



PONTIFICIA **UNIVERSIDAD CATÓLICA** DEL PERÚ

Esta obra ha sido publicada bajo la licencia Creative Commons
Reconocimiento-No comercial-Compartir bajo la misma licencia 2.5 Perú.

Para ver una copia de dicha licencia, visite
<http://creativecommons.org/licenses/by-nc-sa/2.5/pe/>



PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ

FACULTAD DE CIENCIAS E INGENIERÍA



DISEÑO E IMPLEMENTACIÓN DE UN
SISTEMA EXPERTO PARA UNA PLATAFORMA
DE CONTACT CENTER

TESIS PARA OPTAR EL TÍTULO DE
INGENIERO DE LAS TELECOMUNICACIONES
PRESENTADO POR

Aldo Javier Lovera Raffo

LIMA – PERÚ

2008

Resumen

El presente proyecto de tesis consiste en el diseño e implementación de un Sistema Experto para el Contact Center de una aerolínea, que permita al cliente resolver sus inquietudes de forma inmediata. La solución se centrará en mejorar el módulo de correo electrónico de la plataforma haciendo uso del conocimiento extensivo de los Sistemas Expertos, permitiendo cubrir las deficiencias originadas por la gran demanda actual de atención en ese medio.

El primer capítulo está centrado en brindar los conocimientos básicos necesarios para el desarrollo del proyecto.

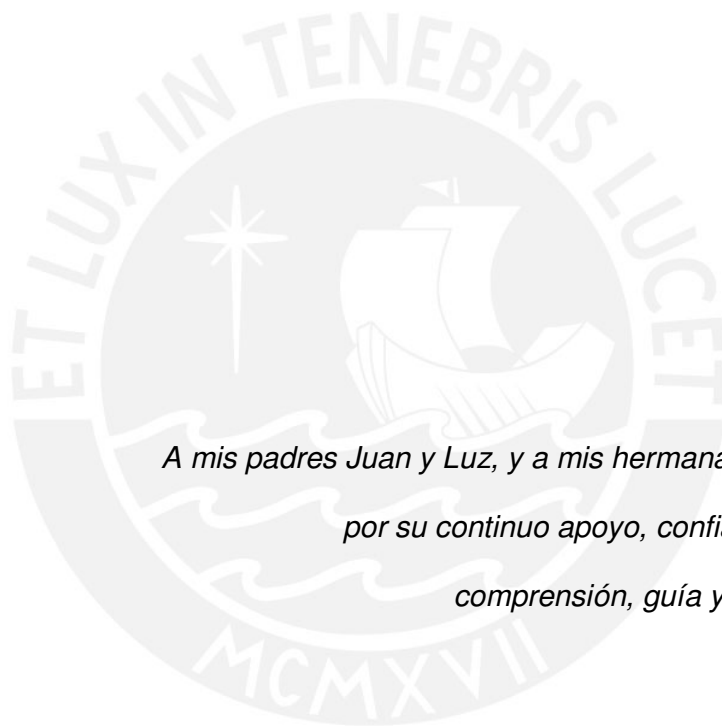
El segundo capítulo busca presentar el Contact Center actual de una aerolínea en particular, realizar el análisis de las necesidades de la empresa y proponer una solución a la medida. Se presentan también los alcances de la tesis.

El tercer capítulo presenta las herramientas, componentes y arquitectura de la solución, así como una breve descripción de su funcionamiento.

El cuarto capítulo, por su parte, describe todo el proceso de implementación del sistema e integración de los servicios.

El quinto capítulo está enfocado al análisis de la confiabilidad del sistema, para conocer si el prototipo se cumple con las necesidades de la empresa.

Por último, se presentan las conclusiones y recomendaciones del presente trabajo, además de proponer algunos trabajos futuros que permitan expandir las funcionalidades de la solución.



*A mis padres Juan y Luz, y a mis hermanas Marie y Givi,
por su continuo apoyo, confianza, paciencia,
comprensión, guía y profundo amor.*

Agradecimientos

Quiero empezar dando gracias a Dios por todas las cosas que me ha dado durante mi vida; una vida saludable, una familia maravillosa y afectuosa, una persona con quien compartir mis sentimientos y experiencias, unos amigos con los cuales puedo contar siempre. Cada una de estas bendiciones me ha permitido culminar el presente proyecto y una de las etapas más importantes de mi vida.

Gracias a mis padres y hermanas, personas muy especiales para mí que me han ayudado a crecer, madurar, valorar lo que la vida me ofrece, confiar en mí mismo y vencer los obstáculos que se me presenten. A ellos les debo todo lo que soy y he logrado en mi vida.

Gracias a mis asesores, Ing. Enrique Larios e Ing. Arturo Díaz, por estar siempre prestos a ayudarme y darme ánimos para seguir con el proyecto hasta el final. No sólo depositaron su confianza en mí durante mi vida universitaria; me brindaron algo más valioso, su gran amistad.

Gracias a Claudia Macedo por su amor, cariño y comprensión. Su apoyo incondicional, fortaleza y confianza me han permitido obtener logros que parecían inalcanzables y nunca rendirme.

Gracias a todos mis amigos, la universidad sin ustedes no hubiera sido la misma. Cada momento que hemos pasado juntos ha sido inolvidable.

Índice

Índice	v
Lista de Figuras	vii
Lista de Tablas.....	viii
Introducción.....	1
Capítulo 1: Conceptos Básicos	2
1.1 CRM.....	2
1.2 El Contact Center	3
1.2.1 Tipos de Contact Centers	4
1.2.2 Modelos de TI de Contact Centers	5
1.2.2.1 Call Center TDM	5
1.2.2.2 Computer Telephony Integration.....	6
1.2.2.3 IP Contact Center	7
1.2.2.4 Contact Center Robusto.....	8
1.2.3 Componentes de un Contact Centers	9
1.2.4 Arquitectura de un Contact Center	10
1.2.5 Funcionamiento de un Contact Center.....	11
1.2.6 Tecnologías Involucradas en un Contact Center	12
1.2.6.1 Universal Queueing	12
1.2.6.2 Automatic Call Distributor (ACD).....	12
1.2.6.3 Interactive Voice Response (IVR)	12
1.2.6.4 Speech Recognition.....	13
1.2.6.5 Text to Speech Synthesis (TTS)	13
1.2.6.6 Dialers.....	14
1.3 Sistemas Expertos	14
1.3.1 Características de un Sistema Experto	16
1.3.2 Ventajas de un Sistema Experto.....	17
1.3.3 Componentes de un Sistema Experto.....	18
1.3.3.1 Componente Humano	18
1.3.3.2 Base de Conocimiento	19
1.3.3.3 Subsistema de Adquisición de Conocimiento.....	20
1.3.3.4 Control de la Coherencia	20
1.3.3.5 Motor de Inferencia.....	20
1.3.3.6 Subsistema de Adquisición de Información	21
1.3.3.7 Interfase de Usuario	21
1.3.3.8 Subsistema de Ejecución de Órdenes.....	21
1.3.3.9 Subsistema de Explicación	21
1.3.3.10 Subsistema de Aprendizaje	22
1.3.4 Tipos de Sistemas Expertos.....	23
1.3.4.1 Rule-based	23
1.3.4.2 Object-oriented.....	24
1.3.4.3 Logic-based	24
1.3.4.4 Probabilísticos	24
1.3.5 Evolución de los Sistemas Expertos.....	25

1.4	Necesidad de un Contact Center utilizando Sistemas Expertos	26
1.5	Casos de éxito	27
Capítulo 2:	Análisis de Requerimientos	29
2.1	Escenario de aplicación del sistema	30
2.2	Historia de la empresa	30
2.2.1	Visión de la empresa	31
2.2.2	Misión de la empresa	31
2.2.3	Valores de la empresa	31
2.3	Situación actual del Contact Center	31
2.3.1	Situación de la demanda	34
2.4	Necesidad de la empresa	35
2.5	Solución Propuesta	36
2.5.1	Alcances del Proyecto	37
Capítulo 3:	Arquitectura de la Solución	39
3.1	Herramientas Tecnológicas	39
3.1.1	Postfix	41
3.1.2	SpamAssassin	42
3.1.3	MySQL	42
3.1.4	JESS	43
3.1.5	Lenguaje de Programación	44
3.2	Componentes de la solución	45
3.3	Funcionamiento	46
3.4	Consideraciones de la Solución	47
3.5	Beneficios de la Solución	48
Capítulo 4:	Implementación del prototipo	50
4.1	Servidor de Base de Datos	50
4.2	Servidor de Correo	53
4.3	Servidor de Aplicaciones	57
4.3.1	Aplicación Java	57
4.3.2	Sistema Experto	60
Capítulo 5:	Plan de Pruebas	63
5.1	Introducción	63
5.2	Desarrollo de pruebas del sistema	64
5.3	Resultados	66
Conclusiones, Recomendaciones y Trabajos Futuros		70
6.1	Recomendaciones	70
6.2	Trabajos Futuros	71
6.3	Conclusiones	72
Bibliografía		73
Anexos		78

Lista de Figuras

FIGURA 1-1: INTEGRACIÓN DE DIFERENTES MEDIOS DE COMUNICACIÓN.....	4
FIGURA 1-2: ARQUITECTURA DE UN CALL CENTER TDM.....	6
FIGURA 1-3: ARQUITECTURA DEL CALL CENTER “FUJITSU” CON TECNOLOGÍA CTI.....	7
FIGURA 1-4: ARQUITECTURA DEL “IP CONTACT CENTER” DE ALTITUDE	8
FIGURA 1-5: ARQUITECTURA DE UN CONTACT CENTER ROBUSTO	9
FIGURA 1-6: ARQUITECTURA DE UN CONTACT CENTER	11
FIGURA 1-7: ÁREAS DE LA INTELIGENCIA ARTIFICIAL.....	15
FIGURA 1-8: RELACIÓN ENTRE EL DOMINIO DEL	16
FIGURA 1-9: COMPONENTES DEL SISTEMA EXPERTO.....	19
FIGURA 1-10: ENCADENAMIENTO DE REGLAS	23
FIGURA 1-11: DIAGRAMA PARA CONSTRUIR UNA RED BAYESIANA	25
FIGURA 2-1: ARQUITECTURA DEL CONTACT CENTER DE LA AEROLINEA	33
FIGURA 2-2: TRÁFICO ENTRANTE DEL SERVIDOR DE CORREOS DE LA PLATAFORMA.....	35
FIGURA 3-1: ARQUITECTURA DE LA SOLUCIÓN PROPUESTA	40
FIGURA 3-2: ARQUITECTURA MODULAR DE POSTFIX	41
FIGURA 3-3: ARQUITECTURA DEL PROTOTIPO	45
FIGURA 3-4: DIAGRAMA DEL FUNCIONAMIENTO.....	46
FIGURA 4-1: ESQUEMA DE LA DB	51
FIGURA 4-2: DISTRIBUCIÓN DE DEMONIOS POSTFIX DURANTE EL FILTRADO DE CONTENIDOS AFTER-QUEUE.....	55
FIGURA 4-3: DIAGRAMA DE FLUJO DE LA APLICACIÓN JAVA	58
FIGURA 4-4: ELEMENTOS DEL JESS	60
FIGURA 4-5: IMPLEMENTACIÓN DE LA BASE DE CONOCIMIENTOS.....	61
FIGURA 5-1: GRÁFICA DE TIEMPO DE ATENCIÓN DE LOS CORREOS	67
FIGURA 5-2: GRÁFICA DE USO DEL SERVIDOR	68

Lista de Tablas

TABLA 4-1: TABLA CORREOS	51
TABLA 4-2: TABLA CLIENTE	52
TABLA 4-3: TABLA PAIS	52
TABLA 4-4: TABLA CIUDAD.....	52
TABLA 4-5: TABLA HORARIO.....	52
TABLA 4-6: TABLA CLASE.....	53
TABLA 4-7: TABLA TARIFA.....	53
TABLA 4-8: TABLA VUELO	53
TABLA 4-9: TABLA CLIENTEXVUELO.....	53
TABLA 4-10: TABLA CLAVES	53
TABLA 4-11: CONTENIDO DEL ARCHIVO /ETC/POSTFIX/MAIN.CF	54
TABLA 4-12: CONTENIDO DEL SCRIPT /ETC/POSTFIX/JAVA_MAIL.SH	55
TABLA 4-13: CONTENIDO DEL SCRIPT /ETC/POSTFIX/MASTER.CF	56
TABLA 5-1: RESULTADOS DE LAS PRUEBAS DE CONFIABILIDAD	66



Introducción

En los últimos años, las empresas buscan brindar un mejor servicio al cliente, de manera que pueda establecer una relación más estrecha con él y obtener, como resultado de ello, beneficios económicos y una ventaja sostenible en el mercado. Diversas empresas en el mundo han encontrado en el Contact Center, una herramienta con la capacidad de interactuar con los clientes sin importar el canal de comunicación.

Por otro lado, el desarrollo de la inteligencia artificial está cambiando poco a poco nuestra forma de vida. Un punto clave es el desarrollo de sistemas que puedan razonar, aprender y obtener conclusiones como un ser humano, e incluso mejor que éste.

En los Contact Centers, los agentes deben atender las consultas y dudas de un gran número de clientes, dedicando a cada uno un tiempo de atención determinado. Este tiempo de atención provocará que otros clientes que hayan ingresado recientemente al sistema tengan que esperar por algún agente libre, produciendo disconformidad y relaciones negativas.

Se requiere realizar una mejora en el Contact Center para que se pueda atender a un número mayor de clientes. Por ello, el tema de la presente tesis consiste en diseñar e implementar un Sistema Experto que pueda integrarse al Contact Center de una empresa que busque eliminar estos defectos de su plataforma de atención al cliente.



Capítulo 1

Conceptos Básicos

1.1 CRM

El Customer Relationship Management (CRM), o “Gestión de Relaciones con el Cliente” es definido como un enfoque empresarial por el cual se puede conocer la conducta de los clientes e influir en ella por medio de una comunicación congruente, para aumentar su nivel de captación, retención, lealtad y rentabilidad [SWI2001]. Es una estrategia utilizada por muchas empresas para incrementar la satisfacción de sus clientes y optimizar la rentabilidad de sus negocios.

Se basa en 4 pilares fundamentales [GRA2001]:

1. Almacenamiento de la información de los clientes de tal forma que podamos mantener y cultivar relaciones a lo largo del tiempo. La

información será demográfica y de contacto, así como de su comportamiento en el uso de los servicios y compra de productos.

2. Análisis de los datos para poder diferenciar un individuo de los demás y poder establecer con él una relación personalizada.
3. Comunicación directa e interactiva con el cliente para construir una relación sólida.
4. Adaptación estructural y funcional en la organización.

El CRM se apoya en nuevas tecnologías para lograr una acelerada toma de decisiones administrativas y aumentar la utilidad de la información recopilada, obteniendo un mejor conocimiento sobre clientes potenciales y leales [MIC2004].

1.2 El Contact Center

Se define como un centro de atención multicanal, donde las interacciones que se produzcan entre la empresa y el cliente pueden provenir de cualquier canal o medio de comunicación (teléfono, chat, e-mail, web, fax, videollamada) [COM2007, CYS2007]. Ello permite centralizar en un único repositorio toda la información de los canales que actualmente se estén utilizando en la empresa. En la figura 1-1 se muestra como los diferentes medios de comunicación se integran en el Contact Center y éste se relaciona con las diferentes áreas de la empresa.



FIGURA 1-1: INTEGRACIÓN DE DIFERENTES MEDIOS DE COMUNICACIÓN

Fuente: “Contact Center” [FUJ2007]

1.2.1 Tipos de Contact Centers

Los Contact Center se comunican con los usuarios de diversas maneras, y dependiendo de que lado inicie la comunicación se puede definir el tipo de Contact Center [FLU2005, AVA2007].

Se tiene un Inbound Contact Center cuando la comunicación es iniciada por el cliente. En estos sistemas el cliente buscará comunicarse con un representante de determinada compañía para obtener asistencia técnica o de emergencia, resolver dudas de facturación, entre otros.

Cuando el Contact Center es quien inicia la comunicación, se tiene un Outbound Contact Center. En la mayoría de casos la comunicación con el cliente se realiza solamente por medios telefónicos, tendiendo a tener una menor complejidad que en el caso anterior.

Son muchas las empresas que buscan implementar un Inbound Contact Center para mejorar las relaciones con sus clientes. Como ejemplo se podría considerar el caso de una Aerolínea que mediante sus diferentes canales de atención pueda resolver las inquietudes de los clientes sobre diversos vuelos y promociones así como un método eficiente de reserva y compra de boletos.

Existe una clase mejorada de Inbound Contact Centers, los “Self Service Contact Centers” [AVA2007]. Aquí la interacción entre el cliente y un agente humano se está dejando de lado en gran parte para pasar a una interacción entre cliente y un sistema automatizado, obteniendo así un ahorro de costos significativo y una rápida atención a los clientes.

1.2.2 Modelos de TI de Contact Centers

Con el pasar de los años, los modelos de TI fueron evolucionando debido a las diferentes tecnologías que surgieron. Algunos de estos modelos son [LAR2007]:

1.2.2.1 Call Center TDM

Este modelo es una infraestructura de hardware sencilla donde el núcleo era una central PBX con la inteligencia para poder derivar la llamada entrante de un cliente a un pool de agentes (Automatic Call Distributor), tal como se muestra en la figura 1-2. Es la solución más simple para atención de clientes; sin embargo, algunos factores como el manejo de las colas de espera y reportes de gestión del sistema, lo convierte en un sistema con muchas limitaciones.

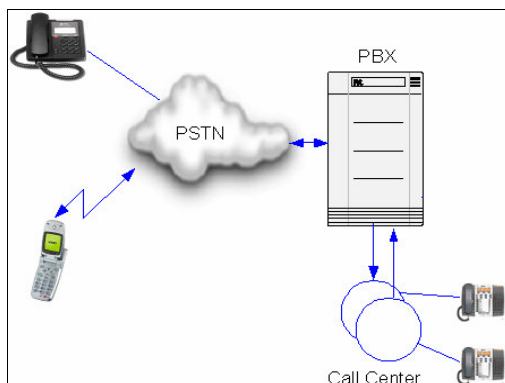


FIGURA 1-2: ARQUITECTURA DE UN CALL CENTER TDM

Fuente: “Una Revisión de los Modelos de Tecnologías de Información de Contact Centers” [LAR2007]

1.2.2.2 Computer Telephony Integration

Posteriormente se buscó la convergencia de voz y datos en los Call Centers, obteniéndose así el modelo Computer Telephony Integration (CTI). En este modelo se cuenta con una mayor cantidad de elementos, como se aprecia en la figura 1-3, siendo el principal, el Servidor CTI, que se encarga de enviar a los agentes del Call Center la información de los clientes en el momento que se produce la llamada.

Este modelo supera las limitaciones del modelo anterior; sin embargo no brinda facilidad para integrar mejoras, no pudiendo cubrir la necesidad naciente de atender nuevos canales de comunicación.

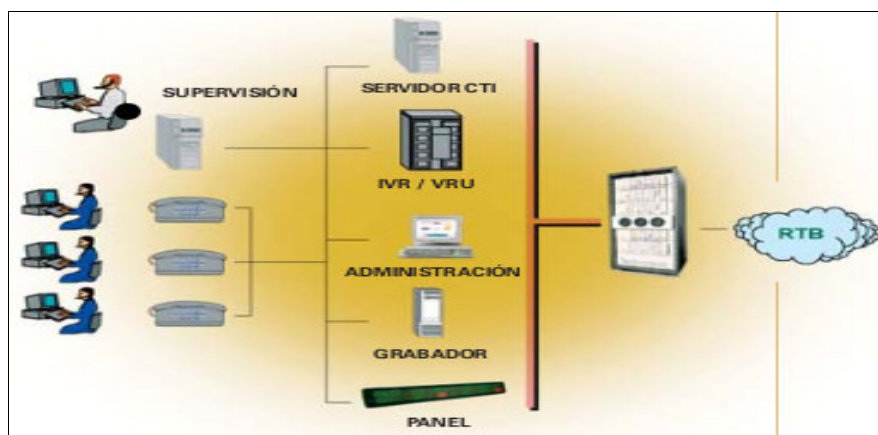


FIGURA 1-3: ARQUITECTURA DEL CALL CENTER “FUJITSU” CON TECNOLOGÍA CTI

Fuente: “Contact Center” [FUJ2007]

1.2.2.3 IP Contact Center

Debido al uso masivo de Internet y del comercio electrónico, la atención telefónica era cada vez la opción menos llamativa. Los clientes exigían nuevas vías de interacción con las organizaciones, y éstas se vieron en la necesidad de habilitar nuevos canales de atención.

Los modelos de Call Center eran en ese momento muy rígidos, no se podían implementar nuevas mejoras fácilmente; es así que surge la necesidad de transformar estos modelos empleando tecnología IP para alcanzar la flexibilidad deseada. Este nuevo modelo (mostrado en la figura 1-4) permitía, entre otras cosas, integrar los nuevos canales de atención, pasando a ser llamado Contact Center.

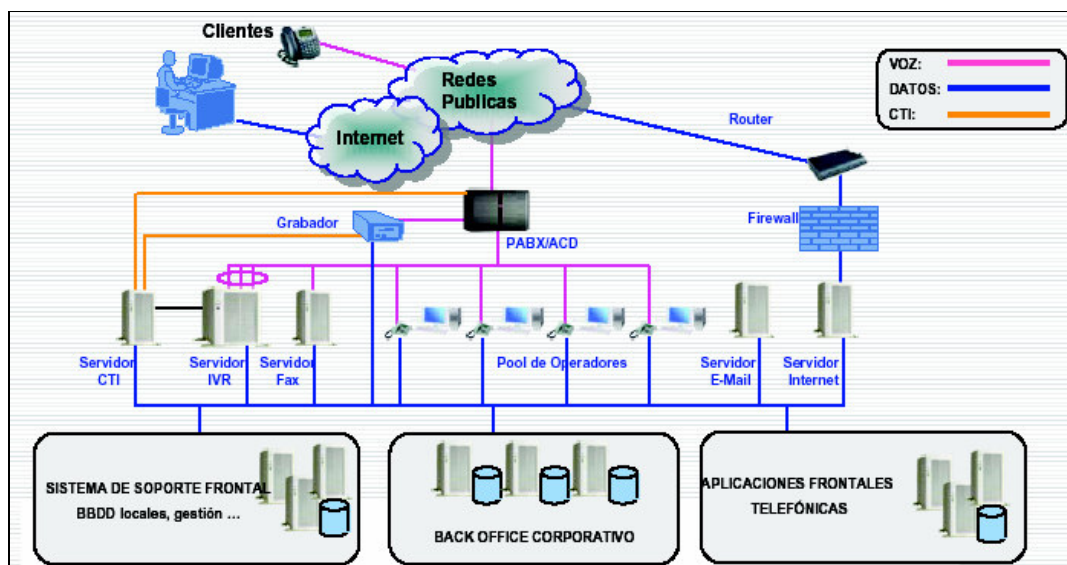


FIGURA 1-4: ARQUITECTURA DEL “IP CONTACT CENTER” DE ALTITUDE

Fuente: “Como implantar un Contact Center de Excelencia” [ALT2007]

1.2.2.4 Contact Center Robusto

Surgieron nuevas consideraciones en el diseño de los Contact Centers. Se requería de una plataforma de alta confiabilidad y disponibilidad que además realice el balanceo de carga entre los equipos responsables de las actividades críticas. Es así, como surge el Contact Center Robusto que, como se aprecia en la figura 1-5, consiste en un sistema redundante que garantiza el funcionamiento de la plataforma en caso de fallas o sobrecarga.

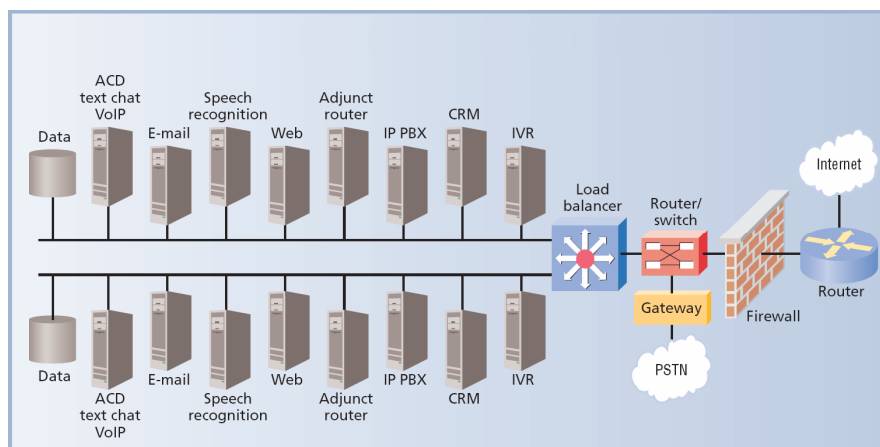


FIGURA 1-5: ARQUITECTURA DE UN CONTACT CENTER ROBUSTO

Fuente: “Una Revisión de los Modelos de Tecnologías de Información de Contact Centers” [LAR2007]

1.2.3 Componentes de un Contact Centers

Los principales componentes de un Contact Center son [CLE2007, FLO2001]:

- **Gateway VoIP.** Dispositivo que realiza la conversión de la señal de voz tradicional de las llamadas provenientes de la PSTN en paquetes IP (SIP, H.323) y viceversa.
- **IP-PBX o Centralita IP.** Dispositivo hardware o software que se encarga de conmutar el tráfico telefónico de VoIP.
- **Web Server.** Servidor que administra las aplicaciones web y maneja las sesiones de agentes, supervisores y administradores. Un Contact Center cuenta con servidores de diferentes aplicaciones de acuerdo a los canales de comunicación con los que trabaje, por que también podríamos tener servidores de chat, correo electrónico, fax, etc.

- **Contact Center Server.** Software gestor de contactos que selecciona el mejor método para distribuir a los contactos en los grupos de agentes, independientemente del medio de comunicación empleado.
- **File Server.** Dispositivo que almacena archivos físicos como grabaciones, scripts de emails, sesiones de chat, etc.
- **Database Server.** Dispositivo que contiene una extensa base de datos del sistema, sus configuraciones, y la data histórica de los clientes.
- **Gateway Análogo.** Dispositivo que nos permite reutilizar la telefonía tradicional POTS.
- **IP-Phones / SoftPhones.** Terminales de los agentes para la realización o recepción de llamadas que pueden ser hardware o software. En el caso de los Softphones, son una aplicación multimedia que trabaja con VoIP y que le permitirá al agente realizar y recibir llamadas desde la PC.

1.2.4 Arquitectura de un Contact Center

Se muestra a continuación la arquitectura básica de un Contact Center, donde se puede observar los componentes expuestos en el punto anterior.

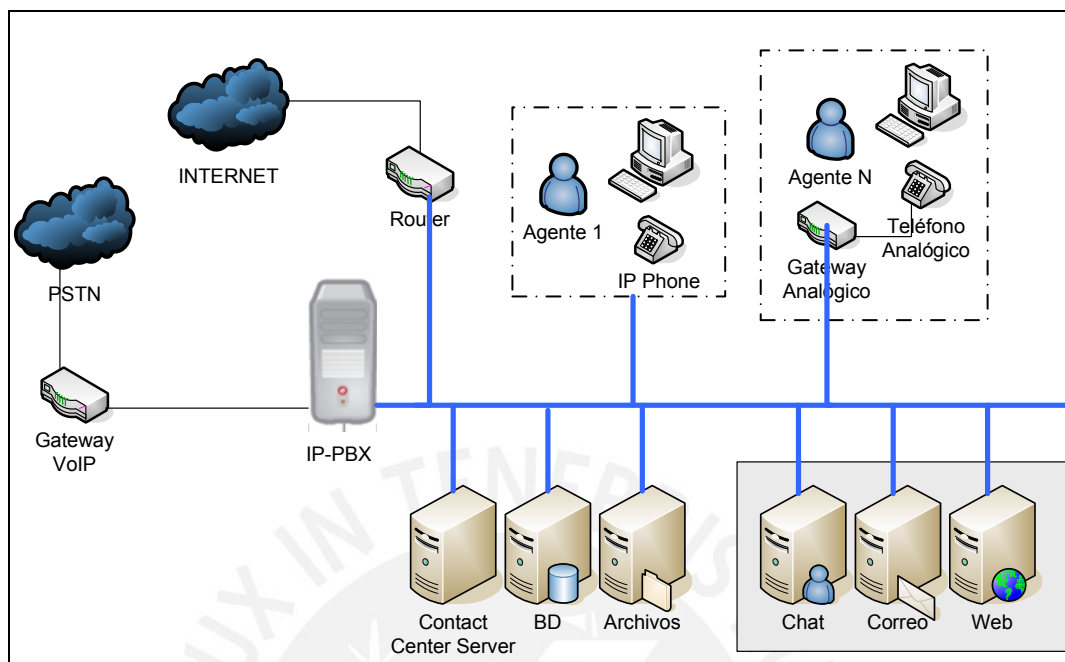


FIGURA 1-6: ARQUITECTURA DE UN CONTACT CENTER

1.2.5 Funcionamiento de un Contact Center

Cuando una llamada llega desde la PSTN pasa a través del Gateway IP que convertirá la señal telefónica en paquetes IP (protocolos SIP, H.323 entre otros) que serán transferidos a la Centralita IP. Si el cliente accede por otros canales, realizará una conexión con el servidor respectivo.

El Contact Center Server identificará a cada uno de los contactos que han ingresado, así como el medio de comunicación empleado para luego determinar el agente calificado que atenderá a un determinado cliente. Una vez tomada la decisión, se establece la comunicación entre el cliente y el agente pero al mismo tiempo el Contact Center Server almacena los datos de la transacción y envía la información del cliente al agente y supervisor respectivo [CLE2007].

1.2.6 Tecnologías Involucradas en un Contact Center

Se muestra a continuación algunas de las tecnologías empleadas por los Contact Centers:

1.2.6.1 Universal Queueing

Tecnología por la cual todos los canales y medios de contacto (telefonía, fax, web, correo electrónico, mensajería instantánea, entre otros) son integrados en una misma cola para un procesamiento y manejo estandarizado [FLU2005, GAR2007].

1.2.6.2 Automatic Call Distributor (ACD)

Sistema diseñado para procesar eficientemente grandes cantidades de llamadas entrantes y asegurar que sean atendidas en un tiempo determinado. Al principio, los ACDs asignaba las llamadas en una distribución First-Come First-Served (FCFS) al agente que ha estado disponible por un largo tiempo; las mejoras implementadas luego permitían asignar llamadas basadas en reglas de enrutamiento definidas por la organización [FLU2005, DAT2007, STR2007].

1.2.6.3 Interactive Voice Response (IVR)

Sistema automático de procesamiento de llamadas programado para responder con mensajes de voz preestablecidos a entradas DTMF del teléfono del cliente. Los IVRs permitirán que el cliente pueda navegar y seleccionar la información

deseada, para esto el cliente deberá presionar determinadas teclas en respuesta a las opciones que el IVR anuncia.

Actualmente este sistema de voz está migrando a uno de video: Interactive Voice & Video Response (IV²R). El cliente, además de escuchar el menú del IVR, podrá observarlo y recibir un video de presentación mientras espera a ser atendido por el agente entre otras funcionalidades [FLU2005, STR2007, SUR2007].

1.2.6.4 Speech Recognition

Proceso de convertir un mensaje hablado en texto, con el uso de un sistema capaz de extraer la información lingüística de las emisiones de un ser humano, compararla con parámetros de referencia (mediante un entrenamiento del mismo) y en base a reglas, decidir la expresión apropiada. Sus aplicaciones son similares a los IVRs, pero la toma de decisiones no es en base a entradas DTMF sino a lo que el cliente dice [FLU2005, FLO2006].

1.2.6.5 Text to Speech Synthesis (TTS)

Proceso inverso al Speech Recognition. Consiste en obtener voz de manera sintética, una transcripción fonética de un texto junto con la prosodia asociada.

Es un proceso multidisciplinar, pues para implementar un sistema que realice una síntesis adecuada se deben tener conocimientos de Lingüística, Fonética, Prosodia, Procesamiento del Lenguaje Natural, Procesamiento Digital de Señales, Modelado de Sistemas.

Entre sus aplicaciones se tienen los Sistemas de Respuesta Automatizadas, donde estas respuestas con dinámicas o no han sido grabadas previamente [FLU2005, FLO2006, ICM2007].

1.2.6.6 Dialers

Sistema que automatiza el inicio y discado de llamadas en un Outbound Contact Center y las conectan a los agentes. Empleando Predictive Dialers, el sistema realiza un gran número de intentos y conectan al cliente que ha contestado la llamada con un agente, generando a veces tiempos de espera en el cliente; empleando Preview Dialers, el agente por medio de un botón realiza la llamada evitando así generar un tiempo de espera [FLU2005, COS2007].

1.3 Sistemas Expertos

Los sistemas expertos (SE), tal como se muestra en la figura 1-7, son una rama de la Inteligencia Artificial que hace un uso extensivo del conocimiento especializado para resolver algún problema al nivel de un humano experto, al nivel de una persona que es especialista en el tema [GIA1998].

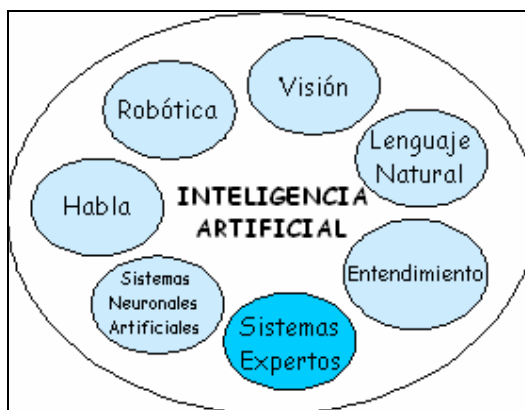


FIGURA 1-7: ÁREAS DE LA INTELIGENCIA ARTIFICIAL

Fuente: “Expert Systems Principles and Programming” [GIA1998]

Actualmente el término “Sistema Experto” es usado para referirse a todo sistema que emplee tecnologías de sistemas expertos. Estas tecnologías pueden estar conformadas por lenguajes de sistemas expertos (p.e. CLIPS, JESS), programas y hardware especialmente diseñados para el desarrollo y ejecución de Sistemas Expertos.

La mayoría de personas consideran sinónimos los términos como *expert system*, *knowledge-based system* (sistema basado en conocimiento) y *knowledge-based expert system* (sistema experto basado en conocimiento). Eso implicaría que un sistema que no contenga especialización alguna, tan solo conocimiento general, sea mal llamado “Sistema Experto” [KRI1996].

El conocimiento que tenga una persona experta sobre la resolución de problemas específicos es conocido como el *dominio del conocimiento* del experto, este dominio está limitado por el *dominio del problema*, tal como se muestra en la figura 1-8. Un sistema experto, al ser una imitación del experto,

no tendrá conocimiento sobre otros *dominios de conocimiento* por más que se encuentren dentro del mismo *dominio del problema* [GIA1998].



FIGURA 1-8: RELACIÓN ENTRE EL DOMINIO DEL CONOCIMIENTO Y DEL PROBLEMA

Fuente: "Expert Systems Principles and Programming" [GIA1998]

1.3.1 Características de un Sistema Experto

Un sistema experto es usualmente diseñado para tener las siguientes características [CAS1998, GIA1998, WIL2007]:

- **Alta Performance.** El sistema debe ser capaz de dar una respuesta de gran calidad, igualando o superando la respuesta de un experto en el tema.
- **Adecuado tiempo de respuesta.** El sistema debe entregar una respuesta en un tiempo igual o menor al que requiere el experto en tomar una decisión.
- **Buena confiabilidad.** El sistema debe ser confiable y no propenso a fallas.
- **Entendible.** El sistema debe de poder explicar paso a paso el razonamiento seguido durante la ejecución, en lugar de ser solo una

caja negra. Esta característica es muy importante pues permitirá que el sistema experto justifique sus conclusiones y durante una depuración del sistema, verificar el correcto razonamiento.

- **Flexibilidad.** El sistema, debido a la gran cantidad de conocimiento que puede contener, debe contar con algún mecanismo eficiente para añadir, cambiar o eliminarlos.

1.3.2 Ventajas de un Sistema Experto

Un gran número de ventajas vuelven a los SE muy atractivos [CAS1998, GIA1998, WIL2007]:

- **Disminuyen costos.** El costo de entregar respuestas por usuario es muy bajo.
- **Disminuyen riesgos.** Podemos usar los sistemas en entornos que son muy peligrosos para el ser humano.
- **Son Permanentes.** A diferencia del humano experto que puede retirarse, renunciar o fallecer, los conocimientos del sistema siempre estarán ahí.
- **Especialización múltiple.** El conocimiento de muchos expertos puede estar contenida dentro del sistema, aumentando el Dominio del Conocimiento del mismo.
- **Incrementan la confiabilidad.** El sistema puede corroborar las decisiones tomadas por otro sistema, incrementando la confiabilidad de que una decisión es correcta.

- **Explicación.** El sistema puede explicar en detalle todo el razonamiento que lo llevo a una determinada conclusión. Un experto humano puede estar muy cansado o sin ganas de explicar paso a paso su razonamiento.
- **Respuesta continua.** En aplicaciones de tiempo-real, se obtendrá siempre una respuesta, a diferencia del humano experto que puede reducir su eficiencia por stress o fatiga.
- **Tutor inteligente.** El sistema puede actuar como un tutor inteligente dejando que el estudiante ejecute programas y entienda el razonamiento.
- **Base de datos inteligente.** El sistema puede ser usado para acceder a una base de datos de una manera inteligente

1.3.3 Componentes de un Sistema Experto

Como vemos en la figura 1-9, los componentes típicos de un SE son [CAS1998]:

1.3.3.1 Componente Humano

Conformado por un grupo de expertos que suministran el conocimiento de un determinado tema, un grupo de ingenieros del conocimiento que traducen el conocimiento otorgado a un lenguaje que el sistema experto pueda comprender y los usuarios finales del sistema. La colaboración de este grupo humano es considerada como el elemento más importante en el desarrollo del sistema experto.

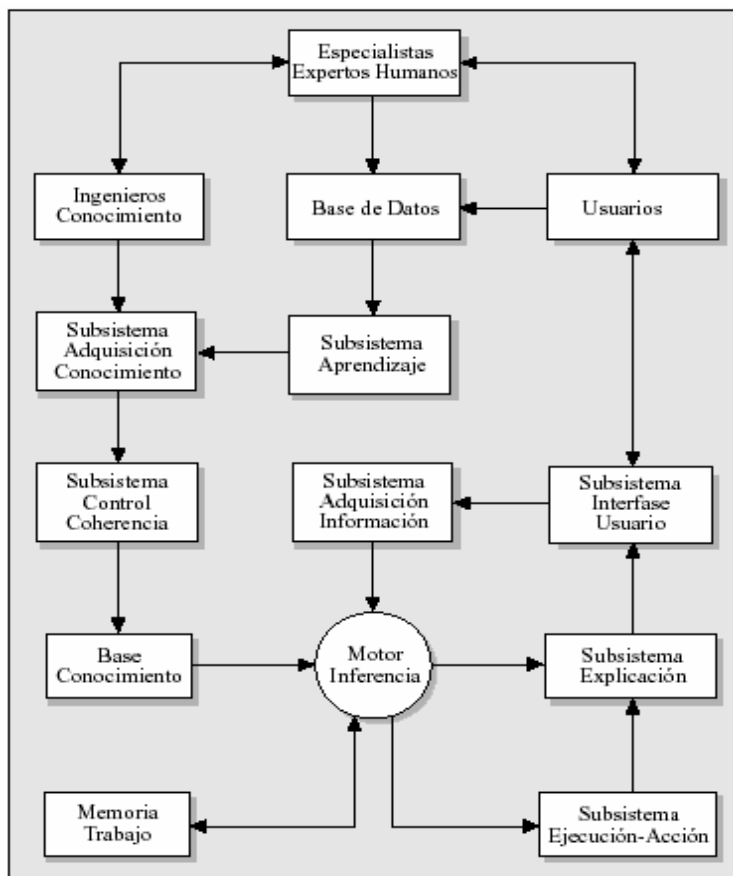


FIGURA 1-9: COMPONENTES DEL SISTEMA EXPERTO

Fuente: "Sistemas Expertos y modelos de Redes Probabilísticas" [CAS1998]

1.3.3.2 Base de Conocimiento

Elemento ordenado y estructurado suministrado por los expertos que contiene los conocimientos que se requieren para el desarrollo del sistema. Se requiere que los expertos humanos repiensen, reorganicen, y reestructuren la base de conocimiento.

Se debe tener presente que al hablar del conocimiento nos referimos a afirmaciones de validez general como reglas, distribuciones de probabilidad, etc. mientras que al hablar de datos nos referimos a la información relacionada

con una aplicación particular. Los primeros se almacenan permanentemente en la *Base de Conocimiento*, mientras que los segundos se almacenan en la *Memoria de trabajo*.

1.3.3.3 Subsistema de Adquisición de Conocimiento

Controla el flujo de nuevos conocimientos que va desde el experto humano hacia la *Base de Datos*, determinando si el conocimiento recientemente obtenido es redundante o no. En caso que no lo sea, la información es llevada a la *Base de Conocimiento* para su almacenamiento.

1.3.3.4 Control de la Coherencia

Es un componente esencial que controla la consistencia de la *base de datos* y evita que un conocimiento inconsistente brindado por el experto ingrese. Informa sobre las restricciones que se debe cumplir para que el conocimiento sea coherente con lo existente en la *base de conocimiento*, ayudando a los expertos humanos a dar información estable.

1.3.3.5 Motor de Inferencia

Corazón de todo sistema experto cuya función principal es obtener conclusiones aplicando el conocimiento contenido en la *Base de Conocimiento* a los datos.

Las conclusiones del *motor de inferencia* pueden estar basadas en conocimiento determinista o conocimiento probabilístico. En el último caso, la

propagación de incertidumbre es la tarea principal del motor de inferencia, que permite sacar conclusiones bajo incertidumbre.

1.3.3.6 Subsistema de Adquisición de Información

Obtiene información del usuario que es empleada por el *Motor de Inferencia* como conocimiento adicional para poder continuar con el proceso de inferencia y llegar a una conclusión, esto ocurre cuando el conocimiento inicial del sistema es muy limitado.

Esa información es obtenida por medio de la *Interfase de Usuario* y debe comprobarse la consistencia de la misma antes de ser almacenada.

1.3.3.7 Interfase de Usuario

Enlace entre el sistema experto y el usuario que por medio de mecanismos eficientes permitirá mostrar y obtener la información de forma fácil y agradable.

1.3.3.8 Subsistema de Ejecución de Órdenes

Componente que permite al sistema experto iniciar acciones basadas en las conclusiones obtenidas por el motor de inferencia. La explicación de las razones de porque se inician estas acciones se dan a conocer al usuario por medio del *Subsistema de Explicación*.

1.3.3.9 Subsistema de Explicación

Componente que explica el proceso seguido por el motor de inferencia o por el subsistema de ejecución.

En muchos casos, es necesaria la explicación de las conclusiones debido a los riesgos asociados con las acciones a ejecutar.

1.3.3.10 Subsistema de Aprendizaje

Este componente permitirá que nuestro sistema pueda aprender. El aprendizaje puede ser estructural o paramétrico. Por aprendizaje estructural nos referimos a algunos aspectos relacionados con la estructura del conocimiento (reglas, distribuciones de probabilidad, etc.) como la inclusión de una nueva regla en la *base de conocimiento*. Por aprendizaje paramétrico nos referimos a estimar los parámetros necesarios para construir la base de conocimiento.

Los sistemas expertos pueden también obtener experiencia a partir de los datos disponibles. Estos datos pueden ser obtenidos por expertos y no expertos y pueden utilizarse por el *subsistema de adquisición de conocimiento* y por el *subsistema de aprendizaje*.

Después de analizar cada uno de los componentes de los sistemas expertos, podemos ver que estos sistemas pueden realizar varias tareas como:

- Adquirir conocimiento y verificar la coherencia del mismo, ayudando así expertos humanos a dar conocimiento coherente.
- Almacenar (memorizar) conocimiento.
- Preguntar cuando se requiere un nuevo conocimiento.
- Aprender de la base de conocimiento y de los datos disponibles.

- Realizar inferencias y razonamiento en base a conocimientos deterministas y de incertidumbre.
- Explicar las conclusiones o acciones tomadas.

1.3.4 Tipos de Sistemas Expertos

De acuerdo a la forma en la cual abordan los problemas, tenemos los siguientes tipos de SE:

1.3.4.1 Rule-based

Dan solución a problemas deterministas. La base de conocimiento contiene el conjunto de reglas que definen el problema (figura 1-10), y el motor de inferencia obtiene las conclusiones utilizando un mecanismo de razonamiento lógico sobre estas reglas [CAS1998].

Es una forma simple y poderosa de representar el conocimiento.

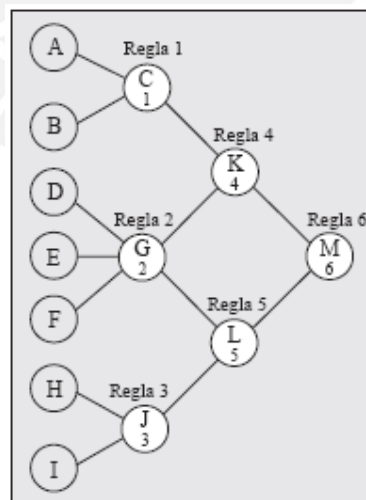


FIGURA 1-10: ENCADENAMIENTO DE REGLAS

Fuente: "Sistemas Expertos y modelos de Redes Probabilísticas" [CAS1998]

1.3.4.2 Object-oriented

Se definen objetos y métodos para representar el problema, estos métodos se encargarán capturar más conocimiento y actualizar el conocimiento base [WAS1989].

A diferencia de los SE basados en reglas, no se requiere de grandes cantidades de reglas para resolver un problema complejo.

1.3.4.3 Logic-based

En este caso, se establecen declaraciones de los elementos (símbolos) del problema a resolver y luego se definen algunas Reglas de Inferencia. El sistema generará todas las inferencias posibles [KRI1996].

Su capacidad de realizar una búsqueda más precisa es limitada, se requeriría de muchas reglas de inferencia y declaraciones para alcanzar el estado deseado.

1.3.4.4 Probabilísticos

Dan solución a problemas que generan incertidumbres. Se emplea el razonamiento probabilístico para poder eliminar esas incertidumbres, definiendo una distribución conjunta de las variables para describir relaciones de dependencia entre ellas y emplear fórmulas probabilísticas para llegar a alguna conclusión. Para poder definir la distribución conjunta de variables, se emplean modelos de redes probabilísticas, entre los cuales tenemos a las redes de Markov y de Bayes [CAS1998], que parten de una representación gráfica de las relaciones entre las variables, como vemos en la figura 1-13.

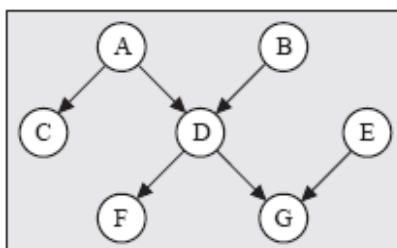


FIGURA 1-11: DIAGRAMA PARA CONSTRUIR UNA RED BAYESIANA

Fuente: "Sistemas Expertos y modelos de Redes Probabilísticas" [CAS1998]

1.3.5 Evolución de los Sistemas Expertos

Los sistemas expertos para propósitos reales surgieron en los años 70. El primero de ellos fue el DENDRAL, desarrollado por Edgard Feigenbaum en LISP durante 10 años, cuyo dominio de conocimiento era la inferencia de estructuras moleculares (Química). Si bien no separaba el conocimiento del motor de inferencia, fue un buen punto de partida para muchos programadores de la época. Posteriormente surgió MYCIN (1974), sistema desarrollado por Ted Shortliffe en LISP para el diagnóstico de enfermedades infecciosas de la sangre y receta de medicinas de acuerdo a las características de la persona.

Después de estos 2, surgieron muchos otros sistemas que presentaban muchas mejoras y resolvía problemas de diferentes disciplinas como la medicina interna (INTERNIST), geología (PROSPECTOR), computación (XCON). Para el año 1989 ya se tenían alrededor de 2200 sistemas expertos, generando una inversión por parte de los negocios de un billón de dólares [CAS1998, GIA1998, WIL2007].

Al mismo tiempo que los Sistemas Expertos evolucionaban, también lo hacían los diferentes lenguajes de programación, ofreciendo cada vez más facilidades a los programadores para implementar SE. Entre los más difundidos se tienen al

PROLOG (PROGramation et LOGique), CLIPS (C Language Integrated Production System) y JESS (Java Expert System Shell).

Actualmente los sistemas expertos ofrecen diferentes soluciones, tal es el caso de la empresa italiana Expert System [EXP2007], que desde 1989 ofrece soluciones SE para el manejo óptimo de la data de las empresas. Se deja de lado la Gestión del conocimiento para pasar a la Inteligencia Semántica, donde el texto es analizado por el SE de la misma forma que una persona lo haría.

1.4 Necesidad de un Contact Center utilizando Sistemas Expertos

Para las empresas es muy importante y necesario contar con un Contact Center, esta plataforma permite integrar todos los canales de comunicación que emplea la organización para relacionarse con sus clientes, aprovechando toda la información que tiene de los mismos para brindarles un mejor servicio, afianzar su lealtad y optimizar la rentabilidad del negocio.

En la actualidad existen muchos Contact Centers conformados por sistemas automatizados (IVR, IV2R, entre otros), que según estudios recientes de la compañía norteamericana de investigación de mercados Leo J. Shapiro y Associates, genera en los clientes mucha satisfacción al recibir respuestas rápidas a sus dudas e información clara, detallada y precisa de lo que buscan [COR2007]. Esto también genera satisfacción a la empresa, debido a que estos sistemas le brindan ahorro de personal, permiten atender a muchos clientes y generar mejores tratos con ellos.

Sin embargo, en casos en los cuales el cliente requiera de una interacción con el agente, por más eficiente que éste sea, sólo podrán atender un número determinado de consultas en un tiempo dado. En estos casos, la solución directa estaría en aumentar el número de agentes para evitar cuellos de botella, pero no es una solución rentable.

Es en este escenario donde podríamos hacer uso de los Sistemas Expertos, es decir, implementar una mejora en el Contact Center instalando en algunos de los módulos de atención un sistema que imite al agente, atendiendo ambas las solicitudes de los clientes por diferentes canales al mismo tiempo

El Contact Center, al emplear un Sistema Experto, contará con un agente virtual que brindará respuestas de igual o mayor calidad que el agente humano en un tiempo mucho menor lo que le permitirá atender a muchos más clientes. Además este agente virtual no es susceptible a la fatiga o stress que pueda sentir su contraparte humana, entregando siempre una respuesta continua en aplicaciones de tiempo real. Todo esto otorga grandes beneficios a la empresa.

1.5 Casos de éxito

Se muestra a continuación algunas soluciones de la empresa italiana Expert System [EXP2007], líder en implantar Sistemas Expertos semánticos para mejorar plataformas Contact Center existentes.

- Implementó un Sistema Experto para la empresa Linear Asssicurazioni, corredora de seguros italiana. La necesidad de implementar el sistema

COGITO® Contact Mail para el ordenamiento y respuesta de correo electrónico fue consecuencia del crecimiento de la empresa. Los correos que llegaban eran analizados y procesados; dependiendo de su contenido el SE generaba una respuesta automática (en caso de preguntas frecuentes), semiautomática (el operador interviene en una porción del mensaje) o customizada (el mensaje es enviado a un operador e incrementa el Conocimiento Base del sistema.

- Implementó un SE para la empresa americana AskMeNow, líder en sistemas inteligentes de respuesta por celular. Cuando un cliente quiere información de alguna materia, la solicita por SMS; el sistema realiza una búsqueda y responde por el mismo medio. Se implementó la solución COGITO® Contact Mobile, que permitía al cliente realizar pedidos de una forma más flexible, no limitarlo a determinadas expresiones. El SE tiene la capacidad de entender, extraer, filtrar y categorizar la información contenida en los mensajes y enviar una respuesta adecuada por SMS. Las mejoras del sistema se reflejaron en una mayor acogida por parte de los usuarios y en una reducción de costos de entrenamiento de personal.



Capítulo 2

Análisis de Requerimientos

La finalidad del presente proyecto de tesis es mejorar una plataforma de comunicaciones, conocida como Contact Center, utilizando el conocimiento especializado de los sistemas expertos para buscar que la atención al cliente sea más rápida y eficiente, además de generar ahorro para la empresa en personal humano.

Para el desarrollo del prototipo se debe de tener en cuenta a la empresa cliente, su giro de negocio, necesidades y el tipo de Contact Center con el que cuentan.

2.1 Escenario de aplicación del sistema

Para el presente proyecto se creó una empresa peruana ficticia perteneciente al sector aviación comercial llamada AlcaLines. Esta empresa se basa en el perfil de LAN Airlines, aerolínea chilena considerada como una de las de mayor flujo de pasajeros en Sudamérica y entre las 40 con mayores ingresos en el mundo [LAN2007].

Se eligió esta empresa para el desarrollo de la aplicación debido a que el rubro al que pertenece busca constantemente alternativas que permitan mejorar su posicionamiento en el mercado peruano e incrementar la satisfacción de sus clientes, así como contar con ventajas competitivas.

Partimos del supuesto de que la empresa cuenta actualmente con un Contact Center IP, este cuenta con los módulos de telefonía, mensajería instantánea, correo electrónico, navegación web y ha permitido mejorar la calidad del servicio de atención al cliente con respecto a consultas, reservas y compras; sin embargo, la empresa considera que es necesario recurrir a diferentes tecnologías para lograr mayores mejoras en la plataforma.

2.2 Historia de la empresa

AlcaLines inicia sus operaciones en el mes de abril del 2003 y a la fecha, con más de 4 años de experiencia, es la línea aérea peruana con mayor participación en el mercado nacional (57%) y de gran renombre en el mercado sudamericano (Brasil, Ecuador, Colombia, Chile, Argentina, Bolivia). [LAN2007]

Nació en base al esfuerzo, trabajo y creatividad de sus fundadores y trabajadores, brindando en sus inicios servicios a diversas partes del territorio nacional (Lima, Cuzco, Arequipa, Juliaca, Iquitos, Piura, Tacna, Trujillo) y luego de 3 años a diversas capitales de Sudamérica (Sao Paulo, Quito, Bogotá, Santiago de Chile, Buenos Aires, La Paz).

AlcaLines se distingue en el mercado nacional e internacional por su servicio de calidad y su sistema de atención al cliente.

2.2.1 Visión de la empresa

“Ser una de las diez mejores aerolíneas del mundo, brindando a nuestros clientes un servicio de primera.”

2.2.2 Misión de la empresa

“Ofrecer a nuestros pasajeros un viaje que sea expedito y agradable de principio a fin, con la posibilidad de elegir entre múltiples destinos.”

2.2.3 Valores de la empresa

Integridad, Seguridad, Honradez, Calidad de Servicio y Eficiencia.

2.3 Situación actual del Contact Center

La aerolínea cuenta con un Contact Center IP ubicado en Lima. Este Contact Center muestra una arquitectura basada en estándares abiertos, permitiendo

compatibilidad con servidores comerciales, y la mayoría de sus componentes están implementados en software libre.

El Contact Center IP, tal como se muestra en la figura 2-1, consta de los siguientes componentes:

- Una Centralita IP Asterisk [AST2007] que implementa las funcionalidades de IVR, monitoreo de llamadas, CDR, Call Recording, Call ID, ACD, Dialers predictivos para llamadas outbound.
- Un Contact Center Server SCU (Servidor de Contactos Universales) [CLE2007] para la gestión unificada de las comunicaciones multimedia (telefonía, correo electrónico, chat) por medio del enrutamiento basado en habilidades.
- Un Servidor Web Apache [APA2007] que almacena las distintas páginas que conforman el sitio web de la empresa. El servidor cuenta con diversos módulos que hacen posible el uso de Webphones y ventanas de Chat embebidas.
- Un Servidor de Correo Postfix [POS2007] que gestiona el envío y recepción de correos electrónicos, tanto de sus clientes como proveedores.
- Un Servidor de Mensajería Instantánea Jabber [JAB2007] que gestiona las comunicaciones de Chat (por medio de ventanas embebidas en el sitio web) entre los clientes y los agentes del Contact Center.
- Un Servidor de Base de Datos MySQL [MYS2007] que mantiene la información de los clientes y su data histórica.

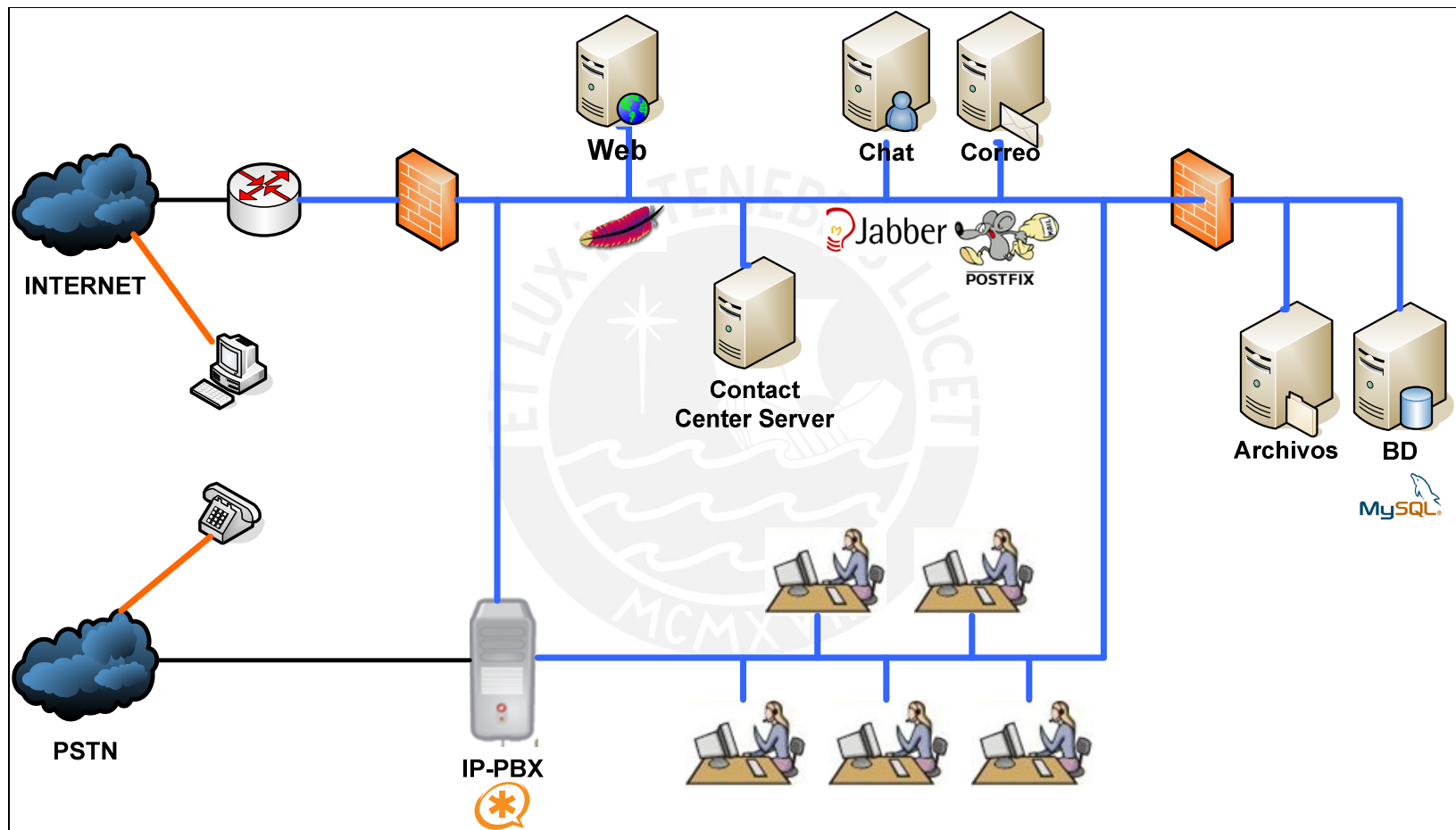


FIGURA 2-1: ARQUITECTURA DEL CONTACT CENTER DE LA AEROLINEA

- Un Servidor de Archivos que almacena grabaciones de las llamadas, registro llamadas, sesiones de Chat, entre otros archivos importantes.

El Contact Center es de tipo inbound/outbound ya que atiende las comunicaciones entrantes para resolver consultas de los clientes y gestionar reservas y compras de pasajes; y establece comunicaciones salientes para ofertar promociones a sus clientes más representativos (servicio diferenciado) y realizar encuestas que permitan mejorar la calidad de servicio de la empresa. La atención brindada por los 3 medios de comunicación es a través de la interacción con agentes; para cubrir toda la demanda, en el horario de 8 a.m. a 8 p.m. se cuenta con 30 agentes (15 en simultáneo que laboran medio tiempo) y en el horario restante se cuenta con 12 agentes (6 en simultáneo que laboran medio tiempo), teniendo 42 agentes en total.

2.3.1 Situación de la demanda

AlcaLines se encuentra en un continuo aumento del número de clientes gracias al prestigio que ha ganado durante estos años, a los novedosos programas como AlcaPass, AlcaOferta y a la flexibilidad de medios para que el cliente realice sus operaciones. Esto ha producido un incremento de las comunicaciones que ingresan al Contact Center, en especial el módulo de correo electrónico. En los últimos 3 meses, el flujo de correos entrantes por consultas de horarios y tarifas de los diferentes vuelos ha aumentado considerablemente; la figura 2-2 muestra el tráfico entrante de correos de la plataforma, donde se observa que de los 827 correos que aproximadamente se

reciben al día, el 54.17% son correos de consultas en general y el 45.83% restante son correos no deseados o SPAM.

De acuerdo a las estrategias de crecimiento de la empresa, se estima un aumento similar del flujo de correos entrantes para los siguientes meses.

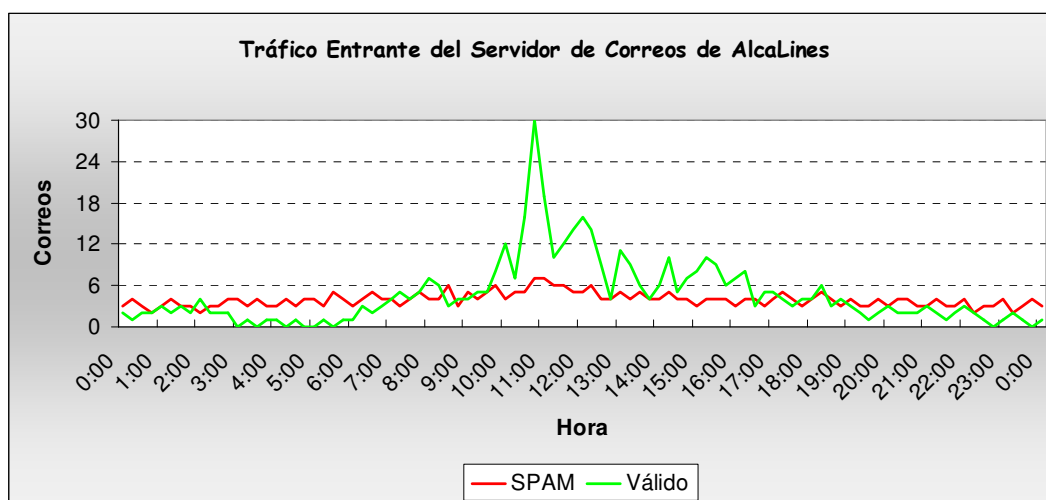


FIGURA 2-2: TRÁFICO ENTRANTE DEL SERVIDOR DE CORREOS DE LA PLATAFORMA

2.4 Necesidad de la empresa

Para AlcaLines, la atención rápida y eficiente al cliente es uno de sus principales pilares. Es por ello que ha considerado como parte de su plan estratégico buscar una solución rápida, eficiente y escalable para hacer frente a la creciente demanda de atención por correo electrónico y garantizar continuidad y estabilidad de la plataforma.

Frente a esta necesidad se presentan 2 alternativas: contratar agentes que se dediquen a la atención de clientes por correo electrónico o invertir en un

sistema de respuesta automático eficiente y escalable que atienda ese tipo de consultas.

La alternativa que la empresa elija deberá permitir que el Contact Center esté en condiciones de:

- Recibir y atender ordenadamente los correos de consulta de los clientes respecto a los horarios y las tarifas de los vuelos de la empresa.
- Brindar una respuesta adecuada y coherente.
- Obtener el menor tiempo de respuesta posible.
- Almacenar tanto el correo de consulta como el correo de respuesta.
- Mantener un resumen de las consultas y operaciones del cliente para gestiones futuras de CRM.

2.5 Solución Propuesta

Luego de un análisis de la problemática de la empresa y de las alternativas disponibles, se propone como solución el sistema de respuesta automático. Se tendrá como elemento principal a un Sistema Experto (SE) que permitirá resolver las consultas del cliente de la misma manera que un agente especializado lo haría, y en base a un dominio de conocimiento delimitado por la información que maneja la aerolínea.

La idea de contratar un grupo de agentes para cubrir las necesidades de la empresa no es rentable a largo plazo [CLE2007]. Cada nuevo agente involucra un conjunto de costos interrelacionados como costos operativos, beneficios y seguros laborales que por ley merece; además, conforme crezca la empresa y

aumente la demanda, requeriremos contratar más personal incurriendo en mayores costos para la empresa.

Al invertir en el sistema, a diferencia del caso anterior, sólo se requerirá de una inversión inicial y eventualmente de algunos costos de mantenimiento para ofrecer alta disponibilidad o para ampliar la capacidad del sistema en caso la demanda aumente, haciendo de la propuesta un proyecto más rentable. Otro punto a favor es que esta solución implementa una mejora a la plataforma, permitiendo explotar al máximo el módulo en cuestión. Esto brinda también un valor agregado a la empresa; actualmente el fuerte mercado competitivo exige a las empresas constantes cambios para adecuarse mejor a las necesidades del cliente y son las soluciones TI las que lo hacen posible.

2.5.1 Alcances del Proyecto

- El proyecto buscará implementar un sistema experto que permita al cliente interactuar con un agente virtual del Contact Center de la aerolínea.
- El sistema será un prototipo desarrollado únicamente para el módulo de correo electrónico de la plataforma de la empresa.
- El prototipo deberá dar solución de la manera más óptima a las consultas del cliente sobre tarifas y horarios de los vuelos de la aerolínea.
- El prototipo deberá responder en un tiempo mucho menor al de un agente humano.

- El prototipo deberá remitir las consultas a un agente en caso que no cuente con el conocimiento suficiente para resolverlas.
- Toda información de las consultas será almacenada en Base de Datos para futuros estudios de mercado.





Capítulo 3

Arquitectura de la Solución

La arquitectura de la solución planteada para el Contact Center de la aerolínea AlcaLines se muestra en la figura 3-1. Dado que la plataforma ya cuenta con varios servidores implementados, usaremos algunos de ellos (Base de datos, Correo electrónico) adicionando configuraciones e implementaremos otros para el desarrollo de la solución.

3.1 Herramientas Tecnológicas

Para implementación del prototipo haremos uso de las siguientes herramientas:

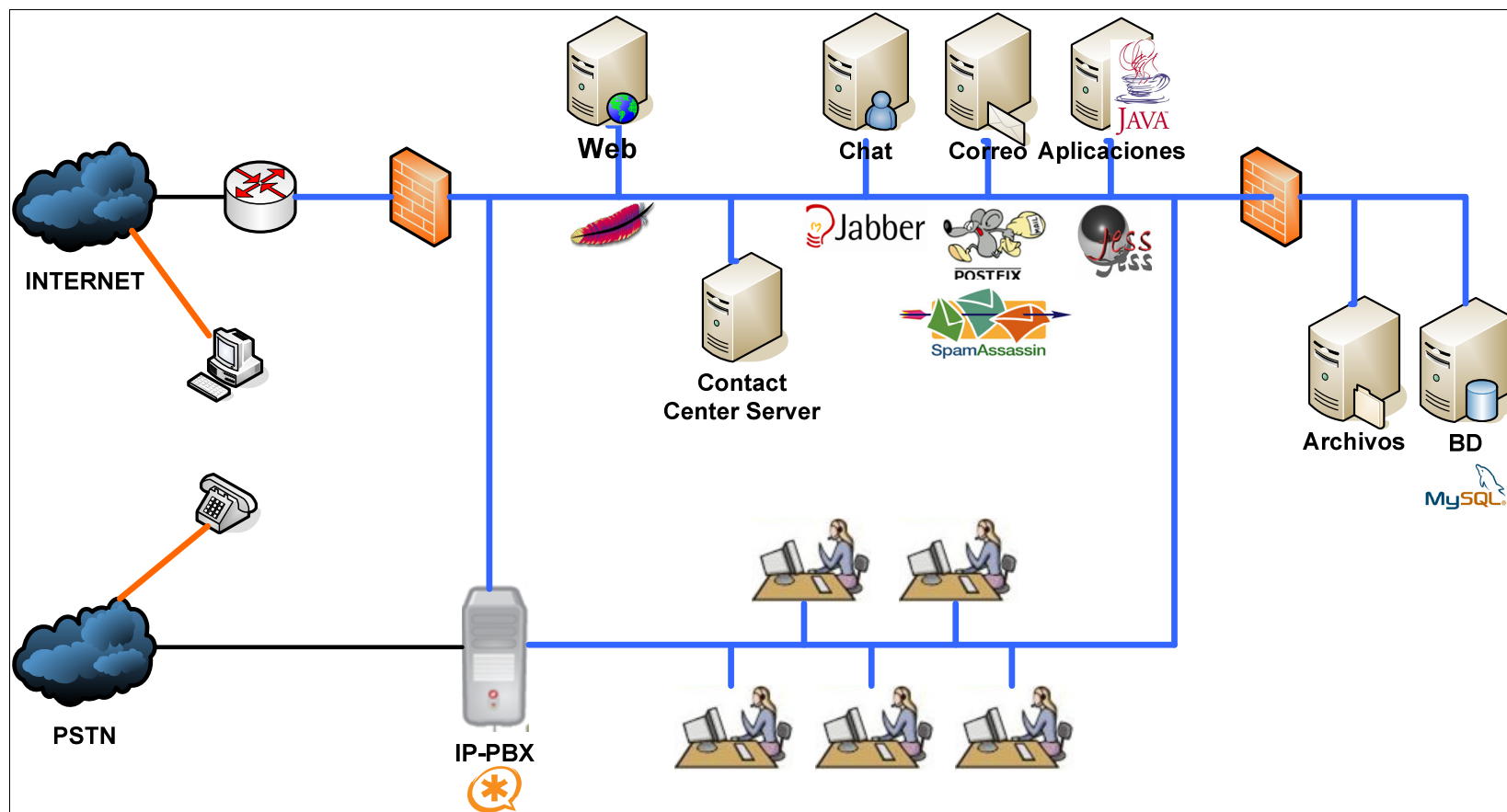


FIGURA 3-1: ARQUITECTURA DE LA SOLUCIÓN PROPUESTA

3.1.1 Postfix

Es un Agente de Transporte de Correos (MTA) de código abierto, un software encargado del enrutamiento y envío de correos electrónicos a sus respectivos destinatarios, soportado por plataformas GNU/Linux y MAC OS X. [POS2007]

Fue desarrollada por Wietse Zweitze Venema [WIE2007] en el centro de investigación de IBM Thomas J. Watson bajo el nombre de IBM Secure Mailer(2001) como una alternativa al Sendmail [SEN2007], ofreciendo una forma más rápida, fácil y segura de administrar el servicio.

Postfix se caracteriza por ser un programa modular, ya que delega el funcionamiento a pequeños módulos (demonios, colas) que realizan una tarea específica, como se muestra en la figura 3-2. Esto hace también que el programa sea más ligero y eficiente en el manejo de recursos que otras alternativas. Otro punto fuerte es la seguridad, cubriendo aquellos agujeros que existían en el Sendmail y brindando compatibilidad para el uso de listas negras [PNG2007].

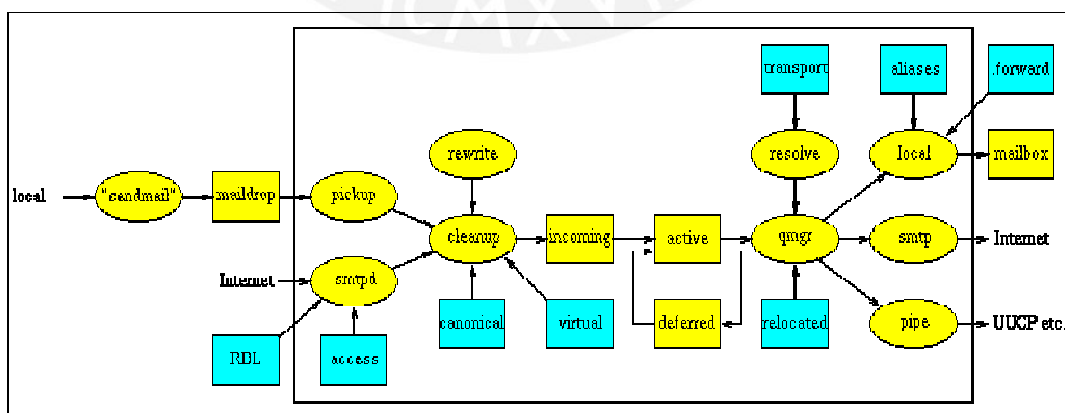


FIGURA 3-2: ARQUITECTURA MODULAR DE POSTFIX

Fuente: "Postfix" [POS2007]

3.1.2 SpamAssassin

Es un filtro inteligente de correos electrónicos de código abierto desarrollado por la Apache Software Foundation [APA2007] y especializado en la detección de correos no deseados.

Es utilizado en la mayoría de MTA existentes (Sendmail, Exim, Postfix) gracias a su arquitectura modular, su facilidad de uso e integración, y su gran capacidad para identificar SPAM. Esto último es posible gracias a un conjunto de programas implementados en lenguaje Perl [PER2008] que realizan pruebas en las cabeceras y contenidos del correo, utilizan algoritmos de búsqueda bayesiana, acceden a listas negras y bases de datos de identificación de SPAM, entre otros mecanismos. [SPA2008]

3.1.3 MySQL

Es un Sistema de Gestión de Base de Datos Relacional (RDBMS) desarrollado por MySQL AB y recientemente adquirido por Sun Microsystems. [MYS2008]

Actualmente, la base de datos MySQL se ha convertido en la base de datos de código abierto más popular del mundo gracias a su rapidez, alta confiabilidad y facilidad de uso. Adicionalmente, es una base de datos multiplataforma (GNU/Linux, MAC OS, Windows, HP-UX, AIX) y la preferida para las aplicaciones de nueva generación basadas en soluciones LAMP (Linux, Apache, MySQL, PHP/Python/Perl). [LAM2007]

Posee una licencia dual, se puede usar como un producto de código abierto bajo los términos de la licencia GNU General Public License o comprar una licencia comercial que viene acompañada de soporte y diversos servicios.

3.1.4 JESS

El Java Expert System Shell es una herramienta que permite implementar Sistemas Expertos (SE) basados en reglas. Esta fue elaborada totalmente en lenguaje Java por Ernest Friedman-Hill en Sandia National Laboratories como parte de un proyecto interno de investigación. [SAN2007]

JESS proporciona tanto un motor de inferencia como un entorno de desarrollo flexible que permitirá elaborar programas en Java con la capacidad de razonar en base al conocimiento insertado mediante reglas. Para la implementación de un SE podemos usar el lenguaje del JESS (línea de comandos) y acceder al API de Java, o utilizar lenguaje Java e importar la librería del JESS [FRI2003]. Una vez definidas las reglas, se aplicarán sobre un conjunto de eventos con la ayuda del razonamiento progresivo [JES2007] y del algoritmo Rete [FOR1982], haciendo el trabajo más rápido que si se contara con un conjunto de sentencias condicionales en cascada.

Existe en el mercado un gran número de motores de inferencia, pero pocos son los que ofrecen robustez, confiabilidad y un entorno de desarrollo de SE. Dentro de este reducido grupo, son contados los que ofrecen una interacción con el lenguaje Java de la forma que JESS brinda. Tal es el caso de Mandarax, una librería Java de código abierto (LGPL) que hace posible la implementación de SE basados en reglas con ayuda del razonamiento regresivo [MAN2008]. Mandarax utiliza un algoritmo diferente al Rete que le permite obtener premisas de una inferencia a partir de un resultado conocido; sin embargo, este tipo de razonamiento no es la mejor opción para nuestra

solución. Nuestro sistema utilizará el razonamiento progresivo para determinar una respuesta en base al conjunto de premisas dadas por el cliente.

Si bien JESS era considerado en sus inicios como un clon de CLIPS [CLI2007] hecho en Java; actualmente se ha convertido en una herramienta diferente, gracias a las continuas mejoras, y altamente utilizada, gracias a su rápido, robusto y ligero motor de inferencia. [JES2007]

A diferencia de las demás herramientas a emplear en la solución, JESS no se encuentra bajo licencia alguna de software libre o de código abierto. Para poder utilizar el JESS, se debe de adquirir una licencia de los laboratorios Sandia, ya sea académica (gratuita) o comercial (costo acondicionado a la aplicación). Esta licencia autoriza la modificación libre del código fuente del JESS, pero no su libre distribución. [JES2007]

3.1.5 Lenguaje de Programación

El lenguaje de programación que será usado para el desarrollo del prototipo es Java, desarrollado por Sun Microsystems en 1991 y dado a conocer en 1995.

Entre sus características principales tenemos que se trata de un lenguaje de programación orientado a objetos (POO) y multiplataforma, no se requiere de mayores cambios para ejecutar un programa en diferentes sistemas operativos.

Actualmente Java es un lenguaje muy popular gracias a su sencillez, potencia, seguridad y fiabilidad; sin embargo, según la aplicación, presenta una velocidad de ejecución lenta con respecto al lenguaje C. [JAV2007]

En mayo del 2007, Sun Microsystems liberó la mayor parte de sus tecnologías Java bajo la licencia GNU General Public License; es por ello que se considera como código abierto.

3.2 Componentes de la solución

Una vez conocidas las herramientas, debemos conocer los componentes necesarios para el desarrollo de nuestra aplicación.

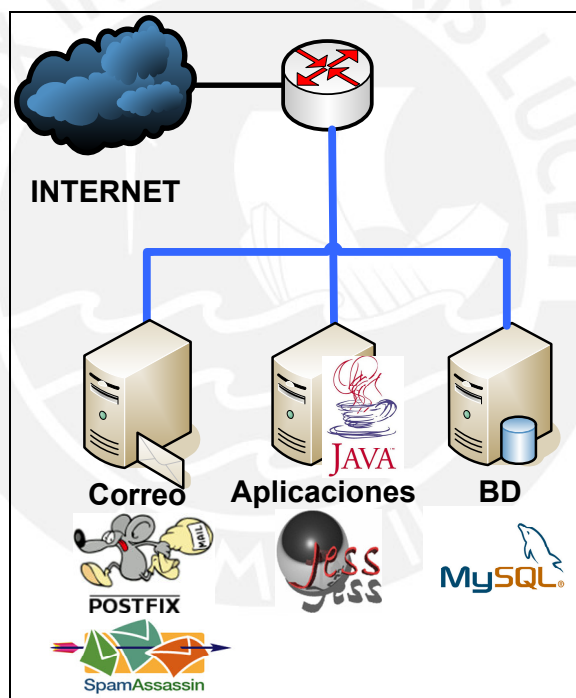


FIGURA 3-3: ARQUITECTURA DEL PROTOTIPO

La arquitectura recomendada, tal como se muestra en la figura 3-3, es la siguiente:

- Un servidor de correos Postfix 2.3 con el filtro de contenidos SpamAssassin 3.2.4.

- Un servidor de base de datos MySQL 5.0.
- Un servidor de aplicaciones donde se ejecutará el programa Java. Debe contar con el Java Development Kit 6 y los paquetes jess.jar y mysql-connector-java-5.0.7-bin.jar para importar las clases que permitan utilizar el JESS 7.0p1 y conectarse con la base de datos MySQL, respectivamente.

Sin embargo, para el desarrollo del prototipo y la elaboración de las pruebas, se implementarán los servicios mencionados dentro de un mismo equipo. El equipo a utilizar es una desktop con procesador Intel Pentium 4 de 2.80GHz y 512 MB de memoria RAM; y los servicios correrán sobre el sistema operativo GNU/Linux Ubuntu 7.10 (Gutsy Gibbon) con el kernel 2.6.22.

3.3 Funcionamiento

El funcionamiento de la aplicación se muestra en la figura 3-4.

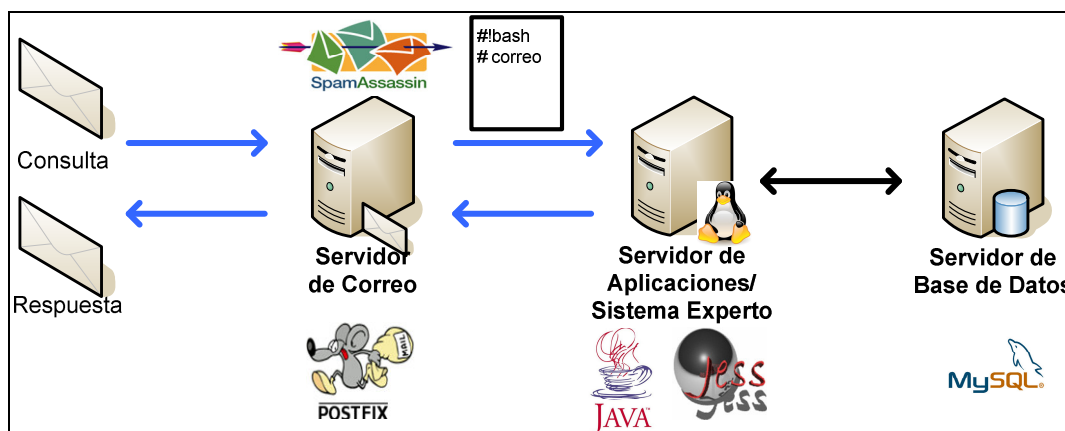


FIGURA 3-4: DIAGRAMA DEL FUNCIONAMIENTO

Cuando un cliente realice consultas de horarios y precios de vuelos por medio del correo electrónico, el servidor Postfix de la aerolínea recibirá y entregará inmediatamente el correo al SpamAssassin, el cual determinará si el correo es no SPAM; una vez marcado el correo, será enviado al servidor de aplicaciones. Para lograrlo, implementaremos un ejecutable en Bash-Scripting [COO2005] que permita imprimir el contenido del correo en un puerto determinado que nuestra aplicación Java se encontrará escuchando.

La aplicación Java recibirá el mensaje y analizará las cabeceras más relevantes; primero verificará que el correo no se encuentre marcado como SPAM, luego validará los datos del cliente en la base de datos para conocer si puede o no utilizar el sistema. Si la respuesta es afirmativa, se procederá al análisis minucioso del mensaje para extraer un conjunto de palabras claves; esta información será almacenada en base de datos y enviada al JESS.

El sistema experto, en base a las palabras claves y al gran conjunto de reglas que obtuvo de la información almacenada en base de datos, formulará una respuesta adecuada a la consulta y la enviará a la aplicación Java. Esta generará el correo de respuesta, lo almacenará en la base de datos y lo remitirá al servidor Postfix para que se encargue de enviárselo al cliente.

3.4 Consideraciones de la Solución

- Se buscará la implementación de un modelo de referencia que, si bien en este caso únicamente se ocupa de brindar información de vuelos,

puede ser ampliado para abarcar otros servicios, tales como reservas o compras de pasajes.

- Este sistema será de uso exclusivo de los clientes de la aerolínea. No se realizará implementación del sistema de registro.
- Se contará con un modelo de base de datos simplificado de la aerolínea. No se incluirá información sobre escalas de viaje o capacidad de pasajeros.
- El sistema solicitará ayuda a un agente físico en caso su base de conocimiento no le permita resolver las consultas. Posteriormente, el agente podrá insertar el nuevo conocimiento para que el sistema aprenda y cada vez se convierta más autosuficiente.

3.5 Beneficios de la Solución

- La solución propuesta permitirá que la plataforma de la aerolínea brinde mayor flexibilidad al cliente en el instante en que realice sus consultas. Desde una computadora o un teléfono móvil, el cliente podrá enviar un correo para obtener información de manera casi inmediata y transparente.
- El costo de inversión de solución es reducido gracias a que la mayoría de herramientas no requieren de licencias para ser utilizadas. A largo plazo, esta alternativa ofrece una mayor rentabilidad que el aumento de personal.

- La solución, al ser implementada con software y estándares abiertos, brinda flexibilidad y escalabilidad, además de compatibilidad con otras soluciones que pueden implementarse a futuro en la plataforma de la aerolínea.
- La solución servirá de base para el desarrollo de mejoras en los diversos módulos del Contact Center de la empresa, haciendo siempre uso del conocimiento extensivo de los Sistemas Expertos.





Capítulo 4

Implementación del prototipo

En la medida de lo posible, se buscará realizar las instalaciones de las herramientas utilizando los repositorios de Ubuntu, permitiendo un adecuado manejo del sistema y mayor facilidad para las actualizaciones.

4.1 Servidor de Base de Datos

Como primer paso de la implementación, se procederá con la instalación de la base datos MySQL. Para esto se utilizará el comando “sudo apt-get install mysql-server-5.0”.

Posteriormente se implementará el modelo Entidad-Relación mostrado en la figura 4-1. Por motivos de simplicidad, el modelo se diseñó como una versión resumida de un esquema de Base de Datos de una aerolínea real, tomando en cuenta sólo la información necesaria para cumplir los alcances del proyecto

como destinos, horarios y tarifas de los vuelos comerciales que ofrece la empresa.

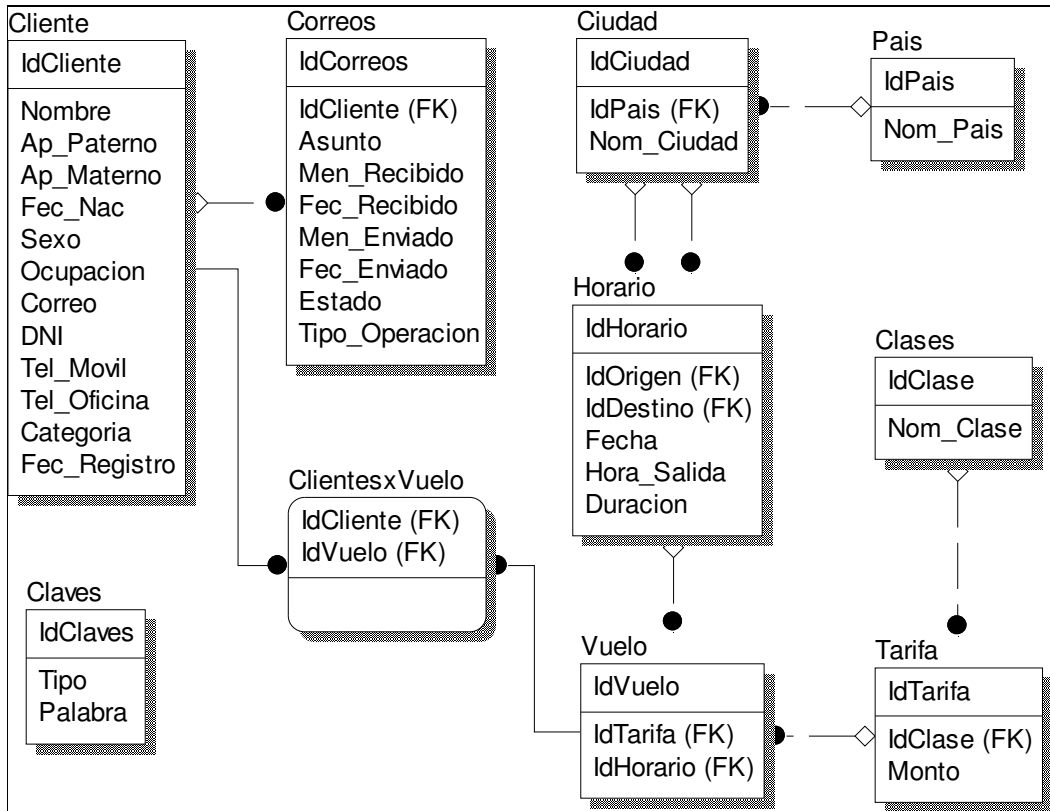


FIGURA 4-1: ESQUEMA DE LA DB

La descripción de cada uno de los campos de las tablas del modelo se detallará a continuación:

TABLA 4-1: TABLA CORREOS

Campo	Tipo Dato	Descripción
IdCorreos	Int	Llave primaria. Identificador del correo.
IdCliente	Int	Llave foránea. Cliente que envió el correo.
Asunto	Varchar	Asunto del correo.
Men_Recibido	Varchar	Mensaje del correo recibido.
Fec_Recibido	Date	Fecha y hora del correo recibido.
Men_Enviado	Varchar	Mensaje del correo enviado.
Fec_Enviado	Date	Fecha y hora del correo enviado.
Estado	Int	Estado del correo: 0 (Pendiente), 1 (Atendido).
Tipo_Operacion	Varchar	Resumen de la consulta del cliente

TABLA 4-2: TABLA CLIENTE

Campo	Tipo Dato	Descripción
IdCliente	Int	Llave primaria. Identificador de cliente.
Nombre	varchar	Nombre del cliente.
Ap_Paterno	varchar	Apellido paterno del cliente.
Ap_Materno	varchar	Apellido materno del cliente.
Fec_Nac	Date	Fecha de nacimiento del cliente.
Sexo	Char	Sexo del cliente.
DNI	varchar	Documento Nacional de Identidad del cliente.
Ocupación	varchar	Ocupación laboral del cliente.
Correo	varchar	Correo electrónico del cliente.
Tel_Movil	varchar	Número del teléfono celular del cliente.
Tel_Casa	varchar	Número del teléfono fijo del cliente.
Tel_Oficina	varchar	Número del teléfono laboral del cliente.
Categoría	Int	Categoría del cliente: 0 (normal), 1 (Premium).
Fec_Registro	Date	Fecha de registro

TABLA 4-3: TABLA PAIS

Campo	Tipo Dato	Descripción
IdPais	Int	Llave primaria. Identificador del país.
Nom_Pais	varchar	Nombre del país.

TABLA 4-4: TABLA CIUDAD

Campo	Tipo Dato	Descripción
IdCiudad	Int	Llave primaria. Identificador de la ciudad.
IdPais	Int	Llave foránea. País de la ciudad.
Nom_Ciudad	varchar	Nombre de la ciudad.

TABLA 4-5: TABLA HORARIO

Campo	Tipo Dato	Descripción
IdHorario	Int	Llave primaria. Identificador del horario de vuelo.
IdOrigen	Int	Llave foránea. Ciudad de origen.
IdDestino	Int	Llave foránea. Ciudad de destino.
Fecha	Date	Fecha del vuelo
Hora_Salida	varchar	Hora de salida del vuelo
Duración	Int	Duración del vuelo.

TABLA 4-6: TABLA CLASE

Campo	Tipo Dato	Descripción
IdClase	int	Llave primaria. Identificador de la clase.
Nom_Clase	varchar	Nombre de la clase.

TABLA 4-7: TABLA TARIFA

Campo	Tipo Dato	Descripción
IdTarifa	int	Llave primaria. Identificador de la tarifa.
IdClase	int	Llave foránea. Clase de la tarifa.
Monto	int	Valor en soles de la tarifa.

TABLA 4-8: TABLA VUELO

Campo	Tipo Dato	Descripción
IdVuelo	int	Llave primaria. Identificador del vuelo.
IdHorario	int	Llave foránea. Horario del vuelo.
IdTarifa	int	Llave foránea. Tarifa del vuelo.

TABLA 4-9: TABLA CLIENTEXVUELO

Campo	Tipo Dato	Descripción
IdCliente	int	Llave primaria. Relación fuerte. Cliente que consulta.
IdVuelo	int	Llave primaria. Relación fuerte. Vuelo consultado.

TABLA 4-10: TABLA CLAVES

Campo	Tipo Dato	Descripción
IdClaves	int	Llave primaria. Identificador de la palabra clave.
Tipo	int	Tipo de palabra clave.
Palabra	varchar	Palabra clave.

4.2 Servidor de Correo

Como siguiente paso de la implementación, se instalarán el servidor de correos Postfix y el filtro de correos SpamAssassin empleando el comando “sudo apt-get install postfix spamassassin”.

Se procederá a modificar el archivo de configuración `/etc/postfix/main.cf`, archivo principal del Postfix que permite adaptarlo a nuestras necesidades. En este archivo, tal como se muestra en la tabla 4-11, se añadirá información sobre el dominio del servidor, las redes a las que el servidor pertenece, el tamaño máximo de buzón de correo local, tipos de notificaciones, entre otros.

TABLA 4-11: CONTENIDO DEL ARCHIVO /ETC/POSTFIX/MAIN.CF

```

...
smtpd_banner = $myhostname ESMTP $mail_name (Ubuntu)
myhostname = arudo-desktop
myorigin = /etc/mailname
mydestination = alcalines.com, arudo-desktop, localhost.localdomain, localhost
mynetworks = 192.168.2.0/24, 127.0.0.0/8
mailbox_size_limit = 0
recipient_delimiter = +
inet_interfaces = all
notify_classes = bounce, resource, software, delay, bounce2
...

```

Para realizar el envío de los correos al servidor de aplicaciones en tiempo real, se usará el Filtrado de Contenidos After-Queue de Postfix [POS2007] con algunas modificaciones. Este tipo de filtrado se lleva a cabo luego del encolamiento de correos, brindando mayores beneficios que el filtrado convencional, como el número de filtros que puedan emplearse en paralelo.

Como vemos en la figura 4-2, Postfix recibirá correos de la red con ayuda del demonio `smtpd` y luego los encolará con el demonio `qmgr`. Luego, con ayuda de los agentes de entrega `local`, `smtp` y `pipe`, Postfix entregará los correos a destinatarios locales, a otros MTAs o a comandos externos, respectivamente. Es a este último demonio, al que se le asociará un filtro de contenido externo, el cual, al final del análisis, utilizará al demonio `sendmail` para insertar los

correos no eliminados nuevamente en las colas del Postfix y sean entregados finalmente a sus destinatarios.

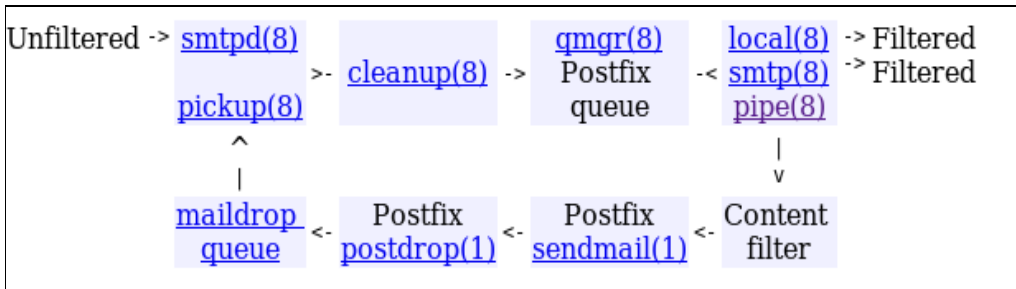


FIGURA 4-2: DISTRIBUCIÓN DE DEMONIOS POSTFIX DURANTE EL FILTRADO DE CONTENIDOS AFTER-QUEUE

Fuente: "Postfix" [POS2007]

El filtro de contenido, tal como se muestra en la tabla 4-12, estará conformado por el script Bash /etc/postfix/mail_java.sh.

TABLA 4-12: CONTENIDO DEL SCRIPT /ETC/POSTFIX/JAVA_MAIL.SH

```

#!/bin/sh
INSPECT_DIR=/var/spool/filter
SENDMAIL="/usr/sbin/sendmail -i"
SPAMASSASSIN="/usr/bin/spamassassin -x"
DESTINO="localhost 55055 " #IP del servidor y el puerto (55055)
TIEMPO="-q 0" #Tiempo para cerrarla conexion.

# Codigos de Salida
EX_TEMPFAIL=75

# Limpiar variables al finalizar o en caso de error.
trap "rm -f in.$$" 0 1 2 3 15

# Inicio.
cd $INSPECT_DIR || { echo $INSPECT_DIR no existe; salida $EX_TEMPFAIL; }
cat | $SPAMASSASSIN> in.$$ && cat in.$$ | nc $TIEMPO $DESTINO || {
    echo No se puede grabar el correo en el archivo; exit $EX_TEMPFAIL; }

$SENDMAIL "$@" < in.$$
exit $?
    
```


El script enviará el correo entrante al SpamAssassin (el cual conserva su configuración por defecto), donde se insertará la cabecera “X-Spam-Status” indicando si se trata de SPAM (“Yes”) o no (“No”); el correo resultante es almacenado luego en el archivo temporal in.\$\$ para su posterior análisis. Con ayuda de la herramienta NetCat [NET2007], se enviará el contenido del correo a través de una conexión TCP/IP con el servidor de aplicaciones utilizando el puerto 55055, logrando de esta forma que la aplicación Java obtenga los correos conforme llegan al buzón de consultas. Finalmente, el correo es reinsertado al Postfix para su almacenamiento en el buzón.

Se debe realizar algunas configuraciones antes de poder utilizar el script anteriormente presentado. Como primer paso se creará al usuario “filtrador”, el cual será dueño del script y de la carpeta temporal /var/spool/filter/. Paso seguido, se modificará el archivo /etc/postfix/master.cf, como se muestra en la tabla 4-13, para activar el filtrado de contenidos en los correos que llegan vía SMTP y asociar al agente pipe con el script. Se establece como límite que el Postfix ejecute este script un máximo de 10 veces en el mismo instante, valor comúnmente recomendado para equipos desktop.

TABLA 4-13: CONTENIDO DEL SCRIPT /ETC/POSTFIX/MASTER.CF

```
#=====
# service type private unpriv chroot wakeup maxproc command + args
#           (yes) (yes) (yes) (never) (100)
#
#=====
...
smtp net n - - - - smtpd -o content_filter=filter:dummy
filter unix - n n - 10 pipe flags=Rq user=filtrador
argv=/etc/postfix/mail_java.sh -f ${sender} -- ${recipient}
...
```

4.3 Servidor de Aplicaciones

El paso final de la implementación está conformado por el desarrollo de la aplicación Java que procesará el correo recibido y del JESS que generará el mensaje de respuesta.

4.3.1 Aplicación Java

Primero se instalará el Java Development Kit 6 con el comando “sudo apt-get install sun-java6-jdk”. Para el desarrollo se necesitará de las librerías:

- commons-dbcp-1.2.1.jar
- commons-pool-1.3.jar
- mysql-connector-java-5.0.7-bin.jar
- mail.jar
- jess.jar

Las tres primeras librerías brindan los métodos necesarios para implementar el pool de conexiones y la conexión con la Base de Datos MySQL, la cuarta permite el envío de correos electrónicos de respuesta y la quinta permite utilizar el JESS dentro de nuestra aplicación.

La aplicación se implementará de manera tal que siga el flujo mostrado en la figura 4-3. Se contará con una clase que escuche las conexiones entrantes en el puerto 55055 (sockets) para obtener los correos enviados por el Postfix. Cada vez que se detecte una nueva conexión, se almacenará el correo entrante y se iniciará un hilo que se encargue de su procesamiento.

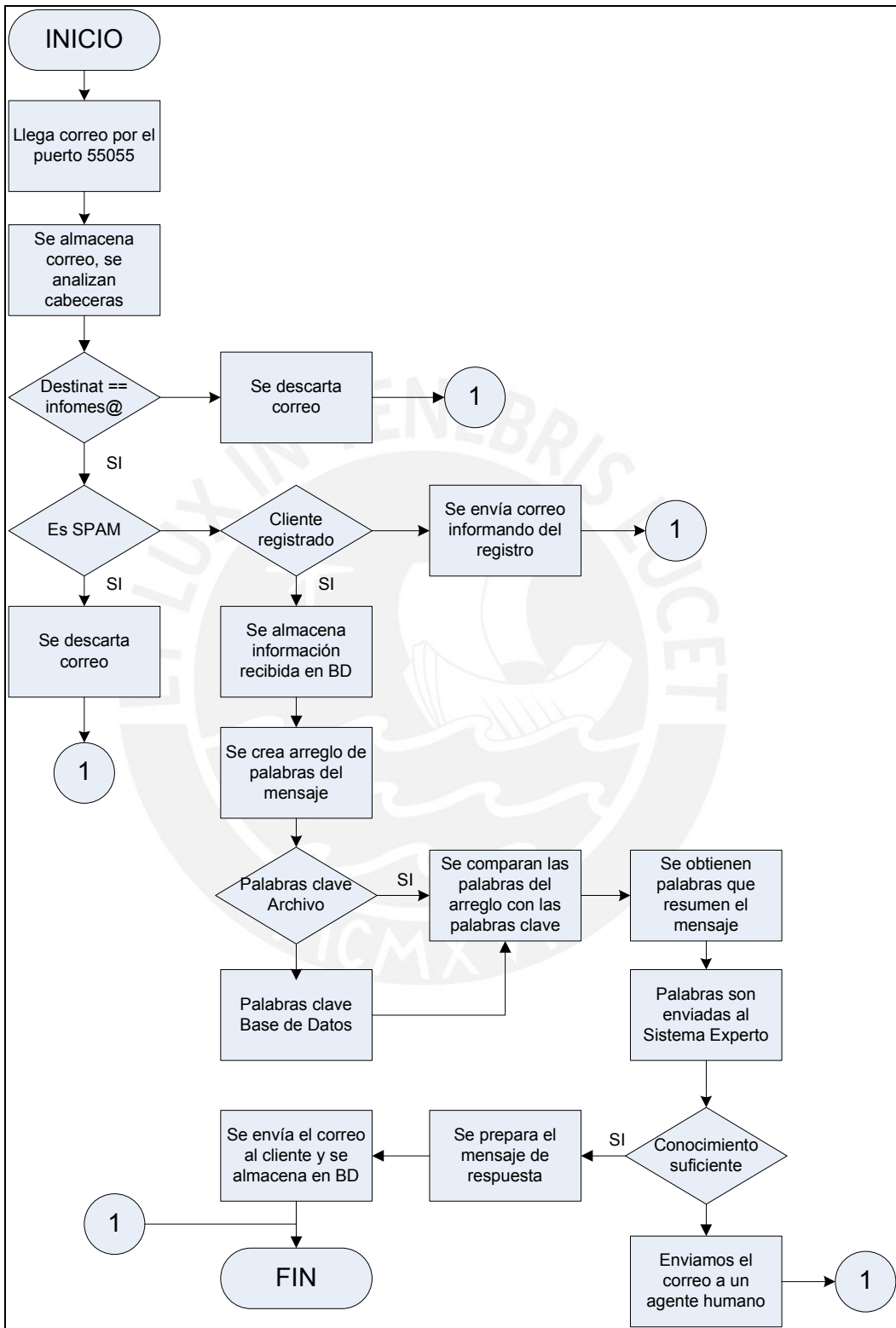


FIGURA 4-3: DIAGRAMA DE FLUJO DE LA APLICACIÓN JAVA

Debido a que el correo entrante se almacena como una cadena de caracteres, la primera labor del hilo será enviarlo a una etapa de análisis de cabeceras para obtener el remitente, destinatario, asunto, mensaje y el indicador de SPAM. Una vez se determine que el correo está dirigido a la cuenta del sistema de consultas “informes@alcalines.com”, se comprobará que no sea SPAM; caso contrario, será descartado. Con la información del remitente, se validará en base de datos si el usuario puede utilizar el servicio; caso contrario se le envía un correo que sugiere registrarse.

Validado el usuario, se procede a almacenar el correo en base de datos, marcándolo como pendiente, para luego iniciar el análisis del contenido del mensaje. Se desmenuzará el mensaje, almacenando las palabras en un ArrayList, luego se dará inicio a la búsqueda de palabras claves en él mismo. Para lograrlo, se emplearán archivos de texto que contengan palabras claves; se comparará cada una de las palabras del arreglo con el contenido de los archivos, obteniendo así un Bean que resuma la solicitud del cliente. A manera de brindar redundancia, se podrá utilizar la base de datos para la búsqueda de palabras claves, en caso la lectura de los archivos falle.

La información del Bean se enviará al JESS para que, junto con las reglas de la Base de Conocimiento y el motor de inferencia, determine la respuesta adecuada a la consulta. El resultado será entregado de vuelta al programa para componer el correo de respuesta, enviárselo al cliente y almacenarlo en la base de datos (marcándolo como atendido) junto con un resumen de la consulta. En caso la información recolectada no sea suficiente, el sistema enviará el correo a la cuenta de un agente para que la consulta sea atendida; luego será el

agente el que entrene al sistema para que en el futuro pueda resolver este tipo de consultas.

4.3.2 Sistema Experto

Antes de la implementación del Sistema Experto, se debe tener en cuenta los elementos del JESS con los que se interactuará:

- **Base de Conocimientos**, contiene un conjunto de reglas definidas del sistema.
- **Memoria del trabajo**, contiene datos de entrada o hechos que el sistema considerará como premisas.
- **Motor de Inferencia**, corazón del sistema experto que seleccionará las reglas que cumplen con lo almacenado en la memoria de trabajo (pattern matcher), establecerá el orden de ejecución (agenda) y ejecutará una por una (execution engine).

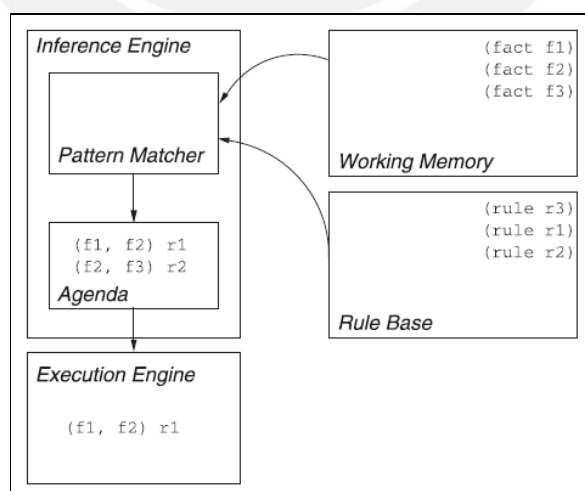


FIGURA 4-4: ELEMENTOS DEL JESS

Fuente: "JESS in Action" [FRI2003]

Toda BC se construye a partir de una Base de Hechos (BH), un conjunto de datos en bruto del dominio del problema sobre el cual trabajará la ingeniería del conocimiento [KNO2007]; para la solución, la BH está conformada por la base de datos, ya que almacena la información de los destinos, horarios y tarifas de los vuelos de la empresa.

Debido a la gran cantidad de información almacenada y posibilidad de combinaciones, la BC contará con un enorme número de reglas; además, cada vez que se agregue información de nuevos vuelos o se omitan vuelos antiguos, deberá de actualizarse. Teniendo en cuenta estos problemas de escalabilidad, se implementará un método Java que acceda a la información en base de datos y cree todas las reglas posibles en código JESS, tal como se muestra en la figura 4-5. Todas estas reglas serán almacenadas en el archivo “reglas.clp”, el cual, con ayuda del API nativo del JESS, será cargado en el motor de inferencia.

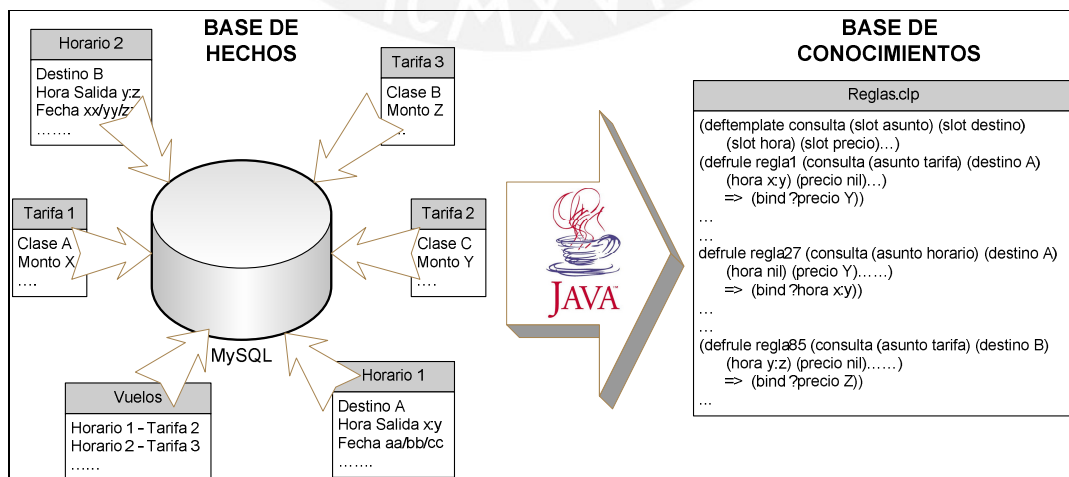
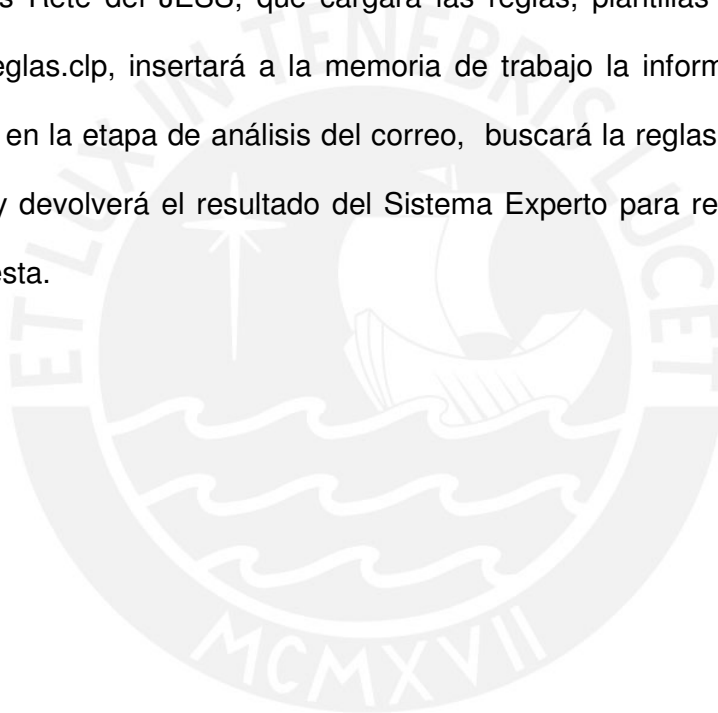


FIGURA 4-5: IMPLEMENTACIÓN DE LA BASE DE CONOCIMIENTOS

Este método también deberá de agregar al archivo una estructura de datos o plantilla que permita que el motor de inferencia interactúe con los hechos de la memoria de trabajo, así como una función que permita enviar los resultados del JESS al programa en Java empleando Hash Tables [HAS2007].

Una vez definido el archivo reglas.clp se implementará la clase AnalizaExperto.java para utilizar al JESS. Primero se instanciará al motor de inferencias Rete del JESS, que cargará las reglas, plantillas y funciones del archivo reglas.clp, insertará a la memoria de trabajo la información del Bean generado en la etapa de análisis del correo, buscará la reglas que resuelva la consulta y devolverá el resultado del Sistema Experto para redactar el correo de respuesta.





Capítulo 5

Plan de Pruebas

5.1 Introducción

Con el fin de evaluar las ventajas del Sistema Experto y definir la confiabilidad y rapidez de la arquitectura implementada, se procederá a la etapa de pruebas.

La primera parte de este plan de pruebas consistirá en obtener el grado de confiabilidad del sistema, para ello se procederá a generar un número considerable de correos de contenido variable y estudiar el comportamiento del sistema.

La segunda parte consistirá en determinar el tiempo promedio que le toma al sistema analizar y responder los correos. Estos resultados serán recolectados durante la ejecución de la primera prueba.

La tercera y última parte presentará un análisis de la carga del sistema durante las pruebas de confiabilidad.

5.2 Desarrollo de pruebas del sistema

Teniendo en cuenta que las pruebas buscan analizar el comportamiento del sistema, se procederá a automatizar el envío de los correos a la aplicación Java con el comando “cat correo_x | /usr/sbin/spamassassin/ -x | nc -q 0 localhost 55055”, utilizado en el script *java_mail.sh* tal como se observa en la tabla 4-12. El comando realizará una lectura del archivo *correo_x* (archivo en texto plano que contiene el formato de un correo convencional) y lo enviará al SpamAssassin para el análisis anti-SPAM; al finalizar, el correo resultante será enviado a la aplicación Java para su procesamiento.

Los correos utilizados en las pruebas deberán describir diferentes tipos de operaciones, como solicitudes de reservas de pasajes, compra de pasajes, consultas sobre horarios y tarifas de vuelo, de restricciones y cambios, informes de quejas y sugerencias, solicitudes de reembolso, de forma que se pueda analizar las acciones que toma el sistema en escenarios para los que fue o no diseñado. Entre las posibles acciones, se buscará conocer si el sistema responde a las consultas de manera precisa (respuesta correcta), con inconsistencias y resultados parciales (respuesta errada) o si delega el trabajo a algún agente por falta de información.

Para el cálculo del tiempo que le toma a la aplicación responder cada correo, se utilizará el método Java “System.currentTimeMillis()” en el instante en que

se recibe el correo en el puerto 55055 y en el que se envía la respuesta al servidor, determinando así el tiempo del sistema en milisegundos en el que se realizó cada evento. Luego se procederá a calcular la diferencia entre estos 2 tiempos.

Para el desarrollo de estas pruebas, se deberá tener presente el tráfico entrante de correos en el Contact Center de AlcaLines, mostrada en la figura 2-2; el número total de correos que se analizarán estará definido por la cantidad de correos que recibe la plataforma en la hora cargada o pico. Durante la hora pico (de 10:00 a 11:00 hrs.) se tienen 112 correos entrantes, donde 28 son SPAM y 84 son correos de consultas; por lo tanto, para las pruebas se enviarán a la aplicación 5 ráfagas de correos, tal como se muestra continuación:

- 1ra ráfaga: 4 SPAMs, 12 correos válidos.
- 2da ráfaga: 5 SPAMs, 7 correos válidos.
- 3ra ráfaga: 5 SPAMs, 16 correos válidos.
- 4ta ráfaga: 7 SPAMs, 30 correos válidos.
- 5ta ráfaga: 7 SPAMs, 18 correos válidos.

Para que las pruebas de confiabilidad del sistema sean más reales, se solicitó la colaboración de un grupo de personas en la generación de los 84 correos de consultas, de los cuales el 73.81% (62 correos) realizaban consultas que el sistema se encontraba en capacidad de responder y el 28.19% (22 correos) hacían referencia a temas que se encuentran fuera del dominio de conocimiento del sistema.

Otro factor que también se debe considerar es que el sistema experto cuenta con un total 284 reglas, obtenidas automáticamente en base a la información de 2 días de vuelos.

5.3 Resultados

Los 112 correos fueron analizados por nuestro sistema y se obtuvieron los siguientes resultados:

TABLA 5-1: RESULTADOS DE LAS PRUEBAS DE CONFIABILIDAD

Tipo Correo	Respuesta		Desvío	
	Correcta	Errada	Agente	Descarte
SPAM	-	-	-	28
Tarifa - Horario	57	3	2	-
Otros	0	0	22	-

En la tabla 5-1 se puede apreciar que de los 62 correos de consultas de tarifas y horarios de vuelos, 57 fueron contestados correctamente, 3 fueron contestados de manera parcial o con información equivocada y 2 derivados a un agente por falta de información. Con respecto a los correos de consultas no soportadas por el sistema, fueron desviados en su totalidad a un agente para que pueda resolverlas, ninguna fue eliminada. En el caso de los 28 correos SPAM, el sistema se encargo de descartarlos antes de procesarlos.

De lo dicho anteriormente, vemos que el sistema contesto correctamente 57 de 62 correos de consultas de horarios y tafias de vuelos; ofreciendo una confiabilidad del 91.94%.

Para el cálculo del tiempo de procesamiento del sistema, se tomó en cuenta los tiempos de los 84 correos de consultas; los 28 correos SPAM no fueron considerados ya que no son procesados por el sistema.

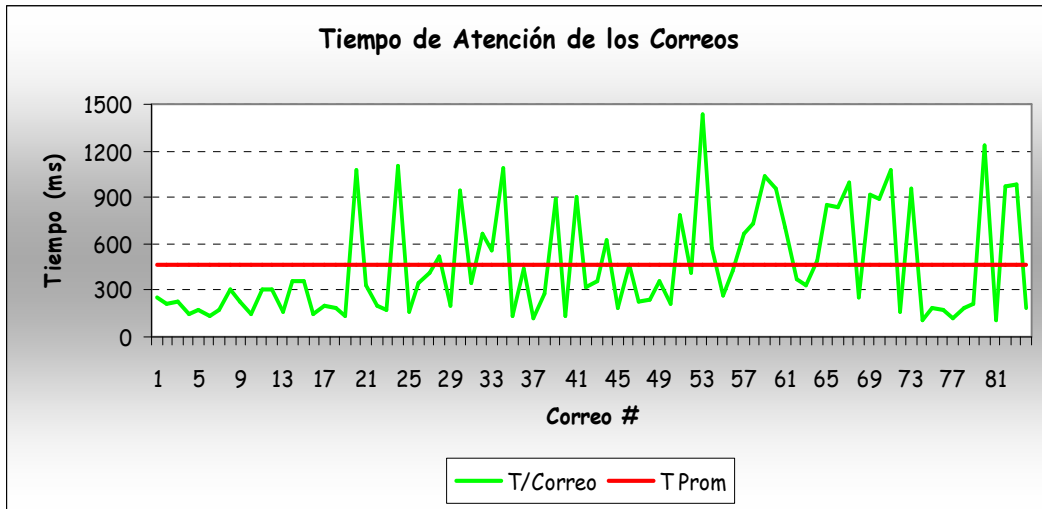


FIGURA 5-1: GRÁFICA DE TIEMPO DE ATENCIÓN DE LOS CORREOS

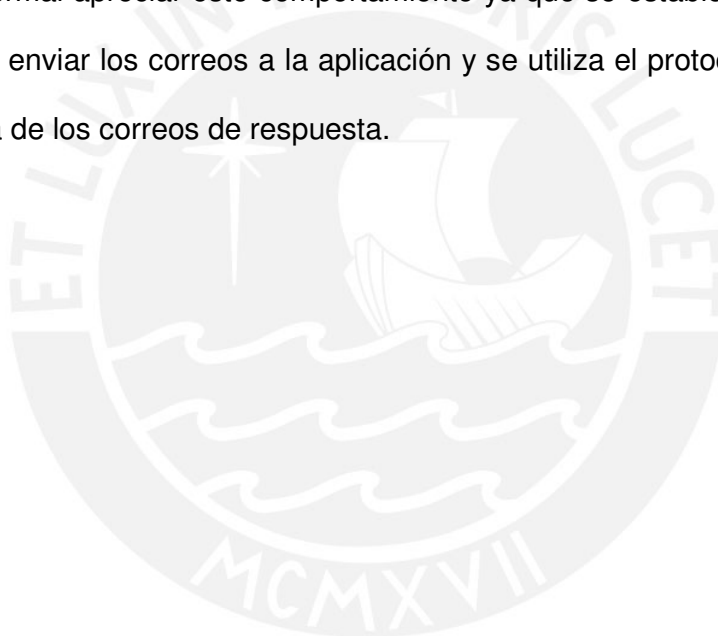
Se puede observar en la figura 5-1, los diferentes tiempos de procesamiento del sistema de cada uno de los 84 correos recibidos por el sistema. En promedio, el tiempo que le toma al sistema responder una consulta es de 461 milisegundos, tal como lo indica la línea de color rojo.

Para poder apreciar la carga del servidor durante la ejecución de las pruebas, se utilizó una herramienta propia de la distribución llamada “Monitor del sistema”. Esta herramienta muestra tanto la carga del CPU como el estado de la memoria del equipo y el uso de recursos de red.



FIGURA 5-2: GRÁFICA DE USO DEL SERVIDOR

En la figura 5-2, se puede observar las estadísticas obtenidas durante las pruebas con los 112 correos. En la gráfica de carga de CPU se pueden distinguir 5 eventos importantes, que corresponden a las 5 ráfagas de correos generadas durante las pruebas; en el primer caso los correos llegaron a la aplicación de manera secuencial, no de forma conjunta como ocurre en los restantes, donde la carga del procesador se encuentra al 100% durante el análisis de casi toda la ráfaga. En el caso de la gráfica de uso de recursos de red, es normal apreciar este comportamiento ya que se establecen conexiones TCP para enviar los correos a la aplicación y se utiliza el protocolo SMTP para la entrega de los correos de respuesta.



Conclusiones, Recomendaciones y Trabajos Futuros

6.1 Recomendaciones

Se sugieren, a continuación, algunas recomendaciones a tener en cuenta sobre el presente trabajo de tesis:

- Se recomienda que, de acuerdo a la proyección en el uso del servicio, se empleen servidores independientes para los servicios de base de datos, correo electrónico y aplicaciones. Esto permitirá distribuir la carga de cada uno de estos servicios, incrementar la velocidad de procesamiento de solicitudes y eliminar posibles cuellos de botella en la plataforma.

- Se recomienda contar con sistemas operativos más robustos, estables y orientados a la administración de servicios como Red Hat Enterprise Linux 5.1, CentOS 5.1 o Debian.
- Se recomienda implantar un sistema de redundancia para la solución propuesta y ofrecer así un servicio de alta confiabilidad y disponibilidad. Se debe considerar también la posibilidad de extender este sistema en toda la plataforma y contar así con un Contact Center IP robusto.
- Se recomienda aprovechar toda la información de aquellos correos que el sistema no pueda procesar para contribuir al entrenamiento del mismo.

6.2 Trabajos Futuros

Se proponen a continuación diversos trabajos derivados del presente proyecto:

- Implementar el sistema del proyecto actual en los módulos restantes del Contact Center, como el Chat y la telefonía. Esto permitirá que la plataforma dependa cada vez menos de agentes físicos y pueda soportar mayor cantidad de comunicaciones.
- Extender las funcionalidades del sistema, de forma que pueda atender otras operaciones de la aerolínea, como por ejemplo, la reserva y compra de pasajes. En estos casos, se debería de considerar el uso de *triggers* y *Jobs* para mantener actualizadas las reglas del sistema.
- Implementar mecanismos que faciliten a los agentes la labor de entrenar al sistema. Esto ayudará a que el sistema aprenda de forma más rápida y eficiente.

- Incluir en el sistema nuevos e innovadores servicios que permita una integración más fuerte con los demás canales del Contact Center.

6.3 Conclusiones

Finalizado el presente trabajo de tesis, se puede llegar a las siguientes conclusiones:

- De los resultados obtenidos durante el desarrollo de las pruebas, podemos concluir que el sistema es fiable. El obtener un grado de confiabilidad del 91.94% nos indica que nuestra solución es viable, que el sistema es válido.
- El sistema implementado ofrece una velocidad de respuesta muy alta, alrededor de 461 milisegundos según lo obtenido en la etapa de pruebas. Este valor corrobora la teoría base del presente proyecto, el Sistema Experto permitirá que la atención al cliente sea mucho más rápida y eficiente.
- El sistema permite, no sólo dar solución a la problemática generada por la alta demanda, sino también, aumentar la satisfacción de los clientes, el prestigio de la empresa y su posición en el mercado.
- Conforme el sistema vaya aprendiendo, se volverá cada vez más independiente, disminuyendo así la carga de trabajo que le asigna al agente.

Bibliografía

- [CAS1998] CASTILLO, ENRIQUE. “Sistemas Expertos y Modelos de Redes Probabilísticas”. España. 1998.
- [CLE2007] CLEMENTE MEDINA, PAMELA. Tesis de Grado: “Análisis y Diseño de un Proyecto de Migración de un Call Center a un Contact Center utilizando tecnología IP”. PUCP, Perú. 2007.
- [COR2007] CORROCHANO, BERTHA; HEREDERO SANTIAGO. Artículo: “El Aspect Contact Center Satisfaction Index 2007 detecta mejoras año a año en las interacciones cliente-empresa y observa una relación directa entre malas experiencias e impacto negativo en los resultados”. Madrid, España. Mayo 2007.
- URL: http://www.acceso.com/display_release.html?id=36929
- [COO2005] COOPER, MENDEL. “Advance Bash-Scripting Guide: an in-depth exploration of the art of shell scripting “, 2005.
- URL: http://www.linuxtopia.org/online_books/advanced_bash_scripting_guide/
- [FLU2005] FLUSS, DONNA. “The Real-Time Contact Center: Strategies, Tactics and Technologies for Building a Profitable Service and Sales Operation”. Ed. AMACOM. EE.UU. 2005.
- [FLO2001] FLORES, JUAN. Artículo: “Escenarios reales de IP Contact Center.” Comunicaciones World. Noviembre 2001.
- URL: <http://www.soluziona.com>

- [FLO2006] FLORES, ANDRÉS. Presentación: “Aplicaciones de Tratamiento de Señales: Síntesis de Voz y Sistemas de Diálogo Automático”. Perú. 2006.
- [FOR1982] FORGY, CHARLES. "Rete: A Fast Algorithm for the Many Pattern/ Many Object Pattern Match Problem". Artificial Intelligence vol.19. 1982.
- [FRI2003] FRIEDMAN-HILL ERNEST. “JESS in Action: Rule-Based System in Java”. Ed Manning. EE.UU. 2003
- [GIA1998] GIATARRANO, JOSEPH; RYLEY, GARY. “Expert Systems Principles and Programming” .1998.
- [GRA2001] GRANADOS MOLINA, ENRIQUE. Artículo: “CMR y SERVICIOS DE INFORMACIÓN: La biblioteca como un negocio de la nueva economía”. Lima. 2001. URL: <http://www.bibliosperu.com>
- [KRI1996] KRISHNAMOORTHY, C.; RAJEEV, S. “Artificial Intelligence and Expert Systems for Engineers”. CRC Press LLC. 1996.
- [LAR2007] LARIOS; ENRIQUE. Reporte: “Una Revisión de los Modelos de Tecnologías de Información de Contact Centers”. Convención Interamericana de Ingeniería en Computación, Sistemas e Informática, Mérida, Venezuela. Noviembre 2007.
- [SWI2001] SWIFT, RONALD. “Accelerating Customer Relationships: Using CRM and Relationship Technologies”. Ed. Prentice-Hall, New York, EEUU. 2001.
- [WAS1989] WASHINGTON, EDDIE; ALI, MOONIS. “An Object-Oriented Expert System for Coal-Fired MHD Power Plant Fault Monitoring and

Diagnosis”. ACM Press. New York, EEUU. 1989

URL: <http://portal.acm.org/citation.cfm?id=66636>

[WIL2007] WILENSKY, ROBERT. Presentación: “Expert Systems”

URL: <http://www.cs.berkeley.edu>

[ALT2007] Altitude. Presentación: “Como implantar un Contact Center de Excelencia” Barcelona, España. Febrero 2007.

URL: <http://www.altitude.es>

[ANO2007] Anomy Sanitizer. URL: <http://mailtools.anomy.net/>

[APA2007] Apache. URL: <http://www.apache.org>

[AST2007] Asterisk. URL: <http://www.asterisk.org>

[AVA2007] Avaya. URL: <http://www.avaya.com/>

[CLI2007] CLIPS. URL: <http://www.ghq.net/clips/CLIPS.html>

[COS2007] CosmoCom. URL: <http://www.cosmocom.com/>

[COM2007] Comparex. Presentación: “Soluciones de Contact Center”

URL: <http://www.comparex.es>

[CYS2007] Cystelcom Sistemas. URL: <http://www.cystelcom.com>

[DAT2007] Database System Corp.

URL: <http://www.databasesystemscorp.com/>

[EXP2007] Expert System. URL: <http://www.expertsystem.net/>

[FUJ2007] Fujitsu. Presentación: “Contact Center”.

URL: <http://www.fujitsu.com>

[GAR2007] Gartner Inc. URL: <http://www.gartner.com/>

[GOC2007] Go4Costumers. URL: <http://www.go4customer.com/>

[HAS2007] Hash Tables.

URL: http://www.cs.auckland.ac.nz/software/AlgAnim/hash_tables.html

[ICM2007] ICMI. URL: <http://www.incoming.com/Default.aspx>

[JAB2007] Jabber. URL: <http://www.jabber.org>

[JES2007] Java Expert System Shell. URL: <http://herzberg.ca.sandia.gov/jess/>

[JAV2007] Java Sun. URL: <http://java.sun.com/>

[KNO2007] Knowledge Engineering.

URL: <http://www.epistemics.co.uk/Notes/61-0-0.htm>

[LAN2007] LAN Airlines. URL: <http://www.lan.com/index-es-cl.html>

[LAM2007] LAMP O'Reilly.

URL: <http://www.onlamp.com/pub/a/onlamp/2001/01/25/lamp.html>

[MAN2008] Mandarax. URL: <http://mandarax.sourceforge.net/>

[MIC2004] Microsoft. Artículo: "El CRM como estrategia de negocio".

URL: http://www.microsoft.com/spain/empresas/soluciones/guia_cr_m.msp

[MYS2008] MySQL. URL: <http://www.mysql.com>

[NET2007] NETCAT. URL: <http://netcat.sourceforge.net/>

[PER2008] Perl. URL: <http://www.perl.com>

[POS2007] Postfix. URL: <http://www.postfix.org>

[PNG2007] Postfix: La nueva generación.

URL: <http://www.uco.es/ccs/sistemas/postfix/intro.html>

[SAN2007] Sandia National Laboratories. URL: <http://www.sandia.gov/>

[SEN2007] Sendmail. URL: <http://www.sendmail.org>

- [SPA2008] SpamAssassin. URL: <http://spamassassin.apache.org/>
- [STR2007] Strategic Contact. URL: <http://www.strategiccontact.com/index.asp>
- [SUR2007] Surf. Presentación: "IV2R". URL: <http://www.surf-com.com>
- [WIE2007] Wietse's Home Page. URL: <http://www.porcupine.org/wietse/>



Anexos

Anexo1: Script de la Base de Datos.

Se presenta en este anexo el script para la implementación de la estructura de base de datos.

Anexo2: Código de la implementación.

Se presenta en este anexo el código Java de la implementación del sistema.

