

UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS.

FACULTAD 2.



Trabajo de Diploma para optar por el título de
Ingeniero en Ciencias Informáticas.

**Título: Sistema de Gestión de Encuestas Telefónicas
a través de Respuesta de Voz Interactiva.**

Autores: Yasmin Diaz Espinosa.

Ricardo Ernesto Sori Echerri.

Tutor: Ing. Alejandro Rodríguez Reyes.

Ciudad de La Habana, 2011.

Resumen.

La encuesta es un método de la investigación de mercados que sirve para obtener información específica de una muestra de la población mediante el uso de cuestionarios estructurados que se utilizan para obtener datos precisos de las personas encuestadas.

Existen varios tipos de encuestas, entre ellas las telefónicas. Estas consisten en una entrevista vía telefónica con cada encuestado. Sus principales ventajas son:

- ✓ Se puede abarcar un gran número de personas en menos tiempo que la entrevista personal.
- ✓ Sus costos suelen ser bajos.
- ✓ Son de fácil administración (hoy en día, existen softwares especializados para la gestión de encuestas telefónicas).

La Empresa de Telecomunicaciones de Cuba S.A. (ETECSA) necesita conocer cuantitativa y cualitativamente la aceptación de los servicios que ofrece, para así poder estar al tanto de que tan satisfechos se sienten sus clientes. Se desarrolló un Sistema de Gestión de Encuestas Telefónicas, donde no solo ETECSA sino diferentes empresas y entidades puedan consumir este servicio. Para la gestión de estas encuestas telefónicas se utilizó como software: Asterisk, el cual provee funcionalidades de una central telefónica, además es de código abierto, lo cual facilita su obtención sin tener que pagar licencias de forma sistemática. Provee funcionalidades como la creación de Respuesta de Voz Interactiva (en inglés Interactive Voice Response), que es utilizada por empresas y entidades que reciben o efectúan grandes cantidades de llamadas, a fin de reducir la necesidad de personal y costos.

Índice de Contenido.

Resumen.....	2
Índice de Figuras.....	8
Índice de Tablas.....	9
Introducción.....	1
CAPÍTULO I. FUNDAMENTACIÓN TEÓRICA.....	4
1.1. Introducción.....	4
1.2. ¿Qué es una IVR?.....	4
1.2.1. Tecnología IVR.....	4
1.3. Ventajas de la Respuesta de Voz Interactiva para las empresas.....	4
1.4. Sistemas de Encuestas Telefónicas mediante una IVR en el mundo.....	5
1.5. Necesidad del Sistema de Gestión de Encuestas para Cuba.....	6
1.6. Herramientas Utilizadas.....	7
1.6.1. Metodología utilizada para el desarrollo del proyecto.....	7
1.6.2. Sistemas Gestores de Bases de Datos (SGBD).....	8
1.6.3. Herramientas de Base de Datos.....	9
1.6.4. Lenguajes de programación WEB.....	10
1.6.5. Servidores para aplicaciones web.....	12
1.6.6. Frameworks.....	13
1.6.7. Control de Versiones.....	15
1.6.8. IDE de Desarrollo.....	16
1.6.9. Software para PBX.....	16
1.7. Definiciones.....	17

1.8. Conclusiones Parciales.....	17
CAPÍTULO II. EXPLORACIÓN Y PLANIFICACIÓN.....	18
2.1. Introducción.....	18
2.2. Objeto de Automatización.....	18
2.3. Propuesta del Sistema.....	18
2.4. Características del Sistema.....	19
2.4.1. Características Funcionales del Sistema.....	19
2.4.2. Características no Funcionales del Sistema.....	20
2.4.2.1. Características de apariencia o interfaz externa.....	20
2.4.2.2. Características de usabilidad.....	20
2.4.2.3. Características de rendimiento.....	20
2.4.2.4. Características de portabilidad.....	20
2.4.2.5. Características de seguridad.....	20
2.4.2.6. Características de software.....	21
2.4.2.7. Características de hardware.....	21
2.5. Arquitectura del sistema propuesto.....	21
2.5.1. Arquitectura cliente servidor.....	22
2.5.2. Estilos Arquitectónicos.....	24
2.6. Fase de Exploración.....	26
2.6.1. Flujo de procesos del sistema propuesto.....	26
2.6.2. Personas involucrados en el sistema.....	26
2.6.3. Historias de Usuario.....	26
2.7. Planificación.....	34

2.7.1.	Estimación de esfuerzo por historias de usuarios.	34
2.7.2.	Plan de Iteraciones.	35
2.7.3.	Plan de duración de las iteraciones.	35
2.7.4.	Plan de entregas.	36
2.8.	Conclusiones Parciales.....	36
CAPÍTULO III. DISEÑO DEL SISTEMA.		37
3.1.	Introducción.	37
3.2.	Patrones de Diseño.	37
3.2.1.	Patrones GOF que implementa el CEDRUX.....	37
3.2.2.	Patrones para Asignar Responsabilidades (GRASP) que implementa el CEDRUX.....	37
3.3.	Tarjetas Cargo o Clase, Responsabilidad y Colaboración (CRC).	38
3.4.	Diseño de la Base de Datos	42
3.5.	Conclusiones Parciales.....	43
CAPÍTULO IV. IMPLEMENTACIÓN Y PRUEBAS.		44
4.1.	Introducción.	44
4.2.	Iteración 1.....	44
4.3.	Iteración 2.....	52
4.4.	Iteración 3.....	52
4.5.	Pruebas.	53
4.5.1.	Pruebas unitarias.....	54
	Los errores se pudieron corregir a tiempo, por lo que la prueba arrojó un resultado satisfactorio.	55
4.5.2.	Pruebas de aceptación.	55
4.6.	Conclusiones Parciales.....	55

CAPÍTULO V. ESTUDIO DE FACTIBILIDAD.....	56
5.1. Introducción.....	56
5.2. Modelo matemático. COCOMO II.....	56
5.3. Características del Proyecto.....	57
5.3.1. Entradas Externas.....	57
5.3.2. Salidas Externas.....	57
5.3.3. Consultas Externas.....	58
5.3.4. Archivos Lógicos Internos.....	58
5.3.5. Archivos de Interfaz Externos.....	58
5.3.6. Puntos de Función Desajustados.....	59
5.4. Cálculos de instrucciones fuentes, esfuerzo, tiempo de desarrollo, cantidad de hombres y costo. 59	
5.4.1. Cálculo de instrucciones fuentes.....	59
5.4.2. Cálculo de esfuerzo nominal.....	60
5.4.3. Precedentes (PREC).....	61
5.4.4. Flexibilidad de Desarrollo (FLEX).....	61
5.4.5. Cohesión del Equipo (TEAM).....	61
5.4.6. Solución de riesgos (RESL).....	62
5.4.7. Madurez del proceso (PMAT).....	62
5.4.8. Ajuste del esfuerzo nominal.....	62
5.5. Cálculo del tiempo de desarrollo del software.....	64
5.6. Cálculo del costo total del proyecto.....	64
Cálculo de la cantidad de personas destinadas al proyecto.....	65

5.7. Resultados.....	65
5.8. Análisis de costo.....	66
5.9. Conclusiones Parciales.....	66
Conclusiones Generales.....	67
Recomendaciones.....	67
Referencias Bibliográficas.....	¡Error! Marcador no definido.
Bibliografía.....	¡Error! Marcador no definido.
Anexos.....	¡Error! Marcador no definido.
6.1. Flujo de Proceso del Sistema.....	¡Error! Marcador no definido.
6.2. Pruebas de Aceptación.....	¡Error! Marcador no definido.
6.3. Tablas de COCOMO II.....	¡Error! Marcador no definido.
Glosario de Términos.....	¡Error! Marcador no definido.

Índice de Figuras.

Figura 1. Subsistemas del proceso automatizado de encuestas telefónicas.	19
Figura 2. Propuesta de arquitectura del sistema.	22
Figura 3. Arquitectura Cliente Servidor.	23
Figura 4. Estilo Arquitectónico Modelo Vista Controlador.....	25
Figura 5. Diagrama de Clases.	39
Figura 6. Tablas de la BD.	43

Índice de Tablas.

Tabla 1 Personas relacionadas con el sistema.	26
Tabla 2 Historia de usuario: Crear Encuesta.....	27
Tabla 3 Historia de usuario: Modificar Encuesta.	27
Tabla 4 Historia de usuario: Eliminar Encuesta.....	28
Tabla 5 Historia de usuario: Buscar Encuesta.....	28
Tabla 6 Historia de usuario: Insertar Teléfono.....	29
Tabla 7. Historia de usuario: Importar Teléfonos.....	29
Tabla 8 Historia de usuario: Modificar Teléfono.	30
Tabla 9 Historia de usuario: Eliminar Teléfono.....	30
Tabla 10 Historia de usuario: Buscar Teléfono.....	31
Tabla 11 Historia de usuario: Asignar Teléfonos.....	31
Tabla 12 Historia de usuario: Mostrar Reportes.	32
Tabla 13 Historia de usuario: Realizar Encuesta.....	33
Tabla 14 Historia de usuario: Integración Web – IVR.....	33
Tabla 15 Estimación del esfuerzo por HU.	34
Tabla 16 Plan de duración de iteraciones.	35
Tabla 17 Plan de Entregas.....	36
Tabla 18. Clase EncuestaController.....	39
Tabla 19. Clase Encuesta.	39
Tabla 20. Clase ReportesController.	40
Tabla 21. Clase TelefonosController.....	40
Tabla 22. Clase Registro.....	40

Tabla 23. Clase Pregunta.	41
Tabla 24. Clase Respuesta.....	41
Tabla 25. Clase Teléfono.....	41
Tabla 26. Clase Encuesta_Telefono.	42
Tabla 27 Historias de Usuario abordadas en la primera iteración.	44
Tabla 28 Tarea #1 de la historia de usuario: Crear Encuesta.....	45
Tabla 29 Tarea #2 de la historia de usuario: Crear Encuesta.....	45
Tabla 30 Tarea #1 de la historia de usuario: Modificar Encuesta.	45
Tabla 31 Tarea #2 de la historia de usuario: Modificar Encuesta.	46
Tabla 32 Tarea #1 de la historia de usuario: Eliminar Encuesta.....	46
Tabla 33 Tarea #1 de la historia de usuario: Buscar Encuesta.	46
Tabla 34 Tarea #2 de la historia de usuario: Buscar Encuesta.	47
Tabla 35 Tarea #1 de la historia de usuario: Insertar Teléfono.	47
Tabla 36 Tarea #2 de la historia de usuario: Insertar Teléfono.	47
Tabla 37. Tarea #1 de la historia de usuario: Importar Teléfonos.....	48
Tabla 38. Tarea #2 de la historia de usuario: Importar Teléfonos.....	48
Tabla 39 Tarea #1 de la historia de usuario: Modificar Teléfono.	48
Tabla 40 Tarea #2 de la historia de usuario: Modificar Teléfono.	49
Tabla 41 Tarea #1 de la historia de usuario: Eliminar Teléfono.....	49
Tabla 42 Tarea #1 de la historia de usuario: Buscar Teléfono.	50
Tabla 43 Tarea #2 de la historia de usuario: Buscar Teléfono.	50
Tabla 44 Tarea #1 de la historia de usuario: Asignar Teléfonos.....	50
Tabla 45 Tarea #2 de la historia de usuario: Asignar Teléfonos.....	51

Tabla 46 Tarea #1 de la historia de usuario: Mostrar Reporte.....	51
Tabla 47 Tarea #2 de la historia de usuario: Mostrar Reporte.....	51
Tabla 48 Historias de Usuario abordadas en la segunda iteración.....	52
Tabla 49 Tarea #1 de la historia de usuario: Realizar Encuesta.	52
Tabla 50 Historias de Usuario abordadas en la tercera iteración.	53
Tabla 51 Tarea #1 de la historia de usuario: Integración WEB - IVR.....	53
Tabla 52 Entradas Externas.....	57
Tabla 53 Salidas Externas.	57
Tabla 54 Consultas Externas.	58
Tabla 55 Archivos Lógicos Internos.	58
Tabla 56 Archivos de Interfaz Externos.....	58
Tabla 57 Puntos de Función Desajustados.....	59
Tabla 58 Características del Sistema y Valores.	60
Tabla 59 Valores asignados a las variables.	62
Tabla 60 Clasificación de los multiplicadores del esfuerzo.....	63
Tabla 61 Multiplicadores de esfuerzo aplicados al software.....	63
Tabla 62 Resultados.....	66
Anexo 6. Tabla 63 Prueba de aceptación: HU Crear Encuesta.....	¡Error! Marcador no definido.
Anexo 7. Tabla 64 Prueba de aceptación: HU Crear Encuesta.....	¡Error! Marcador no definido.
Anexo 8. Tabla 65 Prueba de aceptación: HU Crear Encuesta.....	¡Error! Marcador no definido.
Anexo 9. Tabla 66 Prueba de aceptación: HU Crear Encuesta.....	¡Error! Marcador no definido.
Anexo 10. Tabla 67 Prueba de aceptación: HU Crear Encuesta.....	¡Error! Marcador no definido.
Anexo 11. Tabla 68 Prueba de aceptación: HU Modificar Encuesta.	¡Error! Marcador no definido.

Anexo 12. Tabla 69 Prueba de aceptación: HU Modificar Encuesta. ¡Error! Marcador no definido.

Anexo 13. Tabla 70 Prueba de aceptación: HU Modificar Encuesta. ¡Error! Marcador no definido.

Anexo 14. Tabla 71 Prueba de aceptación: HU Modificar Encuesta. ¡Error! Marcador no definido.

Anexo 15. Tabla 72 Prueba de aceptación: HU Modificar Encuesta. ¡Error! Marcador no definido.

Anexo 16. Tabla 73 Prueba de aceptación: HU Eliminar Encuesta. ¡Error! Marcador no definido.

Anexo 17. Tabla 74 Prueba de aceptación: HU Buscar Encuesta. ¡Error! Marcador no definido.

Anexo 18. Tabla 75 Prueba de aceptación: HU Buscar Encuesta. ¡Error! Marcador no definido.

Anexo 19. Tabla 76 Prueba de aceptación: HU Insertar Teléfono. ¡Error! Marcador no definido.

Anexo 20. Tabla 77 Prueba de aceptación: HU Insertar Teléfono. ¡Error! Marcador no definido.

Anexo 21. Tabla 78 Prueba de aceptación: HU Insertar Teléfono. ¡Error! Marcador no definido.

Anexo 22. Tabla 79. Prueba de aceptación: HU Importar Teléfonos. ¡Error! Marcador no definido.

Anexo 23. Tabla 80. Prueba de aceptación: HU Importar Teléfonos. ¡Error! Marcador no definido.

Anexo 24. Tabla 81. Prueba de aceptación: HU Importar Teléfonos. ¡Error! Marcador no definido.

Anexo 25. Tabla 82 Prueba de aceptación: HU Modificar Teléfono. ¡Error! Marcador no definido.

Anexo 26. Tabla 83 Prueba de aceptación: HU Modificar Teléfono. ¡Error! Marcador no definido.

Anexo 26. Tabla 84 Prueba de aceptación: HU Modificar Teléfono. ¡Error! Marcador no definido.

Anexo 27. Tabla 85 Prueba de aceptación: HU Eliminar Teléfono. ¡Error! Marcador no definido.

Anexo 28. Tabla 86 Prueba de aceptación: HU Buscar Teléfono. ¡Error! Marcador no definido.

Anexo 29. Tabla 87 Prueba de aceptación: HU Buscar Teléfono. ¡Error! Marcador no definido.

Anexo 30. Tabla 88 Prueba de aceptación: HU Asignar Teléfonos. ¡Error! Marcador no definido.

Anexo 31. Tabla 89 Prueba de aceptación: HU Asignar Teléfonos. ¡Error! Marcador no definido.

Anexo 32. Tabla 90 Prueba de aceptación: HU Asignar Teléfonos. ¡Error! Marcador no definido.

Anexo 33. Tabla 91 Prueba de aceptación: HU Mostrar Reportes. ¡Error! Marcador no definido.

- Anexo 34. Tabla 92 Prueba de aceptación: HU Realizar Encuesta..... **¡Error! Marcador no definido.**
- Anexo 35. Tabla 93 Peso del Factor de Complejidad [Boehm 1995/2]..... **¡Error! Marcador no definido.**
- Anexo 36. Tabla 94 Multiplicadores de Esfuerzo del Modelo de Diseño Temprano [COCOMO II]. **¡Error! Marcador no definido.**
- Anexo 37. Tabla 95 Factores de Escala [COCOMO II]. **¡Error! Marcador no definido.**

Introducción.

Con el paso de los años, las grandes, medianas y pequeñas empresas se han dado cuenta que su principal prioridad debe ser el cliente y que este merece ser tratado de la forma más cómoda y elegante posible.

La forma convencional de conocer el criterio de los clientes eran las encuestas personales, estas se realizaban de forma manual mediante una entrevista cara a cara con el encuestado. A medida que se fueron perfeccionando las tecnologías se fue puliendo la forma en que se realizaban las encuestas en el mundo, surgieron las encuestas telefónicas guiadas por un operador, encargado de comunicarse con los clientes e iniciar una conversación basada en preguntas y respuestas simples.

Este método ya no es suficiente para saciar las necesidades de empresas y entidades comercializadoras. Las encuestas abarcan múltiples disciplinas: economía, marketing, sociología, comunicación, entre otros, incluso son decisivas para fijar precios y salarios, por lo que los operadores no son suficientes para satisfacer la demanda de encuestas existente.

Cuba no está exenta de tales necesidades. Liberar de carga a los operadores y encuestar a gran cantidad de personas al mismo tiempo dispersas geográficamente se ha convertido en un reto para la Empresa de Telecomunicaciones de Cuba y otras empresas que requieran este servicio.

Llamar de forma manual a cada abonado y encuestar a cada cliente por separado, es la forma en que se realizan las encuestas en el país, estas permiten valorar cuantitativa y cualitativamente la aceptación de un servicio o producto determinado, o simplemente poder realizar estudios de mercado.

Por lo explicado anteriormente se puede inferir que Cuba no dispone de un sistema automatizado que realice la gestión de encuestas telefónicas utilizando una IVR¹ como sistema telefónico capaz de interactuar con el humano a través de grabaciones de voz.

Analizando lo expuesto se plantea como *problema científico* ¿Cómo automatizar el proceso de realización de las encuestas telefónicas?

¹ *Interactive Voice Response*, Respuesta de Voz Interactiva.

Para responder a esta interrogante se define como *objeto de estudio* los procesos de gestión de encuestas telefónicas.

Dadas estas condiciones y pretendiendo ofrecer la mejor solución posible es que se plantea como *objetivo general* desarrollar un sistema automatizado que gestione encuestas telefónicas utilizando una IVR.

Se enmarcan en un *campo de acción* los procesos de gestión de encuestas telefónicas utilizando una IVR.

Para darle un completo y exhaustivo cumplimiento al objetivo trazado se proponen las siguientes *tareas de la investigación*:

- ✓ Realizar un estudio del funcionamiento de todas las herramientas necesarias para la creación del sistema.
- ✓ Estudio, instalación y configuración del software para centrales telefónicas: Asterisk, para la construcción de una IVR que permita automatizar el proceso de encuestas vía telefónica.
- ✓ Estudio de las formas actuales y convencionales para la creación de encuestas.

Para lograr una solución satisfactoria, la investigación se basa en la siguiente *idea a defender*, con el sistema se logrará automatizar el proceso de gestión de encuestas vía telefónica utilizando una IVR.

Para dar soporte a la idea planteada anteriormente se utilizaron los siguientes *métodos de investigación*:

- ✓ Analítico–Sintético: Se utilizó durante el proceso de revisión bibliográfica para conocer el funcionamiento de las IVR en Asterisk como centralita telefónica y al realizar el análisis de las principales técnicas, tecnologías, metodologías de desarrollo de software para lograr la confesión de la aplicación.
- ✓ Inductivo-deductivo: Permitió transitar de lo particular a lo general y viceversa favoreciendo objetivamente el enlace que se establece en la realidad entre lo singular y lo general ya que ambas se complementan mutuamente en el proceso de desarrollo científico.
- ✓ Modelación: Se crearon abstracciones que explican la realidad, por ejemplo, todos los modelos y diagramas presentados.
- ✓ Análisis Histórico – lógico: Se utilizó durante la investigación de las tendencias actuales existentes en el mundo respecto a los sistemas de encuestas telefónicas.
- ✓ Experimento: Se utilizó para la comprobación de fenómenos y modelos durante la construcción del sistema.

El aporte práctico esperado del trabajo es: La obtención de una aplicación que permita gestionar encuestas telefónicas utilizando una IVR.

El presente trabajo de diploma consta de cinco partes fundamentales en el transcurso de la investigación realizada: resumen, introducción, desarrollo, conclusiones y bibliografía. El desarrollo a su vez viene dado por cinco capítulos, donde concentrará la mayor parte de la información emitida:

Capítulo 1: Fundamentación Teórica: En este capítulo es donde se expone todo lo relacionado con la información recopilada luego de realizar el estado del arte. Se detallan las tecnologías, tendencias y metodologías utilizadas durante la solución del problema planteado.

Capítulo 2: Exploración y Planificación: Se detallan los artefactos generados durante las fases de exploración y planificación del proyecto.

Capítulo 3: Diseño del Sistema: Durante el desarrollo de este capítulo se exponen las principales características y artefactos del flujo de diseño de la metodología de desarrollo.

Capítulo 4: Implementación y Pruebas: En este capítulo se detalla todo lo relacionado con los procesos de implementación y pruebas del sistema.

Capítulo 5: Estudio de Factibilidad: Se lleva a cabo un estudio de factibilidad del sistema, en el cual se especifica si el producto es fiable o no de acuerdo a los resultados alcanzados al finalizar el capítulo.

CAPÍTULO I. FUNDAMENTACIÓN TEÓRICA.

1.1. Introducción.

En este capítulo se realiza un estudio de herramientas existentes en el resto del mundo, relacionadas con encuestas utilizando IVR. Se definen conceptos generales, lo novedoso del producto, así como las tendencias, metodologías y herramientas de desarrollo utilizadas. Todo ello con el fin de adquirir conocimientos, los cuales serán el punto de partida del presente trabajo.

1.2. ¿Qué es una IVR?

Consiste en un sistema telefónico que es capaz de recibir una llamada e interactuar con el humano a través de grabaciones de voz y el reconocimiento de respuestas simples, como "sí", "no" u otras. Es un sistema automatizado de respuesta interactiva, orientado a entregar y/o capturar información a través del teléfono, permitiendo el acceso a servicios de información u otras operaciones. (1)

1.2.1. Tecnología IVR.

La IVR, para brindar mejores servicios, involucra otras tecnologías como:

DTMF (Detección de Marcación por Tonos): Brinda la posibilidad de reconocer que teclas ha presionado el usuario por el tono de las mismas.

TTS (Texto a Voz): Ofrece la capacidad de transformar texto a audio que escucha el operador.

ASR (Reconocimiento Automático de Voz): Brinda la posibilidad de reconocer las palabras del usuario y aceptarlas como órdenes.

1.3. Ventajas de la Respuesta de Voz Interactiva para las empresas.

Las compañías que reciben al día miles de llamadas, ya sea de consultas o de reclamaciones, obtienen con las IVR varias ventajas importantes. La posibilidad de que el usuario hable con la máquina del Call Center (Centro de Llamadas) permite en primer lugar, liberar de trabajo a los operadores y reducir el costo de las encuestas y, en segundo término, aumentar la velocidad con la que se realizan, así como administrarlas más fácilmente. (3)

1.4. Sistemas de Encuestas Telefónicas mediante una IVR en el mundo.

Realizar una investigación de mercado suele ser un proceso prolongado y costoso, pero existen nuevas oportunidades para que las empresas logren una mejor comprensión de las necesidades de sus clientes. Utilizando una variedad de herramientas, el gerente de una empresa promedio puede ahora realizar desde encuestas básicas para los clientes hasta estudios de mercado extensos sin tener que auxiliarse de terceros para realizar esta tarea. (3)

Una de estas empresas es CIO, es una marca registrada de CIO Argentina S.R.L., tiene su razón de ser en la necesidad de cambio y adaptación en el área informática que ha surgido en los últimos años en diversas empresas y organizaciones. (4)

CIO es una consultora de investigación especializada en desarrollos de marca, comunicación y negocios. Asesora a compañías e instituciones en tres áreas centrales: Marcas y Mercados, Gestión Estratégico y Comunicación.

Esta empresa brinda servicio de encuestas telefónicas con la más avanzada tecnología CATI² para Call Center. Cuentan con un equipo de trabajo especializado en planificar encuestas y sondeos telefónicos, que cuida todos los detalles para lograr un trabajo de gran valor agregado desde el campo hasta el análisis de los resultados. Permite la realización de tracking telefónico para encuestas automáticas con tecnología Asterisk sobre plataformas Linux con 100 canales concurrentes con capacidad de 30.000 encuestas diarias. (5)

Otra de las principales empresas que se encarga del estudio, creación y utilización de IVR, es la empresa DSC³, es un proveedor líder en sistemas de teléfono de contacto del centro integrado con el software de telefonía por computadora. Son capaces de crear encuestas telefónicas automatizadas utilizando una IVR, estas se pueden iniciar por llamadas de salida predeterminadas o en respuesta a personas que llamen. Puede enrutar llamadas a los agentes o a las instalaciones de servicio o transferir la llamada a una extensión al exterior. Esta ventaja mejora el sistema de llamadas del centro telefónico de las actuales características robustas, de marcación automática, transmisión de voz, distribución de llamadas y grabación de llamadas digitales. (6)

² *Computer Assisted Telephone Interview*, Entrevista Telefónica Asistida por Computadoras.

³ *DataBase System Corporation*, Corporación de Sistemas de Bases de Datos.

MKTCalls es otra de las empresas inmersas en el mundo de la telefonía automática. Es una empresa de marketing corporativo digital, utiliza servicios de mensajes pregrabados a teléfonos fijos y celulares y encuestas telefónicas interactivas automáticas a través de una IVR. El sistema permite realizar llamadas automáticamente a números de teléfono almacenados en una base de datos, difundir una o varias preguntas al interlocutor y registrar las respuestas. La empresa se beneficia, al contar con un acertado diagnóstico mediante la aplicación de una encuesta. Esta tecnología permite obtener resultados de gran confiabilidad en tiempo real a un muy bajo costo. Además aproximadamente entre el 15% y el 25% de los llamados telefónicos resultan encuestas completas. (7)

La principal deficiencia que presentan estas tres empresas en conjunto es que no exponen en su página oficial gran documentación acerca del software que utilizan para la realización de encuestas telefónicas con tecnología IVR. Además no especifican ningún vínculo donde se pueda acceder y descargar el software de manera libre o privativa. No exponen su código fuente para que pueda ser reutilizado por otras personas que quieran tratar con esta tecnología. En general, los clientes que quieran consumir este servicio deben contactar con cada una de las empresas mencionadas anteriormente.

Puesto a que no se pudo observar cómo funcionaba el servicio de encuestas telefónicas utilizando una IVR en estas compañías, ni se obtuvo ninguna versión tangible del software, es que se propone como tarea principal desarrollar un sistema que gestione las encuestas telefónicas en Cuba a través de una IVR.

1.5. Necesidad del Sistema de Gestión de Encuestas para Cuba.

Las encuestas hoy en día constituyen uno de los métodos más usados para obtener información específica de una muestra de la sociedad. Cada empresa por muy bien que funcione necesita saber que tan importante es el producto que comercializan, si posee la calidad requerida, así como, si los clientes están satisfechos con los resultados finales. Cuba necesita economizar el sistema de encuestas que utiliza actualmente, dejar atrás las personales y adentrarse en el mundo de las nuevas tecnologías, tales como las encuestas telefónicas automatizadas.

Este método ha alcanzado en los últimos años un auge increíble. Una de las ventajas es el coste y la rapidez, ya que la muestra que se quiera tomar de la sociedad puede estar dispersa geográficamente y de ser así se tuviese que invertir muchos más recursos para lograr el objetivo final. Es una técnica muy rápida ya que varios usuarios pueden estar conectados participando al mismo tiempo, sin necesidad de

hacerlos esperar. A los encuestados les puede resultar una vía más motivadora y sencilla de responder las preguntas.

No por la gran cantidad de ventajas y comodidades que posee quiere decir que no tenga puntos débiles, el cuestionario debe ser reducido, sencillo y sin preguntas abiertas.

En Cuba esta tecnología es prácticamente nueva, ya que las encuestas telefónicas se realizan de la manera convencional. ETECSA u otras entidades que deseen realizar encuestas telefónicas a un gran número de abonados deben disponer de una enorme cantidad de operadores o un gran margen de tiempo, debido que tienen que llamar uno por uno a los clientes y en caso que este, luego de interactuar con el encuestador no desee responder las preguntas elaboradas pues simplemente se retira y resulta para el operador una pérdida de tiempo, lo que con un sistema automatizado se resuelve en cuestión de minutos sin tardar el proceso por situaciones como la antes planteada.

Lo novedoso del desarrollo de un sistema de gestión de encuestas telefónicas a través de una IVR es la automatización de estas y no disponer de personal de trabajo innecesario cuando se puede encuestar a gran cantidad de clientes concurrentemente a través de grabaciones de voz. Además libera de trabajo a los operadores y resulta una vía más cómoda de realizar una encuesta.

1.6. Herramientas Utilizadas.

1.6.1. Metodología utilizada para el desarrollo del proyecto.

La Programación Extrema (PX), mejor conocida por su nombre en inglés eXtreme Programming (XP), es una de las llamadas metodologías ágiles de desarrollo de software más exitosas de los últimos tiempos. Por tal motivo es que se escogió esta metodología para la realización del sistema propuesto en este proyecto.

1.6.1.1. ¿Por qué XP?

Se decidió utilizar esta metodología debido a que la misma promueve la comunicación y el coraje entre todos los miembros del equipo de desarrollo. Se basa en miradas, entrevistas, da la posibilidad de consumir palabras de otras personas, cada miembro del equipo puede exponer sus dudas, ideas, sin tener la obligación de embellecer las mismas.

Debido a que no se puede predecir el futuro es que se tiene en cuenta XP por el nivel de simplicidad que ofrece, no se hace necesario invertir esfuerzos en algo que pueda llegar a tener valor en un futuro. Se caracteriza por la retroalimentación tanto del cliente, de los miembros del equipo, o sea en cuestión a todo lo que se mueve alrededor del tema. Es una metodología ágil debido a que los individuos e interacciones son más importantes que los procesos y herramientas, se hace mucho más imprescindible construir un buen equipo que el entorno. Un software que funcione es más importante que documentación exhaustiva, ya que los documentos deben ser cortos y con la información lo más centrada posible. La colaboración con el cliente es más importante que la negociación de contratos. Este punto es tan importante como los demás porque de una comunicación correcta, interacción constante del cliente y el equipo de desarrollo pues se logra una buena marcha del proyecto y un éxito consigo. Y por último, la respuesta ante el cambio es más importante que el seguimiento de un plan, esto indica que estar preparado para los cambios que puedan ocurrir durante el ciclo de vida del software y dar respuesta a los mismos es mucho más importante que seguir una planificación estricta y poco flexible. (17)

1.6.2. Sistemas Gestores de Bases de Datos (SGBD).

Un sistema gestor de bases de datos es una aplicación que permite a los usuarios definir, crear, mantener la base de datos y proporciona un acceso controlado a la misma.

El propósito general de estos sistemas es manejar de manera clara, sencilla y ordenada un conjunto de datos que seguidamente se convertirán en información relevante para una organización.

Para la aplicación que se desea implementar se ha escogido utilizar el siguiente SGBD:

1.6.2.1. PostgreSQL

Es uno de los Sistemas Gestores de Bases de Datos más utilizados por la comunidad de software libre por las razones siguientes:

Cumple con las propiedades ACID (Atomicidad, Consistencia, Aislamiento y Durabilidad) y soporta el lenguaje común de acceso a los datos: SQL⁴.

Es multiplataforma y posee buenas interfaces de instalación y administración. Aproxima los datos a un modelo Objeto-Relacional, y es capaz de manejar completas rutinas y reglas. Ejemplos de su avanzada

⁴ *Structured Query Language*, Lenguaje de Consultas Estructurado.

funcionalidad son consultas SQL declarativas, control de concurrencia multiversión, soporte multiusuario, transacciones y optimización de consultas.

Está basado en el proyecto POSTGRES, de la universidad de Berkeley. Es una derivación libre (Open Source) de este proyecto, y utiliza el lenguaje SQL92/SQL99. Fue el pionero en muchos de los conceptos existentes en el Sistema Objeto-Relacional actual, incluido, más tarde en otros sistemas de gestión comercial. Incluye características de la orientación a objetos, como la herencia, tipos de datos, funciones, restricciones, disparadores e integridad transaccional.

Lleva más de una década de desarrollo, siendo hoy en día un sistema bastante avanzado, que tiene soporte nativo para los lenguajes de programación: C, C++, Java, Python y PHP⁵.

Presenta las siguientes características:

- ✓ Soporta distintos tipos de datos: además del soporte para los tipos base, también soporta datos de tipo fecha, monetarios, elementos gráficos, datos sobre redes y cadenas de bits. Además permite la creación de tipos de datos propios.
- ✓ Incorpora una estructura de datos array.
- ✓ Permite la declaración de funciones propias, así como la definición de disparadores.
- ✓ Soporta el uso de índices, reglas y vistas.
- ✓ Incluye herencia entre tablas.
- ✓ Permite la gestión de diferentes usuarios, como también los permisos asignados a cada uno de ellos. (9)

1.6.3. Herramientas de Base de Datos

1.6.3.1. PgAdmin III

PgAdmin III es una aplicación gráfica para gestionar el gestor de bases de datos PostgreSQL, siendo una de las más completa y popular con licencia de código abierto. Está diseñado para responder a las necesidades de todos los usuarios, desde escribir consultas SQL simples hasta desarrollar bases de datos complejas. La interfaz gráfica soporta todas las características de PostgreSQL y facilita enormemente la administración. La aplicación también incluye un editor SQL con resaltado de sintaxis.

⁵ *Hipertext Pre – Processor, Pre - Procesador de Hipertexto.*

1.6.4. Lenguajes de programación WEB

Actualmente existen diferentes lenguajes de programación para desarrollar en la web, estos han ido surgiendo debido a las tendencias y necesidades de las plataformas.

En el sistema de gestión de encuestas que se desea desarrollar se ha decidido utilizar lenguajes de programación del lado del cliente, tales como, HTML⁶ y Java Script, y del lado del servidor, PHP.

1.6.4.1. PHP

PHP es un lenguaje interpretado de alto nivel embebido en páginas HTML y ejecutado en el servidor. PHP inició como una modificación a Perl escrita por Rasmus Lerdorf a finales de 1994. Su primer uso fue el de mantener un control sobre quién visitaba su curriculum en su web. (10). Es un lenguaje del lado del servidor, puede realizar accesos a bases de datos, conexiones en red y otras tareas para crear la página final que verá el cliente. (11)

El cliente solamente recibe una página con el código HTML resultante de la ejecución de la PHP. Como la página resultante contiene únicamente código HTML, es compatible con todos los navegadores.

La seguridad es uno de sus puntos más importantes, el hecho de que en muchas ocasiones PHP se encuentra instalado sobre servidores Unix o Linux, que son de sobra conocidos como veloces y seguros lo demuestra. Además, PHP permite configurar el servidor de modo que se permita o rechacen diferentes usos, lo que puede hacer al lenguaje más o menos seguro dependiendo de las necesidades de cada cual.

Ventajas:

- ✓ Es un lenguaje multiplataforma.
- ✓ Completamente orientado a la web.
- ✓ Capacidad de expandir su potencial utilizando la enorme cantidad de módulos (llamados exts o extensiones).
- ✓ Posee una amplia documentación en su página oficial.
- ✓ Es libre, por lo que se presenta como una alternativa de fácil acceso para todos.
- ✓ Soporta el paradigma de Programación Orientada a Objetos.
- ✓ Posee una biblioteca nativa de funciones sumamente amplia.
- ✓ No requiere definición de tipos de variables.

⁶ *HyperText Markup Language*, Lenguaje de Marcado de HiperTexto.

- ✓ Rapidez. PHP generalmente es utilizado como modulo de Apache, lo que lo hace extremadamente veloz. Está completamente escrito en C, así que se ejecuta rápidamente utilizando poca memoria.
 - ✓ Puede interactuar con muchos motores de bases de datos tales como MySQL, MS SQL, Oracle, Informix y PostgreSQL. Siempre se podrá disponer de ODBC ⁷ para situaciones que lo requieran.
- (12)

Desventajas:

- ✓ Si bien PHP no obliga a quien lo usa a seguir una determinada metodología a la hora de programar (muchos otros lenguajes tampoco lo hacen), aun estando dirigido a alguna en particular, el programador puede aplicar en su trabajo cualquier técnica de programación y/o desarrollo que le permita escribir código ordenado, estructurado y manejable. Un ejemplo de esto son los desarrollos que en PHP se han hecho del patrón de diseño Modelo Vista Controlador (MVC), que permiten separar el tratamiento y acceso a los datos, la lógica de control y la interfaz de usuario en tres componentes independientes.
- ✓ La ofuscación de código es la única forma de ocultar las fuentes.
- ✓ El manejo de errores no es tan sofisticado como Cold Fusion o ASP⁸.
- ✓ Se necesita instalar un servidor web.
- ✓ Todo el trabajo lo realiza el servidor y no delega al cliente.
- ✓ La legibilidad del código puede verse afectada al mezclar sentencias HTML y PHP.
- ✓ La programación orientada a objetos es aún muy deficiente para aplicaciones grandes.(12)

1.6.4.2. HTML.

Es un lenguaje estático para el desarrollo de sitios web.

Ventajas:

- ✓ Sencillo, permite describir hipertexto.
- ✓ El texto es presentado de forma estructurada y agradable.
- ✓ No necesita de grandes conocimientos cuando se cuenta con un editor de páginas web.
- ✓ Archivos pequeños.

⁷ *Open DataBase Connectivity*, Conectividad Abierta de Bases de Datos.

⁸ *Active Server Pages*, Páginas Servidoras Activas.

- ✓ Despliegue rápido.
- ✓ Lenguaje de fácil aprendizaje.
- ✓ Lo admiten todos los exploradores. (13)

Desventajas:

- ✓ Lenguaje estático.
- ✓ La interpretación de cada navegador puede ser diferente.
- ✓ Guarda muchas etiquetas que pueden no ser necesarias y dificultan la corrección.
- ✓ El diseño es lento.
- ✓ Las etiquetas son muy limitadas. (13)

1.6.4.3. JavaScript.

Este es un lenguaje interpretado, no requiere compilación. No es orientado a objetos ni dispone de herencias.

Ventajas:

- ✓ Lenguaje de scripting seguro y fiable.
- ✓ Los script tienen capacidades limitadas, por razones de seguridad.
- ✓ El código JavaScript se ejecuta en el cliente. (13)

Desventajas:

- ✓ Código visible por cualquier usuario.
- ✓ El código debe descargarse completamente.
- ✓ Puede poner en riesgo la seguridad del sitio con el problema llamado XSS (significa en inglés Cross Site Scripting renombrado a XSS por su similitud con las hojas de estilo CSS⁹). (13)

1.6.5. Servidores para aplicaciones web.

Un servidor web es un programa que está diseñado para transferir hipertextos, páginas web o páginas HTML. Se ejecuta en un ordenador manteniéndose a la espera de peticiones por parte de un cliente (un navegador web) y responde a estas peticiones adecuadamente mediante una página web que se exhibirá en el navegador o mostrando un mensaje si se detectó algún error.

⁹ *Cascading Style Sheets*, Hojas de Estilo de Cascada.

Se decidió utilizar el servidor Apache para alojar la aplicación web.

1.6.5.1. Apache.

Apache es el servidor web por excelencia, su alto nivel de configuración, robustez y estabilidad hacen que cada vez millones de servidores reiteren su confianza en este programa.

La popularidad de este software libre ampliamente reconocido en muchos ámbitos empresariales y tecnológicos es debido a las siguientes razones:

- ✓ Funciona en diferentes Sistemas Operativos, lo que lo hace prácticamente universal.
- ✓ Apache es una tecnología de software libre.
- ✓ Apache es un servidor altamente configurable de diseño modular. Es muy sencillo ampliar las capacidades del servidor Web Apache. Otra cosa importante es que cualquiera que posea una buena experiencia en la programación de C o Perl puede escribir un módulo para realizar una función determinada.
- ✓ Apache trabaja con gran cantidad de lenguajes script. Teniendo todo el soporte que se necesita para tener páginas dinámicas.
- ✓ Apache te permite personalizar las respuestas ante los posibles errores que puedan ocurrir en el servidor. Es posible configurar Apache para que ejecute un determinado script cuando ocurra un error en concreto.
- ✓ Tiene un alto nivel de configuración en la creación y gestión de logs. Apache permite la creación de ficheros de log a medida del administrador, de este modo puedes tener un mayor control sobre lo que sucede en el servidor. (14)

1.6.6. Frameworks.

Se refiere a una estructura software compuesta de componentes personalizables e intercambiables para el desarrollo de una aplicación.

1.6.6.1. Mapeador de Doctrine

Se utilizó para el mapeo de todas las clases del diseño, esta herramienta permitió ahorrar tiempo y esfuerzos para realizar esta labor. La herramienta se conecta a la base de datos y permite el mapeo de todas las tablas que se deseen, la única deficiencia es que no reconoce las relaciones entre las clases y hay que redefinírselas para que el mapeo quede con la calidad requerida.

Es una estructura conceptual y tecnológica de soporte definida, normalmente con artefactos o módulos de software concretos, con base en la cual otro proyecto de software puede ser organizado y desarrollado.

Se utilizó en la aplicación el Cedrux, que a su vez viene integrado por diferentes framework, tales como: Zend Framework, Doctrine, UCID y ExtJS de JavaScript.

1.6.6.2. ExtJS.

ExtJS es una librería JavaScript que permite construir aplicaciones web, es del lado del cliente. Posee un conjunto de ventajas y tiene un sistema dual de licencias: comercial y código abierto.

Ventajas:

- ✓ Código reutilizable.
- ✓ Orientada a la programación de interfaces tipo desktop en la web.
- ✓ Soporte comercial. (15)

Desventajas:

- ✓ La librería es muy grande. (15)

1.6.6.3. Zend Framework.

Se trata de un framework para desarrollo de aplicaciones Web y servicios Web con PHP. Brinda soluciones para construir sitios web modernos, robustos y seguros. Tiene una ventaja muy importante, es desarrollado por Zend y esta a su vez está respalda comercialmente a PHP.

Presenta entre otras las siguientes características:

- ✓ Simplifica la gestión de archivos de configuración.
- ✓ Proporciona los componentes que forma la infraestructura del patrón Modelo-Vista-Controlador.
- ✓ Proporciona una capa de acceso a bases de datos, construida sobre PDO¹⁰.
- ✓ Proporciona mecanismos de filtrado y validación de entradas de datos. (16)

¹⁰ *PHP Data Objects* (capa de abstracción de acceso a datos para PHP).

1.6.6.4. Doctrine ORM¹¹.

- ✓ El Framework Doctrine es un potente y completo sistema de mapas de relaciones de objetos para PHP 5.2+, con una base de datos con capas de abstracción incorporada. Además es multiplataforma.
- ✓ Una de sus principales características es la posibilidad de escribir consultas de base de datos en un dialecto orientado a objetos de propiedad SQL llamada Doctrine (Lenguaje de consulta DQL¹²), inspirado en hibernate HQL¹³. Esto proporciona a los desarrolladores una poderosa alternativa a SQL que mantiene la flexibilidad sin necesidad de duplicar código innecesario.

1.6.6.5. UCID.

Este framework fue desarrollado en la Universidad de las Ciencias Informáticas (UCI), y es el encargado del trabajo con las vistas, integrado a su vez por ExtJS.

1.6.7. Control de Versiones

Se le llama control de versiones a la gestión de los diversos cambios que se realizan sobre los elementos de algún producto o una configuración del mismo. Se decidió utilizar el TortoiseSVN.

1.6.7.1. TortoiseSVN

TortoiseSVN es un cliente gratuito de código abierto para el sistema de control de versiones Subversion. Maneja ficheros y directorios a lo largo del tiempo. Los ficheros se almacenan en un repositorio central. El repositorio es prácticamente lo mismo que un servidor de ficheros ordinario, salvo que recuerda todos los cambios que se hayan hecho a sus ficheros y directorios. Esto permite que pueda recuperar versiones antiguas de sus ficheros y examinar la historia de cuándo y cómo cambiaron sus datos, y quién hizo el cambio. Esta es la razón por la que mucha gente piensa que Subversion, y los sistemas de control de versiones en general, son una especie de máquinas del tiempo.

Los sistemas de control de versiones también son sistemas de manejo de configuración del software. Estos sistemas están diseñados específicamente para manejar árboles de código fuente, y tienen muchas características que son específicas para el desarrollo de software, tales como el entendimiento nativo de los lenguajes de programación, o proporcionan herramientas para compilar software. Subversion, sin

¹¹ *Object Relational Mapper*, Mapeador de Relaciones de Objetos.

¹² *Doctrine Query Language*, Lenguaje de Consulta de Doctrine.

¹³ *Hibernate Query Language*, Lenguaje de Consulta de Hibernate.

embargo, no es uno de estos sistemas; es un sistema general que puede ser utilizado para manejar cualquier colección de ficheros, incluyendo código fuente. (17)

1.6.8. IDE¹⁴ de Desarrollo

Un entorno de desarrollo integrado (en inglés Integrated Development Environment o IDE) es un programa compuesto por una serie de herramientas que utilizan los programadores para desarrollar código. Esta herramienta puede estar pensada para su utilización con un único lenguaje de programación o bien puede dar cabida a varios de estos. Las herramientas que normalmente componen un entorno de desarrollo integrado son las siguientes: un editor de texto, un compilador, un intérprete, unas herramientas para la automatización, un depurador, un sistema de ayuda para la construcción de interfaces gráficas de usuario y, opcionalmente, un sistema de control de versiones.

Hoy en día los entornos de desarrollo proporcionan un marco de trabajo para la mayoría de los lenguajes de programación existentes en el mercado. Además es posible que un mismo entorno de desarrollo tenga la posibilidad de utilizar varios lenguajes de programación. (18)

1.6.8.1. Zen Studio Neon

Se trata de un programa de la empresa Zend, una de las que impulsa PHP, orientada a desarrollar aplicaciones web.

Es un editor de texto para páginas PHP que proporciona un buen número de ayudas desde la creación y gestión de proyectos.

Zend Studio Neon está basado en Eclipse y es una plataforma de software de código abierto.(19)

1.6.9. Software para PBX.

1.6.9.1. Asterisk 1.8.

Asterisk es un programa de software libre (bajo licencia GPL, acrónimo de *General Public License*, Licencia Pública General) que proporciona funcionalidades de una central telefónica (PBX¹⁵). Como cualquier PBX, se puede conectar un número determinado de teléfonos para hacer llamadas entre sí.

¹⁴ *Integrated Development Environment*, Entorno de Desarrollo Integrado.

¹⁵ *Private Branch Exchange*, Central Telefónica Privada.

1.7. Definiciones

ERP: Se llama ERP a los sistemas de planificación empresariales que administran aspectos de producción, distribución y otros en una compañía. (20)

IVR: Consiste en un sistema telefónico que es capaz de recibir una llamada e interactuar con el humano a través de grabaciones de voz. Es un sistema de respuesta interactiva, orientado a entregar y/o capturar información automatizada a través del teléfono permitiendo el acceso a los servicios de información y operaciones autorizadas. (21)

1.8. Conclusiones Parciales

En el presente capítulo se arribó a la conclusión que es necesaria la implementación de un sistema que gestione las encuestas telefónicas automáticamente en Cuba, ya que esta tecnología es prácticamente nueva en Cuba y significaría un ahorro de recursos y personal técnico para la realización de encuestas. Se describieron algunos conceptos sobre la metodología, herramientas a utilizar, así como los lenguajes de programación que se escogieron para dar solución al problema.

CAPÍTULO II. EXPLORACIÓN Y PLANIFICACIÓN.

2.1. Introducción.

En el presente capítulo se hace alusión a las fases de exploración y planificación, las cuales son propias de la metodología ágil utilizada durante el desarrollo del sistema propuesto. También se detallan todos los artefactos que son generados durante el transcurso de las fases, así como la planificación del tiempo y el esfuerzo.

2.2. Objeto de Automatización.

El uso de las encuestas tradicionales en el mundo es casi obsoleto, debido a la gran cantidad de recursos que se necesitan para realizarlas, además de que imposibilita que muchas personas sean encuestadas por la posición geográfica en la que viven.

Una solución fueron las encuestas telefónicas, al contar con un dispositivo telefónico muchas veces hasta en los lugares más recónditos, se posibilita una forma de encuestar más efectiva en todos los aspectos.

Estas encuestas se realizan con la interacción telefónica entre operadores especializados y el personal encuestado, intercambiando preguntas y respuestas armónicamente.

Se considera que este proceso se puede automatizar, sustituir ese operador por un sistema computarizado que sea capaz de interactuar amablemente con los entrevistados, respetando todas las políticas y restricciones para realizar una buena encuesta, sin que las personas se sientan predispuestas a interactuar con una máquina. La aplicación informática debe ser capaz además de almacenar todos los datos obtenidos durante las encuestas, así como publicar los resultados vía web.

2.3. Propuesta del Sistema.

El sistema está diseñado para empresas que, como ETECSA, necesiten de este servicio de encuestas telefónicas automatizadas para estudios de mercado o medición de calidad de sus productos o servicios.

El mismo cuenta con dos subsistemas independientes que se integran para automatizar el proceso de realización de encuestas telefónicas, una aplicación web para administrar las encuestas, teléfonos y mostrar los resultados obtenidos, y una IVR para interactuar con los encuestados. Posee además funcionalidades que permiten la autenticación y gestión de usuarios dentro del sistema, este es implementado dentro del módulo de seguridad del framework que se utilizó.

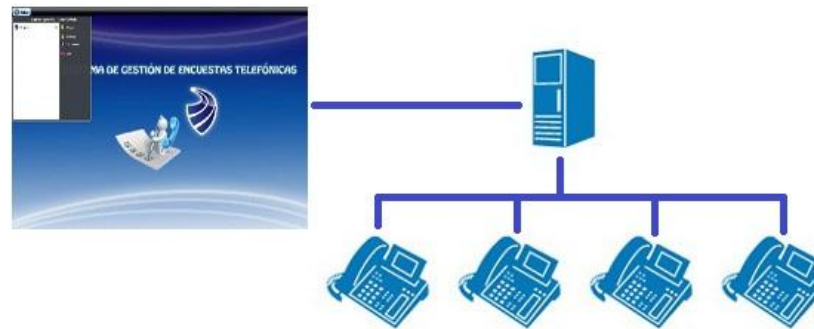


Figura 1. Subsistemas del proceso automatizado de encuestas telefónicas.

La aplicación web debe permitirle a un operador crear, modificar y eliminar encuestas telefónicas, así como consultar sus resultados. Además, debe posibilitar insertar, importar, modificar, eliminar y buscar teléfonos, además de asignárselos a las encuestas a realizarse.

El sistema IVR debe ser capaz de guiar al encuestado mediante grabaciones de voz, ya sean mensajes enunciativos o preguntas, en el proceso de realización de una o más encuestas.

2.4. Características del Sistema.

2.4.1. Características Funcionales del Sistema.

Significan las necesidades de los clientes para resolver un determinado problema o simplemente alcanzar el objetivo trazado.

Las características funcionales identificadas por el cliente fueron:

- ✓ Gestionar Encuesta, incluye Eliminar, Modificar y Añadir una encuesta.
- ✓ Gestionar Teléfonos, incluye Eliminar, Modificar, Añadir y Asignar un teléfono.
- ✓ Realizar Encuesta, se genera una IVR encargada de llamar a los teléfonos que le han sido asignados a la encuesta seleccionada.
- ✓ Mostrar Resultados, se grafican las respuestas de los abonados que han sido encuestados.

2.4.2. Características no Funcionales del Sistema.

Son aquellas características que no se refieren directamente a las funciones específicas que proporciona el sistema, sino a las propiedades emergentes de éste.

2.4.2.1. Características de apariencia o interfaz externa.

Los colores son agradables y contrastantes, el contenido fácil de manipular y la navegación organizada y amena.

2.4.2.2. Características de usabilidad.

Los administradores del sistema tienen un amplio acceso a toda la información que contiene la aplicación, además de que pueden configurar y utilizar todas las prestaciones que ofrece el sistema.

2.4.2.3. Características de rendimiento.

El sistema aprovecha los recursos disponibles en función de mejorar su rendimiento, su eficiencia, así como el tiempo de respuesta. Para lograr esto se utilizaron técnicas de elaboración de sitios web que faciliten un acceso rápido a las páginas.

2.4.2.4. Características de portabilidad.

El sistema cuenta con tres servidores, el de Bases de Datos, el Servidor web y uno para Asterisk; estos, conjuntamente con la aplicación pueden estar en una misma estación conformando una arquitectura centralizada, aunque es mucho más recomendable utilizar una arquitectura distribuida aislando los servidores en estaciones independientes. El sistema puede ser utilizado desde cualquier sistema operativo, ya que las técnicas de desarrollo web a utilizar son multiplataforma.

2.4.2.5. Características de seguridad.

La seguridad en este sistema tiene una extraordinaria importancia, por lo que se manejaron varios aspectos para la protección de la información:

Confidencialidad: La información puede ser manejada solamente por usuarios autenticados en el sistema.

Integridad: La información que se maneja dentro del sistema debe ser objeto de una cuidadosa protección contra la corrupción y estados de inconsistencia.

Disponibilidad: La información dentro de la aplicación debe estar disponible en todo momento para los usuarios autenticados, los mecanismos para lograr la seguridad necesaria del sistema no son un impedimento para acceder a la información.

2.4.2.6. Características de software.

Para acceder al sistema se necesita un navegador web que brinde soporte para DOM 2.0 y Java Script, ejemplo Mozilla Firefox. Además se necesita tener el FlashPlayer instalado.

2.4.2.7. Características de hardware.

En el cliente se requiere una PC con 256 MB¹⁶ de RAM¹⁷ como mínimo, un procesador igual o mayor a 1.40 GHZ¹⁸ y una tarjeta de red, todos los servidores deben tener 1 GB¹⁹ de RAM, 160 GB de disco duro mínimo, además deben de tener UPS²⁰, un lector de DVD²¹ o CD²² y una tarjeta de red.

2.5. Arquitectura del sistema propuesto.

La arquitectura de software consiste en un conjunto de patrones y abstracciones coherentes que proporcionan el marco de referencia necesario para guiar la construcción del Software.

El operador es el responsable de definir los números telefónicos que serán encuestados, luego, el cliente interactúa con un IVR, después de responder a las preguntas se elabora un registro con los resultados, los cuales más tarde se expondrán en la web, donde se podrá acceder a ellos. Tanto la aplicación web como el servidor Asterisk interactúan con el sistema gestor de base de datos PostgreSQL, el cual permitirá leer los datos registrados de la interacción con el cliente, así como modificar los ya existentes.

¹⁶ **MegaBytes**

¹⁷ **Random Access Memory, Memoria de Acceso Aleatorio**

¹⁸ **Gigahertz**

¹⁹ **GigaBytes**

²⁰ **Uninterruptible Power Supply, Sistema de Alimentación Ininterrumpida**

²¹ **Digital Versatile Disc, Disco Versátil Digital**

²² **Compact Disc, Disco Compacto**

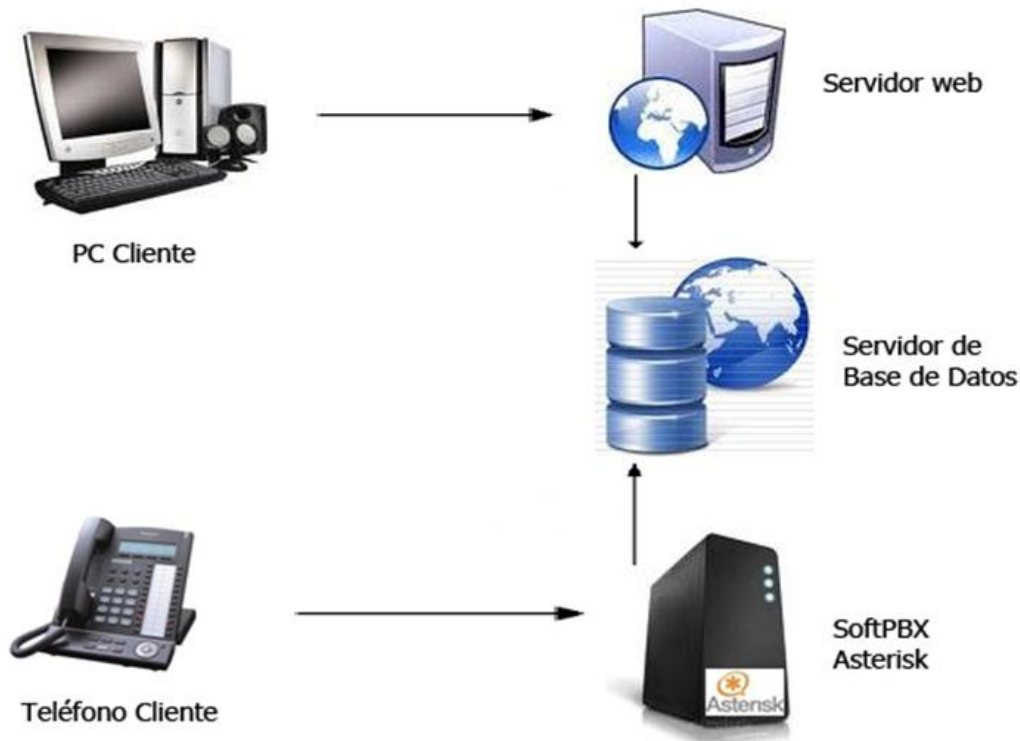


Figura 2. Propuesta de arquitectura del sistema.

2.5.1. Arquitectura cliente servidor.

Se decidió utilizar la arquitectura Cliente-Servidor para la realización del sistema debido a que se puede acceder desde cualquier lugar. Esta arquitectura consiste básicamente en un cliente que realiza peticiones a otro programa (servidor) que le da respuesta. En esta arquitectura la capacidad de proceso está repartida entre los clientes y los servidores, aunque son más importantes las ventajas de tipo organizativo debido a la centralización de la gestión de la información y la separación de responsabilidades, lo que facilita y clarifica el diseño del sistema. La separación entre cliente y servidor es una separación de tipo lógico, donde el servidor no se ejecuta necesariamente sobre una sola máquina ni es necesariamente un sólo programa.

La red cliente-servidor es aquella red de comunicaciones en la que todos los clientes están conectados a un servidor, en el que se centralizan los diversos recursos y aplicaciones con que se cuenta; y que los

pone a disposición de los clientes cada vez que estos son solicitados. Esto significa que todas las gestiones que se realizan se concentran en el servidor, de manera que en él se disponen los requerimientos provenientes de los clientes que tienen prioridad, los archivos que son de uso público y los que son de uso restringido, los archivos que son de sólo lectura y los que, por el contrario, pueden ser modificados. (22)

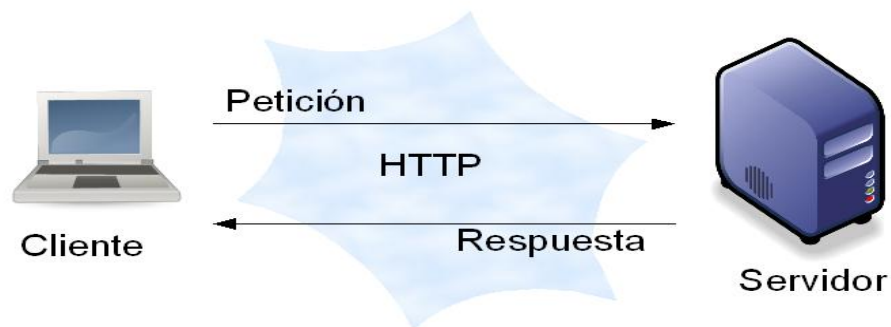


Figura 3. Arquitectura Cliente Servidor.

Ventajas:

- ✓ *Centralización del control:* Los accesos, recursos y la integridad de los datos son controlados por el servidor de forma que un programa cliente defectuoso o no autorizado no pueda dañar el sistema. Esta centralización también facilita la tarea de poner al día datos u otros recursos.
- ✓ *Escalabilidad:* Se puede aumentar la capacidad de clientes y servidores por separado. Cualquier elemento puede ser aumentado (o mejorado) en cualquier momento, o se pueden añadir nuevos nodos a la red (clientes y/o servidores).
- ✓ *Fácil mantenimiento:* Al estar distribuidas las funciones y responsabilidades entre varios ordenadores independientes, es posible reemplazar, reparar, actualizar, o incluso trasladar un servidor, mientras que sus clientes no se verán afectados por ese cambio (o se afectarán mínimamente). Esta independencia de los cambios también se conoce como encapsulación.

Desventajas:

- ✓ La congestión del tráfico ha sido siempre un problema en el paradigma de Cliente - Servidor. Cuando una gran cantidad de clientes envían peticiones simultaneas al mismo servidor, puede ser que cause muchos problemas para éste (a mayor número de clientes, más problemas para el servidor).
- ✓ El software y el hardware de un servidor son generalmente muy determinantes. Un hardware regular de un ordenador personal puede no poder servir a cierta cantidad de clientes. Normalmente se necesita software y hardware específico, sobre todo en el lado del servidor, para satisfacer el trabajo. (23)

2.5.2. Estilos Arquitectónicos

Un estilo arquitectónico define las reglas generales de organización en términos de un patrón y las restricciones en la forma y la estructura de un grupo numeroso y variado de sistemas de software. En una forma más específica, un estilo determina el vocabulario de componentes y conectores que pueden ser utilizados en instancias de este estilo, con un conjunto de restricciones en las descripciones arquitectónicas.

2.5.2.1. Modelo Vista Controlador

Para el desarrollo de la aplicación web SGET²³ se eligió el estilo arquitectónico Modelo Vista Controlador debido a la organización en partes que posee; separa los datos, la interfaz de usuario y la lógica de control en tres componentes distintos:

El Modelo es el objeto que representa los datos del programa. Maneja los datos y controla todas sus transformaciones. El Modelo no tiene conocimiento específico de los Controladores o de las Vistas, ni siquiera contiene referencias a ellos. Es el propio sistema el que tiene encomendada la responsabilidad de mantener enlaces entre el Modelo y sus Vistas, y notificar a las Vistas cuando cambia el Modelo.

La Vista es el objeto que maneja la presentación visual de los datos representados por el Modelo. Genera una representación visual del Modelo y muestra los datos al usuario. Interactúa con el Modelo a través de una referencia al propio Modelo.

²³ *Phone Surveys Management System*, Sistema de Gestión de Encuestas Telefónicas.

El Controlador es el objeto que proporciona significado a las órdenes del usuario, actuando sobre los datos representados por el Modelo. Cuando se realiza algún cambio, entra en acción, bien sea por cambios en la información del Modelo o por alteraciones de la Vista. Interactúa con el Modelo a través de una referencia al propio Modelo.

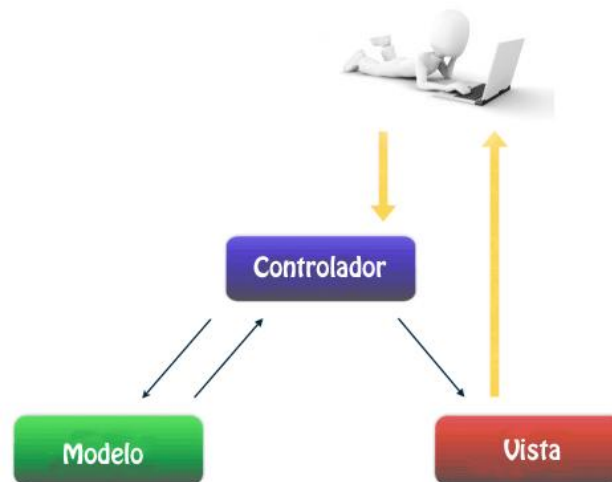


Figura 4. Estilo Arquitectónico Modelo Vista Controlador

Ventajas del MVC

Este modelo de arquitectura presenta varias ventajas:

- ✓ Hay una clara separación entre los componentes de un programa; lo cual permite implementarlos por separado.
- ✓ Hay un API²⁴ muy bien definido; cualquiera que use el API, podrá reemplazar el Modelo, la Vista o el Controlador, sin aparente dificultad.
- ✓ La conexión entre el Modelo y sus Vistas es dinámica; se produce en tiempo de ejecución, no en tiempo de compilación. (24)

²⁴ *Application Programming Interface*, Interfaz de Programación de Aplicaciones

2.6. Fase de Exploración

Esta fase es la primera de todas en la metodología XP. 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 durante el desarrollo del sistema.

2.6.1. Flujo de procesos del sistema propuesto.

El proceso comienza con la introducción de datos por parte de un operador, el sistema comprueba la corrección de los datos y lo autentica. Una vez autenticado este contará con todos los privilegios dentro de la aplicación; será capaz de crear, modificar y eliminar encuestas, además de poder analizar los datos de todas las encuestas almacenados en la base de datos y publicar sus resultados. Por otra parte, se podrá consultar los resultados publicados vía web de las encuestas que se hayan realizado. Para mayor entendimiento ver **Anexos # 1, 2, 3, 4, 5, 6 y 7.**

2.6.2. Personas involucrados en el sistema.

Se definen como involucrados en el sistema a todas aquellas personas que realizan una función o interactúan con el sistema 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 interactuar con el sistema mientras se realiza la encuesta.
Operador	Es el encargado de definir los números telefónicos a ser encuestados, gestionar las encuestas, analizar los resultados y publicarlos vía web.

2.6.3. Historias de Usuario

Las historias de usuarios representan una breve descripción del comportamiento del sistema, en ellas se emplea la terminología del cliente excluyendo el lenguaje técnico. Se emplean para hacer estimaciones de tiempo. El lenguaje de estas debe ser corto y lo más sencillo posible. Durante la fase de exploración se identificaron 12 historias de usuario, las cuales se detallan a continuación.

Se decidió tomar como 1 semana, 5 días laborables, trabajando 8 horas diarias, lo cual suma las 40 horas de trabajo semanal que propone XP.

Tabla 2 Historia de usuario: Crear Encuesta.

Historia de Usuario.	
Número: 1.	Usuario: Operador.
Nombre historia: Crear Encuesta.	
Prioridad en negocio: Alta.	Riesgo en desarrollo: Medio.
Puntos estimados: 2/5.	Iteración asignada: 1.
Programador responsable: Yasmin Diaz Espinosa – Ricardo E. Sori Echerri.	
Descripción: El operador introducirá en la aplicación web los datos de la encuesta (nombre, pregunta, respuestas, audios de la pregunta y las respuestas, y una descripción).	
Observaciones:	

Tabla 3 Historia de usuario: Modificar Encuesta.

Historia de Usuario.	
Número: 2.	Usuario: Operador.
Nombre historia: Modificar Encuesta.	
Prioridad en negocio: Media.	Riesgo en desarrollo: Medio.
Puntos estimados: 1/5.	Iteración asignada: 1.
Programador responsable: Yasmin Diaz Espinosa – Ricardo E. Sori Echerri.	
Descripción: El operador busca una encuesta por coincidencia textual, el sistema muestra todas las encuestas encontradas, se selecciona la deseada y se muestra un formulario donde se modifican los	

datos. El sistema muestra un mensaje de confirmación.
Observaciones: En caso de no encontrarse ninguna encuesta de acuerdo a la búsqueda se muestra la lista de encuestas en blanco.

Tabla 4 Historia de usuario: Eliminar Encuesta.

Historia de Usuario.	
Número: 3.	Usuario: Operador.
Nombre historia: Eliminar Encuesta.	
Prioridad en negocio: Media.	Riesgo en desarrollo: Medio.
Puntos estimados: 1/5.	Iteración asignada: 1.
Programador responsable: Yasmin Diaz Espinosa – Ricardo E. Sori Echerri.	
Descripción: El operador busca una encuesta por coincidencia textual, el sistema muestra un formulario con todas las encuestas encontradas, se selecciona la deseada y se elimina, mostrando un mensaje de confirmación.	
Observaciones: En caso de no encontrarse ninguna encuesta de acuerdo a la búsqueda se muestra la lista de encuestas en blanco.	

Tabla 5 Historia de usuario: Buscar Encuesta.

Historia de Usuario.	
Número: 4.	Usuario: Operador.
Nombre historia: Buscar Encuesta.	
Prioridad en negocio: Alta.	Riesgo en desarrollo: Medio.
Puntos estimados: 4/5.	Iteración asignada: 1.

Programador responsable: Yasmin Diaz Espinosa – Ricardo E. Sori Echerri.
Descripción: El operador busca una encuesta por coincidencia textual y el sistema es capaz de mostrar los resultados encontrados en la búsqueda.
Observaciones: En caso de no encontrarse ninguna encuesta de acuerdo a la búsqueda se muestra la lista de encuestas en blanco.

Tabla 6 Historia de usuario: Insertar Teléfono.

Historia de Usuario.	
Número: 5.	Usuario: Operador.
Nombre historia: Insertar Teléfono.	
Prioridad en negocio: Alta.	Riesgo en desarrollo: Medio.
Puntos estimados: 1/5.	Iteración asignada: 1.
Programador responsable: Yasmin Diaz Espinosa – Ricardo E. Sori Echerri.	
Descripción: El operador introducirá en la aplicación web los números telefónicos posibles a encuestar.	
Observaciones:	

Tabla 7. Historia de usuario: Importar Teléfonos.

Historia de Usuario.	
Número: 6.	Usuario: Operador.
Nombre historia: Importar Teléfonos.	
Prioridad en negocio: Alta.	Riesgo en desarrollo: Medio.
Puntos estimados: 1/5.	Iteración asignada: 1.

Programador responsable: Yasmin Diaz Espinosa – Ricardo E. Sori Echerri.
Descripción: El operador podrá insertar teléfonos a la base de datos mediante la importación de un fichero que contenga los números.
Observaciones:

Tabla 8 Historia de usuario: Modificar Teléfono.

Historia de Usuario	
Número: 7.	Usuario: Operador.
Nombre historia: Modificar Teléfono.	
Prioridad en negocio: Media.	Riesgo en desarrollo: Medio.
Puntos estimados: 1/5.	Iteración asignada: 1.
Programador responsable: Yasmin Diaz Espinosa – Ricardo E. Sori Echerri.	
Descripción: El operador busca un número de teléfono por coincidencia numérica, el sistema muestra todas los teléfonos encontrados, se selecciona el deseado y se muestra un formulario donde se modifican los datos. El sistema muestra un mensaje de confirmación.	
Observaciones: En caso de no encontrarse ningún teléfono de acuerdo a la búsqueda se muestra la lista de teléfonos en blanco.	

Tabla 9 Historia de usuario: Eliminar Teléfono.

Historia de Usuario.	
Número: 8.	Usuario: Operador.
Nombre historia: Eliminar Teléfono.	
Prioridad en negocio: Media.	Riesgo en desarrollo: Medio.

Puntos estimados: 1/5.	Iteración asignada: 1.
Programador responsable: Yasmin Diaz Espinosa – Ricardo E. Sori Echerri.	
Descripción: El operador busca un teléfono por coincidencia numérica, el sistema muestra un formulario con todos los teléfonos encontrados, se selecciona el deseado y se elimina, mostrando un mensaje de confirmación.	
Observaciones: En caso de no encontrarse ningún teléfono de acuerdo a la búsqueda se muestra la lista de teléfonos en blanco.	

Tabla 10 Historia de usuario: Buscar Teléfono.

Historia de Usuario.	
Número: 9.	Usuario: Operador.
Nombre historia: Buscar Teléfono.	
Prioridad en negocio: Alta.	Riesgo en desarrollo: Medio.
Puntos estimados: 2/5.	Iteración asignada: 1.
Programador responsable: Yasmin Diaz Espinosa – Ricardo E. Sori Echerri.	
Descripción: El operador busca un teléfono por coincidencia numérica y el sistema es capaz de mostrar los resultados encontrados en la búsqueda.	
Observaciones: En caso de no encontrarse ningún teléfono de acuerdo a la búsqueda se muestra la lista de teléfonos en blanco.	

Tabla 11 Historia de usuario: Asignar Teléfonos.

Historia de Usuario.	
Número: 10.	Usuario: Operador.

Nombre historia: Asignar Teléfonos.	
Prioridad en negocio: Alta.	Riesgo en desarrollo: Medio.
Puntos estimados: 2/5.	Iteración asignada: 1.
Programador responsable: Yasmin Diaz Espinosa – Ricardo E. Sori Echerri.	
Descripción: El operador selecciona una encuesta y le asigna los teléfonos a los que se le aplicará dicha encuesta.	
Observaciones: En caso de no existir ningún teléfono no se podrá realizar la encuesta hasta que sea insertado al menos uno.	

Tabla 12 Historia de usuario: Mostrar Reportes.

Historia de Usuario.	
Número: 11.	Usuario: Operador.
Nombre historia: Mostrar Reportes.	
Prioridad en negocio: Media.	Riesgo en desarrollo: Medio.
Puntos estimados: 1/5.	Iteración asignada: 1.
Programador responsable: Yasmin Diaz Espinosa – Ricardo E. Sori Echerri.	
Descripción: El sistema muestra un formulario con todas las encuestas que están activas, el operador o el cliente buscan y seleccionan la encuesta que desean consultar sus resultados y el sistema muestra un gráfico con dichos resultados.	
Observaciones: En caso de no encontrarse ninguna encuesta de acuerdo a la búsqueda se muestra la lista de encuestas en blanco.	

Tabla 13 Historia de usuario: Realizar Encuesta.

Historia de Usuario.	
Número: 12.	Usuario: Operador.
Nombre historia: Realizar Encuesta.	
Prioridad en negocio: Muy Alta.	Riesgo en desarrollo: Muy Alto.
Puntos estimados: 3.	Iteración asignada: 2.
Programador responsable: Yasmin Diaz Espinosa – Ricardo E. Sori Echerri.	
Descripción: Después que el operador defina los números a encuestar y ordene que se realice la encuesta, el sistema comienza la interacción con el cliente mediante preguntas y almacena los datos de las respuestas en un registro para ser publicados posteriormente.	
Observaciones: Una vez que se realice la llamada y el cliente rechace ser encuestado se termina el proceso.	

Tabla 14 Historia de usuario: Integración Web – IVR.

Historia de Usuario.	
Número: 13.	Usuario: Administrador.
Nombre historia: Integración Web – IVR.	
Prioridad en negocio: Muy Alta.	Riesgo en desarrollo: Muy Alto.
Puntos estimados: 2.	Iteración asignada: 3.
Programador responsable: Yasmin Diaz Espinosa – Ricardo E. Sori Echerri.	
Descripción: Luego de contar con todas las funcionalidades web realizadas y la IVR, se realizará la integración de estas dos partes. Una vez finalizada se podrá contar con el Sistema de Gestión de Encuestas Telefónicas en su primera versión.	
Observaciones:	

2.7. Planificación

Durante esta fase se priorizan las historias de usuarios y se estima cuánto esfuerzo requiere cada una de ellas. Se utiliza como métrica la medida de puntos. Un punto se considera como una semana ideal de trabajo, donde los miembros de los equipos de desarrollo trabajan sin interrupciones.

2.7.1. Estimación de esfuerzo por historias de usuarios.

Para el desarrollo del sistema propuesto se realizó una estimación del esfuerzo de cada una de las historias de usuarios, obteniéndose como resultados los que se muestran a continuación:

Tabla 15 Estimación del esfuerzo por HU.

Historia de Usuario.	Puntos de Estimación.
Crear Encuesta.	2/5
Modificar Encuesta.	1/5
Eliminar Encuesta.	1/5
Buscar Encuesta.	4/5
Insertar Teléfono.	1/5
Importar Teléfonos.	1/5
Modificar Teléfono.	1/5
Eliminar Teléfono.	1/5
Buscar Teléfono.	2/5
Asignar Teléfonos	2/5
Mostrar Reportes.	1/5
Realizar Encuesta.	3

Integración Web – IVR.	2
------------------------	---

2.7.2. Plan de Iteraciones.

Luego de ser identificadas las historias de usuarios y estimado el esfuerzo que requieren es que se procede a la planificación de la implementación del SGET. Para una mejor organización y desempeño del equipo de desarrollo se ha establecido la implementación dividida en tres iteraciones:

Iteración 1

Durante esta iteración se realizará la implementación de las historias de usuarios 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 y 11. Luego de finalizada esta iteración se contará con la primera versión de la aplicación web SGET.

Iteración 2

La iteración número 2 será la encargada de la implementación de la historia de usuario 12. Una vez que se finalice se contará con una versión de prueba de la IVR que estará integrada con la aplicación web SGET.

Iteración 3

Durante la iteración 3 se integrará la aplicación web con la IVR, esta iteración será la encargada de la implementación de la historia de usuario 13, al finalizar se dispondrá de una versión 1.0 del SGET, el cual se pondrá a prueba para verificar su funcionamiento.

2.7.3. Plan de duración de las iteraciones.

Este plan es el encargado de exponer cada una de las historias de usuarios que se implementarán en las iteraciones anteriormente expuestas, además de una estimación del tiempo que se demorará en desarrollarlas. El orden en que se desarrollará se muestra a continuación.

Tabla 16 Plan de duración de iteraciones.

Iteración.	Orden de la Historias de usuario a implementar.	Duración total.
1	Crear Encuesta. Modificar Encuesta. Eliminar Encuesta. Buscar Encuesta.	3 semanas.

	Insertar Teléfono. Importar Teléfonos. Modificar Teléfono. Eliminar Teléfono. Buscar Teléfono. Asignar Teléfonos. Mostrar Reportes.	
2	Realizar Encuesta.	3 semanas.
3	Integración Web – IVR.	2 semanas.

2.7.4. Plan de entregas.

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

Tabla 17 Plan de Entregas.

Módulo.	Final de la Iteración 1. (23 de Abril del 2011).	Final de la Iteración 2. (21 de Mayo del 2011).	Final de la Iteración 3. (30 de Mayo del 2011).
SGET	Web v0.1	IVR v0.1	SGET v1.0

2.8. Conclusiones Parciales.

Durante este capítulo se abordó todo lo concerniente a la fase de exploración y planificación, donde se hizo uso de cada uno de los artefactos que generan. Además se priorizaron las historias de usuarios y se acordó el alcance de las versiones.

CAPÍTULO III. DISEÑO DEL SISTEMA.

3.1. Introducción.

Durante este capítulo se logrará un diseño lo más simple posible de la solución propuesta, siempre y cuando se logre un correcto funcionamiento del sistema. Se abordarán junto con el diseño del sistema los artefactos que se generan durante el desarrollo de esta fase.

3.2. Patrones de Diseño.

Expresan esquemas para definir estructuras de diseño (o sus relaciones) con las que construir sistemas software. Es una solución estándar para un problema común de programación, una técnica para flexibilizar el código haciéndolo satisfacer ciertos criterios, un proyecto o estructura de implementación que logra una finalidad determinada, un lenguaje de programación de alto nivel, una manera más práctica de describir ciertos aspectos de la organización de un programa y la forma de un diagrama de objeto o de un modelo de objeto.

3.2.1. Patrones GOF²⁵ que implementa el CEDRUX.

Decorador: Como complemento del CEDRUX el Zend Framework implementa este patrón, el cual es el encargado de asignarle responsabilidades a objetos y añadir funcionalidades de manera dinámica.

Singleton: Garantiza la existencia de una única instancia para una clase y la creación de un mecanismo de acceso global a dicha instancia. Zend Framework posee una instancia única del controlador frontal disponible mediante este patrón para lograr una vía de entrada única a las solicitudes.

Facade: Permite utilizar una interfaz común para un conjunto de interfaces del módulo, haciendo que el mismo sea más fácil de usar.

Factory: Proporciona una interfaz para la creación de objetos interdependientes o interrelacionados, sin especificar sus clases concretas. Zend Framework utiliza este patrón en la implementación de un conjunto de clases para el acceso a datos.

3.2.2. Patrones para Asignar Responsabilidades (GRASP²⁶) que implementa el CEDRUX

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 solo pueda ser creada por el objeto que contiene la

²⁵ *Gang Of Four, Banda De Cuatro*

²⁶ *General Responsibility Assignment Software Patterns, Patrones de Software Generales para la Asignación de Responsabilidades*

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 dependencias con otras clases. Las diferentes clases controladoras solo 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 Cargo o Clase, Responsabilidad y Colaboración (CRC).

La metodología XP no recomienda la utilización de diagramas de clases, debido a la gran cantidad de documentación que requiere, aunque tampoco excluye su elaboración, en su lugar orienta la realización de Tarjetas de Cargo o CRC. Estas tarjetas se utilizan para el diseño de software orientado a objetos. Son un puente de comunicación entre diferentes participantes. Todas las ideas que se exponen son buenas, no se censuran.

Para un mejor entendimiento se muestra a continuación un diagrama de clases con sus funcionalidades y relaciones, además de las tarjetas cargo.

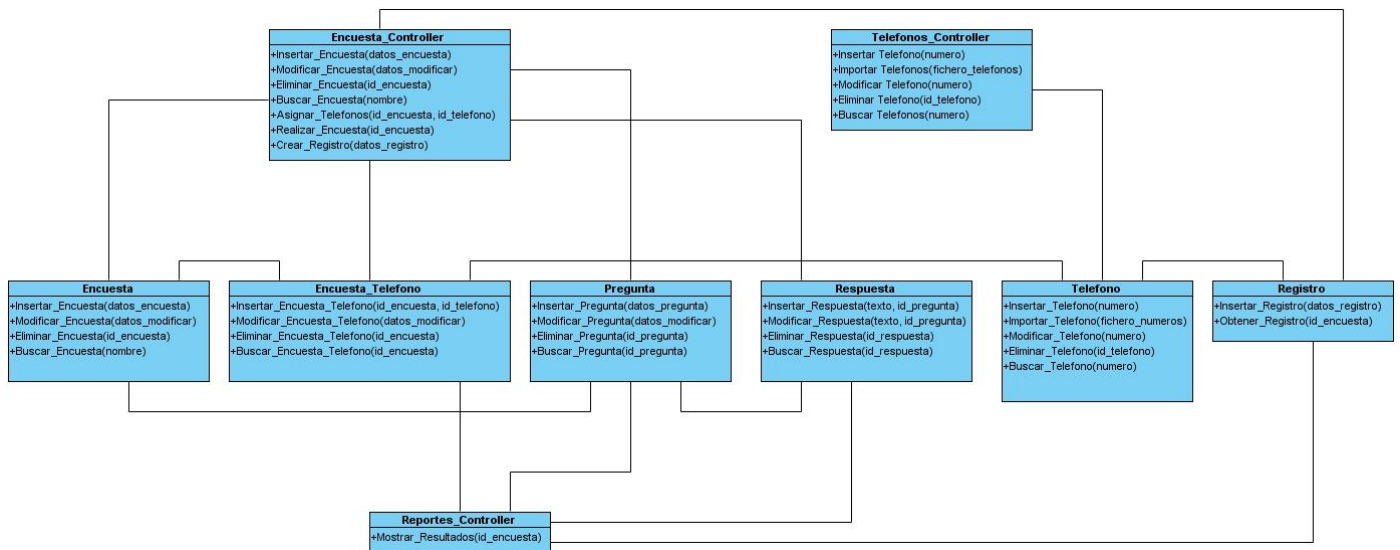


Figura 5. Diagrama de Clases.

Tabla 18. Clase EncuestaController.

Clase: EncuestaController.	
Responsabilidad.	Colaboración.
Adicionar una encuesta. Modificar los datos de una encuesta. Eliminar una encuesta. Buscar encuestas. Asignar teléfonos. Obtener datos de las encuestas. Efectuar las encuestas y crear registros después de finalizadas.	Encuesta. Respuesta. Pregunta. Telefono. Registro. Encuesta_Telefono.

Tabla 19. Clase Encuesta.

Clase: Encuesta.	
Responsabilidad.	Colaboración.

<p>Adicionar una encuesta. Modificar los datos de una encuesta. Eliminar una encuesta. Buscar una encuesta.</p>	<p>EncuestaModel.</p>
--	-----------------------

Tabla 20. Clase ReportesController.

Clase: ReportesController.	
Responsabilidad.	Colaboración.
<p>Mostrar reporte de una encuesta.</p>	<p>Encuesta. Encuesta_Telefono. Registro. Respuesta. Pregunta.</p>

Tabla 21. Clase TelefonosController.

Clase: TelefonosController.	
Responsabilidad.	Colaboración.
<p>Insertar un teléfono. Importar teléfonos. Modificar un teléfono. Eliminar un teléfono. Buscar teléfonos.</p>	<p>Encuesta. Telefono. Encuesta_Telefono. Registro. Respuesta. Pregunta.</p>

Tabla 22. Clase Registro.

Clase: Registro.	
Responsabilidad.	Colaboración.

Crear un registro después de aplicársele una encuesta a un número determinado.	RegistroModel.
--	----------------

Tabla 23. Clase Pregunta.

Clase: Pregunta.	
Responsabilidad.	Colaboración.
Insertar una pregunta. Modificar una pregunta. Eliminar una pregunta. Buscar una pregunta.	PreguntaModel.

Tabla 24. Clase Respuesta.

Clase: Respuesta.	
Responsabilidad.	Colaboración.
Insertar una respuesta. Modificar una respuesta. Eliminar una respuesta. Buscar una respuesta.	RespuestaModel.

Tabla 25. Clase Teléfono.

Clase: Teléfono.	
Responsabilidad.	Colaboración.
Insertar un teléfono. Importar teléfonos. Modificar un teléfono. Eliminar un teléfono. Buscar teléfonos.	TelefonoModel.

Tabla 26. Clase Encuesta_Telefono.

Clase: Encuesta_Telefono.	
Responsabilidad.	Colaboración.
Asignar teléfonos a las encuestas.	Encuesta_TelefonoModel.

3.4. Diseño de la Base de Datos

Una base de datos correctamente diseñada permite obtener acceso a información exacta y actualizada. Puesto que un diseño correcto es esencial para lograr los objetivos fijados para la base de datos. Un buen diseño de base de datos es, por tanto, aquél que:

- ✓ Divide la información en tablas basadas en temas para reducir los datos redundantes.
- ✓ Ayuda a garantizar la exactitud e integridad de la información.
- ✓ Satisface las necesidades de procesamiento de los datos y de generación de informes. (26)

Por tales motivos es que es imprescindible durante el desarrollo de un sistema un correcto diseño de la base de datos. A continuación se muestra el diagrama de entidad- relación del sistema propuesto:

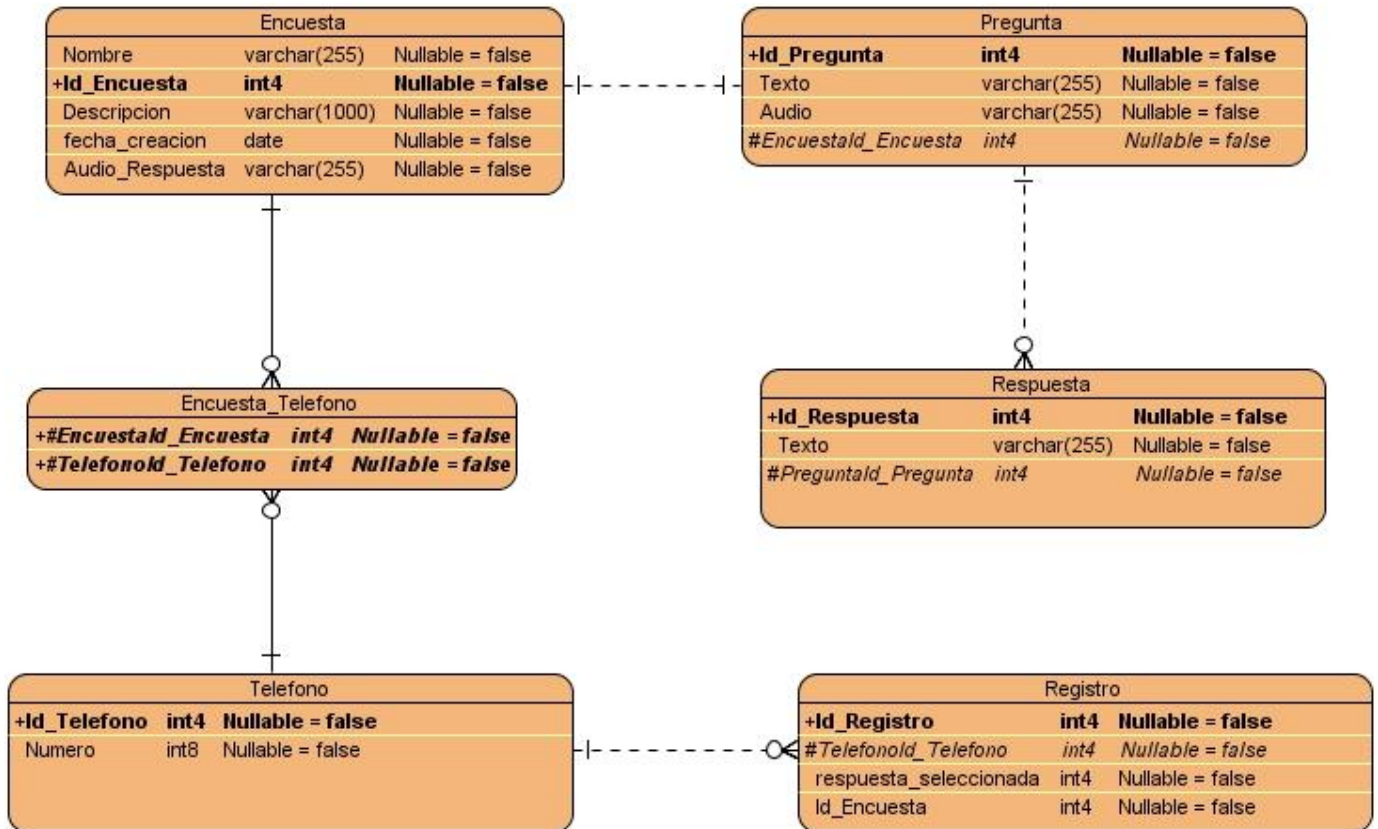


Figura 6. Diagrama Entidad-Relación.

3.5. Conclusiones Parciales.

Al concluir este capítulo se obtuvieron todos los artefactos generados durante el desarrollo de la fase de diseño, donde se realizó una descripción de los patrones de diseño, tarjetas CRC, así como el diseño de la base de datos en general.

CAPÍTULO IV. IMPLEMENTACIÓN Y PRUEBAS.

4.1. Introducción.

XP plantea que se requiere prueba y comprobación extra del funcionamiento del sistema antes de que este se pueda liberar al cliente. En esta fase, los nuevos cambios pueden todavía ser encontrados y debe tomarse la decisión de si se incluyen o no en el release actual. En el presente capítulo se detallan las tres iteraciones llevadas a cabo durante la construcción del sistema. Durante el transcurso de las iteraciones se realiza la implementación de las historias de usuario seleccionadas para cada una de estas. A continuación se detalla cada una de las iteraciones.

4.2. Iteración 1.

Durante esta iteración se abordarán las historias de usuario referentes a la versión web que se desea implementar.

Tabla 27 Historias de Usuario abordadas en la primera iteración.

Historias de Usuario.	Estimación (Días).	Real (Días).
Crear Encuesta.	2/5	2/5
Modificar Encuesta.	1/5	1/5
Eliminar Encuesta.	1/5	1/5
Buscar Encuesta.	2/5	4/5
Insertar Teléfono.	1/5	1/5
Importar Teléfonos.	1/5	1/5
Modificar Teléfono.	1/5	1/5
Eliminar Teléfono.	1/5	1/5
Buscar Teléfono.	2/5	2/5
Asignar Teléfonos.	2/5	2/5
Mostrar Reportes.	1/5	1/5

Total.	3 semanas.	3 semanas.
--------	------------	------------

Tareas de las historias de usuario abordadas en la iteración.

Tabla 28 Tarea #1 de la historia de usuario: Crear Encuesta.

Tarea de Ingeniería.	
Número Tarea: 1.	Número Historia de Usuario: 1.
Nombre Tarea: Diseñar la interfaz de Crear Encuesta.	
Tipo de Tarea: Desarrollo.	Puntos Estimados: 2/5.
Fecha Inicio: 19/01/11.	Fecha Fin: 20/01/11.
Programador Responsable: Yasmin Diaz Espinosa - Ricardo E. Sori Echerri.	
Descripción: Se diseña la interfaz de la funcionalidad Crear Encuesta, permitiendo que se introduzcan los datos para la creación de la encuesta.	

Tabla 29 Tarea #2 de la historia de usuario: Crear Encuesta.

Tarea de Ingeniería.	
Número Tarea: 2.	Número Historia de Usuario: 1.
Nombre Tarea: Implementar la funcionalidad que permita la creación de las encuestas.	
Tipo de Tarea: Desarrollo.	Puntos Estimados: 1/5.
Fecha Inicio: 21/01/11.	Fecha Fin: 22/01/11.
Programador Responsable: Yasmin Diaz Espinosa - Ricardo E. Sori Echerri.	
Descripción: Se implementa la funcionalidad Crear Encuesta, permitiendo la confección de la encuesta con los datos introducidos (nombre de la encuesta, texto de la pregunta, audio de la pregunta, texto y audio de las respuestas, y una breve descripción), estos son guardados en sus tablas correspondientes en la base de datos.	

Tabla 30 Tarea #1 de la historia de usuario: Modificar Encuesta.

Tarea de Ingeniería.	
Número Tarea: 1.	Número Historia de Usuario: 2.
Nombre Tarea: Diseñar la interfaz de Modificar Encuesta.	

Tipo de Tarea: Desarrollo.	Puntos Estimados: 1/5.
Fecha Inicio: 22/01/11.	Fecha Fin: 23/01/11.
Programador Responsable: Yasmin Diaz Espinosa - Ricardo E. Sori Echerri	
Descripción: Se diseña la interfaz de la funcionalidad Modificar Encuesta, permitiendo que se modifiquen los campos deseados.	

Tabla 31 Tarea #2 de la historia de usuario: Modificar Encuesta.

Tarea de Ingeniería.	
Número Tarea: 2.	Número Historia de Usuario: 2.
Nombre Tarea: Implementar la funcionalidad que permita la modificación de las encuestas.	
Tipo de Tarea: Desarrollo.	Puntos Estimados: 1/5.
Fecha Inicio: 23/01/11.	Fecha Fin: 24/01/11.
Programador Responsable: Yasmin Diaz Espinosa - Ricardo E. Sori Echerri	
Descripción: Se implementa la funcionalidad Modificar Encuesta, permitiendo la actualización de los datos que se deseen de la encuesta, estos son modificados en sus tablas correspondientes en la base de datos.	

Tabla 32 Tarea #1 de la historia de usuario: Eliminar Encuesta.

Tarea de Ingeniería	
Número Tarea: 1.	Número Historia de Usuario: 3.
Nombre Tarea: Implementar la funcionalidad que permita la eliminación de las encuestas.	
Tipo de Tarea: Desarrollo.	Puntos Estimados: 4/5.
Fecha Inicio: 24/01/11.	Fecha Fin: 25/01/11.
Programador Responsable: Yasmin Diaz Espinosa - Ricardo E. Sori Echerri.	
Descripción: Se implementa la funcionalidad Eliminar Encuesta, que permite eliminar la encuesta seleccionada. Los datos de la encuesta son borrados de su tabla correspondiente en la base de datos.	

Tabla 33 Tarea #1 de la historia de usuario: Buscar Encuesta.

Tarea de Ingeniería.

Número Tarea: 1.	Número Historia de Usuario: 4.
Nombre Tarea: Diseñar la interfaz de Buscar Encuesta.	
Tipo de Tarea: Desarrollo.	Puntos Estimados: 1/5
Fecha Inicio: 25/01/11.	Fecha Fin: 26/01/11.
Programador Responsable: Yasmin Diaz Espinosa - Ricardo E. Sori Echerri.	
Descripción: Se diseña la interfaz de la funcionalidad Buscar Encuesta, que permite la entrada de valores para buscar en las encuestas según su coincidencia textual.	

Tabla 34 Tarea #2 de la historia de usuario: Buscar Encuesta.

Tarea de Ingeniería.	
Número Tarea: 2.	Número Historia de Usuario: 4.
Nombre Tarea: Implementar la funcionalidad que permita la búsqueda de las encuestas.	
Tipo de Tarea: Desarrollo.	Puntos Estimados: 1/5.
Fecha Inicio: 26/01/11.	Fecha Fin: 27/01/11.
Programador Responsable: Yasmin Diaz Espinosa - Ricardo E. Sori Echerri.	
Descripción: Se implementa la funcionalidad Buscar Encuesta, permitiendo listar todas las encuestas encontradas después de buscar por coincidencia textual con los valores introducidos.	

Tabla 35 Tarea #1 de la historia de usuario: Insertar Teléfono.

Tarea de Ingeniería	
Número Tarea: 1.	Número Historia de Usuario: 5.
Nombre Tarea: Diseñar la interfaz de Insertar Teléfono.	
Tipo de Tarea: Desarrollo.	Puntos Estimados: 1/5.
Fecha Inicio: 27/01/11.	Fecha Fin: 28/01/11.
Programador Responsable: Yasmin Diaz Espinosa - Ricardo E. Sori Echerri.	
Descripción: Se diseña la interfaz de la funcionalidad Insertar Teléfono, permitiendo que se introduzcan los datos para la inserción del teléfono.	

Tabla 36 Tarea #2 de la historia de usuario: Insertar Teléfono.

Tarea de Ingeniería.

Número Tarea: 2.	Número Historia de Usuario: 5.
Nombre Tarea: Implementar la funcionalidad que permita la inserción de los teléfonos.	
Tipo de Tarea: Desarrollo.	Puntos Estimados: 1/5.
Fecha Inicio: 28/01/11.	Fecha Fin: 29/01/11.
Programador Responsable: Yasmin Diaz Espinosa - Ricardo E. Sori Echerri.	
Descripción: Se implementa la funcionalidad Insertar Teléfono, permitiendo la creación del teléfono con los datos introducidos (número de teléfono), estos son guardados en sus tablas correspondientes en la base de datos.	

Tabla 37. Tarea #1 de la historia de usuario: Importar Teléfonos.

Tarea de Ingeniería	
Número Tarea: 1.	Número Historia de Usuario: 6.
Nombre Tarea: Diseñar la interfaz de Importar Teléfonos.	
Tipo de Tarea: Desarrollo.	Puntos Estimados: 1/5.
Fecha Inicio: 1/02/11.	Fecha Fin: 2/02/11.
Programador Responsable: Yasmin Diaz Espinosa - Ricardo E. Sori Echerri.	
Descripción: Se diseña la interfaz de la funcionalidad Importar Teléfonos, permitiendo la posibilidad de importar desde un fichero una lista de teléfonos para ser insertados en la base de datos.	

Tabla 38. Tarea #2 de la historia de usuario: Importar Teléfonos.

Tarea de Ingeniería.	
Número Tarea: 2.	Número Historia de Usuario: 6.
Nombre Tarea: Implementar la funcionalidad que permita la importación de teléfonos desde un fichero.	
Tipo de Tarea: Desarrollo.	Puntos Estimados: 1/5.
Fecha Inicio: 3/02/11.	Fecha Fin: 4/02/11.
Programador Responsable: Yasmin Diaz Espinosa - Ricardo E. Sori Echerri.	
Descripción: Se implementa la funcionalidad Importar Teléfonos, permitiendo la lectura de un fichero y posterior inserción de teléfonos en la base de datos.	

Tabla 39 Tarea #1 de la historia de usuario: Modificar Teléfono.

Tarea de Ingeniería.

Número Tarea: 1.	Número Historia de Usuario: 6.
Nombre Tarea: Diseñar la interfaz de Modificar Teléfono.	
Tipo de Tarea: Desarrollo.	Puntos Estimados: 1/5.
Fecha Inicio: 4/02/11.	Fecha Fin: 5/02/11.
Programador Responsable: Yasmin Diaz Espinosa - Ricardo E. Sori Echerri.	
Descripción: Se diseña la interfaz de la funcionalidad Modificar Teléfono, permitiendo que se modifiquen los datos deseados.	

Tabla 40 Tarea #2 de la historia de usuario: Modificar Teléfono.

Tarea de Ingeniería.	
Número Tarea: 2.	Número Historia de Usuario: 6.
Nombre Tarea: Diseñar la interfaz de Modificar Teléfono.	
Tipo de Tarea: Desarrollo.	Puntos Estimados: 1/5.
Fecha Inicio: 5/02/11.	Fecha Fin: 6/02/11.
Programador Responsable: Yasmin Diaz Espinosa - Ricardo E. Sori Echerri.	
Descripción: Se implementa la funcionalidad Modificar Teléfonos, permitiendo la actualización de los datos que se deseen del teléfono, estos son modificados en sus tablas correspondientes en la base de datos.	

Tabla 41 Tarea #1 de la historia de usuario: Eliminar Teléfono.

Tarea de Ingeniería.	
Número Tarea: 1.	Número Historia de Usuario: 7.
Nombre Tarea: Implementar la funcionalidad que permita la eliminación de los teléfonos.	
Tipo de Tarea: Desarrollo.	Puntos Estimados: 1/5.
Fecha Inicio: 6/02/11.	Fecha Fin: 7/02/11.
Programador Responsable: Yasmin Diaz Espinosa - Ricardo E. Sori Echerri.	
Descripción: Se implementa la funcionalidad Eliminar Teléfono, que permite eliminar el teléfono seleccionado. Los datos del teléfono son borrados de su tabla correspondiente en la base de datos.	

Tabla 42 Tarea #1 de la historia de usuario: Buscar Teléfono.

Tarea de Ingeniería.	
Número Tarea: 1.	Número Historia de Usuario: 8.
Nombre Tarea: Diseñar la interfaz de Buscar Teléfono.	
Tipo de Tarea: Desarrollo.	Puntos Estimados: 2/5.
Fecha Inicio: 8/02/11.	Fecha Fin: 9/02/11.
Programador Responsable: Yasmin Diaz Espinosa - Ricardo E. Sori Echerri.	
Descripción: Se diseña la interfaz de la funcionalidad Buscar Teléfono, que permite la entrada de valores para buscar en los teléfonos según su coincidencia textual.	

Tabla 43 Tarea #2 de la historia de usuario: Buscar Teléfono.

Tarea de Ingeniería.	
Número Tarea: 2.	Número Historia de Usuario: 8.
Nombre Tarea: Implementar la funcionalidad que permita la búsqueda de los teléfonos.	
Tipo de Tarea: Desarrollo.	Puntos Estimados: 1/5.
Fecha Inicio: 10/02/11.	Fecha Fin: 11/02/11.
Programador Responsable: Yasmin Diaz Espinosa - Ricardo E. Sori Echerri.	
Descripción: Se implementa la funcionalidad Buscar Teléfono, permitiendo listar todos los teléfonos encontrados después de buscar por coincidencia textual con los valores introducidos.	

Tabla 44 Tarea #1 de la historia de usuario: Asignar Teléfonos.

Tarea de Ingeniería.	
Número Tarea: 1.	Número Historia de Usuario: 9.
Nombre Tarea: Diseñar la interfaz de Asignar Teléfonos.	
Tipo de Tarea: Desarrollo.	Puntos Estimados: 2/5.
Fecha Inicio: 12/02/11.	Fecha Fin: 13/02/11.
Programador Responsable: Yasmin Diaz Espinosa - Ricardo E. Sori Echerri.	
Descripción: Se diseña la interfaz de la funcionalidad Asignar Teléfonos, que permite asignar teléfonos a la encuesta seleccionada.	

Tabla 45 Tarea #2 de la historia de usuario: Asignar Teléfonos.

Tarea de Ingeniería.	
Número Tarea: 2.	Número Historia de Usuario: 9.
Nombre Tarea: Implementar la funcionalidad que permita la asignación de los teléfonos a las encuestas.	
Tipo de Tarea: Desarrollo.	Puntos Estimados: 2/5.
Fecha Inicio: 14/02/11.	Fecha Fin: 15/02/11.
Programador Responsable: Yasmin Diaz Espinosa - Ricardo E. Sori Echerri.	
Descripción: Se implementa la funcionalidad Asignar Teléfonos, permitiendo asignar los teléfonos uno a uno o todos los teléfonos existentes a la encuesta seleccionada registrándose los cambios en la base de datos.	

Tabla 46 Tarea #1 de la historia de usuario: Mostrar Reporte.

Tarea de Ingeniería.	
Número Tarea: 1.	Número Historia de Usuario: 10.
Nombre Tarea: Diseñar la interfaz de Mostrar Reporte.	
Tipo de Tarea: Desarrollo.	Puntos Estimados: 1/5.
Fecha Inicio: 16/02/11.	Fecha Fin: 17/02/11.
Programador Responsable: Yasmin Diaz Espinosa - Ricardo E. Sori Echerri.	
Descripción: Se diseña la interfaz de la funcionalidad Mostrar Reporte, que permite mostrar los resultados en una gráfica de la encuesta seleccionada.	

Tabla 47 Tarea #2 de la historia de usuario: Mostrar Reporte.

Tarea de Ingeniería.	
Número Tarea: 2.	Número Historia de Usuario: 10.
Nombre Tarea: Implementar la funcionalidad que permita mostrar los reportes de las encuestas.	
Tipo de Tarea: Desarrollo.	Puntos Estimados: 1/5.
Fecha Inicio: 18/02/11.	Fecha Fin: 19/02/11.
Programador Responsable: Yasmin Diaz Espinosa - Ricardo E. Sori Echerri.	
Descripción: Se implementa la funcionalidad Mostrar Reporte, permitiendo obtener de la base de datos	

los resultados de las encuestas realizadas para ser mostrados en una gráfica.

4.3. Iteración 2.

Durante el transcurso de la presente iteración se registrará la historia de usuario concerniente a la IVR, la cual será la encargada de la interacción con el cliente.

Tabla 48 Historias de Usuario abordadas en la segunda iteración.

Historias de Usuario.	Estimación.	Real.
Realizar Encuesta.	3	3
Total.	3 semanas.	3 semanas.

Tareas de las historias de usuario abordadas en la iteración.

Tabla 49 Tarea #1 de la historia de usuario: Realizar Encuesta.

Tarea de Ingeniería.	
Número Tarea: 1.	Número Historia de Usuario: 11.
Nombre Tarea: Implementar la funcionalidad que permita realizar las encuestas.	
Tipo de Tarea: Desarrollo.	Puntos Estimados: 3.
Fecha Inicio: 17/02/11.	Fecha Fin: 18/02/11.
Programador Responsable: Yasmin Diaz Espinosa - Ricardo E. Sori Echerri.	
Descripción: Se implementa la funcionalidad Realizar Encuesta, permitiendo llamar a los teléfonos previamente fijados, la interacción entre el cliente y la IVR y la creación de un registro con los datos necesarios (id de la encuesta, id del teléfono y la respuesta seleccionada) después de finalizada la encuesta.	

4.4. Iteración 3.

Durante esta iteración se implementará la historia de usuario restante que involucra funcionalidades concernientes a la integración de la aplicación web con el sistema IVR. Al culminar esta, se consta de un producto listo para su puesta en funcionamiento.

Tabla 50 Historias de Usuario abordadas en la tercera iteración.

Historias de Usuario.	Estimación.	Real.
Integración Web – IVR.	2	2
Total.	2 semanas.	2 semanas.

Tareas de las historias de usuario abordadas en la iteración.

Tabla 51 Tarea #1 de la historia de usuario: Integración WEB - IVR.

Tarea de Ingeniería.	
Número Tarea: 1.	Número Historia de Usuario: 12.
Nombre Tarea: Implementar la funcionalidad que permita integrar la aplicación web con la IVR.	
Tipo de Tarea: Desarrollo.	Puntos Estimados: 2.
Fecha Inicio: 20/02/11.	Fecha Fin: 21/02/11.
Programador Responsable: Yasmin Diaz Espinosa - Ricardo E. Sori Echerri.	
Descripción: Se implementa la funcionalidad Integración WEB - IVR, permitiendo la interrelación del sistema de administración web con el sistema telefónico para la realización de las encuestas, obteniéndose la primera versión del SGET.	

4.5. Pruebas.

Uno de los pilares de la eXtreme Programming es el proceso de 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 detección. 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, encargadas de verificar el código y diseñada por los programadores, y pruebas de aceptación o pruebas funcionales destinadas a evaluar si al final de una iteración se consiguieron las funcionalidades requeridas, diseñadas por el cliente final. (27)

4.5.1. Pruebas unitarias

Las pruebas unitarias o pruebas de unidad consisten en comprobaciones (manuales o automatizadas) desarrolladas por los programadores. Las cuales se realizan para verificar que el código correspondiente a un módulo concreto se comporta de manera esperada.

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. (28)

En el transcurso del desarrollo de la aplicación se aplicaron una serie de este tipo de pruebas, que ayudaron a corregir errores encontrados dentro del código fuente del sistema. Un ejemplo lo constituye el siguiente fragmento:

```
var nombreenc = new Ext.form.TextField({
  id:'nombre',
  allowBlank:false,
  width:'385',
  fieldLabel:'Nombre de la Encuesta'
});
```

Este código permitía que al campo “nombre de la encuesta” se le pudieran introducir datos no deseados como caracteres extraños, lo que se pudo corregir mediante una validación utilizando una expresión regular en JavaScript.

```
var validarnombre=/^[a-zA-Z áéíóúñÑ.0-9]*$/;
```

```
var nombreenc = new Ext.form.TextField({
  id:'nombre',
  allowBlank:false,
```

```

regex:validarnombre,
width:'385',
fieldLabel:'Nombre de la Encuesta'
});

```

Tabla 52. Cantidad de Pruebas Unitarias realizadas.

Cantidad de Pruebas Realizadas.	Cantidad de Pruebas Satisfactorias.	Cantidad de Errores Encontrados.	Cantidad de Errores Corregidos.
37	29	8	8

Los errores se pudieron corregir a tiempo, por lo que las pruebas arrojaron un resultado satisfactorio.

4.5.2. Pruebas de aceptación.

Las pruebas de aceptación son consideradas como pruebas de caja negra. Los clientes son responsables de verificar que los resultados de estas pruebas sean correctos. Así mismo, en caso de que fallen varias pruebas, deben indicar el orden de prioridad de resolución. Una historia de usuario no se puede considerar terminada hasta tanto pase correctamente todas las pruebas de aceptación.

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. (28)

Para un mayor entendimiento ver **Anexos 8, 9, ..., 37.**

4.6. Conclusiones Parciales.

En este capítulo se detallaron las historias de usuario de cada una de las iteraciones definidas, haciendo uso de las tareas de ingeniería identificadas por historia de usuario, elaborando a su vez las pruebas de aceptación para confirmar que las historias de usuario han sido implementadas correctamente.

CAPÍTULO V. ESTUDIO DE FACTIBILIDAD.

5.1. Introducción.

En el presente capítulo es donde se realiza la estimación de los recursos necesarios para el desarrollo del proyecto. Es un análisis comprensivo que sirve para recopilar los datos relevantes sobre el desarrollo de un proyecto y en base a ello tomar la mejor decisión. (29) Para la realización del sistema propuesto se realizará un estudio de factibilidad que se centra en la estimación de tamaño, esfuerzo y planificación.

5.2. Modelo matemático. COCOMO II.

COCOMO ²⁷ es un modelo matemático de base empírica utilizado para estimación de costes de software. Incluye tres submodelos, cada uno ofrece un nivel de detalle y aproximación, cada vez mayor, a medida que avanza el proceso de desarrollo del software: básico, intermedio y detallado. Este modelo fue desarrollado por Barry W. Boehm a finales de los años 70 y comienzos de los 80, exponiéndolo detalladamente en su libro "Software Engineering Economics". (30)

Características.

Pertenece a la categoría de modelos de subestimaciones basados en estimaciones matemáticas. Está orientado a la magnitud del producto final, midiendo el tamaño del proyecto, en líneas de código principalmente.

Inconvenientes.

- ✓ Los resultados no son proporcionales a las tareas de gestión ya que no tiene en cuenta los recursos necesarios para realizarlas.
- ✓ Se puede desviar de la realidad si se indica mal el porcentaje de líneas de comentarios en el código fuente.
- ✓ Es un tanto subjetivo, puesto que está basado en estimaciones y parámetros que pueden ser vistos de distinta manera por distintos analistas que usen el método.
- ✓ Se miden los costes del producto, de acuerdo a su tamaño, pero no la productividad.
- ✓ La medición por líneas de código no es válida para orientación a objetos.

²⁷ COⁿstructive CO^st MO^del, *Modelo Constructivo de Costes*

- ✓ Utilizar este modelo puede resultar un poco complicado, en comparación con otros métodos (que también sólo estiman). (26)

5.3. Características del Proyecto.

Para la estimación del proyecto se hace necesaria la obtención de los Puntos de Función Desajustados, los cuales vienen dados por la suma de las Entradas, Salidas y Consultas Externas, así como los Archivos Lógicos Internos y Archivos Interfaz Externos.

5.3.1. Entradas Externas.

Las Entradas Externas no son más que entrada de datos del usuario o de control que ingresan desde el exterior del sistema para agregar y/o cambiar datos a un archivo lógico interno. (31)

Tabla 53 Entradas Externas.

Nombre de la entrada externa.	Cantidad de ficheros.	Cantidad de elementos de datos.	Clasificación (Simple, Media, Compleja).
Crear Encuesta.	1	6	Simple.
Modificar Encuesta.	1	6	Simple.
Eliminar Encuesta.	1	1	Simple.
Buscar Encuesta.	1	1	Simple.
Insertar Teléfono.	1	1	Simple.
Modificar Teléfono.	1	1	Simple.
Eliminar Teléfono.	1	1	Simple.
Buscar Teléfono.	1	1	Simple.
Total.		18	

5.3.2. Salidas Externas.

Las Salidas Externas no son más que salida de datos de usuario o de control que deja el límite del sistema de software. (31)

Tabla 54 Salidas Externas.

Nombre de la salida externa.	Cantidad de	Cantidad de	Clasificación (Simple,
------------------------------	-------------	-------------	------------------------

	ficheros.	elementos de datos.	Media, Compleja).
Realizar Encuesta.	1	1	Simple.
Mostrar Reportes.	1	1	Simple.
Total.		2	

5.3.3. Consultas Externas.

Las Consultas Externas no son más que una combinación única de entrada-salida, donde una entrada causa y genera una salida inmediata, como un tipo de solicitud externa. (31)

Tabla 55 Consultas Externas.

Nombre de la Petición.	Cantidad de ficheros.	Cantidad de elementos de datos.	Clasificación (Simple, Media, Compleja).
Total.		0	

5.3.4. Archivos Lógicos Internos.

Incluye cada archivo lógico, es decir cada grupo lógico de datos que es generado, usado, o mantenido por el sistema de software. (31)

Tabla 56 Archivos Lógicos Internos.

Nombre de la Petición.	Cantidad de ficheros.	Cantidad de elementos de datos.	Clasificación (Simple, Media, Compleja).
Base Batos: Encuesta.	1	6(tablas).	Simple.
IVR.	1	1	Simple.
Total.		7	

5.3.5. Archivos de Interfaz Externos.

Archivos transferidos o compartidos entre sistemas de software. (31)

Tabla 57 Archivos de Interfaz Externos.

Nombre de la interfaz externa.	Cantidad de ficheros.	Cantidad de elementos de datos.	Clasificación (Simple, Media, Compleja).
	0	0	

Total.		0	
---------------	--	----------	--

5.3.6. Puntos de Función Desajustados.

Los Puntos de Función procuran cuantificar la funcionalidad de un sistema de software. La meta es obtener un número que caracterice completamente al sistema. Son útiles estimadores ya que están basados en información que está disponible en las etapas tempranas del ciclo de vida del desarrollo de software. (31)

Tabla 58 Puntos de Función Desajustados.

Elementos.	Simple.		Medio.		Complejo.		Subtotal.
	No.	Peso.	No.	Peso.	No.	Peso.	
Entradas Externas.	18	3	0	4	0	6	54
Salidas Externas.	2	4	0	5	0	7	8
Consultas Externas.	0	3	0	4	0	6	0
Fichero Lógico Interno.	7	7	0	10	0	15	49
Fichero Interfaz Externo.	0	5	0	7	0	10	0
Total (UFP):							111

5.4. Cálculos de instrucciones fuentes, esfuerzo, tiempo de desarrollo, cantidad de hombres y costo.

5.4.1. Cálculo de instrucciones fuentes.

Luego de contar con la cantidad de puntos de función desajustados es que se procede a calcular la cantidad de instrucciones fuentes:

$$SLOC = UFP \times ratio$$

$$SLOC = 111 \times 59$$

$$SLOC = 6549$$

Donde:

SLOC: Cantidad de instrucciones fuente (6549).

UFP : Puntos de función desajustados. (111)

ratio : Conversión de puntos de función desajustados a líneas de código para el lenguaje PHP [59].

Partiendo de la ecuación anterior se obtiene el siguiente resultado:

KSLOC : **SLOC/1000** (Tamaño del software a desarrollar).

$$KSLOC = 6549/1000 = 6.5$$

Las características del sistema y valores obtenidos anteriormente han sido plasmadas en la siguiente tabla:

Tabla 59 Características del Sistema y Valores.

Características.	Valor.
Puntos de función desajustados.	111
Lenguaje (PHP).	59
Instrucciones fuentes por puntos de función.	6549 SLOC
Instrucciones fuentes.	6.5 KSLOC

5.4.2. Cálculo de esfuerzo nominal.

Posteriormente se realiza el cálculo del esfuerzo nominal para lo cual se utiliza la ecuación que se muestra a continuación:

$$PM_{Nominal} = Ax(Size)^B$$

$$PM_{Nominal} = 2.94 x(6.5)^{0.99}$$

$$PM_{Nominal} = 18.76 \text{ meses/hombre}$$

Donde:

PM_{Nominal}: Esfuerzo nominal requerido en meses – hombres.

Size: Tamaño estimado del software en Puntos de Función sin Ajustar. (6.5 KSLOC)

A: Constante que se utiliza para capturar los efectos multiplicativos en el esfuerzo requerido de acuerdo al crecimiento de tamaño del software. (2.94).

B: Constante denominada Factor escalar y su valor está dado por la resultante de los aspectos positivos sobre los negativos que presenta el proyecto.

$$B = 0.91 + 0.01 \times \sum (W_i)$$

$$B = 0.91 + 0.01 \times 7.74$$

$$B = 0.99$$

Donde:

W_i : Variables escalares que indican las características que el proyecto presenta en lo que a su complejidad y entorno de desarrollo se refiere.

5.4.3. Precedentes (PREC).

El factor de precedencia (PREC) toma en cuenta el grado de experiencia previa en relación al producto a desarrollar, tanto en aspectos organizacionales como en el conocimiento del software y hardware a utilizar.

5.4.4. Flexibilidad de Desarrollo (FLEX).

El factor de flexibilidad (FLEX) considera el nivel de exigencia en el cumplimiento de los requerimientos preestablecidos, plazos de tiempos y especificaciones de interfaz.

5.4.5. Cohesión del Equipo (TEAM).

El factor de escala de Cohesión del Equipo explica los recursos de turbulencia y entropía del proyecto debido a dificultades en la sincronización de los implicados en el proyecto, usuarios, clientes, desarrolladores y los que lo mantienen. Estas dificultades pueden aparecer por las diferentes culturas y objetivos de los implicados; dificultades en conciliar objetivos y la falta de experiencia y de familiaridad de los implicados en trabajar como un equipo. (32)

5.4.6. Solución de riesgos (RESL).

Este factor combina dos factores de medida de Ada COCOMO “Minuciosidad del diseño por revisión del diseño del producto (PDR)” y “Eliminación de riesgos por PDR”. La medida de RESL es la media pesada subjetiva de las características de la lista. (32)

5.4.7. Madurez del proceso (PMAT).

El procedimiento para determinar PMAT se obtiene a través del Modelo de Madurez de Capacidad del Instituto de Ingeniería del Software (CMM). El período de tiempo para medir la madurez del proceso es el momento en el que el proyecto comienza. (32)

La siguiente tabla muestra los valores asignados a cada una de estas variables:

Tabla 60 Valores asignados a las variables.

Nombre.	Valor.	Justificación.
PREC	1.24	Existen varios proyectos similares a nivel internacional, pero no se encontró ninguno a nivel nacional.
FLEX	1.01	Cuenta con alta flexibilidad en cuanto a los requerimientos establecidos inicialmente.
TEAM	1.10	El equipo de desarrollo presenta una alta cohesión.
RESL	2.83	No se identificaron grandes riesgos.
PMAT	1.56	Se cuenta con la experiencia necesaria como para que el software cumpla con las funcionalidades requeridas.
Total(SF)	7.74	

5.4.8. Ajuste del esfuerzo nominal.

Cada nivel de medida de cada esfuerzo nominal tiene un peso asociado. El peso se llama multiplicador de esfuerzo (EM). Los EM se usan para ajustar el esfuerzo Meses-persona nominal. Hay 7 multiplicadores de esfuerzo para el Modelo de Diseño Anticipado y 17 para el modelo de Post-Arquitectura.

Tabla 61 Clasificación de los multiplicadores del esfuerzo.

Multiplicadores del esfuerzo.	
Del product.	RCPX: Confiabilidad y Complejidad del producto.
	RUSE: Reusabilidad Requerida.
De la Plataforma.	PDIF: Dificultad de la Plataforma.
Del Proyecto.	FCIL: Facilidades.
	SCED: Cronograma de Desarrollo Requerido.
Del Personal.	PERS: Aptitud del Personal.
	PREX: Experiencia del Personal.

Tabla 62 Multiplicadores de esfuerzo aplicados al software.

Nombre.	Valor.	Justificación.
RCPX	1.30	La confidencialidad del sistema es alta.
RUSE	1.00	Pretende reutilizarse una parte del código.
PDIF	1.00	La plataforma de desarrollo es estable, el uso de la memoria y almacenamiento es normal.
PREX	0.71	Nivel de experiencia en el personal en cuanto a la utilización del lenguaje y herramientas.
PERS	0.83	La aptitud del personal es alta.
FCIL	0.73	La utilización de entornos de desarrollo integrados facilita en gran medida el desarrollo de la aplicación.
SCED	1.00	El sistema se desarrolló en el tiempo establecido.
Total(EM)	0.56	

$$PM_{ajustado} = PM_{Nominal} \times \prod (ME_t)$$

$$PM_{ajustado} = 18.76 \text{ meses / hombre} \times 0.56$$

$$PM_{ajustado} = 10.51 \text{ meses / hombre}$$

$$PM_{ajustado} \approx 11 \text{ meses / hombre}$$

5.5. Cálculo del tiempo de desarrollo del software.

El tiempo requerido para el desarrollo del proyecto está dado por:

$$TDEV = C \times (PM_{ajustado})^F$$

$$TDEV = 3.64 \times 10.51^{0.26}$$

$$TDEV = 6.71$$

$$TDEV \approx 7$$

Donde se tienen como variables:

C: Constante con valor igual 3.64.

PM_{ajustado} = 10.51 meses/hombre.

F: Es un valor que se calcula de la siguiente forma:

$$F = D + 0.2 \times 0.01 \times \sum SF$$

$$F = 0.24 + 0.2 \times 0.01 \times 7.74$$

$$F \approx 0.26$$

Donde se tiene como variables:

D : Constante cuyo valor es 0.24.

SF : Valor de los factores de escala.

5.6. Cálculo del costo total del proyecto.

Para el cálculo del costo total correspondiente al proyecto en cuestión, COCOMO II propone:

$$C = CHM \times PM$$

$$C = 157 \times 10.51$$

$$C = 1650.07$$

$$C \approx 1650$$

Donde se tiene como variables:

C: Costo total.

CHM: Costo teniendo en cuenta el salario de todos los obreros, el cual se calcula de la siguiente forma:

$$CHM = CH \times sal$$

$$CHM = 1.57 \times 100.$$

$$CHM = 157$$

Donde se tiene como variables:

sal : Salario medio por cada trabajador estimado en 100.

CH : Cantidad de personas destinadas al proyecto:

Cálculo de la cantidad de personas destinadas al proyecto.

$$CH = PM / TDEV$$

$$CH = 10.51 / 6.71$$

$$CH = 1.57$$

$$CH \approx 2$$

Donde se tiene como variables:

$$PM = 10.51$$

$$TDEV = 6.71$$

5.7. Resultados.

A continuación son mostrados en una tabla los resultados obtenidos luego de efectuados todos los cálculos necesarios para determinar el costo y esfuerzo requeridos para el desarrollo de la aplicación SGET.

Tabla 63 Resultados.

Cálculo de.	Valor.
Esfuerzo.	11 meses/hombre.
Tiempo de desarrollo.	7 meses.
Cantidad de hombres.	2 hombres.
Salario medio.	100 pesos.
Costo.	1650 pesos.

5.8. Análisis de costo.

El Sistema de Gestión de Encuestas Telefónicas (SGET) no trae consigo grandes gastos en cuanto a costo monetario se refiere, pues al usarse gran cantidad de herramientas de software libre no se incurren en los gastos de tener que pagar las licencias de dichas herramientas, además solo se incluye el gasto del salario de los desarrolladores, por lo cual la implementación de dicho sistema es factible.

5.9. Conclusiones Parciales.

En el capítulo se realizó un análisis detallado de la factibilidad de la solución propuesta, llegando a la conclusión de que es una propuesta económica y viable para su desarrollo. Para un mayor entendimiento ver **Anexos 38, 39 y 40.**

Conclusiones Generales.

Llegado a este punto se puede decir que se le dio cumplimiento a todos los objetivos planteados para la construcción del sistema.

- ✓ Se logró evidenciar la ventaja que posee un sistema automatizado para la realización de las encuestas sobre la manera tradicional. Esta aplicación, basada en una PBX y utilizando una IVR para la interacción con las personas posibilita la realización de encuestas concurrentes a múltiples encuestados, trayendo consigo un gran beneficio económico y de mercado para las empresas, posibilitando el desarrollo del marketing y técnicas para aumentar la satisfacción del cliente.
- ✓ Después de un estudio e investigación de técnicas y herramientas actuales se seleccionaron las más adecuadas para darle solución al problema planteado, priorizando aquellas que fueran libres para no tener que invertir en licencias innecesariamente. Esto posibilitó la construcción de la aplicación SGET sin gasto por licencias de softwares.
- ✓ Se obtuvo una aplicación WEB capaz de gestionar encuestas telefónicas y mostrar sus resultados una vez realizadas.
- ✓ Se construyó una IVR encargada de moderar e interactuar con los encuestados en el proceso de la realización de las encuestas.

Después de integrar todas las funcionalidades implementadas se obtuvo un sistema automatizado que realiza encuestas telefónicas utilizando una IVR para la interacción con las personas. Además, posibilita consultar los resultados de las encuestas, que pueden ser vitales a la hora de tomar decisiones de acuerdo a la opinión de los encuestados.

Recomendaciones.

- ✓ Se pueden reutilizar las funcionalidades que ofrece la aplicación para la construcción de un sistema de votaciones telefónico, permitiendo que las personas disquen un número de servicio fijado para interactuar con la IVR y hacer su elección.
- ✓ Agregar una funcionalidad para importar listas de teléfonos separadas por categorías para la realización de encuestas personalizadas a un sector definido de la población.
- ✓ Poner a prueba el sistema con una gran cantidad de teléfonos, como la red pública de ETECSA, para probar el buen funcionamiento del sistema desarrollado, así como el manejo de un gran flujo de datos.