



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 INTERACTIVO DE RESPUESTA DE
VOZ (IVR) PILOTO PARA LA RESERVA DE
BOLETOS DEL FERROCARRIL CUZCO –
MACHU PICHU**

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

David Alfonso Ortega Gallegos

LIMA – PERÚ

2007

RESUMEN

El proyecto de tesis consiste en el estudio, diseño e implementación de un sistema IVR IP de interfaz telefónica bilingüe (español- inglés) para la reserva de boletos del ferrocarril de Cuzco para el viaje desde la estación de San Pedro hasta la ciudadela Machu – Picchu (Aguas Calientes).

Para esto se realizará un previo estudio teórico de las tecnologías IVR para luego proceder al desarrollo del proyecto en fases de diseño e implementación.

Este sistema consistirá en una arquitectura conformada por tres servidores: El primero será una PBX IP implementada en software libre, el segundo un servidor de requerimientos que tramitará pedidos y almacenará la lógica del sistema, y el tercero un servidor de Base de datos que sigue el modelamiento desarrollado en este trabajo.

Una vez hecha la implementación, ésta será sometida a pruebas de esfuerzo para determinar la cantidad de llamadas y el uso de CPU en el servidor PBX para finalmente dar las conclusiones y recomendaciones del caso.

DEDICATORIA

Esta tesis está dedicada a mis padres David y Gloria
por su constante apoyo, confianza y
sobre todo: por su amor.



AGRADECIMIENTOS

Quisiera empezar agradeciendo a Dios por su continuo cuidado y guía; así como por la salud que me dado para poder llegar hasta este punto de mi vida.

Agradezco infinitamente a mis padres por su esfuerzo en brindarme la educación que tengo, por su cuidado y su apoyo, así como su motivación y guía. Agradezco también a mis abuelos por su gran cariño.

Al ingeniero Enrique Larios, mi asesor, por su constante ayuda y preocupación en el desarrollo de éste trabajo y sobretodo por la confianza depositada en mí y por su amistad.

Al ingeniero Arturo Díaz Rosemberg, por su apoyo en la revisión y el plan de pruebas del sistema y su buena disposición para ayudarme a generar los resultados finales del proyecto.

Al doctor Carlos Silva y al ingeniero David Chávez, por su guía y constante preocupación en el cumplimiento de los objetivos primordiales. Quiero agradecer a ambos también su apoyo en momentos de dificultad, no sólo en lo concerniente a lo académico, sino en su apoyo emocional para con los míos.

Quiero agradecer a Mayra Alejandra Barrios por su ayuda con la corrección lingüística del presente trabajo y su ayuda con las locuciones en el desarrollo del proyecto. ¡Gracias por ser la voz de mi sistema, Mayra! Muchas gracias por tu incondicional apoyo, comprensión y cariño.

A mis profesores, a todos y cada uno, por haberme dado los conocimientos necesarios para llegar hasta aquí, y finalmente a mis compañeros de promoción y amigos de toda la vida: sin Uds. esta gran travesía no hubiera sido lo que fue. Gracias por los geniales recuerdos.

Finalmente, a todos de nuevo... ¡Muchas gracias!

INDICE

LISTA DE FIGURAS	8
LISTA DE TABLAS	9
INTRODUCCIÓN	10
CAPÍTULO 1 ESTUDIO TEÓRICO DE LA TECNOLOGÍA IVR	12
1.1 Voz sobre IP	12
1.1.1 Protocolos de Señalización	13
1.1.1.1 SIP	13
1.1.1.2 IAX	14
1.1.1.3 Comparaciones entre ambos protocolos	14
1.1.2 Codecs	15
1.1.2.1 G.711	16
1.1.2.2 G.729	16
1.1.2.3 GSM	17
1.2 CTI: Computer Telephone Integration	17
1.2.1 Definición	17
1.2.2 Aplicaciones	17
1.3 IVR - Interactive Voice Response	18
1.3.1 Definición	18
1.3.2 Justificación de los Sistemas IVR	19
1.3.2.1 Sistema	19
1.3.2.2 Usuario	21
1.3.3 Evolución de los sistemas IVR y Tipos	21
1.3.4 Partes y Tecnologías anexas de los Sistemas IVR	24
1.3.4.1 PBX	24
1.3.4.2 IVR (VRU)	25
1.3.4.3 TTS (Text to Speech)	25
1.3.4.4 ASR (Automatic Speech Recognition)	25
1.3.5 Sistemas IVR en el Mercado	25
1.3.5.1 Mundo Empresarial	25
1.3.5.2 Mercados Mundiales	27
1.3.6 Similitudes y Diferencias entre software libre y soluciones propietarias	29
CAPÍTULO 2 ANÁLISIS Y ARQUITECTURA DE LA SOLUCIÓN	31
2.1. Definición del problema a resolver	31
2.2. Alcances y Limitaciones del Proyecto	34

2.3.	Propuesta de la Arquitectura de la Solución.....	35
2.3.1.	Tecnologías de la Solución	36
2.3.1.1	PBX IP	36
2.3.1.2	Lenguaje de Programación	37
2.3.1.2.1	Asterisk - Java	37
2.3.1.3	Base de Datos del Sistema	40
2.3.1.4	Servidor de Aplicaciones	41
2.3.1.5	Servidor TTS.....	41
2.3.1.6	GUI.....	42
2.3.1.7	SOX	43
CAPÍTULO 3 DISEÑO E IMPLEMENTACIÓN DE LA SOLUCIÓN		44
3.1.	Diseño de la Solución.....	44
3.1.1.	Descripción del Servicio	44
3.1.1.1	Registro.....	44
3.1.1.2	Reserva	45
3.1.1.3	Pago.....	45
3.1.2.	Diagramas de Flujo de la Solución	45
3.1.2.1	Aplicación Telefónica.....	45
3.1.2.2	Aplicación Web	49
3.1.3.	Modelo de Entidad Relación	50
3.1.3.1	Diccionario de Datos	51
3.1.3.2	Relación de Procedimientos	55
3.1.4.	Estándares del Sistema.....	57
3.1.5.	Diseño de la Aplicación IVR	57
3.1.5.1	Aplicación Telefónica.....	57
3.1.5.1.1	Validación	58
3.1.5.1.2	Requerimiento de Datos.....	59
3.1.5.1.3	Informe.....	60
3.1.5.1.4	Registro.....	61
3.1.5.2	Aplicación Web.....	62
3.1.6.	Diccionario de Clases del Sistema.....	63
3.1.7.	Estimación de Tráfico	69
3.2.	Implementación de la Arquitectura.....	69
3.2.1.	Servidor PBX	69
3.2.1.1	Configuración Asterisk.....	70
3.2.1.2	Configuración Festival	71
3.2.2.	Servidor de Requerimientos (AGI)	72
3.2.3.	Servidor de Base de Datos	73
3.2.4.	Pantallas de la Aplicación Web Prototipo	74
3.2.4.1	Interfaz de Ingreso.....	74
3.2.4.2	Interfaz de Inscripción	75
3.2.4.3	Interfaz de Búsquedas	76
3.2.4.4	Interfaces de Datos	77
CAPÍTULO 4 PRUEBAS DE DESEMPEÑO		80
4.1.	Introducción	80
4.2.	Configuración.....	81
4.2.1	Servidor PBX	82
4.2.2	Servidor de Requerimientos.....	82

4.2.3	Sistema SIPP.....	83
4.3.	Resultados.....	83
4.3.1	Número de Llamadas Concurrentes	84
4.3.2	Uso de CPU.....	85
4.3.3	Memoria.....	87
CONCLUSIONES, RECOMENDACIONES Y TRABAJOS FUTUROS		88
5.1.	Recomendaciones.....	88
5.2.	Trabajos Futuros.....	89
5.3.	Conclusiones	90
BIBLIOGRAFÍA		91
ANEXOS		94



LISTA DE FIGURAS

FIGURA 1.1. COSTO COMPARATIVO UNA LLAMADA PROCESADA CON UN AGENTE HUMANO CONTRA UN IVR DE RECONOCIMIENTO DE VOZ	19
FIGURA 1.2. COSTO DE RECUPERACIÓN EN EL TIEMPO DE LA INVERSIÓN INICIAL EN UN IVR DE RECONOCIMIENTO VOCAL (NUANCE)	20
FIGURA 1.3. COSTOS PROMEDIO POR LLAMADA (SPEECH WORKS).....	20
FIGURA 1.4. EVOLUCIÓN DE TECNOLOGÍAS USADAS EN SISTEMAS IVR MOSTRANDO COSTO VERSUS SOFISTICACIÓN	22
FIGURA 1.5. EVOLUCIÓN DE MERCADO CHINO EN SISTEMAS IVR MOSTRANDO TASAS DE INCREMENTO	28
FIGURA 1.6. EMPRESAS QUE COMPARTEN EL MERCADO CHINO EN SISTEMAS IVR.....	28
FIGURA 2.1. RECORRIDO GEOGRÁFICO DEL TREN CUZCO – MACHU PICCHU.....	32
FIGURA 2.2. TOTAL DE VISITAS A MACHU PICCHU DESDE EL AÑO 2003 AL 2005..	33
FIGURA 2.2. VISTANTES EXTRANJEROS VÍA TREN.....	33
FIGURA 2.3. ARQUITECTURA DEL DESARROLLO	36
FIGURA 2.4. DISEÑO AGI DEL PAQUETE ASTERISK-JAVA	39
FIGURA 3.2. DIAGRAMA DE LA SOLUCION – APLICACIÓN WEB.....	49
FIGURA 3.3. DIAGRAMA DE MODELO ENTIDAD RELACIÓN	50
FIGURA 3.3. ETAPA DE VALIDACIÓN	58
FIGURA 3.3. ETAPA DE REQUERIMIENTO DE DATOS.....	60
FIGURA 3.4. LOG DEL SERVIDOR DE REQUERIMIENTOS	73
FIGURA 3.5. INTERFAZ DE INGRESO	74
FIGURA 3.6. INTERFAZ DE INSCRIPCIÓN	75
FIGURA 3.6. INTERFAZ DE INSCRIPCIÓN	76
FIGURA 3.7. INTERFAZ DE RESULTADO DE BÚSQUEDA	77
FIGURA 3.8. INTERFAZ DE DATOS DEL USUARIO	78
FIGURA 3.9. INTERFAZ DE DATOS DE LA RESERVA.....	79
FIGURA 4.1. ARQUITECTURA DE PRUEBAS.....	81
FIGURA 4.2. LLAMADAS CONCURRENTES – PRUEBA 1.....	84
FIGURA 4.3. LOG DEL SIPP	84
FIGURA 4.4. LLAMADAS CONCURRENTES – PERIODO TOTAL	85
FIGURA 4.5. USO DE CPU CON SERVIDOR TTS.....	86
FIGURA 4.6. USO DE CPU SIN SERVIDOR TTS.....	86
FIGURA 4.7. USO DE MEMORIA.....	87

LISTA DE TABLAS

TABLA 1.1. DISTINTOS CODECS PARA VOIP	16
TABLA 1.2. EJEMPLOS DE INTERACCIÓN DE DIFERENTES TIPOS DE TECNOLOGÍA IVR.....	23
TABLA 1.3. ENCUESTA AL SECTOR CLIENTES SOBRE PREFERENCIAS EN SISTEMAS IVR Y REPUTACION DE LOS MISMOS	26
TABLA 2.1. PRECIOS DE IDA/VUELTA DE LOS VAGONES PERURAIL	34
TABLA 2.2. ALGUNOS MÉTODOS DE LA CLASE BASEAGISCIPT	40
TABLA 3.1. LOCUCIONES GRABADAS PARA LA APLICACIÓN TELEFÓNICA IVR ...	46
TABLA 3.1. LOCUCIONES GRABADAS PARA LA APLICACIÓN TELEFÓNICA IVR ...	47
TABLA 3.1. LOCUCIONES GRABADAS PARA LA APLICACIÓN TELEFÓNICA IVR ...	48
TABLA 3.2. TABLA USUARIO	51
TABLA 3.3. TABLA VAGÓN	52
TABLA 3.4. TABLA VIAJE.....	53
TABLA 3.5. TABLA OPERADOR.....	53
TABLA 3.6. TABLA RESERVA	54
TABLA 3.7. TABLA DE PROCEDIMIENTOS DEL PAQUETE PKG_RESERVA	55
TABLA 3.8. TABLA DE PROCEDIMIENTOS DEL PAQUETE PKG_TRABAJO	56
TABLA 3.9 VARIABLES ESPERADAS EN PRIMERA ETAPA DE REQUERIMIENTO DE DATOS	59
TABLA 3.10. CLASE BUSUARIO	63
TABLA 3.11. CLASE BRESERVA	64
TABLA 3.12. CLASE DUSUARIO	66
TABLA 3.16. ARCHIVO FESTIVAL.SCM – LENGUAJE ESPAÑOL.....	71
TABLA 3.17. ARCHIVO FESTIVAL.SCM – INTERACCION ASTERISK.....	71
TABLA 3.18. ARCHIVO /ETC/ASTERISK/FESTIVAL.CONF	72
TABLA 3.19. ARCHIVO FASTAGI-MAPPING.PROPERTIES	72
TABLA 4.1. CONFIGURACIÓN DE SIP.CONF	82
TABLA 4.2. ARCHIVO FASTAGI.PROPERTIES	83
TABLA 4.3. COMANDO SIPP EJECUTADO.....	83

Introducción

En los últimos años, hemos observado un evolucionar en los sistemas telefónicos, no sólo en el avance tecnológico de estos y de la red que los une; sino, de los servicios y aplicaciones telefónicas que empiezan a aparecer con el objetivo de mejorar la interacción con el usuario y la eficiencia en comunicar mensajes así como ofrecer automatización a ciertos procesos.

Los IVR (Interactive Voice Response) son aplicaciones de voz interactivas que aceptan como entrada tanto tonos marcados por el usuario, como la voz del mismo ofreciendo distintos tipos de respuesta según la programación del propio sistema; pudiendo de esta forma ofrecer servicios de información, encuestas y transacciones telefónicas.

Las tecnologías IVR's han tenido un gran apogeo, en un comienzo en bancos y grandes empresas, llegando hasta las tiendas que ofrecen interfaces de compra vía telefónica, ahorrando enormemente en capital humano para hacer este requerimiento.

Estos sistemas han evolucionado desde interfaces vía DTMF hasta reconocimiento de voz (Speech recognition IVR, Guided Speech IVR) en los cuales el usuario puede usar su voz para no sólo hacer selecciones en el sistema, sino también para hacer preguntas de manera natural.

En este proyecto, se realizará un breve estudio teórico de los conceptos ligados a la telefonía IP, así como los concernientes a los IVR's y el análisis del impacto de estos en la industria de las telecomunicaciones y la economía.

En base a los estudios teóricos, se procederá a hacer el diseño e implementación de un sistema IVR IP de interfaz telefónica bilingüe (español- inglés) para la reserva de boletos del ferrocarril de Cuzco para el viaje desde la estación de San Pedro hasta la ciudadela Machu – Picchu (Aguas Calientes). Con esto se buscará automatizar el proceso de reserva y compra de los boletos usando herramientas tecnológicas que permitan una mejor interacción con los turistas.

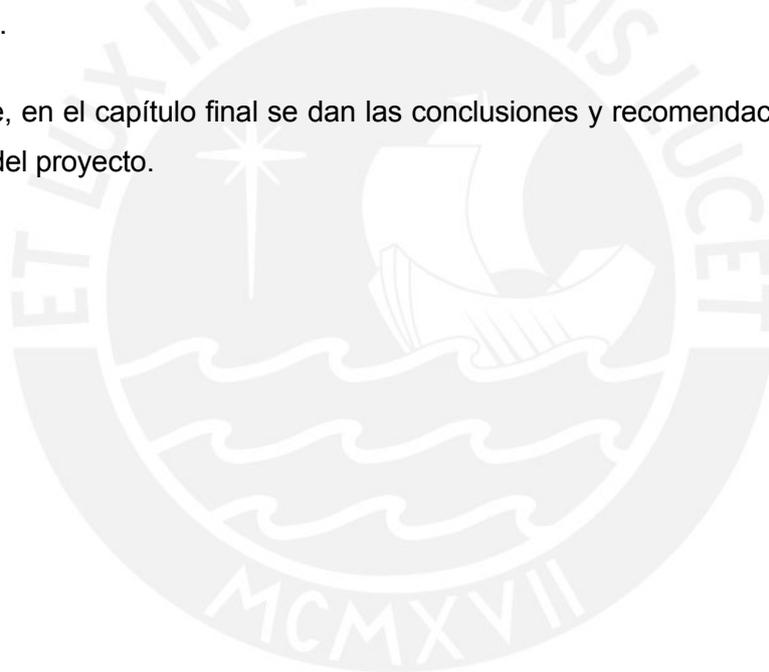
El trabajo se ha organizado de la siguiente forma:

En el primer capítulo se presenta una introducción teórica al mundo de la telefonía IP y los IVR, así como también las tecnologías que se aplican en conjunto para brindar al usuario distintas capacidades, comparando finalmente las tecnologías propietarias con las de software libre.

En el segundo capítulo se presenta el análisis y la arquitectura de la solución tras el estudio de la realidad y las variables que involucran el problema, elaborando los alcances del sistema.

En el tercer y cuarto capítulo se muestra el diseño de la solución e implementación de la misma, para luego iniciar un plan de pruebas del sistema funcional cuyos resultados son analizados.

Finalmente, en el capítulo final se dan las conclusiones y recomendaciones así como la viabilidad del proyecto.



Capítulo 1

Estudio Teórico de la Tecnología IVR

En este capítulo se analizarán los conceptos de Voz sobre IP y los relacionados a las tecnologías de los sistemas IVR

1.1 Voz sobre IP

Llamamos “Voz sobre IP” (VoIP) a las tecnologías que, trabajando en conjunto, permiten transmitir voz a través de las conexiones a Internet. Los paquetes de voz son encapsulados en el protocolo de Internet IP siendo codificados con un codec y usando protocolos específicos para realizar la señalización; los cuales serán estudiados brevemente en el presente capítulo.

Por otro lado, llamamos Telefonía IP a la convergencia de voz, datos y vídeo en la misma infraestructura, es decir en la misma red, la cual genera procedimientos simplificados de configuración y administración así como también un menor costo en el capital de inversión y mantenimiento.

El gran desarrollo de los codecs y la tecnología ha permitido que las comunicaciones de Voz sobre IP ofrezcan cada vez mayor calidad, siendo las favoritas en muchos casos para el público debido al bajo presupuesto requerido y simplicidad.

1.1.1 Protocolos de Señalización

Los protocolos de señalización son los encargados del establecimiento y gestión de mensajes de estado entre los puntos extremos que participan en una llamada. Estos protocolos indicarán el paso de los flujos de voz, encapsulados en paquetes RTP/RTCP por la red hasta llegar a su destino. En esta sección analizaremos los protocolos SIP y IAX, haciendo referencia al protocolo H323 de la ITU.

1.1.1.1 SIP

El Protocolo de Inicialización y Señalización (SIP) nace en el año 1996 como la respuesta de la IETF (Internet Engineering Task Force) a las dificultades presentadas por el protocolo H.323 de la ITU, convirtiéndose en un estándar en el año 1999 [RFC2543] siendo luego mejorado reemplazado en el 2002 [RFC3261].

Al igual que H.323, el protocolo SIP hace uso de RTP y UDP para hacer transferencia de voz, usando una única petición para enviar la información que se requiere, teniendo como ventaja una mayor rapidez y eficiencia en muchos casos que otros protocolos como H323, trasmite la información en modo texto que hace mucho más fácil la interpretación. SIP hace uso del protocolo SDP (Protocolo de Descripción de Sesión) para poder negociar antes de empezar el flujo RTP, los acuerdos realizados como el codec a usar y el tipo de media a enviar.

La sencillez y familiaridad que existe entre SIP y otros protocolos como HTTP o SMTP ha posicionado a este protocolo en un excelente lugar en la industria y la preferencia de los programadores; sin embargo SIP no puede atravesar NATs (Traductores de direcciones de red) que abundan en la vasta arquitectura del Internet de hoy en día. Esto se da debido a que la señalización y los flujos RTP son transmitidos por puertos diferentes, siendo asignado para los flujos RTP un puerto aleatorio, y por ende difícil para controlar la traducción. Existen otras estrategias como los NAT Transversal,

ICE, TURN y los servidores STUN las cuales dependen mucho de un tipo de escenario para una determinada utilización. Aún así, SIP es hoy en día uno de los protocolos de VoIP más populares.

1.1.1.2 IAX

El protocolo IAX fue creado por Digium para que los servidores Asterisk puedan establecer troncales y así comunicarse entre ellos. Su nombre: Inter Asterisk Exchange.

IAX posee la capacidad de englobar múltiples sesiones en un sólo flujo lo que puede significar un tremendo ahorro en ancho de banda. [QUI2006]

La gran ventaja de IAX frente a otros protocolos de VoIP es su naturalidad ante los NATs, ya que no cuenta con el problema previamente descrito para el protocolo SIP; sino que atraviesa los traductores debido a que tanto la señalización como los flujos RTP se transmiten por el mismo puerto UDP (4569).

La versión actual del protocolo es el IAX2 y ha sido deliberadamente diseñado para trabajar a través de firewalls (contrafuegos) y NATs, manteniendo los túneles creados al mínimo como una cuestión de seguridad, haciendo de IAX el protocolo más simple de implementar. Sin embargo, la desventaja es que IAX aún no cuenta con el respaldo de la estandarización. [IAX2005]

1.1.1.3 Comparaciones entre ambos protocolos

Tras haber estudiado ambos protocolos, podemos notar los siguientes puntos importantes que los diferencian claramente:

- SIP es un protocolo capaz de transportar no solo voz sino también video, datos y cualquier flujo que se le asigne; pues como ya fue indicado anteriormente, es un protocolo de inicialización y establecimiento de sesión. IAX es un protocolo que solamente transporta voz y video haciéndolo menos flexible ante SIP.

- SIP requiere del uso de un puerto de control así como también un puerto para cada flujo RTP, por lo que necesitará 3 puertos como mínimo en total. Esto es una desventaja comparada con la capacidad de IAX de usar un único puerto sin importar el número de flujos existentes, habilidad que lo hace más resistente frente a NATs sin necesidad de otras técnicas.
- IAX tiene muy bien diferenciadas las funciones de capa 2 y capa 3, indicando que la señalización y el audio tienen estados definidos, pudiendo ser más eficaz en situaciones como una terminación de llamada, donde en un extremo la comunicación se corta abruptamente. SIP debe implementar estándares adicionales a los definidos en el estándar [RFC3261] pues su comportamiento nativo es bastante pobre.
- SIP está implementado en casi cualquier teléfono IP que podamos hallar actualmente en el mercado mientras que unos pocos cuentan con IAX2.

Aún cuando el protocolo H.323 puede aún ser aplicado en terminación de llamadas, éste es un protocolo antiguo y complejo diseñado para desempeñarse sobre cualquier red.

Para el desarrollo del presente trabajo, se ha decidido utilizar el protocolo SIP dada su orientación al mundo IP, la estandarización existente y el apoyo de la IETF.

1.1.2 Codecs

Los codecs (codificador – decodificador) son modelos matemáticos que nos brindan la información suficiente para poder interpretar información como si esta fuera completa. Estos codecs son importantes ya que con ellos enviamos la cantidad de datos suficientes de acuerdo a un tipo de calidad esperado aligerando la carga de los paquetes de voz.

TABLA 1.1. DISTINTOS CODECS PARA VOIP

Fuente:[VAN2005]

Codec	Tasa de bits de datos	Licencia
G.711	64 kbps	No
G.726	16, 24, o 32 kbps	No
G.723.1	5.3 o 6.3 kbps	Sí
G.729A	8 kbps	Sí
GSM	13 kbps	No
iLBC	13.3 kbps o 15.2 kbps	No
Speex	Variable (2.15 y 22.4 kbps)	No

1.1.2.1 G.711

Desarrollado por la ITU, es el codec nativo de la PSTN y su tasa de transmisión es de 64 kbps que usa compansión (comprensión-expansión) [VAN2005], un tipo de comprensión que dependiendo de la zona usa la ley μ en Norteamérica y la ley A en el resto del mundo.

1.1.2.2 G.729

Este codec fue desarrollado también por la ITU y es usado en aplicaciones de VoIP dada su mínima tasa que es de 8 kbps, ofreciendo gran calidad de audio debido al uso de la Predicción Lineal de Código Algebraico de Estructura Conjugada (CS-ACELP). Contrastando el poco ancho de banda utilizado, observamos la gran capacidad computacional requerida, que puede ser un gran inconveniente en sistemas convencionales. Si bien es un codec que requiere licencia, existen implementaciones de uso gratuito.

1.1.2.3 GSM

El codec GSM (Global System of Mobile communications) proviene del conocido sistema de comunicaciones móviles y es del tipo RPE-LTP (Regular Pulse Excitation Long-Term Prediction) Proporciona una tasa de 13kbps ofreciendo una buena calidad con gran simpleza de proceso para aplicaciones de tiempo real, mientras que los códigos CELP requieren un tiempo para el proceso DSP o caso contrario, requieren un procesador digital de señales para su reproducción en tiempo real. [BER2006].

1.2 CTI: Computer Telephone Integration

1.2.1 Definición

Llamamos CTI al conjunto de componentes tanto de hardware como de software capaz de traducir señales entre sistemas telefónicos y computadores con el objetivo de manipular llamadas para integrarlas a servicios especiales o fijar una ruta idónea para su atención por medio de procesos computacionales. Para esto la interacción entre una PBX y una base de datos es básica.

El uso de la tecnología CTI se está incrementando en un 30% anualmente especialmente en el segmento SOHO (Small Office Home Office) [BEA2006] siendo los sistemas de respuesta vocal interactivos (IVR) la aplicación más popular de esta tecnología dada su cercanía al público en general.

1.2.2 Aplicaciones

Con el avance y popularidad de la Voz sobre IP, las aplicaciones en CTI han crecido, así como la posibilidad de desarrollo en software para el manejo, proceso y ruteo de llamadas. Entre las posibles aplicaciones podemos encontrar [DOD2002]:

- Control y Gestión de llamadas entrantes a un cliente.

- Softphones (Teléfonos implementados en software)
- Reconocimiento automático de llamadas basado en el identificador de llamadas o el ANI (Número automático de identificación) idóneos para el desarrollo de centros de contacto.
- Respuesta interactiva vocal automática para servicios de información o transacciones telefónicas
- Rastreo de llamadas y eventos en un sistema telefónico desde un servidor.
- Enrutamiento de llamadas según habilidades (SBR – Skill Based Routing) que permite un evaluar los datos ingresados vía un IVR en una llamada y su redirección al agente más idóneo para la resolución de problemas

1.3 IVR - Interactive Voice Response

1.3.1 Definición

Los IVR (Sistemas Interactivos de Respuesta de Voz) – también llamados VRU por Unidad de Respuesta Vocal – son sistemas que ofrecen aplicaciones a usuarios por medio del teléfono como dispositivo de entrada de información. Los IVRs contienen programados mensajes y respuestas estáticas y/o dinámicas de acuerdo a los requerimientos del usuario y alcances del propio sistema.

Los sistemas IVR correctamente diseñados solo deben requerir a los usuarios contar con un teléfono y una línea POTS (Sistema antiguo de telefonía plana) y deben ser capaces de atender requerimientos de tal forma que sean usados por cualquier persona, desde cualquier lugar y a cualquier hora. [YAR2003]

Estos sistemas han evolucionado desde sistemas de respuesta de teclas hasta el más preciso reconocimiento vocal gramatical, evolución que veremos en esta sección

1.3.2 Justificación de los Sistemas IVR

1.3.2.1 Sistema

El panorama previo a la existencia de los sistemas IVRs, que es aún muy usado en muchas empresas del negocio telefónico, no nos es ajeno: un grupo de agentes sentados con un teléfono recibiendo llamadas telefónicas direccionándolas (o haciéndolas), con un sistema de espera en el mayor de los casos.

En un estudio realizado en Mayo del año 2001, El Grupo Gartner reportó que el costo promedio de un agente humano por transacción telefónica era de \$5.50 contra los \$0.45 de un sistema IVR. Reducción de costos similares son apreciados cuando un sistema IVR de reconocimiento de voz es implantado. [YAR2003]

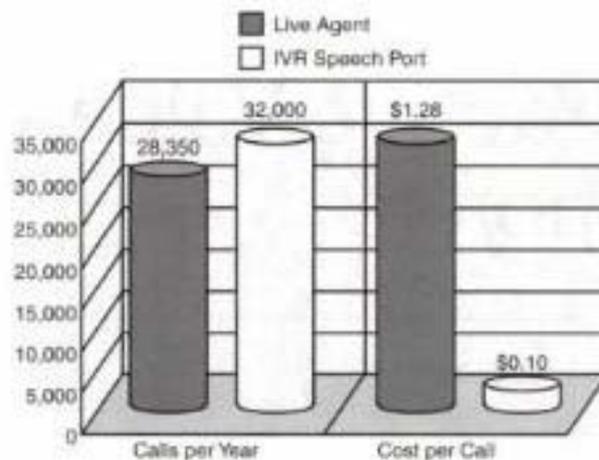


FIGURA 1.1. COSTO COMPARATIVO UNA LLAMADA PROCESADA CON UN AGENTE HUMANO CONTRA UN IVR DE RECONOCIMIENTO DE VOZ

Fuente: [YAR2003]

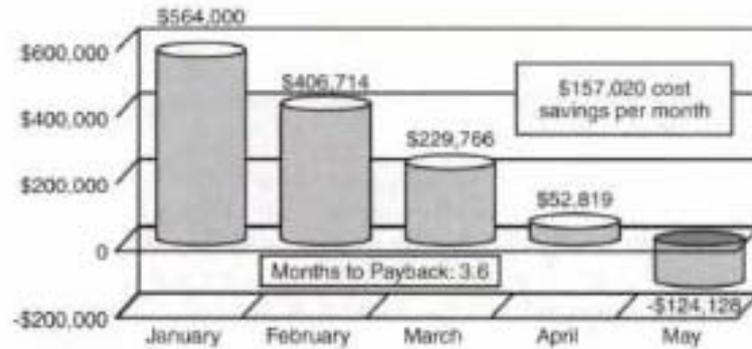


FIGURA 1.2. COSTO DE RECUPERACIÓN EN EL TIEMPO DE LA INVERSIÓN INICIAL EN UN IVR DE RECONOCIMIENTO VOCAL (NUANCE)

Fuente: [YAR2003]

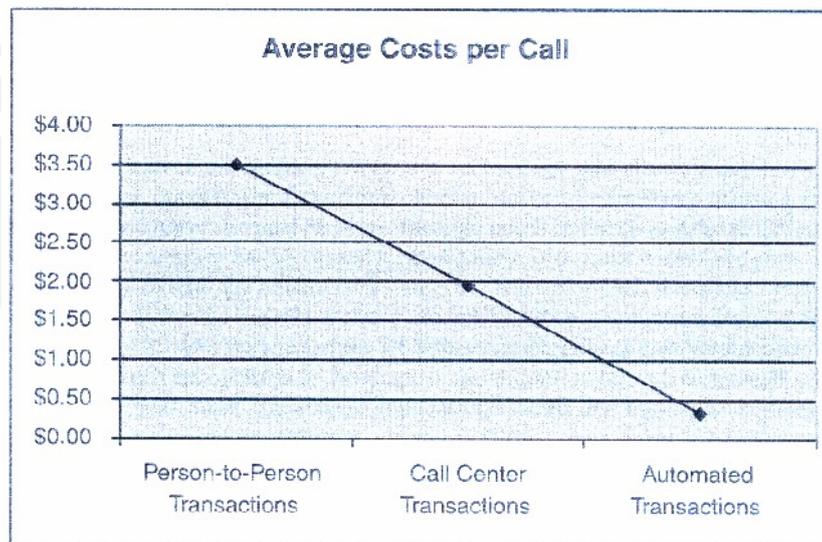


FIGURA 1.3. COSTOS PROMEDIO POR LLAMADA (SPEECH WORKS)

Fuente: [SHA2002]

1.3.2.2 Usuario

Si bien las personas siempre son algo renuentes ante el hecho de interactuar con una máquina que les despliegue opciones, [BRO2004] un hecho real sobre los IVRs es que todos podemos usarlos. Su facilidad de uso y el hecho de contar con el teléfono como equipo usuario, un dispositivo ya posicionado y parte de la vida de todos, hace que desde un niño hasta un anciano puedan acceder a la información que un IVR esta dispuesto a ofrecer.

1.3.3 Evolución de los sistemas IVR y Tipos

Los sistemas IVR han ido evolucionando con el tiempo, siendo su mayor aliado en crecimiento el procesamiento digital de señales (PDS) que ha hecho posible el reconocimiento de voz, así como otras herramientas como TTS, haciendo a los IVRs cada vez más “humanos” e interactivos.

Como podemos observar en la Figura 1.4, no en todos los casos la evolución ha traído un nuevo tipo de IVR reemplazando a los demás sino que, muchas tecnologías son mezcladas de acuerdo a las necesidades de la empresa que implanta al sistema y la comodidad que desea brindar a sus clientes.

Inicialmente, en los IVR basados en tonos duales de multifrecuencia (DTMF) el usuario sólo tiene la capacidad de elegir entre los menús que le son desplegados vocalmente presionando las teclas e introduciendo códigos o números de tarjeta de crédito vía el teclado telefónico.

Más adelante llegaron los sistemas de reconocimiento de voz, que podían detectar tan sólo frases claves como “reservación de aerolíneas”, parte de un menú que puede ser hablado por los usuarios; sin embargo, el sistema IVR aún mantiene al usuario en un ambiente de opciones fijas.

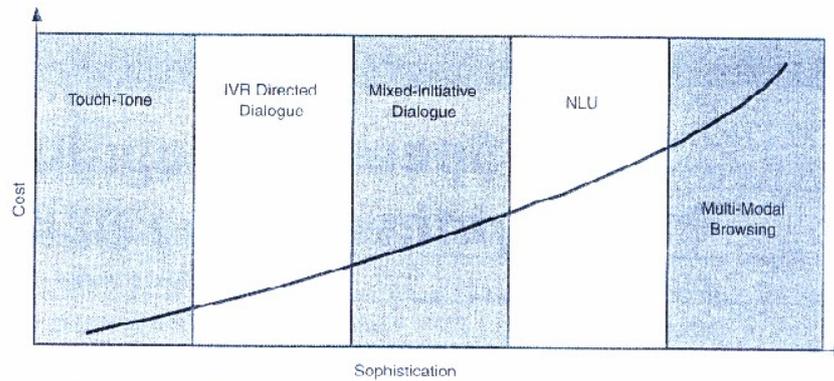


FIGURA 1.4. EVOLUCIÓN DE TECNOLOGÍAS USADAS EN SISTEMAS IVR MOSTRANDO COSTO VERSUS SOFISTICACIÓN

Fuente: [SHA2002]

El avance del procesamiento digital de señales nos trae IVRs capaces de entender requerimientos del usuario pronunciados en oraciones gramaticalmente entendibles, siendo inicialmente en los “Sistemas de iniciativa mezclada” respuestas a preguntas hechas por el sistema, para luego pasar a requerimientos propios del usuario, dándoles libertad de hablar y construir oraciones con sentido completo así como preguntas, todo esto posible gracias a la tecnología de “Entendimiento Natural de lenguaje” (NLU). Esto es posible mediante el uso de VXML (Voice Extensible Markup Language) el cual es un lenguaje de programación creado para la interacción vía voz con elementos de navegación en el mundo del Internet.

Como el tipo de IVR más avanzado podemos encontrar el de búsqueda multimodal, el cual puede regresar distintos tipos de contenido como video o imágenes a un Terminal que lo soporte (PDA, celular, etc)

Una comparación entre la interacción entre el usuario y el sistema en los diferentes tipos de IVR puede verse en la Tabla 1.2.

**TABLA 1.2. EJEMPLOS DE INTERACCIÓN DE DIFERENTES TIPOS DE
TECNOLOGÍA IVR**

Fuente: [SHA2002]

Tecnología	Interacción Usuario - Computadora
Tonos de marcado	C: Presione 1 para reservaciones en aerolíneas
IVRs de diálogo dirigido	C: ¿Desea escuchar reservaciones en aerolíneas o cronogramas de vuelo?
	U: Reservaciones de aerolínea
Diálogo Iniciativo Mixto	C: ¿Desea escuchar reservaciones en aerolíneas o cronogramas de vuelo?
	U: Deseo viajar de Boston a New York mañana por la mañana
NLU	C: United Airlines, ¿cómo puedo ayudarlo?
	U: ¿Me podría reservar un asiento de primer clase a Hong Kong? Quiero un vuelo temprano por la mañana saliendo pasado mañana
Navegación Multimodal	U: Muéstreme el pronóstico del clima
	C: (Información regresa sobre el teléfono)

1.3.4 Partes y Tecnologías anexas de los Sistemas IVR

Como parte de la arquitectura básica de un sistema IVR tenemos principalmente las líneas telefónicas que pueden ser 24 líneas (un T1) y los respectivos números telefónicos de éstas.

1.3.4.1 PBX

El Private Branch Exchange es el dispositivo encargado de realizar la conmutación de las llamadas entrantes y las llamadas salientes al sistema, permitiendo conectar teléfonos de manera interna sin intervención del proveedor telefónico, lo que permite a empresas mantener redes internas con comunicación libre de facturación.

Las PBX nos ofrecen interfaces a determinado número de troncales de distinto tipo por medio de las cuales los anexos o extensiones podrán realizar o recibir llamadas al exterior, realizando así la conmutación.

Las PBX – IP son las centralitas basada en software capaces de transmitir voz sobre IP basándose en protocolos previamente descritos y poder interactuar también con la red telefónica pública conmutada (PSTN)

Algunas PBX poseen un software ACD (Authomatic Call Distribution), el cual posee una programación de cómo encaminar las llamadas y que hacer en casos de espera.

Las interfaces más comunes en una PBX son:

- FXO: (Foreign Exchange Office)
- FXS: (Foreign Exchange Station)
- E&M (Conexión entre centrales)
- BRI (Acceso básico ISDN)

- PRI (Acceso Primario ISDN)

Las PBX – IP usarán sólo interfaces de este tipo para comunicarse con la PSTN. Estas interfaces se adquieren a manera de tarjetas o módulo compatibles con una PC.

1.3.4.2 IVR (VRU)

Es el corazón del sistema el cual se conecta directamente al PBX (o también a la PSTN en otros casos) para poder contestar automáticamente ciertos requerimientos. De acuerdo a su tipo puede contar con los siguientes tipos de componentes:

1.3.4.3 TTS (Text to Speech)

Este software es el encargado de generar voz natural en tiempo real de cualquier texto arbitrario que se le introduzca al sistema o que es leído de una base de datos.

1.3.4.4 ASR (Automatic Speech Recognition)

Este software está dedicado al reconocimiento vocal y debe ser instalado y configurado en el mismo IVR además de ser condicionado a la forma de hablar en las regiones que se desean cubrir.

1.3.5 Sistemas IVR en el Mercado

1.3.5.1 Mundo Empresarial

Se ha explicado ya la utilidad de los IVRs para poder recibir requerimientos de los clientes de mejor y más eficiente forma que agentes humanos. Los IVR actualmente se encuentran presentes en sistemas de comunicación, servicios financieros, servicios de salud entre otros. Sin embargo, el avance en el

mercado de los IVRs no está en la incursión en nuevos mercados sino en el avance tecnológico y soporte que las empresas puedan brindar a sus clientes así como mejoras y calidad en los servicios que ofrecen.

Una de las empresas más fuertes según Beasty Collin [BEA2006] es “Avaya” que estableciendo una alianza con “Intervoice”, alcanzó en el año 2005 un crecimiento del 20% en ventas. En el 2005 Avaya lanza el “Avaya Voice Portal”, una solución basada en el protocolo IP que soporta SIP y es perfecto para la actual transición a telefonía IP.

Otra empresa como Nortel Networks ha incluido también soporte para VoIP y VXML (Voice XML), pero el retraso en la inclusión a estas tecnologías podría haber sido la razón por la cual hayan obtenido la menor puntuación en el ítem “satisfacción del consumidor” mostrado en la siguiente tabla :

TABLA 1.3. ENCUESTA AL SECTOR CLIENTES SOBRE PREFERENCIAS EN SISTEMAS IVR Y REPUTACION DE LOS MISMOS

Fuente: [BEA2006]

IVR	Satisfacción del Cliente	Funcionalidad	Dirección Empresarial	Top 3 Verticales
Avaya	3.5	3.6	4	Servicios financieros, de Salud y Comunicaciones
Genesys Telecommunications Labs	4	4.2	3.9	Comunicaciones, Finanzas, Salud
Nortel Networks	3.3	4	4	Comunicaciones, Servicios Financieros, Salud, Tecnología

La empresa más reconocida actualmente es Genesys Telecommunications Laboratory, derrocando a Avaya e Intervoice, recibiendo el mayor puntaje en el campo “satisfacción del cliente”. La plataforma Genesys Voice Platform ofrecerá interconexión con sistemas open source, enfocándose en una solución robusta que incluya VXML.

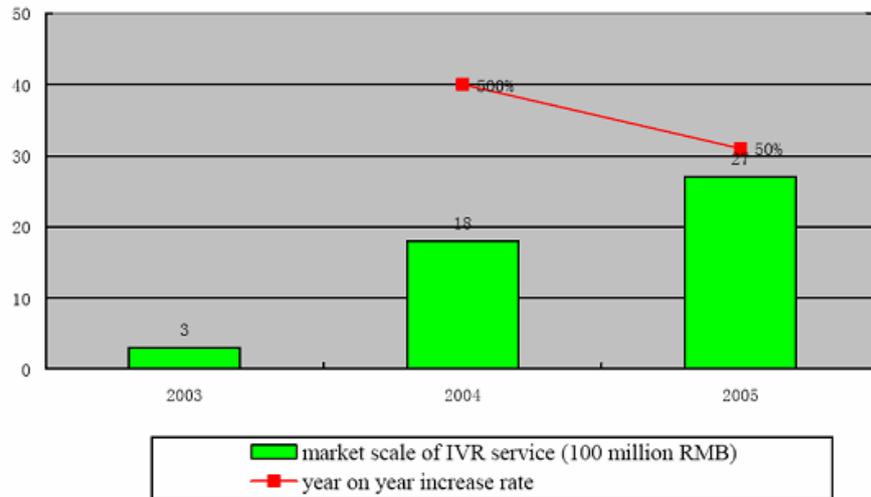
1.3.5.2 Mercados Mundiales

Es cierto afirmar que el mercado de los sistemas IVR se ha incrementado alrededor de un 90% en la última década desde su aparición. El apoyo de tecnologías nuevas como VXML y sistemas integrados han disparado las ventas dado el ahorro generado al sector empresarial. [FRO2006]

En los Estados Unidos estos sistemas se incrementaron notablemente en el 2004 y el 2005. En el 2005 el mercado generó aproximadamente \$564.8 millones de dólares en sistemas IVR, esperándose un crecimiento acelerado a una tasa ponderada de crecimiento anual cercana a 13.4 por ciento. Los pronósticos ofrecen un crecimiento del mercado en el 2008 y el 2009 para luego estabilizarse y más adelante experimentar ligeros descensos. [FRO2006]

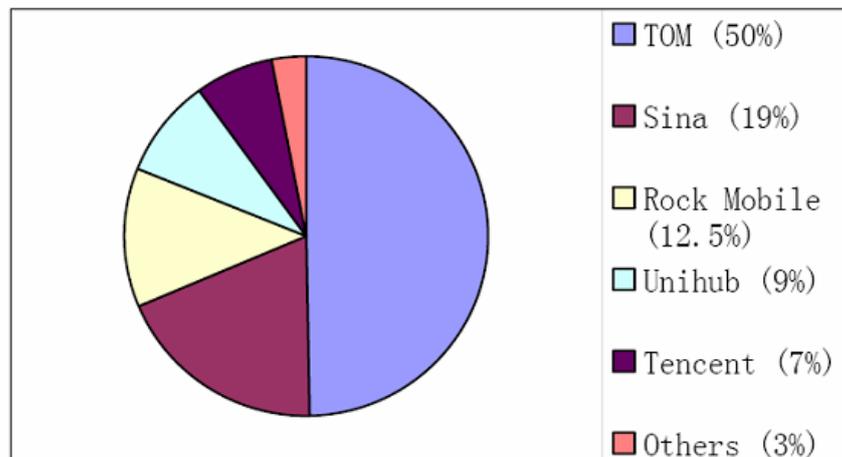
Similar, pero aún más prometedor es el caso de China, cuyo mercado aún se encuentra en la fase formadora, pero a una velocidad impresionante gracias al giro innovador que han dado al servicio. Si bien el mercado actual se encuentra centralizado, hay una gran diversidad de empresas ya establecidas y con un porcentaje del mercado, que si bien no es alto en un análisis occidental, nos arroja grandes cifras en usuarios dada la densidad poblacional de este país del oriente.

En China, los servicios IVR han alcanzado un auge como servicios de valor agregado a las empresas de telefonía celular, de tal modo que un usuario marcará un determinado número para ser informado sobre el clima, el deporte, ingresar a una sala de Chat con comandos de voz o hacer órdenes de compra. [ZHA2006]



**FIGURA 1.5. EVOLUCIÓN DE MERCADO CHINO EN SISTEMAS IVR
MOSTRANDO TASAS DE INCREMENTO**

Fuente: [ZHA2006]



**FIGURA 1.6. EMPRESAS QUE COMPARTEN EL MERCADO CHINO EN
SISTEMAS IVR**

Fuente: [ZHA2006]

Desde la segunda mitad del 2003, los sistemas IVR para móviles se volvieron los servicios de valor agregado que más remesas dejaron a las empresas, dejando incluso atrás a los populares SMS (Short Message Service) y MMS (Multimedia Message Service).

Entre las empresas que ofrecen estos servicios tenemos a TOM que logró \$24.5 millones de dólares en el primer cuarto del 2004, SINA que obtuvo \$10.6 millones de dólares y TENCENT con \$3.78 millones de dólares entre otras con alto grado de facturación.

1.3.6 Similitudes y Diferencias entre software libre y soluciones propietarias

Los “Softswitches” son los dispositivos encargados del control y procesamiento de llamadas sobre una red de conmutación de paquetes.

Uno de los softswitches más populares y recientes es “Asterisk” de Digium, el cual nos da la ventaja de prescindir de un hardware dedicado a esta tarea en telecomunicaciones pues son capaces de implementar centrales basadas en software, reduciendo los precios y cambiando radicalmente la telefonía, siendo tan sólo otro servicio corriendo en un servidor.

Muchas empresas como Dialogic (ahora parte de Intel) y Natural Microsystems están introduciendo al mercado versiones sólo en software de sus equipos de telefonía computarizada y conmutadores, pudiendo ser soluciones levantadas en servidores comunes, por lo que el crecimiento de los softswitches es una tendencia actual y real que acerca al usuario final a este tipo de sistemas que antes eran solo sistemas cerrados ofrecidos a grandes empresas.

Asterisk es un sistema de implementación libre de una central telefónica o PBX que permite todas las funcionalidades de ésta pudiendo servir también como una pasarela a la red telefónica. Para usar Asterisk sólo se requiere como mínimo una computadora personal o servidor al que se le pueden añadir periféricos

como tarjetas para poder hacer conexiones hacia la red telefónica. Si bien Asterisk puede funcionar sobre muchos sistemas operativos, GNU/Linux es aquel que ofrece mayor soporte, obteniendo de ésta forma una solución basada en software libre, libre de necesidad de licencias y amplias posibilidades de configuración de servicios como los IVR.

Frente a otros competidores como las soluciones de Microsoft como lo son Speech Server 2007, sobre un Windows Sever 2003 o similar, Asterisk nos brinda una solución de costo mucho menor debido a las licencias de uso que las soluciones de Microsoft necesitan.

En contraste, las interfaces de Microsoft para la realización de IVR son mucho más amigables y comprensibles, mientras que nativamente en Asterisk sólo se configuran IVRs mediante la manipulación de archivos de texto, aunque actualmente muchas interfaces gráficas s en diferentes lenguajes de programación han aparecido.

Por otro lado, Asterisk también tiene debilidades como lo son la incertidumbre sobre la cantidad de CPU disponible para llevar acabo las tareas de conmutación y servicios telefónicos en un determinado instante; la escalabilidad y la falta de conocimiento del número de usuarios que podrá soportar un servidor corriendo Asterisk, lo que requiere estudios y pruebas de desempeño.

Asterisk es un sistema joven que aún tiene un largo camino por recorrer y cuyas versiones superiores seguro incluirán factores que la competencia si ofrece como es el soporte nativo de VXML, importante en la nueva generación de IVRs.

Capítulo 2

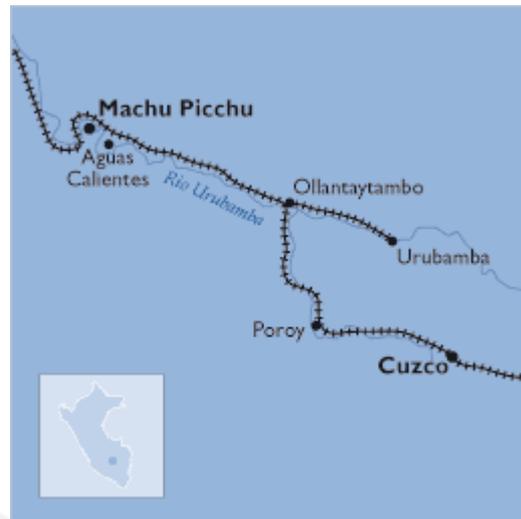
Análisis y Arquitectura de la Solución

En este capítulo se definirá el problema o caso práctico que se resuelve en esta tesis, se darán los alcances y limitaciones del proyecto y se dará una propuesta de solución sobre la cual se hará un diseño posterior.

2.1. Definición del problema a resolver

La ciudad del Cuzco es reconocida mundialmente como la capital del imperio Inca dado su legado histórico y su riqueza arquitectónica. La mayor atracción para los turistas que visitan la ciudad imperial es la ciudadela de Machu Picchu, ubicada a 130 kilómetros al noroeste de la Ciudad del Cuzco, en la cresta del cerro Machu Picchu (Montaña Nueva) ubicado en el valle de Urubamba, a 2490 metros sobre el nivel del mar.

El método de acceso más utilizado para acceder al complejo turístico de Machu Picchu es el viaje en tren que dura aproximadamente 3 horas desde la Ciudad del Cuzco – aunque hace pequeñas paradas – hasta el pueblo de Aguas Calientes desde donde se toman buses para subir hasta la ciudadela ubicada en la cima del cerro.



**FIGURA 2.1. RECORRIDO GEOGRÁFICO DEL TREN CUZCO – MACHU
PICCHU**

Fuente: [ORI2005]

Dado el gran número de turistas extranjeros y nacionales que optan por el tren como vía de acceso – un total de 411,709 visitantes extranjeros y 128,595 nacionales en el 2005 ascendiendo a un máximo mensual de 51,453 extranjeros y 23,204 nacionales [INC-Cuzco] – el siguiente proyecto busca elevar el flujo turístico mediante la implantación de un sistema de reservas telefónicas para visitantes, automatizando el proceso para que los propios turistas puedan hacer sus reservas y luego sólo proceder a pagarlas.

De ésta forma no sólo los turistas de cualquier procedencia sino también los agentes de viaje podrán realizar reservas a corto y largo plazo en una interfaz vocal que verificará disponibilidad en tiempo real.

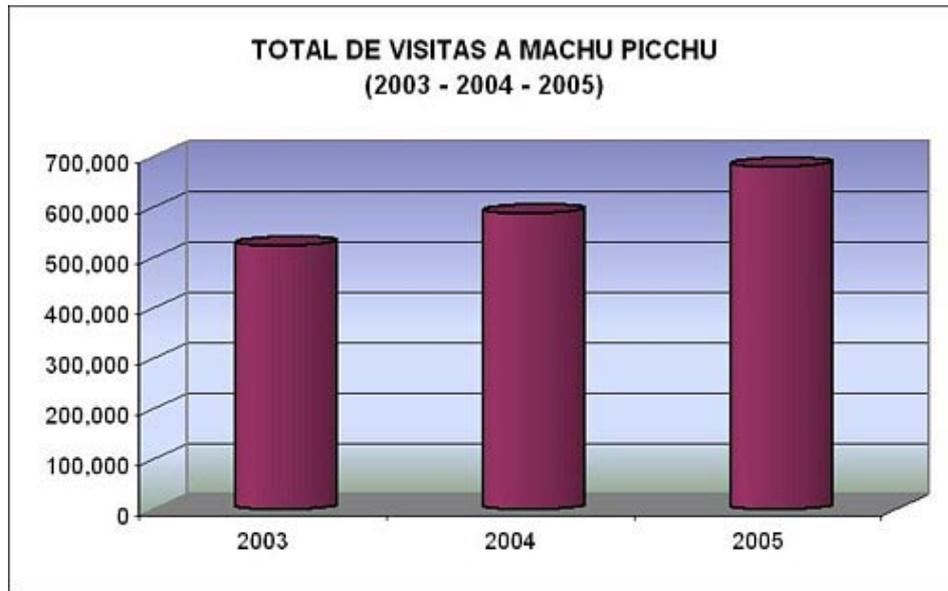


FIGURA 2.2. TOTAL DE VISITAS A MACHU PICCHU DESDE EL AÑO 2003 AL 2005

Fuente: [TN2005]



FIGURA 2.2. VISTANTES EXTRANJEROS VÍA TREN

Fuente: [TN2005]

De acuerdo al servicio ofrecido por Perurail (Orient Express), las variables principales del sistema son los horarios de salida y llegada así como los tipos de vagones (clases) en los cuales un turista o agente puede reservar asientos.

TABLA 2.1. PRECIOS DE IDA/VUELTA DE LOS VAGONES PERURAIL

Fuente: [ORI2005]

SERVICIO (Ruta Cusco-Machu Picchu)	PRECIOS Ida y vuelta		PRECIOS Ida o vuelta	
	Dólares	Nuevos Soles	Dólares	Nuevos Soles
Hiram Bingham	US\$ 476.00	S/. 1,608.88	No aplica	No aplica
Vistadome	US\$ 101.15	S/. 341.89	US\$ 59.50	S/. 201.11
Backpacker	US\$ 65.45	S/. 221.22	US\$ 41.65	S/. 140.78

2.2. Alcances y Limitaciones del Proyecto

- El proyecto será un prototipo de sistema IVR IP de respuesta a tonos de marcado multifrecuencia. Para la reserva de boletos para el tren de Cuzco a Machu Picchu
- El prototipo IVR será multi idioma, considerando inicialmente el español y el inglés.
- Las reservas realizadas tendrán una vigencia de 2 días para ser pagadas, de lo contrario caducarán en el sistema.
- La reserva a realizar deberá ser por lo menos con 1 día de antelación a la fecha del inicio del viaje.

- Cada reserva constará de hasta un máximo de 6 asientos
- La validación de usuarios se realizará por medio de códigos de inscripción autogenerados.
- El sistema enviará un correo electrónico con los datos de la reserva al correo electrónico especificado por el usuario.
- El proyecto incluirá una interfaz web de administración de reservas e inscripción de códigos de reserva.

2.3. Propuesta de la Arquitectura de la Solución

La Arquitectura propuesta contará de tres elementos:

- La PBX-IP
- El servidor de aplicaciones
- La Base de Datos

El servidor de aplicaciones se encargará de hacer las conexiones a base de datos y de recibir la información de entrada que dará el usuario al sistema IVR así como también procesar la información para brindar una respuesta adecuada, albergando el script del IVR mismo. La PBX-IP será una solución en software libre con la configuración de anexos; mientras que finalmente la base de datos guardará las tablas necesarias para el sistema así como la información de usuarios y reservas.

El servidor de aplicaciones se encargará también de alojar el sistema web de inscripción de usuarios y requerimiento de información de la base de datos.

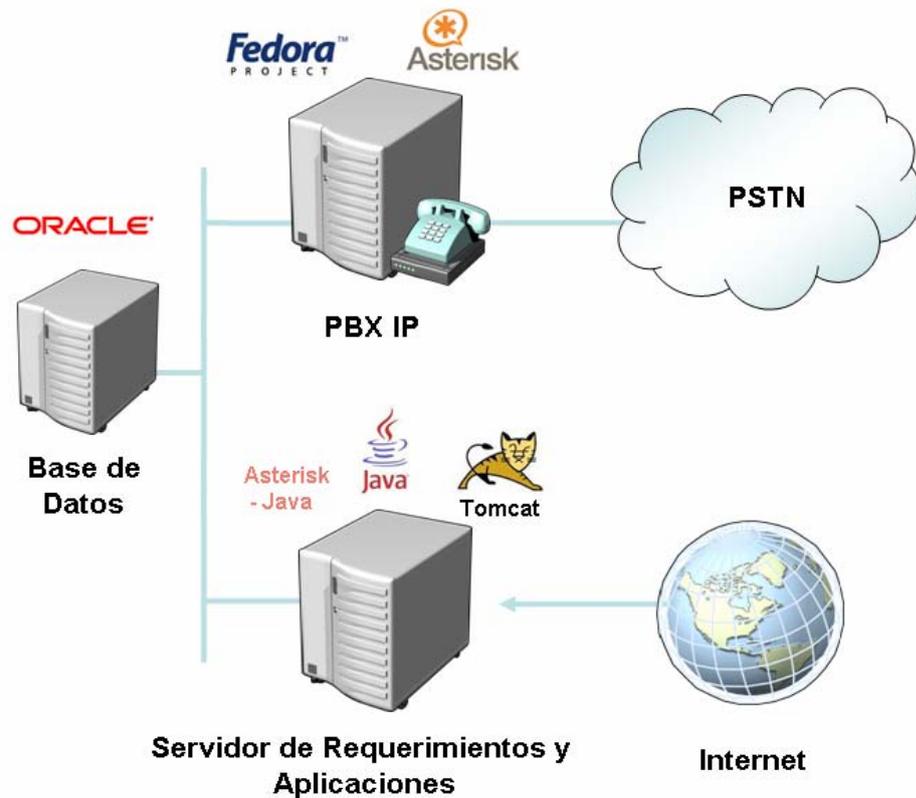


FIGURA 2.3. ARQUITECTURA DEL DESARROLLO

2.3.1. Tecnologías de la Solución

A continuación se detallarán las herramientas para el diseño e implementación del sistema, indicando la justificación de la debida elección.

2.3.1.1 PBX IP

El servidor PBX estará basado en el sistema operativo Linux (Fedora Core 5) operando con Asterisk. Éste software de código abierto es una aplicación de una central telefónica creado en 1999 por Mark Spencer de Digium. [VAN2005] Asterisk se encuentra bajo licencia GPL (GNU General Public License) y actualmente se encuentra en su versión 1.4.1.

Asterisk soporta muchos protocolos de VoIP, como los descritos en el capítulo anterior y presenta capacidades que anteriormente sólo eran encontradas en equipos propietarios, llevadas a cabo por medio de implementaciones en software y arquitecturas funcionales.

Para el desarrollo del proyecto de tesis, se utilizará Asterisk en su versión 1.4.1. dada la capacidad de esta versión de interactuar con paquetes de software como Asterisk – Java, ha ser descritos posteriormente.

2.3.1.2 Lenguaje de Programación

El lenguaje de programación elegido es Java. Éste fue desarrollado por Sun Microsystems a principios del año 1990 y se ha convertido en un lenguaje de programación muy popular dada su robustez, simple sintaxis e interoperabilidad. Entre las características principales de Java tenemos:

- Lenguaje de programación Orientado a objetos (POO) que permite la reutilización de código, agilizando el desarrollo de software en la creación de sistemas de mayor complejidad.
- Lenguaje multiplataforma: Funciona de manera similar en diferentes sistemas operativos. Esto se debe a la interpretación del lenguaje realizado por la máquina virtual Java JVM; dado que normalmente un lenguaje compilado es traducido y adaptado a un archivo ejecutable para una determinada plataforma. [MOL2006]
- Tiene capacidades de extender funcionalidades de un servidor web, las cuales se explicarán en el punto 2.3.1.5

Para la realización de este proyecto se ha utilizado la versión 1.5.0_08 del kit de desarrollo Java (JDK)

2.3.1.2.1 Asterisk - Java

El API Asterisk – Java ofrece un conjunto de clases que permite la creación de aplicaciones que puedan controlar y monitorear centrales PBX basadas en

Asterisk 1.0, 1.2 y trabajos futuros en gestión de la versión 1.4. Actualmente este paquete está en su versión 0.3 y está registrado bajo licencia Apache Versión 2.0.

El paquete Asterisk – Java está desarrollado mediante el protocolo FastAGI, por lo que se permite poner en funcionamiento un servidor de requerimientos que recibirá y mandará comandos por un socket TCP. Para esto será necesaria la configuración del archivo `extensions.conf` ubicado en `/etc/asterisk` del servidor PBX.

A continuación se describirá el diseño del soporte de FastAGI del API Asterisk-Java en su paquete `org.asteriskjava.fastagi`. Éste se basa en tres importantes interfaces: `AgiServer`, `AgiScript` y `MappingStrategy`.

- La interfaz `AgiServer` tiene como responsabilidad escuchar los requerimientos AGI provenientes de un servidor Asterisk y luego elegir el proceso para ese requerimiento, invocarlos para proveer los medios para enviar comandos a Asterisk y recibir la respuesta correspondiente. El API Asterisk-Java incluye ya la implementación de esta interfaz en `DefaultAgiServer`.
- `AgiServer` usa una estrategia de mapeo (`MappingStrategy`) para la selección del proceso, y esto se basa en la lectura del recurso y verificando la URL, esto es llamado `ResourceBundleMappingStrategy`.

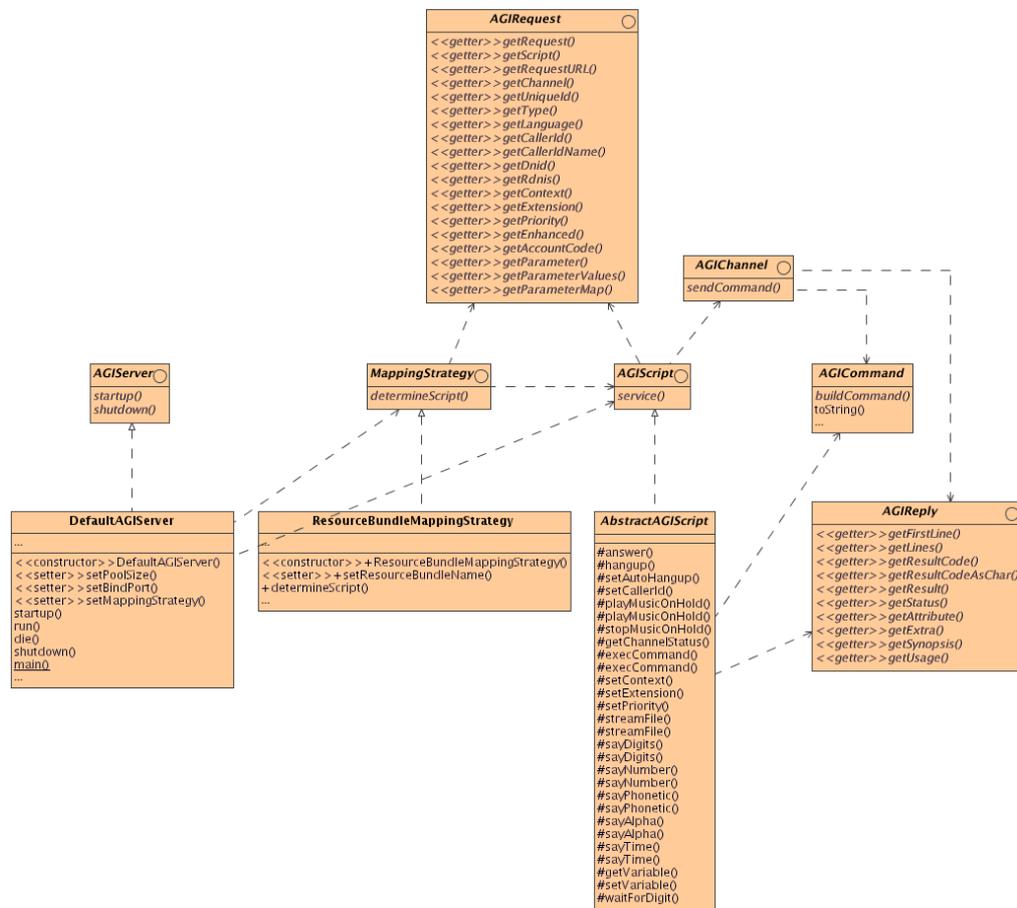


FIGURA 2.4. DISEÑO AGI DEL PAQUETE ASTERISK-JAVA

Fuente: [ASJ2007]

- La tercera interfaz es el AgiScript, el cual se refiere al código mismo invocado para atender un requerimiento. AgiScript es para Asterisk-Java lo que un servlet es para un contenedor de servlets (servlet container). La interfaz AgiScript es bastante simple, usa un método llamado service() al cual se le pasan el AgiRequest y el AgiChannel, permitiendo enviar comandos Agi hacia Asterisk. [ASJ2007]

Para el desarrollo del presente proyecto se utilizarán principalmente los métodos definidos en la clase BaseAgiScript, la cual nos ofrece control sobre acciones de la propia central para pedir y brindar información necesaria para la implementación del sistema IVR. Esta clase está ubicada en

org.asteriskjava.fastagi y sus métodos son el reflejo de los comandos AGI como métodos de una clase extensible en Java, lo que nos permitirá ya en este ambiente generar otras clases necesarias para la interacción con la base de datos.

TABLA 2.2. ALGUNOS MÉTODOS DE LA CLASE BASEAGISCIPT

Tipo	Nombre	Descripción
void	answer()	Contesta el canal
int	exec(String aplicacion)	Ejecuta un comando de una aplicación dado
String	getData(String archivo)	Reproduce un archivo y espera datos para almacenar
void	streamFile(String archivo)	Reproduce un archive dado
void	Saydigits(String cadena)	Reproduce una cadena de dígitos

2.3.1.3 Base de Datos del Sistema

La base de datos elegida para el sistema IVR es Oracle 9i, sistema de base de datos relacional orientado a funcionar más en sincronía con las necesidades del Internet; por lo que es el sistema preferido por las empresas dada su robustez y fidelidad que han hecho de Oracle Corporation la segunda empresa de software más grande del mundo.

Para la interacción con la base de datos desde los Scripts generados en Java para el funcionamiento del Sistema IVR y su interacción con el sistema de base de datos se utilizará el paquete ojdbc14.jar de Oracle 9i JDBC Drivers en su versión 9.2.0.8 compatible con JDK 1.5.

Para una mejor organización de las sentencias SQL (*Structured Query Language*) se utilizarán paquetes y procedimientos almacenados en el mismo servidor de base de datos con el fin de poder hacer transacciones para el desarrollo del sistema.

2.3.1.4 Servidor de Aplicaciones

El servidor de aplicaciones elegido para albergar la aplicación web de inscripción inicial del flujo del servicio IVR es el Apache Tomcat 5.5.12, el cual se encuentra bajo licencia *Open Source* (Código Abierto) y fue creado en conjunto por Apache Software Foundation y Sun Microsystems.

Este servidor de aplicaciones es idóneo para el tipo de páginas web dinámicas que formarán parte de la aplicación dado que se basará en Java Servlets, clases de java que extienden funcionalidades de servidor y otras basadas en el modelo cliente – servidor. Para el funcionamiento de los Servlets, Apache Tomcat nos presenta el contenedor de servlets trabajando en conjunción con el servidor web necesario para el correcto funcionamiento del dinamismo de páginas web.

Una de las ventajas del uso de Apache Tomcat es que al estar desarrollado en Java, sólo requerirá de una máquina virtual Java JVM para funcionar correctamente indistintamente de la plataforma operativa en la que se encuentre.

2.3.1.5 Servidor TTS

Como parte de la tecnología anexa usada en los sistemas IVR, descritos en el capítulo anterior, se ha visto necesaria la inclusión de un servidor TTS (Text to Speech) en el desarrollo del presente proyecto. Para esto se ha elegido el sistema “Festival”, desarrollado por el CSTR (The Centre for Speech Technology Research) de la Universidad de Edimburgo, Inglaterra. “Festival” es un marco de trabajo que permite construir sistemas de síntesis de voz. La versión actual de este sistema es el 2.0 y que está disponible para descarga,

ofreciendo soporte tanto en inglés americano y británico así también como en español.

2.3.1.6 GUI

El diseño de la interfaz gráfica del aplicativo web que acompaña al servicio telefónico es un juego de páginas en HTML dada la sencillez de este lenguaje y la necesidad de tan sólo un navegador web como *Mozilla Firefox* o *Internet Explorer* para su acceso.

La arquitectura MVC o Modelo-Vista-Controlador se encarga de separar la presentación, la lógica de control y el estado de la aplicación con el objetivo de hacer el sistema modular; es decir, una parte puede ser cambiada sin alterar la otra.

El “controlador” será el encargado de recibir los requerimientos y es el responsable de tomar acciones apropiadas en respuesta a cada requerimiento. El “modelo” está referido a la representación del estado de la aplicación en la base de datos y los JavaBeans; Finalmente la “vista” tomará la información provista por el “controlador” y el “modelo” y la presenta al usuario. Cabe notar que la “vista” conformada por páginas dinámicas no debe iniciar ningún tipo de cambio en el estado del modelo ni ningún proceso, pues esto es tarea del “controlador”.

La estática nativa de las páginas HTML se ha solucionado insertando código java en las mismas y así convertir estas interfaces en JSPs (Java Server Pages), que serán utilizadas por la “vista” de MVC en conjunto con el marco Apache Struts y sus action servlets, que se encargarán de la estructura de la aplicación dejando la lógica y negocio del sistema como parte con mayor carga para el programador.

La versión del marco Apache Struts usado es 1.2.2, y este se encuentra bajo licencia Apache.

2.3.1.7 SOX

SOX es una utilidad vía línea de comandos que permite la conversión de archivos de audio a distintos tipos de formatos desde los más conocidos como MP3, CD-R, Ogg entre otros; además de presentar la opción de poder aplicar distintos filtros con el fin de incorporar efectos a las pistas de sonido. SOX se encuentra bajo licencias GPL (General Public Licence) y LGPL (Lesser General Public Licence – GNU Library) y está programado en lenguaje C. [SOX2007]

SOX, creado por el suizo Lance Norskog, existe oficialmente desde 1995, año de su lanzamiento y se ha ido perfeccionando por colaboración de la comunidad Open Source siendo ahora capaz de reproducir y grabar directamente pistas desde Linux y Unix..

Para este proyecto de tesis, SOX fue utilizado para la grabación y conversión de pistas en el formato GSM ya descrito en el capítulo anterior, para que puedan ser reproducidos por Asterisk en el canal.

Capítulo 3

Diseño e Implementación de la Solución

En este capítulo describe el diseño del proyecto presentando los flujos del mismo para luego describir la configuración especial de los elementos del sistema necesarios para el funcionamiento del mismo.

3.1. Diseño de la Solución

3.1.1. Descripción del Servicio

El servicio ofrecido a los turistas por el cual ellos podrán realizar reservas en el tren Cuzco Machu Picchu se divide en 3 partes funcionales:

3.1.1.1 Registro

Durante su llegada a la ciudad de Lima, los turistas serán atendidos en los aeropuertos mientras hacen fila para pasar por migraciones por un grupo de señoritas que los interrogarán sobre sus intenciones de viajar a Cuzco y visitar Machu Pichu. Ante una respuesta positiva, éstas le pedirán su nombre completo, correo electrónico y número de pasaporte que ingresarán en una PC Pocket o un equipo portátil que contendrá un aplicativo web que les

generará un código y será impreso y entregado al llegar a ventanilla de migraciones.

3.1.1.2 Reserva

Usando cualquier terminal telefónico, el turista podrá realizar la llamada al servicio automático por medio de la cual hará su reserva. Dado que la reserva no podrá cambiarse tras haber sido hecha, el usuario contará con las opciones necesarias para realizar cambios y escuchará un resumen de su reserva el cual confirmará. Si la reserva no ha sido confirmada, el usuario podrá colgar en cualquier momento y dar uso a su código de nuevo, sin embargo; tras haber sido confirmada la reserva, el código se bloqueará y no podrá volver a usarse mientras no haya sido liberado.

3.1.1.3 Pago

El turista se acercará a cualquiera de las agencias autorizadas a pagar su reserva, donde un agente lo atenderá y entregará sus pasajes. El agente contará también con un aplicativo web por el medio del cual fijará las posiciones de reserva como pagadas e irremovibles. En caso la reserva permanezca sin ser pagada por 2 días será eliminada del sistema, liberando el código para su reutilización.

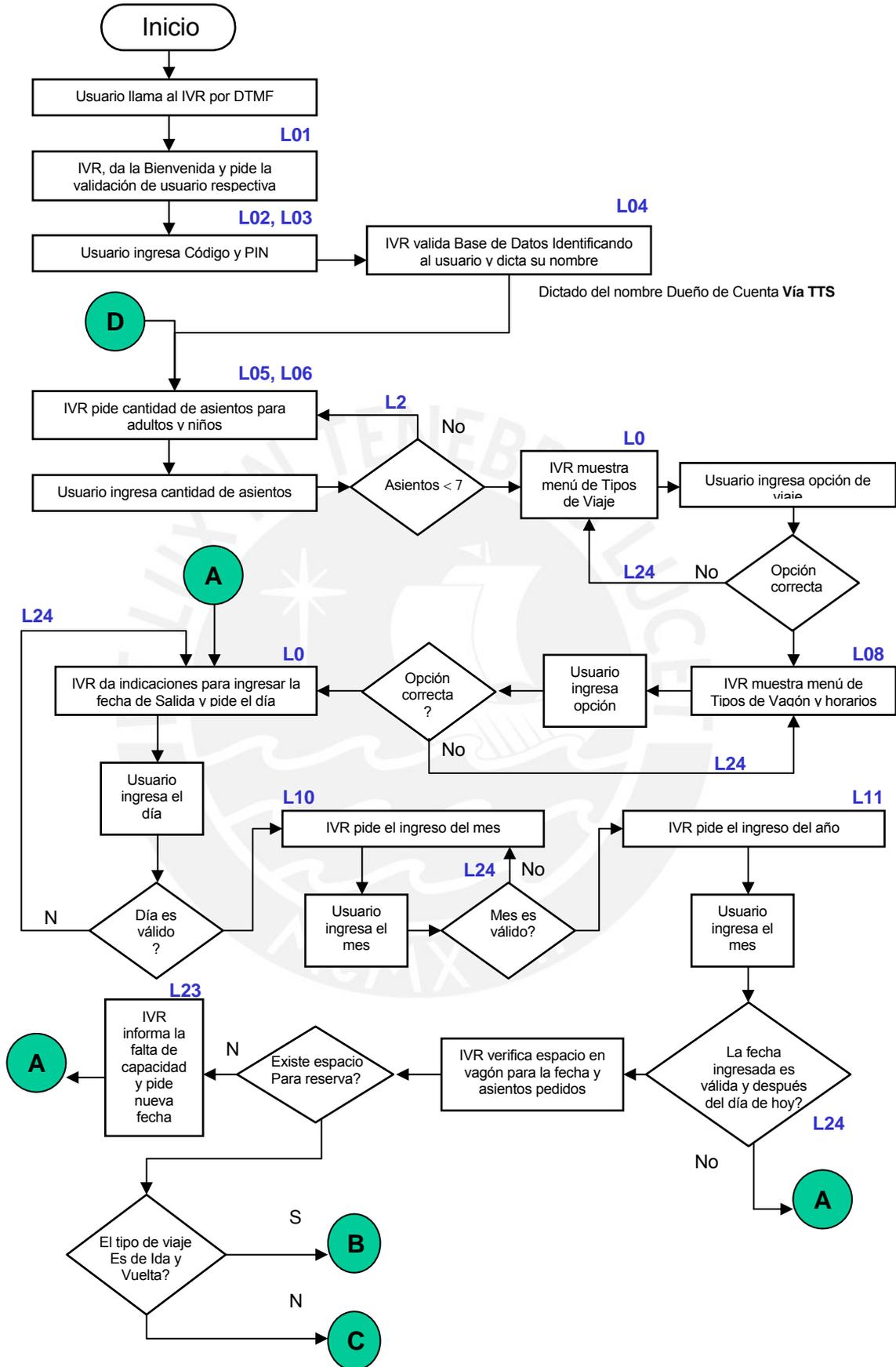
3.1.2. Diagramas de Flujo de la Solución

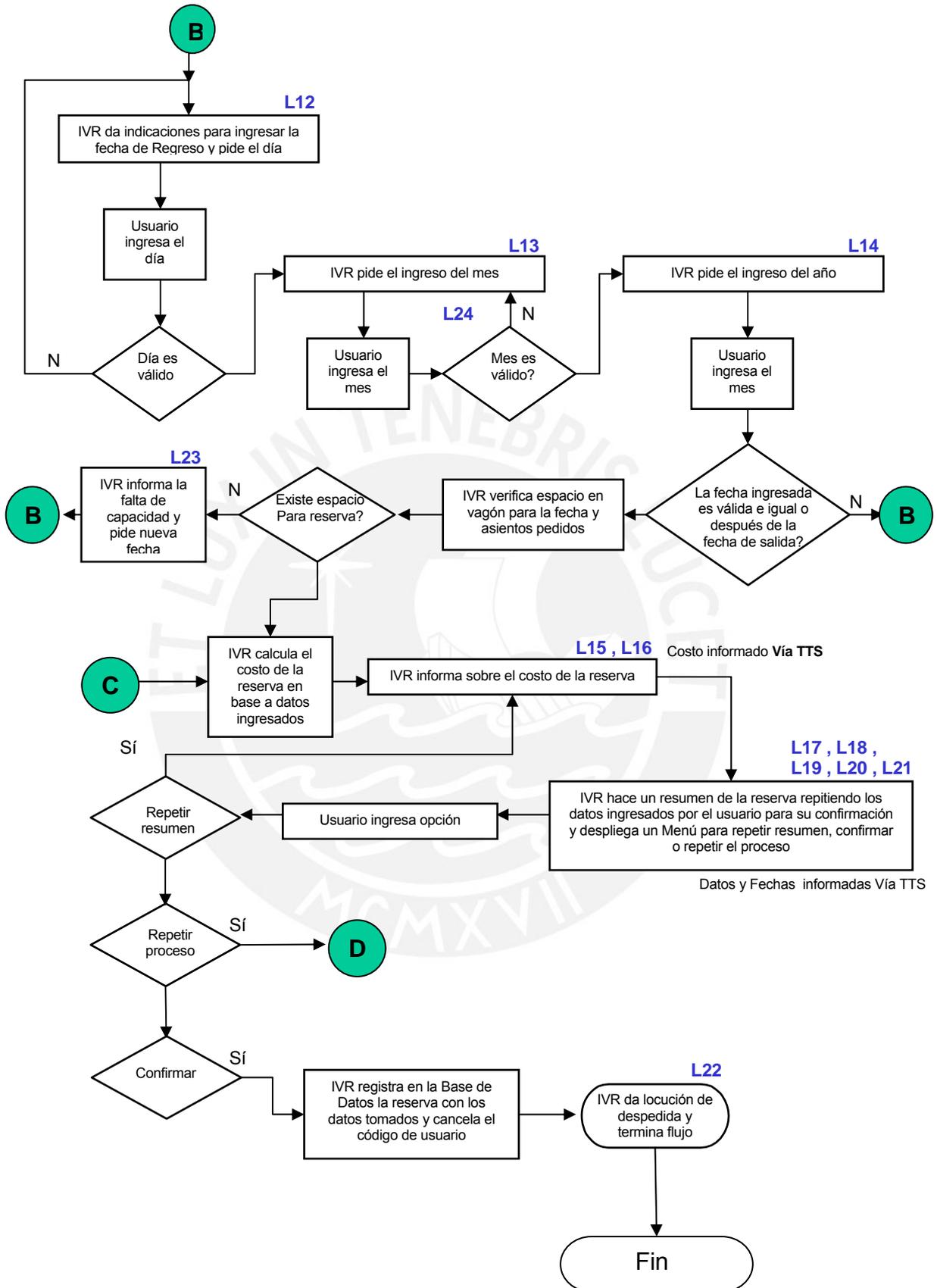
A continuación se presentarán los flujogramas o diagramas de flujo diseñados para el sistema IVR prototipo:

3.1.2.1 Aplicación Telefónica

Tras presentar el diagrama de flujo de la aplicación telefónica, se presentará la lista de locuciones grabadas y su posición será indicada en los diferentes pasos del diagrama de flujo.

**FIGURA 3.1. DIAGRAMA DE FLUJO DE LA SOLUCIÓN –
APLICACIÓN TELEFÓNICA**



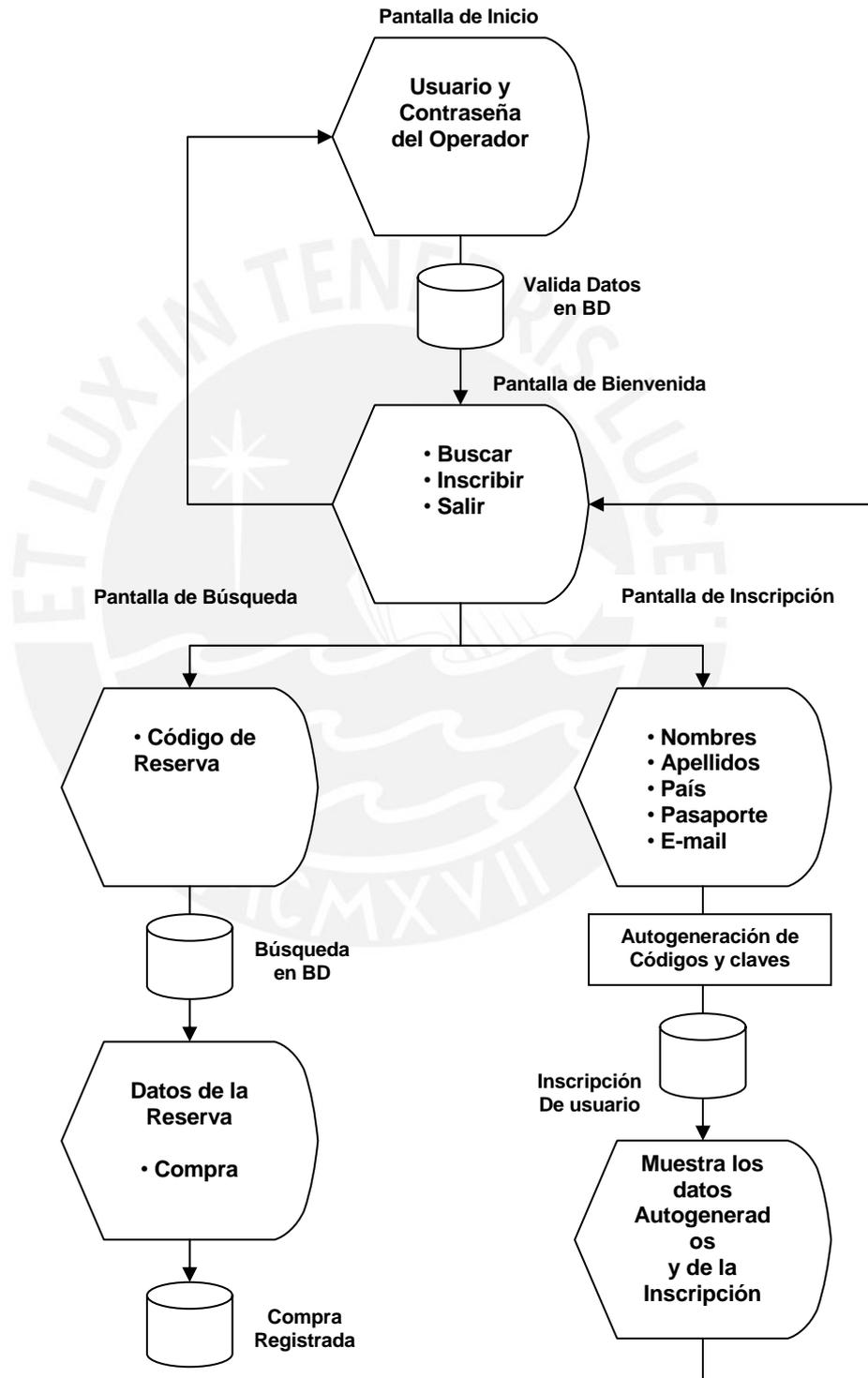


**TABLA 3.1. LOCUCIONES GRABADAS PARA LA APLICACIÓN TELEFÓNICA
IVR**

Código	Mensaje de la Locución
L01	"Bienvenido al Sistema de Reservas del Ferrocarril Cuzco Machu Picchu"
L02	"Por favor Ingrese cu código de usuario, al finalizar presione numeral (#)"
L03	"Ingrese su clave secreta"
L04	"Cuenta perteneciente a..."
L05	"Asientos a reservar: Contará con un total de 6 asientos para adultos y niños. Ingrese la cantidad de adultos"
L06	"Ingrese la cantidad de niños"
L07	"Tipo de Viaje: Para un viaje sólo de ida ONE WAY marque 1, para un viaje de ida y vuelta ROUND TRIP marque 2"
L08	"Tipo de Vagón: Para Backpacker de las 6.15 am marque 1. Para Vistadome de las 6 am marque 2. Para Vistadome de las 7 am marque 3. Para Hiram Bighman de las 9 marque 4"
L09	"Fecha de salida Cuzco - Machu Picchu: Al finalizar presione numeral (#)Ingrese el día de salida"
L10	Ingrese el mes de salida
L11	"Ingrese el año de salida, use 4 dígitos"
L12	"Fecha de regreso Machu Picchu – Cuzco : Al finalizar presione numeral (#)Ingrese el día de regreso"
L13	"Ingrese el mes de regreso"
L14	"Ingrese el año de regreso, use al menos 2 dígitos"
L15	"El costo de su reserva asciende a"
L16	"...Dólares"
L17	Su Reserva es la siguiente: Total de asientos reservados..."
L18a	"Vagón: Backpacker de las 6.15 am"
L18b	"Vagón: Vistadome de las 6.00 am"
L18c	"Vagón: Vistadome de las 7.00 am"
L18d	"Vagón: Hiram Bingham de las 9.00 am"
L19	"Saldrá de la ciudad del Cuzco el ..."
L20	"Y volverá a la misma el ..."
L21	"Para Confirmar marque Asterisco (*) Para Repetir el resumen de reserva marque 1. Si encontró un error marque 2 para repetir el proceso."
L22	"Su reserva ha sido realizada. Acérquese a cualquiera de nuestras oficinas con su código de reserva para cancelar sus pasajes. Muchas gracias por usar nuestro servicio. Hasta pronto."
L23	"Actualmente no contamos con espacio suficiente en la fecha elegida, intente una nueva fecha"
L24	"Su ingreso no fue válido, inténtelo de nuevo"

3.1.2.2 Aplicación Web

FIGURA 3.2. DIAGRAMA DE LA SOLUCION – APLICACIÓN WEB



3.1.3. Modelo de Entidad Relación

El modelo de Datos fue definido según el siguiente gráfico:

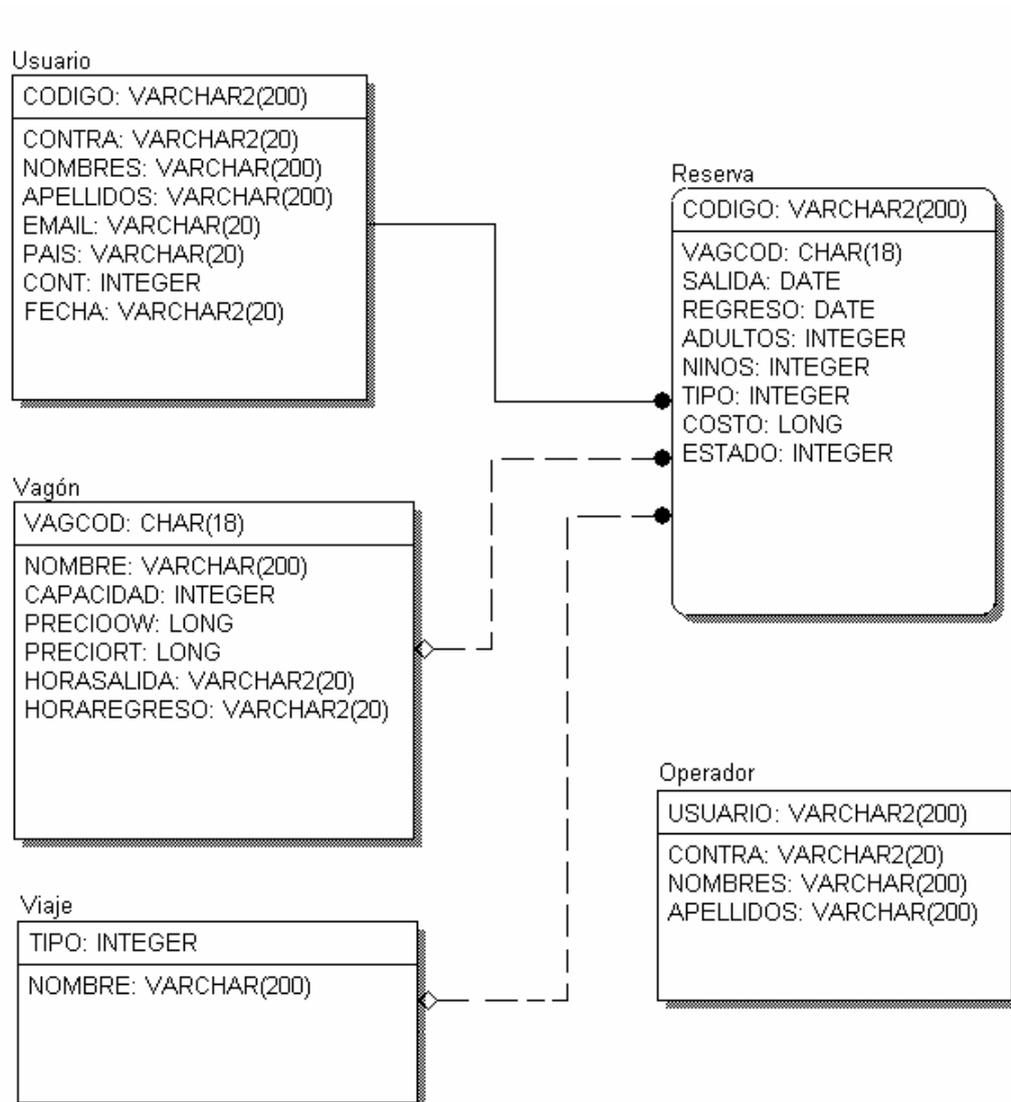


FIGURA 3.3. DIAGRAMA DE MODELO ENTIDAD RELACIÓN

3.1.3.1 Diccionario de Datos

A continuación se presentará la descripción de cada uno de los datos de las tablas del modelo antes mostrado:

Tabla Usuario:

TABLA 3.2. TABLA USUARIO

Nombre	Tipo de Dato	Descripción
CODIGO	VARCHAR2	Llave primaria. Es el código identificador del usuario en el sistema telefónico IVR.
CONTRA	VARCHAR2	Contraseña correspondiente al código de usuario y que valida su entrada al sistema.
NOMBRES	VARCHAR2	Nombres del Usuario.
APELLIDOS	VARCHAR2	Apellidos del Usuario.
EMAIL	VARCHAR2	E-mail válido del usuario.
PASAPORTE	VARCHAR2	Número de Pasaporte
CONT	INTEGER	Contador de Acción de usuario: 0 = Usuario sin reserva. 1 = Usuario con reserva hecha. 2 = Reserva comprada.
FECHA	VARCHAR2	Fecha de inscripción en el sistema IVR.

Tabla Vagón:

TABLA 3.3. TABLA VAGÓN

Nombre	Tipo de Dato	Descripción
VAGCOD	CHAR	Llave primaria. Identificador del tipo o clase de vagón.
NOMBRE	VARCHAR2	Nombres del Vagón.
CAPACIDAD	INTEGER	Capacidad de asientos del vagón.
PRECIOOW	LONG	Precio Sólo de Ida (One Way) por asiento en el vagón.
PRECIORT	LONG	Precio de Ida y Vuelta (Round Trip) por asiento en el vagón.
HORASALIDA	VARCHAR2	Hora de Salida del vagón de la ciudad del Cuzco a Machu Picchu
HORAREGRESO	VARCHAR2	Hora de Regreso del vagón de la ciudad del Machu Picchu a Cuzco

Tabla Viaje:

TABLA 3.4. TABLA VIAJE

Nombre	Tipo de Dato	Descripción
TIPO	INTEGER	Llave primaria. Identificador del tipo o viaje: = 1 : One Way = 2: Round Trip
NOMBRE	VARCHAR	Nombres del tipo de viaje.

Tabla Operador:

TABLA 3.5. TABLA OPERADOR

Nombre	Tipo de Dato	Descripción
USUARIO	VARCHAR2	Llave primaria. Identificador del operador que utilizará el sistema web de inscripción y administración del sistema IVR.
CONTRA	VARCHAR2	Contraseña que valida identidad del operador
NOMBRES	VARCHAR2	Nombres del Operador
APELLIDOS	VARCHAR2	Apellidos del Operador

Tabla Reserva:

TABLA 3.6. TABLA RESERVA

Nombre	Tipo de Dato	Descripción
CODIGO	VARCHAR2	Llave primaria. Relación Fuerte. Referencia al Código de usuario con lo que vincula a la Tabla Reserva como hija de la tabla Usuario.
VAGCOD	CHAR	Relación Débil. Referencia al Código del tipo de vagón en el cual se hará la reserva.
SALIDA	DATE	Fecha de salida de la ciudad de Cuzco a Machu Picchu.
REGRESO	DATE	Fecha de Regreso de Machu Picchu a la ciudad del Cuzco.
ADULTOS	INTEGER	Cantidad de adultos en la reserva
NINOS	INTEGER	Cantidad de niños en la reserva.
TIPO	INTEGER	Relación débil. Referencia a la llave primario de la tabla Viaje, indicándose el tipo de reserva.
COSTO	LONG	Costo calculado de la reserva.

3.1.3.2 Relación de Procedimientos

**TABLA 3.7. TABLA DE PROCEDIMIENTOS DEL PAQUETE
PKG_RESERVA**

PKG_RESERVA	
Nombre:	T_VERIFICA_CONTRASENA
Descripción:	Verifica, recibiendo un código de usuario y una contraseña del servicio telefónico IVR, la existencia del usuario y su capacidad de hacer reservas. Regresa la cuenta de usuarios con estas características, la cual puede ser cero o uno.
Nombre:	T_OBTEN_CAPACIDAD_1
Descripción:	Obtiene la capacidad reservada de asientos, retornando la suma de las variables ADULTOS y NIÑOS en la fecha de salida y en un determinado vagón.
Nombre:	T_OBTEN_CAPACIDAD_2
Descripción:	Obtiene la capacidad reservada de asientos, retornando la suma de las variables ADULTOS y NIÑOS en la fecha de regreso y en un determinado vagón.
Nombre:	T_OBTEN_NOMBRE
Descripción:	Regresa los nombres y apellidos del Usuario cuyo código identificador es dado con el fin de armar la cadena a ser pronunciada por el servidor TTS
Nombre:	T_OBTEN_CAP_VAGON
Descripción:	Dado un determinado código de vagón, retornará la capacidad total del Vagón con fines de proveer datos a la lógica del negocio para el cálculo de espacio libre.

Nombre:	T_OBTEN_COSTO
Descripción: Regresará los costos por asiento de un vagón cuyo código ha sido dado. Tanto el costo de ida (PRECIOOW) como de ida y vuelta (PRECIORT) son devueltos para que el servidor dado el caso, elija el adecuado a usar.	
Nombre:	T_RESERVA
Descripción: Con todos los datos obtenidos, hará la inscripción en la tabla RESERVA del requerimiento del usuario para luego proceder a bloquear el código de usuario así éste no podrá volver a usarse.	

**TABLA 3.8. TABLA DE PROCEDIMIENTOS DEL PAQUETE
PKG_TRABAJO**

PKG_TRABAJO	
Nombre:	T_ELIMINAR
Descripción: JOB que se ejecutará diariamente borrando aquellos usuarios que no hayan comprado su reserva hasta dos días después de haberla hecho. El código será liberado.	

3.1.4. Estándares del Sistema

El sistema IVR cuenta con los siguientes estándares:

- Los códigos de usuario serán números autogenerados y consecutivos de 6 dígitos empezando por el 100000.
- Las contraseñas serán números aleatorios de cuatro dígitos.
- El código de usuario será bloqueado tras la inscripción de la reserva, por lo que éste no podrá ser usado nuevamente.
- El plazo para la compra de la reserva hecha telefónicamente es de dos días a partir del día de reserva. Luego de este plazo la reserva será anulada y el código de reserva liberado para su uso.

3.1.5. Diseño de la Aplicación IVR

3.1.5.1 Aplicación Telefónica

La arquitectura en general funciona de la siguiente forma:

- El servidor AGI escuchará requerimientos por parte del Servidor Telefónico por el puerto 4573
- El servidor de Base de Datos tomará requerimientos a través del puerto 1521.
- El software TTS ubicado en el servidor PBX oír requerimientos por el puerto 1314

Para un mejor entendimiento del diseño de la aplicación telefónica, ésta ha sido dividida en 4 procesos: Validación, requerimiento de datos, Informe y Registro.

3.1.5.1.1 Validación

Esta primera etapa consiste en la validación del usuario en el sistema. Para esto, tras dar la bienvenida, el sistema pedirá un código de usuario y una contraseña. Con estos datos almacenados en el servidor de requerimientos (FASTAGI) se hará un llamado al método ValidaEntrada que invoca al procedimiento T_VERIFICA_CONTRASENA ubicado en el paquete almacenado PKG_RESERVA el cual buscará en la tabla USUARIO y contará los registros que contenga el usuario y la contraseña ingresada, y que a su vez se encuentre hábil de ser usado para generar una reserva. De ser positiva la búsqueda cuya respuesta sólo podrá tener los valores de “0” o “1” (No encontrado y encontrado respectivamente), este valor regresará al servidor el cual permitirá romper el bucle que se activará siempre tras un valor igual a “0” e invocará al método “getNombre” el cual hará una obtención de los nombres y apellidos del usuario mediante el procedimiento almacenado T_OBTEN_NOMBRE. Los datos regresarán al servidor FASTAGI el cual armará una cadena de formato especial con el nombre completo del usuario y con ésta, hará un requerimiento al software Festival TTS de pronunciación de esta cadena, confirmando al usuario la validez de su entrada y su identificación.

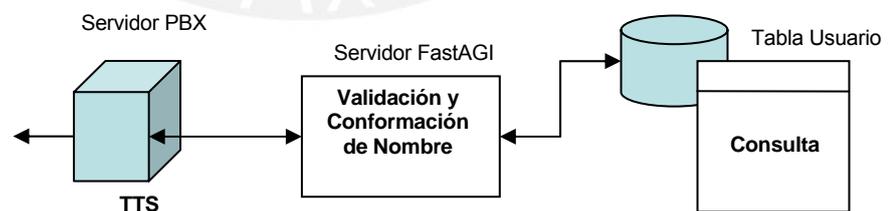


FIGURA 3.3. ETAPA DE VALIDACIÓN

3.1.5.1.2 Requerimiento de Datos

En la primera etapa de requerimiento de datos el sistema reproducirá locuciones y esperará entradas para los campos “número de asientos a reservar”, “tipo de viaje” y “tipo da vagón” haciendo los requerimientos al usuario por medio de las locuciones L05, L06, L07 y L08. Estos datos serán validados según los datos de la Tabla 3.2, reproduciendo una locución de error en caso se marque una opción no esperada y repitiendo el proceso de pedido del campo erróneo.

TABLA 3.9 VARIABLES ESPERADAS EN PRIMERA ETAPA DE REQUERIMIENTO DE DATOS

Campo	Variable esperada
Número de asientos a reservar	Número entre 1 y 6
Tipo de Viaje	Opciones de Menú: = 1 Sólo de ida: ONE WAY = 2 Ida y vuelta: ROUND TRIP
Tipo de Vagón	Opciones de Menú: = 1 Backpacker de las 6.15 am = 2 Vistadome de las 6.00 am = 3 Vistadome de las 7.00 am = 4 Hiram Bighman de las 9.00am

Cabe resaltar que todas los ingresos del usuario son tomados como variables del tipo “String”, para luego ser transformadas al tipo numérico y así hacer la validación ya antes explicada.

En la segunda parte de esta etapa se leerán la o las fechas – según el tipo de viaje – ingresadas por el usuario para la reserva de su viaje y se validará la existencia de espacio para la reserva en la base de datos. Para esto, se utilizarán las variables recogidas en la primera etapa de requerimiento de datos y hacer una consulta en la base de datos sumando la cantidad de personas con reservas en la fecha dada y en el vagón registrado.

Previamente el servidor ha verificado la validez de la fecha y que esta sea posterior al día en el que se está haciendo la llamada de reserva. En el caso de una segunda fecha necesaria para el tipo de viaje “2”, se verificará que la fecha de regreso ingresada no sólo sea posterior al día en el que se hace la reserva sino también posterior a la primera fecha y luego se procede a la verificación de capacidad.

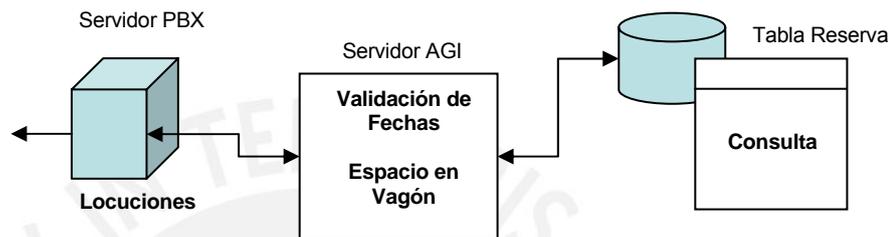


FIGURA 3.3. ETAPA DE REQUERIMIENTO DE DATOS

Finalmente el servidor FASTAGI obtendrá la capacidad total del vagón y por medio de una operación matemática verificará si hay espacio suficiente para la reserva.. Esto se hace mediante los métodos FechaSalida(), FechaLlegada(), ObtenerCapacidadVagon() y evaluacion() que invocan al los procedimientos T_OBTEN_CAPACIDAD_LIBRE y T_OBTEN_CAP_VAGON del paquete PKG_RESERVA usando las tablas RESERVA y VAGON

De ser positiva la respuesta y si es necesario el ingreso de una nueva fecha, este procesos empezará nuevamente; de lo contrario el sistema informará la falta de capacidad pidiendo el ingreso de una nueva fecha de viaje.

3.1.5.1.3 Informe

Una vez que todos los campos son ingresados y verificados como válidos, el sistema procederá a informar el costo al cual asciende la reserva. Para esto obtendrá de la base de datos el costo por asiento

del tipo de vagón respectivo por medio del método `ObtenerPrecioVagon()` que invoca al procedimiento `T_OBTEN_COSTO` del paquete `PKG_RESERVA` usando la tabla `VAGON`.

Con esto el servidor `FASTAGI` hará el cálculo matemático y tras obtener el costo total, transformará la variable numérica a una cadena de caracteres y conformará la cadena especial con la que se ejecutará el software `TTS` para que pronuncie el costo y lo informe al usuario.

Tras la información del costo, el sistema hará un resumen total de la reserva buscando la confirmación del usuario. Para esto utilizará todas las variables que fueron ingresadas en el proceso y que fueron ya validadas y serán colocadas en un párrafo de resumen mezclando locuciones y pronunciación de las fechas y número de asientos vía `TTS`. Tras esto, el servidor dará las opciones de confirmación o repetición del proceso en caso de querer hacer algún cambio, lo que regresa al punto inicial de requerimiento de datos.

3.1.5.1.4 Registro

El proceso final de registro se da tras la confirmación de reserva y consiste básicamente en la inscripción en la Tabla `RESERVA` de todos los datos recogidos durante el proceso. Para esto se utilizará el método `setReserva` que invoca al procedimiento `T_RESERVA` del paquete `PKG_RESERVA`.

Finalmente, con la reserva ya realizada el código de usuario será bloqueado por lo que no se podrán realizar más reservas con éste.

3.1.5.2 Aplicación Web

La aplicación web fue diseñada bajo la arquitectura Modelo-Vista-Controlador utilizando el marco Apache Struts 1.2.2.

Todo requerimiento hecho empezará en el controlador de Struts, cuyo corazón son las clases `ActionServlets`, las cuales son mapeados a la extensión `*.do`. El controlador de servlets buscará la acción correcta a seguir tras ser invocada. Esta información es leída de los archivos de configuración Struts en instancias de la clase `ActionMapping`.

La clase `ActionMapping` contará también con clases asociadas llamadas `ActionForms`, que son `JavaBeans` que representan datos de ingreso desde un formulario HTML. Estos beans serán inicializados automáticamente en caso de ser necesarios y rellenos con parámetros. Adicionalmente, estos datos pueden ser validados.

Finalmente las acciones (`ActionServlets`) son invocados para una determinada URL y cuando es completada, el controlador redirecciona el requerimiento al componente "Vista", compuesto por la página web, representados por objetos `ActionForward` en el objeto `ActionMapping`.

Para la implementación del sistema web prototipo, se crearon `ActionServlets` y el archivo `config-struts` en el cual se almacenó el mapeo respectivo a acciones y `ActionForms` requeridos para el diseño.

3.1.6. Diccionario de Clases del Sistema

TABLA 3.10. CLASE BUSUARIO

Clase	Atributos	Métodos	Descripción
BUusuario	1. String codigo 2. String contrasena 3. String nombres 4. String apellidos 5. String pais 6. String email 7. String cont	1.getCodigo	Obtiene el código del usuario.
		2.setCodigo	Establece o cambia el código del usuario.
		3.getContrasena	Obtiene la contraseña del usuario.
		4.setContrasena	Establece o cambia la contraseña del usuario.
		5. getNombres	Obtiene los nombres del usuario.
		6. setNombres	Establece o cambia los nombres del usuario.
		7. getApellidos	Obtiene los nombres del usuario.
		8. setApellidos	Establece o cambia los apellidos del usuario.
		9. getPais	Obtiene el país de procedencia del usuario.
		10. setPais	Establece o cambia el país del usuario.

		11. getEmail	Obtiene el correo electrónico del usuario.
		12. setEmail	Establece o cambia el correo electrónico del usuario.
		13. getCont	Obtiene el estado de cuenta del usuario.
		14. set Cont	Cambia el estado de cuenta del usuario.

TABLA 3.11. CLASE BRESERVA

Clase	Atributos	Métodos	Descripción
BReserva	1. String codigo	1.getCodigo	Obtiene el código de la reserva.
	2. String tipo	2.setCodigo	Establece el código de la reserva.
	3. String vagon	3.getTipo	Obtiene el tipo de viaje.
	4. String fecha1	4.setTipo	Establece el tipo de viaje.
	5. String fecha2	5.getVagon	Obtiene el tipo de vagón.
	6. Int adultos	6.setVagon	Establece el tipo de vagón.
	7. Int ninos		

	8. Double costo	7. getFecha1	Obtiene la fecha de salida.			
		8. setFecha1	Establece la fecha de salida.			
			9. getFecha2	Obtiene la fecha de regreso.		
			10. setFecha2	Establece la fecha de regreso.		
			11. getAdultos	Obtiene la cantidad de adultos en la reserva.		
			12. setAdultos	Establece la cantidad de adultos en la reserva.		
			13. getNinos	Obtiene la cantidad de niños en la reserva.		
			14. setNinos	Establece la cantidad de niños en la reserva.		
			15. getCosto	Obtiene el costo de la reserva.		
			16. setCosto	Establece o cambia el costo de la reserva.		

TABLA 3.12. CLASE DUSUARIO

Clase	Métodos	Tipo	Descripción
DUSUARIO	1. getNombre	String	Obtiene el nombre y el apellido del dueño de la cuenta de la tabla "USUARIO" devolviendo el string para que sea pronunciado por el servidor TTS tras la invocación respectiva.
	2. validaEntrada	boolean	Valida la entrada de la cuenta de un usuario al sistema mediante la verificación de su Código de usuario y contraseña devolviendo "true" en caso positivo y "false" en caso negativo. Verificará también si el código de usuario se encuentra libre de realizar reservas.

TABLA 3.13. CLASE DVAGON

Clase	Métodos	Tipo	Descripción
DVAGÓN	1. getCapacidadvagon	Int	Obtiene la cantidad de asientos que tiene un respetivo vagón dado su código de vagón.
	2. getPreciovagon	double	Obtiene el precio de un vagón según su tipo “One way” o “Round trip” consultando la tabla “VAGON”.
	3. fechaSalida	Int	Obtiene la cantidad de asientos totales ocupados del vagón elegido en la fecha de salida indicada
	4. fechaRegreso	int	Obtiene la cantidad de asientos totales ocupados del vagón elegido en la fecha de regreso indicada.

TABLA 3.14. CLASE DRESERVA

Clase	Métodos	Tipo	Descripción
DRESERVA	1. Evaluacion00	boolean	Invocando a las funciones “getCapacidadvagon” y “fechasalida” de la clase DVagon , evaluará si es posible la inscripción de la reserva basándose en la capacidad libre del vagón en la fecha indicada y en el número de asientos a reservar, devolviendo “true” en caso positivo y “false” en caso negativo.
	2. Evaluacion01	boolean	Invocando a las funciones “getCapacidadvagon” y “fechallegada” ” de la clase DVagon, evaluará si es posible la inscripción de la reserva basándose en la capacidad libre del vagón en la fecha indicada y en el número de asientos a reservar, devolviendo “true” en caso positivo y “false” en caso negativo.
	3. SetReserva	void	Realiza la inscripción de la reserva con todos los datos obtenidos tras el ingreso de tonos del usuario, por medio del uso de la clase BRESERVA

3.1.7. Estimación de Tráfico

Se conoce de los datos presentados en el Capítulo 2 que un máximo de 50,000 turistas mensuales son los que viajan a Machu Picchu vía tren. A continuación se hará una estimación de tráfico para el sistema suponiendo que la empresa deseará hacer una migración al nuevo sistema IVR de reservas.

Con un flujo de 50,000 turistas, se estima un total de 1667 turistas por día entrando a la ciudad del Cuzco, se asumirá el escenario más difícil en el que existirá una total demanda.

Usando la fórmula de tráfico y usando una duración promedio de 3 minutos y medio:

$$\text{Tráfico} = [(\text{Número de Llamadas}) \times (\text{Duración por Llamada})] / 1 \text{ hora}$$

Obtenemos un total de 97.3 Erlangs, los cuales calculados usando la fórmula del Erlang B a un 1% de pérdida se obtienen 115 líneas o llamadas concurrentes.

En [QUI2006] se realizaron pruebas sobre un servidor de 1GB de memoria RAM que arrojaron resultados de hasta 199 llamadas concurrentes, por lo que un servidor con esta configuración podrá soportar el tráfico estimado.

3.2. Implementación de la Arquitectura

3.2.1. Servidor PBX

El servidor PBX se encuentra implementado en una PC de procesador Intel Pentium 4, 512MB RAM corriendo el sistema operativo Fedora Core 5 con el kernell 1.2.17

3.2.1.1 Configuración Asterisk

La versión más reciente de Asterisk podrá descargarse de [AST2007] y detalles de su compilación se encuentran en [MOL2006].

El archivo `extensions.conf`, el cual se encuentra en `/etc/asterisk` y contiene el plan de discado (`dialplan`) del contexto “sistema”, será configurado asignando un número al sistema IVR, el cuál será la extensión 601. Esta extensión hará referencia al servidor AGI y al nombre mapeado en **fastagi-mapping.properties**, archivo de propiedades que será explicado más adelante. Adicionalmente se implementaron 2 extensiones llamadas “usuario0” y “usuario1” en los archivos `iax.conf` y `sip.conf` respectivamente para pruebas iniciales en el mismo contexto.

Adicionalmente se creó la carpeta “tesis” en la ruta `/var/lib/asterisk/sounds` para almacenar las locuciones grabadas tanto en inglés como en español, siendo las rutas `/var/lib/asterisk/sounds/tesis/es` y `/var/lib/asterisk/sounds/tesis/en`.

TABLA 3.15. ARCHIVO EXTENSIONS.CONF

<pre> [sistema] exten => 1234,1,Dial(IAX2/usuario0) exten => 1234,2,Hangup() exten => 1235,1,Dial(SIP/usuario1) exten => 1235,2,Hangup() exten => 600,1,Answer exten => 600,2,SetMusicOnHold(default) exten => 600,5,Background(/tesis/en/00) exten => 1,1,AgI(agi://192.168.1.33/ivr_es.agi) exten => 1,2,AgI(agi://192.168.1.33/ivr_en.agi) </pre>

3.2.1.2 Configuración Festival

Una vez instalado Asterisk, se procederá a la instalación de Festival, software usado para TTS (ya explicado previamente). Para esto se utilizará el comando “yum install festival”

Una vez instalado, procedemos a instalar el paquete de Festival en español disponible en [FES2007], descargando el archivo de extensión .tar.gz y descomprimiéndolo.

Para programar el lenguaje a ser utilizado, alteraremos el archivo festival.scn , ubicado en /usr/share/festival, al cual le agregaremos el siguiente script para usar por defecto el lenguaje español en modo servidor:

TABLA 3.16. ARCHIVO FESTIVAL.SCM – LENGUAJE ESPAÑOL

```
(language_spanish)
(set! voice_default 'voice_pc_diphone)
```

Una vez hecho esto, agregaremos un script en el mismo archivo que permitirá que Asterisk interactúe directamente con Festival, el cual define la forma en que Asterisk pasará las cadenas de texto a ser reproducidas:

TABLA 3.17. ARCHIVO FESTIVAL.SCM – INTERACCION ASTERISK

```
;; Command for Asterisk begin
(define (tts_textasterisk string mode)
  "(tts_textasterisk STRING MODE)
  Apply tts to STRING. This function is specifically designed for
  use in server mode so a single function call may synthesize the string.
  This function name may be added to the server safe functions."
  (utt.send.wave.client (utt.wave.resample (utt.wave.rescale (utt.synth
```

```
(eval (list 'Utterance 'Text string))) 5) 8000)))  
;;; Command for Asterisk end
```

Finalmente se crea un usuario en el archivo `festival.conf` ubicado en `/etc/asterisk` en el cual indicaremos las características de la conexión entre Asterisk y Festival, indicando la ubicación del host Festival, en el caso de este proyecto encontrándose juntos: el puerto y el comando, entre otros:

TABLA 3.18. ARCHIVO /ETC/ASTERISK/FESTIVAL.CONF

```
[general]  
host=localhost  
port=1314  
usecache=yes  
cachedir=/var/cache/asterisk/festival/  
festivalcommand=(tts_textasterisk "%s" 'file')(quit)\n
```

3.2.2. Servidor de Requerimientos (AGI)

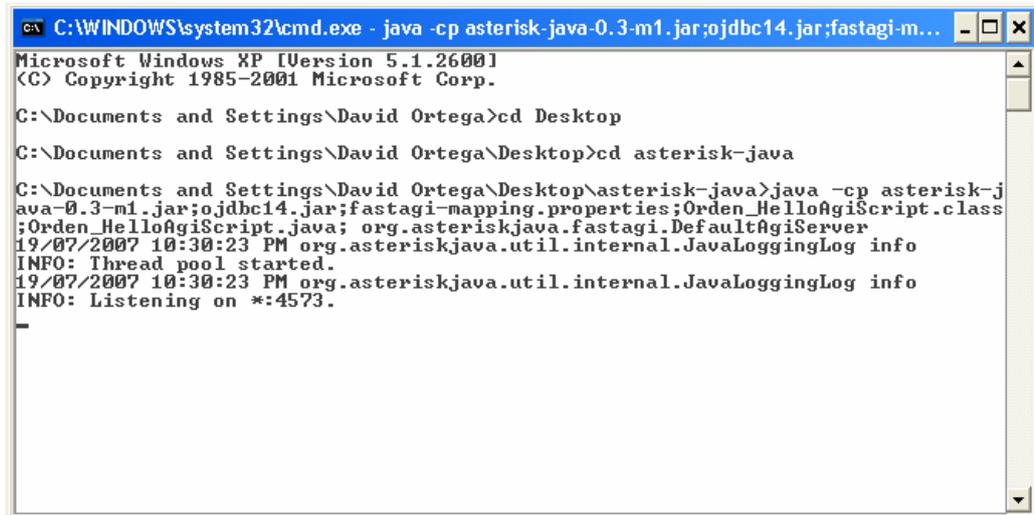
El servidor de requerimientos se encuentra implementado en una PC con procesador Intel Centrino Duo2, con 2 GB de memoria RAM usando una plataforma Windows XP.

Para esto debemos contar con el paquete Asterisk-Java en una ubicación del sistema de archivos, y el archivo **fastagi-mapping.properties**, el cual será el encargado de mapear el script realizado en Java correspondiente a la lógica del negocio del IVR a una extensión “.agi”, la cual será empleada para invocar el script remotamente, utilizando el socket ya explicado previamente.

TABLA 3.19. ARCHIVO FASTAGI-MAPPING.PROPERTIES

```
ivr.agi_es = IVRAgiScript  
ivr.agi_en = IVRAgienScript
```

El archivo antes mencionado referencia al código propio del IVR llamado IVRAgiScript.java y IVRAgienScript.java, el cual se colocará también en la misma ubicación del sistema de archivos que el archivo de mapeo, o en su defecto se colocará la ruta entera de la ubicación de IVRAgiScript.java en el archivo **fastagi-mapping.properties**.



```

C:\WINDOWS\system32\cmd.exe - java -cp asterisk-java-0.3-m1.jar;ojdbc14.jar;fastagi-m...
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\David Ortega>cd Desktop

C:\Documents and Settings\David Ortega\Desktop>cd asterisk-java

C:\Documents and Settings\David Ortega\Desktop\asterisk-java>java -cp asterisk-j
ava-0.3-m1.jar;ojdbc14.jar;fastagi-mapping.properties;Orden_HelloAgiScript.class
;Orden_HelloAgiScript.java; org.asteriskjava.fastagi.DefaultAgiServer
19/07/2007 10:30:23 PM org.asteriskjava.util.internal.JavaLoggingLog info
INFO: Thread pool started.
19/07/2007 10:30:23 PM org.asteriskjava.util.internal.JavaLoggingLog info
INFO: Listening on *:4573.
  
```

FIGURA 3.4. LOG DEL SERVIDOR DE REQUERIMIENTOS

Con esto realizado sólo queda levantar el servidor, ejecutando `org.asteriskjava.fastagi.DefaultAgiServer`, el cual levantará el hilo respectivo para recibir los requerimientos FASTAGI vía el puerto 4573.

3.2.3. Servidor de Base de Datos

El servidor de base de datos fue implementado en una PC de procesador Intel Pentium 4, 512MB RAM corriendo el sistema operativo Microsoft Windows XP con el Sistema Oracle 9i instalado.

La implementación del mismo se basó en la creación de la base de datos y la programación de los paquetes de procedimientos almacenados así como la implementación del *JOB* que funciona como un procedimiento de ejecución automática.

La implementación del JOB utiliza DBMS_JOB, el cual es un paquete Oracle PL/SQL el cual permite agendar un procedimiento que será ejecutado a una determinada hora, especificando también la periodicidad del mismo, en caso sea una operación.

El sistema usará el JOB “Eliminar” para borrar del sistema a los usuarios que no hayan cancelado el monto de la reserva, pasados los dos días de plazo para la compra.

3.2.4. Pantallas de la Aplicación Web Prototipo

3.2.4.1 Interfaz de Ingreso

La primera interfaz del sistema web prototipo es la interfaz de ingreso, la cual requerirá un usuario y una contraseña válida para poder ingresar al sistema

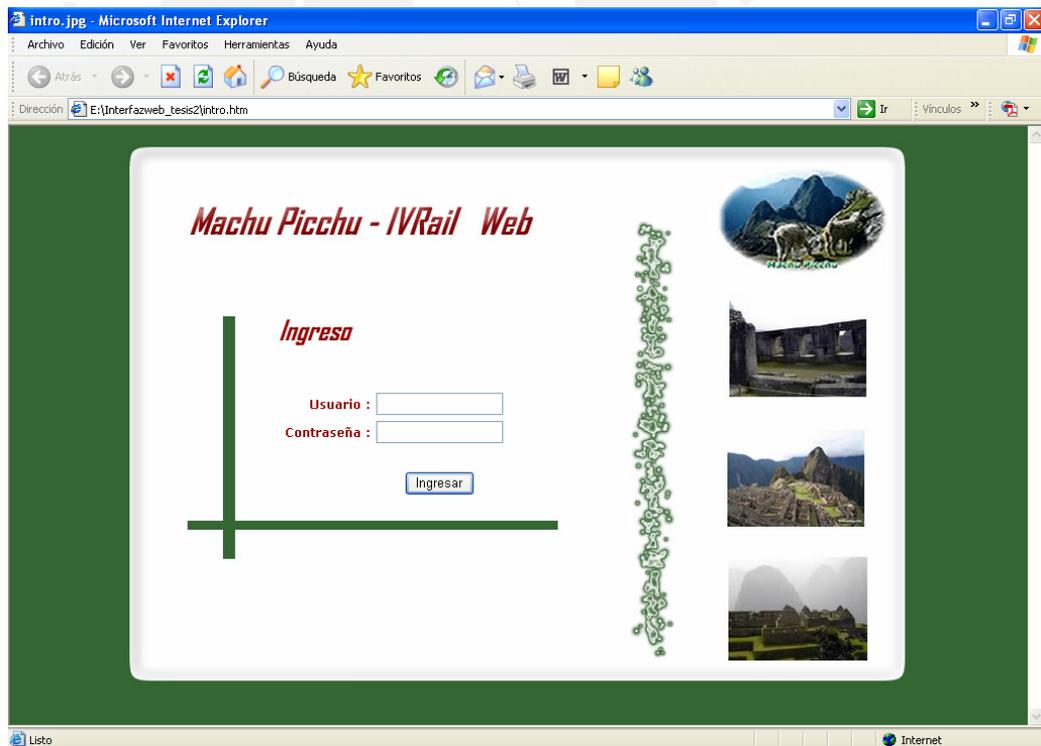


FIGURA 3.5. INTERFAZ DE INGRESO

Tras el ingreso por esta interfaz y la autenticación hecha, el sistema mostrará por defecto la interfaz de inscripción como interfaz de bienvenida y habrá obtenido de la base de datos todos los datos del operador del sistema web.

3.2.4.2 Interfaz de Inscripción

La interfaz de inscripción nos permitirá generar códigos de usuario y contraseñas para el ingreso validado en el sistema telefónico. Este consta de un formulario en el cual se pedirán todos los datos necesarios para la inscripción. El sistema mandará un correo electrónico automáticamente con los datos y presentará los códigos generados para hacer posible su impresión de ser necesarios.

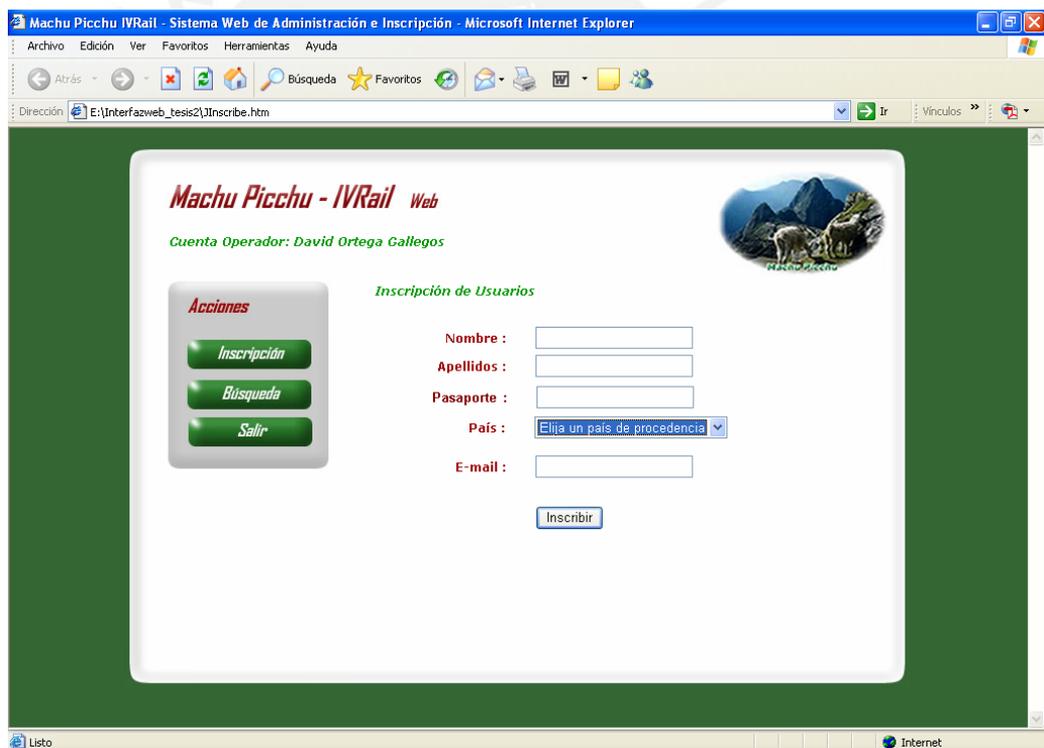


FIGURA 3.6. INTERFAZ DE INSCRIPCIÓN

Podemos observar el menú de acciones el cuál permanecerá en todo el sistema con las opciones: “Inscripción”, “Búsqueda” y “Salir”. La primera hará referencia a esta misma interfaz, la segunda a la interfaz de búsqueda a ser descrita a continuación y la tercera para abandonar la sesión y salir del sistema.

3.2.4.3 Interfaz de Búsquedas

La interfaz de búsqueda de usuarios se hará por tres campos: Nombres, Apellidos, y código de usuario, siendo este último el identificador principal. Para la compra de la reserva es necesario conocer el código de usuario, caso contrario el usuario deberá acercarse e identificarse con su pasaporte para poder comprar la reserva.

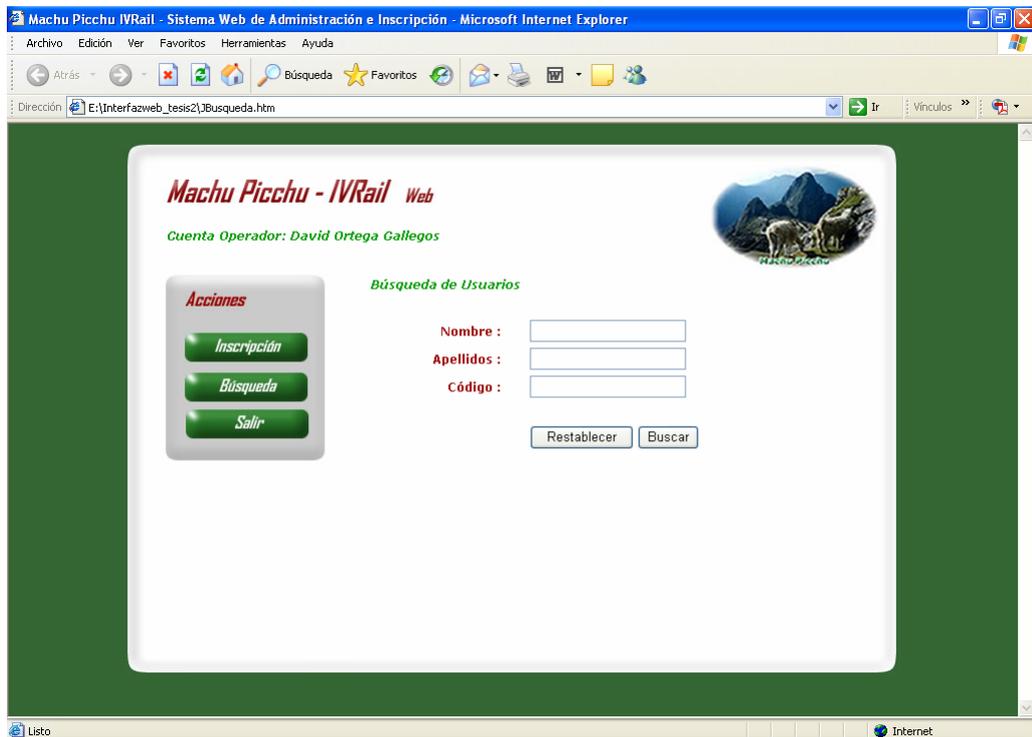


FIGURA 3.6. INTERFAZ DE INSCRIPCIÓN

Tras la búsqueda, se obtendrá un cuadro similar al de la figura 3.7, mostrándose todos los resultados obtenidos según el criterio de búsqueda utilizado.

La interfaz de búsqueda mostrará botones como los vistos a la derecha e izquierda de la tabla de resultados, usados respectivamente para ingresar a las interfaces de datos del usuario y la reserva.



FIGURA 3.7. INTERFAZ DE RESULTADO DE BÚSQUEDA

3.2.4.4 Interfaces de Datos

El sistema cuenta con interfaces de despliegue de datos: La interfaz de datos del usuario y la interfaz de datos de la reserva.

La interfaz de datos del usuario muestra los datos ingresados por el mismo sistema en el momento de la inscripción e informará también sobre el estado de la cuenta, el cual puede tomar tres valores:

- Libre: Este estado indica que el usuario no ha realizado ninguna reserva en el sistema y que su código está disponible para ser usado.
- Reserva Realizada: Este estado significa que el usuario tiene una reserva hecha y que tendrá dos días para comprar la reserva.

- Boleto Comprado: Indica que el o los boletos con las reservas ya fueron comprados.

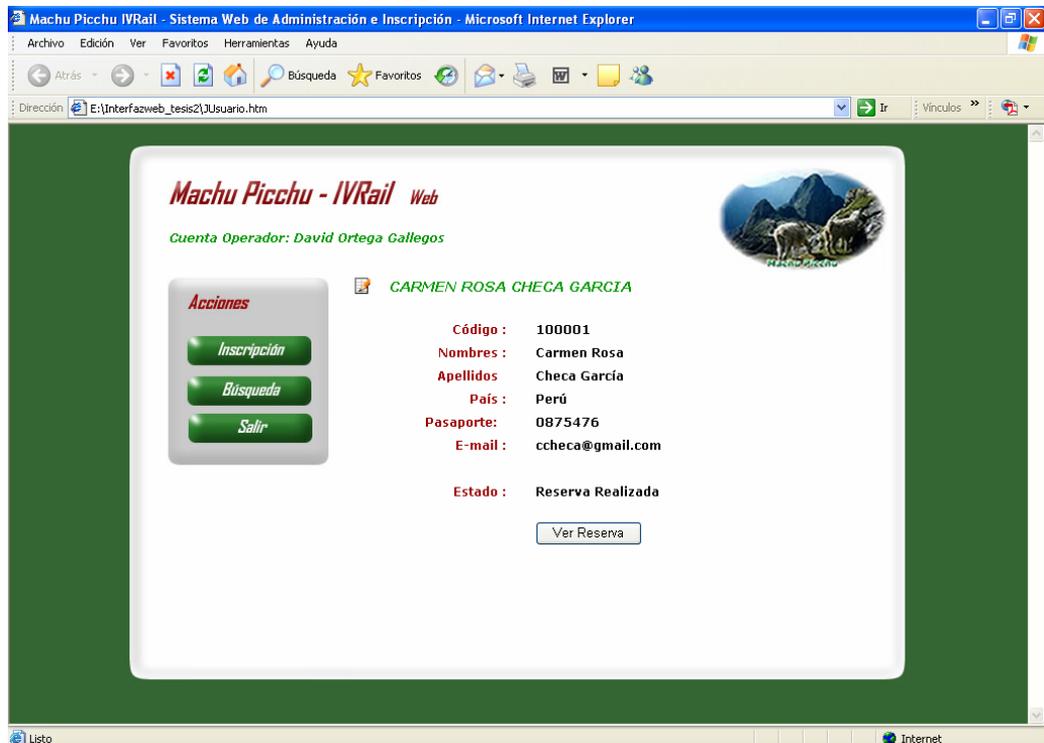


FIGURA 3.8. INTERFAZ DE DATOS DEL USUARIO

Los estados “Reserva Realizada” y “Reserva comprada” constarán a su vez de un botón que permitirá el ingreso a la interfaz de datos de la Reserva.

La interfaz de Datos de la Reserva mostrará todos los datos almacenados en la tabla Reserva de la base de datos y podrá estar eventualmente ligado a un proceso de impresión de tickets. Esta interfaz nos mostrará también el estado y un botón en caso el estado sea “Reserva Realizada” para poder comprar la reserva, como se puede observar en la figura 3.9



FIGURA 3.9. INTERFAZ DE DATOS DE LA RESERVA

Capítulo 4

Pruebas de Desempeño

En este capítulo se describirá el proceso de pruebas realizado así como los elementos usados para los fines de evaluación y se mostrarán los resultados obtenidos para el Sistema IVR.

4.1. Introducción

Con el fin de evaluar el sistema IVR y definir su desempeño en la arquitectura implementada se utilizará el software SIPP. SIPP es un sistema de pruebas de código abierto desarrollado por Hewlett Packard (HP). Se aprovecharán las capacidades de generación de tráfico de Sipp para emular a muchos usuarios SIP llamando al sistema IVR y realizando requerimientos a la base de datos.

Sipp cuenta con configuraciones por defecto llamadas escenarios éstas son:

- UAC - User Agent Client
- UAS – User Agent Server

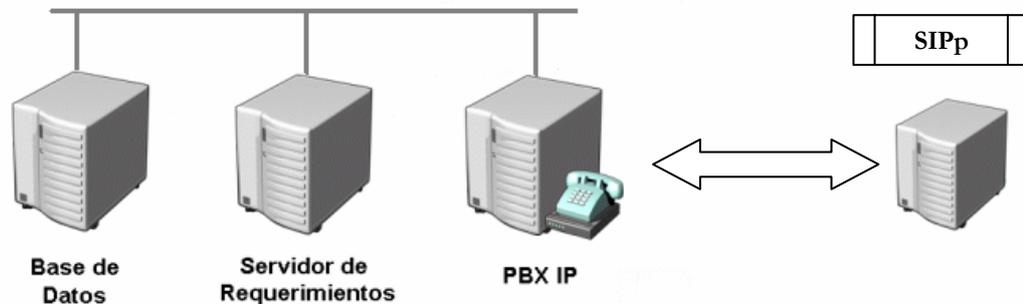


FIGURA 4.1. ARQUITECTURA DE PRUEBAS

Si bien pueden usarse estas configuraciones, SIPp también da la capacidad de crear escenarios personalizados en archivos XML, para la prueba específica de aplicaciones que requieran intercambiar datos más allá de un simple intercambio RTP, es decir una simulación de llamadas.

En las pruebas realizadas se configuró la arquitectura para ser cargada. Las llamadas harán un requerimiento automático a base de datos, para luego ejecutar una sentencia TTS y mantener la llamada mediante la reproducción de un archivo de audio de 80 minutos de duración, con el fin de poder obtener la cantidad de llamadas que el servidor PBX puede soportar en un determinado instante.

A su vez, se analizará el uso de CPU y memoria total tanto con el uso del servidor TTS como sin él con el fin de realizar comparaciones que permitan emitir recomendaciones adecuadas. Estos monitoreos fueron realizados mediante el uso del software *Cacti*.

4.2. Configuración

Para la realización de las pruebas se configuró el sistema de la siguiente forma:

4.2.1 Servidor PBX

El servidor PBX fue configurado creando un usuario [SIPP] en el archivo `/etc/asterisk/sip.conf`, por el cual el sistema antes descrito generará las llamadas a la PBX, produciendo tráfico.

La configuración necesaria para esto es la que se muestra en la siguiente tabla.

TABLA 4.1. CONFIGURACIÓN DE SIP.CONF

<pre>[sipp] Type=friend Host=dynamic context=sistema user=sipp disallow=all allow=ulaw language=es</pre>
--

Así también el Asterisk fue configurado para su interacción con *Cacti*, tal como se especifica en [DIA2007].

4.2.2 Servidor de Requerimientos

Para el servidor de requerimientos, se generó un script de pruebas y se puso en funcionamiento el servidor AGI manipulado la variable “PoolSize” que regula el número máximo de requerimientos (hilos) que pueden generarse en el servidor e iniciándola en 500.

Para la manipulación de ésta variable, se generó el archivo `fastagi.properties`, y se colocó en el *Classpath* al momento de la ejecución.

4.3.1 Número de Llamadas Concurrentes

El número de llamadas concurrentes máximo obtenidos fue 167. La figura 4.2 nos muestra el incremento de llamadas según la tasa de 4 llamadas por minuto. Cuando la llamada 168 ingresa el servidor no puede continuar con el servicio dado que es incapaz de alzar un canal SIP, perdiendo las llamadas siguientes. La Figura 4.3 nos muestra el log del SIPP.

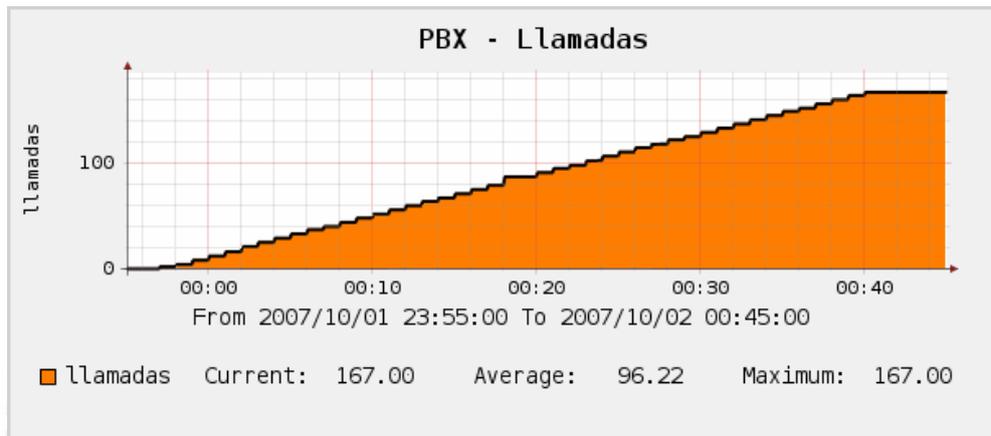


FIGURA 4.2. LLAMADAS CONCURRENTES – PRUEBA 1

Start SIPP shell		
Start Time	: 2007-10-01 23:59:30	
Last Reset Time	: 2007-10-02 00:47:14	
Current Time	: 2007-10-02 00:47:14	
Counter Name	Periodic value	Cumulative value
Elapsed Time	: 00:00:00:442	: 00:47:44:817
Call Rate	: 0.000 cps	: 0.064 cps
Incoming call created	: 0	: 0
OutGoing call created	: 0	: 184
Total Call created		: 184
Current Call	: 167	
Successful call	: 0	: 0
Failed call	: 0	: 17
Response Time 1	: 00:00:00:000	: 00:00:00:163
Call Length	: 00:26:01:081	: 00:23:36:854
--- Test Terminated ---		
2007-10-02 00:47:14: Aborting call with Call-Id '99-2780@192.168.1.33'.		
sipp: There were more errors, enable -trace_err to log them.		
C:\Program Files\SIPP>		

FIGURA 4.3. LOG DEL SIPP

La Figura 4.4 nos muestra las llamadas concurrentes en todo el periodo de pruebas: podemos observar que en ambos casos el número de llamadas de concurrentes es el mismo, dado que SIPp hace uso de G.711 que no hace uso de recursos de CPU, [QUI2006] por lo que el número de llamadas depende mayormente de la memoria del servidor.

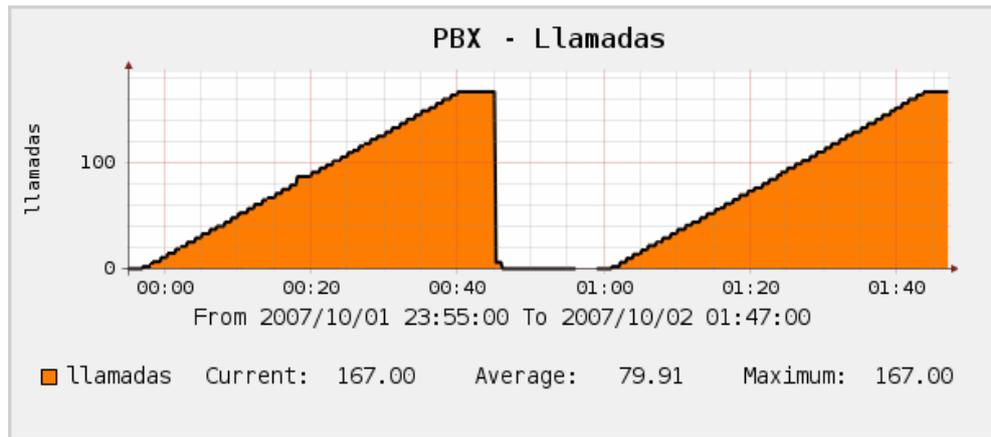


FIGURA 4.4. LLAMADAS CONCURRENTES – PERIODO TOTAL

4.3.2 Uso de CPU

Durante las pruebas se midió el uso de CPU para determinar la influencia en el uso del servidor TTS Festival. Para esto se presentan los gráficos de uso de CPU obtenidos de uso de CPU.

Los resultados, tal cual como son mostrados en las figuras 4.5 y 4.6, muestran que el uso del servidor Festival TTS incrementó en un 20% considerando sólo el uso del servicio TTS una sólo vez. Los gráficos muestran momentos similares a las 00:45 usando el servidor Festival y 1:42 sin el uso del mismo, lo que nos permite una comparación

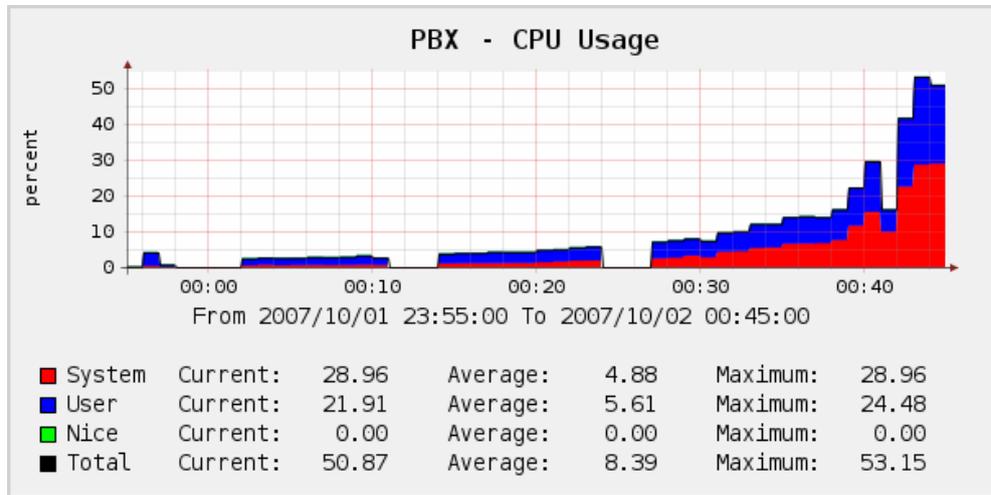


FIGURA 4.5. USO DE CPU CON SERVIDOR TTS

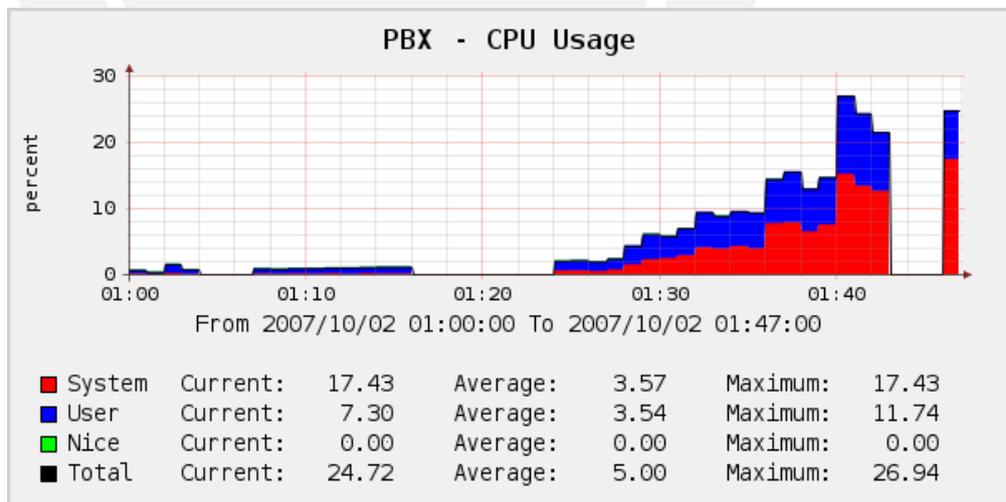


FIGURA 4.6. USO DE CPU SIN SERVIDOR TTS

4.3.3 Memoria

En la figura 4.7 se muestra el uso de memoria durante el periodo completo de pruebas.

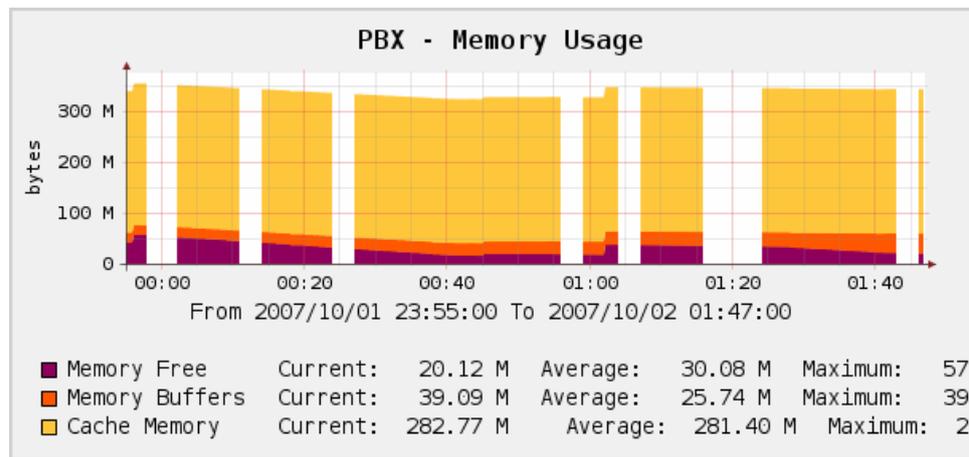


FIGURA 4.7. USO DE MEMORIA

Comparando esta gráfica tomando en cuenta los tiempos de ocurrencia de llamadas de la figura 4.4, podemos observar que las llamadas se inician alrededor de la hora 00:00 y hasta que finalicen a las 00:40 y que en este tiempo la memoria libre fue decreciendo linealmente debido al establecimiento paulatino de las llamadas. Desde las 00:40 hasta las 1:00 existe un comportamiento constante dado que el servidor se encontraba en un periodo de estabilización para luego empezar de nuevo un comportamiento similar en la segunda prueba.

Conclusiones, Recomendaciones y Trabajos Futuros

En este capítulo se presentarán las recomendaciones, conclusiones del proyecto de tesis y los trabajos futuros que pueden estar derivados de éste.

5.1. Recomendaciones

- Protocolo: Se recomienda el uso del protocolo SIP, como fue utilizado en el siguiente proyecto; sin embargo también es viable la utilización del protocolo IAX2 que consume menor ancho de banda y es mucho más robusto.
- Hardware: Se recomienda que para la puesta en producción del proyecto o de un servicio similar basado en la misma arquitectura se utilicen servidores de gran poder, especialmente para el servidor PBX y el servidor de Requerimientos, ya que la capacidad de mantener conexiones está directamente relacionada a la cantidad de memoria y robustez de estos computadores tanto para mantener llamadas concurrentes como para abrir conexiones al servidor de requerimientos. En el caso de este proyecto se utilizó para el servidor de requerimientos un servidor con 2GB de RAM, por lo que se recomienda que todos los servidores tengan memoria igual o

superior a 1GB. En las pruebas realizadas se verificó que la cantidad de memoria utilizada (512MB) era suficiente para soportar el tráfico esperado del sistema., sin embargo se recomienda la ampliación a 1GB, para garantizar el correcto desempeño del servidor TTS y brindar calidad.

- Codec: Dada la importancia del ancho de banda sin dejar de lado la calidad, se recomienda el uso de GSM que sobresale por su simpleza y calidad.
- Sistemas Operativos: Para la implementación del servidor PBX en un ambiente de producción se recomienda el uso de CentOS Linux, siguiendo la línea Red Hat del sistema piloto realizado. Para la implementación del servidor de Base de datos se recomienda el sistema operativo Solaris dada la confiabilidad y la administración de recursos que provee y para el servidor web se recomienda utilizar CentOS. Se recomienda para servidor de requerimientos el uso de Windows Server 2003 y de esta forma utilizar el driver gratuito para Windows que ofrece Oracle siguiendo la arquitectura del proyecto.
- El servidor web recomendado para este caso es Tomcat, dada la poca carga para el sistema de administración web.

5.2. Trabajos Futuros

Diferentes trabajos pueden ser derivados del presente proyecto, entre ellos:

- Implementar un servicio IVR similar o mucho más avanzado usando VXML (Voice Extensible Markup Language) o JvXvML (Java Vxml) permitiendo así la creación de IVRs de nueva generación como lo son los de diálogo dirigido o los NLU.
- Desarrollar otros proyectos CTI como lo son centros de Contactos o sistemas de enrutamiento de llamadas según habilidades, para los cuales un IVR es la etapa inicial de interacción con el usuario.

5.3. Conclusiones

Al terminar el presente proyecto de tesis se puede concluir lo siguiente:

- Se realizó el estudio de las tecnologías IVR, sus tipos y evoluciones y las razones por las cuales han prevalecido y destacado en el mercado dada la optimización que brindan en el intercambio de información, reduciendo costos de operación y mantenimiento.
- Se estudió cada uno de los componentes de la arquitectura del sistema, comprobándose la exitosa interacción entre tecnologías libres y propietarias mediante uso de drivers y/o middlewares como vendría a ser para el caso del proyecto, el servidor AGI basado en Java.
- Se comprobó y estudió la capacidad del API Asterisk-Java como controlador de la PBX Asterisk en sus distintas interfaces AGI y la coexistencia de diferentes conexiones que pueden existir desde el ámbito de programación Java que maneja.
- Se implementó exitosamente el sistema IVR-IP piloto usando una arquitectura de tres servidores ya descrita a lo largo del presente trabajo, demostrando su viabilidad y su enorme potencial de aplicación en empresas de diversos ámbitos que requieran brindar una interfaz telefónica simple y amigable a sus clientes.
- Se midió la capacidad real del sistema en los servidores implementados por medio de pruebas de desempeño observando que las capacidades del sistema satisfacían el tráfico estimado en el punto 3.1.7. Se comprobó la capacidad del servidor Asterisk al soportar 167 llamadas concurrentes y se midió el uso de CPU del servidor TTS.

BIBLIOGRAFÍA

- [BEA2006] Beasty Colin, 'INTERACTIVE VOICE RESPONSE Customer Relationship Management'; Apr 2006; 10, 4; ABI/INFORM Global. pg. 27
- [BRO2004] Brown Lois, 'Lost In IVR: The Hidden Costs of Pushing High-Value Customers Through Self-Service'
Customer Interaction Solutions, 2004
- [DIA2007] Diaz Arturo, 'Diseño e Implemetación del Centro de Operaciones y Gestión de la Red Académica Peruana en Software Libre'
PUCP, 2006
- [DOD2002] Dodd Annabel, 'The Essential Guide to Telecommunications'
Prentice Hall, 2002
- [FRO2006] Frost & Sullivan, 'U.S. IVR (Interactive Voice Response) System Markets'
URL: www.researchandmarkets.com/reports/358843
Frost & Sullivan, Febrero del 2006
- [RFC2543] Handley, *et al*, 'SIP: Session Initiation Protocol'
IETF, Junio de 1999
URL : <http://www.ietf.org/rfc/rfc2543.txt>
- [MAH2004] Mahler Paul, 'VoIP Telephony with Asterisk'
Signate, 2004
- [MOL2006] Molina José, 'Implementación de servicios VoIP sobre Asterisk'
Universidad Politécnica de Barcelona – UPC, 2006

- [TN2005] Portal T-NEWS
URL: <http://www.tnews.com.pe/actualidad/ac-20.htm>
Información de contacto: bdatos@tnews.com.pe
Consultada el 19 de diciembre del 2006
- [QUI2006] Quintana Diego, 'Diseño e Implementación de una Red de Telefonía IP con software Libre en la RAAP'
PUCP, 2006
- [RFC3261] Rosenberg, *et. al*, 'SIP: Session Initiation Protocol'
IETF, Junio del 2002
<http://www.ietf.org/rfc/rfc2543.txt>
- [STR2007] Sitio oficial de Apache Struts URL: <http://struts.apache.org/>
Consultada el 2 de junio del 2007
- [AST2007] Sitio oficial de Asterisk URL: www.asterisk.org/
Digium Inc, 2007
- [ASJ2007] Sitio oficial de Asterisk-Java
Stefan Reuter, 2007
URL: <http://asterisk-java.org/>
Información de contacto: stefan.reuter@reucon.com.
Consultada el 7 de octubre del 2006
Secciones: Descargas, Diseño.
- [FES2007] Sitio oficial de Festival
URL: <http://www.cstr.ed.ac.uk/projects/festival>
Consultada el 5 de mayo del 2007

- [ORI2007] Sitio oficial de Orient-Express URL: <http://www.orient-express.com/>
Consultada el 2 de diciembre del 2006
Secciones: Cuzco, Perurrail.
- [SOX2007] Sitio oficial de SOX URL: <http://sox.sourceforge.net/>
Consultada el 4 de marzo del 2007
Secciones: Proyecto, Descripción, FAQ.
Consultada el 5 de enero del 2007
Secciones: Descargas, Soporte.
- [SHA2002] Sharma Chetan, 'VoiceXML'
Wiley, 2002
- [IAX2005] Spencer & Millar, 'Inter-Asterisk EXchange (IAX) Version 2 - Draft'
Network Working Group, Julio del 2005
- [BER2006] Technical University of Berlin – GSM Lossy Speech Library
URL: <http://www.cs.tu-berlin.de/~jutta/toast.html>
Información de contacto: jutta@pobox.com
Consultada el 16 de julio de 2007
- [VAN2005] Van Meggelen Jim et al. 'Asterisk The future of Telephony'
O'Reilly, 2005
- [YAR2003] Yarberry William, 'Computer telephony Integration'
CRC Press, 2003
- [ZHA2006] Zhao Jielun, 'Mobile Value-Added Service Market in China'
Dansmarks Tekniske Universitet – DTU, 2006

ANEXOS

Anexo 1: Comandos Utilizados

Este anexo incluye una relación y explicación de los principales comandos utilizados en el sistema para inicialización de sus diversos componentes.

Anexo 2: Scripts de la Implementación del IVR

Este anexo comprende el código fuente del sistema IVR

Anexo 3: Scripts del Modelo de datos y Procedimientos Almacenados

Este anexo incluye los scripts necesarios para la implementación de la base de datos.

Anexo 4: Scripts fuente del Sistema Web Prototipo

Este anexo comprende el código fuente de la etapa controladora del sistema web descrito en el proyecto.

Anexo 5: Recursos Multimedia

Este anexo, incluido en el disco compacto adjunto contiene imágenes en formato .PNG del proceso de pruebas, recursos de audio utilizados en el proyecto y videos explicativos de prueba del sistema IVR.

