

**UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS**  
**FACULTAD 2**



**Título:** Sistema de integración de servicios de comunicación a usuarios (SISCU).

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

**Autores:** Claudia Rodríguez Crespo

Odaimys Rodríguez García.

**Tutor:** Ing. Yasser Azán Basallo.

**Co-Tutor:** Ing. Gerardo Rodríguez Fernández.

Ciudad de la Habana, Junio del 2011.



**DECLARACIÓN DE AUTORÍA.**

Declaro ser autor de la presente tesis, reconociendo a la Universidad de las Ciencias Informáticas los derechos patrimoniales de manera exclusiva.

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

---

Claudia Rodríguez Crespo

Firma Autor

---

Odaimys Rodríguez García

Firma Autor

---

Firma del Tutor

Ing. Yasser Azán Basallo

---

Firma del Co-Tutor

Ing. Gerardo Rodríguez Fernández



**Dedicatoria Odaimys:**

Dedico esta tesis a mis padres por ser los motores que impulsan mi andar por la vida, por darme el valor para enfrentar las situaciones difíciles, por apoyarme en las decisiones que tomo aun cuando no sean las correctas y sobre todo por creer ciegamente en que puedo lograr lo que me proponga.

**Dedicatoria Claudia:**

Quiero dedicar este trabajo a las 2 personas que me dieron la vida, a mi mamá Odalis que ha sido mi amiga, mi guía, la persona que ha estado siempre a mi lado y a mi papá Jorge Luis por ser mi ejemplo a seguir, mi amigo y el más cariñoso de los padres. Para ustedes, que son a las personas que le debo lo que soy y que han sido los motores que me han impulsado a llegar hasta aquí, es hoy este trabajo.

## **RESUMEN.**

Producto del problema existente con la recepción de la información por parte de los usuarios y a la importancia de que la misma llegue a sus destinatarios, la Universidad de las Ciencias Informáticas (UCI) tuvo la necesidad de integrar en una misma aplicación los servicios de correo electrónico, mensajería instantánea, telefonía fija y servicio de mensajes cortos, con el propósito de suponer menos tiempo y esfuerzo al usuario a la hora de realizar el envío de mensajes por estas vías. Para ello se le asignó la tarea al proyecto “Comunicaciones Unificadas” del Centro de Telemática de la Facultad 2.

En la actualidad existen varios sistemas que integran algunos de estos servicios y agregan otros similares, presentándose todos a nivel internacional. Luego de realizar un análisis de estas aplicaciones, comparar sus características con las requeridas y tener en cuenta los recursos disponibles, se decidió desarrollar un sistema de integración de servicios de comunicación a usuarios.

Para guiar el proceso de desarrollo del software se seleccionó la metodología XP y para la implementación del sistema se utilizó el lenguaje de programación PHP como lenguaje del lado del servidor, siguiendo el patrón arquitectónico Modelo-Vista-Controlador, además de utilizar el servidor de Comunicaciones Unificadas: Elastix como módulo de administración de Asterisk y servidor de correo electrónico, así como un servidor de mensajería instantánea y servidor web del cual se consume el servicio de envío de mensajes cortos.

**ÍNDICE DE IMÁGENES.**

Imagen 1 Diagrama de Clases.....	19
Imagen 2 Arquitectura Propuesta.....	22
Imagen 3 Arquitectura Cliente Servidor.....	23
Imagen 4 Modelo-Vista-Controlador.....	24
Imagen 5 Diagrama de Entidad Relación de la Base de Datos. ....	48

**ÍNDICE DE TABLAS.**

Tabla 1 Personas relacionadas con el sistema. ....	25
Tabla 2 Adicionar cliente. ....	25
Tabla 3 Modificar cliente. ....	26
Tabla 4 Eliminar cliente. ....	27
Tabla 5 Adicionar contacto. ....	27
Tabla 6 Modificar contacto. ....	28
Tabla 7 Eliminar contacto. ....	29
Tabla 8 Enviar correo masivo. ....	29
Tabla 9 Crear lista de correos. ....	30
Tabla 10 Modificar lista de correos. ....	31
Tabla 11 Eliminar lista de correos. ....	32
Tabla 12 Enviar mensaje instantáneo masivo. ....	32
Tabla 13 Crear lista de usuarios. ....	33
Tabla 14 Modificar lista de usuarios. ....	34
Tabla 15 Eliminar lista de usuarios. ....	35
Tabla 16 Enviar SMS masivo. ....	35
Tabla 17 Crear listas de telefonía móvil. ....	36
Tabla 18 Modificar listas de telefonía móvil. ....	37
Tabla 19 Eliminar listas de telefonía móvil. ....	37
Tabla 20 Enviar mensaje de voz. ....	38
Tabla 21 Crear directorio. ....	39
Tabla 22 Modificar directorio. ....	40
Tabla 23 Eliminar directorio. ....	40
Tabla 24 Estimación de esfuerzo por historia de usuario. ....	41
Tabla 25 Plan de duración de iteraciones. ....	44
Tabla 26 Plan de entregas. ....	45
Tabla 27 Tiempo de las tareas desarrolladas en la primera iteración. ....	49
Tabla 28 Tiempo de las tareas desarrolladas en la segunda iteración. ....	50
Tabla 29 Tiempo de las tareas desarrolladas en la tercera iteración. ....	50
Tabla 30 Tiempo de las tareas desarrolladas en la cuarta iteración. ....	51
Tabla 31 Tiempo de las tareas desarrolladas en la quinta iteración. ....	51
Tabla 32 Cantidad de Pruebas Unitarias realizadas. ....	53
Tabla 33 Entradas externas. ....	57
Tabla 34 Salidas externas. ....	57
Tabla 35 Consultas externas. ....	58
Tabla 36 Fichero lógico interno. ....	58
Tabla 37 Interfaces Externas. ....	58
Tabla 38 Puntos de función desajustados. ....	59
Tabla 39 Peso del factor de complejidad. ....	59
Tabla 40 Características del sistema. ....	60
Tabla 41 Factores de escala del sistema. ....	62
Tabla 42 Factores de escala. ....	62
Tabla 43 Multiplicadores de esfuerzo. ....	63
Tabla 44 Multiplicadores de esfuerzo del modelo del diseño temprano. ....	64
Tabla 45 Resultados. ....	66

**ÍNDICE DE CONTENIDO.**

INTRODUCCIÓN.....	1
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA.....	6
1.1 Introducción.....	6
1.2 Conceptos vinculados al campo de acción.....	6
1.2.1 Correo electrónico.....	6
1.2.2 Mensajería instantánea.....	6
1.2.3 Telefonía fija.....	7
1.2.4 Telefonía IP (Internet Protocol).....	7
1.2.5 Telefonía móvil.....	7
1.2.6 Sistema de mensajes cortos.....	8
1.3 Unificación de servicios de comunicación.....	8
1.4 Sistema Integral de Gestión de Entidades (CEDRUX).....	9
1.5 Metodologías de desarrollo de software.....	11
1.6 Herramientas.....	12
1.6.1 Sistemas gestores de bases de datos.....	12
1.6.2 Lenguajes de programación web.....	13
1.6.3 Servidores para aplicaciones web.....	16
1.6.4 Entorno de Desarrollo Integrado de Desarrollo.....	17
1.7 Conclusiones parciales.....	17
CAPÍTULO 2: EXPLORACIÓN Y PLANIFICACIÓN.....	18
2.1 Introducción.....	18
2.2 Características del sistema.....	18
2.2.1 Objeto de Automatización.....	18
2.2.2 Propuesta del Sistema.....	18
2.2.3 Características no funcionales del sistema.....	20
2.3 Arquitectura.....	21
2.3.1 Propuesta de Arquitectura del Sistema.....	22
2.3.2 Arquitectura Cliente Servidor.....	22
2.3.3 Patrones de arquitectura.....	23
2.4 Exploración.....	24
2.4.1 Flujo de procesos del sistema propuesto.....	24
2.4.2 Personas relacionadas con el sistema.....	25
2.4.3 Historias de Usuarios.....	25
2.5 Planificación.....	41
2.5.1 Estimación de esfuerzo por historias de usuarios.....	41
2.5.2 Plan de Iteraciones.....	42
2.5.3 Plan de duración de las iteraciones.....	44
2.5.4 Plan de entregas.....	45
2.6 Conclusiones parciales.....	45
CAPÍTULO 3: DISEÑO DEL SISTEMA.....	46
3.1 Introducción.....	46
3.2 Patrones de Diseño.....	46
3.3 Tarjetas Cargo o Clase, Responsabilidad y Colaboración (Tarjetas CRC).....	47
3.4 Diseño de la Base de datos.....	48

3.5 Conclusiones parciales.....	48
CAPÍTULO 4: IMPLEMENTACIÓN Y PRUEBA.....	49
4.1 Introducción.....	49
4.2 Implementación.....	49
4.2.1 Primera Iteración.....	49
4.2.2 Segunda Iteración.....	50
4.2.3 Tercera Iteración.....	50
4.2.4 Cuarta Iteración.....	51
4.2.5 Quinta Iteración.....	51
4.3 Prueba.....	52
4.3.1 Prueba Unitaria.....	52
4.3.2 Prueba Aceptación.....	54
4.4 Conclusiones parciales.....	55
CAPÍTULO 5: ESTUDIO DE FACTIBILIDAD.....	56
5.1 Introducción.....	56
5.2 Modelo COCOMO II.....	56
5.3 Características del Proyecto.....	56
5.3.1 Entradas externas.....	56
5.3.2 Salidas externas.....	57
5.3.3 Consultas externas.....	58
5.3.4 Archivos lógicos internos.....	58
5.3.5 Archivos de interfaz externos.....	58
5.3.6 Puntos de función desajustados.....	59
5.4 Cálculos de instrucciones.....	60
5.4.1 Cálculos de instrucciones.....	60
5.4.2 Cálculo de esfuerzo nominal.....	60
5.5 Ajuste del esfuerzo nominal.....	63
5.5.1 Ajuste del esfuerzo nominal clasificados en categorías.....	63
5.5.2 Cálculo del tiempo de desarrollo del software.....	64
5.5.3 Cálculo del costo total del proyecto.....	65
5.6 Resultados.....	66
5.7 Análisis de costo.....	66
5.8 Conclusiones parciales.....	66
CONCLUSIONES.....	67
RECOMENDACIONES.....	68
REFERENCIAS BIBLIOGRÁFICAS.....	69
BIBLIOGRAFÍA.....	<b>¡Error! Marcador no definido.</b>

## **INTRODUCCIÓN.**

La evolución de las tecnologías de la informática y las comunicaciones es un hecho que no pasa desapercibido producto a los inminentes cambios que han ocurrido y que continúan haciéndose para mejorar e incrementar los servicios que brindan. El salto ha sido impresionante, del año 1645 cuando Blaise Pascal construyó una máquina con ruedas dentadas destinada a la realización de operaciones de suma y resta a la actualidad que se cuenta con potentes ordenadores que realizan disímiles operaciones en cuestiones de segundos.

La Informática trajo consigo un grupo de tecnologías que son élite en el campo de la comunicación interpersonal que aunque no son tan antiguas como la propia informática, se encuentran entre las más populares y utilizadas por las personas a nivel mundial. Un ejemplo es la telefonía móvil y la Internet que se sitúan entre lo más novedoso de las tecnologías de nueva generación. Ambas surgen paralelamente como resultado de una intensa búsqueda por mejorar la comunicación entre los usuarios así como para acortar las distancias, pensada únicamente para el beneficio de grandes sectores gubernamentales y de empresas privadas. Sin embargo, el desarrollo de estos dos tipos de tecnologías llegan a un punto en el cual convergen, y es cuando la red de comunicaciones a nivel mundial se abre y sobrepasa las expectativas de sus creadores; la Internet deja de ser para uso exclusivo de la milicia y el gobierno, y se transforma en un medio de interacción social que actualmente está presente en todos los ámbitos de la vida diaria.

La Internet contribuyó al desarrollo del correo electrónico, servicio de gran utilidad en la actualidad, que posibilita el envío de mensajes haciendo uso de un ordenador o dispositivo conectado a la red, su principal ventaja radica en la capacidad permitida para los archivos adjuntos, siendo esta superior a la que se puede enviar haciendo uso de la mensajería instantánea, quien también se desarrolló gracias a las infinitas posibilidades que brinda la Internet y es una forma de comunicación en tiempo real entre dos o más personas que en la actualidad ha experimentando un aumento importante en cuanto a aceptación de los usuarios.

Por otra parte, la telefonía móvil, significó un paso de avance en la telefonía tradicional, no solo porque permite llevar el teléfono a todas partes, sino también porque agrega la posibilidad del envío de mensajes SMS (Short Message Service).

Cuba trata de no mantenerse ajeno al desarrollo que tienen a nivel global estas tecnologías, para lograr tan difícil tarea cuenta con varias instituciones que se encargan de impulsar los avances en las esferas de las comunicaciones y la informática. Una de las instituciones más destacadas en este ámbito es la Universidad de las Ciencias Informáticas (UCI).

La UCI es una universidad productiva, cuya misión es producir software y servicios informáticos (1), dentro de sus características fundamentales se destacan: la vinculación de los estudiantes al trabajo en proyectos productivos, la realización de investigaciones y formación de futuros profesionales que se encargarán del desarrollo de la rama.

En la universidad existen un grupo de servicios que posibilitan la comunicación entre los usuarios; ejemplo de estos: correo electrónico, mensajería instantánea y telefonía fija, además del servicio de mensajes cortos que no es muy usado ya que fue recientemente implementado en la universidad. La mayoría de las informaciones en la UCI se difunden haciendo uso de estos servicios producto de la gran aceptación que tienen los mismos en la comunidad universitaria.

Actualmente ocurre un problema con la recepción de la información por los usuarios, debido a que en ocasiones el medio por el que se envía la misma no se encuentra disponible o el usuario no está haciendo uso de él en ese momento, lo que trae consigo: desinformación, desconocimiento y tardía comunicación interpersonal. El hecho de que todos estos servicios se encuentran separados unos de otros, conlleva que el usuario deje de utilizar alguno, ya que le quita tiempo y le supone un esfuerzo adicional hacer uso de todos.

Hoy en día en la universidad no existe una aplicación que unifique los servicios de comunicación interpersonal y que aumente la probabilidad de que la información llegue a los destinatarios por cualquiera que haya sido la vía de envío, posibilitando que el mensaje llegue y su emisor realice un esfuerzo menor.

A partir de la situación problemática anteriormente planteada, se deriva el siguiente **problema a resolver**: ¿Cómo eliminar la desinformación y tardía comunicación en los procesos de envío de mensajes que hacen uso de los medios de comunicación interpersonal existentes en la UCI?

Por las características de la investigación, se define como **objeto de estudio** los procesos de envío de mensajes que hacen uso de los medios de comunicación interpersonal y como **campo de acción** los procesos de envío de mensajes a través de mensajería instantánea, correo electrónico, telefonía fija y SMS en la UCI.

El **objetivo general** del trabajo de diploma es desarrollar una plataforma que unifique los servicios de comunicación entre usuarios existentes en la UCI. Se definen como **objetivos específicos**:

- Realizar un estudio de los procesos de envío de mensajes existentes en la universidad.
- Modelar el sistema de integración de servicios de comunicación.
- Desarrollar una aplicación Web que permita adicionar, modificar o eliminar tanto contactos como clientes.
- Realizar pruebas al Sistema de integración de servicios de comunicación a usuarios.

Para dar solución al objetivo propuesto se definen las siguientes **tareas de investigación**:

- Identificación de los sistemas de integración de servicios de comunicación existentes en el ámbito internacional y nacional para lograr la realización de un producto que satisfaga las necesidades del cliente.
- Determinación de cuáles de los servicios de comunicación existentes en la UCI se pueden integrar en un sistema único y si son los solicitados por el cliente.
- Selección de las tecnologías y metodologías de desarrollo de software que se utilizarán en la investigación.
- Estudio del Sistema Integral de Gestión de Entidades (CEDRUX) para lograr una familiarización con dicho marco de trabajo posibilitando la realización del producto.
- Realización de un estudio de los servicios de correo electrónico, mensajería instantánea, telefonía fija y SMS, para conocer cómo se desempeñan y usarlos en el sistema a implementar.
- Identificación de las características funcionales y no funcionales del sistema solicitadas por el cliente.

A raíz de lo antes expuesto surge como **idea a defender**: con la creación de un sistema que integre los servicios de comunicación existentes en la universidad se permitirá a los usuarios de esta mitigar la desinformación y la tardía comunicación.

Para el desarrollo del trabajo se utilizaron los siguientes métodos:

#### **Métodos teóricos:**

- Análisis - Síntesis: Se utilizó para la realización de un estudio exhaustivo acerca de los servicios informáticos que permiten la comunicación interpersonal así como arribar a conclusiones de utilidad para la investigación y precisar las características del trabajo a realizar.

- Histórico - Lógico: Para determinar las tendencias actuales de los servicios existentes en la universidad y la existencia de otras aplicaciones que unifiquen los mismos.
- Modelación: Se utilizó para la realización de los modelos y diagramas presentados en la aplicación a desarrollar.

### **Métodos empíricos:**

- Entrevista: Son realizadas frecuentemente para conocer las necesidades del cliente, así como obtener las características tanto funcionales como no funcionales del sistema.

El aporte práctico del presente trabajo es una aplicación web que integre los servicios de correo electrónico, mensajería instantánea, telefonía fija y móvil, existentes en la Universidad de las Ciencias Informáticas.

Este trabajo consta de cinco capítulos en los cuales se describe todo el proceso por el que transita la investigación.

**Capítulo 1. Fundamentación Teórica:** Contiene una descripción del proceso que será automatizado y un breve estudio de los principales conceptos tratados en el trabajo. Se expone el estado del arte, las principales características de las tecnologías y herramientas seleccionadas para la solución del problema actual, así como de la metodología de desarrollo empleada.

**Capítulo 2. Exploración y Planificación:** Se hace énfasis en las características del sistema a desarrollar, la propuesta del mismo, se definen las iteraciones a realizar y la duración que presentarán, así como se describen las historias de usuarios y la estimación de esfuerzo que presentan las mismas.

**Capítulo 3. Diseño del sistema:** Se describen las Tarjetas de Cargo o Clase, Responsabilidad y Colaboración (CRC), así como los patrones de diseño y la estructura que presentará la base de datos a utilizar.

**Capítulo 4. Implementación y Pruebas:** Se expone todo lo relacionado a los procesos de implementación y pruebas del sistema, destacándose la identificación de las tareas de ingeniería y la correcta ejecución de las pruebas a realizar.

**Capítulo 5. Estudio de Factibilidad:** Se lleva a cabo un estudio del sistema para comprobar si es factible o no la realización del mismo, para ello se tiene en cuenta el factor tiempo de trabajo, así como presupuesto a utilizar.

## **CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA.**

### **1.1 Introducción.**

En el presente capítulo se expone el resultado de una investigación acerca de los servicios de telefonía fija, mensajería instantánea, correo electrónico y mensajería SMS. También un estudio sobre los sistemas que realizan la unificación de estos servicios así como las principales características de las tecnologías, metodologías y herramientas más adecuadas para desarrollar una aplicación informática de este tipo.

### **1.2 Conceptos vinculados al campo de acción.**

Con el estudio realizado se han encontrado algunos conceptos relacionados con el campo de acción, que son de vital importancia para un mejor entendimiento del contenido de la investigación. A continuación se presentan los más importantes.

#### **1.2.1 Correo electrónico.**

El correo electrónico es un servicio de red que permite a los usuarios enviar y recibir mensajes en fracciones de segundos, con textos, sonidos e imágenes. El emisor de un correo electrónico manda los mensajes a un servidor y éste, a su vez, se encarga de enviárselos al servidor del receptor. Para poder ver el correo electrónico es necesario que el receptor se conecte con su servidor. Se puede enviar el mensaje a uno o varios remitentes al mismo tiempo, con dirección visible o encriptada, con listas de distribución públicas o privadas.

#### **1.2.2 Mensajería instantánea.**

La mensajería instantánea conocida también en inglés como IM, permite conversar con otra gente mediante mensajes instantáneos, unirse a foros de debate con temas específicos, enviar ficheros y comunicarse con otros usuarios en tiempo real sin importar las distancias.

En los programas actuales, habitualmente, se envía cada frase de texto al terminarse de escribir y en algunos hasta se permite dejar mensajes aunque la otra parte no esté conectada al estilo de un contestador automático.

Los clientes de mensajería instantánea más utilizados son ICQ, Yahoo! Messenger, Windows Live Messenger, Pidgin, AIM (AOL Instant Messenger) y Google Talk.

### **1.2.3 Telefonía fija.**

La telefonía fija realiza el transporte de voz en tiempo real entre dos terminales, estando ambos, o al menos el terminal de origen, conectados a una red de telecomunicaciones. Aunque pueden concebirse usos privados de la telefonía fija (telefonía “en grupo cerrado” de usuarios), lo habitual es ligar la telefonía fija con el servicio telefónico fijo disponible al público. Este servicio asigna a cada abonado un número del plan nacional o internacional de numeración telefónica (habitualmente con vinculación geográfica) y le permite, además de efectuar y recibir llamadas nacionales e internacionales, hacer uso de una serie de servicios adicionales. (2)

### **1.2.4 Telefonía IP (Internet Protocol).**

La telefonía IP surge como una alternativa a la telefonía tradicional, brindando nuevos servicios al cliente y una serie de beneficios económicos y tecnológicos. Dicha telefonía reúne la transmisión de voz y de datos, lo que posibilita la utilización de las redes informáticas para efectuar llamadas telefónicas. Además, ésta tecnología al desarrollar una única red encargada de cursar todo tipo de comunicación, ya sea de voz, datos o video, se denomina red convergente o red multiservicios. (3)

### **1.2.5 Telefonía móvil.**

La telefonía móvil, también llamada telefonía celular, está formada por dos grandes partes: una red de comunicaciones (o red de telefonía móvil) y los terminales (o teléfonos móviles). Estos últimos desempeñan un papel fundamental en el desarrollo de esta telefonía permitiendo gracias a su portabilidad comunicarse desde casi cualquier lugar, siendo su principal función la comunicación de voz y el envío de mensajes de texto, aunque han ido adquiriendo otras funcionalidades tales como: cámaras fotográficas, calculadoras, agendas electrónicas, reproductores multimedia, entre otros.

### **1.2.6 Sistema de mensajes cortos.**

Conocido por la sigla SMS del inglés Short Message System (Sistema de Mensajes Cortos), con este servicio se puede enviar mensajes de texto desde cualquier móvil a otro, sin importar de qué compañía sea. Si el teléfono al que se envía el mensaje está apagado o fuera de cobertura el mensaje se almacena en la red y se entrega en cuanto el teléfono se conecta de nuevo a la red. (4)

### **1.3 Unificación de servicios de comunicación.**

Para la realización de una aplicación que permita el envío de mensajes a través de los servicios de telefonía fija, mensajería instantánea, correo electrónico y mensajería SMS; se analizaron otros sistemas que presentan funcionalidades similares a las que se desean implementar. Arrojando como resultado a nivel mundial:

La solución de Comunicaciones Unificadas de Microsoft la cual permite a empresas pequeñas, medianas y corporativas integrar los sistemas de comunicación (Telefonía fija y en movilidad, E-Mail, mensajería instantánea, videoconferencia y fax) dentro de las aplicaciones de negocio y gestionar de manera unificada todos los sistemas de comunicación reduciendo así los costes operativos y de mantenimiento. Mensajería Instantánea, Presencia, Audio y videoconferencia, integración con Outlook e integración de telefonía IP son algunas de las funcionalidades que la empresa Integra-T proporciona a sus clientes a través de la implantación de la solución de Comunicaciones Unificadas de Microsoft. (5)

Otro de los sistemas encontrados es el proyecto Titán presentado por Facebook el 15 de noviembre del 2010, el cual es un sistema que emplea algunos de los servicios que se desean integrar en la aplicación solicitada. La idea es que Facebook se convierta en un eslabón central en todas las comunicaciones electrónicas. Cada usuario podrá disponer de una dirección @facebook.com, y cuando alguien envíe un e-mail a esta dirección, el usuario lo recibirá en el dispositivo definido por su status. En otras palabras, al encontrarse conectado a Facebook, el mensaje será presentado como notificación de chat. Al responder, el remitente recibirá la respuesta por correo electrónico. Si el usuario no está conectado a Facebook, el mensaje será redirigido a su teléfono móvil. Así, el usuario no necesitará adaptarse a la forma en que el mensaje le haya sido enviado. (6) Dicho proyecto en estos momentos se encuentra en estado de prueba, pues no ha sido posible que todo el mundo lo pueda usar debido al enorme volumen de mensajes que se envían (más de cuatro mil millones al día), pero se trabaja para lograr su implantación total.

Y como último punto del estudio mencionar al Elastix que es una distribución de “Software Libre” de Servidor de Comunicaciones Unificadas que integra en un solo paquete algunas tecnologías clave como: VoIP PBX, Fax, Mensajería Instantánea y Email.

Elastix implementa gran parte de su funcionalidad sobre 4 programas de software muy importantes que son Asterisk, Hylafax, Openfire y Postfix. La parte del sistema operativo se basa en CentOS, una popular distribución Linux orientada a servidores. Cada uno de estos programas son desarrollados y mantenidos por diferentes compañías y comunidades. Donde está la grandeza de Elastix es en la creación de una interfaz Web común para la administración de estos servicios y la integración de los mismos de forma sumamente fácil y sencilla. (7)

Luego de realizado el análisis exhaustivo de los sistemas encontrados y observar que estos integran varios de los servicios que se desean, mas no realizan una gestión de los cliente así como los contactos que poseen cada uno, además de apreciar que en nuestro país no existe una aplicación como la que se desea implementar; se hace necesario desarrollar un sistema que unifique el envío de mensajes, ya sean de texto o voz y haga uso de los servicios que se presentan en la universidad, dándole solución a los problemas presentados en dicho centro.

#### **1.4 Sistema Integral de Gestión de Entidades (CEDRUX).**

Para el desarrollo del sistema se seleccionó el Sistema Integral de Gestión de Entidades (CEDRUX) que es una solución de software cubana desarrollada por la UCI basada en los sistemas ERP (Enterprise Resource Planning, Planeación de Recursos Empresariales). Este marco de trabajo emplea como lenguaje de programación PHP y como gestor de base de datos PostgreSQL, además de estar integrado por diferentes framework como son ExtJs, Doctrine, Zend Framework y UCID (acrónimo de Unidad de Compatibilidad Integrada de Desarrollo).

##### **➤ ExtJS Framework.**

ExtJS es una librería construida con Javascript ligera, de alto rendimiento y compatible con la mayoría de los navegadores que existen en la actualidad que permite construir páginas e interfaces web dinámicas, su potencia radica en la rica colección de componentes para el diseño del lado del cliente haciendo uso extensivo de Ajax. Entre los componentes que esta librería ofrece se encuentran cuadros de diálogo, menús, tablas editables, layouts, paneles, pestañas y todo lo necesario para construir atractivos

desarrollos. Utiliza además Dynamic Hypertext Markup Language (DHTML) y Cascading Style Sheets (CSS) Esta librería incluye:

- Componentes UI del alto performance y personalizables.
- Modelo de componentes extensibles.
- Un API fácil de usar.
- Licencias Open Source y comerciales. (8)

Esta librería ofrece muchas ventajas como la posibilidad de crear aplicaciones complejas utilizando componentes predefinidos, un manejador de layouts, la ventana flotante, existe un balance entre Cliente-Servidor, entre otros.

#### ➤ **Doctrine Framework.**

Doctrine es un mapeador de objetos relacionales (ORM) para PHP 5.2.3. En otras palabras es la capa de abstracción situada entre la base de datos y la programación. Entre las principales características de Doctrine se encuentran la posibilidad de escribir en las consultas de la base de datos, lo que proporciona a los desarrolladores una poderosa alternativa a SQL que mantiene la flexibilidad, sin la duplicación de código. Además es multiplataforma. (9)

#### ➤ **Zend Framework.**

Zend Framework se trata de un framework para desarrollo de aplicaciones Web y servicios Web con PHP, brinda soluciones para construir sitios web modernos, robustos y seguros. Además, es Open Source y trabaja con PHP 5. (10)

Entre las ventajas más visibles de Zend Framework se encuentran:

- Estandariza los procesos más frecuentes, dotándolos de gran robustez.
- Facilita el mantenimiento de las aplicaciones.
- Ofrece muchas facilidades para el acceso a recursos avanzados (Servidor Web por ejemplo) que de otro modo resultan bastante más costosos de desarrollar.
- A diferencia de otros Frameworks, es posible utilizarlo en modo "desacoplado", es decir, aquellas clases o componentes que sean necesarios en cada proyecto, sin arrastrar todo el framework detrás para cualquier pequeña necesidad.

➤ Tiene el respaldo de la propia ZEND, creadora de PHP, lo que asegura su continuidad futura tanto como la del propio lenguaje PHP. (11)

➤ **UCID Framework.**

Es el Framework encargado del trabajo con las vistas. Abarca la integración de ExtJs Framework con el sistema incluyendo el integrador de interfaz, el generador de interfaz dinámica y la impresión de documentos. Integra la iconografía, los diferentes temas de escritorio de la aplicación, el multilinguaje. (12)

### **1.5 Metodologías de desarrollo de software.**

Una metodología de desarrollo de software es un conjunto de pasos y procedimientos que deben seguirse para desarrollar un software y que este cumpla con los requerimientos del usuario. Las metodologías de desarrollo de software se pueden clasificar en: Ágiles y Robustas o Tradicionales.

Las metodologías ágiles son aquellas que desarrollan software en cortos lapsos de tiempo. Además de que exigen poca documentación y sirven para trabajar en proyectos donde los requisitos no se conocen con exactitud. Ejemplos de estas son: XP (Extreme Programming), SCRUM, entre otras.

Las metodologías robustas o tradicionales están hechas para dar solución a proyectos, con soluciones complejas y de gran tamaño, donde es esencial una documentación amplia y detallada de cada paso en el desarrollo del software. Ejemplos de estas son: Microsoft Solution Framework (MSF), Rational Unified Process (RUP), entre otras.

#### **Metodología de Desarrollo de Software a utilizar en el desarrollo del sistema.**

Extreme Programming (XP) es la metodología que se decidió utilizar gracias a las características que presenta las cuales se adaptan en gran medida a las necesidades presentadas en el proyecto así como a las condiciones de trabajo. Es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores y propiciando un buen clima de trabajo donde el software es liberado en entregas frecuentes tan pronto como sea posible. Se basa en realimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios. Está diseñada para mitigar los riesgos en proyectos

con corto tiempo de entrega planteado y con continuos cambios de requerimientos. Es una de las metodologías de desarrollo de software más exitosas en la actualidad y su fortaleza radica en tener como parte del equipo, al usuario final, pues es uno de los requisitos para llegar al éxito del proyecto.

Extreme Programming es un conjunto de buenas prácticas para la ejecución de proyectos de software. Está basado en un conjunto de reglas simples que agilizan y simplifican el desarrollo de software en uno o varios equipos de desarrollo, con los siguientes objetivos:

- Adaptarse a los cambios de requerimientos, y ejecutarlos satisfactoriamente.
- Evitar retrasos en los proyectos basados en la comunicación, la colectividad y el trabajo en equipo.
- Mejorar la productividad y el desempeño del equipo, asegurando el bienestar del mismo.
- Incrementar y asegurar la calidad del sistema.
- Involucrar al cliente y/o usuario final del sistema para alcanzar la satisfacción del mismo. (13)

## **1.6 Herramientas.**

### **1.6.1 Sistemas gestores de bases de datos.**

Un Sistema Gestor de Bases de Datos (SGBD) es una colección de programas cuyo objetivo es servir de interfaz entre la base de datos, el usuario y las aplicaciones, además de permitir crear y mantener una Base de datos, asegurando su integridad, confidencialidad y seguridad. Algunos ejemplos de SGBD son Oracle, DB2, PostgreSQL, MySQL, MS SQL Server, etc.

Un SGBD debe permitir:

- Definir una base de datos: especificar tipos, estructuras y restricciones de datos.
- Construir la base de datos: guardar los datos en algún medio controlado por el mismo SGBD
- Manipular la base de datos: realizar consultas, actualizarla, generar informes. (14)

### **Sistemas gestores de bases de datos a utilizar.**

Para el sistema a implementar se ha decidido la utilización de PostgreSQL en su versión 8.3, el cual es un sistema de gestión de bases de datos objeto-relacional, es una derivación libre y su código fuente completo está disponible. Utiliza el lenguaje SQL92/SQL99 y su desarrollo es realizado por un equipo de desarrolladores en su mayoría voluntarios extendido por todo el mundo, que se comunican vía Internet. Se trata de un proyecto comunitario y no está controlado por compañía alguna.

Sus características técnicas le hacen una de las bases de datos más potentes y robustas del mercado. Funciona muy bien con grandes cantidades de datos y una alta concurrencia de usuarios accediendo a la vez al sistema. Soporta distintos tipos de datos: además del soporte para los tipos base, también soporta datos de tipo fecha, monetarios, elementos gráficos, cadenas de bits, etc. También permite la creación de tipos propios. 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. Puede ser utilizado, modificado y distribuido por cualquiera gratuitamente, para cualquier propósito ya sea con fines privados, comerciales o académicos. Puede ser utilizado en los principales sistemas operativos: Linux, Unix, Mac OS, Beos, Windows y presenta soporte para los lenguajes más populares: PHP, C, C++, Perl, Python, entre otros.

### **Ventajas de PostgreSQL.**

PostgreSQL se caracteriza por ser un sistema estable, de alto rendimiento, gran flexibilidad ya que funciona en la mayoría de los sistemas Unix, tiene características que permiten extender fácilmente el sistema, además de instalación ilimitada ya que con PostgreSQL, nadie puede demandarlo por violar acuerdos de licencia, puesto que no hay costo asociado a la licencia del software. Ahorros considerables en costos de operación: PostgreSQL ha sido diseñado y creado para tener un mantenimiento y ajuste mucho menor que otros productos, conservando todas las características, estabilidad y rendimiento. El código fuente está disponible para todos sin costo. Si su equipo necesita extender o personalizar PostgreSQL de alguna manera, pueden hacerlo con un mínimo esfuerzo, sin costos adicionales. Existen varias herramientas gráficas de alta calidad para administrar las bases de datos (pgAdmin, pgAccess) y para hacer diseño de bases de datos (Tora, Data Architect). (15)

### **1.6.2 Lenguajes de programación web.**

Un lenguaje de programación es un idioma artificial diseñado para expresar acciones que deben ser ejecutadas por un ordenador, es el medio de comunicación entre el programador y la máquina, como todo lenguaje está formado por un conjunto de símbolos y reglas sintácticas y semánticas que definen su estructura y significado. (16)

La programación web permite la creación de sitios dinámicos en Internet. Esto se consigue generando los contenidos del sitio a través de una base de datos mediante lenguajes de script como pueden ser PHP, ASP o ASP.NET. Dominando la programación web podrá crear sitios dinámicos como periódicos digitales o tiendas virtuales. (17)

### **Lenguajes de programación web a utilizar.**

En el desarrollo de la aplicación se ha decidido la utilización del lenguaje de programación Hypertext Pre-Processor (PHP) del lado del servidor, mientras que del lado del cliente se propone utilizar los lenguajes HyperText Markup Language (HTML) y Java script.

#### **➤ PHP.**

El lenguaje de programación a utilizar es PHP debido a que es el establecido según el marco de trabajo CEDRUX con el que se decidió crear el sistema.

PHP es el lenguaje del lado servidor más extendido en la web. Nacido en 1994, se trata de un lenguaje de creación relativamente reciente, aunque con la rapidez con la que evoluciona Internet parezca que ha existido toda la vida. Es un lenguaje que ha tenido una gran aceptación en la comunidad de desarrolladores, debido a la potencia y simplicidad que lo caracterizan, así como al soporte generalizado en la mayoría de los servidores.

PHP permite embeber sus pequeños fragmentos de código dentro de la página HTML y realizar determinadas acciones de una forma fácil y eficaz, combinando los conocimientos sobre el desarrollo HTML. Es decir, con PHP se puede escribir scripts dentro del código HTML. Por otra parte, ofrece un sinfín de funciones para la explotación de bases de datos de una manera llana, sin complicaciones y soporta la programación orientada a objetos, es multiplataforma, permite la conexión con distintos gestores de bases de datos y es libre. Existen varios lenguajes de programación que cumplen con estas características como son: Python, PHP, Java, entre otros.

#### **➤ HTML.**

Hypertext Markup Language, o simplemente HTML, es un lenguaje de programación muy sencillo que se utiliza para crear los textos y las páginas web. Es justamente un lenguaje que se basa en las marcas para crear los hipertextos. Está compuesto por etiquetas que definen la estructura y el formato del documento

que verá el usuario a través de la web. Esas etiquetas son leídas por el navegador o visualizador, es decir; el programa que se utiliza para navegar, y que es el que ejecuta las funciones creadas en HTML.

Una de las características es que, además del texto (que es el contenido en sí), este lenguaje permite que se creen enlaces entre distintas partes del mismo documento o entre distintas fuentes de información a través de hiperenlaces o hipervínculos, e incluso insertar otros elementos como imágenes y sonidos.

HTML no presenta ningún compilador, por lo tanto, algún error de sintaxis que se presente, éste no lo detectará y se visualizará en la forma como éste lo entienda. Una de las principales ventajas que tiene es la gran variedad de navegadores y exploradores que lo soportan, convirtiéndose en el formato más usado para la transmisión de documentos electrónicos a través de Internet.

Actualmente existe un gran número de navegadores por lo que a veces se complica que todos puedan interpretar el código HTML de la misma manera. Por esa razón es que quienes crean las páginas, chequean que estas puedan ser leídas al menos por los navegadores más populares.

#### ➤ **Javascript.**

JavaScript es un lenguaje de programación interpretado, multiplataforma y parcialmente orientado a objetos, basado en prototipos, dinámico y desarrollado por Netscape para incrementar las funcionalidades del lenguaje HTML. Permite crear efectos especiales para interactuar con el visitante y funciones básicas que son soportadas por la mayoría de los navegadores que se utilizan diariamente.

Ventajas de JavaScript:

- Permite elaborar aplicaciones web que simulen características de aplicaciones escritorios.
- Es un código “interpretado” por el cliente.
- 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 cualquier usuario aún estando protegido con las leyes del copyright. (18)

Uno de los inconvenientes que presenta el lenguaje es que requiere estar activado en los navegadores para que su uso sea posible por estos. Por otro lado, no le proporciona al programador control total de la página web. En ocasiones los desarrolladores deben realizar diferentes implementaciones de código

JavaScript para sus aplicaciones, debido a que el código no es interpretado de igual modo por todos los navegadores.

### **1.6.3 Servidores para aplicaciones web.**

Un servidor web es un programa que está diseñado para transferir hipertextos, páginas web o páginas HTML: textos complejos con enlaces, figuras, formularios, botones y objetos incrustados como animaciones o reproductores de música. El programa implementa el protocolo HTTP (HyperText Transfer Protocol) que pertenece a la capa de aplicación del modelo OSI.

El servidor web es un programa que se aloja en un servidor que escucha peticiones HTTP las cuales son atendidas y procesadas por este programa. Atendiendo a las peticiones hechas, gestiona las páginas web o ejecuta códigos en el servidor para dar respuesta, ya sea información o un mensaje detallado de error.

#### **Servidor para aplicación web a utilizar.**

Para el sistema a implementar se ha decidido la utilización del servidor Apache, el cual se desarrolló dentro del proyecto HTTP Server de la Apache Software Foundation. Apache presenta entre otras características informes de errores altamente configurables, índice de directorios, un directorio de alias; negociación de contenidos, gestión de recursos multi hilo, integración de imágenes del lado del servidor; reescritura de las URL; comprobación de la ortografía de las URL y manuales online, bases de datos de autenticación y negociado de contenido, aunque carece de una interfaz gráfica que ayude en su configuración. Tiene amplia aceptación en la red por ser altamente configurable de diseño modular lo que aumenta sus capacidades, es el más usado, empleado en el 48% de los sitios web en el mundo. Es un servidor de distribución libre, código abierto (Open source) y multiplataforma.

Se decidió utilizar el servidor Apache para publicar la aplicación por todas las bondades que ofrece este software, por ser extensible, popular y brindar las siguientes ventajas:

- Ayuda en la mejora del posicionamiento.
- Es un software libre.
- Multiplataforma.
- Open Source, modular y presenta mensajes de error altamente configurables.

#### **Ventajas de Apache.**

- Altamente configurable: Para los que saben, casi cualquier cosa que se necesite hacer con un servidor web se puede hacer con Apache.
- Estabilidad: Ha probado su estabilidad y funcionamiento en una inmensa cantidad de proyectos.
- Independencia de la plataforma: Está disponible en variedad de plataformas.
- Código abierto: Esto permite por ejemplo, producir binarios para plataformas en las que no hay un binario oficial o compilarlo para un mejor rendimiento en una plataforma específica. (19)

#### **1.6.4 Entorno de Desarrollo Integrado de Desarrollo.**

Un Entorno de Desarrollo Integrado ó IDE de Desarrollo es un entorno de programación empaquetado como un programa de aplicación, en otras palabras, consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica. Estos pueden ser aplicaciones por sí solos o pueden formar parte de aplicaciones existentes. Un IDE de Desarrollo favorece el desarrollo rápido y factible debido a que provee un marco de trabajo completamente amigable.

#### **IDE de desarrollo seleccionado.**

Para el desarrollo de la aplicación se ha decidido utilizar el Zend Studio Neon ya que se trata de un programa de Zend que es uno de los mayores impulsores de PHP.

Zend Studio Neon es una versión de Zend está basado en el proyecto PHP Development Tools (PDT) de Eclipse. Visualmente es más agradable, su vista y funcionalidades están adaptadas para que sean similares a las anteriores versiones de Zend Studio. Entre sus principales ventajas se presenta:

- Presenta un buen depurador.
- Permite agilizar el trabajo de desarrollo.
- Posee infinitas opciones que permiten un desarrollo profesional de las aplicaciones.

#### **1.7 Conclusiones parciales.**

Durante el capítulo se realizó un estudio sobre los servicios de comunicación existentes en la actualidad con funcionalidades similares a la aplicación que se desea desarrollar, además de definir las tecnologías, herramientas y metodología de desarrollo de software a utilizar para la implementación. El aprovechamiento de estas tecnologías, la correcta selección y uso de las herramientas permitirá lograr un sistema que cumpla los requisitos propuestos y satisfaga las expectativas del cliente.

## **CAPÍTULO 2: EXPLORACIÓN Y PLANIFICACIÓN.**

### **2.1 Introducción.**

En el presente capítulo se hace alusión a todo lo referente con las fases de exploración y planificación que son propias de la metodología de desarrollo elegida para la implementación de la aplicación, se realizará un análisis acerca de las características con las que va a contar el sistema y se mostrará una propuesta de solución para dar respuesta a la problemática que dio origen al mismo. Además de tratar los artefactos generados en cada iteración del trabajo, una estimación del período de duración de cada fase y una propuesta de la arquitectura que presentará el producto.

### **2.2 Características del sistema**

#### **2.2.1 Objeto de Automatización.**

El envío de mensajes es un proceso que se encuentra automatizado a través de múltiples servicios, donde cada cual realiza esta acción haciendo uso de una plataforma independiente, por lo que si se desea enviar una información por varias vías podría llegar a ser una tarea engorrosa y a su vez suponerle un esfuerzo mayor al cliente, por lo tanto, se decidió desarrollar una aplicación que permita a los usuarios haciendo uso de una misma plataforma realizar el envío de informaciones posibilitando que estas lleguen por todas las vías disponibles en la UCI, dígame correo electrónico, mensajería instantánea, telefonía y mensajería SMS.

#### **2.2.2 Propuesta del Sistema.**

El Sistema de integración de servicios de comunicación a usuarios tiene como propósito unificar los servicios de envío de mensajes que se encuentran vigentes en la UCI. Este sistema se plantea como una aplicación web a la cual puede acceder cualquier usuario previamente autenticado. La aplicación permitirá el envío de mensajes vía correo electrónico, mensajería instantánea, telefonía fija y servicio de mensajes cortos, así como dará la posibilidad de gestionar los clientes y contactos que posee cada usuario, con esto se facilitará el trabajo de los mismos que ya no tendrán que utilizar cada una de las plataformas que posibilitan el envío de mensajes de forma individual sino que se presentarán todos en un único sistema.

Para realizar el envío se utilizan librerías como: Zend\_Mail, que forma parte de Zend Framework y se usan para enviar el correo electrónico, XMPPHP que es una librería jabber de PHP que se usa para enviar el mensaje instantáneo, Soap\_Client que también es una librería de PHP que facilita la conexión con el servidor web así como SSH2 para conectarse con el servidor Asterisk.

Para el correcto funcionamiento de este sistema se necesita un servidor Asterisk, un servidor web que brinde los servicios de mensajes cortos, así como un servidor de mensajería instantánea y otro de correo electrónico.

A continuación se muestra un diagrama de clases el cual no es objetivo de la metodología seleccionada pero ayudará a propiciar un mejor entendimiento del sistema desarrollado.

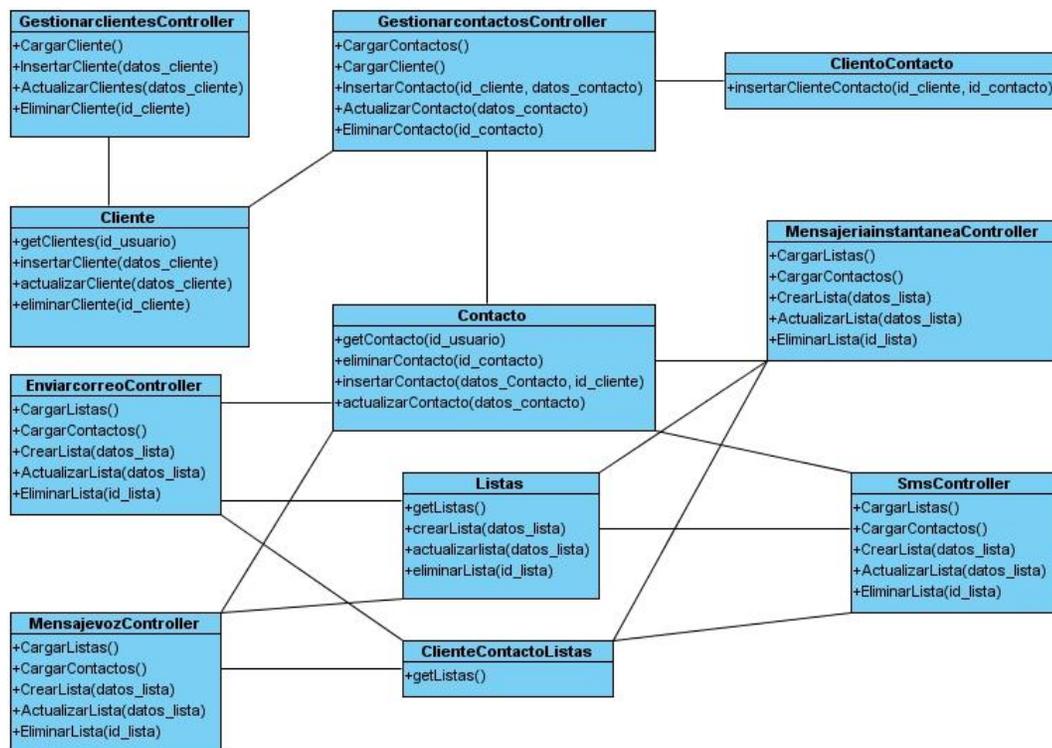


Imagen 1 Diagrama de Clases

**Seguridad en la aplicación implementada:**

La seguridad en la aplicación web se maneja gracias a la ayuda del marco de trabajo CEDRUX, este se compone por módulos y subsistemas. De cada uno de estos sistemas se gestionan las funcionalidades que brindan y de estas sus acciones. De igual forma se gestionan los roles, servicios consumidos y

ofertados a otros subsistemas, los usuarios y los permisos que tienen cada uno de estos dentro de las aplicaciones.

De esta forma se puede permitir o denegar el acceso tan profundamente como se desee, quedando definido un sistema de administración centralizada de seguridad que puede ser configurado para una política restrictiva tan severa como se desee. Cada usuario tendrá su perfil y el administrador podrá configurarlo según entienda los permisos que deben tener.

### **2.2.3 Características no funcionales del sistema.**

Las características no funcionales hacen relación a las características del sistema que aplican de manera general como un todo, más que a rasgos particulares del mismo, para la realización de la aplicación se identificaron las siguientes características no funcionales.

➤ **Apariencia o interfaz externa.**

No debe tener animaciones ni imágenes pesadas que comprometan la rapidez de la aplicación. El diseño deberá ser atractivo con colores que no resulten agotadores a la vista.

➤ **Rendimiento.**

Para el funcionamiento óptimo se seguirán técnicas de elaboración de sitios web, que faciliten el acceso rápido a las páginas. La eficiencia del sistema estará dada en gran medida por el aprovechamiento de los recursos y la propuesta debe ser rápida con un tiempo de respuesta mínimo.

➤ **Seguridad.**

- **Confiabilidad.**

Deben ser exactas y seguras las salidas del software, además de realizarse salvadas de seguridad para no perder la información en caso de algún fallo. La información manejada por el sistema debe estar protegida de acceso no autorizado y divulgación, los administradores del sistema tienen todos los privilegios en la aplicación y los usuarios del sistema podrán enviar los mensajes así como gestionar sus contactos.

- **Integridad.**

La información manejada por el sistema debe ser objeto a una cuidadosa protección contra la corrupción y estados de inconsistencia.

- **Disponibilidad.**

La aplicación deberá estar disponible en todo momento para aquellas personas con acceso y los mecanismos utilizados para lograr la seguridad no deben ocultar o retrasar a los usuarios para obtener los servicios deseados en un momento dado.

➤ **Soporte.**

El sistema deberá estar bien documentado para cuando exista la necesidad de realizar cambios, mejoras, solucionar algún problema o agregar nuevas funcionalidades.

➤ **Legales.**

El sistema es propiedad de la Universidad de las Ciencias Informáticas.

➤ **Portabilidad.**

El sistema podrá ser accedido desde cualquier sistema operativo. La aplicación, el servidor de Base de Datos y demás servidores pueden estar en la misma máquina conformando una arquitectura centralizada, aunque es recomendable tener una arquitectura distribuida con los servidores en máquinas diferentes.

➤ **Software.**

En las computadoras de los usuarios solo se requiere un navegador Web Mozilla Firefox 2.0 u otro que brinde soporte para DOM 2.0 y Java Script, además de cualquier distribución de Linux o Windows. En el caso del servidor de aplicaciones web se deberá tener instalado cualquier versión igual o superior al Apache 2.0 y como servidor de base de datos se requiere tener instalado PostgreSQL 8.3.

➤ **Hardware.**

En el cliente se requiere una máquina con 256 MB de RAM como mínimo, un procesador igual o mayor a 1.40 GHZ y una tarjeta de red, todos los servidores deben tener 1GB de RAM, 160 GB de disco duro mínimo, además deben de tener UPS, un lector de DVD o CD y una tarjeta de red.

### 2.3 Arquitectura.

La arquitectura de software es, a grandes rasgos, una vista del sistema que incluye los componentes principales del mismo, la conducta de esos componentes según se percibe desde el resto del sistema y las formas en que los componentes interactúan y se coordinan para alcanzar la misión del sistema. La vista arquitectónica es una vista abstracta, aportando el más alto nivel de comprensión y la supresión o diferimiento del detalle inherente a la mayor parte de las abstracciones. (20)

### 2.3.1 Propuesta de Arquitectura del Sistema.

Un usuario sentado frente a un ordenador accede a la vista del sistema, que es la capa de presentación o interfaz del mismo, esta se conecta con la controladora y esta a su vez con el modelo siendo este el encargado de actualizar y obtener los datos de la Base de Datos. La capa controladora, sería la encargada de indicarle a la vista las actualizaciones a realizar utilizando los cambios hechos en el modelo; además de controlar la conexión a establecer con los distintos servidores.

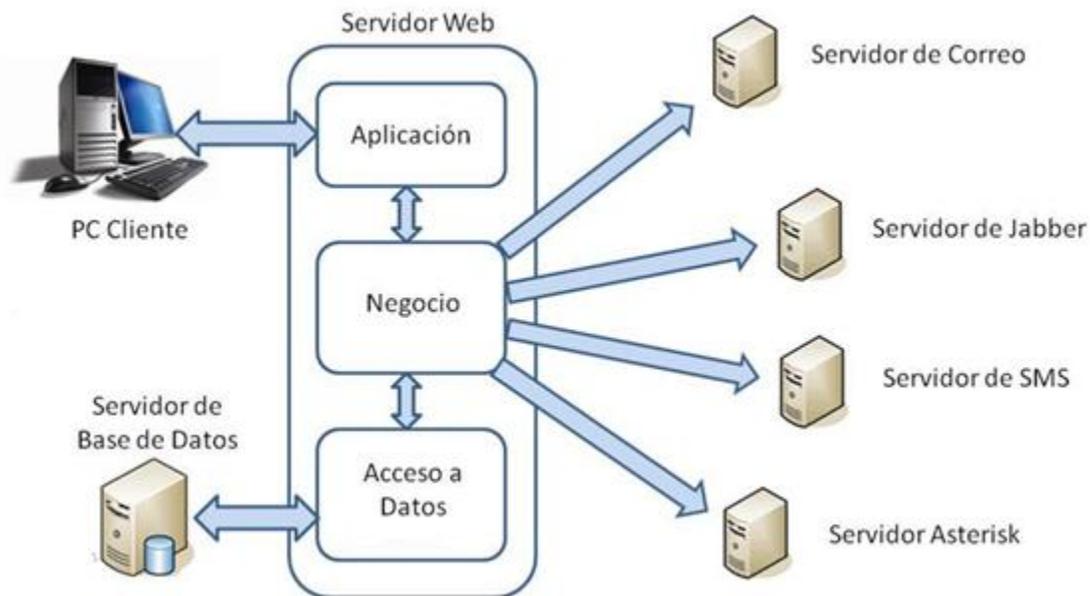


Imagen 2 Arquitectura Propuesta.

### 2.3.2 Arquitectura Cliente Servidor.

La arquitectura cliente-servidor se divide en dos partes claramente diferenciadas, la primera es la parte del servidor y la segunda la de un conjunto de clientes. Normalmente, el servidor es una máquina bastante potente que actúa de depósito de datos y funciona como un sistema gestor de base de datos (SGBD). Mientras que los clientes suelen ser estaciones de trabajo que solicitan varios servicios al servidor, ambas partes deben estar conectadas entre sí mediante una red.

Este tipo de arquitectura es muy utilizada en la actualidad, debido a que es la más avanzada y la que mejor ha evolucionado en estos últimos años. El modelo cliente/servidor se recomienda, en particular, para redes que requieran un alto grado de fiabilidad. Las principales ventajas son:

- Recursos centralizados: Debido a que el servidor es el centro de la red, puede administrar los recursos que son comunes a todos los usuarios, por ejemplo: una base de datos centralizada se utilizaría para evitar problemas provocados por datos contradictorios y redundantes.
- Seguridad mejorada: La cantidad de puntos de entrada que permite el acceso a los datos no es importante.
- Administración al nivel del servidor: Los clientes no desempeñan un papel importante en este modelo, requieren menos administración.
- Red escalable: Gracias a esta arquitectura, es posible quitar o agregar clientes sin afectar el funcionamiento de la red y sin la necesidad de realizar mayores modificaciones. (21)



**Imagen 3 Arquitectura Cliente Servidor.**

### 2.3.3 Patrones de arquitectura.

Aquellos que expresan un esquema organizativo estructural fundamental para sistemas de software.

#### **Modelo-Vista-Controlador.**

El MVC está formado por tres niveles:

- Modelo: Nivel que representa la lógica del negocio, es la información que maneja la aplicación.
- Vista: Transforma el modelo en una página web que le permite al usuario interactuar con la misma.
- Controlador: Nivel encargado de procesar las interacciones del usuario con la aplicación y realizar los cambios necesarios en el modelo o la vista.

Este patrón tiene como objetivo principal separar la lógica de negocio de la interfaz, lo que permite una mayor robustez de los sistemas que son desarrollados con el MVC. Este estilo tiene soportes de vistas

múltiples, dado que las vistas se hallan separadas del modelo y no hay dependencia directa del modelo con respecto a la vista, la interfaz de usuario puede mostrar múltiples vistas de los mismos datos simultáneamente. Entre las ventajas más notables del patrón se encuentran:

- La conexión entre el Modelo y sus Vistas es totalmente dinámica, la misma se produce en tiempo de ejecución.
- Permite implementar los componentes por separado gracias a la clara separación que existe entre ellos.
- Posee un API bien definido, lo que permite reemplazar el Modelo, la Vista o el Controlador sin aparente dificultad.

A continuación se brinda una imagen que muestra las relaciones entre el modelo, la vista y el controlador:



**Imagen 4 Modelo-Vista-Controlador.**

## 2.4 Exploración.

La Fase de Exploración es la primera con la que cuenta la metodología XP, en esta fase los clientes plantean a grandes rasgos las historias de usuario que son de interés para la entrega del producto. Al mismo tiempo, el equipo de desarrollo se familiariza con las herramientas, tecnologías y prácticas que se utilizarán para el desarrollo de la aplicación. La duración de esta fase depende principalmente de la familiaridad que tengan los programadores con la tecnología.

### 2.4.1 Flujo de procesos del sistema propuesto.

El sistema cuenta con un proceso principal, en el cual el usuario debe autenticarse, el sistema confirma la autenticación, si no es correcta el sistema posibilita que vuelva a autenticarse, si por el contrario es correcta el usuario seleccionará la vía que va a utilizar para enviar el mensaje, que puede ser Enviar

Mensaje E-mail, Enviar Mensaje Instantáneo, Enviar Mensaje SMS o Enviar Mensaje de Voz; siendo estos sub-procesos de este proceso principal. Los diagramas de los sub-procesos y el proceso principal se pueden observar en el Anexo 1.

**2.4.2 Personas relacionadas con el sistema.**

**Tabla 1 Personas relacionadas con el sistema.**

Personas Relacionadas con el Sistema	Justificación.
Usuario del Sistema	Es la persona que se encarga de insertar, eliminar o modificar los datos de sus clientes, contacto, listas de contactos, así como de redactar los mensajes que desea enviar por cualquiera de las diferentes vías disponibles.
Administrador del Sistema	Además de poder realizar las mismas tareas de un usuario del sistema, es el responsable de eliminar y asignar los roles a los usuarios registrados.

**2.4.3 Historias de Usuarios.**

Las historias de usuario son las técnicas utilizadas en XP para especificar los requisitos del sistema, en estas el cliente describe brevemente las características que debe poseer, se escriben en forma concreta y lo más entendibles posibles o sea en el lenguaje de los usuarios, desde su perspectiva aunque los desarrolladores pueden brindar también su ayuda en la identificación de las mismas. Durante la fase de exploración se identificaron 22 historias de usuario, las cuales se describen a continuación.

**Tabla 2 Adicionar cliente.**

Historia de Usuario	
Número:1	Usuario: Usuario del sistema
Nombre historia: Adicionar cliente	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alto
Puntos estimados: 3/5	Iteración asignada:1

<b>Programador responsable:</b> Odaimys Rodríguez – Claudia Rodríguez
<b>Descripción:</b> El usuario introducirá en la aplicación web los datos del cliente (nombre, 1er apellido, 2do apellido, sexo, edad) y pulsará el botón “Crear Cliente” mostrándose un mensaje de información notificando que la acción se ejecutó correctamente.
<b>Observaciones:</b> <ul style="list-style-type: none"> <li>- Los campos obligatorios son solamente: “nombre” y “1er apellido”.</li> <li>- Los datos insertados deben tener en caso de “edad” 3 caracteres máximos.</li> <li>- En caso de haber algún problema con los datos del cliente (como es la falta de un campo obligatorio) se marcará el campo en el que se ha cometido el error con una línea de color rojo, y al pasar el mouse sobre el mismo se mostrará una notificación con la descripción del error.</li> </ul>

**Tabla 3 Modificar cliente.**

<b>Historia de Usuario</b>	
<b>Número:</b> 2	<b>Usuario:</b> Usuario del sistema
<b>Nombre historia:</b> Modificar cliente	
<b>Prioridad en negocio:</b> Alta	<b>Riesgo en desarrollo:</b> Alto
<b>Puntos estimados:</b> 2/5	<b>Iteración asignada:</b> 1
<b>Programador responsable:</b> Odaimys Rodríguez – Claudia Rodríguez	
<b>Descripción:</b> El usuario tendrá la posibilidad de modificar en la aplicación web los datos del cliente seleccionado (nombre, 1er apellido, 2do apellido, sexo, edad) si no se selecciona ningún cliente el botón “Modificar Cliente” debe estar inhabilitado. Luego de seleccionar el cliente y marcar el botón “Guardar” se muestra un mensaje de información notificando que la acción se ejecutó correctamente.	
<b>Observaciones:</b> <ul style="list-style-type: none"> <li>- Solo se habilitará el botón “Modificar Cliente” si existe un cliente seleccionado.</li> </ul>	

- En caso de haber algún problema con los datos del cliente (como es la falta de un campo obligatorio) se marcará el campo en el que se ha cometido el error con una línea de color rojo, y al pasar el cursor sobre el mismo se mostrará una notificación con la descripción del error.

**Tabla 4 Eliminar cliente.**

Historia de Usuario	
<b>Número:</b> 3	<b>Usuario:</b> Usuario del sistema
<b>Nombre historia:</b> Eliminar cliente	
<b>Prioridad en negocio:</b> Alta	<b>Riesgo en desarrollo:</b> Alto
<b>Puntos estimados:</b> 2/5	<b>Iteración asignada:</b> 1
<b>Programador responsable:</b> Odaimys Rodríguez – Claudia Rodríguez	
<p><b>Descripción:</b> El usuario tendrá la posibilidad de eliminar de la aplicación web los datos del cliente seleccionado (nombre, 1er apellido, 2do apellido, sexo, edad) si no se selecciona ningún cliente el botón “Eliminar Cliente” debe estar inhabilitado. Luego de seleccionar el cliente y marcar el botón “Eliminar Cliente” se mostrará un mensaje para confirmar la eliminación, y una vez seleccionada la opción de confirmación se muestra un mensaje de información notificando que la acción se ejecutó correctamente.</p>	
<p><b>Observaciones:</b></p> <p>- Solo se habilitará el botón “Eliminar Cliente” si existe un cliente seleccionado.</p>	

**Tabla 5 Adicionar contacto.**

Historia de Usuario	
<b>Número:</b> 4	<b>Usuario:</b> Usuario
<b>Nombre historia:</b> Adicionar contacto.	
<b>Prioridad en negocio:</b> Alta	<b>Riesgo en desarrollo:</b> Alto

<b>Puntos estimados:</b> 3/5	<b>Iteración asignada:</b> 1
<b>Programador responsable:</b> Odaimys Rodríguez – Claudia Rodríguez	
<p><b>Descripción:</b> El usuario selecciona el botón “Crear Contacto”, mostrándose una ventana con todos sus clientes y luego de elegir al que desea adicionarle un nuevo contacto y pulsar el botón “Seleccionar” aparece una nueva ventana en la que se introducirán los datos del contacto (tipo de contacto, contacto, descripción) además de mostrarse los datos del cliente que se seleccionó. Luego de presionar el botón “Crear” se mostrará un mensaje de información notificando que la acción se ejecutó correctamente.</p>	
<p><b>Observaciones:</b></p> <ul style="list-style-type: none"> <li>- Los campos obligatorios son: “tipo de contacto” y “contacto”.</li> <li>- En caso de haber algún problema con los datos del contacto (como es la falta de un campo obligatorio) se marcará el campo en el que se ha cometido el error con una línea de color rojo, y al pasar el cursor sobre el mismo se mostrará una notificación con la descripción del error.</li> </ul>	

**Tabla 6 Modificar contacto.**

<b>Historia de Usuario</b>	
<b>Número:</b> 5	<b>Usuario:</b> Usuario
<b>Nombre historia:</b> Modificar contacto	
<b>Prioridad en negocio:</b> Alta	<b>Riesgo en desarrollo:</b> Alto
<b>Puntos estimados:</b> 2/5	<b>Iteración asignada:</b> 1
<b>Programador responsable:</b> Odaimys Rodríguez – Claudia Rodríguez	
<p><b>Descripción:</b> El usuario tendrá la posibilidad de modificar en la aplicación web los datos del contacto seleccionado (tipo de contacto, contacto, descripción) si no se selecciona ningún contacto el botón “Modificar Contacto” debe estar inhabilitado. Luego de seleccionar el contacto y marcar la opción “Guardar” se muestra un mensaje de información como que la acción se ejecutó correctamente.</p>	

**Observaciones:**

- Solo se habilitará el botón “Modificar Contacto” si existe un contacto seleccionado.
- En caso de haber algún problema con los datos del contacto (como es la falta de un campo obligatorio) se marcará el campo en el que se ha cometido el error con una línea de color rojo, y al pasar el cursor sobre el mismo se mostrará una notificación con la descripción del error.

**Tabla 7 Eliminar contacto.**

Historia de Usuario	
<b>Número:</b> 6	<b>Usuario:</b> Usuario
<b>Nombre historia:</b> Eliminar contacto.	
<b>Prioridad en negocio:</b> Alta	<b>Riesgo en desarrollo:</b> Alto
<b>Puntos estimados:</b> 2/5	<b>Iteración asignada:</b> 1
<b>Programador responsable:</b> Odaimys Rodríguez – Claudia Rodríguez	
<b>Descripción:</b> El usuario tendrá la posibilidad de eliminar de la aplicación web los datos del contacto seleccionado (tipo de contacto, contacto, descripción) si no se selecciona ningún contacto el botón “Eliminar Contacto” debe estar inhabilitado. Luego de seleccionar el contacto y marcar el botón “Eliminar Contacto” se mostrará un mensaje para confirmar la eliminación, y una vez seleccionada la opción de confirmación se muestra un mensaje de información notificando que la acción se ejecutó correctamente.	
<b>Observaciones:</b>	
- Solo se habilitará el botón “Eliminar Contacto” si existe un contacto seleccionado.	

**Tabla 8 Enviar correo masivo.**

Historia de Usuario	
<b>Número:</b> 7	<b>Usuario:</b> Usuarios del sistema

<b>Nombre historia:</b> Enviar correo masivo	
<b>Prioridad en negocio:</b> Alta	<b>Riesgo en desarrollo:</b> Medio
<b>Puntos estimados:</b> 1	<b>Iteración asignada:</b> 2
<b>Programador responsable:</b> Odaimys Rodríguez – Claudia Rodríguez	
<p><b>Descripción:</b> Si el usuario realiza la acción de pulsar el botón “Nuevo Correo” se le mostrará una ventana con las listas de correo y los contactos de los cuales dispone, luego de seleccionar a los que desea enviar el mensaje, se mostrará una nueva ventana para que introduzca los datos del correo (asunto, cuerpo del correo), finalizando el envío al presionar el botón “Enviar”. Finalmente, se muestra un mensaje de información notificando que la acción se ejecutó correctamente.</p>	
<p><b>Observaciones:</b> - Solo se habilitará el botón “Seleccionar” si existe un contacto o lista de contacto seleccionada.</p>	

Tabla 9 Crear lista de correos.

Historia de Usuario	
<b>Número:</b> 8	<b>Usuario:</b> Usuarios del sistema
<b>Nombre historia:</b> Crear lista de correos	
<b>Prioridad en negocio:</b> Media	<b>Riesgo en desarrollo:</b> Medio
<b>Puntos estimados:</b> 4/5	<b>Iteración asignada:</b> 2
<b>Programador responsable:</b> Odaimys Rodríguez – Claudia Rodríguez	
<p><b>Descripción:</b> El usuario presiona el botón “Crear Lista Correo” que muestra una ventana para introducir los datos de la lista (nombre) y presiona “Aceptar”, lo cual origina una nueva ventana que permite elegir los contactos que formarán parte de dicha lista, para realizar esta acción basta con seleccionar el contacto y pulsar el botón “Agregar” o “Quitar” según se desee, finaliza la creación presionando el botón “Crear”</p>	

que muestra un mensaje de información notificando que la acción se ejecutó correctamente.

**Observaciones:**

- El campo “nombre” de la lista es obligatorio.
- En caso de haber algún problema con los datos de la lista (como es la falta de un campo obligatorio) se marcará el campo en el que se ha cometido el error con una línea de color rojo, y al pasar el cursor sobre el mismo se mostrará una notificación con la descripción del error.
- Solo se habilitarán los botones “Agregar” y “Quitar” si existe un contacto seleccionado en sus respectivos espacios.

**Tabla 10 Modificar lista de correos.**

Historia de Usuario	
<b>Número:</b> 9	<b>Usuario:</b> Usuarios del sistema
<b>Nombre historia:</b> Modificar lista de correos	
<b>Prioridad en negocio:</b> Media	<b>Riesgo en desarrollo:</b> Medio
<b>Puntos estimados:</b> 3/5	<b>Iteración asignada:</b> 2
<b>Programador responsable:</b> Odaimys Rodríguez – Claudia Rodríguez	
<p><b>Descripción:</b> El usuario presiona el botón “Modificar Lista Correo” que muestra una ventana con los contactos que posee esa lista y los que se le pueden añadir, para realizar estas acciones basta con seleccionar el contacto y pulsar el botón “Agregar” o “Quitar” según convenga, finaliza la modificación presionando el botón “Guardar” que muestra un mensaje de información notificando que la acción se ejecutó correctamente.</p>	
<p><b>Observaciones:</b></p> <ul style="list-style-type: none"> <li>- Solo se habilitará el botón “Modificar Lista Correo” si existe una lista seleccionada.</li> <li>- En caso de haber algún problema con los datos de la lista (como es la falta de un campo obligatorio) se marcará el campo en el que se ha cometido el error con una línea de color rojo, y al pasar el cursor sobre</li> </ul>	

el mismo se mostrará una notificación con la descripción del error.  
 - Solo se habilitarán los botones "Agregar" y "Quitar" si existe un contacto seleccionado en sus respectivos espacios.

**Tabla 11 Eliminar lista de correos.**

<b>Historia de Usuario</b>	
<b>Número:</b> 10	<b>Usuario:</b> Usuarios del sistema
<b>Nombre historia:</b> Eliminar lista de correos.	
<b>Prioridad en negocio:</b> Media	<b>Riesgo en desarrollo:</b> Medio
<b>Puntos estimados:</b> 3/5	<b>Iteración asignada:</b> 2
<b>Programador responsable:</b> Odaimys Rodríguez – Claudia Rodríguez	
<b>Descripción:</b> El usuario tendrá la posibilidad de eliminar de la aplicación web la lista que desee, para ello debe seleccionarla, si no se selecciona ninguna lista el botón "Eliminar Lista Correo" debe estar inhabilitado. Luego de seleccionar la lista y marcar el botón "Eliminar Lista Correo" se mostrará un mensaje de confirmación de eliminación, y una vez seleccionada la opción de confirmación se muestra un mensaje de información notificando que la acción se ejecutó correctamente.	
<b>Observaciones:</b> - Solo se habilitará el botón "Eliminar Lista Correo" si existe alguna lista seleccionada.	

**Tabla 12 Enviar mensaje instantáneo masivo.**

<b>Historia de Usuario</b>	
<b>Número:</b> 11	<b>Usuario:</b> Usuarios del sistema
<b>Nombre historia:</b> Enviar mensaje instantáneo masivo.	
<b>Prioridad en negocio:</b> Alta	<b>Riesgo en desarrollo:</b> Medio

<b>Puntos estimados:</b> 1	<b>Iteración asignada:</b> 3
<b>Programador responsable:</b> Odaimys Rodríguez – Claudia Rodríguez	
<b>Descripción:</b> Si el usuario realiza la acción de pulsar el botón “Nuevo Mensaje Instantáneo” se le mostrará una ventana con las listas de usuarios y los contactos de los cuales dispone, luego de seleccionar a los que desea enviar el mensaje, se mostrará una nueva ventana para que introduzca los datos del mensaje (cuerpo del mensaje), finalizando el envío al presionar el botón “Enviar”. Finalmente, se muestra un mensaje de información notificando que la acción se ejecutó correctamente.	
<b>Observaciones:</b> - Solo se habilitará el botón “Seleccionar” si existe un contacto o lista de contacto seleccionada.	

Tabla 13 Crear lista de usuarios.

Historia de Usuario	
<b>Número:</b> 12	<b>Usuario:</b> Usuarios del sistema
<b>Nombre historia:</b> Crear lista de usuarios	
<b>Prioridad en negocio:</b> Media	<b>Riesgo en desarrollo:</b> Medio
<b>Puntos estimados:</b> 4/5	<b>Iteración asignada:</b> 3
<b>Programador responsable:</b> Odaimys Rodríguez – Claudia Rodríguez	
<b>Descripción:</b> El usuario presiona el botón “Crear Lista Usuarios” que muestra una ventana para introducir los datos de la lista (nombre) y presiona “Aceptar”, lo cual origina una nueva ventana que permite elegir los contactos que formarán parte de dicha lista, para realizar esta acción basta con seleccionar el contacto y pulsar el botón “Agregar” o “Quitar” según se desee, finaliza la creación presionando el botón “Crear” que muestra un mensaje de información notificando que la acción se ejecutó correctamente.	
<b>Observaciones:</b> - El campo “nombre” de la lista es obligatorio.	

- En caso de haber algún problema con los datos de la lista (como es la falta de un campo obligatorio) se marcará el campo en el que se ha cometido el error con una línea de color rojo, y al pasar el cursor sobre el mismo se mostrará una notificación con la descripción del error.
- Solo se habilitarán los botones “Agregar” y “Quitar” si existe un contacto seleccionado en sus respectivos espacios.

**Tabla 14 Modificar lista de usuarios.**

Historia de Usuario	
<b>Número:</b> 13	<b>Usuario:</b> Usuarios del sistema
<b>Nombre historia:</b> Modificar lista de usuarios	
<b>Prioridad en negocio:</b> Media	<b>Riesgo en desarrollo:</b> Medio
<b>Puntos estimados:</b> 3/5	<b>Iteración asignada:</b> 3
<b>Programador responsable:</b> Odaimys Rodríguez – Claudia Rodríguez	
<p><b>Descripción:</b> El usuario presiona el botón “Modificar Lista Usuarios” que muestra una ventana con los contactos que posee esa lista y los que se le pueden añadir, para realizar estas acciones basta con seleccionar el contacto y pulsar el botón “Agregar” o “Quitar” según convenga, finaliza la modificación presionando el botón “Guardar” que muestra un mensaje de información notificando que la acción se ejecutó correctamente.</p>	
<p><b>Observaciones:</b></p> <ul style="list-style-type: none"> <li>- Solo se habilitará el botón “Modificar Lista Usuarios” si existe una lista seleccionada.</li> <li>- En caso de haber algún problema con los datos de la lista (como es la falta de un campo obligatorio) se marcará el campo en el que se ha cometido el error con una línea de color rojo, y al pasar el cursor sobre el mismo se mostrará una notificación con la descripción del error.</li> <li>- Solo se habilitarán los botones “Agregar” y “Quitar” si existe un contacto seleccionado en sus respectivos espacios.</li> </ul>	

Tabla 15 Eliminar lista de usuarios.

Historia de Usuario	
<b>Número:</b> 14	<b>Usuario:</b> Usuarios del sistema
<b>Nombre historia:</b> Eliminar lista de usuarios	
<b>Prioridad en negocio:</b> Media	<b>Riesgo en desarrollo:</b> Medio
<b>Puntos estimados:</b> 3/5	<b>Iteración asignada:</b> 3
<b>Programador responsable:</b> Odaimys Rodríguez – Claudia Rodríguez	
<p><b>Descripción:</b> El usuario tendrá la posibilidad de eliminar de la aplicación web la lista que desee, para ello debe seleccionarla, si no se selecciona ninguna lista el botón “Eliminar Lista Usuarios” debe estar inhabilitado. Luego de seleccionar la lista y marcar el botón “Eliminar Lista Usuarios” se mostrará un mensaje de confirmación de eliminación, y una vez seleccionada la opción de confirmación se muestra un mensaje de información notificando que la acción se ejecutó correctamente.</p>	
<p><b>Observaciones:</b></p> <p>- Solo se habilitará el botón “Eliminar Lista Usuarios” si existe alguna lista seleccionada.</p>	

Tabla 16 Enviar SMS masivo.

Historia de Usuario	
<b>Número:</b> 15	<b>Usuario:</b> Usuarios del sistema
<b>Nombre historia:</b> Enviar SMS masivo	
<b>Prioridad en negocio:</b> Alta	<b>Riesgo en desarrollo:</b> Medio
<b>Puntos estimados:</b> 1	<b>Iteración asignada:</b> 4
<b>Programador responsable:</b> Odaimys Rodríguez – Claudia Rodríguez	

<p><b>Descripción:</b> Si el usuario realiza la acción de pulsar el botón “Nuevo SMS” se le mostrará una ventana con las listas de telefonía móvil y los contactos de los cuales dispone, luego de seleccionar a los que desea enviar el mensaje, se mostrará una nueva ventana para que introduzca los datos del mensaje (cuerpo del mensaje), finalizando el envío al presionar el botón “Enviar”. Finalmente, se muestra un mensaje de información notificando que la acción se ejecutó correctamente.</p>
<p><b>Observaciones:</b></p> <ul style="list-style-type: none"> <li>- Solo se habilitará el botón “Seleccionar” si existe un contacto o lista de contacto seleccionada.</li> </ul>

Tabla 17 Crear listas de telefonía móvil.

<b>Historia de Usuario</b>	
<b>Número:</b> 16	<b>Usuario:</b> Usuarios del sistema
<b>Nombre historia:</b> Crear listas de telefonía móvil	
<b>Prioridad en negocio:</b> Media	<b>Riesgo en desarrollo:</b> Medio
<b>Puntos estimados:</b> 4/5	<b>Iteración asignada:</b> 4
Programador responsable: Odaimys Rodríguez – Claudia Rodríguez	
<p><b>Descripción:</b> El usuario presiona el botón “Crear Lista Telefonía Móvil” que muestra una ventana para introducir los datos de la lista (nombre) y presiona “Aceptar”, lo cual origina una nueva ventana que permite elegir los contactos que formarán parte de dicha lista, para realizar esta acción basta con seleccionar el contacto y el botón “Agregar” o “Quitar” según se desee, finaliza la creación presionando el botón “Crear” que muestra un mensaje de información notificando que la acción se ejecutó correctamente.</p>	
<p><b>Observaciones:</b></p> <ul style="list-style-type: none"> <li>- El campo “nombre” de la lista es obligatorio.</li> <li>- En caso de haber algún problema con los datos de la lista (como es la falta de un campo obligatorio) se marcará el campo en el que se ha cometido el error con una línea de color rojo, y al pasar el cursor sobre el mismo se mostrará una notificación con la descripción del error.</li> </ul>	

- Solo se habilitarán los botones “Agregar” y “Quitar” si existe un contacto seleccionado en sus respectivos espacios.

**Tabla 18 Modificar listas de telefonía móvil.**

<b>Historia de Usuario</b>	
<b>Número:</b> 17	<b>Usuario:</b> Usuarios del sistema
<b>Nombre historia:</b> Modificar listas de telefonía móvil.	
<b>Prioridad en negocio:</b> Media	<b>Riesgo en desarrollo:</b> Medio
<b>Puntos estimados:</b> 3/5	<b>Iteración asignada:</b> 4
<b>Programador responsable:</b> Odaimys Rodríguez – Claudia Rodríguez	
<b>Descripción:</b> El usuario presiona el botón “Modificar Lista Telefonía Móvil” que muestra una ventana con los contactos que posee esa lista y los que se le pueden añadir, para realizar estas acciones basta con seleccionar el contacto y pulsar el botón “Agregar” o “Quitar” según convenga, finaliza la modificación presionando el botón “Guardar” que muestra un mensaje de información notificando que la acción se ejecutó correctamente.	
<b>Observaciones:</b>	
<ul style="list-style-type: none"> <li>- Solo se habilitará el botón “Modificar Lista Telefonía Móvil” si existe una lista seleccionada.</li> <li>- En caso de haber algún problema con los datos de la lista (como es la falta de un campo obligatorio) se marcará el campo en el que se ha cometido el error con una línea de color rojo, y al pasar el cursor sobre el mismo se mostrará una notificación con la descripción del error.</li> <li>- Solo se habilitarán los botones “Agregar” y “Quitar” si existe un contacto seleccionado en sus respectivos espacios.</li> </ul>	

**Tabla 19 Eliminar listas de telefonía móvil.**

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

<b>Número:</b> 18	<b>Usuario:</b> Usuarios del sistema		
<b>Nombre historia:</b> Eliminar listas de telefonía móvil.			
<b>Prioridad en negocio:</b> Media		<b>Riesgo en desarrollo:</b> Medio	
<b>Puntos estimados:</b> 3/5		<b>Iteración asignada:</b> 4	
<b>Programador responsable:</b> Odaimys Rodríguez – Claudia Rodríguez			
<p><b>Descripción:</b> El usuario tendrá la posibilidad de eliminar de la aplicación web la lista que desee, para ello debe seleccionarla, si no se selecciona ninguna lista el botón “Eliminar Lista Telefonía Móvil” debe estar inhabilitado. Luego de seleccionar la lista y marcar el botón “Eliminar Lista Telefonía Móvil” se mostrará un mensaje de confirmación de eliminación, y una vez seleccionada la opción de confirmación se muestra un mensaje de información notificando que la acción se ejecutó correctamente.</p>			
<p><b>Observaciones:</b> - Solo se habilitará el botón “Eliminar Lista Telefonía Móvil” si existe alguna lista seleccionada.</p>			

**Tabla 20 Enviar mensaje de voz.**

Historia de Usuario			
<b>Número:</b> 19	<b>Usuario:</b> Usuarios del sistema		
<b>Nombre historia:</b> Enviar mensaje de voz			
<b>Prioridad en negocio:</b> Alta		<b>Riesgo en desarrollo:</b> Medio	
<b>Puntos estimados:</b> 1		<b>Iteración asignada:</b> 5	
<b>Programador responsable:</b> Odaimys Rodríguez – Claudia Rodríguez			
<p><b>Descripción:</b> Si el usuario realiza la acción de pulsar el botón “Nuevo Mensaje de Voz” se le mostrará una ventana que permitirá introducir el número de teléfono desde el cual se va a grabar el mensaje, además de las listas de telefonía fija y los contactos de los cuales dispone el usuario autenticado, luego de</p>			

seleccionar a los que le desea enviar el mensaje y presionar el botón “Aceptar” el teléfono que está usando timbrará, dará las instrucciones para grabar el mensaje y al colgar se realizará el envío automático. Mostrando un mensaje de información notificando que la acción se ejecutó correctamente.

**Observaciones:**

- Solo se habilitará la opción “Aceptar” si existe un contacto o lista de contacto seleccionado.

**Tabla 21 Crear directorio.**

<b>Historia de Usuario</b>	
<b>Número:</b> 20	<b>Usuario:</b> Usuarios del sistema
<b>Nombre historia:</b> Crear directorio	
<b>Prioridad en negocio:</b> Media	<b>Riesgo en desarrollo:</b> Bajo
<b>Puntos estimados:</b> 4/5	<b>Iteración asignada:</b> 5
<b>Programador responsable:</b> Odaimys Rodríguez – Claudia Rodríguez	
<p><b>Descripción:</b> El usuario presiona el botón “Crear Directorio” que muestra una ventana para introducir los datos del directorio (nombre) y presiona “Aceptar”, lo cual origina una nueva ventana que permite elegir los contactos que formarán parte de dicha lista, para realizar esta acción basta con seleccionar el contacto y pulsar el botón “Agregar” o “Quitar” según se desee, finaliza la creación presionando el botón “Crear” que muestra un mensaje de información notificando que la acción se ejecutó correctamente.</p>	
<p><b>Observaciones:</b></p> <ul style="list-style-type: none"> <li>- El campo “nombre” del directorio es obligatorio.</li> <li>- En caso de haber algún problema con los datos del directorio (como es la falta de un campo obligatorio) se marcará el campo en el que se ha cometido el error con una línea de color rojo, y al pasar el cursor sobre el mismo se mostrará una notificación con la descripción del error.</li> <li>- Solo se habilitarán los botones “Agregar” y “Quitar” si existe un contacto seleccionado en sus respectivos espacios.</li> </ul>	

Tabla 22 Modificar directorio.

<b>Historia de Usuario</b>	
<b>Número:</b> 21	<b>Usuario:</b> Usuarios del sistema
<b>Nombre historia:</b> Modificar directorio	
<b>Prioridad en negocio:</b> Media	<b>Riesgo en desarrollo:</b> Bajo
<b>Puntos estimados:</b> 3/5	<b>Iteración asignada:</b> 5
<b>Programador responsable:</b> Odaimys Rodríguez – Claudia Rodríguez	
<p><b>Descripción:</b> El usuario presiona el botón “Modificar Directorio” que muestra una ventana con los contactos que posee el directorio y los que se le pueden añadir, para realizar estas acciones basta con seleccionar el contacto y pulsar el botón “Agregar” o “Quitar” según convenga, finaliza la modificación presionando el botón “Guardar” que muestra un mensaje de información notificando que la acción se ejecutó correctamente.</p>	
<p><b>Observaciones:</b></p> <ul style="list-style-type: none"> <li>- Solo se habilitará el botón “Modificar Directorio” si existe un directorio seleccionado.</li> <li>- En caso de haber algún problema con los datos del directorio (como es la falta de un campo obligatorio) se marcará el campo en el que se ha cometido el error con una línea de color rojo, y al pasar el cursor sobre el mismo se mostrará una notificación con la descripción del error.</li> <li>- Solo se habilitarán los botones “Agregar” y “Quitar” si existe un contacto seleccionado en sus respectivos espacios.</li> </ul>	

Tabla 23 Eliminar directorio.

<b>Historia de Usuario</b>	
<b>Número:</b> 22	<b>Usuario:</b> Usuarios del sistema

<b>Nombre historia:</b> Eliminar directorio	
<b>Prioridad en negocio:</b> Media	<b>Riesgo en desarrollo:</b> Bajo
<b>Puntos estimados:</b> 3/5	<b>Iteración asignada:</b> 5
<b>Programador responsable:</b> Odaimys Rodríguez – Claudia Rodríguez	
<p><b>Descripción:</b> El usuario tendrá la posibilidad de eliminar de la aplicación web el directorio que desee, para ello debe seleccionarlo, si no se selecciona ningún directorio el botón “Eliminar Directorio” debe estar inhabilitado. Luego de seleccionar el directorio a eliminar y marcar la opción “Eliminar Directorio” se mostrará un mensaje de confirmación de eliminación, y una vez seleccionada la opción de confirmación pues se muestra un mensaje de información notificando que la acción se ejecutó correctamente.</p>	
<p><b>Observaciones:</b></p> <p>- Solo se habilitará el botón “Eliminar Directorio” si existe algún directorio seleccionado.</p>	

## 2.5 Planificación.

En esta fase se establece la prioridad de cada historia de usuario así como se realiza una estimación del esfuerzo necesario de cada una de ellas, se toman acuerdos sobre el contenido de la entrega y se determina un cronograma en conjunto con el cliente. Las estimaciones de esfuerzo asociado a la implementación de las historias se establecen utilizando como medida el punto, este es considerado como una semana de trabajo, semanas de 40 horas, en la que el equipo de desarrollo labora de forma ininterrumpida.

### 2.5.1 Estimación de esfuerzo por historias de usuarios.

Para el desarrollo del sistema propuesto en este trabajo se realizó una estimación del esfuerzo para cada una de las historias de usuario identificadas, llegándose a los resultados que se muestran en la siguiente tabla:

**Tabla 24 Estimación de esfuerzo por historia de usuario.**

Historia de Usuario	Puntos de estimación
---------------------	----------------------

Adicionar cliente.	3/5
Modificar cliente	2/5
Eliminar cliente	2/5
Adicionar contacto.	3/5
Modificar contacto.	2/5
Eliminar contacto.	2/5
Enviar correo masivo.	1
Crear lista de correos.	4/5
Modificar lista de correos.	3/5
Eliminar lista de correos.	3/5
Enviar mensaje instantáneo masivo.	1
Crear lista de usuarios.	4/5
Modificar lista de usuarios.	3/5
Eliminar lista de usuarios.	3/5
Enviar SMS masivo.	1
Crear listas de telefonía móvil.	4/5
Modificar listas de telefonía móvil.	3/5
Eliminar listas de telefonía móvil.	3/5
Enviar mensaje de voz.	1
Crear directorio.	4/5
Modificar directorio	3/5
Eliminar directorio	3/5

### 2.5.2 Plan de Iteraciones.

Luego de identificadas las historias de usuario y estimado el esfuerzo dedicado a la realización de cada una de estas historias se procede a la realización de la planificación de la etapa de implementación del sistema a desarrollar. En base a lo antes mencionado se decide realizar esta en cinco iteraciones, las cuales se detallan a continuación.

### **Iteración 1.**

La iteración 1 será la encargada de la implementación de las historias de usuario de 1, 2, 3, 4, 5 y 6. Esta iteración servirá de base para el trabajo en las fases venideras pues se encarga de la implementación de algunas de las historias de usuarios fundamentales para el desarrollo del sistema.

### **Iteración 2.**

La iteración 2 será la encargada de la implementación de las historias de usuario de 7, 8, 9 y 10. Al finalizar esta iteración se contará con la segunda versión de prueba de la aplicación la que va a contar con todo lo relacionado con el servicio de correo electrónico.

### **Iteración 3.**

La iteración 3 tendrá como principal objetivo la implementación de las historias de usuario 11, 12, 13 y 14 que no dejan de ser importantes pero el cliente solicitó se implementaran luego de culminar con el servicio de correo, al finalizar esta iteración se contará con una versión de prueba que abarcará lo relacionado al servicio de mensajería instantánea unido al correo electrónico terminado en la fase anterior.

### **Iteración 4.**

La iteración 4 tendrá como principal objetivo la implementación de las historias de usuario 15, 16, 17 y 18 que presentan la misma importancia que las tratadas en la iteración 3, y al igual que estas el cliente solicitó su implementación luego de culminar con el servicio de mensajería instantánea, al finalizar esta iteración se contará con una versión de prueba que abarcará lo relacionado al servicio de mensajería SMS, además de mensajería instantánea y correo electrónico tratados en las fases anteriores .

### **Iteración 5.**

La iteración 5 por ser la última no pierde importancia solo que se solicitó por el cliente su desarrollo en la última iteración, esta tendrá como tarea la implementación de las historias de usuario 19, 20, 21 y 22 que son las encargadas de complementar el resultado del producto requerido por el cliente. Al terminar esta iteración se contará con una versión 1.0 del producto final quedando solamente como tarea poner en funcionamiento el sistema para evaluar su desempeño.

2.5.3 Plan de duración de las iteraciones.

Tabla 25 Plan de duración de iteraciones.

Iteración	Orden de la Historias de usuario a implementar.	Duración total
1	Adicionar cliente. Modificar cliente. Eliminar cliente. Adicionar contacto. Modificar contacto. Eliminar contacto.	3 semanas
2	Enviar correo masivo. Crear lista de correo. Modificar lista de correo. Eliminar lista de correo.	3 semanas
3	Enviar mensaje instantáneo masivo. Crear lista de usuarios. Modificar lista de usuarios. Eliminar lista de usuarios.	3 semanas
4	Enviar SMS masivo. Crear listas de telefonía móvil. Modificar listas de telefonía móvil. Eliminar listas de telefonía móvil.	3 semanas
5	Enviar mensaje de voz. Crear directorio. Modificar directorio. Eliminar directorio.	3 semanas

**2.5.4 Plan de entregas.**

A continuación se presenta el plan de entregas ideado para la fase de implementación. Como producto del mismo se hará una versión del sistema al finalizar cada iteración en la fecha aproximada que se indica en la siguiente tabla.

**Tabla 26 Plan de entregas.**

Módulo	Final Iteración 1 (10 - marzo - 2011)	Final Iteración 2 (1 - abril - 2011)	Final Iteración 3 (22 - abril - 2011)	Final Iteración 4 (13 - mayo - 2011)	Final Iteración 5 (6 - junio - 2011)
SISCU	Versión 0.1	Versión 0.2	Versión 0.3	Versión 0.4	SISCU 1.0

**2.6 Conclusiones parciales.**

Como parte del presente capítulo se abordó todo lo referente a las fases de exploración y planificación desarrolladas para la correcta implementación de la aplicación, haciendo una descripción de la duración y de los artefactos generados durante el transcurso de las mismas.

## CAPÍTULO 3: DISEÑO DEL SISTEMA.

### 3.1 Introducción.

En este capítulo se tratarán los diferentes artefactos generados en esta fase así como un diseño de la solución más sencilla a implementar, la definición de la arquitectura, patrones de diseño web y diseño de la base de datos, entre otros aspectos.

### 3.2 Patrones de Diseño.

Los patrones son una solución ya probada y documentada a un problema en un determinado contexto o situación. Se puede asegurar que un sistema para que quede bien estructurado debe estar lleno de patrones. Cada patrón hace una descripción de la solución de cada problema, y de esa forma la solución se puede utilizar una y otra vez, sin repetir las mismas acciones.

Para el desarrollo de la aplicación se tuvieron en cuenta algunos de los patrones GRASP y GOF, implementados por CEDRUX los cuales son vitales para entender el funcionamiento de dicho marco de trabajo y con esto lograr la obtener el objetivo trazado.

#### **Patrones para Asignar Responsabilidades (GRASP) que implementa el CEDRUX.**

- Experto: El uso de este patrón posibilita la asignación de responsabilidad, la clase tiene la información necesaria para la realización de la asignación. El uso de este patrón permitirá que las clases controladoras manejen las peticiones del cliente y las clases del modelo son las encargadas del acceso a datos, esto proporciona que se conserve el encapsulamiento y además se dé soporte a un bajo acoplamiento y una alta cohesión.
- Creador: Es el responsable de la creación de una nueva instancia de alguna clase. Existe un único script PHP que crea una única instancia del controlador frontal, dicho controlador frontal (Dispatcher) este último es el encargado de instanciar las clases controladoras y éstas, a su vez, sólo crean objetos de la clase Zend\_View. El uso de este patrón permite crear las dependencias mínimas necesarias entre las clases, lo que favorece al mantenimiento del sistema.
- Alta cohesión: Es el responsable de mantener la complejidad manejable, se caracteriza por asignar una responsabilidad de manera que la cohesión permanezca alta. Este patrón permitirá tener clases fáciles de mantener, entender y reutilizar.

- Bajo acoplamiento: Permite soportar bajas dependencias, un bajo impacto del cambio e incremento de la reutilización. Este patrón se tiene en cuenta por la importancia de realizar un diseño de clases independiente que puedan soportar los cambios de una manera fácil y permitan la reutilización. El uso de los patrones Experto y Creador favorecen al bajo acoplamiento entre las clases del sistema.
- Controlador: Es el responsable de gestionar un evento de entrada al sistema. Zend Framework colabora en la utilización de este patrón, ya que este define un controlador frontal que conlleva a que todas las solicitudes son dirigidas a un único script PHP que se encarga de instanciar al controlador frontal y redirigir las llamadas.

### **Patrones Gang of Fourt (GOF) que implementa el CEDRUX.**

- Decorator: Es un patrón estructural que añade funcionalidad a una clase dinámicamente. Zend Framework implementa este patrón en la clase `Zend_View`, encargada de asignarle responsabilidades a objetos de manera dinámica y configurarlos con nuevos atributos.
- Facade: Es un patrón estructural y consiste en crear una única clase de manejo más fácil, que permita acceder a un conjunto numeroso y complicado de clases. La fachada satisface a la mayoría de los clientes, sin ocultar las funciones de menor nivel a aquellos que necesiten acceder a ellas. (22) En la solución se pone de manifiesto este patrón en las clases pertenecientes al paquete “services” del componente las cuales sirven de fachada a las funcionalidades brindadas por el mismo al resto de los componentes.
- Factory: Es un patrón creacional que se utiliza cuando es necesario crear diferentes objetos, todos pertenecientes a la misma familia. Este permite configurar en tiempo de ejecución un sistema con una familia u otra de objetos. Zend Framework utiliza este patrón en la implementación de un conjunto de clases para el acceso a datos.
- Singleton: Es un patrón creacional que tiene como propósito garantizar una única instancia de una clase, proporcionando un punto de acceso global a la misma. Zend Framework posee una instancia única del controlador frontal disponible mediante este patrón para lograr una vía de entrada única a las solicitudes.

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

El diagrama de tarjetas CRC es una técnica de modelado orientado a objetos que permite identificar las clases y sus responsabilidades. Las tarjetas CRC son una herramienta de ayuda al refinamiento de

clases. Consiste en elaborar para cada clase una tarjeta con los siguientes datos: Nombre, Responsabilidades, Colaboraciones.

Las tarjetas determinan el comportamiento de cada actividad. Las tarjetas CRC identificadas durante el desarrollo del sistema se encuentran anexas, específicamente en el Anexo 2.

### 3.4 Diseño de la Base de datos.

El diseño de la base de datos en una tarea de vital importancia para la correcta implementación de una aplicación web, en esta se describen las relaciones y los tipos de datos que se van a usar para lograr un correcto funcionamiento del sistema. En el caso de la aplicación que se está desarrollando se identificaron 6 tablas, de las cuales 2 de ellas son producto de la relación de mucho a mucho existente entre otras de las tablas. A continuación se muestra el diagrama Entidad-Relación diseñado para la aplicación:

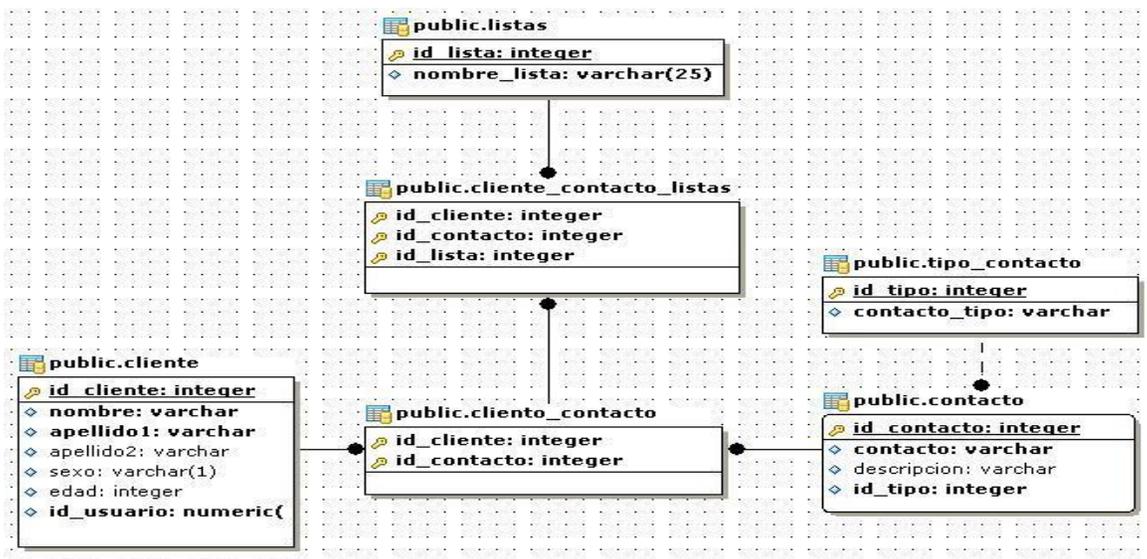


Imagen 5 Diagrama de Entidad Relación de la Base de Datos.

### 3.5 Conclusiones parciales.

En el presente capítulo se abordó lo referente a la fase de diseño del proyecto, que contempla los patrones de diseño que favorecen el buen desempeño de la siguiente fase y describe el diseño de la base de datos mediante el modelo físico con todas las entidades que se manejan en el sistema. A partir de este punto se puede continuar con las siguientes fases en el desarrollo del sistema atendiendo a que se cumplan los requerimientos que se proponen implementar.

## CAPÍTULO 4: IMPLEMENTACIÓN Y PRUEBA.

### 4.1 Introducción.

El presente capítulo describe las iteraciones realizadas en la etapa de codificación del proyecto, las pruebas de aceptación efectuadas sobre el mismo y las tareas de ingeniería generadas en cada iteración. Ajustándose a la planificación realizada, se llevaron a cabo cinco iteraciones de desarrollo sobre el sistema, cumpliendo con lo planteado por la metodología que expone que para la correcta implementación de un software se debe obtener al culminar cada iteración un producto funcional, el cual debe ser probado por el equipo de desarrollo y mostrado al cliente para lograr que este emita su opinión a los desarrolladores con el propósito de conformar un producto listo para su puesta en producción.

### 4.2 Implementación.

Durante el transcurso de las iteraciones se realiza la implementación de las historias de usuario seleccionadas para cada una de estas. Al inicio de las mismas, se lleva a cabo una revisión del plan de iteraciones y se modifica de ser necesario. (23) A continuación se detallan las cinco iteraciones desarrolladas en el sistema y se exponen las tareas generadas por cada historia de usuario.

#### 4.2.1 Primera Iteración.

Durante esta iteración se abordaron algunas de las historias de usuario de mayor prioridad y se construyó la base de la arquitectura del sistema con el fin de obtener un producto para ser mostrado al cliente y obtener sugerencias de este.

**Tabla 27 Tiempo de las tareas desarrolladas en la primera iteración.**

Historia de Usuario	Estimación	Real
Adicionar cliente.	3/5	3/5
Modificar cliente.	2/5	2/5
Eliminar cliente.	2/5	2/5
Adicionar contacto.	3/5	3/5
Modificar contacto.	2/5	2/5

Eliminar contacto.	2/5	2/5
<b>TOTAL</b>	<b>3</b>	<b>3</b>

**Tareas de las historias de usuario desarrolladas en la primera iteración.**

Las tareas de las historias de usuarios desarrolladas en la primera iteración se encuentran en el Anexo 4.

**4.2.2 Segunda Iteración.**

Durante el transcurso de la presente iteración se concluyó la implementación de las funcionalidades concernientes al servicio de correo electrónico.

**Tabla 28 Tiempo de las tareas desarrolladas en la segunda iteración.**

Historia de Usuario	Estimación	Real
Enviar correo masivo.	1	1
Crear lista de correos.	4/5	4/5
Modificar lista de correos.	3/5	3/5
Eliminar lista de correos.	3/5	3/5
<b>TOTAL</b>	<b>3</b>	<b>3</b>

**Tareas de las historias de usuario desarrolladas en la segunda iteración.**

Las tareas de las historias de usuarios desarrolladas en la segunda iteración se encuentran en el Anexo 4.

**4.2.3 Tercera Iteración.**

Durante el transcurso de la presente iteración se concluyó la implementación de las funcionalidades concernientes al servicio de mensajería instantánea.

**Tabla 29 Tiempo de las tareas desarrolladas en la tercera iteración.**

Historia de Usuario	Estimación	Real
Enviar mensaje instantáneo masivo.	1	1
Crear lista de usuarios.	4/5	4/5
Modificar lista de usuarios.	3/5	3/5
Eliminar lista de usuarios.	3/5	3/5

<b>TOTAL</b>	<b>3</b>	<b>3</b>
--------------	----------	----------

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

Las tareas de las historias de usuarios desarrolladas en la tercera iteración se encuentran en el Anexo 4.

#### 4.2.4 Cuarta Iteración.

Durante el transcurso de la presente iteración se concluyó la implementación de las funcionalidades concernientes al servicio de mensajería SMS.

**Tabla 30 Tiempo de las tareas desarrolladas en la cuarta iteración.**

Historia de Usuario	Estimación	Real
Enviar SMS masivo.	1	1
Crear listas de telefonía móvil.	4/5	4/5
Modificar listas de telefonía móvil.	3/5	3/5
Eliminar listas de telefonía móvil.	3/5	3/5
<b>TOTAL</b>	<b>3</b>	<b>3</b>

#### Tareas de las historias de usuario desarrolladas en la cuarta iteración.

Las tareas de las historias de usuarios desarrolladas en la cuarta iteración se encuentran en el Anexo 4.

#### 4.2.5 Quinta Iteración.

En el transcurso de esta iteración se implementaron las historias de usuario restantes que involucraban las funcionalidades concernientes al servicio de telefonía. Al culminar esta, se cuenta con un producto listo para su puesta en funcionamiento.

**Tabla 31 Tiempo de las tareas desarrolladas en la quinta iteración.**

Historia de Usuario	Estimación	Real
Enviar mensaje de voz.	1	1
Crear directorio.	4/5	4/5
Modificar directorio	3/5	3/5
Eliminar directorio	3/5	3/5

TOTAL	3	3
-------	---	---

**Tareas de las historias de usuario desarrolladas en la quinta iteración.**

Las tareas de las historias de usuarios desarrolladas en la quinta iteración se encuentran en el Anexo 4.

**4.3 Prueba.**

El objetivo de las pruebas de software es detectar el mayor número posible de errores. Para lograrlo existen diferentes técnicas. Pese a que se ha realizado una división entre pruebas de caja blanca y caja negra, no deben considerarse técnicas excluyentes: se consiguen mejores resultados al combinar ambos enfoques. Por otro lado, el proceso de pruebas no se debe plantear exclusivamente en las fases finales de la implementación del producto, incluso algunos autores definen la necesidad de dar la vuelta completamente al modelo de desarrollo lineal y comenzar la elaboración del software por la especificación de los casos de prueba. (24)

XP divide las pruebas en dos grupos: pruebas unitarias, desarrolladas por los programadores y encargadas de verificar el código de forma automática y las pruebas de aceptación, destinadas a evaluar si al final de una iteración se consiguió la funcionalidad requerida además de comprobar que dicha funcionalidad sea la esperada por el cliente. (23)

**4.3.1 Prueba Unitaria.**

Las pruebas unitarias son una de las piedras angulares de XP, las mismas consisten en comprobaciones (manuales o automatizadas) desarrolladas por los programadores que se realizan para verificar que el código correspondiente a un módulo concreto se comporta de manera esperada. Todos los módulos deben de pasar las pruebas unitarias antes de ser liberados o publicados. Por otra parte, las pruebas deben ser definidas antes de realizar el código. Que todo código liberado pase correctamente las pruebas unitarias es lo que habilita que funcione la propiedad colectiva del código. En este sentido, el sistema y el conjunto de pruebas debe ser guardado junto con el código, para que pueda ser utilizado por otros desarrolladores, en caso de tener que corregir, cambiar o re-codificar parte del mismo. (25) Estas pruebas no generan artefactos y no son directamente palpables para el cliente lo cual no quiere decir que no sean de vital importancia para el desarrollo del proyecto.

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

```
var TFApell1 = new Ext.form.TextField({
    id : 'apellido1',
    allowBlank : false,
    fieldLabel:perfil.etiquetas.lbTitMsgApellido
});
```

Este código permitía que al campo “apellido del cliente” se le pudieran introducir datos no deseados como caracteres extraños y números, lo que se pudo corregir mediante una validación utilizando una expresión regular en JavaScript.

```
var letras=/^[a-zA-Záéíóúñ]*$/
```

```
var TFApell1 = new Ext.form.TextField({
    id : 'apellido1',
    allowBlank : false,
    maskRe: letras,
    fieldLabel:perfil.etiquetas.lbTitMsgApellido
});
```

**Tabla 32 Cantidad de Pruebas Unitarias realizadas.**

Cantidad de Pruebas Realizadas.	Cantidad de Pruebas Satisfactorias.	Cantidad de Errores Encontrados.	Cantidad de Errores Corregidos.
26	19	7	7

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

### 4.3.2 Prueba Aceptación.

Las pruebas de aceptación son creadas en base a las historias de usuarios, en cada ciclo de la iteración del desarrollo. El cliente debe especificar uno o diversos escenarios para comprobar que una historia de usuario ha sido correctamente implementada.

Las pruebas de aceptación son consideradas como “pruebas de caja negra” (“Black box system tests”). Los clientes son responsables de verificar que los resultados de estas pruebas sean correctos. Asimismo, en caso de que fallen varias pruebas, deben indicar el orden de prioridad de resolución.

Una historia de usuario no se puede considerar terminada hasta tanto pase correctamente todas las pruebas de aceptación.

Dado que la responsabilidad es grupal, es recomendable publicar los resultados de las pruebas de aceptación, de manera que todo el equipo esté al tanto de esta información. (25)

Las pruebas de aceptación son definidas por el cliente, quien en el caso de la aplicación SISCO solicitó la realización de pruebas a las funcionalidades de mayor prioridad, así como también incluyó algunas que no tienen la prioridad máxima pero que determinó era importante realizarles pruebas. Como resultados de dichas pruebas se generaron artefactos descritos en las tablas situadas en el Anexo 4, las cuales constatan de nueve campos donde se recogen las siguientes informaciones:

- Código Caso de Prueba: Será el identificador de la prueba a la cual hace referencia.
- Nombre Historia de Usuario: Contendrá el nombre de la historia de usuario a la cual hace referencia.
- Nombre de la persona que realiza la prueba: Nombre del miembro del equipo que realiza la prueba.
- Descripción de la Prueba: Se realiza una breve descripción de la funcionalidad que se desea probar.
- Condiciones de Ejecución: Describe las condiciones que deben cumplirse para poder realizar la prueba.
- Entradas / Pasos de Ejecución: Se describen cada uno de los pasos llevados a cabo durante el desarrollo de la prueba, teniendo en cuenta las entradas realizadas por el usuario, teniendo como propósito fundamental, observar si se obtiene el resultado esperado.
- Resultado Esperado: Se realiza una breve descripción del resultado esperado con la realización de la prueba.
- Evaluación de la Prueba: Evaluación emitida de acuerdo con el resultado que se obtiene de la prueba realizada.

En el caso de las evaluaciones realizadas, se pueden obtener tres resultados:

- Satisfactoria: Cuando el resultado obtenido en la prueba es exactamente el esperado por el usuario.
- Parcialmente Satisfactoria: Cuando el resultado no es el óptimo o esperado por el cliente.
- Insatisfactoria: Cuando el resultado de la prueba genera un error de codificación en la aplicación, trayendo consigo la invalida la historia de usuario.

#### **4.4 Conclusiones parciales.**

En el capítulo se especificaron las tareas que se realizaron en las cinco iteraciones realizadas para el desarrollo del sistema. Se abordaron las pruebas realizadas, así como la importancia de las mismas en la metodología XP, destacándose las pruebas de aceptación, las cuales miden la satisfacción del cliente con el producto desarrollado.

## **CAPÍTULO 5: ESTUDIO DE FACTIBILIDAD.**

### **5.1 Introducción.**

En el presente capítulo se realiza un estudio de factibilidad para la realización del sistema propuesto, haciendo una estimación del tamaño, la planificación, así como esfuerzo necesario para llevar a cabo el mismo, todo esto se debe a la gran importancia que presenta en la planificación de proyectos de software la estimación, siendo esta la encargada de determinar con cierto grado de certeza los recursos necesarios para la implementación y correcta elaboración del producto a entregar al cliente.

### **5.2 Modelo COCOMO II.**

COCOMO II fue propuesto y desarrollado por Barry Boehm, este modelo permite realizar estimaciones en función del tamaño del software, y de un conjunto de factores de costo y de escala. Los factores de costo describen aspectos relacionados con la naturaleza del producto, hardware utilizado, personal involucrado, y características propias del proyecto.

COCOMO II posee tres modelos denominados Composición de Aplicación, Diseño Temprano y Post-Arquitectura. Cada uno de ellos orientados a sectores específicos del mercado de desarrollo de software y a las distintas etapas del desarrollo de software. (26)

El estudio de factibilidad de este sistema se realizó mediante el modelo de Diseño Temprano el cual usa Puntos de Función No Ajustados como métrica de medida. Este modelo se utiliza en las primeras etapas de un proyecto software, cuando se conoce muy poco sobre el tamaño del producto que se va a desarrollar, la naturaleza de la plataforma objetivo, la naturaleza del personal involucrado en el proyecto o especificaciones detalladas del proceso que se va a usar.

### **5.3 Características del Proyecto.**

#### **5.3.1 Entradas externas.**

Entrada de datos del usuario o de control que ingresan desde el exterior del sistema para agregar y/o cambiar datos a un archivo lógico interno. En el caso específico de la aplicación desarrollada, se cuenta con 16 entradas externas, presentadas en la tabla que se muestra a continuación.

**Tabla 33 Entradas externas.**

Nombre de la entrada externa.	Cantidad de ficheros.	Cantidad de elementos de datos.	Clasificación (Simple, Media, Compleja)
Adicionar cliente	1	5	simple
Modificar cliente	1	5	simple
Adicionar contacto	1	3	simple
Modificar contacto	1	3	simple
Enviar correo electrónico	1	2	simple
Adicionar listas correo	1	1	simple
Modificar listas correo	1	1	simple
Enviar mensaje instantáneo	1	1	simple
Adicionar listas usuarios	1	1	simple
Modificar listas usuarios	1	1	simple
Enviar SMS	1	1	simple
Adicionar listas de SMS	1	1	simple
Modificar listas de SMS	1	1	simple
Enviar mensaje de voz	1	1	simple
Adicionar directorio	1	1	simple
Modificar directorio	1	1	simple
<b>Total</b>		29	

### 5.3.2 Salidas externas.

Salida de datos de usuario o de control que deja el límite del sistema de software. En el caso específico de la aplicación desarrollada, se cuenta con 4 salidas externas, presentadas en la tabla que se muestra a continuación:

**Tabla 34 Salidas externas.**

Nombre de la salida externa.	Cantidad de ficheros.	Cantidad de elementos de datos.	Clasificación (Simple, Media, Compleja)
Enviar correo electrónico	1	1	simple
Enviar mensaje instantáneo	1	1	simple
Enviar SMS	1	1	simple
Enviar mensaje de voz	1	1	simple
<b>Total</b>		4	

### 5.3.3 Consultas externas.

Se definen como 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 este caso no fueron identificados.

**Tabla 35 Consultas externas.**

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

### 5.3.4 Archivos lógicos internos.

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

**Tabla 36 Fichero lógico interno.**

Nombre de la petición.	Cantidad de ficheros.	Cantidad de elementos de datos. (tablas)	Clasificación (Simple, Media, Compleja)
Base Batos: Pacsoft	1	6	Simple.
<b>Total</b>		6	

### 5.3.5 Archivos de interfaz externos

Archivos transferidos o compartidos entre sistemas de software. En el caso específico de la aplicación desarrollada, fueron identificados 4 archivos de interfaz externos, presentadas en la tabla que se muestra a continuación:

**Tabla 37 Interfaces Externas**

Nombre de la interfaz externa.	Cantidad de ficheros.	Cantidad de elementos de datos.	Clasificación (Simple, Media, Compleja)
Servidor Correo	1	3	simple

Servidor Jabber	1	2	simple
Servidor Asterik	1	2	simple
Servidor Web	1	2	simple
<b>Total</b>		9	

### 5.3.6 Puntos de función desajustados.

La tabla que se presentará a continuación está basada en las características del sistema anteriormente expuestas, el producto de la cantidad existente de cada una de ellas y el peso correspondiente a las mismas da como resultado final, los puntos de función desajustados pertenecientes al proyecto.

**Tabla 38 Puntos de función desajustados.**

Elementos.	Simple.		Medio.		Complejo.		Subtotal.
	No.	Peso.	No.	Peso.	No.	Peso.	
Entradas externas.	29	3	0	4	0	6	87
Salidas externas.	4	4	0	5	0	7	16
Consultas externas.	0	3	0	4	0	6	0
Fichero lógico interno.	6	7	0	10	0	15	42
Fichero interfaz interno.	9	5	0	7	0	10	45
<b>Total (UFP):</b>							<b>190</b>

**Tabla 39 Peso del factor de complejidad.**

Tipo de función.	Peso del factor de complejidad.		
	Bajo.	Promedio.	Alto.
Entradas externas (Inputs).	3	4	6
Salidas externas (Outputs).	4	5	7
Archivo lógicos internos (Archivos).	7	10	15
Archivos externos de interfase (Interfases).	5	7	10
Consultas externas (Queries).	3	4	6

**5.4 Cálculos de instrucciones.**

**5.4.1 Cálculos de instrucciones.**

Una vez obtenida la cantidad total de puntos de función desajustados pertenecientes al proyecto, se procede a calcular la cantidad de instrucciones fuentes del mismo, para lo cual se utiliza la ecuación que se muestra seguidamente:

$$SLOC = UFP \times ratio$$

Donde:

**SLOC:** Cantidad de instrucciones fuente: 11210.

**UFP:** Puntos de función desajustados: 190.

**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:

$$SLOC = 190 \times 59$$

$$SLOC = 11210$$

**KSLOC** tamaño del software a desarrollar.

$$KSLOC = 11.21$$

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

**Tabla 40 Características del sistema.**

Características	Valor
Puntos de función desajustados.	190
Lenguaje (PHP).	59
Instrucciones fuentes por puntos de función.	11210 SLOC
Instrucciones fuentes.	11,21 KSLOC

**5.4.2 Cálculo de esfuerzo nominal.**

Posteriormente se procede al cálculo del esfuerzo nominal, ecuación que se toma como base tanto en el método de diseño preliminar, al cual se hace alusión anteriormente, como en el modelo Post arquitectura, ambos definidos por COCOMO II.

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

$$PM_{Nominal} = 2.94 \times (11.21)^{0.97}$$

$$PM_{Nominal} = 30.89 \text{ meses/hombre}$$

Donde:

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

**Size:** Tamaño estimado del software en Puntos de Función sin Ajustar. (11,21 KSLOC)

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

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

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

$$B = 0.91 + 0.01 \times 6.32$$

$$B = 0.97$$

Donde:

**W** 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 con el 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.

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

**Tabla 41 Factores de escala del sistema.**

Nombre.	Valor.	Justificación.
PREC	1,24	Se contaba con experiencia en otros proyectos.
FLEX	1,01	Cuenta con alta flexibilidad en cuanto a los requerimientos establecidos inicialmente.
TEAM	1,10	Se presentó una buena sincronización entre el cliente y los desarrolladores.
RESL	1,41	No se identificaron riesgos significativos.
PMAT	1,56	El software da cumplimiento a funcionalidades solicitadas por el cliente.
<b>Total(SF)</b>	6,32	

**Tabla 42 Factores de escala.**

Nombre.	VLO	LO	NOM	HI	VHI	XHI
<b>PREC</b>	6.20	4.96	3.72	2.40	1.24	0.00
<b>FLEX</b>	5.07	4.05	3.04	2.03	1.01	0.00
<b>RESL</b>	7.07	5.65	4.24	2.83	1.41	0.00
<b>TEAM</b>	5.48	4.38	3.29	2.19	1.10	0.00
<b>PMAT</b>	7.80	6.24	4.68	3.12	1.56	0.00

### 5.5 Ajuste del esfuerzo nominal.

El esfuerzo calculado anteriormente es un valor nominal y debe ser ajustado en base a las características del proyecto para lo cual se tiene un conjunto de Multiplicadores de Esfuerzo (MEi) que representan las características del proyecto y expresan su impacto en el desarrollo total del producto de software.

#### 5.5.1 Ajuste del esfuerzo nominal clasificados en categorías.

Los 7 multiplicadores de esfuerzo 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.

**Tabla 43 Multiplicadores de esfuerzo.**

Nombre.	Valor.	Justificación.
<b>RCPX</b>	1,30	La complejidad que presenta la aplicación es alta.
<b>RUSE</b>	1,00	La aplicación reutiliza código.
<b>PDIF</b>	1,00	Uso de memoria y almacenamiento normal, plataforma estable.
<b>PREX</b>	0,71	La experiencia del equipo con la herramienta y lenguaje es alta.
<b>PERS</b>	1,00	La capacidad del equipo de desarrollo es normal.
<b>FCIL</b>	0,73	La aplicación es fácil de utilizar.

<b>SCED</b>	1,00	El desarrollo del sistema se realizó en el tiempo planificado.
<b>Total(EM)</b>	0,67	

Tabla 44 Multiplicadores de esfuerzo del modelo del diseño temprano.

	XLO	VLO	LO	NOM	HI	VHI	XHI
<b>RCPX</b>	0.73	0.81	0.98	1.00	1.30	1.74	2.38
<b>RUSE</b>	XXXX	XXXX	0.95	1.00	1.07	1.15	1.24
<b>PDIF</b>	XXXX	XXXX	0.87	1.00	1.29	1.81	2.61
<b>PERS</b>	2.12	1.62	1.26	1.00	0.83	0.63	0.50
<b>PREX</b>	1.59	1.33	1.12	1.00	0.87	0.71	0.62
<b>FCIL</b>	1.43	1.30	1.10	1.00	0.87	0.73	0.62
<b>SCED</b>	XXXX	1.43	1.14	1.00	1.00	1.00	XXXX
<b>USR1</b>	XXXX	1.00	1.00	1.00	1.00	1.00	XXXX
<b>USR2</b>	XXXX	1.00	1.00	1.00	1.00	1.00	XXXX

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

$$PM_{ajustado} = 30.89 \text{ meses/hombre} \times 0.67$$

$$PM_{ajustado} = 20.81 \text{ meses/hombre}$$

### 5.5.2 Cálculo del tiempo de desarrollo del software.

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

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

$$TDEV = 3.64 \times (20.81)^{0.25}$$

$$TDEV = 7.84$$

Donde:

C: Constante con valor 3.64.

PM<sub>ajustado</sub> = 20.81 meses/hombre.

F = 0.26

**CALCULAR**

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

$$F = 0.24 + 0.2 \times 0.01 \times 6.32$$

$$F = 0.25$$

Donde:

**D** : Constante cuyo valor es 0.24.

**SF** : Valor de los factores de escala.

### 5.5.3 Cálculo del costo total del proyecto.

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

$$C = CHM \times PM$$

$$C = 265.57 \times 20.81$$

$$C = 5527.48$$

#### CALCULAR CHM.

Donde:

**C**: Costo total.

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

$$CHM = CH \times sal$$

$$CHM = 2.66 \times 100$$

$$CHM = 265.57$$

Donde:

**sal** : Salario medio por cada trabajador.

**CH** : Cantidad de personas destinadas al proyecto, lo que es calculado a través de la ecuación:

$$CH = \frac{PM}{TDEV}$$

$$CH = \frac{20.81}{7.84}$$

$$CH = 2.66$$

### 5.6 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.

**Tabla 45 Resultados**

Cálculo de:	Valor
Esfuerzo.	20,81 meses/hombre
Tiempo de desarrollo.	10 meses
Cantidad de hombres.	2 hombres.
Salario medio.	100 pesos
Costo.	5527,48 pesos

### 5.7 Análisis de costo.

El desarrollo de un producto siempre tiene un costo de producción, el cual debe ser justificado en base a los beneficios reportados por el mismo. El sistema que se propone en este trabajo no conlleva a grandes gastos, puesto que solo es influyente el salario de los desarrolladores, por lo cual se concluye que su implementación es factible. Esto se debe en gran medida a la utilización de herramientas libres que no requieren el pago de licencia.

### 5.8 Conclusiones parciales.

En el presente capítulo se realizó un análisis de factibilidad de la solución propuesta, efectuando una comparación entre los costos de producción y los beneficios que aporta su puesta en funcionamiento se arribó a la conclusión que el desarrollo del producto SISCO es viable.

## **CONCLUSIONES.**

En el transcurso de la investigación se logró dar cumplimiento al objetivo trazado, para ello se realizó un estudio de los procesos de envío de mensajes existentes en la universidad, se modeló una sistema que unifica los servicios de correo electrónico, mensajería instantánea, telefonía móvil y telefonía fija. Se desarrolló una aplicación web que permite a los usuarios que accedan a la misma para gestionar sus clientes, contactos y listas de contactos; además de enviar mensajes por las diferentes vías disponibles y se realizaron pruebas al sistema para verificar que cumpliera con los requerimientos especificados por el cliente.

La solución desarrollada permite mayor información de los usuarios, mejor comunicación entre los emisores del mensaje y los receptores, además de contribuir a mitigar la tardía comunicación entre los usuarios, ya que al enviarse por todas las vías es menos probable que el mensaje no llegue a su destinatario.

## **RECOMENDACIONES.**

Con el objetivo de que el sistema tenga un elevado índice de utilización se recomienda incorporar otras funcionalidades, como por ejemplo: permitir no solo el envío sino también la recepción de mensajes desde la propia plataforma, guardar un registro con los mensajes enviados por cada usuario que acceda al sistema, así como programar el envío de los mensajes y añadirle la posibilidad de una configuración visual de los diferentes servidores a utilizar.

## REFERENCIAS BIBLIOGRÁFICAS

1. Sitio Oficial de la Universidad de las Ciencias Informáticas. [En línea] UCI. Disponible [http://www.uci.cu].
2. Telefonía fija. [En línea] Disponible en: [http://es.wikitel.info/wiki/Telefon%C3%ADa\_fija].
3. Qué es la Telefonía IP. *Universidad de Chile: Dirección de Servicios de Tecnologías de Información*. [En línea] Disponible en: [http://www.telefoniaip.uchile.cl/capacitacion\_telefonia.htm].
4. Mensajes de Texto (SMS). *Entel*. [En línea] Disponible en: [http://personas.entelpcs.cl/PortalPersonas/appmanager/entelpcs/personas?\_nfpb=true&\_pageLabel=P3001662871250439532266].
5. Integración de Comunicaciones Unificadas de Microsoft. *Servicios de Integración con Microsoft Comunicaciones Unificadas*. [En línea] Disponible en: [http://www.teldat.com/es/page.php?cnt\_id=servicios-integracion-microsoft-comunicaciones-unificadas].
6. Unifica e-mail, sms y mensajería instantánea. Facebook lanza sistema todo-en-uno de correo electrónico. [En línea] 2010. Disponible en: [http://diarioti.com/gate/n.php?id=28117].
7. **Muñoz, Alfio**. *Elastix A Ritmo de Merengue*. 2009-2010.
8. Desarrollo en Web. Blog sobre desarrollo de aplicaciones web en Java, Phython y JavaScript. [En línea] Disponible en: [http://blogs.antartec.com/desarrolloweb/2008/10/extjs-lo-bueno-lo-malo-y-lo-feo/].
9. Útiles herramientas para desarrolladores de PHP. [En línea] Disponible en: [http://www.tecnovi.net/2009/08/utiles-herramientas-para-desarrolladores-de-php].
10. **Leopoldo, Carlos**. Zend Framework, una introducción. *techtastico*. [En línea] 27 de Noviembre de 2007. Disponible en: [http://techtastico.com/post/zend-framework-una-introduccion/].
11. Más sobre Zend Framework. [En línea] Disponible en: [http://www.proyectosbds.com/software-y-programacion-a-medida/programacion-php-lamp-zf/mas-sobre-zend-framework/121/].
12. **García Hernández, Alién, Viltres Ramírez, Damián**. *Implementación de los Componentes Configuración, Nomencladores y Recuperaciones del subsistema Inventario del Sistema de Gestión Integral CEDRUX*. 2009.
13. Prácticas de Extreme programming: Manejo de proyectos, recargado. [En línea] Disponible en: [http://www.sg.com.mx/sg2011/sessions/pr%C3%A1cticas-de-extreme-programming-manejo-de-proyectos-recargado].
14. ¿Qué es un Sistema Gestor de Bases de Datos o SGBD? [En línea] Disponible en: [http://www.cavsi.com/preguntasrespuestas/que-es-un-sistema-gestor-de-bases-de-datos-o-sgbd/].
15. **Espinoza, Humberto**. PostgreSQL Una Alternativa de DBMS Open Source. [En línea] 2005. Disponible en: [http://www.lgs.com.ve/pres/PresentacionES\_PSQL.pdf].

Referencias Bibliográficas

16. Lenguajes de Programación. [En línea] 2009. Disponible en: [www.lenguajes-de-programacion.com].
17. Curso online Master en Programacion Web. [En línea] Disponible en: [http://www.ciberaula.com/curso/masterprogramacionweb/].
18. Cursos de creacion y diseño de paginas web. [En línea] Disponible en: [http://www.cursopaginasweb.net/curso-paginas-web/introduccion-javascript.htm].
19. Ventajas y Desventajas de Apache Web Server. [En línea] Disponible en: [http://www.cerocerouno.com.ar/?p=228].
20. **Billy, Carlos.** Introducción a la Arquitectura de Software. [En línea] Disponible en: [http://www.willydev.net/descargas/prev/IntroArq.pdf].
21. Entorno cliente/servidor. [En línea] Disponible en: [http://es.kioskea.net/contents/cs/csintro.php3].
22. **Gamma, Erich y otros.** Design Patterns: Elements of Reusable Object-Oriented Software.
23. **Ilizastegui Arriba, Damián y Plá Rodríguez, José Antonio.** *Sistema para la integración continua de proyectos y el control de builds en la empresa Procyon Soluciones.* Ciudad de la Habana : s.n., 2007.
24. **Tuya, Javier, Ramos Román, Isabel y Dolado Cosín, Javier.** *Técnicas cuantitativas para la gestión en la ingeniería del software.* 2007.
25. **Joskowicz, Ing. José.** Reglas y Prácticas en eXtreme Programming. [En línea] 2008. Disponible en: [http://iie.fing.edu.uy/~josej/docs/XP%20-%20Jose%20Joskowicz.pdf].
26. **Moreno, Ana M<sup>a</sup>.** Estimación de Proyectos Software: COCOMO II. [En línea] Disponible en: [http://www.eici.ucm.cl/Academicos/ygomez/descargas/Ing\_Sw2/apuntes/cocomo\_manual\_espanol.pdf].