éxito ya que se podía definir

más de un maestro al mismo tiempo. Cuando uno de ellos dejaba de funcionar se podía conmutar a otro maestro y la operación no se detenía (non-stop). Sin embargo la sincronización entre los maestros era compleja y el retardo podía llegar a ser lo suficiente como para que se pudiesen leer datos aún no actualizados entre los maestros. Peor aún, cuando un maestro salía de servicio era muy complejo actualizarlo, al momento exacto en que había salido de servicio.

#### **10.4.2.3. Redundancia utilizando Clusters**

La tecnología de clusters se comenzó a desarrollar hace ya algunos años con el objetivo de poder unir conjunto de computadoras para realizar trabajos cooperativos. De esta manera se dice que eran “débilmente acopladas” y que se podía ver a ese conjunto como si fuese una única computadora.

El objetivo de los clusters era incrementar la potencia de cálculo (o procesamiento) uniendo varios procesadores provenientes de distintas computadoras, pero cada una de ellas era autónoma. El otro gran objetivo era lo relacionado con la seguridad. De esta forma se podía contar con un sistema de Alta Disponibilidad (High Availability) y sistemas inmunes a fallas (fail-over). De tal forma que cuando un nodo dejara de participar del cluster, el sistema como un todo pudiera seguir funcionando sin ningún problema.

MySQL Cluster es una versión de alta disponibilidad y alta redundancia adaptada para el entorno de computación distribuida. Usa el motor de almacenamiento **NDB Cluster** para permitir la ejecución de varios servidores MySQL en un cluster. Este motor de almacenamiento está disponible en las distribuciones binarias a partir de la versión 5.0.

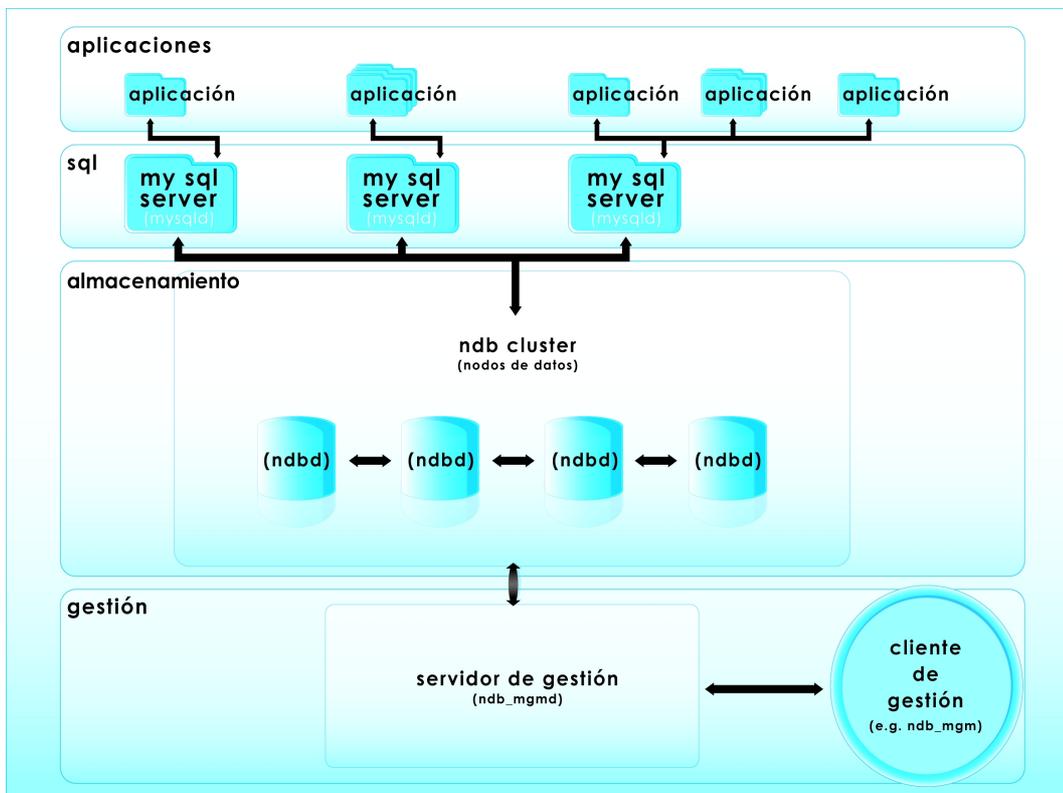


Figura 8: Distribución de clientes y almacenamiento bajo clusters

## **11. Autorización de Prestaciones Médicas**

### **11.1. Introducción**

Los clientes de cualquier tipo de empresa cada vez reclaman más y mejores servicios, y es por eso que existen departamentos en las empresas dedicados a satisfacer esas necesidades. En gran parte con la incorporación de tecnología se pueden prestar mejores servicios. Una empresa que no sea capaz de satisfacer las necesidades de sus clientes está destinada a ser reemplazada por otra más atenta a los deseos de los mismos.

Los afiliados a empresas de Medicina Prepagas u Obras Sociales, son más proclives a este comportamiento, ya que a la hora de consumir los servicios lo hacen por problemas de salud, están impacientes por resolver su problema ya que éste afecta su calidad de vida y trastoca los demás aspectos de su vida. De manera que siempre hay que brindarle mayores servicios.

En el Capítulo 3, se vio la problemática común que afecta a muchos afiliados de empresas de salud. Ahora nos dedicaremos a ver con mayor grado de detalle la lógica de negocio que se ha implementado en el AT.

### **11.2. Autómata finito**

Un autómata finito o máquina de estado finito es un modelo matemático de un sistema que recibe una cadena constituida por símbolos de un alfabeto y determina si esa cadena pertenece al lenguaje que el autómata reconoce. Existen dos tipos distintos de autómatas: finitos determinísticos y autómatas no determinísticos.

#### **11.2.1. Autómata finito determinístico**

Es aquel autómata finito cuyo estado de llegada está unívocamente determinado por el estado inicial y el carácter leído por el autómata.

### 11.2.2. Autómata finito no determinístico

Es aquel que presenta cero, una o más transiciones por el mismo carácter del alfabeto. Un autómata finito no determinista también puede o no tener más de un nodo inicial.

### 11.2.3. Fundamento Matemático

Este tipo de autómatas tiene las siguientes características:

1. Es un conjunto finito de estados y un conjunto de transiciones de estado a estado, que se dan sobre símbolos de entrada tomados de un alfabeto  $\Sigma$ .
2. Para cada símbolo de entrada existe exactamente una transición a partir de cada estado (posiblemente de regreso al mismo estado).
3. Un estado, por lo general denotado como  $q_0$  es el estado inicial, en el que el autómata comienza.
4. Algunos estados (tal vez ninguno, proceso continuo) están designados como final o de aceptación.

Un autómata finito determinístico es una quinta upla  $(Q, \Sigma, \delta, q_0, F)$  donde:

- $Q$  es un conjunto finito de estados.
- $\Sigma$  un alfabeto de entrada finito.
- $q_0$  elemento de  $Q$ , estado inicial.
- $F \subseteq Q$  el conjunto de estados finales o de aceptación.
- $\delta$  es la función de transición.  $\delta : Q \times \Sigma \rightarrow Q$  que determina el único estado siguiente para el par  $(q_1, \sigma)$  correspondiente al estado actual  $q_1$  y la entrada  $\sigma$ .

Un conjunto de funciones de transición  $\Delta = [\delta_0, \dots, \delta_n]$ : Que determinan el cambio de estado del autómata según cierta condición. Se simboliza de la siguiente manera:

$$\delta_i: (q_h, b) \Rightarrow q_t$$

Donde:

- $\delta_i$  representa a la función de transición
- $q_h$  es el estado en que se encuentra actualmente el autómata
- $b \in \Sigma$  es el símbolo de entrada introducido al autómata
- $q_t$  es el estado nuevo del autómata, o sea, el nuevo estado actual

Generalmente el término autómata finito determinístico se abrevia como DFA (Deterministic Finite Automata). Usaremos  $M = (Q, \Sigma, q_0, F, \delta)$  para indicar el conjunto de estados, el alfabeto, el estado inicial, el conjunto de estados finales y la función asociadas con el DFA M.

Se puede construir un diagrama para que ayude a determinar los distintos miembros o cadenas del lenguaje. Tal diagrama tiene la forma de un grafo dirigido con información adicional, y se llama diagrama de transición. Los nodos del grafo corresponden a los estados del DFA y se usan para señalar, en ese momento, hasta qué lugar se analizó la cadena. Por lo general  $q_0$  es el estado inicial, marcando con una flecha ( $\rightarrow$ ), el comienzo del autómata. Algunos estados están designados como final o aceptación indicados por un doble círculo. Los símbolos del alfabeto son las etiquetas de los arcos del grafo. Si cuando ha sido tratada la cadena en su totalidad se termina en un estado de aceptación entonces la cadena es aceptada por el lenguaje. Si M es un AFD, entonces el lenguaje aceptado por M es:

$$L(M) = \{w \in \Sigma^* \mid w \text{ es aceptada por } M\}.$$

Por tanto,  $L(M)$  es el conjunto de cadenas que hacen que M pase de su estado inicial a un estado de aceptación.

### **11.3. Autómata telefónico como una máquina de estados finito**

En el punto anterior vimos la base matemática de nuestro AT. Ahora nos concentraremos en relacionar cada uno de los elementos de una máquina de estados finitos. Para nuestro AT tenemos:

Elementos	Descripción
$q_0$	Estado inicial, el AT se encuentra a la espera de que ingrese una llamada.
$\Sigma$	El alfabeto de un AT son los números del 0-9, * y #
Q	El conjunto de todos los estados posibles en que se puede encontrar un AT
F	Los estados que no pueden evolucionar a otro estado, por ejemplo no existe el afiliado.
$\delta$	Función de transición. Representa la lógica de negocio que permite evolucionar de un estado $q_n$ al estado $q_{n+1}$ . Por ejemplo al ingresar un afiliado se fija en la BD y como se encontraba en ella pasa al estado de solicitar el código de la prestación.
$\sigma$	Entrada ingresada en un estado. En nuestro AT es el conjunto de dígitos, por ejemplo 132345434500 es el número de afiliado que sirve como entrada para el estado $q_{12}$ , que representa la verificación del afiliado.

Tabla 11.1: Descripción de Elementos

## 11.4. Lógica de Negocio

La lógica de negocio es la parte del sistema que se encarga de las tareas relacionadas con los procesos. Es la forma más sencilla de referirse a los procesos de gestión que se llevan a cabo en un programa de computadoras. Anteriormente a esto se lo denominaba el "backoffice". A medida que los sistemas avanzaron en complejidad se prefirió llamarlos Sistemas de Capas. En ellos, existía una capa de presentación (frontend) y una capa de proceso (backend). Luego esto cambió y comenzó a definirse a los sistemas como multicapas. En ellos cada capa (inferior) brindaba servicio a otra (superior), dispuestas horizontalmente. Posteriormente este modelo evoluciona y se incluyen capas verticales que le dan servicios a varias capas horizontales.

En el caso del AT también está diseñado en capas debido a que

existe una capa que se relaciona con la comunicación (socket, bind y red) por donde ingresa una llamada, un thread que se dispara cuando ingresa una llamada y se encarga de gestionar dicha llamada, capas que se encargan de la persistencia de los datos y capas que implementan los pasos del proceso en función a los valores ingresados por el teclado.

### 11.4.1. Inicio del autómata

Durante la fase de inicio, el AT abre un puerto (socket) y se queda a la espera de peticiones. En la siguiente tabla vemos el código para llevar a cabo esto:

#### Programa Principal

```
1 #
2 # Programa Main
3 #
4
5 port = (ARGV[1] || 4573).to_i
6 host = (ARGV[0] || '0.0.0.0')
7 server = TCPServer.new(host, port)
8 while (session = server.accept)
9   Thread.new(session) do |s|
10     Qualium.new(s)
11   end
12 end
```

Tabla 11.2: Descripción de proceso principal

Como podemos ver el programa principal tiene escasas seis líneas. En las líneas 5-6 inicializa las variables port y host. La línea 7 crea el objeto *server*, es de tipo *TCPServer*, éste objeto tiene toda la descripción del socket y el host sobre el cual se ha abierto. Introduciendo la orden *netstat -tanp* nos muestra:

## Procesos TCP Server

```
# netstat -tanp

Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp    0      0 127.0.0.1:8000         0.0.0.0:*               LISTEN      2322/nasd
tcp    0      0 127.0.0.1:2208        0.0.0.0:*               LISTEN      2245/hpiod
tcp    0      0 0.0.0.0:51210        0.0.0.0:*               LISTEN      3239/skype
tcp    0      0 0.0.0.0:111          0.0.0.0:*               LISTEN      1862/rpcbind
tcp    0      0 0.0.0.0:56626        0.0.0.0:*               LISTEN      1906/rpc.statd
tcp    0      0 0.0.0.0:4573         0.0.0.0:*               LISTEN      5445/ruby
tcp    0      0 127.0.0.1:2207        0.0.0.0:*               LISTEN      2250/python
tcp    0      0 192.168.0.1:47702    190.25.17.143:7995     ESTABLISHED 3239/skype
tcp    0      0 192.168.0.1:42540    209.85.163.91:80      TIME_WAIT   -
tcp    0      0 :::1:38400           :::*                    LISTEN      4556/java
tcp    0      0 :::2401              :::*                    LISTEN      2287/xinetd
tcp    0      0 :::22                :::*                    LISTEN      2270/sshd
tcp    0      0 :::24444             :::*                    LISTEN      4556/java
```

Tabla 11.3: Listado de la orden netstat

Como podemos ver, la línea que esta subrayada se encuentra un proceso ruby (el interprete de nuestro programa) que se encuentra en estado de LISTEN, escuchando peticiones TCP sobre el socket 4573 sobre todas las direcciones IP (0.0.0.0) del servidor. En la línea 8, el método accept del objeto server queda a la espera de una conexión, en cuanto arribe una conexión al socket 4573 se iniciará un thread, sobre dicho thread correrá un objeto de tipo Qualium donde le pasará las características de la comunicación TCP. El objeto Qualium implementa toda la lógica de negocios.

Para que se pueda establecer una llamada, tiene que estar configurado el canal SIP por el cual va a ingresar dicha llamada. En el archivo de configuración sip.conf, se establecen las características del interno SIP. Veamos la siguiente tabla:

```
1 [376]
2 type=peer
3 context=llamadaip
4 nat=no
5 dtmfmode=auto
6 insecure=port,invite
7 qualify=yes
8 disallow=all
9 allow=gsm
10 allow=alaw
11 allow=ulaw
12 allow=g729
```

Tabla 11.4: Descripción del Canal SIP

En la línea 1 se define el nombre del canal, lo siguiente son las características que le asignamos al canal. La línea 2 establece que el canal es solamente entrante, la línea 3 define el nombre del contexto con el cual está relacionado este canal, dicho contexto se define en el archivo `enxtensions.conf`.

En la siguiente figura vemos un softphone X-Lite que está registrado como usuario `qualium`, desde acá el usuario podrá llamar al interno `376@qualium.com.ar`, si está registrado en otra central, si estuviera registrado en la misma central donde corre el AT, simplemente ingresaría el interno 376.



Figura 9: Teléfono softphone

Teniendo un softphone activo podrá llamar a la extensión 376, dicha extensión está relacionada con el contexto *llamadaip*, nos deberemos fijar en el archivo *extentions.conf* como está configurado dicho contexto, esto es lo que relaciona la extensión SIP con el AT:

Tabla: Descripción del contexto *llamadaip*

```

1  [llamadaip]
2
3  exten => s,1,Answer
4  exten => s,n,Background(qualium/interno_erroneo)
5  exten => s,n,Hangup() ; Extensióspecial (Sin Destino)
6  exten => 376,1,Answer ; Answer the line
7  exten => 376,n,agi(agi://localhost:4573/100)
8  exten => 376,n,Hangup
9
10 exten => 1376,1,Answer ; Answer the line
11 exten => 1376,n,agi(agi://localhost:4573/110)
12 exten => 1376,n,Hangup
13
14 exten => 388,1,Answer ; Answer the line
15 exten => 388,n,agi(agi://localhost:6773/100)
16 exten => 388,n,Hangup

```

Tabla 11.5: Descripción contexto *llamadaip*

La tabla 11.5 describe el contexto *llamadaip* definido dentro del archivo *extensions.conf*. Como se ve en un mismo contexto se puede definir más de una extensión o visto de otro modo varias extensiones pueden tener el mismo contexto. El conjunto de extensiones conforma lo que se llama el plan de discado (Dial Plan), para la central IP.

En las líneas 3-5 damos un mensaje de error si la extensión a la cual se llama no es ninguna de las siguientes: 376, 1376 o 388. Si se marca la extensión 376 lo primero que se hace es responder la llamada (línea 6), luego se llama a la función *agi* y se le pasa como argumento *agi://localhost:4573/100*, *agi://* le indica que haga uso del protocolo *agi* pasando la llamada a un TCP Server que se encuentra escuchando en la misma máquina (*localhost*) donde corre la Central Telefónica, en el socket 4573 y le pasa como argumento el valor 100. En este momento le atenderá el proceso perteneciente al AT que se ha diseñado. El argumento con el valor 100 se utiliza para tomar alguna decisión lógica dentro del AT. Por ejemplo en la línea 11 le pasa como argumento un valor de 110, con esto yo puedo hacer cosas distintas en función que una proviene de la extensión 376 y el otro de la 1376. Por ejemplo en el 376 están las autorizaciones médicas y en la extensión 1376 se encuentran las autorizaciones bioquímicas.

De esta forma se ha cerrado el circuito de configuraciones que se tienen que llevar a cabo en la Central Telefónica IP.

## 11.4.2. Proceso

En el punto anterior vimos el arranque del AT y lo que sucede en la central telefónica IP cuando arriba una llamada. Ahora veremos el proceso del AT propiamente dicho. En la línea 10 de la Tabla 11.2 vemos que cuando arriba una llamada la atiende un thread que ejecuta a un objeto *Qualium*, este es el objeto que tiene la lógica de negocio de nuestro proceso. Veremos algunas de las partes de nuestro AT.

Toda la lógica de negocio se encuentra dentro de la clase *Qualium*, alguna de sus partes son:

```
class Qualium
    VERSION = "0.1.9"
    @globalConfig = nil
    @connectionStatus=false
```

```

@xgi_init=[]
# Parametros IVR
def initialize(session)
  # Inicializo el archivo de logger
  @log=Logger.new("application.log", 5, 1000*1024)
  @log.level = Logger::DEBUG
  @log.datetime_format = "%d-%m-%Y %H:%M:%S"
  @log.debug("Inicializacion")

begin
  @horaInicioSesion=Time.now.strftime("%H:%M:%S")
  @cliente_init={}
  @session=session
  # Abro la BD
  @bd=DB.new("aplix","localhost","qualium","capriati")
  @bd.connection
  @voipgi=Voipgi.new(session,"loglib.log")
  @n2s=Num2speech.new(session,@voipgi)
  #@file = File.new("testfile.txt", "w")
  #@file.puts "entramos aca"
rescue
  @log.fatal("Error en inicializacion #{$!} linea
#{__LINE__}")
end
main
playSound("despedida")
end

```

Tabla 11.6: Descripción clase Qualium

En la tabla 11.6 tenemos el comienzo de la definición de la clase Qualium. Ésta, define tres variables locales que se utilizarán en diversos momentos, estas son @globalConfig, @connectionStatus y @xgi\_init. De éstas @xgi\_init es un arreglo donde se guardaran todas las características de la llamada, como el interno al cual se llamó, el número telefónico desde donde se efectuó la llamada, entre otros. Luego define el método initialize, éste es el método constructor del objeto, es decir, el método que se llama una única vez cuando se construye el objeto. Este es el

momento de realizar inicializaciones como conectarse a la BD, el logger, establece la hora, minutos y segundos del comienzo de la llamada. Crea dos objetos fundamentales uno de ellos es *voipgi*, por medio de él se realizará toda la relación de la sesión con el protocolo AGI y el otro es *n2s*, éste objeto realiza toda la conversión de número a letras (number-to-speech). Luego invoca al método *main*, donde comienza el código del objeto.

```
def main
  answer
  inicializaVariables
  playSound("bienvenida")
  return if !verificarCliente
  return if !configurarCliente
  return if !verificarListaBlanca
  return if !verificarListaNegra
  return if !solicitarCodigoRegion
  return if !solicitarTipoEfector
  return if !solicitarEfector
  return if !obtenerEspecialidadEfector
  return if !solicitarClave
  return if !verificarMensajes
  return if !solicitarPrestadorClinica
```

Tabla 11.7: Descripción proceso main

El método main lo primero que hace es llamar al método *answer*, éste método se encarga de inicializar al arreglo *@xgi\_init* luego llama al método *inicializaVariables* que se encarga de inicializar todas las variables del objeto. Luego llama al objeto *playSound* y le pasa como argumento *bienvenida*, éste es el nombre del archivo de sonido que tiene que reproducir a la persona que llama. A partir de acá el diseño del objeto es similar, retorne si el método llamado dio erróneo, así tenemos el primer llamado *return if !verificarCliente* en este caso el método *verificarCliente* tiene toda la lógica de negocio para procesar el ingreso y validación de un número de cliente.

## Método verificarCliente

```
def verificarCliente
  encontro=nil
  @puntoProceso=1
  case @cliente
```

```

when 0:
  (1..3).each do |i|
    playSound("ingrese")
    @cliente=playSoundGetDigits("codigo_empresa",6,@esperaIngreso)
    puts "numero cliente #{@cliente}"
    # si cliente es -1 cortaron la llamada
    if @cliente.to_i==-1
      @codigoError=1
      registrarTransaccion
      return nil
    end
    if @cliente.length==0
      playSound("timeout")
      @cliente=0
    next
    end
    sql="select codigoCliente from EmpresaConvenios" +
      " where codigoCliente=\"#{@cliente}\""
    sth=@bd.execute(sql)
    if sth.rows==0
      playSound("codigo_empresa")
      playSound("no_valido")
      @log.fatal("Codigo Empresa inexistente #{@cliente}")
      playSound("timeout")
    next
    end
    encontro=1
    break
  end
  if !encontro
    playSound("operacion_vencida")
    @codigoError=1
    registrarTransaccion
    return nil
  end
  return true
else
  sql="select codigoCliente from EmpresaConvenios" +
    " where codigoCliente=\"#{@cliente}\""
  sth=@bd.execute(sql)

```

```

    if sth.rows==0
      playSound("cliente_inexistente")
      @log.fatal("Cliente inexistente #{cliente}")
      @codigoError=2
      registrarTransaccion
      return nil
    end
    return true
  end
end

```

Tabla 11.7: Descripción método verificarCliente

Como vemos el resultado de este método puede ser verdadero (true) o falso (nil), en el caso de ser verdadero continúa con la ejecución de la sentencia siguiente, que tiene la misma forma: retorne si el método configurarCliente luego de ser ejecutado ha dado como resultado un valor erróneo.

Esta filosofía de diseño es muy apropiada para el diseño de un AT, el objeto principal es muy fácil de leer y la complicación la tienen los métodos a los que va llamando.

### 11.4.3. Finalización

La finalización del AT se puede llevar a cabo por dos situaciones:

1. Se terminó de ejecutar el método main
2. Alguno de los métodos llamados retornó con un estado erróneo.

En ambos casos continúa ejecutando el método initialize, la instrucción siguiente es llamar al método playSound pasandole como argumento despedida. Éste es un archivo que tiene el audio de la despedida del sistema. Como no existen más sentencias que ejecutar el objeto se destruye y luego finaliza el thread que lo estaba ejecutando.

Algo importante de destacar es que a la extensión 376, puede ser llamada varias veces mientras se ejecuta un hilo (thread). Como cada uno es independiente, nunca va a dar ocupado. Así se pueden estar cursando decenas de hilos AT al mismo momento y se podrá seguir llamando a la

extensión 376. Esto es una clara diferencia con la telefonía analógica tradicional, donde el canal es físico, no lógico como lo es en telefonía IP.

## **12. Conclusiones**

A través del desarrollo de este trabajo queda demostrado que la ayuda que ofrecen los Automatas Telefónicos es indiscutible. No obstante, para que la funcionalidad que cumplen sea eficiente, las aplicaciones que implementan tienen que estar bien acotadas y enfocadas en un tema determinado, con relativamente pocas opciones a cubrir. En el presente estudio se demostró que es posible crear diversas aplicaciones que cubran un amplio espectro.

La incorporación de telefonía IP permite reducir los costos establecidos por la telefonía convencional y además puede extender el radio geográfico de cobertura a nivel nacional e incluso internacional.

Una de las condiciones que se tiene que tener en cuenta, es que están involucradas muchas tecnologías y, por lo tanto, esto sólo puede ser implementado por un grupo de trabajo que se enfoque en su temática. Son grupos interdisciplinarios donde el integrador juega un papel fundamental a la hora de manejar el equipo de trabajo.

Los AT pueden estar apoyados por tecnologías complementarias tales como aplicaciones Web o el soporte de mensajes de texto (SMS) por medio del teléfono móvil. Todo eso constituye el combo de opciones que puede dar una empresa a sus clientes.

Creemos que la tecnología está continuamente avanzando, por lo cual debemos estar en un constante proceso de adaptación. Estamos inmersos en el cambio, por lo tanto una de nuestras fortalezas o debilidades será nuestra adaptación a ella. Esta tesis se ha basado en los Automatas Telefónicos y la telefonía IP para acceder a ellos.

### **12.1. Investigaciones y desarrollos futuros**

Una tecnología complementaria a esto sería la utilización del servicio de mensajes cortos (SMS).

La telefonía móvil nos provee de una tecnología que la podríamos sumar a nuestro AT, incorporando, de esta manera, una mayor plasticidad. Podemos transformar a nuestro AT desde un tipo de funcionamiento en modo sincrónico a un modo asincrónico. La llamada telefónica es fundamentalmente sincrónica o en tiempo real. En cambio el

envío, recepción, procesamiento y posterior respuesta de un SMS sería un modo totalmente asincrónico.

La gran ventaja sería el incremento de cobertura. Esto está basado en que la red inalámbrica de los celulares, cada vez es mayor. Por otro lado, para muchas personas es mucho más sencillo enviar un SMS (o varios) con los datos del prestador, afiliado y prestación a realizar. Y en algún momento (asincrónico) obtendrá la respuesta, sin tener que perder tiempo en esperarla. Por ejemplo, al otro día puede verificar todas las respuestas del sistema. Esto reduce aún más el tiempo de obtener el código de autorización de la prestación.

La comunicación entre un celular y la PC se puede implementar por medio de un cable. El problema es que cada marca y modelo tiene su propio conector. Para evitar esto lo ideal es hacerlo por medio de tecnología inalámbrica, ya sea por Infra Rojo o Bluetooth. Con cualquiera de estas dos tecnologías nos independizamos de la marca y modelo del móvil. Obviamente el dispositivo que utilizemos tienen que tener alguna de estas dos tecnologías inalámbricas disponibles.

Como hemos visto antes, para poder interactuar con el Autómata Telefónico podemos hacerlo por medio de la telefonía fija o la telefonía móvil. Pero en éste último caso el costo incurrido al llamar desde un teléfono móvil a un teléfono fijo es mucho más caro (aprox. 3 veces), que si lo hiciera de móvil a móvil.

Para solucionar este problema se puede utilizar un GSM Gateway. Éste dispositivo permite transformar una llamada bajo tecnología GSM (Sistema Global para las Comunicaciones Móviles) a telefonía fija o a telefonía IP (existen las dos salidas). Recordemos que en este momento en la Argentina el 87% de los teléfonos móviles soportan esta tecnología y las redes de todas las prestatarias de telefonía móvil. De esta manera la llamada entraría por un interno de la central telefónica IP, por un canal SIP o por un canal Zap.

Como pudimos observar, a lo largo del trabajo, tomando como caso de estudio el servicio de las Obras Sociales, hemos logrado cumplir con el objetivo de analizar las tecnologías necesarias para poder diseñar, desarrollar y finalmente implementar un Autómata Telefónico. Y finalmente podemos concluir es factible y recomendable comenzar a implementar este tipo de tecnología en las Obras Sociales, ya que facilitaría y ahorraría mucho trabajo a la vez que mejoraría la calidad del servicio que se le brinda al usuario.

## 13. Bibliografía

1	ROSENBERG, J., SCHULZRINNE, H. y CAMARILLO, SIP: Session Initiation Protocol, RFC 3261, junio 2002.
2	JOHNSTON, A.B., SIP, understanding the Session Initiation Protocol, second edition, 2004, Artech House Publishers, Boston.
3	Manuel Moreno Martín, Manuel Alvarez- Campana Fernández-Corredor, Joan Vinyes Sanz, Una primera aproximación al protocolo SIP, Revista AHCIET,
4	<a href="http://www.cs.columbia.edu/sip">http://www.cs.columbia.edu/sip</a>
5	David Gomillion, Barrie Dempster, "Building Telephony Systems with Asterisk", Packt Publishing Ltd., ISBN 1-904811-15-9, February 2006
6	Neil Cherry, "Linux Smart Homes for Dummies", Wiley Publishing Inc., ISBN-10: 0-7645-9823-6, p. 37-89
7	Digium Install Guide, Digium Press
8	Oded Agam,"Voice over IP Testing - A Practical Guide", RADCOM White Paper, Bitpipe Inc. 2001
9	Mark Spencer, "Introduction to the Asterisk Open Source PBX", Linux Support Services, Inc.,
10	Avaya™ IVR Designer, Release 5.0, Avaya Press
11	Call Centre Operation Guideline, Office of Information and Communications Technology
12	Chris James, "Financial Services: Transforming Contact Centers into High-Performing Sales Channels", Best Practices, LLC
13	Jonathan Davidson, "Voice over IP Fundamentals", Cisco Press
14	Gustavo Artigas, "Cómo planificar Profesionalmente, Instalar e Implementar en la empresa un sistema VoIP", Siemens
15	Réal Bergevin, Allen Wyatt, "Contact Center for Dummies", Wiley Publishing Inc., ISBN-10: 0-471-75819-1

16	Jim Van Meggelen, Jared Smith, Leif Madsen; Asterisk The Future of Telephony, O'Reilly, ISBN: 0-596-00962-3
17	Steven Shepard, Convergencia de las telecomunicaciones, McGraw-Hill, ISBN 0-07-138785-4
18	Timothy Kelly, VoIP For Dummies, Wiley Publishing, Inc., ISBN-13: 978-0-7645-8843-3
19	Maik Schmidt, Enterprise Integration with Ruby, The Pragmatic Programmers LLC., ISBN 0-9766940-6-9
20	Leif Madsen, Jared Smith, Jim Van Meggelen, Chris Tooley, An Introduction to Asterisk, The Asterisk Documentation Project
21	David Endler, Mark Collier, Hacking Exposed VoIP: Voice over IP Security Secrets & Solutions, McGraw Hill, ISBN-13: 978-0-07-226364-0
22	Leif Madsen, Jared Smith, Steven Sokol, Wasim Baig, The Hitchhiker's Guide to Asterisk
23	Brett Schwarz, Asterisk Open-Source PBX System, February 2004, Linux Journal
24	Cornelis Hoogendoorn, Next Generation Networks and VoIP, Siemens Press
25	Leif Madsen, Jared Smith, Jim Van Meggelen, Asterisk : The Future of Telephony, O'Reilly, ISBN: 0-596-00962-3
26	Theodore Wallingford, VoIP Hacks, O'Reilly, ISBN: 0-596-10133-3
27	Thomas Porter, Practical VoIP Security, Syngress Publishing, Inc., ISBN: 1597490601
28	Matthew G. Naugle, Illustrated TCP/IP, Wiley Computer Publishing, ISBN: 0471196568
29	Flavio Eduardo de Andrade Gonçalves, Asterisk PBX Guia de Configuração
30	Cisco Systems, Voice over IP Fundamentals, Cisco Press
31	Chris Roberts, Voice Over IP Security, Centre for Critical Infrastructure Protection, Wellington, New Zealand
32	Paul Mahler, VoIP Telephony with Asterisk, ISBN 09759992-0-6