

**Universidad de las Ciencias Informáticas.  
Facultad 2.**



*Trabajo de Diploma para optar por el título de Ingeniero Informático.*

**Título: “Creación Gráfica de IVR para Asterisk.”**

**Autor:** Yanelay Villamón Costa.

**Tutor:** Ing. Yanerys Gourrie Fernández.

**Co-tutor:** Ing. Arianna Pérez Carmentates.

**Consultor:** Lic. Alain Pérez Balart.

**Ciudad de La Habana, Junio 2011.**

Declaro que soy el único autor de este trabajo y autorizo al Centro de Telemática de la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

Para que así conste firmo la presente a los 16 días del mes de Junio del año 2011.

---

Firma del Autor  
Yanelay Villamón Costa.

---

Firma del Tutor.  
Ing. Yanerys Gourrie Fernández.

---

Firma del Co – tutor.  
Ing. Arianna Pérez Carmenates.

---

Firma del Consultor:  
Lic. Alain Pérez Balart.

**Tutor:**

**Nombre y Apellidos:** Ing. Yanerys Gourrie Fernández.

**Sexo:** F.                    **Institución:** UCI.

**Dirección de la institución:** Carretera a San Antonio de los Baños, km 2 ½, Boyeros, Ciudad de La Habana.

**Correo electrónico:** [ygourrie@uci.cu](mailto:ygourrie@uci.cu)                    **Teléfono del trabajo:** 8373768.

**Título de la especialidad de graduado:** Ingeniera en Ciencias Informáticas.

**Año de graduación:** 2010. **Institución donde se graduó:** Universidad de las Ciencias Informáticas.

**Co - tutor:**

**Nombre y Apellidos:** Ing. Arianna Pérez Carmenates.

**Sexo:** F.                    **Institución:** UCI.

**Dirección de la institución:** Carretera a San Antonio de los Baños, km 2 ½, Boyeros, Ciudad de La Habana.

**Correo electrónico:** [acarmenates@uci.cu](mailto:acarmenates@uci.cu)                    **Teléfono del trabajo:** 8373768.

**Categoría Docente:** Instructor.

**Título de la especialidad de graduado:** Ingeniera en Ciencias Informáticas.

**Año de graduación:** 2008.                    **Institución donde se graduó:** Universidad de las Ciencias Informáticas.

**Consultor:**

**Nombre y Apellidos:** Lic. Alain Pérez Balart.

**Sexo:** M.                    **Institución:** UCI.

**Dirección de la institución:** Carretera a San Antonio de los Baños, km 2 ½, Boyeros, Ciudad de La Habana.

**Correo electrónico:** [alainuci@uci.cu](mailto:alainuci@uci.cu)                    **Teléfono del trabajo:** 8358150.

**Categoría Docente:** Instructor.

**Título de la especialidad de graduado:** Licenciado en Ciencias de la Computación.

**Año de graduación:** 2005                    **Institución donde se graduó:** Universidad de Oriente.



*Si salgo llevo, si llevo entro, si entro triunfo.  
Fidel Castro.*

*El presente Trabajo de Diploma quiero dedicárselo a mis padres, que a ellos les debo todo lo que soy, sus vidas las han consagrado a educarme y pues aquí tienen el fruto de su entrega, de su amor, comprensión y de su esfuerzo por verme hoy hecha una mujer de bien, formada de Ingeniera, una profesional.*

*A mi amiga Tahimí porque éste en sus inicios fue un trabajo de ambas, del equipo que siempre hemos formado juntas y ahora aunque no estuviste junto a mí para discutirlo quiero que sepas que también es tuyo. Nunca olvides que eres una mujer de éxitos.*

*En especial a DIOS por llevarme por el buen camino, por tomar el control de mi vida,  
por hacerme la persona que soy hoy, por darme paz en momentos de desesperación.*

*Gracias a mis padres que me han dado todo el apoyo, el amor y la fuerza para poder  
llegar hasta aquí.*

*Gracias papi por tu confianza, por guiarme correctamente, por saber que te tengo ahí  
para mí, por demostrarme tu amor, porque eres el apoyo y la fuerza que he tenido  
siempre, por ser el mejor papá en todo el mundo.*

*Gracias mami por ser mi amiga por encima de todas las cosas, por sentirme continuamente  
a mi lado, por tu cariño sin límites, por tus palabras siempre en el momento adecuado,  
porque eres el ser que me trajo a la vida.*

*Con mucho amor a esa persona que llegó a mi vida sin ser esperada: mi novio Yasiel, mi  
pareja durante toda esta etapa, mi amigo, mi confidente, gracias por tu ayuda  
incondicional por estar ahí cuando más lo necesité, por los momentos felices, por hacerme  
saber y sentir que me amas, por ser mi solecito. Te amo.*

*A mis abuelos paternos que hoy no están físicamente, pero que los siento en mi corazón y  
que sé que en el lugar donde se hallen están observando este momento.*

*A mi abuela Ana por su cariño, por ser su nieta querida.*

*A mi tío Carlín por ser mi segundo padre, por poder contar contigo, por defenderme y entenderme.*

*A mi familia por el apoyo incondicional, por ser parte de lo que soy.*

*A mi suegra María Elena por cuidar de mí, por abrirme las puertas de su hogar, por quererme.*

*A mi suegro Rafael por poder contar con su ayuda siempre que lo he necesitado.*

*A la familia de mi novio por ser como han sido conmigo, siempre haciéndome sentir parte de ellos.*

*A mi tutora Yanerys Gourrie por dedicarme tiempo, por siempre haberme atendido con paciencia brindándome su ayuda incondicional, por haber estado ahí en los momentos más difíciles de este trabajo de diploma, por su preocupación.*

*A mi Co- tutora Arianna Pérez por aclararme dudas, por sus consejos, por su paciencia.*

*A mi consultor Alain Pérez Balart por sus conocimientos que me han guiado correctamente para la realización de este trabajo.*

*A mi amiga Viviana por ser la hermana que nunca tuve, por estar ahí cuando estoy feliz, cuando estoy triste. Gracias por existir.*

*A mi amiga Tahimí por ayudarme tanto de forma incondicional durante estos cinco años, por tu comprensión, por tus palabras de consuelo, por tu perseverancia, por tu optimismo siempre.*

*A mi amiga Yeny y a sus padres porque sé de su afecto hacia mí, por ayudarme, porque nunca dejé de contar con ustedes.*

*A la moñón, mi amiga Yasmin por auxiliarme en los momentos que te lo he pedido, por otros que te has brindado voluntaria, por preocuparte por mi tesis, por pasarnos ratos conversando mucho.*

*A Elmer por ser un buen amigo, por socorrerme, por estar al tanto de mi docencia, por hacerte cargo de mi equipaje en Venezuela.*

*A mis amistades y compañeros por ser todos un apoyo, una compañía, por ayudarme en los momentos necesitados, por orientarme en la realización de este trabajo de diploma. A Claudia, a Dianelys, Danae, Javier, Denia, Maipú, Rainer, Karelis.*

*A mis compañeros de residencia, a el Gago, gracias por brindarme la laptop siempre que la necesité, a Yandy por invitarnos a su casa y pasar una semana de la victoria inolvidable, a Irma y a Noel por las palizas en el dominó de ambas partes y por enseñarme más acerca de DIOS.*



*A todos los amigos que he tenido el gran placer de conocer durante mi vida.*

*A los buenos profesores de los que he aprendido durante mi labor como estudiante en las anteriores escuelas cursadas y principalmente en la UCI por su cooperación a formarme*

*como profesional. A Alain, Xiomara, Leydis, Marisol de la Caridad Paterson, Yanetsy, Mayda Gil, Carlos Molina, Deivis, Daira, Idalis, Carlos Milian, Denis Buedo, Daimara, Antonio, Maidelis, Erick Pérez, Castillo, Roberto Sarmiento.*

*A mi profesor de teatro César Cutén por haberme ayudado a integrarme como artista aficionada, es un sueño que me llevo cumplido.*

*A mis compañeros de teatro por compartir tan buenos momentos y por aprender siempre de cada uno.*

*Al tribunal de tesis en todo este período por ayudarme a perfeccionar este trabajo con críticas constructivas.*

*A la Federación Estudiantil Universitaria por dejarme ser parte de sus dirigentes estudiantiles, me llevo muchos momentos importantes.*

*A nuestro Comandante en Jefe Fidel Castro por sus ideas.*

*A la Revolución por darme la oportunidad de formarme hoy como profesional.*

**Resumen.**

Las Respuestas de Voz Interactiva (IVR) es un sistema telefónico capaz de interactuar con el humano a través de grabaciones de voz o reconocimiento de respuestas simples, orientado a entregar y/o capturar información a través del teléfono, permitiendo el acceso a determinados servicios.

Las IVR se han convertido en un elemento muy importante para las telecomunicaciones, por ser en muchas ocasiones las que realizan el primer contacto con el cliente, para enrutar de manera adecuada su llamada o dar respuesta a un requerimiento completo.

Actualmente la necesidad de brindar nuevas prestaciones ha desplegado una cantidad notable de servicios de Respuesta de Voz Interactiva, por lo que la creación de IVR es algo que está en constante utilización por las ventajas que brinda para los abonados. En el centro Telemática de la Universidad de las Ciencias Informáticas estas siglas se evidencian casi todo el tiempo, donde un grupo de desarrolladores se encargan de crear IVR con el objetivo de ampliar esta área, principalmente en el territorio nacional. La creación de las mismas resulta engorrosa y compleja ya que se realiza directamente a código, por lo que los desarrolladores deben utilizar un notable tiempo para implementarlas, además poseer conocimientos y estudiarlas. Para facilitar la creación de IVR se decide diseñar un sistema informático, que mediante una interfaz gráfica genere IVR.

Con esta propuesta de sistema se garantiza la comodidad y facilidad de creación de IVR para los desarrolladores, minimizando el tiempo de desarrollo para el proyecto.

**Palabras claves:**

IVR, Call Center, Asterisk.

**Índice de Contenido.**

INTRODUCCIÓN.....	2
CAPÍTULO I: FUNDAMENTACIÓN TEÓRICA.....	8
1.1.    Introducción:.....	8
1.2.    Conceptos Relacionados.....	8
1.2.1.    Asterisk.....	8
1.2.2.    Call Center.....	8
1.2.3.    IVR.....	8
1.2.4.    Intérprete.....	9
1.3.    Análisis de soluciones existentes en el mundo actual.....	9
1.3.1.    Creación de IVR directamente a código.....	9
1.3.2.    Elastix.....	9
1.3.3.    Quickfuse.....	10
1.4.    Metodología de Desarrollo.....	11
1.4.1.    ¿Por qué utilizar XP?.....	12
1.5.    Lenguaje de desarrollo.....	13
1.5.1.    Lenguajes de desarrollo en el lado del Cliente.....	13
1.5.2.    Lenguaje de desarrollo en el lado del Servidor.....	15
1.6.    Herramientas.....	15
1.6.1.    Dreamweaver.....	15
1.6.2.    Zend Studio.....	16
1.6.3.    Herramienta CASE.....	16
1.7.    Servidor Web.....	17
1.7.1.    Apache.....	18
1.8.    Conclusiones:.....	18
CAPÍTULO II: EXPLORACIÓN Y PLANIFICACIÓN.....	19

2.1. Introducción.....	19
2.2. Objeto de Automatización.....	19
2.3. Propuesta del sistema .....	19
2.3.1. ¿Por qué Aplicación Web y no Aplicación de Escritorio? .....	19
2.4. Características del Sistema.....	20
2.4.1. Características funcionales del sistema.....	20
2.4.2. Características no funcionales del sistema.....	21
2.5. Arquitectura.....	21
2.5.1. Arquitectura Cliente – Servidor.....	22
2.6. Fase Exploración.....	23
2.6.1. Flujo de Procesos del Sistema Propuesto.....	23
2.6.2. Personas Relacionadas con el Sistema.....	26
2.6.3. Historias de Usuario.....	26
2.7. Planificación.....	33
2.7.1. Estimación de esfuerzo por historias de usuarios.....	33
2.7.2. Plan de Iteraciones.....	34
2.7.3. Plan de Duración de las Iteraciones.....	34
2.7.4. Plan de Entregas.....	35
2.8. Conclusiones Parciales.....	35
<b>CAPÍTULO III: DISEÑO DEL SISTEMA.....</b>	<b>36</b>
3.1. Introducción.....	36
3.2. Patrones de Diseño .....	36
3.2.1. Patrones GOF (Gang of Four, Banda de Cuatro).....	36
3.2.2. Patrones generales de software para asignación de responsabilidades (GRASP).....	37
3.3. Tarjetas de Cargo o Clase, Responsabilidad y Colaboración (CRC).....	38
3.4. Tareas de Ingeniería.....	42
3.4.1. Tareas de Ingeniería correspondiente a la Historia de Usuario # 1.....	42

Para ver todas las Tareas de Ingeniería, ver Anexos # 8, 9, ..., 36.....	43
3.5. Conclusiones Parciales.....	43
<b>CAPÍTULO IV: EVALUACIÓN DE LA PROPUESTA.....</b>	<b>44</b>
4.1 Introducción.....	44
4.2 Método Delphi.....	44
4.2.1 Características del Método Delphi.....	44
4.2.2 Proceso de Selección de expertos.....	45
4.2.3 Elaboración y realización de las encuestas.....	48
4.2.4 Conclusiones de la validación por método Delphi.....	50
4.3 Pruebas.....	51
4.3.1 Pruebas en el análisis.....	51
4.3.2 Pruebas Unitarias.....	51
4.3.3 Pruebas de Aceptación.....	52
4.4 Casos de Prueba.....	53
4.4.1 Diseño de casos de prueba de la Historia de Usuario # 1.....	53
4.5 Conclusiones Parciales.....	54
<b>CONCLUSIONES.....</b>	<b>55</b>
<b>RECOMENDACIONES.....</b>	<b>56</b>
<b>TRABAJOS CITADOS.....</b>	<b>57</b>
<b>BIBLIOGRAFÍA.....</b>	<b>59</b>
<b>ANEXOS.....</b>	<b>62</b>
Figura 11. Anexo 1. Interfaz Diseñar IVR.....	62
Figura 12. Anexo 2. Interfaz Salvar IVR.....	63
Figura 13. Anexo 3. Interfaz Validar IVR.....	64
Figura 14. Anexo 4. Interfaz Listar IVR.....	65

Figura 15. Anexo 5. Interfaz Eliminar IVR. ....	66
Figura16. Anexo 6. Interfaz Mostrar IVR Gráficamente. ....	67
Figura 17 Anexo 7. Interfaz Modificar IVR. ....	68
Tareas de Ingeniería correspondiente a la Historia de Usuario # 1. ....	69
Anexo 8. Tabla 38 Tarea de Ingeniería # 2: Componente Colgar. ....	69
Anexo 9. Tabla 39 Tarea de Ingeniería # 3: Componente Escribir mensaje. ....	69
Anexo 10. Tabla 40 Tarea de Ingeniería # 4: Componente Subir archivo de voz. ....	70
Anexo 11. Tabla 41 Tarea de Ingeniería # 5: Componente Grabar mensaje. ....	70
Anexo 12. Tabla 42 Tarea de Ingeniería # 6: Componente Reproducir mensaje. ....	70
Anexo 13. Tabla 43 Tarea de Ingeniería # 7: Componente Conector. ....	71
Anexo 14. Tabla 44 Tarea de Ingeniería # 8: Componente Condicional múltiple. ....	71
Anexo 15. Tabla 45 Tarea de Ingeniería # 9: Componente Condicional. ....	72
Anexo 16. Tabla 46 Tarea de Ingeniería # 10: Componente Recibir dígitos. ....	72
Anexo 17. Tabla 47 Tarea de Ingeniería # 11: Componente comparar dígitos. ....	73
Anexo 18. Tabla 48 Tarea de Ingeniería # 12: Componente Transferir llamada. ....	73
Anexo 19. Tabla 49 Tarea de Ingeniería # 13: Componente Tiempo de espera. ....	74
Tareas de Ingeniería correspondiente a la Historia de Usuario # 2. ....	74
Anexo 20. Tabla 50 Tarea de Ingeniería # 14: Agregar componentes. ....	74
Anexo 21. Tabla 51 Tarea de Ingeniería # 15: Editar componentes. ....	75
Anexo 22. Tabla 52 Tarea de Ingeniería # 16: Eliminar componentes. ....	75
Anexo 23. Tabla 53 Tarea de Ingeniería # 17: Enlazar componentes. ....	75
Tareas de Ingeniería correspondiente a la Historia de Usuario # 3. ....	76
Anexo 24. Tabla 54 Tarea de Ingeniería # 18: Generar código php. ....	76
Anexo 25. Tabla 55 Tarea de Ingeniería # 19: Generar posiciones de componentes. ....	76
Tareas de Ingeniería correspondiente a la Historia de Usuario # 4. ....	77
Anexo 26. Tabla 56 Tarea de Ingeniería # 20: Salvar fichero .php. ....	77

Anexo 27. Tabla 57 Tarea de Ingeniería # 21: Salvar fichero .conf. ....	77
Tareas de Ingeniería correspondiente a la Historia de Usuario # 5. ....	78
Anexo 28. Tabla 58 Tarea de Ingeniería # 22: Validar sintaxis de la IVR. ....	78
Anexo 29. Tabla 59 Tarea de Ingeniería #23: Validar valores de los componentes.....	79
Tareas de Ingeniería correspondiente a la Historia de Usuario # 6. ....	79
Anexo 30. Tabla 60 Tarea de Ingeniería # 24: Listar IVR. ....	79
Tareas de Ingeniería correspondiente a la Historia de Usuario # 7. ....	80
Anexo 31. Tabla 61 Tarea de Ingeniería # 25: Eliminar IVR. ....	80
Tareas de Ingeniería correspondiente a la Historia de Usuario # 8. ....	80
Anexo 32. Tabla 62 Tarea de Ingeniería # 26: Leer fichero .php. ....	80
Anexo 33. Tabla 63 Tarea de Ingeniería # 27: Leer fichero .conf. ....	81
Anexo 34. Tabla 64 Tarea de Ingeniería # 28: Crear objeto JSON.....	81
Tareas de Ingeniería correspondiente a la Historia de Usuario # 9. ....	81
Anexo 35. Tabla 65 Tarea de Ingeniería # 29: Mostrar IVR gráficamente .....	81
Tareas de Ingeniería correspondiente a la Historia de Usuario # 10. ....	82
Anexo 36. Tabla 66 Tarea de Ingeniería # 30: Modificar IVR. ....	82
Anexo 37. Encuesta para la selección de los expertos.....	83
Anexo 38. Tabla de resultado del coeficiente de competencia realizada a los expertos.....	84
Anexo 39. Expertos seleccionados. ....	84
Diseño de casos de prueba de la Historia de Usuario # 2. ....	86
Anexo 41. Tabla 71 Prueba # 2.1: Diseñar IVR.....	86
Diseño de casos de prueba de la Historia de Usuario # 3. ....	86
Anexo 42. Tabla 72 Prueba # 3.1: Generar IVR. ....	86
Diseño de casos de prueba de la Historia de Usuario # 4. ....	87
Anexo 43. Tabla 73 Prueba # 4.1: Salvar IVR correcta. ....	87
Anexo 44. Tabla 74 Prueba # 4.2: Salvar IVR correcta sin nombre.....	88
Anexo 45. Tabla 75 Prueba # 4.3: Salvar IVR correcta sin dirección.....	88

Anexo 46. Tabla 76 Prueba # 4.4: Salvar IVR correcta sin nombre ni dirección. ....	89
Anexo 47. Tabla 77 Prueba # 4.5: Salvado de IVR correcta cancelado. ....	90
Anexo 48. Tabla 78 Prueba # 4.6: Salvar IVR incorrecta. ....	91
Anexo 49. Tabla 79 Prueba # 4.7: Salvar IVR incorrecta sin nombre. ....	92
Anexo 50. Tabla 80 Prueba # 4.8: Salvar IVR incorrecta sin dirección. ....	93
Anexo 51. Tabla 81 Prueba # 4.9: Salvar IVR incorrecta sin nombre ni dirección. ....	94
Anexo 52. Tabla 82 Prueba # 4.10: Salvado de IVR incorrecta cancelado. ....	95
Anexo 53. Tabla 83 Prueba # 4.11: IVR incorrecta sin salvar. ....	95
Diseño de casos de prueba de la Historia de Usuario # 5. ....	96
Anexo 54. Tabla 84 Prueba # 5.1: Validar IVR incorrecta. ....	96
Anexo 55. Tabla 85 Prueba # 5.2: Validar IVR correcta. ....	96
Diseño de casos de prueba de la Historia de Usuario # 6. ....	97
Anexo 56. Tabla 86 Prueba # 6.1: Listar IVR. ....	97
Diseño de casos de prueba de la Historia de Usuario # 7. ....	97
Anexo 57. Tabla 87 Prueba # 7.1: Eliminar IVR. ....	97
Anexo 58. Tabla 88 Prueba # 7.2: Eliminado de IVR cancelado. ....	98
Anexo 59. Tabla 89 Prueba # 7.3: Eliminar otra IVR. ....	99
Diseño de casos de prueba de la Historia de Usuario # 8. ....	99
Anexo 60. Tabla 90 Prueba # 8.1: Interpretar código de IVR. ....	99
Diseño de casos de prueba de la Historia de Usuario # 9. ....	100
Anexo 61. Tabla 91 Prueba # 9.1: Mostrar IVR gráficamente. ....	100
Diseño de casos de prueba de la Historia de Usuario # 10. ....	101
Anexo 62. Tabla 92 Prueba # 10.1: Modificar IVR. ....	101
Anexo 63. Tabla 93 Prueba # 10.2: Modificado de IVR cancelado. ....	101
Anexo 64. Tabla 94 Prueba # 10.3: Modificar otra IVR. ....	102
GLOSARIO DE TÉRMINOS. ....	103



## **Introducción.**

Desde la antigüedad, el hombre siempre ha tenido la necesidad de comunicarse. La comunicación entre las personas es un fenómeno donde se han desarrollado múltiples métodos de expresión que han contribuido al desarrollo del lenguaje y a lograr un entendimiento entre un grupo de individuos. Los psiquiatras suelen llamar comunicación a todo procedimiento por medio del cual una mente puede influir de alguna forma sobre otra. Con el desarrollo de la civilización y de las lenguas escritas, surge también la comunicación a distancia con el objetivo de facilitar el desarrollo del comercio entre las diferentes naciones. Poco a poco se han ido perfeccionando los disímiles medios de comunicación, desde el papel, la impresión y los servicios postales hasta el surgimiento del teléfono, las telecomunicaciones e internet.

Así, desde los inicios de la especie, la comunicación fue evolucionando hasta llegar a la más sofisticada tecnología, para lograr acercar espacios y tener mayor velocidad en el proceso.

Las Telecomunicaciones, del prefijo griego tele (distancia o lejos), significan comunicación a distancia y abarcan toda transmisión, emisión o recepción de signos, señales, datos o imágenes, voz, sonido o información de cualquier naturaleza que se efectúa a través de cables, radioelectricidad, medios ópticos, físicos u otros sistemas electromagnéticos. (1) Es una técnica que consiste en la transmisión de un mensaje desde un punto a otro, usualmente con la característica adicional de ser bidireccional.

A nivel mundial, la dinámica de las telecomunicaciones en el mundo ha provocado la creación de los Centros de Atención de Llamadas, más conocidos en inglés como Call Center y actualmente las mismas han tenido un desarrollo vertiginoso principalmente por la contribución de la microelectrónica, la óptica, la ciencia del espacio, la computación e informática, entre otras. Cada una de estas áreas ha proveído a las telecomunicaciones de distintos soportes tecnológicos: mayor capacidad de transmisión, mejores materiales conductores, diversificación de interconexión, capacidad de manejo y transferencia de diferentes signos, señales e imágenes, simultáneamente.

Cuba, a pesar de ser subdesarrollada y del tercer mundo, no se ha quedado atrás y ha empleado recursos en el desarrollo de las telecomunicaciones. En la actualidad la Empresa de Telecomunicaciones de Cuba S.A. (ETECSA) ha logrado incrementar en diez veces la cantidad de teléfonos instalados desde su creación. Se completó la red de fibra óptica que ha permitido que la tele selección nacional llegue a todas las provincias del país y ha comenzado a prestar mayores servicios en el área de la telefonía pública. Se completaron acuerdos con Venezuela para la construcción de un nuevo cable submarino de 1552 KM, y una capacidad de 180 GB/S, que pondrá a Cuba en un alto nivel de comunicaciones con el resto del

mundo. Además la División de Servicios Móviles de ETECSA, CUBACEL S.A se dedicó a prestar los servicios de comunicación celular en toda la isla.

La Universidad de las Ciencias Informáticas (UCI) se ha convertido en una potente empresa productora de software en el país, la misma fue creada por el Comandante en Jefe Fidel Castro en el año 2002, con el objetivo de contribuir al desarrollo informático del país y crear profesionales altamente calificados y comprometidos con la Revolución Socialista. En ella se realizan diversos proyectos de software para comercializar con otros países e informatizar el territorio nacional. Estos proyectos están organizados en centros productivos con un perfil específico. El centro de Telemática, perteneciente a la facultad 2 desarrolla un proyecto que ofrece el servicio de montaje de Call Center en la hermana República Bolivariana de Venezuela, dentro del país y en la propia Universidad.

Uno de los servicios que brinda un Call Center es generar Respuesta de Voz Interactiva, más conocida en inglés como Interactive Voice Response (IVR) que consiste en un sistema telefónico que es capaz de recibir una llamada e interactuar con el ser humano a través de grabaciones de voz y el reconocimiento de respuestas simples, como “sí”, “no”, u otras. (2)

Los sistemas automatizados de reconocimiento de voz permiten a los llamantes usar directamente palabras o frases que estos sistemas reconocen y convierten en comandos que se comunican de forma interactiva con aplicaciones informáticas.

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 o por respuesta hablada y en función de cada respuesta, la aplicación IVR realiza una serie de acciones sobre la base de datos.

Los sistemas de respuesta vocal interactiva son de uso indispensable en los servicios de atención de llamadas, ya que proporcionan una descongestión necesaria en dichos servicios.

Los sistemas IVR proporcionan grandes beneficios para los centros de atención al cliente como son:

- Reducción de tiempos de esperas.
- Se comportan como receptores de llamadas en horarios fuera de atención al público.
- Aumento de la disponibilidad del servicio.
- Identificación y verificación de la identidad del usuario llamante.

En la Universidad de las Ciencias Informáticas, específicamente en el Centro de Telemática la creación y manejo de estas IVR son algo cotidiano que permanentemente está en constante utilización.

Actualmente en el mundo y en Cuba la creación de estas IVR se realiza directamente a código, el cuál será interpretado por Asterisk (sistema que simula una planta telefónica). Cada IVR se implementa de manera muy específica en dependencia de las características propias de cada una, por lo que la creación de las mismas resulta muy engorrosa, compleja, incómoda y con notable pérdida de tiempo tanto para los desarrolladores como para el proyecto, además los desarrolladores deben tener amplios conocimientos para poder desarrollar la inmensa variedad de IVR que se requieren a medida que las Telecomunicaciones se desarrollan y surge la necesidad de brindar nuevos servicios.

Todo lo antes mencionado demuestra una situación problemática que define el siguiente **problema a resolver**: ¿Cómo facilitar el proceso de creación de las IVR para un sistema Asterisk?

Para responder a esta interrogante surge como **objeto de estudio**: Los procesos de creación de IVR.

Para complementar el objeto de estudio se tiene como **campo de acción**: Los procesos de creación de IVR para un sistema Asterisk.

Se plantea como **objetivo general**: Diseñar una aplicación informática que permita generar IVR a través de una interfaz gráfica para un sistema Asterisk.

Para cumplir el objetivo planteado se proponen las siguientes **tareas de investigación**:

- Análisis y conceptualización de la información que existe sobre la creación de IVR en la Universidad y en el mundo.
- Identificación de sistemas que creen IVR en el mundo y estudiar sus principales características y funcionalidades.
- Identificar las funcionalidades que pueden contener una IVR.
- Análisis del sistema Asterisk para identificar los mecanismos al ejecutar las IVR.
- Definición y propuestas de las herramientas, lenguajes y metodologías a emplear en el desarrollo de la aplicación.
- Definir cada una de las funcionalidades correspondientes a la aplicación.
- Definir las clases del sistema para luego poder obtener una correcta implementación.
- Realizar un prototipo no funcional para complementar el diseño de la aplicación.

- Realizar pruebas al análisis para verificar la correcta definición de las funcionalidades correspondientes a la aplicación.
- Realizar un diseño de casos de pruebas para dejar plasmado el procedimiento de como tiene que responder la aplicación ante cada una de sus funcionalidades.

Teniendo en cuenta lo anterior se realiza un diagnóstico sobre la problemática a resolver detectando la necesidad de una nueva herramienta que garantice la efectividad del proceso de creación de las IVR, de una forma personalizada, simple y asequible para los desarrolladores por lo que se define como **idea a defender**:

El proceso de creación de IVR para el sistema Asterisk puede simplificarse con el diseño de una aplicación informática que permita generar IVR a través de una interfaz gráfica.

Los **métodos teóricos** utilizados fueron:

**El Analítico – Sintético** para poder establecer comparaciones de criterios y determinar los elementos comunes y generales de los criterios considerados que permitieron arribar a conclusiones parciales y finales sobre las IVR.

La **Inducción – Deducción** permite a partir del estudio de hechos aislados se pueda arribar a proposiciones generales y que a partir de conocimientos generales se puedan inferir casos particulares teniendo como punto de partida un razonamiento lógico.

El **Histórico – Lógico** en la búsqueda de una visión holística del fenómeno, buscando su condicionamiento, la lógica interna de su desarrollo, la esencia teórica del fenómeno en cuestión. Es necesario el estudio de la historia de las IVR, para poder comprender la lógica y el funcionamiento de las mismas.

También se hizo uso de **métodos empíricos**:

El **Análisis Documental** permitió apreciar muchos aspectos importantes contenidos en los documentos que aparecen en internet sobre las IVR, todos de actualidad y confiables, los que posibilitaron llegar a conclusiones certeras del tema objeto de estudio.

La **Entrevista**: La entrevista es una conversación planificada entre el investigador y el entrevistado para obtener información. (3) Se realizó al líder del proyecto, el Ing. Erick Pérez Castillo, para conocer cómo se desarrollan las IVR, que conocimientos posee sobre la existencia de las mismas en el mundo, que funcionalidades considera que no deben faltar al sistema objeto de investigación. Además se entrevistó a

estudiantes del Proyecto Call Center que están desarrollando IVR actualmente, para conocer qué aspectos consideran que les dificultan en la realización de las mismas.

**Observación:** El investigador conoce el problema y el objeto de investigación, estudiando su curso natural, sin alteración de las condiciones naturales, es decir que la observación tiene un aspecto contemplativo. Podemos entender la observación como el registro visual de lo que ocurre en una situación real. (4) El tipo de observación utilizada fue la no estructurada, simple o libre. Según Alfredo Díaz Hurbe esta observación es cuando el observador solo desea guiarse por líneas muy generales en la observación, para de ahí desprender con libertad los aspectos específicos que se quieran estudiar con mayor detenimiento. Aunque aquí el observador no se vale de instrumentos para su trabajo, sí lleva consigo, por lo menos, una guía general y simple que le permita conocer como punto de partida, las generalidades del caso que se va a investigar. Se realizó a desarrolladores del proyecto Call Center del centro Telemática de la Universidad de las Ciencias Informáticas, durante la creación de IVR para valorar cómo se realizan las mismas.

**Encuesta:** Se utiliza cuando la información que se realiza puede ser obtenida a partir de la respuesta de una o un grupo de personas, a través del uso de un cuestionario pre elaborado. Se realizaron encuestas a siete expertos en el tema con el objetivo de obtener una garantía sobre la calidad de la propuesta realizada.

El presente documento tiene la siguiente estructura:

**Capítulo 1:** “Fundamentación teórica”: Este capítulo contiene la base teórica para entender el problema planteado, donde se abordan algunos conceptos asociados al dominio del problema para facilitar su entendimiento. Se presenta el análisis realizado para seleccionar la metodología y la herramienta CASE<sup>1</sup> que se utiliza para el desarrollo ingenieril del Módulo de Creación Gráfica de IVR para Asterisk. Además se proponen los lenguajes de desarrollo y las herramientas donde debe ser implementado el sistema.

**Capítulo 2:** “Exploración y Planificación”: Este capítulo contiene la propuesta del sistema, además de la definición de arquitectura, los requerimientos no funcionales y el flujo de eventos del sistema, los requisitos funcionales a través de las Historias de Usuario. En este capítulo se define una planificación por iteraciones para obtener los resultados.

---

<sup>1</sup> *Computer Aided Software Engineering*, Ingeniería de Software Asistida por Computadora.

**Capítulo 3:** “Diseño del Sistema”: Este capítulo contiene todo el diseño del sistema propuesto. Se definen los patrones de diseño así como las clases del negocio a través de las Tarjetas de Cargo o Clase, Responsabilidad y Colaboración. Además se realizan las tareas de Ingeniería.

**Capítulo 4:** “Evaluación de la Propuesta”: Este capítulo contiene una evaluación mediante el uso de los criterios de un panel de expertos y el empleo de técnicas propuestas por el método Delphi. Además del diseño de casos de prueba planteado por la metodología XP, así como las pruebas para validar la correcta selección de los requisitos funcionales.

## Capítulo I: Fundamentación Teórica.

### 1.1. Introducción:

En este primer capítulo se analizan conceptos relacionados con el sistema propuesto, se desarrolla un estudio profundo sobre la situación actual de las IVR en el mundo, en Cuba y en la Universidad de las Ciencias Informáticas, se especifica la metodología de desarrollo escogida, los lenguajes de desarrollo y las herramientas que se proponen utilizar para el futuro desarrollo de una aplicación.

### 1.2. Conceptos Relacionados.

Para un mejor entendimiento del problema a resolver es necesario explicar algunos términos sobre telecomunicaciones. Estos términos se muestran a continuación:

#### 1.2.1. Asterisk.

Proyecto desarrollado por Mark Spencer, miembro fundador de la compañía *Digium*. Es un programa de software libre que convierte un ordenador en una central telefónica multifuncional. Se encarga de integrar funcionalidades de telefonía clásica con nuevas capacidades derivadas de su flexible y potente arquitectura, pudiendo conectar un número determinado de teléfonos para hacer llamadas entre sí.

#### 1.2.2. Call Center.

Un Call Center o Centro de Atención de Llamadas entrantes o salientes es una herramienta de comunicación y relación con los clientes que utiliza el teléfono como medio de comunicación básico, gestionado por personas en conjunto a los recursos humanos, físicos y tecnológicos disponibles, basados en metodologías de trabajo, procesos determinados para atender las necesidades y dar servicios a cada cliente único con el objetivo de atraerlos a la organización. Son operados por una compañía proveedora de servicios que se encarga de administrar, proveer soporte y asistencia al consumidor según los productos, servicios o información necesitada. Puede ser operado independientemente o puede estar interconectado con otros centros, generalmente conectados a una corporación computarizada.

Un Call Center es aquel que provee a la empresa de los elementos necesarios para, con un servicio centralizado vía telefónica, establezca relaciones de mutuo beneficio, con sus clientes y proveedores. (5)

#### 1.2.3. IVR.

Es una solución de última generación para operadores de telefonía fija y móvil destinada a ofrecer sistemas inteligentes de pre atención, autogestión y diálogo interactivo con los abonados. Es un sistema

automatizado de respuesta interactiva orientado a entregar y/o capturar información a través del teléfono, permitiendo el acceso a determinados servicios. Los sistemas automatizados de reconocimiento de voz permiten a los llamantes usar directamente palabras o frases que estos sistemas reconocen y convierten en comandos que se comunican de forma interactiva con aplicaciones informáticas. (6)

#### **1.2.4. Intérprete.**

En las ciencias de la computación, intérprete o interpretador es un programa informático capaz de ejecutar y analizar otros programas escritos en un lenguaje de alto nivel. Los intérpretes realizan la traducción a medida que sea necesaria, típicamente, instrucción por instrucción. Permiten ofrecer al programa interpretado un entorno no dependiente de la máquina donde se ejecuta el intérprete, sino del propio intérprete, esto se conoce comúnmente como máquina virtual. (7)

### **1.3. Análisis de soluciones existentes en el mundo actual.**

Un servicio de Respuesta de Voz Interactiva (IVR) es una solución de última generación para operadores de telefonía fija y móvil destinada a ofrecer sistemas inteligentes de pre atención, autogestión y diálogo interactivo con los abonados que mejoran la productividad de las compañías, minimizando costos y servicios. Actualmente en la mayoría de los países del mundo se brinda el servicio de IVR, pero sin embargo, existen muy pocos sistemas que permiten crearlas para la inmensa utilidad que tienen.

#### **1.3.1. Creación de IVR directamente a código.**

Varios países como Cuba crean las IVR directamente a código para que luego, el mismo sea interpretado por Asterisk, lo cual no es muy factible por lo que se explicó anteriormente en la situación problemática, pero a pesar de estos inconvenientes es la forma más utilizada para la creación de las mismas.

#### **1.3.2. Elastix.**

Fue creado por la compañía ecuatoriana Palo Santo Solutions. Es un sistema de manejo para comunicaciones que compone varios servicios, es una distribución libre de Servidor de Comunicaciones Unificadas que integra en un solo paquete:

- VoIP<sup>2</sup>PBX<sup>3</sup>.
- Fax.

---

<sup>2</sup>*Voice over Internet Protocol*, Protocolo de Internet sobre Voz.

<sup>3</sup>*Private Branch Exchange*, Central Secundaria Privada.

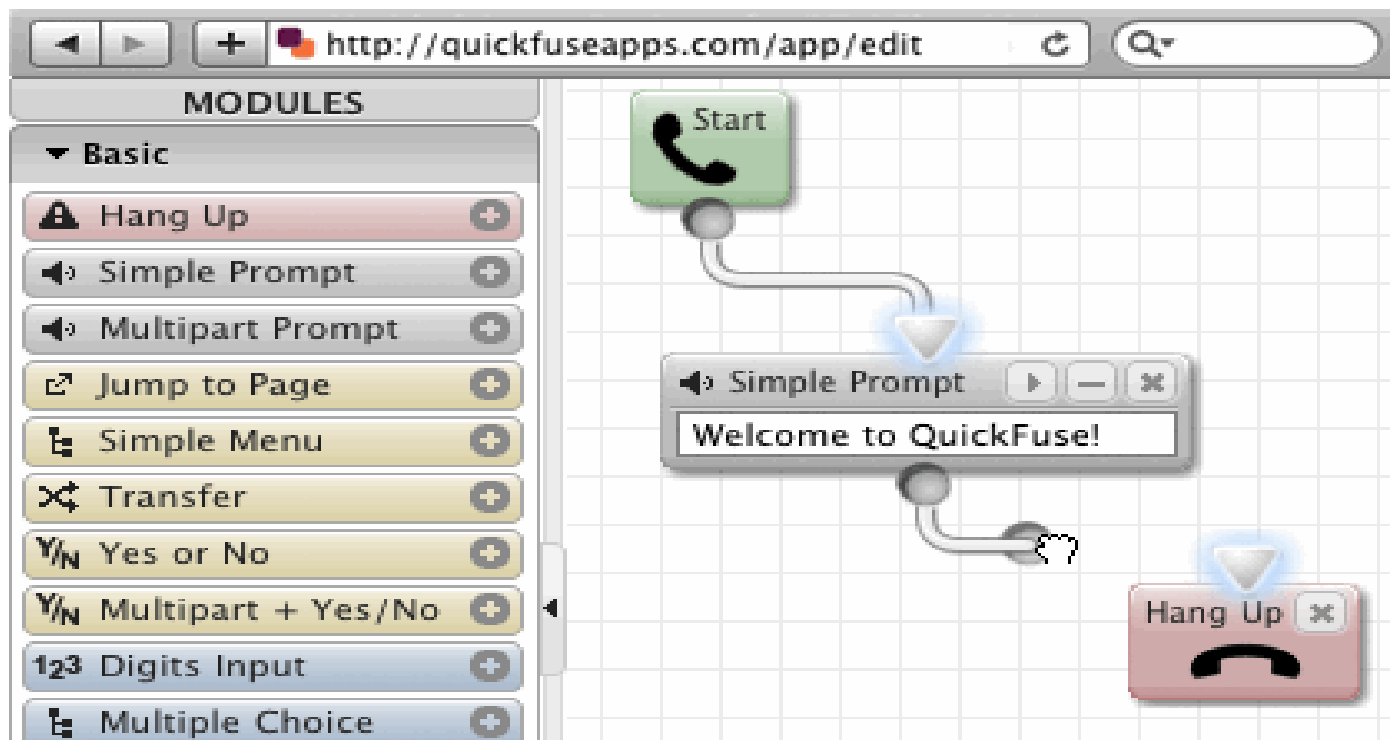


- Mensajería Instantánea.
- Correo electrónico.
- Colaboración.

Elastix es un sistema muy utilizado en la Universidad de las Ciencias Informáticas. Brinda entre sus prestaciones generar IVR, pero tiene como desventaja en esta área las pocas opciones que proporciona por lo que las IVR son muy poco personalizables, no contienen elementos gráficos, ni es factible su uso precisamente porque no posibilita especificar muchas de las características que se desea que contenga la IVR.

### 1.3.3. Quickfuse.

Existe una aplicación gráfica que genera IVR de una forma sencilla y personalizada, a la cual puede acceder cualquier usuario, la misma se encuentra en la siguiente dirección: <http://quickfuseapps.com>.



**Figura 1** Aplicación Web que crea IVR gráficamente.

Esta aplicación de una forma muy simple y solo arrastrando componentes permite la creación de las IVR, conteniendo muchísimas opciones, pero tiene como desventaja principal que no es libre. Luego de diseñar

una IVR brinda solamente la opción de salvar la misma, después de este vínculo el sistema no permite mostrar el código generado y se desconoce el destino de la Respuesta de Voz Interactiva, o sea que el creador de la IVR nunca podrá utilizarla.

Por lo tanto no existe una única aplicación que integre todas las características deseadas:

- Libre.
- Ahorre tiempo para el desarrollo del proyecto.
- Simplifique el trabajo para los desarrolladores.
- Integre funcionalidades variadas para poder crear una IVR, que permitan personalizar la misma.

Por lo que se hace necesario el diseño de un sistema con estos requisitos.

#### **1.4. Metodología de Desarrollo.**

Una metodología de desarrollo de software es un conjunto de procedimientos, técnicas, herramientas y un soporte documental que ayuda a los desarrolladores guiando la realización del nuevo software. No existe una metodología estándar a utilizar en todos los procesos de desarrollo de software.

Para seleccionar una metodología hay que tener en cuenta las características del proyecto que se va a desarrollar. Las metodologías pueden clasificarse en tradicionales y ágiles, las primeras recalcan el uso exhaustivo de documentación durante todo el ciclo del proyecto, imponen una disciplina de trabajo sobre el proceso de desarrollo del software, con el fin de conseguir un software más eficiente. Algunos ejemplos de estas metodologías son:

- RUP (Rational Unified Process, Proceso Racional Unificado).
- MSF (Microsoft Solutions Framework, Marco de Trabajo de Soluciones de Microsoft).

Las metodologías ágiles ponen vital importancia en la capacidad de respuesta a los cambios y en mantener una buena relación con el cliente para llevar al éxito el proyecto. Estas últimas se utilizan para equipos de desarrollo que lo integran pocas personas y para proyectos de corto plazo. Algunos ejemplos de estas metodologías son:

- Crystal Methodologies, Metodologías de Cristal.
- Dynamic System Development Method, Método de Desarrollo de Sistemas Dinámicos (DSDM).
- Adaptive Software Development, Desarrollo de Software Adaptable (ASD)
- Feature – Driven Development, Desarrollo Basado en funcionalidades (FDD)

Una de las metodologías ágiles de desarrollo de software más exitosa en la actualidad es Extreme Programming (XP).



**Figura 2 Flujo de trabajo de la Metodología XP.**

#### **1.4.1. ¿Por qué utilizar XP?**

La programación extrema es una metodología ligera de desarrollo de software que se basa en la simplicidad, la comunicación y la realimentación o reutilización del código desarrollado. (8)

Se resuelve el uso de la misma para el desarrollo de este trabajo de diploma por ser una metodología muy simple, ágil, utilizada para proyectos de corto plazo, pequeño equipo y breve tiempo de entrega.

Esta metodología consiste en una programación rápida o extrema, cuya particularidad es tener como parte del equipo al usuario final, pues es uno de los requisitos para llegar al éxito del proyecto. La simplicidad es la base de XP. Se simplifica el diseño para agilizar el desarrollo y facilitar el mantenimiento. XP, se basa en el trabajo orientado directamente al objetivo, basándose para esto en las relaciones interpersonales y en la velocidad de reacción para la implementación y para los cambios que puedan surgir durante el desarrollo del proceso.

XP se caracteriza porque los individuos e interacciones son más importantes que los procesos y herramientas. Un software que funcione es más importante que la documentación exhaustiva, los documentos deben ser cortos y centrarse en lo fundamental. La colaboración con el cliente es más importante que la negociación de contratos. La respuesta ante un cambio es más importante que el seguimiento de un plan, por lo tanto, la planificación debe ser flexible y abierta.

Con estas características se decide que XP es la metodología más apropiada para el proyecto que se propone. Su ciclo de vida consiste de seis fases:

- Exploración: Los clientes plantean en las historias de usuarios lo que constituye la principal prioridad para la primera versión del producto.
- Planificación de la Entrega (Release): El cliente establece la prioridad de cada historia de usuario, datos que son usados por los programadores para realizar una estimación del esfuerzo necesario de cada una de ellas.
- Iteraciones: Incluye varias iteraciones sobre el sistema antes de ser entregado. El Plan de Entrega está compuesto por iteraciones de no más de tres semanas.
- Producción: En esta fase se realizan pruebas adicionales y revisiones de rendimiento antes de que el sistema sea llevado al entorno del cliente.
- Mantenimiento: Se plantean las tareas de soporte para el cliente. La fase de mantenimiento puede requerir adicionar personal al equipo y realizar cambios en su estructura.
- Muerte del Proyecto: Esta etapa es de suma importancia pues es cuando el cliente no tiene más historias para ser incluidas en el sistema. Lo que requiere que se satisfagan las necesidades del cliente en otros aspectos como rendimiento y confiabilidad del sistema. (9)

## **1.5. Lenguaje de desarrollo.**

Un lenguaje de desarrollo es un idioma artificial que interpreta la computadora. Generalmente se utilizan para crear programas que controlan el funcionamiento de un fenómeno determinado. Estos lenguajes están formados por un conjunto de símbolos y reglas sintácticas y semánticas que definen su estructura.

### **1.5.1. Lenguajes de desarrollo en el lado del Cliente.**

#### **1.5.1.1. HTML.**

Se propone utilizar como lenguaje de programación en el lado del cliente HTML, siglas de Hyper Text Markup Language (Lenguaje de Marcado de Hipertexto) para la creación de las páginas. HTML es el lenguaje de marcado predominante para la elaboración de las páginas web. (10) Se utiliza para detallar la estructura y el contenido en forma de texto y complementarlo con objetos tales como imágenes. Puede describir, hasta cierto punto, la apariencia de un documento, y puede incluir un script, por ejemplo Java Script.

### 1.5.1.2. Java Script –ExtJS.

Para la validación de los datos en el lado del cliente se propone utilizar el lenguaje Java Script integrado con el framework ExtJS.

Java Script es un lenguaje de programación interpretado, se define como orientado a objetos, basado en prototipos, imperativo y dinámico. Se utiliza principalmente en su forma del lado del cliente, implementado como parte de un navegador web permitiendo mejoras en la interfaz de usuario y páginas web dinámicas.

(11)

Todos los navegadores modernos interpretan el código Java Script integrado en las páginas web. Además se propone Java Script ya que tiene como subconjunto de su notación literal de objetos a JSON<sup>4</sup>, el cual no es más que un formato ligero para el intercambio de datos por lo que se propone también su utilización para un correcto funcionamiento del sistema a desarrollar. La simplicidad de JSON ha dado lugar a la generalización de su uso, es fácil para los seres humanos a leer y escribir. Es fácil para las máquinas para analizar y generar. JSON es un lenguaje de texto que es completamente independiente del lenguaje.

JSON está construido en dos estructuras:

- Una colección de pares, nombre/valor. En varios idiomas, esto se realiza como un objeto, registro, estructura, diccionario, tabla hash, lista con llave o una matriz asociativa.
- Una lista ordenada de valores. En la mayoría de idiomas, esto se realiza como una matriz, vector, una lista o secuencia.

ExtJS es un framework Java Script que se utiliza como base para la programación avanzada de aplicaciones y aporta un grupo de funciones o códigos para realizar tareas. Un framework es el conjunto de librerías de código que contienen procesos o rutina listos para usarse. Se refiere a una estructura software compuesta de componentes personalizables e intercambiables para el desarrollo de una aplicación. (12) Además ExtJS ofrece una infraestructura que brinda mayor facilidad para la creación de aplicaciones complejas del lado del cliente. Java Script con ExtJS aporta una interfaz para programación que permite efectos con el navegador que funcionarán para todos los visitantes. Tiene un sistema dual de licencias: comercial y código abierto. Posee un conjunto de ventajas como son:

- Código reutilizable.
- Orientada a la programación de interfaces tipo desktop en la web.

---

<sup>4</sup> *JavaScript ObjectNotation*, Notación de Objetos de JavaScript.

Dispone de un conjunto de componentes para incluir dentro de una aplicación web, como:

- Cuadros y áreas de texto.
- Campos numéricos.
- Combos.
- Editor HTML.
- Paneles divisibles en secciones.

### **1.5.2. Lenguaje de desarrollo en el lado del Servidor.**

#### **1.5.2.1. PHP.**

Del lado del servidor se propone utilizar PHP (Hypertext Pre-processor, Pre- Procesador de Hipertexto), que es un lenguaje de programación interpretado, diseñado originalmente para la creación de páginas web dinámicas. (13) PHP corre en casi cualquier plataforma utilizando el mismo código fuente, pudiendo ser compilado y ejecutado en algo así como 25 plataformas, incluyendo diferentes versiones de Unix, Windows y Macs. Es un lenguaje completamente orientado a la web. Posee una amplia documentación en su página oficial, es libre por lo que se presenta como una alternativa de fácil acceso para todos. PHP es de código abierto, lo que significa que el usuario no depende de una compañía específica para arreglar cosas que no funcionan y no está forzado a pagar actualizaciones anuales para tener una versión que funcione. Permite las técnicas de Programación Orientada a Objetos, contiene una biblioteca de funciones sumamente amplia y manejo de excepciones.

### **1.6. Herramientas.**

#### **1.6.1. Dreamweaver.**

Se decide utilizar la herramienta Dreamweaver para diseñar las páginas que forman parte de la vista, la misma proporciona una potente combinación de herramientas visuales de diseño, funciones de desarrollo de aplicaciones y soporte para la edición del código, características que permiten a los desarrolladores crear los sitios web con una rapidez increíble. Proporciona las herramientas profesionales que requiere un entorno integrado y agilizado.

### **1.6.2. Zend Studio.**

Como herramienta para programar la lógica del negocio se resuelve utilizar Zend Studio o Zend Development Environment, el mismo es un completo entorno de desarrollo integrado para el lenguaje de programación PHP, además es un editor de texto para páginas PHP, que proporciona un gran número de ayudas desde la creación y gestión de proyectos hasta la depuración del código. Una de sus principales ventajas es la de disponer de un debugger para PHP, que permite ejecutar las aplicaciones paso a paso y también ofrece completamiento de código, así como herramientas para la programación. Está escrito en Java, y está disponible para las plataformas Microsoft Windows, Mac OS X y GNU/Linux.

### **1.6.3. Herramienta CASE.**

Las herramientas CASE (Ingeniería de Software Asistida por Computadoras) son el mejor soporte para el proceso de desarrollo de software. Son diversas aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de software. Estas herramientas pueden ayudar en todos los aspectos del ciclo de vida de desarrollo del software en tareas como el proceso de realizar un diseño del proyecto. Son un sistema de software que intenta proporcionar ayuda automatizada a las actividades del proceso de software.

#### **1.6.3.1. Bizagi.**

Se decide utilizar Bizagi ya que es la herramienta por la cual los procesos de negocio son modelados, configurados, organizados e implementados. Es la solución líder BPM (Gestión de Procesos de Negocio, Model Process Bussines) que permite diseñar, integrar, automatizar y monitorear los procesos de negocio por medio de un ambiente gráfico. Es una aplicación utilizada para realizar diagramas y documentar los procesos utilizando el estándar BPMN (representación gráfica para la especificación de los procesos de negocio en un modelo de proceso de negocio).

#### **1.6.3.2. Notación de modelado de procesos de negocio o Business Process Modeling Notation (BPMN).**

Es una notación que modela los procesos de negocio, basada en diagramas de flujo fácil de entender. El objetivo principal de los esfuerzos de BPMN era dar una notación rápidamente comprensible por toda esa gente de negocios, desde el analista de negocio que hace el borrador inicial de los procesos, pasando por los desarrolladores técnicos responsables de implementar la tecnología que llevarán a cabo dichos

procesos, llegando finalmente a la gente de negocio que gestionará y monitorizará esos procesos. (14) Está diseñada para cubrir varios tipos de modelado, permite la creación tanto de segmentos de procesos, como procesos de negocio de comienzo a fin en diferentes niveles de representatividad. BPMN es gráficamente más rico, con menos símbolos fundamentales, pero con más variaciones de éstos, lo que facilita su comprensión por parte de personas no expertas.

- Provee una notación que es fácilmente entendida por todos los usuarios.
- Crea un puente estandarizado para el vacío existente entre el diseño del proceso de negocio y su implementación.
- Asegura que los lenguajes para la ejecución de los procesos de negocio puedan ser visualizados con una notación común.

### 1.7. Servidor Web.

Un servidor web o HTTP es un programa que procesa cualquier aplicación del lado del servidor realizando conexiones bidireccionales y/o unidireccionales y síncronas o asíncronas con el cliente generando o cediendo una respuesta en cualquier lenguaje o aplicación del lado del cliente. (15) Un servidor web opera mediante el protocolo HTTP, de la capa de aplicación del Modelo OSI<sup>5</sup>. Se ejecuta en un ordenador manteniéndose a la espera de peticiones por parte de un cliente (un navegador web) y que responda a estas peticiones adecuadamente, mediante una página web que se exhibirá en el navegador o mostrando el respectivo mensaje si se detectó algún error.

Los servidores web pueden entregar también aplicaciones web, éstas son porciones de código que se ejecutan cuando se realizan ciertas peticiones o respuestas HTTP. Su forma de comportamiento en dichas aplicaciones es el siguiente:

Aplicaciones en el lado del cliente: El cliente web es el encargado de ejecutarlas en la máquina del usuario. El servidor proporciona el código de las aplicaciones al cliente y éste, mediante el navegador, las ejecuta.

Aplicaciones en el lado del servidor: El servidor web ejecuta la aplicación; ésta, una vez ejecutada, genera cierto código HTML; el servidor toma este código recién creado y lo envía al cliente por medio del protocolo HTTP.

---

<sup>5</sup> *Open System Interconnection*, Modelo de Interconexión de Sistemas Abiertos.



### 1.7.1. Apache.

Se propone utilizar el servidor web Apache, ya que es un servidor web http de código abierto para plataformas Unix, Windows y otras, que implementa el protocolo http. Apache es un servidor flexible, rápido y eficiente, continuamente actualizado y adaptado a los nuevos protocolos, es el servidor HTTP más usado. Las principales ventajas que brinda este servidor son las siguientes:

- Multiplataforma.
- Modular: puede ser adaptado a diferentes entornos y necesidades, con los diferentes módulos de apoyo que proporciona.
- Es personalizable, la arquitectura modular de Apache permite construir un servidor hecho a la medida. Además permite la implementación de los últimos y más nuevos protocolos.
- Incentiva la realimentación de los usuarios, obteniendo nuevas ideas, informes de fallos y parches para la solución de los mismos.
- Se desarrolla de forma abierta.
- Extensible: Gracias a ser modular se han desarrollado diversas extensiones entre las que se destaca PHP.
- En cuanto a la administración los archivos de configuración de Apache están en ASCII<sup>6</sup>, por lo que tiene un formato simple, y pueden ser editados tan solo con un editor de texto. Estos son transferibles, lo que permite la clonación efectiva de un servidor. El servidor puede ser administrado vía línea de comandos, lo que hace la administración remota muy conveniente.

### 1.8. Conclusiones:

En este capítulo se realizó un estudio del arte donde se abordan las principales características que tienen las IVR que existen actualmente y la necesidad de crear una herramienta que proporcione mayores ventajas para la confección de las mismas. Se propusieron las principales herramientas, metodologías y lenguajes a utilizar en el desarrollo de una aplicación que permita la creación de IVR de forma gráfica, entre las cuales podemos especificar metodología XP, lenguajes de programación del lado del cliente HTML y Java Script con ExtJS, del lado del servidor PHP y como herramientas se utilizan Dreamweaver, Zend Studio y BizAgi.

---

<sup>6</sup>*American Standard Code for Information Interchange*, Código Estadounidense Estándar para el Intercambio de Información.

## Capítulo II: Exploración y Planificación.

### 2.1. Introducción.

En el presente capítulo se abordan los temas relacionados con las fases de exploración y planificación de la metodología de desarrollo XP, así como se elabora una propuesta del sistema a desarrollar, se explican y se describen las características principales del mismo, se confeccionan las historias de usuario que proporcionan un mayor entendimiento y comprensión del sistema, se presentan los principales artefactos generados, y además se hace un análisis de la estimación del esfuerzo por cada historia de usuario.

### 2.2. Objeto de Automatización.

La creación de IVR es un proceso que necesita ser automatizado ya que la forma actual de su realización resulta muy engorrosa, compleja, incómoda y con notable pérdida de tiempo tanto para los desarrolladores como para el proyecto. Por lo tanto se desea analizar y diseñar una aplicación que permita crear las IVR de forma gráfica brindando comodidad, facilidad y optimización del tiempo para los desarrolladores.

### 2.3. Propuesta del sistema

Se propone el análisis y diseño de un sistema informático que permita la creación de IVR con el objetivo principal de facilitar el desarrollo de las mismas. Este sistema se plantea como una aplicación web, a la cual podrá acceder el cliente y a través de componentes gráficos y relaciones desarrollar la IVR que desee.

#### 2.3.1. ¿Por qué Aplicación Web y no Aplicación de Escritorio?

Una aplicación Web es un sistema informático que los usuarios utilizan accediendo a un servidor web a través de Internet o de una Intranet mediante un navegador. Es decir, es una aplicación de software que se codifica en un lenguaje soportado por los navegadores web en la que se confía la ejecución al navegador.

La razón primordial por la que este sistema se propone como una aplicación web es con la idea futura de integrarlo a la plataforma web de Elastix, obteniendo como resultado un sistema que hará más completo el funcionamiento de esta plataforma, ya que está siendo muy utilizado en el centro de Telemática por ser la mejor solución libre que brinde comunicaciones unificadas. La arquitectura implementada en Elastix corresponde con la escogida para la aplicación propuesta, aunque no necesariamente tiene que ser la

misma, ya que Elastix lo que hace es incorporar en una única solución todos los medios y alternativas de comunicación existentes en el ámbito empresarial.

A partir de este juicio se plantea desarrollar una aplicación web, que además brinda muchas ventajas como se mencionan a continuación:

- Las aplicaciones web son populares debido a que el navegador web como cliente ligero es muy práctico y no hay que instalar ningún programa con CD<sup>7</sup>, descargar o actualizar el software para su utilización. Sus actualizaciones se hacen de una manera muy sencilla, sin necesidad de hacer descargas, instalaciones o comprar físicamente el producto.
- Para su ejecución simplemente basta con teclear su dirección URL<sup>8</sup> en cualquier navegador web.
- No depende de ningún sistema operativo ni configuración de hardware específica.
- Facilidad del control de acceso.
- Naturalmente multiusuario.
- Son aplicaciones multiplataforma por lo que los usuarios no tienen que usar un sistema operativo en específico.
- Ahorran costes de hardware y software.
- Facilitan el trabajo colaborativo y a distancia. (16)

#### **2.4. Características del Sistema.**

Una característica del sistema no es más que una condición o necesidad de un usuario para resolver un problema o alcanzar un objetivo. También se puede decir que es una capacidad que debe estar presente en un sistema para satisfacer un contrato, estándar, especificación u otro documento formal. Las características del sistema pueden dividirse en funcionales y no funcionales.

##### **2.4.1. Características funcionales del sistema.**

Las características funcionales son características requeridas del sistema que expresan una capacidad de acción del mismo – una funcionalidad; generalmente expresada en una declaración en forma verbal. (17)

---

<sup>7</sup> *Compact Disc*, Disco Compacto.

<sup>8</sup> *Uniform Resource Locator*, Localizador de Recurso Uniforme.

#### 2.4.1.1 Funcionalidades del sistema.

Para describir los servicios que se espera que el sistema propuesto suministre se definieron diez funcionalidades, las cuales se detallan en las Historias de Usuario del presente capítulo.

#### 2.4.2. Características no funcionales del sistema.

Las características no funcionales del sistema tienen que ver con características que de una u otra forma puedan limitar el sistema. Se aplican al sistema en su totalidad.

##### 2.4.2.1. Software.

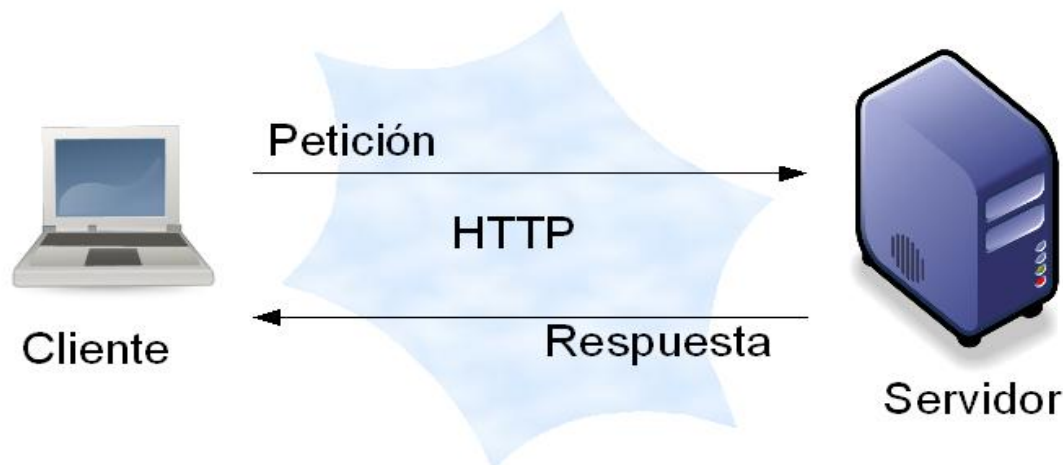
Considerar como característica técnica mínima que en las computadoras del cliente se requiera un sistema operativo con un navegador web Mozilla Firefox, Internet Explorer o cualquier otro que brinde soporte para Java Script. En el servidor de aplicaciones web se necesita un servidor instalado con un intérprete del lenguaje PHP.

#### 2.5. Arquitectura.

La Arquitectura no es más que una vista estructural de alto nivel que define los estilos o grupos de estilos adecuados para cumplir con las características no funcionales de un software. Es el resultado de ensamblar un cierto número de elementos arquitectónicos para satisfacer determinados requisitos o funcionalidades. Existen varios tipos de arquitecturas donde se puede mencionar el modelo vista controlador, SOAP<sup>9</sup>, la arquitectura n-capas, cliente/servidor entre otras. La arquitectura de software es importante como disciplina debido a que los sistemas de software crecen de forma tal que resulta muy complicado que sean diseñados, especificados y entendidos por un solo individuo. (18) Para el desarrollo de la aplicación se propone una arquitectura cliente – servidor , estructurada en dos capas.

---

<sup>9</sup>Simple **ObjectAccess Protocol**, Protocolo de Acceso de Objetos Simples.



**Figura 3 Arquitectura Cliente – Servidor.**

### **2.5.1. Arquitectura Cliente – Servidor.**

La arquitectura cliente – servidor consiste básicamente en un cliente que realiza peticiones a otro programa (el servidor) que le da respuesta.

En esta arquitectura el remitente de una solicitud es conocido como cliente. Sus principales características son:

- Es quien inicia solicitudes o peticiones.
- Espera y recibe las respuestas del servidor.
- Normalmente interactúa directamente con los usuarios finales mediante una interfaz gráfica de usuario.

Al receptor de la solicitud enviada por el cliente se conoce como servidor. Sus principales características son:

- Al iniciarse esperan a que lleguen las solicitudes de los clientes.
- Tras la recepción de una solicitud, la procesan y luego envían la respuesta al cliente.
- Por lo general, aceptan conexiones desde un gran número de clientes.
- No es frecuente que interactúen directamente con los usuarios finales.

Se proponen dos capas para proporcionar una forma efectiva de separación de responsabilidades. Las capas que se proponen son las siguientes:

- Capa de presentación: Es la que ve el usuario, presenta el sistema al usuario, le comunica la información y captura la información del usuario en un mínimo de proceso. También conocida como interfaz gráfica. Esta capa se comunica únicamente con la capa de negocio.
- Capa de negocio: Es donde residen los programas que se ejecutan, se reciben las peticiones del usuario y se envían las respuestas tras el proceso. También se conoce como lógica del negocio, ya que es aquí donde se establecen todas las reglas que deben cumplirse. Esta capa se comunica con la capa de presentación, para recibir las solicitudes y presentar los resultados.

## **2.6. Fase Exploración.**

En esta fase, los clientes plantean a grandes rasgos las historias de usuario que son de interés para la primera entrega del producto. Al mismo tiempo el equipo de desarrollo se familiariza con las herramientas, tecnologías y prácticas que se utilizarán en el proyecto. Se prueba la tecnología y se exploran las posibilidades de la arquitectura del sistema construyendo un prototipo.

### **2.6.1. Flujo de Procesos del Sistema Propuesto.**

El usuario solicita la acción que desea realizar, Diseñar, Editar o Eliminar IVR.

- Diseñar IVR: El usuario interactúa con el sistema permitiendo adicionar y editar los componentes que conformarán la IVR. Luego puede seleccionar Salvar IVR donde el sistema genera el código correspondiente a cada componente y lo guarda en ficheros.
- Editar IVR: El sistema muestra un listado de las IVR salvadas en el servidor web donde el cliente escoge la IVR a modificar. El sistema a través del intérprete lee el código fuente y busca estructuras sintácticas que correspondan con los componentes que conforman la IVR y lo almacena en un objeto JSON que envía a la PC<sup>10</sup> cliente donde se cargan los componentes de forma gráfica. El cliente sigue trabajando en el diseño de la IVR.
- Eliminar IVR: El sistema muestra un listado de las IVR salvadas en el servidor web donde el cliente escoge la IVR a eliminar. El sistema elimina los ficheros relacionados con la IVR seleccionada.

---

<sup>10</sup> *Personal Computer*, Computadora Personal.

Para un mayor entendimiento de lo expuesto anteriormente ver las imágenes que a continuación se muestran:

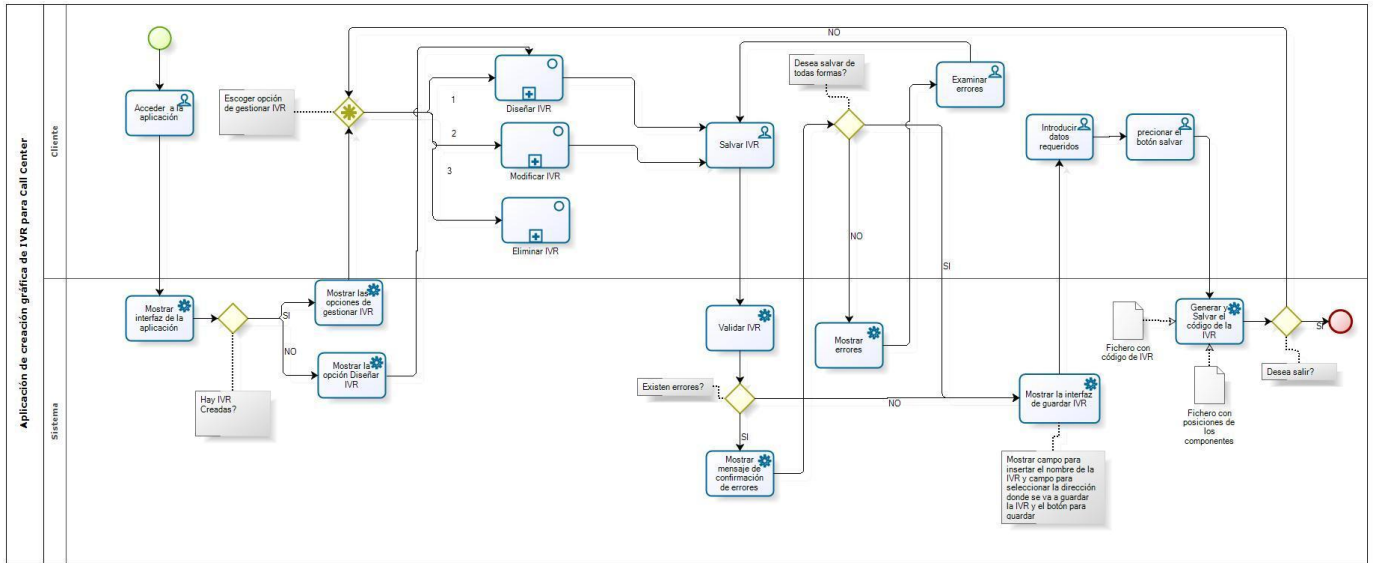


Figura 4 Proceso del sistema Aplicación de creación gráfica de IVR para Asterisk.

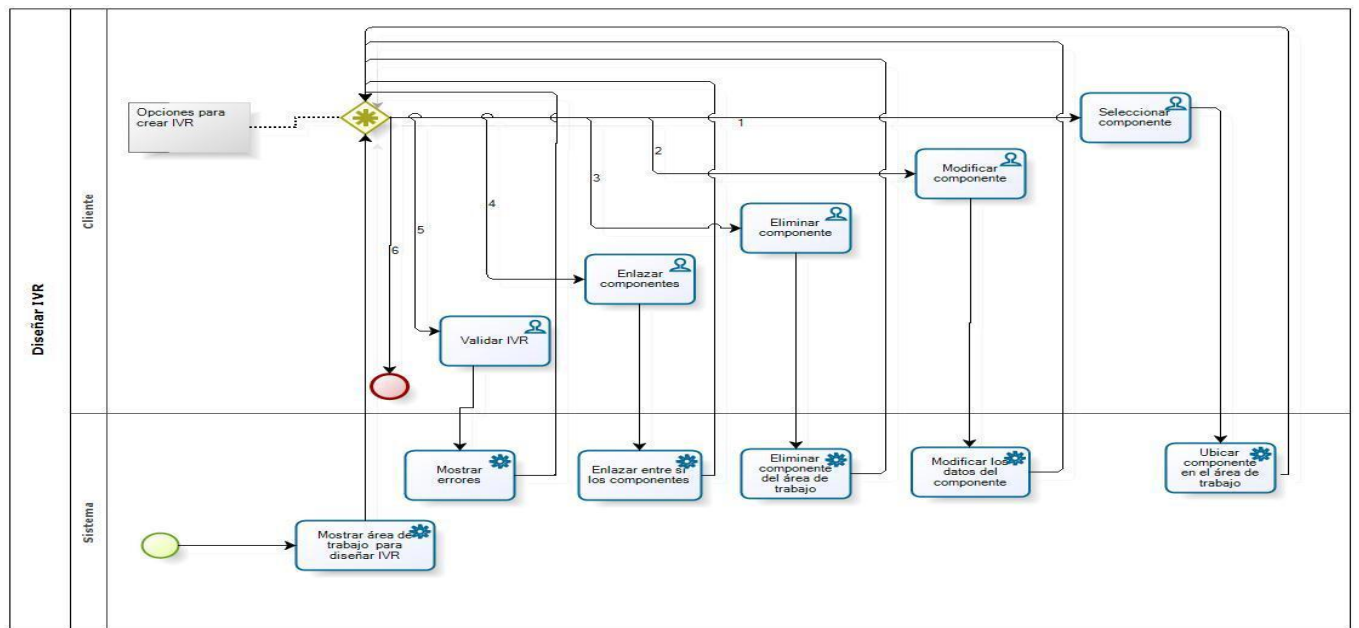
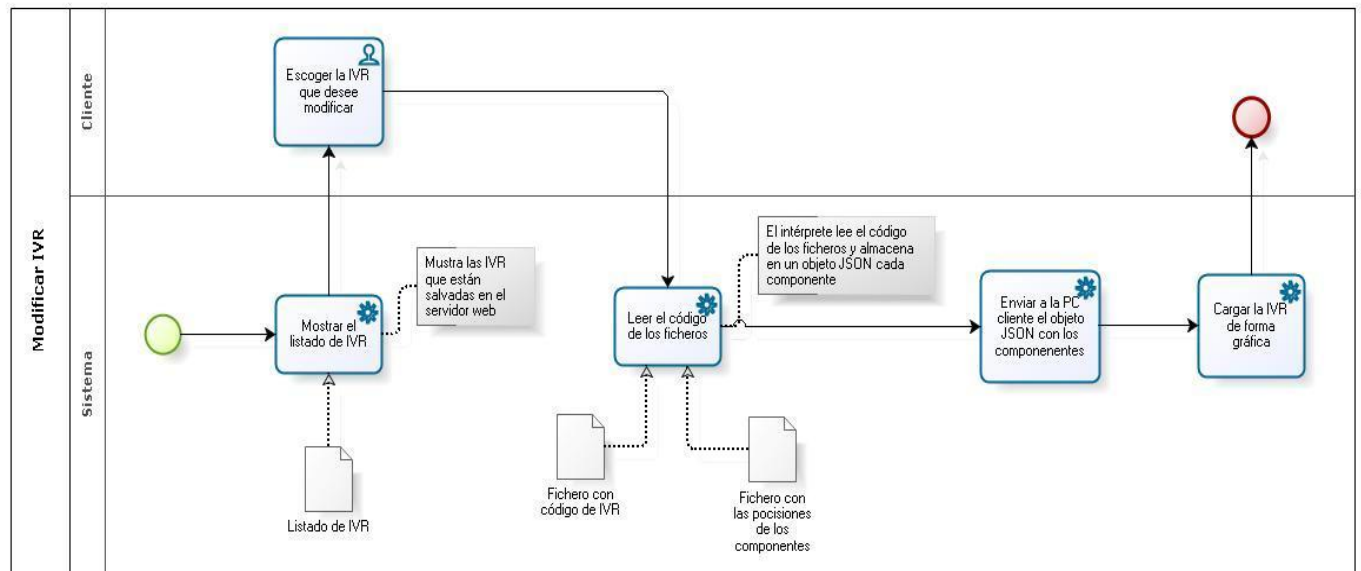
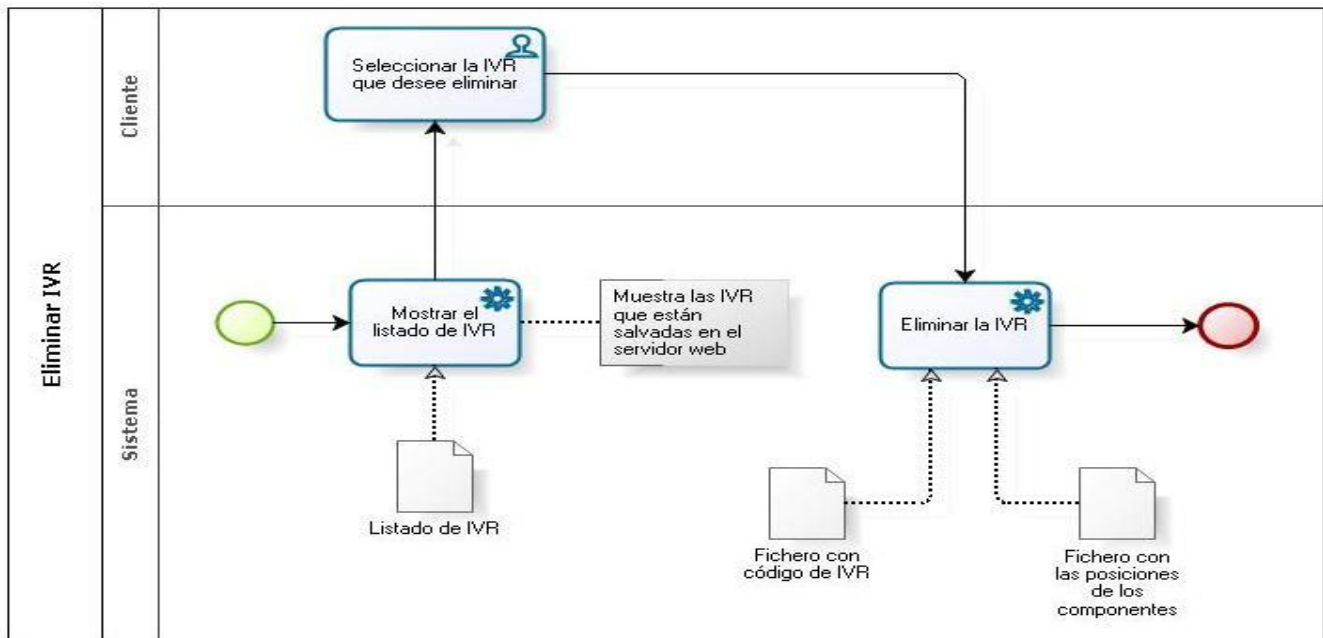


Figura 5 Proceso del sistema Diseñar IVR.



powered by BizAgi Process Modeler

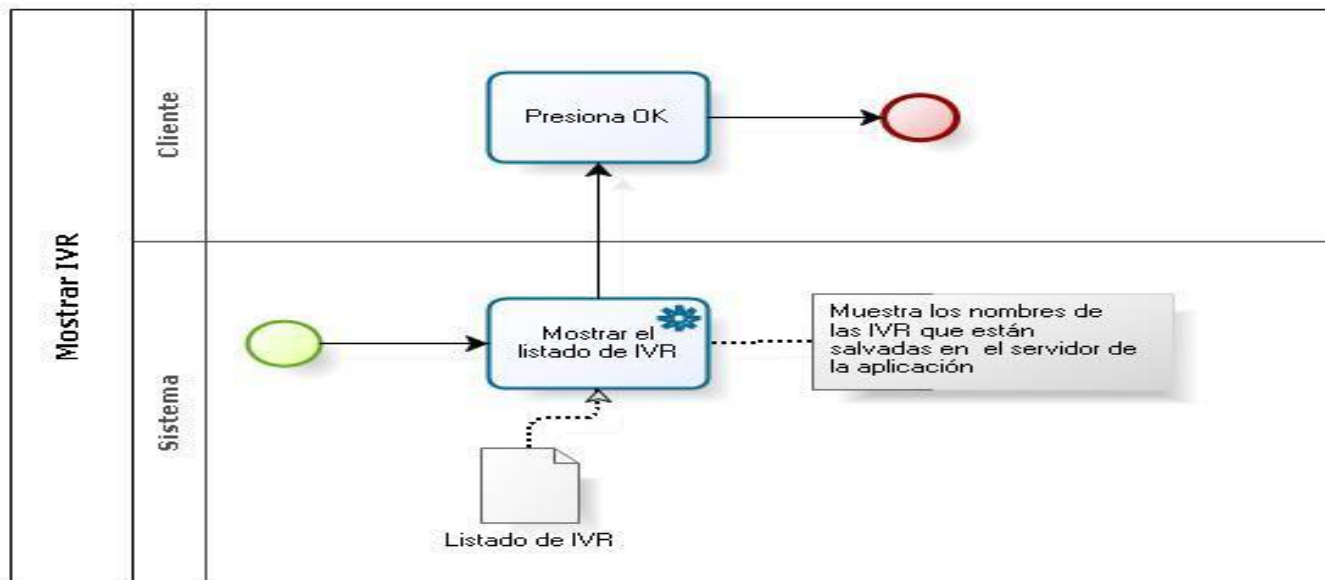
Figura 6 Proceso del sistema Modificar IVR.



powered by BizAgi Process Modeler

Figura 7 Proceso del sistema Eliminar IVR.





powered by BizAgi Process Modeler

Figura 8 Proceso del sistema Mostrar IVR.

### 2.6.2. Personas Relacionadas con el Sistema.

Se define como persona relacionada con el sistema a toda aquella que obtiene un resultado o interactúa con él de una forma u otra.

Tabla 1 Personas Relacionadas con el Sistema.

Personas Relacionadas con el Sistema.	Justificación.
Cliente.	Es la persona que se encarga de diseñar, modificar y eliminar las IVR.

### 2.6.3. Historias de Usuario.

Las historias de usuario son utilizadas en las metodologías de desarrollo ágiles como XP para especificar los requerimientos funcionales del sistema, las mismas son escritas por el cliente y especifican qué debe hacer el sistema, por lo que son cortas y escritas en el lenguaje del usuario.

**2.6.3.1. La Prioridad en el negocio.**

**Alta:** Se le otorga a las historias de usuario que resultan funcionalidades fundamentales en el desarrollo del sistema, a las que el cliente define como principales para el control integral del sistema.

**Media:** Se le otorga a las historias de usuario que resultan para el cliente como funcionalidades a tener en cuenta, sin que estas tengan una afectación sobre el sistema que se esté desarrollando.

**Baja:** Se le otorga a las historias de usuario que constituyen funcionalidades que sirven de ayuda al control de elementos asociados al equipo de desarrollo, a la estructura y no tienen nada que ver con el sistema en desarrollo.

**2.6.3.2. El Riesgo en su desarrollo.**

**Alta:** Cuando en la implementación de las historias de usuario se consideran la posible existencia de errores que lleven a la inoperatividad del código.

**Media:** Cuando pueden aparecer errores en la implementación de la historia de usuario que puedan retrasar la entrega de la versión.

**Baja:** Cuando pueden aparecer errores que serán tratados con relativa facilidad sin que traigan perjuicios para el desarrollo del proyecto. El cliente y el equipo de desarrollo trabajan en conjunto para definir como agrupar las historias de usuario para su lanzamiento.

**Tabla 2 Historia de Usuario: Generar Componentes.**

<b>Historia de Usuario.</b>	
<b>Número:</b> 1.	<b>Usuario:</b> Programador.
<b>Nombre historia:</b> Generar Componentes.	
<b>Prioridad en negocio:</b> Alta.	<b>Riesgo en desarrollo:</b> Alto.
<b>Puntos estimados:</b> 3.	<b>Iteración asignada:</b> 1.
<b>Programador responsable:</b> Programador.	
<b>Descripción:</b> Permite definir los distintos componentes que pueden conformar una IVR. Como son: ➤ Comenzar.	

- Colgar.
- Escribir mensaje.
- Subir archivo de voz.
- Grabar mensaje.
- Reproducir mensaje.
- Conector.
- Condicional múltiple.
- Condicional (Si o No).
- Recibir dígitos.
- Comparar dígitos.
- Transferir Llamada.
- Tiempo de espera.

**Tabla 3 Historia de Usuario: Diseñar IVR.**

<b>Historia de Usuario.</b>	
<b>Número:</b> 2.	<b>Usuario:</b> Usuario.
<b>Nombre historia:</b> Diseñar IVR.	
<b>Prioridad en negocio:</b> Alta.	<b>Riesgo en desarrollo:</b> Alto.
<b>Puntos estimados:</b> 3.	<b>Iteración asignada:</b> 1.
<b>Programador responsable:</b> Programador.	
<b>Descripción:</b> Permite agregar, eliminar o editar componentes para diseñar la IVR.	
<b>Prototipo de Interfaz:</b> Ver prototipo de interfaz en <a href="#">Anexo 1.</a>	

**Tabla 4 Historia de Usuario: Generar IVR.**

<b>Historia de Usuario.</b>
-----------------------------

<b>Número:</b> 3.	<b>Usuario:</b> Sistema.		
<b>Nombre historia:</b> Generar IVR.			
<b>Prioridad en negocio:</b> Alta.		<b>Riesgo en desarrollo:</b> Alto.	
<b>Puntos estimados:</b> 2.		<b>Iteración asignada:</b> 2.	
<b>Programador responsable:</b> Programador.			
<b>Descripción:</b> Permite enviar un objeto al servidor, donde se va a generar el código correspondiente a cada propiedad del objeto.			

**Tabla 5 Historia de Usuario: Salvar IVR.**

<b>Historia de Usuario.</b>			
<b>Número:</b> 4.	<b>Usuario:</b> Usuario.		
<b>Nombre historia:</b> Salvar IVR.			
<b>Prioridad en negocio:</b> Alta.		<b>Riesgo en desarrollo:</b> Alto.	
<b>Puntos estimados:</b> 1.		<b>Iteración asignada:</b> 2.	
<b>Programador responsable:</b> Programador.			
<b>Descripción:</b> Permite salvar en un fichero, en la dirección deseada el código de la IVR que puede ser interpretado por Asterisk y salvarlo en el Servidor de la Aplicación. Permite salvar en un fichero de configuración la posición (x, y) de los componentes en el Servidor de la Aplicación.			
<b>Prototipo de Interfaz:</b> Ver prototipo de interfaz en <a href="#">Anexo 2.</a>			

**Tabla 6 Historia de Usuario: Validar IVR.**

<b>Historia de Usuario.</b>	
<b>Número:</b> 5.	<b>Usuario:</b> Sistema.
<b>Nombre historia:</b> Validar IVR.	
<b>Prioridad en negocio:</b> Media.	<b>Riesgo en desarrollo:</b> Media.
<b>Puntos estimados:</b> 2.	<b>Iteración asignada:</b> 2.
<b>Programador responsable:</b> Programador.	
<b>Descripción:</b> Permite validar cada campo que exista en los componentes que se agreguen al área de trabajo así como que la sintaxis de la IVR esté correctamente.	
<b>Prototipo de Interfaz:</b> Ver prototipo de interfaz en <a href="#">Anexo 3.</a>	

**Tabla 7 Historia de Usuario: Listar IVR.**

<b>Historia de Usuario.</b>	
<b>Número:</b> 6.	<b>Usuario:</b> Sistema.
<b>Nombre historia:</b> Listar IVR.	
<b>Prioridad en negocio:</b> Baja.	<b>Riesgo en desarrollo:</b> Bajo.
<b>Puntos estimados:</b> 1.	<b>Iteración asignada:</b> 3.
<b>Programador responsable:</b> Programador.	
<b>Descripción:</b> Permite mostrar en un listado todas las IVR creadas por la aplicación.	
<b>Prototipo de Interfaz:</b> Ver prototipo de interfaz en <a href="#">Anexo 4.</a>	

**Tabla 8 Historia de Usuario: Eliminar IVR.**

<b>Historia de Usuario.</b>	
<b>Número:</b> 7.	<b>Usuario:</b> Usuario.
<b>Nombre historia:</b> Eliminar IVR.	
<b>Prioridad en negocio:</b> Baja.	<b>Riesgo en desarrollo:</b> Medio.
<b>Puntos estimados:</b> 1.	<b>Iteración asignada:</b> 3.
<b>Programador responsable:</b> Programador.	
<b>Descripción:</b> Permite al usuario eliminar la IVR creada que desee.	
<b>Prototipo de Interfaz:</b> Ver prototipo de interfaz en <a href="#">Anexo 5.</a>	

Tabla 9 Historia de Usuario: Interpretar código de IVR.

<b>Historia de Usuario.</b>	
<b>Número:</b> 8.	<b>Usuario:</b> Sistema.
<b>Nombre historia:</b> Interpretar código de IVR.	
<b>Prioridad en negocio:</b> Media.	<b>Riesgo en desarrollo:</b> Medio.
<b>Puntos estimados:</b> 3.	<b>Iteración asignada:</b> 4.
<b>Programador responsable:</b> Programador.	
<b>Descripción:</b> Encargado de leer el código fuente y buscar estructuras sintácticas que correspondan con los componentes que conforman la IVR almacenándolos en un objeto JSON.	
<b>Observaciones:</b>	

Tabla 10 Historia de Usuario: Mostrar IVR gráficamente.

<b>Historia de Usuario.</b>	
<b>Número:</b> 9.	<b>Usuario:</b> Sistema.
<b>Nombre historia:</b> Mostrar IVR gráficamente.	
<b>Prioridad en negocio:</b> Media.	<b>Riesgo en desarrollo:</b> Medio.
<b>Puntos estimados:</b> 1.	<b>Iteración asignada:</b> 4.
<b>Programador responsable:</b> Programador.	
<b>Descripción:</b> Permite a través de un objeto JSON mostrar desde el lado del cliente cada componente de forma gráfica.	
<b>Prototipo de Interfaz:</b> Ver prototipo de interfaz en <a href="#">Anexo 6.</a>	

**Tabla 11 Historia de Usuario: Modificar IVR.**

<b>Historia de Usuario.</b>	
<b>Número:</b> 10.	<b>Usuario:</b> Usuario.
<b>Nombre historia:</b> Modificar IVR.	
<b>Prioridad en negocio:</b> Media.	<b>Riesgo en desarrollo:</b> Medio.
<b>Puntos estimados:</b> 1.	<b>Iteración asignada:</b> 4.
<b>Programador responsable:</b> Programador.	
<b>Descripción:</b> Permite al usuario modificar la IVR creada que desee.	
<b>Prototipo de Interfaz:</b> Ver prototipo de interfaz en <a href="#">Anexo 7.</a>	

**2.7. Planificación.**

En la fase de planificación se realiza la estimación del esfuerzo que costará la implementación de cada historia de usuario, como en la metodología ágil XP las métricas son libres, puede utilizarse cualquier criterio para medir el desempeño del proyecto en cuestión. Una de las métricas más utilizadas en este tipo de metodología es la medida de puntos; un punto en esta métrica es considerado como una semana de trabajo, la cuál contará de seis días hábiles, donde los miembros de los equipos de desarrollo trabajan sin interrupciones.

**2.7.1. Estimación de esfuerzo por historias de usuarios.**

Se ha realizado una estimación del esfuerzo por cada una de las historias de usuario identificadas en esta fase, a continuación se muestran los resultados:

**Tabla 12 Estimación del esfuerzo por HU.**

<b>Historia de Usuario.</b>	<b>Puntos de estimación.</b>
Generar Componentes.	3.
Diseñar IVR.	3.
Generar IVR.	2.
Salvar IVR.	1.
Validar IVR.	2.
Listar IVR.	1.
Eliminar IVR.	1.
Interpretar código de IVR.	3.
Mostrar IVR gráficamente.	1.
Modificar IVR.	1.



### 2.7.2. Plan de Iteraciones.

Una vez identificadas las historias de usuario y estimado el esfuerzo dedicado a la realización de cada una de estas historias de usuario, se realiza la planificación de la implementación del trabajo, la cual se divide en iteraciones para organizar el trabajo y lograr un mejor desempeño y resultados más satisfactorios del equipo de desarrollo. Se han definido tres iteraciones, las cuales se detallan a continuación:

#### Iteración 1.

La iteración número 1 se encargará de la implementación de la historia de usuario número 1 y 2, estas tienen una gran prioridad dada la complejidad de las mismas. Al finalizar esta iteración se contará con la primera versión de prueba del sistema.

#### Iteración 2.

En la iteración número dos serán implementadas las historias de usuario 3, 4 y 5 que también tienen una prioridad muy importante, agregando ya la posibilidad de generar, validar y salvar el código de la IVR creada.

#### Iteración 3.

La iteración número tres se encargará de la implementación de las historias de usuario 6 y 7 permitiendo mostrar un listado con las IVR salvadas y poder eliminar algunas de ellas.

#### Iteración 4.

La iteración número cuatro se encargará de la implementación de las historias de usuario 8, 9 y 10 que permitirán modificar una IVR creada regresando el código de la misma a sus componentes gráficos, estas completarán el resultado de las iteraciones precedentes. Al terminar esta iteración se contará con la versión 1.0 del producto final y como resultado de esta el sistema se pondrá en funcionamiento para evaluar su resultado.

### 2.7.3. Plan de Duración de las Iteraciones.

El Plan de duración de las iteraciones es el encargado de relacionar las historias de usuario que van a ser implementadas con cada una de las iteraciones, además muestra la duración de cada iteración y el orden de realización de las historias de usuario.

#### Tabla 13 Plan de Duración de Iteraciones.

<b>Iteración.</b>	<b>Orden de la Historias de usuario a implementar.</b>	<b>Duración total.</b>
1.	Generar Componentes. Diseñar IVR.	6 semanas.
2.	Generar IVR. Salvar IVR. Validar IVR.	5 semanas.
3.	Listar IVR. Eliminar IVR.	2 semanas.
4.	Interpretar código de IVR. Mostrar IVR gráficamente. Modificar IVR.	5 semanas.

**2.7.4. Plan de Entregas.**

En este plan de entrega se expone la fecha estimada en que se liberarán los releases de cada una de iteraciones.

**Tabla 14 Plan de Entregas.**

<b>Módulo.</b>	<b>Final de la Iteración 1. 36 días.</b>	<b>Final de la Iteración 2. 24 días.</b>	<b>Final de la Iteración 3. 12 días.</b>	<b>Final de la Iteración 4. 36 días.</b>
Call-Center.	Web v0.1.	IVR v0.1.	IVR v0.1.	IVR v1.0.

**2.8. Conclusiones Parciales.**

En el desarrollo de este capítulo se abordaron las características fundamentales del sistema que se propone y se hizo una descripción de cada uno de los artefactos que se generan durante el transcurso de la fase de Exploración y Planificación.

## Capítulo III: Diseño del Sistema.

### 3.1. Introducción.

En el presente capítulo se detalla la construcción del sistema, las tareas generadas por cada historia de usuario, la arquitectura y los patrones usados en la aplicación.

### 3.2. Patrones de Diseño

Los patrones de diseño son soluciones simples y elegantes a problemas específicos y comunes del diseño orientado a objetos, basadas en la experiencia y se ha demostrado que funciona. (19) Son la base para la búsqueda de soluciones a problemas comunes en el desarrollo de software y otros ámbitos referentes al diseño de interacción o interfaces. Un patrón de diseño es una solución a un problema de diseño, una descripción de clases y objetos comunicándose entre sí adaptada para resolver un problema de diseño general en un contexto particular, el cual es el encargado de identificar Clases, Instancias, Roles, Colaboraciones y la distribución de responsabilidades. Los patrones de diseño son el esqueleto de las soluciones a problemas comunes en el desarrollo de software. (20)

#### 3.2.1. Patrones GOF (Gang of Four, Banda de Cuatro).

Los patrones de diseño del grupo de GOF se clasifican en 3 grandes categorías basadas en su propósito: creacionales, estructurales y de comportamiento. (21)

- Abstract Factory: Proporciona una interfaz para la creación de objetos interdependientes o interrelacionados, sin especificar sus clases concretas. Consiste en utilizar una clase constructora abstracta con unos cuantos métodos definidos y otros abstractos. Este patrón se ve evidenciado en la clase Componente del sistema propuesto.
- Singleton: Está diseñado para restringir la creación de objetos pertenecientes a una clase o el valor de un tipo a un único objeto. Su intención consiste en garantizar que una clase sólo tenga una instancia y proporcionar un punto de acceso global a ella. Este patrón se ve evidenciado en la clase Controladora del sistema propuesto.

### 3.2.2. Patrones generales de software para asignación de responsabilidades (GRASP).

Los patrones GRASP describen los principios fundamentales de diseño de objetos para la asignación de responsabilidades. Constituyen un apoyo para la enseñanza que ayuda a entender el diseño de objeto esencial y aplica el razonamiento para el diseño de una forma sistemática, racional y explicable. (22)

- Creador: Se aplica para la asignación de responsabilidades a las clases relacionadas con la creación de objetos, de forma tal que una instancia de un objeto sólo pueda ser creada por el objeto que contiene la información necesaria para ello. El uso de este patrón permite crear las dependencias mínimas necesarias entre las clases, lo que favorece al mantenimiento del sistema y ofrece mejores oportunidades de reutilización.
- Experto: Se aplica para la asignación de responsabilidades a las clases de forma tal que las mismas contengan la información necesaria para poder ejecutar una acción específica. El uso de este patrón permitirá a los objetos valerse de su propia información para hacer lo que se les pide, favorece la existencia de mínimas relaciones entre las clases, lo que permite contar con un sistema robusto y fácil de mantener.
- Bajo Acoplamiento: El acoplamiento mide la fuerza con que una clase está conectada a otra, de esta forma una clase con bajo acoplamiento debe tener un número mínimo de dependencia con otras clases. Las diferentes clases controladoras sólo dependen de un único controlador frontal para realizar sus funcionalidades. Este patrón se tuvo presente debido a la importancia que se le atribuye a realizar un diseño de clases independientes que puedan soportar los cambios de una manera fácil y que a su vez permitan la reutilización
- Alta cohesión: Se aplica para realizar un diseño que evite contener clases con un alto grado de abstracción, que asuman responsabilidades que podían haber delegado a otros objetos o que tengan responsabilidades muy complejas. Se tienen las clases controladoras que se encargan de ejecutar acciones de acuerdo a las peticiones que le llegan y las clases de acceso a datos que interactúan con el modelo, de forma tal que se elimina la sobrecarga de funcionalidades en las clases controladoras.
- Controlador: Se aplica para realizar las asignaciones en cuanto al manejo de los eventos del sistema y definir sus operaciones. (23)

**3.3. Tarjetas de Cargo o Clase, Responsabilidad y Colaboración (CRC).**

Para el desarrollo de sistemas que utilizan como metodología XP no se requiere de la realización del diagrama de clases, debido a que la metodología contiene una variante denominada tarjetas de Cargo o Clase, Responsabilidad y Colaboración (CRC).

Las tarjetas CRC son una técnica simple e informal pero efectiva que ha sido propuesta tanto para el modelado conceptual como para el diseño detallado de sistemas Orientado a Objetos. Se analizan basándose en sus responsabilidades con respecto al sistema. Una tarjeta CRC establece 3 dimensiones las cuales identifican el rol de un objeto en análisis y/o diseño: nombre de la clase, responsabilidades y colaboraciones. Las mismas determinan el comportamiento de cada actividad.

**Tabla 15 Tarjeta CRC Clase Controladora.**

<b>Clase: Controladora.</b>	
<b>Responsabilidad.</b>	<b>Colaboración.</b>
Recibir las peticiones del cliente y mandar a ejecutar las acciones a la clase Tejedora.  Mostrar IVR.	Tejedora.

**Tabla 16 Tarjeta CRC Clase Tejedora.**

<b>Clase: Tejedora.</b>	
<b>Responsabilidad.</b>	<b>Colaboración.</b>
Generar IVR. Salvar IVR. Modificar IVR. Validar IVR. Diseñar IVR.	Controladora.  IVR.

Tabla 17 Tarjeta CRC Clase IVR

Clase: IVR.	
Responsabilidad.	Colaboración.
Estructurar y organizar los componentes.	Tejedora. Componentes.

Tabla 18 Tarjeta CRC Clase Componente.

Clase: Componente.	
Responsabilidad.	Colaboración.
Generar Código.	IVR.

Tabla 19 Tarjeta CRC Clase Descolgar.

Clase: Comenzar.	
Responsabilidad.	Colaboración.
Generar Código.	Componente.

Tabla 20 Tarjeta CRC Clase Colgar.

Clase: Descolgar	
Responsabilidad.	Colaboración.
Generar Código.	Componente.

Tabla 21 Tarjeta CRC Clase Escribir mensaje.

Clase: Escribir mensaje.	
Responsabilidad.	Colaboración.
Generar Código.	Componente.

Tabla 22 Tarjeta CRC Clase Subir archivo de Voz.

Clase: Subir archivo de Voz.	
<b>Responsabilidad.</b>	<b>Colaboración.</b>
Generar Código.	Componente.

Tabla 23 Tarjeta CRC Clase Grabar mensaje.

Clase: Grabar mensaje.	
<b>Responsabilidad.</b>	<b>Colaboración.</b>
Generar Código.	Componente.

Tabla 24 Tarjeta CRC Clase Reproducir mensaje.

Clase: Reproducir mensaje.	
<b>Responsabilidad.</b>	<b>Colaboración.</b>
Generar Código.	Componente.

Tabla 25 Tarjeta CRC Clase Conector.

Clase: Conector.	
<b>Responsabilidad.</b>	<b>Colaboración.</b>
Generar Código.	Componente.

Tabla 26 Tarjeta CRC Clase Condicional múltiple.

Clase: Condicional múltiple.	
<b>Responsabilidad.</b>	<b>Colaboración.</b>
Generar Código.	Componente.

Tabla 27 Tarjeta CRC Clase Condicional.

Clase: Condicional	
<b>Responsabilidad.</b>	<b>Colaboración.</b>
Generar Código.	Componente.

Tabla 28 Tarjeta CRC Clase Recibir dígitos.

Clase: Recibir dígitos.	
<b>Responsabilidad.</b>	<b>Colaboración.</b>
Generar Código.	Componente.

Tabla 29 Tarjeta CRC Clase Comparar dígitos.

Clase: Comparar dígitos.	
<b>Responsabilidad.</b>	<b>Colaboración.</b>
Generar Código.	Componente.

Tabla 30 Tarjeta CRC Clase Transferir llamada.

Clase: Transferir llamada.	
<b>Responsabilidad.</b>	<b>Colaboración.</b>
Generar Código.	Componente.

Tabla 31 Tarjeta CRC Clase Tiempo de espera.

Clase: Tiempo de espera.	
<b>Responsabilidad.</b>	<b>Colaboración.</b>
Generar Código.	Componente.



A continuación se muestra una imagen con un diseño de las clases del sistema para un mejor entendimiento de lo expresado en las tarjetas CRC.

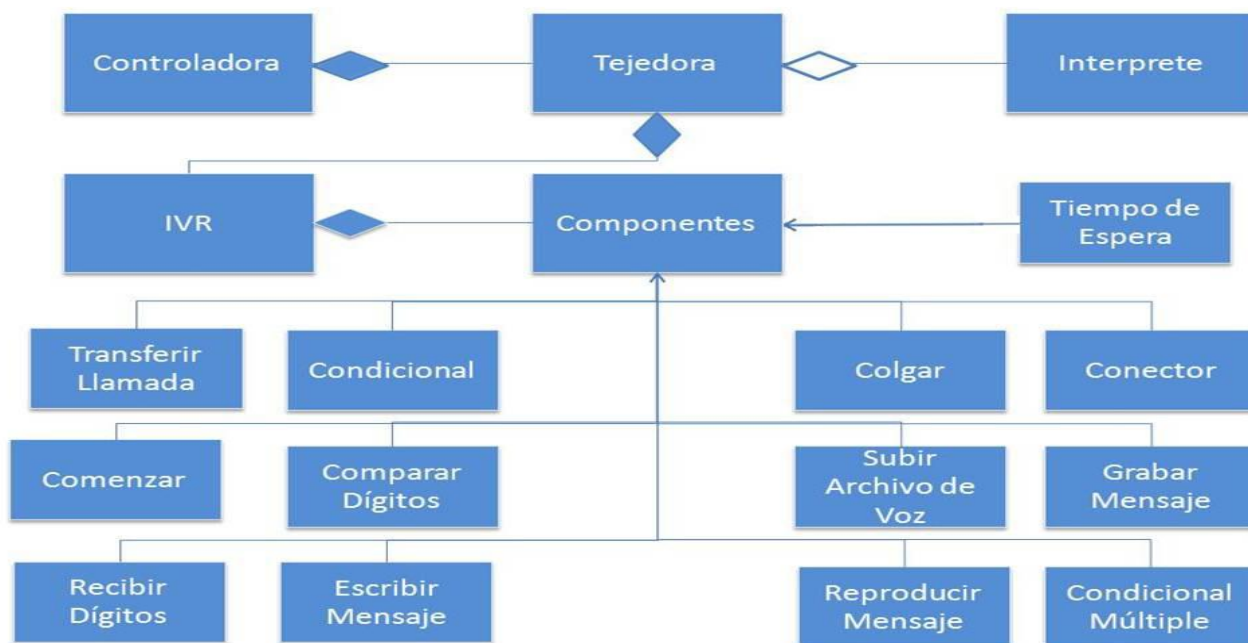


Figura 9: Diseño de Clases.

### 3.4. Tareas de Ingeniería.

Las tareas de ingeniería son escritas por el equipo de desarrollo a partir de las historias de usuario elaboradas por el cliente, las cuales describen las tareas correspondientes a cada Historia de Usuario, dando un detalle más profundo para realizar una implementación de las mismas y estimando un tiempo más cercano a la realidad para cada una de ellas.

#### 3.4.1. Tareas de Ingeniería correspondiente a la Historia de Usuario # 1.

Tabla 32 Tarea de Ingeniería # 1: Componente Comenzar.

Tarea de Ingeniería.	
Número Tarea: 1.	Número Historia de Usuario: 1.

<b>Nombre Tarea:</b> Componente Comenzar.	
<b>Tipo de Tarea:</b> Desarrollo.	<b>Puntos Estimados:</b> 0.23/3 semanas.
<b>Fecha Inicio:</b> Definir por el programador.	<b>Fecha Fin:</b> Definir por el programador.
<b>Programador Responsable:</b> Programador.	
<b>Descripción:</b> Implementar el componente comenzar, el cuál le dará inicio a la IVR, siempre saldrá por defecto en el área de trabajo.	

Para ver todas las Tareas de Ingeniería, ver [Anexos # 8, 9, ..., 36](#).

### 3.5. Conclusiones Parciales.

En este capítulo se describieron las fases de diseño, realizando una detallada descripción del diseño del sistema. Se especificaron cada una de las tareas a realizar en las cuatro iteraciones del sistema propuestas.

## Capítulo IV: Evaluación de la Propuesta.

### 4.1 Introducción.

En este capítulo se hace una evaluación mediante el uso de los criterios de un panel de expertos y el empleo de técnicas propuestas por el método Delphi con el objetivo de obtener una garantía sobre la calidad de la propuesta realizada. Se hace necesario evaluar el trabajo para contar con opiniones especializadas en el tema antes de continuar con la implementación del mismo.

Además se hace el diseño de casos de prueba planteado por la metodología XP, así como las pruebas para validar la correcta selección de los requisitos funcionales.

### 4.2 Método Delphi.

El método Delphi fue desarrollado por la Corporación Rand, es una metodología de investigación multidisciplinaria para la realización de pronósticos y predicciones a corto o largo plazo en varias ramas de las ciencias y la técnica, es considerada una de las técnicas subjetivas de pronosticación más confiables, para el desarrollo de este método se necesita la opinión de un grupo de expertos los cuales son consultados individualmente mediante cuestionarios para que sus planteamientos no influyan en los ajenos.

Consiste en la selección de un grupo de expertos a los que se les pregunta su opinión sobre cuestiones referidas a acontecimientos del futuro. Las estimaciones de los expertos se realizan en sucesivas rondas, anónimas, al objeto de tratar de conseguir consenso, pero con la máxima autonomía por parte de los participantes. (24)

#### 4.2.1 Características del Método Delphi.

Para validar la presente propuesta de forma eficiente se escogió el Método Delphi, teniéndose en cuenta que posee las siguientes características:

➤ Anonimato: Se garantizó que no existiera contacto entre los participantes, los gestores de este trabajo identificaron a cada participante y recogieron sus respuestas. Garantizando que no existiera la posibilidad de que un miembro del grupo fuera influenciado por la reputación de otro de los miembros.

- Iteración: A pesar de que el método permite realizar tantas rondas como sean necesarias para obtener mejores resultados, no fue necesario hacer más de una iteración por la concordancia obtenida en el criterio de los expertos en la primera ronda.
- Retroalimentación controlada: Los resultados totales de las rondas previas no son entregados a los participantes solo circula una parte de la información seleccionada. Lo cual no fue necesario, por lo anteriormente expuesto.
- Resultados estadísticos: La información que se presentó a los expertos no fue sólo desde el punto de vista de la mayoría, sino que se presentaron todas las opiniones indicando el grado de acuerdo que se obtuvo.

#### 4.2.2 Proceso de Selección de expertos.

El grupo de expertos se conformó con especialistas que poseen un amplio conocimiento y experiencia en el desarrollo de IVR, medimos algunos parámetros obligatorios a cumplir por los mismos, estos son:

- Graduado de nivel superior.
- Mantenga una vida activa en el desarrollo de productos informáticos.
- Conocimientos sobre el manejo y creación de IVR.
- Ocho meses de experiencia laboral como mínimo.

Escoger correctamente los expertos que conformarán el equipo propicia obtener resultados con calidad y una opinión grupal con alto nivel de consenso. Un experto debe ser una persona experimentada en el tema que se trata, capaz de ofrecer valoraciones objetivas y hacer recomendaciones al respecto.

Por tal razón esta selección se realizó atendiendo fundamentalmente a la experiencia y los resultados obtenidos por cada encuestado, entre ellos encontramos líderes, analistas y desarrolladores de proyectos, garantizando de esta forma que las opiniones brindadas sean confiables y válidas para el objetivo propuesto.

Para ver la encuesta realizada para la selección de los expertos: [Anexo 37](#).

La selección de los expertos se realizó de acuerdo a la valoración de sus competencias, para ello se hizo necesario calcular el coeficiente de competencia (k), basado en los resultados obtenidos de la encuesta de autovaloración, atendiendo a los resultados de estas se procede al cálculo mediante la fórmula:

$$K = \frac{1}{2} (k_c + k_a).$$

Donde

K: es el coeficiente de competencia.

Kc: es el coeficiente de conocimiento.

Ka: es el coeficiente de argumentación.

El coeficiente de conocimientos se obtuvo de la primera pregunta del cuestionario que es la pregunta que recoge una autoevaluación del posible experto.

**Tabla 33 Cálculo del coeficiente de conocimiento.**

0	1	2	3	4	5	6	7	8	9	10
					X					

En esta tabla las casillas están enumeradas del 0 al 10 y representan el conocimiento que se tiene de un tema determinado en esa escala. El posible experto deberá marcar con una X la casilla que refleje el conocimiento que posee respecto al tema investigado. De esta forma, si marca la casilla 0 es que no tiene conocimiento alguno del tema y viceversa si marca la casilla 10 es que tiene un gran conocimiento. Una vez realizada la selección, el número que representa la casilla se multiplicará por 0.1 para poder ajustarla a la teoría de las probabilidades. El coeficiente de argumentación se obtuvo de la segunda pregunta del cuestionario que es la pregunta que recoge las fuentes que le han servido para argumentar el conocimiento que el experto tiene de la temática presente.

**Tabla 34 Cálculo del coeficiente de argumentación.**

No.	Fuentes de argumentación.	Grado.		
		Alto.	Medio.	Bajo.
1.	Experiencia.			
2.	Documentación válida y elaborada por expertos nacionales.			
3.	Documentación válida y elaborada por expertos extranjeros.			
4.	Intercambio de información a través de foros o sitios no confiables			
5.	Conocimiento propio.			
6.	Intuición.			
	<b>TOTALES.</b>			

El experto debe marcar en este caso los elementos que le permitan argumentar su evaluación del nivel de conocimiento que seleccionó en la pregunta 1. La selección del experto en la pregunta se traduce a puntos, según la siguiente escala:

**Tabla 35 Escala para el cálculo del coeficiente de argumentación.**

No.	Fuentes de argumentación.	Grado.		
		Alto.	Medio.	Bajo.
1.	Experiencia.	0.3	0.25	0.15
2.	Documentación válida y elaborada por expertos nacionales.	0.25	0.2	0.1
3.	Documentación válida y elaborada por expertos extranjeros.	0.3	0.2	0.1
4.	Intercambio de información a través de foros o sitios no confiables.	0.05	0.05	0.05
5.	Conocimiento propio.	0.05	0.05	0.05
6.	Intuición.	0.05	0.05	0.05
	<b>TOTALES.</b>	1	0.8	0.05

Una vez calculados los coeficientes de conocimiento  $K_c$  y de argumentación  $K_a$  se calculó el coeficiente de competencias. Para interpretar el coeficiente de competencia se utilizó la siguiente escala:

Si  $0.8 < K < 1.0$ , el coeficiente de competencia es alto.

Si  $0.5 < K < 0.8$ , el coeficiente de competencia es medio.

Si  $K < 0.5$ , el coeficiente de competencia es bajo

Los resultados obtenidos del coeficiente de competencia de los cuestionarios de autoevaluación aplicado a los expertos seleccionados pueden verse en el [Anexo 38](#).

#### **4.2.2.1 Cantidad de Expertos seleccionados.**

Aunque no hay forma de determinar el número óptimo de expertos para participar en una encuesta Delphi, estudios realizados por investigadores de la Corporación Rand señalan que si bien parece necesario un mínimo de siete expertos habida cuenta que el error disminuye notablemente por cada experto añadido hasta llegar a los siete expertos, no es aconsejable recurrir a más de 30 expertos, pues la mejora en la previsión es muy pequeña y normalmente el incremento en coste y trabajo de investigación no compensa la mejora. (24)

Los expertos seleccionados para proceder a la validación fueron aquellos cuyos resultados mostraron un coeficiente de competencia alto y medio. De los siete expertos a los que se les realizó las encuestas de autovaloración, todos fueron seleccionados para continuar con la ejecución del método.

Para consultar los datos de los expertos seleccionados ver [Anexo 39](#).

#### 4.2.3 Elaboración y realización de las encuestas.

Una vez seleccionados los expertos se prosigue con la elaboración de la encuesta de validación, se confeccionó una encuesta con ocho preguntas, que debían ser categorizadas en (Muy adecuado (C1), Bastante adecuado (C2), Adecuado (C3) Poco adecuado (C4), No adecuado (C5)).

La encuesta se muestra en el [Anexo 40](#), para la confección de la misma se tuvieron en cuenta los objetivos fundamentales de la presente propuesta de aplicación.

##### 4.2.3.1 Determinación de la concordancia entre los expertos.

Un alto nivel de acuerdo entre los expertos, le dará mayor validez a la propuesta, por tanto, se hizo necesario calcular el Coeficiente de Concordancia de **Kendall** que ayudó a comprobar el grado de coincidencia en las valoraciones realizadas por los expertos.

La fórmula planteada por **Kendall** para calcular el Coeficiente de Concordancia es:

$$W = \frac{12s}{k^2(N^3 - N)}$$

Donde S: Suma de los cuadrados de las desviaciones observadas de la media de S<sub>j</sub> (rangos), esto es:

$$S = \sum_{j=1}^n (S_j - \bar{S})^2 \text{ donde } \bar{S} = \frac{\sum_{j=1}^n S_j}{N}$$

N: Número de entidades, (objetos, individuos) ordenados, es decir cantidad de preguntas realizadas a los expertos.

$\bar{S}$ : Suma de los rangos dividido entre la cantidad de preguntas realizadas.

K: Cantidad de expertos seleccionados para la validación.

W: Concordancia entre los expertos.

Cuando se tienen más de dos expertos, los rangos (S<sub>j</sub>) se calculan de la siguiente manera: Se le asignan valores a las categorías (Muy adecuado (5), Bastante adecuado (4), Adecuado (3), Poco adecuado (2), No adecuado (1)), esto se muestra en la siguiente tabla.

Tabla 36 Cálculo del Coeficiente de Kendall.

Preguntas.	E1.	E2.	E3.	E4.	E5.	E6.	E7.	Sj.	PA.	PP.
P1	5	4	5	3	4	5	3	29	82,8%	5,8
P2	5	4	5	5	5	5	5	34	97,1%	6,8
P3	5	5	5	5	5	5	5	35	100%	7
P4	4	4	5	2	3	4	3	25	71,4%	5
P5	4	4	5	4	4	4	5	30	85,7%	6
P6	5	4	4	5	5	5	5	33	94,2%	6,6
P7	3	3	5	3	4	3	4	25	71,4%	5
P8	5	4	5	4	5	5	4	32	91,4%	6,4
Total	36	32	39	31	35	36	34	243	86,7%	48,6

PA: por ciento de aceptación.

PP: promedio de puntos

En dependencia de la evaluación que el experto dé a cada pregunta, será el valor asociado que se pondrá en la tabla anterior.

Cálculo de W:

$$\bar{S} = \frac{\sum_{j=1}^n S_j}{N}; \text{ donde } N = 8$$

Por lo tanto quedaría:

$$\bar{S} = 243/8 = 30,3.$$

$$S = \sum_{j=1}^n (S_j - \bar{S})^2 = (29 - 30,3)^2 + (34 - 30,3)^2 + (35 - 30,3)^2 + (25 - 30,3)^2 + (30 - 30,3)^2 + (33 - 30,3)^2 + (25 - 30,3)^2 + (32 - 30,3)^2 = 103,92$$

Luego:

K = 7 y N = 8

$$W = \frac{12s}{k^2(N^3 - N)}$$

$$W = (12 * 103,92) / (7^2 (8^3 - 8)) = 0,0504$$



W expresa el grado de concordancia entre los cinco expertos al dar un orden evaluativo a las preguntas sometidas a valoración. Este coeficiente siempre será positivo y su valor estará comprendido en el rango de 0 a 1.

Cálculo del Chi Cuadrado Real:

$$X^2 = K (N - 1) W$$

$$X^2 = 7(8 - 1) * 0,0504 = 2,46$$

Si  $X^2 \text{ real} < X^2 (\infty, N - 1)$  entonces existe concordancia entre los expertos.

El Chi cuadrado calculado se compara con los valores de tablas estadísticas dadas, con  $\infty=0.05$  para un nivel de confianza del 95%.

$$X^2_{\text{real}} < X^2(\alpha, N - 1)$$

$$2,46 < X^2(0.95, 7)$$

$$2,46 < 14.067140$$

Por lo tanto, se puede concluir que existe concordancia entre los expertos.

#### 4.2.4 Conclusiones de la validación por método Delphi.

En la siguiente gráfica se muestran los resultados de la validación, basándose en cada una de las respuestas emitidas por los expertos.

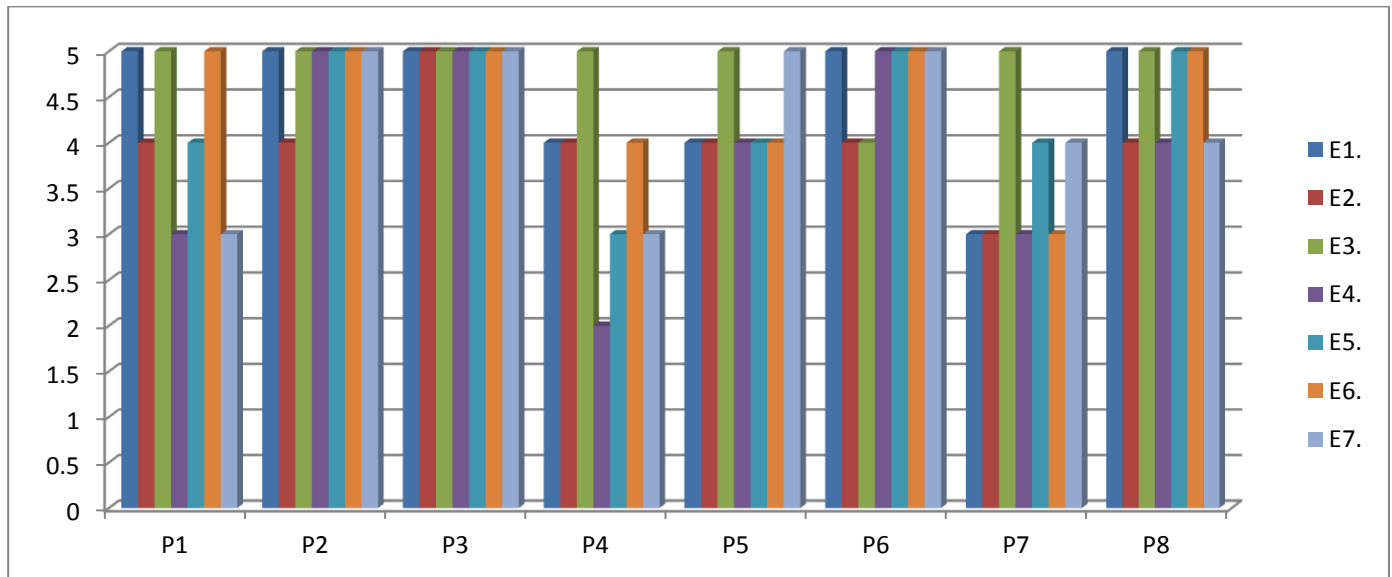


Figura 10 Resultados de la validación.

Analizando la tabla número 36 donde se muestran los resultados de la validación, detallando las preguntas con la puntuación otorgada por los expertos a cada uno de ellos, se puede concluir que la propuesta cumple con los objetivos trazados en un por ciento alto, por lo que el resultado de la validación mediante este método es positivo. El promedio de puntos está siempre por encima de los 3 puntos y el por ciento de aceptación de cada uno de los criterios es superior al 60%. Se obtiene un valor final de 86,7 % de aceptación.

### **4.3 Pruebas.**

XP anima a probar constantemente tanto como sea posible. Esto permite aumentar la calidad de los sistemas reduciendo el número de errores no detectados y disminuyendo el tiempo transcurrido entre la aparición de un error y su corrección. También permite aumentar la seguridad de evitar efectos colaterales no deseados a la hora de realizar modificaciones y refactorizaciones. XP divide las pruebas del sistema en dos grupos: pruebas unitarias y pruebas de aceptación o pruebas funcionales destinadas a evaluar si al final de una iteración se consiguió la funcionalidad requerida diseñadas por el cliente final. Este tipo de pruebas, constituyen uno de los mayores avances en la programación desde la aparición de la orientación a objetos y aunque son recomendadas, su uso o no, queda a merced de los desarrolladores.

#### **4.3.1 Pruebas en el análisis.**

Las pruebas se realizan durante todo el proceso de desarrollo del software. Para verificar la correcta definición de las funcionalidades correspondientes a la aplicación, se le realizaron pruebas a los requisitos funcionales del sistema. Estas pruebas se pueden hacer por listas de chequeo o a través de un prototipo no funcional, el cual se le muestra al cliente y éste define con su criterio si están correctamente definidas o no.

Para la realización de estas pruebas se escogió el método de un prototipo no funcional, ya que las listas de chequeo no están hasta el momento diseñadas para la metodología XP en la Universidad de las Ciencias Informáticas. Con la realización de las mismas se confirmó la correcta definición de los requisitos funcionales de aplicación.

#### **4.3.2 Pruebas Unitarias.**

Son las encargadas de que en el transcurso de la implementación de un sistema, cada uno de los desarrolladores tiene que ir probando constantemente lo que va obteniendo para garantizar que las

funcionalidades exigidas por el cliente estén siendo implementadas correctamente. Estas no generan artefactos y no son directamente palpables para el cliente.

Las pruebas unitarias proporcionan beneficios tales como:

- El programador puede realizar cambios de forma segura respaldado por efectivos casos de prueba.
- Permite saber si una determinada funcionalidad se puede agregar al sistema existente sin alterar el funcionamiento actual del mismo.
- Brindan al programador una inmediata retroalimentación de como está realizando su trabajo. (25)

#### 4.3.3 Pruebas de Aceptación.

Las pruebas funcionales son la mejor forma de probar la aplicación de extremo a extremo: desde la petición realizada por un navegador hasta la respuesta enviada por el servidor. Las pruebas funcionales prueban todas las capas de la aplicación. En realidad, son muy similares a lo que se hace manualmente cada vez que se añade o modifica una acción y se prueban dichos cambios en el navegador para comprobar que todo funciona bien al pulsar sobre los enlaces y botones y que todos los elementos se muestran correctamente en la página. En otras palabras, lo que se hace es probar un escenario correspondiente a la historia de usuario que se acaba de implementar en la aplicación.

Las pruebas de aceptación son más importantes que las pruebas unitarias dado que significan la satisfacción del cliente con el producto desarrollado y el final de una iteración y el comienzo de la siguiente. Las pruebas de aceptación se elaboran a lo largo de la iteración, en paralelo con el desarrollo del sistema, y adaptándose a los cambios que el sistema sufra. (25)

Dichas pruebas estarán divididas en las siguientes secciones:

- **Código:** Servirá como identificador de la prueba realizada, a su vez será sugerente al nombre de la prueba a la que hace referencia.
- **Número Historia de Usuario:** Tendrá el número de la HU a la que hace referencia la prueba a realizar.
- **Nombre:** Nombre que se le da a la prueba a realizar.
- **Descripción:** Se describe la funcionalidad que se desea probar.

- **Condiciones de Ejecución:** Mostrará las condiciones que deben cumplirse para poder llevar a cabo el caso de prueba.
- **Entradas / Pasos de Ejecución:** Se hará la descripción de cada uno de los pasos seguidos durante el desarrollo de la prueba, se tendrá en cuenta cada una de las entradas que hace el usuario con el objetivo de ver si se obtiene el resultado esperado.
- **Resultado esperado:** Se hará una breve descripción del resultado que se espera obtener con la prueba realizada.
- **Evaluación de la prueba:** Acorde al resultado de la prueba realizada se emitirá una evaluación sobre la misma:

Bien: Cuando el resultado de la prueba es exactamente el esperado por el usuario.

Parcialmente bien: Cuando el resultado no es completamente el esperado por el cliente o usuario de la aplicación y muestra resultados erróneos o fuera de contexto.

Mal: Cuando el resultado de la prueba realizada genera un error de codificación en la aplicación o muestra como resultado elementos no deseados o fuera de contexto, trayendo como consecuencia que la funcionalidad requerida por el cliente no tenga resultado, lo que invalida también la UH.

#### 4.4 Casos de Prueba.

##### 4.4.1 Diseño de casos de prueba de la Historia de Usuario # 1.

**Tabla 37 Prueba # 1.1: Generar componentes.**

<b>Caso de Prueba Aceptación.</b>	
<b>Código:</b> 1.1.	<b>Número Historia de Usuario:</b> 1.
<b>Nombre:</b> Generar componentes.	
<b>Descripción:</b> Prueba para verificar la funcionalidad Generar componentes.	
<b>Condiciones de Ejecución:</b> La aplicación tiene que estar abierta.	
<b>Entrada/Pasos de ejecución:</b>	
➤ Cada componente será creado con la forma y con el código correspondiente a él.	

**Resultado Esperado:** Aparecerán todos los componentes creados en la aplicación en el menú de componentes. Se cargarán con la forma programada cuando el usuario desee utilizarlos y se salvarán con el código correspondiente a él.

**Evaluación de la Prueba:** Evaluar después de la implementación.

Para ver todos los diseños de Casos de Prueba, ver [Anexos # 41, 42,...., 64.](#)

#### **4.5 Conclusiones Parciales.**

En este capítulo se seleccionaron los expertos que verdaderamente cumplían con los niveles de competencia suficiente para así llegar a comprobar que el diseño del sistema propuesto tuvo resultados satisfactorios. Se abordó acerca de la importancia de las pruebas en la metodología XP, en especial las pruebas de aceptación, ya que estas miden la satisfacción del cliente con el producto desarrollado.

**Conclusiones.**

Con el desarrollo de este trabajo de diploma se logró encontrar la respuesta a la interrogante: ¿Cómo facilitar el proceso de creación de las IVR para un sistema Asterisk?

Dicha respuesta se manifiesta con la realización de un diseño de una futura aplicación que permita la creación gráfica de IVR.

Este sistema propuesto permitirá facilitar el desarrollo de las IVR, logrando ser una aplicación clave para los proyectos productivos del centro Telemática.

Por lo que la aplicación diseñada cumple satisfactoriamente el objetivo trazado para este trabajo ya que se definieron todas las bases para guiar el proceso de desarrollo de software permitiendo la modelación del sistema.

En el presente trabajo de diploma se definieron las tecnologías, herramientas y metodología de desarrollo de software que se necesitarán para la implementación y correcto funcionamiento de la aplicación. El aprovechamiento de estas tecnologías, la correcta selección y uso de las herramientas permitirá lograr un sistema que cumpla las características definidas.

**Recomendaciones.**

- Se recomienda la implementación de la aplicación propuesta y diseñada, ya que la continuidad de este proyecto, llevando a la práctica lo analizado en el presente trabajo de diploma, es un eslabón muy importante para resolver el problema que actualmente se evidencia a la hora de crear IVR.
- Se recomienda la utilización de la aplicación propuesta después de su implementación, específicamente en los proyectos productivos del centro Telemática por el vínculo directo que tienen con las IVR.

### Trabajos Citados.

1. **buenastareas.** buenastareas. [En línea] [Citado el: 10 de Noviembre de 2010.] Disponible en:[<http://www.buenastareas.com/ensayos/Telecomunicaciones/74673.html>].
2. **softwaredecobranza.** softwaredecobranza. [En línea] [Citado el: 20 de Noviembre de 2010.] Disponible en:[<http://www.softwaredecobranza.com/interactive-voice-response.aspx>].
3. Diseño metodológico de la Investigación de la población y muestra. Métodos y diseño experimental.
4. *Diseño metodológico de la investigación de la población y muestra. Métodos y diseño experimental.*

### Martinto, MSc. Pedro Carlos Pérez.

5. **Gestiopolis.** Gestiopolis. [En línea] [Citado el: 10 de Enero de 2011.] Disponible en:[<http://www.gestiopolis.com/canales/demarketing/articulos/61/callcenter.htm>].
6. **Lcc.** Lcc. [En línea] [Citado el: 20 de Octubre de 2010.] Disponible en:[<http://www.lcc.uma.es/~eat/services/llamadas/llamadas.html>].
7. **tapachulaenlinea.** tapachulaenlinea. [En línea] [Citado el: 5 de Diciembre de 2010.] Disponible en:[<http://www.tapachulaenlinea.com/~tutoriales/?p=339>].
8. **Slideshare.** Slideshare. [En línea] [Citado el: 1 de Febrero de 2011.] Disponible en:[<http://www.slideshare.net/urumisama/programacin-extrema-4488942>].
9. **univalle.** univalle. [En línea] [Citado el: 12 de Diciembre de 2010.] Disponible en:[<http://eisc.univalle.edu.co/materias/WWW/material/lecturas/xp.pdf>].
10. **masadelante.** masadelante. [En línea] [Citado el: 16 de Diciembre de 2010.] Disponible en:[<http://www.masadelante.com/faqs/html>].
11. **blogspot.** blogspo. [En línea] [Citado el: 10 de Enero de 2011.] Disponible en:[<http://3plej-life.blogspot.com/2011/02/javascript-orientado-objetos-clases.html>].
12. **Gutierrez, Javier J.** Isi. [En línea] [Citado el: 27 de Enero de 2011.] Disponible en:[[http://www.isi.us.es/~javierj/investigacion\\_ficheros/Framework.pdf](http://www.isi.us.es/~javierj/investigacion_ficheros/Framework.pdf)].
13. **ganacontupropiaweb.** ganacontupropiaweb. [En línea] [Citado el: 15 de Enero de 2011.] Disponible en:[[http://www.ganacontupropiaweb.com/webmaster/lenguajes\\_de\\_programacion\\_de\\_paginas\\_web.htm](http://www.ganacontupropiaweb.com/webmaster/lenguajes_de_programacion_de_paginas_web.htm)].
14. **Barriento, Manuel Sánchez.** Aprendergratis. [En línea] 2 de Noviembre de 2008. [Citado el: 15 de Enero de 2011.] Disponible en:[<http://www.aprendergratis.com/introduccion-a-bpmn.html>].
15. **Hincapie, Daniel.** Danielhincapie. [En línea] [Citado el: 12 de Febrero de 2011.] Disponible en:[<http://danielhincapie.com/PRACTICAGENERAL/servidores.html>].



16. **esenciahumana.** esenciahumana. [En línea] [Citado el: 25 de Enero de 2011.] Disponible en:[<http://www.esenciahumana.com.mx/Servicios/AplicacionesWeb/VentajasBeneficiosAplicaciones.html>].
17. **Synergix.** Wordpress. [En línea] [Citado el: 15 de Mayo de 2011.] Disponible en:[<http://synergix.wordpress.com/2008/07/07/requisito-funcional-y-no-funcional/>].
18. **Erika Camacho, Fabio Cardeso y Gabril Nuñez.** *Arquitecturas de Software*. Abril, 2004.
19. **Software, Ingeniero.** Ingenierosoftware. [En línea] [Citado el: 2 de Mayo de 2011.] Disponible en:[<http://www.ingenierosoftware.com/analisisydiseno/patrones-diseno.php>].
20. **Tedeschi, Nicolás.** Microsoft. [En línea] [Citado el: 10 de Marzo de 2011.] Disponible en:[<http://msdn.microsoft.com/es-es/library/bb972240.aspx>].
21. **Fire, Angel.** Geektheplanet. [En línea] 11 de Mayo de 2011. [Citado el: 26 de Mayo de 2011.] Disponible en:[<http://geektheplanet.net/5462/patrones-gof.xhtml>].
22. **Saavedra, Jorge.** Wordpress. [En línea] [Citado el: 2 de Mayo de 2011.] Disponible en:[<http://jorgesaavedra.wordpress.com/category/patrones-grasp>].
23. **Larman, C.** "UML y patrones" Tomo I Capítulos 18. *"UML y patrones" Tomo I Capítulos 18*. 2009-2010.
24. **Astigarraga, Eneko.** EL MÉTODO DELPHI.
25. **Pérez, Isaías Carrillo.** *METODOLOGIA DE DESARROLLO DE SOFTWARE*. s.l. : 2, 2005.

**Bibliografía.**

1. **buenastareas.** buenastareas. [En línea] [Citado el: 10 de Noviembre de 2010.] Disponible en:[<http://www.buenastareas.com/ensayos/Telecomunicaciones/74673.html>].
2. **softwaredecobranza.** softwaredecobranza. [En línea] [Citado el: 20 de Noviembre de 2010.] Disponible en:[softwaredecobranza. softwaredecobranza. [En línea] [Citado el: 12 de Noviembre de 2010.] Disponible en:[<http://www.softwaredecobranza.com/interactive-voice-response.aspx>].].
3. **tapachulaenlinea.** tapachulaenlinea. [En línea] [Citado el: 5 de Diciembre de 2010.] Disponible en:[<http://www.tapachulaenlinea.com/~tutoriales/?p=339>].
4. **univalle.** univalle. [En línea] [Citado el: 12 de Diciembre de 2010.] Disponible en:[<http://eisc.univalle.edu.co/materias/WWW/material/lecturas/xp.pdf>].
5. **masadelante.** masadelante. [En línea] [Citado el: 16 de Diciembre de 2010.] Disponible en:[<http://www.masadelante.com/faqs/html>].
6. **blogspo.** blogspo. [En línea] [Citado el: 10 de Enero de 2011.] Disponible en:[<http://3plej-life.blogspot.com/2011/02/javascript-orientado-objetos-clases.html>].
7. **ganacontupropiaweb.** ganacontupropiaweb. [En línea] [Citado el: 15 de Enero de 2011.] Disponible en:[[http://www.ganacontupropiaweb.com/webmaster/lenguajes\\_de\\_programacion\\_de\\_paginas\\_web.htm](http://www.ganacontupropiaweb.com/webmaster/lenguajes_de_programacion_de_paginas_web.htm)].
8. **esenciahumana.** esenciahumana. [En línea] [Citado el: 25 de Enero de 2011.] Disponible en:[<http://www.esenciahumana.com.mx/Servicios/AplicacionesWeb/VentajasBeneficiosAplicaciones.html>].
9. **Larman, C.** “UML y patrones” Tomo I Capítulos 18. “UML y patrones” Tomo I Capítulos 18. 2009-2010.
10. **solcre.** solcre. [En línea] [Citado el: 29 de Enero de 2011.] Disponible en:[[http://www.solcre.com/files/ventajas\\_de\\_las\\_aplicaciones\\_web.pdf](http://www.solcre.com/files/ventajas_de_las_aplicaciones_web.pdf)].
11. **ERIKA CAMACHO, FABIO CARDESO, GABRIEL NUÑEZ.** prof.usb. [En línea] [Citado el: 4 de Febrero de 2011.] Disponible en:[<http://prof.usb.ve/lmendoza/Documentos/PS-6116/Guia%20Arquitectura%20v.2.pdf>].
12. **icomparable.** icomparable. [En línea] [Citado el: 10 de Febrero de 2011.] Disponible en:[<http://icomparable.blogspot.com/2008/11/arquitecturas-n-tier-ventajas-y.html>].
13. **programacionextrema.** programacionextrema. [En línea] [Citado el: 2 de Marzo de 2011.] Disponible en:[<http://www.programacionextrema.org/>].
14. **efrenteicom.** efrenteicom. [En línea] [Citado el: 10 de Noviembre de 2010.] Disponible en:[<http://www.efrenteicom.com/>].

15. **buenastareas.** buenastareas. [En línea] [Citado el: 10 de Noviembre de 2010.] Disponible en:[<http://www.buenastareas.com/ensayos/Telecomunicaciones/74673.html>].
16. **tmcnet.** tmcnet. [En línea] [Citado el: 12 de Noviembre de 2010.] Disponible en:[<http://ivr.tmcnet.com/>].
17. **articulosinformativos.** articulosinformativos. [En línea] [Citado el: 12 de Noviembre de 2010.] Disponible en:[[http://www.articulosinformativos.com.mx/Respuesta\\_de\\_Voz\\_Interactiva-a953452.html](http://www.articulosinformativos.com.mx/Respuesta_de_Voz_Interactiva-a953452.html)].
18. **cash-pot.** cash-pot. [En línea] [Citado el: 20 de Noviembre de 2010.] Disponible en:[<http://www.cash-pot.com.ar/interactive-voice-response.aspx>].
19. **asterisk-es.** asterisk-es. [En línea] [Citado el: 29 de Noviembre de 2010.] Disponible en:[<http://comunidad.asterisk-es.org/index.php?title=Introduccion>].
20. **Aguilar, Javier Vásquez.** gestiopolis. [En línea] [Citado el: 3 de Diciembre de 2010.] Disponible en:[<http://www.gestiopolis.com/canales/demarketing/articulos/61/callcenter.htm>].
21. **certificacionelastix.** certificacionelastix. [En línea] [Citado el: 7 de Diciembre de 2010.] Disponible en:[[http://www.certificacionelastix.com/index.php?option=com\\_content&view=article&id=2&Itemid=2](http://www.certificacionelastix.com/index.php?option=com_content&view=article&id=2&Itemid=2)].
22. **seetio.** seetio. [En línea] [Citado el: 10 de Diciembre de 2010.] Disponible en:[<http://www.seetio.com/blog/quickfuse-crea-proprias-aplicaciones-voz-interactivas-ivr-online/>].
23. **chuidiang.** chuidiang. [En línea] [Citado el: 15 de Diciembre de 2010.] Disponible en:[<http://www.chuidiang.com/ood/metodologia/extrema.php>].
24. **desarrolloweb.** desarrolloweb. [En línea] [Citado el: 20 de Diciembre de 2010.] Disponible en:[<http://www.desarrolloweb.com/articulos/que-es-html.html>].
25. **maestrosdelweb.** maestrosdelweb. [En línea] [Citado el: 7 de Enero de 2011.] Disponible en:[<http://www.maestrosdelweb.com/editorial/%C2%BFque-es-javascript/>].
26. **Alvarez, Miguel Angel.** dmaspv. [En línea] [Citado el: 11 de Enero de 2011.] Disponible en:[[http://dmaspv.com/files/page/07042011180222\\_manual%20de%20jquery%20en%20pdf%20desarrollo\\_webcom.pdf](http://dmaspv.com/files/page/07042011180222_manual%20de%20jquery%20en%20pdf%20desarrollo_webcom.pdf)].
27. **portalprogramas.** portalprogramas. [En línea] [Citado el: 15 de Enero de 2011.] Disponible en:[<http://gratis.portalprogramas.com/Microsoft-Project.html>].
28. **catarina.** udlap. [En línea] [Citado el: 20 de Enero de 2011.] Disponible en:[[http://catarina.udlap.mx/u\\_dl\\_a/tales/documentos/lis/rea\\_c\\_ji/capitulo2.pdf](http://catarina.udlap.mx/u_dl_a/tales/documentos/lis/rea_c_ji/capitulo2.pdf)].
29. **Dany.** Rowman. [En línea] [Citado el: 20 de Enero de 2011.] Disponible en:[<http://es.scribd.com/doc/22659190/Manual-De-BizAqi>].

30. **geneura.** geneura. [En línea] [Citado el: 25 de Enero de 2011.] Disponible en:[<http://geneura.ugr.es/~gustavo/apache/>].
31. **Apache.** Apache. [En línea] [Citado el: 13 de Marzo de 2011.] Disponible en:[<http://www.apache.org>].
32. **Presman, Roger S.** *Ingeniería de Software. Un enfoque práctico.*
34. **Gestiopolis.** Gestiopolis. [En línea] [Citado el: 10 de Enero de 2011.] Disponible en:[<http://www.gestiopolis.com/canales/demarketing/articulos/61/callcenter.htm>].
35. **Lcc.** Lcc. [En línea] [Citado el: 20 de Octubre de 2010.] Disponible en:[<http://www.lcc.uma.es/~eat/services/llamadas/llamadas.html>].
36. **Slideshare.** Slideshare. [En línea] [Citado el: 1 de Febrero de 2011.] Disponible en:[<http://www.slideshare.net/urumisama/programacin-extrema-4488942>].
37. **Gutierrez, Javier J.** Isi. [En línea] [Citado el: 27 de Enero de 2011.] Disponible en:[[http://www.isi.us.es/~javierj/investigacion\\_ficheros/Framework.pdf](http://www.isi.us.es/~javierj/investigacion_ficheros/Framework.pdf)].
38. **Hincapie, Daniel.** Danielhincapie. [En línea] [Citado el: 12 de Febrero de 2011.] Disponible en:[<http://danielhincapie.com/PRACTICAGENERAL/servidores.html>].
39. **Software, Ingeniero.** Ingenierosoftware. [En línea] [Citado el: 2 de Mayo de 2011.] Disponible en:[<http://www.ingenierosoftware.com/analisisydiseno/patrones-diseno.php>].
40. **Tedeschi, Nicolás.** Microsoft. [En línea] [Citado el: 10 de Marzo de 2011.] Disponible en:[<http://msdn.microsoft.com/es-es/library/bb972240.aspx>].
41. **Fire, Angel.** Geektheplanet. [En línea] 11 de Mayo de 2011. [Citado el: 26 de Mayo de 2011.] Disponible en:[<http://geektheplanet.net/5462/patrones-gof.xhtml>].
42. **Saavedra, Jorge.** Wordpress. [En línea] [Citado el: 2 de Mayo de 2011.] Disponible en:[<http://jorgesaavedra.wordpress.com/category/patrones-grasp>].
43. **Pérez, Isaías Carrillo.** *METODOLOGIA DE DESARROLLO DE SOFTWARE.* s.l. : 2, 2005.
44. **Saavedra, Jorge.** Wordpress. [En línea] [Citado el: 2 de Mayo de 2011.] Disponible en:[<http://jorgesaavedra.wordpress.com/category/patrones-grasp>].
45. **Larman, C.** "UML y patrones" Tomo I Capítulos 18. "UML y patrones" Tomo I Capítulos 18. 2009-2010.
46. **Astigarraga, Eneko.** EL MÉTODO DELPHI.
47. **Pérez, Isaías Carrillo.** *METODOLOGIA DE DESARROLLO DE SOFTWARE.* s.l. : 2, 2005

Anexos.

Figura 11. Anexo 1. Interfaz Diseñar IVR.

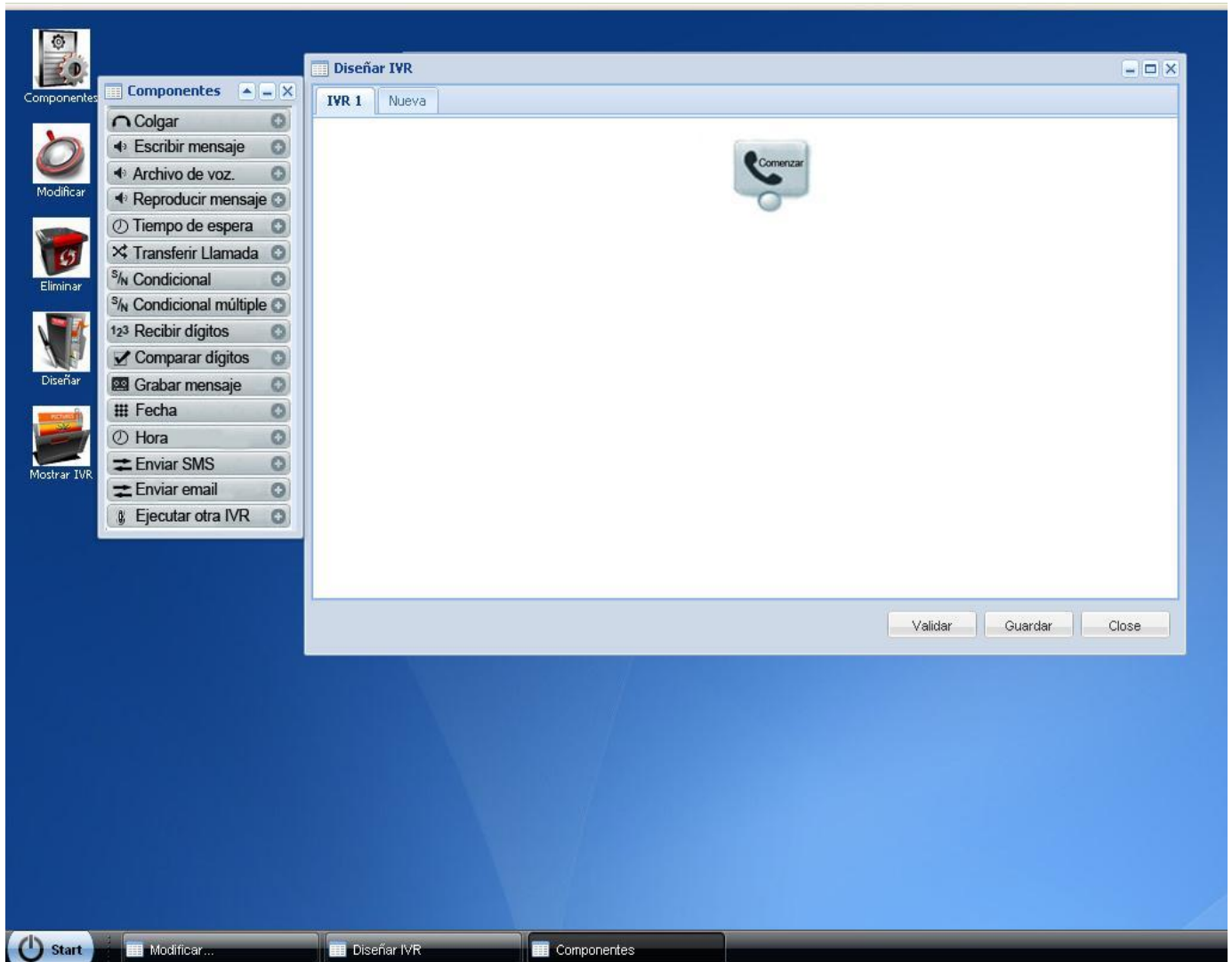


Figura 12. Anexo 2. Interfaz Salvar IVR.

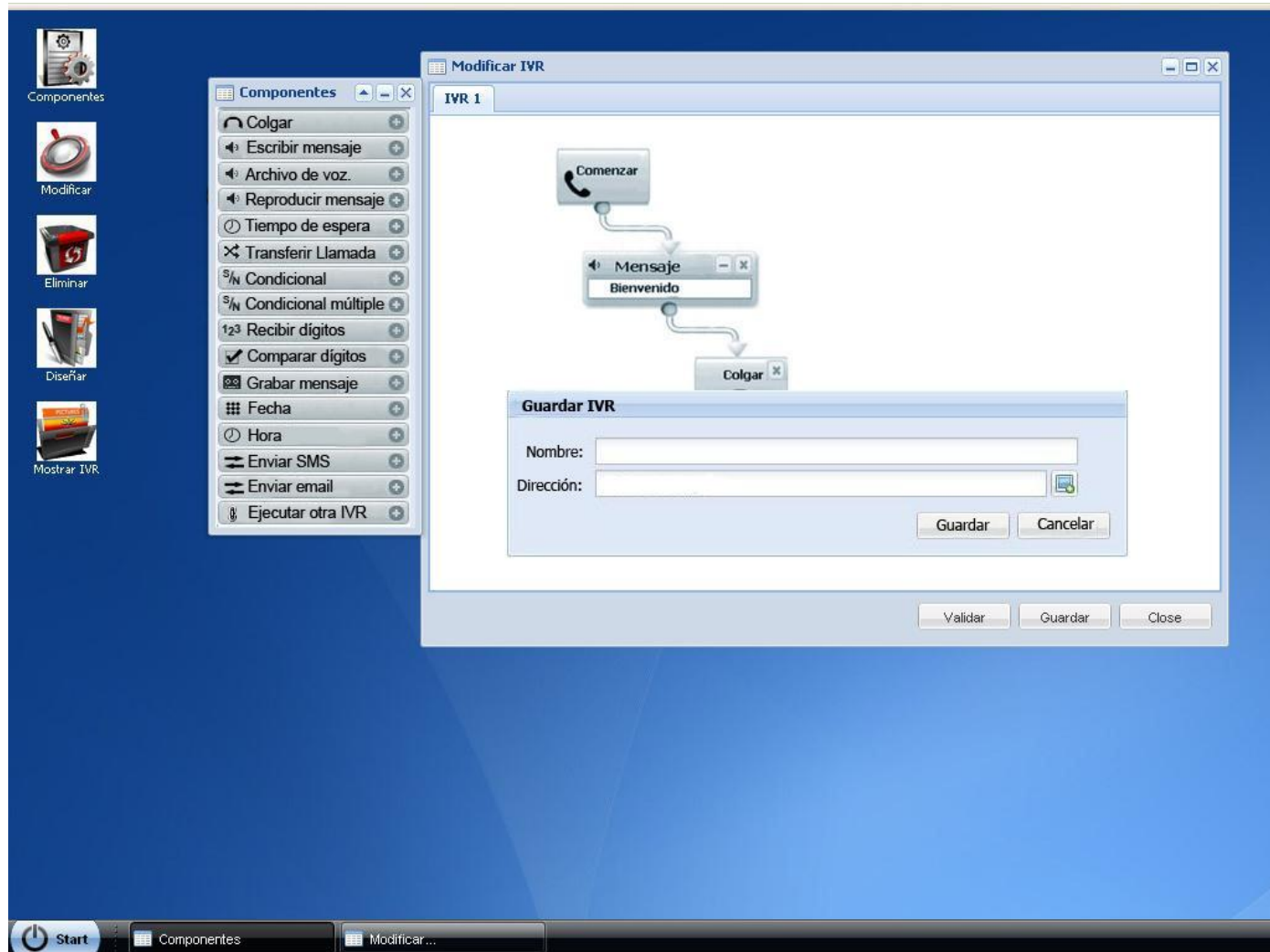


Figura 13. Anexo 3. Interfaz Validar IVR.

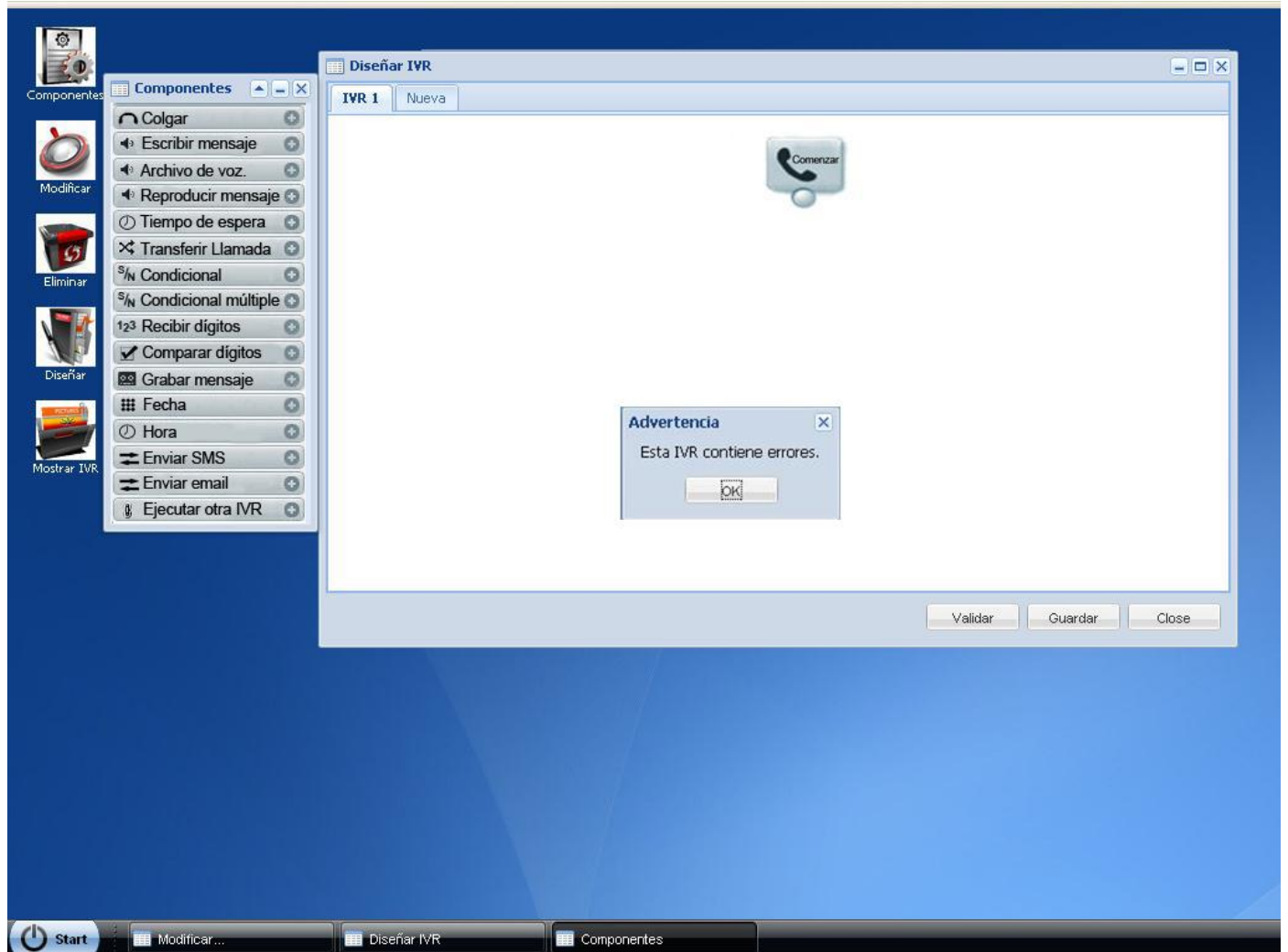


Figura 14. Anexo 4. Interfaz Listar IVR.

