



# **ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL**

## **Facultad de Ingeniería Eléctrica y Computación**

“Diseño, implementación y mantenimiento de un sistema básico de Call Center con modo de operación dual sobre la plataforma de telefonía IP Elastix”

### **EXAMEN DE GRADO (COMPLEXIVO)**

Previo a la obtención del Título de:

## **INGENIERO EN COMPUTACIÓN ESPECIALIZACIÓN**

### **“SISTEMAS DE INFORMACIÓN”**

Presentado por:

**Alex Leopoldo Villacís Lasso**

**GUAYAQUIL – ECUADOR**

## TRIBUNAL DE SUSTENTACIÓN



---

MSIG. Soldimar Matamoros E.  
EVALUADORA



---

MSc. Patricia Chavez Burbano  
EVALUADORA

## DECLARACIÓN EXPRESA

“La responsabilidad del contenido de este proyecto de graduación nos corresponden exclusivamente, y el patrimonio intelectual de la misma a la ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL”

(Reglamento de Graduación de la ESPOL)

A handwritten signature in blue ink, reading "Alex Leopoldo Villacís Lasso", written over a horizontal line.

Alex Leopoldo Villacís Lasso

## RESUMEN

Este informe técnico detalla la implementación de un sistema básico de Call Center, el Elastix CallCenter, actualmente en su versión 2.2.0-8, el cual permite realizar automáticamente llamadas desde un sistema Asterisk, y conectarlas con uno de un grupo de agentes, para propósitos de encuestas o telemarketing.

En el primer capítulo se exponen los antecedentes del proyecto. Se empieza con la descripción del problema que enfrenta el primer cliente para el cual se desarrolló la solución. Se exponen las justificaciones para crear un sistema nuevo y se muestran los objetivos generales y específicos, así como una estimación del alcance y las limitaciones que puede tener el proyecto.

En el segundo capítulo se realiza un análisis del estado inicial del cliente. Como parte de este análisis, se muestran los requerimientos del cliente inicial, y los cambios en estos requerimientos al ser extendido el proyecto para publicación general. Luego se procede a listar las plataformas y



tecnologías disponibles al inicio de la implementación. Aquí se procede a expandir el análisis de la situación del cliente y se derivan las funcionalidades que debe tener el sistema para poder satisfacer los requerimientos del cliente.

En el tercer capítulo se hacen descripciones cortas de los conceptos claves que están involucrados en el desarrollo del proyecto. Como parte de esto se procede a listar las plataformas de software que proveen implementaciones de los conceptos y funcionalidades requeridas, y que son la infraestructura sobre la cual se construye esta solución.

En el cuarto capítulo, se describe el diseño del proyecto, desde el punto de vista funcional y técnico. Se describe los roles de los diversos usuarios del sistema, la manera en que estos usuarios interactúan con el sistema, y la manera como se ingresan y exportan datos entre el sistema y otros sistemas que el cliente puede poseer, así como una descripción técnica de los componentes del sistema.

En el quinto capítulo, se indican las experiencias y resultados obtenidos durante la implementación del sistema. Se detallan en particular las limitaciones expuestas por las decisiones iniciales de implementación, y los cambios introducidos para superar tales limitaciones. Se concluye con una

descripción del estado actual del proyecto y las limitaciones no resueltas.

En el sexto capítulo, se dedica atención especial a los problemas de diseño pendientes que tiene el sistema, tal como está implementado actualmente. Se detallan las maneras en que estas limitaciones de diseño obstaculizan la introducción de características deseables, y se indican sugerencias para una posible reimplementación del sistema, de una forma que solucione estas limitaciones.

## ÍNDICE GENERAL

RESUMEN.....	IV
ÍNDICE GENERAL.....	VII
ABREVIATURAS .....	XI
GLOSARIO .....	XV
ÍNDICE DE FIGURAS.....	XVII
INTRODUCCIÓN.....	XIX
1 Antecedentes.....	1
1.1 Descripción del problema .....	1
1.2 Justificación .....	4
1.3 Formulación del problema .....	8
1.4 Objetivos.....	9
1.4.1 Objetivo general .....	9
1.4.2 Objetivos específicos.....	10
1.5 Alcance del proyecto .....	10
2 Análisis de estado inicial y requerimientos .....	14
2.1 Plataformas y tecnologías utilizadas inicialmente.....	14
2.2 Solución del problema .....	16

2.3	Identificación de requerimientos .....	17
3	Marco teórico .....	26
3.1	Telefonía VoIP .....	26
3.2	Descripción de una IP-PBX .....	30
3.3	Protocolos SIP e IAX .....	33
3.4	Controladores DAHDI .....	36
3.5	Asterisk – Plataforma para aplicaciones de comunicaciones .....	37
3.6	FreePBX – Interfaz web para configuración de Asterisk.....	47
3.7	Elastix – Sistema de comunicaciones unificadas .....	49
4	Descripción del proyecto.....	53
4.1	Descripción de las funciones del sistema .....	53
4.1.1	Roles de los usuarios del sistema .....	54
4.1.2	Interacción con los usuarios .....	63
4.1.3	Integración con otros sistemas y transferencia de datos.....	64
4.1.4	Requerimientos de rendimiento, confiabilidad y seguridad .....	69
4.2	Diagramas de roles e interacciones .....	71
4.3	Descripción Técnica de la Solución .....	72
4.3.1	Componentes del sistema .....	73
4.3.2	Administración del sistema.....	75

5	Implementación del proyecto .....	78
5.1	Descripción de la implementación .....	78
5.1.1	El Servidor .....	79
5.1.2	El Cliente .....	82
5.1.3	La Base de Datos .....	107
5.1.4	El Demonio .....	119
5.2	Arquitectura inicial de la Consola de Agente y el Marcador .....	138
5.3	Introducción del manejo de llamadas entrantes .....	142
5.4	Limitaciones de la arquitectura inicial de la Consola de Agente ...	144
5.5	Introducción del protocolo ECCP (Elastix CallCenter Protocol)....	147
5.6	Limitaciones observadas del Marcador monoproceso .....	155
5.7	Introducción del Marcador multiproceso .....	158
5.8	Implementación de agentes dinámicos.....	161
5.9	Situación actual y limitaciones pendientes .....	164
6	ANÁLISIS DE RESULTADOS .....	166
6.1	Evaluación de cumplimiento de objetivos .....	166
6.2	Características deseables a implementar .....	168
6.3	Obstáculos planteados por diseño actual .....	172
6.4	Puntos de rediseño en base de datos .....	174

6.5	Lenguajes de programación y bibliotecas de software para rediseño	
	175	
	CONCLUSIONES Y RECOMENDACIONES .....	179
	ANEXOS .....	185
	BIBLIOGRAFÍA .....	268

## ABREVIATURAS

ADPCM	Adaptive Differential PCM	PCM adaptivo diferencial
AJAX	Asynchronous JavaScript And XML	JavaScript y XML asíncrono
AMI	Asterisk Manager Interface	Interfaz de administración de Asterisk
ASCII	American Standard Code for Information Interchange	Código estándar americano para intercambio de información
CDR	Call Detail Record	Registro de detalle de llamada
CRM	Customer Relationship Management	Administración de relaciones con clientes
CRUD	Create, Read, Update and Delete	Crear, leer, actualizar y borrar
CSV	Comma-Separated Values	Valores separados por comas
DAHDI	Digium/Asterisk Hardware Device Interface	Interfaz de dispositivo hardware de Digium/Asterisk
DHCP	Dynamic Host Configuration Protocol	Protocolo dinámico de configuración de host
DID	Direct Inward Dialing	Marcado entrante directo
DNC	Do Not Call	No llamar
DTMF	Dual-Tone Multi-Frequency	Señalización de tono dual

	signalling	multifrecuencia
ECCP	Elastix Call Center Protocol	Protocolo de call center de Elastix
GUI	Graphical User Interface	Interfaz gráfica de usuario
HTML	HyperText Markup Language	Lenguaje de marcas de hipertexto
HTTP	Hypertext Transfer Protocol	Protocolo de transferencia de hipertexto
IAX	Inter-Asterisk eXchange	Intercambio entre Asterisk
iLBC	Internet Low Bitrate Codec	Codec de baja tasa de bits de Internet
IMAP	Internet Message Access Protocol	Protocolo de acceso a mensajes de Internet
ISDN	Integrated Services Digital Network	Red digital de servicios integrados
IVR	Interactive Voice Response	Respuesta de voz interactiva
LAMP	Linux, Apache, MySQL, PHP	
LAN	Local Area Network	Red de área local
MGCP	Media Gateway Control Protocol	Protocolo de control de pasarela de media
PBX	Private Branch eXchange	Ramal privado de conmutación
PCM	Pulse Code Modulation	Modulación por impulsos



		codificados
PDF	Portable Document Format	Formato de documento portable
POP3	Post Office Protocol version 3	Protocolo de oficina postal versión 3
PS	PostScript	
PSTN	Public Switched Telephone Network	Red telefónica conmutada pública
RFC	Request For Comments	Petición para comentarios
RPC	Remote Procedure Call	Llamada de procedimiento remoto
RTP	Real-time Transport Protocol	Protocolo de transporte en tiempo real
SDP	Session Description Protocol	Protocolo de descripción de sesión
SIP	Session Initiation Protocol	Protocolo de inicio de sesión
SMS	Short Message Service	Servicio de mensaje corto
SMTP	Simple Mail Transfer Protocol	Protocolo simple de transferencia de correo
SOAP	Simple Object Access Protocol	Protocolo simple de acceso a objetos
TCP	Transmission Control Protocol	Protocolo de control de

		transmisión
TIFF	Tagged Image File Format	Formato de archivo de imagen etiquetado
TLS	Transport Layer Security	Seguridad de capa de transporte
TTS	Text-To-Speech	Texto a voz
UDP	User Datagram Protocol	Protocolo de datagrama de usuario
URI	Uniform Resource Identifier	Identificador uniforme de recurso
URL	Uniform Resource Locator	Localizador uniforme de recurso
VoIP	Voice over IP	Voz a través de IP
XML	eXtensible Markup Language	Lenguaje de marcas extensible
XMPP	eXtensible Messaging and Presence Protocol	Protocolo de mensajería y presencia extensible

## GLOSARIO

Agente	Persona contratada por el call-center para atender llamadas telefónicas, bien marcadas, o asignadas a la persona luego de ingresar al PBX.
Campaña	Sesión de funcionamiento en un call-center, de uno a varios días de duración, orientado al marcado de una lista particular de números con un tema en particular. El tema de la campaña determina la posible selección de números a marcar, el guión a usar y las preguntas de los formularios a llenar con cada llamada. Con la extensión de funcionalidad a la recepción de llamadas del exterior, esta definición cubre lo que luego se conoce como campaña saliente.
E/S	Entrada/Salida, una operación de acceso a almacenamiento secundario.
Guión	También llamado script, un esquema detallado que indica al agente gran parte o la totalidad del discurso que debe decirle a la persona contactada. Este esquema incluye, si es necesario, ramas de decisiones que el agente debe seguir en caso de que la persona contactada exprese interés, o no, en partes del discurso.

**Payload**      Contenido de un mensaje en el cual se transmite un fragmento del flujo de datos de audio o video, representado con el codec negociado con anterioridad.

## ÍNDICE DE FIGURAS

Figura 3.1: Relación entre los componentes de una pila de protocolos VoIP	30
Figura 3.2: Interacción de Asterisk con elementos de telefonía externos	38
Figura 4.1: Sistema Call Center y roles de los usuarios	65
Figura 4.2: Flujo de información cargada y recolectada	67
Figura 4.3: Protocolos usados para interacción de agente	71
Figura 5.1: Consola de Agente en pantalla de login	85
Figura 5.2: Consola de Agente en atención de llamada	85
Figura 5.3: Listado de Campañas Salientes	86
Figura 5.4: Lista de números en DNC	87
Figura 5.5: Lista de URLs Externos	87
Figura 5.6: Lista de Colas Reservadas	89
Figura 5.7: Interfaz de carga de contactos	89
Figura 5.8: Lista de Campañas Entrantes	90
Figura 5.9: Lista de Agentes Estáticos	91
Figura 5.10: Lista de Usuarios ECCP	92
Figura 5.11: Lista de Agentes Dinámicos	92
Figura 5.12: Lista de Tipos de Recesos	94
Figura 5.13: Lista de Formularios	94
Figura 5.14: Editor de Formularios	95

Figura 5.15: Vista Previa de Formulario.....	95
Figura 5.16: Reporte de Recesos .....	96
Figura 5.17: Detalle de Llamadas .....	98
Figura 5.18: Reporte de Llamadas por Hora.....	98
Figura 5.19: Reporte de Llamadas por Agente .....	99
Figura 5.20: Reporte de Tiempo de Espera .....	100
Figura 5.21: Reporte de Tiempo de Sesión .....	101
Figura 5.22: Reporte de Éxito de Llamadas Entrantes .....	101
Figura 5.23: Gráfico de Llamadas por hora .....	102
Figura 5.24: Reporte de Información de Agente .....	103
Figura 5.25: Monitoreo de Agentes .....	104
Figura 5.26: Troncales por hora .....	104
Figura 5.27: Monitoreo de Llamadas Entrantes .....	106
Figura 5.28: Monitoreo de Campañas.....	106
Figura 5.29: Módulo de Configuración .....	107
Figura 5.30: Arquitectura de primera implementación de programa marcador con actualización a tabla <code>current_calls</code> . .....	141
Figura 5.31: Arquitectura de programa marcador monoproceso con protocolo ECCP .....	154
Figura 5.32: Arquitectura de programa marcador multiproceso con ECCP	160

## INTRODUCCIÓN

Para todos los negocios y empresas, una parte importante de las actividades requeridas para su funcionamiento es el mantener el contacto con sus clientes. Este contacto incluye, pero no se limita, a la búsqueda de clientes nuevos con los cuales comerciar, el anuncio de nuevos productos y servicios a los clientes existentes, encuestas de satisfacción con el servicio recibido, atención y servicio técnico. Para entablar el contacto con los clientes, una de las herramientas disponibles es la llamada telefónica entre el cliente y un representante de la empresa.

Muchas empresas están implementando sistemas de telefonía interna basados en protocolos IP. Uno de tales sistemas telefónicos es Elastix. Para satisfacer la necesidad de contacto con el cliente, surge la posibilidad de usar el sistema de telefonía IP para administrar las llamadas entre la empresa y el cliente. Un uso óptimo de la central telefónica IP requiere, además de la llamada en sí, la automatización del marcado, la administración de los números telefónicos y la información recogida y la generación de reportes y monitoreos de la actividad de llamado. El sistema Call Center desarrollado en la presente tesis es una implementación de tales requerimientos sobre Elastix.

# **CAPÍTULO 1**

## **1 Antecedentes**

En este capítulo se muestran los problemas a los cuales se enfrentó la empresa cliente al momento de iniciar un call-center. Se menciona el procedimiento intentado inicialmente y los problemas encontrados, los cuales justifican la creación de un sistema automatizado. Finalmente se especifican los objetivos general y específicos y el alcance del proyecto.

### **1.1 Descripción del problema**

En el año 2007, la empresa CARTIMEX S.A., dedicada a la distribución y venta de equipos de computación y otros artículos electrónicos, decidió implementar un call-center dedicado al telemercadeo. Este call-center estaría orientado a la generación de llamadas a clientes potenciales, con el fin de informarles sobre promociones y nuevos productos disponibles. Si se despertaba el



interés del cliente, la llamada debía de terminar con el inicio de una venta. Toda la información concerniente a la llamada realizada debía de ser registrada para poder refinar las llamadas futuras y para llegar a dar término a las ventas iniciadas a través de las llamadas. La intención última de este call-center era la de incrementar las ventas a través de ofertas a los clientes contactados, que de otro modo no se hubiesen enterado de la disponibilidad de los productos de la empresa.

Con este plan de call-center en mente, la empresa contrató alrededor de una decena de personas para que iniciaran las llamadas (con plan de expansión futura), y proveerles con computadores que ejecutaban softphones. Para dotar a los softphones con el acceso a las líneas telefónicas al exterior, la empresa eligió usar una implementación telefónica basada en la distribución GNU/Linux orientada a telefonía ELASTIX. Sin embargo, ELASTIX, a pesar de proveer la capacidad telefónica base, no tenía entonces asistencia alguna explícitamente orientada al funcionamiento de un call-center. Para suplir la carencia, el administrador del call-center intentó hacerlo funcionar con el procedimiento detallado a continuación:

- El administrador del call-center obtiene o recibe una lista de

números telefónicos a marcar, con el nombre de la persona a la cual se espera contactar. La lista de números se divide en partes iguales, tantas como agentes haya disponibles.

- El administrador elabora, o recibe de sus superiores, un guión que cada agente debe de seguir para la interacción con el cliente, y además una lista de preguntas o encuesta a llenar.
- Se elabora un libro de cálculo en Excel que contiene en una hoja el guión a seguir, el formulario a llenar en otra hoja y el listado de los números a marcar en una última hoja. Este listado contiene, además del nombre y el número, columnas para anotar respuestas del formulario.
- Los agentes reciben una versión del libro de cálculo, la cual abren en su computador asignado. Cada versión difiere en la lista de números asignada al agente.
- Para cada número en su lista, el agente debe de digitarlo en su softphone y esperar al resultado de la marcación. Es responsabilidad del agente anotar en su hoja de cálculo si la llamada se conectó, o si ha fallado.
- En caso de que la llamada se haya conectado, el agente anota en la hoja de cálculo la hora, con segundos, en que inicia la exposición, y a continuación recita el guión de

interacción provisto en el libro. En base a las preguntas del formulario (también incluido en la hoja de cálculo), el agente llena el renglón correspondiente al número marcado con las respuestas. Las columnas dedicadas a las preguntas del formulario también indican si se ha iniciado una venta como resultado de la llamada.

- Al cerrar la llamada, el agente anota en la hoja correspondiente la hora, con segundos, en que terminó la llamada. A continuación, el agente marca el siguiente número.

El procedimiento indicado anteriormente es casi completamente manual. En cada paso, el agente debe de intervenir para seguir los detalles del procedimiento. Como resultado, se introduce la posibilidad de errores y se confía en la honestidad de los agentes contratados. Los inconvenientes se exponen a continuación y justifican una implementación automatizada.

## **1.2 Justificación**

La búsqueda de una solución automatizada para la realización de las llamadas a los clientes se justifica, debido a los inconvenientes

que implica trabajar de manera manual. Los mismos se listan a continuación:

- La lista de contactos asignada a cada agente es administrada individualmente por cada uno de ellos. Los agentes deben dedicar atención a llevar la cuenta de cada número marcado en medio de una tabla de números telefónicos similares. Para cada contacto, se debe mantener seleccionado el renglón en la hoja, a la vez que se salta hacia atrás y hacia delante hacia las hojas del guión y de las preguntas del formulario. Adicionalmente, el agente debe localizar la columna de la pregunta actual para anotar la respuesta correspondiente al contacto en curso. Esta situación introduce la posibilidad de cometer errores, de los cuales el más frecuente es el de anotar la información en el renglón o columna equivocados.
- Los administradores del call-center desean tener a la mano estadísticas de tiempos de atención. La razón de reunir tales estadísticas es la de medición y control de la eficiencia de cada agente (porcentaje de tiempo dedicado a la atención) y de la eficiencia de la campaña como un todo. Estas estadísticas deben ser reunidas manualmente, ya que no hay

asistencia alguna por parte del computador. Además, como fuente de tales estadísticas, el instante de inicio y final de atención de la llamada deben ser anotados manualmente por el agente en la hoja de cálculo. Este método, además de estar sujeto a error humano, depende de la buena fe de cada agente para su precisión. Un agente malintencionado puede anotar tiempos falsos de inicio y de final, defraudando a la empresa por la diferencia de tiempo, y esta diferencia sólo se podría descubrir luego de una auditoría contra los registros de llamadas realizadas por la central telefónica.

- De la misma manera, el tiempo que un agente invierte en almorzar, ir al baño y otras interrupciones (programadas o no) debe ser anotado por el administrador o por el propio agente, con el consiguiente riesgo de error y malicia humanas. Otro tanto ocurre con el tiempo total durante el cual el agente estuvo disponible para atender llamadas.
- No existe una manera rápida de extraer un reporte de las respuestas entregadas por los clientes. Para hacerlo, es necesario consolidar manualmente las hojas de cálculo actualizadas de todos los agentes que participaron. Otra vez, en esta fase puede introducirse error humano al sumar las contribuciones de todos los agentes.

- Una vez que se ha terminado de atender una llamada, el agente inicia la siguiente llamada en un momento de su propia elección. No es posible hacer que el agente atienda inmediatamente una llamada luego de concluida la anterior. El tiempo que transcurre entre el fin de la llamada y el inicio de la siguiente se considera, por parte de los administradores, desperdiciado. Los administradores del callcenter de CARTIMEX consideraron esto un problema grave.
- No existe manera de aprovechar el tiempo muerto entre el inicio del marcado de la llamada y el instante en que la llamada es contestada del lado remoto (si es que es contestada). Para aprovechar el tiempo indicado, es necesario que de alguna manera se prediga que el agente estará pronto desocupado, para poder iniciar el marcado de la siguiente llamada mientras el agente todavía está conectado a la llamada actual. Esta situación también fue considerada como un problema, ya que el agente resulta subutilizado.

Con todos estos inconvenientes, los administradores del callcenter decidieron aprobar el desarrollo de un sistema automatizado para la

administración de las llamadas del callcenter, y además consideraron adecuado encargar el desarrollo de tal sistema a la empresa responsable del proyecto ELASTIX, con el que contaban originalmente.

Por su parte, Palosanto Solutions justifica el desarrollo del software de gestión de call-center como un método para adquirir experiencia y para disponer de una solución estándar para futuros proyectos. La adquisición de conocimientos es valiosa para el soporte técnico debido a que con frecuencia un sistema con Asterisk y Elastix se usa como un call-center. La disponibilidad de una solución estándar de call-center es importante, debido a que ofrece una razón adicional para instalar ELASTIX en sus sistemas, y por lo tanto para contratar servicio técnico o hardware de telefonía.

### **1.3 Formulación del problema**

A partir de los inconvenientes que fueron expuestos anteriormente, se puede concluir que el problema al cual se enfrenta la empresa CARTIMEX, es el uso de un procedimiento de trabajo completamente manual que conduce a una dependencia desmesurada en la labor humana. Esta forma de trabajo es resultado de la inexistencia inicial de un sistema de gestión de call-

center compatible con ELASTIX. El uso de la labor manual ocurre en áreas del procedimiento que son repetitivas y operan sobre conjuntos moderadamente grandes de datos, y que por lo tanto son candidatos ideales para la automatización. Las áreas donde la intervención humana es problemática y podría reemplazarse por programas son: distribución de llamadas, marcación de llamadas, administración de los datos de formularios, consolidación de datos recolectados, auditoría de las llamadas, seguimiento de descansos y monitoreo del funcionamiento de la operación.

## **1.4 Objetivos**

### **1.4.1 Objetivo general**

El objetivo general que la empresa CARTIMEX desea alcanzar con el desarrollo del sistema de call-center es disminuir la cantidad de errores humanos y las ineficiencias existentes en el procedimiento de llamadas original, por medio de la automatización de la gestión de números y marcado.



### **1.4.2 Objetivos específicos**

Para el caso específico de la empresa CARTIMEX, los objetivos fueron los siguientes:

- Maximizar el número de llamadas realizadas por agente, en un periodo de tiempo dado.
- Minimizar tiempos muertos entre llamadas.
- Automatizar la auditoría de tiempos de inicio y fin de llamadas, y su duración.
- Monitorear y llevar el control de los descansos requeridos por los agentes, con miras a minimizar tiempo perdido.
- Minimizar el número de errores humanos en la marcación, en la auditoría de los agentes y en los reportes y recolección de datos.
- Permitir el monitoreo en tiempo real de los agentes, que permita tomar acciones correctivas inmediatas en caso de desviación.

### **1.5 Alcance del proyecto**

Los alcances del proyecto del sistema de call-center acordados entre CARTIMEX y Palosanto Solutions son:

- Marcación automática de llamadas con distribución a los agentes.
- Carga de números a marcar en el sistema a partir de archivos de texto.
- Visualización, para cada agente, de la información de la llamada que se está atendiendo, junto con el guión de interacción, y el formulario a rellenar. Esta visualización se llama en otras partes del documento como CONSOLA DE AGENTE.
- Auditoría de las llamadas realizadas, lo cual incluye inicio y fin de la llamada, duración, y, resultado de la marcación.
- Extracción de la información recolectada de cada campaña, en un formato tabular fácilmente manipulable. Esta información debe de incluir el detalle de cada llamada realizada y la información recolectada del formulario.
- Auditoría de los instantes en los cuales los agentes empiezan y terminan de atender llamadas, y de los descansos autorizados.
- Reportes estáticos y en tiempo real sobre el funcionamiento del sistema de call-center y de la actividad de los agentes.

Los siguientes puntos quedan fuera del alcance del proyecto:

- Integración con posibles sistemas de administración de relaciones con clientes (CRM).
- Introducción al sistema de números a marcar a partir de fuentes distintas a archivos de texto, como por ejemplo, bases de datos en otro servidor.
- Resolución de problemas relacionados con los softphones u otras terminales telefónicas, en su conexión con el sistema ELASTIX. Los softphones fueron puestos en funcionamiento antes de iniciar el desarrollo del proyecto de callcenter, y su configuración y mantenimiento es responsabilidad del contrato de soporte de ELASTIX (en la parte VoIP) y de la gente encargada del provisionamiento de los computadores de los agentes, que es gente interna a la empresa que gestiona el callcenter. Se consideraron excepciones en los casos en que el cliente demostró que el comportamiento del softphone es distinto como resultado de problemas con el software del callcenter.
- Garantías sobre el hardware de telefonía en la central ELASTIX, las cuales están cubiertas por los contratos de soporte técnicos existentes.
- Problemas de configuración de las troncales VoIP usadas para las llamadas generadas, los cuales deben de ser

resueltos a través de los contratos de soporte técnico existentes.

- Generación de respaldos de la información en la central. Este aspecto es responsabilidad del administrador del sistema.

## **CAPÍTULO 2**

### **2 Análisis de estado inicial y requerimientos**

En este capítulo se enumeran los recursos disponibles para la construcción de una solución al problema del cliente. Luego se construye una descripción de las características de esta solución. Por último se identifican los requerimientos que deben ser satisfechos por la solución propuesta.

#### **2.1 Plataformas y tecnologías utilizadas inicialmente**

El cliente original CARTIMEX tenía a su disposición el siguiente hardware y software, el cual fue utilizado para la primera versión del CallCenter:

- Una red LAN ordinaria Ethernet, a la cual están conectados todos los demás equipos.

- Un servidor Elastix, el cual consiste en un equipo servidor compatible Intel x86, con una instalación de la distribución de telefonía Elastix. Los componentes individuales que conforman una instalación de Elastix se describen con más detalle en la sección 3.7.
- Una tarjeta telefónica de líneas analógicas, la cual se usaba para generar y recibir llamadas, conectadas a las líneas telefónicas PSTN provistas por la red telefónica.
- Un conjunto de estaciones PC clientes (se empezó con 10). Cada una de las PC clientes estaba dedicada a un agente y tenían instalado una versión del sistema operativo Microsoft Windows, en la cual estaba a su vez instalado el navegador Mozilla Firefox, para estandarización inicial.
- Cada una de las estaciones PC tiene instalado además un softphone, un software que implementa un teléfono VoIP dentro de la PC. Este teléfono VoIP está configurado para realizar y recibir llamadas a través del servidor Elastix.

Debido a que el servidor Elastix tiene una interfaz administrativa y de aplicaciones basada en web, el desarrollo estuvo orientado desde un principio a construir toda la interfaz del CallCenter como un conjunto de módulos de aplicación web, incluyendo la interfaz de agente. En particular, la naturaleza web de la interfaz de agente,

unido al requerimiento de actualización en tiempo real, tiene implicaciones expuestas en el capítulo 5 en la discusión de la arquitectura de la Consola de Agente.

## **2.2 Solución del problema**

En base a la formulación del problema indicada en el capítulo anterior, se llegó a la conclusión de que se requiere un sistema orientado a: generación automática de llamadas por medio del programa servidor de telefonía (Asterisk, sección 3.5) y usando el hardware telefónico y/o conexiones troncalizadas VoIP disponibles. El sistema debe de funcionar con intervención humana mínima a nula para el acto de generación de llamadas, y con auditoría automática del inicio y final de cada llamada generada. Asociado a cada registro de llamada debe existir la capacidad de adjuntar datos recolectados por entrevista con el cliente contactado, en base a un formulario que también debe de definirse dentro del sistema. La solución propuesta para automatizar el procedimiento manual es la implementación del sistema de gestión de llamadas conocido ahora como Elastix CallCenter.

Para poder tratar con los volúmenes de números a marcar y datos recogidos, deben de implementarse importación en lote de los números a marcar y exportación de los datos recolectados (estado

de la llamada generada para el número, y datos del formulario). El sistema debe poder soportar la ausencia anunciada temporal de un agente (llamado en el medio DESCANSO o BREAK) y detener el flujo de llamadas para este agente en particular. Por último, el sistema debe de implementar las características accesorias a las capacidades discutidas: administración de agentes, administración de campañas, administración de formularios, definición de tipos de BREAK y generación de reportes.

### **2.3 Identificación de requerimientos**

Los requerimientos para el sistema a desarrollar que fueron identificados al principio son los siguientes:

1. El sistema debe ser capaz de generar llamadas automáticamente. Al ser realizada la llamada, esta debe de ser asignada eventualmente a un agente disponible. Sin embargo, las llamadas pueden fallar en ser asignadas por diversos motivos (porque el número está ocupado, no está asignado, porque el cliente se cansa de esperar a que se le asigne un agente). Por lo tanto, el sistema debe llevar el registro del destino final del intento de marcado, para propósitos de estadísticas. Para poder superar estados temporales de fallos, el sistema debe ser capaz de reintentar



el marcado de la llamada luego de un intervalo de tiempo, hasta un máximo de intentos definido por campaña.

2. La lista de números telefónicos a marcar para las llamadas debe de ser ingresable al sistema en forma masiva, con cientos o miles de números a llamar. Cada lote de números marcados está asociado con una ejecución de llamadas de uno o varios días sobre un tema determinado, de aquí en adelante llamado “campaña”.
3. Para propósitos de auditoría, el sistema debe ser capaz de llevar la pista de los agentes disponibles para atender llamadas y su estado dentro de cada sesión de atención. Así, el sistema debe de poder registrar el momento en que el agente inició y finalizó su sesión de atención. También deben de distinguirse entre estados activo, ocupado y en pausa por ausencia autorizada (de aquí en adelante llamado BREAK), así como el motivo de la ausencia. El seguimiento de estado ocupado de los agentes es también implícitamente necesario para no realizar más llamadas simultáneas de las que son posibles de atender con el conjunto de agentes disponibles.
4. Como parte de la auditoría, el sistema debe implementar un inventario de los agentes conocidos por el sistema. Esto incluye la creación, actualización y eliminación de agentes.

La creación de los agentes requiere además asignación de contraseña para la autenticación de los mismos al momento del inicio de sesión.

5. El sistema debe ser capaz de realizar un esfuerzo para determinar si una llamada en particular atendida por un agente va a finalizar próximamente, en base al historial de duración promedio de una llamada de la campaña. Entonces, si el sistema decide que la llamada se terminará pronto, se debe de iniciar el marcado de una nueva llamada que será colocada en una cola, para que sea asignada al agente apenas la llamada anterior concluya. Esta característica del sistema se denomina PREDICCIÓN. Debido a la posibilidad de que la predicción no sea del todo precisa, se debe de tener un porcentaje de confiabilidad de que la llamada atendida realmente será finalizada.
6. El sistema debe de implementar una Consola de Agente. La Consola de Agente es una interfaz de usuario que muestra en una sola vista la información que el agente requiere para atender la llamada asignada. Esta información incluye el estado en que está el agente (ocioso, atendiendo una llamada, en break), datos de la llamada que le ha sido

- asignada y un recordatorio del guión de atención que debe usar para esta campaña.
7. La Consola de Agente debe de permitir al agente colgar la llamada asignada. La interfaz gráfica debe también permitir que el agente indique su ingreso y salida de las pausas autorizadas (breaks).
  8. El sistema debe incluir la capacidad de guardar datos de encuestas asociadas a la campaña. La definición de la campaña debe de incluir los formularios a usar para la atención. La Consola de Agente debe de mostrar los formularios de la campaña para que sean llenados con la información de la entrevista con el cliente. La información que se ha recolectado durante la campaña debe de poderse exportar desde el sistema en un formato fácilmente manipulable para futuro análisis.
  9. El sistema debe permitir la definición de formularios para asociar a las campañas realizadas. En esto se incluye la definición de los campos asociados al formulario y el tipo de dato básico de cada campo.
  10. El sistema debe de permitir definir la lista de descansos autorizados (BREAK). Esto incluye agregar, actualizar y

eliminar los tipos de descansos que los agentes pueden elegir en la Consola de Agente.

11. Se deben de elaborar reportes sobre el rendimiento de la campaña. Los reportes solicitados son los siguientes:

- Reporte, por intervalo de fechas, de los tiempos que cada agente invirtió en los diversos BREAK con una sumatoria total por agente, por BREAK y un gran total.
- Reporte de las llamadas realizadas por los agentes en un intervalo de fechas, con detalle de los instantes de inicio y final, la duración de la llamada y cuánto tiempo esperó el cliente para ser atendido luego de que se conectó la llamada.
- Reporte de distribución por horas (histograma) del número de llamadas realizadas por el sistema, por intervalo de fechas.
- Reporte de agregado de llamadas atendidas por cada agente, por intervalo de fechas, con especificación del número de llamadas, total de tiempo de llamadas, promedio de la duración y duración de la llamada más larga.
- Reporte, por intervalo de fechas y por campaña, de la distribución por intervalos (histograma) del tiempo que

tardó el cliente desde que fue contactado hasta que fue atendido por un agente, con especificación de promedio de espera y máxima espera en el intervalo.

- Reporte, por intervalo de fechas, del tiempo invertido en la atención de llamadas por cada agente, del tiempo total de la sesión del agente y del porcentaje del tiempo atendiendo llamadas con respecto al total de login (porcentaje de atención).

El sistema fue inicialmente implementado con las especificaciones indicadas anteriormente. Durante las pruebas y posterior puesta en funcionamiento por parte de la empresa CARTIMEX, esta empresa agregó un nuevo requerimiento: la gestión de llamadas que ingresan al sistema. La primera versión del sistema sólo era capaz de generar llamadas por su cuenta, desde este punto denominadas LLAMADAS SALIENTES. Sin embargo, se puso en evidencia que también era necesario atender las llamadas que los clientes de CARTIMEX realizaban desde el exterior, fuera de las decisiones de marcación del sistema. Estas llamadas que llegan al sistema también deben ser atendidas por agentes, sus datos deben ser recolectados y sus estadísticas deben de ser recogidas. A estas llamadas se las denominará de aquí en adelante LLAMADAS

ENTRANTES. Al introducir el manejo de las llamadas entrantes se agrega una nueva fuente de llamadas a administrar en el sistema, una nueva dimensión a los reportes generados y una necesidad adicional de exportación de datos. Este requerimiento fue entonces implementado en la segunda versión del sistema.

Luego de la implementación exitosa del sistema en CARTIMEX, otras empresas que usaban Elastix como su sistema de telefonía expresaron interés en usar el software desarrollado para implementar sus propios callcenter. Esto incluye tanto empresas nacionales como internacionales. En el transcurso de la implementación del software ya desarrollado en otras empresas, además de la corrección de errores intrínseca al uso generalizado, se pusieron en evidencia nuevos requerimientos:

1. Los clientes de otros países deben de implementar sus callcenter de acuerdo a su legislación local. Un requerimiento frecuentemente impuesto es la tabla DNC (Do Not Call), que consiste en una lista de números telefónicos a los que no deben realizarse llamadas automáticas, bajo pena de sanciones. Esta tabla debe poder cargarse en el sistema para ser considerada en la carga de números a marcar y para la generación de llamadas en sí.

2. En el funcionamiento de los callcenter, es frecuente que el cliente contactado indique que no puede atender la llamada actual, pero provee un horario en el cual la llamada será aceptada. A este requerimiento se lo denomina AGENDAMIENTO, y debe de ser soportado por el sistema, en la forma de una llamada generada en un momento posterior. El agendamiento puede ser ciego (cualquier agente disponible al momento de generar la segunda llamada puede atender), o requerir que el mismo agente que atendió la primera llamada atiende también la segunda.
3. Los implementadores del callcenter encontraron que, algunas veces, durante la generación de llamadas salientes, la llamada resulta conectada, pero el otro lado no es una persona que puede atender la llamada, sino una máquina contestadora. O puede ocurrir que haya silencio y la llamada se cuelga por parte del agente al poco tiempo de ser conectada. A esta condición de la llamada se la denomina LLAMADA CORTA, y se estableció que esta llamada debe de considerarse como fallida, a pesar de haber sido conectada con el número destino.
4. Un reporte en tiempo real del estado de todos los agentes del callcenter. Este reporte muestra el estado actualizado del

agente, el número que está atendiendo y el tiempo que está transcurriendo en atención, en estado ocioso, o en break.

5. Un reporte en tiempo real del desarrollo de la campaña. Este reporte muestra, para una campaña en particular, el progreso que se realiza en cada llamada generada o recibida por el sistema, las llamadas en espera de atención, y las llamadas siendo atendidas por los agentes.
6. Algunas empresas expresaron interés en el desarrollo de interfaces de agente alternativas a la consola ordinaria del Elastix CallCenter. Para poder acomodar este requerimiento, se creó el protocolo de callcenter de Elastix (ECCP), el desarrollo el mismo se explica ampliamente en la sección 5.5



## **CAPÍTULO 3**

### **3 Marco teórico**

En este capítulo se discuten los conceptos necesarios para construir el proyecto de Call Center. Se empieza por definir la telefonía sobre IP, una IP-PBX y las tecnologías DAHDI, SIP e IAX. Luego se introduce la herramienta principal, Asterisk, con un detalle de las características que permiten implementar una PBX. A continuación se presenta FreePBX, la interfaz de configuración de Asterisk. Por último se presenta a Elastix, el proyecto de telefonía que reúne todos los componentes anteriores.

#### **3.1 Telefonía VoIP**

En computación, VoIP se refiere al conjunto de técnicas y protocolos que se usan para proveer equivalentes de los servicios telefónicos convencionales a través de una red que usa protocolos de Internet. [1] Usualmente, por servicios telefónicos

convencionales, se entendía la transmisión de voz, pero las tecnologías de VoIP también se emplean para transmitir servicios de fax, mensajería SMS, e incluso videollamadas y otros tipos de multimedia. La red que transmite los protocolos IP puede ser la Internet pública, o una red local LAN dentro de una organización.

Para poder funcionar, los protocolos que conforman una comunicación VoIP deben encargarse de las mismas tareas que deben realizarse en una llamada a través de la red tradicional PSTN. Estas tareas son la señalización y negociación de un canal de comunicación, la digitalización de la señal analógica de audio y/o vídeo y la codificación (*encoding*) de la versión digital para su eventual transmisión.

La señalización (*signalling*) consiste en el intercambio de información entre dos puntos que culmina en el establecimiento de dos flujos de comunicación para transmitir la conversación en sí. Esto involucra negociaciones tales como determinar si el usuario o entidad en el sistema contactado existe y está disponible, la ubicación en la red donde debe enviarse el flujo de medios (en caso de ser distinto de la ubicación del sistema contactado), la negociación del formato de codificación de entre los formatos

soportados por los dos sistemas participantes, la indicación del progreso de la llamada y posibles errores al establecer esta llamada y por último, el mantenimiento y finalización de la sesión de la llamada. [2][3] A continuación, o en paralelo con la señalización, el flujo digitalizado de los datos de audio y/o video debe que ser dividido y encapsulado en paquetes para ser transmitido a través de la red IP.

Un codec es el algoritmo y el modelo matemático que se usa para codificar de forma digital, y generalmente comprimir, la información de audio o video. El codec se usa para la conversión entre la señal analógica y la representación de los datos de media dentro del paquete a transmitir. Cada codec hace un balance entre la fidelidad, el ancho de banda usado para transmitir, la resistencia frente a la degradación por pérdida de paquetes y el esfuerzo invertido por la fuente y el destino para ejecutar el algoritmo del códec. [2] La elección del codec es una parte de la negociación hecha a través de los protocolos de señalización.

Para la señalización se pueden usar varios protocolos, los cuales incluyen los siguientes estándares con implementaciones de código abierto:

- SIP
- IAX
- H.323

Algunos de los codecs de audio que pueden ser negociados al establecer la comunicación entre dos puntos de la comunicación son los siguientes:

- PCM, el muestreo de audio sin compresión alguna
- G.711, estándar ITU que define los codecs mu-law y a-law
- ADPCM
- GSM, el códec definido como parte del estándar GSM del European Telecommunications Standards Institute (ETSI)
- G.722
- G.726
- iLBC
- Speex
- G.729

La figura 3.1 muestra una pila de protocolos VoIP típica en una aplicación.

### 3.2 Descripción de una IP-PBX

Un sistema de PBX es un combinación de hardware y software que cumple las funciones de un switch telefónico, el cual permite

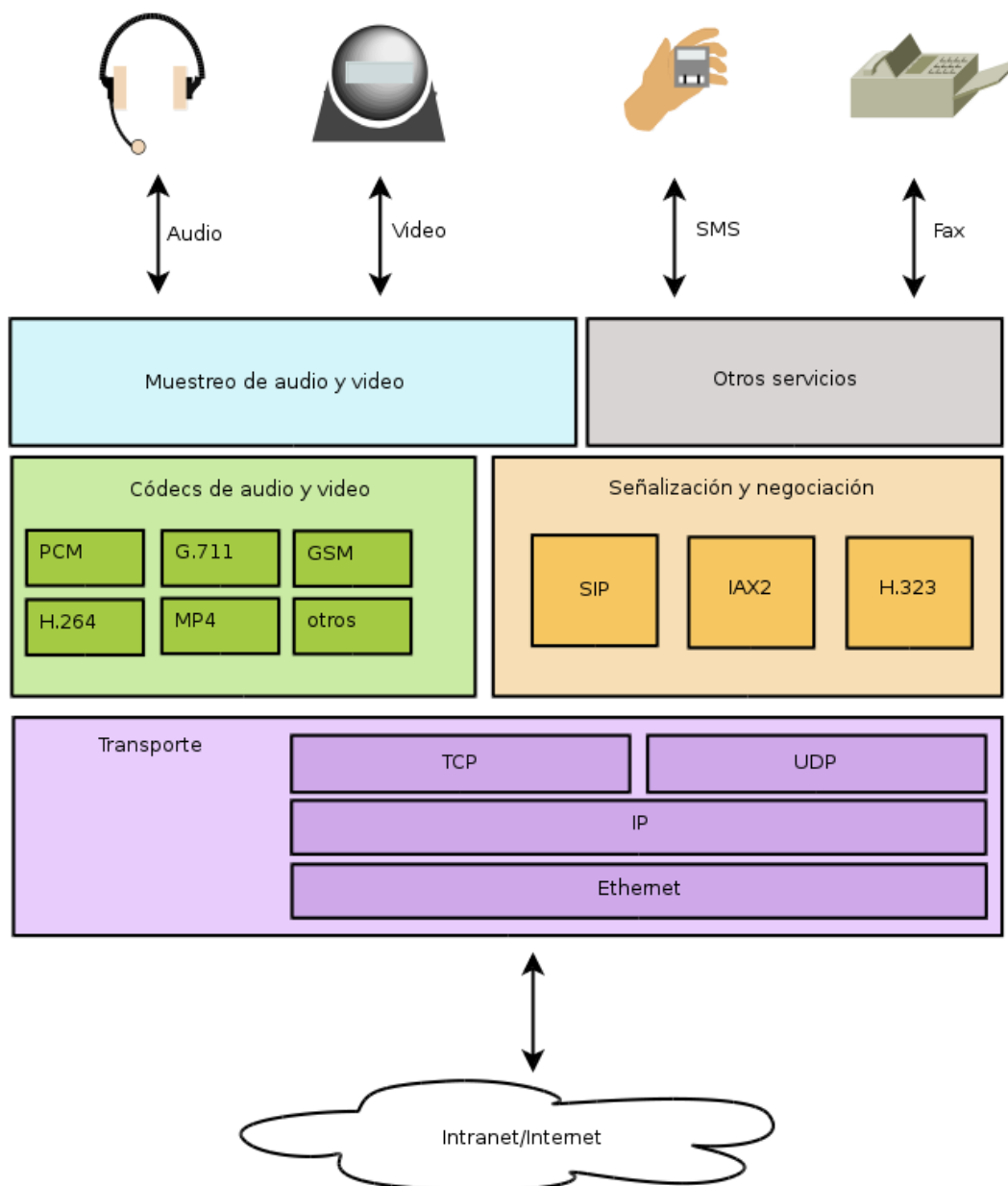


Figura 3.1: Relación entre los componentes de una pila de protocolos VoIP

interconectar llamadas dentro de un grupo de extensiones, con independencia de la red pública (PSTN). El primer beneficio otorgado por el uso de una PBX es el poder realizar llamadas entre las extensiones internas sin incurrir en los costos de llamadas realizadas a través de la red pública. Un segundo beneficio básico se deriva de la centralización de las líneas en la PBX, ya que generalmente la PBX tiene conexiones con la red pública, las cuales pueden ser compartidas entre todas las extensiones configuradas. Con esto se logra que una cantidad pequeña de líneas públicas puedan usarse por un número mucho mayor de extensiones.

Además de los beneficios básicos expuestos arriba, la relativa separación de la red pública permite que la PBX pueda, según las capacidades del sistema, incluir características adicionales. Las funciones que pueden implementarse dentro de una PBX incluyen, aunque no están limitadas, a las siguientes:

- Transferencia de llamadas
- Hunt Group – una llamada que ingresa a través de una línea (por ejemplo, de la PSTN) puede ser asignada a una extensión de un grupo configurado. La selección del algoritmo para elegir la extensión asignada depende de las

capacidades de la PBX (extensión más baja, round-robin, extensión más ociosa, u otros).

- Colas – esta funcionalidad es una extensión del Hunt Group, pero con la capacidad de mantener al usuario en espera en caso de que todas las extensiones configuradas para recibir las llamadas estén ocupadas o no disponibles, para asignar la llamada en modo FIFO a la extensión que se desocupe a continuación.
- Grabación de llamadas – el sistema puede, si tiene el almacenamiento adecuado, grabar conversaciones de las llamadas ruteadas a través de la PBX. Según la arquitectura del sistema, la decisión de cuándo grabar una llamada puede depender de las reglas definidas por el administrador de la PBX. Previa autorización del administrador, uno o ambos participantes de la conversación también pueden iniciar o detener la grabación.
- Buzón de voz – si una extensión no está disponible para contestar una llamada ruteada, el sistema puede ofrecer guardar una grabación para que sea escuchada por el usuario de la extensión en un instante posterior.
- Conferencias – tres o más extensiones se comunican entre sí de forma que todos pueden escuchar la conversación de

todos los demás. Las extensiones pueden unirse o abandonar la conferencia en cualquier momento. Usualmente se ofrecen controles para autorizar la entrada a la conferencia y para invitar a nuevas extensiones a unirse a ella.

- IVR – el usuario puede, a través del teclado del teléfono, interactuar con la PBX y elegir opciones presentadas a través de grabaciones. Generalmente esta funcionalidad se usa para implementar menús de selección que presentan y organizan funcionalidades adicionales al usuario.

Una IP-PBX es un sistema de PBX que hace uso de extensiones telefónicas y líneas salientes de tipo VoIP. Si las capacidades del sistema lo permiten, la funcionalidad VoIP puede combinarse con soporte de extensiones telefónicas analógicas tradicionales y con señalización digital ISDN. En este último caso, la IP-PBX provee funcionalidades de gateway entre las tecnologías VoIP y las conexiones analógicas o ISDN.

### **3.3 Protocolos SIP e IAX**

Una parte fundamental de la operación de un software VoIP, sea éste el softphone, teléfono físico VoIP, o servidor de aplicaciones



VoIP (como por ejemplo Asterisk), es el protocolo de señalización telefónica. Los protocolos de señalización VoIP más populares, de entre los implementados por Asterisk, son SIP e IAX.

### **Protocolo SIP**

El estándar SIP, definido originalmente en su versión 1 en el RFC 2543 y actualizado a su versión 2 en el RFC 3261, es un protocolo de señalización orientado a texto, en el cual los mensajes enviados consisten de peticiones con comandos como cadenas de texto, con URIs que definen el destino a ser contactado, cabeceras adicionales que consisten en texto que definen el nombre de la cabecera y su valor asociado y un cuerpo de mensaje opcional según se requiera por la petición. En este sentido, SIP está modelado de forma parecida al protocolo HTTP, y utiliza muchas de sus convenciones, incluyendo valores de estado de las respuestas y sintaxis de URIs. Generalmente el protocolo SIP se transmite a través de UDP o TCP, en el puerto 5060 para tráfico sin encriptar, o el puerto 5061 para tráfico encriptado con TLS. Sin embargo, también se ha implementado la negociación SIP a través de WebSockets.

El protocolo SIP hace uso de varios otros estándares para poder implementar por completo la negociación de una sesión telefónica. Los dos estándares principales que se usan conjuntamente con SIP son: SDP para la negociación de las capacidades de medios (audio, video y otros) en una sesión en particular, y RTP para la transmisión real del payload. Ya que SIP se encarga únicamente de la señalización, el protocolo SDP sirve para negociar el transporte (UDP/TCP, IP y puerto) a emplear para enviar los paquetes RTP.

### **Protocolo IAX**

IAX es un protocolo de señalización en formato binario, que fue introducido inicialmente como una implementación interna de Asterisk, para ser usada en la interconexión entre servidores Asterisk. Eventualmente, IAX, y su sucesor, IAX2, fueron descritos como un RFC puramente informativo (que no es parte de esfuerzo alguno de estandarización), el RFC 5456. Además de ser soportado por el propio Asterisk, IAX también está implementado en algunos softphones y teléfonos físicos VoIP, así como algunos otros softwares de telefonía VoIP. El puerto por omisión para IAX es el puerto 4569.

Además de ser un formato binario, en oposición a SIP que funciona en modo texto, IAX difiere de SIP en que el protocolo es autocontenido, y el payload se transmite en el mismo flujo de paquetes que contiene la señalización, en el mismo puerto. Esta característica permite simplificar la configuración de los firewalls que deben dejar pasar tráfico IAX, a diferencia de SIP, en que se debe lidiar con el puerto de señalización y un puerto de media negociado. Otra diferencia es que se soporta el envío de la media de múltiples conversaciones en un solo paquete IAX (troncalización). La troncalización permite ahorrar paquetes de datos y reducir la fracción del paquete dedicada a las cabeceras IP.

### **3.4 Controladores DAHDI**

Al implementar una solución telefónica que usa las líneas telefónicas tradicionales de la PSTN (analógicas o digitales) y que debe de funcionar en un computador de propósito general, surge de inmediato la necesidad de que el computador se comuniquen con la línea telefónica directamente. Para esto, se han creado tarjetas de interfaz con la línea telefónica que se enchufan al computador, de las cuales Digium (la compañía que auspicia y respalda Asterisk) es uno de los fabricantes principales.

Las tarjetas de telefonía, así como las tarjetas de empresas que buscan fabricar hardware compatible con Asterisk, varían en su forma de exponer sus capacidades de telefonía al sistema. Con el propósito de que Asterisk (u otro software telefónico) sea capaz de hacer uso del hardware telefónico, se construye la interfaz de software llamada Zaptel, luego renombrada DAHDI. Esta interfaz de controladores, al ser implementada por el controlador de dispositivo de la tarjeta específica, presenta una abstracción de hardware telefónico para uso de Asterisk. Con esta interfaz estandarizada se vuelve posible que un sistema de líneas telefónicas tradicionales pueda integrarse e interactuar con tecnologías VoIP.

### **3.5 Asterisk – Plataforma para aplicaciones de comunicaciones**

El software de comunicaciones telefónicas Asterisk es el corazón de la gran mayoría de instalaciones de telefonía VoIP de código abierto, tanto las instalaciones construidas desde cero, como las administradas a través de interfaces web como FreePBX y las distribuciones dedicadas a la telefonía, como Elastix.

La definición corta de Asterisk de acuerdo a su portal web oficial es: “Asterisk es una infraestructura de código abierto para la construcción de aplicaciones de comunicaciones. Asterisk está por

debajo de sistemas IP PBX, gateways VoIP, servidores de conferencias y otras soluciones personalizadas. Es usado por negocios pequeños, grandes negocios, callcenters, proveedores de telefonía y agencias gubernamentales, en todo el mundo, Asterisk es software libre y de código abierto. Asterisk está auspiciado por Digium.” [9]

La figura 3.2 muestra la relación de Asterisk con varias otras tecnologías mencionadas:

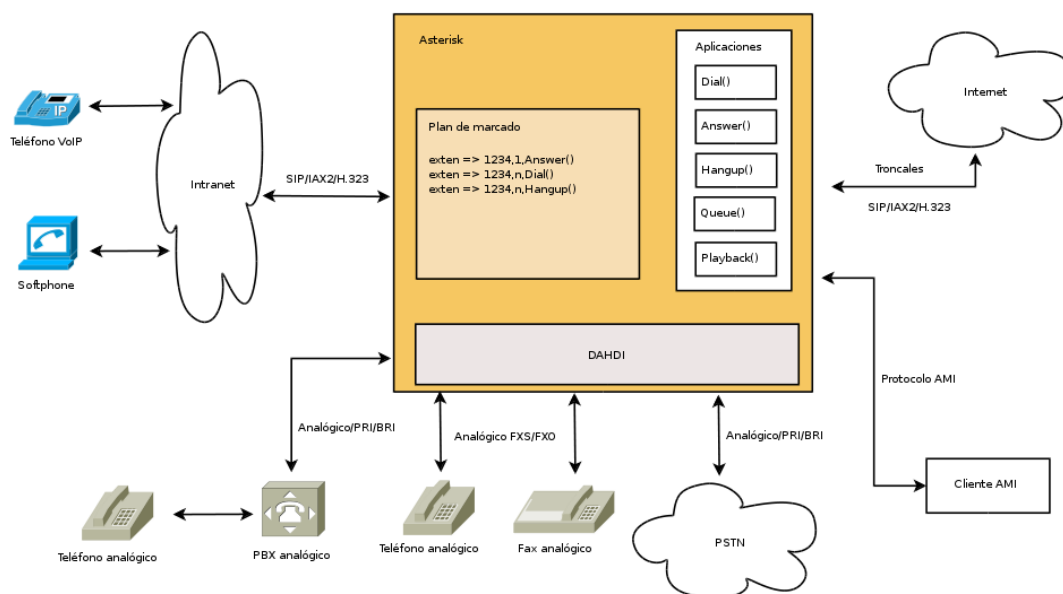


Figura 3.2: Interacción de Asterisk con elementos de telefonía externos

El software Asterisk, por sí mismo, *no es una aplicación telefónica lista para usar*. Lo que provee Asterisk es un conjunto de herramientas y componentes orientados a la telefonía, más un

mecanismo de configuración, similar a un lenguaje de programación. Con estos componentes, el administrador de sistema, o una herramienta de más alto nivel, como FreePBX, puede construir un sistema que provea los servicios de telefonía requeridos. La modularidad de estos componentes es lo que permite la flexibilidad de implementación de Asterisk.

Dos de los conceptos fundamentales necesarios para configurar Asterisk son:

- Canales
- Plan de marcado

### **Canales**

En Asterisk, un canal es la abstracción de una instancia de una tecnología telefónica, que es capaz de introducir llamadas al sistema Asterisk, o de transmitir llamadas desde el sistema. [10] Existen varios tipos de canales disponibles en Asterisk, los cuales corresponden a cada una de las tecnologías telefónicas soportadas en la instancia de Asterisk:

- DAHDI/nnn: Es una conexión con una tarjeta telefónica de tipo DAHDI. El número nnn corresponde al número de canal de entre todos los canales DAHDI de todas las tarjetas

combinadas. El rango de canales disponibles se configura en el archivo `/etc/asterisk/chan_dahdi.conf` .

- SIP/unacuenta/hexid: Es un canal de tecnología VoIP SIP que se ha identificado como la cuenta unacuenta . El valor hexid es necesario porque las troncales VoIP pueden, si están así configuradas, rutear un número arbitrario de llamadas simultáneas usando las mismas credenciales, como se hace comúnmente en las troncales SIP. Las cuentas SIP se configuran en el archivo `/etc/asterisk/sip.conf` .
- IAX/unacuenta/hexid: Es un canal de tecnología VoIP IAX. Al igual que en SIP, se pueden rutear un número arbitrario de llamadas con las mismas credenciales. Las cuentas IAX se configuran en el archivo `/etc/asterisk/iax.conf` .

Los tipos de canales mencionados arriba son los más utilizados. Adicionalmente, hay soporte para otras tecnologías telefónicas que no son tan frecuentemente utilizadas en la actualidad, como por ejemplo:

- OOH323: soporte H.323
- mISDN: soporte para tarjetas telefónicas ISDN independiente de DAHDI
- MGCP: soporte de Media Gateway Control Protocol

- Motif: soporte para colocar y recibir llamadas de Google Voice
- Mobile: soporte para llamadas y SMS usando dispositivos Bluetooth

Además de los tipos de canales que representan directamente tecnologías de telefonía, existen al menos dos tipos de canales especiales, los cuales se mencionan porque se usan como parte de la arquitectura del CallCenter:

- Local: Este tipo de canal es un seudónimo para indicar que el marcado debe realizarse, no a través de una tecnología telefónica directa, sino a través del plan de marcado, el cual se explica a continuación. Por ejemplo, un canal de tipo Local que marca a la extensión 1234 en un contexto llamado from-internal aparece de la forma: Local/1234@from-internal . Su uso dentro del CallCenter deriva del hecho de que al iniciar la llamada a través del plan de marcado, todas las reglas de transformación definidas en el plan de marcado se aplican. Por lo tanto, es posible usar las mismas reglas de selección de troncales telefónicas que se usarían si se marcara directamente el número desde un teléfono.
- Agent: Este tipo de canal se comporta como un alias que



puede ser asumido por diversas extensiones, o incluso, por llamadas que provienen desde el exterior. Para hacer uso de un canal de tipo Agent, se debe de definir el canal en el archivo `/etc/asterisk/agents.conf` y luego conducir una llamada realizada por el dispositivo interesado a un contexto que ejecute la aplicación `AgentLogin`, o (de forma equivalente) una llamada conectada directamente a la misma aplicación `AgentLogin`. El uso del canal de tipo Agent permite que se adquiera este seudónimo con cualquier extensión que esté siendo usada para la comunicación.

### **Plan de marcado**

Un plan de marcado (`dialplan`) es la fuente de configuración que usa Asterisk, la cual generalmente se carga desde el archivo `/etc/asterisk/extensions.conf`, pero también puede cargarse desde una base de datos. El archivo `/etc/asterisk/extensions.conf` puede a su vez hacer referencia a más archivos que definen porciones del plan de marcado, para propósitos de organización. Cada llamada recibida por Asterisk especifica una secuencia de caracteres que representan el número o extensión a contactar, la cual se denomina cadena de marcado (`dialstring`). Esta cadena de marcado se examina en busca de una coincidencia, la cual determina el

comportamiento posterior de Asterisk. Dentro del plan de marcado se reúnen cuatro conceptos importantes: contextos, extensiones, prioridades y aplicaciones [2].

Un contexto es una sección del plan de marcado, agrupada bajo un nombre simbólico (por ejemplo from-internal), que define grupos de extensiones. Cada extensión definida dentro de un contexto está completamente aislada de interactuar con extensiones de cualquier otro contexto, excepto si se configura deliberadamente esta interacción [2]. Esto permite, por ejemplo, configurar dos (o más) contextos pertenecientes a dos empresas distintas servidas por el mismo servidor Asterisk, con números de extensiones que se duplican, pero que permanecen independientes. Otra capacidad importante posible con contextos es el de nivel de privilegios. Las extensiones en contextos distintos pueden tener privilegios distintos (por ejemplo, ser capaces o no de realizar llamadas internacionales).

Todas las tecnologías de canales que pueden introducir llamadas a ser manejadas por Asterisk requieren indicar el contexto dentro del cual se buscará la secuencia de marcado usada. Las funcionalidades provistas por el mismo son las que definen el rol de

una extensión como “interna”, “llamada del exterior”, u otra clasificación requerida por el administrador.

Una extensión, dentro de un contexto, es un nombre (que puede ser una cadena numérica como 1234, pero no está estrictamente limitada a esto) que encapsula una secuencia de pasos que definen acciones a tomar para la llamada [2]. Un contexto puede tener una cantidad arbitraria de extensiones. Cada uno de los pasos por los que se conduce la llamada está definido por una línea de texto dentro del contexto, bajo la siguiente sintaxis:

`exten => EXTENSION,PRIORIDAD,APLICACIÓN([parámetros])`

La prioridad es el número del paso dentro de la definición de la extensión. Todas las extensiones deben definir el paso (prioridad) 1, y los siguientes pasos pueden estar numerados explícitamente, o puede usarse la etiqueta n (para NEXT, siguiente) para facilidad de configuración.

Una aplicación, en Asterisk, es una orden que ejecuta una acción específica en la llamada. Para cada tarea común que se puede hacer sobre la llamada, hay una aplicación para ejecutarla. Por

ejemplo:

- `Answer()`: marcar una llamada como contestada
- `Hangup()`: colgar la llamada, finalizándola
- `Playback(archivo)`: reproducir un archivo de audio en la llamada
- `Background(archivo)`: reproducir un archivo de audio, pero si se digita una secuencia de dígitos a través del teclado del teléfono (transmitido con DTMF), la aplicación salta a la extensión indicada. Esta aplicación es la base para construir menús interactivos.
- `Goto()`: saltar a un nuevo contexto, extensión y/o prioridad arbitrarios. Útil para construir bucles.
- `Dial()`: generar una nueva llamada telefónica que marca al canal indicado, y conecta la llamada nueva con la existente. Esta es la manera principal por la cual el plan de marcado genera llamadas salientes usando el contexto.

Además de las capacidades implementadas por el plan de marcado y las aplicaciones que se invocan a través de él, Asterisk ofrece otro mecanismo de control, el cual es aprovechado por el CallCenter. Este mecanismo, llamado AMI, consiste en un protocolo de red que funciona por omisión en el puerto TCP 5038 en el cual escucha

Asterisk. El cliente AMI inicia una conexión TCP/IP a este puerto y pasa por la autenticación configurada en el archivo `/etc/asterisk/manager.conf`. Entonces el cliente AMI tiene acceso a eventos que son comunicados a través de la conexión y que representan acciones que Asterisk ejecuta en tiempo real. Además, en la misma conexión, la aplicación puede enviar comandos para pedir reportes sobre el estado actual de Asterisk y para iniciar acciones sobre las llamadas en curso.

Algunos de los comandos que pueden lanzarse a través de AMI, de interés al CallCenter son:

- Generación de nuevas llamadas sin necesidad de marcado manual. Esta capacidad es la base de la marcación automática del CallCenter.
- Control de colas, incluyendo agregar y quitar agentes, además de pausado de colas.
- Asignación de extensiones a agentes que usan el tipo de canal Agent, así como la revocatoria de tal asignación.
- Reportes del estado de canales, colas y agentes.

Algunos de los eventos que son reportados por la conexión AMI y que son de interés al CallCenter son:

- Creación de nuevo canal
- Colgado de una llamada en un canal
- Inicio del marcado de un número
- Término de la generación de una llamada (cuando se sabe si el marcado ha tenido éxito)
- Ingreso y salida de una llamada a una cola
- Enlace entre dos canales activos
- Ingreso y salida de agente al sistema (es decir, activación o desactivación del canal de tipo Agent)

### **3.6 FreePBX – Interfaz web para configuración de Asterisk**

FreePBX es un proyecto de código abierto orientado a la configuración de un sistema Asterisk a través de una interfaz web. Con el uso de FreePBX, el administrador del sistema ya no requiere armar a mano la totalidad de un plan de marcado coherente para sus necesidades. En lugar de esto, el administrador instala FreePBX (o usa la copia ya instalada por la distro de telefonía como Elastix) y organiza los requerimientos de telefonía usando la interfaz web, previo login con usuario y contraseña.

Para el administrador, FreePBX organiza las capacidades de telefonía de Asterisk para implementar un IP-PBX que ofrece entre

sus capacidades básicas, las siguientes:

1. Extensiones internas, las cuales pueden ser de tipo VoIP (SIP o IAX), DAHDI, H.323, o CUSTOM (para permitir al administrador introducir a mano la tecnología telefónica). Estas extensiones internas tienen un número de extensión asignado en el contexto from-internal.
2. Troncales, las cuales son conexiones telefónicas que conectan múltiples llamadas simultáneas y que representan la intercomunicación con el exterior. Al igual que con las extensiones, se puede elegir la tecnología telefónica de la troncal.
3. Rutas salientes, las cuales consisten en patrones de dígitos que al ser marcados, dirigen la llamada generada por una extensión interna hacia una troncal que se elige de un grupo de troncales disponibles. Se pueden aplicar transformaciones para manipular el número a marcar antes de que sea presentado a la troncal.
4. Rutas entrantes, las cuales consisten en reglas para el manejo de llamadas que ingresan al sistema, aplicadas según la troncal y la secuencia marcada (DID). La llamada entrante puede ser manipulada y enviada a varios tipos de destinos en el plan de marcado, incluyendo IVRs, colas,

extensiones individuales, grupos de timbrado y otros servicios implementados en el plan de marcado.

5. Colas, las cuales se asocian a una extensión que se puede marcar desde un teléfono para entrar como usuario regular. La configuración de la cola usada por FreePBX es la que eventualmente se utiliza para ejecutar el CallCenter.
6. IVRs, las cuales FreePBX puede armar con las aplicaciones de Asterisk expuestas anteriormente. Esto incluye la asignación de dígitos para salir del IVR y mandar la llamada a cualquier otra entidad de telefonía definida, incluyendo extensiones.

### **3.7 Elastix – Sistema de comunicaciones unificadas**

La definición y propósito del proyecto Elastix de acuerdo a la página oficial del proyecto son: “Elastix es un software de código abierto para el establecimiento de comunicaciones unificadas. Pensando en este concepto el objetivo de Elastix es el de incorporar en una única solución todos los medios y alternativas de comunicación existentes en el ámbito empresarial.” [12]

El objetivo principal al cual está orientado Elastix, es la de presentar en una solo paquete instalable varias funcionalidades útiles



orientadas a la comunicación empresarial, de las cuales la más importante es la telefonía VoIP. La siguiente lista es una muestra de las características que Elastix integra en su instalación base:

- Telefonía tradicional (PSTN) y VoIP. Esto se logra incluyendo los proyectos de DAHDI para la interconexión telefónica PSTN, Asterisk para la telefonía VoIP (incluyendo los soportes SIP e IAX) y delegando gran parte de la configuración del plan de marcado a FreePBX.
- Soporte para recepción y envío de faxes, tanto a través de líneas analógicas como VoIP. Esto se logra haciendo uso de los proyectos HylaFax (<http://www.hylafax.org>) para la cola de faxes y manipulación de archivos de fax, ghostscript para la conversión de archivos PDF y PS a TIFF (el formato gráfico nativo de envío y recepción de fax) y el proyecto IAXModem (<http://iaxmodem.sourceforge.net>) que presenta una interfaz de terminal modem para HylaFAX y la convierte en una llamada VoIP en el protocolo IAX para la interacción con Asterisk. Elastix provee administración de colas de fax y notificación vía correo de envío y recepción de faxes.
- Integración de cuentas de correo. Para esta funcionalidad, Elastix hace uso de Postfix (<http://www.postfix.org>) para correo entrante (SMTP), Cyrus (<http://cyrusimap.org>) para el

soporte de consulta de correos vía IMAP/POP3, filtrado de spam usando SpamAssassin (<http://spamassassin.apache.org>) y RoundCube (<http://roundcube.net/>) como un cliente IMAP de correo a través de una interfaz web. Elastix además provee creación y administración de cuentas de correo y de dominios de correo, configuración de mensajes de vacaciones, activación y desactivación de filtro de spam. Adicionalmente se provee configuración de listas de correo usando Mailman (<http://www.list.org>).

- Capacidad para chat usando el protocolo XMPP, con el servidor Openfire (<http://www.igniterealtime.org/projects/openfire/>).
- Herramientas de administración del sistema, las cuales incluyen configuración de interfaces de red, de servidor DHCP, respaldo y restauración de configuración, detector de hardware de telefonía DAHDI, creación masiva de extensiones a partir de archivos subidos al servidor (Lote de Extensiones) y escaneo de la red para la detección de teléfonos VoIP que son subsecuentemente configurables desde el servidor central Elastix (Configurador de Endpoint).
- Varios reportes, entre los cuales están llamadas realizadas,

bitácora de Asterisk, uso simultáneo de canales, tiempo usado por extensión, y llamadas perdidas.

La distribución Elastix está basada en CentOS 5. La instalación de Elastix, y su administración fuera del ámbito de telefonía, es casi idéntica a la de un sistema CentOS (basado en RedHat).

Como parte de los requerimientos de las herramientas de telefonía y administración, Elastix instala las siguientes herramientas de desarrollo, las cuales se aprovechan a su vez para la implementación del CallCenter:

- MySQL, la base de datos de código abierto. En Elastix, esta base de datos almacena la información persistente para FreePBX, RoundCube, el configurador de endpoint, vTiger (si está instalado) y a2billing (si está instalado).
- PHP 5.1, el lenguaje de programación en el cual está escrito FreePBX, Elastix (incluyendo sus herramientas de sistema), RoundCube, vTiger, y a2billing.

## **CAPÍTULO 4**

### **4 Descripción del proyecto**

En este capítulo se muestra detalles del diseño del sistema Call Center. Se muestra la descripción del sistema en función de roles e interacción de usuarios e interacción con otros sistemas. Luego se discuten características de las interacciones entre los usuarios y el sistema. Por último se muestran los componentes de la solución implementada y los pasos a seguir para una sesión de trabajo.

#### **4.1 Descripción de las funciones del sistema**

Para describir las funciones del sistema Elastix CallCenter, se muestran primero los roles de los usuarios, con las tareas que deben de realizar, y las funcionalidades que el sistema le ofrece para realizarlas. Luego se explica la manera en que los usuarios en estos roles interactúan con el sistema. A continuación, se expone el

intercambio de datos entre el sistema Elastix CallCenter y el exterior. Por último se mencionan los requerimientos de rendimiento y seguridad, y la manera en la cual el sistema va a cumplirlos.

#### **4.1.1 Roles de los usuarios del sistema**

Para el diseño e implementación del sistema del call-center, los roles que pueden asumir los usuarios son los siguientes: agente, administrador de campañas, y supervisor de campañas.

##### **El agente**

Este es el usuario que está encargado de atender las llamadas que son mediadas por el sistema de call-center.

Las tareas asignadas al agente son las siguientes:

- Recibir la llamada que le ha sido asignada, y seguir el guión de interacción que ha sido elaborado por la campaña.
- Mantener el interrogatorio con el contacto telefónico con el objetivo de llenar el o los formularios que han sido asociados con la campaña.
- En caso de que el contacto esté interesado en la tema

de la campaña, pero no pueda atender en el momento en que recibió la llamada, el agente debe agendar una llamada nueva a realizar al mismo número. Esta llamada, según el tema de la campaña, puede ser redirigida al mismo agente, o si es aceptable, a cualquier agente disponible. Generalmente el agendamiento incluye un horario negociado en el cual el contacto puede atender el tema de la campaña.

- Anunciar descansos aprobados. Durante la sesión de atención de llamadas, el agente requiere ausentarse de la atención por intervalos. Las razones de estas ausencias de la atención pueden ser variadas. Por ejemplo, ausencia para almuerzo, para usar el baño, por reunión, por interrupción no planificada. Durante estos intervalos de descanso, el sistema no debe de rutear llamadas al agente.

El sistema requiere ofrecer las siguientes funcionalidades y servicios al agente:

- El sistema marca automáticamente un número de la campaña. Eventualmente la llamada resultante de esta marcación resulta asignada a un agente

disponible. Esto libera al agente de tener que elegir cuál número marca a continuación.

- Se lleva la pista de cuál número (contacto) está siendo atendido por el agente. Todas las interacciones que registran datos durante la atención a este número terminan asociadas a este contacto. El agente se libera de la necesidad de mantener esta asociación de forma explícita.
- El guión de atención para el agente está almacenado como parte de la campaña. Por lo tanto, este guión se puede consultar en cualquier momento durante la atención.
- El formulario de preguntas a realizar durante la atención también está registrado en el sistema, para ser consultado durante la atención. Las respuestas llenadas en el formulario son registradas en el sistema para la llamada actual. El agente ya no tiene entonces que llevar la cuenta del número al que corresponden las respuestas del formulario.
- El sistema ofrece un formulario de agendamiento en el cual se anota el horario de atención tentativa, y si el mismo agente debe de atender la nueva llamada. El

registro de la nueva llamada a realizar es otra tarea que el agente ya no tiene que realizar.

- Para poder marcar los descansos, se implementa una interfaz que permite al agente marcar el inicio y final de un descanso, además del tipo de descanso. El sistema se encarga de proveer la fecha y hora de los respectivos inicio y final. Con esta capacidad, el agente se libra de anotar tales horas.

### **El administrador de campañas**

Esta es la persona que se encarga de informar al sistema de nuevas campañas. Esta persona puede ser la misma persona que el supervisor de campañas u otra distinta.

Las tareas asignadas al administrador de campañas son las siguientes:

- Creación de una nueva campaña. Esta actividad incluye la alimentación de la lista de contactos a marcar al sistema, ingreso del guión, y selección de los formularios apropiados.
- Ingreso de formularios al sistema. Esta tarea puede incluir la redacción inicial de las preguntas de los



formularios. Cada formulario ingresado debe ser luego asociado a una campaña en el sistema.

- Recolección de los datos recogidos por la campaña luego de haber finalizado. Con el procedimiento manual implementado antes del desarrollo del sistema, esto requería la consolidación en un solo documento (libro de cálculo) de la información de todos los agentes.
- Crear y administrar colas de llamadas en el sistema. Estas colas se asignan luego a campañas específicas para su funcionamiento.
- Administrar troncales telefónicas para la generación de las llamadas. Si se considera necesario, estas troncales pueden ser usadas como dedicadas a campañas específicas.
- Crear y administrar los agentes asignados a la atención de las llamadas.

El sistema requiere ofrecer las siguientes funcionalidades y servicios al administrador de campañas:

- Se provee una interfaz de creación de campañas. En esta interfaz, el administrador de campañas puede

especificar la cola a usar para la generación de las llamadas, los formularios a asociar a esta campaña, el guión que se debe de usar, y otros parámetros de configuración. Además, se provee el mecanismo para cargar la lista de contactos a marcar.

- Se muestra un listado de las campañas que han sido creadas. Además de proveer una manera de modificar los parámetros de una campaña ya creada, esta interfaz da acceso a la descarga de los datos recogidos en la campaña. Esta descarga corresponde a la consolidación de los datos recogidos por todos los agentes, ya clasificados para análisis por parte del administrador de campañas.
- El sistema ofrece un módulo de creación de formularios. Los formularios creados por este mecanismo se componen de campos con una etiqueta a ser usada como la pregunta, y un tipo de dato (validación básica). Luego de crearse el formulario, este queda disponible para ser asociado a múltiples campañas. Los formularios se muestran en la consola de agente mencionada anteriormente, al ser asignada una llamada correspondiente a la campaña. La

descarga de los datos recogidos incluye, para cada contacto, todos los valores recogidos de todos los formularios.

- La creación y administración de colas y troncales, como entidades telefónicas independientes, se delega a FreePBX. La interfaz de administración de campañas lee la base de datos de FreePBX para alimentar la selección de colas y de troncales para la creación de la campaña.
- Se dispone de un módulo para la creación y administración de agentes. A diferencia de la administración de colas y troncales, FreePBX no tiene una interfaz dedicada a la administración de agentes. Sin asistencia de FreePBX, el administrador tendría que modificar archivos de texto usando la consola de texto. Por lo tanto, la interfaz provista libra al administrador de la posibilidad de cometer errores.
- El sistema posee además un módulo para la creación y administración de descansos. Sólo los tipos de descansos creados a través de este módulo podrán ser usados por los agentes durante las campañas. Esto impone un nivel de organización a la auditoría de

descansos consultada luego por el supervisor de campañas.

### **El supervisor de campañas**

Esta es la persona encargada de supervisar el funcionamiento en tiempo real de cada una de las campañas, y de intervenir para remover interrupciones y distracciones.

Las tareas asignadas al supervisor de campañas son las siguientes:

- Minimizar la ocurrencia de agentes ociosos.
- Detectar síntomas de interrupciones en el servicio telefónico, y ayudar a resolverlos si es posible.
- Autorizar el uso de descansos por parte de los agentes, y limitar el posible abuso de tales descansos.
- Asistir en casos excepcionales que no están cubiertos por los guiones o formularios de las campañas.
- Reunir y analizar estadísticas sobre el funcionamiento del callcenter, a fin de detectar patrones no evidentes durante el funcionamiento en vivo de la campaña.
- Definir los tipos de descansos a los cuales los agentes están autorizados.

El sistema requiere ofrecer las siguientes funcionalidades y servicios al supervisor de campañas:

- Monitoreo centralizado del estado actual de las actividades de la campaña. Este monitoreo ocurre con relativa independencia de la supervisión física de los agentes.
- Notificación del estado de actividad de cada agente. Esto consiste en si el agente ha iniciado sesión, si está atendiendo una llamada, y si ha tomado un descanso.
- Reportes que muestren, para las campañas (en curso y terminadas), las estadísticas de atención y actividad. Estos reportes incluyen: Detalle de Estado de Llamadas, Reporte de Descansos, Llamadas por Hora, Tiempo de Espera en Cola, y Tiempo Total de Atención.
- Reportes sobre la eficiencia de cada agente individual durante intervalos de tiempo.

#### **4.1.2 Interacción con los usuarios**

Para el agente, la interacción con el sistema consiste en el enrutamiento periódico y automático de las llamadas. Cada una de las llamadas que terminan siendo asignadas al agente debe de ir acompañada de una actualización de la información mostrada en la consola de agente. Esta información incluye el número contactado, el nombre del contacto (si está disponible), y la información extra asociada al contacto (en caso de haber estado presente). Además debe de actualizarse el guión asociado a la campaña de la cual se sacó la llamada enrutada, así como el formulario. En caso de agendamiento, el agente debe de llenar un formulario adicional que indica la información de horario para la generación de la nueva llamada.

Para el administrador de campañas, la interacción con el sistema es más tradicional, y se asemeja a una interacción ordinaria CRUD. En este esquema se ajustan la creación y administración de campañas, la administración de descansos, la creación y actualización de agentes, y el ingreso de formularios.

Para el supervisor de campañas, la interacción con el sistema tiene dos modos principales. En el primer modo de interacción, el supervisor solamente consulta los reportes que muestran los parámetros de rendimiento de las campañas, colas, y agentes. En el segundo modo de interacción, el supervisor consulta una vista del estado del sistema que se actualiza en tiempo real, y que permite al supervisor actuar de inmediato en el caso de que ocurran problemas con el flujo de llamadas.

La figura 4.1 muestra los roles de los usuarios en relación al sistema Call Center.

#### **4.1.3 Integración con otros sistemas y transferencia de datos**

Para el funcionamiento del sistema de CallCenter, el principal mecanismo de transferencia de datos desde y hacia el sistema consiste en la subida y descarga de archivos CSV. De los módulos que manejan subidas de archivos para la alimentación del sistema, el más importante es el módulo de alimentación de números a marcar para las campañas salientes. En un estado temprano del análisis, se empezó con la idea de tratar con un listado simple de números. Sin

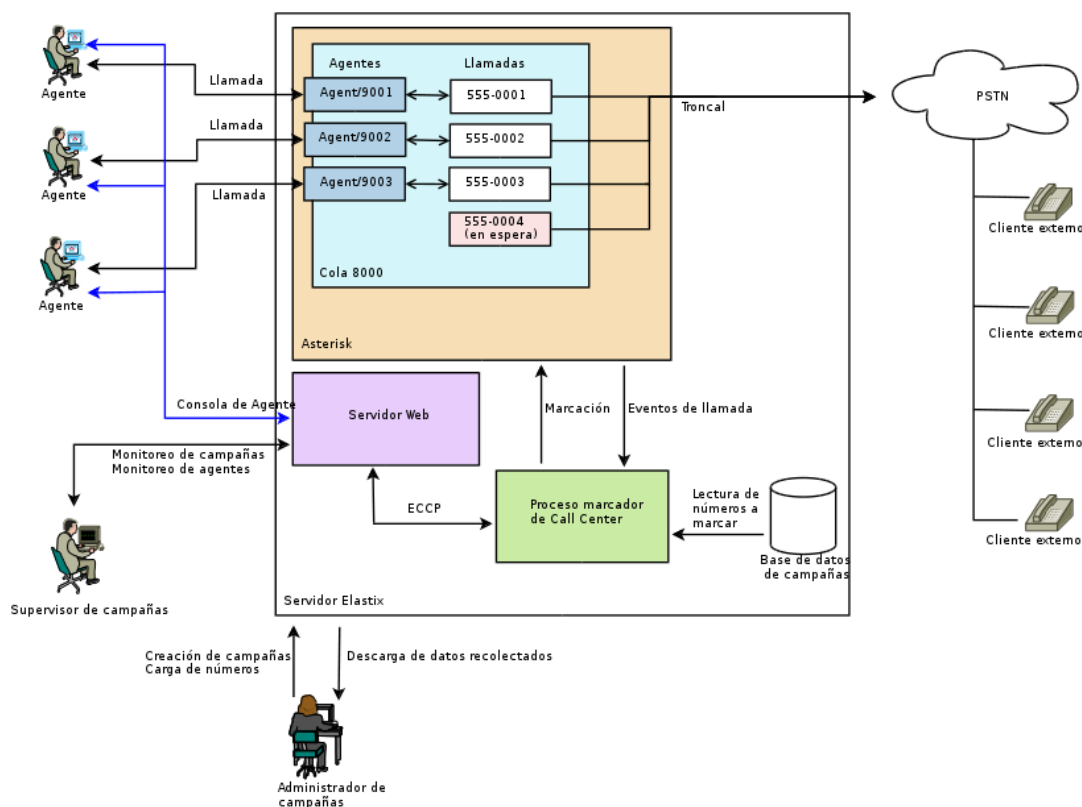


Figura 4.1: Sistema Call Center y roles de los usuarios

embargo, se encontró que frecuentemente es necesario que el agente conozca información adicional sobre el número cuya llamada ha sido conectada. Esta información adicional, en la forma de atributos, debe también de introducirse como parte de la campaña. Estos atributos se representan como columnas adicionales en el archivo CSV, al lado de la primera columna que contiene el número telefónico a marcar. Para representar las etiquetas de cada atributo, se utiliza la primera fila del archivo CSV.



Para la descarga de los datos de la campaña en formato CSV, el formato generado es más complejo. Esto se debe a que el formato debe representar el resultado del llamado a todos los números, así como los datos recogidos por todos los formularios. Luego de algunas pruebas, se llegó a las siguientes columnas mínimas para representar la información:

- Número telefónico marcado
- Estado final de la marcación
- Número del agente que atendió la llamada
- Fecha y hora de la atención
- Duración de la llamada
- Uniqueid (valor asignado por Asterisk a la llamada durante el procesamiento)
- Código de fallo (en caso de que la llamada fallase)
- Texto asociado al código de fallo

Además de estas columnas mínimas, la descarga de la información de la campaña incluye como columnas adicionales los valores recogidos por los formularios que sean asociados a la campaña. Todas las columnas adicionales llevan como primera fila, la etiqueta del campo

del formulario asociado.

La figura 4.2 muestra el flujo de información cargada y recolectada en una campaña:

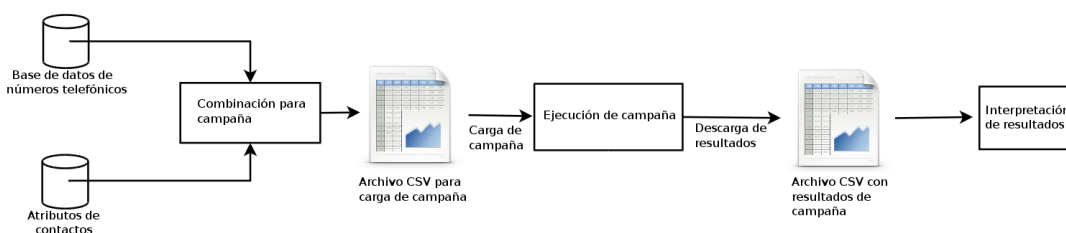


Figura 4.2: Flujo de información cargada y recolectada

En el caso de las llamadas entrantes, no existe una lista predeterminada de números, puesto que las llamadas ingresan desde el exterior al sistema con independencia de las acciones del marcador. Por lo tanto, la capacidad del sistema de cargar con anticipación la información sobre el contacto que atiende el agente es limitada. Sin embargo, si se ha definido una campaña entrante que administra una cola en particular, es posible asociar formularios de recolección de datos, y también es posible descargar datos que hayan sido recogidos por la campaña. El formato de

recolección de datos es similar al de las campañas salientes, con la diferencia que los números que aparecen listados no contienen un código de fallo, pero tienen un estado de atención que indica si la llamada fue atendida, o se ha perdido sin haber sido asignada a un agente.

Además de la transferencia de datos implementada con los archivos CSV, se observó que la consola de agente requiere, según las necesidades del implementador, asociarse con otras interfaces web. Esto toma frecuentemente la forma de un URL con parámetros que pueden ser reemplazados por propiedades y atributos de la llamada. Al recibir una llamada, la campaña define la plantilla del URL que debe de abrirse, y entonces se realiza el reemplazo de las propiedades.

Se excluye explícitamente del análisis de integración con otros sistemas, todo intento de hacer que el sistema de marcación cargue durante el proceso de la campaña, el número en sí a marcar, o atributos correspondientes a él, por cualquier medio de RPC con otro computador o fuente de datos.

#### **4.1.4 Requerimientos de rendimiento, confiabilidad y seguridad**

Dentro del sistema del CallCenter, los principales requerimientos de rendimiento y confiabilidad se centran en el funcionamiento de la consola de agente.

1. La consola de agente debe de comportarse como una interfaz de tiempo real. Esto significa que no basta con que la consola de agente muestre la información de una llamada asignada al agente. Además la consola debe mostrar esta información de forma inmediata, o con un mínimo retardo.
2. El mecanismo escogido para implementar la actualización en tiempo real debe de ser escalable con el número de agentes totales, y también con el número de agentes simultáneamente conectados al sistema.
3. El supervisor del CallCenter requiere reportes que se actualicen en tiempo real: Monitoreo de Agentes, Monitoreo de Campañas, Monitoreo de Colas.

Para cumplir con estos requerimientos de tiempo real, se debe establecer un mecanismo de comunicación periódico o

constante entre el sistema de CallCenter, y la implementación escogida para la interfaz de la Consola de Agente. Se requirieron varios intentos para realizar esto correctamente, los cuales se detallan en las secciones 5.4 a 5.5. El mecanismo usado para la Consola de Agente sirve también para los reportes del supervisor.

Los requerimientos de seguridad se centran en la separación y limitación de privilegios:

- Los agentes deben autenticarse en el sistema para poder iniciar una sesión.
- Ningún agente debe poder interferir en los datos o atención de otro agente.
- Ningún agente debe tener acceso a los módulos de configuración de campañas, agentes y otros parámetros reservados al administrador del Call Center.

Para implementar la separación entre los roles es suficiente con hacer uso del soporte de grupos de la plataforma Elastix, la cual restringe los módulos accesibles a un subconjunto indicado por el administrador.

## 4.2 Diagramas de roles e interacciones

En la figura 4.3, el lado cliente (agente) consiste del navegador que muestra la interfaz Elastix, más la extensión telefónica (física o softphone) asignada para atender las llamadas. Se asume que la extensión ya está registrada correctamente en el sistema Elastix. Los diagramas completos de interacciones se muestran en el anexo 2.

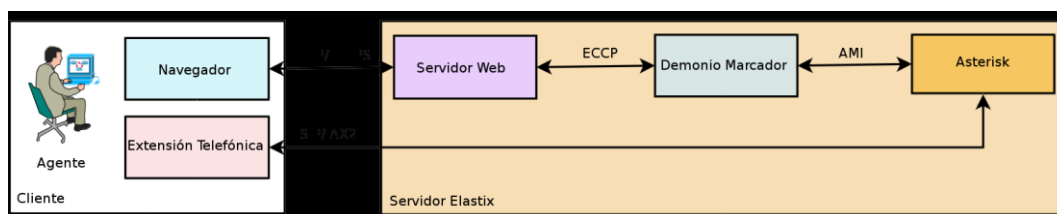


Figura 4.3: Protocolos usados para interacción de agente

En todas estas interacciones, el navegador se comunica con el sistema Elastix a través del protocolo ordinario HTTP. El servidor web en el sistema Elastix se comunica a su vez con el proceso marcador con el protocolo ECCP, y finalmente, el proceso marcador se comunica con Asterisk a través del protocolo AMI.

Para la interacción de sesión completa, se asume que el agente parte de la interfaz de consola de agente, en su modo de petición

de credenciales de login. A través de la interacción de sesión mostrada en el anexo 2, se llega eventualmente al estado de espera de atención de llamadas, hasta que se manda a finalizar la sesión.

Para la interacción de atención de la llamada, se asume que el agente parte de la interfaz de consola de agente, en su modo de espera de atención de llamadas. La interacción de atención mostrada en el anexo 2 parte desde el momento en que Asterisk asigna una llamada conectada al agente, y el agente termina viendo esta información. En la versión de la interacción con el agente estático, este agente no tiene la opción de rechazar la llamada una vez asignada.

### **4.3 Descripción Técnica de la Solución**

En esta sección se muestran los componentes necesarios para implementar el sistema. A continuación se listan los pasos que siguen el administrador y el agente para trabajar con el sistema.

### 4.3.1 Componentes del sistema

Para satisfacer los requerimientos, las principales herramientas de las que se dispone son las disponibles en el servidor Elastix. En vista de que Elastix es esencialmente una plataforma LAMP, los componentes residentes en el servidor de la solución que se implementa consisten en los siguientes:

- Un esquema de base de datos administrado por el motor de base de datos MySQL provisto por Elastix. Esto cubre todas las necesidades de almacenamiento persistente requeridos por todos los demás componentes.
- Un conjunto de módulos de Elastix para servir contenido a través de la página Web. Estos módulos están escritos en PHP por ser este lenguaje de programación el elegido para las porciones Web de la plataforma Elastix, así como de FreePBX. Estos módulos hacen uso de las facilidades provistas por las bibliotecas Elastix. Tales facilidades incluyen: autenticación básica de usuarios, definición de usuarios y grupos con autorizaciones configurables para cada módulo, soporte básico de



internacionalización, plantillas HTML, bibliotecas para formularios y reportes. Esto cubre todos los módulos orientados a CRUD básico.

- De entre los módulos Web de Elastix, se señalan en especial los módulos orientados a tiempo real: la Consola de Agente, Monitoreo de Campañas, Monitoreo de Agentes, Monitoreo de Colas Entrantes. Además de hacer uso de las facilidades provistas por Elastix, estos módulos implementan técnicas Comet para permitir el envío de eventos. Estos eventos se usan para actualizar la interfaz en tiempo real.
- Un programa que corre de forma persistente como un servicio de sistema (daemon), escrito en PHP. Este programa es el componente central del sistema de Call Center. Este programa es el encargado de conectarse a Asterisk para determinar el estado de los agentes y las colas, recibir eventos, y generar las llamadas salientes. Además se tiene la responsabilidad de escribir en la base de datos el resultado de la generación de llamadas, y de retransmitir a todos los clientes interesados los eventos de inicio y fin de llamadas.

En el lado de cada uno de los agentes, la solución que se implementa reutiliza los componentes instalados para realizar el método manual del call-center con el sistema Elastix:

- Navegador web compatible capaz de visualizar la interfaz Elastix.
- Para los agentes, un teléfono VoIP asociado con una extensión existente en el sistema Elastix.

#### **4.3.2 Administración del sistema**

Para poder hacer uso del sistema CallCenter para las campañas entrantes o salientes, el administrador debe realizar las siguientes operaciones:

1. Crear una o más colas telefónicas para uso y monitoreo por parte del demonio marcador en la interfaz administrativa de FreePBX.
2. Crear tantos registros de agentes como personas tenga disponibles en el módulo de agentes del Call Center.
3. Asignar los agentes requeridos a cada una de las colas en la interfaz administrativa de FreePBX. Cada campaña que haga uso de estas colas ruteará las

llamadas a los agentes asignados.

4. Crear al menos un formulario en la interfaz de creación de formularios del CallCenter. Estos formularios pueden ser reutilizados entre campañas, y se requiere de al menos uno para asociar a cada campaña creada.
5. En el caso de las campañas entrantes, se deben reservar las colas que serán monitoreadas para las llamadas. Esto se hace en el módulo de colas entrantes del CallCenter.
6. Crear los tipos de descansos (breaks) autorizados para los agentes. Estos descansos se registran en el módulo Administrador de Breaks.
7. Una vez creada la cola, el formulario, y los agentes, el administrador debe de crear la campaña. Para la campaña saliente, se debe además proveer el archivo con la lista de números a marcar, la cual incluye además las propiedades de cada contacto a marcar.

Una vez preparado el sistema, el agente empieza a trabajar:

- Al iniciar sesión, cada agente empieza a recibir y atender llamadas. La información recogida se

acumula en la base de datos.

- Luego de terminada la campaña, el administrador usa el módulo de administración de campañas para descargar los datos recogidos.

## **CAPÍTULO 5**

### **5 Implementación del proyecto**

En este capítulo se describe la manera en que se implementó el proyecto Call Center a través de todos sus componentes. Luego se muestran detalles de la primera versión del marcador y de la Consola de Agente, las limitaciones encontradas y las nuevas capacidades requeridas por los clientes. Se muestra cómo la introducción del protocolo ECCP permitió la reescritura de la Consola de Agente y cómo la adopción de un modelo multiproceso resolvió un problema de rendimiento observado. Por último se listan las limitaciones pendientes conocidas.

#### **5.1 Descripción de la implementación**

Como parte de la implementación del sistema, se muestran los siguientes componentes que forman la solución: el servidor, el cliente, la base de datos y el demonio de marcación.

### 5.1.1 El Servidor

El primer componente de la solución de CallCenter es el servidor Elastix en sí. Una instalación Elastix es una especialización de un sistema CentOS 5.x, y como tal, ofrece los componentes estándar de un servidor GNU/Linux: kernel Linux, bibliotecas C (glibc), administración de paquetes de software (rpm, yum), shell de sistema (bash) y utilidades de configuración. Como parte de la instalación Elastix, el proyecto Call Center hace uso adicional de los siguientes componentes de software:

- DAHDI, los controladores de kernel y el API de usuario que permiten exponer las capacidades de la tarjeta telefónica al resto del sistema. Este componente se usa únicamente cuando se requiere hacer uso de líneas físicas analógicas o digitales. Este fue el caso del cliente original CARTIMEX, donde se contrató una línea ISDN E1 para la salida de las llamadas. Los 30 canales de llamadas que provee esta línea quedan entonces disponibles para uso por las capas superiores. En caso de una instalación telefónica con troncales SIP o IAX2, la capa DAHDI es

opcional.

- Asterisk, en su versión 1.4 o superior. El paquete Asterisk implementa las variadas tecnologías de telefonía bajo un esquema unificado. Estas tecnologías incluyen acceso a los canales de la troncal DAHDI (en caso de estar presente), las troncales VoIP (en caso necesario) y los clientes telefónicos SIP e IAX2. Además, Asterisk implementa la abstracción del plan de marcado, necesario para tener acceso a las colas y extensiones a través de sus números de extensión. Finalmente, Asterisk ofrece el protocolo AMI a través del puerto 5038/tcp, lo cual permite el control y monitoreo de Asterisk en tiempo real por parte de FreePBX y el demonio marcador del Call Center.
- MySQL, desde la versión 5.0. El esquema usado por el Call Center se detalla en la sección 5.1.3. Además MySQL es la base de datos que se usa para almacenar la configuración de FreePBX.
- PHP, desde la versión 5.1. Este es el lenguaje de programación en el cual están escritos los componentes de FreePBX, Elastix, los módulos de

administración de Call Center y el demonio marcador. PHP se usa en las modalidades de lenguaje embebido en el servidor web Apache y como intérprete independiente para el demonio marcador.

- FreePBX, en su versión 2.6 o superior. El paquete FreePBX organiza las capacidades de Asterisk bajo un GUI de control. Todas las operaciones de configuración que no son explícitamente orientadas al Call Center se delegan a FreePBX. Las operaciones de configuración delegadas incluyen la creación y administración de las extensiones SIP e IAX2, la configuración de troncales para la salida de las llamadas generadas, la creación de las colas de llamadas y la asignación de agentes a cada una de las colas.
- El framework de Elastix, desde la versión 1.6. Este componente es responsable de proveer el GUI de uso común de los módulos, el esquema de autenticación y autorización, una administración estándar de reportes y formularios y el acceso a bibliotecas Javascript.



Además de los componentes de software mencionados anteriormente, el servidor debe estar conectado a la red local, y requiere además que todas las terminales clientes (navegadores y teléfonos VoIP) están conectados a la misma red local.

### **5.1.2 El Cliente**

El siguiente componente del sistema del Call Center es el cliente. En el sistema, el cliente consiste de la terminal remota en la red local, la cual contiene un navegador Web. Con este navegador, se usa el protocolo estándar HTTP o HTTPS para mostrar la interfaz del módulo Elastix, la cual es una página Web con imágenes y Javascript.

Para la mayoría de los módulos, es suficiente generar el contenido HTML del reporte o del formulario a partir de la base de datos. Para los módulos que requieren funcionamiento en tiempo real (Consola de Agente, Monitoreo de Campañas, Monitoreo de Agentes, Monitoreo de Llamadas Entrantes), el servidor debe además enviar los cambios al navegador cuando se detecta un cambio en el estado monitoreado. La verificación de estos cambios y su

correspondiente envío se implementan a través de AJAX con código Javascript en el navegador. Para la primera versión de cada módulo se usaron peticiones periódicas del estado monitoreado al servidor cada pocos segundos. Luego de constatar problemas de rendimiento, se implementaron otras técnicas como Long Polling y Server Sent Events. Los detalles del análisis que condujo a este cambio se muestran en la sección 5.4.

En el caso específico de la Consola de Agente, el cliente debe adicionalmente incluir una extensión telefónica registrada en el Elastix, de tecnología SIP o IAX2. El número asociado a esta extensión se usa cuando se requiere el login del agente. Se pueden usar extensiones VoIP físicas, o softphones que se ejecutan en la misma terminal que el navegador.

Los siguientes módulos que implementan las funcionalidades web del Elastix Call Center:

### **Consola de Agente**

Este módulo implementa la interfaz que los agentes usan para ver la información que corresponde a la llamada que atienden actualmente. Luego de que el agente se autentica (véase Figura 5.1) se le muestra el modo de atención (véase Figura 5.2) Para cada llamada atendida, el módulo tiene tres pestañas que corresponden respectivamente a la información de la llamada, el guión de atención para la campaña y los formularios a llenar para la llamada. Si la campaña define URLs externos, se crean pestañas adicionales para mostrarlos. A la izquierda se encuentran los botones para colgar la llamada y para entrar o salir de pausa.

### **Llamadas salientes**

Esta sección contiene los siguientes módulos:

- Campañas: muestra el listado de las campañas de llamadas salientes que se han creado en el sistema, visibles en la Figura 5.3. Se ofrecen opciones para modificar los parámetros de la campaña y para descargar los datos recolectados. El reporte contiene botones para mostrar la pantalla de creación de una nueva campaña, para activación y desactivación de

una campaña y para el borrado definitivo de una campaña y sus datos.

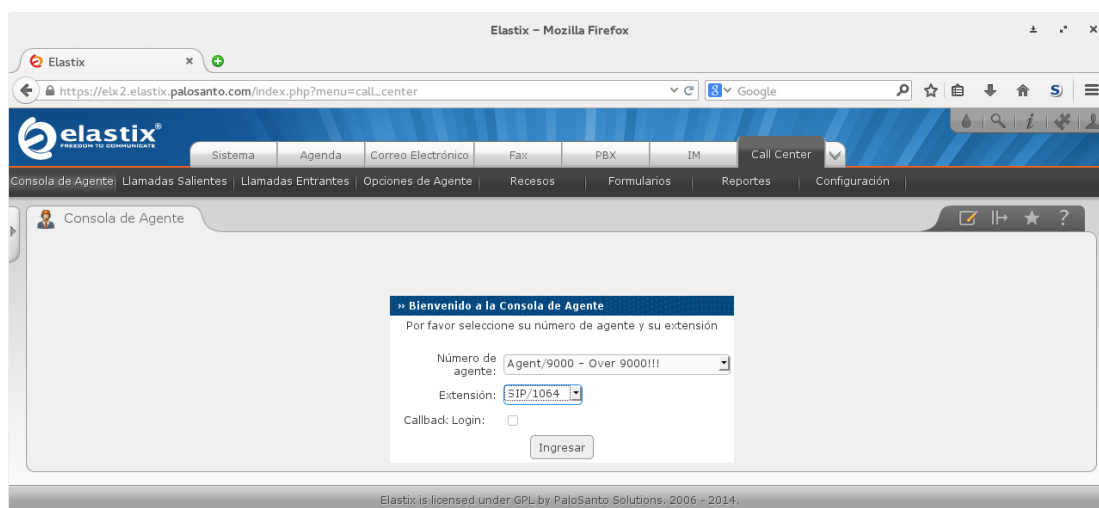


Figura 5.1: Consola de Agente en pantalla de login

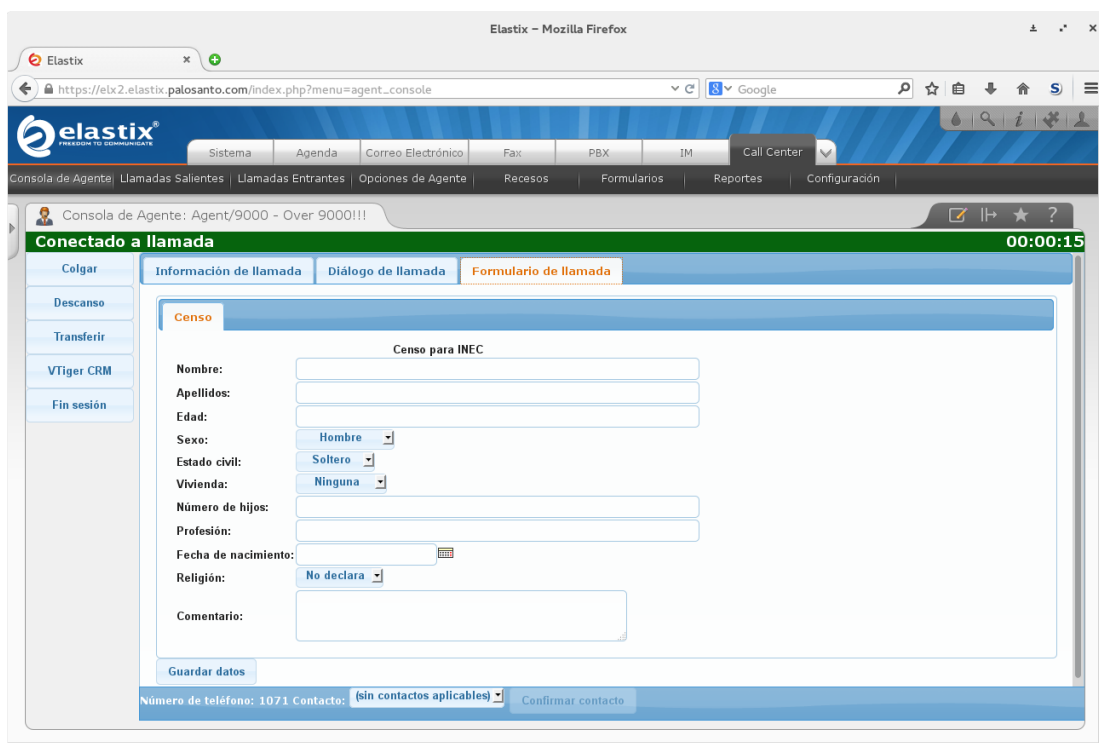


Figura 5.2: Consola de Agente en atención de llamada

The screenshot shows the Elastix web interface in Mozilla Firefox. The main content area is titled 'Listado de campañas' and features a table of campaigns. The table has the following columns: Nombre, Rango Fechas, Horario Diario, Intentos, Troncal, Cola, Llamadas completadas, Promedio, Estado, and Opciones. The data rows are as follows:

Nombre	Rango Fechas	Horario Diario	Intentos	Troncal	Cola	Llamadas completadas	Promedio	Estado	Opciones
<input type="radio"/> Campaña de prueba 2	2011-01-28 - 2012-01-28	00:00:00 - 23:59:00	5	(Dialplan)	8000	12	157	Inactiva	[Editar] [Datos CSV]
<input type="radio"/> Campaña de prueba 3	2011-03-14 - 2011-04-14	00:00:00 - 23:59:00	5	(Dialplan)	8000	17	188	Inactiva	[Editar] [Datos CSV]
<input type="radio"/> Campaña de prueba 4	2011-07-29 - 2013-12-31	00:00:00 - 23:59:00	5	(Dialplan)	8000	2	137	Terminada	[Editar] [Datos CSV]
<input type="radio"/> Prueba campaña noviembre 17	2011-11-17 - 2012-11-17	00:00:00 - 23:59:00	5	(Dialplan)	8000	1	1,066	Terminada	[Editar] [Datos CSV]
<input type="radio"/> Gatito	2012-02-17 - 2013-12-31	00:00:00 - 23:59:00	5	(Dialplan)	8000	45	176	Terminada	[Editar] [Datos CSV]

The interface also includes a left sidebar with navigation options like 'Campañas', 'Lista de No Llamar', and 'URLs Externos'. The top navigation bar contains options like 'Sistema', 'Agenda', 'Correo Electrónico', 'Fax', 'PBX', 'IM', and 'Call Center'. The footer indicates 'Elastix is licensed under GPL by PaloSanto Solutions. 2006 - 2014.'

Figura 5.3: Listado de Campañas Salientes

- Lista de No Llamar: muestra la lista de todos los números ingresados como DNC (Do Not Call), como se muestra en la Figura 5.4. El botón para agregar muestra la pantalla con opción de carga de archivos con números (por omisión) o ingreso de números individuales.

Lista de Teléfonos

Números telefónicos	Fecha de Ingreso	Estado
<input type="checkbox"/> 1777000000	2014-09-16 18:18:28	Activo
<input type="checkbox"/> 1777000001	2014-09-16 18:18:28	Activo
<input type="checkbox"/> 1777000002	2014-09-16 15:29:59	Activo
<input type="checkbox"/> 1777000003	2014-09-16 15:29:59	Activo
<input type="checkbox"/> 1777000004	2014-09-16 15:29:59	Activo
<input type="checkbox"/> 1777000005	2014-09-16 15:29:59	Activo
<input type="checkbox"/> 1777000006	2014-09-16 15:29:59	Activo
<input type="checkbox"/> 1777000007	2014-09-16 15:29:59	Activo
<input type="checkbox"/> 1777000008	2014-09-16 15:29:59	Activo
<input type="checkbox"/> 1777000009	2014-09-16 15:29:59	Activo
<input type="checkbox"/> 1777000010	2014-09-16 15:29:59	Activo
<input type="checkbox"/> 1777000011	2014-09-16 15:29:59	Activo
<input type="checkbox"/> 1777000012	2014-09-16 15:29:59	Activo
<input type="checkbox"/> 1777000013	2014-09-16 15:29:59	Activo
<input type="checkbox"/> 1777000014	2014-09-16 15:29:59	Activo

Figura 5.4: Lista de números en DNC

URLs Externos

Activo	Se abre en	Plantilla de URL	Descripción	Opciones
Yes	Nueva ventana	http://www.google.com/#q={NAME}	Búsqueda de Google sobre el nombre del contacto	[Editar]
Yes	JSONP	http://localhost/~alex/php/practica/php/jsonptest.php?phone={__PHONE__}&sz={Campo1}&y={Campo2}&z={Campo3}	Algún URL raro para probar JSONP.	[Editar]
Yes	Marco embebido	http://localhost/~alex/	Página local	[Editar]

Figura 5.5: Lista de URLs Externos

- URLs Externos: muestra los URLs disponibles para asociar a campañas, como en la Figura 5.5. Cada vez que se atiende una llamada, se puede cargar en el navegador un URL que apunte a una interfaz personalizada, con variables en la plantilla que se pueden reemplazar por atributos de la llamada.

### **Llamadas entrantes**

Esta sección contiene los siguientes módulos:

- Colas: muestra las colas definidas en el sistema que han sido reservadas para las llamadas entrantes. Este módulo aparece en la Figura 5.6.
- Clientes: muestra una interfaz (en la Figura 5.7) para cargar una lista de contactos a ser asociado con los números de teléfono recibidos en llamadas entrantes.
- Campañas entrantes: muestra la lista de campañas entrantes definidas en el sistema, como en la Figura 5.8. Al igual que en el módulo Campañas, se pueden activar, desactivar y borrar con los botones, y crear nuevas campañas con una pantalla separada

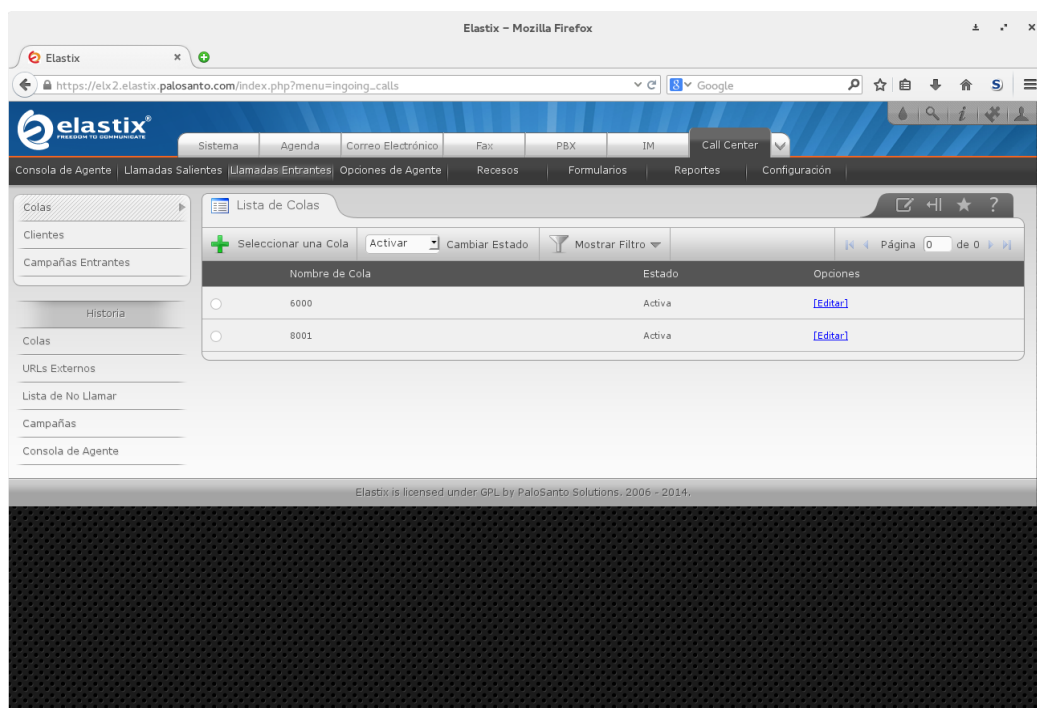


Figura 5.6: Lista de Colas Reservadas

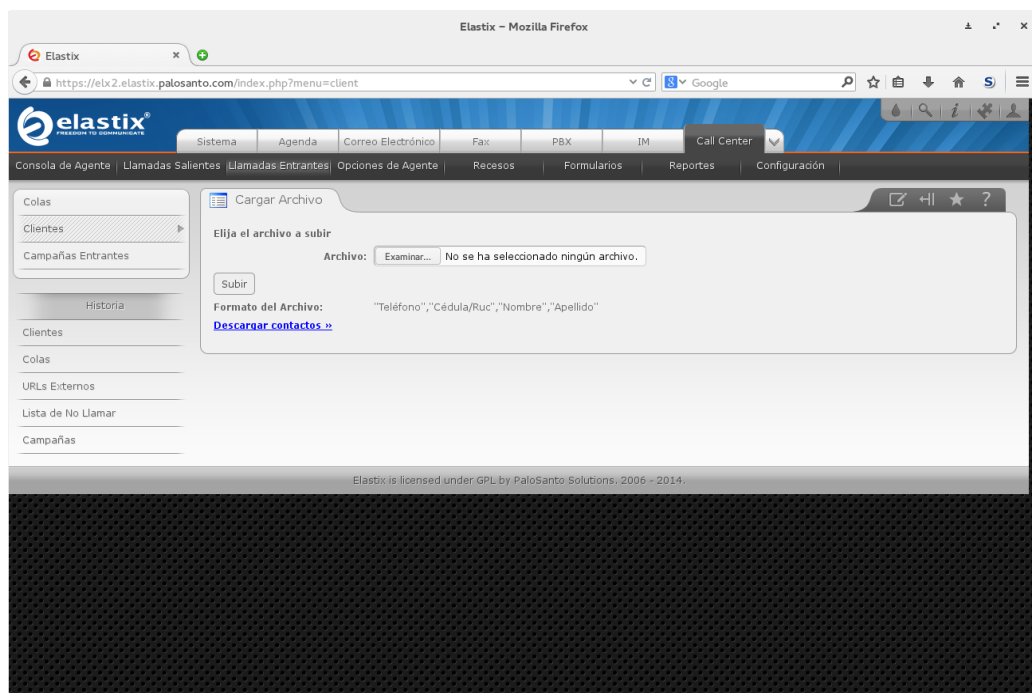


Figura 5.7: Interfaz de carga de contactos



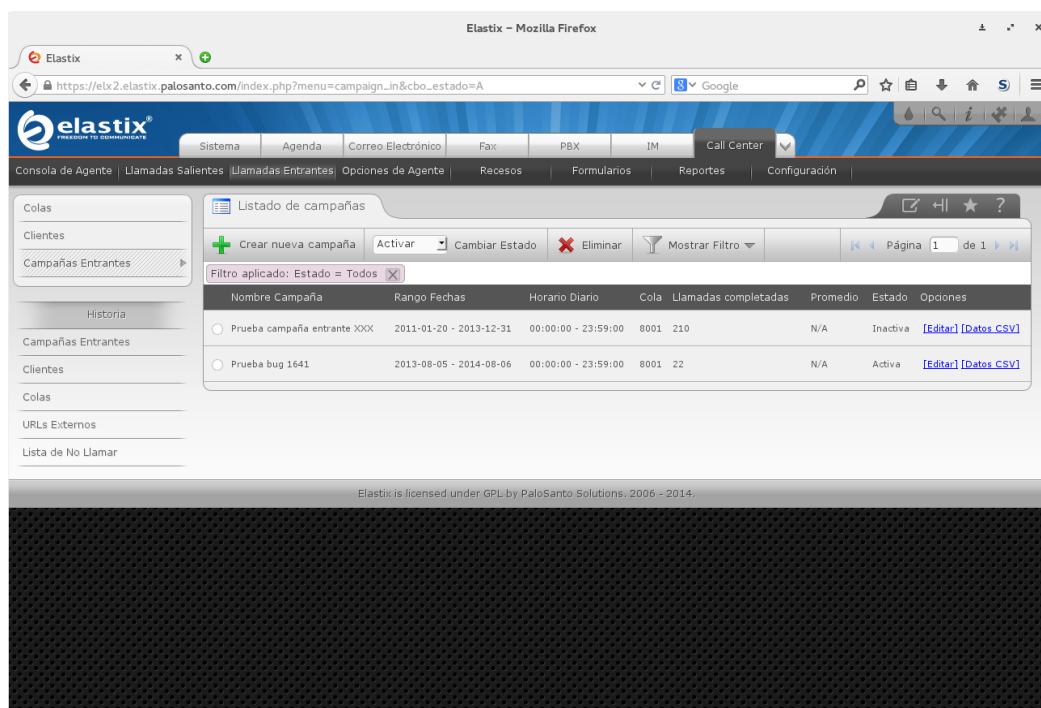


Figura 5.8: Lista de Campañas Entrantes

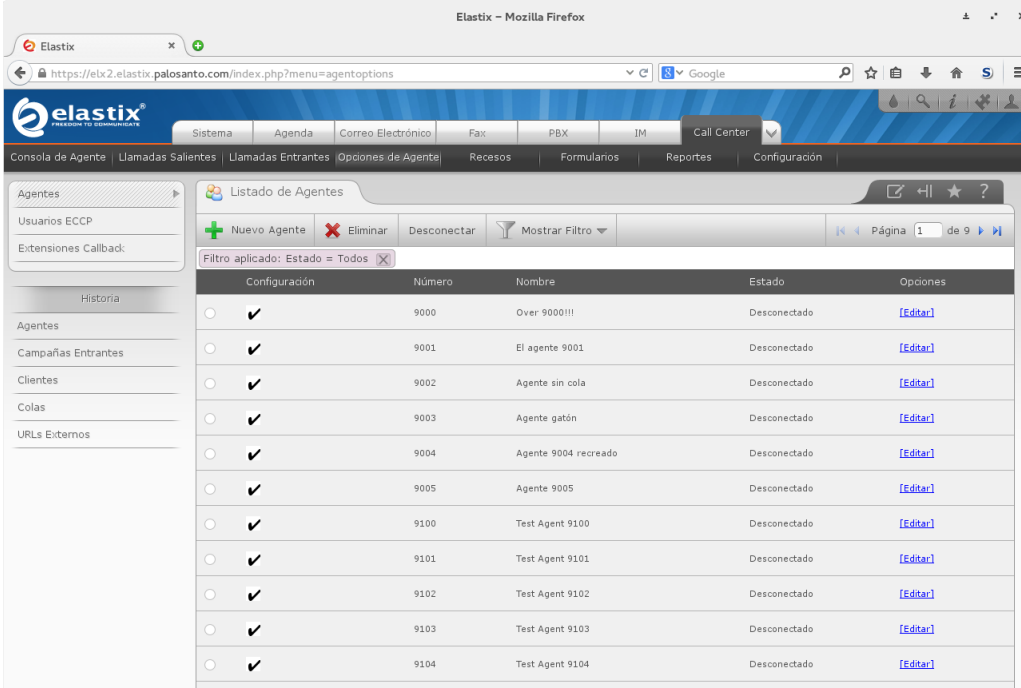
## Opciones de Agente

Esta sección contiene los siguientes módulos:

- **Agentes:** muestra los agentes estáticos que han sido configurados para el Call Center (véase la Figura 5.9). Se incluye la pantalla para definir nuevos agentes estáticos con su secuencia de autenticación telefónica (a digitar para AgentLogin) y la contraseña del protocolo ECCP.
- **Usuarios ECCP:** muestra la lista de credenciales ECCP para conexión al sistema (véase la Figura

5.10), la cual no debe de confundirse con la contraseña asociada a un agente específico. Una credencial para el usuario agentconsole debe de existir siempre, para uso de la nueva implementación de la Consola de Agente.

- Extensiones Callback: muestra los agentes dinámicos definidos para el Call Center, como en la Figura 5.11. Se incluye una pantalla para asignar la extensión del agente dinámico con sus contraseñas.



The screenshot shows the Elastix web interface in a Mozilla Firefox browser. The page title is 'Listado de Agentes'. The interface includes a navigation menu with options like 'Sistema', 'Agenda', 'Correo Electrónico', 'Fax', 'PBX', 'IM', and 'Call Center'. Below the navigation, there are tabs for 'Consola de Agente', 'Llamadas Salientes', 'Llamadas Entrantes', 'Opciones de Agente', 'Recesos', 'Formularios', 'Reportes', and 'Configuración'. The main content area shows a table of agents with the following data:

Configuración	Número	Nombre	Estado	Opciones
<input checked="" type="checkbox"/>	9000	Over 9000!!!	Desconectado	[Editar]
<input checked="" type="checkbox"/>	9001	El agente 9001	Desconectado	[Editar]
<input checked="" type="checkbox"/>	9002	Agente sin cola	Desconectado	[Editar]
<input checked="" type="checkbox"/>	9003	Agente gatón	Desconectado	[Editar]
<input checked="" type="checkbox"/>	9004	Agente 9004 recreado	Desconectado	[Editar]
<input checked="" type="checkbox"/>	9005	Agente 9005	Desconectado	[Editar]
<input checked="" type="checkbox"/>	9100	Test Agent 9100	Desconectado	[Editar]
<input checked="" type="checkbox"/>	9101	Test Agent 9101	Desconectado	[Editar]
<input checked="" type="checkbox"/>	9102	Test Agent 9102	Desconectado	[Editar]
<input checked="" type="checkbox"/>	9103	Test Agent 9103	Desconectado	[Editar]
<input checked="" type="checkbox"/>	9104	Test Agent 9104	Desconectado	[Editar]

Figura 5.9: Lista de Agentes Estáticos

Lista de usuarios ECCP

+ Nuevo usuario ECCP    ✖ Eliminar

Página 1 de 1

Nombre	Opciones
agentconsole	<a href="#">[Modificar]</a>
extremeVoxUser	<a href="#">[Modificar]</a>
monitor	<a href="#">[Modificar]</a>

Elastix is licensed under GPL by PaloSanto Solutions. 2006 - 2014.

Figura 5.10: Lista de Usuarios ECCP

Extensiones Callback

+ Nueva extensión callback    ✖ Eliminar    Desconectar    ⚙ Mostrar Filtro

Página 1 de 1

Filtro aplicado: Estado = Todos

Número	Nombre	Estado	Opciones
1064	Alex	Desconectado	<a href="#">[Editar]</a>
1071	Cuenta A	Desconectado	<a href="#">[Editar]</a>
7337	Prueba 7337	Desconectado	<a href="#">[Editar]</a>

Elastix is licensed under GPL by PaloSanto Solutions. 2006 - 2014.

Figura 5.11: Lista de Agentes Dinámicos

## **Recesos**

Este módulo lista los tipos de recesos que han sido autorizados para los agentes del sistema (véase la Figura 5.12). A través de los botones se pueden activar y desactivar los recesos, y crear nuevos tipos de recesos.

## **Formularios**

Esta sección contiene los siguientes módulos:

- Diseñador de formularios: muestra los formularios que han sido creados en el sistema para ser asociados con las campañas (véase Figura 5.13). Los formularios pueden ser activados, desactivados y eliminados. Para creación y modificación se dispone de una pantalla (véase Figura 5.14) donde se indican nombre y descripción del formulario, además de una lista de los campos definidos. Cada fila de un campo definido contiene la etiqueta del formulario, una lista desplegable para el tipo de formulario y un botón para quitar el campo en caso necesario. La última fila se usa para agregar nuevos campos. Las filas se pueden arrastrar para modificar el orden de presentación de los campos.

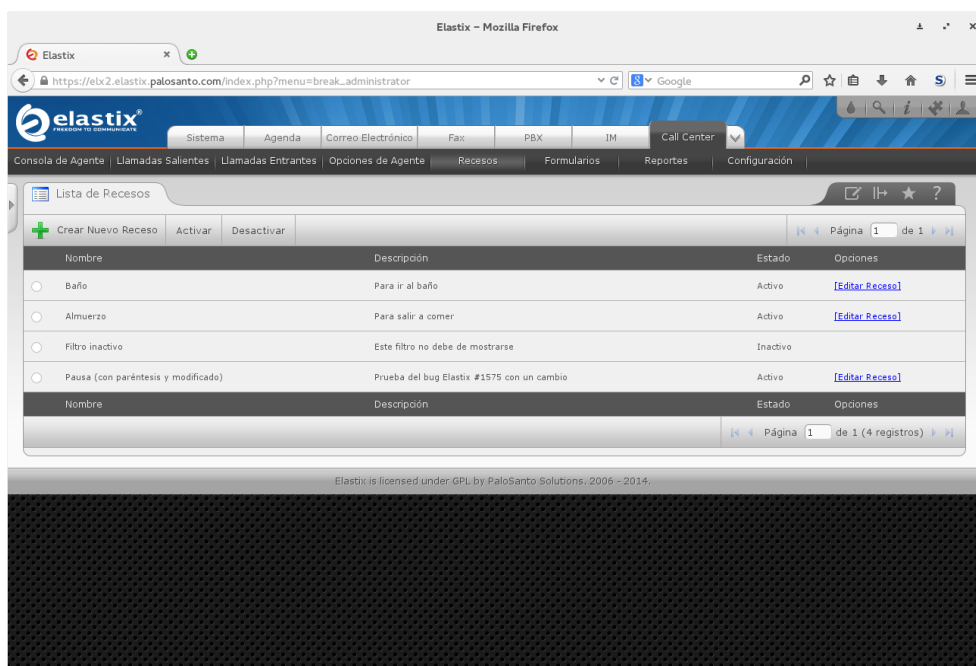


Figura 5.12: Lista de Tipos de Recesos

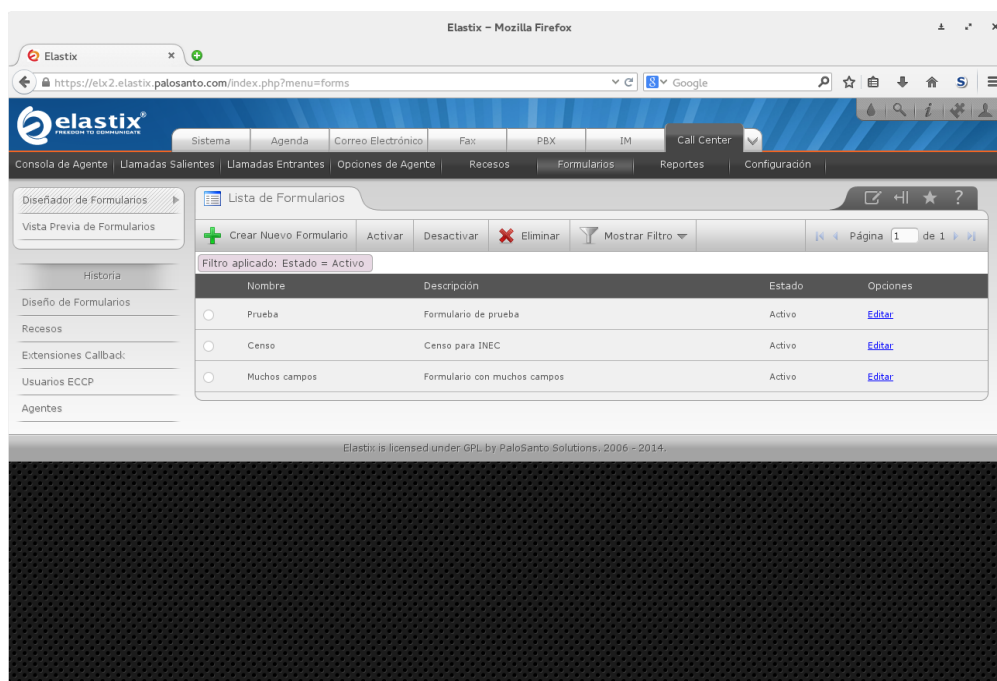


Figura 5.13: Lista de Formularios

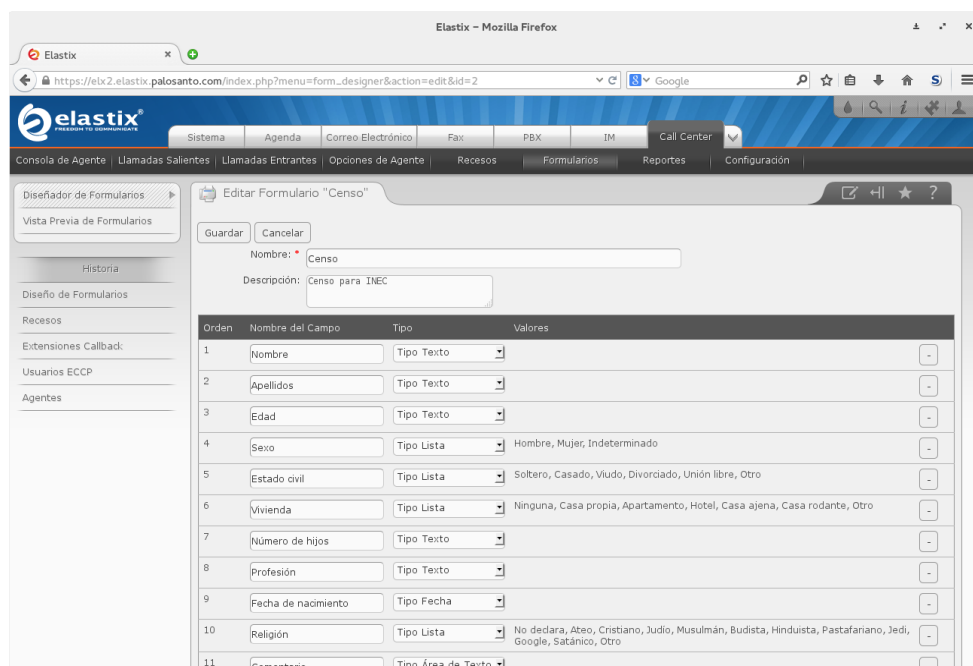


Figura 5.14: Editor de Formularios

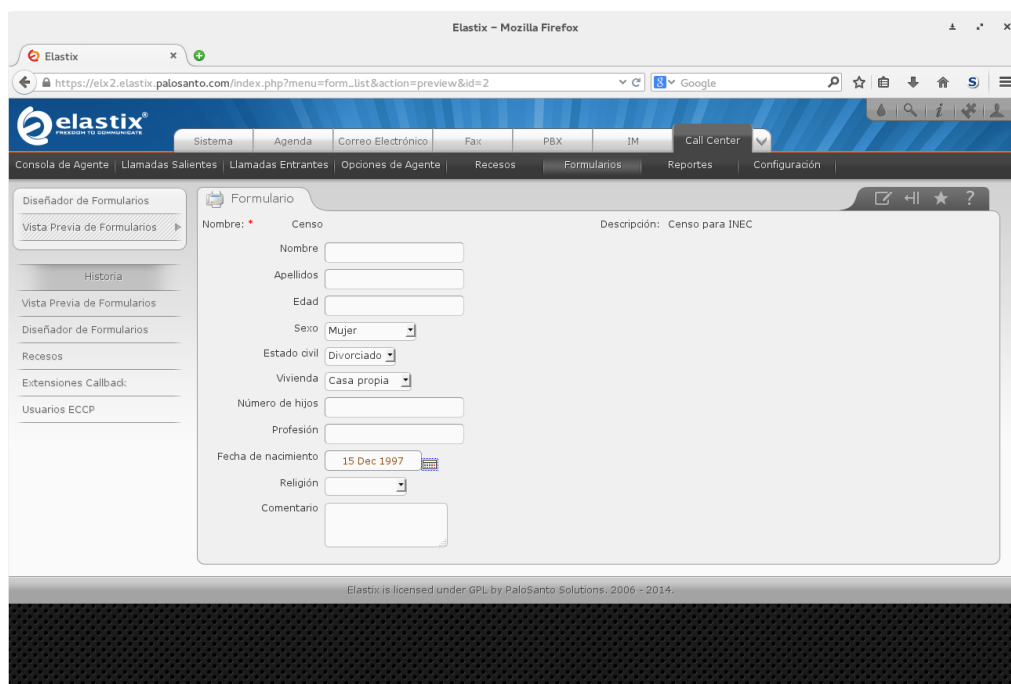


Figura 5.15: Vista Previa de Formulario

- Vista Previa de Formularios: lista los formularios creados y permite elegir cada uno para mostrar una vista del aspecto final del formulario, como en la figura 5.15.

## Reportes

Esta sección contiene los siguientes módulos:

- Reporte de Recesos: se listan los agentes activos, con el tiempo invertido en cada receso conocido, y un total por agente como última columna, para un intervalo de fechas indicado por el filtro (por omisión el día actual). La última fila contiene el tiempo invertido en recesos por todos los agentes. Un ejemplo se muestra en la Figura 5.16.

The screenshot shows the Elastix web interface in Mozilla Firefox. The browser address bar shows the URL: [https://elx2.elastix.palasantos.com/index.php?menu=reports\\_incoming\\_calls\\_fecha\\_init=16+Dec+2014&txt](https://elx2.elastix.palasantos.com/index.php?menu=reports_incoming_calls_fecha_init=16+Dec+2014&txt). The page title is 'Elastix - Mozilla Firefox'. The navigation menu includes: Sistema, Agenda, Correo Electrónico, Fax, PBX, IM, Call Center, Consola de Agente, Llamadas Salientes, Llamadas Entrantes, Opciones de Agente, Recesos, Formularios, Reportes, Configuración. The 'Reportes' menu is active, and the 'Reporte de Recesos' sub-menu is selected. The main content area shows a filter for 'Fecha Inicio' (16 Dec 2012) and 'Fecha Fin' (16 Dec 2014). Below the filter is a table with the following data:

No. de Agente	Nombre Agente	Hold	Baño	Almuerzo	Filtro inactivo	Pausa (con paréntesis y modificado)	Total
9000	Over 9000!!!	00:00:00	00:06:45	00:00:00	00:00:00	00:00:00	00:06:45
9001	El agente 9001	00:00:00	00:00:00	00:00:00	00:00:00	00:00:00	00:00:00
9002	Agente sin cola	00:00:00	00:00:00	00:00:00	00:00:00	00:00:00	00:00:00
9003	Agente gatón	00:00:00	00:00:00	00:00:00	00:00:00	00:00:00	00:00:00
9004	Agente 9004 resreado	00:00:00	00:00:00	00:00:00	00:00:00	00:00:00	00:00:00
1064	Alex	00:00:54	00:59:25	00:00:10	00:00:00	00:00:35	01:01:04
1071	Cuenta A	00:00:00	00:00:00	00:00:00	00:00:00	00:00:00	00:00:00
7337	Prueba 7337	00:00:00	00:00:00	00:00:00	00:00:00	00:00:00	00:00:00
9005	Agente 9005	00:00:00	00:00:00	00:00:00	00:00:00	00:00:00	00:00:00
9100	Test Agent 9100	00:00:00	00:00:00	00:00:00	00:00:00	00:00:00	00:00:00
9101	Test Agent 9101	00:00:00	00:00:00	00:00:00	00:00:00	00:00:00	00:00:00
9102	Test Agent 9102	00:00:00	00:00:00	00:00:00	00:00:00	00:00:00	00:00:00
9103	Test Agent 9103	00:00:00	00:00:00	00:00:00	00:00:00	00:00:00	00:00:00
9104	Test Agent 9104	00:00:00	00:00:00	00:00:00	00:00:00	00:00:00	00:00:00
9105	Test Agent 9105	00:00:00	00:00:00	00:00:00	00:00:00	00:00:00	00:00:00
9106	Test Agent 9106	00:00:00	00:00:00	00:00:00	00:00:00	00:00:00	00:00:00
9107	Test Agent 9107	00:00:00	00:00:00	00:00:00	00:00:00	00:00:00	00:00:00
9108	Test Agent 9108	00:00:00	00:00:00	00:00:00	00:00:00	00:00:00	00:00:00

Figura 5.16: Reporte de Recesos

- **Detalle de Llamadas:** se listan las llamadas colocadas durante un intervalo de tiempo, filtrado por tipo de llamada, prefijo de teléfono, número de agente y número de cola. Para cada llamada se muestran varios instantes de interés – fecha y hora de inicio de atención, fecha y hora de final de atención, duración, tiempo de espera en la cola y estado final de la llamada. Un ejemplo se muestra en la Figura 5.17.
- **Llamadas por hora:** se lista una cuenta total y por cola de llamadas durante un intervalo de hora del día, filtrado por estado y por cola. Un ejemplo se muestra en la Figura 5.18.
- **Llamadas por agente:** se lista, para cada combinación de agente y cola encontrados, el número de llamadas atendidas, el total de tiempo invertido en atención, el promedio de tiempo de atención y el tiempo más largo invertido en atender una llamada. Un ejemplo se muestra en la figura 5.19.



The screenshot shows the 'Detalle de Llamadas' page in the Elastix system. The left sidebar contains various reports and monitoring options. The main area features a filter section with the following settings:

- Fecha Inicio: 09 Feb 2011
- Fecha Fin: 16 Feb 2011
- Tipo: (Cualquier tipo)
- Teléfono: (empty)
- No. Agente: (Todos los agentes)
- Cola: (Todas las colas)

The data table below shows a list of call records with the following columns: No. Agente, Agente, Fecha Inicio, Hora Inicio, Fecha Fin, Hora Fin, Duración, Tiempo Espera, Cola, Tipo, Teléfono, Transferencia, and Estado.

No. Agente	Agente	Fecha Inicio	Hora Inicio	Fecha Fin	Hora Fin	Duración	Tiempo Espera	Cola	Tipo	Teléfono	Transferencia	Estado
9000	Over 9000!!!	2011-02-09	12:14:34	2011-02-09	12:17:39	00:03:05	00:00:01	8000	Saliente	1065		Success
9000	Over 9000!!!	2011-02-09	12:17:53	2011-02-09	12:21:27	00:03:34	00:00:00	8000	Saliente	1065		Success
9000	Over 9000!!!	2011-02-09	12:21:35	2011-02-09	12:33:58	00:12:23	00:00:01	8000	Saliente	1065		Success
9000	Over 9000!!!	2011-02-09	12:34:04	2011-02-09	12:35:02	00:00:58	00:00:01	8000	Saliente	1065		Success
9000	Over 9000!!!	2011-02-09	12:35:12	2011-02-09	12:35:49	00:00:37	00:00:00	8000	Saliente	1065		Success
9000	Over 9000!!!	2011-02-09	12:36:07	2011-02-09	12:36:25	00:00:18	00:00:00	8000	Saliente	1065		Success
9000	Over 9000!!!	2011-02-09	12:37:36	2011-02-09	12:38:49	00:01:13	00:00:01	8000	Saliente	1065		Success
9000	Over 9000!!!	2011-02-09	12:38:56	2011-02-09	12:39:14	00:00:18	00:00:00	8000	Saliente	1065		Success
9000	Over 9000!!!	2011-02-09	12:39:42	2011-02-09	12:39:52	00:00:10	00:00:00	8000	Saliente	1065		ShortCall
9000	Over 9000!!!	2011-02-09	13:39:31	2011-02-09	13:40:21	00:00:50	00:00:01	8000	Saliente	1065		Success
9000	Over 9000!!!	2011-02-11	17:37:49	2011-02-11	17:42:16	00:04:27	00:00:01	8000	Saliente	1065		Success

Figura 5.17: Detalle de Llamadas

The screenshot shows the 'Llamadas por hora' page in the Elastix system. The left sidebar contains various reports and monitoring options. The main area features a table showing the number of calls per hour for two different agents (6000 and 8001) and a 'Todas' (All) column.

Hora	6000	8001	Todas
00:00	0	0	0
01:00	0	0	0
02:00	0	0	0
03:00	0	0	0
04:00	0	0	0
05:00	0	0	0
06:00	0	0	0
07:00	0	0	0
08:00	0	0	0
09:00	1	6	7
10:00	1	13	14
11:00	1	24	25
12:00	0	37	37
13:00	2	14	16
14:00	0	15	15
15:00	0	32	32
16:00	0	65	65
17:00	0	57	57
18:00	2	20	22
19:00	0	0	0

Figura 5.18: Reporte de Llamadas por Hora

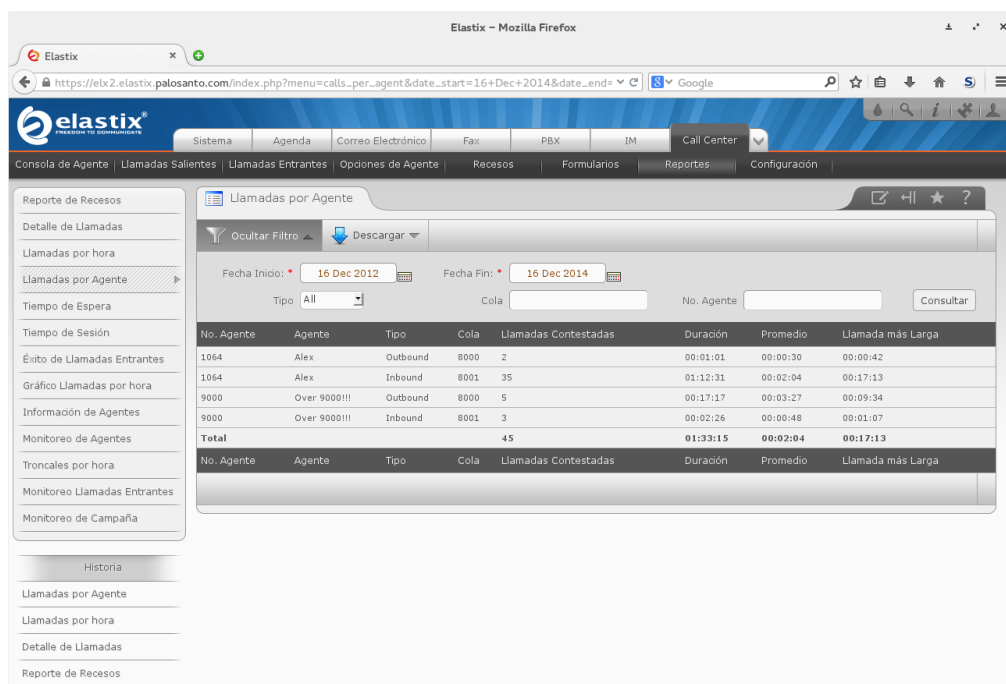


Figura 5.19: Reporte de Llamadas por Agente

- Tiempo de Espera: se lista, para cada cola, la clasificación por tiempo que la llamada estuvo en espera en la cola antes de ser atendida, en intervalos de 10 segundos. Se muestra además el tiempo total y promedio. Un ejemplo se muestra en la Figura 5.20.
- Tiempo de Sesión: se lista, para cada agente, el total de sesiones realizadas durante un intervalo de fechas especificado (por omisión el día actual). Se detalla el tiempo invertido en llamadas entrantes y salientes, y el porcentaje de servicio, definido como la fracción de

tiempo total de sesión invertido en atención de llamadas. Un ejemplo se muestra en la Figura 5.21

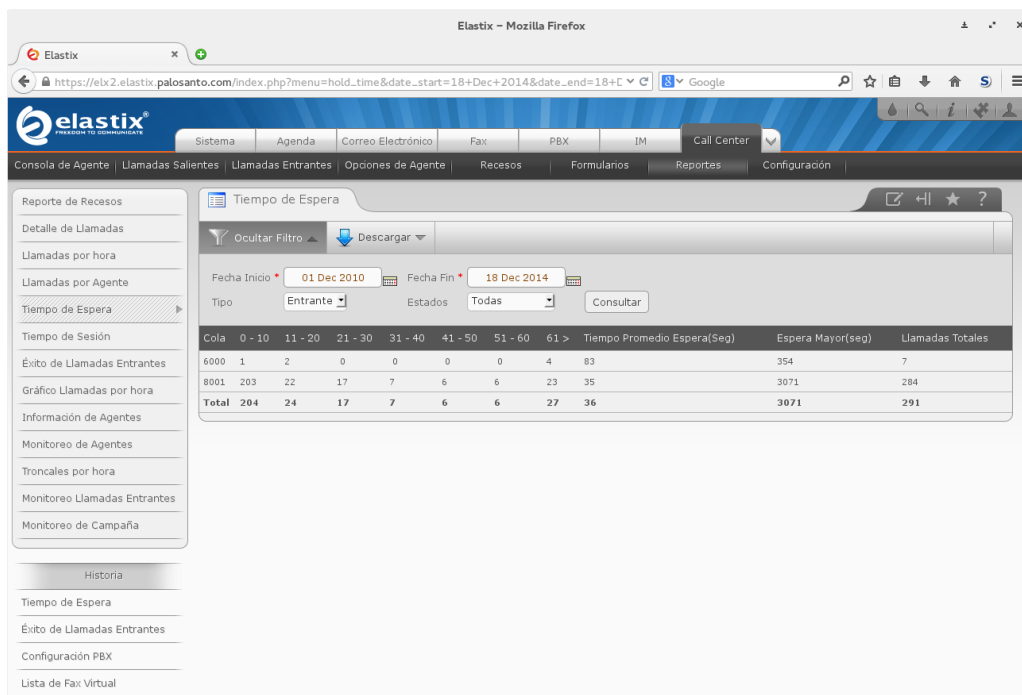


Figura 5.20: Reporte de Tiempo de Espera

- **Éxito de Llamadas Entrantes:** se lista, para cada cola, contadores de estados finales de llamadas entrantes. Un ejemplo se muestra en la Figura 5.22.
- **Gráfico Llamadas por hora:** se lista la misma información que el reporte de Llamadas por hora, y además un gráfico de líneas que muestra los intervalos horarios y líneas según el estado. Un ejemplo se muestra en la Figura 5.23.

Reporte de Recesos

Detalle de Llamadas

Llamadas por hora

Llamadas por Agente

Tiempo de Espera

**Tiempo de Sesión**

Éxito de Llamadas Entrantes

Gráfico Llamadas por hora

Información de Agentes

Monitoreo de Agentes

Troncales por hora

Monitoreo Llamadas Entrantes

Monitoreo de Campaña

Historia

Tiempo de Sesión

Tiempo de Espera

Llamadas por Agente

Llamadas por hora

**Tiempo de Sesión**

Ocultar Filtro Descargar

Página 1 de 1

Fecha Inicial: 23 Sep 2014 Fecha Final: 16 Dec 2014 Cola Entrante: Todas Tipo: Detallado Consultar

Agente	Nombre	Fecha Inicial	Fecha Final	Total Sesión	Llamadas entrantes	Llamadas salientes	Tiempo en Llamadas	Servicio(%)	Estado
1064	Alex	2014-09-24 13:31:35	2014-09-24 13:31:55	00:00:20	00:00:00	00:00:00	00:00:00	0.00	
1064	Alex	2014-09-25 17:35:59	2014-09-25 17:36:11	00:00:12	00:00:00	00:00:00	00:00:00	0.00	
1064	Alex	2014-09-25 17:37:10	2014-09-25 17:38:20	00:01:10	00:00:00	00:00:00	00:00:00	0.00	
1064	Alex	2014-09-26 16:12:48	2014-09-26 16:14:05	00:01:17	00:00:00	00:00:00	00:00:00	0.00	
9000	Over 9000!!!	2014-12-16 18:27:35	2014-12-16 18:29:19	00:01:44	00:00:00	00:00:00	00:00:00	0.00	
9460	Test Agent 9460	2014-09-29 15:33:02	2014-09-29 15:40:26	00:07:24	00:00:00	00:00:00	00:00:00	0.00	
<b>Total</b>				<b>00:12:07</b>	<b>00:00:00</b>	<b>00:00:00</b>	<b>00:00:00</b>		

Agente Nombre Fecha Inicial Fecha Final Total Sesión Llamadas entrantes Llamadas salientes Tiempo en Llamadas Servicio(%) Estado

Página 1 de 1 (6 registros)

Figura 5.21: Reporte de Tiempo de Sesión

Reporte de Recesos

Detalle de Llamadas

Llamadas por hora

Llamadas por Agente

Tiempo de Espera

Tiempo de Sesión

**Éxito de Llamadas Entrantes**

Gráfico Llamadas por hora

Información de Agentes

Monitoreo de Agentes

Troncales por hora

Monitoreo Llamadas Entrantes

Monitoreo de Campaña

Historia

Éxito de Llamadas Entrantes

Tiempo de Sesión

Tiempo de Espera

Llamadas por Agente

**Éxito de Llamadas Entrantes**

Ocultar Filtro Descargar

Página 1 de 1

Fecha Inicio: 01 Dec 2013 Fecha Fin: 16 Dec 2014 Buscar

Cola	Exitosas	Abandonadas	Tiempo de espera	Llamadas Totales
8001	0	7	00:12:47	7
6000	0	0	00:00:00	0
<b>Total</b>	<b>0</b>	<b>7</b>	<b>00:12:47</b>	<b>7</b>

Cola Exitosas Abandonadas Tiempo de espera Llamadas Totales

Figura 5.22: Reporte de Éxito de Llamadas Entrantes

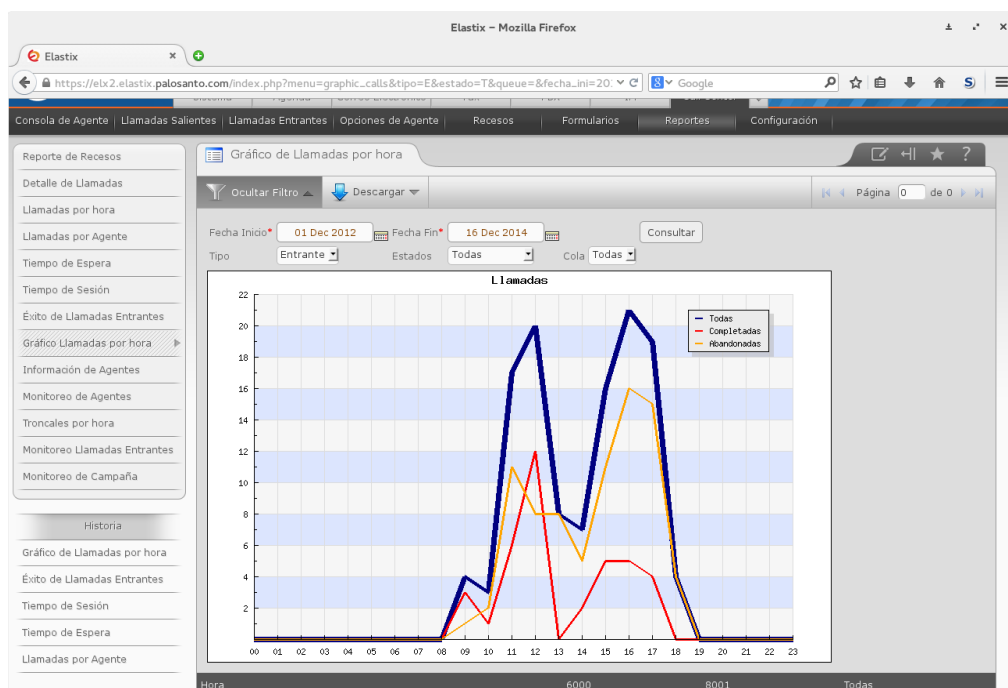


Figura 5.23: Gráfico de Llamadas por hora

- **Información de Agentes:** se muestra información detallada sobre la atención de un solo agente durante un intervalo de fechas indicado (por omisión el día actual). Se permite especificar una sola cola entrante, y se detalla inicio, final y duración de la sesión, la cuenta de llamadas entrantes, número de llamadas por hora, duración de atención por llamada, y un reporte de duración de pausas. Un ejemplo se muestra en la Figura 5.24.
- **Monitoreo de Agentes:** se muestran las colas monitoreadas para llamadas, y los agentes asignados

a cada una de las colas. Para cada combinación de agente y cola, se muestra el estado del agente, el total de llamadas atendidas a través de la cola, el tiempo total de sesión y de atención a llamadas. Este reporte se actualiza en tiempo real a través del protocolo ECCP. Un ejemplo se muestra en la Figura 5.25.

- Troncales por hora: se muestra el uso de las troncales definidas para las llamadas atendidas, clasificado por intervalo horario. Un ejemplo se muestra en la Figura 5.26.

The screenshot shows the Elastix web interface in Mozilla Firefox. The main content area displays the 'Información del Agente' report for agent Alex. The report includes the following data:

**Fecha Inido:** 01 Sep 2013  
**Fecha Fin:** 16 Dec 2014  
**Cola:** 8001  
**No. Agente:** 1064

**NOMBRE AGENTE:** Alex

**DATOS SOBRE LA CONEXION**

Primera Conexión	2013-09-09 12:32:51
Última Desconexión	2014-09-26 16:14:05
Tiempo de Conexión	05:04:52
Conteo de Conexiones	26

**LLAMADAS ENTRANTES**

Conteo llamadas entrantes	7 Llamada(s) (7 Monitoreadas, 0 No Monitoreadas)
Llamadas/h	1.38
Tiempo Llamadas entrantes	00:24:15
Media Llamadas entrantes	00:03:28 (Sólo monitoreadas)

**MOTIVO NO-DISPONIBILIDAD**

Receso	Conteo	Hora	Porcentaje de no-disponibilidad
Baño	7	00:21:21	98.46 %
Pausa (con paréntesis y modificado)	1	00:00:20	1.54 %

Figura 5.24: Reporte de Información de Agente

Reporte de Recesos

Detalle de Llamadas

Llamadas por hora

Llamadas por Agente

Tiempo de Espera

Tiempo de Sesión

Éxito de Llamadas Entrantes

Gráfico Llamadas por hora

Información de Agentes

Monitoreo de Agentes

Troncales por hora

Monitoreo Llamadas Entrantes

Monitoreo de Campaña

Historia

Monitoreo de Agentes

Información de Agentes

Gráfico Llamadas por hora

Éxito de Llamadas Entrantes

Cola	Número	Agente	Estado actual	Total de llamadas	Tiempo total de sesión	Tiempo total de llamadas
8000	Agent/9000	Over 9000!!!	LOGOUT 00:33:22	0	00:01:44	00:00:00
	Agent/9001	El agente 9001	LOGOUT	0	00:00:00	00:00:00
	Agent/9003	Agente gatón	LOGOUT	0	00:00:00	00:00:00
	SIP/1064	Alex	LOGOUT	0	00:00:00	00:00:00
SIP/1071	Cuenta A	LOGOUT	0	00:00:00	00:00:00	
<b>TOTAL</b>	<b>Agentes: 5</b>			<b>0</b>	<b>00:01:44</b>	<b>00:00:00</b>
8001	Agent/9000	Over 9000!!!	LOGOUT 00:33:22	0	00:01:44	00:00:00
	Agent/9003	Agente gatón	LOGOUT	0	00:00:00	00:00:00
	SIP/1064	Alex	LOGOUT	0	00:00:00	00:00:00
	SIP/1071	Cuenta A	LOGOUT	0	00:00:00	00:00:00
<b>TOTAL</b>	<b>Agentes: 4</b>			<b>0</b>	<b>00:01:44</b>	<b>00:00:00</b>
8010	Agent/9100	Test Agent 9100	LOGOUT	0	00:00:00	00:00:00
	Agent/9101	Test Agent 9101	LOGOUT	0	00:00:00	00:00:00
	Agent/9102	Test Agent 9102	LOGOUT	0	00:00:00	00:00:00
	Agent/9103	Test Agent 9103	LOGOUT	0	00:00:00	00:00:00
	Agent/9104	Test Agent 9104	LOGOUT	0	00:00:00	00:00:00
	Agent/9105	Test Agent 9105	LOGOUT	0	00:00:00	00:00:00
	Agent/9106	Test Agent 9106	LOGOUT	0	00:00:00	00:00:00
	Agent/9107	Test Agent 9107	LOGOUT	0	00:00:00	00:00:00
	Agent/9108	Test Agent 9108	LOGOUT	0	00:00:00	00:00:00

Figura 5.25: Monitoreo de Agentes

Reporte de Recesos

Detalle de Llamadas

Llamadas por hora

Llamadas por Agente

Tiempo de Espera

Tiempo de Sesión

Éxito de Llamadas Entrantes

Gráfico Llamadas por hora

Información de Agentes

Monitoreo de Agentes

Troncales por hora

Monitoreo Llamadas Entrantes

Monitoreo de Campaña

Historia

Troncales por hora

Reporte de Recesos

Consola de Agente

Reporte de Troncales usadas por Hora en el día

Ocultar Filtro

Descargar

Página 1 de 1

Fecha Inicio: 01 Jan 2008

Fecha Fin: 08 Jan 2015

Troncal: (All) Mostrar

Periodo de Tiempo	Entradas	Respondidas	Abandonadas	En cola	Sin monitorear
09:00:00 - 10:00:00	7	6	1	0	0
10:00:00 - 11:00:00	14	12	2	0	0
11:00:00 - 12:00:00	25	14	11	0	0
12:00:00 - 13:00:00	37	29	8	0	0
13:00:00 - 14:00:00	16	8	8	0	0
14:00:00 - 15:00:00	15	9	6	0	0
15:00:00 - 16:00:00	32	21	11	0	0
16:00:00 - 17:00:00	67	51	16	0	0
17:00:00 - 18:00:00	62	42	20	0	0
18:00:00 - 19:00:00	23	19	4	0	0
19:00:00 - 20:00:00	2	2	0	0	0
<b>TOTAL</b>	<b>300</b>	<b>213</b>	<b>87</b>	<b>0</b>	<b>0</b>

Periodo de Tiempo Entradas Respondidas Abandonadas En cola Sin monitorear

Página 1 de 1 (11 registros)

Figura 5.26: Troncales por hora

- **Monitoreo Llamadas Entrantes:** se listan las colas de llamadas entrantes, con contadores de llamadas en todos los estados conocidos, para el día actual. Este reporte se actualiza en tiempo real a través del protocolo ECCP. Un ejemplo se muestra en la Figura 5.27.
- **Monitoreo de Campaña:** se muestra, para una campaña en particular, información actualizada en tiempo real en varios paneles. En el panel izquierdo se muestran las llamadas generadas por el demonio marcador, con el Caller ID y su instante de marcación. En el panel derecho se muestran los agentes de la cola configurada, con su estado, el número de teléfono de su llamada asignada, y el instante de inicio de atención. El panel inferior, que inicialmente está oculto, muestra una bitácora de las actividades del marcador. Un ejemplo se muestra en la Figura 5.28.

### **Configuración**

Este módulo permite configurar varios parámetros que afectan a todo el sistema Call Center, como se muestra en la Figura 5.29. Los parámetros configurables incluyen el



servidor Asterisk a usar (normalmente localhost), el umbral de llamada corta, el porcentaje de servicio y la espera máxima de inactividad del agente.

Cola	Entrantes	Respondidas	Abandonadas	Llamadas en espera	Sin monitoreo
8000	0	0	0	0	0
8001	1	0	1	0	0
11627	0	0	0	0	0
<b>TOTAL</b>	<b>1</b>	<b>0</b>	<b>1</b>	<b>0</b>	<b>0</b>

Figura 5.27: Monitoreo de Llamadas Entrantes

**Campaña:** (Campaña entrante) Prueba bug 1641

Fecha inicio: 2013-08-05      Fecha final: 2014-08-06      Horario: 00:00:00 - 23:59:00

Cola: 8001      Intentos: N/D

Total llamadas: 22      Llamadas en cola: 0      Llamadas conectadas: 0

Llamadas sin pista: 0      Llamadas abandonadas: 17      Llamadas terminadas: 5

Promedio Duración de Llamada: 00:03:51      Duración Máxima de Llamada: 00:17:13

Estado	Núm. Telf.	Troncal	Desde	Agente	Estado	Núm. Telf.	Troncal	Desde
				Agent/9000	No logon	-	-	-
				Agent/9003	No logon	-	-	-
				SIP/1064	No logon	-	-	-
				SIP/1071	No logon	-	-	-

Ver registro de campaña:

Figura 5.28: Monitoreo de Campañas

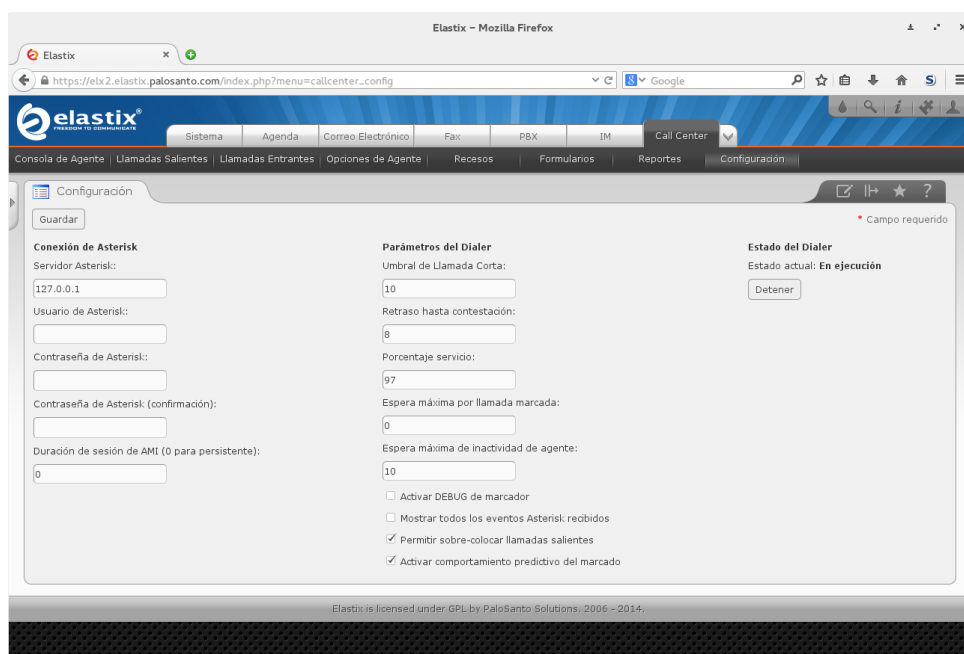


Figura 5.29: Módulo de Configuración

### 5.1.3 La Base de Datos

Para el almacenamiento de los datos del sistema Call Center, se usa el motor de base de datos MySQL. El esquema usado por la implementación se creó bajo el nombre “call\_center”. A continuación se detallan las tablas que fueron creadas.

- Tabla “agent”: esta tabla representa a los agentes que están registrados en el sistema. Cada uno de los agentes tiene asociado un número de agente (que en el caso de los agentes dinámicos, corresponde al número de extensión), un tipo de agente (estático o

dinámico), una etiqueta descriptiva del agente, una contraseña (el PIN a introducir para agentes estáticos), un estado de activo o inactivo y una contraseña para la interacción ECCP.

- Tabla “audit”: esta tabla registra los inicios y finales de sesión para cada agente que utiliza el sistema, así como los inicios y finales de los descansos que cada uno de los agentes toma. Esta tabla hace referencia al agente que inicia la sesión.
- Tabla “break”: en esta tabla se registran los tipos de descansos que han sido configurados en el sistema. Cada descanso tiene un nombre, una descripción y un estado de activo o inactivo.
- Tabla “campaign”: esta tabla registra las campañas salientes creadas en el sistema. Para cada campaña, se registra la siguiente información:
  - Etiqueta descriptiva de la campaña.
  - Fechas del intervalo durante el cual la campaña va a ejecutarse. La especificación del intervalo permite ejecutar campañas promocionales por un intervalo limitado, así como la preparación

por adelantado de campañas que inician en una fecha futura.

- El horario en el cual se generan las llamadas. Esto permite que la campaña genere llamadas sólo en horas laborables, o sólo fuera de ellas, como requiera el administrador.
- Número de intentos máximos para cada llamada generada.
- Troncal a través de la cual se requiere que se generen las llamadas salientes. Si no se especifica, se usa el plan de marcado de FreePBX para generar las llamadas salientes.
- Contexto y número de cola al cual hay que dirigir las llamadas salientes para que sean atendidas por agentes. Casi siempre, el contexto es “from-internal”, el contexto usado para el marcado de extensiones internas de FreePBX.
- Máximo número de llamadas simultáneas a generar para esta campaña. Este límite existe para poder acomodar la generación de llamadas salientes a través de tecnologías

telefónicas que tienen un límite fijo de llamadas simultáneas. El ejemplo más prominente es ISDN (23 llamadas para T1 o 30 llamadas para E1). Para tecnologías telefónicas sin un límite establecido, es suficiente con asignar un valor mucho mayor que el número de agentes.

- Número de llamadas completadas y promedio de la duración de la llamada. Estos valores se usan para asistir en la predicción de cuelgue de una llamada atendida.
- Texto del guión que debe usarse para cada llamada atendida. Este texto, que generalmente es un fragmento de código HTML, se muestra en la consola de agente cuando la llamada es asociada al agente.
- Estado de campaña activa, inactiva, o terminada. Este último estado se alcanza cuando se han intentado todos los números de la campaña, y todos han tenido éxito, o alcanzado el número máximo de reintentos.

- Referencia a un URL opcional de una página web que debe abrirse cuando se atiende la llamada.
- Tabla “calls”: esta es la tabla en la cual se registran los números a marcar de las llamadas salientes para cada campaña, así como el resultado de haber marcado el número. Para cada número marcado, se almacena la siguiente información:
  - Fecha y hora de varios eventos recibidos para la llamada. Los instantes de eventos registrados son: inicio de marcado (Originate), recepción de resultado del marcado (OriginateResponse), entrada de la llamada a la cola (Join), inicio de atención por parte del agente (Link/Bridge), final de atención por parte del agente (Unlink/Hangup). Al examinar la fecha y hora de varios eventos, es posible analizar la calidad del servicio. Por ejemplo, el tiempo de espera en la cola se muestra en el reporte Hold Time.
  - Estado transitorio o final de la llamada. El estado inicial es el estado vacío,

correspondiente a un número que no se ha marcado. Al ser realizada la llamada, este campo se actualiza con estados sucesivos de timbrado, entrada en cola y éxito de llamada. Si la llamada falla en alguno de los pasos, se registra un estado de fallo.

- Valor de Uniqueid de Asterisk. Este valor corresponde a un identificador asignado por Asterisk a la llamada realizada. Este identificador puede usarse para localizar el CDR correspondiente a la llamada realizada, en el reporte de Elastix.
- Cantidad de reintentos. Este es el número de veces que se ha intentado marcar a este número luego de que el intento anterior ha fallado. La marcación se considera fallida al alcanzar el número de reintentos máximo indicado por la campaña.
- Duración de la atención de la llamada por el agente, en segundos.
- Duración del intervalo entre el ingreso a la cola y el inicio de la atención, en segundos.

- Bandera de DNC (Do Not Call). Si, al momento de intentar el primer marcado, la llamada fue encontrada en la tabla de no llamar, se activa esta bandera y se ignora la llamada.
  - Fechas y horas de intervalo de agendamiento, en caso de reagendarse la llamada.
  - Agente que debe de atender la llamada reagendada.
  - Código y descripción de la causa de fallo del último intento de marcado.
  - Troncal que fue usada para enviar la llamada saliente.
- Tabla “call\_attribute”: esta tabla se usa para almacenar los atributos asociados a cada contacto que se usará en una llamada saliente. Estos atributos fueron especificados en el archivo CSV que se usó para alimentar la campaña. Se incluye la etiqueta usada para el atributo, el valor y el número de orden de columna que se usó en el archivo CSV.
  - Tabla “current\_calls”: esta es una tabla auxiliar que fue usada para implementar la primera versión de la actualización en tiempo real de la Consola de Agente.



Actualmente la Consola de Agente no usa esta tabla, pero se conserva debido a que existen herramientas desarrolladas por terceros que consultan esta tabla.

- Tabla “dont\_call”: en esta tabla se almacena la lista de teléfonos que no deben de marcarse, para poder cumplir con leyes nacionales que obligan a los sistemas de automarcado a obedecer una lista de teléfonos excluida de telemarketing.
- Tabla “form”: esta es la tabla que define los formularios disponibles para las campañas. Esta tabla define el nombre del formulario, la descripción larga y el estado de activo o inactivo.
- Tabla “form\_field”: esta tabla representa los campos que pertenecen a cada uno de los formularios definidos. Para cada campo de un formulario, se define la etiqueta o nombre, el tipo de campo, una lista de valores admitidos (para el tipo de lista de selección) y el orden en el cual el campo aparece en el formulario.
- Tabla “form\_data\_recollected”: esta tabla almacena, para cada llamada atendida y para cada campo del formulario asociado a la campaña, el valor de la

información que fue proporcionada por el cliente contactado y llenada por el agente.

- Tabla “campaign\_form”: tabla de asociación entre un formulario y una campaña.
- Tabla “campaign\_external\_url”: en esta tabla se almacenan los URLs de páginas web externas que pueden ser asociados con campañas, para que sean abiertos cuando una llamada resulta conectada al agente. Se define la plantilla del URL a abrir, una descripción del URL, una bandera de activación y la manera en la que la plantilla puede abrirse (nueva ventana, embebido, o como jsonp).
- Tabla “queue\_call\_entry”: en esta tabla se listan todas las colas que se han definido, que han sido reservadas para las campañas entrantes. Toda cola que no se mencione en esta tabla puede usarse para campañas salientes. En esta tabla se registran el número de la cola, el estatus de activo o inactivo y un script asociado a la cola.
- Tabla “contact”: esta tabla se usa para almacenar información sobre un cliente potencial que llama al sistema, a partir del número de CallerID

proporcionado con la llamada. Si se encuentra la información del contacto, se la muestra cuando el agente recibe la información de la llamada. Se almacena el número a asociar y el nombre y apellidos del contacto.

- Tabla “call\_entry”: esta tabla es análoga a la tabla “calls”, pero registra la información de las llamadas entrantes, que no son generadas por el propio programa marcador. Se registra la siguiente información:
  - Referencia al agente que manejó la llamada.
  - Referencia a la cola a través de la cual entró la llamada.
  - Referencia a un registro de contacto, si tal contacto fue identificado.
  - Número de Caller ID reportado por la fuente de llamadas a Asterisk.
  - Fecha y hora de entrada a la cola, de inicio de la llamada y de final de la llamada.
  - Estado transitorio o final de la llamada. Debido a que la llamada ya existe al entrar en la cola, no existe estado de fallo de marcación.

- Duración del intervalo entre entrada a la cola e inicio de la atención, en segundos.
  - Valor de Uniqueid de Asterisk. Al igual que para la llamada saliente, este identificador puede usarse para localizar el CDR de la llamada una vez concluida.
  - Referencia a la campaña que coordina esta llamada entrante, si existe.
  - Troncal que recibió la llamada.
- Tabla “campaign\_entry”: en esta tabla se registran las campañas de llamadas entrantes. Una campaña entrante permite asociar, por un intervalo definido, uno o más formularios de atención que serán usados para atender las llamadas entrantes. Se registra la referencia a la cola de llamadas entrantes, las fechas de inicio y final de intervalo de la campaña, las horas de inicio y final de la atención durante el día, el estado de campaña activa o inactiva, el texto del guión que debe usarse para cada llamada atendida y una referencia a un URL de una página web que debe abrirse cuando se atiende la llamada.

- Tabla “current\_call\_entry”: tabla auxiliar que fue introducida para implementar la primera versión de la actualización en tiempo real de la Consola de Agente. Al igual que con la tabla “current\_calls”, en la actualidad se conserva para beneficio de herramientas de terceros que usan esta tabla.
- Tabla “campaign\_form\_entry”: tabla de asociación entre una campaña entrante y un formulario.
- Tabla “form\_data\_recollected\_entry”: esta tabla almacena, para cada llamada entrante atendida, y para cada campo del formulario asociado a la campaña, el valor de la información que fue proporcionada por el cliente contactado y llenada por el agente. Esta tabla es análoga a la tabla “form\_data\_recollected”.
- Tabla “eccp\_authorized\_clients”: en esta tabla se almacena información de autenticación para usuarios del protocolo ECCP. Se almacena el usuario autorizado y un hash de la contraseña asignada.
- Tabla “call\_progress\_log”: esta tabla almacena información adicional sobre el progreso de las operaciones sobre cada llamada de una campaña. Se

registra, para cada llamada realizada, la campaña correspondiente, el instante del cambio de estado de la llamada, el nuevo estado de la llamada, el número de reintento, el agente asignado, la troncal usada y la duración de la llamada (de estar disponible). Esta tabla es de principal interés para el reporte de Monitoreo de Campaña.

- Tabla “valor\_config”: esta tabla almacena valores de configuración que pueden modificarse a través del módulo de configuración (callcenter\_config).

El diagrama Entidad/Relación correspondiente a la base de datos se muestra en el anexo 3.

#### **5.1.4 El Demonio**

El componente más importante para el funcionamiento del sistema Call Center es el demonio de marcación. Este programa hace uso de los siguientes componentes de Asterisk para la generación de llamadas: la aplicación Queue y la aplicación AgentLogin, el protocolo AMI y los eventos de progreso de llamada.

### **La aplicación Queue()**

En el programa Asterisk, una llamada que sigue un plan de marcado puede ser conducida hacia diversos módulos, llamados aplicaciones, los cuales fueron expuestos de manera general en la sección 3.5. Una de las aplicaciones es la aplicación Queue(), la cual se invoca de la siguiente manera en el plan de marcado (tomando como ejemplo una cola llamada 8000):

```
exten => EXTENSION,PRIORIDAD,Queue(8000, [parámetros])
```

Cuando una llamada conectada en el plan de marcado ejecuta la aplicación Queue(), la llamada entra en una cola de espera. Asociado a esta cola de espera pueden existir cero o más miembros de cola - un canal telefónico con su tecnología asociada - como por ejemplo "SIP/agenteventas", "IAX2/recepcion", o (esto es importante), "Agent/9000". La llamada que entra en la aplicación Queue() será asignada a uno de los agentes asociados, en caso de que el agente no esté ocupado atendiendo una llamada, y si el canal es alcanzable. En caso de que todos los agentes disponibles estén ocupados, la llamada entrante quedará en espera y

escucha música de fondo hasta que alguno de los agentes se libere y la llamada pueda ser asignada. Al ser asignada la llamada a un agente, el canal correspondiente será timbrado si es necesario, o conectada directamente si es posible (para tecnología Agent).

Además del comportamiento indicado, la aplicación Queue() emite los siguientes eventos de interés a través del protocolo AMI (discutido más adelante):

- Join: una llamada ha ingresado a la cola, antes de decidir si se puede asignar un agente.
- QueueMemberAdded: un canal agente ha sido agregado como miembro de una cola
- QueueMemberRemoved: un canal agente ha sido quitado como miembro de una cola

El protocolo AMI expone los comandos QueueAdd y QueueRemove, los cuales agregan o quitan, respectivamente, un canal telefónico como miembro de una cola. Estos comandos son de interés principalmente para implementar los agentes dinámicos. Además se expone el comando QueuePause, el cual permite “pausar” un agente.



Un agente pausado no recibe llamadas aunque el canal correspondiente está disponible. El comando `QueuePause` es la base de implementación de los descansos (breaks) en el CallCenter.

La configuración de las colas disponibles para la aplicación `Queue()` se realiza en el archivo `/etc/asterisk/queues.conf` . Un ejemplo típico de una sección de configuración de cola (escrita por FreePBX) se muestra en el anexo 4.

La configuración de una cola puede tener dos tipos de agentes: estáticos y dinámicos. Un agente estático está presente de forma permanente como una entrada `member=xxx` en la configuración de la cola (en el ejemplo anterior, los canales `Agent/9000` y `Agent/9001` son miembros estáticos). Un agente estático no puede ser removido a través del comando `QueueRemove`, y se lo considera disponible si el canal telefónico correspondiente es alcanzable. Para la configuración del Call Center, todos los agentes estáticos (al inicio, los únicos disponibles) se configuran a través de la tecnología Agent. La razón de esta decisión es que un agente conectado a través de la

tecnología Agent está con el teléfono ya abierto, y por lo tanto recibirá la llamada asignada inmediatamente, sin opción a ignorarla o rechazarla, por requerimiento expreso del cliente inicial.

Un agente dinámico no aparece como miembro de una cola en la configuración de `queues.conf`, aunque todavía se lo debe declarar en la configuración de la cola en FreePBX. El control de la disponibilidad del agente dinámico se la realiza a través de los comandos `QueueAdd` y `QueueRemove` de AMI. Para el funcionamiento del Call Center, los agentes dinámicos son canales VoIP ordinarios que corresponden a extensiones internas, como `SIP/12345` o `IAX2/99887`. Esto permite que el agente mantenga cerrado el teléfono la mayor parte del tiempo, y lo levante al recibir una llamada asignada de la cola.

### **La aplicación AgentLogin()**

Luego de la aplicación `Queue()`, la otra aplicación de importancia para el funcionamiento del demonio de Call Center es la aplicación `AgentLogin()`. La aplicación `AgentLogin()` es el mecanismo estándar para hacer uso de

una definición de agente estático. Como toda aplicación de plan de marcado de Asterisk, `AgentLogin()` puede invocarse de la siguiente manera en un plan de marcado:

```
exten => EXTENSION,PRIORIDAD,AgentLogin(9000)
```

Esta invocación causa que la extensión usada para la llamada sea conectada con el agente `Agent/9000`. En todos los lugares que hagan referencia a `Agent/9000`, este canal es ahora alcanzable y se conectará con la llamada en curso que ejecuta la aplicación `AgentLogin()`. Esta llamada en curso escucha música en espera hasta que una llamada de otra parte (por ejemplo, de una cola) se conecte, en cuyo caso se establece la comunicación entre las dos llamadas. Si la llamada conectada es luego terminada, la primera llamada (la que ejecutó `AgentLogin()` en el plan de marcado) vuelve al estado de música en espera. Las maneras de terminar la ejecución de `AgentLogin()` son terminar la llamada, o con el comando `Agentlogoff` de AMI.

Las primeras versiones del demonio de Call Center instalaban contextos que llamaban a la aplicación

AgentLogin() en el plan de marcado. Pero las versiones actuales aprovechan la capacidad del comando Originate de AMI de conectar una llamada directamente con una aplicación, y por lo tanto invocan la aplicación AgentLogin() explícitamente sin pasar por un contexto del plan de marcado. Esta manera de invocar la aplicación permite reducir el número de cambios requeridos en el plan de marcado de Asterisk, y encapsula el uso especializado de AgentLogin() dentro del código del demonio de marcación.

La aplicación AgentLogin() emite los siguientes eventos de interés a través del protocolo AMI (discutido más adelante):

- Agentlogin: una llamada ha ejecutado la aplicación AgentLogin() y se ha autenticado correctamente con el número PIN asociado al agente estático.
- Agentlogoff: una llamada previamente conectada con AgentLogin() ha terminado esta conexión, sea porque se cierra la llamada, o por el comando Agentlogoff de AMI.

### **Comandos de reporte “queue show” y “agent show”**

Una de las tareas básicas del demonio de Call Center es el de regular el ritmo de generación de llamadas, de forma que se generen llamadas para ocupar a todos los agentes ociosos, pero sin llegar a superar la disponibilidad de agentes, lo que produciría una acumulación de llamadas en espera en la cola. Para implementar la regulación del ritmo de llamadas generadas, se requiere interrogar a Asterisk sobre el estado de las colas y los agentes. Las herramientas básicas para tal interrogación son los comandos “queue show” y “agent show”.

Los comandos “queue show” y “agent show” son comandos disponibles como parte de la consola de Asterisk. Esta consola de Asterisk es una interfaz de línea de comandos que permite al administrador del sistema interactuar con una instancia de Asterisk en ejecución, por medio de órdenes escritas de forma interactiva, o como parámetros de línea de comando. Además, todos los comandos de la consola de Asterisk están disponibles para todos los clientes AMI a través del comando Command, el cual ejecuta un comando como si se escribiese en la consola de Asterisk, y devuelve

el contenido del resultado de ejecutar el comando.

El comando “queue show” reporta el estado instantáneo de todas las colas de llamadas en Asterisk (o la cola especificada como parámetro del comando). Este estado incluye los agentes disponibles para atender llamadas, llamados Members en el reporte, con sus atributos de atención. Si hay llamadas no asignadas a agentes, se reportan como una lista adicional al final del reporte.

En el anexo 5 se muestra un ejemplo de una ejecución del comando “queue show” que reporta sobre las colas 8000, 8001 y 8100.

El comando “agent show” reporta el estado de disponibilidad de los agentes estáticos que se registran mediante la aplicación AgentLogin(). El anexo 5 contiene un ejemplo que muestra a los agentes 9000 y 9001 en estado no registrado.

### **El protocolo AMI**

Para que una aplicación externa a Asterisk pueda controlarlo y hacer uso de sus capacidades de telefonía, se requiere un

canal de comunicación y un protocolo estándar. Asterisk provee este protocolo estándar a través de AMI (Asterisk Manager Interface).

Asterisk expone el protocolo AMI a través del puerto TCP 5038, usualmente en la interfaz localhost. El protocolo AMI consiste en líneas de texto ASCII, separadas por secuencias de salto de línea (bytes 0x0d,0x0a o 0x0a). Estas líneas de texto se agrupan a su vez en secuencias llamadas paquetes, separadas por otra secuencia de salto de línea (una línea vacía). Las líneas de texto contienen un atributo y un valor para este atributo, separados por dos puntos y espacio.

Existen tres tipos de paquetes reconocidos en el protocolo AMI:

- Petición (Action) enviada por el cliente AMI al servidor.
- Respuesta (Response) enviada por el servidor al cliente AMI.
- Evento (Event) enviado por el servidor al cliente AMI.

Cuando el cliente AMI requiere un servicio de Asterisk, se envía un paquete de petición. Este paquete contiene como primer atributo, una línea de la forma "Action: XYZ", donde

XYZ es el nombre de la acción que se requiere de Asterisk. El paquete puede contener atributos adicionales en el paquete, los cuales dependen de la petición requerida.

El servidor Asterisk, al recibir el paquete de petición, lo procesa y envía un paquete de respuesta. Este paquete se distingue porque su primer atributo es una línea de la forma "Response: XYZ", donde XYZ es una texto que informa el resultado del proceso de la petición (el cual es "Success" en caso de no haber error alguno). Al igual que con el paquete de petición, la respuesta contiene atributos adicionales que dependen de la petición enviada.

Como un ejemplo, el siguiente intercambio muestra la manera en la que un cliente AMI inicia una sesión autenticada AMI con Asterisk. El cliente abre la conexión TCP al puerto 5038 y envía un paquete de petición como el siguiente, seguido de una línea vacía:

Action: login

Username: usuarioami

Secret: unacontrasenia



Si el servidor Asterisk acepta las credenciales, contestará algo como lo siguiente:

Response: Success

Message: Authentication accepted

El último tipo de paquete es el paquete de evento. Este paquete puede ser enviado por el servidor Asterisk en cualquier momento, sin que el cliente lo pida. El propósito primario de los eventos es el de informar en tiempo real a los clientes AMI conectados sobre la actividad de llamadas, colas, canales y demás objetos de telefonía administrados por Asterisk.

Por ejemplo, el siguiente es un ejemplo del evento Dial, el cual se emite en el momento en que Asterisk ejecuta la aplicación Dial() en el plan de marcado para contactar a una extensión o número externo:

Event: Dial

Privilege: call,all

SubEvent: Begin

Channel: Local/1065@from-internal-fd98;2

Destination: SIP/1065-00000003

CallerIDNum: <unknown>

CallerIDName: <unknown>

UniqueID: 1286581757.4

DestUniqueID: 1286581758.5

Dialstring: 1065

### **Eventos de progreso de llamada**

El demonio de Call Center, para implementar el monitoreo y control de llamadas, requiere conectarse como un cliente AMI al servidor Asterisk, y a continuación registrar código que será llamado para cada evento AMI que resulte de interés para el monitoreo. El código que maneja tales eventos forma una fracción importante del código total del demonio de Call Center. Los eventos de interés para el demonio de Call Center son los siguientes:

- Newchannel: se emite este evento cuando Asterisk crea un nuevo canal telefónico.
- Dial: se emite este evento cuando Asterisk inicia el marcado de un número a través de una troncal que

pertenece a una ruta saliente. Para un mismo intento de llamada, se pueden emitir múltiples eventos Dial que corresponden a intentos fallidos en troncales sucesivas.

- **OriginateResponse:** se emite este evento cuando Asterisk tiene una llamada activa resultante de una petición AMI Originate, o cuando el intento de llamada se considera fallido. Los dos escenarios más importantes para este evento son el intento de generar una llamada saliente y la llamada que inicia la sesión del agente estático a través de la aplicación `AgentLogin()`.
- **Join:** se emite este evento cuando una llamada entra a una de las colas administradas por la aplicación `Queue()`. Si la cola a la cual se entra es una de las colas reservadas para llamadas entrantes, se registra la nueva llamada en este momento.
- **Link/Bridge:** se emite este evento cuando un canal de llamada queda enlazado con otro canal y se empieza a intercambiar audio entre los dos canales. Para el demonio de Call Center, el caso principal de interés es el enlace del canal de la llamada generada o entrante

con el canal correspondiente al agente. El evento se emite con el nombre Link en Asterisk 1.4 y Bridge en Asterisk 1.6 pero el significado es el mismo.

- Unlink: se emite este evento cuando dos canales de llamada que fueron previamente enlazados se separan.
- Hangup: se emite este evento cuando un canal activo termina. Este evento se emite en varios escenarios. El primero es cuando un agente estático falla en su autenticación de inicio de sesión y cierra la llamada. Otro escenario es el de una llamada ordinaria que termina. También puede recibirse el evento como parte de un fallo en un intento de generación de llamada, en cuyo caso, se registra información adicional sobre el motivo de fallo.
- Agentlogin: se emite este evento cuando un agente estático completa la autenticación de la aplicación AgentLogin() y queda disponible para recibir llamadas de las colas registradas. En caso de que se reconozca que el inicio de sesión corresponde a un agente, se actualiza la auditoría para marcar el inicio de sesión.

- **Agentlogoff:** se emite este evento cuando un agente estático finaliza su sesión en la aplicación AgentLogin() y deja de estar disponible para recibir llamadas. Se actualiza la auditoría para marcar el final de sesión si el agente es uno de los agentes administrados por el sistema.
- **QueueMemberAdded:** se emite este evento cuando se agrega un agente dinámico a una de las colas administradas por la aplicación Queue(). Si el agente dinámico es una extensión registrada por el sistema, se actualiza la auditoría para marcar el inicio de sesión de forma similar al manejo del evento Agentlogin.
- **QueueMemberRemoved:** se emite este evento cuando se quita un agente dinámico de una de las colas administradas por la aplicación Queue(). De forma similar al evento Agentlogoff, se actualiza la auditoría para marcar el final de sesión.
- **PeerStatus:** se emite este evento cuando ocurre un cambio en el estado de registro de una extensión VoIP. Estos cambios pueden incluir transiciones a estado registrado, a no registrado, o a no alcanzable.

En el caso de la transición a no alcanzable, el demonio de Call Center verifica si la extensión VoIP es un agente dinámico con sesión activa, y si es así, procede a terminar la sesión de agente.

### **Estrategia de funcionamiento del demonio de Call Center**

Al reunir todos los componentes anteriores, se llega a la estrategia básica de funcionamiento del demonio de Call Center, la cual es la siguiente:

- Periódicamente (cada pocos segundos), se evalúan todas las campañas activas y en el intervalo horario de funcionamiento. Para cada campaña, su cola correspondiente es examinada con el comando “queue show” y las llamadas atendidas actualmente por los agentes son identificadas con el comando “agent show”.
- Para cada cola, se identifica el número de agentes ociosos, definidos como agentes disponibles que no están actualmente atendiendo llamadas. Si hay llamadas que están siendo atendidas por agentes, se calcula para cada una de ellas la probabilidad de que la llamada termine luego de un intervalo indicado por

el tiempo promedio que tarda una nueva llamada desde que se marca hasta que se contesta. La formula usada para la predicción es una probabilidad Erlang acumulada, que para el caso de una sola llamada, se reduce a una probabilidad exponencial. El número de llamadas activas cuya probabilidad de terminar es mayor a un nivel de confianza guardado en la configuración se suma al número de agentes ociosos. De este número se resta el número de llamadas que están en proceso de marcación por iteraciones anteriores, y el resultado es el número nuevo de llamadas a generar.

- Para cada campaña, se leen de la base de datos el número de llamadas nuevas a generar. Para esta lectura, se da prioridad a los contactos que no se han marcado previamente, frente a los contactos con intentos fallidos. De los contactos leídos, se eliminan aquellos que constan en la lista de DNC (Do Not Call).
- Con los contactos restantes, se ejecuta el comando AMI Originate. El comando Originate es el medio principal por el cual se manda a generar la llamada. Este comando recibe como parámetros principales la

cadena de marcado y la combinación de extensión, contexto y prioridad. Para el caso particular del demonio de Call Center, la cadena de marcado se construye como `Local/XXXX@from-internal` para hacer uso del plan de marcado del sistema, o como `SIP/nombredetruncal/XXXX` para hacer uso de una troncal directa. Además, la extensión con la que se conecta el marcado corresponde a la extensión asignada a la cola de la campaña. Con esto se asegura la conexión entre la llamada marcada y la cola donde los agentes esperan atender la llamada.

- Cada una de las invocaciones del comando AMI `Originate` tiene un parámetro denominado `Async`, el cual indica que la llamada debe de realizarse de forma asíncrona, y devolver el control al cliente de forma inmediata. Más tarde, cuando Asterisk haya terminado de marcar el número telefónico, se emite el evento `OriginateResponse` con el resultado del marcado.
- Mediante los eventos de progreso de llamada, se sigue la pista de cada llamada a medida que progresa. Eventualmente, si la llamada es exitosa, Asterisk asigna un agente en la cola para que atienda la



llamada. El evento Link indica cuál agente fue asignado para atender la llamada, lo cual queda registrado por el demonio de Call Center.

- El agente atiende al contacto y llena los formularios. El proceso se repite cuantas veces sea necesario para agotar todas las llamadas salientes, o hasta que la campaña salga de horario de atención.

Para el caso de las llamadas entrantes, el procedimiento es parecido, pero sin los pasos de lectura de contactos existentes a marcar. En lugar de eso, el demonio de Call Center sólo empieza a monitorear la llamada cuando ésta entra en la cola reservada para las llamadas entrantes.

## **5.2 Arquitectura inicial de la Consola de Agente y el Marcador**

La primera implementación del marcador del Call Center consistía en un solo proceso de trabajo. Al arrancar el programa, se realiza una secuencia estándar de llamadas a fork() y de manipulación del ID de sesión que culminan en un proceso que queda ejecutándose como un servicio de sistema. Este proceso inmediatamente llama otra vez a fork() para crear un proceso hijo, el cual realiza realmente las tareas del marcador. El primer proceso monitorea al segundo y

lo reinicia en caso de que el proceso hijo finalice anormalmente.

Al iniciarse, el proceso de trabajo abre conexiones a la base de datos. A continuación, se leen las configuraciones del marcador desde la tabla valor\_config. Uno de los datos leídos de valor\_config consiste en las credenciales para conectarse a AMI. Luego se abre la conexión al servidor Asterisk en el puerto de AMI y se instalan los manejadores de eventos de Asterisk. Por último se da marcha al bucle de eventos.

El proceso de trabajo ejecuta secuencialmente las siguientes operaciones dentro del bucle de eventos:

- Verificación del lapso de mínimo de 3 segundos entre generaciones de llamadas.
- Lectura de las campañas que están activas en el instante de verificación.
- Generación de las llamadas de las campañas activas, según su estado de reintento.
- Verificación de actividad en las conexiones de red abiertas por el proceso. Inicialmente, la única conexión abierta era la conexión AMI.

- Manejo de los eventos AMI de interés para el demonio de CallCenter, listados anteriormente.

Para el manejo de las llamadas generadas, el proceso mantiene una lista de registros de las llamadas en curso. En esta lista se almacena el momento en que se generó la llamada, el número marcado, los canales involucrados y eventualmente el agente que Asterisk asigna para manejar la llamada. Para cada transición de interés, el estado actualizado de la llamada se actualiza en la base de datos, en la tabla "calls". Adicionalmente, para el evento Link recibido cuando una llamada se conecta con un agente, se inserta un registro nuevo en la tabla "current\_calls", el mismo que se borra con el correspondiente evento "Hangup". La Figura 5.30 muestra un diagrama de este modelo de programa.

La primera implementación de la Consola de Agente fue una implementación basada enteramente en el examen de la tabla "current\_calls". Para lograr una aproximación de un comportamiento en tiempo real en esta versión de la Consola de Agente, se realizaba lo siguiente:

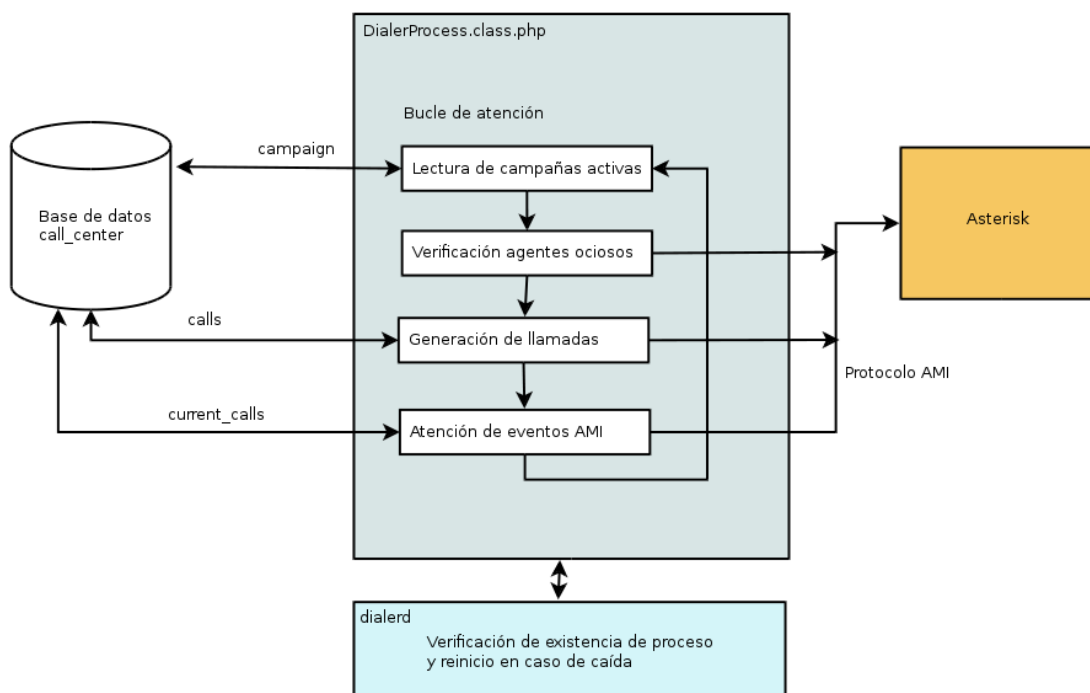


Figura 5.30: Arquitectura de primera implementación de programa marcador con actualización a tabla current\_calls.

- El módulo genera el HTML inicial para la vista de la consola, más las funciones de Javascript que implementan la interactividad. Una de ellas instala un timer que ejecuta una petición al servidor Elastix cada 4 segundos.
- Al recibir la petición periódica desde el navegador, el servidor consulta la tabla "current\_calls" para determinar si el agente de la sesión está conectado a una llamada.
- Si el estado de conexión actual es distinto del estado consultado en una petición anterior (guardado en el estado de la sesión), se lee toda la información asociada a la

llamada: propiedades de la campaña y definiciones de los formularios asociados.

- Se genera una nueva versión de la vista de la consola con los datos actualizados.

### **5.3 Introducción del manejo de llamadas entrantes**

En la primera versión del sistema de Call Center, sólo se implementó la generación y manejo de llamadas hacia el exterior a partir de la lista en la base de datos. En este modo de operación se puede usar cualquier cola existente como destino para las llamadas generadas, con el único requisito de que la cola tenga los agentes correspondientes configurados. Poco después de que se implementara este modo de operación, el cliente presentó un nuevo requerimiento para un segundo modo de operación.

Además de las llamadas que se generan desde la empresa con el propósito de buscar clientes, el cliente también recibe llamadas desde el exterior. Una fracción importante de estas llamadas son de personas interesadas en los productos de la empresa, y que son potenciales oportunidades de venta. La empresa tiene interés de atender tales llamadas a través de su equipo de agentes y de hacer uso de la administración que ya se implementó para las llamadas

generadas desde la empresa. Para lograr esto, estas llamadas son conducidas a través de un IVR que desemboca en una cola. A partir de este punto, las llamadas pueden ser administradas por el proceso de Call Center.

Para la implementación de este modo de operación, se empezó por implementar la reserva de una o más colas para la atención de llamadas entrantes. Tales colas, escritas en la tabla "queue\_call\_entry" quedan entonces excluidas de ser elegidas para posibles campañas salientes que se estén realizando simultáneamente. Al entrar en una de las colas reservadas, el proceso de Call Center registra la llamada en una tabla separada, la tabla "call\_entry". En esta tabla se escriben los instantes en que la llamada entra en la cola, se asigna un agente y el momento en que termina la llamada, de forma parecida a como se hace con las llamadas salientes autogeneradas. Para la Consola de Agente, se creó una nueva tabla, "current\_call\_entry", la cual contiene la información de las llamadas que están siendo actualmente atendidas por agentes, de forma análoga al uso de la tabla "call\_entry". La implementación de la Consola de Agente fue modificada de forma correspondiente para examinar también esta tabla de forma periódica.

Por último, el mismo requerimiento de recolectar información con el uso de formularios que fue descrito para las llamadas autogeneradas existe también para las llamadas entrantes. Para acomodar este requerimiento, se introduce la campaña entrante. Una campaña entrante se define como la reservación de una cola de llamadas entrantes para la recolección de llamadas durante un intervalo de tiempo, con uno o más formularios asociados. La administración de las campañas entrantes se realiza a través de un nuevo módulo, llamado precisamente Campañas Entrantes. Este módulo implementa la creación, actualización y desactivación de las campañas entrantes, la asignación de formularios a partir del mismo conjunto disponible para las campañas salientes, y la descarga en formato CSV de los datos recolectados en los formularios para una campaña entrante, de forma idéntica a como se realiza para las campañas salientes.

#### **5.4 Limitaciones de la arquitectura inicial de la Consola de Agente**

Como se mencionó anteriormente en la sección 5.2, la primera implementación de la Consola de Agente intentaba conseguir una actualización de información en tiempo real a través de la lectura periódica de las tablas “current\_calls” y “current\_call\_entry” cada 4

segundos. Aunque esta implementación bastó como una prueba de concepto, este modelo no es fácilmente escalable para una cantidad superior a 10 agentes. A medida que se agregan agentes en atención simultánea, se notaron las siguientes deficiencias:

- Cada verificación de cada agente requiere una inversión de recursos por parte del servidor Elastix. Primero se requiere el ancho de banda de la red usada para recibir la petición y para enviar su respuesta. Luego cada verificación necesita consultar las tablas de la base de datos, “current\_calls” y “current\_call\_entry”, para constatar si hay cambios con respecto al estado anterior. Estas consultas implican acceso al disco del servidor y toman tiempo. Por último, la verificación debe interpretar los resultados y decidir el nuevo estado del agente. Este esfuerzo crece de forma proporcional con el número de agentes en atención simultánea.
- Una gran fracción de las verificaciones periódicas para el agente encuentran que no han habido cambios respecto a la última verificación. Esto es consecuencia de la siguiente observación: cada agente que atiende llamadas pasa gran parte del tiempo en un estado que no cambia. Si un agente está atendiendo una llamada, esta atención dura algunos



minutos, durante las cuales las verificaciones periódicas devuelven el mismo resultado que la verificación inmediatamente anterior. De la misma manera, si un agente ha indicado un break, este break también dura algunos minutos, durante los cuales no cambia el resultado de la verificación.

- Debido a que la verificación al servidor Elastix se realiza cada 4 segundos, la interfaz de la Consola de Agente puede demorar hasta 4 segundos en actualizarse cuando se presenta un verdadero cambio en el estado del agente. Este retraso en la actualización puede interferir en la presentación del agente frente al usuario atendido, en el caso de que el agente cambie de ocioso a ocupado.

La combinación de los tres puntos indicados tiene como resultado que con una cantidad de una docena o más de agentes, el servidor Elastix dedique una fracción muy grande del CPU a responder peticiones de estado de la Consola de Agente. Muchas de estas peticiones resultan en el mismo estado en que estaba el agente, con la excepción de los pocos agentes que cambian de estado en un intervalo cualquiera de 4 segundos (muchas veces ninguno). Con una cantidad suficientemente grande de agentes, el servidor

no puede responder de manera oportuna. Como consecuencia, la actualización de las interfaces de cada agente empieza a retrasarse. Adicionalmente, cualquier intento de reducir el intervalo de petición con el propósito de recortar el retraso de actualización resulta en un incremento de la carga observada de CPU.

### **5.5 Introducción del protocolo ECCP (Elastix CallCenter Protocol)**

Poco después de que se iniciara el análisis del problema con la primera versión de la Consola de Agente, un desarrollador externo a la empresa Palosanto Solutions ofreció una oportunidad de un desarrollo conjunto. Según este desarrollador, la Consola de Agente basada en HTML y Javascript era una implementación con capacidades limitadas, y se proponía el desarrollo de una nueva Consola de Agente. Esta extensión a la Consola de Agente no estaría basada en HTML, sino que sería implementada como un programa separado, instalable en la computadora de cada agente.

Para permitir la implementación de este programa, surge de inmediato la necesidad de permitir el control de las actividades del sistema de CallCenter de forma remota. Luego de rondas de negociación con el desarrollador, se llegaron a las siguientes conclusiones:

- Se debe de construir un protocolo que permita a un programa remoto (la interfaz alterna de Call Center) recibir información sobre el progreso de la campaña para un agente específico, y que permita controlar el estado del agente (login y logout de agente, estado de pausa). Hasta ahora, estas actividades se realizaban con el código del módulo Elastix dentro de la Consola de Agente.
- No se debe de dar al programa remoto a desarrollar el control directo del protocolo AMI, ya que AMI no provee todas las funcionalidades del Call Center, y además da acceso a capacidades en exceso de las estrictamente necesarias para el control. El acceso directo arriesga interacción indeseada (accidental o deliberada) con Asterisk.
- Por las mismas razones, no es adecuado dar acceso remoto a la base de datos del Call Center, y adicionalmente el protocolo de base de datos no puede notificar sobre la actividad de Asterisk. Esto obligaría a consultar periódicamente la base de datos como lo hace la primera versión de la Consola de Agente, con la dificultad adicional de realizarse a través de la red en lugar de en la misma máquina.
- En consecuencia, se requiere un protocolo de red que

permita mandar peticiones al sistema de Call Center y recibir sus respuestas, además de enviar notificaciones de estado al cliente conectado.

Los dos principales protocolos estructurados con bibliotecas disponibles al momento de implementar la funcionalidad eran XMLRPC y SOAP. Ambos soportan el envío de mensajes estructurados para implementar llamadas de procedimientos, pero las bibliotecas disponibles no soportan el envío de mensajes asíncronos desde el servidor al cliente. Debido a eso se decidió implementar desde cero un nuevo protocolo, basado en XML.

El protocolo XML que se construyó fue denominado Elastix Call Center Protocol, abreviado a ECCP. Las características que se implementaron en ECCP fueron las siguientes:

- La comunicación entre el cliente y el servidor consiste en el intercambio de documentos XML bien formados, que conforman los paquetes. Existen tres tipos de paquetes: requerimiento, respuesta y evento.
- El protocolo define dos niveles de autenticación. Cuando el cliente se conecta, sólo está disponible un requerimiento que enumera el resto de requerimientos soportados, y el

requerimiento que inicia una sesión autenticada. El inicio de sesión es el primer nivel de autenticación, el cual requiere de un usuario y contraseña. Al autenticarse correctamente, están disponibles los requerimientos que leen información sobre campañas, colas y agentes. Para poder modificar el estado de un agente de Call Center, se requiere un segundo nivel de autenticación, con una contraseña asociada al agente.

- El cliente puede indicar opcionalmente que está interesado únicamente en eventos relacionados a un agente en particular. Este filtro es adecuado para implementar una Consola de Agente. Si se desactiva el filtro, se reciben los eventos de todos los agentes, lo cual es adecuado para implementar un monitor de progreso de campañas.
- Un cliente puede ordenar a través del ECCP las siguientes operaciones que afectan al agente: inicio de sesión ECCP, finalizar sesión ECCP, filtrar eventos por agente, inicio de sesión de agente, finalizar sesión de agente, guardar información de formulario para una llamada, transferencia de llamada, pausar agente (break), finalizar pausa de agente, colgar llamada, agendar llamada.
- El protocolo ECCP soporta requerimientos para obtener la

siguiente información: lista de requerimientos soportados, obtener estado del agente, obtener guión asociado a la cola entrante, obtener lista de campañas, obtener información detallada sobre campaña, obtener información sobre una llamada de una campaña, listar descansos disponibles, obtener resumen de actividad de agente, leer variables de canal de la llamada atendida.

- Los siguientes eventos se emiten desde el servidor para indicar el progreso de la atención de las llamadas: agente enlazado (a llamada), agente desenlazado, agente inició sesión, agente terminó sesión, agente falló en iniciar sesión, agente inició un descanso, agente finalizó un descanso.

La implementación del protocolo ECCP fue agregada al código del proceso marcador de Call Center para que fuese puesta a prueba por el desarrollador externo en su aplicación. En paralelo con estas pruebas, se observó que las capacidades implementadas en el protocolo ECCP pueden ser usadas en una reimplementación de la Consola de Agente basada en HTML, con el modelo de notificación Javascript denominada actualmente Comet.

El modelo Comet de notificación Javascript consiste en una serie de estrategias para que un servidor de aplicaciones Web pueda hacer llegar una notificación de un evento a una interfaz HTML con Javascript, sin que esta última tenga que contactar periódicamente al servidor para buscar nuevos eventos. Para la reimplementación de la Consola de Agente se implementaron dos estrategias Comet, y se elige una u otra según las capacidades del navegador Web:

- Long-Polling: soportado en todos los navegadores incluyendo Internet Explorer. En esta estrategia, el navegador contacta al servidor para pedir eventos. Si no hay un evento disponible, entonces el servidor deja abierta la conexión sin mandar una respuesta. La conexión permanece sin respuesta hasta que se tenga al menos un nuevo evento, o hasta que transcurra un tiempo de expiración definido por el programador. En cualquiera de las dos condiciones, el servidor envía la respuesta y cierra la petición. A continuación el navegador genera una nueva conexión, y se repite el ciclo.
- Server-Sent Events: soportado en Firefox y Google Chrome, y las versiones más recientes de Internet Explorer. En esta estrategia, se hace uso de un objeto especial Javascript EventSource que inicia una conexión al servidor, e interpreta

una respuesta de tipo text/event-stream. El servidor puede entonces mandar mensajes sucesivos en la conexión de respuesta, separados por pausas de longitud arbitraria, sin cerrar la conexión.

Para la nueva implementación de la Consola de Agente, el código PHP que se ejecuta en el contexto del servidor Web recibe una conexión (para Long-Polling o Server-Sent Events, según la capacidad detectada). A su vez, este código PHP implementa un cliente ECCP que se conecta al proceso marcador del Call Center que se ejecuta en el mismo servidor. Este cliente ECCP implementa un bucle de proceso de mensajes, los cuales alimentan el flujo Comet al recibir el evento correspondiente de ECCP. Con esta reimplementación se logra una gran reducción de carga de CPU, ya que el bucle de eventos permanece ocioso hasta que se recibe un evento que requiera una actualización del estado de la Consola de Agente. También se logra una reducción del tráfico de red requerido, además de un retraso mucho menor entre ocurrencia del evento y la actualización de la interfaz de la Consola de Agente. Finalmente, la implementación de un cliente de referencia logró ejercitar la implementación y definición del protocolo ECCP, con lo que se verificó que el protocolo es adecuado para la construcción



de un programa cliente real. Esta Consola de Agente, con la arquitectura mostrada en la Figura 5.31, reemplazó a la implementación anterior a partir de la versión 2.1.0-0 del Call Center.

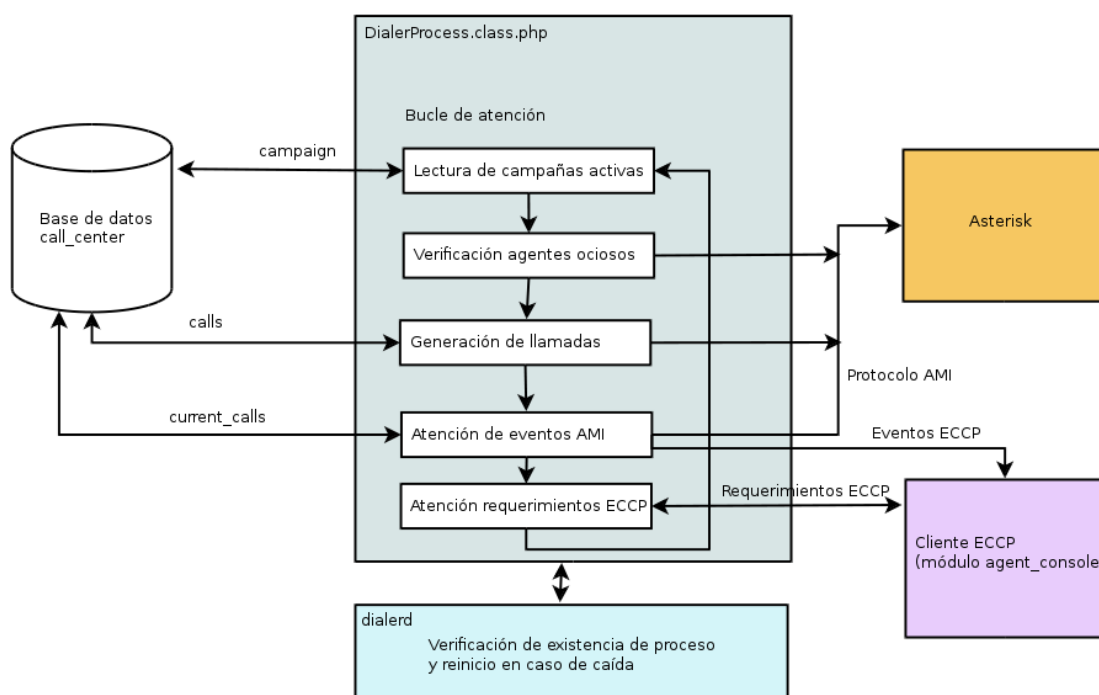


Figura 5.31: Arquitectura de programa marcador monoprocés con protocolo ECCP

Luego de implementado el protocolo ECCP y la nueva Consola de Agente, se observó que el modelo de actualización en tiempo real con un bucle de mensajes podía aplicarse a otros módulos. Los siguientes módulos de monitoreo fueron escritos o reimplementados

para usar el protocolo ECCP de forma parecida a la Consola de Agente:

1. Reporte de monitoreo de agentes
2. Reporte de monitoreo de llamadas entrantes
3. Monitoreo de campañas

De estos módulos, los dos primeros fueron implementados antes del protocolo ECCP, y usaban peticiones periódicas al servidor Elastix para su actualización. Al ser migrados al ECCP, se obtuvieron los mismos beneficios que al migrar la Consola de Agente, especialmente porque las lecturas a la base de datos para los reportes eran costosas y poco adecuadas para realizar actualizaciones periódicamente.

## **5.6 Limitaciones observadas del Marcador monoproceso**

En la sección 5.2 se mencionó la arquitectura de la primera implementación del proceso marcador. Las dos características de interés en esta primera implementación son que toda la implementación está ejecutándose en un solo proceso y que las lecturas y escrituras a la base de datos se realizan en el mismo hilo de ejecución que atiende la llegada de mensajes AMI, y más tarde las peticiones ECCP. Al implementar el Call Center en un cliente en Venezuela que tenía unos 50 agentes simultáneos, se detectaron

limitaciones en la implementación que producían demoras de varias decenas de segundos entre el inicio de la atención por parte del agente y la actualización de la interfaz de la Consola de Agente. Del análisis de la situación del cliente, se observó lo siguiente:

- El servidor del cliente era un equipo con múltiples procesadores (8), pero debido a la arquitectura de un solo proceso, no es posible usar procesadores adicionales para las tareas del programa marcador.
- El manejo de los eventos de AMI en el proceso marcador involucra actualizaciones a la base de datos. Toda lectura o escritura a la base de datos es una operación de E/S que bloquea la ejecución del programa hasta que la operación se completa. Debido a que un solo proceso maneja los eventos AMI, y además realiza accesos a la base de datos en el manejo de AMI, el procesamiento se enlentece y los eventos AMI sin procesar se acumulan. Esto es uno de los motivos del retraso observado.
- Durante tráfico pesado de llamadas, el cliente AMI queda inundado de eventos, de los cuales sólo una fracción pueden ser de interés. En estas condiciones, la ejecución de un requerimiento AMI, como por ejemplo la petición del estado de una cola, es más lento de enviar y recibir. Tales

requerimientos AMI, enviados dentro del manejador de un evento AMI anterior, son otra causa del retraso observado.

- Se identificaron instancias (datos de llamadas entrantes, estado de agentes, manejo de llamadas en pausas) donde los datos del estado que fueron escritos a la base de datos en una parte del programa fueron consultados posteriormente en otra parte. Debido a que el proceso de marcador es el único que actualiza estos datos, las lecturas posteriores de los mismos datos son innecesarias.
- Muchas de las consultas disponibles a través del protocolo ECCP (información de campaña, formularios, llamadas, pausas disponibles) constituyen accesos a la base de datos. Ya que el mismo proceso maneja el ECCP y el manejo de AMI, toda consulta a la base de datos retrasa el procesamiento de eventos AMI. Esta es una tercera causa del retraso observado.
- La colocación de las llamadas salientes requiere algunas peticiones AMI para consultar el estado de las colas y los agentes y además lecturas a la base de datos para cargar las llamadas a generar. Otra vez, debido a que el mismo proceso maneja los eventos AMI y la colocación de llamadas, el manejo de los eventos AMI se retrasa mientras se recoge

la información para el siguiente ciclo de generación de llamadas. Si se intenta retrasar la colocación de llamadas para dar prioridad al manejo de eventos AMI (un paliativo intentado en el cliente) eventualmente la generación de llamadas se paraliza porque todo el procesamiento se dedica al manejo de AMI y el protocolo ECCP.

### **5.7 Introducción del Marcador multiproceso**

Debido a las deficiencias observadas arriba, se llegó a la conclusión de que se debía reimplementar el programa de marcación del Call Center con una nueva arquitectura que solucione los problemas indicados. Las especificaciones para la construcción del programa fueron las siguientes:

- El dialer nuevo será multiproceso. Deben existir al menos tres procesos: el proceso dedicado a manejo AMI, el proceso de colocación de llamadas, y el proceso que maneja el protocolo ECCP.
- Se requiere coordinar el trabajo entre todos los procesos para proveer las mismas funcionalidades de la primera implementación. Para eso, se requiere construir un mecanismo que permita el intercambio de mensajes entre los procesos. Como mínimo, se requiere esto para que cada

proceso pueda ser terminado de forma ordenada. A menos que se requiera lo contrario, todos los mensajes transmitidos entre procesos serán asíncronos. No debe requerirse que el emisor del mensaje requiera un mensaje de vuelta del receptor.

- De entre los procesos mencionados, el proceso de eventos AMI requiere manejar los eventos con el mínimo de demora. Por lo tanto, toda interacción que requiera bloquear el proceso en espera de una respuesta o resultado debe quitarse. La consecuencia de esto es que el proceso de eventos AMI no debe interactuar directamente con la base de datos.
- El proceso de colocación de llamadas no tiene requerimientos fuertes de tiempo real. Por lo tanto, a este proceso se le encarga la tarea de recibir mensajes de actualización desde el proceso AMI.
- Adicionalmente, el proceso de colocación de llamadas envía el listado de llamadas que fueron generadas al proceso AMI, donde se registran en una lista. Esta lista se actualiza en memoria con cada evento AMI recibido. Otro tanto ocurre con la lista de agentes. De esta manera se evita esperar por la base de datos con cada actualización o verificación del

estado de llamadas y agentes.

- Cuando se requiere notificar un evento de ECCP, el proceso AMI envía un mensaje al proceso ECCP con los datos sacados de sus listas de objetos. Si el evento ECCP requiere más información, se lee de la base de datos en el proceso ECCP, no en el proceso de eventos AMI.

La arquitectura multiproceso resultante se muestra en la Figura 5.32.

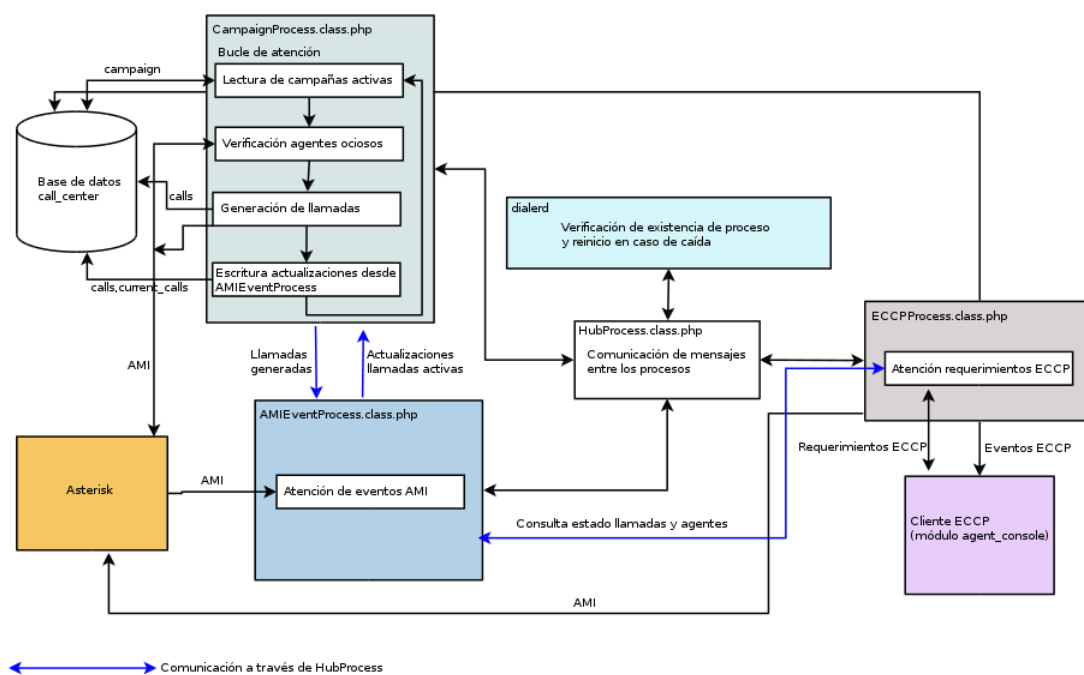


Figura 5.32: Arquitectura de programa marcador multiproceso con ECCP

Luego de implementar la nueva versión del programa marcador del Call Center con estas especificaciones, el servidor redujo

enormemente la carga de CPU, y fue capaz de atender los 50 a 100 agentes simultáneos que el cliente requería. Los cambios realizados fueron entonces integrados a partir de la versión 2.1.99-0.alpha del Call Center y declarados estables a partir de la versión 2.2.0-0.

## **5.8 Implementación de agentes dinámicos**

Desde la implementación inicial del Call Center, el modelo de manejo de los agentes ha consistido en crear canales de tipo Agent/XXXX para agregarlos a las colas como sea necesario. Cuando se usa una extensión SIP o IAX para conectarse al canal de tipo Agent, se inicia una llamada desde el servidor Elastix hacia la extensión. Esta llamada debe permanecer abierta durante toda la duración de la sesión, y si se cierra esta llamada, el marcador registra un final de sesión. El agente escucha música de espera durante todo el intervalo en el cual no está atendiendo llamadas.

En algunas implementaciones del Call Center, particularmente para atención de llamadas entrantes, se requiere un sistema de funcionamiento distinto. En estos casos, el agente debe de iniciar sesión en el Call Center pero permanece con el teléfono cerrado hasta que una llamada se le asigna, en cuyo caso la extensión del



agente timbra. Entonces el agente debe de contestar la llamada y usar la interfaz de la Consola de Agente de la manera acostumbrada. A este modo de funcionamiento también se lo denomina Callback Login.

Para la implementación de este modo de funcionamiento, ya no es adecuado el uso de la aplicación AgentLogin(). La razón es que esta aplicación sólo puede usarse en una llamada activa, la cual es la llamada que inicia la sesión del agente. Para poder mantener el teléfono cerrado, es necesario timbrar la extensión directamente, lo cual a su vez requiere agregar la extensión (SIP o IAX) como miembro de la cola de atención.

Para acomodar el nuevo requerimiento, se introduce el concepto de Agente Dinámico. Un agente dinámico es una extensión SIP o IAX preexistente (creada a través de FreePBX) que se registra como un nuevo agente en el módulo de Agentes Dinámicos. De forma similar a como se realiza para los agentes con el canal Agent/xxxx que funcionan con el modo anterior, las extensiones marcadas como Agentes Dinámicos se agregan a la definición de la cola en la configuración de cola de FreePBX. Sin embargo, deben de añadirse en el listado marcado como Agentes Dinámicos, y de la forma

S1234 (para la extensión SIP 1234) o I1234 (para la extensión IAX 1234).

Dentro del código del marcador, el manejo de un agente dinámico es distinto. Ya que no se usa la aplicación `AgentLogin()`, no es posible manejar los inicios y finales de sesión a través de los eventos `Agentlogin` y `Agentlogoff` de AMI. En lugar de esto se instalan manejadores de los eventos `QueueMemberAdded` y `QueueMemberRemoved`. Los agentes dinámicos empiezan en un estado de no pertenencia a la cola, según el reporte del comando "queue show". Un agente dinámico se considera que inició sesión si se detecta al menos un evento `QueueMemberAdded` que involucre este agente. Este evento se lanza luego de completar el ingreso como miembro a la cola por parte de la extensión. De forma análoga, se marca el final de la sesión del agente cuando se detecta al menos un evento `QueueMemberRemoved`, que marca el momento en que la extensión fue retirada como miembro de la cola.

Cuando se requiere iniciar la sesión del agente dinámico, la página de login de la Consola de Agente cambia para mostrar las extensiones configuradas como agentes dinámicos, más un campo de contraseña. La contraseña se compara con la que fue ingresada

al registrar el agente dinámico, y si es correcta, se envía un requerimiento ECCP loginagent, igual que con el agente estático. El programa marcador verifica que el agente es realmente un agente dinámico, y a continuación ejecuta el comando AMI QueueAdd sobre la lista de colas a las cuales pertenece el agente en cuestión. Esta lista se la obtiene de la base de datos interna de Asterisk con el comando AMI “database show”, y es la misma lista que se configura en la interfaz de FreePBX.

### **5.9 Situación actual y limitaciones pendientes**

En la versión actual del sistema Elastix CallCenter (2.2.0-7) se tiene la arquitectura de tres procesos para el programa marcador, el modo dual de funcionamiento de llamadas salientes y llamadas entrantes, la elección entre agentes estáticos y dinámicos y el protocolo ECCP que permite el uso de eventos asíncronos e implementaciones estilo Comet para los módulos de Consola de Agente, Monitoreo de Campañas, Monitoreo de Agentes y Monitoreo de Llamadas Entrantes. Se conocen las siguientes características y decisiones de diseño que pueden todavía limitar el funcionamiento del Call Center:

- El proceso ECCP es un solo hilo de ejecución que intenta atender secuencialmente todos los requerimientos de todos

los agentes conectados. Varios de los requerimientos leen y escriben a la base de datos. Si una lectura de la base de datos tarda mucho tiempo, esto detiene todas las otras peticiones del resto de clientes hasta que la lectura se complete. Los efectos de las consultas lentas sobre el protocolo ECCP se han mitigado hasta ahora mediante índices sobre las tablas de la consulta requerida.

- Es posible construir un cliente de monitoreo que consulte una misma información sobre muchas entidades sucesivas usando múltiples peticiones sucesivas de una forma que frena la atención del resto de clientes. Por ejemplo pedir las colas a la que pertenece un agente, luego el siguiente agente, y así sucesivamente. Se ha comprobado que es mucho más eficiente ejecutar una sola petición que lista todas las entidades de interés, que pedir en secuencia la información sobre cada entidad.

## **CAPÍTULO 6**

### **6 ANÁLISIS DE RESULTADOS**

En este capítulo se estudia hasta qué punto fueron cumplidos los objetivos inicialmente propuestos al construir el sistema de Call Center. A continuación se enumeran funcionalidades deseables que no están escritas todavía, con una lista de obstáculos que dificultan la implementación de tales funcionalidades. Por último se discuten maneras de mejorar el esquema de datos y herramientas de programación a usar en un posible rediseño del sistema.

#### **6.1 Evaluación de cumplimiento de objetivos**

El objetivo general propuesto, de disminuir la cantidad de errores humanos y las ineficiencias en el procedimiento manual, fue alcanzado. Se eliminaron los errores de digitación de números de teléfono y los errores al llenar los datos del formulario. Al remover el

paso de digitación del número a contactar, se redujo el tiempo que los agentes pasan esperando hasta atender la siguiente llamada. Como ejemplo, una implementación del Elastix Call Center hecha por IPTegra para Sistem Contact Center en Barranquilla, Colombia logró un incremento de atención al cliente del 300% con respecto a la situación inicial sin un software dedicado al manejo de un callcenter. [13]

Con respecto a los objetivos específicos:

- El número de llamadas por agente fue maximizado a través de la automatización de la generación de llamadas y por el marcado predictivo. Las interrupciones en la atención por parte del agente son ahora analizables a través de los módulos de Reportes de Recesos y de Tiempo de Sesión.
- El tiempo muerto entre llamadas fue minimizado a través de los mismos mecanismos que maximizaron el número de llamadas.
- El reporte CSV de las llamadas generadas para cada campaña contiene los campos de fecha de generación de la llamada y de duración de la atención. De esta manera se cumple el objetivo de automatización de la auditoría de tiempo de inicio y duración de las llamadas.

- El objetivo de monitorear los descansos de los agentes se cumple a través del módulo de Reporte de Recesos.
- Los errores humanos en los pasos del procedimiento de trabajo del callcenter fueron en gran parte eliminados por la implantación del sistema Call Center. El factor humano fue eliminado de la marcación, la auditoría y los reportes. Todavía existe la posibilidad de errores humanos en la interpretación de los datos recolectados de la campaña, luego de su exportación en formato CSV.
- El objetivo de monitoreo en tiempo real se cumple a través de los reportes de Monitoreo de Campañas y Monitoreo de Agentes.

## **6.2 Características deseables a implementar**

La construcción del proyecto Call Center ha dado como resultado un sistema que cumple con los requerimientos que se establecieron al inicio. Esto fue posible debido a que el software de telefonía Asterisk provee aplicaciones que se corresponden directamente a aspectos específicos del diseño del Call Center. La aplicación Queue() es una representación directa de la cola de atención requerida para la generación de las llamadas salientes y la atención de las llamadas entrantes, y la aplicación AgentLogin() implementa

la especificación de que el agente debe recibir de inmediato la llamada, sin demora para contestar.

Un segundo factor que permitió la construcción del sistema fue el uso del protocolo AMI. El protocolo AMI de Asterisk define eventos que señalan transiciones de estado de todas las llamadas realizadas. La existencia de eventos de progreso de la llamada en el protocolo dio como resultado que se pueda implementar la notificación en tiempo real de la llamada atendida y el registro de la duración de la llamada.

Finalmente, el uso de FreePBX suministró una estructura para las capacidades de Asterisk, con lo cual se elimina la necesidad de construir tal estructura desde cero antes de implementar las características específicas del Call Center. Sin embargo, todos estos factores permiten soportar características adicionales a las ya implementadas.

Al realizar las implementaciones del sistema Call Center, se compiló una lista de varias características que serían útiles de implementar. Tales características no han sido escritas todavía, debido a razones como limitaciones del diseño actual de la base de datos, dificultades



en la migración de los datos en instalaciones existentes o insuficiente análisis para implementar la característica de una manera genérica. Las características deseables a implementar son:

- Soporte para generación de llamadas salientes que, en lugar de conectarse a un agente humano, reproduzcan una grabación al cliente contactado, opcionalmente con parámetros de la llamada que elijan fragmentos de grabación o TTS. Esta funcionalidad está fuera del alcance inicialmente definido para el sistema Call Center, el cual conduce llamadas a ser atendidas por agentes humanos.
- Múltiples números telefónicos por contacto. Esta característica implica que cada número sea intentado durante la campaña hasta que uno de ellos conteste, y luego el resto de los números pueden ser ignorados. Esta característica requiere una migración de las bases de datos existentes para separar la información del contacto de la lista de números a marcar.
- Formularios con campos condicionales. Actualmente los campos de los formularios son completamente independientes entre sí. La asignación o elección de un valor en un campo no influye de manera alguna en la presencia o rango de valores disponibles en el resto de los campos. Con

campos condicionales, sería posible definir grupos de campos cuya presencia requiere que otro campo tenga un valor específico.

- Reciclado de contactos o números. Esta característica permite seleccionar un subconjunto de los números o contactos de una campaña en base a un criterio de estado de la llamada (fallida, llamada corta) o según los valores seleccionados a través de un formulario. Esta funcionalidad requiere un análisis más extenso para ser implementada de forma general.
- Soporte para transferencia entre agentes. Esta característica permite que un agente que atiende una llamada, pero no tiene un conocimiento o autorización requerido para satisfacer al cliente, pueda mandar la llamada a un segundo agente que sí lo tenga. Actualmente hay soporte para transferencia de llamadas, pero la transferencia termina la atención de la llamada dentro del Call Center. Para implementar esta característica, se requiere el soporte de sesiones de atención (a continuación) o una capacidad equivalente para registrar la reconexión de la llamada transferida a otro agente.

- Sesiones de atención. Esta característica consiste en poder agrupar todas las interacciones para un mismo contacto dentro de una campaña como una sola sesión de atención. Esto incluye la llamada original más todos los agendamientos y transferencias realizados.

### **6.3 Obstáculos planteados por diseño actual**

Los siguientes puntos conocidos plantean obstáculos para la implementación de nuevas características y para la mejora del rendimiento del sistema Call Center:

- En la base de datos, la tabla de llamadas salientes (tabla “calls”) combina la información del contacto, la información del número a marcar y el estado del intento de la marcación. Esta consolidación no causó problemas para la implementación inicial, pero es el principal obstáculo para la implementación de múltiples números por contacto. Además esta consolidación dificulta el análisis post-ejecución del historial de marcación de las llamadas, puesto que sólo se conserva el estado del último intento.
- La tabla usada para la auditoría (tabla “audit”) guarda los intervalos de inicio y de final de las sesiones de los agentes y además los intervalos de inicio y final de los descansos

tomados. Pero no hay una relación explícita en la base de datos de que un descanso fue tomado durante una sesión específica, y por lo tanto debe deducirse de forma indirecta a través de los intervalos de inicio y fin. De la misma manera, no hay una relación explícita entre la sesión de un agente y las llamadas atendidas durante una sesión específica del agente.

- En el programa marcador, el manejo de todas las conexiones ECCP en un solo proceso es el principal obstáculo potencial a mejoras en el rendimiento. La razón por la cual se adoptó esta arquitectura en lugar de mantener o iniciar nuevos procesos o hilos para atender las peticiones es que todas las peticiones ECCP deben referenciar una sola fuente de información compartida. Esta consiste en el conjunto de objetos creados para representar llamadas y agentes en el proceso de atención de eventos AMI (AMIEventProcess.class.php). Sin embargo, el manejo en un solo proceso obliga a serializar las peticiones de todos los clientes.

#### 6.4 Puntos de rediseño en base de datos

Algunos de los obstáculos mencionados pueden ser resueltos con un rediseño de las tablas de la base de datos. Los cambios propuestos al esquema son los siguientes:

- División de la tabla “calls” en tres tablas separadas. Las tablas sugeridas son “campaign\_contact”, “contact\_number” y “call\_attempt”. La tabla “campaign\_contact” debe conservar la clave foránea hacia la tabla “campaign” y los registros de las tablas “form\_data\_recolected” (datos del formulario) y “call\_attribute” (atributos del contacto) deben de apuntar a esta tabla. La tabla “contact\_number” se destina a guardar el número telefónico y debe almacenar la clave foránea hacia la tabla “campaign\_number” para la relación de muchos-a-uno. Si es necesario, también debe de almacenar el número total de reintentos hacia este número. La tabla “call\_attempt” se debe usar para el historial de reintentos a un solo número telefónico y debe almacenar, además de la clave foránea hacia “contact\_number”, el estado de éxito, los tiempos de eventos del intento de llamada, los códigos de error, el agente que atendió la llamada y el valor de Uniqueid de Asterisk. El agendamiento se puede implementar agregando los tiempos de inicio y final de horario de agendamiento a la

tabla “campaign\_number” para el número elegido.

- La tabla de auditoría debe de separarse en dos tablas. Se sugiere “agent\_session” y “agent\_break”. La tabla “agent\_session” debe de contener los inicios y finales de sesión de los agentes. Las tablas “call\_attempt” (del rediseño) y “call\_entry” deben tener claves foráneas hacia “agent\_session” para indicar la atención dentro de una sesión de agente. De la misma forma, la tabla “agent\_break” debe de tener una clave foránea hacia “agent\_session”, para indicar un descanso tomado dentro de una sesión de agente.

## **6.5 Lenguajes de programación y bibliotecas de software para rediseño**

Para un proyecto que consiste de varios componentes con tareas bien definidas, como el sistema Call Center, no existe un obstáculo fundamental que impida que cada componente sea implementado usando bibliotecas de software distintas, o incluso lenguajes de programación distintos. El componente de los módulos en el GUI de Elastix debe de implementarse en PHP, ya que éste es el lenguaje en el cual está escrito el resto del GUI. Por otra parte, el programa marcador no está obligado a ser escrito en el mismo lenguaje. Esto conduce a analizar si es posible obtener beneficios al

reimplementar el programa marcador en un lenguaje de programación distinto, con las bibliotecas disponibles para el nuevo lenguaje.

Se ha mencionado en la sección 6.3 que las peticiones ECCP son serializadas debido al acceso compartido al conjunto de objetos que representa las llamadas en atención y los agentes. Si una petición ECCP requiere un acceso de base de datos lento, todas las otras peticiones de otros clientes ECCP quedarán estancadas mientras se espera la respuesta. La razón de este inconveniente es la ausencia de un mecanismo en PHP para aplicar entrada/salida asíncrona (asynchronous I/O) a una conexión de base de datos.

Si se mantiene el lenguaje de programación PHP, es posible resolver la situación si se implementa cada manejador de una petición ECCP como una máquina de estado finito. Cada conexión está ociosa, o en un estado del manejador. Cada estado debe corresponder a un acceso a la base de datos, el cual debe de realizarse en un proceso separado iniciado con el único propósito de bloquearse esperando la respuesta desde la base de datos. Mientras se espera la respuesta de la base de datos, el proceso intenta atender la petición de otro cliente. Aunque es posible

implementar esta solución, los detalles de los manejadores de requerimientos se vuelven engorrosos fácilmente.

Sin embargo, existe una arquitectura alterna de programación disponible en el lenguaje de programación Python. Los programas escritos en Python pueden usar la biblioteca de red eventlet (<http://eventlet.net>) para organizar la creación de hilos verdes (green threads/greenlets). Un hilo verde es un hilo de ejecución cuya ejecución o interrupción está a cargo de la lógica del programa que lo creó, en lugar del sistema operativo. En el caso de eventlet, un hilo verde sigue su ejecución hasta que está a punto de ejecutar una acción de lectura o escritura a través de un socket de red, o una operación de base de datos. En ese momento, el hilo verde agrega el socket a una lista de sockets que se monitorean por actividad, e inmediatamente cede el control al siguiente hilo verde disponible.

Un programa que usa eventlet tiene las siguientes características:

- Cada rutina que se ejecuta dentro de un hilo verde se programa como si el socket (u operación de base de datos) fuese a bloquear la ejecución hasta terminar la operación de entrada/salida. La biblioteca eventlet se encarga del acto de



ceder el control a otro hilo verde y de monitorear la actividad de los sockets. No es necesario reescribir el programa como una máquina de estados.

- Un hilo verde administrado por eventlet sólo cede el control al momento de iniciar una operación de socket. Esto vuelve mucho más fácil razonar sobre el estado compartido entre todos los hilos, a diferencia de la programación multihilo ordinaria, la cual requiere candados u otros mecanismos de sincronización explícita para proteger el estado compartido.
- Debido a que el estado es compartido, no es necesario enviar mensajes para implementar comunicación entre varios procesos. Esto simplifica la programación en comparación al modelo actual del programa marcador.

Por estas razones, la reescritura del programa marcador en Python con la biblioteca eventlet es una posible solución al problema potencial de bajo rendimiento debido a la serialización de las peticiones ECCP por múltiples clientes. Esta solución sería tomada luego de agotar las opciones de optimización de las peticiones a la base de datos usando índices.

## CONCLUSIONES Y RECOMENDACIONES

### Conclusiones

1. La construcción del sistema Call Center sobre Elastix fue capaz de alcanzar el objetivo general de reducir las ineficiencias y errores humanos en la atención de las llamadas a los clientes. Esto se evidenció en la empresa piloto luego de la implementación y se verificó independientemente en el caso de estudio de IPTegra. [13]
2. El uso de la aplicación AgentLogin() de Asterisk para mantener al agente siempre listo para recibir una llamada, sumado a la generación automática de llamadas, alcanzó el objetivo de maximizar el número de llamadas atendidas por agente y de minimizar los tiempos muertos.
3. El uso de la base de datos para almacenar los números a marcar es un elemento fundamental para cumplir con el objetivo de eliminar los errores humanos en la marcación y el seguimiento de

la información de la llamada.

4. El objetivo del monitoreo en tiempo real de la actividad de las campañas puede implementarse a través de peticiones de la base de datos, pero fue necesaria una definición formal de un protocolo orientado a eventos (el ECCP) para que este monitoreo fuese eficiente.
5. El software Asterisk demostró ser suficientemente flexible para soportar la implementación de las características del Call Center. Esto se debe al conjunto de aplicaciones implementadas en Asterisk, sumado a la interfaz AML que permite control por parte de un agente externo.
6. La aplicación Queue() de Asterisk resultó suficiente para el propósito de asignar las llamadas generadas a los agentes. Sin embargo, el comportamiento de esta aplicación de Asterisk no es suficientemente flexible si se requiere que la decisión de asignación dependa de condiciones indicadas por el programa marcador. Como ejemplo, se requiere programación especial para implementar la característica de llamada agendada a un agente específico.
7. La aplicación AgentLogin() de Asterisk se debe emplear únicamente para el escenario en el cual el agente debe de recibir

una llamada de forma inmediata, sin la demora o la decisión de contestar el teléfono. Para cualquier otro escenario, una extensión ordinaria es más cómoda para el usuario.

8. La capacidad de configuración provista por FreePBX (incluido como parte de Elastix) es apenas suficiente para adecuar las colas, las extensiones y los agentes. Esto se debe a que para que el Call Center funcione correctamente es necesario preparar los agentes de la cola de forma especial para indicar que son agentes permanentes con el canal Agent. Para una integración más profunda del Call Center con el resto del sistema, se requiere que el código del Call Center modifique directamente las tablas que usa FreePBX para su configuración.
9. Las bibliotecas del framework de Elastix proveen un modelo de programación consistente, pero algunas funcionalidades pueden mejorarse. Por ejemplo, el modelo de privilegios soporta grupos con módulos permitidos y denegados, pero no hay un mecanismo estandarizado para que una aplicación de terceros como el Call Center cree un nuevo grupo. Esto conduce a que el administrador del sistema administre los grupos y permisos manualmente para construir la estructura de privilegios necesaria para el Call Center.
10. El código del marcador del Call Center es capaz de ser mejorado o incluso reescrito y remodelado sin que se presenten problemas

en instalaciones existentes. Por el contrario, la estructura de la base de datos es difícil de cambiar debido a la necesidad de preservar los datos existentes. Esto dificulta la implementación de características nuevas.

### **Recomendaciones**

1. Se recomienda investigar un mecanismo con el cual una aplicación pueda definir una cola con una condición de asignación a un agente completamente programable. Este mecanismo puede servir a los sistemas como el Call Center y otros proyectos que se benefician del control sobre cuál agente atiende una llamada.
2. Se recomienda implementar un proceso por el cual la lista de números a marcar para una campaña pueda obtenerse, no desde la base de datos local alimentada desde un archivo CSV, sino desde un sistema externo. El servidor externo podría soportar XML-RPC, SOAP o un protocolo de base de datos. Se puede ofrecer la opción de cargar todos los números al inicio de la campaña, o de cargar porciones del listado a medida que progresa la campaña.
3. Actualmente la información que se puede mostrar sobre una llamada de una campaña entrante es limitada (sólo nombres y

cédula), por lo que es recomendable implementar un método para que el programa marcador pueda obtener información sobre el cliente correspondiente al número de la llamada entrante. Este método puede depender de una base de datos local, o ser consultado desde otro servidor.

4. Para futuras versiones, se requiere construir un módulo para la asignación de agentes a las colas. Este módulo necesariamente duplicará parte de la funcionalidad de colas de FreePBX, pero puede hacer cumplir los requerimientos del canal Agent para los agentes estáticos.
5. Para mejorar la utilización de los números, se debe implementar una manera para reciclar los números de teléfono de una campaña saliente. Esta característica debe permitir la selección de algunos números telefónicos de una campaña terminada, y construir una nueva campaña con la selección. El criterio de selección puede ser el estado de éxito o fallo del marcado, una respuesta seleccionada en el formulario llenado, o una combinación de lo anterior.
6. Es recomendable implementar la capacidad de que el supervisor de la campaña pueda escuchar la conversación en progreso entre un agente y un cliente. Esta capacidad permite que el supervisor pueda evaluar la calidad del diálogo de los agentes, e intervenir de

inmediato si ocurren problemas. El lugar más adecuado para implementar tal funcionalidad es en el reporte de Monitoreo de Campañas.

7. Se deben analizar los puntos mencionados en la sección 6.4 para el rediseño de la base de datos, con el objetivo de construir una nueva versión del sistema Call Center. Esta versión nueva sería capaz de implementar correctamente las nuevas características mencionadas en la sección 6.2, y evitando o solucionando los obstáculos indicados en la sección 6.3. Esta nueva versión debe proveer un mecanismo de migración para los datos almacenados en las instalaciones existentes del Call Center.

## **ANEXOS**



# ANEXO 1

## MANUALES DE USUARIO

### Manual de usuario - Administrador

#### Instalación

El paquete de software del Elastix Call Center puede ser instalado de dos formas distintas:

- A través de la interfaz web de Elastix, en el módulo Addons.
- A través del programa de línea de comando “yum”.

Los requisitos de instalación del paquete Elastix Call Center son los siguientes:

- Un sistema Elastix en funcionamiento normal.
- Una conexión a internet que sea usable por el sistema Elastix.

Para realizar la instalación como Addon:

1. Ingresar a la interfaz web de Elastix con un navegador web, con credenciales de administrador.
2. Completar el proceso de registro de Elastix, si no se lo ha hecho todavía.
3. Elegir el módulo de Addons en el menú del GUI de Elastix.

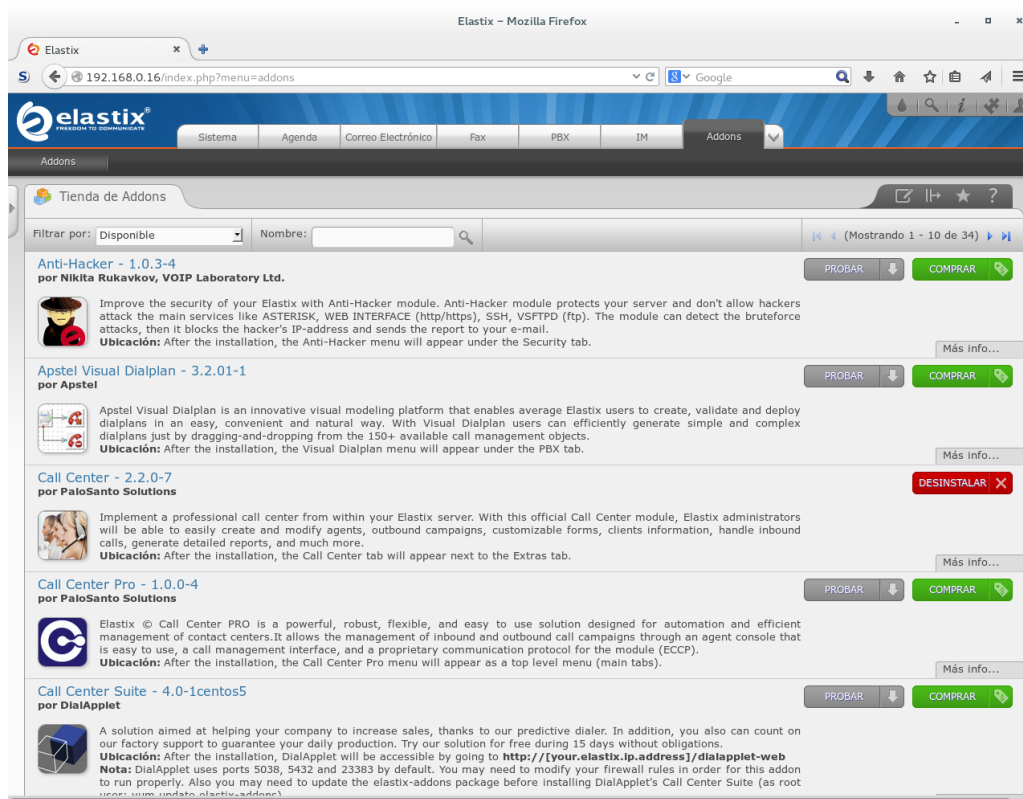


Figura 1: Interfaz de módulo de Addons

4. Buscar el módulo de Elastix Call Center en el listado de addons presentado.
5. Hacer clic en el botón Instalar correspondiente al Elastix Call Center.
6. La interfaz web mostrará una barra de descarga. Luego de que la barra se complete, el Elastix Call Center estará instalado.

Para realizar la instalación como un RPM ordinario:

1. Iniciar una sesión SSH hacia el computador que ejecuta Elastix, bajo el usuario root.

2. Ejecutar el siguiente comando: `yum install elastix-callcenter`
3. Aprobar la confirmación de descarga, y si es necesario, la instalación de la clave de verificación RPM.
4. El paquete RPM será descargado e instalado.

A través de cualquiera de los dos métodos, la instalación del programa incluye la creación de los módulos de la interfaz web, la instalación del esquema de la base de datos con registros iniciales y la instalación para el arranque automático (pero NO el inicio) del marcador telefónico. Todos los módulos están disponibles para el grupo administrador, pero no para otros grupos.

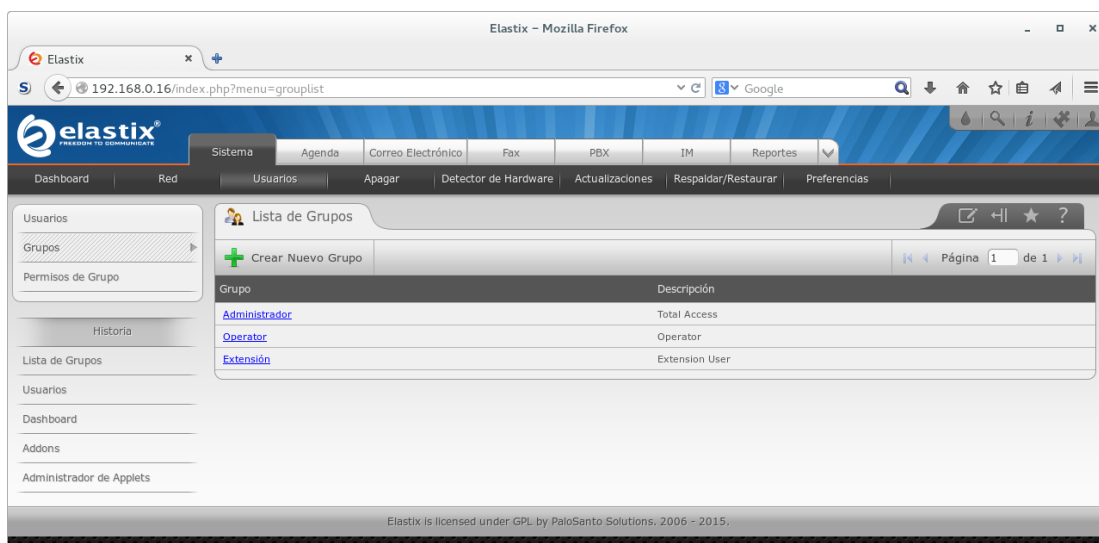


Figura 2: Interfaz de módulo de Grupos

Para la configuración de acceso de los módulos:

1. Decidir un plan de usuarios y grupos para los agentes y supervisores del Call Center. Este plan puede consistir en un usuario web Elastix para cada agente y supervisor, o un solo usuario compartido para todos los agentes más otro para todos los supervisores.
2. En el módulo Grupos del menú Sistema, cree un grupo para todas las cuentas de agentes. Este grupo debe tener acceso al módulo de Consola de Agente.
3. En el mismo módulo Grupos, cree un grupo para todas las cuentas de supervisores. Este grupo debe tener acceso a todos los reportes.
4. A criterio del administrador, se pueden crear otros grupos para acceso al menú de breaks, de administración de agentes, y otros módulos del Call Center.

### **Configuración de Elastix**

Antes de la configuración específica del Call Center, el sistema Elastix debe ser preparado para su funcionamiento como una IP-PBX ordinaria. Los siguientes componentes se deben de configurar y verificar antes de continuar con la preparación del Call Center:

- Al menos una cola telefónica, la cual servirá para acumular las llamadas en espera de ser atendidas por los agentes. Las colas para campañas entrantes deben de estar separadas de las colas que se

dedicarán a las campañas salientes.

- Al menos una troncal para la generación o recepción de llamadas desde el Elastix hacia el exterior. Esta troncal puede usar cualquier tecnología disponible (SIP, IAX, DAHDI u otra).
- Al menos una ruta saliente que permita un marcado desde el Elastix que haga uso de la troncal configurada anteriormente.
- Para atención de llamadas entrantes, se requiere al menos una ruta entrante que enlace el DID asignado por el proveedor de telefonía con una cola configurada anteriormente para campañas entrantes.
- Extensiones VoIP para los agentes, en cantidad al menos igual al número de agentes simultáneos en atención. Las extensiones VoIP deben ser de tecnología SIP o IAX2.

Todos estos parámetros se configuran desde el módulo de FreePBX embebido en el módulo Configuración del menú PBX.

### **Configuración de agentes**

El administrador del Call Center debe decidir los tipos de agentes que deben ser usados según el estilo de atención deseado para el sistema:

- Agentes estáticos, los cuales mantienen una llamada durante toda la sesión de atención, y escuchan música de espera hasta que se le asigna una llamada, sin opción a decidir si contestan o no.

- Agentes dinámicos, los cuales mantienen el teléfono cerrado durante la sesión de atención hasta que reciben una llamada asignada por el sistema. Entonces su extensión timbrará y el agente debe contestar la llamada para atender al cliente.

Los dos tipos de agentes pueden mezclarse libremente en una misma cola y campaña, en caso necesario.

### Para configurar agentes estáticos:

1. Elegir el módulo Agentes del menú Opciones de Agente.

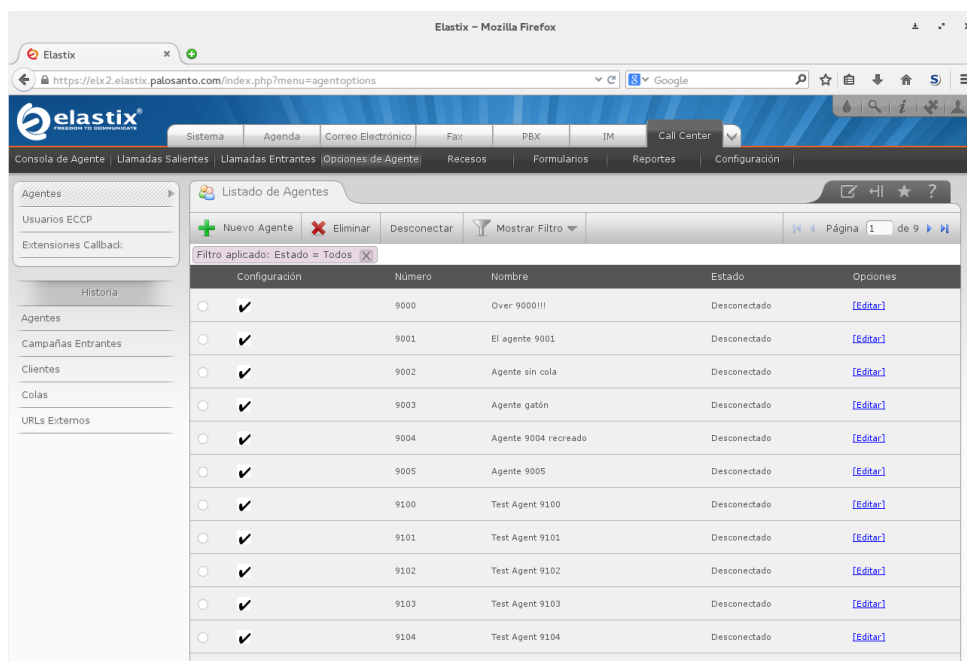


Figura 3: Interfaz de módulo de Agentes con lista de agentes

2. Pulsar en el botón Nuevo Agente
3. Llenar el nombre de agente, su número de agente y su número de

clave.

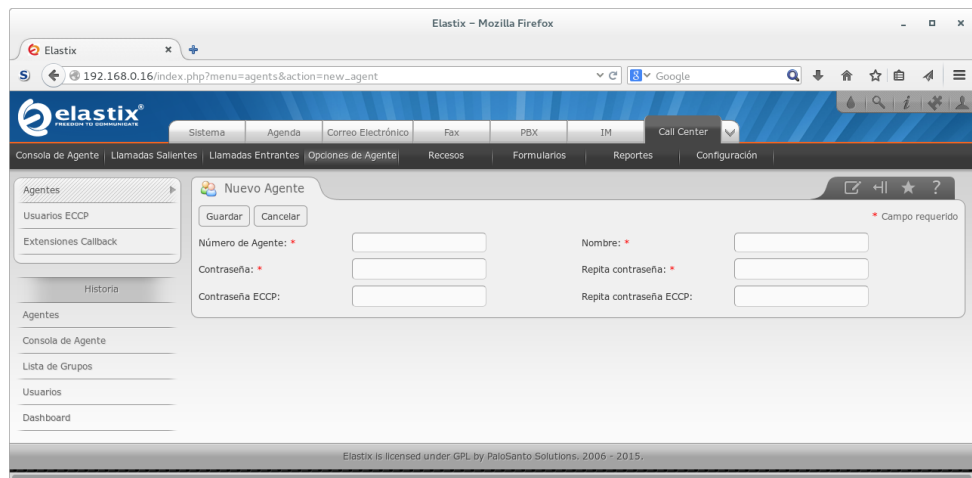


Figura 4: Interfaz de módulo de Agentes en creación de nuevo agente

4. Repetir para todos los agentes necesarios.
5. Elegir el módulo de FreePBX embebido del menú PBX, submódulo Colas.
6. Para cada cola en la que debe atender un agente, se debe agregar en el recuadro de texto de Agentes estáticos el texto Annnn, donde nnnn es el número de agente asignado previamente. Por ejemplo, para el agente 9000 se agrega el texto "A9000" al recuadro de Agente estáticos.

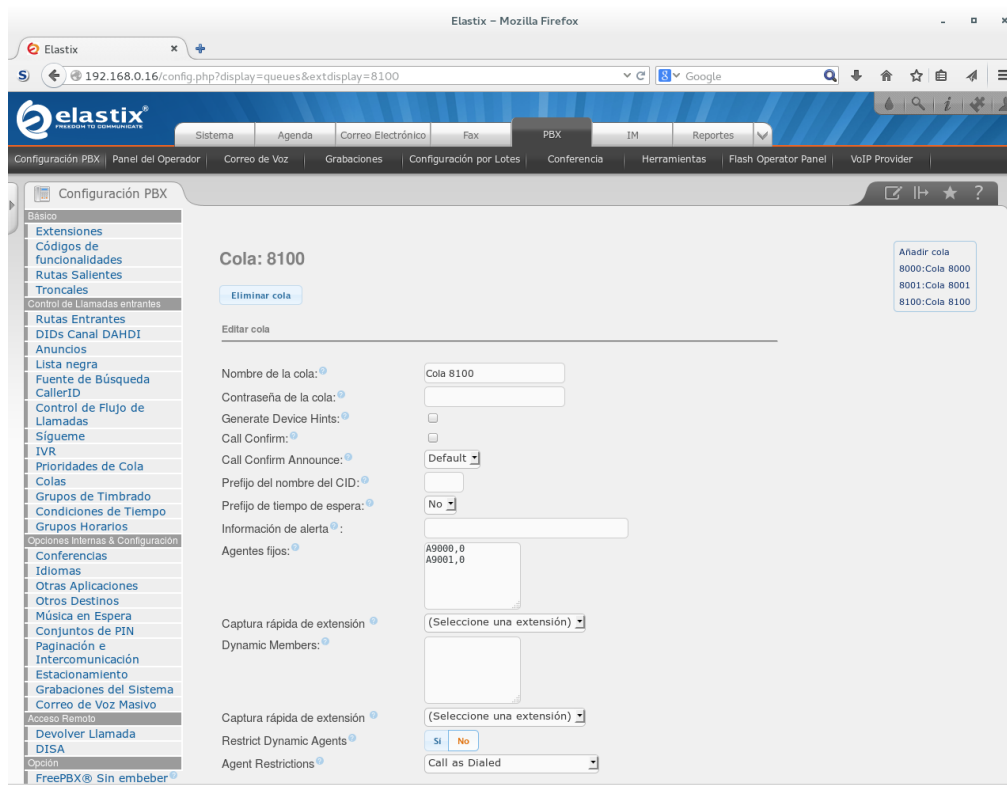


Figura 5: Interfaz de FreePBX embebido mostrando módulo de configuración de cola

7. Guardar y aplicar los cambios en FreePBX embebido.

Para configurar agentes dinámicos:

1. Elegir el módulo Extensiones CallBack del menú Opciones de Agente.



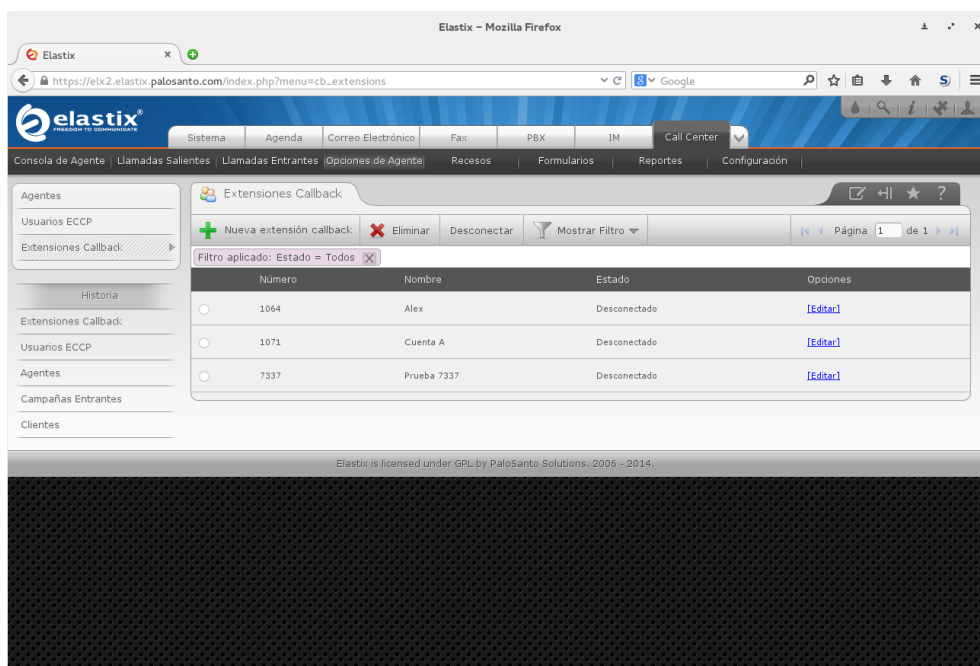


Figura 6: Interfaz de módulo de Extensiones Callback con lista de extensiones

2. Pulsar en el botón Nueva Extensión Callback.
3. Llenar el nombre del agente, elegir una extensión libre, y asignar una contraseña.

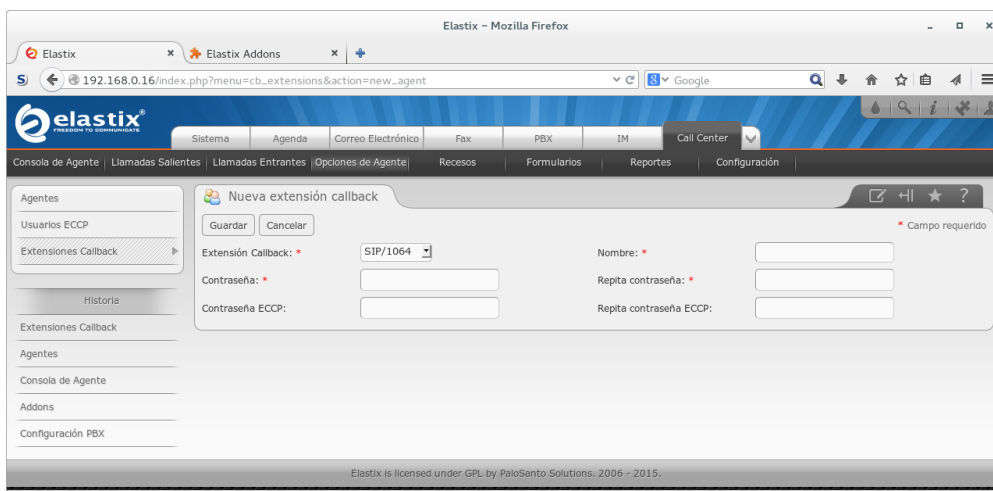


Figura 7: Interfaz de módulo de Extensiones Callback en elección de nueva extensión

4. Repetir para todos los agentes necesarios.
5. Elegir el módulo de FreePBX embebido del menú PBX, submódulo Colas.
6. Para cada cola en la que debe atender un agente, se debe agregar en el recuadro de texto de Agentes dinámicos el texto Snnnn o Innnn, donde nnnn es el número de extensión correspondiente al agente, y S o I corresponde a la extensión SIP o IAX2, respectivamente.
7. Guardar y aplicar los cambios en FreePBX embebido.

Luego de la creación de los agentes, se recomienda que los agentes ingresen con su usuario de Elastix y a continuación intenten iniciar sesión en la Consola de Agente para verificar que el procedimiento se puede realizar correctamente.

### **Creación de descansos autorizados**

El administrador (o el supervisor de Call Center, en caso de estar autorizado) debe crear los descansos permitidos a los agentes durante su sesión de atención. Para definir los descansos se debe:

## 1. Elegir el módulo Recesos del menú Call Center.

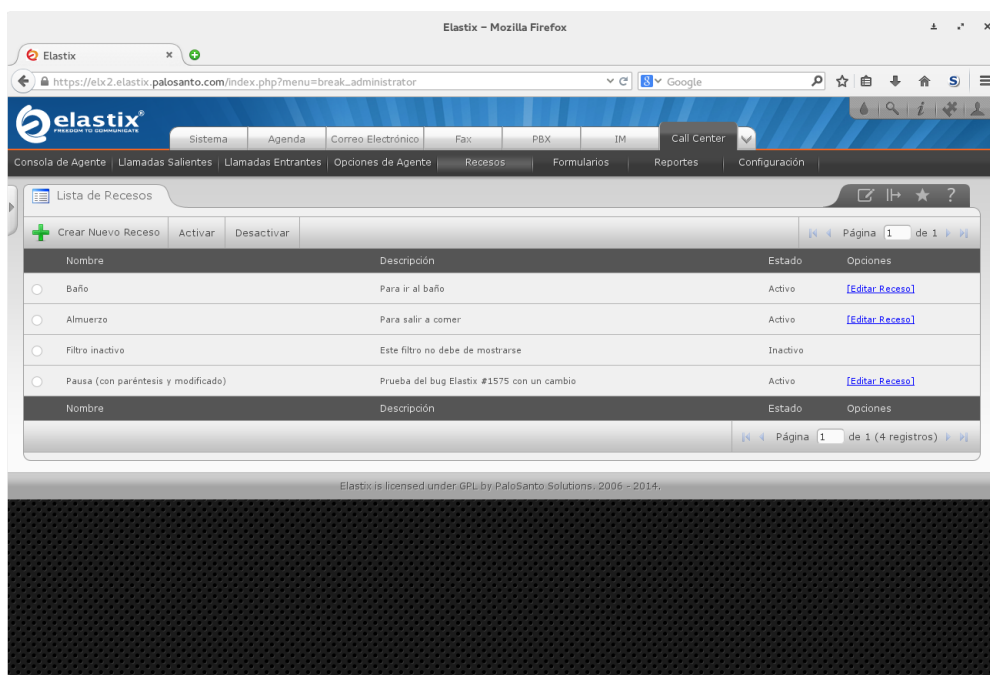


Figura 8: Interfaz de módulo de Recesos

2. Pulsar en el botón Crear Nuevo Receso.
3. Ingresar el nombre del receso a crear.
4. Repetir para todos los tipos de descansos que se deben autorizar.

## Creación de formularios

Para poder recoger datos en las campañas (entrantes o salientes) se requiere definir un formulario que indique la estructura de los datos recolectados. Un mismo formulario puede ser compartido por más de una campaña. Un formulario puede tener los siguientes tipos de campos:

- Etiqueta: un texto estático que aparece en el formulario sin opción a recoger datos.
- Texto: un recuadro de texto de una sola línea, para preguntas con respuestas cortas.
- Área de texto: un recuadro de texto que permite múltiples líneas o textos largos, para preguntas con respuestas largas, o anotaciones.
- Lista: una lista desplegable con varias opciones, para preguntas con respuesta de opción múltiple.
- Fecha: un campo con un calendario para elegir una fecha específica.

Los pasos a seguir para la creación de un formulario son:

1. Elegir el módulo Diseño de Formularios del menú Formularios.

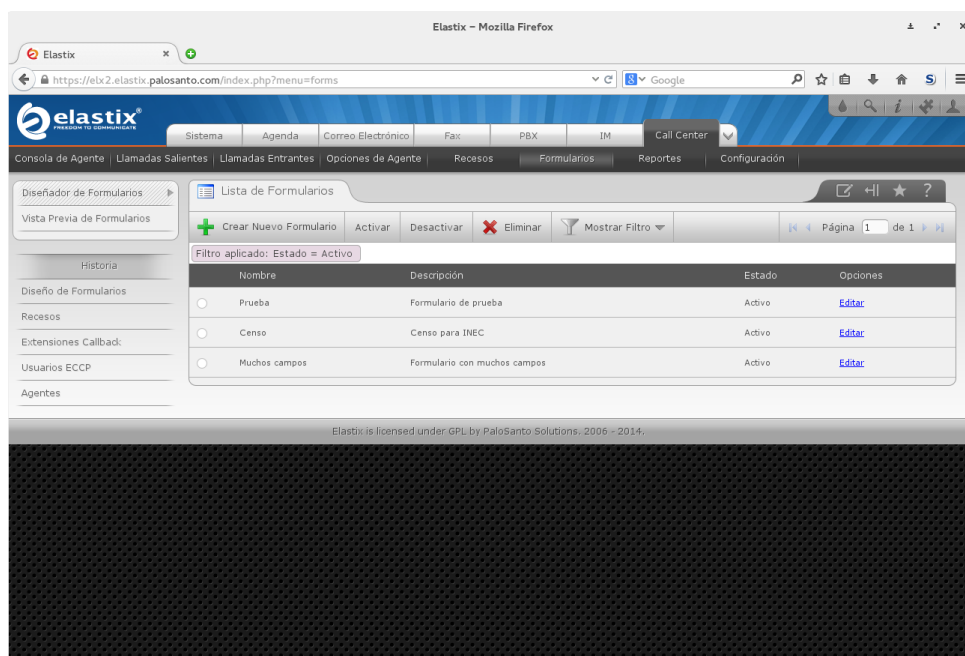


Figura 9: Interfaz de módulo de Diseñador de Formularios con lista de

## formularios

2. Pulsar en el botón Nuevo Formulario.
3. Llenar el nombre y descripción del formulario.
4. En la lista proporcionada, elegir un texto para el nombre de cada campo, el tipo de campo, y si es necesario, las opciones de la lista. Las filas de los campos del formulario pueden ser arrastradas para cambiar el orden de los campos. El botón con el signo menos a la derecha de cada fila elimina la fila del formulario. La última fila tiene un botón con el signo más para añadir un nuevo campo.

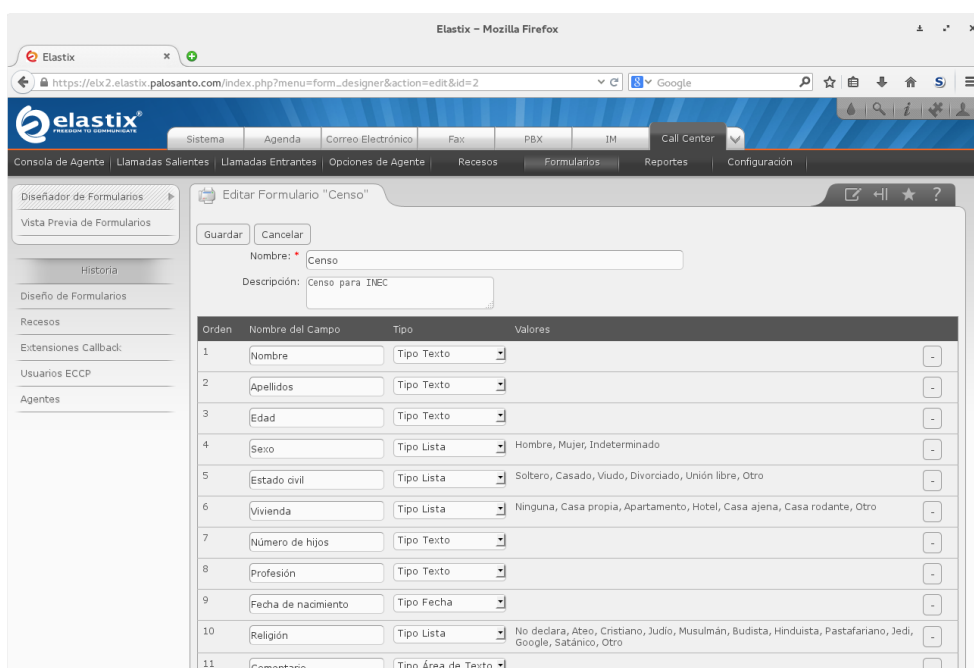


Figura 10: Interfaz de módulo de Diseñador de Formularios en creación de formulario

#### 5. Guardar las modificaciones.

Estos pasos se deben repetir para todos los formularios planeados para las campañas.

### **Creación de una campaña saliente**

El administrador (o el supervisor de Call Center, en caso de estar autorizado) debe crear las nuevas campañas salientes con la lista de números a marcar. Para cada uno de los números, el Elastix Call Center generará una llamada que conecte este número telefónico con la cola de los agentes, donde la llamada será asignada a un agente disponible.

Para crear una campaña saliente, se requieren tener listos los siguientes elementos:

- Un archivo CSV que represente una base de datos de números a marcar. Estos números deben estar en el formato de marcación que usaría una extensión ordinaria para marcar el número a través de las rutas salientes configuradas en el Elastix. Por ejemplo, si la ruta saliente requiere marcar 9 como prefijo para elegir la troncal al exterior, la base de números debe incluir este 9 como prefijo de todos los números. Cada renglón del archivo CSV debe tener el número a marcar como primera columna. El resto de columnas se dedican a

atributos que serán listados en la Consola de Agente al ser asignada al agente la llamada correspondiente. Opcionalmente, el primer renglón del archivo CSV puede dedicarse a etiquetas que indican el nombre del atributo en cada columna.

- Al menos una cola que NO esté reservada para llamadas entrantes, con agentes disponibles.
- Al menos un formulario para recolección de datos.
- Un guión de atención.

La creación de la campaña saliente requiere de los siguientes pasos:

- Elegir el módulo Campañas del menú Llamadas Salientes.

The screenshot displays the Elastix web interface. The top navigation bar includes 'Sistema', 'Agenda', 'Correo Electrónico', 'Fax', 'PBX', 'IM', and 'Call Center'. The main content area shows the 'Listado de campañas' module with a table of campaigns. The table columns are: Nombre, Rango Fechas, Horario Diario, Intentos, Troncal, Cola, Llamadas completadas, Promedio, Estado, and Opciones. The data rows are as follows:

Nombre	Rango Fechas	Horario Diario	Intentos	Troncal	Cola	Llamadas completadas	Promedio	Estado	Opciones
<input type="radio"/> Campaña de prueba 2	2011-01-28 - 2012-01-28	00:00:00 - 23:59:00	5	(Dialplan)	8000	12	157	Inactiva	[Editar] [Datos CSV]
<input type="radio"/> Campaña de prueba 3	2011-03-14 - 2011-04-14	00:00:00 - 23:59:00	5	(Dialplan)	8000	17	188	Inactiva	[Editar] [Datos CSV]
<input type="radio"/> Campaña de prueba 4	2011-07-29 - 2013-12-31	00:00:00 - 23:59:00	5	(Dialplan)	8000	2	137	Terminada	[Editar] [Datos CSV]
<input type="radio"/> Prueba campaña noviembre 17	2011-11-17 - 2012-11-17	00:00:00 - 23:59:00	5	(Dialplan)	8000	1	1,066	Terminada	[Editar] [Datos CSV]
<input type="radio"/> Gatito	2012-00-17 - 2013-12-31	00:00:00 - 23:59:00	5	(Dialplan)	8000	45	176	Terminada	[Editar] [Datos CSV]

Figura 11: Interfaz de módulo de Campañas Salientes con lista de campañas

- Pulsar el botón de Crear Nueva Campaña.
- Llenar los campos requeridos de nombre, fechas de validez de la campaña, horario diario de la campaña, número de cola, formulario(s) a usar, guión de atención y archivo CSV con números a marcar. El resto de campos pueden dejarse en sus valores por omisión.

The screenshot shows the 'Nueva campaña' (New Campaign) form in the Elastix interface. The form includes the following fields and options:

- Nombre:** Text input field.
- Rango Fechas:** Two date pickers for 'Inicio' (Start) and 'Fin' (End).
- Horario Diario:** Two sets of time pickers for 'Hora inicial' (Start time) and 'Hora final' (End time).
- Formulario:** A dropdown menu currently set to 'Censo básico', with 'Administrar Formularios' link below it.
- URLs Externos:** A dropdown menu set to '(No external URL)', with 'Administrar URLs Externos' link below it.
- Troncal:** A dropdown menu set to '(Por Plan Marcado)', with 'Administrar Troncales' link below it.
- Max. canales a usar:** Text input field with value '23'.
- Contexto:** Text input field with value 'from-internal'.
- Cola:** Text input field with value '8000 Cola 8000', with 'Administrar Colas' link below it.
- Intentos:** Text input field with value '5'.
- Archivo de Llamadas:** A file selection button labeled 'Examinar...' with the message 'No se ha seleccionado ningún archivo.' below it.
- Codificación de Archivo de Llamadas:** A dropdown menu set to 'UTF-8 - Universal'.
- Guión:** A rich text editor with a toolbar and a text area.

Figura 12: Interfaz de módulo de Campañas Salientes en creación de campaña

- Guardar los campos de la nueva campaña.



Los números de la campaña empezarán a marcarse al cumplirse las siguientes condiciones:

- Fecha del sistema entre intervalo de validez de la campaña.
- Hora del sistema dentro del horario de la campaña.
- Al menos uno de los agentes asignados a la cola ha iniciado su sesión y está ocioso.

### **Creación de una campaña entrante**

El administrador (o el supervisor de Call Center, en caso de estar autorizado) debe crear las nuevas campañas entrantes. Toda llamada que ingrese a la cola de la campaña entrante quedará en espera hasta ser asignada a un agente. Al ser atendida la llamada, el agente verá el guión y los formularios asignados a la campaña para ser llenados.

Para crear una campaña entrante, se requieren tener listos los siguientes elementos:

- Al menos una cola que esté reservada para llamadas entrantes, con agentes disponibles. Para reservar una cola para llamadas entrantes, se debe entrar al módulo Colas del menú Llamadas Entrantes y elegir la cola a reservar.

- Al menos una ruta entrante que conduzca llamadas recibidas por el sistema hacia la cola de los agentes. Esta ruta entrante se configura en el módulo de FreePBX embebido.
- Al menos un formulario para recolección de datos.
- Un guión de atención.

La creación de la campaña entrante requiere de los siguientes pasos:

- Elegir el módulo Campañas Entrantes del menú Llamadas Entrantes.

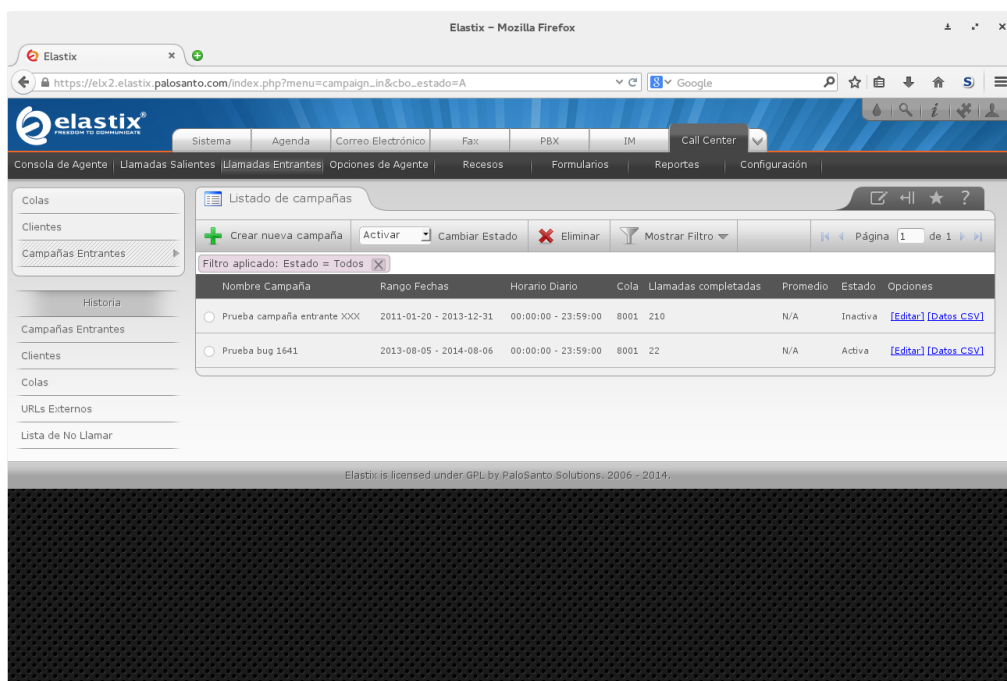


Figura 13: Interfaz de módulo de Campañas Entrantes con lista de campañas

- Pulsar el botón de Nueva Campaña.
- Llenar los campos requeridos de nombre, fechas de validez de la

campaña, horario diario de la campaña, número de cola, formulario(s) a usar y guión de atención. El resto de campos pueden dejarse en sus valores por omisión.

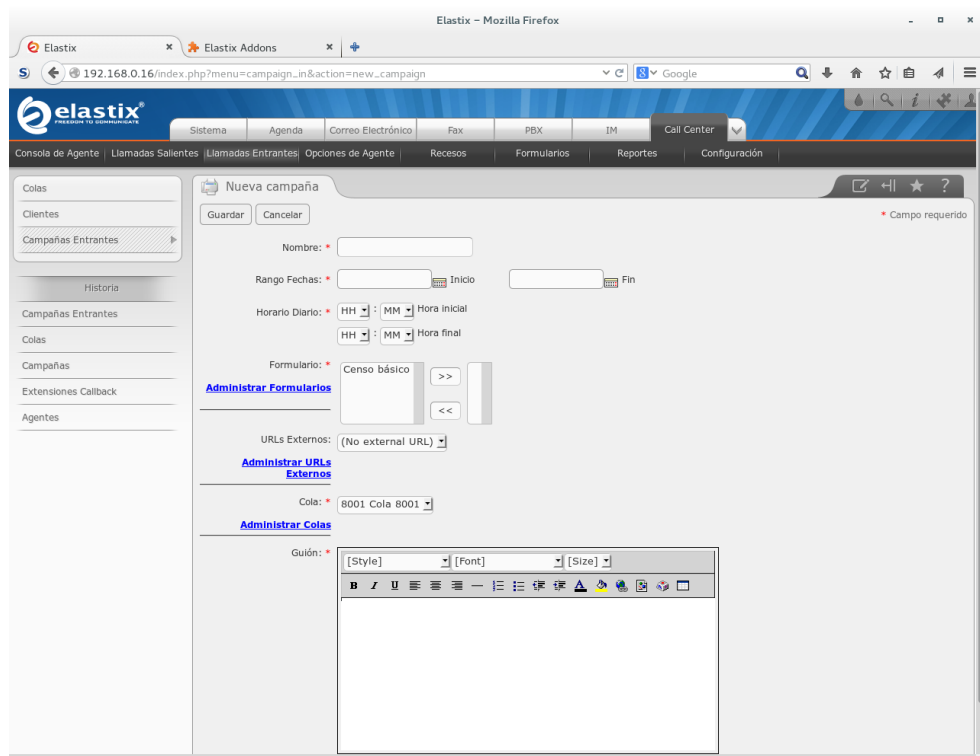


Figura 14: Interfaz de módulo de Campañas Entrantes en creación de campaña

- Guardar los campos de la nueva campaña.

La campaña entrante se usará al cumplirse las siguientes condiciones:

- Fecha del sistema entre intervalo de validez de la campaña.
- Hora del sistema dentro del horario de la campaña.

- Al menos uno de los agentes asignados a la cola ha iniciado su sesión y está ocioso.
- Una llamada del exterior termina entrando en la cola que fue reservada para la campaña.

### **Descarga de datos recolectados**

Una vez terminada la campaña (saliente o entrante), los datos recolectados se descargan de la siguiente manera:

- Elegir el módulo Campañas o Campañas Entrantes, según corresponda.
- Localizar en la lista la campaña de la cual se descargarán los datos.
- Hacer clic en el enlace “Datos CSV”, lo cual inicia la descarga de los datos.

Los datos recolectados aparecen como una lista de los números marcados, uno por renglón, con los siguientes campos:

- Número marcado o recibido.
- Estado final del intento de marcado.
- Número del agente que atendió la llamada.
- Fecha de inicio de la atención de la llamada.
- Duración de la atención, en segundos.

- Valor del campo Uniqueid asignado por Asterisk, para identificación de la llamada.
- Código y texto de fallo de llamada, si la llamada no fue exitosa.
- Columnas de los atributos de la llamada, los mismos que fueron indicados por el archivo. CSV usado para cargar los datos inicialmente.
- Columnas de los campos de cada formulario asignado a la campaña.

## **Manual de usuario - Agente**

### **La consola de agente**

El módulo de la Consola de Agente contiene la información que el agente requiere para atender cada llamada asignada por el sistema Call Center. Este módulo tiene dos modos principales: el modo de login y el modo de atención. A su vez, el modo de login se comporta de manera distinta para el agente estático y el agente dinámico.

Los requisitos para que un agente pueda iniciar una sesión de atención son:

- Una cuenta de usuario web en el sistema Elastix. Esta cuenta debe pertenecer a un grupo que esté autorizado al módulo de la Consola de Agente.
- Un navegador web.
- Una extensión telefónica configurada en el sistema Elastix, con tecnología SIP o IAX2.
- Un teléfono VoIP (sea un teléfono físico o un softphone) registrado correctamente con la extensión telefónica configurada.
- Para un agente estático, se requiere configurar el número de agente estático en el módulo Agentes del sistema Call Center.
- Para un agente dinámico, se requiere registrar la extensión VoIP del

agente como un agente dinámico en el módulo Extensiones Callback.

Para iniciar una sesión de atención como agente estático:

- Iniciar la sesión de usuario en Elastix y navegar al módulo de Consola de Agente.
- Verificar que la casilla de verificación de Callback Login está desactivada.

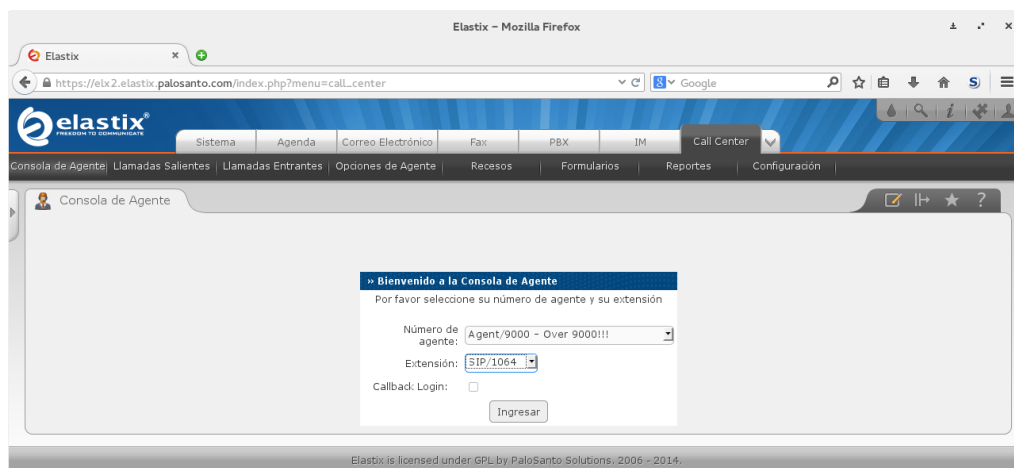


Figura 15: Interfaz del módulo de Consola de Agente en modo de login de agente estático

- Elegir el número de agente configurado en la lista desplegable Número de Agente.
- Elegir la extensión VoIP asignada al agente en la lista desplegable Extensión.
- Presionar el botón Ingresar.

- La extensión seleccionada debe timbrar en este momento. El agente debe entonces contestar esta llamada.
- En la llamada se reproduce una grabación que indica que se debe ingresar la contraseña numérica a través del teclado del teléfono, seguido de la tecla numeral (#). El agente debe seguir estas instrucciones.
- Si la contraseña es correcta, se reproduce un audio indicando que el agente está logoneado. Al mismo tiempo la interfaz de la Consola de Agente cambia al modo de atención.
- El agente debe mantener activa la llamada en su teléfono durante toda la sesión de atención. El cuelgue del teléfono se considera un fin de sesión y hace que la interfaz de la Consola de Agente vuelva al modo de login.
- Para finalizar la sesión de atención, el agente estático debe cerrar su teléfono o pulsar en el botón Fin Sesión de la interfaz de atención.

Para iniciar una sesión de atención como agente dinámico:

- Iniciar la sesión de usuario en Elastix y navegar al módulo de Consola de Agente.
- Activar la casilla de verificación de Callback Login en caso necesario.



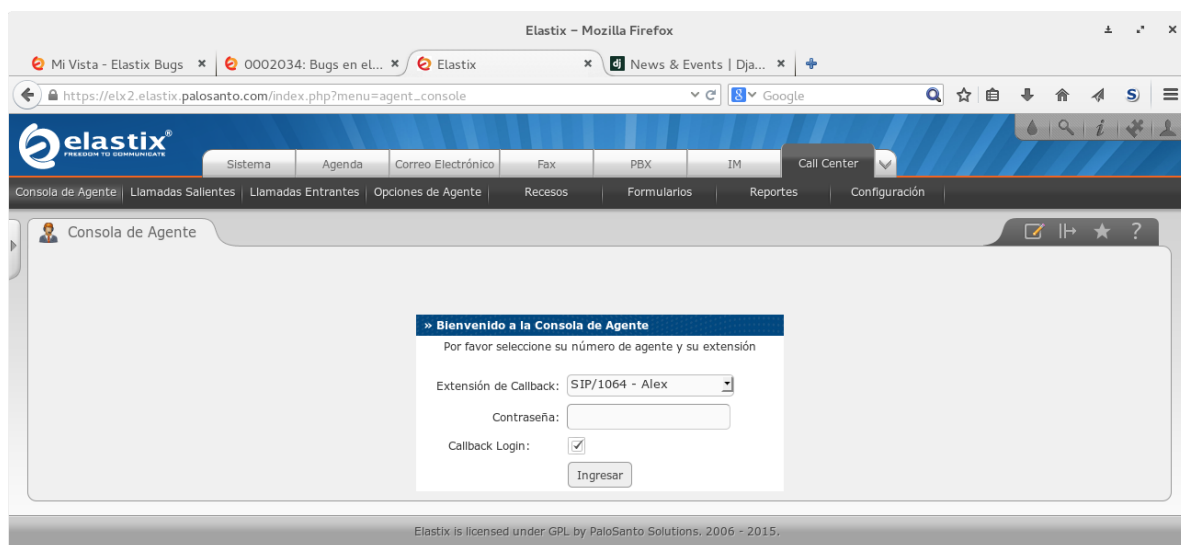


Figura 16: Interfaz del módulo de Consola de Agente en modo de login de agente dinámico

- Elegir la extensión asignada para el agente en la lista desplegable Extensión de Callback.
- Ingresar la contraseña asignada.
- Presionar el botón Ingresar.
- La interfaz de la Consola de Agente cambia de inmediato al modo de atención, sin llamar a la extensión seleccionada.
- El agente debe mantener su teléfono cerrado durante la sesión. Cuando se asigna una llamada, el teléfono timbrará y el agente debe contestar.
- Para finalizar la sesión de atención, el agente dinámico debe pulsar en el botón Fin Sesión de la interfaz de atención.

En caso de login exitoso, se muestra el modo de atención:

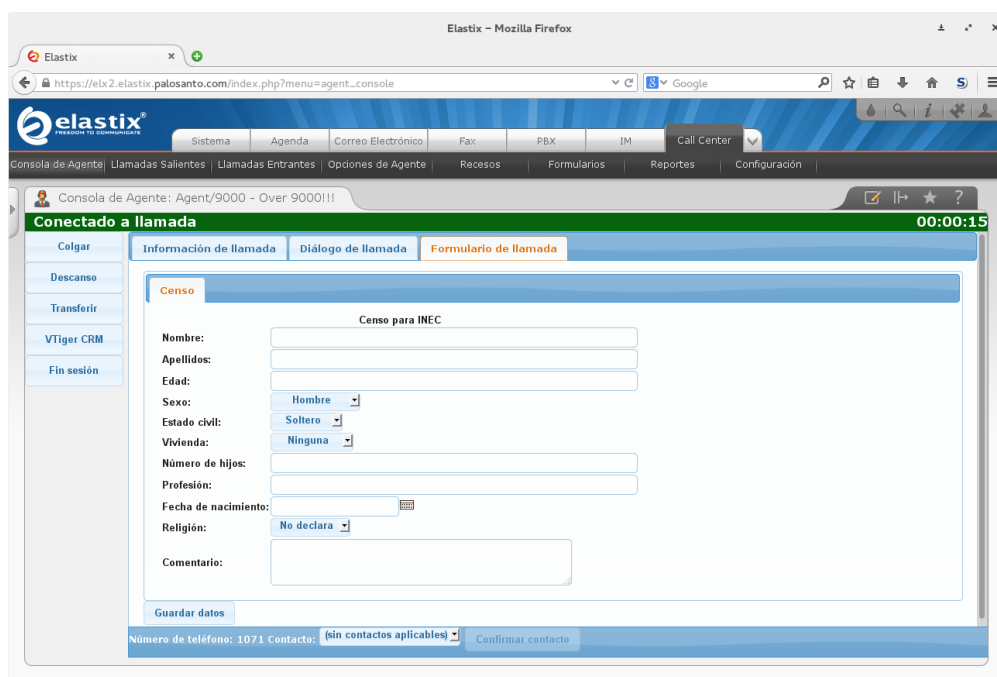


Figura 17: Interfaz del módulo de Consola de Agente en modo de atención.

El modo de atención contiene las siguientes áreas:

- La barra de estado en el borde superior del módulo.
- La barra de botones en el lado izquierdo.
- El área de pestañas con la información de la campaña.

La barra de estado muestra el estado actual de la sesión del agente. Los estados posibles son:

- Ocioso: la barra muestra el mensaje “No hay llamada activa” en fondo

azul.

- Ocupado: la barra muestra el mensaje “Conectado a llamada” en fondo verde. El cronómetro del lado derecho lleva la cuenta del tiempo invertido en atención de la llamada actual.
- En descanso: la barra muestra el mensaje “En descanso” en fondo rojo. El cronómetro del lado derecho lleva la cuenta del tiempo invertido en los descansos de la sesión actual.

La barra de botones contiene los siguientes controles que afectan a la sesión de atención actual:

- Colgar: finaliza la llamada del cliente asignado al agente.
- Descanso: muestra un diálogo con una lista desplegable de los recesos autorizados. Al elegir uno el agente entra en el modo de receso, en el cual no recibe llamadas.



Figura 18: Diálogo de selección de receso

- Transferir: muestra un diálogo con cuadro de texto para digitar una extensión a la cual transferir la llamada. Al aceptar, el agente se desconecta de la llamada actual.
- VTiger CRM: abre VTiger CRM en una nueva pestaña del navegador. Sólo aparece si VTiger CRM está instalado en Elastix.
- Fin Sesión: finaliza la sesión de atención del agente y cambia el módulo al modo de login.

El área de pestañas contiene tres pestañas que muestran información sobre la llamada:

- Información de la llamada: se muestran los atributos asociados a la llamada que fueron ingresados en el archivo CSV.

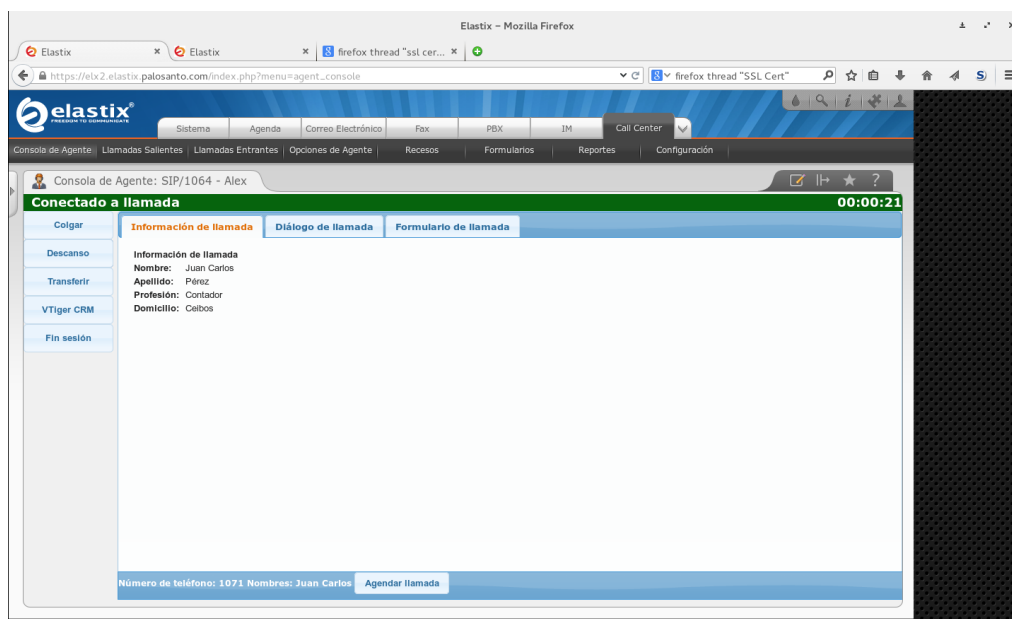


Figura 19: Interfaz de Consola de Agente con información de la llamada

- Diálogo de llamada: se muestra el guión de atención asociado a la campaña.

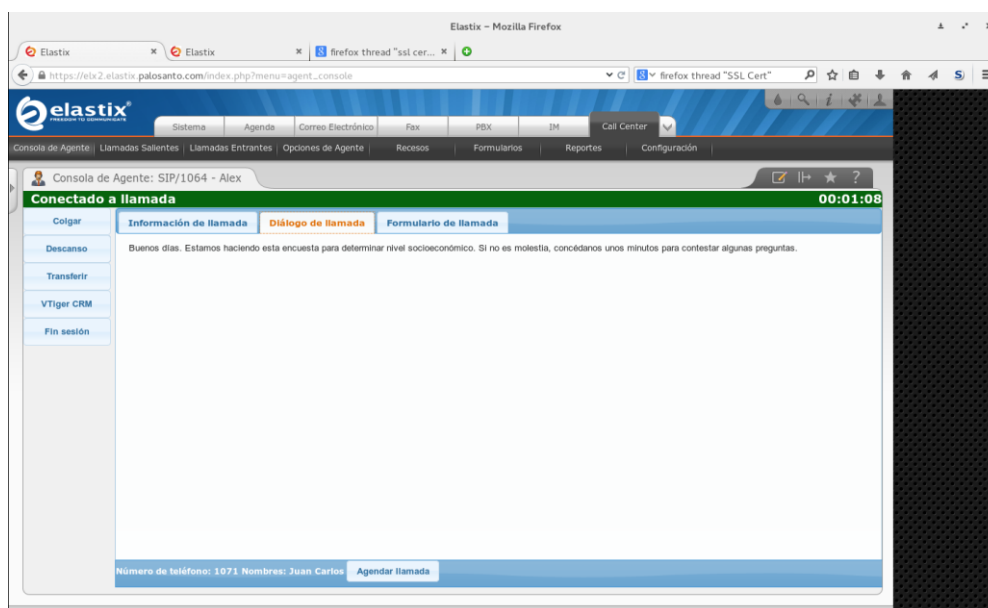


Figura 20: Interfaz de Consola de Agente con texto de guión

- Formulario de llamada: se muestra el formulario asociado con la campaña de la llamada actual, listo para ser llenado. La figura 17 muestra la interfaz de la Consola de Agente con el formulario de llamada.

Para las llamadas que pertenecen a una campaña saliente, se muestra en el borde inferior del módulo un botón para agendar una llamada. Este agendamiento muestra un diálogo donde se puede elegir fecha y hora de la llamada agendada y si el propio agente debe atender.

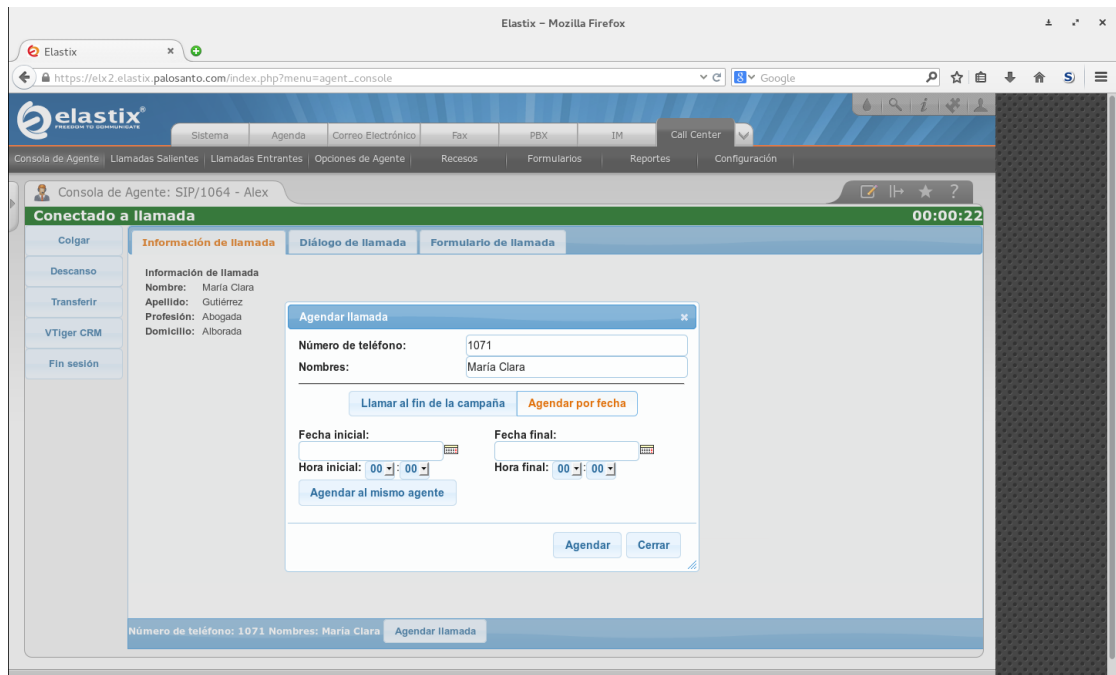


Figura 21: Interfaz de Consola de Agente con diálogo de agendamiento

## Manual de usuario - Supervisor

Para el monitoreo en tiempo real de la actividad del Call Center, hay dos módulos importantes:

- Monitoreo de campañas
- Monitoreo de agentes

### Monitoreo de actividad de campañas

The screenshot displays the Elastix web interface for monitoring campaigns. The browser address bar shows the URL: `https://elx2.elastix.palosanto.com/index.php?menu=campaign_monitoring#/details/incoming/2`. The interface includes a navigation menu with options like 'Sistema', 'Agenda', 'Correo Electrónico', 'Fax', 'PBX', 'IM', and 'Call Center'. The main content area is titled 'Monitoreo de Campaña' and shows details for a campaign named '(Campaña entrante) Prueba bug 1641'.

Key statistics displayed include:

- Fecha inicio:** 2013-08-05
- Fecha final:** 2014-08-06
- Horario:** 00:00:00 - 23:59:00
- Cola:** 8001
- Intentos:** N/D
- Total llamadas:** 22
- Llamadas en cola:** 0
- Llamadas conectadas:** 0
- Llamadas sin pista:** 0
- Llamadas abandonadas:** 17
- Llamadas terminadas:** 5
- Promedio Duración de Llamada:** 00:03:51
- Duración Máxima de Llamada:** 00:17:13

Below the statistics, there are two tables:

Llamadas siendo marcadas:				Agentes:				
Estado	Núm. Telf.	Troncal	Desde	Agente	Estado	Núm. Telf.	Troncal	Desde
				Agent/9000	No logon	-	-	-
				Agent/9003	No logon	-	-	-
				SIP/1064	No logon	-	-	-
				SIP/1071	No logon	-	-	-

At the bottom of the interface, there is a checkbox labeled 'Ver registro de campaña:'.

Figura 22: Interfaz del módulo de Monitoreo de Campaña

El módulo de Monitoreo de Campañas muestra cuatro áreas principales:

- El área superior tiene una lista desplegable para elegir la campaña a monitorear. La información de la campaña seleccionada se muestra en el resto de las etiquetas. Además información de la cola asociada a la campaña seleccionada se actualiza de inmediato en los demás paneles.
- El área central izquierda muestra las llamadas que se generan pero todavía no han sido asignadas a un agente. Esto incluye las llamadas que se están marcando y las llamadas que se han conectado y entrado a la cola, pero no pueden ser atendidas porque todos los agentes están ocupados. A medida que las llamadas se asignan a agentes, estas desaparecen de esta área y aparecen en el área central derecha.
- El área central derecha muestra los agentes de la cola asignada a la campaña. Cada agente está ocioso, o atendiendo una llamada. El listado muestra el número correspondiente a la llamada marcada o recibida.
- El área inferior sólo se muestra al activar la casilla de verificación. Esta área contiene una bitácora de la actividad de la campaña. Esta actividad incluye el acto del marcado, la conexión de la llamada, el ingreso de la llamada a la cola y la asignación a un agente.



## Monitoreo de agentes

Cola	Número	Agente	Estado actual	Total de llamadas	Tiempo total de sesión	Tiempo total de llamadas
8000	Agent/9000	Over 9000!!!!	LOGOUT 00:33:22	0	00:01:44	00:00:00
	Agent/9001	El agente 9001	LOGOUT	0	00:00:00	00:00:00
	Agent/9003	Agente gatón	LOGOUT	0	00:00:00	00:00:00
	SIP/1064	Alex	LOGOUT	0	00:00:00	00:00:00
	SIP/1071	Cuenta A	LOGOUT	0	00:00:00	00:00:00
<b>TOTAL</b>		<b>Agentes: 5</b>		<b>0</b>	<b>00:01:44</b>	<b>00:00:00</b>
8001	Agent/9000	Over 9000!!!!	LOGOUT 00:33:22	0	00:01:44	00:00:00
	Agent/9003	Agente gatón	LOGOUT	0	00:00:00	00:00:00
	SIP/1064	Alex	LOGOUT	0	00:00:00	00:00:00
	SIP/1071	Cuenta A	LOGOUT	0	00:00:00	00:00:00
<b>TOTAL</b>		<b>Agentes: 4</b>		<b>00:01:44</b>	<b>00:00:00</b>	
8010	Agent/9100	Test Agent 9100	LOGOUT	0	00:00:00	00:00:00
	Agent/9101	Test Agent 9101	LOGOUT	0	00:00:00	00:00:00
	Agent/9102	Test Agent 9102	LOGOUT	0	00:00:00	00:00:00
	Agent/9103	Test Agent 9103	LOGOUT	0	00:00:00	00:00:00
	Agent/9104	Test Agent 9104	LOGOUT	0	00:00:00	00:00:00
	Agent/9105	Test Agent 9105	LOGOUT	0	00:00:00	00:00:00
	Agent/9106	Test Agent 9106	LOGOUT	0	00:00:00	00:00:00
	Agent/9107	Test Agent 9107	LOGOUT	0	00:00:00	00:00:00
	Agent/9108	Test Agent 9108	LOGOUT	0	00:00:00	00:00:00

Figura 23: Interfaz del módulo de Monitoreo de Agentes

El módulo de monitoreo de agentes muestra todas las colas asignada al sistema, con todos los agentes administrados con el Call Center. Para cada combinación de cola y agente se muestra:

- Canal del agente, en formato Agent/xxxx para agentes estáticos, o SIP/xxxx e IAX/xxxx para agentes dinámicos.
- Nombre descriptivo del agente.
- Estado actual del agente. Este estado puede ser la palabra LOGOUT, o iconos correspondientes a los estados de ocioso, ocupado, o en receso. Si el agente ha iniciado sesión al menos una vez en el día, se

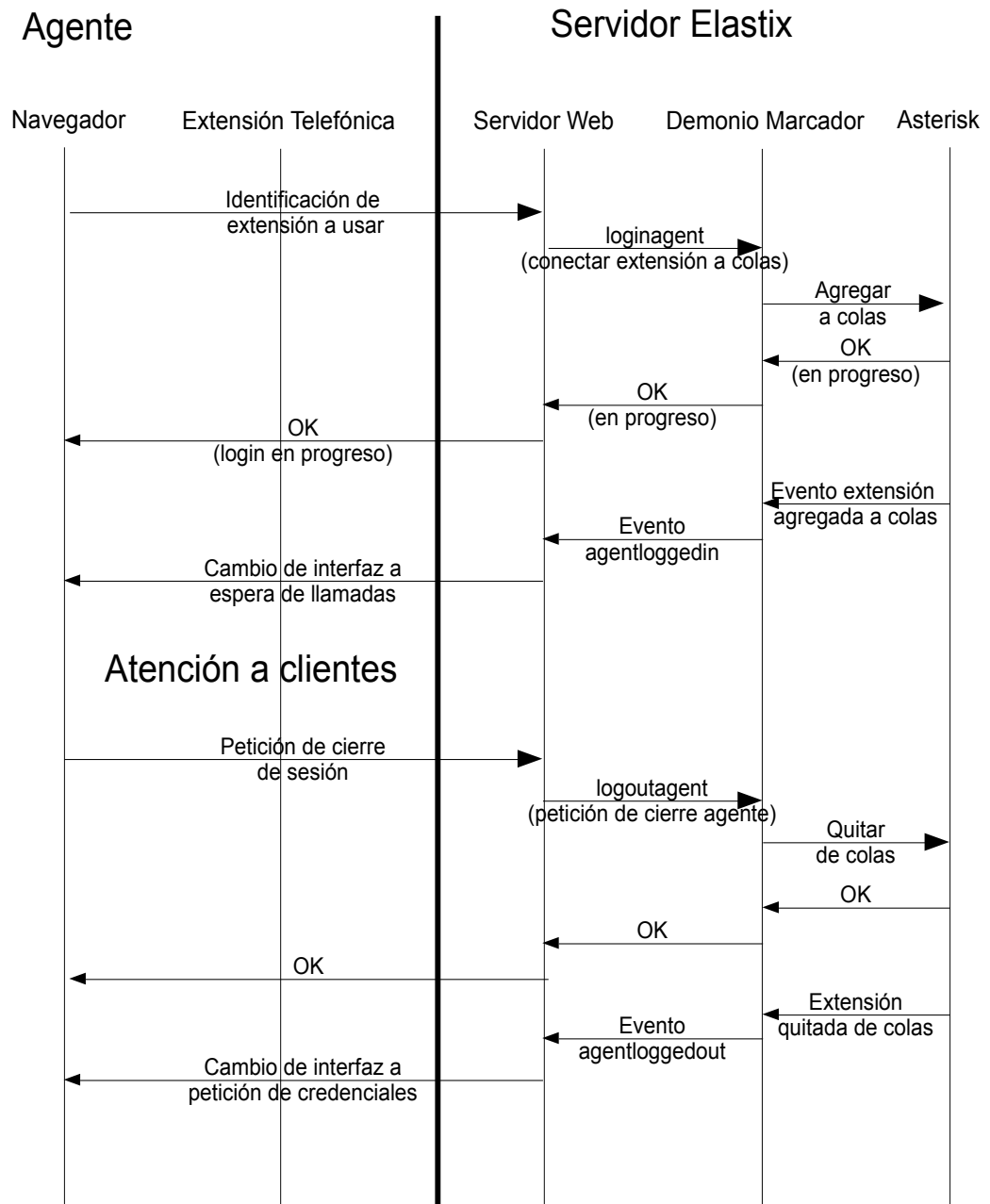
muestra también un contador del tiempo desde que ingresó al estado mostrado.

- Cuenta de llamadas atendidas por el agente a través de la cola indicada.
- Tiempo total de todas las sesiones iniciadas por el agente durante el día, incluyendo la sesión en curso, si existe.
- Tiempo total invertido en atención de llamadas que entraron por la cola durante el día.

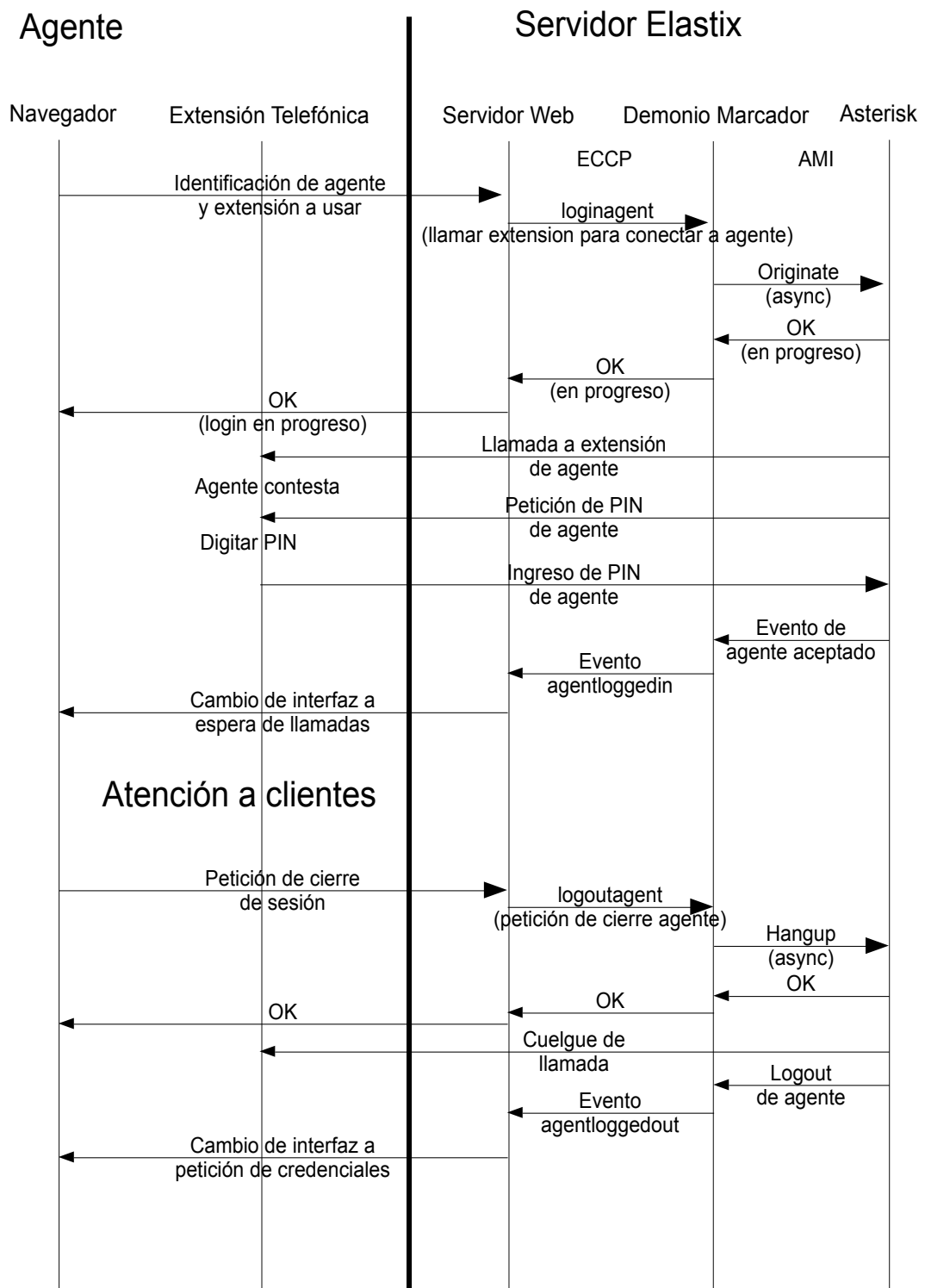
Todas las columnas a excepción de las dos primeras se actualizan en tiempo real al progresar las campañas.

## ANEXO 2

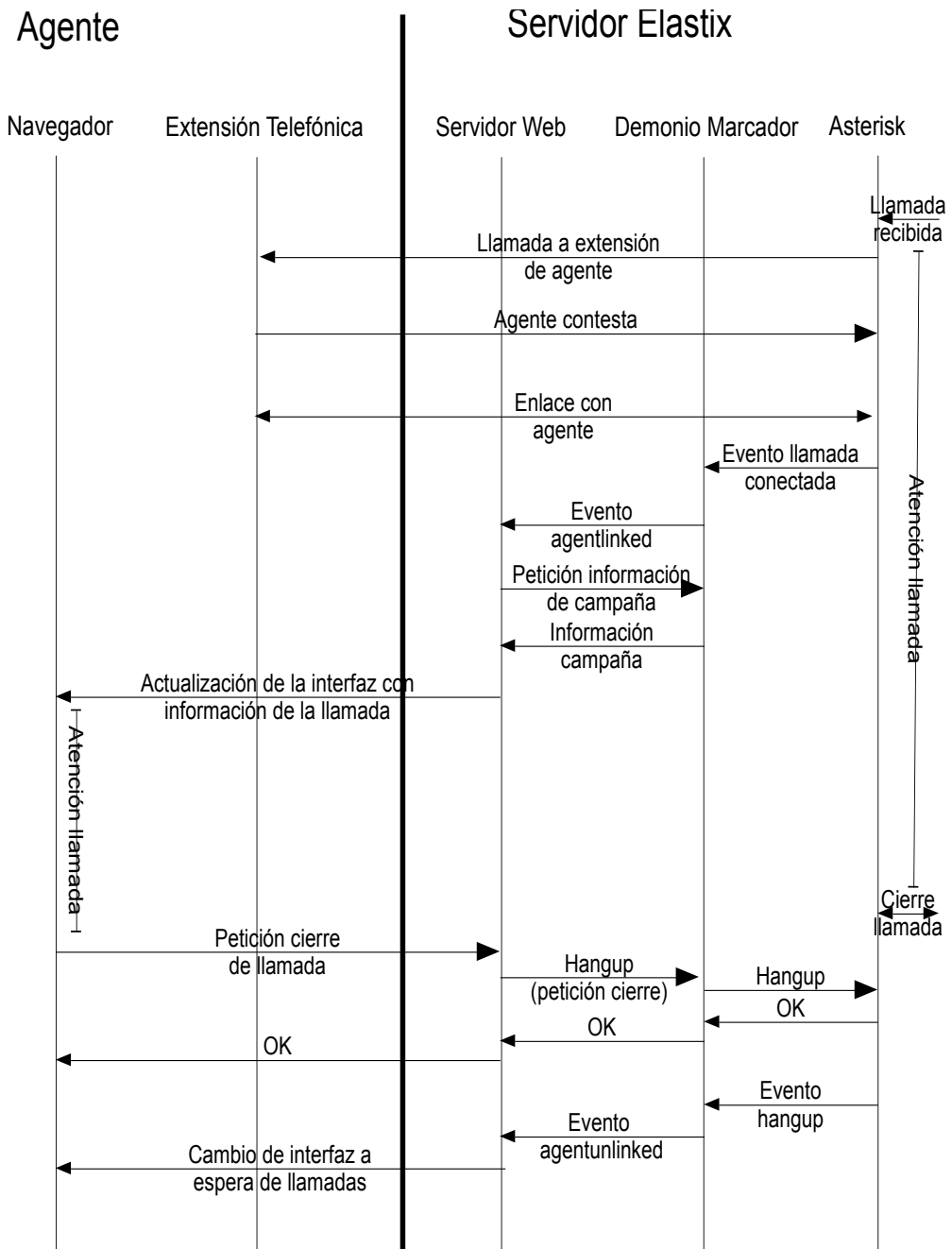
### DIAGRAMAS DE INTERACCIÓN



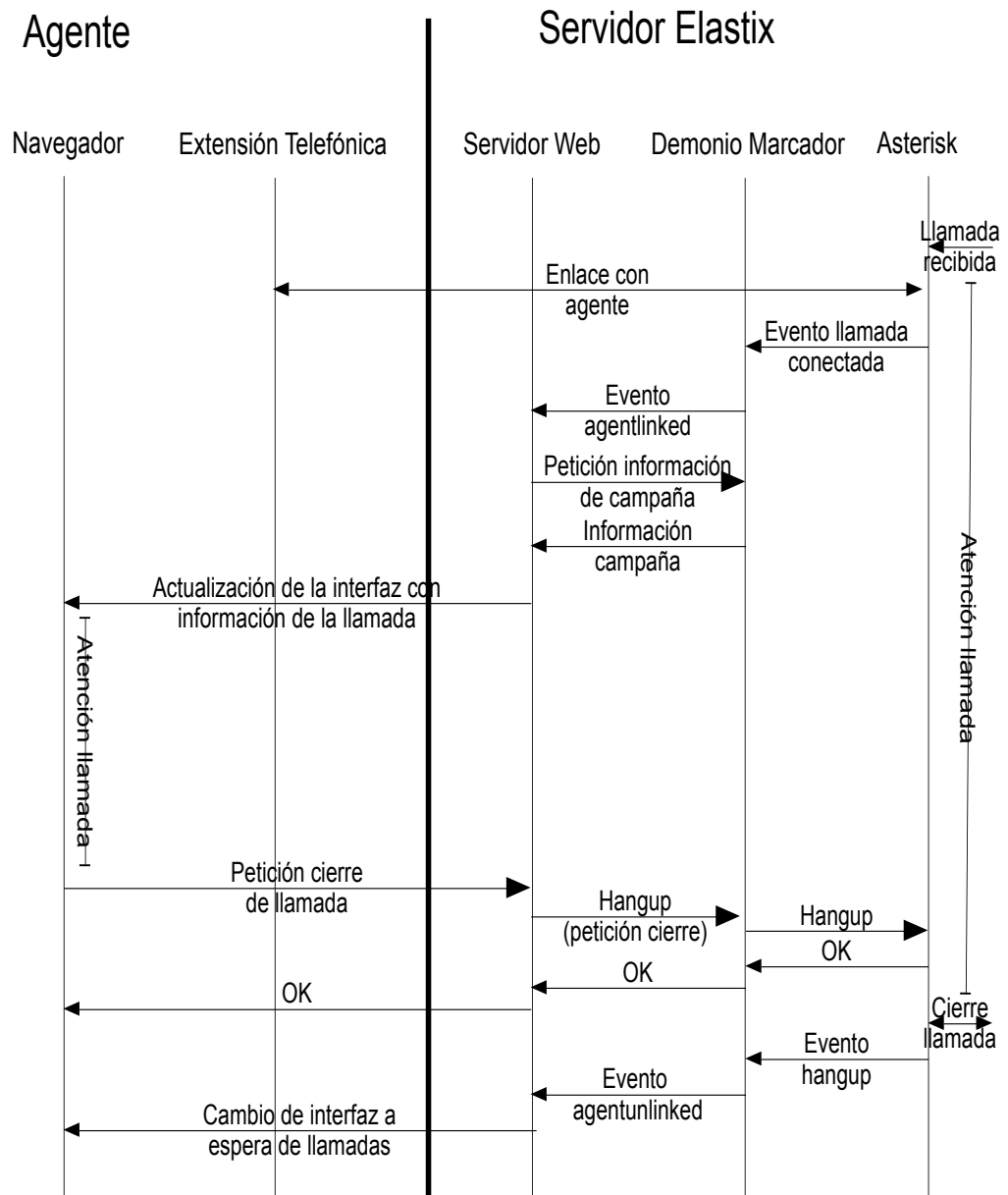
Interacción para una sesión de agente dinámico



Interacción para una sesión de agente estático



Interacción para una atención de llamada para agente dinámico



Interacción para la atención de llamada para agente estático



## ANEXO 4

### EJEMPLO DE CONFIGURACIÓN DE COLA EN ASTERISK

```
[8000]
announce-frequency=0
announce-holdtime=no
announce-position=no
autofill=no
eventmemberstatus=no
eventwhencalled=no
joinempty=yes
leavewhenempty=no
maxlen=0
periodic-announce-frequency=0
queue-callswaiting=silence/1
queue-thereare=silence/1
queue-youarenext=silence/1
reporholdtime=no
retry=5
ringinuse=yes
servicelevel=60
strategy=ringall
timeout=15
weight=0
wrapuptime=0
member=Agent/9000,0
member=Agent/9001,0
```



## ANEXO 5

### EJEMPLOS DE SALIDA DE COMANDOS DE ASTERISK

#### Comando “queue show”:

```
[root@elx2 asterisk]# asterisk -rnx 'queue show'
8100 has 0 calls (max unlimited) in 'ringall' strategy (0s holdtime,
0s talktime), W:0, C:0, A:0, SL:0.0% within 60s
  Members:
    Agent/9001 (ringinuse enabled) (Unavailable) has taken no
calls yet
    Agent/9000 (ringinuse enabled) (Unavailable) has taken no
calls yet
  No Callers

default has 0 calls (max unlimited) in 'ringall' strategy (0s
holdtime, 0s talktime), W:0, C:0, A:0, SL:0.0% within 0s
  No Members
  No Callers

8001 has 0 calls (max unlimited) in 'ringall' strategy (0s holdtime,
0s talktime), W:0, C:0, A:0, SL:0.0% within 60s
  Members:
    Agent/9000 (ringinuse enabled) (Unavailable) has taken no
calls yet
  No Callers

8000 has 0 calls (max unlimited) in 'ringall' strategy (0s holdtime,
0s talktime), W:0, C:0, A:0, SL:0.0% within 60s
  Members:
    Agent/9000 (ringinuse enabled) (Unavailable) has taken no
calls yet
  No Callers
```

#### Comando “agent show”:

```
[root@elx2 asterisk]# asterisk -rnx 'agent show'
9000          (Over 9000!!!) not logged in (musiconhold is 'default')
9001          (Agente 9001) not logged in (musiconhold is 'default')
2 agents configured [0 online , 2 offline]
```

## ANEXO 6

### PROTOCOLO ECCP

Protocolo ECCP v0.1

Revisión 5

Estatus: Beta

#### Objetivo

El protocolo ECCP es un protocolo basado en XML cuya finalidad es la de proveer de un API de comunicaciones a través de un puerto TCP para que aplicaciones externas puedan comunicarse con el motor de call center que provee Elastix.

La carencia de un mecanismo de comunicación entre el motor de call center y las aplicaciones cliente, que permita la transmisión de eventos asincrónicos, fue una de las motivaciones para la creación del ECCP. Antes del ECCP una aplicación cliente debía enviar peticiones periódicas al servidor para enterarse de la ocurrencia de eventos asincrónicos.

Por otro lado, era necesaria una estandarización del protocolo de comunicaciones con el motor de call center con miras a permitir escalabilidad y organización.

#### Convenciones usadas en este documento

En la mayor parte de este documento, cuando se refiera a “atributos” y “elementos”, estos términos estarán relacionados con su definición dentro del lenguaje XML.

Cuando se mencione la palabra servidor (a secas) se estará haciendo referencia al servidor de call center, que viene junto con el módulo de call center de Elastix.

Este programa también recibe el nombre de “dialerd”.

Toda fecha tendrá formato yyyy-mm-dd y toda hora tendrá formato hh:mm:ss. La zona horaria con la que se opera es siempre la que se ha configurado en el servidor que ejecuta el programa dialerd.

La palabra 'CallCenter' (sensitiva a mayúsculas) se utilizará para referirse al módulo CallCenter de Elastix.

#### Arquitectura



## Comunicaciones TCP/IP

La comunicación será TCP en el puerto 20005.

### Sesiones

El protocolo ECCP está orientado a sesiones y tiene la capacidad de soportar múltiples sesiones de comunicación. De esta manera puede atender de esta manera a varias aplicaciones cliente a la vez.

Una sesión quedará establecida luego de que la aplicación cliente se haya autenticado correctamente. La sesión permanecerá activa hasta que la aplicación cliente termine la sesión o hasta que ocurra un timeout por inactividad.

El valor del timeout por inactividad será de 5 minutos, luego de lo cual el servidor terminará la sesión.

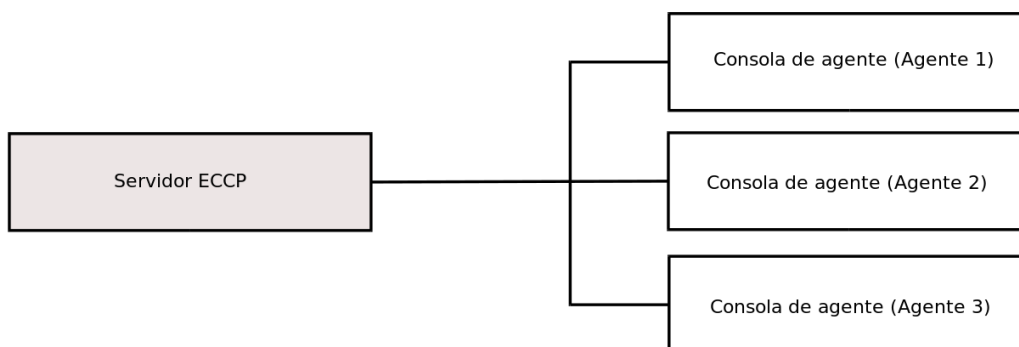
### Separación entre cliente y agente

El servidor se comunica con aplicaciones cliente y es la aplicación cliente la que gestiona el ingreso de los agentes al sistema. Una aplicación cliente puede manejar a uno o más agentes.

Esto último abre las puertas a dos escenarios que se explican a continuación.

Escenario 1: Consolas de agente que se conectan directamente al servidor.

Aquí cada consola de agente se encarga de gestionar el ingreso de un solo agente. Para el servidor cada consola de agente es vista como una aplicación cliente.

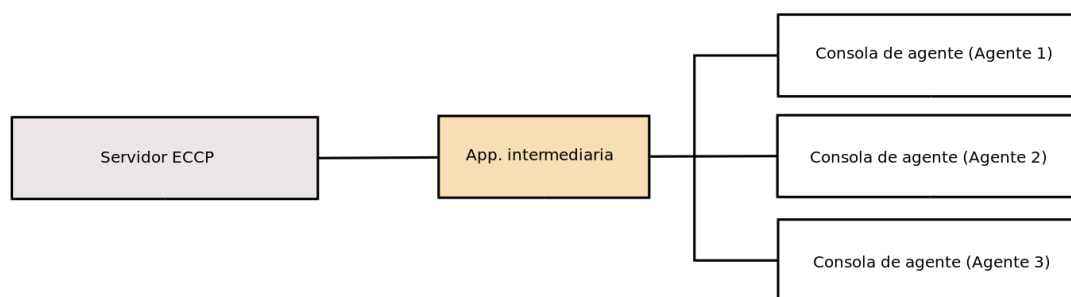


Como vemos en el ejemplo anterior el servidor ha iniciado 3 sesiones, una para cada aplicación cliente, que en este caso vendría a ser cada consola de agente.

Este esquema es el más sencillo de implementar.

Escenario 2: Consolas de agente que se conectan mediante una aplicación cliente intermediaria.

Aquí la aplicación cliente intermediaria se encarga de gestionar el ingreso de varios agentes. En este escenario, desde el punto de vista del servidor, la aplicación cliente intermediaria es vista como una aplicación cliente, mientras que las consolas de agente no.



Como vemos en el ejemplo anterior, el servidor ha iniciado UNA SOLA SESION en lugar de tres.

En un futuro nos referiremos a esta aplicación intermediaria como *servidor de concurrencia*. Un servidor de concurrencia es útil en varios sentidos. Uno de ellos es que se liberan recursos del servidor Elastix, pues el servidor de concurrencia puede estar instalado en un equipo remoto. El servidor Elastix ya no tiene que encargarse de gestionar las sesiones para las aplicaciones 'cliente'.

El servidor de concurrencia también puede servir de servidor de caché para descongestionar los requerimientos al servidor Elastix. Por ejemplo, podría hacer caché de información de las campañas.

### Consideraciones de seguridad

Para mantener cierto control sobre las aplicaciones cliente que se conectan al servidor, este último mantendrá un registro de IPs autorizadas para conectarse. En caso de que un cliente intente conectarse desde una dirección IP no autorizada, el servidor simplemente cerrará la conexión con un mensaje "Connection refused" y escribirá un mensaje de error en el log respectivo.

### **Tipos de paquetes**

Existen tres tipos de paquetes en el protocolo ECCP:

- evento
- requerimiento
- respuesta

Cada paquete se representa como un documento XML bien formado, cuyas etiquetas son definidas a continuación.

#### **Evento**

Un evento es generado por un acontecimiento asincrónico ocurrido en el Servidor.

A continuación su sintaxis básica:

```
<event>
```

```
...
```

```
</event>
```

**IMPORTANTE:** para extensión futura del protocolo, pueden definirse eventos adicionales a los descritos en este documento, así como etiquetas y atributos adicionales en eventos definidos. Un cliente de ECCP DEBE IGNORAR todo evento que no sepa manejar. De la misma manera, el cliente DEBE IGNORAR cualquier etiqueta o atributo que no sepa manejar en el evento que recibe.

#### **Requerimiento**

Un requerimiento es un mensaje enviado por la aplicación cliente hacia el servidor. El servidor contestará este mensaje con la respuesta apropiada.

En el requerimiento se incluye un atributo llamado id. Este id no es otra cosa que una cadena de caracteres que sirve para poder identificar la respuesta. De este modo se minimiza la posibilidad de que ocurran problemas ocasionados por condiciones de carrera. El identificador consta de dos partes unidas por un punto: la primera parte está constituida por el timestamp UNIX y la segunda parte está compuesta de un entero aleatorio de 6 caracteres. Un ejemplo de identificador es el siguiente:

```
1292899827.394772
```

A continuación de un ejemplo de un requerimiento:

```
<request id="1292899827.394772">
```

```
...
```

```
</request>
```

## Respuesta

Un requerimiento puede producir una o varias respuestas. Las respuestas se relacionan con los requerimientos a través del identificador del requerimiento. A continuación su sintaxis básica:

```
<response id=identificador>
...
</response>
```

**IMPORTANTE:** para extensión futura del protocolo, pueden definirse etiquetas y atributos adicionales en respuestas definidas. El cliente ECCP DEBE IGNORAR cualquier etiqueta o atributo que no sepa manejar en la respuesta que recibe.

## Autenticación de Requerimientos

Algunos de los Requerimientos listados en este manual ordenan actualizaciones en el estado del sistema que deben ser autorizadas únicamente para un agente en particular. Cuando un Requerimiento requiere ser autenticado, el método usado es el siguiente:

- Cada agente tiene asignada una contraseña conocida por el servidor.
- El requerimiento "login" de la aplicación, en caso de tener éxito, devuelve un texto aleatorio como contenido del tag "app\_cookie". De aquí en adelante a este texto se lo llamará cookie.
- Todo requerimiento que deba ser autenticado debe tener dos argumentos obligatorios, agent\_number y agent\_hash .
- El parámetro agent\_number es el nombre del agente, como "Agent/9000"
- El parámetro agent\_hash es un valor resultante de aplicar el hash MD5 a la concatenación del texto cookie, el nombre del agente, y la contraseña del agente, en este orden. Es decir, para el agente Agent/9000 con contraseña de valor "unaclavesecreta":

```
s = cookie + "Agent/9000" + "unaclavesecreta"
agent_hash <-- MD5(s)
```

...donde cookie contiene el valor devuelto en app\_cookie por el login, y el signo más (+) indica la operación de concatenación de cadenas.

Todo requerimiento que haga referencia a un argumento agent\_hash debe de recibir el valor de hash calculado de la manera indicada anteriormente.

## Mensajes de error

Los requerimientos definidos en este documento pueden reportar errores de

la siguiente manera:

```
<response id=identificador>
  <failure>
    <code>XXX</code>
    <message>Algún mensaje</message>
  </failure>
</response>
```

o de la siguiente manera:

```
<response id=identificador>
  <request_response>
    <failure>
      <code>XXX</code>
      <message>Algún mensaje</message>
    </failure>
  </request_response>
</response>
```

La diferencia entre los dos métodos es que el primero indica errores de formato o de protocolo que ocurren antes del procesamiento específico del requerimiento, mientras que el segundo método indica un error dentro del procesamiento específico del requerimiento.

Un requerimiento se considera fallido si contiene la etiqueta <failure>. Dentro de la etiqueta se almacena un código de error y un mensaje descriptivo. El código de error está vagamente basado en los códigos de error de HTTP, pero no se debe de contar con esta relación.

Los códigos de error más comunes son los siguientes:

<b>Código de error</b>	<b>Significado</b>
400	Petición mal formada
401	No autorizado
404	No se encuentra el objeto
417	Una condición esperada falla en cumplirse
500	Error interno del programa
501	No implementado

### **Casos comunes de errores**

Respuestas a requerimientos sin previo inicio de sesión:

En caso de requerimientos sin previo inicio de sesión, el servidor responderá así:

```
<response id=identificador>
  <failure>
    <code>401</code>
    <message>Unauthorized</message>
  </failure>
</response>
```

Respuestas a requerimientos mal formateados:

En caso de que el servidor no pueda entender algún requerimiento, responderá así:

```
<response>
  <failure>
    <code>400</code>
    <message>Bad request</message>
  </failure>
</response>
```

Como se puede observar, el atributo id no se incluye.

### **Modelo de funcionamiento de CallCenter**

A continuación se describe el modelo de funcionamiento del sistema CallCenter.

Esta explicación proporciona una referencia para comprender los tipos de requerimientos que son necesarios en el protocolo ECCP, así como los parámetros necesarios y opcionales en cada protocolo.

La asignación de llamadas en el CallCenter se implementa a través de las colas de Asterisk. Cada cola se identifica con un número, y tiene dos tipos de actores: agentes y llamadas. Un agente es una extensión de Asterisk que se ha registrado como perteneciente a la cola, para recibir llamadas. Una llamada puede, a través de órdenes dadas al Asterisk vía AMI (Asterisk Manager Interface), o vía un marcado directo a la central Asterisk, entrar a una cola. Al entrar, oír música de espera hasta que un agente se libere, en cuyo caso la llamada será enlazada con este agente.

Los agentes pueden ser de tipo dinámico o estático. Los agentes dinámicos pueden logonearse o deslogonearse de la cola arbitrariamente, y casi



siempre son extensiones reales de la central. Un agente estático siempre pertenece a la cola, aunque puede estar en estado no-disponible. Para el funcionamiento actual del CallCenter, todos los agentes estáticos deben ser canales de tipo Agent/9000, en oposición a los agentes dinámicos, que son extensiones cuyos canales deben ser de tipo SIP/4321 o IAX2/4321. Estas cadenas (Agent/9000, SIP/4321, IAX2/4321) identifican internamente al agente que puede atender llamadas. Tanto los agentes estáticos como dinámicos pueden pertenecer a múltiples colas.

Un agente estático tiene un número asociado, que es independiente de la extensión telefónica usada para hacer uso de este agente. Por ejemplo, una cola puede tener un agente 9000, el cual es usado por la extensión 1064. Para que una extensión haga uso de un agente estático, debe de completar el procedimiento de 'login' a la cola. En Elastix, se ha configurado el número \*8888 como login de cola arbitraria. Al marcar este número, la extensión oye una grabación para ingresar el número de agente que se va a usar. Luego de digitado, se reproduce otra grabación para que ingrese la contraseña del agente. Si se digita correctamente la contraseña, la extensión ingresa a la cola como un agente, y la extensión reproduce música de espera hasta que se le asigne una llamada. Para finalizar el login a la cola, se cuelga la extensión, o se ejecuta el método Agentlogoff del AMI indicando el canal Agent/XXXX que desea terminar la sesión. Al realizar login y logout a través del CallCenter, el sistema lleva la auditoría de cuánto tiempo ha estado logoneado cada agente.

Un agente dinámico es una extensión ordinaria de tipo SIP o IAX2. El logoneo a la cola consiste en un QueueAdd a cada cola de interés, y no requiere una contraseña internamente. Al agregarse a la cola, la extensión permanece colgada hasta que se asigna una llamada, en cuyo caso la extensión timbrará y el agente debe de contestar la llamada. Para finalizar el login a la cola, se ejecuta el método QueueRemove del AMI indicando el canal que desea deslogonearse. Al igual que con los agentes estáticos, el sistema lleva la auditoría de cuánto tiempo ha estado logoneado cada agente.

En el resto del documento, toda instancia de Agent/XXXX (por ejemplo Agent/9000) debe entenderse que se refiere a cualquier tipo de agente, sea estático o dinámico. Por lo tanto, también son aceptables SIP/4321 o IAX2/4321.

El CallCenter soporta dos modos principales de operación: campañas salientes y campañas entrantes. Una campaña tiene asociada una cola, a través de la cual se procesarán las llamadas asociadas a esta campaña. Tanto la campaña saliente, como la campaña entrante, pueden estar en

funcionamiento simultáneo en la misma central, siempre que se asignen colas separadas para cada función. Para una campaña entrante, se configura la central y los planes de marcado para que una llamada que viene del exterior termine en una cola que ha sido registrada como reservada para campañas entrantes (Interfaz Web: Call Center-->Ingoing Calls-->Queues). Para una campaña saliente, se debe configurar una nueva campaña (Call Center-->Outgoing Calls-->Campaigns) con una lista de números telefónicos a marcar. El sistema entonces marcará cada uno de los números, asociando como extensión a la cola donde esperan los agentes.

El sistema, luego de aceptar el login de los agentes, empieza a marcar o esperar llamadas. Para cada llamada entrante, se crea un nuevo registro de la llamada en la base de datos. Para las llamadas salientes, se actualiza el tiempo de duración de la llamada en la base de datos de los teléfonos cargados. El sistema notifica al agente cuando una llamada ha sido asignada por Asterisk, y le transmite el ID de la campaña, el ID de la llamada asociada, los atributos de la llamada que hayan sido cargados en la base de datos, el texto asociado que debe recitar el agente (el llamado "script" de la campaña), y los IDs de los formularios que se pueden usar para recoger datos sobre la llamada. La interfaz del lado del cliente recoge los datos para llenar el formulario, y se los transmite al sistema CallCenter para ser almacenados. (NOTA: la interfaz antigua del CallCenter no tiene todavía soporte para asociar formularios a una campaña entrante)

Para el caso de las llamadas entrantes, existe una lista de contactos identificada por número telefónico. El sistema identifica el número vía el servicio de Caller-ID e intenta mostrar los atributos del contacto que corresponde a una llamada entrante. Sin embargo, puede ocurrir que existan múltiples contactos que tienen el mismo número telefónico asociado. En este caso el CallCenter transmite la información de todos los contactos que coinciden con el número telefónico, y el agente humano debe de confirmar cuál contacto fue el que se ha recibido.

El sistema CallCenter permite que un agente pause la recepción de llamadas como parte de un receso controlado, también controlado y auditado. A estos recesos se los llama "breaks". Cuando un agente debe ingresar a un break, se selecciona desde la interfaz el tipo de break que debe de usar. Luego, al terminar el break el agente lo indica a través de la interfaz, y empieza a recibir llamadas otra vez.

## **Listado de eventos**

Evento "agentlinked"

Este evento se genera cuando un agente ha sido enlazado con una llamada ingresada a la cola. Internamente se ha recibido el evento Link/Bridge para la llamada.

Elementos informativos:

- `agent_number`: Agente que ha sido enlazado con la llamada. Este identificador está de la forma 'Agent/9000'.
- `remote_channel`: Canal que representa la llamada que ha sido enlazada. Puede estar en una variedad de formatos, pero el más frecuente es PROTOCOLO/XXX-YYYY.

Por ejemplo: 'SIP/unatroncal-0000001'.

- `call_type`: Uno de los dos valores 'incoming' u 'outgoing', para indicar si la llamada es entrante o saliente, respectivamente.
- `campaign_id`: ID de la base de datos de la campaña, un entero. Este valor es requerido para identificar una llamada saliente. Para una llamada entrante, este valor es opcional, e incluso puede estar omitido, si la llamada entrante no está asociada a una campaña.
- `call_id`: ID de la base de datos de la llamada, un entero. Con este ID, y el valor de `call_type` (y `campaign_id` para llamadas salientes), el código cliente puede identificar la llamada a procesar para formularios.
- `phone`: El número que fue marcado, o el Caller-ID que fue recibido.
- `status`: El estado de la llamada. Usualmente es 'Success' para llamada saliente y 'activa' para llamada entrante.
- `uniqueid`: ID interno de Asterisk que identifica a la llamada remota.
- `datetime_originate`: (Sólo llamada saliente) Fecha y hora en que se inició el marcado de la llamada.
- `datetime_originateresponse`: (Sólo llamada saliente) Fecha y hora en que se recibió respuesta del marcado de la llamada.
- `datetime_join`: Fecha y hora en que la llamada ingresó a la cola.
- `datetime_linkstart`: Fecha y hora en que la llamada fue enlazada al agente.
- `retries`: (Opcional) Número de veces que la llamada ha sido reintentada.
- `trunk`: Troncal a través de la cual se recibió la llamada. Por ejemplo: DAHDI/4 o SIP/unatroncal
- `queue`: (Sólo llamada entrante) Cola que recibió la llamada
- `call_attributes`: (Opcional) Atributos asociados a la llamada en la base de datos. Esta es una lista que contiene múltiples elementos 'attribute'. Cada elemento attribute tiene los siguientes elementos:
  - `label`: Etiqueta de texto asociada al atributo
  - `value`: Valor de texto del atributo

- order: Orden en que se sugiere que se muestren los atributos
- matching\_contacts: (Opcional, sólo llamada entrante) Lista de contactos cuyo número de teléfono coincide con el Caller-ID de la llamada. Si hay más de un contacto, la interfaz debe usar el requerimiento setcontact para indicar cuál contacto debe asociarse con la llamada recibida. Este elemento es una lista que contiene múltiples elementos 'contact'. Cada elemento contact tiene un atributo 'id' que debe usarse en el requerimiento setcontact. Además cada contacto tiene múltiples elementos 'attribute'. Cada elemento attribute contiene los siguientes elementos:
  - label: Etiqueta de texto asociada al atributo del contacto
  - value: Valor de texto del atributo del contacto
  - order: Orden en que se sugiere que se muestren los atributos del contacto
- call\_survey: (Opcional) Lista de formularios con los datos recogidos para esta llamada. Este elemento sólo está presente si se ha recogido previamente datos para esta llamada. Este elemento es una lista que contiene múltiples elementos 'form'. Cada elemento form tiene un atributo 'id' que debe usarse en el requerimiento saveformdata. Además cada form tiene múltiples elementos 'field'. Cada elemento field tiene un atributo 'id' que pueden usarse también en el requerimiento saveformdata. El elemento field tiene además los siguientes elementos:
  - label: Etiqueta del elemento de formulario
  - value: Valor que fue recogido en el formulario
- campaignlog\_id: ID del registro de bitácora. Para monitoreo de campaña.

NOTA: para averiguar los formularios aplicables a una llamada que no ha sido previamente llenada, se debe usar el requerimiento getcampaigninfo.

#### Evento "agentunlinked"

Este evento se genera cuando se deshace el enlace entre un agente y la llamada previamente enlazada. Internamente se ha recibido el evento Hangup del lado remoto.

#### Elementos informativos:

- agent\_number: Agente que ha sido desenlazado de la llamada. Este identificador está de la forma 'Agent/9000'.
- call\_type: Uno de los dos valores 'incoming' u 'outgoing', para indicar si

la llamada es entrante o saliente, respectivamente.

- `campaign_id`: (Opcional) ID de la base de datos de la campaña, un entero. Este valor es requerido para identificar una llamada saliente. Para una llamada entrante, este valor es opcional, e incluso puede estar omitido, si la llamada entrante no está asociada a una campaña.
- `call_id`: ID de la base de datos de la llamada, un entero. Con este ID, y el valor de `call_type` (y `campaign_id` para llamadas salientes), el código cliente puede identificar la llamada a procesar para formularios.
- `phone`: El número que fue marcado, o el Caller-ID que fue recibido.
- `datetime_linkend`: Fecha y hora en que la llamada fue desconectada del agente.
- `duration`: Duración en segundos de la llamada.
- `shortcall`: Bandera puesta a 1 si la llamada es demasiado corta según la configuración del sistema, o 0 si es una llamada de duración normal.
- `campaignlog_id`: ID del registro de bitácora. Para monitoreo de campaña.

#### Evento "agentloggedin"

Este evento se genera cuando el agente indicado ha ingresado al sistema. La separación de esto como un evento permite que la respuesta al requerimiento `loginagent` regrese de inmediato mientras el agente se demora digitando la clave.

Nota: no debe de confundirse este evento con el requerimiento `loginagent`.

Elementos informativos:

- `agent`: Agente que ha podido logonearse. Este identificador está de la forma 'Agent/9000'.

#### Evento "agentloggedout"

Este evento se genera cuando el agente indicado ha sido deslogoneado del sistema, sea por voluntad del agente (requerimiento `logoutagent`), o porque la extensión asociada ha sido colgada o se perdió la comunicación.

Nota: no debe de confundirse este evento con el requerimiento `logoutagent`.

Elementos informativos:

- `agent`: Agente que ha sido deslogoneado. Este identificador está de la forma 'Agent/9000'.

#### Evento "agentfailedlogin"

Este evento se genera cuando el agente indicado ha intentado logonearse al sistema pero el intento ha fallado. Una posible causa es que el agente no haya digitado la contraseña de ingreso a la cola.

Elementos informativos:

- agent: Agente que ha intentado logonearse. Este identificador está de la forma 'Agent/9000'.

Evento "pausestart"

Este evento se genera cuando el agente indicado entra en una pausa o un hold.

Elementos informativos:

- agent\_number: Agente que ha entrado en pausa. Este identificador está de la forma 'Agent/9000'.
- pause\_class: Uno de los tipos 'break' o 'hold'
- pause\_type: ID de la pausa que se ha indicado en el requerimiento pauseagent. Sólo presente para pause\_class igual a break.
- pause\_name: Nombre de la pausa correspondiente al ID indicado por pause\_type. Sólo presente para pause\_class igual a break.
- pause\_start: Fecha registrada como inicio de la pausa. Formato yyyy-mm-dd hh:mm:ss

Evento "pauseend"

Este evento se genera cuando el agente indicado sale de una pausa o hold.

Elementos informativos:

6. agent\_number: Agente que ha salido de pausa. Este identificador está de la forma 'Agent/9000'.
7. pause\_class: Uno de los tipos 'break' o 'hold'
8. pause\_type: ID de la pausa que se ha indicado en el requerimiento pauseagent. Sólo presente para pause\_class igual a break.
9. pause\_name: Nombre de la pausa correspondiente al ID indicado por pause\_type. Sólo presente para pause\_class igual a break.
10. pause\_start: Fecha registrada como inicio de la pausa. Formato yyyy-mm-dd hh:mm:ss
11. pause\_end: Fecha registrada como final de la pausa. Formato yyyy-mm-dd hh:mm:ss
12. pause\_duration: Duración de la pausa en segundos.

## Evento "callprogress"

Este evento se genera para indicar el progreso de la llamada, específicamente la transición de estado de la llamada referenciada. Este evento no es normalmente generado para todos los clientes. Para recibir este evento, el cliente ECCP debe de ejecutar el requerimiento "callprogress" con la bandera "enable" puesta a 1.

### Elementos informativos:

- **datetime\_entry:** Fecha y hora en la que se produjo la transición de estado
- **campaign\_type:** Tipo de la campaña, uno de 'incoming' u 'outgoing'
- **campaign\_id:** ID en la base de datos de la campaña a la que pertenece la llamada. Este valor siempre está presente para una llamada saliente, pero puede no estar presente para una llamada entrante.
- **call\_id:** ID en la base de datos de la llamada que se hace referencia en evento.
- **new\_status:** El nuevo estado al cual ha llegado la llamada. Los eventos reportados pueden ser uno de los siguientes:
  - **Placing:** se ha empezado a colocar la llamada saliente. El único parámetro válido es retry.
  - **Dialing:** se ha empezado a marcar la llamada saliente. Se conocen los valores de retry y trunk.
  - **Ringing:** la llamada saliente está timbrando y va a conectarse a la cola. Se conocen los valores de uniqueid, retry, trunk.
  - **OnQueue:** la llamada está en la cola esperando ser asignada un agente. Se conocen los valores de uniqueid, retry, trunk.
  - **Failure:** la llamada no ha podido ser conectada. Los valores válidos son dependientes del estado en el cual se detectó el fallo.
  - **OnHold:** la llamada ha entrado en hold. Se conocen los valores de uniqueid, retry, trunk.
  - **OffHold:** la llamada ha salido de hold. Se conocen los valores de uniqueid, retry, trunk.
- **retry:** Para llamadas salientes, el número de intento de marcado que corresponde a este cambio de estado. Siempre 0 para la llamada entrante.
- **uniqueid:** ID único de Asterisk para la llamada correspondiente al intento.
- **trunk:** Troncal a través de la cual se realiza la llamada saliente, o se recibe la llamada entrante.

- phone: número de teléfono de callerid o marcado
- queue: número de la cola por la que se espera que pase la llamada

### **Listado de requerimientos**

#### Requerimiento "getrequestlist"

Este requerimiento sirve para listar la lista completa de todos los requerimientos conocidos por esta versión del programa.

Atributos o argumentos:

Ninguno.

Respuesta:

- requests: Elemento que contiene uno o más elementos de tipo 'request'. Cada elemento 'request' tiene como contenido el nombre de un requerimiento.

#### Requerimiento "login"

Este requerimiento sirve para autenticar una aplicación cliente y permitirle establecer una sesión de comunicaciones con el servidor. El servidor ECCP no transmitira evento alguno hasta que el cliente haya completado un login correcto con el servidor.

Atributos o argumentos:

- username: Nombre de usuario asignado al cliente
- password: Contraseña correspondiente al nombre de usuario

Respuesta:

- app\_cookie: Cadena de texto arbitraria que se genera como parte del login. El cliente debe de almacenar esta cadena para poder realizar el requerimiento loginagent.

### **Encriptación de la clave**

La clave va "encriptada" con MD5. El hash se calcula en base a una semilla, la cual es un secreto compartido entre la aplicación cliente y el servidor. Es decir que la aplicación cliente previamente se tendrá que configurar para conocer dicho secreto.

TODO: secreto compartido no implementado todavía. Se requiere acordar manera de usar semilla. Por ahora se acepta clave en plaintext, o md5 de la misma clave.



### Requerimiento "logout"

El requerimiento "logout" sirve para terminar la sesión entre la aplicación cliente y el servidor. Luego de ejecutar este requerimiento, el servidor cerrará la sesión TCP con el cliente.

Atributos o argumentos:

Este requerimiento no tiene atributos.

### Requerimiento "filterbyagent"

Este requerimiento sirve para indicar al servidor ECCP que el cliente sólo está interesado en los eventos del agente indicado en la petición, y que los eventos asociados a otros agentes no deben ser enviados a esta sesión. De esta manera el cliente no debe de recibir eventos que no le interesa procesar, por ser de agentes distintos al que es representado por el cliente. Este filtrado afecta

únicamente a los eventos - cualquier requerimiento que haga referencia a otro

agente será contestado correctamente, incluso luego de aplicar el filtro. El filtro es local para la sesión activa - otras sesiones simultáneas con otros clientes no se ven afectadas.

Atributos o argumentos:

- `agent_number`: Agente que se filtra. Este identificador está de la forma 'Agent/9000'. También se puede especificar la palabra 'any' para deshacer el filtro y volver a escuchar eventos de todos los agentes.

### Requerimiento "getagentstatus"

El requerimiento "getagentstatus" informa el estado actual de disponibilidad del agente indicado.

Atributos o argumentos:

- `agent_number`: Agente que se interroga. Este identificador está de la forma 'Agent/9000'.

Respuesta:

- `status`: tiene uno de los siguientes posibles valores: `offline`: Agente no ha iniciado sesión en la cola `online`: Agente ha iniciado sesión, y está ocioso `oncall`: Agente ha iniciado sesión, y está atendiendo una

llamada paused: Agente ha iniciado sesión, y está pausado (en break)

- channel: contiene el canal de Asterisk usado para el login del agente.
- extension: contiene la extensión (derivada del canal) usada para el login del agente. Si los elementos "channel" y "extension" están ausentes con "status" igual a "offline", el agente está deslogueado. Si están presentes con "status" igual a "offline", el agente está en proceso de ingresar la contraseña de ingreso a la cola, a través del canal y extensión indicados. En todos los otros valores de "status", los elementos "channel" y "extension" están siempre presentes.
- onhold: 1 si el agente ha iniciado un hold con el requerimiento "hold", o 0.
- pauseinfo: elemento que contiene información de la pausa (break) en la que está puesto el agente. Sólo está presente en status paused. Contiene los siguientes elementos:
  - pauseid: ID de la pausa usada en pauseagent
  - pausename: Nombre de la pausa correspondiente al ID de pauseid
  - pausestart: Fecha y hora correspondiente al inicio de la pausa. Si la fecha es la misma fecha de hoy, sólo se reporta la hora. Formato [yyyy-mm-dd] hh:mm:ss.
- remote\_channel: canal de la llamada que ha sido enlazada con el agente. Sólo presente si el agente está atendiendo una llamada.
- callinfo: elemento que contiene información de la llamada enlazada al agente. Sólo está presente si el agente está atendiendo una llamada. Contiene los siguientes elementos:
  - calltype: Valor de "outgoing" o "incoming"
  - campaign\_id: ID de la campaña correspondiente a la llamada, o ausente si la llamada entrante no pertenece a una campaña.
  - callid: ID de la llamada.
  - callnumber: Número telefónico marcado (llamada saliente) o Caller-ID (llamada entrante) correspondiente a la llamada enlazada.
  - dialstart: Fecha y hora en formato [yyyy-mm-dd] hh:mm:ss para inicio del marcado de la llamada saliente. Ausente para llamadas entrantes.
  - dialend: Fecha y hora en formato [yyyy-mm-dd] hh:mm:ss para final del marcado de la llamada saliente. Ausente para llamadas entrantes.
  - queuestart: Fecha y hora en formato [yyyy-mm-dd] hh:mm:ss para entrada de la llamada a la cola de la campaña.
  - linkstart: Fecha y hora en formato [yyyy-mm-dd] hh:mm:ss para enlace de la llamada al agente.

- `queuenum`: Cola a través de la cual se asignó la llamada al agente. Sólo está presente si la llamada ha sido asignada a través de una cola.

### Requerimiento "getmultipleagentstatus"

Este requerimiento es una optimización del requerimiento "getagentstatus". La diferencia principal es que el requerimiento "getmultipleagentstatus" permite interrogar el estado de muchos agentes a la vez. De esta manera se reducen la cantidad de llamadas ECCP necesarias para implementar monitoreos.

Atributos o argumentos:

- `agents`: Lista de agentes para los que se requiere el reporte de estado de los agentes. Este elemento contiene múltiples elementos 'agent\_number', cada uno de los cuales identifica al agente, en formato Agent/9000.

Respuesta:

- `agents`: Lista de agentes para los que se reporta el estado actual. Este elemento contiene múltiples elementos 'agent', los cuales contienen los mismos campos que el elemento `getagentstatus_response` del requerimiento "getagentstatus", a excepción de `pauseinfo`. Además, cada elemento 'agent' contiene los siguientes campos:
  - `agent_number`: Agente identificado por el elemento

Véase `getagentstatus` para los detalles de los campos.

### Requerimiento "loginagent"

Este requerimiento inicia la conexión de una extensión, bajo la identidad de un agente específico, hacia una cola. En la implementación actual, Asterisk inicia una llamada hacia la extensión, la cual al ser contestada, reproduce una petición de contraseña. Al digitar la contraseña correctamente, el agente queda logoneado a la cola, y se emite el evento "agentloggedin". Si no se contesta la llamada, o si la contraseña se digita incorrectamente, se emite el evento "agentfailedlogin". La respuesta a este requerimiento indica únicamente que la llamada hacia el agente ha sido iniciada, no que el agente haya sido aceptado en la cola. Opcionalmente, se puede especificar un intervalo máximo de inactividad. Si el agente no tiene actividad alguna desde un login exitoso hasta que transcurre el intervalo, el dialer deslogonea automáticamente al agente.

Para mantener activa una sesión de agente con un timeout, véase el requerimiento pingagent.

Atributos o argumentos:

- agent\_number: El número de agente a logonear. Este identificador está de la forma 'Agent/9000'.
- agent\_hash: Hash de autenticación para el agente.
- extension: El número de extensión a ingresar bajo la identidad del agente.
- password: (opcional) Clave telefónica del agente (no se usa en la implementación actual)
- timeout: (opcional) Intervalo máximo de inactividad, en segundos.

El elemento "status" en la respuesta tiene uno de los siguientes posibles valores:

- logging: Llamada ha sido marcada y espera respuesta
- logged-in: Agente ya estaba anteriormente logoneado con ECCP en la misma extensión
- logged-out: No se puede iniciar el proceso de login. Posibles causas:
  - Agente indicado no existe
  - Extensión indicada no existe
  - Agente ya logoneado con un medio distinto al ECCP
  - Agente ya logoneado en una extensión distinta

Requerimiento "logoutagent"

Este requerimiento termina la sesión de un agente en las colas a la que pertenece. Luego de finalizada la sesión, se emite un evento "agentlogoff".

Atributos o argumentos:

- agent\_number: El número de agente a logonear. Este identificador está de la forma 'Agent/9000'.
- agent\_hash: Hash de autenticación para el agente.

Requerimiento "getqueuescript"

Este requerimiento recupera el texto asociado a una cola de campaña entrante. Este texto es el que debe mostrarse al agente cuando se recibe una llamada entrante en una cola monitoreada, y no existe una campaña entrante creada que use la cola en cuestión. Si existe una campaña entrante, debe

usarse su texto asociado en lugar del que se recupera con este requerimiento.

Atributos o argumentos:

- queue: Número de la cola para el que se recupera el script.

Respuesta:

- script: Texto libre que debe de presentarse en la interfaz gráfica del agente.

Requerimiento "getcampaignlist"

Este requerimiento lista todas las campañas del sistema.

Atributos o argumentos:

- campaign\_type: Opcional. Si se especifica, debe ser 'incoming' o 'outgoing'. Si no se especifica, se leen todas las campañas.
- status: Opcional. Si se especifica, debe ser 'active', 'inactive', 'finished'.
- filtername: Opcional. Si se especifica, el nombre de la campaña se filtrará para que contenga este texto.
- datetime\_start: Opcional. El inicio de la campaña deberá ser igual o posterior a esta fecha. Formato yyyy-mm-dd.
- datetime\_end: Opcional. El final de la campaña deberá ser igual o posterior a esta fecha. Formato yyyy-mm-dd.
- offset: Opcional. Si se especifica, debe también indicarse limit. Se devuelven registros a partir del número indicado, con el primer registro siendo 0. Se asume 0 por omisión en caso de especificar limit sin offset.
- limit: Opcional. Si se especifica, se devuelve como máximo el número de registros indicado por el parámetro. Si no se especifica, se devuelven todos los registros posibles.

Respuesta:

- campaigns: Elemento que contiene cero o más elementos 'campaign'. Cada elemento 'campaign' contiene:
  - id: ID de la campaña
  - type: Tipo de la campaña: 'outgoing' o 'incoming'
  - name: Nombre de la campaña
  - status: Una de 'active', 'inactive', 'finished'

## Requerimiento "getcampaiginfo"

Este requerimiento recupera información estática sobre la campaña. Por información estática, se entiende la información que no cambia como consecuencia del progreso de las llamadas pertenecientes a la campaña.

Atributos o argumentos:

- `campaign_type`: Estilo de la campaña. Uno de los valores 'incoming','outgoing'
- `campaign_id`: ID de la campaña.

Respuesta:

- `name`: Nombre de la campaña
- `type`: Tipo de la campaña, uno de 'incoming' u 'outgoing'
- `startdate`: Fecha desde la cual tiene vigencia la campaña.
- `enddate`: Fecha hasta la cual tiene vigencia la campaña
- `working_time_starttime`: Hora desde la que se considera activa la campaña
- `working_time_endtime`: Hora hasta la que se considera activa la campaña
- `queue`: Cola usada para la campaña
- `retries`: (Sólo campañas salientes) Máximo número de reintentos para mismo número
- `context`: (Sólo campañas salientes) Contexto Asterisk desde el que se marca
- `maxchan`: (Sólo campañas salientes) Máximo número de llamadas a realizar
- `trunk`: (Opcional, sólo campañas salientes) Troncal a usar para las llamadas. Si no aparece, las llamadas se generan a través del plan de marcado de FreePBX
- `status`: Uno de los valores 'active', 'inactive', 'finished' (sólo campañas salientes)
- `urltemplate`: Si no está vacío, una plantilla para poder construir un URL a una página Web que debe de cargarse cuando la llamada es conectada. La plantilla de URL puede contener identificadores con nombres entre llaves, como por ejemplo {HomeAddress}, los cuales deben de ser reemplazados con los verdaderos valores escapados para cada llamada conectada. Los identificadores disponibles son los nombres de los atributos de la llamada (con distinción de mayúsculas y minúsculas), además de los siguientes atributos por omisión de la llamada:
  - `{__AGENT_NUMBER__}` Identificador del canal de agente, por ejemplo Agent/9000

- {\_\_CALL\_TYPE\_\_} uno de los valores 'incoming', 'outgoing'
- {\_\_CAMPAIGN\_ID\_\_} ID interno de la campaña en curso
- {\_\_CALL\_ID\_\_} ID interno de la llamada actualmente conectada
- {\_\_PHONE\_\_} Número marcado o Caller-ID de la llamada conectada
- {\_\_REMOTE\_CHANNEL\_\_} Identificador de canal de la llamada conectada
- urlopentype: La manera recomendada de abrir el URL indicado por urltemplate: window: El URL debe de abrirse en una ventana nueva del navegador, o equivalente. embedded: El URL debe de abrirse en un marco embebido dentro de la consola. jsonp: El URL debe interpretarse como código Javascript que debe de inyectarse en la consola Web.
- script: Texto a usar como el guión de la campaña. NOTA: en la implementación actual, el texto almacenado se genera con un widget de edición que agrega estilos HTML/CSS en este campo. El cliente debe estar preparado para mostrar tales estilos, o de filtrarlos y mostrar únicamente el texto.
- forms: Elemento que contiene uno o más elementos 'form'
  - form: Elemento presente una o más veces. Representa un formulario que se puede usar para recoger información sobre las llamadas asociadas a la campaña. El elemento tiene los siguientes atributos:
    - id: ID de la base de datos para el formulario.
    - name: Nombre corto del formulario
    - description: Descripción larga del formulario
 Además, el elemento contiene uno o más elementos:
    - field: Campo de entrada de información para este formulario. Cada elemento field tiene los siguientes atributos:
      - order: Orden de presentación del campo
      - id: ID de base de datos del campo de entrada
 Además cada elemento field contiene a su vez los siguientes elementos:
      - label: Etiqueta asociada al campo
      - type: Uno de los siguientes: TEXT TEXTAREA DATE LABEL LIST
      - maxsize: (Opcional) Especifica la longitud máxima de la entrada, en caracteres
      - default\_value: (Opcional) Un valor por omisión sugerido al

cliente que muestra el formulario. ATENCION: este valor debe de ser guardado como el valor elegido, como si hubiese sido ingresado por el agente.

- options: (Opcional): Sólo presente cuando type es LIST. Contiene múltiples elementos 'value'.
  - value: Uno de los múltiples valores que se aceptan como entrada para este campo.

### Requerimiento "getcampaignstatus"

Este requerimiento recupera información dinámica o volátil sobre la campaña.

Esta información incluye el estado de los agentes disponibles en la cola de la campaña activa, el número de llamadas en los varios estados reconocidos, y las llamadas activas que todavía no han sido asignadas a agentes.

Atributos o argumentos:

- campaign\_type: Estilo de la campaña. Uno de los valores 'incoming','outgoing'
- campaign\_id: ID de la campaña.
- datetime\_start: Opcional. Si se especifica, las estadísticas que involucren llamadas se calcularán sólo con las llamadas realizadas a partir de la fecha indicada (yyyy-mm-dd)

Respuesta:

- statuscount: elemento que contiene los siguientes atributos:
  - total: Para llamadas salientes, este es el total de llamadas registradas como parte de la campaña. Para llamadas entrantes, este es el total de llamadas que se han recibido hasta el momento en la cola.
  - pending: Cantidad de llamadas que no ha sido marcada todavía en ningún intento.
  - placing: Para llamadas salientes, cantidad de llamadas para las que se ha mandado a Asterisk la orden de generar la llamada, pero todavía no se recibe noticia de éxito o fracaso de la generación (en el evento OriginateResponse). No definido para llamadas entrantes.
  - ringing: Para llamadas salientes, cantidad de llamadas para las que se recibió noticia de éxito en marcado, pero todavía no entran a la cola de la campaña. No definido para llamadas entrantes.
  - onqueue: Para llamadas salientes o entrantes, cantidad de



llamadas conectadas que esperan en la cola a ser asignadas a un agente.

- success: Para llamadas salientes o entrantes, cantidad de llamadas que fueron asignadas a un agente. Para llamadas salientes, esta cuenta incluye las llamadas que han sido terminadas correctamente luego de ser atendidas por un agente.
- onhold: Para llamadas salientes o entrantes, cantidad de llamadas que han sido puestas en espera.
- failure: Para llamadas salientes, cantidad de llamadas para las cuales el marcado ha fallado al menos una vez. No definido para llamadas entrantes.
- shortcall: Para llamadas salientes, cantidad de llamadas que fueron asignadas a un agente, pero duraron muy poco como para representar una interacción adecuada. No definido para llamadas entrantes.
- noanswer: Para llamadas salientes, cantidad de llamadas que se desconectaron luego de haber sido conectadas (según OriginateResponse) pero antes de haber entrado a la cola de la campaña. No definido para llamadas entrantes.
- abandoned: Para llamadas salientes y entrantes, cantidad de llamadas que se desconectaron luego de haber entrado a una cola, pero antes de ser asignadas a un agente.
- finished: Para llamadas entrantes, cantidad de llamadas ya terminadas luego de ser atendidas por un agente. No definido para llamadas salientes.
- losttrack: Para llamadas entrantes, cantidad de llamadas que no pudieron ser vigiladas hasta terminar porque se reinició el marcador. No definido para llamadas salientes.
- agents: Agentes asociados a la cola de la campaña. Este elemento es una lista que contiene múltiples elementos 'agent'. Cada elemento contiene los siguientes atributos:
  - agentchannel: Canal del agente de la cola (normalmente Agent/NNNN)
  - status: Estado del agente. Puede ser 'online', 'offline', 'oncall', 'paused' como se describe en la respuesta del requerimiento 'getagentstatus'
  - callid: ID en la base de datos de la llamada que atiende el agente. Sólo está definido si el agente está logoneado y atendiendo alguna llamada.
  - callnumber: Número telefónico del cliente al que se conecta la llamada. Sólo está definido si el agente está logoneado y

atendiendo alguna llamada.

- callclient: Canal asignado a la llamada del cliente atendido. Sólo está definido si el agente está logoneado y atendiendo alguna llamada.
- pausename: Nombre de la pausa en la que está el agente. Sólo está definido si el agente está en pausa.
- pauseid: ID de la pausa en la que está el agente. Sólo está definido si el agente está en pausa.
- pausestart: Hora en la que inició la pausa, en formato HH:MM:SS de 24 horas. Sólo está definido si el agente está en pausa.
- dialstart: Hora en la que inició la marcación de la llamada que atiende el agente. Sólo definido para campañas salientes, y si el agente está atendiendo una llamada.
- dialend: Hora en la que terminó la marcación de la llamada que atiende el agente. Sólo definido para campañas salientes, y si el agente está atendiendo una llamada.
- queuestart: Hora en la que entró en la cola la llamada atendida por el agente. Sólo definido si el agente está atendiendo una llamada.
- linkstart: Hora en la que la llamada fue conectada con el agente. Sólo definido si el agente está atendiendo una llamada.
- trunk: Troncal a través de la cual se recibe llamada entrante, o se realiza la llamada saliente. Sólo definido si el agente está atendiendo una llamada.
- activecalls: Llamadas de la campaña que todavía no han sido asignadas a un agente. Este elemento es una lista que contiene múltiples elementos 'activecall'. Cada elemento contiene los siguientes atributos:
  - callid: ID en la base de datos de la llamada conectada al cliente
  - callnumber: Número telefónico de la llamada conectada al cliente
  - callstatus: Estado de la llamada de la campaña. Puede ser uno de los valores 'Placing' (sólo campañas salientes), 'Ringing' (sólo campañas salientes), 'OnQueue' (campañas entrantes o salientes)
  - dialstart: Hora en la que inició la marcación de la llamada. Solamente definido para campañas salientes.
  - dialend: Hora en la que se terminó la marcación de la llamada. Solamente definido para campañas salientes, y si el estado en 'callstatus' es 'ringing' o 'onqueue'.
  - queuestart: Hora en la que la llamada ingresó a la cola. Solamente definido para campañas salientes, y si el estado en 'callstatus' es 'onqueue'.

- trunk: Troncal a través de la cual se recibe llamada entrante, o se realiza la llamada saliente.
- stats: Contadores globales de la campaña Este elemento contiene los siguientes atributos:
  - total\_sec: Suma de todas las duraciones en segundos de todas las llamadas completadas.
  - max\_duration: Máxima duración de llamada en la campaña.

#### Requerimiento "getcampaigueuewait"

Este requerimiento calcula un histograma de la cantidad de segundos que tarda una llamada en la campaña elegida desde que entra a la cola hasta que es asignada a un agente. El histograma se calcula en intervalos de 5 segundos hasta 30 segundos, más una categoría de llamadas atendidas luego de 30 segundos, y además una cuenta de llamadas abandonadas.

Atributos o argumentos:

- campaign\_type: Estilo de la campaña. Uno de los valores 'incoming','outgoing'
- campaign\_id: ID de la campaña.

Respuesta:

- histogram: lista que contiene elementos "interval". Cada elemento interval contiene los siguientes campos:
  - lower: Valor inferior de tiempo de espera (inclusive) abarcado por intervalo
  - upper: Valor superior de tiempo de espera (inclusive) abarcado por intervalo. Para el último intervalo, se omite este elemento porque representa las llamadas iguales o mayores a 30 segundos.
  - count: Número de llamadas que han caído dentro del intervalo
- abandoned: Contiene el número de llamadas que fueron abandonadas, es decir, que entraron a la cola, pero terminaron antes de ser asignadas a un agente.

#### Requerimiento "getcallinfo"

Este requerimiento recupera información sobre una llamada específica de una campaña. Actualmente sólo se recupera información que se puede leer

desde la base de datos, y no información sobre la posible llamada en curso.

Atributos o argumentos:

- `campaign_type`: Tipo de la campaña, uno de 'incoming' u 'outgoing'
- `campaign_id`: ID en la base de datos de la campaña a la que pertenece la llamada. Este valor es opcional para una llamada entrante, pero es requerido para una llamada saliente.
- `call_id`: ID en la base de datos de la llamada que se interroga

Respuesta:

- `agent_number`: Agente que ha sido enlazado con la llamada. Este identificador está de la forma 'Agent/9000'.
- `call_type`: Uno de los dos valores 'incoming' u 'outgoing', para indicar si la llamada es entrante o saliente, respectivamente.
- `campaign_id`: ID de la base de datos de la campaña, un entero. Este valor es requerido para identificar una llamada saliente. Para una llamada entrante, este valor es opcional, e incluso puede estar omitido, si la llamada entrante no está asociada a una campaña.
- `call_id`: ID de la base de datos de la llamada, un entero. Con este ID, y el valor de `call_type` (y `campaign_id` para llamadas salientes), el código cliente puede identificar la llamada a procesar para formularios.
- `phone`: El número que fue marcado, o el Caller-ID que fue recibido.
- `status`: El estado de la llamada. Usualmente es 'Success' para llamada saliente y 'activa' para llamada entrante.
- `uniqueid`: ID interno de Asterisk que identifica a la llamada remota.
- `datetime_originate`: (Sólo llamada saliente) Fecha y hora en que se inició el marcado de la llamada.
- `datetime_originateresponse`: (Sólo llamada saliente) Fecha y hora en que se recibió respuesta del marcado de la llamada.
- `datetime_join`: Fecha y hora en que la llamada ingresó a la cola.
- `datetime_linkstart`: Fecha y hora en que la llamada fue enlazada al agente.
- `retries`: (Opcional) Número de veces que la llamada ha sido reintentada.
- `trunk`: (Sólo llamada entrante) Troncal a través de la cual se recibió la llamada. Por ejemplo: DAHDI/4 o SIP/unatroncal
- `queue`: (Sólo llamada entrante) Cola que recibió la llamada
- `call_attributes`: (Opcional) Atributos asociados a la llamada en la base de datos. Esta es una lista que contiene múltiples elementos 'attribute'. Cada elemento attribute tiene los siguientes elementos:
  - `label`: Etiqueta de texto asociada al atributo
  - `value`: Valor de texto del atributo

- order: Orden en que se sugiere que se muestren los atributos
- matching\_contacts: (Opcional, sólo llamada entrante) Lista de contactos cuyo número de teléfono coincide con el Caller-ID de la llamada. Si hay más de un contacto, la interfaz debe usar el requerimiento setcontact para indicar cuál contacto debe asociarse con la llamada recibida. Este elemento es una lista que contiene múltiples elementos 'contact'. Cada elemento contact tiene un atributo 'id' que debe usarse en el requerimiento setcontact. Además cada contacto tiene múltiples elementos 'attribute'. Cada elemento attribute contiene los siguientes elementos:
  - label: Etiqueta de texto asociada al atributo del contacto
  - value: Valor de texto del atributo del contacto
  - order: Orden en que se sugiere que se muestren los atributos del contacto
- call\_survey: (Opcional) Lista de formularios con los datos recogidos para esta llamada. Este elemento sólo está presente si se ha recogido previamente datos para esta llamada. Este elemento es una lista que contiene múltiples elementos 'form'. Cada elemento form tiene un atributo 'id' que debe usarse en el requerimiento saveformdata. Además cada form tiene múltiples elementos 'field'. Cada elemento field tiene un atributo 'id' que pueden usarse también en el requerimiento saveformdata. El elemento field tiene además los siguientes elementos:
  - label: Etiqueta del elemento de formulario
  - value: Valor que fue recogido en el formulario

### Requerimiento "setcontact"

Este requerimiento ordena asignar el ID de contacto para la llamada entrante, en el caso de que existan múltiples contactos cuyo número telefónico coincida con el Caller-ID anunciado para la llamada entrante. No es necesario que la llamada esté activa al momento de asignar.

### Atributos o argumentos:

- agent\_number: Agente que está asignando el contacto. Este identificador está de la forma Agent/9000.
- agent\_hash: Hash de autenticación para el agente.
- call\_id: ID en la base de datos de la llamada que se asigna su contacto
- contact\_id: ID en la base de datos del contacto a asociar con esta llamada.

### Requerimiento "saveformdata"

Este requerimiento ordena almacenar la información recogida de los formularios, y la asocia a una llamada en particular.

#### Atributos o argumentos:

- agent\_number: Agente que manda a guardar los datos. Este identificador está de la forma Agent/9000
- agent\_hash: Hash de autenticación para el agente.
- campaign\_type: Tipo de la campaña, uno de 'incoming' u 'outgoing'
- call\_id: ID en la base de datos de la llamada que se almacena
- forms: Colección de múltiples elementos 'form':
  - form: Posee el siguiente atributo:
    - id: ID en la base de datos del formulario a usar. Este valor se puede obtener con las llamadas getcallinfo y getcampaigninfo. Contiene múltiples elementos 'field':
    - field: Posee el siguiente atributo:
      - id: ID en la base de datos del campo de formulario a usar. Este valor se puede obtener con las llamadas getcallinfo y getcampaigninfo. Contiene como texto de la etiqueta el valor a almacenar en la base de datos.

### Requerimiento "hold"

Este requerimiento se usa para poner una llamada en hold. Internamente se manda la llamada a una de las extensiones de parqueo. Al poner la llamada en hold, el abonado remoto debería escuchar música en espera, y el agente escucha la música de la cola, pero no recibirá más llamadas.

#### Atributos o argumentos:

- agent\_number: Agente que se va a pausar. Este identificador está de la forma 'Agent/9000'.
- agent\_hash: Hash de autenticación para el agente.

### Requerimiento "unhold"

Este requerimiento se usa para recuperar la llamada que se haya mandado a hold, y volver a hablar con el abonado.

#### Atributos o argumentos:

- agent\_number: Agente que se va a pausar. Este identificador está de la forma 'Agent/9000'.
- agent\_hash: Hash de autenticación para el agente.

#### Requerimiento "transfercall"

Este requerimiento se usa para transferir la llamada que atiende el agente, a otra extensión o a otro número externo. Al momento de transferir la llamada, el agente queda libre para recibir otra llamada, como si se hubiese colgado la llamada. Actualmente no está soportada la transferencia hacia otro agente de campaña, o hacia una cola de campañas.

#### Atributos o argumentos:

- agent\_number: Agente que se va a pausar. Este identificador está de la forma 'Agent/9000'.
- agent\_hash: Hash de autenticación para el agente.
- extension: Número de la extensión o número externo al cual mandar la llamada.

#### Requerimiento "atxfercall"

Este requerimiento se usa para transferir la llamada que atiende el agente, a otra extensión o a otro número externo, con confirmación del número transferido.

Al momento de transferir la llamada, el agente se conecta con el número destino de la transferencia, y si el agente usa el requerimiento hangup para colgar, queda libre para recibir otra llamada. Actualmente no está soportada la transferencia hacia otro agente de campaña, o hacia una cola de campañas.

#### Atributos o argumentos:

- agent\_number: Agente que se va a pausar. Este identificador está de la forma 'Agent/9000'.
- agent\_hash: Hash de autenticación para el agente.
- extension: Número de la extensión o número externo al cual mandar la llamada.

#### Requerimiento "pauseagent"

Este requerimiento se usa para poner al agente en pausa debido a un break conocido.

El ID a usar para el break se obtiene con el requerimiento "getpauses".

Atributos o argumentos:

- agent\_number: Agente que se va a pausar. Este identificador está de la forma 'Agent/9000'.
- agent\_hash: Hash de autenticación para el agente.
- pause\_type: ID de base de datos de la pausa (break) a usar.

Requerimiento "unpauseagent"

Este requerimiento se usa para sacar al agente de la pausa en la cual se haya puesto previamente con el requerimiento "pauseagent".

Atributos o argumentos:

- agent\_number: Agente que se va a sacar de pausa. Este identificador está de la forma 'Agent/9000'.
- agent\_hash: Hash de autenticación para el agente.

Requerimiento "getpauses"

Este requerimiento se usa para obtener el listado de las pausas (breaks) que el agente puede usar para entrar en una pausa auditada.

Atributos o argumentos:

Este requerimiento no tiene atributos:

Respuesta:

- pause: Elemento múltiple. Contiene el siguiente atributo:
  - id: ID en la base de datos de la pausa
- Contiene las siguientes etiquetas:
  - name: Nombre corto de la pausa
  - status: Estado de la pausa. Puede valer 'A' para activa, 'I' para inactiva. Sólo deben usarse las pausas activas.
  - type: Tipo de pausa. Actualmente sólo se devuelven las pausas de tipo 'B' (break). La pausa de tipo 'H' (hold) está reservada para uso interno.
- description: Descripción larga de la pausa.

Requerimiento "hangup"



Este requerimiento manda a colgar la llamada en curso que está atendiendo el agente. Se requiere que el agente esté logoneado y tenga una llamada activa.

Atributos o argumentos:

- `agent_number`: Agente que se va a sacar de pausa. Este identificador está de la forma 'Agent/9000'.
- `agent_hash`: Hash de autenticación para el agente.

Requerimiento "schedulecall"

Este requerimiento intenta crear un nuevo registro de llamada en la campaña a partir de la llamada que está atendiendo actualmente. Este requerimiento sólo funciona si el agente está atendiendo una llamada de una campaña saliente. Para la llamada reagendada, es posible especificar un intervalo de fechas (dentro del intervalo de fechas de la campaña) donde se debe de intentar la llamada nueva, así como un intervalo de horas en la que se intenta la llamada (también dentro del intervalo de horas de la campaña). También es posible marcar la llamada para que se intente contactar al mismo agente que atendió la llamada original, siempre que esté definido el contexto "llamada-agendada" en los contextos de Asterisk. Opcionalmente (por paridad con capacidades de la consola web para el agente) es posible especificar un número distinto para marcar, así como un nuevo nombre de contacto.

Los atributos que se hayan definido en la carga de las llamadas, se heredan al agendar la llamada. Los valores recogidos en formularios (si existen) para la llamada NO SE HEREDAN en la llamada agendada.

Atributos o argumentos:

- `agent_number`: Agente que va a agendar la llamada. Este identificador está de la forma 'Agent/9000'. Se espera que el agente esté atendiendo una llamada de campaña saliente.
- `agent_hash`: Hash de autenticación para el agente.
- `schedule`: Elemento que agrupa los siguientes subelementos:
  - `date_init`: Fecha desde la cual debe intentarse llamada agendada. Formato YYYY-MM-DD.
  - `date_end`: Fecha hasta la cual debe intentarse llamada agendada. Formato YYYY-MM-DD.
  - `time_init`: Inicio de horario en que se intenta la llamada agendada. Formato HH:MM:SS.
  - `time_end`: Final de horario en que se intenta la llamada agendada. Formato HH:MM:SS.

Si se especifica el elemento 'schedule', todos los subelementos son obligatorios.

Si no se especifica 'schedule', se asume que la llamada puede realizarse en cualquier momento dentro del horario de la campaña. Como detalle de implementación, la llamada tenderá a realizarse al final de todas las otras llamadas.

- **sameagent:** Si se especifica 1, se marcará la llamada para que se dirija al mismo agente que atendió la llamada original. Esto requiere obligatoriamente especificar el horario con el elemento 'schedule'. Si 'sameagent' se omite o se especifica como 0, la llamada será ruteada a cualquier agente disponible en el momento de ser realizada.
- **newphone:** Si se especifica, se marcará a este número en lugar de al número asociado a la llamada original.
- **newcontactname:** Si se especifica, el primer atributo de la llamada (que se asume es el nombre del contacto) se sobrescribe con el texto indicado.

Mantenido por compatibilidad con la consola web.

#### Requerimiento "getagentqueues"

Este requerimiento lista las colas a las que está suscrito un agente. Esta información era anteriormente reportada por los eventos agentloggedin y agentloggedout pero fue extraída a un requerimiento separado por razones de eficiencia de implementación

Atributos o argumentos:

- **agent\_number:** Agente que se interroga. Este identificador está de la forma 'Agent/9000'.

Respuesta:

- **queues:** Lista de colas a la que pertenece el agente. Este elemento contiene múltiples elementos 'queue', que contienen cada uno una cola a la que pertenece el agente.

#### Requerimiento "getmultipleagentqueues"

Este requerimiento cumple una función parecida a la del requerimiento getagentqueues. La diferencia principal es que el requerimiento getmultipleagentqueues permite preguntar la información de una cantidad arbitraria de agentes simultáneamente. El uso de este requerimiento se recomienda en lugar de una secuencia de llamadas a getagentqueues cuando se requiere la información de colas de múltiples agentes.

#### Atributos o argumentos:

- **agents:** Lista de agentes para los que se requiere el reporte de pertenencia a las colas. Este elemento contiene múltiples elementos 'agent\_number', cada uno de los cuales identifica al agente, en formato Agent/9000.

#### Respuesta:

- **agents:** Lista de agentes para los que se reporta la pertenencia a las colas. Este elemento contiene múltiples elementos 'agent', los cuales contienen los siguientes campos:
  - **agent\_number:** Agente identificado por el elemento
  - **queues:** Lista de colas a las cuales pertenece el agente. Este elemento contiene múltiples elementos 'queue', los cuales contienen el identificador de la cola.

#### Requerimiento "getagentactivitysummary"

Este requerimiento lista, para un rango de fechas, un reporte de actividad de cada uno de los agentes activos. Este reporte de actividad incluye el total de segundos de todas las sesiones, el total de llamadas recibidas y el total de segundos de llamada en cada una de las colas, y el inicio y final de la última sesión y de la última pausa del agente.

#### Atributos o argumentos:

- **datetime\_start:** Fecha inicial del rango, en formato yyyy-mm-dd. Si se omite, se asume la fecha actual.
- **datetime\_end:** Fecha final del rango, en formato yyyy-mm-dd. Si se omite, se asume la fecha actual.

#### Respuesta:

- **agents:** Lista de agentes encontrados en el rango de fechas indicado. Este elemento contiene múltiples elementos 'agent', los cuales contienen los siguientes campos:
  - **agentchannel:** Agente descrito por el elemento. Este identificador está de la forma 'Agent/9000'.
  - **agentname:** Nombre del agente descrito.
  - **logintime:** Segundos que el agente ha pasado en sesiones durante el intervalo especificado. No se incluyen sesiones activas en el momento de la petición.
  - **lastsessionstart:** Si hay al menos una sesión en el intervalo

especificado, este elemento indica el inicio de la última sesión, en formato yyyy-mm-dd hh:mm:ss. De otro modo no aparece.

- lastsessionend: Si hay al menos una sesión en el intervalo especificado, y la sesión ha terminado, este elemento indica el final de la última sesión, en formato yyyy-mm-dd hh:mm:ss. De otro modo no aparece. Si el agente está actualmente en una sesión activa, lastsessionstart estará presente, y lastsessionend NO estará presente.
- lastpausestart: Si hay al menos una pausa en el intervalo especificado, este elemento indica el inicio de la última pausa, en formato yyyy-mm-dd hh:mm:ss. De otro modo no aparece. No hay garantía alguna de que la última pausa informada sea posterior al último inicio de sesión.
- lastpauseend: Si hay al menos una pausa en el intervalo especificado, y la pausa ha terminado, este elemento indica el final de la última pausa, en formato yyyy-mm-dd hh:mm:ss. De otro modo no aparece. Si el agente está actualmente en pausa, lastpausestart estará presente, y lastpauseend NO estará presente.
- callsummary: elemento que contiene dos subelementos, 'incoming' y 'outgoing' según los tipos de campañas. Cada uno de los elementos tiene cero o más elementos 'queue'. Cada elemento queue tiene un atributo 'id' que indica el número de la cola. Además del atributo, existen los siguientes elementos:
  - sec\_calls: suma del tiempo en segundos en llamadas atendidas que entraron a través de la cola indicada bajo el tipo de campaña indicado.
  - num\_calls: número de llamadas atendidas que entraron a través de la cola.

#### Requerimiento "getchanvars"

Este requerimiento lista, para el agente indicado, las variables de canal definidas para la llamada activa que esté atendiendo. Este requerimiento permite construir contextos y/o AGIs que asignen variables de canal que luego pueden obtenerse a través de este requerimiento.

#### Atributos o argumentos:

- agent\_number: Agente que va a agendar la llamada. Este identificador está de la forma 'Agent/9000'.
- agent\_hash: Hash de autenticación para el agente.

#### Respuesta:

- clientchannel: Canal Asterisk correspondiente a la llamada remota que se atiende. Por ejemplo: SIP/troncal/5551234-abcdef. Este es el canal del cual se reportan las variables de canal.
- chanvars: Lista de variables de canal encontradas en el canal de la llamada. Este elemento contiene múltiples elementos 'chanvar', los cuales contienen los siguientes campos:
  - label: Nombre de la variable de canal
  - value: Valor de la variable de canal

#### Requerimiento "callprogress"

Este requerimiento activa o desactiva la emisión de eventos de progreso de llamada. Por omisión los eventos de progreso de llamada están desactivados, para compatibilidad con versiones anteriores de clientes ECCP. Si un cliente puede hacer uso de los eventos, debe activarlos con este requerimiento.

#### Atributos o argumentos:

- enable: Valor de bandera: 1 para activar, 0 para desactivar

#### Requerimiento "campaignlog"

Este requerimiento lista los eventos de progreso de una campaña determinada, en un intervalo de tiempo. Si se omite el intervalo, se asume el día actual del sistema.

#### Atributos o argumentos:

- campaign\_type: Tipo de la campaña, uno de 'incoming' u 'outgoing'
- campaign\_id: ID en la base de datos de la campaña a listar. Para campañas salientes, este parámetro es requerido. Para campañas entrantes, se omite este parámetro si se deben listar eventos para llamadas entrantes no asociadas a una campaña.
- queue: Cola entrante a seleccionar. Para campañas salientes, este parámetro se ignora. Para campañas entrantes, este parámetro selecciona los eventos asociados a llamadas entrantes no asociadas a una campaña entrante. Si se indica a la vez queue y campaign\_id, campaign\_id tiene prioridad y queue se ignora.
- datetime\_start: Fecha inicial del rango, en formato yyyy-mm-dd. Si se omite, se asume la fecha actual.
- datetime\_end: Fecha final del rango, en formato yyyy-mm-dd. Si se omite, se asume la fecha actual.

Respuesta:

- logentries: Lista de elementos de tipo logentry. Cada elemento logentry tiene los siguientes campos:
  - id: ID interno de base de datos del registro.
  - datetime\_entry: Fecha y hora en la que se produjo la transición de estado
  - campaign\_type: Tipo de la campaña, uno de 'incoming' u 'outgoing'
  - campaign\_id: ID en la base de datos de la campaña a la que pertenece la llamada. Este valor siempre está presente para una llamada saliente, pero puede no estar presente para una llamada entrante.
  - call\_id: ID en la base de datos de la llamada que se hace referencia en evento.
  - new\_status: El nuevo estado al cual ha llegado la llamada. Los eventos reportados pueden ser uno de los siguientes:
    - Placing: se ha empezado a colocar la llamada saliente. El único parámetro válido es retry.
    - Dialing: se ha empezado a marcar la llamada saliente. Se conocen los valores de retry y trunk.
    - Ringing: la llamada saliente está timbrando y va a conectarse a la cola. Se conocen los valores de uniqueid, retry, trunk.
    - OnQueue: la llamada está en la cola esperando ser asignada un agente. Se conocen los valores de uniqueid, retry, trunk.
    - Failure: la llamada no ha podido ser conectada. Los valores válidos son dependientes del estado en el cual se detectó el fallo.
    - OnHold: la llamada ha entrado en hold. Se conocen los valores de uniqueid, retry, trunk.
    - OffHold: la llamada ha salido de hold. Se conocen los valores de uniqueid, retry, trunk.
    - Success: la llamada ha sido conectada con un agente
    - Hangup: la llamada ha sido desconectada del agente.
    - ShortCall: la llamada ha durado muy poco para una interacción con el agente.
  - retry: Para llamadas salientes, el número de intento de marcado que corresponde a este cambio de estado. Siempre 0 para la llamada entrante.
  - uniqueid: ID único de Asterisk para la llamada correspondiente al intento.

- trunk: Troncal a través de la cual se realiza la llamada saliente, o se recibe la llamada entrante.
- phone: número de teléfono de callerid o marcado
- queue: número de la cola por la que se espera que pase la llamada
- agentchannel: Canal de agente que atendió la llamada. Sólo válido para estados Success, Hangup, ShortCall.
- duration: Duración de la llamada, en segundos. Sólo válido para estados Hangup y ShortCall.

Los campos son esencialmente idénticos a los del evento callprogress, a excepción de id, agentchannel. El estado Success corresponde al evento agentlinked, y el estado Hangup/ShortCall corresponde al evento agentunlinked.

#### Requerimiento "getincomingqueuelist"

Este requerimiento lista las colas entrantes disponibles en el sistema.

#### Atributos o argumentos:

- offset: Opcional. Si se especifica, debe también indicarse limit. Se devuelven registros a partir del número indicado, con el primer registro siendo 0. Se asume 0 por omisión en caso de especificar limit sin offset.
- limit: Opcional. Si se especifica, se devuelve como máximo el número de registros indicado por el parámetro. Si no se especifica, se devuelven todos los registros posibles.

#### Respuesta:

- queues: lista de elementos de tipo queue. Cada elemento queue tiene los siguientes campos:
  - id: ID interno de identificación de la cola entrante
  - queue: Número de la cola entrante
  - status: Uno de los valores 'active' o 'inactive'

#### Requerimiento "getincomingqueestatus"

Este requerimiento recupera información dinámica o volátil sobre la cola entrante indicada. Esta información incluye el estado de los agentes disponibles en la cola de la campaña activa, el número de llamadas en los varios estados reconocidos, y las llamadas activas que todavía no han sido asignadas a agentes. En este aspecto, este requerimiento es parecido a

getcampaignstatus. Sin embargo, este requerimiento devuelve información sobre las llamadas que entraron a la cola pero no fueron asignadas como pertenecientes a una campaña entrante.

Atributos o argumentos:

- queue: Número de la cola entrante
- datetime\_start: Opcional. Si se especifica, las estadísticas que involucren llamadas se calcularán sólo con las llamadas realizadas a partir de la fecha indicada (yyyy-mm-dd)

Respuesta:

- statuscount: elemento que contiene los siguientes atributos:
  - total: Total de llamadas que se han recibido hasta el momento en la cola.
  - onqueue: Cantidad de llamadas conectadas que esperan en la cola a ser asignadas a un agente.
  - success: Cantidad de llamadas que fueron asignadas a un agente.
  - onhold: Para llamadas salientes o entrantes, cantidad de llamadas que han sido puestas en espera.
  - abandoned: Cantidad de llamadas que se desconectaron luego de haber entrado a una cola, pero antes de ser asignadas a un agente.
  - finished: Cantidad de llamadas ya terminadas luego de ser atendidas por un agente.
  - losttrack: Para llamadas entrantes, cantidad de llamadas que no pudieron ser vigiladas hasta terminar porque se reinició el marcador.
- agents: Agentes asociados a la cola de la campaña. Este elemento es una lista que contiene múltiples elementos 'agent'. Cada elemento contiene los siguientes atributos:
  - agentchannel: Canal del agente de la cola (normalmente Agent/NNNN)
  - status: Estado del agente. Puede ser 'online', 'offline', 'oncall', 'paused' como se describe en la respuesta del requerimiento 'getagentstatus'
  - callid: ID en la base de datos de la llamada que atiende el agente. Sólo está definido si el agente está logoneado y atendiendo alguna llamada.
  - callnumber: Número telefónico del cliente al que se conecta la llamada. Sólo está definido si el agente está logoneado y



atendiendo alguna llamada.

- callclient: Canal asignado a la llamada del cliente atendido. Sólo está definido si el agente está logoneado y atendiendo alguna llamada.
- pausename: Nombre de la pausa en la que está el agente. Sólo está definido si el agente está en pausa.
- pauseid: ID de la pausa en la que está el agente. Sólo está definido si el agente está en pausa.
- pausestart: Hora en la que inició la pausa, en formato HH:MM:SS de 24 horas. Sólo está definido si el agente está en pausa.
- queuestart: Hora en la que entró en la cola la llamada atendida por el agente. Sólo definido si el agente está atendiendo una llamada.
- linkstart: Hora en la que la llamada fue conectada con el agente. Sólo definido si el agente está atendiendo una llamada.
- trunk: Troncal a través de la cual se recibe llamada entrante. Sólo definido si el agente está atendiendo una llamada.
- activecalls: Llamadas de la campaña que todavía no han sido asignadas a un agente. Este elemento es una lista que contiene múltiples elementos 'activecall'. Cada elemento contiene los siguientes atributos:
  - callid: ID en la base de datos de la llamada conectada al cliente
  - callnumber: Número telefónico de la llamada conectada al cliente
  - callstatus: Estado de la llamada de la campaña. Puede ser uno de los valores 'OnQueue' (campañas entrantes o salientes)
  - queuestart: Hora en la que la llamada ingresó a la cola. Sóloamente definido para campañas salientes, y si el estado en 'callstatus' es 'onqueue'.
  - trunk: Troncal a través de la cual se recibe llamada entrante, o se realiza la llamada saliente.
- stats: Contadores globales de la cola Este elemento contiene los siguientes atributos:
  - total\_sec: Suma de todas las duraciones en segundos de todas las llamadas completadas.
- max\_duration: Máxima duración de llamada en la cola.

#### Requerimiento "pingagent"

Este requerimiento reinicia el contador de inactividad de un agente específico. Si se usa la característica de intervalo de inactividad de login,

este requerimiento debe de invocarse regularmente para indicar que el cliente está todavía activo.

Atributos o argumentos:

- `agent_number`: Agente que va a agendar la llamada. Este identificador está de la forma 'Agent/9000'.
- `agent_hash`: Hash de autenticación para el agente.

## BIBLIOGRAFÍA

- [1] VOIP Wiki, VoIP, <http://www.voip-info.org/wiki/view/VoIP> fecha de consulta 27 de junio de 2013.
- [2] Van Meggelen, J., Madsen, L., Smith, J., “Asterisk: The Future of Telephony, 2nd edition”, O’Reilly Media, 2007
- [3] Goncalves, F., “Building Telephony Systems with OpenSIPS 1.6”, Packt Publishing, 2010
- [4] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., Schooler, E., “SIP: Session Initiation Protocol”, <http://tools.ietf.org/html/rfc3261> fecha de consulta 8 de agosto de 2013
- [5] Spencer, M., Capouch, B., Guy, E., Miller, F., Shumard, K., “IAX: Inter-Asterisk eXchange Version 2”, <http://tools.ietf.org/html/rfc5456> fecha de consulta 8 de agosto de 2013
- [6] Digium, Inc., “IP PBX”, <http://www.asterisk.org/get-started/applications/pbx> fecha de consulta 8 de agosto de 2014
- [7] Digium, Inc., “Digium Telephony Cards”, <http://www.digium.com/en/products/telephony-cards> fecha de consulta 10 de agosto de 2014
- [8] Digium, Inc., “DAHDI”, <http://www.asterisk.org/downloads/dahdi> fecha de consulta 10 de agosto de 2014

- [9] Digium, Inc., “Get Started – What is Asterisk”, <http://www.asterisk.org/get-started>, traducido del inglés, fecha de consulta 11 de agosto de 2014
- [10] VOIP Wiki, Asterisk channels, <http://www.voip-info.org/wiki/view/Asterisk+channels> fecha de consulta 11 de agosto de 2014
- [11] Digium, Inc., “Asterisk Project Wiki”, <https://wiki.asterisk.org/wiki/> fecha de consulta 12 de agosto de 2014
- [12] Palosanto Solutions, “Información del Producto”, <http://www.elastix.org/index.php/es/informacion-del-producto/informacion.html> fecha de consulta 17 de agosto de 2014
- [13] Palosanto Solutions, “Elastix como plataforma de comunicaciones especiales”, <http://www.elastix.org/index.php/es/informacion-del-producto/casos-de-exito/711-elx-special-comm-platform.html> fecha de consulta 10 de febrero de 2015