

UNIVERSIDAD DON BOSCO

FACULTAD DE INGENIERIA
ESCUELA DE COMPUTACION



Diseño y desarrollo de una aplicación para la automatización de servicios de la Universidad Don Bosco por medio de un Sistema de Respuesta Interactiva por Voz (IVR)

TRABAJO DE GRADUACION PARA OPTAR AL GRADO

DE

INGENIERO EN CIENCIAS DE LA COMPUTACION

Presentado por

Jenny Eleonora Romero Reyes

Julio Enrique Gutiérrez Orozco

Mario Alberto Cruz Bautista

Asesor

Ing. Ángel Soriano

Tutor

Ing. Melvin Carias

San Salvador

El Salvador

Centro América

INDICE

LISTA DE FIGURAS	V
LISTA DE DIAGRAMAS	VI
INTRODUCCION	IX
CAPITULO I. EL TEMA	1
1. Definicion del tema.....	1
2. Justificacion.....	1
3. Objetivos	3
3.1 Objetivo General	3
3.2 Objetivos Específicos	3
4. Alcances.....	4
5. Limitaciones	5
6. Análisis de Factibilidad.....	6
6.1 Factibilidad Técnica.....	6
6.2 Factibilidad Económica	7
6.3 Factibilidad Operacional.....	8
CAPITULO II. MARCO TEORICO.....	9
1. Sistema IVR	9
2. Marcacion por Tonos.....	11
3. Texto a voz.....	12
4. Herramientas.....	13
4.1 Java.....	13
4.2 Swing	14
4.3 JTAPI	14
4.3.1 XTAPI.....	15
4.4 TextSound	15
4.5 JDBC.....	15
4.6 RFax.....	16
4.6.1 Java Communication API.....	16

4.7	Java Mail	16
5.	Diagrama de Flujo de Datos.....	17
5.1	Ventajas del enfoque de flujo de datos	17
5.2	Convenciones usadas en diagramas de flujos de datos	17
5.3	Creacion del diagrama de contexto	19
5.4	El diagrama cero	19
5.5	Diagramas hijos.....	20
6.	Lenguaje Unificado de Modelado (UML)	20
6.1	Por que es necsario el UML?	21
6.2	La concepcion del UML	21
6.3	Por que se escogio el UML?	22
6.4	Diagramas del UML.....	22
CAPITULO III. DISEÑO DEL SISTEMA.....		26
1.	Diseno de la Interfaz Grafica de Usuario.....	26
1.1	Ventana principal.....	26
1.2	Barra de Menú	27
1.3	Manejador de Componentes.....	<u>27</u>
1.4	Diseñador	29
1.4.1	Componente "Home"	29
1.4.2	Area de Diseño	29
1.4.3	Barra de Componentes.....	30
a.	Selección	31
b.	Condición	31
c.	Entrada DTMF	32
d.	Consulta SQL	33
e.	Menú.....	34
f.	Actualización SQL.....	37
g.	Respuesta por Voz	38
h.	Respuesta por Fax.....	41
i.	Respuetsa por E-Mail.....	44
j.	Conexión.....	44
k.	Mover	46

1.5	Resumen de Servicio	46
1.6	Selección de funciones de MODEM	47
1.7	Configuración de Email	47
2.	Diseño del Procesamiento	48
2.1	Diagramas de Flujo de Datos	48
2.1	DFD 0	48
2.2	DFD 1	49
2.3	DFD 2	50
2.2	Diagramas de Componentes de Software	51
2.3	Diagramas de Componentes de Hardware	52
2.4	Diagramas de Clases	53
2.5	Diagrama de Hilos	56
2.6	Diagramas de Estructuras	57
2.7	Diagramas de Casos de Uso del Administrador	58
2.8	Diagramas de Casos de Uso del Usuario	59
2.9	Diagramas de Estado	60
2.10	Diagramas de Secuencia	69
3.	Desarrollo de la Solucion	87
3.1	Requerimientos minimos de hardware	87
3.2	Requerimientos de software	87
3.3	Software de instalacion	87
4.	Funcionamiento del Sistema	88
CAPITULO IV. CONCLUSIONES Y RECOMENDACIONES		94
1.	Conclusiones	94
2.	Recomendaciones	95
BIBLIOGRAFIA		96
ANEXOS		
1.	Siglas	
2.	Manual del Usuario	
3.	Ejemplos de uso del sistema IVR	

Lista de Figuras

Figura 1: Escenario IVR	9
Figura 2: Teclado DTMF	11
Figura 3: Conversión de Texto a Voz	12
Figura 4: Conectividad con JDBC	16
Figura 5: Símbolos usados en diagramas de flujos de datos	18
Figura 6: Símbolo UML de una clase	23
Figura 7: Símbolo UML de un objeto	23
Figura 8: Diagrama de caso de uso	23
Figura 9: Diagrama de estado	24
Figura 10: Diagrama de secuencia	24
Figura 11: Símbolo UML de un componente	25
Figura 12: Diagrama de distribución	25
Figura 13: Ventana principal	26
Figura 14: Submenú Design	27
Figura 15: Submenú View	27
Figura 16: Submenú Services	28
Figura 17: Manejador de componentes	38
Figura 18: Ventana de Diseño	29
Figura 19: Selección de componentes	30
Figura 20: Diálogo de configuración del componente Condición	31
Figura 21: Diálogo de configuración del componente Entrada DTMF	32
Figura 22: Diálogo de configuración del componente Consulta SQL	33
Figura 23: Diálogo de configuración del componente Menu	34
Figura 24: Diálogo de selección de tipo de opción	35
Figura 25: Diálogo de configuración de una opción estática	35
Figura 26: Diálogo de configuración de una opción dinámica	36
Figura 27: Diálogo de configuración del componente Actualización SQL	37
Figura 28: Diálogo de configuración del componente Respuesta por Voz	38
Figura 29: Diálogo de selección de tipo de línea	39
Figura 30: Diálogo de configuración de una línea estática	39
Figura 31: Diálogo de configuración de una línea dinámica	40

Figura 32: Diálogo de configuración de un encabezado de página.....	41
Figura 33: Diálogo de configuración de un pie de página	42
Figura 34: Diálogo de configuración del componente Respuesta por Fax	43
Figura 35: Diálogo de configuración del componente Respuesta por E-mail	45
Figura 36: Diálogo de Resumen de Servicio	46
Figura 37: Diálogo de selección de funciones de MODEMs	47
Figura 38: Diálogo de configuración de e-Mail	47
Figura 39: Objeto Attendant recibiendo una llamada	88
Figura 40: Creación del objeto Executor	89
Figura 41: Envío de Email	90
Figura 42: Atendiendo dos llamadas	90
Figura 43: Envío de Email a la cola.....	91
Figura 44: Notificación de que la atención de una llamada ha finalizado	92
Figura 45: Recuperación de un Email de la cola.....	92
Figura 46: Envío de un Email recuperado de la cola.....	93

Lista de Diagramas

Diagrama 1: DFD/0	48
Diagrama 2: DFD/1	49
Diagrama 3: DFD/2	50
Diagrama 4: Componentes de Software	51
Diagrama 5: Componentes de Hardware	52
Diagrama 6: Diagrama de clases	53
Diagrama 7: Diagrama de Hilos	56
Diagrama 8: Diagrama de Estructuras	57
Diagrama 9: Casos de uso del Administrador	58
Diagrama 10: Casos de uso del Usuario	59
Diagrama 11: Diagrama de estado de DtmfComponent	60
Diagrama 12: Diagrama de estado de ConditionComponent	60
Diagrama 13: Diagrama de estado de HomeComponent	61
Diagrama 14: Diagrama de estado de ControlBar	61
Diagrama 15: Diagrama de estado de MyMenuComponent	62
Diagrama 16: Diagrama de estado de StaticOption	62
Diagrama 17: Diagrama de estado de DynamicOption	62
Diagrama 18: Diagrama de estado de Include	63
Diagrama 19: Diagrama de estado de StaticLine	63
Diagrama 20: Diagrama de estado de DynamicLine	63
Diagrama 21: Diagrama de estado de FaxComponent/EmailComponent	64
Diagrama 22: Diagrama de estado de QueryComponent/UpdateComponent	64
Diagrama 23: Diagrama de estado de Design	65
Diagrama 24: Diagrama de estado de MainWindow	66
Diagrama 25: Diagrama de estado de ComponentManager	67
Diagrama 26: Diagrama de estado de Executor	67
Diagrama 27: Diagrama de estado de VoiceComponent	68
Diagrama 28: Diagrama de estado de StaticLine	68
Diagrama 29: Diagrama de estado de DynamicLine	68
Diagrama 30: Diagrama de secuencia del caso Open Design	69
Diagrama 31: Diagrama de secuencia del caso Create Design	69
Diagrama 32: Diagrama de secuencia del caso Close Design	70
Diagrama 33: Diagrama de secuencia del caso Close Design (Design not running)	70

Diagrama 34: Diagrama de secuencia del caso Save Design (Design running)	71
Diagrama 35: Diagrama de secuencia del caso Save as	71
Diagrama 36: Diagrama de secuencia del caso Start service (Other service not running).....	72
Diagrama 37: Diagrama de secuencia del caso Start service (Other service running).....	72
Diagrama 38: Diagrama de secuencia del caso Stop service (Not calls in progress).....	73
Diagrama 39: Diagrama de secuencia del caso Stop Service (Calls in progress - anyway).....	73
Diagrama 40: Diagrama de secuencia del caso Stop service (Calls in progress - Not stop).....	74
Diagrama 41: Diagrama de secuencia del caso Add component	74
Diagrama 42: Diagrama de secuencia del caso Edit component	74
Diagrama 43: Diagrama de secuencia del caso Delete component	75
Diagrama 44: Diagrama de secuencia del caso Link component.....	75
Diagrama 45: Diagrama de secuencia del caso Show statistics	76
Diagrama 46: Diagrama de secuencia del caso Close system (service running)	76
Diagrama 47: Diagrama de secuencia del caso Close system (Design open)	76
Diagrama 48: Diagrama de secuencia del caso Close system (Not constraints)	77
Diagrama 49: Diagrama de secuencia del caso Attend call	78
Diagrama 50: Diagrama de secuencia del caso Start DtmfComponent	79
Diagrama 51: Diagrama de secuencia del caso Start MyMenuComponent	80
Diagrama 52: Diagrama de secuencia del caso Start DynamicOption	81
Diagrama 53: Diagrama de secuencia del caso Start StaticOption	81
Diagrama 54: Diagrama de secuencia del caso Start Condition	82
Diagrama 55: Diagrama de secuencia del caso Start QueryComponent	82
Diagrama 56: Diagrama de secuencia del caso Start UpdateComponent	82
Diagrama 57: Diagrama de secuencia del caso Start VoiceComponent	83
Diagrama 58: Diagrama de secuencia del caso Start FaxComponent/EmailComponent.....	84
Diagrama 59: Diagrama de secuencia del caso Start Include	85
Diagrama 60: Diagrama de secuencia del caso Start DynamicLine.....	85
Diagrama 61: Diagrama de secuencia del caso Start StaticLine.....	86
Diagrama 62: Diagrama de secuencia del caso User Hang up to system.....	86

INTRODUCCION

En muchas empresas modernas, cuando una llamada entrante es para solicitar información básica o para recoger datos que no requieren grandes explicaciones, utilizan una aplicación de reconocimiento de voz o de tonos DTMF, que por medio de un sistema interactivo de respuesta hablada permite mantener diálogos avanzados hombre-maquina.

El giro y la naturaleza de la empresa no son lo importante, sino la naturaleza de las operaciones que deben llevarse a cabo para proporcionar información a quienes lo solicitan. En este sentido, se presenta el diseño y desarrollo de una aplicación de un sistema IVR para ser implementado en la Universidad Don Bosco. La funcionalidad del sistema será demostrada por medio de un caso práctico que consistirá en la prestación de servicios de consulta, reservación y renovación de material bibliotecario. En dicha prueba se utilizara una base de datos experimental, desarrollada únicamente para efectos de la demostración.

El documento se ha organizado en tres capítulos: capítulo I, El tema, capítulo II, Marco Teórico y capítulo III, Diseño del sistema.

En el capítulo I, se plantea el tema sobre una aplicación de respuesta interactiva por voz (IVR), que fue tomado por considerarlo relevante y oportuno dado que la universidad no cuenta con dicho servicio. Contempla además la justificación del por que se ha realizado una aplicación IVR para consulta de información de los estudiantes de la universidad. Seguido a esto se plantean los objetivos generales y específicos, los alcances y limitaciones que implica el desarrollo del sistema propuesto, así como el estudio de factibilidad planteado en las áreas técnica, económica y operacional.

El capítulo II contiene las bases teóricas que fundamentan la tecnología IVR y otros aspectos involucrados en el diseño y desarrollo del sistema.

El capítulo III se refiere al diseño del sistema que comprende 3 partes: los diagramas de flujo de datos, el diseño de la interfaz grafica de usuario y el diseño del procesamiento. Para el desarrollo del sistema, se presenta adjunto un CD-ROM con la aplicación.

CAPITULO I. EL TEMA

1. DEFINICIÓN DEL TEMA

Desarrollo de una aplicación de respuesta interactiva por voz (IVR) de propósito general, a partir de la cual se tendrá la posibilidad de implementar diversos servicios, tales como: consultas de información útil para el proceso de inscripción, información sobre resultados de exámenes y servicios de biblioteca

2. JUSTIFICACIÓN

Los estudiantes son la razón de ser de las instituciones educativas; por tanto, brindar el mejor servicio a los futuros profesionales debe ser una prioridad.

Es conocido por todos los estudiantes que prepararse para el período de inscripción es muy importante, ya que de ello depende el horario de clases al que regularmente tendrá que asistir durante todo un semestre. Por consiguiente, es necesario conocer todas las opciones (horarios) disponibles para las materias que se desean inscribir, pero conseguir estos horarios resulta problemático porque muchas veces se publican un día antes de iniciar el proceso de inscripción, ello conlleva a una gran aglomeración de estudiantes alrededor de las carteleras donde están publicados los horarios, sin mencionar que hay que trasladarse hasta la universidad sólo para obtener dicha información. Esta situación se vuelve mucho más crítica para aquellos estudiantes que trabajan o viven lejos de la universidad.

Otra información útil para la inscripción de materias es saber de antemano el cupo disponible para un curso en particular, lo cual permite tener una perspectiva de la posibilidad real de lograr inscribir según el horario de clases que un estudiante ha planificado.

También un aspecto importante, es que en muchas ocasiones para poder obtener las calificaciones de los exámenes parciales realizados es necesario esperar uno o más días, para que sean publicados en cartelera. Además de esto, cabe mencionar que es común que se dé la situación de que al final de ciclo los reportes de notas que son entregados a los estudiantes vienen con una o más notas que hacen falta, y los alumnos se ven en la tarea de tener que desplazarse hasta la universidad, una o varias veces, para lograr conocer dicha(s) nota(s).

Uno de los servicios importantes que presta la Universidad a sus estudiantes es el del uso de la biblioteca. Sin embargo, algunas operaciones como la renovación de préstamos, la consulta de disponibilidad de algún material, reservación de libros, etc., pueden quitar tiempo a las personas encargadas de atender a los estudiantes, lo cual afecta la rapidez de atención que perciben dichos usuarios que en ese momento estén necesitando realizar un préstamo o una devolución.

Se considera que los problemas antes mencionados pueden minimizarse, ofreciendo una alternativa de consulta rápida y eficiente, mediante una aplicación de respuesta interactiva por voz (IVR).

De todos es sabido que en nuestros días cualquier persona tiene acceso a un dispositivo telefónico, de allí que éste resulta ser el medio idóneo para que los estudiantes puedan realizar consultas telefónicas –desde su casa, lugar de trabajo o desde donde se encuentren- acerca de horarios, calificaciones, cupos, materias, etc.; todo esto y la gama de posibilidades que se pueden lograr con una base ya instalada que puede ampliarse según las necesidades crecientes.

3. OBJETIVOS

3.1. Objetivo General

Desarrollar una base tecnológica para un sistema IVR a partir de la cual se podrán implementar servicios que proporcionen un valor agregado a los estudiantes de la Universidad Don Bosco, lo cual conducirá a prestar un servicio más eficiente a la población estudiantil de dicha institución educativa.

3.2. Objetivos específicos

1. Utilizar la tecnología disponible para automatizar la consulta de información.
2. Proporcionar 24 horas de atención ininterrumpida.
3. Diseñar una interfaz simple que permita administrar fácilmente la funcionalidad del sistema a desarrollar.
4. Reducir los tiempos protocolares de atención, evitando demoras innecesarias.
5. Proporcionar una alternativa de consulta eficiente.

4. ALCANCES

1. Se podrán crear dinámicamente los servicios necesarios a través de la interfaz gráfica del sistema, con sólo arrastrar, colocar y configurar los distintos controles disponibles para la elaboración de servicios.
2. Los medios por los cuales se devolverá la respuesta al usuario son tres:
 - a. Teléfono
 - b. Fax
 - c. Correo electrónico.
3. El método que el sistema IVR utilizará para hacer la conversión de texto a voz será el de sintetización de palabras.
4. La aplicación será independiente de la base de datos de la cual se extraerá la información.
5. Como método de validación se usará se usará el carné del estudiante.
6. Para demostrar la funcionalidad del sistema, se utilizará una base de datos de prueba para realizar un caso práctico

5. LIMITACIONES

1. El usuario solamente podrá introducir información a través del teclado de un teléfono por tonos (tonos DTMF).
2. El sistema IVR utilizará líneas telefónicas analógicas hacia la red telefónica, sea ésta la PBX de la Universidad Don Bosco o directamente la PSTN (Public Switched Telephony Network).
3. El sistema únicamente podrá tener acceso a bases de datos relacionales, a través del uso de DSN (Data Source Name) de ODBC (Open Database Connectivity).
4. El sistema soportará una llamada a la vez debido a que se cuenta con sólo un módem con soporte de voz. Si se requiere soportar varias llamadas al mismo tiempo, deberán adicionarse tantos módems como número de llamadas simultáneas se deseen atender. Si se requiere extender la capacidad con mayor facilidad, puede utilizarse una tarjeta de voz Dialogic D/4PCI la cual cuenta con 4 puertos en una misma tarjeta.
5. Debido a la dificultad de acceso a la información que se encuentra en las bases de datos de la Universidad Don Bosco, el sistema IVR será probado mediante el desarrollo de la prestación del servicio de consulta, reservación y renovación de material bibliotecario por medio de una base de datos experimental.
6. El sistema IVR utilizará únicamente datos en tiempo real proveniente de las bases de datos que se hayan especificado en los servicios, es decir, que la imposibilidad de acceder a una tabla impedirá que el servicio sea proporcionado, siendo necesario enviar un mensaje de disculpas al usuario.
7. Aunque la aplicación sea independiente de la plataforma, el hardware involucrado en sistemas de telefonía (en este caso el MODEM con soporte de voz) no lo es; por tanto, la plataforma sobre la que se podrá implementar el sistema estará limitado por el equipo de telefonía subyacente.

6. ANÁLISIS DE FACTIBILIDAD

El estudio de factibilidad comprende tres niveles: técnica, económica y operacional, los cuales se detallan a continuación.

6.1 Factibilidad Técnica.

Los recursos técnicos con los que cuenta actualmente la Universidad Don Bosco, son equipos que utilizan tecnología apropiada para implementar un sistema IVR. Se dispone de casi todo el equipo que se requiere para la implementación de aplicaciones de esta naturaleza, por lo que no será necesario cambiarlo ni mejorarlo.

Como la aplicación se ha desarrollado en Java, no importa la plataforma de sistema operativo que se utilice, lo cual proporciona una increíble flexibilidad a la hora de implementar la aplicación.

El sistema que se ha desarrollado no afectará el desempeño de las aplicaciones existentes que funcionan actualmente en la universidad.

La UDB cuenta con bases de datos relacionales a partir de la cual se tomarán los datos necesarios para las aplicaciones específicas que se implementarán. A dichas bases de datos se tendrá acceso a través de JDBC (Java Data Base Connectivity).

Se cuentan con servidores en donde sería posible alojar el sistema IVR, pero si se desea puede adquirirse nuevo equipo dedicado a esta función.

Los requerimientos mínimos de hardware son:

- 1 PC Pentium II a 500 Mhz o superior, 128 MB RAM, 4 GB HD

Nota: EL número de ranuras PCI dependerá del número de líneas telefónicas a habilitar.

- 1 MODEM con soporte de voz por cada línea de teléfono habilitada.

6.2 Factibilidad Económica.

La factibilidad económica es la segunda parte de la determinación de recursos. Los recursos básicos a considerar son:

- ✓ **Costo de hardware.** Como se mencionó anteriormente, la universidad ya cuenta con la mayoría de equipo necesario (Servidores, línea telefónica/fax y los dispositivos de red telefónica) para la implementación de la aplicación.

Se requiere invertir en un módem con soporte de voz, cuyo precio de lista F.O.B Miami es de \$35.00. Opcionalmente, en lugar de adquirir un módem, se puede utilizar una tarjeta telefónica Dialogic D/4PCI con capacidad para soportar 4 llamadas simultáneamente, pero el monto de inversión ascendería a \$485.00.

Si se desea dedicar una computadora como servidor IVR y no se cuenta con dicho equipo, entonces se requerirá de una inversión adicional de \$400.00 aproximadamente.

- ✓ **Costo de software.** Este factor también favorece a la universidad ya que para el desarrollo de la aplicación se ha utilizado software de licenciamiento gratuito.
- ✓ **El costo que implica el desarrollo de este sistema.** En este sentido, se potencia el valor de la universidad, ya que se le brindará un sistema que le beneficie en el área de atención al estudiante y cuyo costo de adquisición no representará un desembolso para la universidad ya que la aplicación será desarrollada gratuitamente.

6.3 Factibilidad Operacional.

La aplicación que se implementará será un sistema para el área de atención al estudiante, utilizando tecnología IVR para consultas por teléfono. Los alumnos de la universidad podrán acceder a él con sólo tener una línea telefónica, su número de carnet, que servirá a manera de contraseña.

La interfaz de la aplicación es sencilla y práctica de tal forma que sea fácil de administrar. Además, esta aplicación proporcionará un proceso fácil y rápido para que el alumno pueda realizar consultas con toda comodidad.

Dado que la aplicación a desarrollar es una parte nueva en el área de atención al estudiante, se pretende que ésta se acople fácilmente al desempeño funcional de las demás operaciones.

CAPITULO II. MARCO TEORICO

1. SISTEMA IVR

Un sistema IVR (Interactive Voice Response) consiste en una plataforma hardware más una conexión a red de área local, a la cual se conectan host de bases de datos y la aplicación propiamente dicha.

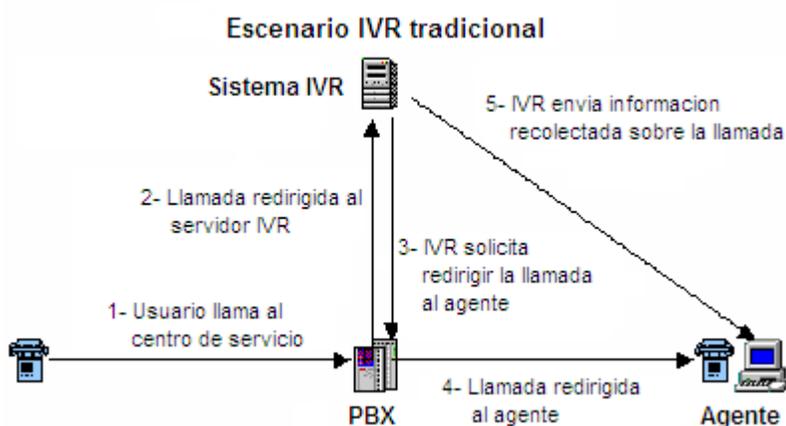


Figura 1. *Escenario IVR*

Cuando los clientes llaman a un centro de servicios automatizados basado en un sistema IVR, una serie de menús grabados les van guiando sobre las diferentes opciones/servicios que se van prestando. Los clientes hacen su elección contestando desde el teclado de su teléfono, si tiene marcación de tonos multifrecuencia (DTMF) o por respuesta hablada y en función de cada respuesta, la aplicación IVR realiza una serie de acciones sobre la base de datos. Entre éstas se incluye proporcionar determinado tipo de información recogida de la base de datos o ficheros de documentos y su lectura, traducidos a voz mediante sistemas de conversión/síntesis texto-voz (TTS).

Por su definición, los sistemas de respuesta vocal interactiva son de uso indispensable en los servicios de atención de llamadas, ya que proporcionan una descongestión necesaria en aquellos servicios saturados por consultas simples a bases de datos o información general.

Los beneficios que proporcionan los sistemas IVR para los centros de atención a clientes son:

- Reducción de costes: los sistemas IVR sustituyen a un gran número de agentes sin disminuir la productividad de la empresa.
- Reducción en el tiempo de espera.
- Actuar como receptor de llamadas en horarios fuera de atención al público.
- Aumento de la disponibilidad del servicio.
- Identificación y verificación de la identidad del usuario llamante.

2. MARCACIÓN POR TONOS

La marcación de tonos multifrecuencia DTMF (*Dial Tone Multifrequency*) consiste en un sistema de marcación basado en la transmisión de un tono de alta frecuencia y otro de baja frecuencia que combinados identifican los dígitos del teclado de un terminal telefónico (0 a 9 y teclas especiales, # *).

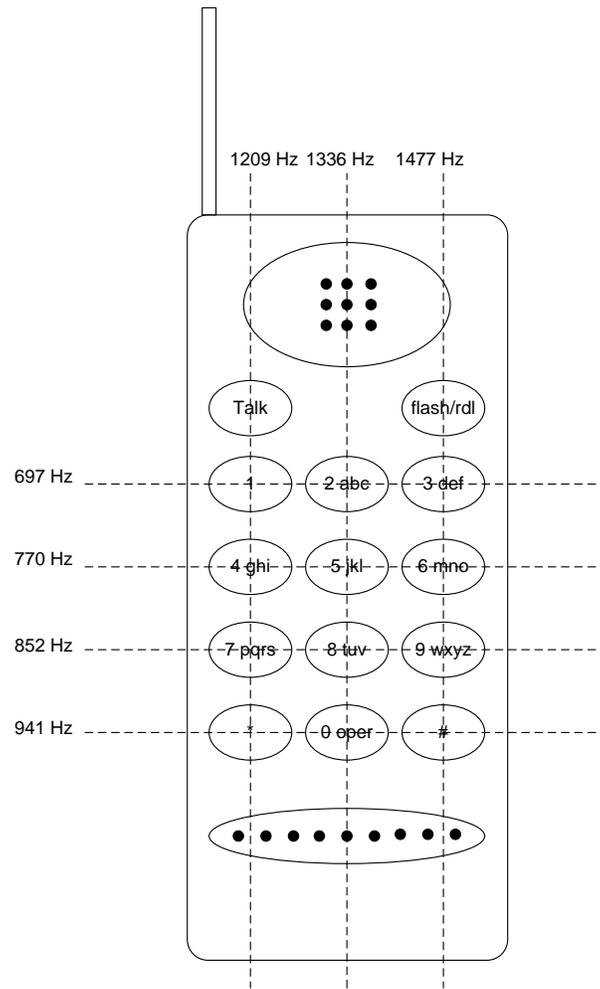


Figura 2.- Teclado DTMF.

Permiten el desarrollo de aplicaciones interactivas guiadas desde menús que indican a los usuarios llamantes la tecla de su teléfono que deben pulsar para acceder a cada servicio. Para casos donde no haya una alta penetración DTMF puede plantearse la utilización de sistemas automatizados de reconocimiento de voz.

3. TEXTO A VOZ

Es el proceso por el cual texto plano es convertido en audio digital listo para su reproducción. Los métodos de conversión de texto a voz se pueden agrupar en tres tipos:

Concatenación de palabras

Este método trabaja uniendo frases y palabras pregrabadas para construir una oración completa.

Sintetización de palabras

Este método genera palabras sintetizadas al aplicar algoritmos matemáticos para simular la cavidad de la boca, la forma de los labios y la posición de la lengua.

Concatenación difónica

Este método concatena segmentos cortos de audio digital para producir un sonido continuo. Cada segmento contiene dos sonidos, uno que inicia el sonido y uno que finaliza el sonido. Un buen ejemplo es la palabra "hello". La palabra "hello" consiste de cuatro segmentos o fonemas: h eh l oe.

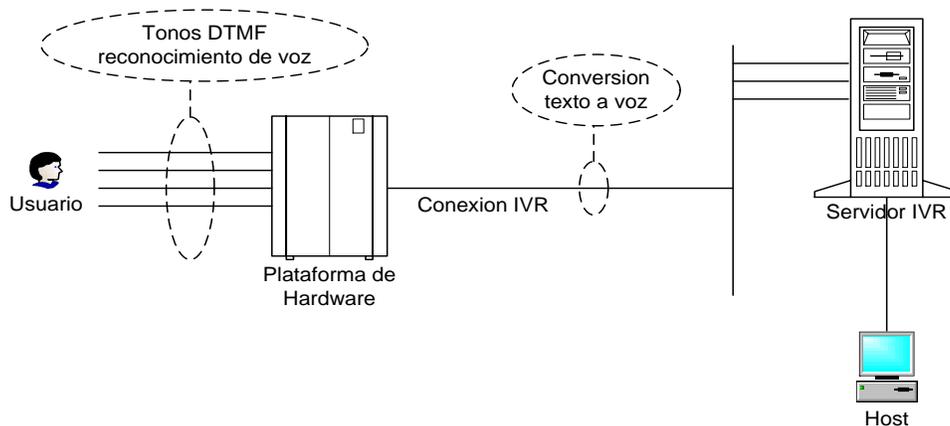


Figura 3.- Conversión de texto a voz.

4. HERRAMIENTAS UTILIZADAS

A continuación se presenta una breve descripción de las herramientas que se utilizan para el desarrollo del sistema IVR. No se trata de dar una explicación completa de cada uno de ellas - porque se requerirían de varios libros para poder hacerlo – sino mostrar una síntesis de sus características y la función que desempeñan dentro del sistema.

4.1 Java

La compañía Sun describe el lenguaje Java como “simple, orientado a objetos, distribuido, interpretado, robusto, seguro, de arquitectura neutra, portable, de altas prestaciones, multitarea y dinámico”.

Java cuenta con un compilador el cual es el encargado de generar un código “neutro” que está preparado para ser ejecutado sobre una “máquina hipotética o virtual”, denominada Java Virtual Machine (JVM). Es esta JVM quien interpreta este código neutro convirtiéndolo a código particular de la CPU o chip utilizada evitando así tener que realizar un programa diferente para cada CPU o plataforma. Entonces, el desarrollo y ejecución de aplicaciones en Java exige que las herramientas para compilar y ejecutar se encuentren accesibles.

Existen distintos programas comerciales con sus respectivos IDEs (Integrated Development Environment) que permiten desarrollar código Java, pero estos no son gratuitos:

Oracle:	Java Developer
Borland:	JBuilder
Microsoft:	J++

Para la elaboración de este sistema se ha utilizado el Java de Sun Microsystems, específicamente el Java2. Java 2 (antes llamado Java 1.2 o JDK 1.2) es la tercera versión importante del lenguaje de programación Java. La compañía Sun, creadora de Java, distribuye gratuitamente el Java Development Kit (JDK). Se trata de un conjunto de programas y librerías que permiten desarrollar, compilar y ejecutar programas en Java. En el JDK de Sun el compilador de Java se llama javac.exe y la JVM se llama java.exe.

Asimismo, Java cuenta con una serie de extensiones como: Swing, JTAPI, JSAPI, JavaComm, JavaMail, JDBC y otros, que sirven como complemento al “Core” de Java.

Sun cuenta con su propio IDE para Java: el Java SunOne, pero debido a la naturaleza de la aplicación que se está desarrollando se ha elegido utilizar Real Java, un simple editor de texto que “colorea” el código Java.

4.2 Swing

Las **JFC, Java™ Foundation Classes** son un conjunto de componentes y características para ayudar a construir los entornos gráficos de los programas o GUIs (Graphical User Interfaces).

Swing es una parte de las JFC que permite incorporar en las aplicaciones elementos gráficos de una forma mucho más versátil y con más capacidades que utilizando el AWT (Abstract Windows Toolkit) básico de Java. Algunas de las características más interesantes son:

1. Cualquier programa que utiliza componentes de Swing puede elegir el aspecto que desea para sus ventanas y elementos gráficos: entorno Windows 95/98/NT, entorno Motif (asociado a sistemas UNIX) o Metal (aspecto propio de Java, común a todas las plataformas).
2. Cualquier componente gráfico de Swing presenta más propiedades que el correspondiente elemento del AWT: Los botones pueden incorporar imágenes, hay nuevos layouts y paneles, menús, etc.
3. Posibilidad de Drag & Drop, es decir de seleccionar componentes con el ratón y arrastrar a otro lugar de la pantalla.

Toda la interfase gráfica del usuario (GUI) del sistema esta implementada con Swing, el cual es parte del paquete estándar javax.swing y viene incluido en el Java 2 SDK

4.3 JTAPI 1.2 (Java Telephony Application Programming Interface)

JTAPI es un paquete de extensión estándar que solamente incluye la especificación de las librerías; y por lo tanto se necesita de terceras partes que brinden dicha implementación para poder utilizar este tipo de paquete. Es decir, JTAPI es una especificación (independiente de la plataforma) y no una implementación (dependiente de la plataforma).

JTAPI contiene las clases necesarias para la comunicación con el sistema telefónico subyacente, por tanto su función dentro del sistema es permitir la interacción del usuario con el sistema IVR a través de la red telefónica (PSTN).

4.3.1 XTAPI 0.0.8: JTAPI Implementación

XTAPI es una implementación de JTAPI que cuenta con un proveedor para Microsoft TAPI (MSTAPI).

Esta versión de XTAPI incluye:

1. El 100% de la implementación del Core Package de JTAPI 1.2
2. El 83% de la implementación del Media Package de JTAPI 1.2

4.4 TextSound V 2.0

Es un convertidor de texto a voz que utiliza el método de sinterización de palabras producido por ByteCool Laboratories y corre sobre Windows 95/98/Me/NT4/2000/XP. Permite generar archivos .wav a 8000Hz Monofónico, que son los tipos de archivos que pueden transmitirse por la línea telefónica a través de MSTAPI.

4.5 JDBC (Java DataBase Connectivity)

JDBC es la API estándar de acceso a Bases de Datos con Java, y se incluye con el Kit de desarrollo de Java (JDK) a partir de la versión 1.1. Para trabajar con JDBC es necesario tener controladores (drivers) que permitan acceder a las distintas Bases de Datos: cada vez hay más controladores nativos JDBC.

Sin embargo, ODBC (Open Database Connectivity, Conectividad Abierta de Bases de Datos) es hoy en día la API más popular para acceso a Bases de Datos. Sin admite este hecho, por lo que, en colaboración con Intersolv (uno de principales proveedores de drivers ODBC) ha diseñado un puente que permite utilizar la API de JDBC en combinación con controladores ODBC.

La función de JDBC es la de permitir conectarse con cualquier base de datos para el cual exista el controlador ODBC respectivo, con el objetivo de poder enviar instrucciones SQL y recibir los resultados de dichas instrucciones.

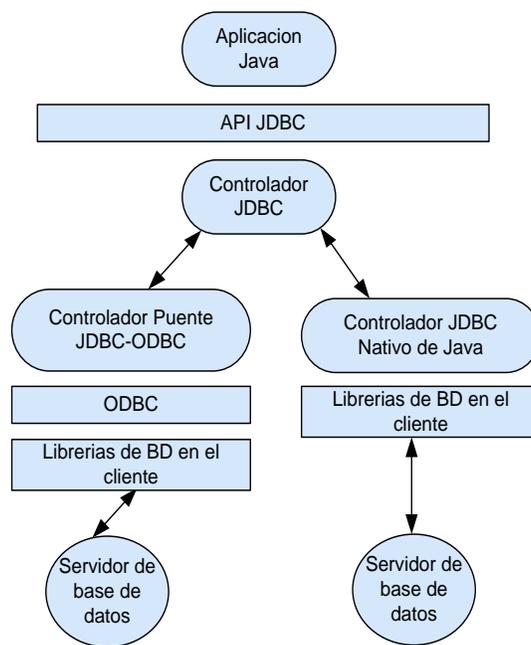


Figura 4. Conectividad con JDBC.

4.6 RFax 1.0

Es un componente de Java para el envío de faxes. Para su funcionamiento Rfax requiere:

1. MODEM clase 1, 2 o 2.0.
2. Java Communications API 2.0
3. Java 1.1 o superior

Este componente será utilizado para el envío de fax cuando así lo requiera el sistema IVR.

4.6.1 Java Communications API 2.0

API que proporciona soporte de comunicación con puertos seriales que cumplen con el estándar RS232 y con puertos paralelos que cumplen con el estándar IEEE 1284. Esta API proporciona los servicios necesarios para la elaboración de RFax.

4.7 JavaMail API 1.3.1

API de Java que brinda funciones de correo electrónico cuya implementación esta incluida en el J2EE (Java 2 Enterprise Edition). Esta API será utilizada para el envío de e-mail cuando así lo requiera el sistema IVR.

5. DIAGRAMA DE FLUJO DE DATOS

Una técnica de análisis estructurada llamada diagramas de flujos de datos (DFD), permite al analista de sistemas reunir una representación grafica de los procesos de datos a lo largo de la organización. Mediante el uso de combinaciones de solamente cuatro símbolos, el analista de sistemas puede crear una representación pictórica de los procesos que eventualmente proporcionarán documentación final del sistema.

5.1 Ventajas del enfoque de flujo de datos

El enfoque de flujo de datos tiene cuatro ventajas principales sobre la explicación narrativa de la forma en que se mueven los datos a través del sistema.

Las ventajas son:

1. Libertad para realizar en forma muy temprana la implementación técnica del sistema.
2. Una mayor comprensión de las interrelaciones de los sistemas y subsistemas.
3. Comunicación del conocimiento del sistema actual a los usuarios por medio de diagramas de flujo de datos.
4. Análisis de un sistema propuesto para determinar si han sido definidos los datos y procesos necesarios.

5.2 Convenciones usadas en diagramas de flujos de datos

Se usan cuatro símbolos básicos para diagramas el movimiento de datos en los diagramas de flujos de datos. Son un cuadrado doble, una flecha, un rectángulo con esquinas redondeadas y un rectángulo de extremo abierto (cerrado al lado izquierdo y abierto del derecho), como se muestra en la siguiente figura:

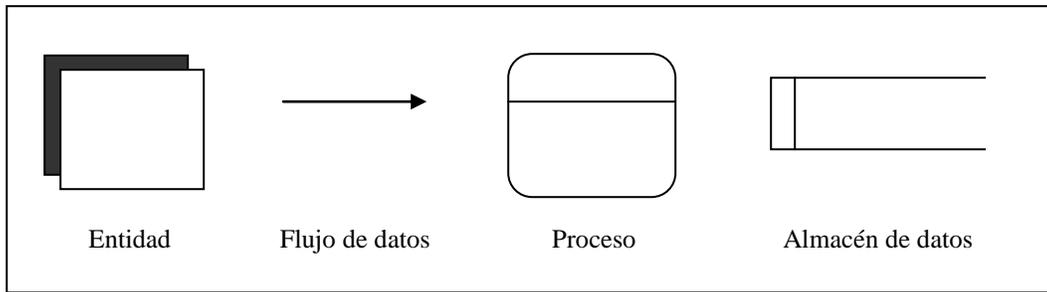


Figura 5. Símbolos usados en diagramas de flujo de datos.

El cuadrado doble es usado para representar una actividad externa (otro departamento, un negocio, una persona o una maquina) que pueden enviar datos o recibirlos del sistema. La misma entidad externa puede ser usada más de una vez en un diagrama de flujo de datos para evitar el cruce de líneas de flujo de datos.

La flecha muestra el movimiento de datos de un punto a otro, esta señala hacia el destino de los datos. Debido a que una flecha representa datos acerca de una persona, lugar o cosa, también debe ser descrita con un nombre.

Un rectángulo con esquinas redondeadas es usado para mostrar la aparición de un proceso de transformación. Los procesos siempre denotan un cambio o transformación de los datos y, por lo tanto, el flujo de datos que sale de un proceso siempre es etiquetado en forma diferente al que entra a el.

El último símbolo básico usado en los diagramas de flujo de datos representa un almacén de datos y es un rectángulo abierto. En los diagramas de flujo de datos el tipo de almacenamiento físico (por ejemplo cinta, disco flexible, disco duro, etc.) no es especificado. En este momento el símbolo de almacenamiento de datos esta simplemente mostrando un recipiente para los datos que permita adición y recuperación de datos. El almacenamiento de datos puede representar un almacenamiento manual, tal como un archivero o un archivo de base de datos computarizado.

5.3 Creación del Diagrama de contexto

Con un enfoque de arriba hacia abajo para diagramar el movimiento de datos, los diagramas se mueven de lo general a lo específico. El diagrama de contexto inicial es un panorama que incluye entradas básicas, el sistema en general y las salidas.

El diagrama de contexto es el nivel más alto en un diagrama de flujo de datos, y contiene solo un proceso que representa al sistema completo. Al proceso le es dado el número cero. Todas las entidades externas son mostradas en el diagrama de contexto, así como los flujos de datos principales que entran y salen de él. El diagrama no contiene ningún almacenamiento de datos.

5.4 El Diagrama 0 (el siguiente nivel)

Un mayor detalle que el que permite el diagrama de contexto se logra “explotando o fragmentando los diagramas”. Las entradas y salidas especificadas en el primer diagrama permanecen constantes en todos los diagramas subsecuentes. Sin embargo, el resto del diagrama original es explotado en acercamientos que involucran de tres a nueve procesos, y muestran almacenes de datos y nuevos flujos de datos de nivel más bajo. Cada diagrama explotado debe usar solamente una hoja de papel.

El Diagrama 0 es la explosión del diagrama de contexto y puede incluir hasta nueve procesos. Cada proceso es numerado con un entero, comenzando, por lo general, en la esquina superior izquierda del diagrama y trabajando hacia la esquina inferior derecha. Los almacenes de datos principales del sistema (representando archivos maestros) y todas las entidades externas son incluidas en el Diagrama 0.

5.5 Diagramas hijos (niveles más detallados)

Cada proceso del Diagrama 0 puede a su vez ser explotado para crear un diagrama hijo más detallado. El proceso del Diagrama 0 que es explotado se le llama *proceso padre*, y el diagrama que resulta es llamado *diagrama hijo*. La regla principal para la creación de diagramas hijos, el balanceo vertical, indica que un diagrama hijo no puede producir salida o recibir entrada que el proceso padre no produzca o reciba. Todos los flujos de datos de entrada o salida del proceso padre deben ser mostrados entrando o saliendo al diagrama hijo.

Al diagrama hijo se le da el mismo número que a su proceso padre en el Diagrama 0. Por ejemplo el proceso 3 explotará al Diagrama 3. Los procesos en el diagrama hijo son numerados usando el número del proceso padre, un punto decimal y un número único para cada proceso hijo. En el Diagrama 3 los procesos serían numerados 3.1, 3.2, 3.3, etc. Si el Diagrama 0 muestra los procesos 1, 2 y 3, entonces los diagramas hijos 1, 2 y 3 están todos en el mismo nivel.

6. LENGUAJE UNIFICADO DE MODELADO (UML)

El éxito de los proyectos de desarrollo de aplicaciones o sistemas se debe a que sirven como enlace entre quien tiene la idea y el desarrollador. El UML (Lenguaje Unificado de Modelado) es una herramienta que cumple con esta función, ya que ayuda a capturar la idea de un sistema para comunicarla posteriormente a quien este involucrado en su proceso de desarrollo; esto se lleva a cabo mediante un conjunto de símbolos y diagramas. Cada diagrama tiene fines distintos dentro del proceso de desarrollo.

La comunicación de la idea es de suma importancia. Antes de la llegada del UML, el desarrollo de sistemas era, con frecuencia, una propuesta al azar. Los analistas de sistemas intentaban evaluar los requerimientos de sus clientes, generar un análisis de requerimientos en algún tipo de notación que ellos mismos comprendieran (aunque el cliente no los comprendiera), dar tal análisis a uno o varios programadores y esperar que el producto final cumpliera con lo que el cliente deseaba.

Dado que el desarrollo de sistemas es una actividad humana, hay muchas posibilidades de cometer errores en cualquier etapa del proceso, y generar un programa difícil de utilizar y no generar una solución al problema original.

6.1 Por que es necesario el UML?

En los principios de la computación, los programadores no realizaban análisis muy profundos sobre el problema por resolver. Si acaso, garabateaban algo en una hoja de papel. Con frecuencia comenzaban a escribir el programa desde el principio, y el código necesario se escribía conforme se requería. En la actualidad, es necesario contar con un plan bien analizado.

Conforme aumenta la complejidad del mundo, los sistemas informáticos también deberán crecer en complejidad.

¿Cómo manejar tanta complejidad? La clave esta en organizar el proceso de diseño de tal forma que los analistas, clientes, desarrolladores y otras personas involucradas en el desarrollo del sistema lo comprendan y convengan con el. El UML proporciona tal organización.

La necesidad de diseños sólidos ha traído consigo la creación de una notación de diseño que los analistas, desarrolladores y clientes acepten como pauta. El UML es esa notación.

6.2 La concepción del UML

El UML es el resultado del trabajo hecho por Grady Booch, James Rumbaugh e Ivar Jacobson. Estos caballeros, trabajaban en empresas distintas durante los años ochenta y principios de los noventa y cada uno diseño su propia metodología para el análisis y diseño orientado a objetos.

Sus metodologías predominaron sobre las de sus competidores. A mediados de los años noventa empezaron a intercambiar ideas entre si y decidieron desarrollar su trabajo en conjunto.

Los anteproyectos del UML empezaron a circular en la industria del software y las reacciones resultantes trajeron consigo considerables modificaciones. Conforme diversos corporativos vieron que el UML era útil a sus propósitos, se conformo un consorcio del UML. Entre los miembros se encuentran DEC, Hewlett-Packard, Intellicorp, Microsoft, Oracle, Texas Instruments y Racional. En 1997 el consorcio produjo la versión 1.0 del UML. El consorcio aumento y genero la versión 1.1. El grupo adopto esta versión a finales de 1997. El OMG (Grupo de Administración de Objetos) se encargo de la conservación del UML y produjo otras dos revisiones en 1998.

El UML ha llegado ha ser el estándar en la industria del software y su evolución continua.

6.3 Por que se escogió el UML

Para el diseño de este Sistema IVR, se decidió utilizar el UML, porque es un sistema de notación que, hoy en día, se ha convertido en estándar en el mundo del desarrollo de sistemas. Además, proporciona la manera de organizar y presentar la idea de un sistema por medio de un conjunto de diagramas que son fácilmente comprensibles.

Los diagramas del UML permiten examinar un sistema desde distintos puntos de vista. Es importante recalcar que en un modelo UML no es necesario que aparezcan todos los diagramas.

¿Por qué es necesario contar con diferentes perspectivas de un sistema? Por lo general, un sistema cuenta con diversas personas implicadas las cuales tienen enfoques particulares en diversos aspectos del sistema. El escrupuloso diseño de un sistema involucra todas las posibles perspectivas, y el diagrama UML proporciona una forma de incorporar una perspectiva en particular. El objetivo es satisfacer a cada persona implicada.

6.4 Diagramas del UML

El UML está compuesto por diversos elementos gráficos que se combinan para conformar diagramas. La finalidad de los diagramas es presentar diversas perspectivas de un sistema, a las cuales se les conoce modelo. Es importante destacar que un modelo UML describe lo que supuestamente hará un sistema, pero no dice como implementar dicho sistema.

A continuación se describirán brevemente los diagramas más comunes del UML y los conceptos que representan.

a. Diagrama de Clase

Muestra las entidades en un sistema y la forma en que tales entidades se relacionan entre sí. Cada clase se representa como un rectángulo con nombre.

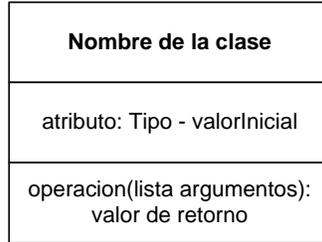


Figura 6. Símbolo UML para representar una clase.

b. Diagrama de Objetos

Muestra las instancias de las clases y sus relaciones entre si. Cada se representa como un rectángulo con nombre.

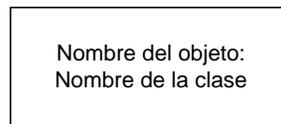


Figura 7. Símbolo UML de un objeto.

c. Diagrama de Casos de uso

Muestra la utilización del sistema. Cada caso de uso aparece como una elipse y cada actor como una figura representativa.

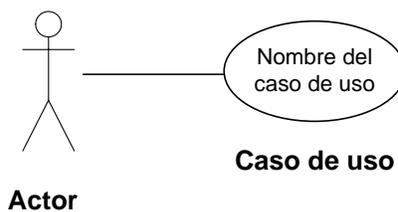


Figura 8. Diagrama de caso de uso.

d. Diagrama de estados

Captura el estado de un objeto durante cierto periodo. Un estado se representa como un rectángulo de vértices redondeados, y una transición entre estados como una línea que los conecta.

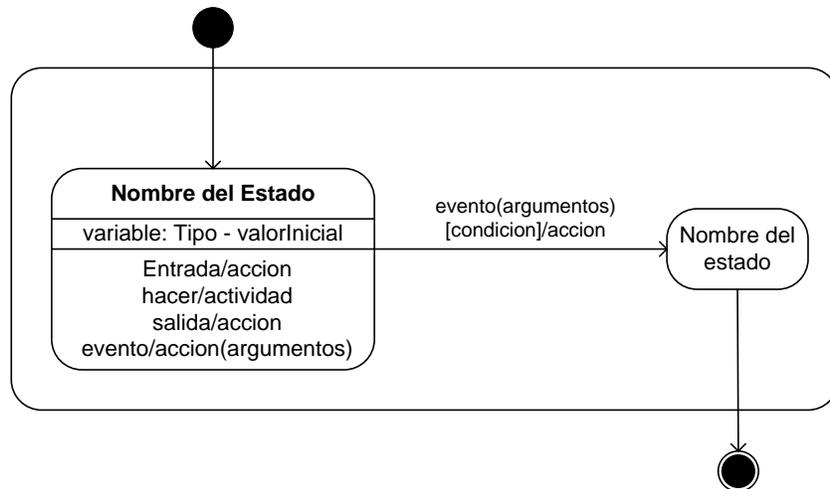


Figura 9. Diagrama de estados.

e. Diagrama de secuencias

Secuencia: representa la forma en que interactúan los objetos entre sí al paso del tiempo. Los objetos se colocan en la parte superior, y el tiempo procede desde la parte superior hacia la parte inferior del diagrama. Las flechas denotan mensajes que van de objeto a objeto.

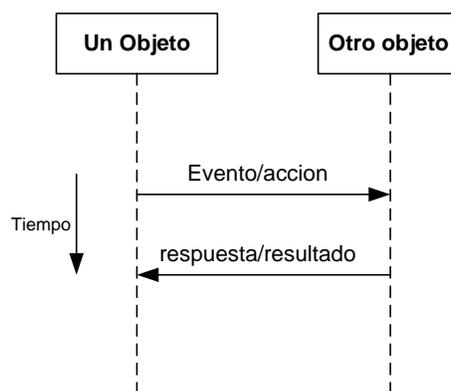


Figura 10. Diagrama de secuencia.

f. Diagrama de distribución

Un componente modela los elementos de software o hardware de un sistema. Cada componente aparece como un rectángulo con dos pequeños rectángulos superpuestos en su borde izquierdo.

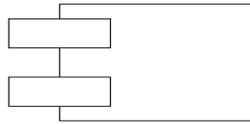


Figura 11. Símbolo UML de un componente

El diagrama de distribución representa la arquitectura física de un sistema de cómputo. Puede mostrar cada equipo de cómputo y dispositivo en el sistema y los componentes que en ellos residen. El equipo de cómputo o nodo se representa como un cubo con los componentes individuales dentro de él.

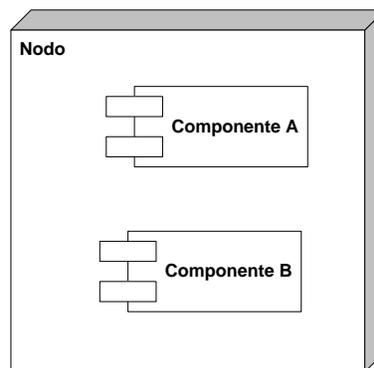


Figura 12. Diagrama de Distribución.

CAPITULO III. DISEÑO DEL SISTEMA

1. DISEÑO DE LA INTERFAZ GRAFICA DE USUARIO

La interfase del sistema IVR esta diseñada para que permita la administración del sistema de una forma muy intuitiva, es decir, que los pasos a seguir para hacer una u otra acción resulten ser obvias.

1.1 Ventana Principal

La ventana de Principal (Fig. 13), como su nombre lo indica, es la encargada de contener todos los elementos que constituyen el sistema IVR. Está formada por tres elementos básicos: La barra de menú, la barra de herramientas y el área asignada a los demás elementos del sistema (Manejador de Componentes, Diseñador, etc.)



Figura 13. Ventana Principal

1.2 Barra de Menú

En el menú se encuentran las operaciones básicas que podrá realizar el sistema. Dicho menú está dividido en tres submenús: Diseño, Servicios, y Aplicación.

a. Menú Diseño (Design)

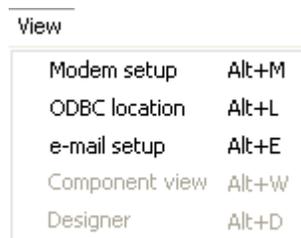
Mediante la utilización del menú Design (Fig. 14), se controla la creación, modificación y cierre del diseño de un servicio dado. Este se encuentra dividido en: New, Open, Save, Save As, Close y Quit.



Design	
New...	Alt+N
Open...	Alt+O
Save...	Alt+S
Save As...	Alt+A
Close	Alt+C
Quit	Alt+Q

Figura 14. Submenú Designs

b. Menú Ver (View)



View	
Modem setup	Alt+M
ODBC location	Alt+L
e-mail setup	Alt+E
Component view	Alt+W
Designer	Alt+D

Figura 15. Submenú View

- MODEM setup: Permite especificar las funciones de los módems instalados en el sistema
- ODBC location: Para fijar la ubicación del archivo ODBC.ini, que es el que contiene las entradas para cada uno de los DSN configurados en el sistema.
- e-mail setup: Se utiliza para fijar el servidor SMTP de correo saliente y la dirección de correo de origen que aparecerá en los correos enviados por el sistema IVR.

- Component view: Si la ventana del “Component View” ha sido cerrada por el usuario durante la edición de un diseño, esta opción permite hacer nuevamente visible dicha ventana.
- Designer: Igual que la opción anterior pero referida a la ventana del diseñador. (Ver figura 15)

c. Menú Servicios (Services)

Utilizado para el manejo de los servicios, siendo sus opciones: Start, Stop y Resume, que permiten poner en funcionamiento un servicio diseñado, detenerlo y poder consultar un resumen de las principales actividades efectuadas por el servicio activo, respectivamente.



Figura 16. Submenú Service

1.3 Manejador de Componentes

Manejador de Componentes es la ventana donde se muestran los objetos creados para un determinado servicio y las dependencias con respecto a los demás componentes (Fig. 17).

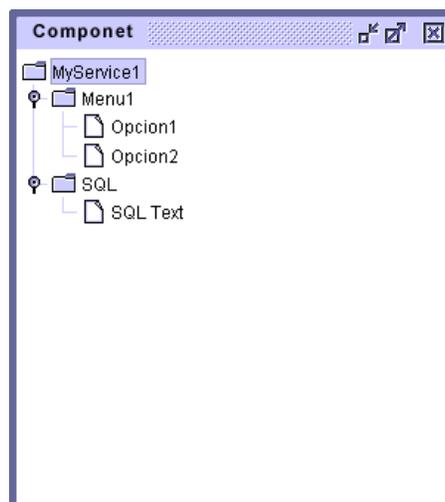


Figura 17. Manejador de Componentes

1.4 Diseñador

La ventana de diseño es el área donde se pueden editar el diseño de los servicios del sistema, y está compuesto por 4 elementos básicos: el componente “Home”, el área de diseño, la barra de herramientas y la barra de estado (Fig. 18).

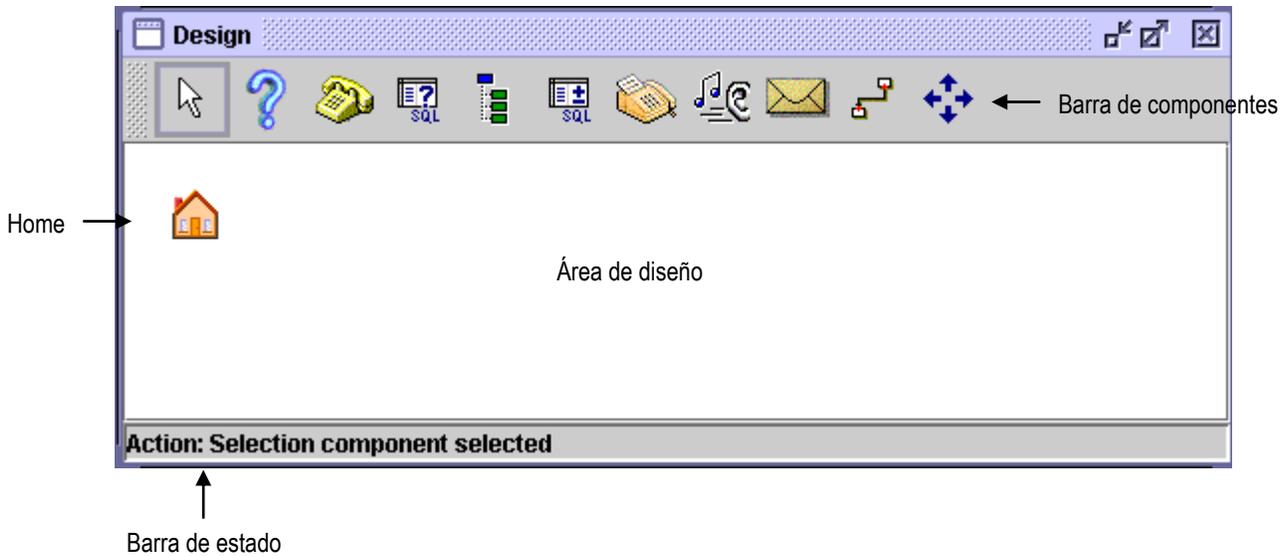


Figura 18. Ventana de Diseño

1.4.1 Componente Home

Es el único componente que siempre va a estar presente en cualquier diseño y que automáticamente aparece en cualquier diseño nuevo que se cree.

1.4.2 Área de Diseño

Es el área en el cual está permitido realizar el diseño de servicios mediante la acción de seleccionar y colocar componentes desde la barra de componentes.

1.4.3 Barra de Componentes

Contiene todos los componentes disponibles con los cuales es posible crear un diseño. Cada componente en la barra puede estar seleccionado o no. Un componente es seleccionado de la barra al hacer click sobre dicho componente. El estado de selección se percibe mediante un recuadro que rodea al componente respectivo o puede visualizarse mediante un mensaje desplegado en la barra de estado del diseñador (Fig. 19).

La barra de componentes no es fija, sino que puede colocarse en cualquier extremo de la ventana de diseño o puede estar flotante.



Figura 19. Selección de componentes

Componentes

Cuando los componentes están en el área de diseño, estos pueden responder a:

1. Doble click para iniciar la edición de su configuración.
2. Click derecho para mostrar un menú emergente con dos opciones: Editar y Borrar

A continuación se describen las funciones de cada uno de los respectivos componentes presentes en la barra:

a. Selección



Función

Cuando se selecciona permite que los componentes situados en el área de diseño puedan responder al doble click o al click derecho.

Datos de Configuración:

Ninguno

b. Condición

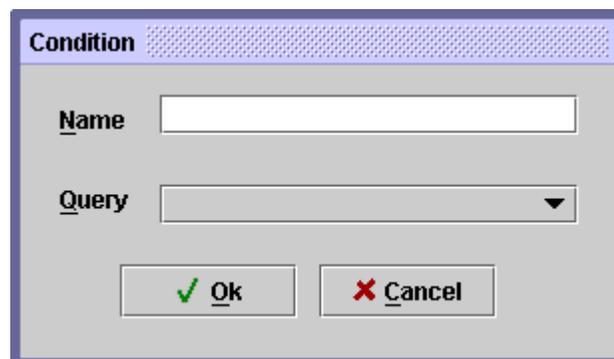


Función

Probar si una condición es verdadera o no. La prueba está basada únicamente en la presencia o ausencia de resultados de una consulta.

Datos de Configuración:

1. Nombre del componente.
2. La consulta en la que se basará la prueba.



The image shows a dialog box titled "Condition". It contains two input fields: "Name" with a text box and "Query" with a dropdown menu. At the bottom, there are two buttons: "Ok" with a green checkmark and "Cancel" with a red X.

Figura 20. Diálogo de configuración del componente Condición

c. Entrada DTMF

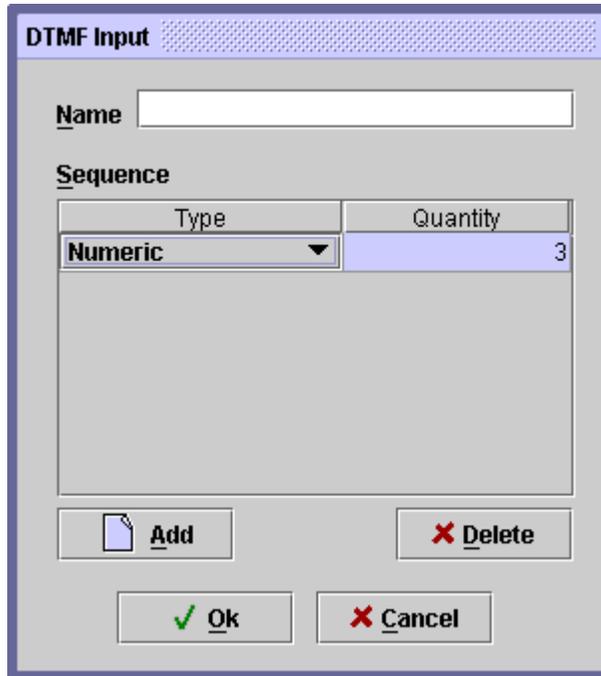


Función

Captar los pulsos DTMF enviados por el usuario.

Datos de Configuración:

1. Nombre del componente
2. Secuencia (dígitos y/o números) que se espera recibir.



The dialog box is titled "DTMF Input". It contains a "Name" field with an empty text box. Below it is a section labeled "Sequence" containing a table with two columns: "Type" and "Quantity". The table has one row with "Numeric" in the "Type" column and "3" in the "Quantity" column. At the bottom of the dialog, there are four buttons: "Add" (with a document icon), "Delete" (with a red X icon), "Ok" (with a green checkmark icon), and "Cancel" (with a red X icon).

Type	Quantity
Numeric	3

Figura 21. Diálogo de configuración del componente Entrada DTMF

d. Consulta-SQL



Función

Realizar conexiones hacia una base de datos con el objetivo de enviar una consulta y guardar los resultados.

Datos de configuración:

1. Nombre del componente
2. Nombre del DSN
3. Instrucción SELECT-SQL

The image shows a dialog box titled "SELECT Statement". It contains the following fields and controls:

- Name:** A text input field.
- DSN:** A dropdown menu.
- SQL Statement:** A large text area for entering the SQL query.
- Buttons:** "Ok" (with a green checkmark) and "Cancel" (with a red X).

Figura 22. Diálogo de configuración del componente Consulta-SQL

e. Menú

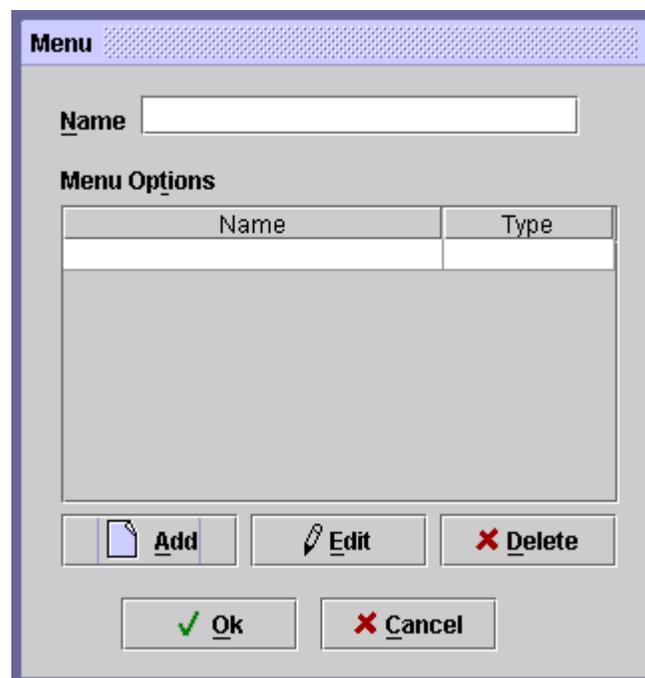


Función

Reproducir un conjunto de opciones de las cuales el usuario que llama debe seleccionar una.

Datos de Configuración

1. Nombre del componente
2. Lista de Opciones



The dialog box is titled "Menu". It contains a text input field labeled "Name". Below it is a section titled "Menu Options" which contains a table with two columns: "Name" and "Type". The table is currently empty. At the bottom of the dialog, there are five buttons: "Add" (with a document icon), "Edit" (with a pencil icon), "Delete" (with a red X icon), "Ok" (with a green checkmark icon), and "Cancel" (with a red X icon).

Figura 23. Diálogo de configuración del componente Menú

Las opciones que pueden agregarse, editarse y eliminarse son de dos tipos: Estáticas o Dinámicas (Fig. 24).

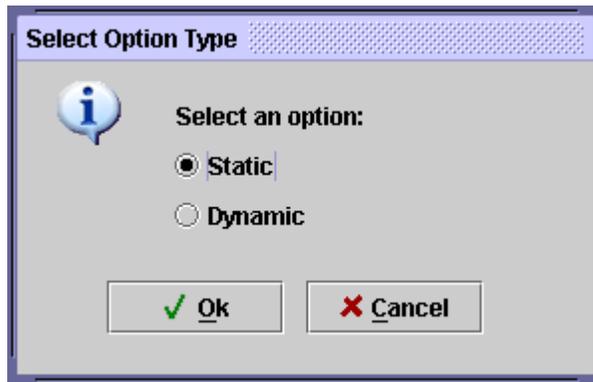


Figura 24. Diálogo de selección de tipo de Opción

Estáticas:

Función

Reproducir un texto fijo.

Datos de Configuración:

1. Nombre de la opción
2. Texto a reproducir

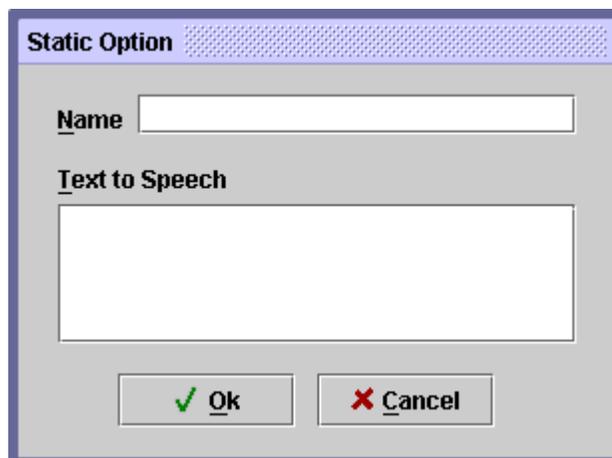


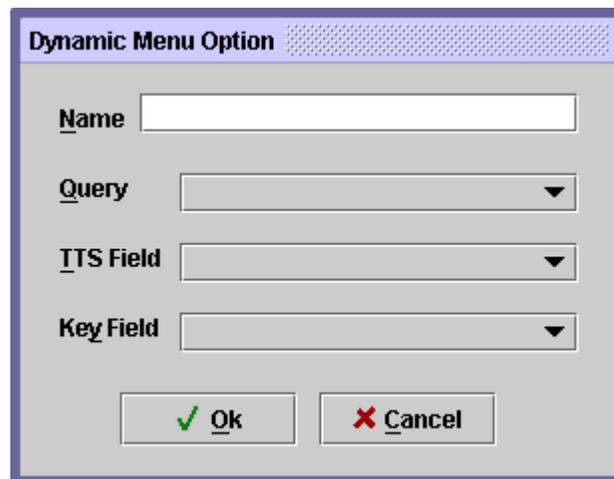
Figura 25. Diálogo de configuración de una opción estática

Dinámicas:

Función: Reproducir el resultado de un campo de una consulta

Datos de Configuración:

1. Nombre de la opción
2. Consulta en la que se basa el menú
3. Campo a reproducir
4. Campo clave asociado del campo a reproducir



The image shows a dialog box titled "Dynamic Menu Option". It contains four input fields: "Name" (a text box), "Query" (a dropdown menu), "ITS Field" (a dropdown menu), and "Key Field" (a dropdown menu). At the bottom, there are two buttons: "Ok" with a green checkmark icon and "Cancel" with a red X icon.

Figura 26. Diálogo de configuración de una opción dinámica

f. Actualización-SQL



Función

Realizar actualizaciones en tablas de una base de datos.

Datos de configuración:

1. Nombre del componente
2. Nombre del DSN
3. Instrucción UPDATE-SQL

The image shows a dialog box titled "UPDATE Statement". It has a light gray background and a blue border. At the top, there is a title bar with the text "UPDATE Statement". Below the title bar, there are three main sections: "Name" with a text input field, "DSN" with a dropdown menu, and "SQL Statement" with a large text area. At the bottom of the dialog, there are two buttons: "Ok" with a green checkmark icon and "Cancel" with a red X icon.

Figura 27. Diálogo de configuración del componente Actualización-SQL

g. Respuesta de Voz



Función

Reproducir un conjunto de líneas de texto

Datos de Configuración:

1. Nombre del componente
2. Lista de líneas

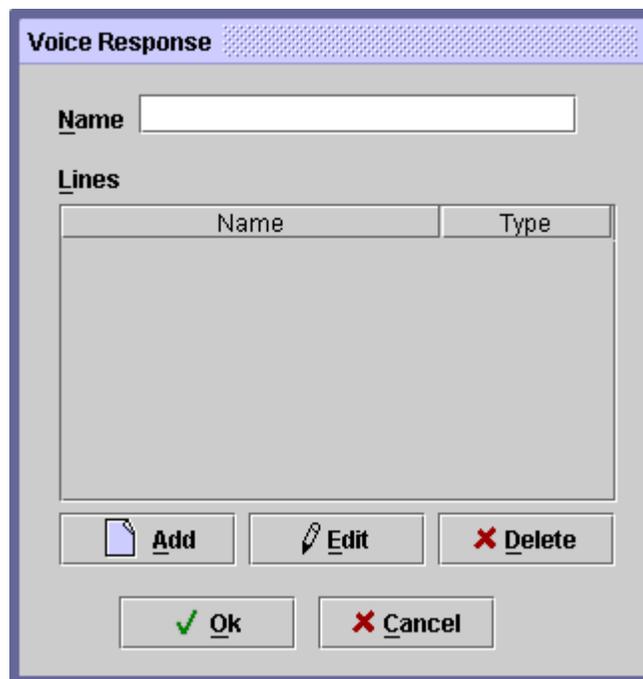


Figura 28. Diálogo de configuración del componente Respuesta de Voz

Las líneas de texto que pueden agregarse, editarse y eliminarse son de dos tipos: Estáticas o Dinámicas (Fig. 29).

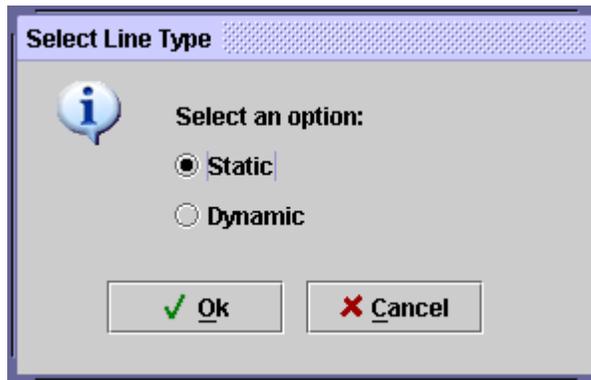


Figura 29. Diálogo de selección de tipo de Línea

Estáticas:

Función: Reproducir un texto fijo.

Datos de Configuración:

1. Nombre de la opción
2. Texto a reproducir

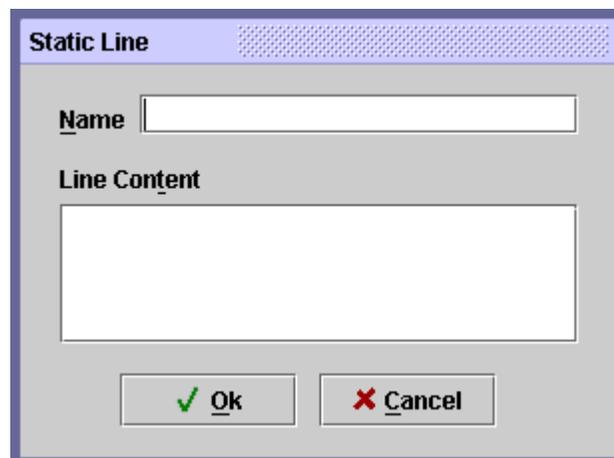


Figura 30. Diálogo de configuración de una línea estática

Dinámicas:

Función: Reproducir el resultado de un campo de una consulta

Datos de Configuración:

1. Nombre de la opción
2. Consulta en la que se basa la respuesta por Voz
3. Campo(s) a reproducir

The image shows a dialog box titled "Dynamic Line". It contains the following elements:

- A text input field labeled "Name".
- A dropdown menu labeled "Query".
- A section titled "Query Fields" containing a table with two columns: "Field Name" and "Selection". The table has three rows, each with an empty cell under "Field Name" and a checkbox under "Selection".
- At the bottom, there are two buttons: "Ok" (with a green checkmark icon) and "Cancel" (with a red X icon).

Figura 31. Diálogo de configuración de una línea dinámica

h. Respuesta por Fax



Función

Enviar fax

Datos de configuración (Fig. 34)

1. Nombre del componente
2. Encabezado del fax
3. Seleccionar la consulta y campo que contiene el número de fax
4. Seleccionar la Consulta y el(los) campo(s) que forman el cuerpo del fax
5. Pie de página del fax

Header y Footer

El encabezado y pie de página están formados por líneas de texto.

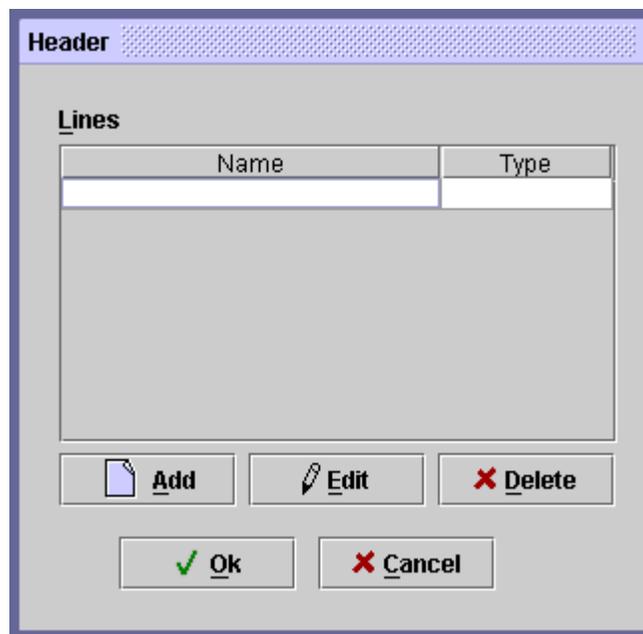


Figura 32. Diálogo de configuración de encabezado de página

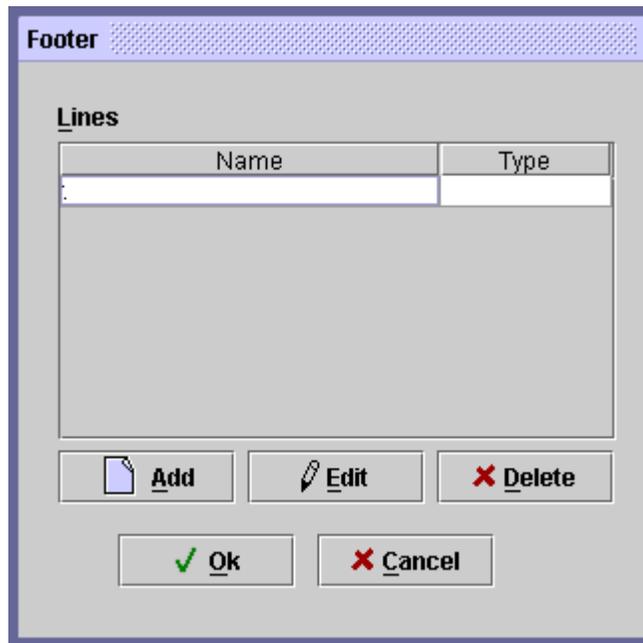


Figura 33. Diálogo de configuración de pie de página

Las líneas de texto pueden ser estáticas o dinámicas y siguen básicamente el mismo patrón de configuración que las líneas de un componente de repuesta de voz.

Fax Response

Name

Header

Fax Number

Query

Query Field

Content

Query

Query Fields

Field Name	Selection
	<input type="checkbox"/>
	<input type="checkbox"/>
	<input type="checkbox"/>

Footer

Figura 34. Diálogo de configuración del componente Respuesta por Fax

i. E-mail



Función

Enviar e-mail

Datos de configuración (Fig. 35)

1. Nombre del componente
2. Encabezado del email
3. Seleccionar la consulta y campo que contiene la dirección de e-mail
4. Seleccionar la Consulta y el(los) campo(s) que forman el cuerpo del e-mail
5. Pie de página del e-mail

Header y Footer

El encabezado y pie de página están formados por líneas de texto la forma de configuración es igual al de el encabezado y pie de página de un componente de fax.

j. Conexión



Función

Colocar conexiones entre componentes. Dichas conexiones representan la dirección de flujo a través de los componentes. Para establecer una conexión únicamente se debe hacer click en el componente origen y luego otro click en el componente destino. Visualmente una conexión se presenta como una línea que parte del componente origen hacia el destino. El sentido de una conexión se simboliza mediante un círculo negro en el componente destino.

No esta permitido:

1. Conectar un componente a si mismo, a excepción del componente Menú.
2. Intentar colocar como componente destino de una conexión al componente Home.
3. Hacer que un componente sea el origen de una conexión cuando éste todavía no forma parte como destino en otra conexión.
4. Colocar dos conexiones entre los mismos componentes pero en sentidos contrarios.

Datos de Configuración:
Ninguno

E-Mail Response

Name

Hader

e-Mail Address

Query

Query Field

Content

Query

Query Fields

Field Name	Selection
	<input type="checkbox"/>
	<input type="checkbox"/>
	<input type="checkbox"/>

Footer

Figura 35. Diálogo de configuración del componente Respuesta por E-mail

k. Mover



Función

Mover componentes situados en el área de diseño mediante la acción de arrastrar y soltar.

Datos de Configuración:

Ninguno

1.5 Resumen de Servicio

Este cuadro de diálogo muestra los datos más relevantes del servicio actualmente en ejecución.

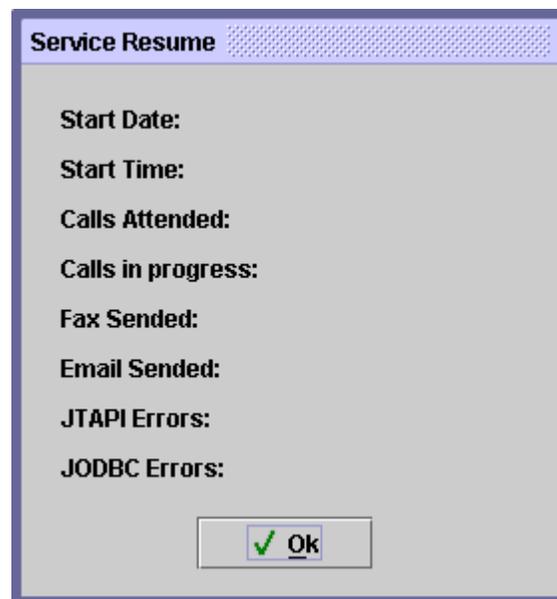


Figura 36. Diálogo de Resumen de Servicio

1.6 Selección de Funciones de Módem

Permite seleccionar las funciones que desempeñarán los módems presentes en el servidor del sistema IVR

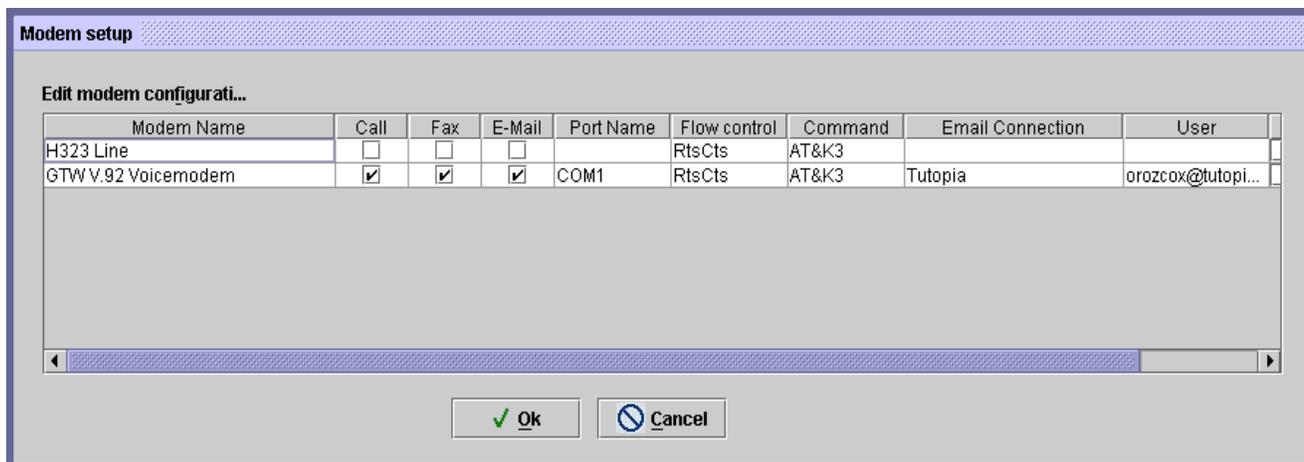


Figura 37. Diálogo de selección de funciones de MODEM

1.7 Configuración de e-mail

Permite seleccionar la dirección o el nombre del servidor SMTP de correo saliente y la dirección de correo electrónico del remitente.

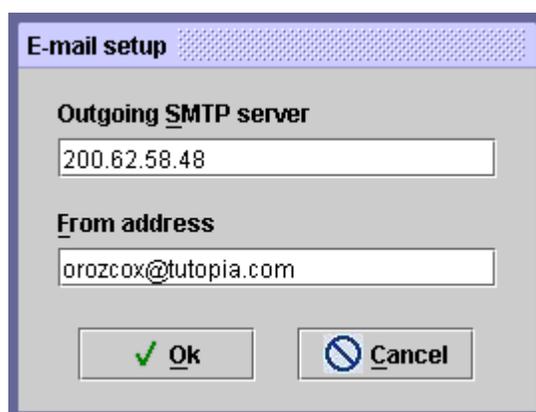


Figura 38. Diálogo de configuración de Email

2. DISEÑO DE PROCESAMIENTO

2.1 Diagrama de Flujo de Datos

2.1.1 Diagrama de contexto

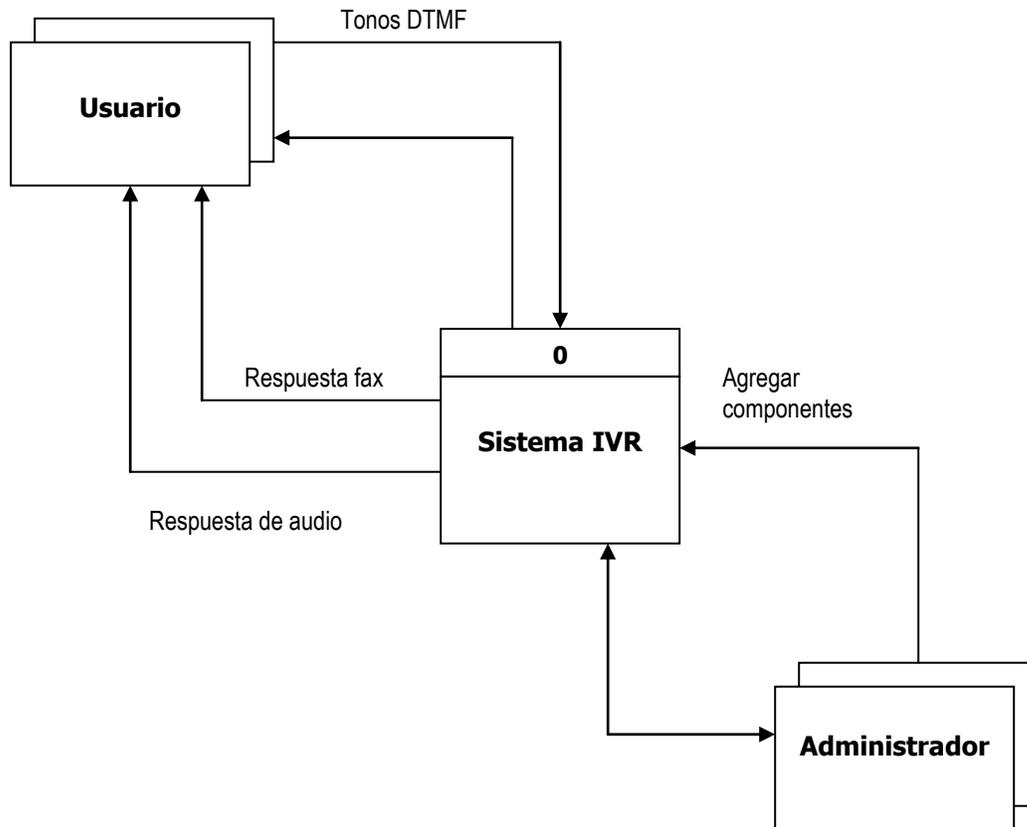


Diagrama 1. Diagrama de contexto

2.1.2 DFD / 1

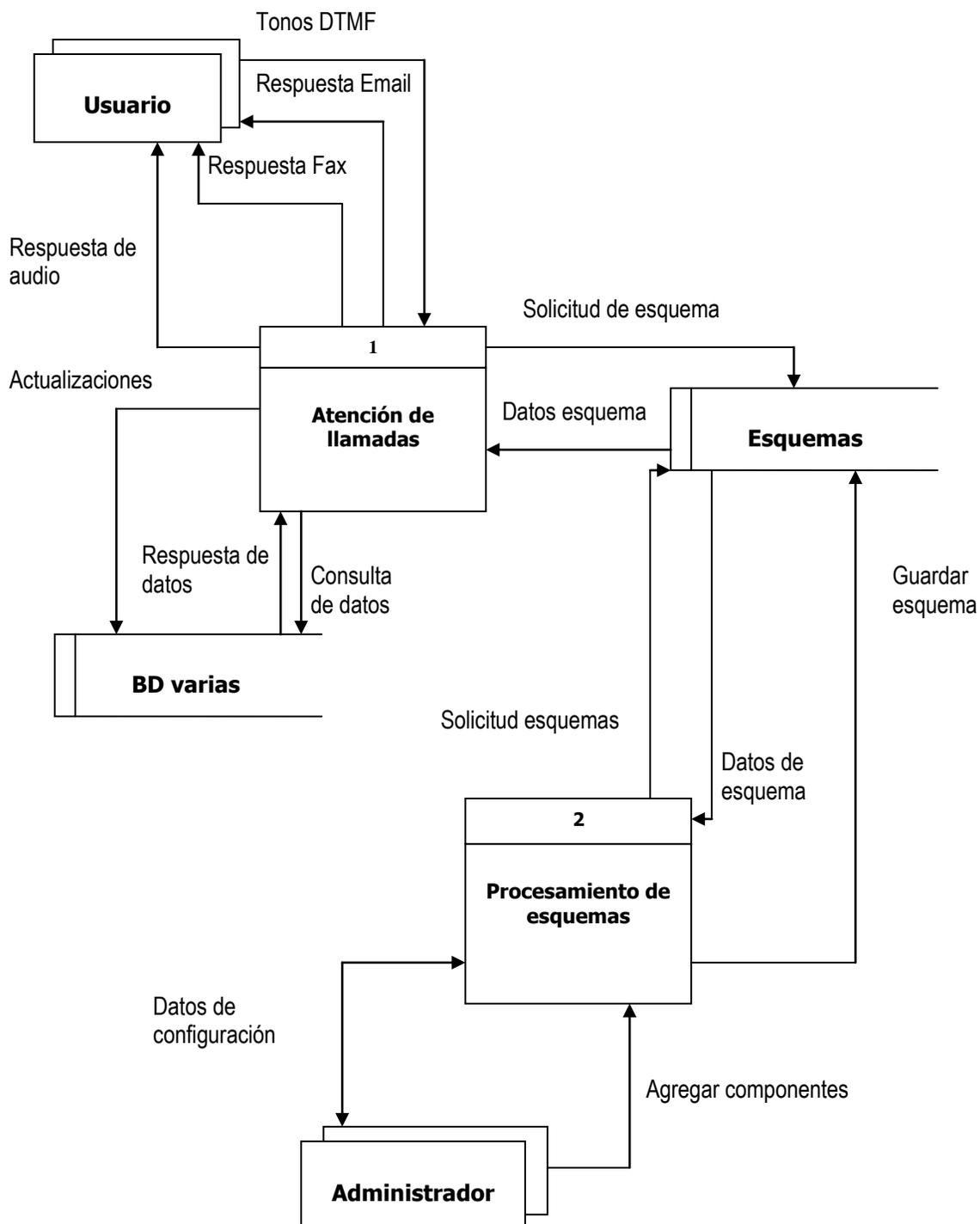


Diagrama 2. DFD/1

2.1.3 DFD / 2

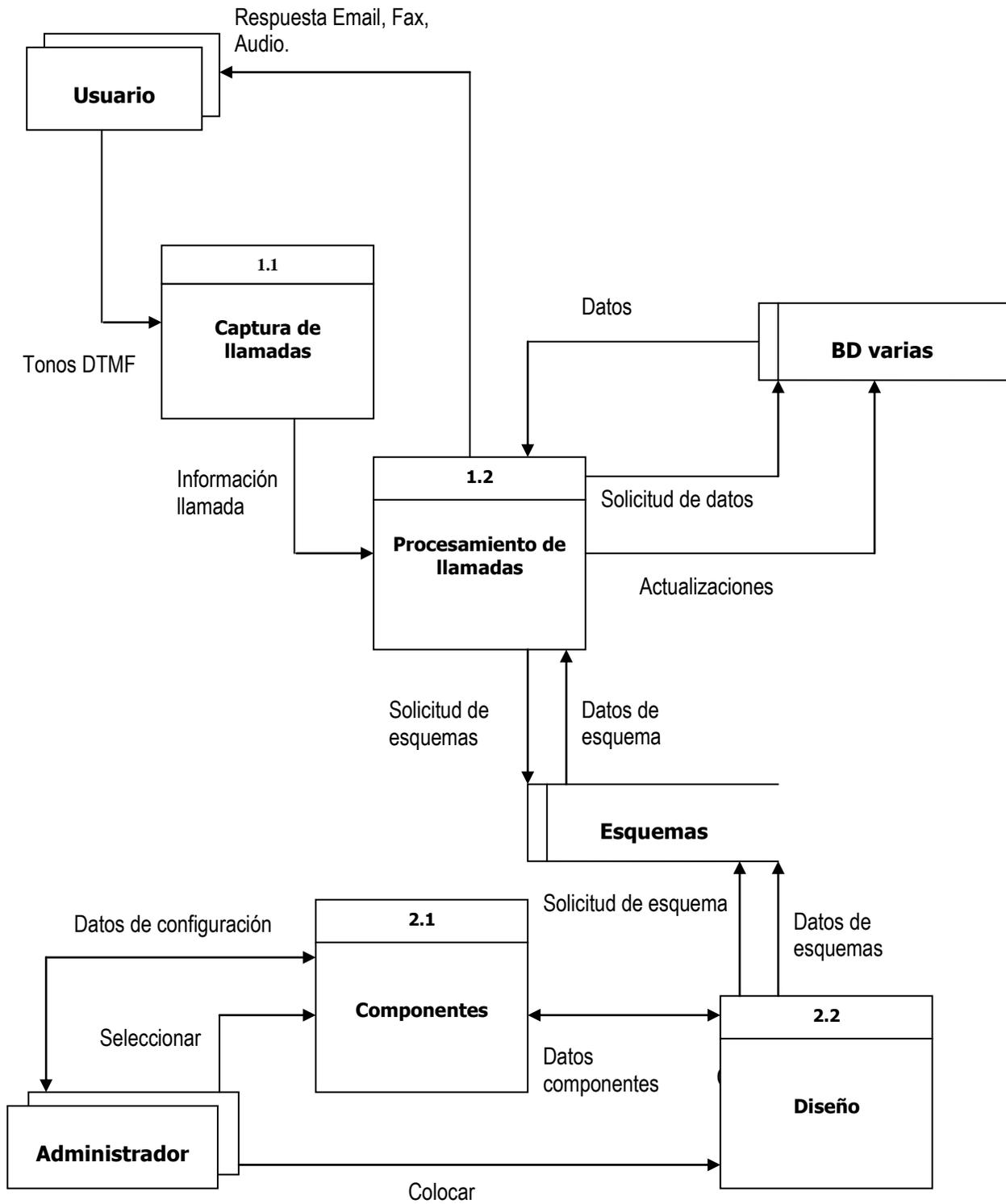


Diagrama 3. DFD/2

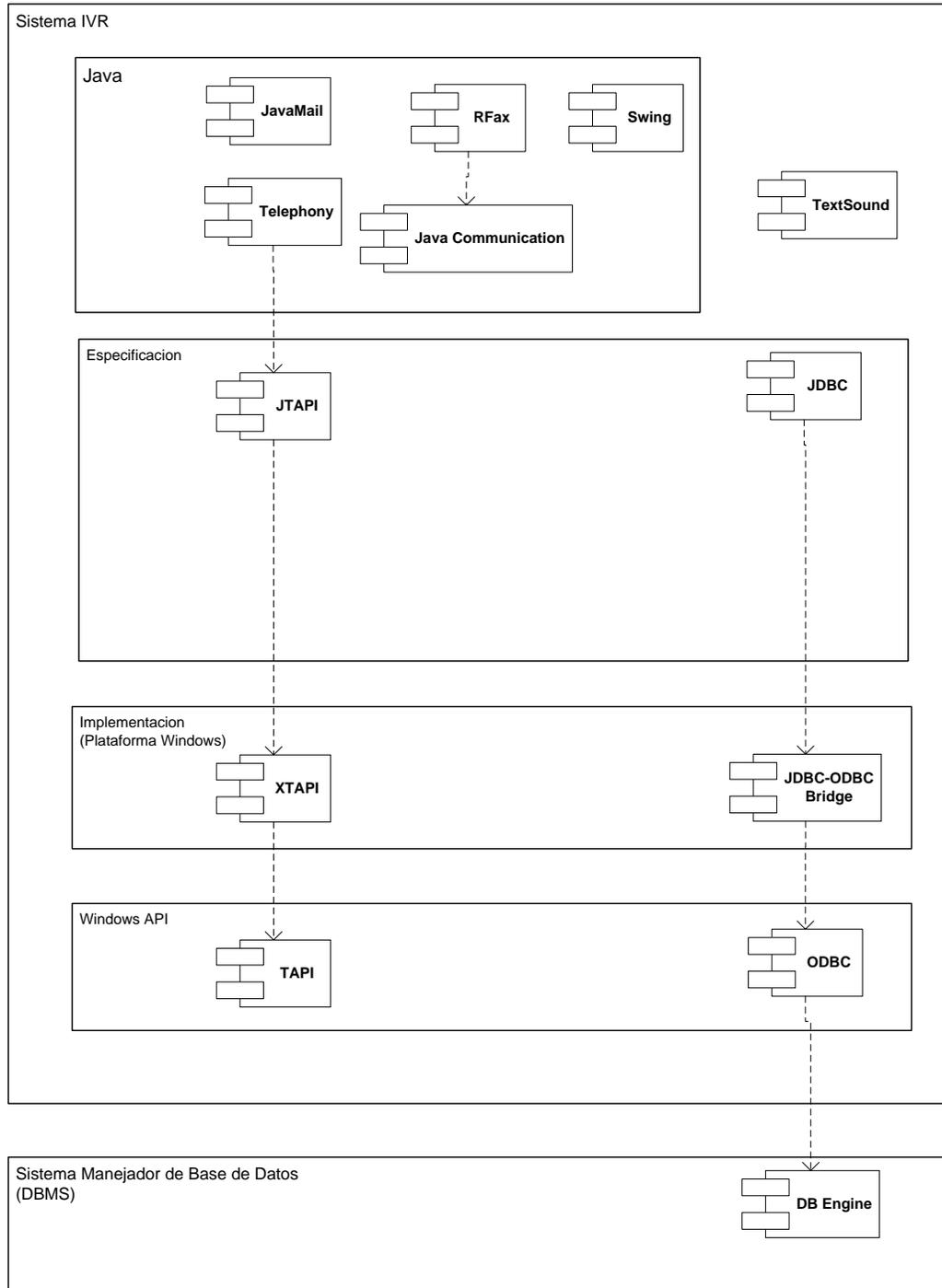


Diagrama 4. Componentes de Software

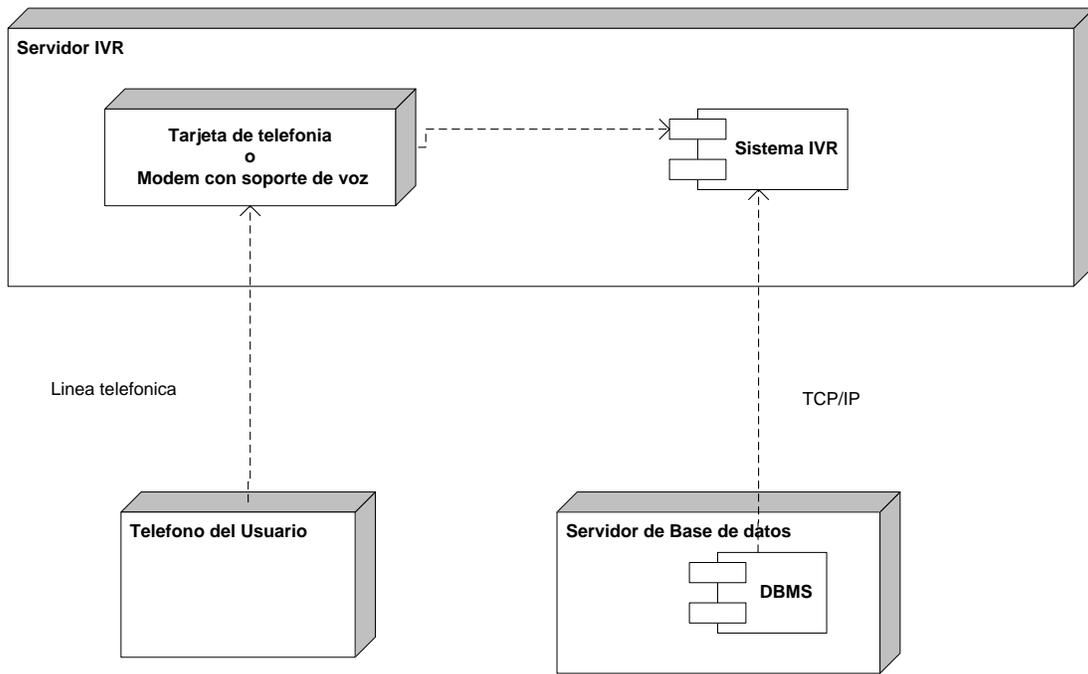


Diagrama 5. Componentes de Hardware

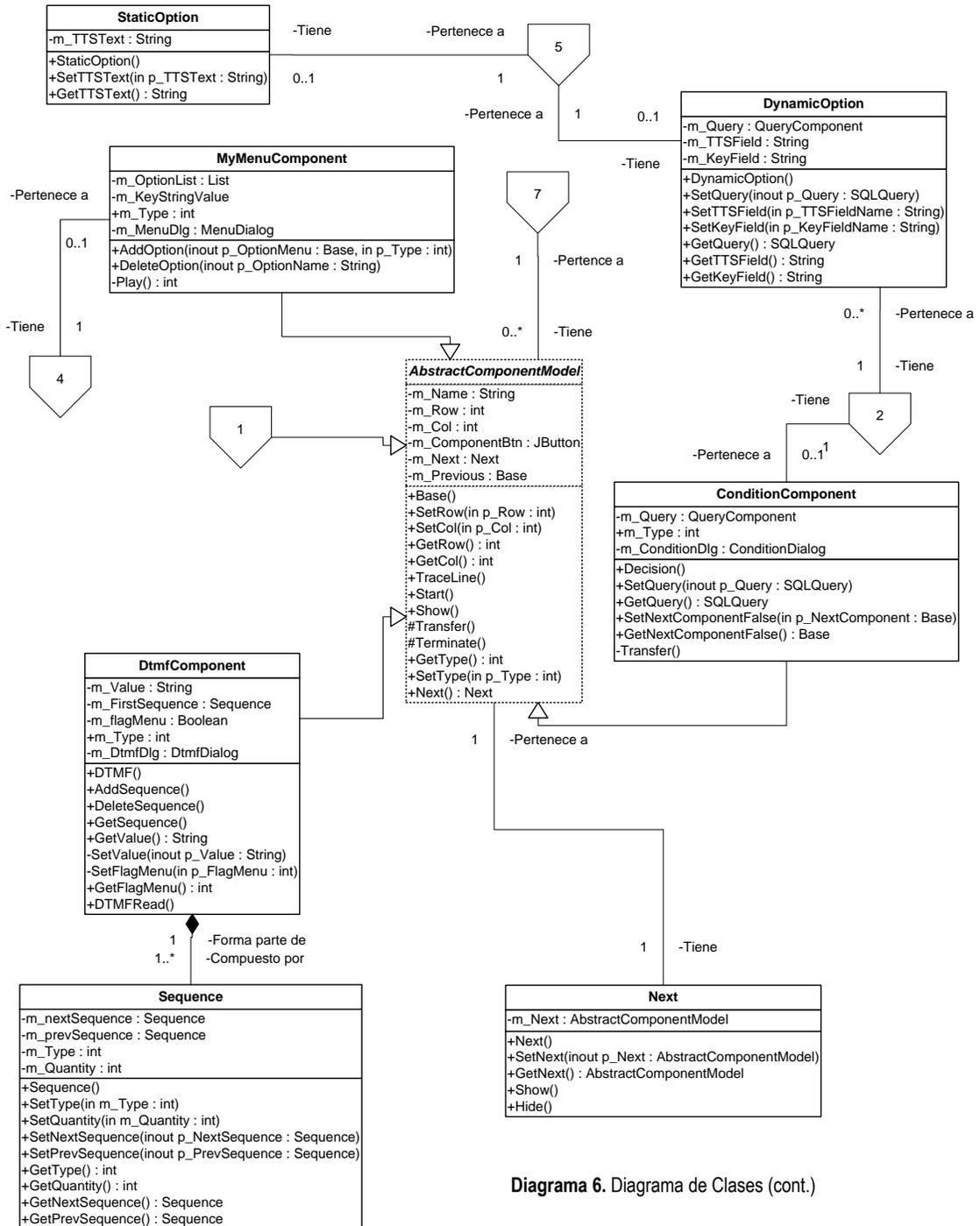


Diagrama 6. Diagrama de Clases (cont.)

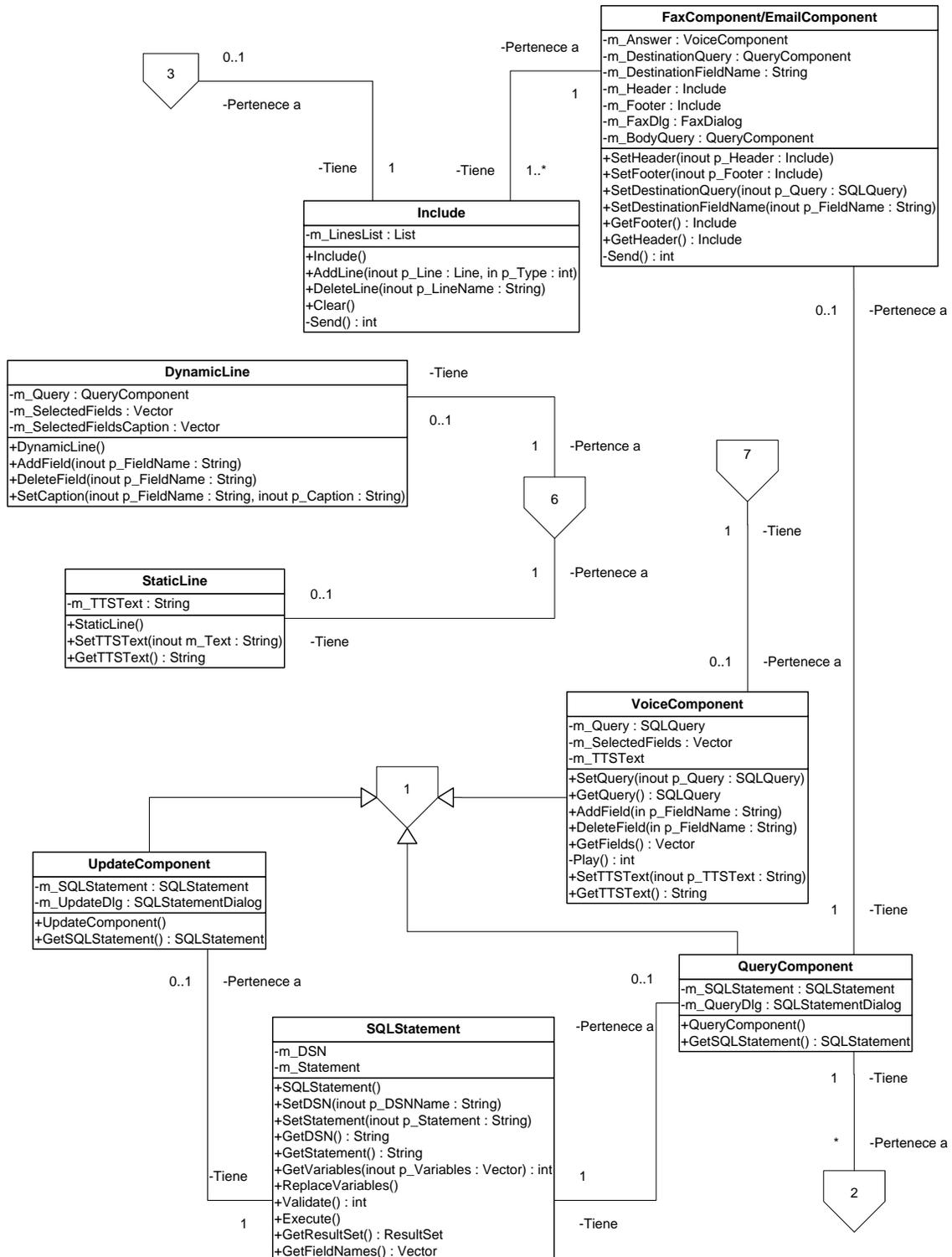


Diagrama 6. Diagrama de Clases (Cont)

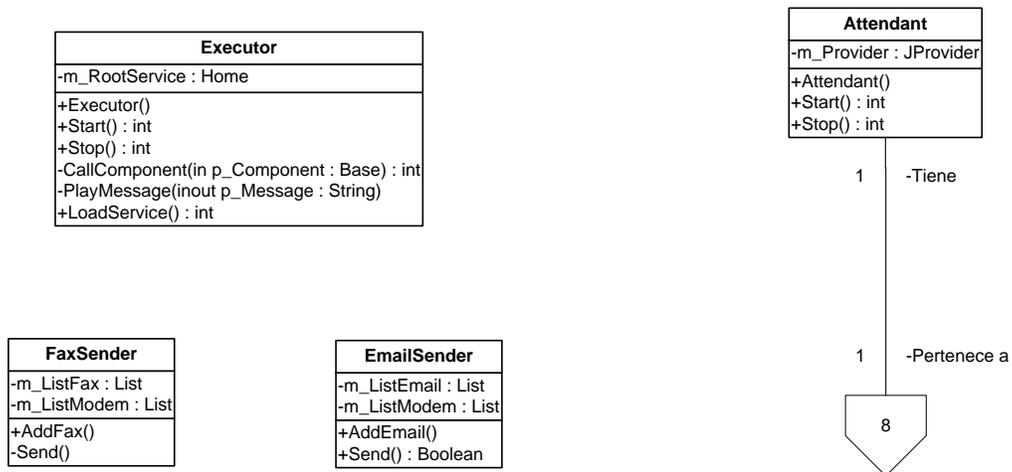


Diagrama 6. Diagrama de Clases (cont.)

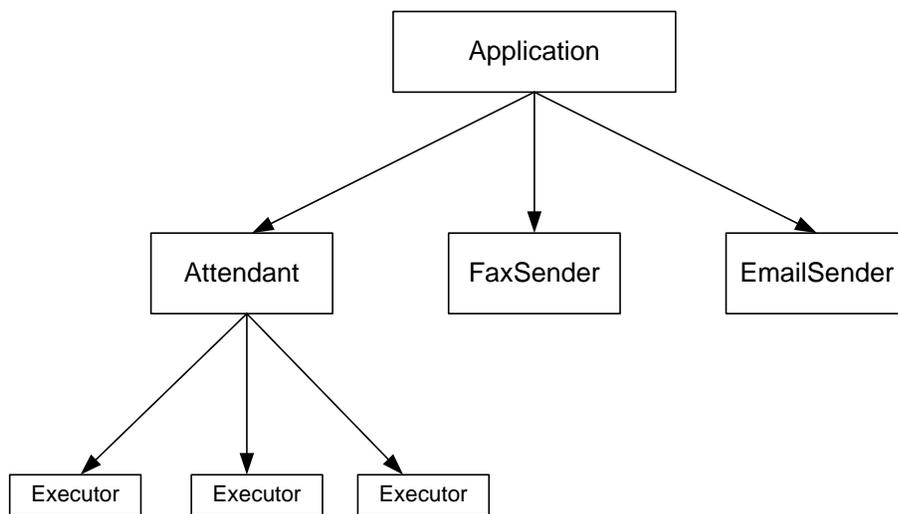
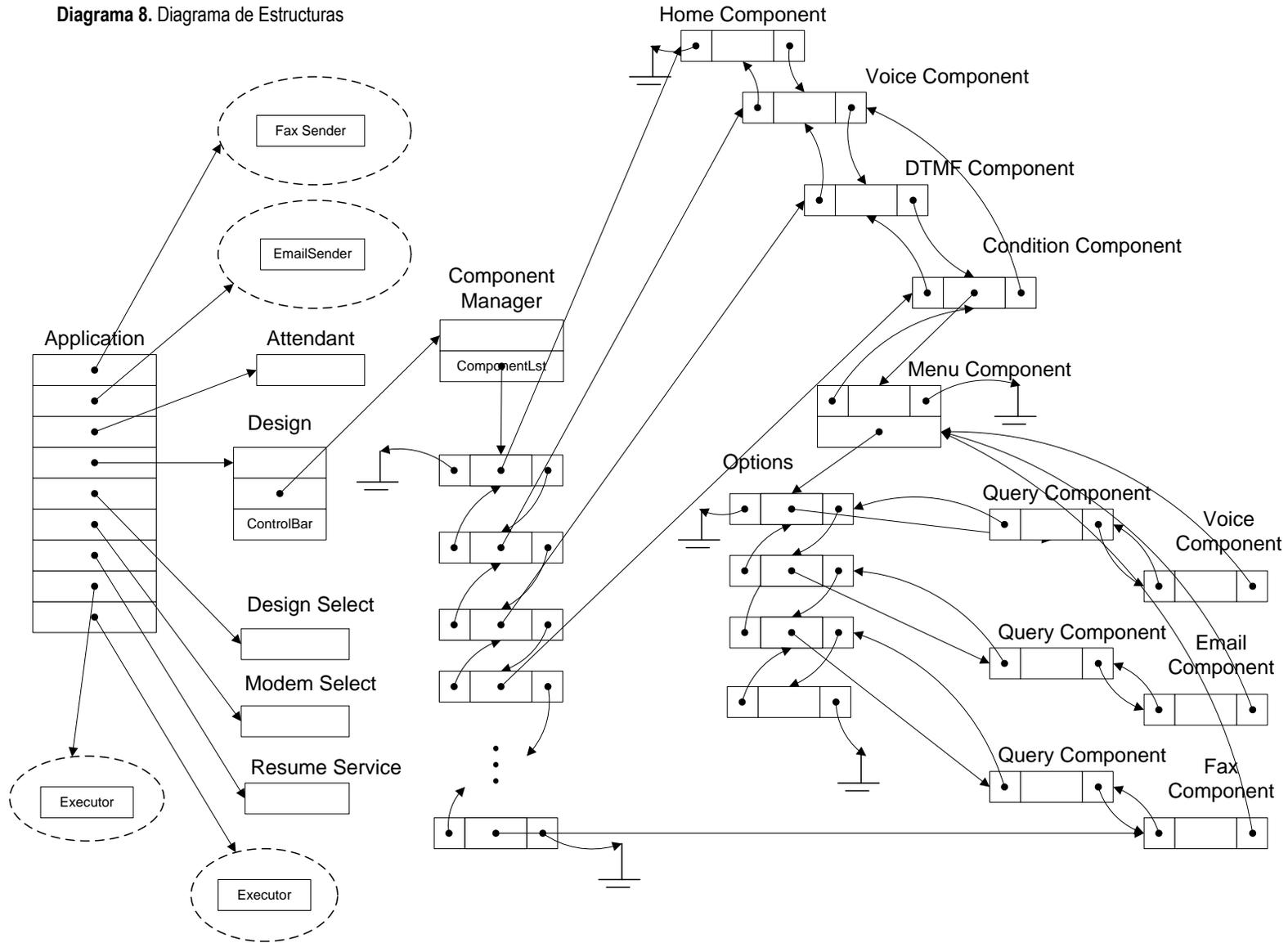


Diagrama 7. Diagrama de Hilos

Diagrama 8. Diagrama de Estructuras



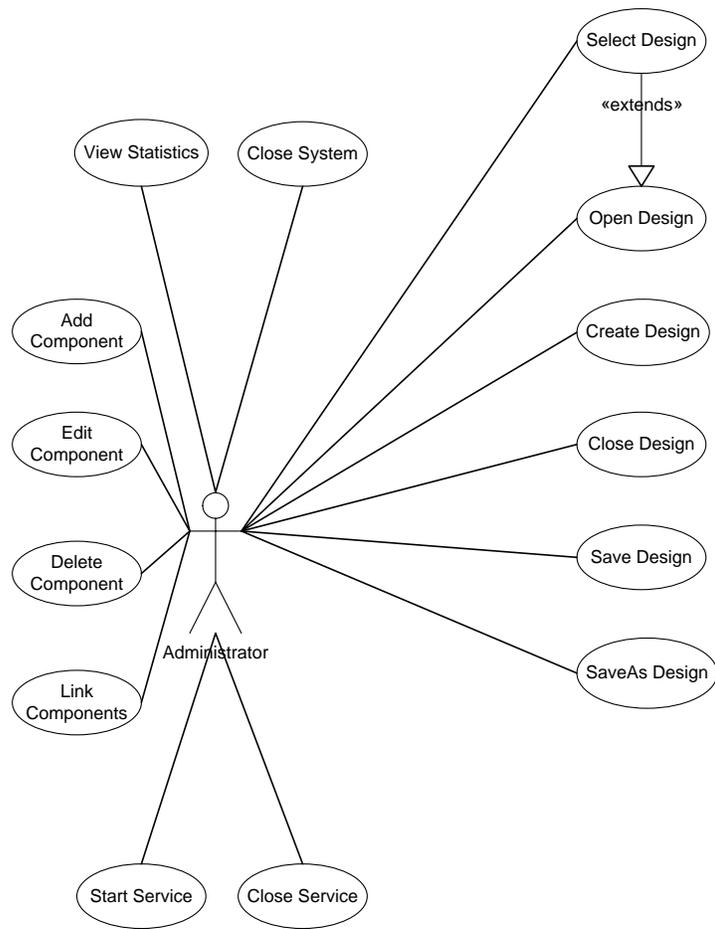


Diagrama 9. Casos de uso del Administrador

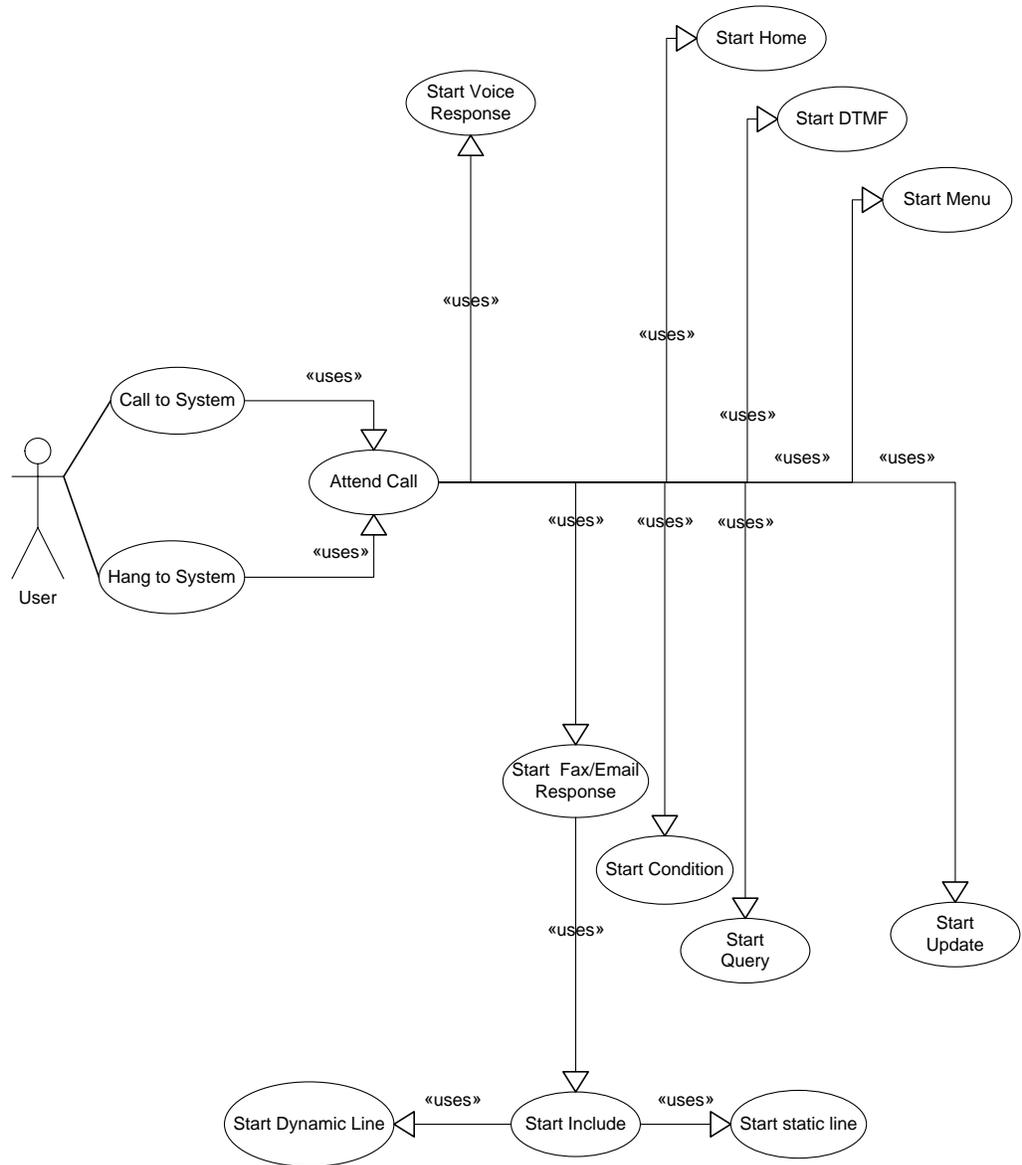


Diagrama 10. Casos de uso del Usuario

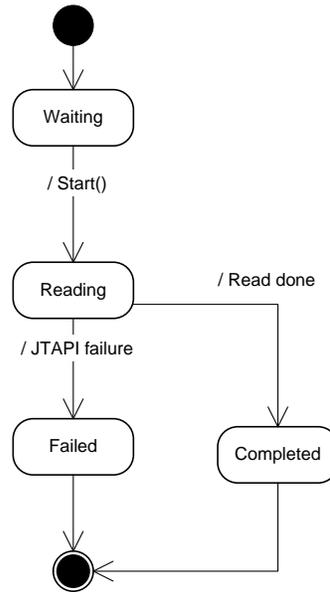


Diagrama 11. Diagrama de estado de DtmfComponent

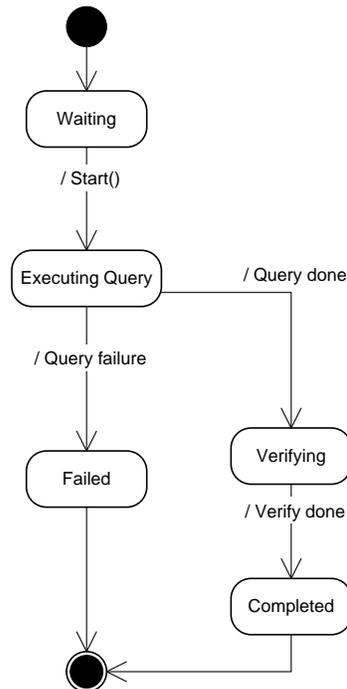


Diagrama 12. Diagrama de estado de ConditionComponent

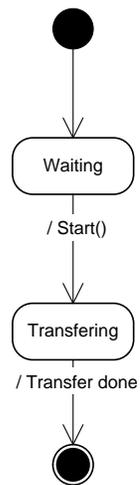


Diagrama 13. Diagrama de estado de HomeComponent

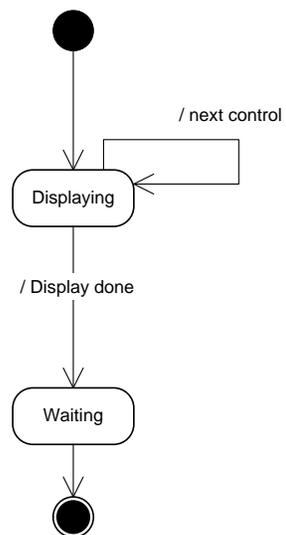


Diagrama 14. Diagrama de estado de ControlBar

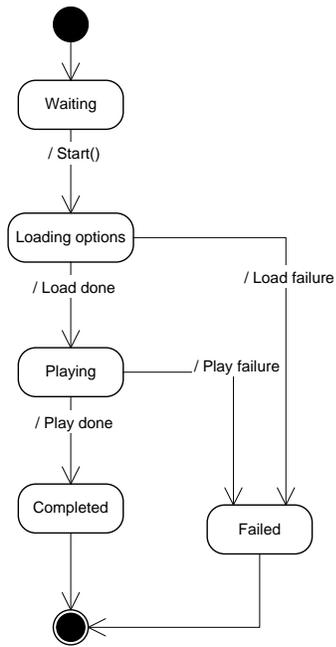


Diagrama 15. Diagrama de estado de MyMenuComponent

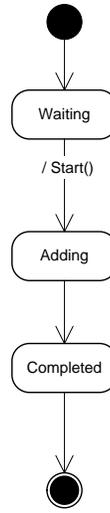


Diagrama 16. Diagrama de estado de StaticOption

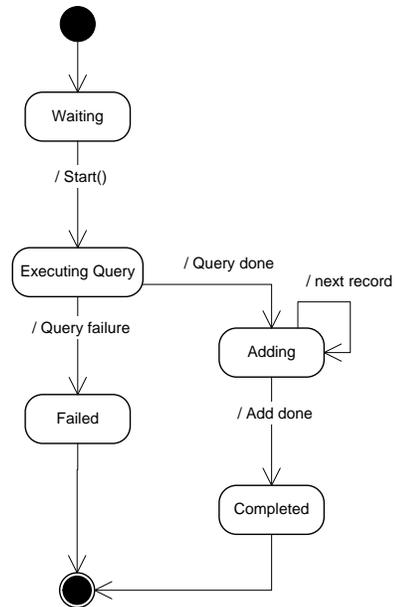


Diagrama 17. Diagrama de estado de DynamicOption

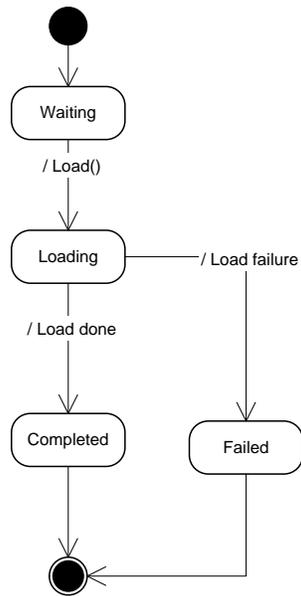


Diagrama 18. Diagrama de estado de Include

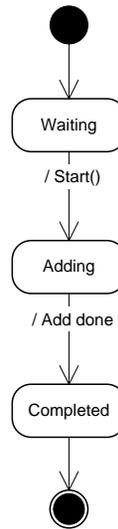


Diagrama 19. Diagrama de estado de StaticLine

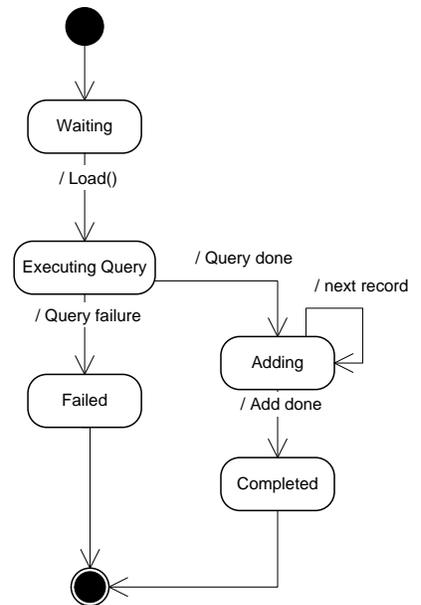


Diagrama 20. Diagrama de estado de DynamicLine

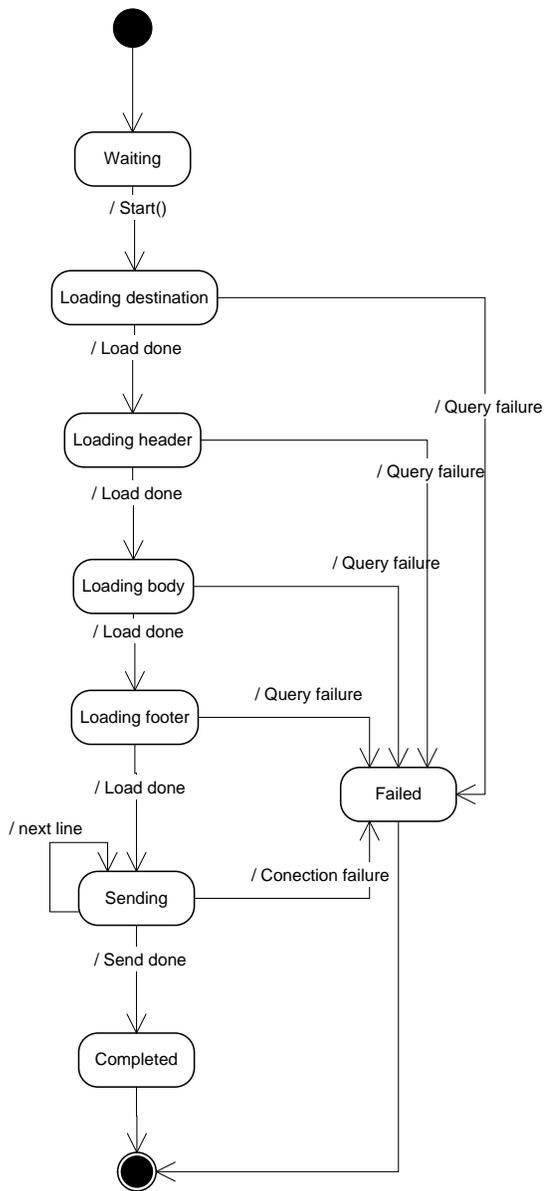


Diagrama 21. Diagrama de estado de FaxComponent, EmailComponent

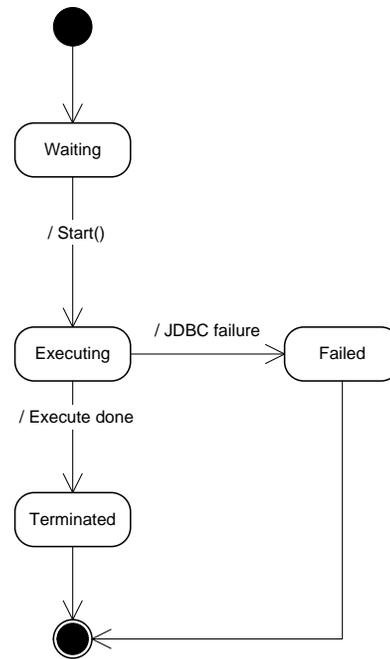


Diagrama 22. Diagrama de estado de QueryComponent, UpdateComponent

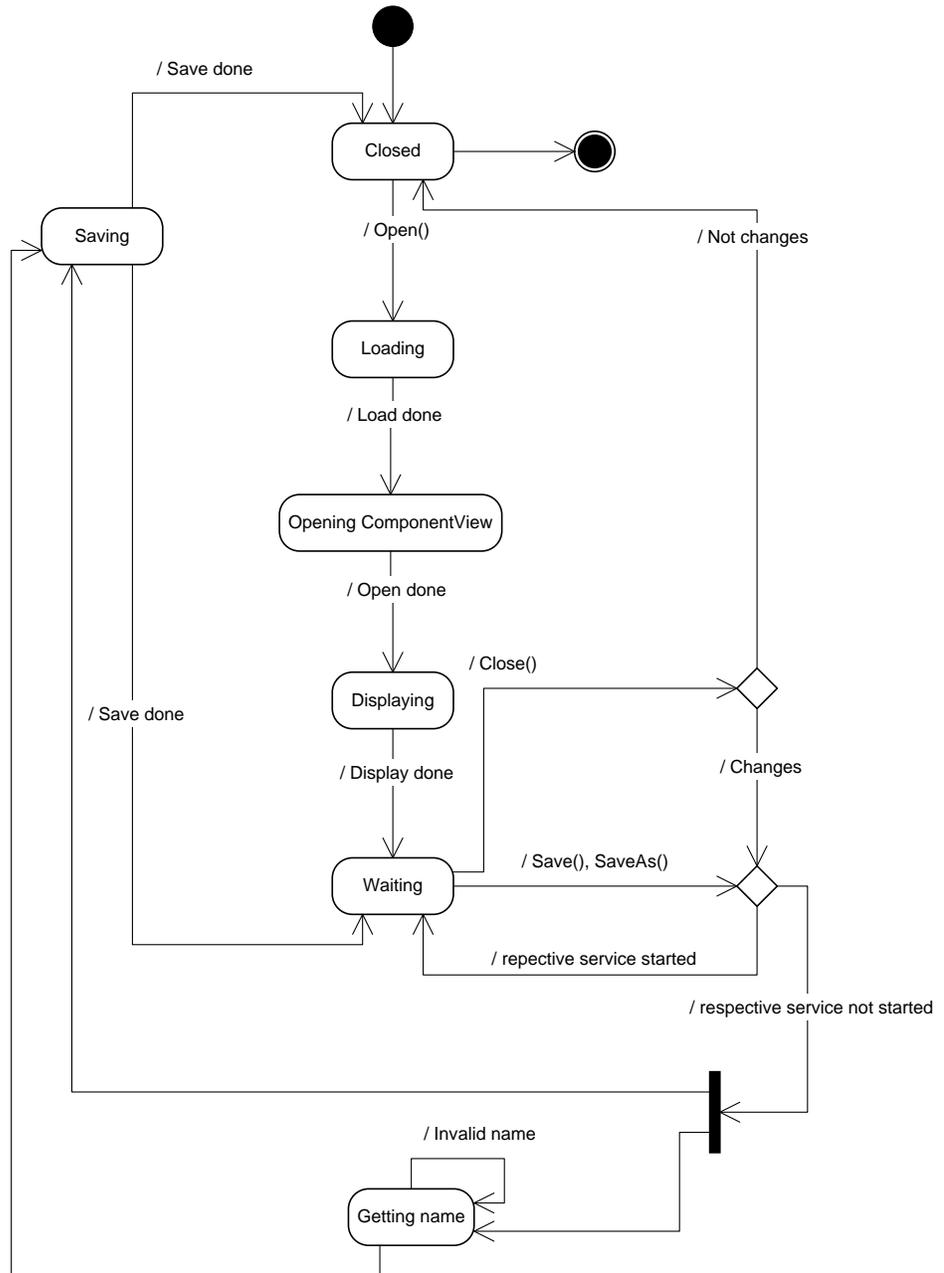


Diagrama 23. Diagrama de estado de Design

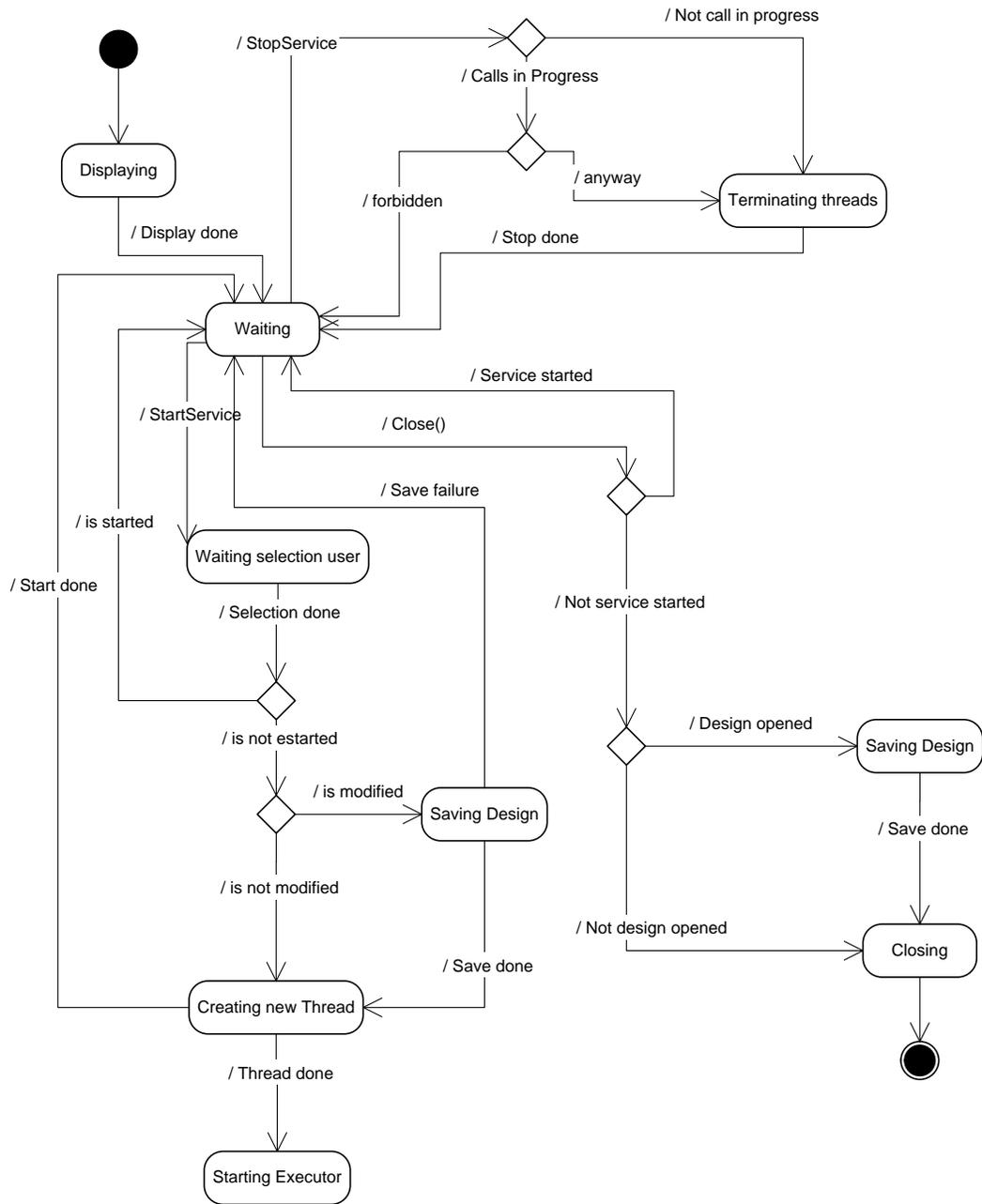


Diagrama 24. Diagrama de estado de MainWindow

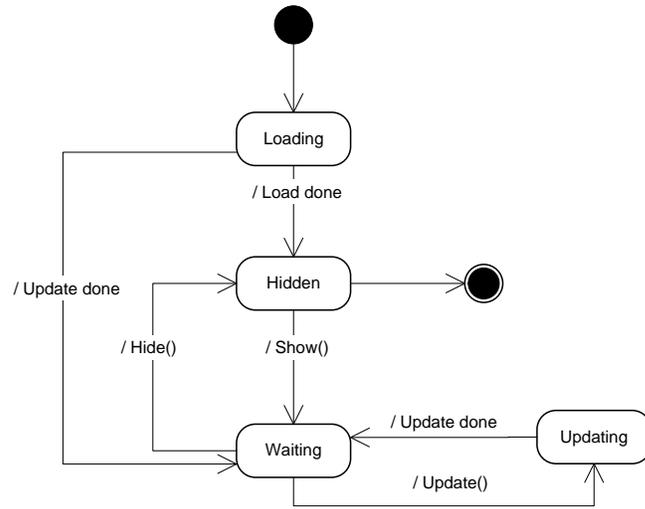


Diagrama 25. Diagrama de estado de ComponentManager

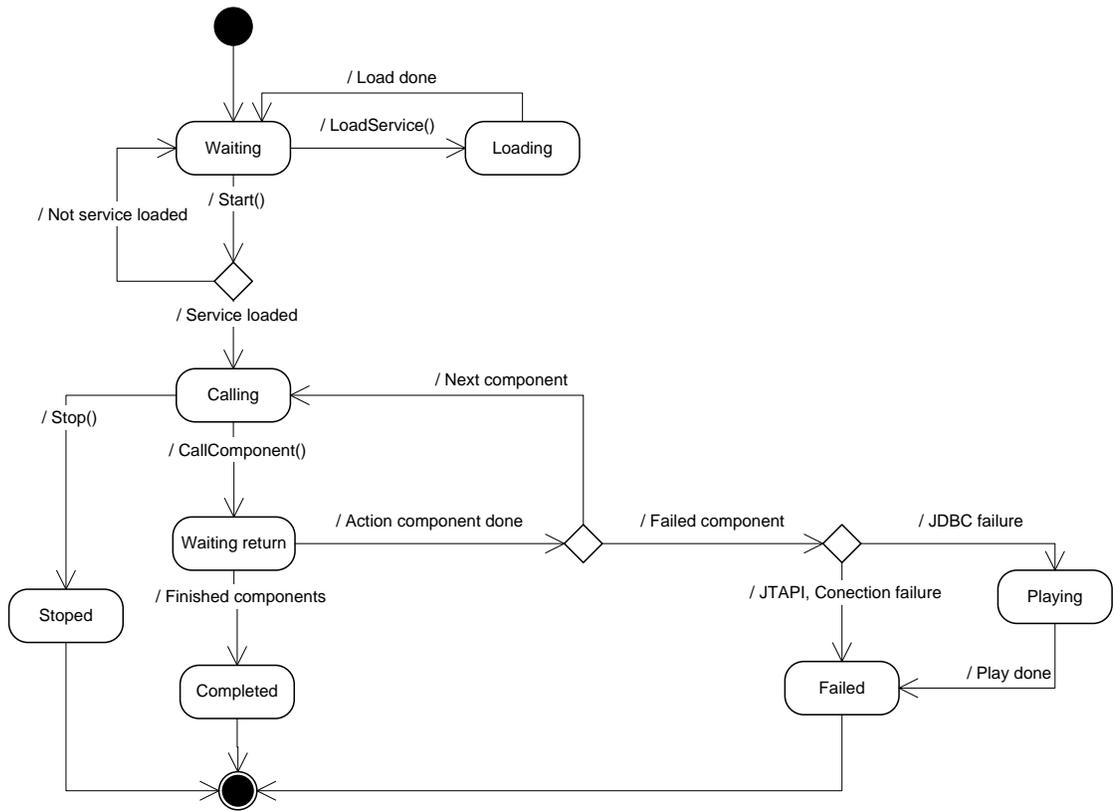


Diagrama 26. Diagrama de estado de Executor

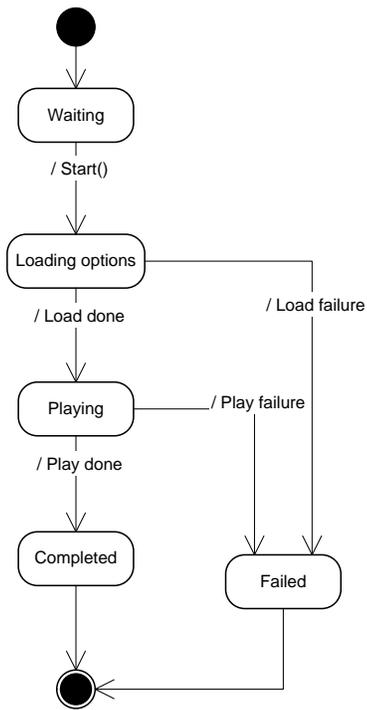


Diagrama 27. Diagrama de estado de VoiceComponent

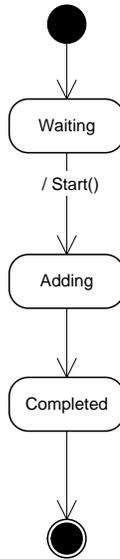


Diagrama 28. Diagrama de estado de StaticLine

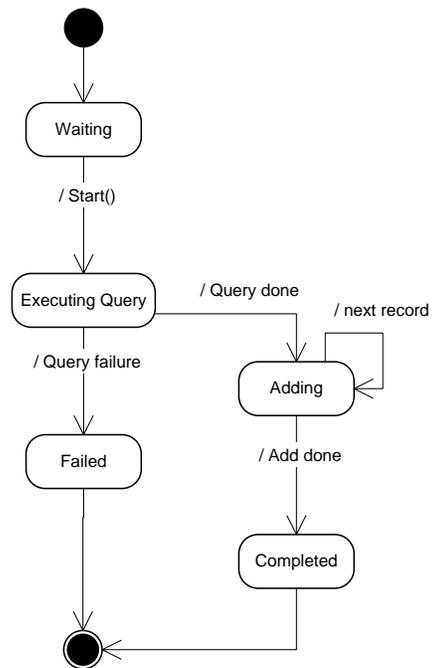


Diagrama 29. Diagrama de estado de DynamicLine

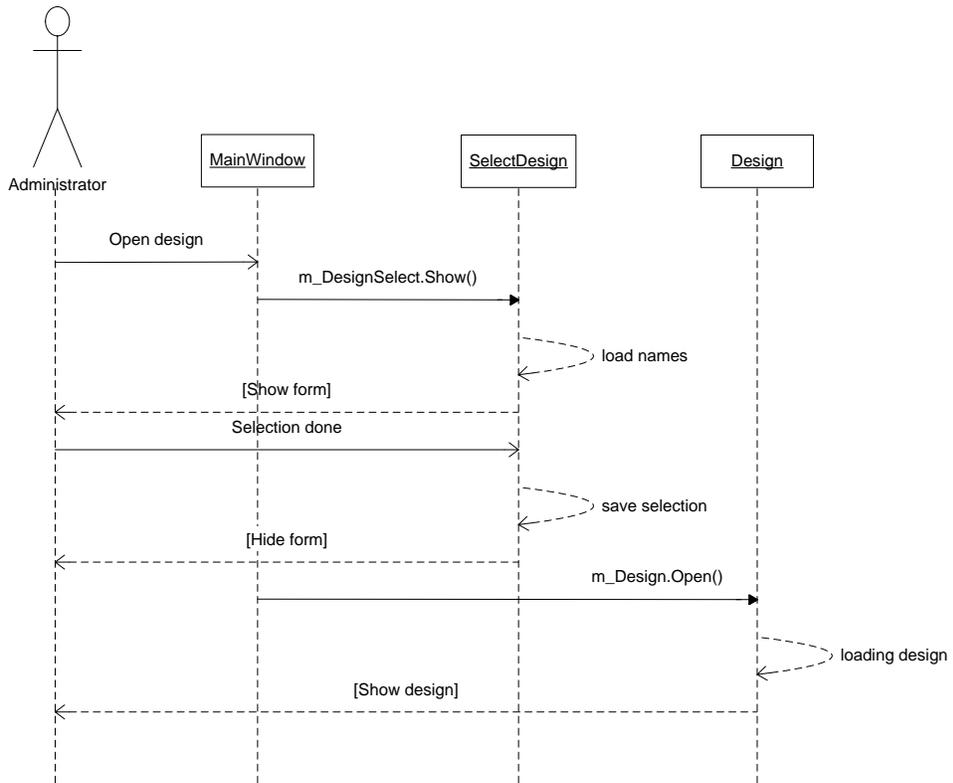


Diagrama 30. Diagrama de secuencia del caso Open Design

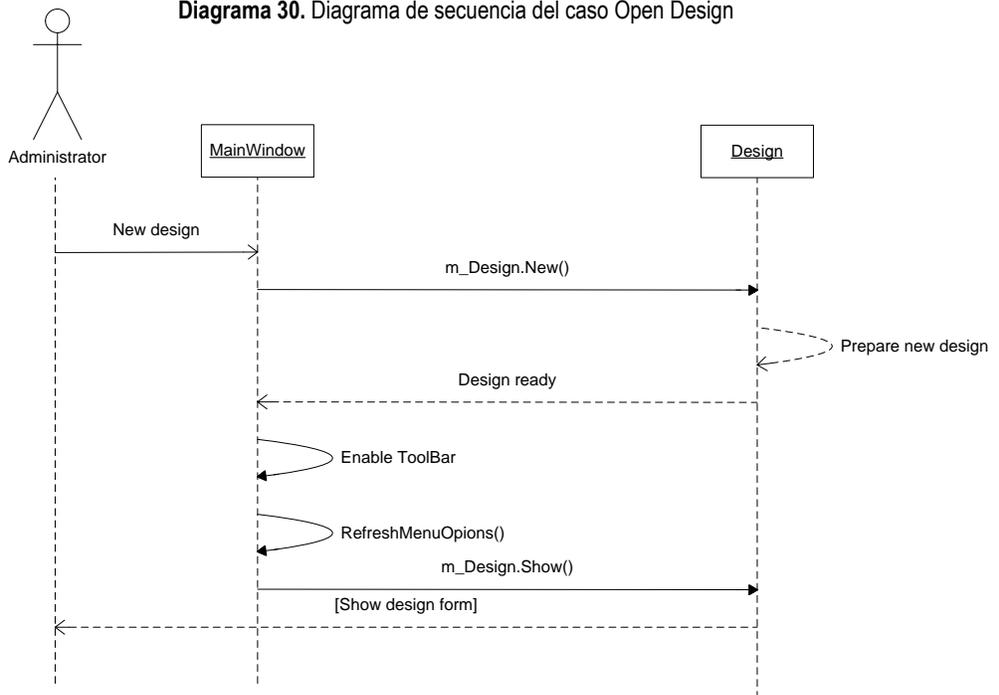


Diagrama 31. Diagrama de secuencia del caso Create Design

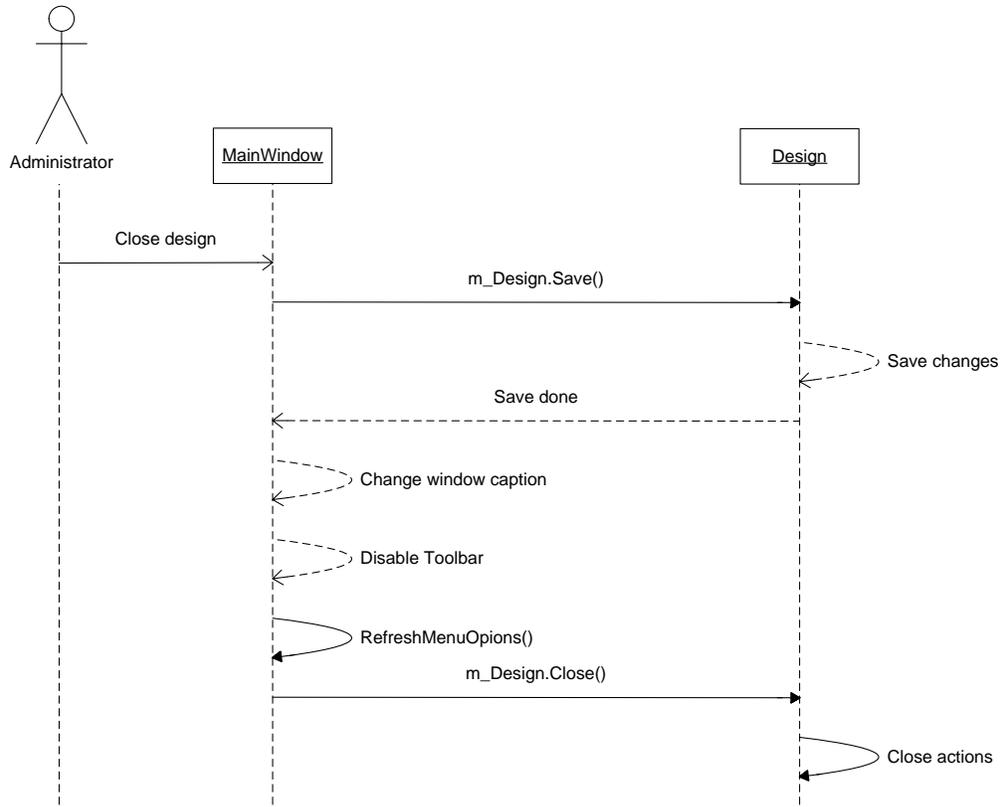


Diagrama 32. Diagrama de secuencia del caso Close Design

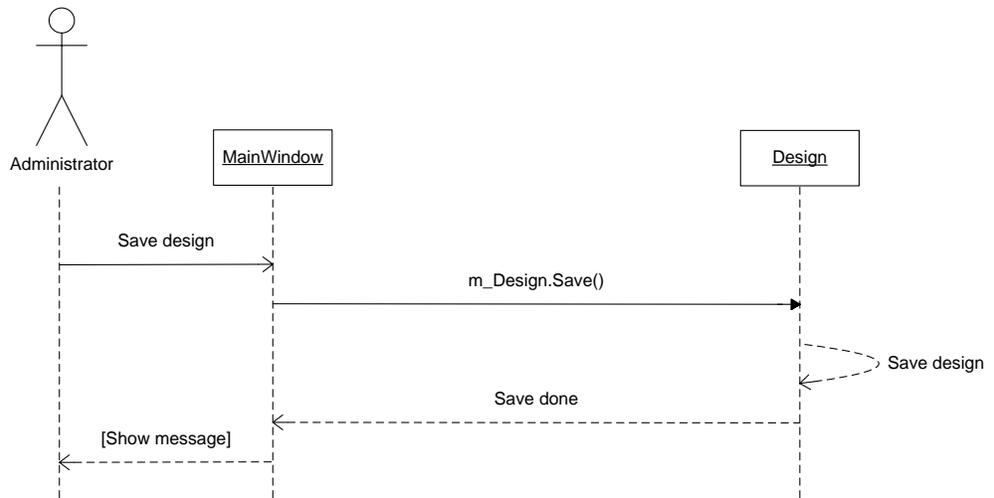


Diagrama 33. Diagrama de secuencia del caso Save Design (Design not running)

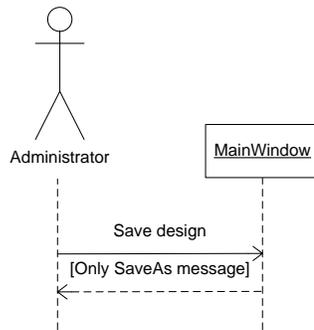


Diagrama 34. Diagrama de secuencia del caso Save Design (Design running)

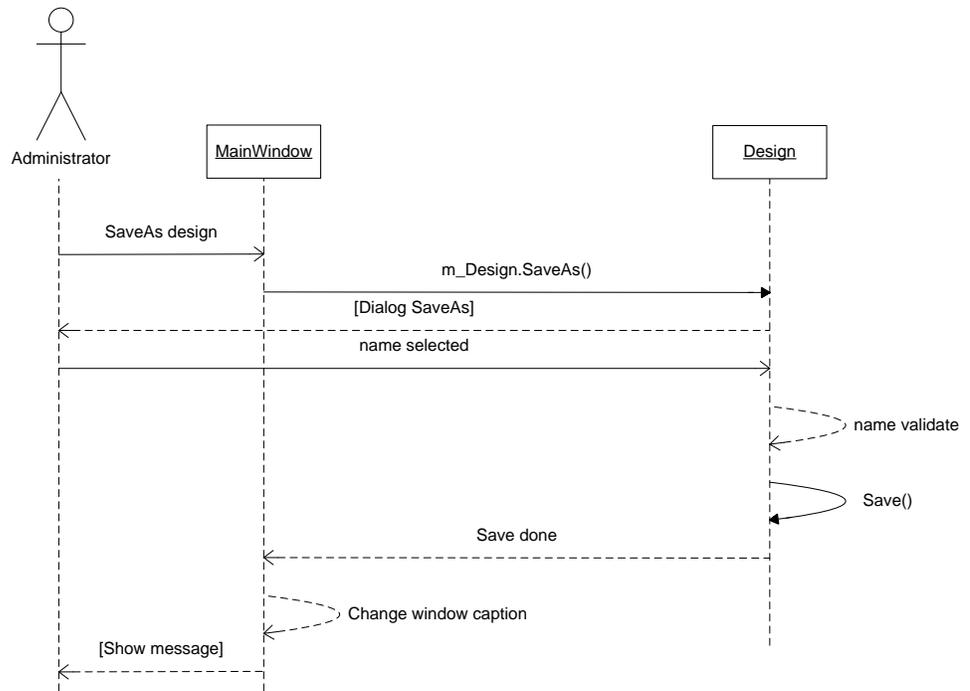


Diagrama 35. Diagrama de secuencia del caso Save as

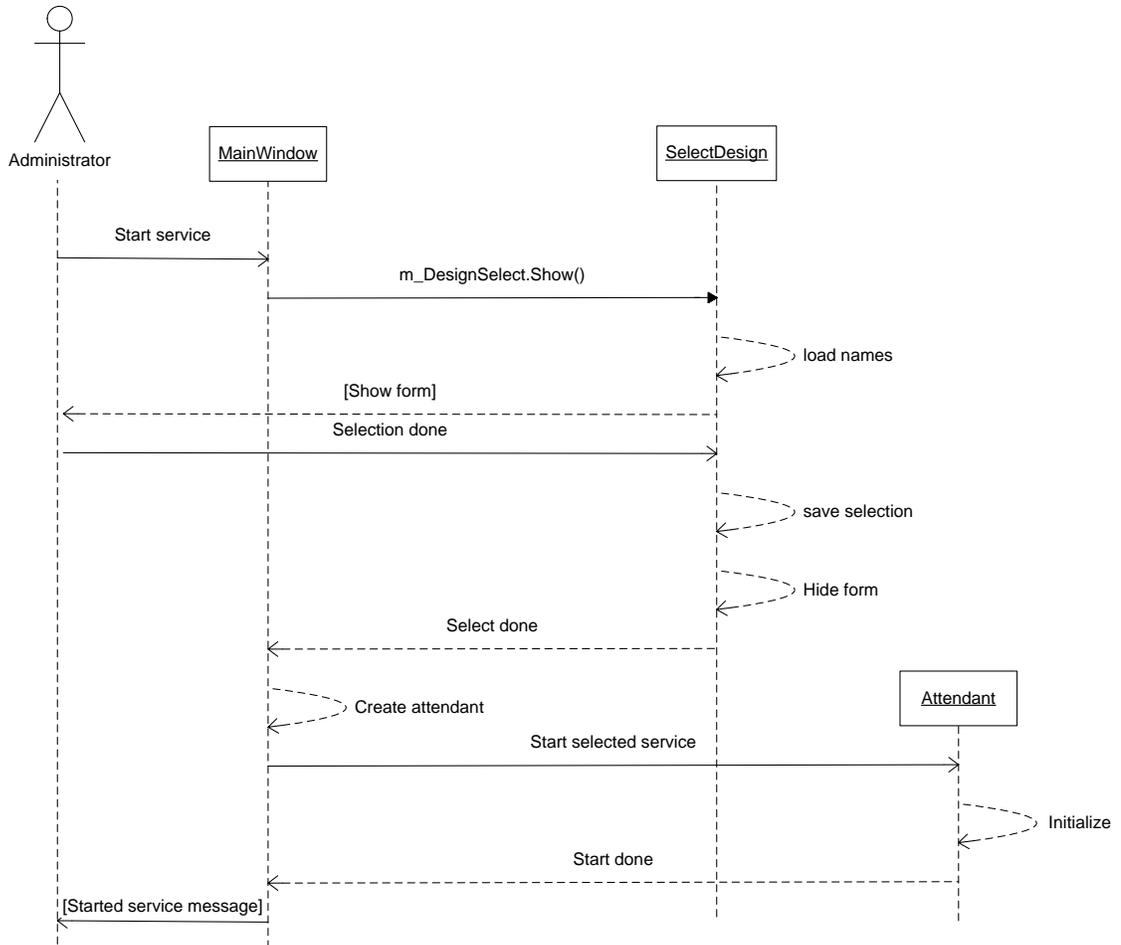


Diagrama 36. Diagrama de secuencia del caso Start Service (Other service not running)

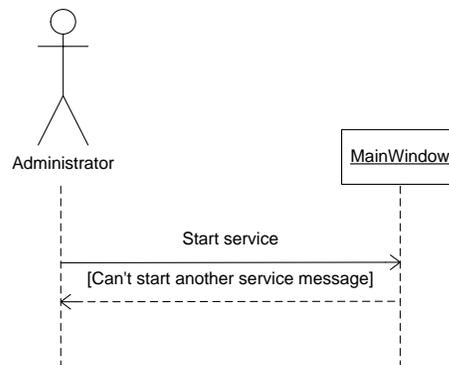


Diagrama 37. Diagrama de secuencia del caso Start Service (Other service running)

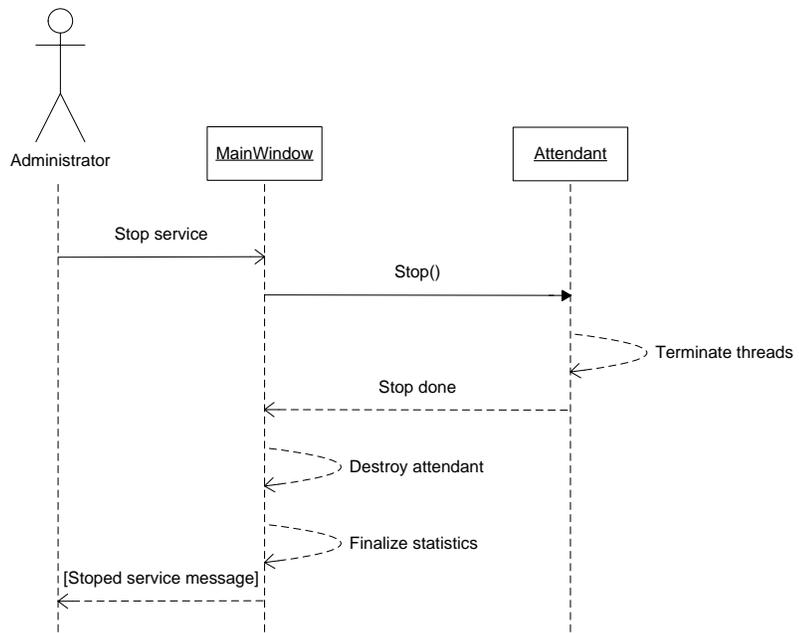


Diagrama 38. Diagrama de secuencia del caso Stop Service (Not calls in progress)

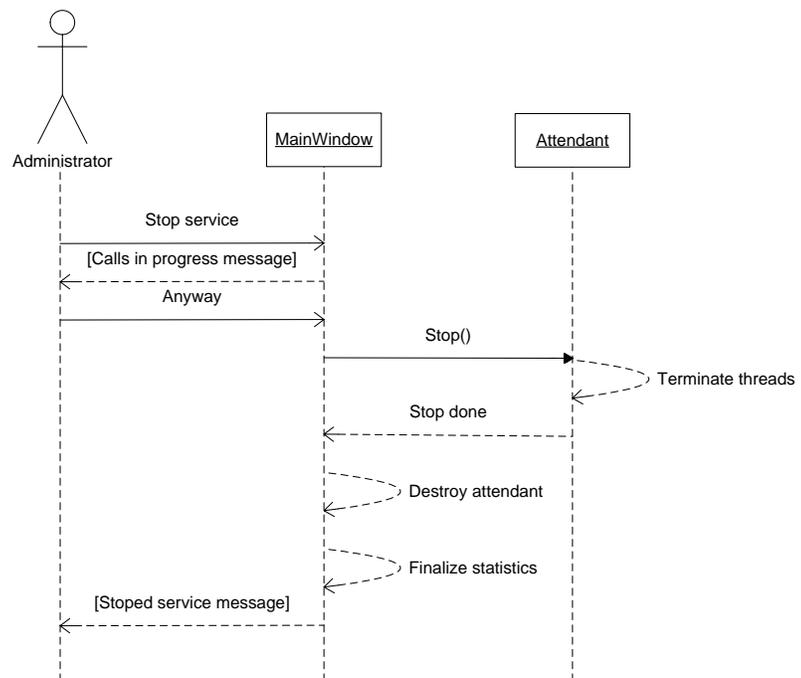


Diagrama 39. Diagrama de secuencia del caso Stop Service (Call in progress - anyway)

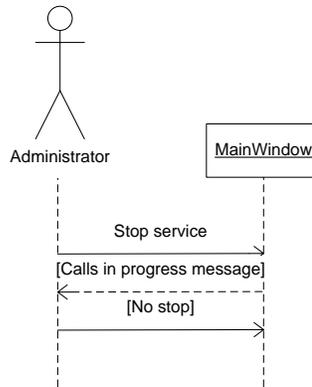


Diagrama 40. Diagrama de secuencia del caso Stop Service (Calls in progress - Not Stop)

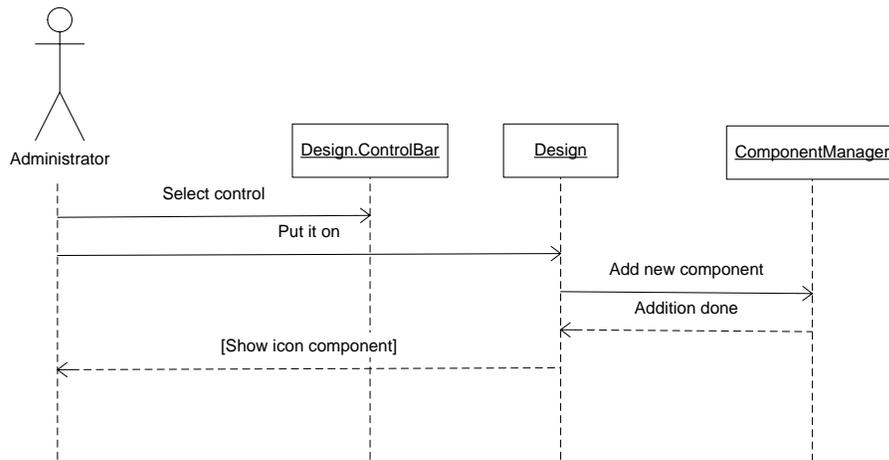


Diagrama 41. Diagrama de secuencia del caso Add component

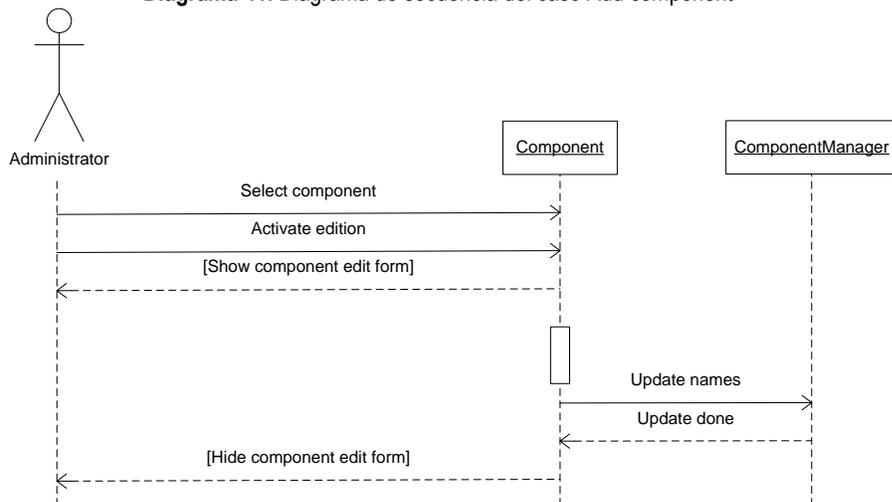


Diagrama 42. Diagrama de secuencia del caso Edit component

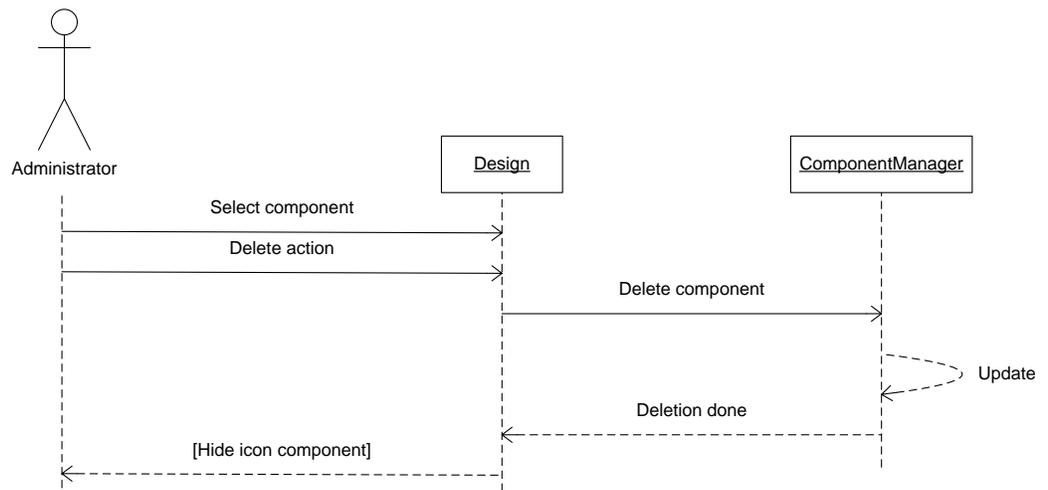


Diagrama 43. Diagrama de secuencia del caso Delete component

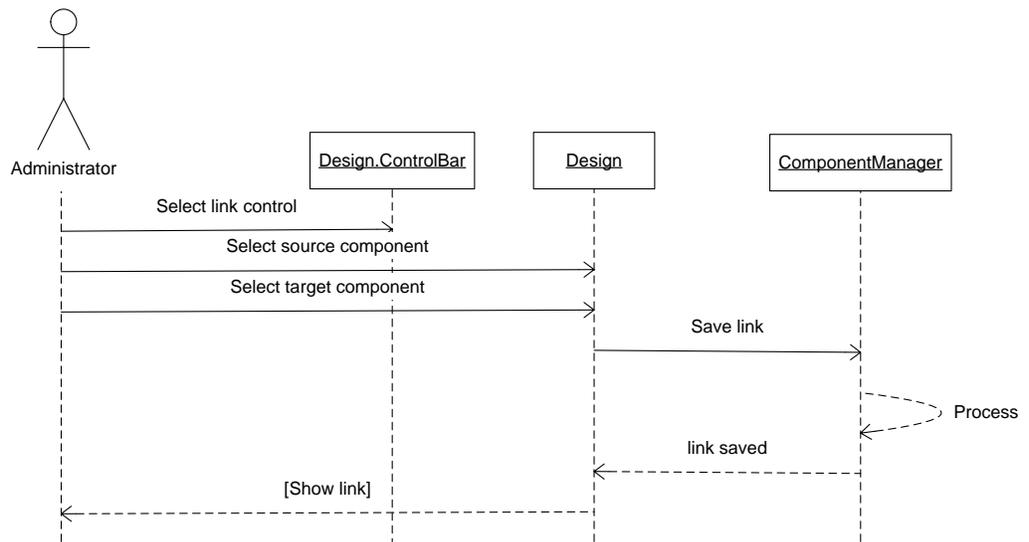


Diagrama 44. Diagrama de secuencia del caso Link component

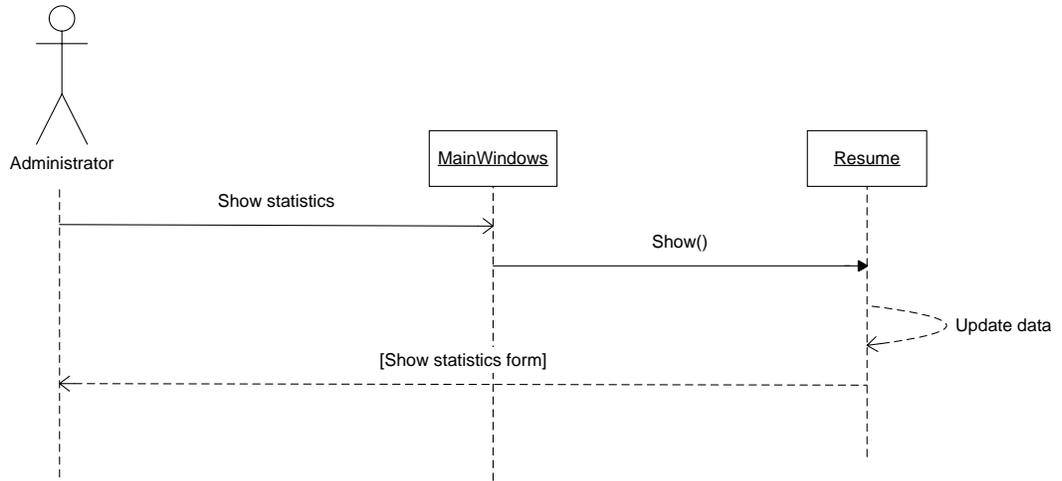


Diagrama 45. Diagrama de secuencia del caso Show Statistics

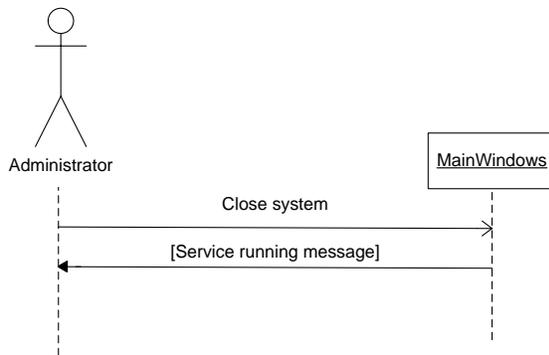


Diagrama 46. Diagrama de secuencia del caso Close System (Service running)

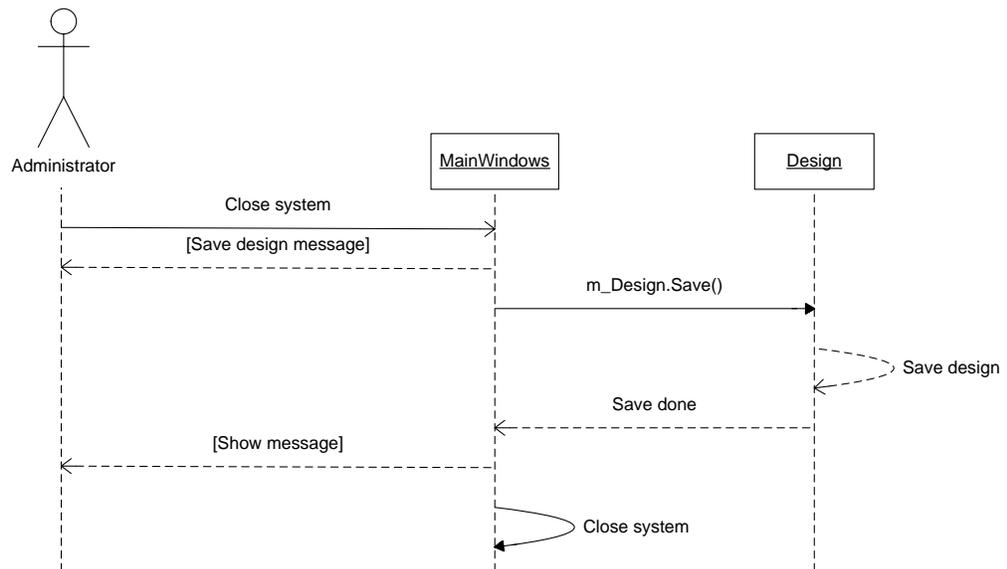


Diagrama 47. Diagrama de secuencia del caso Close system (Design Open)

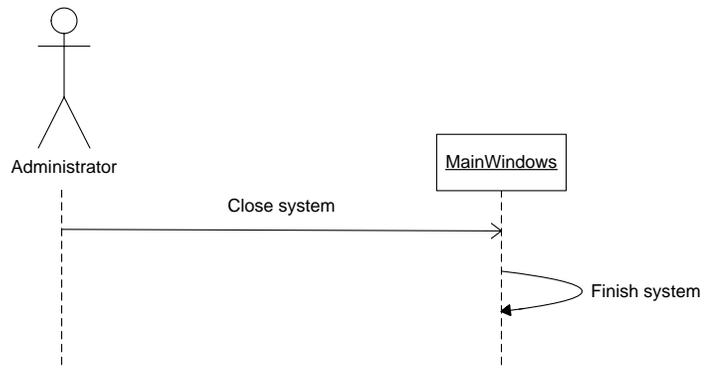


Diagrama 48. Diagrama de secuencia del caso Close system (Not constraints)

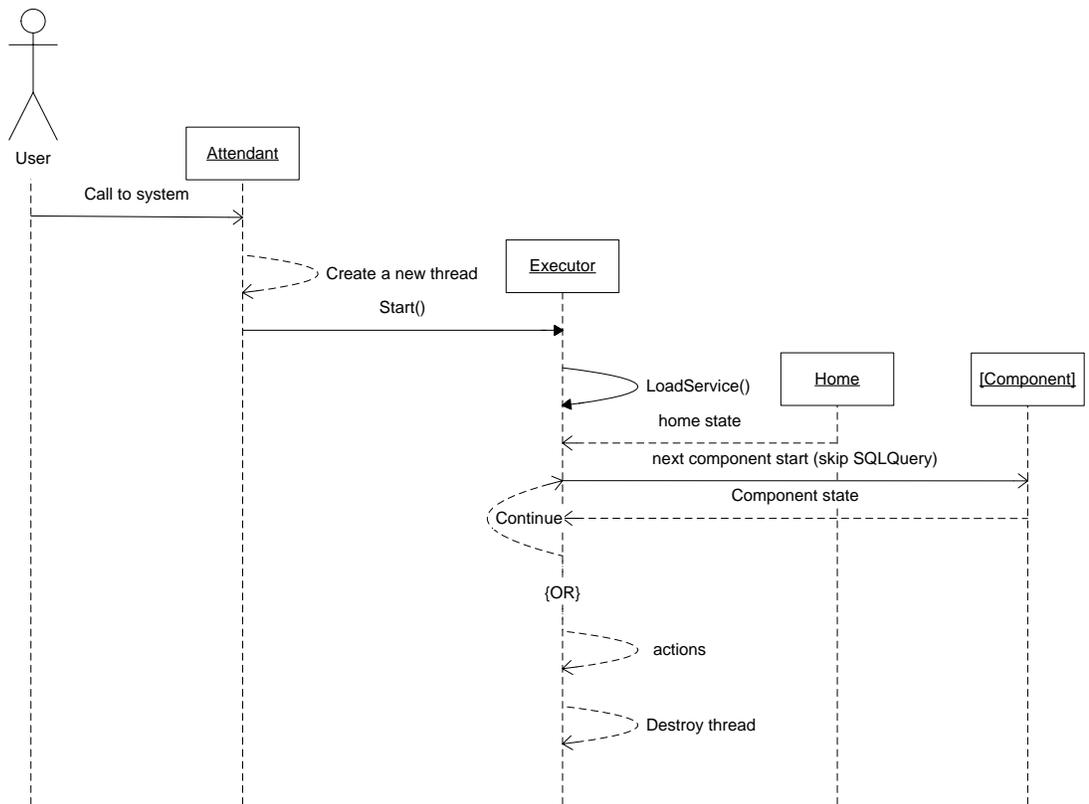


Diagrama 49. Diagrama de secuencia del caso Attend call

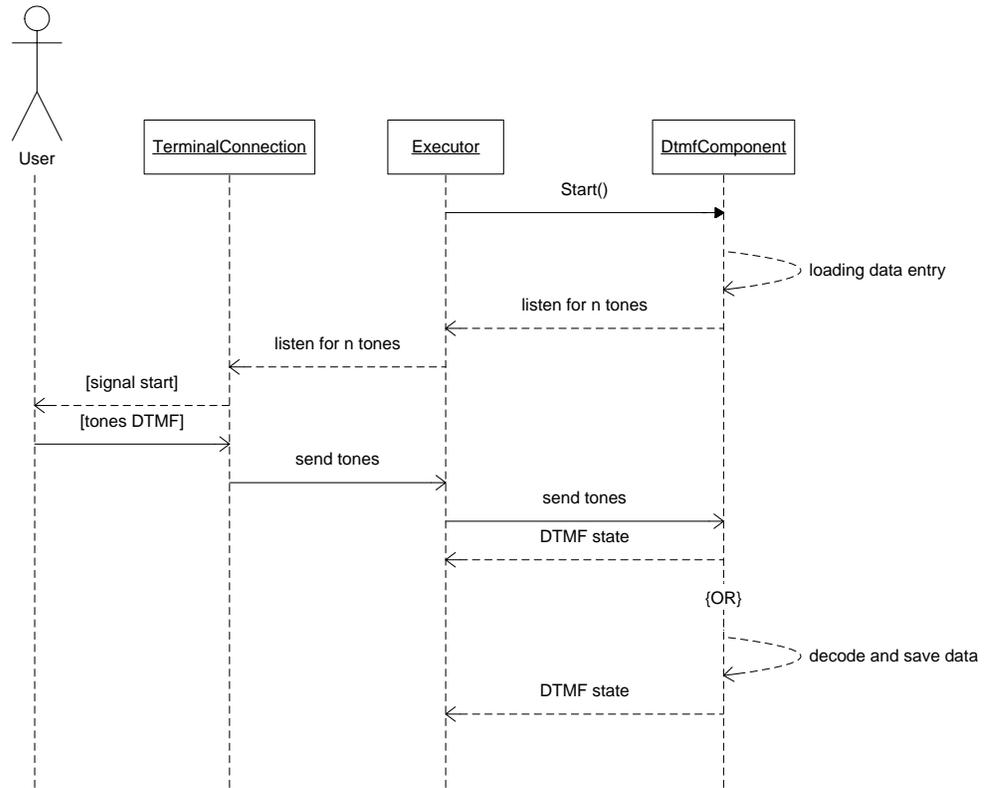


Diagrama 50. Diagrama de secuencia del caso Start DtmfComponent

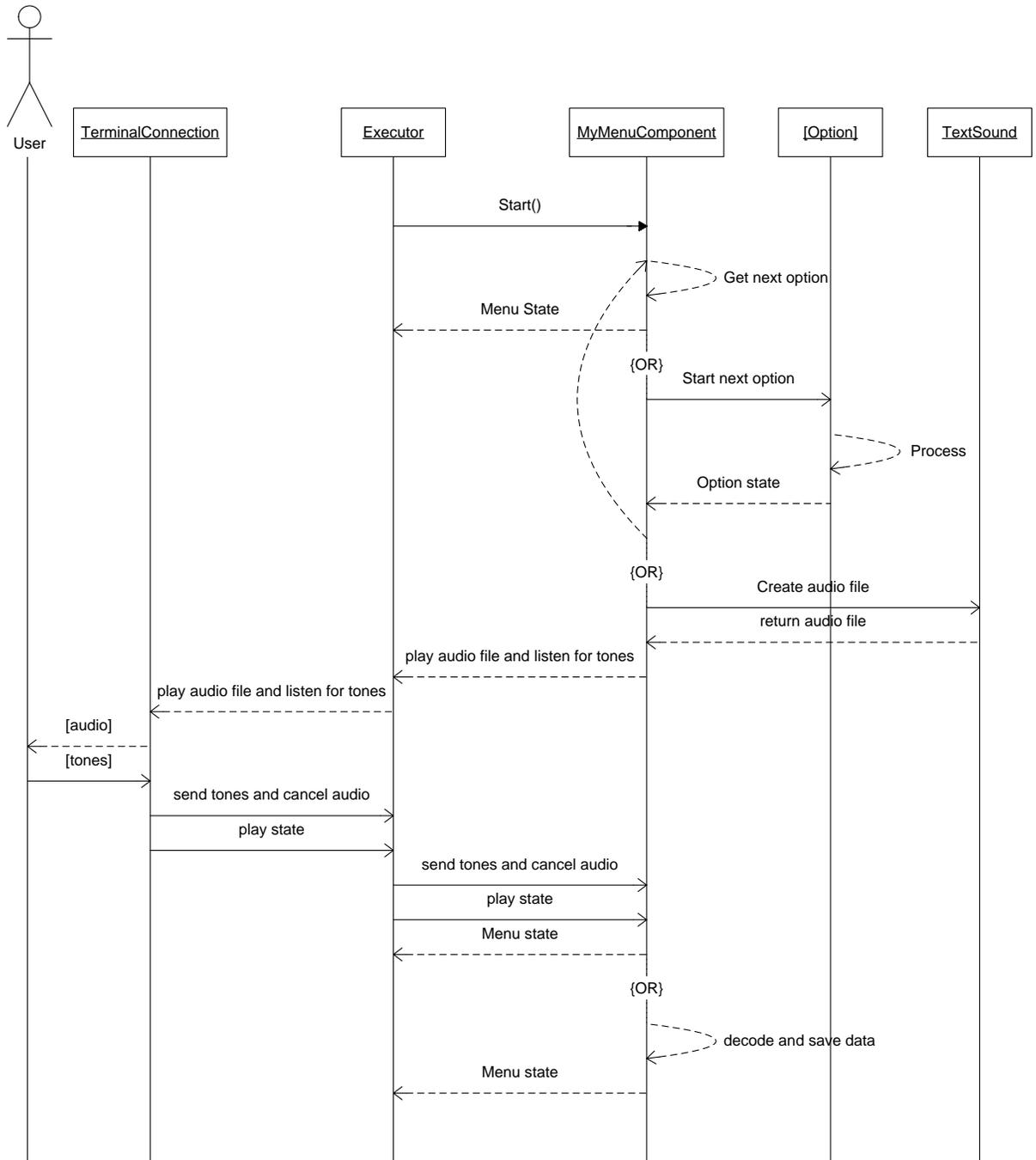


Diagrama 51. Diagrama de secuencia del caso Start MyMenuComponent

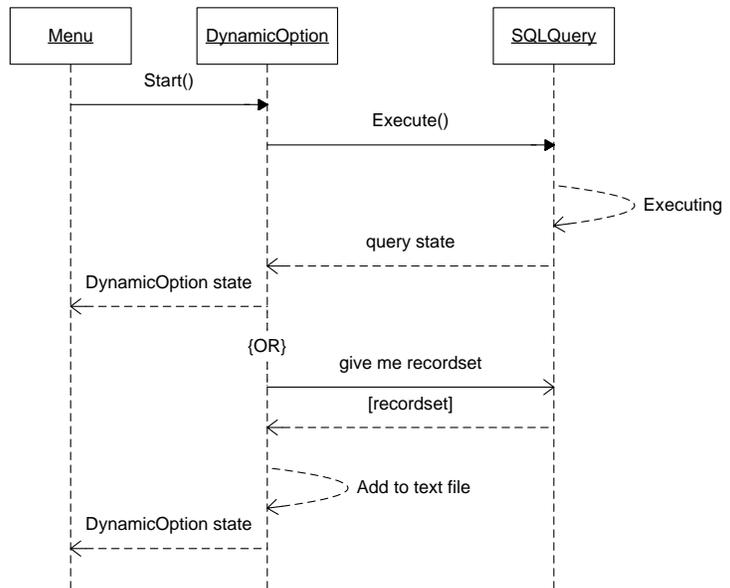


Diagrama 52. Diagrama de secuencia del caso Start Dynamic Option

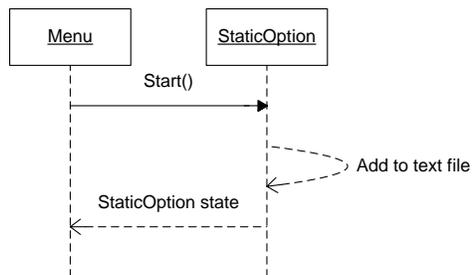


Diagrama 53. Diagrama de secuencia del caso Start Static option)

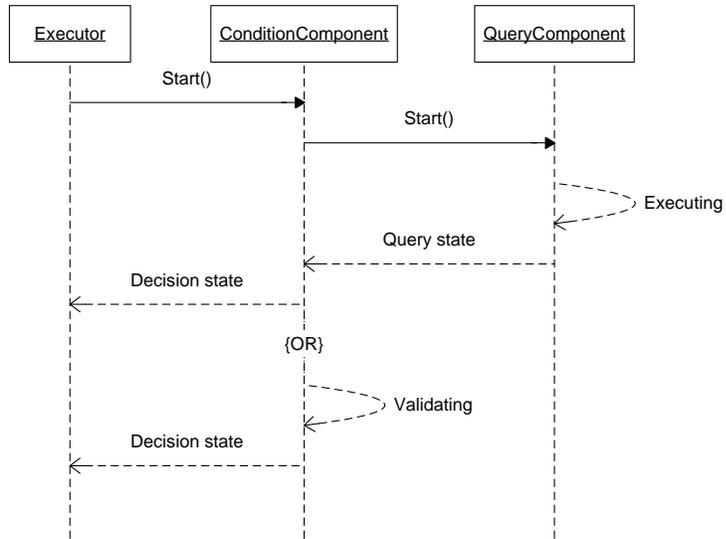


Diagrama 54. Diagrama de secuencia del caso Start Condition)

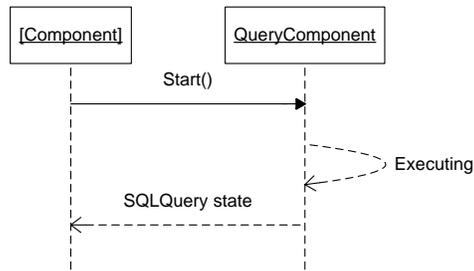


Diagrama 55. Diagrama de secuencia del caso Start QueryComponent)

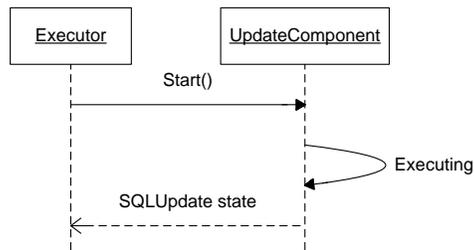


Diagrama 56. Diagrama de secuencia del caso Start UpdateComponent)

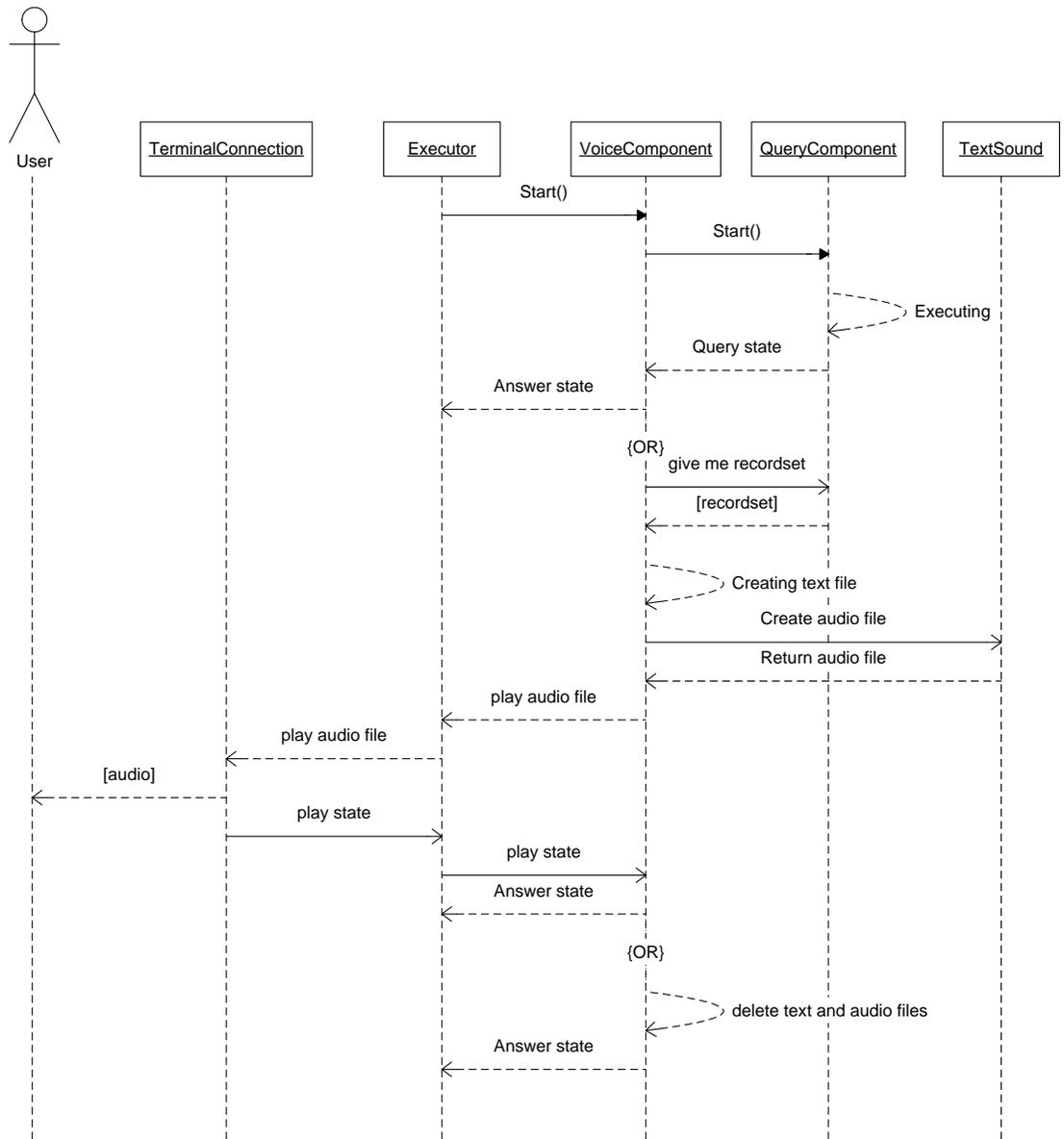


Diagrama 57. Diagrama de secuencia del caso Start VoiceComponent)

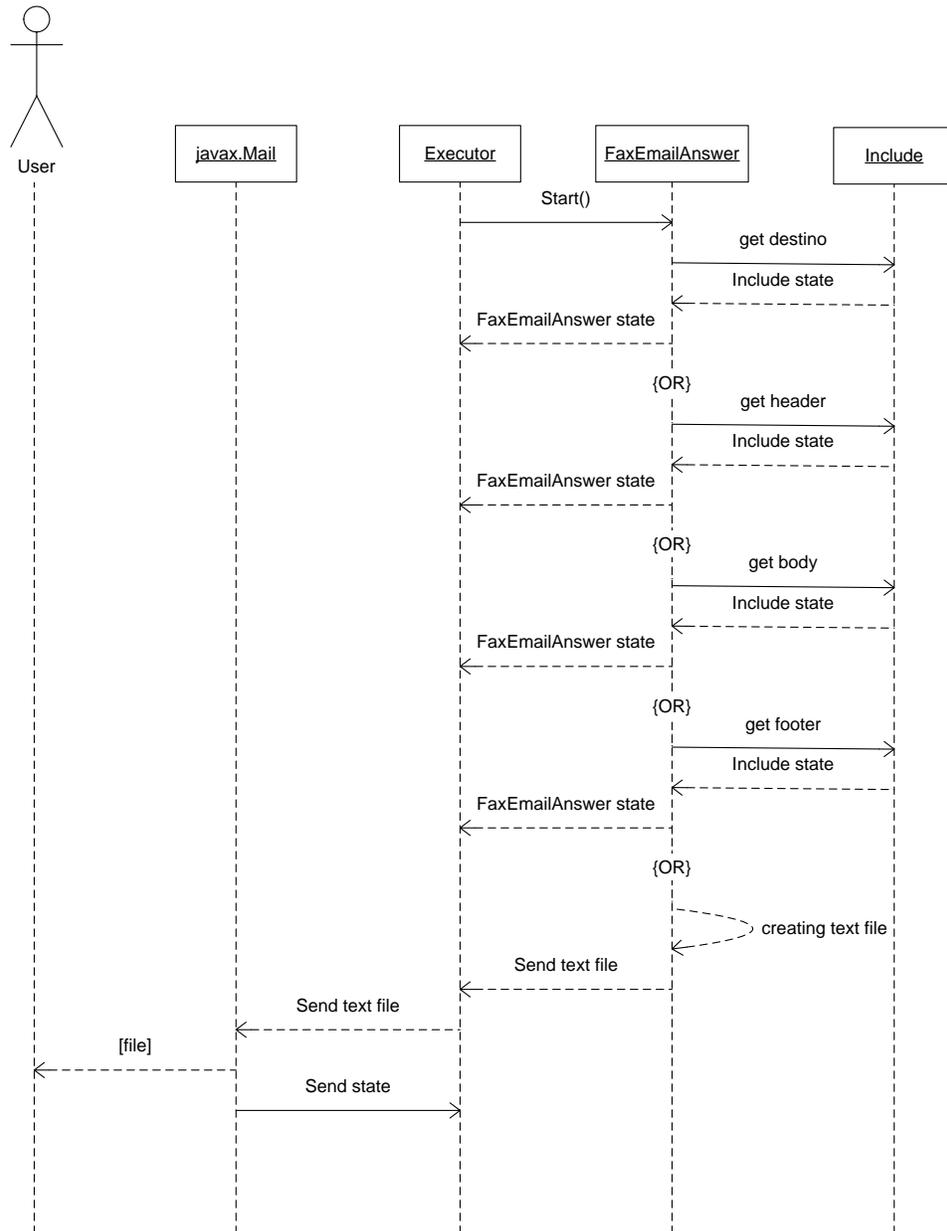


Diagrama 58. Diagrama de secuencia del caso Start FaxComponent/EmailComponent)

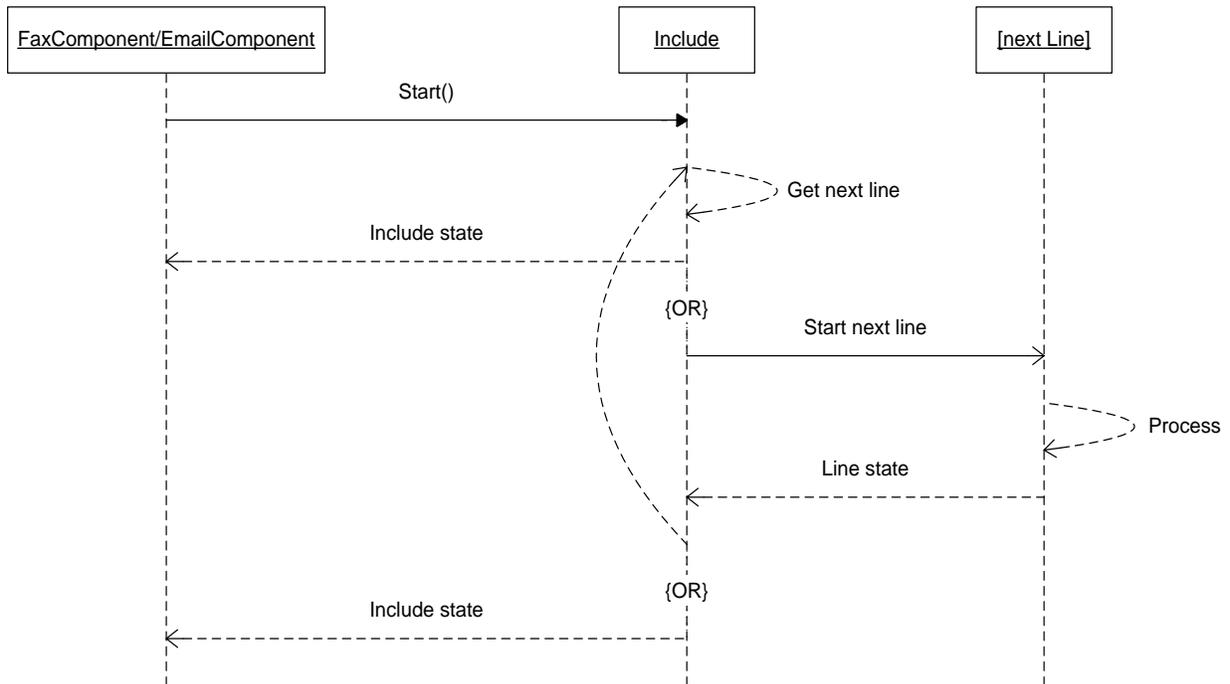


Diagrama 59. Diagrama de secuencia del caso Start Include

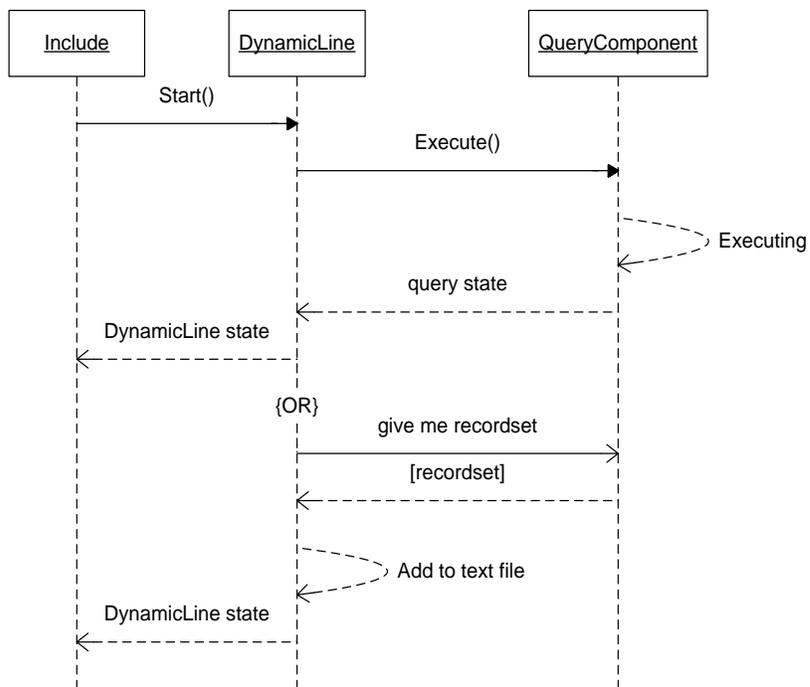


Diagrama 60. Diagrama de secuencia del caso Start DynamicLine

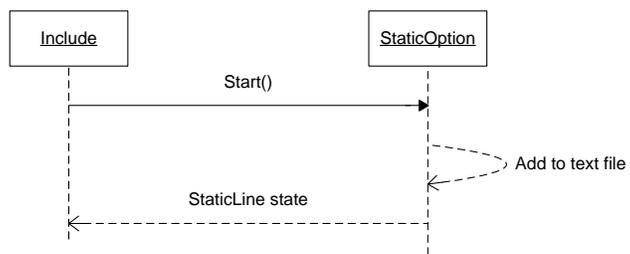


Diagrama 61. Diagrama de secuencia del caso Start StaticLine)

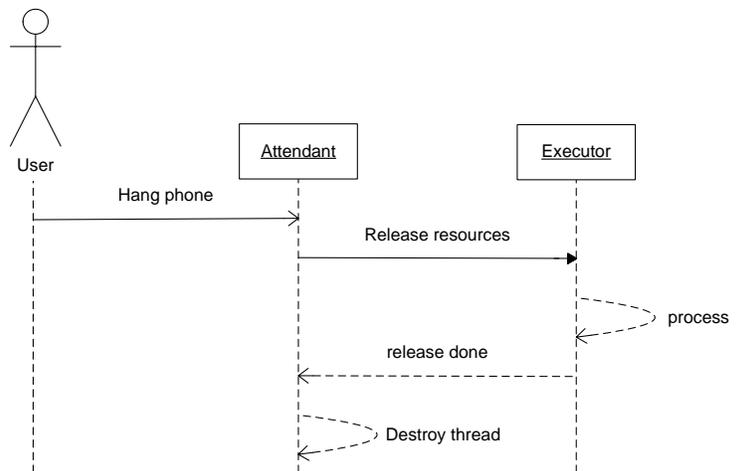


Diagrama 62. Diagrama de secuencia del caso User Hang to system)

3. DESARROLLO DE LA SOLUCIÓN

3.1 Requerimientos mínimos de Hardware

Una computadora con:

- a. 1 CPU 500Mhz.
- b. 1 fax/voice/MODEM
- c. 128 MB RAM
- d. 30 MB HD

3.2 Requerimientos de Software

Windows 98/Me/2000/NT4/XP

3.3 Software de instalación

Adjunto se incluye un CD con todos los programas necesarios para instalar el sistema.

El CD de instalación incluye:

1. Guía de instalación.
2. XTAPI 2.0
3. TextSound 2.0
4. L&H TTS3000 Español
5. JComm 2.0 for Windows
6. RFax 1.0
7. JavaMail 1.3
8. JavaBeans Activation Framework 1.0.2
9. Sistema IVR 1.0

4. FUNCIONAMIENTO DEL SISTEMA

Una vez elaborados los diseños, el sistema permite almacenarlos en el disco. Esta operación se realiza por medio del proceso de **Serialización**, el cual guarda el estado actual de los componentes pertenecientes al servicio diseñado para su posterior recuperación (deserialización), cuando el diseño elaborado sea seleccionado para la ejecución.

Cuando un servicio ha sido seleccionado para su correspondiente ejecución, el sistema IVR es capaz de atender llamadas, siguiendo una secuencia bien definida dependiendo de la configuración y los recursos configurados para la aplicación.

Es de notar, que el sistema puede operar correctamente aun y cuando solo se cuente con una sola interfase hacia la línea telefónica (VoiceModem).

Para ilustrar mejor dicho comportamiento a continuación se describe cada uno los pasos que el sistema realiza, teniendo como base que el sistema cuenta con 2 interfaces hacia líneas telefónicas.

1. En el momento en que se da por iniciado un servicio se crea un agente basado en la clase Attendant. Este es el encargado de verificar el estado de las interfaces hacia la red telefónica (Fig. 39).

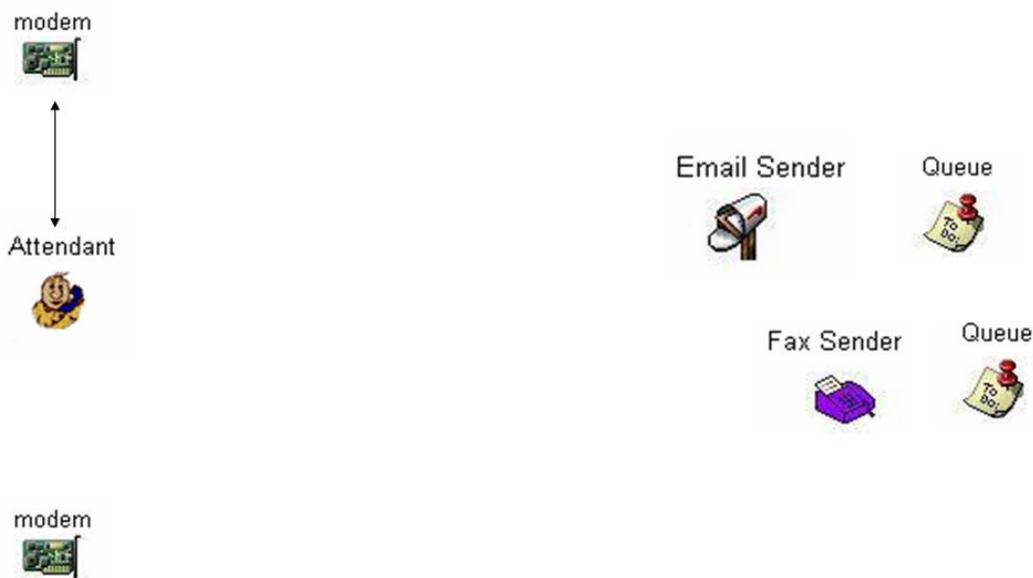


Figura 39. Objeto Attendant recibiendo una llamada.

- Si se recibe una llamada, el Attendant atiende esa llamada, notifica a los agentes MailSender y FaxSender que la interfaz se encuentra ocupada, y crea un objeto basado en la clase Executor indicándole el nombre del servicio que se esta brindando en este momento (Fig. 40).

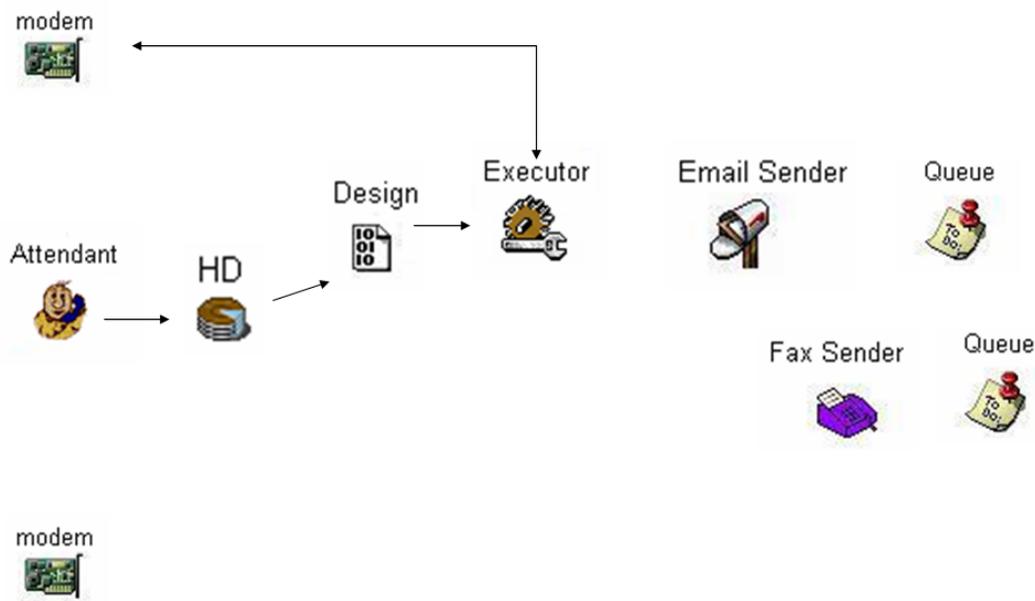


Figura 40. Creación del objeto Executor

- El Executor obtiene una copia de este diseño, **Deserializando** dicho servicio, para proceder a su ejecución
- El sistema es capaz de enviar mensajes a través de la línea telefónica, estos mensajes son elaborados en tiempo real utilizando el método de sinterización de palabras. Dichos mensajes están almacenados en archivos con formato Wav codificados a 8.000 Khz., 8 BIT, mono 7kb/s u law CCITT. Mediante a dichos mensajes y a tonos DTMF, el sistema IVR es capaz de interactuar con el usuario.
- El sistema también es capaz de acceder a diferentes bases de datos utilizando los DSN disponibles en el sistema, permitiendo así poder consultar y actualizar datos de manera automática. Si en el servicio en ejecución existe un QueryComponent o un UpdateComponet, estos acceden a las base de datos en tiempo real, permitiendo un acceso a los datos en línea.

6. Suponiendo que el servicio cuente con los servicios de respuesta a través de fax o correo electrónico, el sistema verifica las interfaces disponibles para poder utilizar otra interfaz, por ejemplo, para conectarse a Internet y enviar una respuesta a través de correo electrónico (Fig. 41).

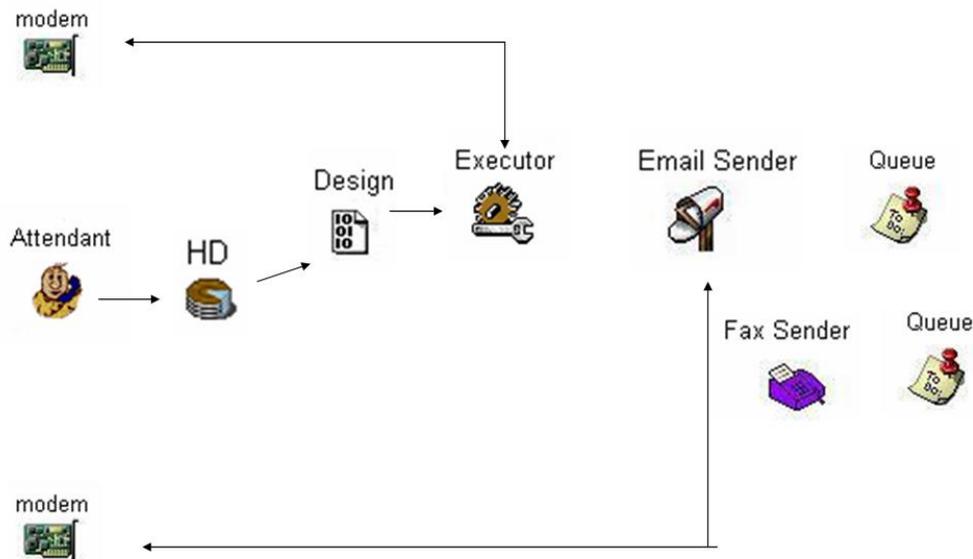


Figura 41. Envío de Email.

7. Si el sistema se encuentra aun atendiendo al usuario y se recibe otra llamada, el agente Attendant la identifica y procede a crear otro objeto Executor (Fig. 42).

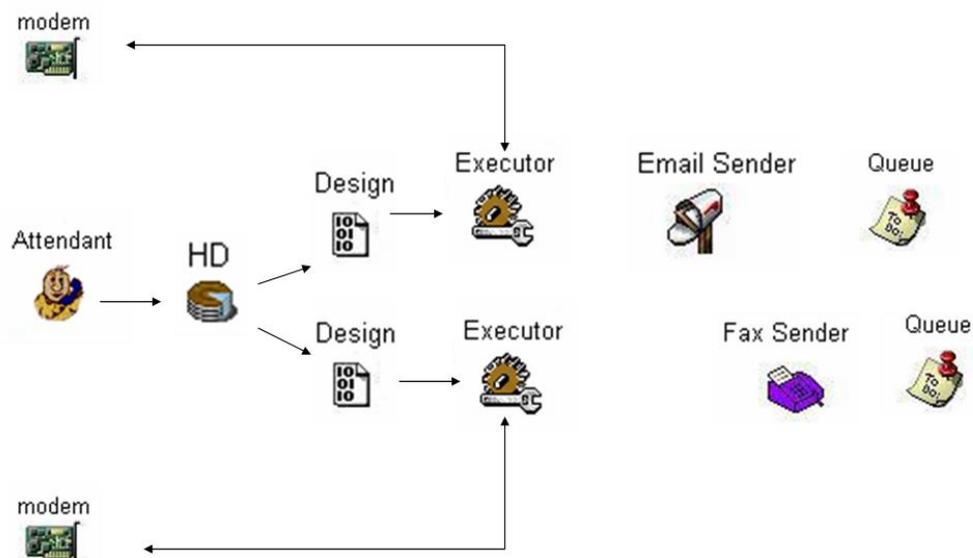


Figura 42. Atendiendo dos llamadas.

8. Si en esta ultima llamada se solicita una respuesta por medio de fax o correo electrónico, el sistema vuelve a verificar el estado de la interfaz, como en este momento no se tiene disponible ninguna, dicha respuesta es almacenada en una cola correspondiente a su método de respuesta (fax o correo electrónico) para que cuando una de las interfaces este disponible, los agentes FaxSender o EmailSender las utilicen y puedan enviar todas la respuestas almacenadas en su cola (Fig.43).

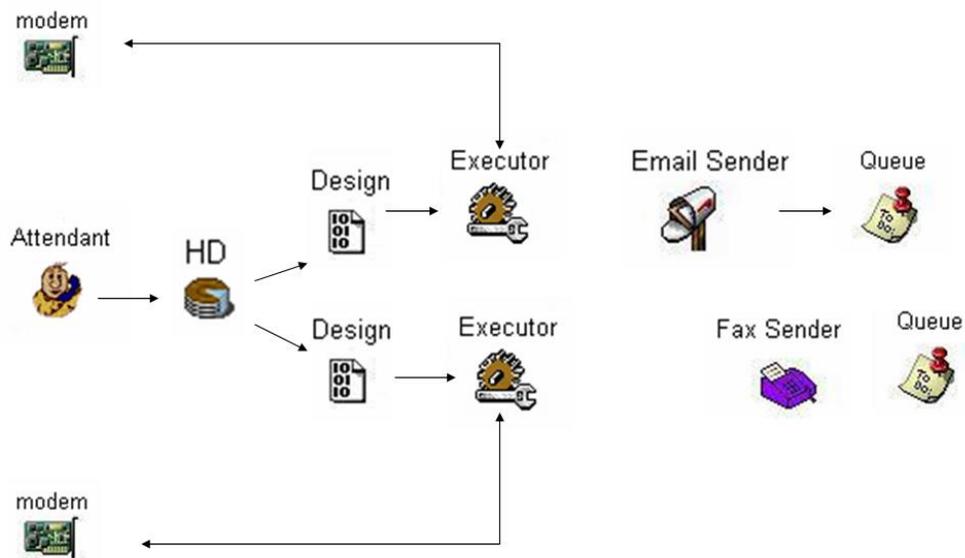


Figura 43. Envío de email a la cola.

9. Una vez el usuario ha finalizado sus peticiones, el Excecuter finaliza la conexión y se destruye el agente, notificando a los demás agentes que existe una interfaz disponible (Fig. 44).

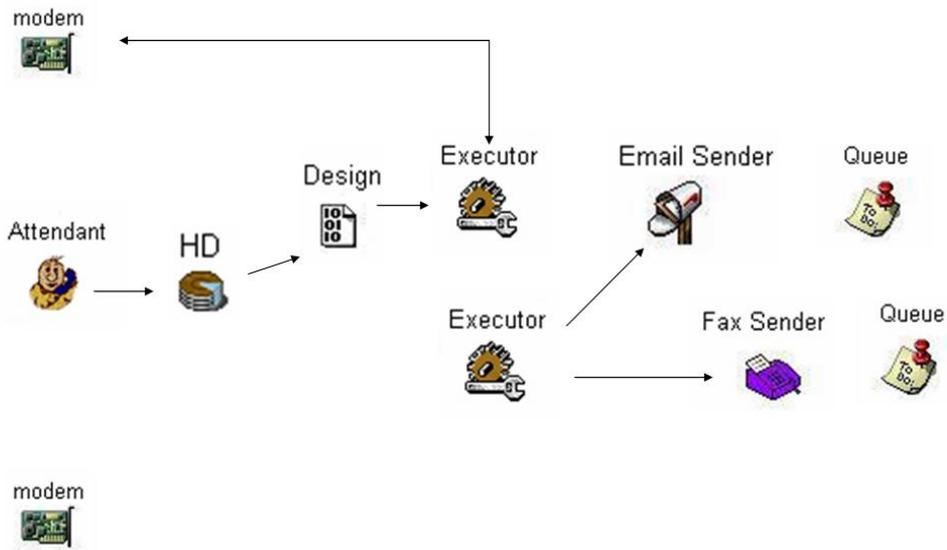


Figura 44. Notificación de que la atención de una llamada ha finalizado.

- Una vez recibida la notificación de que ha quedado disponible una interfaz, en caso de que el EmailSender o el FaxSender tengan algo pendiente en la cola de mensajes, proceden a enviarlo aprovechando que la interfaz está libre.

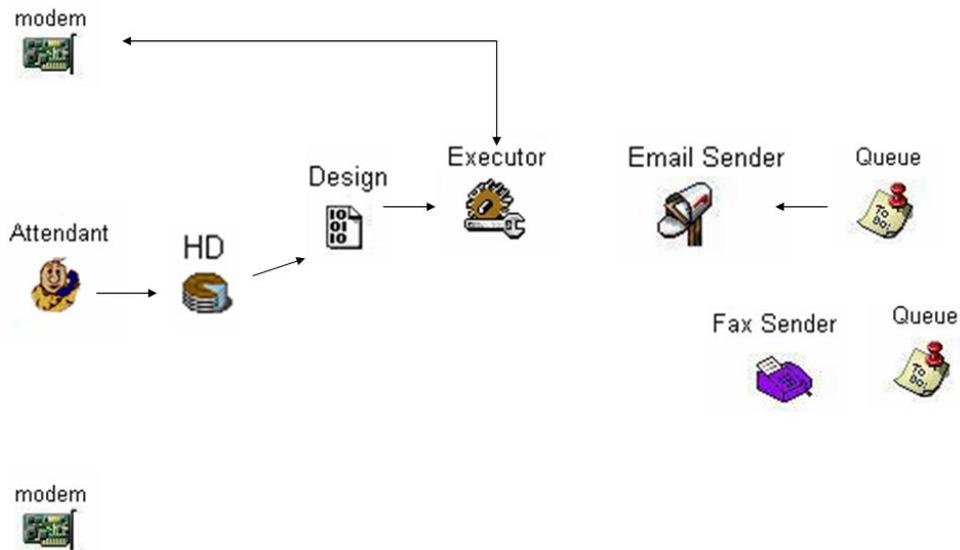


Figura 45. Recuperación de un email de la cola.

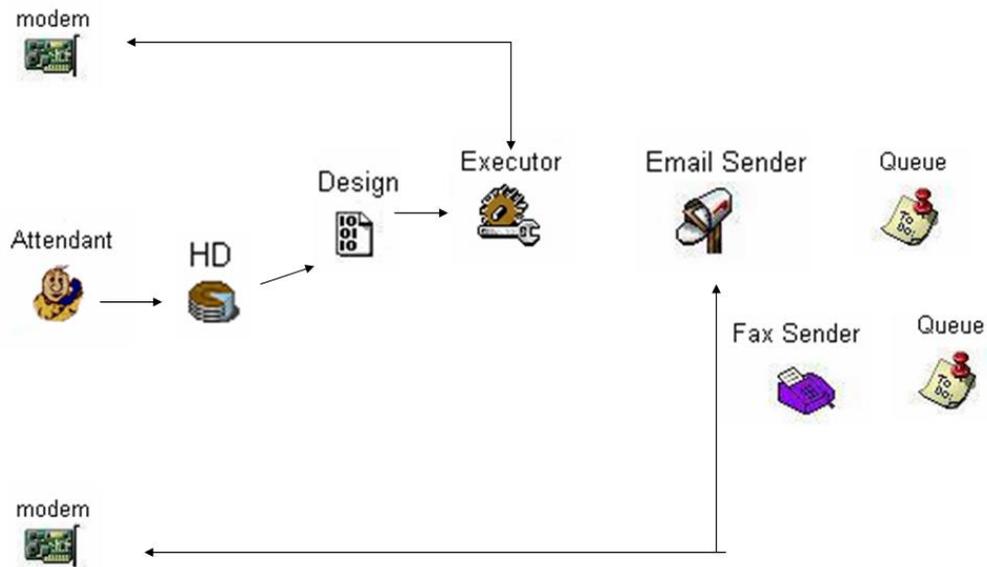


Figura 46. Envío de email recuperado de la cola

11. El proceso se repite análogamente, hasta que se decida finalizar la ejecución de dicho servicio.

12. CAPITULO IV. CONCLUSIONES Y RECOMENDACIONES

1. CONCLUSIONES

1. El hecho de tener que emplear implementaciones de JTAPI, JComm y JavaBeans Activation Framework para un sistema operativo específico, hace que la ventaja de multiplataforma del lenguaje de programación Java se vea reducida; pero teniendo la implementación adecuada de dichas APIs, el sistema IVR puede funcionar en cualquier plataforma.
2. Para hacer funcionar el sistema IVR en otra plataforma, únicamente se necesita una implementación de JTAPI que incluya el Core package y el Media package, así como las implementaciones respectivas de JComm y JavaBeans Activation Framework.
3. MSTAPI no puede reproducir de forma audible por la línea telefónica cualquier tipo de archivo de audio, sino que debe utilizarse un archivo de audio con una codificación u-law, con una velocidad de muestreo de 8000Hz y de carácter monofónica.
4. La utilización de ODBC permite conectarse a cualquier base de datos compatible con este estándar, lo cual elimina la dificultad de tener que utilizar distintas formas de conexión para cada tipo de base de datos.
5. En sistemas de procesos concurrentes, los conceptos de hilo y sincronización constituyen la parte medular sobre la cual se estructura el funcionamiento de dichos sistemas.
6. Con un solo voice/fax/MODEM, el sistema IVR puede brindar sus tres funciones de servicio: atención de llamadas, envío de email y envío de fax.
7. JTAPI acelera el tiempo de desarrollo de sistemas de telefonía, ya que es una API orientada a objetos que cuenta con abstracciones del sistema de telefonía que modelan dicho sistema de una manera más natural que otras APIs.

2. RECOMENDACIONES

1. Para utilizar de manera aceptable todas las capacidades del sistema, los modems instalados deben estar configurados para desempeñar las tres funciones del sistema: atención de llamadas, envío de fax y envío de e-mail.
2. Asegurar la disponibilidad de los recursos (bases de datos y conexiones de telefonía o a la red LAN) a los cuales accede un servicio determinado.
3. En caso de modificaciones estructurales a bases de datos que participan en diseños del sistema IVR, es necesario actualizar dichos diseños mediante la edición de los mismos.
4. Mantener actualizada la información de los usuarios acerca de los números de fax y direcciones de correo electrónico.
5. Verificar la identidad del usuario al inicio de cualquier servicio.

BIBLIOGRAFIA

1. Análisis y Diseño de Sistemas. Kendall & Kendall. Tercera Edición. Prentice Hall Inc. México 1995.
2. Aprendiendo UML en 24 horas. Joseph Schumuller. Pearson Education. México 2000.
3. Aprenda Java como si estuviera en primero. Escuela Superior de Ingenieros Industriales. Universidad de Navarra. Marzo 1999.
4. El Lenguaje de Programación Java. Ken Arnold, Addison Wesley. Tercera Edición
5. Essential JTAPI. Spencer Roberts. First Edition. Prentice Hall.
6. Java Swing. James Elliott, O'Reily & Associates. Second Edition.

Enlaces de Internet

1. http://www.sparxsystems.com.au/UML_Tutorial.htm
2. <http://java.sun.com/products/jdbc/overview.html>
3. <http://www.bytecool.com>
4. <http://sourceforge.net/projects/xtapi>
5. <http://developer.java.sun.com/developer/onlineTraining/JavaMail/>
6. <http://sourceforge.net/projects/xtapi>
7. http://www.java4less.com/java_fax.htm
8. <http://java.sun.com/products/javacomm/>

ANEXOS

ANEXO 1

SIGLAS

API	Application Programming Interface	Interface de Programación de Aplicaciones
AWT	Abstract Windows Toolkit	Herramientas de Ventanas Abstractas
CPU	Central Process Unit	Unidad Central de Procesamiento
DSN	Data Source Name	Nombre de Origen de Datos
DTMF	Dial Tone MultiFrecuency	Marcación por Tono de Multi Frecuencia
GUI	Graphic User Interface	Interface Gráfica de Usuario
IDE	Integrated Development Environment	Entorno de Desarrollo Integrado
IVR	Interactive Voice Response	Respuesta Interactiva por Voz
JDBC	Java DataBase Connectivity	Conectividad Java de Base de Datos
JDK	Java Development Kit	Kit de Desarrollo Java
JFC	Java Fundation Class	Fundación de Clases de Java
JTAPI	Java Telephony API	API de Telefonía Java
JVM	Java Virtual Machine	Máquina Virtual de Java
ODBC	Open DataBase Connectivity	Conectividad Abierta de Base de Datos
PSTN	Public Switched Telephony Network	Red Pública Conmutada de Telefonía
SQL	Structured Query Language	Lenguaje Estructurado de Consultas
TAPI	Telephony API	API de Telefonía
UML	Unified Modelling Language	Lenguaje Unificado de Modelado

ANEXO 2

MANUAL DEL USUARIO

1. Configuración del Sistema

Antes de iniciar la creación de cualquier diseño es necesario establecer la ruta de directorio del archivo .ini del cual se obtendrán las entradas de DSN que pueden ser utilizadas durante la creación de un componente Query o Update. Para ello, la opción de menú *View/ODBC location* abre un cuadro de diálogo en el cual debe seleccionarse dicho archivo .ini. Por defecto, se tiene seleccionado el archivo que utiliza el sistema operativo: ODBC.ini, ubicado en el directorio %systemroot%/WINNT en Windows 2000/NT/XP y %systemroot%/windows en Windows 95/98.

Antes de ejecutar un servicio es necesario configurar ciertos parámetros que son necesarios para que dicho servicio funcione correctamente. En la opción de menú *View/MODEM setup*, deben especificarse los datos para cada uno de los modems que utilizará el sistema y en *View/e-mail setup*, deben colocarse otros datos utilizados para enviar e-mails .

2. Crear un diseño

Se debe seleccionar la opción *Design/New*, lo cual abrirá una nueva ventana del Manejador de Componentes y del Diseñador.

3. Abrir un diseño

Es necesario seleccionar la opción *Design/Open*, lo cual abrirá un cuadro de diálogo (Fig. 1) en el cual debe seleccionarse el nombre del diseño (.ivr) que aparecerá en la ventana del Manejador de Componentes y en la del diseñador.

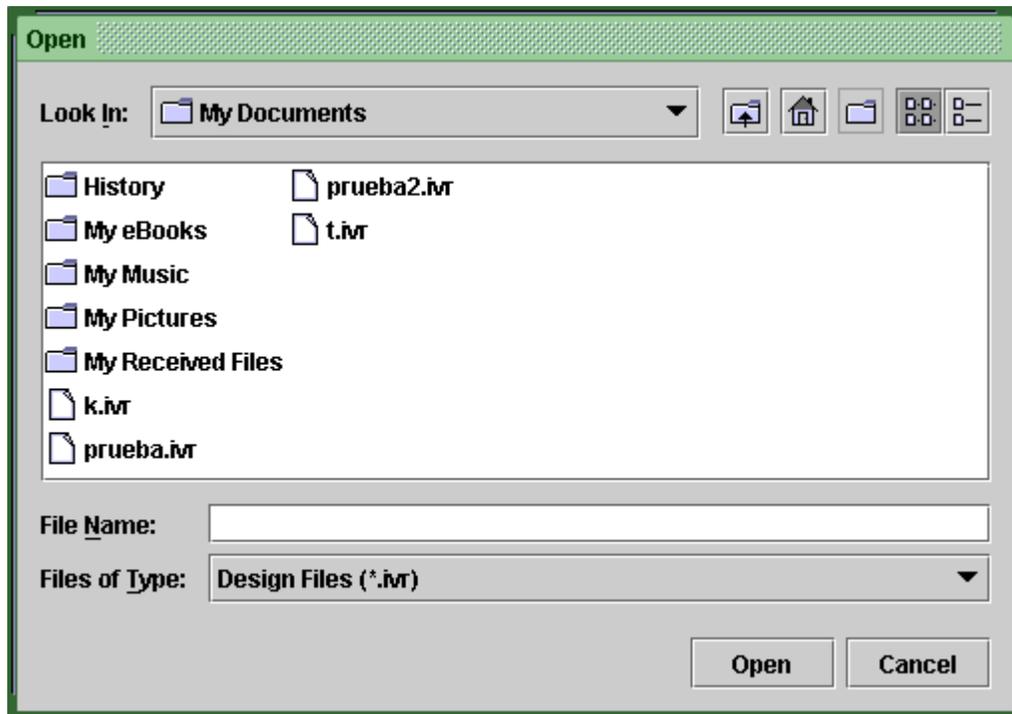


Figura 1. Cuadro de diálogo para abrir un diseño.

4. Edición de un diseño

Un componente es seleccionado de la barra de componentes al hacer clic sobre dicho componente. El estado de selección se percibe mediante un recuadro que rodea al componente respectivo o puede visualizarse mediante un mensaje desplegado en la barra de estado del diseñador (Fig. 2).



Figura 2 Selección de componentes

La barra de componentes no es fija, sino que puede colocarse en cualquier extremo de la ventana de diseño o puede estar flotante.

4.1 Agregar componentes

Debe seleccionarse de la barra de componentes, el componente que se desea agregar y luego dar un clic en un el lugar del área de diseño donde se quiere colocar dicho componente. Puede cambiarse la ubicación del componente dentro del área de diseño al seleccionar el componente Mover y luego “arrastrar y soltar” el componente hacia la nueva posición.

4.2 Agregar conexiones

Para establecer una conexión entre dos componentes, habiendo seleccionado previamente el componente Conexión, únicamente se debe hacer click en el componente origen y después otro click en el componente destino. Visualmente una conexión se presenta como una línea que parte del componente origen hacia el destino. El sentido de una conexión se simboliza mediante un círculo negro en el componente destino.

Cuando el componente origen es un Menú se pide que se seleccione la opción a la cual se establecerá el enlace que se esta creando (Fig. 3).

El nombre completo de una opción esta dado por: <nombre_componente>.<nombre_opcion>



Figura 3. Diálogo de selección de la opción de un Menú que será el origen de una conexión

Cuando el componente origen es una Condición se pide que se seleccione el camino al cual se establecerá el enlace que se esta creando (Fig. 4).

Únicamente hay dos caminos posibles:

True connection: en caso de que el componente Consulta asociado tenga registros. (Pág. 31)

False connection: en caso de que el componente Consulta asociado no tenga registros.

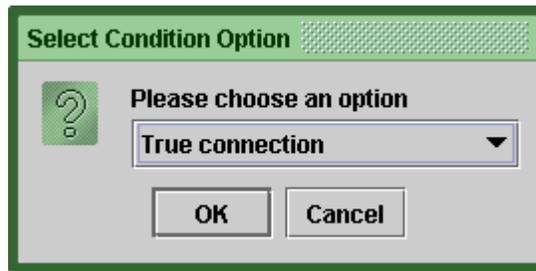


Figura 4. Diálogo de selección del camino de una Condición que será el origen de una conexión

No esta permitido:

5. Conectar un componente a si mismo, a excepción del componente Menú.
6. Intentar colocar como componente destino de una conexión al componente Home.
7. Hacer que un componente sea el origen de una conexión cuando éste todavía no forma parte como destino en otra conexión.
8. Colocar dos conexiones entre los mismos componentes pero en sentidos contrarios.

4.3 Eliminar componentes

Para eliminar un componente, habiendo seleccionado previamente el componente Selección únicamente se da un clic derecho sobre el componente a eliminar y luego seleccionar *Delete* del menú emergente (Fig. 5). Al eliminar un componente, todas las conexiones hacia o desde ese componente son eliminadas automáticamente.



Figura 5. Menú emergente para eliminar un componente del área de diseño.

4.4 Eliminar conexiones

Para eliminar una conexión, habiendo seleccionado previamente el componente Selección, únicamente se da un clic derecho sobre la conexión a eliminar y luego seleccionar la opción *Delete* del menú emergente.

4.5 Ver datos de conexiones

Para ver el componente origen y el componente destino de una conexión, habiendo seleccionado previamente el componente Selección, únicamente se debe hacer clic derecho sobre la conexión y luego seleccionar la opción *Data* del menú emergente (Fig. 6).



Figura 6. Menú emergente para ver datos de una conexión.

4.6 Configurar componentes

Para iniciar la edición de un componente, habiendo previamente seleccionado el componente Selección, debe hacerse doble clic sobre el componente o hacer clic derecho sobre ese componente y luego seleccionar la opción *Edit* del menú emergente (Fig 7). Dicha acción, abre el dialogo de edición de ese componente. Todos los componentes tienen un nombre por defecto el cual puede ser cambiado. Todos los diálogos tienen dos botones de acción:

Ok: Para guardar los cambios realizados en dicho diálogo.

Cancel: Para obviar los cambios realizados en dicho diálogo.



Figura 7. Menú emergente para editar un componente del área de diseño.

a. Configurar una Condición

Se debe seleccionar un componente Consulta del cuadro de selección. Dicho componente Consulta es el que se utilizará como evaluación para determinar si el siguiente componente a llamar durante la ejecución de un servicio será el del camino "True" o el del camino "False" (Fig. 8).

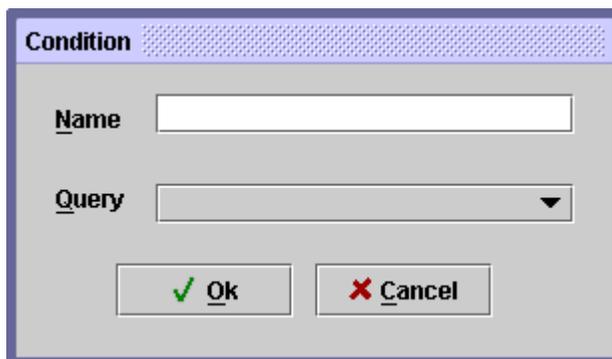
The image shows a dialog box titled "Condition". It has a light blue header bar with the title. Below the header, there are two input fields: "Name" with a text box and "Query" with a dropdown menu. At the bottom, there are two buttons: "Ok" with a green checkmark icon and "Cancel" with a red X icon.

Figura 8. Diálogo de configuración del componente Condición

b. Configurar un Dtmf

1. Se debe especificar el tipo de datos que se espera recibir a través de los tonos *Dtmf*, el cual puede ser numérico o alfanumérico.
 2. Se debe establecer la secuencia de elementos a leer por medio de los tonos. Cada elemento de esta secuencia está formado por el tipo de dato y la cantidad de dígitos. Por medio del botón *Add* se agrega un nuevo elemento a la secuencia y a través del botón *Delete* se elimina uno.
- Se debe tomar en cuenta que para elementos alfanuméricos, la convención para los tonos leídos es:

Primer tono: Indica el número de la tecla donde se encuentra la letra.

Segundo tono: Indica la posición de la letra dentro de la tecla (Fig. 9)

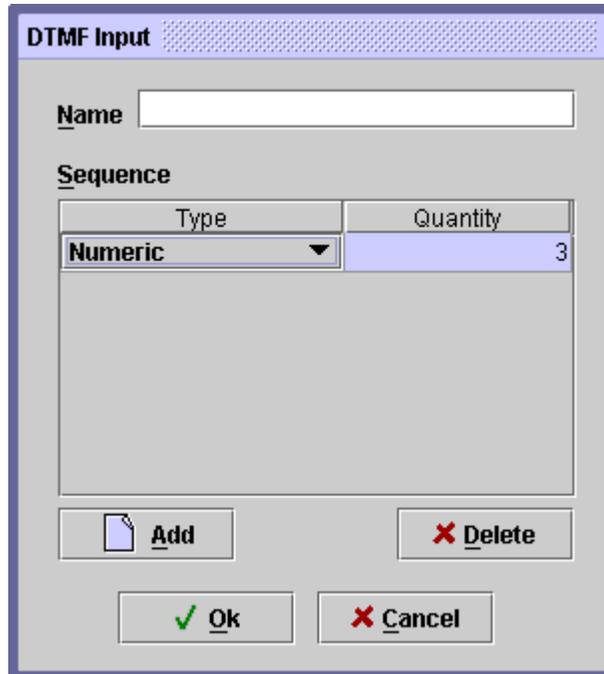


Figura 9. Diálogo de configuración del componente Entrada DTMF

c. Configurar una Consulta

1. Se debe seleccionar el DSN que se utilizará para conectarse a la base de datos. (Fig. 10)
2. Se debe escribir una instrucción SELECT-SQL según el estándar ANSI SQL-92. Dentro de una consulta pueden incluirse como variables los nombres de: componentes dtmf, opciones dinámicas de menús y campos de otras consultas, pero debe utilizarse el operador "&" para concatenar dichas variables. Si la variable es de tipo carácter, deben ponerse apostrofes antes y después de la variable. Algunos ejemplos son:

```
SELECT * FROM Estudiantes WHERE Carnet = ' & Query1.Estudianteld & '
```

```
SELECT * FROM Estudiantes WHERE Estudianteld = & Query1.Estudianteld
```

```
SELECT * FROM Libros WHERE Nombre = ' & Menu1.LibroMombre & '
```

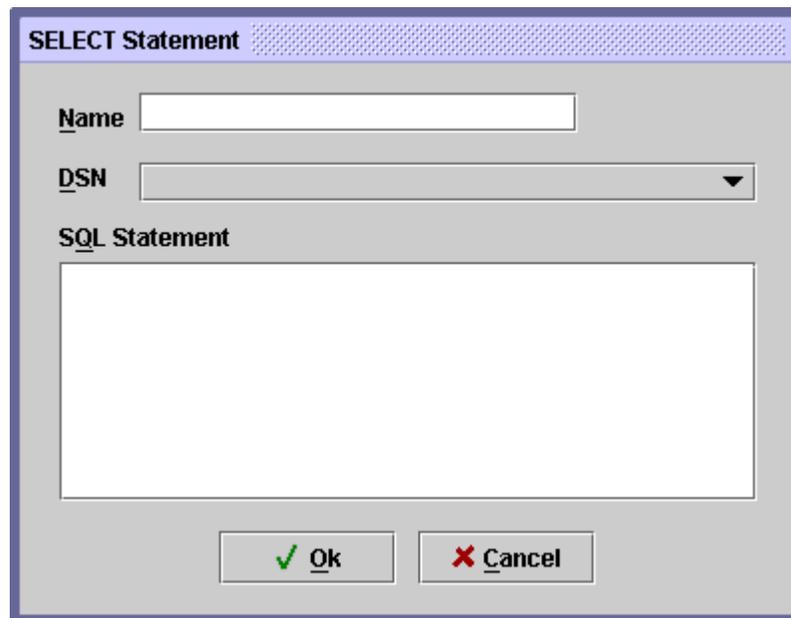


Figura 10. Diálogo de configuración del componente Consulta-SQL

d. Configurar un Menú

Se deben agregar las opciones que se van a reproducir a través de la línea telefónica (Fig. 11).

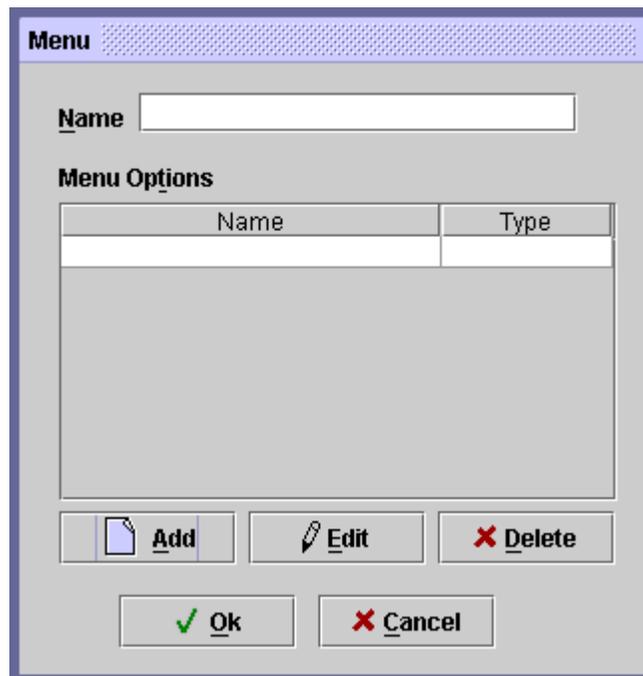


Figura 11. Diálogo de configuración del componente Menú

Para agregar una nueva opción se presiona el botón *Add*, para eliminar se selecciona la opción y después se presiona el botón *Delete* y para editar seleccione la opción y luego presione el botón *Edit*.

Las opciones pueden ser:

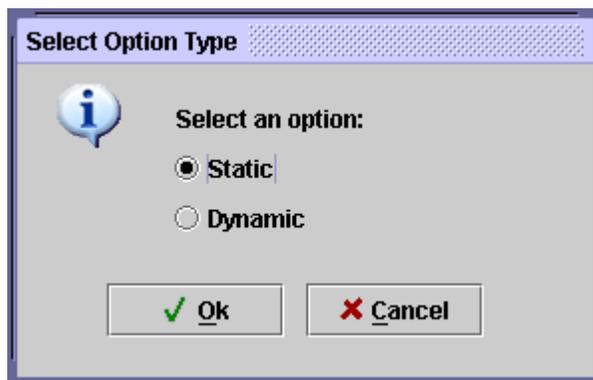


Figura 12. Diálogo de selección de tipo de Opción

1. Opción estática: Texto colocado en tiempo de diseño (Fig. 13).

Únicamente debe introducirse el texto que se desea que sea reproducido por la línea telefónica

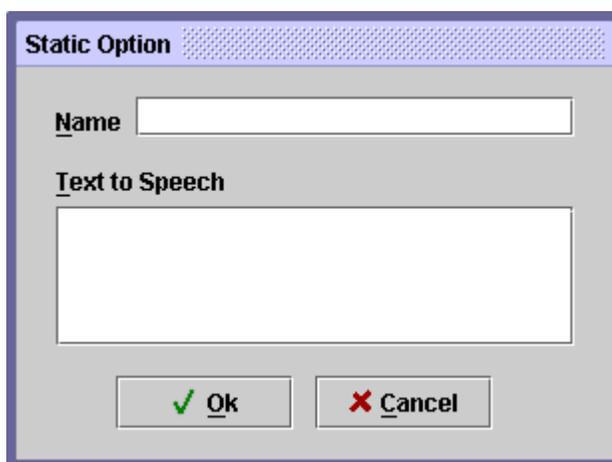


Figura 13. Diálogo de configuración de una opción estática

2. Opción dinámica: Texto generado en tiempo de ejecución a partir de una consulta (Fig. 14).

Debe seleccionarse la consulta de la cual se extraerán los registros que formarán las opciones.

Debe seleccionarse el campo que será reproducido por la línea telefónica.

Debe seleccionarse el campo del cual se obtendrá el valor al momento de que sea escogida una opción.

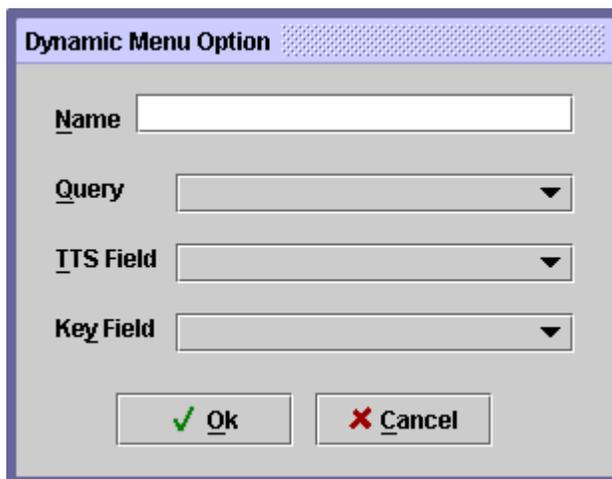
The image shows a dialog box titled "Dynamic Menu Option". It has a light blue header bar with the title. Below the header, there are four rows of controls: 1. A label "Name" followed by a white text input field. 2. A label "Query" followed by a dropdown menu. 3. A label "TTS Field" followed by a dropdown menu. 4. A label "Key Field" followed by a dropdown menu. At the bottom of the dialog, there are two buttons: "Ok" with a green checkmark icon and "Cancel" with a red X icon.

Figura 14. Diálogo de configuración de una opción dinámica

Hay que tomar en cuenta que para cada opción lo que se reproduce por la línea telefónica es:

“Para” <texto_estático_o_dinámico> “marque” <número_opción>

Para que la opción seleccionada por el usuario sea procesada, el usuario debe presionar asterisco, es por eso que por defecto el componente menú dice al inicio un mensaje que dice: “Presione asterisco después de seleccionar la opción”.

e. Configurar una Actualización

1. Se debe seleccionar el DSN que se utilizará para conectarse a la base de datos (Fig. 15)

2. Se debe escribir una instrucción UPDATE-SQL según el estándar ANSI SQL-92. Dentro de una actualización pueden incluirse como variables los nombres de: componentes dtmf, opciones dinámicas de menús y campos de otras consultas, pero debe utilizarse el operador “&” para

concatenar dichas variables. Si la variable es de tipo carácter, deben ponerse apostrofes antes y después de la variable. Algunos ejemplos son:

```
UPDATE Estudiantes SET Telefono = ' & Dtmf1 & ' WHERE Carnet = & Query1.Estudianteld
```

```
UPDATE Estudiantes SET Fax = ' & Dtmf1 & ' WHERE Estudianteld = & Query1.Estudianteld
```

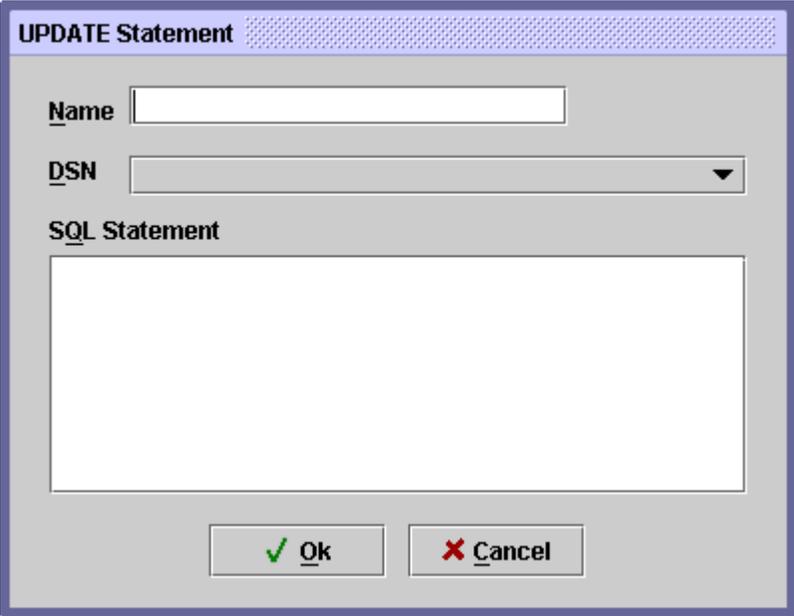


Figura 15. Diálogo de configuración del componente Actualización-SQL

f. Configurar una Respuesta de Voz

Se deben agregar las líneas que se van a reproducir a través de la línea telefónica (Fig. 16)

Para agregar una nueva línea se presiona el botón *Add*, para eliminar se selecciona la línea y después se presiona el botón *Delete* y para editar seleccione la línea y luego presione el botón *Edit*.

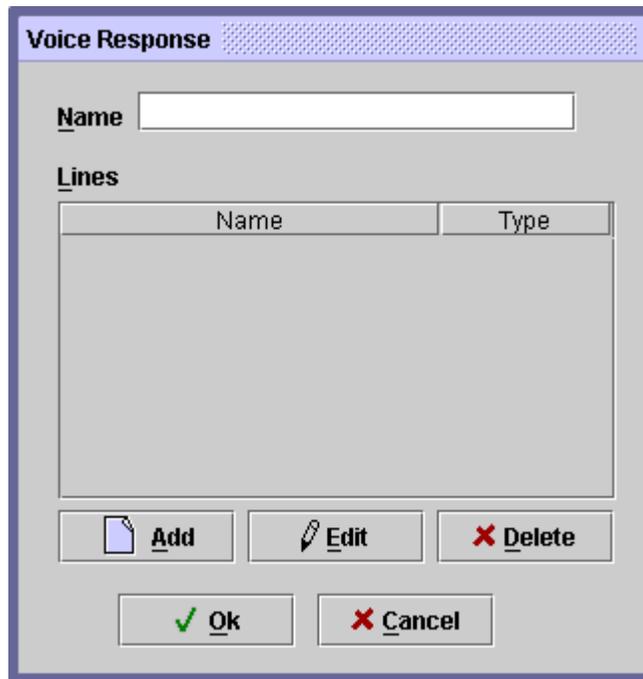


Figura 16. Diálogo de configuración del componente Respuesta de Voz

Las líneas pueden ser (Fig. 17):

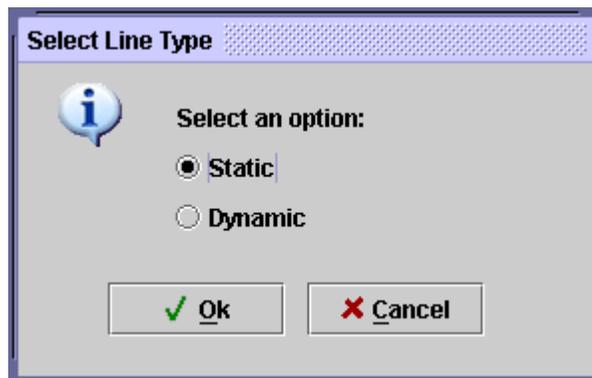
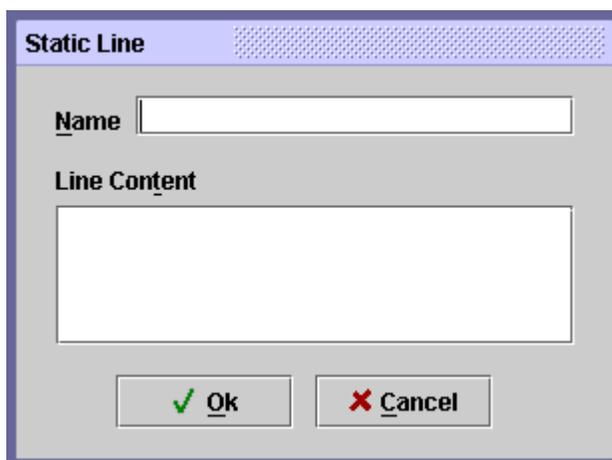


Figura 17. Diálogo de selección de tipo de Línea

1. **Línea estática:** Texto colocado en tiempo de diseño (Fig. 18).

Únicamente debe introducirse el texto que se desea que sea reproducido por la línea telefónica.



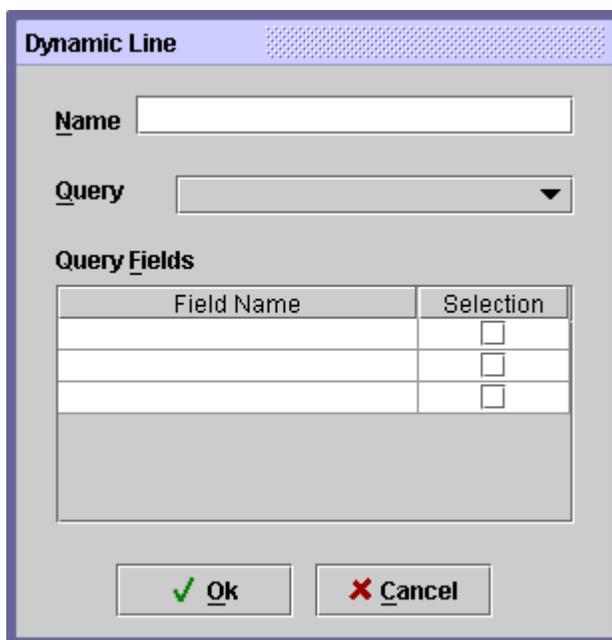
The image shows a dialog box titled "Static Line". It has a light blue header bar with the title. Below the header, there is a text input field labeled "Name". Underneath that is a larger text area labeled "Line Content". At the bottom of the dialog, there are two buttons: "Ok" with a green checkmark icon and "Cancel" with a red X icon.

Figura 18. Diálogo de configuración de una línea estática

2. **Línea dinámica:** Texto generado en tiempo de ejecución a partir de una consulta (Fig. 19).

Debe seleccionarse la consulta de la cual se extraerán los registros que formarán las líneas, es decir, cada registro será una línea.

Deben seleccionarse los campos que formarán parte de la línea.



The image shows a dialog box titled "Dynamic Line". It has a light blue header bar with the title. Below the header, there is a text input field labeled "Name". Underneath that is a dropdown menu labeled "Query". Below the dropdown is a section titled "Query Fields" which contains a table with two columns: "Field Name" and "Selection". The table has three rows, each with an empty cell in the "Field Name" column and a checkbox in the "Selection" column. At the bottom of the dialog, there are two buttons: "Ok" with a green checkmark icon and "Cancel" with a red X icon.

Field Name	Selection
	<input type="checkbox"/>
	<input type="checkbox"/>
	<input type="checkbox"/>

Figura 19. Diálogo de configuración de una línea dinámica

g. Configurar una Respuesta por Fax (Fig. 20)

Fax Response

Name

Hader

Fax Number

Query

Query Field

Content

Query

Query Fields

Field Name	Selection
	<input type="checkbox"/>
	<input type="checkbox"/>
	<input type="checkbox"/>

Footer

Figura 20. Diálogo de configuración del componente Respuesta por Fax

Las partes a configurar a un componente de respuesta por Fax son:

1. Encabezado

Está formado por líneas de forma idéntica a un componente de Voz, con la diferencia de que estas líneas no son reproducidas por la línea telefónica sino que son agregadas al inicio del archivo de texto que será enviado por Fax. Para entrar a su diálogo de configuración, se debe presionar el botón *Header*. (Fig. 21)

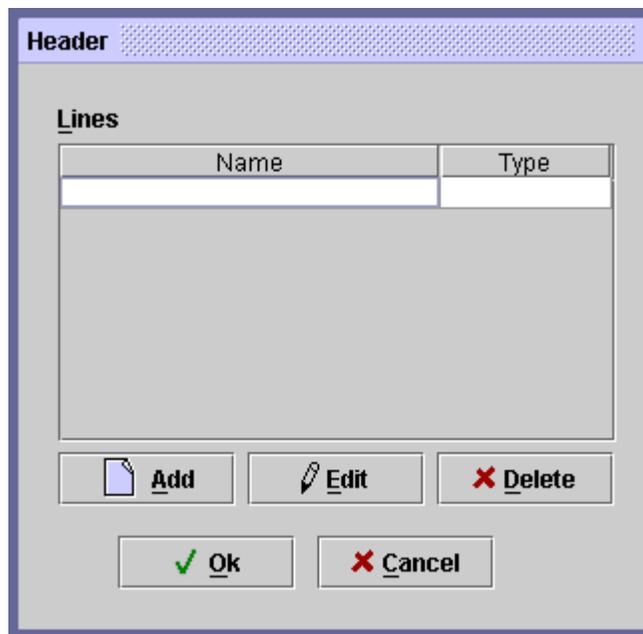


Figura 21. Diálogo de configuración de encabezado de página

2. El destinatario

Se debe seleccionar la consulta que contiene el registro del cual se extraerá el número del fax y luego se selecciona el campo que contiene dicha información (Fig. 20).

3. El cuerpo

Se selecciona la consulta que contiene los registros que formarán parte del contenido del fax y luego se seleccionan los campos de dicha consulta que se necesitan incluir (Fig. 20).

4. El pie de página

Idéntico al encabezado de página, con la diferencia de que sus líneas son agregadas al final del archivo de texto que será enviado por Fax. Para entrar a su diálogo de configuración, se debe presionar el botón *Footer*. (Fig. 22)

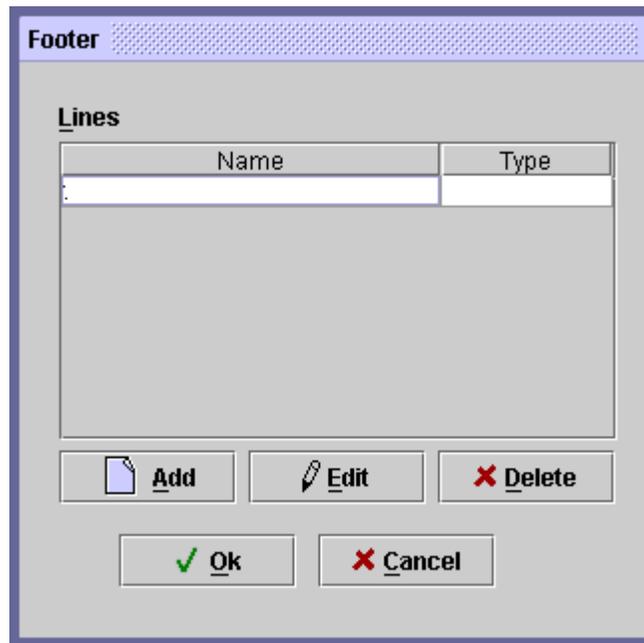


Figura 22. Diálogo de configuración de pie de página

h. Configurar una Respuesta por Email (Fig. 23)

Las partes a configurar a un componente de respuesta por E-mail son:

1. Encabezado

Está formado por líneas de forma idéntica a un componente de Voz, con la diferencia de que estas líneas no son reproducidas por la línea telefónica sino que son agregadas al inicio del archivo de texto que será enviado por Email. Para entrar a su diálogo de configuración, se debe presionar el botón *Header* (Fig. 21).

E-Mail Response

Name

Header

e-Mail Address

Query

Query Field

Content

Query

Query Fields

Field Name	Selection
	<input type="checkbox"/>
	<input type="checkbox"/>
	<input type="checkbox"/>

Footer

Figura 23. Diálogo de configuración del componente Respuesta por E-mail

2. El destinatario

Se debe seleccionar la consulta que contiene el registro del cual se extraerá la dirección de correo electrónico del destino y luego se selecciona el campo que contiene dicha información. (Fig. 23)

3. El cuerpo

Se selecciona la consulta que contiene los registros que formarán parte del contenido del email y luego se seleccionan los campos de dicha consulta que se necesitan incluir.

4. El pie de página

Idéntico al encabezado de página, con la diferencia de que sus líneas son agregadas al final del archivo de texto que será enviado por Email. Para entrar a su diálogo de configuración, se debe presionar el botón *Footer* (Fig. 22).

5. Guardar un diseño

Para guardar un diseño, se debe seleccionar la opción *Design/Save* si se desea guardar el diseño con el mismo nombre que posee al momento de ser abierto; pero si se quiere guardar con un nombre distinto debe utilizarse la opción de menú *Design/Save As*. Si se utiliza *Design/Save* para guardar un nuevo diseño o *Design/Save As* para guardar un diseño existente, se abrirá un cuadro de diálogo para que se especifique la ubicación y el nombre del archivo del diseño (Fig. 24). Dicho archivo tendrá extensión *.ivr*.

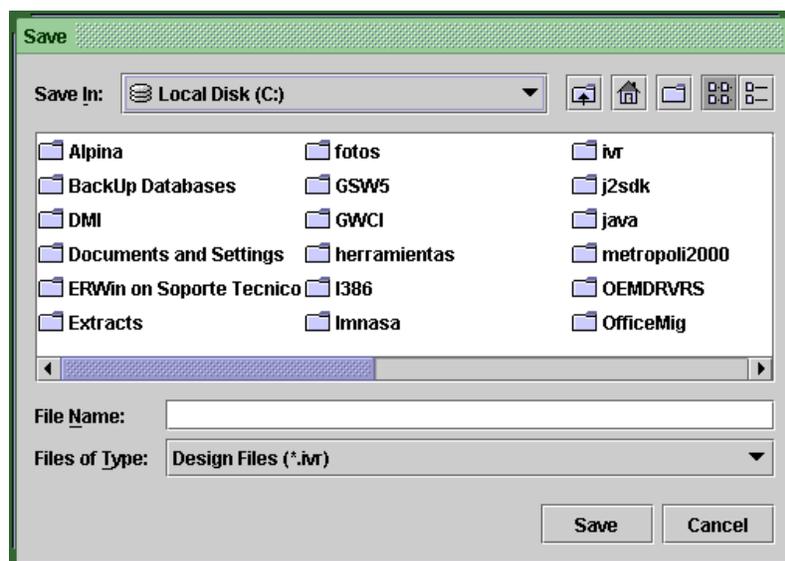


Figura 24. Cuadro de diálogo de las opciones Save y SaveAs del menú Design

Si el directorio y el nombre del archivo coinciden con otro servicio ya existente, se pide una confirmación si se desea sobrescribir con el nuevo diseño (Fig. 25)

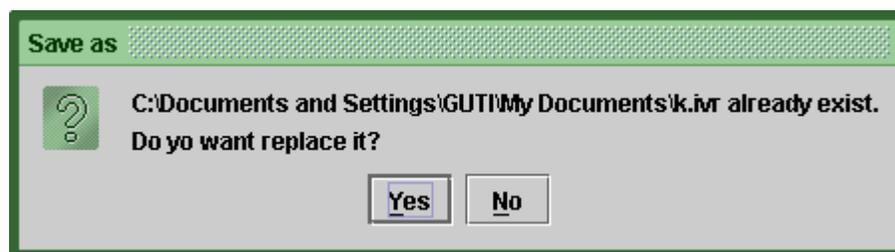


Figura 25. Cuadro de diálogo de confirmación de reemplazo de archivo.

6. Cerrar un diseño

Utilizar la opción de menú *Design/Close*. Si se trata de un nuevo diseño, antes de cerrarlo se pregunta si se desea guardar o no dicho diseño (Fig. 26). Si es un diseño previamente guardado, entonces se pregunta si se desean guardar los cambios realizados en ese diseño (Fig 27)



Figura 26. Diálogo de confirmación para guardar un nuevo diseño

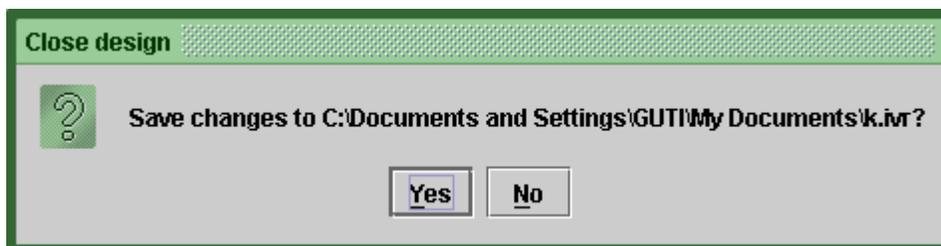


Figura 27. Diálogo de confirmación para guardar cambios de un diseño

7. Ejecución de un servicio

Para ejecutar un servicio hay que seleccionar la opción de menú *Service/Start*, lo cual abrirá un cuadro de diálogo (Fig x de open) para seleccionar el archivo que contiene el diseño que se va a colocar en ejecución.

La ejecución de un servicio provoca que las llamadas puedan ser atendidas por el servicio en ejecución y solo puede haber uno en ejecución a la vez. Si no se encuentran modems configurados para recibir llamadas (ver "Configuración del sistema"), el servicio no puede iniciarse (Fig. 28).



Figura 28. Mensaje de servicio no iniciado por no encontrarse modems configurados para recibir llamadas

8. Detener un servicio

Detener un servicio provoca que las llamadas ya no sean atendidas por el sistema. Para detener un servicio debe seleccionarse la opción de menú *Service/Stop*. Antes de detener el servicio, el sistema pide una confirmación para llevar a cabo dicha acción (Fig. 29).

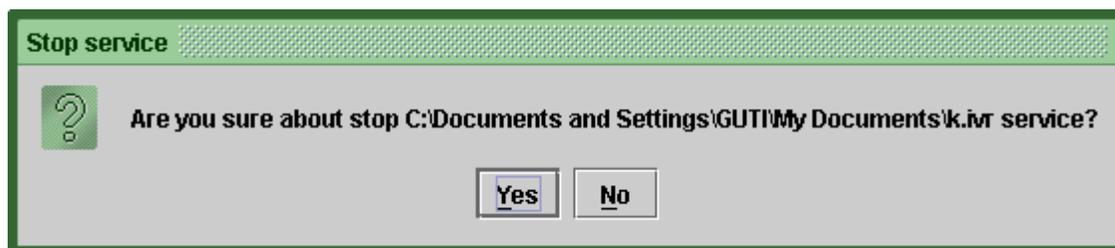


Figura 29. Diálogo de confirmación para detener un servicio

9. Cerrar el Sistema

Para cerrar el sistema debe seleccionarse la opción de menú *Design/Quit*. Si en el momento de cerrar el sistema se encuentra en ejecución un servicio, entonces se pregunta al usuario si aun en esa situación se quiere finalizar la aplicación (Fig 30).



Figura 30. Diálogo de confirmación para cerrar el sistema IVR en caso de que este corriendo un servicio

Si en el momento de cerrar el sistema se encuentra abierto un diseño, entonces se pide una confirmación de que si se desea guardar el nuevo diseño o guardar los cambios realizados a un diseño según sea el caso.

ANEXO 3.

EJEMPLOS DE UTILIZACION DEL SISTEMA IVR

Para la creación de los ejemplos, se utilizará la siguiente base de datos como fuente de información. En los ejemplos se asume que previamente se ha creado el DSN, llamado BDD, para esta base de datos.

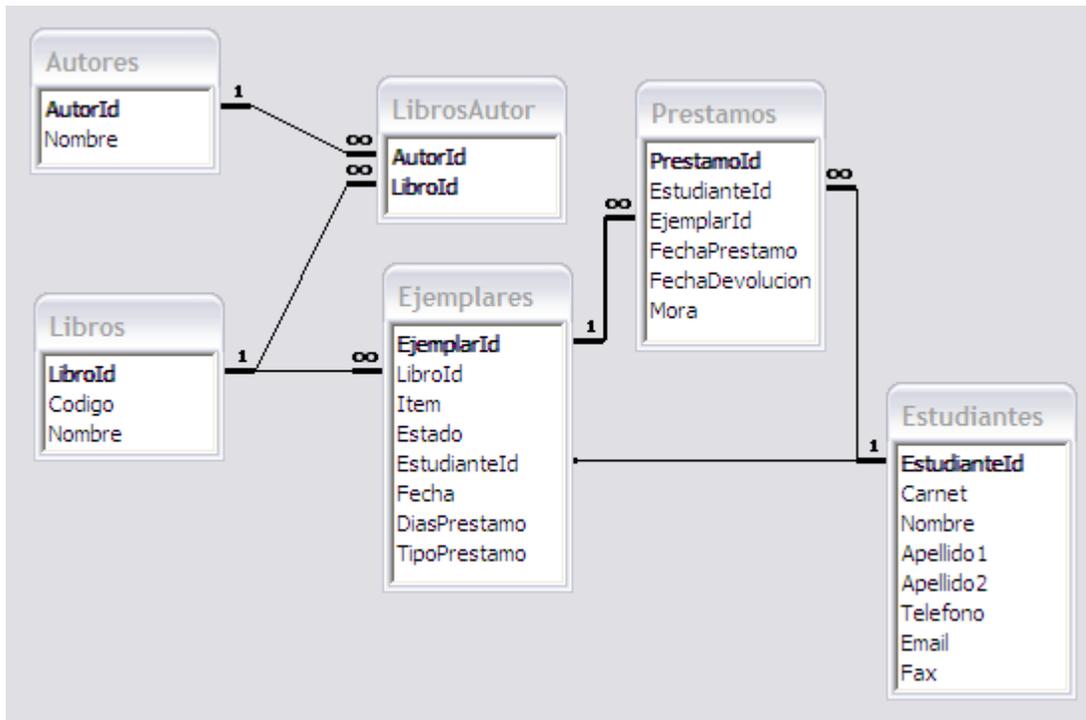


Figura 1. Base de datos de prueba de Biblioteca

A continuación se describen dos ejemplos de la creación de servicios simples para ser utilizados por el sistema IVR. Nótese que estos diseños son para fines de comprensión del uso de la interfaz del sistema IVR, además para la creación de sistema es necesario tener configurado correctamente el sistema y tener configurados los DSN de las bases de datos a utilizar en el servicio.

Ejemplo 1:

El servicio a crear tiene por fin consultar los estudiantes almacenados en una base de datos para luego seleccionar uno, y al final especificar el nombre del estudiante que se seleccionó.

- Ejecutar el sistema.
- Seleccionar la opción *new* del menú *Design*
- Para el inicio del sistema se pretende colocar un mensaje de bienvenida, por lo cual es necesario agregar un VoiceComponent en nuestro diseño.
- Una vez el componente ha sido agregado a nuestro diseño, es necesario configurar el mensaje que enviará al usuario. Para hacer esto es necesario, primero seleccionar en la barra de controles, la opción de selección y luego darle doble clic al componente o mediante el botón derecho del Mouse con el cursor sobre el componente seleccionado *Edit* del menú desplegable. Posteriormente a esto aparecerá el cuadro de diálogo del componente (Fig. 2).

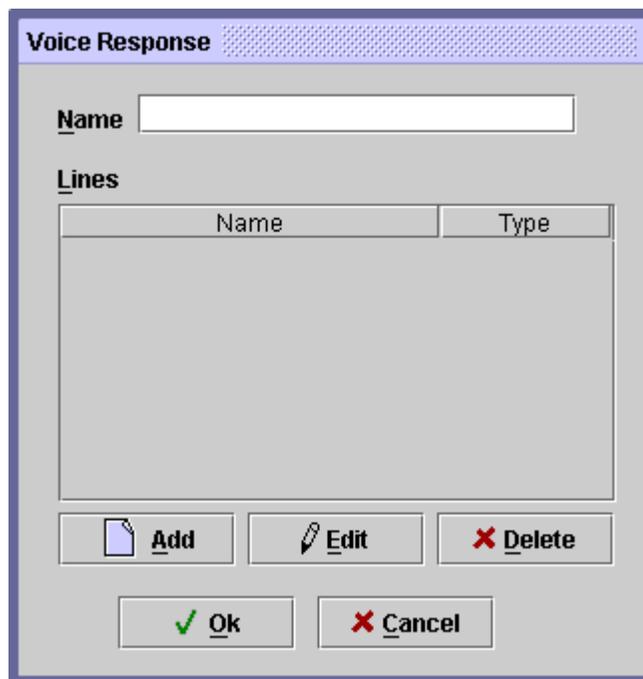


Figura 2. Diálogo de configuración del componente Respuesta de Voz

- Una vez abierto dicho dialogo, es necesario, agregar una línea estática, la cual contendrá el mensaje que se le enviara al usuario. Al dar clic sobre el botón *Add* aparece el siguiente cuadro de dialogo

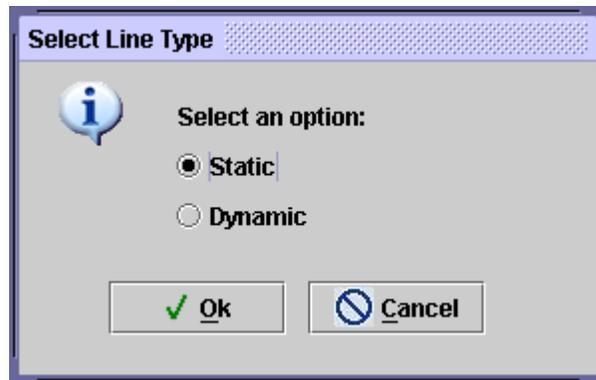


Figura 3. Diálogo para la selección del tipo de línea a agregar.

- Seleccionamos *Static* y nos presenta el dialogo donde colocamos el mensaje de bienvenida (Fig. 3)

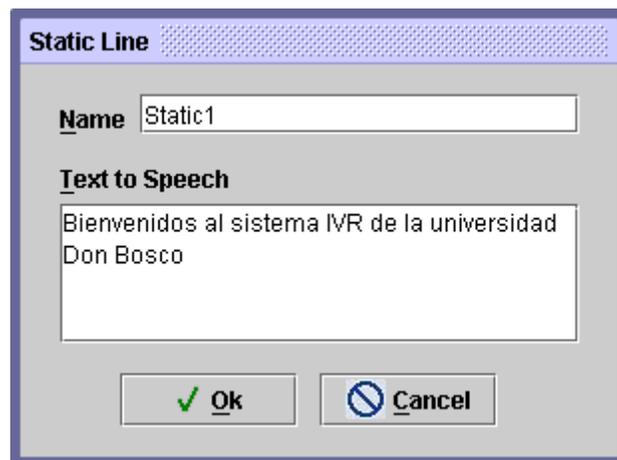


Figura 3. Diálogo para la configuración de una línea Estática.

- Luego agregamos un QueryComponent para obtener una lista de estudiantes almacenados en la base de datos



- Configuramos el componente, seleccionando el DSN conveniente e ingresamos nuestra consulta (Fig 4).

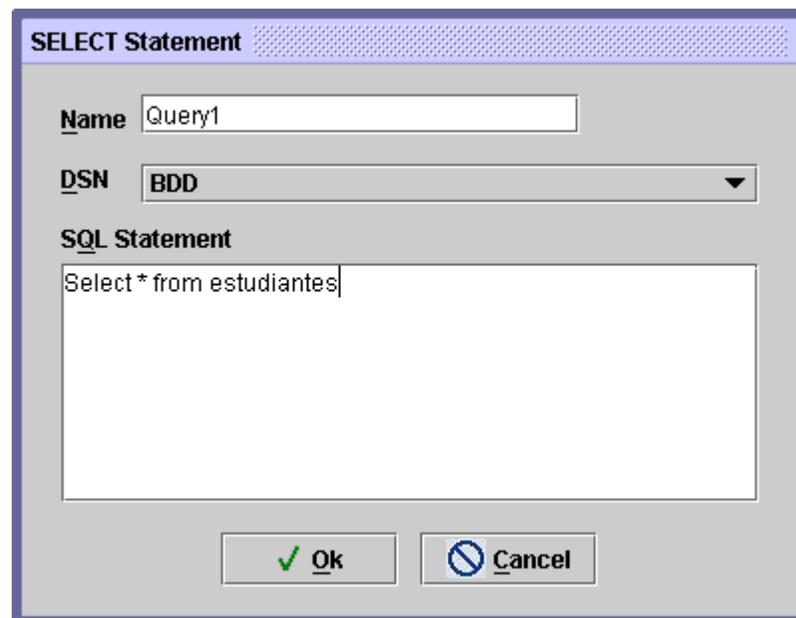


Figura 4. Diálogo para la configuración de un QueryComponent.

- En este momento se procede a crear el menú en el cual se escucharán los distintos nombres de los estudiantes. Para hacer esto procedemos a arrastrar un MyMenuComponent al área de diseño. Después entramos al diálogo de edición del menú y agregamos una opción seleccionando que la opción sea Dinámica, es decir provenientes de la base de datos (Fig. 5).

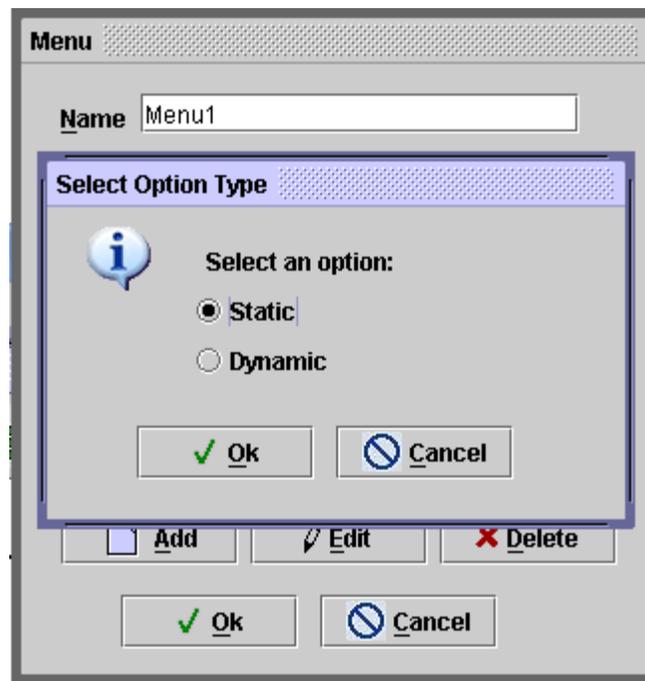


Figura 5. Selección de un tipo de opción de menú.

Dentro de la opción dinámica seleccionamos el Query desde donde obtendremos dichas opciones, y finalmente seleccionamos el campo que sonará y el campo clave para identificar la opción que el usuario ingrese (Fig. 6)

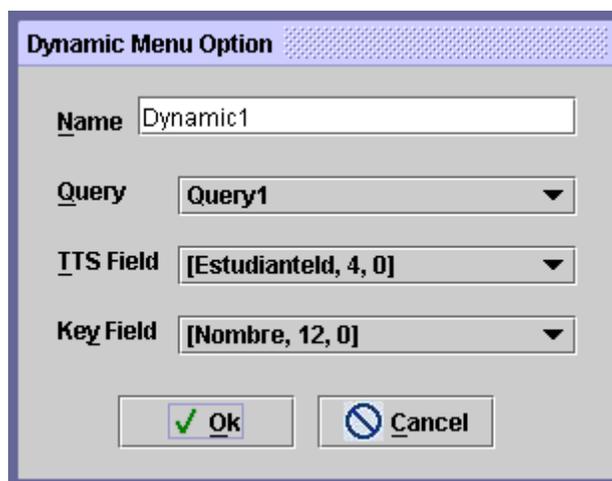


Figura 6. Configuración de una opción dinámica de menú.

- Una vez colocado el menú, debemos obtener los datos de la opción seleccionada, para esto es necesario insertar otro QueryComponent, con el cual obtendremos los datos del estudiante seleccionado en el menú anterior (Fig. 7).

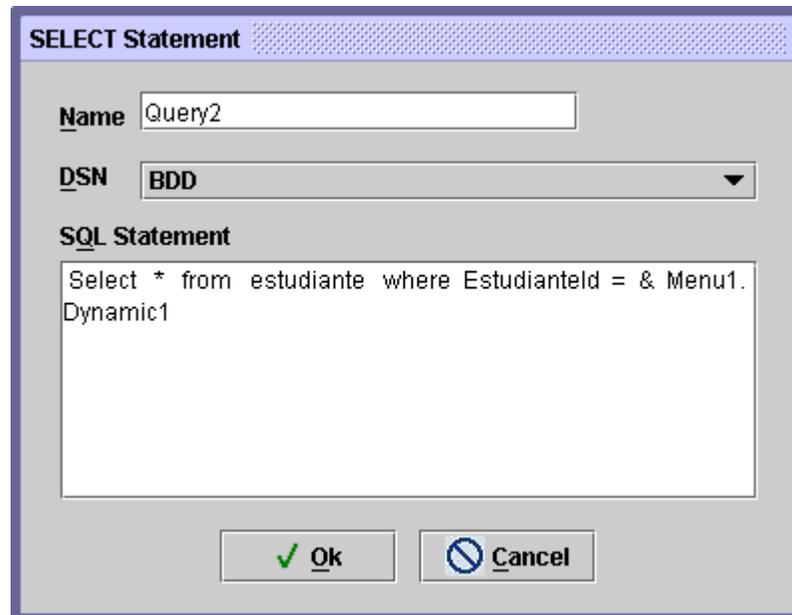


Figura 7. Configuración de un componente SELECT

- Al final basta con agregar otro VoiceComponent en el cual enviaremos como respuesta al usuario un mensaje conteniendo el nombre y el apellido del el usuario seleccionado en el menú.

El nuevo VoiceComponent estará configurado contendrá una línea estática, y una línea dinámica, la primera se utiliza para enviar un mensaje plano, la segunda es la parte donde se envía los datos obtenidos de la base de datos (Fig. 8).

Voice Response

Name Voice2

Lines

Name	Type
Voice2.Static2	Static
Voice2.Dynamic1	Dynamic

Add Edit Delete

Ok Cancel

Figura 8. Configuración de un componente Voice

El mensaje plano que se enviaría se muestra en la figura 9.

Static Line

Name Static2

Text to Speech

Usted ha seleccionado al alumno

Ok Cancel

Figura 9. Configuración de una línea estática.

La configuración de la línea dinámica se muestra en la figura 10, en la cual se selecciona el Query en donde obtenemos los datos (en este caso Query2) y se seleccionan los campos que se enviarán al usuario.

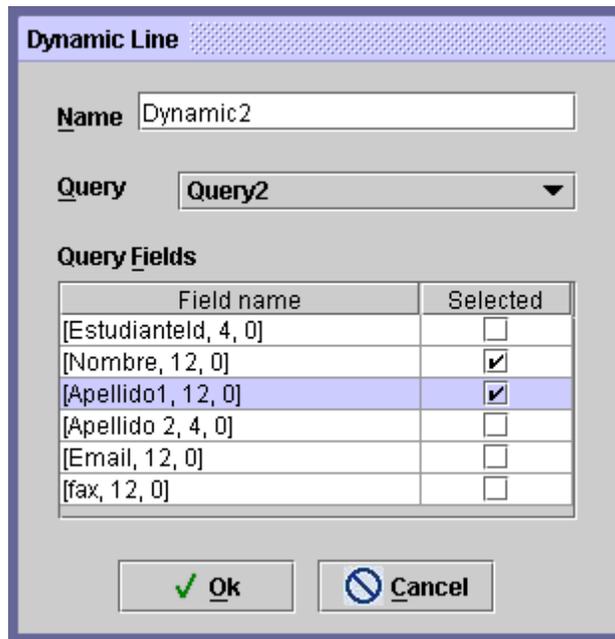


Figura 10. Configuración de una línea dinámica

- Una vez completado este procedimiento, obtendríamos un diseño como el siguiente:

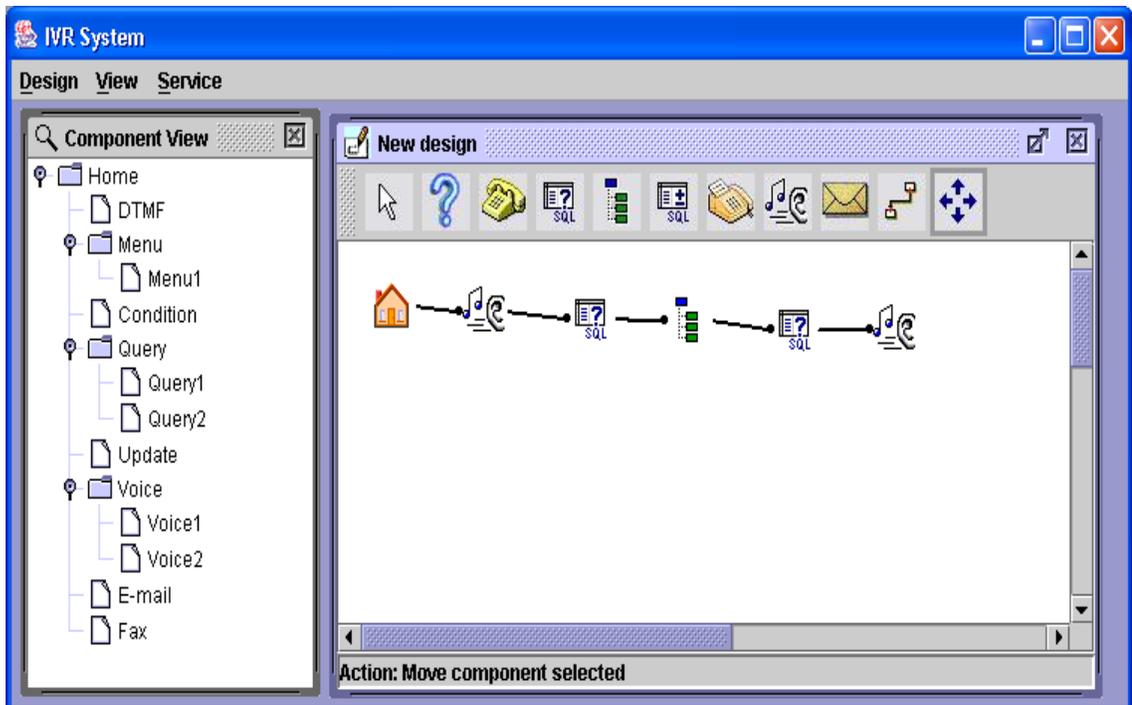


Figura 11. Vista del diseño de un servicio.

- Luego procedemos a guardar el diseño, para esto seleccionamos en el menú *Design* la opción *Save* y lo nombramos (Por ejemplo prueba1.ivr) para proceder a ejecutarlo.

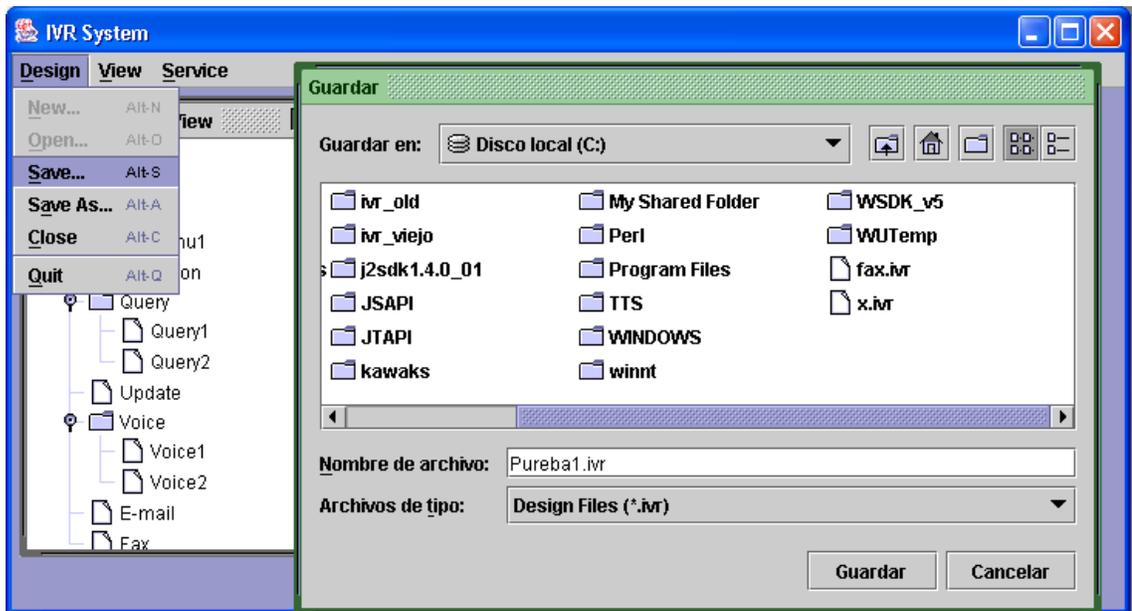


Figura 12. Guardando el diseño de un servicio.

- Para ejecutarlo, en el menú *Service* seleccionamos la opción *Start* y luego seleccionamos el diseño que se ha creado.

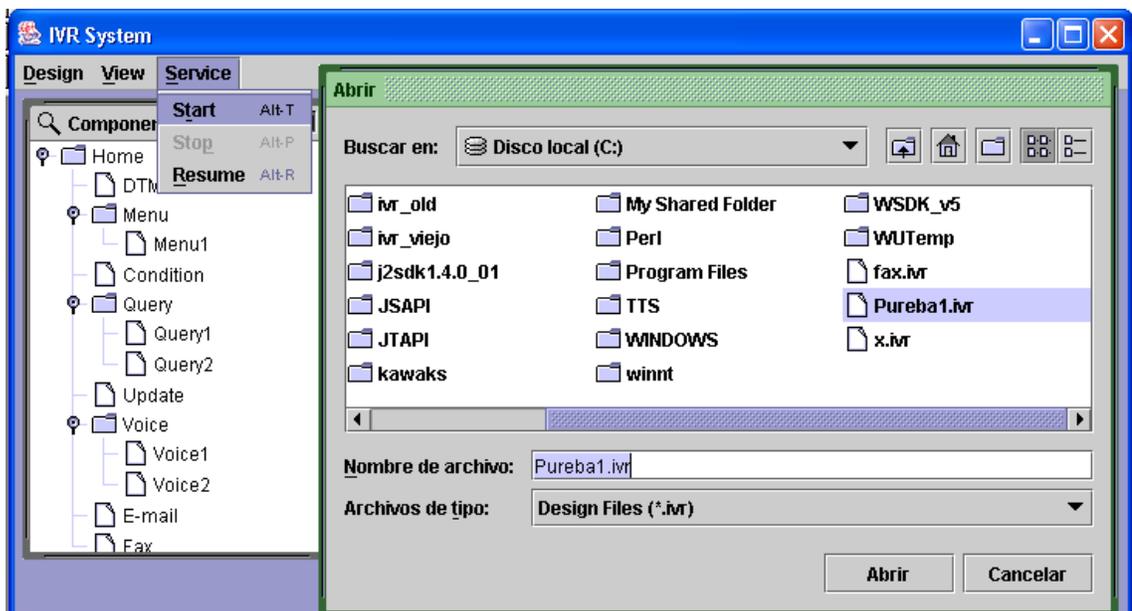


Figura 13. Selección de un diseño para su ejecución.

Una vez se ha seleccionado la opción *abrir*, el sistema esta listo para atender llamadas tomando como base el servicio creado.

Ejemplo 2:

El servicio a crear tiene como fin consultar los estudiantes almacenados en una base de datos para luego seleccionar uno, y al final enviar por correo electrónico los datos del estudiante seleccionado

- Ejecutar el sistema.
- Seleccionar la opción *new* del menú *Design*.
- Para el inicio del sistema se pretende colocar un mensaje de bienvenida, por lo cual es necesario agregar un VoiceComponent en nuestro diseño.
- Una vez el componente ha sido agregado a nuestro diseño, es necesario configurar el mensaje que enviara al usuario. Para hacer esto es necesario, primero seleccionar en la barra de controles, la opción de selección y luego darle doble clic al componente o mediante el botón derecho del Mouse con el cursor sobre el componente seleccionado Edit del menú desplegable, con lo cual aparecerá el cuadro de dialogo del componente (Fig. 14)

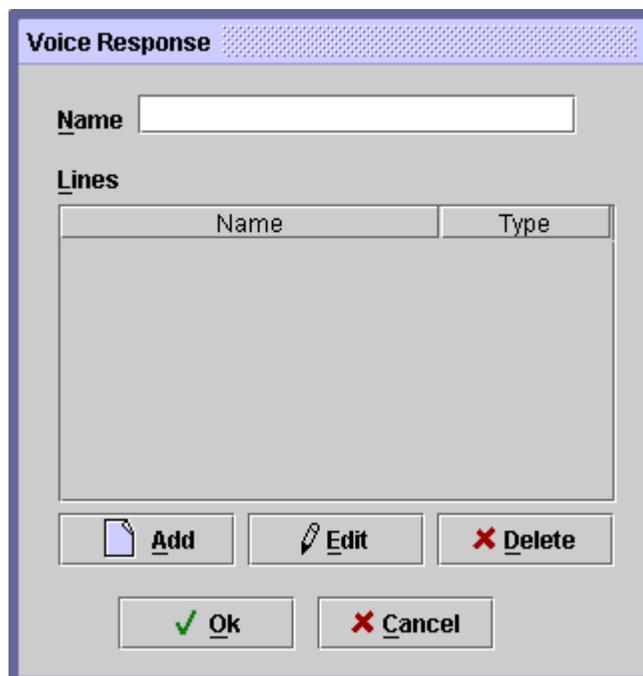


Figura 14. Diálogo de configuración del componente Respuesta de Voz

- Una vez abierto dicho diálogo, es necesario, agregar una línea estática, la cual contendrá el mensaje que se le enviara al usuario. Al dar clic sobre el botón *Add* parece el siguiente cuadro de dialogo:

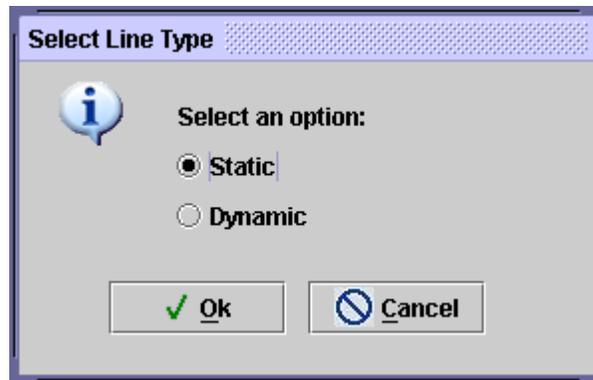


Figura 15. Diálogo para la selección del tipo de línea a agregar.

- Seleccionamos *Static* y nos presenta el diálogo donde colocamos el mensaje de bienvenida (Fig. 16).

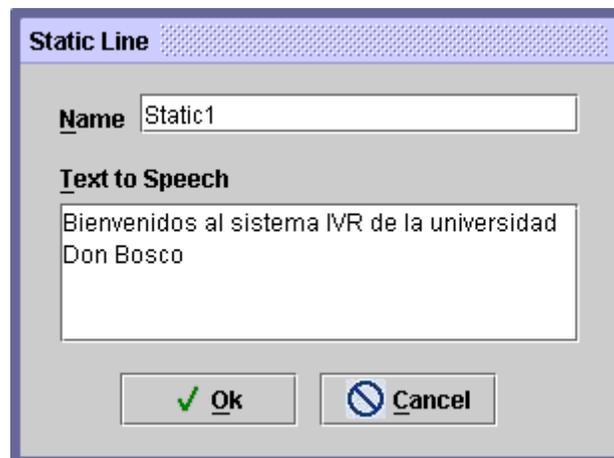


Figura 16. Diálogo para la configuración de una línea estática.

- Luego agregamos un QueryComponent para obtener una lista de estudiantes almacenados en la base de datos

- Configuramos el componente seleccionado el DSN respectivo e ingresamos nuestra consulta (Fig. 17).

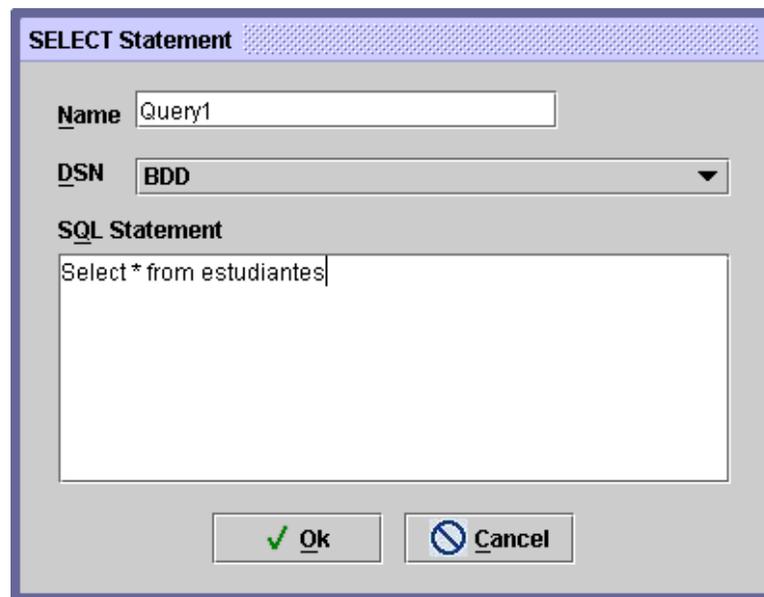


Figura 17. Diálogo para la configuración de un QueryComponent.

- En este momento se procede a crear el menú en el cual se escucharán los distintos nombres de los estudiantes. Para hacer esto procedemos a arrastrar un MyMenuComponent al área de diseño. Luego, seleccionamos que las opciones sean Dinámicas, es decir provenientes de la base de datos:

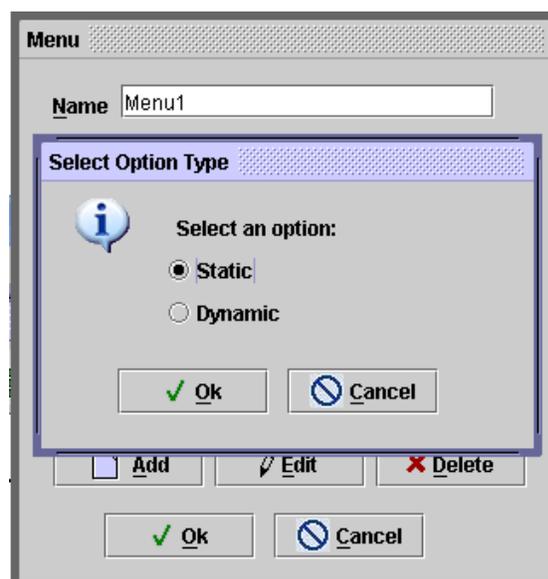
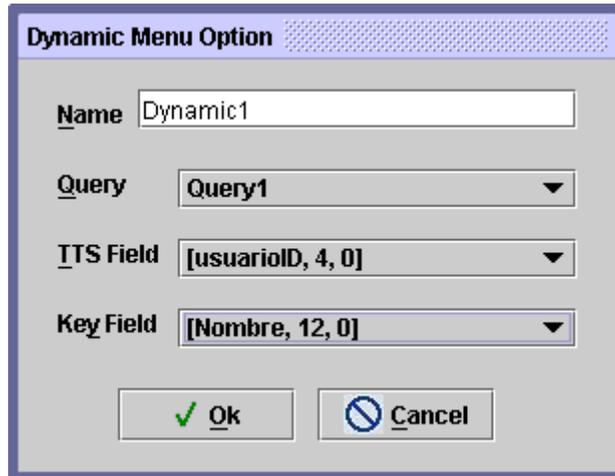


Figura 18. Diálogo para la selección de una opción de menú.

Después de escoger una el tipo dinámico, seleccionamos el Query desde donde obtendremos dichas opciones, y finalmente seleccionamos el campo que sonara y campo clave para identificar la opción que el usuario ha ingresado:



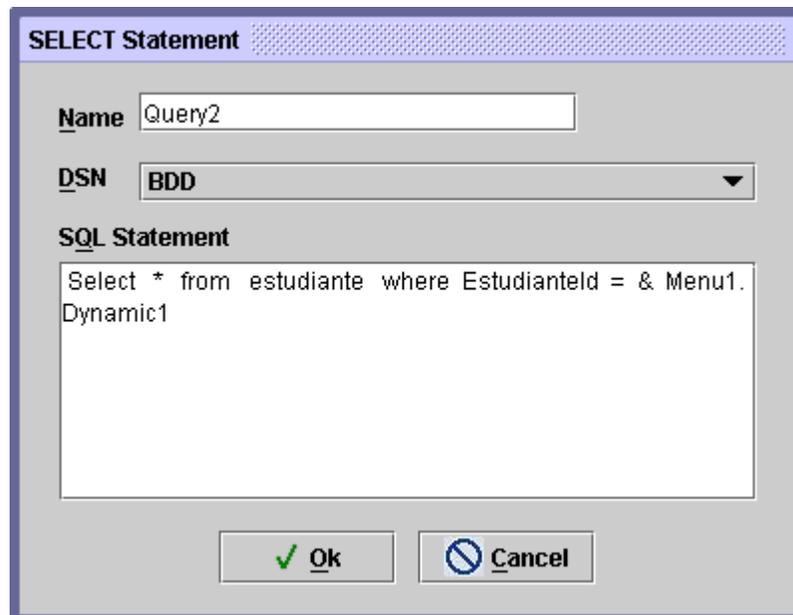
The dialog box titled "Dynamic Menu Option" has the following fields:

- Name:** Dynamic1
- Query:** Query1
- TTS Field:** [usuarioID, 4, 0]
- Key Field:** [Nombre, 12, 0]

Buttons:

Figura 19. Diálogo de configuración de una opción dinámica

- Una vez colocado el menú, debemos obtener los datos de la opción seleccionada, para esto es necesario insertar otro QueryComponent, con el cual obtendremos los datos del estudiante seleccionado en el menú anterior. Luego lo configuramos de la siguiente manera:



The dialog box titled "SELECT Statement" has the following fields:

- Name:** Query2
- DSN:** BDD
- SQL Statement:** Select * from estudiante where Estudianteld = & Menu1. Dynamic1

Buttons:

Figura 20. Dialogo de configuración de un componente SELECT

- Al final basta con agregar el EmailComponent en el cual enviaremos como respuesta al usuario un mensaje conteniendo el nombre y el apellido del el usuario seleccionado en el menú (Fig. 21).

Name Email1

Header

Email Addr...

Query
Query2

Query Field
[Estudianteld, 4, 0]

Content

Query
Query2

Query Fields

Field Name	Selection
[Estudianteld, 4, 0]	<input type="checkbox"/>
[Nombre, 12, 0]	<input checked="" type="checkbox"/>
[Apellido1, 12, 0]	<input checked="" type="checkbox"/>
[Apellido 2, 4, 0]	<input type="checkbox"/>
[Email, 12, 0]	<input type="checkbox"/>
[fax, 12, 0]	<input type="checkbox"/>

Footer

Ok Cancel

Figura 21. Diálogo de configuración de un componente de Email

Se selecciona el Query desde donde se obtiene la dirección de destino a la que va dirigido el correo electrónico, se selecciona el Query con los datos de contenido del correo electrónico y se configuran los encabezados y pie de página que se desean agregar al correo.

Para agregar un encabezado simplemente se da clic en el botón *Header*, y aparece el siguiente cuadro de diálogo:

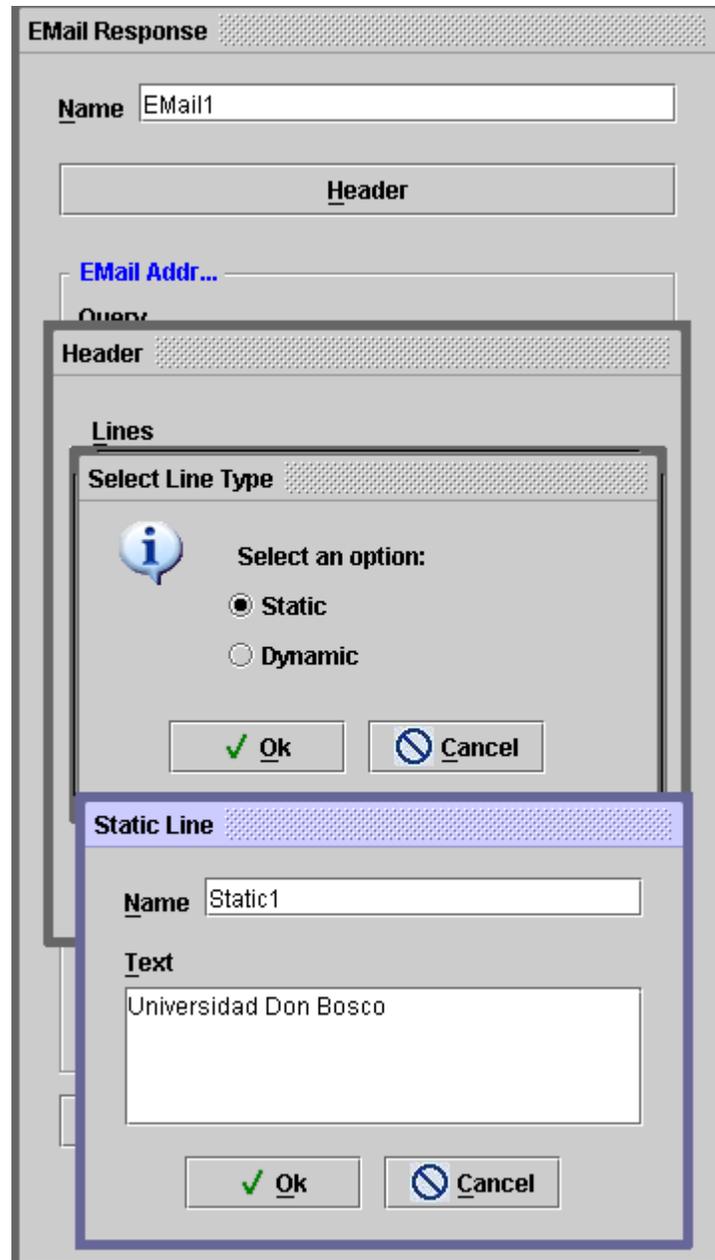


Figura 22. Diálogo de configuración de una línea estática del encabezado de un Email

De la misma manera se configuran los pies de página que se desean en el correo. Simplemente se da clic en el botón Footer, se selecciona el tipo, para este caso estático, y se agrega el mensaje (Fig. 22).

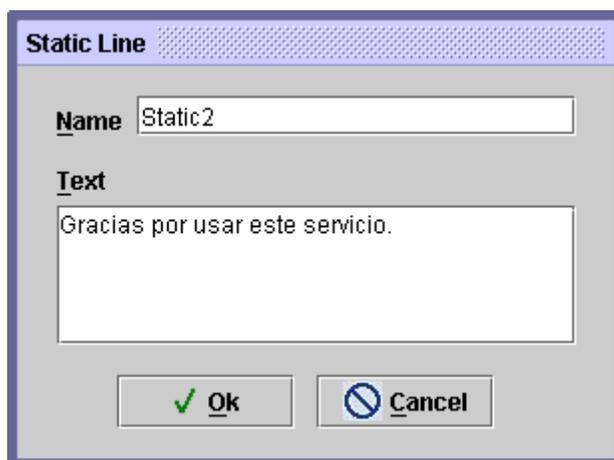


Figura 23. Diálogo de configuración de una línea estática del pie de página de un Email

- Una vez completado este procedimiento el diseño sería como el siguiente:

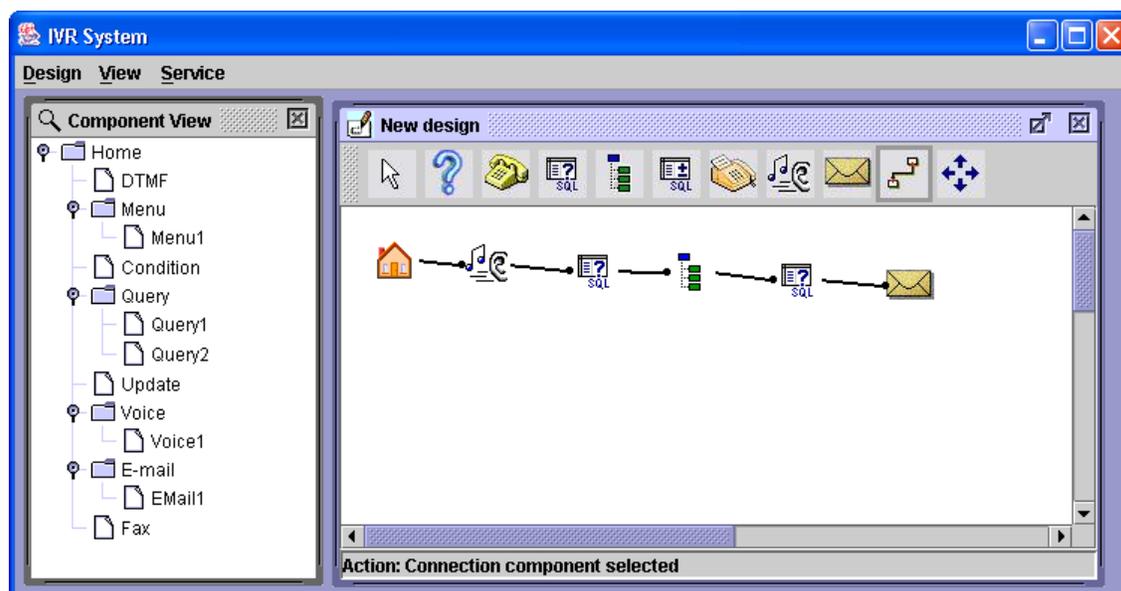


Figura 24. Vista del diseño de un servicio.

- Procedemos a guardar el diseño, para esto seleccionamos en el menú Design la opción Save y lo nombramos (Por ejemplo prueba2.ivr) para proceder a ejecutarlo.
- Para ejecutarlo, en el Service seleccionamos la opción Start y luego seleccionamos en diseño que hemos creado.

Ahora el sistema esta listo para atender llamadas tomando como base el servicio creado.