

**Universidad de las Ciencias Informáticas  
Facultad 2**



**Título: Sistema de Cobro Revertido de Asterisk.**

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

**Autor:** Alberto Aliaga González.

**Tutor(es):** Ing. Yordan Ernesto Estrada Rodríguez

**Co-tutor:** Gerardo Rodríguez Fernández.

**Ciudad de La Habana, 2011**

"..Milito en el grupo de los impacientes, y milito en el bando de los apurados, y de los que siempre presionan para que las cosas se hagan, y de los que muchas veces tratan de hacer más de lo que se puede..."

Fidel Castro Ruz



**DECLARACIÓN DE AUTORÍA**

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

Para que así conste firmo la presente a los \_\_\_\_ días del mes de \_\_\_\_\_ del año \_\_\_\_\_.

\_\_\_\_\_  
Alberto Aliaga González

\_\_\_\_\_  
Yordan Ernesto Estrada Rodríguez

**DATOS DE CONTACTO**

Ing. Yordan E. Estrada Rodríguez: Ingeniero en Ciencias Informáticas graduado en la Universidad en Ciencias Informáticas en el curso 2007-2008. Posee la categoría docente de instructor. Lleva 3 años de graduado.

A mi padre, por ser mi ejemplo y guía, mi mejor amigo y porque nunca faltó un consejo tuyo para alcanzar las metas de mi vida. A mi madre, por traerme a esta vida para regalarme su amor infinito y por sentirse orgullosa de mí. Aún no he encontrado las palabras que de verdad describan mi agradecimiento... para ustedes un beso.

A mi hermana, por consentirme y apoyarme.

A mis abuelos, principio de lo que he sido, soy y seré.

A toda mi familia por el apoyo que me ha brindado a lo largo de mi vida. A mis tíos Adela, Lisbet, Pedro, Yola, Roger, a mi hermano Denis por hacer de su hogar el mío en estos años fuera de casa.

A mi tutor Yordan por ser más que eso, mi amigo. Por todo su interés, apoyo en la realización de esta tesis que también es suya.

A mis compañeros Adila, Javier, Ricardo, Yasmín, Makay, el profe Alejandro, Duany, Portieles, mi cotutor Gerardo, Yeisell, Frank, Odaymis, Claudia y a todos los que me han ayudado y apoyado en la realización de este trabajo.

A ese trío de mosqueteros que viene junto a mí desde la niñez: Tony, Adrián y Yayo, porque no hizo falta la sangre para saber que en ellos tengo los hermanos varones que no tuve.

A los **Trigueños**, por todos los buenos ratos que pasamos en esta Universidad.

A mi novia Yulaini por apoyarme día a día en el desarrollo de mi trabajo.

A todos los que de una forma u otra me han ayudado para llegar hasta aquí, a todos ustedes muchas gracias.

*Alberto*

A la memoria de mi abuelo Juan

## **RESUMEN**

En la actualidad los servicios telefónicos brindados por las PBX (Private Branch Exchange o Central telefónica Secundaria Privada Automática) tienen un amplio alcance en el mercado mundial, generando su utilización un gran impacto en el mismo. Mediante su uso se hace posible la prestación de un gran número de servicios que ofrecen cuantiosas ganancias a las empresas que se desempeñan en este sector; tal es el caso del proyecto Comunicaciones Unificadas en la Universidad de las Ciencias Informáticas (UCI, en lo adelante). En la actualidad, este proyecto cuenta con una plataforma de comunicaciones unificadas, además de una serie de servicios básicos para una plataforma de este tipo, los cuales, sin embargo, no cubren la totalidad de los servicios que esta pudiese ofrecer, aumentando así las ganancias obtenidas por este proyecto. Teniendo en cuenta la situación anteriormente expuesta, se propone el desarrollo de un sistema que haga posible la realización de llamadas de cobro revertido (tipo de llamada donde el usuario al cual va dirigida la misma se adjudica el costo de la comunicación), con el fin de aumentar el número de servicios que posee dicha plataforma y aumentando paralelamente las prestaciones recibidas por parte de clientes, el cual fue desarrollado satisfactoriamente utilizando técnicas y herramientas afines con este propósito.

TABLA DE CONTENIDOS

<b>RESUMEN</b> .....	I
<b>INTRODUCCIÓN</b> .....	4
<b>CAPÍTULO I: FUNDAMENTACIÓN TEÓRICA</b> .....	7
Introducción .....	7
1.1 Características de los Sistemas de Cobro Revertido a nivel mundial. ....	7
1.2 Herramientas y metodología seleccionada.....	9
1.2.1 Plataforma de Comunicaciones utilizada: Elastix.....	9
1.3 Metodología de desarrollo. ....	11
1.4 Herramientas de desarrollo y lenguajes de programación. ....	16
1.5 Lenguajes de programación .....	18
1.5.1 PHP5 .....	18
1.5.2 Java Script.....	18
1.6 Sistema gestor de base de datos: PostgreSQL 8.3 .....	20
1.7 Wamp5.....	21
Conclusiones parciales.....	22
<b>CAPÍTULO II. EXPLORACIÓN Y PLANIFICACIÓN</b> .....	23
Introducción .....	23
2.1 Características .....	23
2.1.1 Objeto de automatización.....	23
2.1.2 Propuesta del Sistema .....	23
2.1.3 Requisitos No Funcionales del sistema.....	23
2.2 Exploración .....	24
2.2.1 Flujo de procesos del sistema propuesto.....	24
2.2.2 Personas relacionadas con el sistema.....	25
2.2.3 Historias de usuarios (HU).....	25
2.3 Planificación.....	31
2.3.1 Estimación de esfuerzo por Historias de Usuario. ....	31
2.3.2 Plan de Iteraciones. ....	32
2.3.3 Plan de duración de las iteraciones. ....	32
2.3.4 Plan de entregas. ....	33
Conclusiones parciales.....	33
<b>CAPÍTULO III: DISEÑO DEL SISTEMA</b> .....	34
Introducción .....	34

3.1 Arquitectura del sistema.....	34
3.1.1 Arquitectura cliente servidor.....	35
3.2 Estilos Arquitectónicos .....	37
3.2.1 Modelo Vista Controlador .....	37
3.3 Patrones de diseño.....	38
3.4 Tarjetas CRC.....	39
3.5 Diseño de Base de datos.....	40
3.5.1 Diagrama Entidad-Relación.....	41
Conclusiones parciales.....	41
<b>CAPÍTULO IV: IMPLEMENTACIÓN Y PRUEBAS .....</b>	<b>42</b>
Introducción .....	42
4.1 Implementación.....	42
4.1.1 Iteración 1 .....	42
4.1.2 Iteración 2 .....	44
4.1.3 Iteración 3 .....	46
4.2 Pruebas .....	47
4.2.1 Pruebas Unitarias. ....	48
4.2.2 Pruebas de Aceptación. ....	49
Conclusiones parciales.....	57
<b>CAPÍTULO V: ESTUDIO DE FACTIBILIDAD .....</b>	<b>58</b>
INTRODUCCIÓN .....	58
4.1 COCOMO II. ....	58
4.2 Características del Proyecto. ....	59
4.2.1 Entradas Externas.....	59
4.2.2 Salidas Externas .....	59
4.2.3 Consultas Externas.....	60
4.2.4 Archivos Lógicos Internos. ....	60
4.2.5 Archivos de Interfaz Externos.....	61
Puntos de Función Desajustados.....	61
4.3 Cálculos de instrucciones fuentes, esfuerzo, tiempo de desarrollo, cantidad de hombres y costo. ....	62
4.3.1 Cálculo de instrucciones fuentes.....	62
4.3.2 Cálculo de esfuerzo nominal. ....	63
Precedentes (PREC).....	63
Flexibilidad de Desarrollo (FLEX).....	63
Solución de riesgos (RESL).....	64
Madurez del proceso (PMAT). ....	64
4.4 Ajuste del esfuerzo nominal.....	65

4.4.1 Multiplicadores de esfuerzo .....	65
4.5 Cálculo del tiempo para el desarrollo del software.....	67
4.6 Cálculo del costo total del proyecto.....	68
4.7 Resultados.....	68
4.8 Análisis del costo .....	69
4.9 Conclusiones parciales.....	69
<b>CONCLUSIONES GENERALES .....</b>	<b>70</b>
<b>RECOMENDACIONES .....</b>	<b>71</b>
BILIOGRAFÍA CITADA .....	<b>¡ERROR! MARCADOR NO DEFINIDO.</b>
BIBLIOGRAFÍA CONSULTADA .....	73

## **INTRODUCCIÓN**

A través de la evolución de las Tecnologías de la Información y las Comunicaciones (TIC) han surgido innumerables cambios en beneficio de la sociedad, no existe duda alguna que las comunicaciones son un factor indispensable para lograr el éxito en cualquier ámbito. Las comunicaciones no son un recurso estático, esto significa que un pequeño incremento de sus medios puede generar una proporción significativa de valor agregado, en el momento de su uso o aplicación.

La inminente y constante necesidad de comunicación en la sociedad actual ha provocado un desarrollo vertiginoso de las telecomunicaciones y con ello un avance en la digitalización y la transmisión de voz sobre las redes de datos, lo cual está abriendo las puertas a un nuevo mercado en pleno auge de expansión y explotación: La telefonía IP (telefonía que reúne la transmisión de voz y de datos, lo que posibilita la utilización de las redes informáticas para efectuar llamadas telefónicas mediante el protocolo Ip).

Esta surge como alternativa de la telefonía tradicional, brindando nuevos servicios al cliente, conjuga dos mundos históricamente separados: la transmisión de voz y la de datos a través de la red Ethernet y ha asumido grandes avances sobre todo lo relacionado con la integración de diversas funcionalidades de convergencia dentro de una misma plataforma. Paralelo al desarrollo de la telefonía IP ha ido creciendo a nivel mundial, la tendencia del uso de plataformas telefónicas basadas en software libre. Cuba no está exenta de estos cambios, en la Universidad de las Ciencias informáticas el Proyecto Comunicaciones Unificadas, del centro Telemática tiene como objeto social la prestación de múltiples servicios de telecomunicaciones entre los que se encuentran la operación, instalación, explotación, y comercialización de una plataforma de comunicaciones unificadas (término comúnmente utilizado por los proveedores de las tecnologías de la información para denominar la integración de los servicios de telefonía, correo electrónico, correo de voz y fax) que incluye Asterisk como software como software que presenta las funcionalidades de una planta telefónica, entre estos servicios se encuentra el servicio de llamadas de cobro revertido, el cual permitirá realizar llamadas de cobro revertido sin necesidad de monedas, tarjetas o contratos. Con este sistema el valor de las llamadas se le atribuye al usuario al cual van destinadas si las mismas son aceptadas por él. Con este servicio los usuarios tendrían una mayor facilidad y flexibilidad

para el uso de la central telefónica, debido a que sería una opción de fácil acceso para este sin tener en cuenta sus opciones de pago disponibles en el momento de su uso.

Es lógico plantear que el desarrollo de un sistema de llamadas de cobro revertido es sólo un paso de avance en el proceso necesario para el cumplimiento de los objetivos del proyecto Comunicaciones unificadas, de esta forma, se logra utilizar la telefonía IP para brindar todos los servicios de una central telefónica tradicional, lograr la conectividad y la interoperabilidad entre la telefonía IP y la telefonía tradicional, además de reducir los gastos por concepto de importaciones que se realizan actualmente en la adquisición de una PBX.

Teniendo en cuenta la situación existente se define el siguiente problema a resolver: **¿Cómo brindar un servicio automatizado de cobro revertido a través Asterisk?**

Hacia la solución del problema planteado se dirige este trabajo de diploma; tomando como objeto de estudio: **los procesos de cobro revertido para sistemas telefónicos** y como campo de acción: **los procesos de cobro revertido en el proyecto de Comunicaciones Unificadas.**

Se traza como objetivo general: **Desarrollar un sistema de cobro revertido para el proyecto Comunicaciones Unificadas.**

Con el objetivo de guiar el curso de la investigación se propusieron las siguientes preguntas científicas:

- ¿Cómo utilizar y administrar la plataforma telefónica?
- ¿Cómo desarrollar una IVR (respuesta de voz interactiva)?

Luego se definen las siguientes tareas investigativas:

1. Realización un estudio del funcionamiento de la plataforma de comunicaciones unificadas utilizada.
2. Realización un estudio del funcionamiento del sistema de administración de la central telefónica.
3. Realización de un estudio del arte de los sistemas de llamadas de cobro revertido existentes.
4. Realización de un estudio para selección de las tecnologías y herramientas necesarias para el desarrollo de este tipo de servicio.
5. Realización de un estudio sobre el funcionamiento de las IVR.

Como posible resultado de este trabajo se espera desarrollar un Sistema de Cobro Revertido a través de Asterisk que sea integrable a la plataforma del proyecto Comunicaciones Unificadas.

El presente trabajo está estructurado en 5 capítulos, a continuación se muestra una breve descripción de cada uno de ellos:

**Capítulo 1:** Fundamentación Teórica, este capítulo comprende el estudio del estado del arte de los elementos fundamentales a tener en cuenta para la solución del problema planteado, se definen la metodología de desarrollo, lenguaje de programación así como el entorno de desarrollo.

**Capítulo 2:** Características del sistema, exploración y planificación, en este capítulo se realizó un análisis de los requisitos que debe satisfacer el sistema, la propuesta de la solución a implementar, además de las historias de usuarios, las cuales describen las funcionalidades del mismo.

**Capítulo 3:** Diseño del sistema. En este capítulo se definirán la arquitectura, los patrones de diseño, estilos arquitectónicos y características generales, además se mostrará un diseño de la base de datos a utilizar.

**Capítulo 4:** Implementación y prueba. En este capítulo se especificarán todas las tareas de la ingeniería definidas para cada historia de usuario así como las pruebas programadas para el sistema.

**Capítulo 5:** Estudio de Factibilidad. En este capítulo se realizará un análisis teniendo en cuenta las características del sistema y el equipo de desarrollo, para determinar cuan factible puede ser el mismo.

## **CAPÍTULO I: FUNDAMENTACIÓN TEÓRICA**

### **Introducción**

En este capítulo se exponen las características principales de los Sistemas de Cobro revertido existente a nivel mundial. Se analizan las metodologías, lenguajes y tecnologías utilizadas para el desarrollo del sistema de cobro revertido de acuerdo con las características de estos, brindando así una propuesta de solución.

### **1.1 Características de los Sistemas de Cobro Revertido a nivel mundial.**

Es importante comenzar explicando características fundamentales de los Centros de Atención de Llamadas (Call center) debido a que es aquí donde se brindan los servicios de cobro revertido. Un centro de atención de llamadas es un área donde las personas, realizan o reciben llamadas desde y/o hacia: clientes, socios comerciales, compañías asociadas u otros. Dichos centros son operados por una compañía de servicios que se encarga de administrar y proveer soporte, además de brindarle asistencia al consumidor según los productos, servicios o información requerida. En ellos se realizan llamadas en función de implementar la venta y los cobros de la empresa. La mayoría de las empresas usan este tipo de centros para interactuar con sus clientes, ya sean empresas de servicio público, firmas de pedidos por catálogo, atención al cliente y soportes operativos, con relación a empresas de software y hardware. Muchos comercios utilizan los Centros de Contacto incluso para el desarrollo de sus funciones internas, incluyen mesas de ayuda y soporte de ventas.

Paralelo al desarrollo de estos centros para la atención de llamadas de usuarios, en el mundo se han desarrollado las centrales telefónicas denominadas PBX. Estos no son más que cualquier central telefónica conectada directamente a la red pública de teléfono por medio de líneas troncales para gestionar, además de las llamadas internas, las entrantes y salientes con autonomía sobre cualquier otra central telefónica (1).

Entre los servicios prestados actualmente por las centrales telefónicas mencionadas anteriormente se encuentran la realización de llamadas, informes de autenticación y salidas de clientes, estadísticas completas del sistema, servicios de mensajería, grabaciones de llamadas y su monitoreo, así como interfaces web para buzones de voz. Además, cuentan con un servicio de llamadas de cobro revertido los

cuales se caracterizan por estar destinados principalmente a aquellos usuarios que deseen contactar con la empresa para obtener cualquier tipo de información sobre los soportes que la misma brinde, de esta forma el cliente llamaría a una extensión determinada donde sería atendido por una operadora que le brindaría la información solicitada y el saldo de la llamada se le adjudicaría a la empresa.

Es necesario mencionar que algunos autores definen por cobro revertido al servicio que permite a las empresas poner a disposición de sus clientes un número para llamar en forma gratuita, siendo el cargo a cuenta de la empresa (2). Entre las empresas que presentan un sistema con las características anteriormente expuestas podemos mencionar Frost&Sullivan y World Premium Rates.

Cuba con el objetivo de alcanzar un desarrollo superior en el mundo de las comunicaciones telefónicas, mantiene un esfuerzo constante en la búsqueda e incremento de los conocimientos y aplicaciones. La Empresa de Telecomunicaciones de Cuba S.A (ETECSA, en lo adelante), se dedica a la comercialización de múltiples productos y servicios en el área de las telecomunicaciones. Actualmente ETECSA cuenta con una alta responsabilidad en el desarrollo socio-económico del país y en la informatización de la sociedad cubana, garantizando la conectividad y el mantenimiento de las redes públicas de telecomunicaciones en todo el territorio. Sin embargo, este es uno de los sectores que presenta gran demanda de reclamaciones por parte de los clientes, además de no poseer un servicio que permita el enlace de la telefonía IP con las redes públicas a disposición de estos.

La Universidad de las Ciencias Informáticas incursiona en el campo de la telefonía IP brindando servicios a través de una plataforma de comunicaciones unificadas en el Centro de Telemática radicado en la Facultad #2 de la UCI, dicho centro no solo se dedica al desarrollo de aplicaciones relacionadas con las telecomunicaciones, sino que ofrece los servicios de montaje de una plataforma de comunicaciones unificadas en diversos proyectos dentro de la misma universidad. Entre los servicios que brinda esta plataforma se encuentra el servicio de telefonía y es posible identificar a Asterisk como software que presenta las funcionalidades de una planta telefónicas, además este proyecto tiene como meta desarrollar una serie de servicios existentes en la telefonía tradicional para esta PBX.

Teniendo en cuenta que hasta el momento de dicha investigación no fue detectado ningún sistema de cobro revertido con las funcionalidades especificadas por el cliente, fue necesario realizar un estudio para

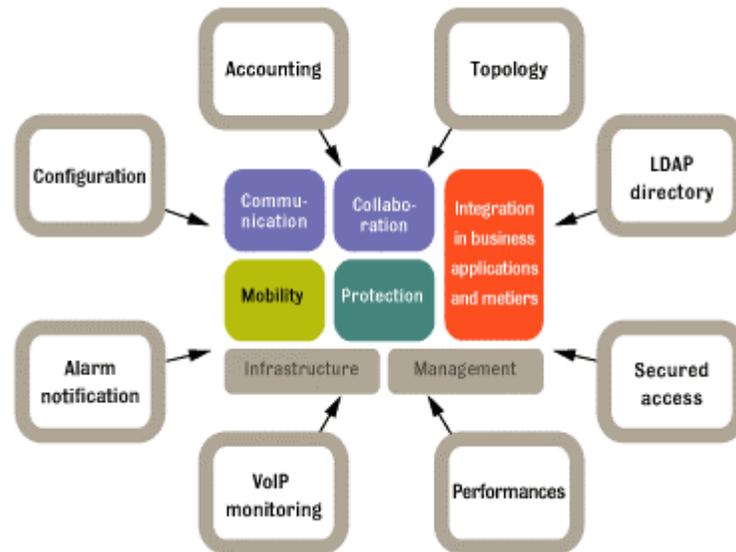
la selección de la metodología de desarrollo acorde para el desarrollo del sistema propuesto, así como una serie de herramientas para la implementación del mismo.

## **1.2 Herramientas y metodología seleccionada**

### **1.2.1 Plataforma de Comunicaciones utilizada: Elastix**

Elastix es una distribución de servidor Linux que presenta módulos para generar varios reportes, este es un software de código abierto bajo la Licencia Pública General Versión 2 para integrar servicios de mensajería instantánea, telefonía, correos de voz y fax, la cual integra varias herramientas disponibles para centrales telefónicas basadas en PBX. Comenzó con una interfaz de informe de llamadas de Asterisk y en la actualidad cuenta con varios medios de comunicación en una solución que se encuentra en constante desarrollo. Posee servicios de correo de voz, fax, soporte para softphones, interfaz de configuración web, interconexión de PBXs, respuesta de IVR configurables, identificación de llamantes y Gestión de relaciones (Customer Relationship Management o CRM). También cuenta con múltiples características y funcionalidades relacionadas con todos los servicios: telefonía IP, servidor de correo, servidor de fax, Conferencias, servidor de mensajería instantánea, entre otros. (3)

Teniendo en cuenta las características del sistema requerido y que dicha plataforma de comunicaciones es la utilizada en el proyecto Comunicaciones Unificadas, se decidió utilizar la herramienta Elastix para el desarrollo del presente trabajo.



**Figura 1** Plataforma de comunicaciones unificadas.

### 1.2.2 Software con características de una central telefónica: Asterisk.

Central telefónica originalmente integrada a la plataforma telefónica utilizada creada por **Mark Spencer** de **DIGIUM, Inc.**, cuyo código fuente está disponible para realizar modificaciones y mejoras. Esta central permite a los desarrolladores, construir nuevos sistemas telefónicos de forma eficiente o migrar de forma gradual los existentes a las nuevas tecnologías. Es un software para una central telefónica conectada directamente a la red pública de teléfono, por medio de líneas troncales, que permite gestionar: realizar, archivar y programar las llamadas internas, las entrantes y salientes con autonomía sobre cualquier otra central telefónica PBX, permite conectividad en tiempo real entre las redes PSTN y PBX, posee un entorno de desarrollo fácil debido a que puede ser programado en el lenguaje C con códigos primitivos o en cualquier otro lenguaje utilizando librerías de tipo AGI (Asterisk Gateway Interface), puede ser ejecutado sobre la plataforma Linux y es a código abierto, convierte un ordenador en un servidor de comunicaciones de voz, entre sus características está la de proveer servicios de buzones de voz con directorios, conferencias, respuesta de voz interactiva así como llamadas en espera. (3)

### **1.3 Metodología de desarrollo.**

En un proyecto de desarrollo de software, la metodología define “quién” debe hacer “qué”, “cuándo” y “cómo” debe hacerlo, para alcanzar un determinado objetivo. La finalidad de una metodología de desarrollo, es garantizar la eficacia y la eficiencia en el proceso de generación de software. En realidad no existe una metodología estándar, sino que las características de cada proyecto, las del equipo de desarrollo, recursos disponibles y tiempo para la elaboración de este exigen la flexibilidad del proceso, adaptándose el mismo al entorno y teniendo como objetivo alcanzar la máxima calidad en lo que se produce (4). Entre las metodologías usadas en la ingeniería se encuentran las metodologías pesadas o tradicionales y las metodologías ágiles o ligeras.

**Las metodologías tradicionales** ponen gran énfasis en la planificación y un uso exhaustivo de documentación, mediante una rigurosa definición de roles, actividades, artefactos. Por su enfoque tradicional no se adaptan a los cambios constantes, por lo que no son metodologías adecuadas para proyectos con un entorno de desarrollo cambiante. (5)

**Las metodologías ágiles** están orientadas a proyectos pequeños donde exista una gran incertidumbre con requisitos desconocidos o variables. El cliente es parte del proceso de desarrollo, lo que posibilita la retroalimentación constante y las respuestas rápidas a los cambios en el negocio debido a su gran capacidad de respuesta a los cambios (5).

En la siguiente tabla se pueden observar las características específicas de cada una de estas metodologías :

**Tabla 1 Comparación entre metodologías**

<b>Metodologías Ágiles</b>	<b>Metodologías Tradicionales</b>
Basada en heurística provenientes de prácticas de producción de código	Basadas en normas provenientes de estándares seguidos por el entorno de desarrollo
Especialmente preparados para cambios durante el proyecto	Cierta resistencia a los cambios
Impuestas internamente (por el equipo)	Impuestas externamente
Procesos menos controlados, con pocos principios	Proceso mucho más controlado, con numerosas políticas/normas.
No existe contrato tradicional o al menos es bastante flexible	Existe un contrato prefijado
El cliente es parte del equipo de desarrollo	El cliente interactúa con el equipo de desarrollo mediante reuniones.
Grupos pequeños (menos de 10 integrantes) y trabajando en el mismo sitio	Grupos grandes y posiblemente distribuidos
Pocos artefactos	Más artefactos
Pocos roles	Más roles
Menos énfasis en la arquitectura de software	La arquitectura de software es esencial y se expresa mediante modelos.

Después de identificar si es necesario utilizar una metodología específica para el desarrollo del producto final, se impone la tarea de decidir cuál de las metodologías existentes será la más adecuada o la mejor, cuál se corresponde más con el producto que se desea obtener, cuál está más acorde con las características del equipo de trabajo, del entorno, del tiempo máximo para la entrega, o sea; deben considerarse aspectos muy importantes antes de adoptar una determinada metodología de desarrollo. De

acuerdo con las características mencionadas, la investigación se centra en las metodologías ágiles, debido a que son capaces de entregar un producto funcional en un breve intervalo de tiempo y con la interacción del cliente. Para ello fue necesario la realización de un estudio sobre las características de las metodologías ágiles conocidas por los desarrolladores, teniendo en cuenta la experiencia con la que cuentan los mismos en su utilización.

### **XP (*Extreme Programming* ó Programación Extrema).**

XP es una metodología ágil que se centra en potenciar las relaciones entre los miembros del equipo de desarrollo como clave fundamental para el éxito, además se basa en la realimentación continua entre el cliente y el equipo de desarrollo, evita elevados e innecesarios costos ya que muchas veces se emplea demasiado tiempo y recursos en cambiar la documentación de la planificación para que se asemeje al código. Para esto, intenta implementar una forma de trabajo que se adapte fácilmente a las circunstancias aumentando considerablemente la productividad del equipo. Además, esta metodología se utiliza para pequeños grupos de trabajo de menos de 10 personas y donde se encuentra al menos un miembro de los clientes en directo intercambio con los desarrolladores.

Para suplir la falta de requisitos, casos de uso, y demás herramientas; XP utiliza historias de usuarios, la historia de usuario es una frase corta que representa alguna función que realizara el sistema. Cada historia de usuario no puede demorar en desarrollarse más de una semana, si así lo requiera, debe segmentarse. Esta metodología consta de seis fases:

- ✓ **Exploración:** los clientes plantean en las historias de usuarios lo que constituye de principal interés para la primera versión del producto.
- ✓ **Planificación de la Entrega (*Release*):** el cliente establece la prioridad de cada historia de usuario, datos que son usados por los programadores para realizar una estimación del esfuerzo necesario de cada una de ellas.
- ✓ **Iteraciones:** es la técnica de desarrollar y entregar componentes incrementales de funcionalidades de un negocio incluye varias iteraciones sobre el sistema antes de ser entregado. El plan de entrega está compuesto por iteraciones de no más de tres semanas.

- ✓ **Producción:** en esta fase se realizan pruebas adicionales y revisiones de rendimiento antes de que el sistema sea llevado al entorno del cliente.
- ✓ **Mantenimiento:** se plantean las tareas de soporte para el cliente. La fase de mantenimiento puede requerir adicionar personal al equipo y realizar cambios en su estructura.
- ✓ **Muerte del Proyecto:** esta etapa es de suma importancia pues es cuando el cliente no tiene más historias para ser incluidas en el sistema. Lo que requiere que se satisfagan las necesidades del cliente en otros aspectos como rendimiento y confiabilidad del sistema.

El riesgo de desarrollo es elevado debido al corto tiempo de entrega planteado y a los continuos cambios de requerimientos. XP está diseñada a mitigar los riesgos en proyectos con estas características (6).

#### **Ventajas:**

- ✓ Al trabajar como un equipo integrado, el código será de mayor calidad desde el mismo momento de crearlo, y tendrá menos fallos.
- ✓ Los programadores novatos aprenderán de los expertos al emparejarse con ellos.
- ✓ Si una pareja realiza un pedazo de código susceptible de ser reutilizado en el proyecto, hay dos programadores que lo saben y que lo reutilizarán cuando puedan (ya que saben cómo funciona), enseñándolo a sus nuevos compañeros. De esta manera, el conocimiento del código ya hecho se propaga de forma natural entre todos los programadores del equipo.
- ✓ El estilo de programación tiende a unificarse.
- ✓ Apropiado para entornos volátiles.
- ✓ Estar preparados para el cambio, significa reducir su coste.
- ✓ Planificación más transparente para los clientes, conocen las fechas de entrega de funcionalidades vitales para su negocio.

- ✓ Permite definir en cada iteración cuales son los objetivos de la siguiente.
- ✓ Permite tener realimentación de los usuarios.
- ✓ La presión está a lo largo de todo el proyecto y no en una entrega final.

**Desventajas:**

- ✓ Delimitar el alcance del proyecto con el cliente.

**Scrum**

Scrum, más que una metodología de desarrollo software, es una forma de auto-gestión de los equipos de programadores. Un grupo de programadores decide cómo hacer sus tareas y cuánto van a tardar en ello. Scrum ayuda a que trabajen todos juntos, en la misma dirección, con un objetivo claro, además permite seguir de forma clara el avance de las tareas a realizar, de forma que los "jefes" puedan ver día a día cómo progresa el trabajo.

Sin embargo, Scrum no indica qué se debe hacer para hacer el código. Debería, por tanto, complementarse con alguna otra metodología de desarrollo. Se lleva bien con las metodologías ágiles y en concreto, con la programación extrema. Es un proceso ágil y liviano que sirve para administrar y controlar el desarrollo de software (4). Este se basa en una serie de aspectos que lo hacen factible y fácil de utilizar. Entre ellos se tiene:

- ✓ Prioriza el trabajo en función del valor que tenga para el negocio, maximizando la utilidad de lo que se construye y el retorno de inversión.
- ✓ Está diseñado especialmente para adaptarse a los cambios en los requerimientos.
- ✓ Los requerimientos y las prioridades se revisan y ajustan durante el proyecto en intervalos muy cortos y regulares (de esta forma se puede adaptar en tiempo real el producto que se está construyendo a las necesidades del cliente).
- ✓ Se busca entregar software que realmente resuelva las necesidades, aumentando la satisfacción del cliente.
- ✓ Procura la construcción del software con calidad.

- ✓ Define las características que debe tener el producto a construir (qué construir, qué no y en qué orden) y remueve cualquier obstáculo que pudiera entorpecer la tarea del equipo de desarrollo.
- ✓ Busca equipos efectivos y productivos (dentro de las posibilidades).
- ✓ Se basa en los principios de inspección continua, adaptación, auto-gestión e innovación.
- ✓ Provee un ámbito propicio para desarrollar sus capacidades profesionales (lo que resulta en un incremento en la motivación de los integrantes del equipo).

### **Metodología seleccionada. Programación Extrema**

Luego del estudio de las metodologías anteriores se decide utilizar XP, teniendo en cuenta que esta metodología presenta características comunes con el sistema propuesto, las cuales se listan a continuación:

- ✓ Está sujeta a cambios continuos.
- ✓ El cliente forma parte del equipo de desarrollo.
- ✓ Está diseñada para proyectos de poca duración.
- ✓ Utilizable en equipos de desarrollo pequeños

A todo esto es posible agregar que en la metodología Scrum son los desarrolladores los que definen el tiempo que necesitan para desarrollar sus tareas y no cuentan con las prioridades de los clientes.

### **1.4 Herramientas de desarrollo y lenguajes de programación.**

El marco de trabajo adoptado para la implementación del sistema es CEDRUX. Este es el primer sistema integral de gestión cubano que sustituye la amplia gama de aplicaciones informáticas usadas para actividad empresarial en el país. Este sistema contiene un módulo de gestión de Capital Humano partiendo de que la única ventaja competitiva de las organizaciones radica precisamente en su capacidad de aprender y adaptarse a los cambios de los ambientes en los que se desarrollan. De aquí que, el Capital Humano es la base de la creación de valores para las organizaciones y no debe omitirse su inclusión en un sistema como este. Está diseñado para modelar y automatizar la mayoría de los procesos en una empresa, permiten unificar u ordenar toda la información de la empresa en un solo lugar. Dentro de las principales características de Cedrux se encuentra su independencia tecnológica. Funciona sobre

plataforma web y puede mantener interoperabilidad con otros sistemas. Es multimoneda, multientidades y como característica particular de Cuba permite hacer las operaciones sobre la doble moneda (peso cubano y peso convertible). Mantiene el control de fechas, es transaccional, con integridad funcional y posibilita el tratamiento estadístico del procesamiento de la información. El mismo ha sido definido por el proyecto ERP-Cuba y constituye un paquete tecnológico para la gestión de entidades (7). A continuación, se detallan las herramientas utilizadas en el desarrollo del sistema que permitirán el desarrollo del mismo. Se debe tener en cuenta que la selección de este marco de trabajo fue realizado después de un análisis de conjunto con el cliente debido a que el mismo solicitó el desarrollo de la aplicación web estableciendo previamente las herramientas presentadas.

#### **1.4.1 Zend Framework 6.0**

Zend Framework (ZF) es un framework de código abierto para desarrollar aplicaciones web y servicios web con PHP 5. ZF es una implementación que usa código 100% orientado a objetos. La estructura de los componentes de ZF es algo único; cada componente está construido con una baja dependencia de otros componentes. Esta arquitectura débilmente acoplada permite a los desarrolladores utilizar los componentes por separado. A menudo se refiere a este tipo de diseño como "use-at-will" (uso a voluntad). Aunque se pueden utilizar de forma individual, los componentes de la biblioteca estándar de Zend Framework conforman un potente y extensible framework de aplicaciones web al combinarse. ZF ofrece un gran rendimiento y una robusta implementación MVC, una abstracción de base de datos fácil de usar, y un componente de formularios que implementa la prestación de formularios HTML, validación y filtrado para que los desarrolladores puedan consolidar todas las operaciones usando de una manera sencilla la interfaz orientada a objetos. Otros componentes, como Zend\_Auth y Zend\_Acl, proveen autenticación de usuarios y autorización diferentes a las tiendas de certificados comunes. También existen componentes que implementan bibliotecas de cliente para acceder de forma sencilla a los servicios web más populares.

(8)

## 1.5 Lenguajes de programación

### 1.5.1 PHP5

PHP es un lenguaje de programación interpretado, diseñado originalmente para la creación de páginas web dinámicas. Es usado principalmente en interpretación del lado del servidor (server-side scripting) pero actualmente puede ser utilizado desde una interfaz de línea de comandos o en la creación de otros tipos de programas incluyendo aplicaciones con interfaz gráfica usando las bibliotecas Qt o GTK+ (9).

#### Ventajas:

- ✓ Es un lenguaje multiplataforma.
- ✓ Completamente orientado al desarrollo de aplicaciones web dinámicas con acceso a información almacenada en una base de datos.
- ✓ El código fuente escrito en PHP es invisible al navegador y al cliente ya que es el servidor el que se encarga de ejecutar el código y enviar su resultado HTML al navegador. Esto hace que la programación en PHP sea segura y confiable.
- ✓ Capacidad de conexión con la mayoría de los motores de base de datos que se utilizan en la actualidad, destaca su conectividad con MySQL y PostgreSQL.
- ✓ Capacidad de expandir su potencial utilizando la enorme cantidad de módulos (llamados ext's o extensiones).
- ✓ Es libre, por lo que se presenta como una alternativa de fácil acceso para todos.
- ✓ Permite aplicar técnicas de programación orientada a objetos.
- ✓ Biblioteca nativa de funciones sumamente amplia e incluida.
- ✓ No requiere definición de tipos de variables aunque sus variables se pueden evaluar también por el tipo que estén manejando en tiempo de ejecución.
- ✓ Tiene manejo de excepciones (desde PHP5).

### 1.5.2 Java Script

Es un lenguaje de programación interpretado, multiplataforma y parcialmente orientado a objetos, desarrollado para incrementar las funcionalidades del lenguaje HTML. Permite crear efectos especiales,

interactuar con el visitante y funciones básicas que son soportadas por la mayoría de los navegadores que se utilizan diariamente.

**Ventajas:**

- ✓ Tiempo de transferencia
- ✓ Permite elaborar aplicaciones Web que simulen características de aplicaciones escritorios.
- ✓ Es un código “interpretado” por el cliente.
- ✓ Es un código orientado a objetos.
- ✓ Es un código integrado a HTML.
- ✓ Reutilización de código de programación.
- ✓ El lenguaje de scripting es seguro y fiable.
- ✓ El código es visible y puede ser leído por cualquiera, incluso si está protegido con las leyes del copyright.

**Ext JS (Extend Java Script)**

Ext JS es una biblioteca de Java Script para el desarrollo de aplicaciones web interactivas usando tecnologías como AJAX, DHTML y DOM. Originalmente construida como una extensión de la biblioteca YUI, en la actualidad puede usarse como extensión para las bibliotecas jQuery y Prototype. Desde la versión 1.1 puede ejecutarse como una aplicación independiente.

**Funcionalidades**

- ✓ Dispone de un conjunto de componentes (widgets) para incluir dentro de una aplicación web, como:
  - ✓ Cuadros y áreas de texto.
  - ✓ Campos para fechas.
  - ✓ Campos numéricos.
  - ✓ Combos.
  - ✓ Radiobuttons y checkboxes.
  - ✓ Editor HTML.

- ✓ Elementos de datos (con modos de sólo lectura, datos ordenables, columnas que se pueden bloquear y arrastrar, etc.).
- ✓ Árbol de datos.
- ✓ Pestañas.
- ✓ Barra de herramientas.
- ✓ Menús al estilo de Windows.
- ✓ Paneles divisibles en secciones.
- ✓ Sliders.

Varios de estos componentes están capacitados para comunicarse con el servidor usando AJAX. También contiene numerosas funcionalidades que permiten añadir interactividad a las páginas HTML, como:

- ✓ Cuadros de diálogo.
- ✓ Alertas para mostrar mensajes de validación e información sobre campos individuales. (10)

### **1.6 Sistema gestor de base de datos: PostgreSQL 8.3**

PostgreSQL es un sistema de gestión de base de datos orientada a objetos y libre, el desarrollo de PostgreSQL no es manejado por una sola compañía, sino que es dirigido por una comunidad de desarrolladores y organizaciones comerciales que trabajan en su desarrollo, dicha comunidad es denominada el PGDG (PostgreSQL Global Development Group). Fue el pionero en muchos de los conceptos existentes en el sistema objeto-relacional actual, incluido más tarde en otros sistemas de gestión comerciales (9).

PostgreSQL es un sistema objeto-relacional, ya que incluye características de la orientación a objetos, como puede ser la herencia, tipos de datos, funciones, restricciones, disparadores, reglas e integridad transaccional. A pesar de esto, PostgreSQL no es un sistema de gestión de bases de datos puramente orientado a objetos.

#### **Ventajas:**

- ✓ Soporte de protocolo de comunicación encriptado por SSL.
- ✓ Extensiones para alta disponibilidad, nuevos tipos de índices, datos espaciales, minería de datos, etc.
- ✓ Incorpora una estructura de datos array.
- ✓ Incorpora funciones de diversa índole: manejo de fechas, geométricas, orientadas a operaciones con redes, etc.
- ✓ Permite la declaración de funciones propias, así como la definición de disparadores.
- ✓ Soporta el uso de índices, reglas y vistas.
- ✓ Incluye herencia entre tablas (aunque no entre objetos, ya que no existen), por lo que a este gestor de bases de datos se le incluye entre los gestores objeto-relacionales.
- ✓ Permite la gestión de diferentes usuarios, como también los permisos asignados a cada uno de ellos.

### **1.7 Wamp5**

WAMP5 es un instalador para Windows; este instala una completa solución WAMP en su computador.

WAMP5 configura Apache - MySQL- PHP5 automáticamente para que trabajen juntos, no importa donde los instale en su disco local.

Este software instala:

- ✓ Apache 2.x.x.
- ✓ PHP 5.x.x
- ✓ MySQL 5.x.x
- ✓ PHPmyadmin
- ✓ SQLitemanager
- ✓ Wampserver service manager

WAMP5 está compuesto por un administrador de servicios como icono en la bandeja del sistema. Este administrador le permite controlar completamente su servidor y proyectos locales. No está diseñado para servidor de producción, pero si como un entorno de desarrollo. Con WAMP5, podrá crear sus scripts localmente en su computador, probarlos y luego cargarlos en su servidor en producción. Se ha

configurado los servidores Apache y MySQL con los archivos de configuración para tener la plataforma más estándar. Usted puede adaptarlas a sus necesidades personales (9).

### **Conclusiones parciales**

En este capítulo se definieron los principales conceptos a tratar en el resto del documento. También se abordaron las características referentes a la metodología de desarrollo escogida y se realizó un estudio de las tecnologías existentes en el mundo actual. Finalmente se dieron a conocer cada una de las herramientas para el desarrollo del sistema con sus características.

## **CAPÍTULO II. EXPLORACIÓN Y PLANIFICACIÓN**

### **Introducción**

Este capítulo permitirá conocer el contexto en el que se expone la solución propuesta. Se describirán los procesos existentes con el objetivo de comprenderlos y se especificarán aquellos que soportará el sistema. También guiará el desarrollo del software hacia el sistema adecuado, teniendo en cuenta las características que debe presentar el mismo. Se hace alusión a la fase de exploración y planificación propias de la metodología de desarrollo utilizada para la implementación del sistema que se propone.

### **2.1 Características**

#### **2.1.1 Objeto de automatización**

Se pretende automatizar el Sistema de cobro revertido en el Proyecto Comunicaciones Unificadas del Centro de Telemática. Mediante este sistema se podrán realizar llamadas de cobro revertido, permitiéndole al encargado de administrar la central telefónica tener una interfaz Web con los datos de las personas que utilizan dicho servicio.

#### **2.1.2 Propuesta del Sistema**

La solución que se propone estará diseñada para la puesta en marcha de un sistema que permita la realización de llamadas de cobro revertido en una central telefónica PBX. El sistema insertará la información de estas llamadas en una base de datos en la cual se guardarán para la facturación de las mismas. De esta base de datos se obtendrán los campos para que el personal encargado de la administración de la central telefónica tenga una interfaz que posibilite mostrar los datos de las llamadas realizadas por este sistema y los clientes que hacen uso de este servicio.

#### **2.1.3 Requisitos No Funcionales del sistema**

Los requisitos no funcionales se definen como las cualidades o propiedades que el producto debe tener, estos representan las características del producto.

### **Usabilidad**

RNF1. El servicio telefónico de cobro revertido solo será usado por los clientes de la central telefónica y el sitio de Reportes solo por el administrador de la central telefónica.

### **Interfaz**

RNF2. La aplicación debe ser amigable y muy profesional permitiendo una interacción amena, presentando los colores asociados al logo del Centro de Telemática (azul, blanco).

### **Confidencialidad**

RNF3. La información debe estar protegida contra accesos no autorizados a través de la gestión de roles.

RNF4. La información debe ser publicada por una persona con los privilegios asignados, o sea, el administrador.

### **Soporte**

RNF5. La aplicación debe ejecutarse sobre los sistemas operativos Windows y Linux.

RNF6. Debe ofrecer la posibilidad de un fácil mantenimiento permitiendo al mismo tiempo que el administrador de la red (autenticado) tenga acceso a la información que almacena el sistema.

RNF7. Se hallará implementado sobre tecnología Web lo que proporcionará su uso mediante el protocolo http.

### **Seguridad**

RNF8. Asegurar que el usuario sea autenticado antes de realizar cualquier operación sobre la aplicación.

RNF9. El sistema deberá estar disponible brindando una posibilidad de uso de 24 horas.

RNF10. El sistema solo puede ser modificado o actualizado por el administrador del mismo.

### **Software**

RNF11. La aplicación deberá ejecutarse en una estación de trabajo que conste con lo siguiente:

Navegador Web Mozilla Firefox.

## **2.2 Exploración**

### **2.2.1 Flujo de procesos del sistema propuesto.**

El Usuario accede al sistema principal marcando el código del servicio: el sistema le dará la bienvenida y requerirá los datos necesarios para establecer la comunicación, de no haber introducido datos correctos el

sistema le brindara 3 intentos para la corrección de los mismos, una vez capturados estos datos de forma correcta se establecerá una comunicación con la persona que recibirá la llamada, poniendo en conocimiento a esta ultima de la persona que requiere el servicio. De ser aceptada la llamada ambas personas establecerán comunicación y al terminar la misma los datos serán registrados en la base de datos para proceder a la facturación de la llamada. En caso que el usuario que solicita el servicio consuma todos los intentos posibles o que el usuario que recibirá la llamada no acepte la llamada, la comunicación usuario-sistema será terminada. Por otra parte el sistema brindara una interfaz web con los datos de estas llamadas a los administradores.

### **2.2.2 Personas relacionadas con el sistema.**

Se define como persona relacionada al sistema toda aquella que obtiene un resultado del valor de uno o varios procesos que se ejecutan en el mismo.

**Tabla 2 Personas relacionadas con el sistema.**

Personas Relacionada con el Sistema.	Justificación.
Usuario.	Es la persona que utiliza el sistema con autorización y se beneficia del servicio que brinda.
Administrador.	Es la persona encargada de chequear los reportes del sistema y de administrar todo lo referente a la central telefónica.

### **2.2.3 Historias de usuarios (HU).**

La identificación de las historias de usuario (HU) es la técnica utilizada para especificar los requisitos del software. Se trata de tablas en las cuales el cliente describe brevemente las características que el sistema debe poseer, sean requisitos funcionales o no funcionales. El tratamiento de las historias de usuario es muy dinámico y flexible. Cada historia de usuario debe ser lo suficientemente comprensible y delimitada para que los programadores puedan implementarla en unas semanas.

Teniendo en cuenta la prioridad en el negocio pueden ser clasificadas en:

**Alta:** Se le otorga a las Historias de Usuario que resultan funcionalidades fundamentales en el desarrollo del sistema, a las que el cliente define como principales para el desarrollo del mismo.

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

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

Teniendo en cuenta el riesgo de desarrollo pueden ser clasificadas en:

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

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

**Baja:** Cuando pueden aparecer errores que serán tratados con relativa facilidad sin que traigan prejuicios para el desarrollo del proyecto

A continuación se muestran las historias de usuarios que serán utilizadas en el desarrollo de la aplicación propuesta y que muestran la información brindada por el cliente.

**Tabla 3 Historia de usuario Reproducir Grabación.**

<b>Historia de Usuario</b>	
<b>Número:</b> 1	<b>Usuario:</b> Cliente
<b>Nombre historia:</b> Reproducir Grabación	
<b>Prioridad en negocio:</b> Alta	<b>Riesgo en desarrollo:</b> Bajo
<b>Puntos estimados:</b> 3/5	<b>Iteración asignada:</b> 1
<b>Programador responsable:</b> Alberto Aliaga González	

<p><b>Descripción:</b> El usuario tendrá contacto con el servicio al introducir el código *33 y el sistema reproducirá el mensaje correspondiente. El mensaje puede ser:</p> <ul style="list-style-type: none"> <li>✓ de bienvenida al sistema.</li> <li>✓ de solicitud de algún dato.</li> <li>✓ de confirmación de un dato introducido por un usuario.</li> </ul>
<p><b>Observaciones:</b></p>

**Tabla 4 Historia de usuario Grabar nombre.**

<b>Historia de Usuario</b>	
<b>Número:</b> 2	<b>Usuario:</b> Cliente
<b>Nombre historia:</b> Grabar nombre.	
<b>Prioridad en negocio:</b> Media	<b>Riesgo en desarrollo:</b> Medio
<b>Puntos estimados:</b> 1	<b>Iteración asignada:</b> 1
<b>Programador responsable:</b> Alberto Aliaga González.	
<b>Descripción:</b> El sistema solicitara al usuario decir su nombre en un plazo de 5 segundos y en este periodo grabara la voz del mismo.	
<b>Observaciones:</b>	

**Tabla 5 Historia de usuario Confirmar Número.**

<b>Historia de Usuario</b>	
<b>Número:</b> 3	<b>Usuario:</b> Cliente

<b>Nombre historia:</b> Confirmar Datos.	
<b>Prioridad en negocio:</b> Alta	<b>Riesgo en desarrollo:</b> Medio
<b>Puntos estimados:</b> 1	<b>Iteración asignada:</b> 1
<b>Programador responsable:</b> Alberto Aliaga González.	
<p><b>Descripción:</b> El sistema solicitará al usuario la confirmación de los datos, en caso de estar incorrecto el usuario tendrá 3 intentos para volver a introducirlo</p> <p>Los datos solicitados por el sistema son:</p> <ul style="list-style-type: none"> <li>✓ Nombre del usuario que solicita el servicio.</li> <li>✓ Numero al cual va dirigida la llamada.</li> <li>✓ Permiso de aceptación de la llamada.</li> </ul>	
<b>Observaciones:</b>	

**Tabla 6 Historia de usuario Establecer comunicación con el cliente número 2**

Historia de Usuario	
<b>Número:</b> 4	<b>Usuario:</b> Sistema
<b>Nombre historia:</b> Establecer comunicación con el cliente número 2.	
<b>Prioridad en negocio:</b> alto	<b>Riesgo en desarrollo:</b> Alto
<b>Puntos estimados:</b> 2	<b>Iteración asignada:</b> 2
<b>Programador responsable:</b> Alberto Aliaga González	
<p><b>Descripción:</b> El sistema establecerá comunicación con el usuario al que está destinada la llamada, reproduciéndole un mensaje para que esta sea aceptada o no.</p>	
<b>Observaciones:</b>	

**Tabla 7 Historia de usuario Establecer comunicación entre ambos usuarios.**

Historia de Usuario
---------------------

<b>Número:</b> 5	<b>Usuario:</b> Sistema
<b>Nombre historia:</b> Establecer comunicación entre ambos usuarios.	
<b>Prioridad en negocio:</b> Alta	<b>Riesgo en desarrollo:</b> Alta
<b>Puntos estimados:</b> 1	<b>Iteración asignada:</b> 2
<b>Programador responsable:</b> Alberto Aliaga González.	
<b>Descripción:</b> Si el usuario al que está destinada la llamada acepta la misma el sistema establecerá comunicación entre ambos usuarios.	
<b>Observaciones:</b>	

**Tabla 8 Historia de usuario Finalizar comunicación.**

Historia de Usuario	
<b>Número:</b> 6	<b>Usuario:</b> Sistema
<b>Nombre historia:</b> Finalizar comunicación	
<b>Prioridad en negocio:</b> Medio	<b>Riesgo en desarrollo:</b> Medio
<b>Puntos estimados:</b> 1/5	<b>Iteración asignada:</b> 2
<b>Programador responsable:</b> Alberto Aliaga González.	
<b>Descripción:</b> Una vez que uno de los dos usuarios haya colgado el teléfono el sistema culminara la comunicación entre ambos usuarios.	
<b>Observaciones:</b>	

**Tabla 9 Historia de usuario Revertir cobro.**

Historia de Usuario	
<b>Número:</b> 7	<b>Usuario:</b> Sistema

<b>Nombre historia:</b> Revertir cobro	
<b>Prioridad en negocio:</b> Alta	<b>Riesgo en desarrollo:</b> Alto
<b>Puntos estimados:</b> 2	<b>Iteración asignada:</b> 3
<b>Programador responsable:</b> Alberto Aliaga González.	
<b>Descripción:</b> Cuando la llamada haya finalizado, el sistema debe de insertar los datos de la misma en la base de datos de forma tal que este sea capaz de identificar las llamadas de cobro revertido para que posteriormente el servicio de facturación atribuya el costo de la llamada al usuario final.	
<b>Observaciones:</b>	

**Tabla 10** Mostrar reportes

Historia de Usuario	
<b>Número:</b> 7	<b>Usuario:</b> administrador
<b>Nombre historia:</b> Mostrar reportes.	
<b>Prioridad en negocio:</b> Medio	<b>Riesgo en desarrollo:</b> Alto
<b>Puntos estimados:</b> 2	<b>Iteración asignada:</b> 3
<b>Programador responsable:</b> Alberto Aliaga González.	
<b>Descripción:</b> La interfaz web del sistema debe mostrar los datos de las llamadas realizadas a través del servicio de código revertido.	
<b>Observaciones:</b>	

### 2.3 Planificación

La metodología XP propone que estimar la duración requerida para implementar las funcionalidades deseadas por el cliente, de informar sobre las consecuencias de determinadas decisiones, organizar la cultura de trabajo solo puede lograrse de forma exitosa con una planificación detallada dentro de cada versión.

Durante la fase de planificación se realiza una estimación del esfuerzo que costará implementar cada historia de usuario, pudiendo utilizarse cualquier criterio para medir el desarrollo del proyecto. La estimación incluye todo el esfuerzo asociado a la implementación de la historia de usuario.

Se debe tener en cuenta que el equipo de desarrollo define una semana de trabajo como la semana de cuarenta (40) horas de trabajo de lunes a viernes.

#### 2.3.1 Estimación de esfuerzo por Historias de Usuario.

Para el desarrollo del sistema propuesto, se realizó una estimación de esfuerzo para cada una de las historias de usuario identificadas.

**Tabla 11 Estimación de esfuerzo por HU.**

Historias de Usuario.	Puntos de Estimación.
Reproducir Grabación	3/5
Grabar nombre	2
Confirmar Número.	2/5
Establecer comunicación con el cliente número 2	2
Establecer comunicación entre ambos usuarios	2
Finalizar comunicación	1/5
Revertir cobro	2

Mostrar reportes	2
------------------	---

**2.3.2 Plan de Iteraciones.**

Después de identificar las historias de usuario del sistema y estimado el esfuerzo dedicado a la realización de cada una de estas se procede a la planificación de la etapa de implementación del proyecto. En base a lo antes planteado se decide realizar esta en tres iteraciones, las cuales se detallan a continuación.

**Iteración 1.**

En esta iteración se implementarán las historias de usuario que tienen están relacionadas con la introducción del usuario en el sistema. Al final de esta se contará con una primera versión de prueba, la cual será mostrada al cliente con el objetivo de obtener una retroalimentación para el grupo de trabajo.

**Iteración 2.**

El objetivo de esta es la implementación de las historias de usuario donde los usuarios logran la comunicación a través del servicio. Al finalizar se contará con una versión de prueba que será mostrada al cliente con el objetivo de realizar cambios necesarios en base a la opinión del mismo.

**Iteración 3.**

Durante el transcurso de esta se implementaron las historias de usuario que terminaran con los objetivos propuestos para el sistema, revertir el cobro de la llamada. Al finalizar la misma se constará de la versión 1.0 del producto final y como resultado de esta, el sistema será puesto en funcionamiento durante un periodo de tiempo para evaluar su desempeño.

**2.3.3 Plan de duración de las iteraciones.**

A continuación se presenta el Plan de duración de iteraciones. Este plan tiene como finalidad mostrar la duración de cada iteración, así como el orden en que serán implementadas las historias de usuario en cada una de las mismas.

**Tabla 12 Plan de duración de iteraciones**

Iteraciones.	Orden de las Historias de Usuarios a Implementar.	Duración de las Iteraciones. (Semanas)
--------------	---	--

Iteración 1.	Reproducir Grabación Grabar nombre. Confirmar Datos.	3
Iteración 2.	Establecer comunicación con el cliente número 2. Establecer comunicación entre ambos usuarios. Finalizar comunicación	3
Iteración 3.	Revertir cobro Mostrar reportes	4

### 2.3.4 Plan de entregas.

**Tabla 13 Plan de entregas**

Sistema.	Final de Iteración 1 28/03/11	Final de Iteración 2 27/04/11	Final de Iteración 3 12/05/11
Sistema de Cobro Revertido	0.1	0.2	1.0

### Conclusiones parciales

En el presente capítulo se abordaron los puntos referentes a las fases de exploración y planificación y características del sistema a desarrollar. La descripción de las Historias de Usuario, como paso fundamental en el ciclo de desarrollo, ha contribuido a la identificación de los requerimientos del sistema, los cuales sirvieron de base para la generación de una serie de artefactos que serán necesarios en la siguiente fase. Con la obtención del plan de entregas y la planificación se ha logrado delimitar el ciclo de desarrollo del sistema.

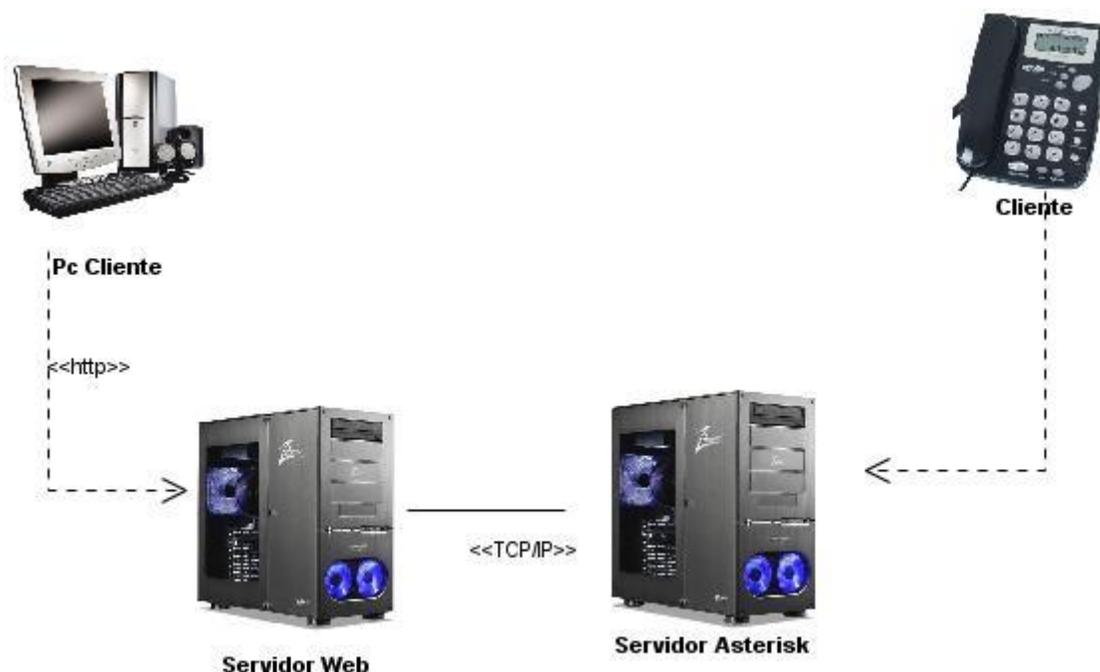
## **CAPÍTULO III: DISEÑO DEL SISTEMA**

### **Introducción**

En este capítulo se define la arquitectura del sistema, así como los estilos arquitectónicos, patrones de diseño utilizados en el desarrollo del mismo. También se mencionan las clases del sistema y las relaciones entre ellas, incluyendo el diagrama entidad-relación de la base de datos utilizada.

### **3.1 Arquitectura del sistema**

La arquitectura de un software se basa en una serie de patrones y abstracciones coherentes que proporcionan el marco de referencia necesario para guiar la construcción del software. En el caso del sistema propuesto el cliente es el responsable de acceder al sistema de cobro revertido, una vez finalizada la llamada se elabora un registro con los resultados, los cuales más tarde se expondrán en la web, donde cada usuario podrá acceder a ellos. Tanto la aplicación web como el servidor Asterisk interactúan con el sistema gestor de base de datos, el cual permitirá leer y modificar los datos registrados de la interacción con el cliente.



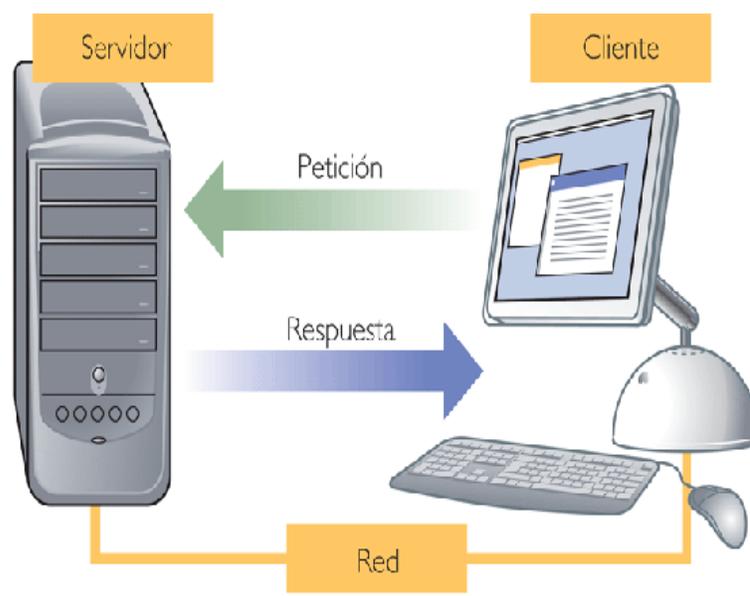
**Figura 2 Arquitectura del sistema propuesto**

### 3.1.1 Arquitectura cliente servidor.

Se decidió utilizar la arquitectura Cliente-Servidor para la realización del sistema debido a que tanto los clientes como el administrador del sistema pueden accederlo 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 debidas 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 (11).

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. Este tipo de red puede utilizarse conjuntamente en caso de que se esté utilizando en una red mixta.



**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 para la arquitectura cliente-servidor debido que a mayor cantidad de clientes que envíen peticiones al servidor mayor congestión tendrá este.
- ✓ El software y el hardware de un servidor son generalmente muy determinantes. Normalmente se necesita software y hardware específico, sobre todo en el lado del servidor, para satisfacer las demandas.

**3.2 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.

**3.2.1 Modelo Vista Controlador**

Para el desarrollo de la aplicación web 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.

**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. Este modelo de arquitectura presenta varias ventajas:

- ✓ Hay una clara separación entre los componentes de un programa; lo cual nos permite implementarlos por separado.
- ✓ La conexión entre el Modelo y sus Vistas es dinámica; se produce en tiempo de ejecución, no en tiempo de compilación.

### 3.3 Patrones de diseño

Un patrón es la descripción de un problema y su solución, que recibe un nombre y que puede emplearse en otros contextos. En teoría, indica la manera de utilizarlo en diversas situaciones. Los patrones no representan ideas ni principios nuevos de la ingeniería de software, por el contrario, intentan codificar el conocimiento y los principios ya existentes. El diseño fue elaborado siguiendo patrones basados en la experiencia, que de manera general constituyen soluciones simples y elegantes a problemas específicos y comunes del diseño orientado a objetos. En este caso se emplearon los patrones GRASP (en inglés General Responsibility Assignment Software Patterns), los que describen los principios fundamentales de la asignación de responsabilidades a objetos, expresados en forma de patrones (12). Los patrones GRASP que se utilizaron son los siguientes:

**Experto:** Dicho patrón es evidenciado en la definición de las clases de acuerdo con las funcionalidades que deben realizar a partir de la información manejada dentro del sistema, como por ejemplo las clases controladoras y las del modelo.

**Alta cohesión:** Este patrón fue utilizado en el diseño del sistema de manera general; donde se agruparon las clases en dependencia de las características a los que se les debía dar respuesta, según la premisa de que cada clase debe implementar las operaciones que estén sobre la misma área funcional.

**Controlador:** Las clase controladora definida: Listareportescontrol es un ejemplo de la aplicación de este patrón, las mismas tendrán a cargo la responsabilidad de manejar los eventos dentro del componente.

Durante el diseño del sistema se emplearon patrones GOF (Gang of four), específicamente:

**Fachada:** La aplicación de este patrón en el sitio se evidencia en la interfaz de servicios simple que es utilizada para gestionar todos los procesos.

**Cadena de Responsabilidad:** Está concebido que ante la ocurrencia de un error al realizarse una determinada consulta a la base de datos el mismo sea manejado por el Modelo, creando una nueva excepción. Dicha excepción debe ser propagada al Controlador, el cual será el encargado de capturarla y enviarla a la Vista ya traducida, esta última por su parte mostrará un mensaje al usuario en un lenguaje entendible notificando el error y sin especificar detalles del mismo. De esta manera se distribuyen las responsabilidades entre las diferentes componentes, evidenciándose por lo tanto el empleo de este patrón.

La Arquitectura Base y el diseño flexible a través del correcto uso de patrones de diseño en la generación de los artefactos necesarios para el desarrollo, posibilitaron crear una entrada apropiada como punto de partida a las actividades de implementación, con la máxima de lograr una mayor calidad del producto y la satisfacción del cliente.

### 3.4 Tarjetas CRC

Las tarjetas CRC (clase, responsabilidad y colaboración) son una técnica para el diseño de software orientado por objetos que determina el comportamiento de cada actividad, en este caso como el servicio web muestra los reportes guardados en la base de datos, la tarjeta CRC queda de la siguiente forma:

**Tabla 14 Tarjeta CRC Listareportecontrol**

Clase: Listareportescontrol	
Responsabilidad	Colaboración
Cargar Información de reportes	Reportes

**Tabla 15 Tarjeta CRC Conexión**

Clase: Conexión	
Responsabilidad	Colaboración
Realizar consultas	CDR

**Tabla 16 Tarjeta CRC Cobrorevertido**

Clase: Cobrorevertido	
Responsabilidad	Colaboración
Respuesta de voz interactiva para el cliente 1.	AGI

**Tabla 17 Tarjeta CRC Cobrorevertido2**

Clase: Cobrorevertido2	
Responsabilidad	Colaboración
Respuesta de voz interactiva para el cliente 2.	AGI

### 3.5 Diseño de Base de datos

En la propuesta de solución brindada el trabajo con la base de datos es esencial, el equipo de desarrollo por las características del sistema decidió trabajar con una de las bases de datos que posee Asterisk, la base de datos asteriskcdrdb, esta base de datos guarda los reportes de las llamadas realizadas en la central telefónica. En la propuesta de solución del sistema los desarrolladores identifican los datos de las llamadas realizadas mediante el servicio para lograr revertir su cobro, luego para lograr mostrar los datos de esta llamada se identifican de la siguiente forma: en una llamada realizada a través del servicio los campos src y channel contienen las extensiones que se comunicaron, si estos campos son diferentes, el valor del campo src es igual al valor contenido en el campo dstchannel y el campo lastapp tiene que

contener el valor Dial, para finalizar el valor del campo disposition debe ser ANSWERED. Si todas estas condiciones se cumplen, es posible identificar a este registro como una llamada de cobro revertido.

### 3.5.1 Diagrama Entidad-Relación

Un diagrama entidad-relación es una herramienta para el modelado de datos de un sistema de información, expresando entidades relevantes para este así como las relaciones entre sí. En el caso del presente sistema, se cuenta con una única tabla en la base de datos como se muestra a continuación:

asteriskcdrdb		
channel	varchar(80)	Nullable = false
calldate	date	Nullable = false
clid	varchar(80)	Nullable = false
src	varchar(80)	Nullable = false
dst	varchar(80)	Nullable = false
dcontext	varchar(80)	Nullable = false
dstchannel	varchar(80)	Nullable = false
lastapp	varchar(80)	Nullable = false
lastdata	varchar(80)	Nullable = false
duration	integer(11)	Nullable = false
billsec	integer(11)	Nullable = false
disposition	varchar(45)	Nullable = false
amaflags	integer(11)	Nullable = false
accountcode	varchar(20)	Nullable = false
uniqueid	varchar(32)	Nullable = false
userfield	varchar(255)	Nullable = false

**Figura 4 Diagrama entidad-relación CDR.**

### Conclusiones parciales

En el presente capítulo se realizó una breve descripción del diseño del sistema, se detallaron las características arquitectónicas que presentara el sistema dando una breve explicación acerca de cómo se utilizan los mismos. Además se explico la lógica del trabajo con la base de datos.

## **CAPÍTULO IV: IMPLEMENTACIÓN Y PRUEBAS**

### **Introducción**

En el presente capítulo se describen las fases de implementación y pruebas propias de la metodología de desarrollo XP. Esta metodología plantea que la implementación de un software debe realizarse de forma iterativa, obteniendo al final de cada iteración un producto funcional que debe ser probado y mostrado al cliente para retroalimentar a los desarrolladores con la opinión de este. También se detallan las tres iteraciones realizadas durante la etapa de codificación del proyecto, así como las tareas de programación desarrolladas en cada una de estas y se hace referencia a las pruebas realizadas sobre el sistema.

### **4.1 Implementación**

Durante el inicio de cada iteración, se lleva a cabo una revisión del plan de iteraciones y se modifica de ser necesario. Todo el trabajo de la iteración es expresado en tareas de programación, cada una de ellas es asignada a un programador como responsable. Estas tareas, pueden escribirse utilizando un lenguaje técnico y no necesariamente deben ser entendibles para el cliente.

A continuación se detalla cada una de las iteraciones del sistema, desarrolladas según la planificación propuesta.

#### **4.1.1 Iteración 1**

Durante el transcurso de esta iteración se implementaron las historias de usuario de que dan la bienvenida al usuario al servicio y con las cuales se capturan todos los datos del mismo.

**Tabla 18 Historias de Usuario desarrolladas en la primera iteración.**

<b>Historias de Usuario</b>	<b>Estimación (Días)</b>	<b>Real (Días)</b>
Reproducir grabación	2/5	2/5
Grabar nombre	2	2
Confirmar datos	3/5	3/5
<b>Total</b>	<b>3 semanas</b>	<b>3 semanas</b>

Tareas de las historias de usuario abordadas en la iteración.

**Tabla 19 Tarea 1 de la historia de usuario Reproducir Grabación.**

Tarea de Ingeniería	
<b>Número Tarea:</b> 1.	<b>Número Historia de Usuario:</b> 1.
<b>Nombre Tarea:</b> Implementar funcionalidad Reproducir grabación	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos Estimados:</b> 2/7
<b>Fecha Inicio:</b> 8/03/11	<b>Fecha Fin:</b> 9/03/11
<b>Programador Responsable:</b> Alberto Aliaga González	
<b>Descripción:</b> Se implementa la funcionalidad Reproducir grabación, permitiendo que el usuario escuche mensajes de bienvenida y reciba instrucciones sobre los pasos para trabajar con el servicio.	

**Tabla 20 Tarea 1 de la historia de usuario Grabar Nombre.**

Tarea de Ingeniería	
<b>Número Tarea:</b> 1.	<b>Número Historia de Usuario:</b> 2.
<b>Nombre Tarea:</b> Implementar la funcionalidad que permita Grabar el nombre del usuario.	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos Estimados:</b> 2
<b>Fecha Inicio:</b> 10/03/11	<b>Fecha Fin:</b> 23/03/11.
<b>Programador Responsable:</b> Alberto Aliaga González	
<b>Descripción:</b> Se implementa la funcionalidad Grabar nombre, la cual permite grabar el nombre del usuario que solicita el servicio una vez que este último le indica el tiempo que tiene para realizar esta acción.	

**Tabla 21 Tarea 1 de la historia de usuario Confirmar Datos.**

Tarea de Ingeniería	
<b>Número Tarea:</b> 1.	<b>Número Historia de Usuario:</b> 3.
<b>Nombre Tarea:</b> Implementar la funcionalidad que permita Confirmar los datos del usuario.	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos Estimados:</b> 2
<b>Fecha Inicio:</b> 24/03/11	<b>Fecha Fin:</b> 28/03/11.
<b>Programador Responsable:</b> Alberto Aliaga González	
<b>Descripción:</b> Se implementa la funcionalidad confirmar datos, la cual permite una vez que el usuario introduce los datos necesarios para utilizar el servicio, estos datos de ser incorrectos el usuario tendrá 3 intentos para volver a introducir cada uno de los mismos, de lo contrario el sistema automáticamente lo sacará del mismo	

#### 4.1.2 Iteración 2

**Tabla 22 Historias de Usuario desarrolladas en la segunda iteración.**

Historias de Usuario	Estimación (Días)	Real (Días)
Establecer comunicación con el cliente número 2.	10	10
Establecer comunicación entre ambos usuarios.	10	10
Finalizar comunicación.	1	1
<b>Total</b>	<b>4 semanas</b>	<b>4 semanas</b>

Tareas de las historias de usuario desarrolladas en la segunda iteración

**Tabla 23 Tarea 1 de la historia de usuario Establecer comunicación con el cliente número 2**

Tarea de Ingeniería
---------------------

<b>Número Tarea:</b> 1.	<b>Número Historia de Usuario:</b> 1.
<b>Nombre Tarea:</b> Implementar la funcionalidad que permita Establecer comunicación con el cliente número 2.	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos Estimados:</b> 2
<b>Fecha Inicio:</b> 29/03/11	<b>Fecha Fin:</b> 11/04/11
<b>Programador Responsable:</b> Alberto Aliaga González	
<b>Descripción:</b> Se implementa la funcionalidad que permita establecer con el cliente número 2 mientras que el cliente 1 se encuentra en espera de la comunicación.	

**Tabla 24 Tarea 1 de la historia de usuario Establecer comunicación entre ambos usuarios.**

Tarea de Ingeniería	
<b>Número Tarea:</b> 1.	<b>Número Historia de Usuario:</b> 2.
<b>Nombre Tarea:</b> Implementar la funcionalidad que permita Establecer comunicación entre ambos usuarios	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos Estimados:</b> 2
<b>Fecha Inicio:</b> 12/03/11	<b>Fecha Fin:</b> 25/04/11
<b>Programador Responsable:</b> Alberto Aliaga González	
<b>Descripción:</b> En dependencia de la respuesta del cliente 2 de si acepta la llamada o no se establece la comunicación entre ambos.	

**Tabla 25 Tarea 1 de la historia de usuario Finalizar comunicación.**

Tarea de Ingeniería	
<b>Número Tarea:</b> 1.	<b>Número Historia de Usuario:</b> 3.
<b>Nombre Tarea:</b> Implementar la funcionalidad que permita Finalizar la comunicación entre ambos usuarios.	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos Estimados:</b> 1/5
<b>Fecha Inicio:</b> 26/03/11	<b>Fecha Fin:</b> 27/04/11

<b>Programador Responsable:</b> Alberto Aliaga González
<b>Descripción:</b> Si el usuario 2 no aceptase la llamada el sistema deberá reproducirle un mensaje al cliente 1 propiciándole dicha información, de lo contrario una vez establecida la comunicación los usuario concluirán su comunicación cuando uno de los dos cuelgue el teléfono.

### 4.1.3 Iteración 3

**Tabla 26 Historias de Usuario desarrolladas en la tercera iteración**

Historias de Usuario	Estimación (Días)	Real (Días)
Revertir cobro	10	10
Mostrar reportes	10	10
<b>Total</b>	<b>4 semanas</b>	<b>4 semanas</b>

Tareas de las historias de usuario desarrolladas en la tercera iteración.

**Tabla 27 Tarea 1 de la historia de usuario Revertir Cobro.**

Tarea de Ingeniería	
<b>Número Tarea:</b> 1.	<b>Número Historia de Usuario:</b> 1.
<b>Nombre Tarea:</b> Implementar la funcionalidad que permita revertir el cobro de las llamadas	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos Estimados:</b> 2
<b>Fecha Inicio:</b> 28/04/11	<b>Fecha Fin:</b> 11/05/11
<b>Programador Responsable:</b> Alberto Aliaga González	
<b>Descripción:</b> Una vez terminada la comunicación entre ambos usuarios el sistema debe ser capaz de revertir el cobro de las llamadas.	

**Tabla 28 Tarea 1 de la historia de usuario Mostrar Reportes.**

Tarea de Ingeniería	
<b>Número Tarea:</b> 1.	<b>Número Historia de Usuario:</b> 3.
<b>Nombre Tarea:</b> Diseñar la interfaz Mostrar reportes	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos Estimados:</b> 2/5
<b>Fecha Inicio:</b> 13/05/11	<b>Fecha Fin:</b> 16/05/11
<b>Programador Responsable:</b> Alberto Aliaga González	
<b>Descripción:</b> Se diseña la interfaz de la funcionalidad Mostrar Reportes, permitiendo que se introduzcan los datos para la búsqueda personalizada de reportes de llamadas.	

**Tabla 29 Tarea 2 de la historia de usuario Mostrar Reportes.**

Tarea de Ingeniería	
<b>Número Tarea:</b> 2.	<b>Número Historia de Usuario:</b> 3.
<b>Nombre Tarea:</b> Implementar la funcionalidad que permita filtrar los resultados de los reportes del servicio atendiendo a el parámetro seleccionado por el administrador de la central telefónica.	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos Estimados:</b> 2
<b>Fecha Inicio:</b> 17/05/11	<b>Fecha Fin:</b> 26/05/11
<b>Programador Responsable:</b> Alberto Aliaga González	
<b>Descripción:</b> Se implementa una funcionalidad que le permite al usuario de la página seleccionar el criterio por el cual desea realizar una búsqueda personalizada, estos criterios son: fecha, extensión, día.	

#### 4.2 Pruebas

Uno de los factores fundamentales de la metodología XP es el proceso de pruebas. XP el cual prueba 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, diseñada por los programadores con el objetivo verificar el código y pruebas de aceptación o pruebas funcionales destinadas a evaluar si al final de una iteración se consiguió la funcionalidad requerida diseñadas por el cliente final.

#### **4.2.1 Pruebas Unitarias.**

En el transcurso de la implementación de un sistema, cada uno de los desarrolladores tiene que ir probando constantemente lo que va obteniendo para garantizar que las funcionalidades exigidas por el cliente estén siendo implementadas correctamente. A estas pruebas se le llama Pruebas Unitarias. Aunque estas no generan artefactos y no son directamente palpables para el cliente, son de vital importancia para el desarrollo de un proyecto. Ejemplo de esto fueron las realizadas al servicio de cobro revertido implementado para saber realmente cual era su respuesta una vez que el sistema solicita una serie de datos y el usuario introduce los mismos de manera correcta o incorrecta, de esta forma el sistema pone en uso las validaciones implementadas de acuerdo con el juego de datos introducido.

**Tabla 30 Ejemplo de descripción de un prueba unitaria**

Clase: Cobrorevertido		
Funcionalidad: Confirmar datos		
Valor de entrada	Valor esperado	Valores obtenidos
\$numero	true	true
\$numero_incorrecto	false	false
Resultado de la prueba: Prueba satisfactoria		

#### **4.2.2 Pruebas de Aceptación.**

Las pruebas de aceptación son las realizadas por el cliente y usuarios finales de la aplicación. En estas serán probadas las funcionalidades exigidas por el cliente y descritas en las historias de usuario, además de los aspectos de seguridad requeridos. Luego de haber superado las pruebas de aceptación podrá considerarse que la aplicación es apta para el uso y despliegue dentro del proyecto.

Como resultado de las pruebas de aceptación se obtendrán artefactos descritos en tablas, estas contarán con los siguientes campos:

- ✓ Código: servirá como identificador de la prueba realizada, a su vez será sugerente al nombre de la prueba a la que hace referencia.
- ✓ Historia de usuario: tendrá el nombre de la historia de usuario a la que hace referencia la prueba a realizar.
- ✓ Nombre de la persona que realiza la prueba: persona que realiza la prueba.
- ✓ Descripción: se describe la funcionalidad que se desea probar.
- ✓ Condiciones de Ejecución: mostrará las condiciones que deben cumplirse para poder llevar a cabo el caso de prueba.
- ✓ Entradas / Pasos de Ejecución: se hará la descripción de cada uno de los pasos seguidos durante el desarrollo de la prueba, se tendrá en cuenta cada una de las entradas que hace el usuario con el objetivo de ver si se obtiene el resultado esperado.
- ✓ Resultado esperado: se hará una breve descripción del resultado que se espera obtener con la prueba realizada.
- ✓ Evaluación de la prueba: acorde al resultado de la prueba realizada se emitirá una evaluación sobre la misma. Esta evaluación tendrá uno de los tres resultados que a continuación se describen:
  1. Satisfactoria: cuando el resultado de la prueba es exactamente el esperado por el usuario.
  2. Parcialmente satisfactoria: cuando el resultado no es completamente el esperado por el cliente o usuario de la aplicación y muestra resultados erróneos o fuera de contexto.

3. Insatisfactoria cuando el resultado de la prueba realizada genera un error de codificación en la aplicación o muestra como resultado elementos no deseados o fuera de contexto, trayendo como consecuencia que la funcionalidad requerida por el cliente no tenga resultado, lo que invalida también la historia de usuario.

Las pruebas de aceptación se llevarán a cabo siguiendo los pasos que se muestran a continuación:

1. Se redactarán los casos de prueba teniendo en cuenta el orden de las historias de usuarios.
2. Se planificarán con el cliente.
3. Se completarán cada uno de los campos de las tablas de las pruebas de aceptación con el resultado de la prueba.

A continuación se presentan las pruebas organizadas de conjunto con el cliente:

**Tabla 31 Prueba de aceptación para la historia de usuario Reproducir grabación**

<b>Caso de Prueba de Aceptación.</b>	
<b>Código: SCR 01-01</b>	<b>Historia de Usuario:</b> Reproducir grabación
<b>Nombre de la persona que realiza la prueba:</b> Alberto Aliaga González.	
<b>Descripción:</b> Prueba para verificar la funcionalidad Reproducir grabación.	
<b>Condiciones de Ejecución:</b> Cualquier usuario que acceda al servicio podrá recibir las instrucciones para el trabajo con el mismo a través de la reproducción de mensajes.	
<b>Entrada/Pasos de ejecución:</b>	
<ul style="list-style-type: none"> <li>✓ Se marcará el código de acceso al servicio.</li> <li>✓ El sistema reproduce el mensaje de bienvenida.</li> </ul>	
<b>Resultado Esperado:</b> El sistema reproduce un mensaje de bienvenida.	
<b>Evaluación de la Prueba:</b> Prueba satisfactoria.	

**Tabla 32 Prueba de aceptación para la historia de usuario Grabar nombre**

<b>Caso de Prueba de Aceptación.</b>
--------------------------------------

<b>Código:</b> SCR 02-01	<b>Historia de Usuario:</b> Grabar nombre
<b>Nombre de la persona que realiza la prueba:</b> Alberto Aliaga González.	
<b>Descripción:</b> Prueba para verificar la funcionalidad Grabar nombre.	
<b>Condiciones de Ejecución:</b> El usuario debe haber recibido la orientación de pronunciar su nombre.	
<b>Entrada/Pasos de ejecución:</b>	
<ul style="list-style-type: none"> <li>✓ El sistema reproduce el mensaje que orienta al usuario pronunciar su nombre en un tiempo determinado.</li> <li>✓ El usuario pronuncia su nombre correctamente y el sistema graba el sonido en la carpeta especificada con el nombre orientado.</li> </ul>	
<b>Resultado Esperado:</b> El usuario pronuncia su nombre correctamente y el sistema graba el sonido en la carpeta especificada con el nombre orientado.	
<b>Evaluación de la Prueba:</b> Prueba satisfactoria.	

**Tabla 33 Prueba de aceptación para la historia de usuario Grabar nombre**

<b>Caso de Prueba de Aceptación.</b>	
<b>Código:</b> SCR 02-02	<b>Historia de Usuario:</b> Grabar nombre
<b>Nombre de la persona que realiza la prueba:</b> Alberto Aliaga González.	
<b>Descripción:</b> Prueba para verificar la funcionalidad Confirmar datos.	
<b>Condiciones de Ejecución:</b> El usuario debe haber recibido la orientación de pronunciar su nombre.	
<b>Entrada/Pasos de ejecución:</b>	
<ul style="list-style-type: none"> <li>✓ El sistema reproduce el mensaje que orienta al usuario pronunciar su nombre. El usuario no pronuncio su nombre.</li> <li>✓ El sistema grabó el ruido ambiente.</li> </ul>	
<b>Resultado Esperado:</b> El sistema grabó el ruido ambiente.	
<b>Evaluación de la Prueba:</b> Prueba satisfactoria.	

**Tabla 34 Prueba de aceptación para la historia de usuario Confirmar datos.**

<b>Caso de Prueba de Aceptación.</b>	
<b>Código:</b> SCR 03-01	<b>Historia de Usuario:</b> Confirmar datos
<b>Nombre de la persona que realiza la prueba:</b> Alberto Aliaga González.	
<b>Descripción:</b> Prueba para verificar la funcionalidad Confirmar datos.	
<b>Condiciones de Ejecución:</b> El sistema reproduce un mensaje para la confirmación del mensaje.	
<b>Entrada/Pasos de ejecución:</b> <ul style="list-style-type: none"> <li>✓ El sistema reproduce el mensaje que orienta al usuario introducir un dato.</li> <li>✓ El usuario introduce el dato y el sistema le reproduce en un mensaje el dato introducido para confirmarlo.</li> <li>✓ El usuario comprueba que el dato es correcto.</li> </ul>	
<b>Resultado Esperado:</b> El usuario comprueba que el dato es correcto.	
<b>Evaluación de la Prueba:</b> Prueba satisfactoria.	

**Tabla 35 Prueba de aceptación para la historia de usuario Confirmar datos.**

<b>Caso de Prueba de Aceptación.</b>	
<b>Código:</b> SCR 03-02	<b>Historia de Usuario:</b> Confirmar datos
<b>Nombre de la persona que realiza la prueba:</b> Alberto Aliaga González.	
<b>Descripción:</b> Prueba para verificar la funcionalidad Confirmar datos.	
<b>Condiciones de Ejecución:</b> El sistema reproduce un mensaje para la confirmación del mensaje.	
<b>Entrada/Pasos de ejecución:</b> <ul style="list-style-type: none"> <li>✓ El sistema reproduce el mensaje que orienta al usuario introducir un dato.</li> <li>✓ El usuario introduce el dato y el sistema le reproduce en un mensaje el dato introducido para confirmarlo.</li> <li>✓ El usuario comprueba que el dato es incorrecto y el sistema le brinda la opción de volver a introducir el dato.</li> </ul>	

**Resultado Esperado:** El usuario comprueba que el dato es incorrecto y el sistema le brinda la opción de volver a introducir el dato.

**Evaluación de la Prueba:** Prueba satisfactoria.

**Tabla 36 Prueba de aceptación para la historia de usuario Establecer comunicación con el cliente número 2**

<b>Caso de Prueba de Aceptación.</b>	
<b>Código:</b> SCR 04-01	<b>Historia de Usuario:</b> Establecer comunicación con el cliente número 2
<b>Nombre de la persona que realiza la prueba:</b> Alberto Aliaga González.	
<b>Descripción:</b> Prueba para verificar la funcionalidad Establecer comunicación con el cliente número 2.	
<b>Condiciones de Ejecución:</b> El usuario 1 debe haber introducido los datos necesarios para establecer la comunicación.	
<b>Entrada/Pasos de ejecución:</b>	
<ul style="list-style-type: none"> <li>✓ El sistema reproduce el mensaje que orienta al usuario número 1 esperar para establecer comunicación con el otro usuario.</li> <li>✓ El sistema establece comunicación con el cliente número 2.</li> </ul>	
<b>Resultado Esperado:</b> El sistema establece comunicación con el cliente número 2.	
<b>Evaluación de la Prueba:</b> Prueba satisfactoria.	

**Tabla 37 Prueba de aceptación para la historia de usuario Establecer comunicación entre los 2 clientes**

<b>Caso de Prueba de Aceptación.</b>	
<b>Código:</b> SCR 05-01	<b>Historia de Usuario:</b> Establecer comunicación entre los 2 clientes
<b>Nombre de la persona que realiza la prueba:</b> Alberto Aliaga González.	
<b>Descripción:</b> Prueba para verificar la funcionalidad Establecer comunicación entre los 2 clientes.	

<b>Condiciones de Ejecución:</b> El sistema debe de haber establecido comunicación con el cliente 2.
<b>Entrada/Pasos de ejecución:</b> <ul style="list-style-type: none"> <li>✓ El sistema reproduce el mensaje que pregunta al usuario número 2 si desea aceptar la llamada.</li> <li>✓ El usuario número 2 acepta la llamada.</li> <li>✓ El sistema establece comunicación entre ambos usuarios.</li> </ul>
<b>Resultado Esperado:</b> El sistema establece entre ambos usuarios.
<b>Evaluación de la Prueba:</b> Prueba satisfactoria.

**Tabla 38 Prueba de aceptación para la historia de usuario Establecer comunicación entre los 2 clientes**

<b>Caso de Prueba de Aceptación.</b>	
<b>Código:</b> SCR 05-02	<b>Historia de Usuario:</b> Establecer comunicación entre los 2 clientes
<b>Nombre de la persona que realiza la prueba:</b> Alberto Aliaga González.	
<b>Descripción:</b> Prueba para verificar la funcionalidad Establecer comunicación entre los 2 clientes.	
<b>Condiciones de Ejecución:</b> El sistema debe de haber establecido comunicación con el cliente 2.	
<b>Entrada/Pasos de ejecución:</b> <ul style="list-style-type: none"> <li>✓ El sistema reproduce el mensaje que pregunta al usuario número 2 si desea aceptar la llamada.</li> <li>✓ El usuario número 2 rechaza la llamada.</li> <li>✓ El sistema reproduce un mensaje al usuario 1 explicándole que la persona a la que va dirigida la llamada no acepto la misma.</li> </ul>	
<b>Resultado Esperado:</b> El sistema debe reproducir un mensaje al usuario 1 explicándole que la persona a la que va dirigida la llamada no acepto la misma.	
<b>Evaluación de la Prueba:</b> Prueba satisfactoria.	

**Tabla 39 Prueba de aceptación para la historia de usuario Finalizar comunicación.**

<b>Caso de Prueba de Aceptación.</b>
--------------------------------------

<b>Código:</b> SCR 06-01	<b>Historia de Usuario:</b> Finalizar comunicación
<b>Nombre de la persona que realiza la prueba:</b> Alberto Aliaga González.	
<b>Descripción:</b> Prueba para verificar la funcionalidad Finalizar comunicación.	
<b>Condiciones de Ejecución:</b> El sistema debe de haber establecido comunicación con el cliente 2.	
<b>Entrada/Pasos de ejecución:</b>	
<ul style="list-style-type: none"> <li>✓ Al terminar la conversación el sistema finaliza la llamada.</li> <li>✓ El sistema termina la comunicación con los clientes.</li> </ul>	
<b>Resultado Esperado:</b> El sistema termina la comunicación con los clientes.	
<b>Evaluación de la Prueba:</b> Prueba satisfactoria.	

**Tabla 40 Prueba de aceptación para la historia de usuario Finalizar comunicación.**

<b>Caso de Prueba de Aceptación.</b>	
<b>Código:</b> SCR 06-02	<b>Historia de Usuario:</b> Finalizar comunicación
<b>Nombre de la persona que realiza la prueba:</b> Alberto Aliaga González.	
<b>Descripción:</b> Prueba para verificar la funcionalidad Finalizar comunicación.	
<b>Condiciones de Ejecución:</b> El sistema debe de haber establecido comunicación con el cliente 2.	
<b>Entrada/Pasos de ejecución:</b>	
El cliente número 2 rechaza la llamada.	
El sistema reproduce mensajes a ambos clientes finaliza la comunicación.	
<b>Resultado Esperado:</b> El sistema reproduce mensajes a ambos clientes finaliza la comunicación.	
<b>Evaluación de la Prueba:</b> Prueba satisfactoria.	

**Tabla 41 Prueba de aceptación para la historia de usuario Revertir cobro.**

<b>Caso de Prueba de Aceptación.</b>	
<b>Código:</b> SCR 07-01	<b>Historia de Usuario:</b> Revertir cobro.
<b>Nombre de la persona que realiza la prueba:</b> Alberto Aliaga González.	

<b>Descripción:</b> Prueba para verificar la funcionalidad Revertir Cobro.
<b>Condiciones de Ejecución:</b> Debe haber finalizado la comunicación entre los usuarios.
<b>Entrada/Pasos de ejecución:</b> <ul style="list-style-type: none"> <li>✓ El sistema guardará en la base de datos el reporte de la llamada de forma tal que sea fácil de identificar el tipo de llamadas.</li> </ul>
<b>Resultado Esperado:</b> Se guardarán los datos de manera que sea fácil distinguir el tipo de llamada.
<b>Evaluación de la Prueba:</b> Prueba satisfactoria.

**Tabla 42 Prueba de aceptación para la historia de usuario Mostrar Reportes**

<b>Caso de Prueba de Aceptación.</b>	
<b>Código:</b> SCR 08-01	<b>Historia de Usuario:</b> Mostrar reportes
<b>Nombre de la persona que realiza la prueba:</b> Alberto Aliaga González.	
<b>Descripción:</b> Prueba para verificar la funcionalidad Mostrar reportes.	
<b>Condiciones de Ejecución:</b> El usuario debe haberse autenticado en el sistema.	
<b>Entrada/Pasos de ejecución:</b> <ul style="list-style-type: none"> <li>✓ El usuario realiza una búsqueda personalizada atendiendo a un criterio de búsqueda seleccionado (Fecha, día, extensión).</li> <li>✓ El portal muestra los resultados de la búsqueda.</li> </ul>	
<b>Resultado Esperado:</b> El sitio muestra los resultados de la búsqueda.	
<b>Evaluación de la Prueba:</b> Prueba satisfactoria.	

**Tabla 43 Prueba de aceptación para la historia de usuario Mostrar Reportes**

<b>Caso de Prueba de Aceptación.</b>	
<b>Código:</b> SCR 09-01	<b>Historia de Usuario:</b> Mostrar reportes
<b>Nombre de la persona que realiza la prueba:</b> Alberto Aliaga González.	
<b>Descripción:</b> Prueba para verificar la funcionalidad Mostrar reportes.	

**Condiciones de Ejecución:** El usuario debe haberse autenticado en el sistema.

**Entrada/Pasos de ejecución:**

- ✓ El usuario realiza una búsqueda personalizada atendiendo a un criterio de búsqueda seleccionado (Fecha, día, extensión) introduciendo datos incorrectos.
- ✓ El portal muestra una alerta aclarando que no existe un resultado para los parámetros solicitados.

**Resultado Esperado:** El portal mostrará una alerta aclarando que no existe un resultado para los parámetros solicitados.

**Evaluación de la Prueba:** Prueba satisfactoria.

**Conclusiones parciales**

En este capítulo se describieron las fases de implementación y prueba. Fueron detalladas además cada una de las tareas que se realizaron en las tres iteraciones del sistema. Se abordó acerca de la importancia de las pruebas en la metodología XP, en especial las pruebas de aceptación, ya que éstas miden la satisfacción del cliente con el producto desarrollado.

## **CAPÍTULO V: ESTUDIO DE FACTIBILIDAD**

### **INTRODUCCIÓN**

En el proceso de desarrollo de software la planificación es muy importante para poder determinar con certeza los medios y recursos que serán necesarios para el desarrollo del mismo, ya sean recursos de hardware, software, esfuerzo, tiempo o costo. En el presente capítulo se realizará un estudio de factibilidad para la realización del sistema propuesto mediante una estimación de tamaño, esfuerzo y planificación necesaria para llevar a cabo la implementación del mismo.

#### **4.1 COCOMO II.**

COCOMO, propuesto y desarrollado por Barry Boehm, es un modelo de estimación de costos que permite determinar el esfuerzo y tiempo que se requiere en un proyecto de software a partir de una medida del tamaño del mismo expresada en el número de líneas de código que se estimen generar para la creación del producto software.

Está basado en la aplicación de ecuaciones matemáticas sobre los Puntos de Función sin ajustar o la cantidad de líneas de código (SLOC, Source Lines of Code) estimados para un proyecto. Estas ecuaciones se encuentran ponderadas por ciertos factores de costo (cost drivers) que influyen en el esfuerzo requerido para el desarrollo del software.

Está adaptado tanto a las necesidades de los diferentes sectores descritos, como al tipo y cantidad de información disponible en cada etapa del ciclo de vida de desarrollo, lo que se conoce por granularidad de la información (13).

Es necesario tener en cuenta que el estudio de la factibilidad de este sistema se realizó mediante el modelo de diseño temprano que se utiliza en las primeras etapas del desarrollo, en las cuales se evalúan las alternativas de hardware y software de un proyecto. En estas etapas se tiene poca información, se conoce muy poco del tamaño del producto a ser desarrollado, de la naturaleza de la plataforma, del personal a ser incorporado al proyecto o aspectos específicos del proceso a utilizar. El nivel de detalle en este modelo es consistente con el nivel general de información disponible y con el nivel general de estimación detallada necesaria en estas etapas, lo que concuerda con el uso de Puntos de Función. Para

estimar tamaño se precisa de los Puntos de Función No Ajustados como métrica de medida y un número reducido de factores de costo.

**4.2 Características del Proyecto.**

Para llevar a cabo la estimación del presente proyecto es necesaria la obtención de los Puntos de Función desajustados, los cuales están dados por la suma de cada una de las entradas, las salidas y las consultas externas del sistema, así como los archivos lógicos internos y de interfaz externos. A continuación se muestran cada una de estas características aplicadas al software en cuestión.

**4.2.1 Entradas Externas.**

Se definen como un proceso elemental mediante el cual ciertos datos cruzan la frontera del sistema desde afuera hacia adentro. En el caso particular de la aplicación propuesta se cuenta con tres entradas externas, especificadas en la siguiente tabla:

**Tabla 44 Entradas externas del sistema**

<b>Nombre de la entrada externa.</b>	<b>Cantidad de ficheros.</b>	<b>Cantidad de elementos de datos.</b>	<b>Clasificación (Simple, Media, Compleja)</b>
Grabar nombre	1	1	Simple
Confirmar datos	1	1	Simple
Establecer comunicación con el cliente 2	1	1	Simple
Mostrar Reportes	7	3	Compleja
<b>Total</b>	<b>10</b>	<b>6</b>	

**4.2.2 Salidas Externas**

Son componentes de entrada y de salida mediante el cual datos simples y datos derivados (esto es, datos que se calculan a partir de otros datos) cruzan las fronteras del sistema desde adentro hacia afuera. Las salidas externas vinculadas al proyecto se describen a continuación.

**Tabla 45 Salidas externas**

Nombre de la salida externa.	Cantidad de ficheros.	Cantidad de elementos de datos.	Clasificación (Simple, Media. Compleja)
Mostrar reportes	7	3	Compleja
<b>Total</b>	<b>7</b>	<b>3</b>	

#### 4.2.3 Consultas Externas.

Son un proceso elemental con componentes de entrada y de salida donde un Actor del sistema rescata datos de uno o más Archivos Lógicos Internos o Archivos de Interfaz Externos. Los datos de entrada no actualizan ni mantienen ningún archivo (lógico interno o de interfaz externo) y los datos de salida no contienen datos derivados (es decir, los datos de salida son básicamente los mismos que se obtienen de los archivos). En el presente caso no fueron identificados.

**Tabla 46 Consultas externas**

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

#### 4.2.4 Archivos Lógicos Internos.

Son un grupo de datos relacionados lógicamente e identificables por el usuario, que residen enteramente dentro de los límites del sistema y se mantienen a través de entradas externas.

**Tabla 47 Archivos lógicos externos**

Nombre de la Petición.	Cantidad de ficheros.	Cantidad de elementos de datos.	Clasificación (Simple, Media. Compleja)
Base Batos: Asteriskcdrdb	1	1(tablas)	Simple.

<b>Total</b>	<b>1</b>	<b>1</b>	
--------------	----------	----------	--

#### 4.2.5 Archivos de Interfaz Externos.

Son un grupo de datos relacionados lógicamente e identificables por el usuario, que se utilizan solamente para fines de referencia. Los datos residen enteramente fuera de los límites del sistema y se mantienen por las Entradas Externas de otras aplicaciones, es decir, cada Archivo de Interfaz Externo es un Archivo Lógico Interno de otra aplicación. En el presente caso no fue identificado ninguno.

**Tabla 48 Archivos de interfaz externos**

Nombre de la interfaz externa.	Cantidad de ficheros.	Cantidad de elementos de datos.	Clasificación (Simple, Media, Compleja)
<b>Total</b>	<b>0</b>	<b>0</b>	

#### Puntos de Función Desajustados.

**Tabla 49 Puntos de Función desajustados**

Elementos.	Simple.		Medio.		Complejo.		Subtotal.
	No.	Peso.	No.	Peso.	No.	Peso.	
Entradas Externas.	3	3	0	4	1	6	15
Salidas Externas.	0	4	0	5	3	7	21
Consultas Externas.	0	3	0	4	0	6	0
Fichero Lógico Interno.	1	7	0	10	0	15	7
Fichero Interfaz Externo.	0	5	0	7	0	10	0
<b>Total (UFP):</b>							<b>43</b>

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

**4.3.1 Cálculo de instrucciones fuentes.**

Luego de contar con la cantidad de puntos de función desajustados es posible proceder a calcular la cantidad de instrucciones fuentes:

$$SLOC = UFP \times ratio$$

Donde:

**SLOC**: Cantidad de instrucciones fuente (2537).

**UFP** : Puntos de función desajustados. (43)

**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:

$$SLOC = 94 \times 59$$

$$SLOC = 2537 = 2.537KSLOC$$

**KSLOC** : Tamaño del software a desarrollar.

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

**Tabla 50 Características y valores del sistema**

Características	Valor
Puntos de función desajustados.	43
Lenguaje (PHP).	59
Instrucciones fuentes por puntos de función.	2537 SLOC
Instrucciones fuentes.	2.53 KSLOC

#### 4.3.2 Cálculo de esfuerzo nominal.

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

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

$$PM_{nominal} = 2.94 \times (2.53)^{1.23}$$

$$PM_{nominal} = 9.2 \text{ meses/hombre}$$

Donde:

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

**$Size$** : Tamaño estimado del software en Puntos de Función sin Ajustar. (4.1 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.

$$B = 0.91 + 0.01 x (W_i)$$

$$B = 0.91 + 0.01 \times 11.33$$

$$B = 1.23$$

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.

#### **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

#### **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.

**Cohesión del Equipo (TEAM).**

El factor de escala denominado Cohesión del Equipo tiene en cuenta las dificultades de sincronización entre los participantes del proyecto: usuarios, clientes y desarrolladores. Estas dificultades pueden surgir por diferencias culturales, dificultad en la conciliación de objetivos, falta de experiencia y familiaridad con el trabajo en equipo.

**Solución de riesgos (RESL).**

Este factor involucra aspectos relacionados al conocimiento de los ítems de riesgo crítico y al modo de abordarlos dentro del proyecto.

**Madurez del proceso (PMAT).**

El procedimiento para determinar el PMAT es establecer el porcentaje de cumplimiento de cada una de las Áreas evaluando el grado de cumplimiento de las metas correspondientes.

A continuación la figura muestra la tabla de los Factores de escala:



	VLO	LO	NOM	HI	VHI	ZHI
PREC	6.20	4.96	3.72	2.48	1.24	0.00
FLEX	5.07	4.05	3.04	2.03	1.01	0.00
RESL	7.07	5.65	4.24	2.83	1.41	0.00
TEAM	5.48	4.38	3.29	2.19	1.10	0.00
PMAT	7.80	6.24	4.68	3.12	1.56	0.00

**Figura 5 Factores de escala COCOMO II**

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

**Tabla 51 Factores de escala del sistema**

Nombre.	Valor.	Justificación.
PREC	2.4	Existen varios proyectos similares a nivel

		internacional, pero no se encontró ninguno a nivel nacional.
FLEX	2.03	Cuenta con alta flexibilidad en cuanto a los requerimientos que demanda el cliente.
TEAM	1.10	El equipo de desarrollo presenta una gran sincronización en el trabajo.
RESL	4.24	No se identificaron grandes riesgos.
PMAT	1.56	Se cuenta con la experiencia para que el software cumpla con las funcionalidades solicitadas por el cliente.
Total(SF)	11.33	

#### **4.4 Ajuste del esfuerzo nominal.**

El esfuerzo nominal calculado debe ser ajustado teniendo en cuenta las características del proyecto, para ello se cuenta con un conjunto de Multiplicadores de Esfuerzo (MEi) que representan las características del proyecto y arrojan su impacto en el desarrollo total del producto de software.

##### **4.4.1 Multiplicadores de esfuerzo**

Existen 7 multiplicadores de esfuerzo, estos son:

Del Producto.

- ✓ RCPX: Confiabilidad y Complejidad del producto
- ✓ RUSE: Reusabilidad Requerida

De la Plataforma.

- ✓ PDIF: Dificultad de la Plataforma

Del Personal.

- ✓ PERS: Aptitud del Personal
- ✓ PREX: Experiencia del Personal

Del Proyecto.

- ✓ FCIL: Facilidades
- ✓ SCED: Cronograma de Desarrollo Requerido

Para otorgar los valores de estos datos se encuentra la tabla de multiplicadores de esfuerzo del modelo del diseño temprano.



	ELO	VLO	LO	NOE	EE	VHI	SHI
RCPX	0.73	0.91	0.99	1.00	1.30	1.74	2.39
RUSE	XXXX	XXXX	0.95	1.00	1.07	1.15	1.24
PDIF	XXXX	XXXX	0.87	1.00	1.29	1.81	2.61
PERE	2.12	1.62	1.26	1.00	0.83	0.63	0.50
PREX	1.50	1.33	1.12	1.00	0.67	0.71	0.62
FCIL	1.42	1.30	1.10	1.00	0.87	0.73	0.62
SCED	XXXX	1.43	1.14	1.00	1.00	1.00	XXXX
DSR1	XXXX	1.00	1.00	1.00	1.00	1.00	XXXX
DSR2	XXXX	1.00	1.00	1.00	1.00	1.00	XXXX

Figura 6 Tabla de multiplicadores de esfuerzo

**Tabla 52 Factores de esfuerzo del sistema**

Nombre.	Valor.	Justificación.
RCPX	1.00	La complejidad del sistema es normal.
RUSE	1.00	Se reutilizará una parte del código.
PDIF	1.00	Uso de memoria y almacenamiento normal, plataforma estable.
PREX	1.12	Experiencia en el personal en cuanto a la utilización del lenguaje y herramientas.

PERS	1.00	Capacidad del personal es normal.
FCIL	1.00	La utilización de entornos de desarrollo integrados facilita en gran medida el trabajo.
SCED	1.00	El sistema se desarrollo en el tiempo establecido.
Total(EM)	1.25	

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

$$PM_{ajustado} = 9.2 \text{ meses/hombre} \times 1.25$$

$$PM_{ajustado} = 11.5 \text{ meses/hombre.}$$

#### 4.5 Cálculo del tiempo para el desarrollo del software.

El tiempo para el desarrollo del software se calcula de la siguiente forma:

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

$$TDEV = 3.64 \times 11.5^{0.005}$$

$$TDEV=3.68$$

Donde:

**C:** Constante con valor 3.64.

$$PM_{ajustado} = 11.5 \text{ meses/hombre.}$$

$$F: 0.005$$

CALCULAR

$$F = D + 0.2 \times 0.01x \quad SF$$

$$F = 0.24 + 0.2 \times 0.01 \times 11.33$$

$$F = 0.005$$

Donde:

$D$  : Constante cuyo valor es 0.24.

$SF$  : Valor de los factores de escala.

#### 4.6 Cálculo del costo total del proyecto.

Para el cálculo del costo total, el modelo matemático utilizado propone la siguiente ecuación:

$$C = CHM \times PM$$

$$C = 100 \times 9.2$$

$$C = 920$$

CALCULAR

Donde:

$C$ : Costo total.

$CHM$ : Costo teniendo en cuenta salario de todos los obreros, el cual se calcula por la siguiente ecuación.

$$CHM = CH \times sal$$

$$CHM = 1 \times 100.$$

$$CHM = 100$$

CALCULAR

Donde:

$sal$  : Salario medio por cada trabajador.

$CH$  : Cantidad de personas destinadas al proyecto.

$$CH = 1.$$

#### 4.7 Resultados.

En la siguiente tabla se muestran los resultados obtenidos luego de haber efectuado todos los cálculos para determinar el costo y esfuerzo requeridos por el proyecto.

<b>Cálculo de:</b>	<b>Valor</b>
Esfuerzo.	9.2 meses/hombre.
Tiempo de desarrollo.	4 meses.
Cantidad de hombres.	1 hombre.
Salario medio.	100 pesos.
Costo.	920 pesos.

#### **4.8 Análisis del costo**

El sistema que se desarrolló en este trabajo no requiere de grandes gastos, debido a que solo es influyente el salario de los desarrolladores, por lo cual se determina que su implementación es factible. Esto se debe en gran magnitud a la utilización de las herramientas libres que no requieren el pago de alguna licencia.

#### **4.9 Conclusiones parciales.**

En el presente capítulo se realizó un análisis de factibilidad de la solución propuesta, arribando a la conclusión de que es posible su desarrollo comparando los costos de producción con los beneficios reportados por su puesta en funcionamiento.

## **CONCLUSIONES GENERALES**

Con el desarrollo de este trabajo se profundizaron los conocimientos sobre la telefonía IP, la utilización de la metodología de desarrollo XP permitió guiar el proceso de desarrollo mediante el cual se obtuvo la implementación de un Sistema de Cobro Revertido, el diseño y la implementación del mismo se rigieron por herramientas y lenguajes de programación como Cedrux, PHP y Postgres SQL, aplicando una arquitectura Cliente-Servidor. Al concluir este trabajo se obtuvieron los siguientes resultados:

- ✓ Se desarrolló un servicio de cobro revertido el cual permite a un usuario comunicarse con otro siempre y cuando este último acepte los términos de la llamada.
- ✓ Se implementó una interfaz Web que posibilita mostrar los reportes del servicio.

Con la construcción de los mismos, se logró poner en práctica los conocimientos adquiridos (Ingeniería de Software, Programación, Telecomunicaciones, entre otros), además de conocimientos que fueron adquiridos durante el desarrollo del trabajo.

Por todo lo antes expuesto se concluye que los objetivos propuestos para el presente proyecto han sido cumplidos satisfactoriamente, incluyéndose una serie de recomendaciones que deben tenerse en cuenta para la futura continuidad del sistema.

## RECOMENDACIONES

A continuación se listan una serie de recomendaciones que en opinión del autor deben ser tenidas en cuenta con el objetivo de realizar un seguimiento de este trabajo:

1. Integrar el sistema a la Plataforma de Comunicaciones Unificadas utilizada en el proyecto.
2. Utilizar en el proyecto Comunicaciones Unificadas una versión superior de Asterisk a la existente teniendo en cuenta que permite un mejor desarrollo de las aplicaciones y servicios además de optimizar el sistema desarrollado.

## GLOSARIO DE TÉRMINOS

### ***Respuesta de Voz Interactiva (IVR).***

Es la traducción de Interactive Voice Response. También se utiliza el término Voice Response Unit (VRU). 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 (14).

### ***Línea troncal.***

Una línea troncal es un enlace que interconecta las llamadas externas de una central telefónica, concentrando y unificando varias comunicaciones simultáneas en una sola señal para un transporte y transmisión a distancia más eficiente (generalmente digital) permitiendo establecer comunicaciones con otra central o una red interna de ellas. Una central telefónica tipo PBX utiliza una línea troncal para poder hacer de la central parte de la red de otras centrales y mantener comunicaciones. Generalmente las líneas troncales de los PBX son enlaces digitales que soportan hasta 30 canales (líneas) de voz para la intercomunicación. Si se llegase a interrumpir la comunicación de la línea troncal, no habría manera de establecer comunicación entre las centrales por ninguno de sus 30 canales (14).

### **Protocolo Ip.**

El protocolo de IP (Internet Protocol) es la base fundamental de la Internet. Porta datagramas de la fuente al destino. El nivel de transporte parte el flujo de datos en datagramas. Durante su transmisión se puede partir un datagrama en fragmentos que se montan de nuevo en el destino. Las principales características de este protocolo son:

- Protocolo orientado a no conexión.
- Fragmenta paquetes si es necesario.
- Direccionamiento mediante direcciones lógicas IP de 32 bits.
- Si un paquete no es recibido, este permanecerá en la red durante un tiempo finito.
- Realiza el "mejor esfuerzo" para la distribución de paquetes.
- Tamaño máximo del paquete de 65535 bytes.
- Sólo se realiza verificación por suma al encabezado del paquete, no a los datos éste que contiene.

El Protocolo Internet proporciona un servicio de distribución de paquetes de información orientado a no conexión de manera no fiable. La orientación a no conexión significa que los paquetes de información, que será emitido a la red, son tratados independientemente, pudiendo viajar por diferentes trayectorias para llegar a su destino. El término no fiable significa más que nada que no se garantiza la recepción del paquete.

La unidad de información intercambiada por IP es denominada datagrama. Tomando como analogía los marcos intercambiados por una red física los datagramas contienen un encabezado y un área de datos. IP no especifica el contenido del área de datos, ésta será utilizada arbitrariamente por el protocolo de transporte. (15)

## BIBLIOGRAFÍA CONSULTADA

1. **Rojano, E.** Asterisk. [En línea] 2007. asterisk.org.
2. **Andrade, Jose Antonio.** Desarrollo de Aplicaciones y Soluciones en la Central Asterisk de Telefonía IP. s.l. : 1, 2006.
3. **Solutions, Palosanto.** Elastix Servidor de comunicaciones unificadas. [En línea] Palosanto Solutions, 2008. 5.
4. **José H. Canós, Patricio Letelier.** *Métodologías Ágiles en el Desarrollo de Software.* Valencia : Universidad Politécnica de Valencia., 2004. 4.
5. **R, Pressman.** *Quinta Edición. Ingeniería de Software. Un Enfoque Práctico.* 2004. 8.
6. **Wells, Don.** Extreme Programming: a gentle introduction. [En línea] www.extremeprogramming.org. 2.
7. **Marianela Tenrero Cabrera, Yoandry Morejón Borbon.** *CedruX, Manual de instalación.* s.l. : Departamento de Tecnología del Centro de Soluciones de Gestión de Entidades (CESGE), Centro de Desarrollo y Asimilación de las Tecnologías de la Unidad de Compatibilización, Integración y Desarrollo (UCID), 2010.
8. **Carrero, Angel.** Programación en Castellano. *Programación en Castellano.* [En línea] [Citado el: 6 de Febrero de 2011.] [http://www.programacion.com/articulo/distintos\\_framework\\_para\\_php\\_379](http://www.programacion.com/articulo/distintos_framework_para_php_379).
9. **Alvarez, Miguel Angel.** DesarrolloWeb.com. *DesarrolloWeb.com.* [En línea] Agosto de 2007. [Citado el: 15 de Octubre de 2010.] desarrolloweb.com.
10. **IDATI.** IDATI. *IDATI.* [En línea] [Citado el: 6 de Febrero de 2011.] <http://www.idati.cl/archives/ext-js-framework-javascriptajax-poderoso-para-aplicaciones-ria>.
11. **Alvarez, Sara.** Desarrollo Web.com. *Desarrollo Web.com.* [En línea] 2007. [Citado el: 6 de Febrero de 2011.] desarrolloweb.com.
12. **Tedeschi, Nicolás.** MSDN. *msdn.microsoft.com.* [En línea] msdn.microsoft.com.
13. **Gallegos, Silvia.** *COCOMO II.* 2007.
14. **Goncalves, Flavio Eduardo.** *Asterisk PBX Guía de configuración.* Rio de Janeiro : s.n., 2007. 3.
15. **Herramientas web para la enseñanza.** Herramientas web para la enseñanza. *Herramientas web para la enseñanza.* [En línea] [Citado el: 12 de mayo de 2011.] <http://neo.lcc.uma.es/evirtual/cdd/tutorial/red/ip.html>.

16. **Snyder, Michael Southwell and Chris.** *Pro PHP Security. New York* . 2005. 10.
17. **Quiñonez, Ernesto.** Introducción a Postgresql. [En línea] postgresql.org.
18. **N., Siminovich.** Design and develop Asterisk-based Voip telephony platforms and services. [En línea] Asterisk.org. 7.
19. **Meggelen, Jim Van.** *Asterik:The future of telephony.* 2009.
20. **David Gomillion, Barrie Dempster.** *Building Telephony Systems Whit Asterisk.* Birgming : s.n., 2006.
21. **Voip-Info.** Voip-Info.org. *Voip-Info.org.* [En línea] [Citado el: 13 de Noviembre de 2010.] <http://www.voip-info.org/>.