

**Universidad de las Ciencias Informáticas**  
Facultad 2



**Título: “Sistema de servicio de SMS basado en Gammu (JAMS)”**

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

**Autor:** José Alexis Hernández Ricardo

**Tutor(es):** Ing. Darién Jesús Álvarez de la Cruz.  
Ing. Yaneisy Cruz Navarro.

**La Habana, Junio 2011**

*“ Los grandes espíritus siempre han encontrado una  
violenta oposición de parte de mentes mediocres. ”*

*Albert Einstein*

## DECLARACIÓN DE AUTORÍA

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

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

---

José Alexis Hernández Ricardo  
**Autor**

---

Ing. Darién Jesús Álvarez de la  
Cruz  
**Tutor**

---

Ing. Yaneisy Cruz  
Navarro  
**Tutor**

## **AGRADECIMIENTOS**

Quisiera agradecer en general a todas las personas a mi alrededor. Primeramente a todos los que me ayudaron, haciendo posible que mi esfuerzo valiera la pena y en segundo lugar a todo el que no me ayudó, al que me subestimó, menospreció o sencillamente me ignoró, pues me forzó a superarme a mí mismo.

Gracias a mis amigos, los de antes, los de ahora y los de siempre, a mis profesores, a los que me han enseñado y los que me han educado.

Gracias especiales a mi novia, que ha sido la luz que me ha guiado por buenos caminos desde que la conocí, a mi padrastro, que se ha convertido en una parte incondicional de mi familia, a mi hermana que siempre ha estado ahí para mí y sobre todo a mi madre que ha sido mi ejemplo a seguir desde que tengo uso de razón.

## **DEDICATORIA**

Dedico este trabajo a mi madre, que ha sido la base donde me he apoyado para alcanzar mis metas, que ha sufrido, llorado, reído, disfrutado y vivido como suyo cada uno de mis días, que me ha enseñado que lo importante no es caer, sino saber levantarse y no con una frase vacía, sino con cada una de sus acciones. A ti madre querida te dedico este trabajo y con él, mi futuro.

## **RESUMEN**

La telefonía y específicamente la telefonía celular, es uno de los medios de comunicación más utilizado en nuestros días. La red GSM presta distintos servicios a los dispositivos interconectados en ella, uno de estos servicios es el SMS. En el centro de Telemática de la Universidad de las Ciencias Informáticas, se utiliza un sistema llamado GAMMU que posibilita la administración de MODEMs GSM para enviar SMS. A pesar de que actualmente GAMMU permite el envío de SMS, presenta para la interacción con sus funcionalidades, una interfaz basada en la consola del sistema, que imposibilita el aprovechamiento al máximo de sus potencialidades, además una de las prioridades del centro, que es el envío masivo de mensajes, es imposible con la forma de interacción actual que presenta el GAMMU. Con la presente investigación se realizó un sistema para dar solución a la situación previamente planteada, creando no solo una aplicación que permita el envío masivo de SMS utilizando MODEMs GSM, sino mejorando la usabilidad del sistema utilizado en el centro.

## **PALABRAS CLAVE**

GSM, SMS, GAMMU, GAMMU SMSD, aplicación, sistema, desarrollo

---

**ÍNDICE**

<b>INTRODUCCIÓN</b> .....	1
<b>CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA</b> .....	6
<b>1.1 Introducción</b> .....	6
<b>1.2 MODEM GSM</b> .....	6
<b>1.3 Sistemas de envío masivo de SMS utilizando MODEM GSM</b> .....	6
1.3.1 <i>SMS masivos</i> .....	7
<b>1.4 GAMMU y GAMMU SMSD</b> .....	8
<b>1.5 Aplicaciones de envío masivo de SMS que utilizan GAMMU SMSD</b> .....	8
<b>1.6 Servicios Web</b> .....	10
<b>1.7 Metodologías de desarrollo de software</b> .....	11
1.7.1 <i>Metodologías de desarrollo de software: Extreme Programming</i> .....	13
1.7.2 <i>Lenguaje de modelado BPMN</i> .....	14
<b>1.8 Lenguajes de programación</b> .....	15
<b>1.9 Framework Doctrine</b> .....	16
<b>1.10 Framework Codeigniter</b> .....	17
<b>1.11 Zend Framework</b> .....	17
<b>1.12 Entorno de desarrollo: Zend Studio 8</b> .....	18
<b>1.13 Herramienta CASE: BizAgi Process Modeler</b> .....	19
<b>1.14 Servidor Web: Apache 2</b> .....	19
<b>1.15 Sistema gestor de bases de datos: PostgreSQL 8.4.7</b> .....	20
<b>1.16 Conclusiones</b> .....	22
<b>CAPÍTULO 2: EXPLORACIÓN Y PLANIFICACIÓN</b> .....	23
<b>2.1 Introducción</b> .....	23
<b>2.2 Objeto de Automatización</b> .....	23
<b>2.3 Propuesta del Sistema</b> .....	23
<b>2.4 Flujo de procesos del sistema propuesto</b> .....	25
<b>2.5 Usuarios relacionados con el sistema</b> .....	26
<b>2.6 Características no funcionales del sistema</b> .....	27
<b>2.7 Fase de Exploración</b> .....	29
2.7.1 <i>Historias de Usuarios</i> .....	30
<b>2.8 Fase de Planificación</b> .....	36
2.8.1 <i>Estimación de esfuerzo por historias de usuarios</i> .....	36
2.8.2 <i>Plan de Iteraciones</i> .....	38
2.8.3 <i>Plan de duración de las iteraciones</i> .....	39
2.8.4 <i>Plan de entregas</i> .....	40
<b>2.9 Conclusiones</b> .....	40
<b>CAPÍTULO 3: DISEÑO DEL SISTEMA</b> .....	41

<b>3.1</b>	<b>Introducción</b>	41
<b>3.2</b>	<b>Arquitectura</b>	41
3.2.1	<i>Arquitectura Utilizada</i>	42
<b>3.3</b>	<b>Patrones de Diseño</b>	43
3.3.1	<i>Patrones para Asignar Responsabilidades (GRASP)</i>	43
3.3.2	<i>Patrones GOF</i>	44
<b>3.4</b>	<b>Tarjetas Cargo o Clase, Responsabilidad y Colaboración (CRC)</b>	44
<b>3.5</b>	<b>Diseño de la Base de datos</b>	45
3.5.1	<i>Modelo Entidad-Relación</i>	46
<b>3.6</b>	<b>Conclusiones</b>	46
<b>CAPÍTULO 4: IMPLEMENTACIÓN Y PRUEBAS</b>		48
<b>4.1</b>	<b>Introducción</b>	48
<b>4.2</b>	<b>Fase de Implementación</b>	48
4.2.1	<i>Primera Iteración</i>	48
4.2.2	<i>Segunda Iteración</i>	49
4.2.3	<i>Tercera Iteración</i>	49
4.2.4	<i>Cuarta Iteración</i>	50
4.2.5	<i>Quinta Iteración</i>	51
<b>4.3</b>	<b>Fase Prueba</b>	51
4.3.1	<i>Prueba Unitaria</i>	52
4.3.2	<i>Prueba Aceptación</i>	52
<b>4.4</b>	<b>Conclusiones</b>	54
<b>CAPÍTULO 5: ESTUDIO DE FACTIBILIDAD</b>		55
<b>5.1</b>	<b>Introducción</b>	55
<b>5.2</b>	<b>Modelo COCOMO II</b>	55
<b>5.3</b>	<b>Características del Proyecto</b>	55
5.3.1	<i>Entradas Externas</i>	55
5.3.2	<i>Salidas Externas</i>	56
5.3.3	<i>Consultas Externas</i>	57
5.3.4	<i>Archivos lógicos internos</i>	58
5.3.5	<i>Archivos de interfaz externos</i>	58
5.3.6	<i>Puntos de función desajustados</i>	58
<b>5.4</b>	<b>Cálculos de instrucciones</b>	59
<b>5.5</b>	<b>Cálculo de esfuerzo nominal</b>	60
<b>5.6</b>	<b>Ajuste del esfuerzo nominal</b>	62
5.6.1	<i>Ajuste del esfuerzo nominal clasificado</i>	62
5.6.2	<i>Cálculo del tiempo de desarrollo del software</i>	63
5.6.3	<i>Cálculo del costo total del proyecto</i>	64
5.6.4	<i>Resultados</i>	65
<b>5.7</b>	<b>Análisis de costo</b>	66

---

5.8 Conclusiones.....	66
CONCLUSIONES .....	67
RECOMENDACIONES.....	68
REFERENCIAS BIBLIOGRÁFICAS .....	69
GLOSARIO DE TÉRMINOS.....	72



## **INTRODUCCIÓN**

Con la evolución del ser humano también han evolucionado sus formas de comunicación. La introducción de las tecnologías influyeron definitivamente en el proceso de transmitir las ideas humanas, dando lugar en la actualidad, a los modernos sistemas de transmisión de información y más específicamente a los sistemas de transmisión a distancia, pudiendo mencionar entre estos el telégrafo, televisión, radio y el teléfono. [1]

Entre los medios de comunicación a distancia actuales, uno de los más usados es el teléfono. Desde su invención a mediados de los años 1876 por Alexander Graham Bell este medio ha evolucionado increíblemente rápido y se ha adaptado a prácticamente cualquier necesidad de comunicación existente en la sociedad actual. A partir de que se inventara el primer teléfono celular en 1973 por el Dr. Martin Cooper, gerente general de sistemas de Motorola, la telefonía celular ha evolucionado hasta convertirse en una de las vías de comunicación más usadas en la actualidad, llegando a implementar opciones, no solo de comunicación vocal como sus predecesores creados por Bell, sino que incluso ha permitido transmisión de texto, imágenes y video.[2]

Para la transmisión de los distintos tipos de datos a través de la telefonía celular se han definido varios estándares, el GSM (del inglés: Global System for Mobile Communications) fue completamente definido para la comunicación mediante dispositivos móviles. Unos de los principales servicios que provee este estándar es el envío de mensajes de texto o SMS (del inglés: Short Message Service). Para gestionar este servicio de mensajes existen los SMSC (del inglés: Short Message Service Center) y las SMS Gateway cuyas funciones, en ambos casos con diferencias, son las de recibir, responder, enviar y reenviar SMS, y por lo tanto pueden actuar como un “puente” entre suscriptores encargándose de realizar estas gestiones, además en caso de que no sea entregable el mensaje pueden realizar un conjunto de acciones como pueden ser informar su pérdida, reportar fallos y reenviarlo en caso de que el destinatario no esté disponible en ese momento. [3][4]

En el centro de TLM (Telemática) de la UCI (Universidad de las Ciencias Informáticas), se desarrollan software que tienen como una de sus principales funciones el envío de SMS, para lo cual se apoyan en sistemas que poseen funcionalidades similares a las de los SMS Gateway, uno de estos es GAMMU, el

cual está definido por sus creadores como un sistema que abarca las aplicaciones, scripts y los drivers para la gestión de diversas funciones en los teléfonos celulares y dispositivos similares [5].

Actualmente el sistema Gammu se utiliza para brindar un servicio de envío de mensajes utilizando MODEM GSM (del inglés: Modulator Demodulator) complementado por la utilización del componente GAMMU SMSD (del inglés: GAMMU SMS Daemon) y aunque dicho sistema permite su administración y configuración y la del GAMMU SMSD y de sus distintos parámetros, esta administración se realiza a nivel de consola, introduciendo manualmente los comandos en la misma, lo cual dificulta grandemente el aprovechamiento de sus funcionalidades. Así mismo, el bajo nivel de administración imposibilita el uso de ciertas características requeridas para el tipo de servicio que se desea crear con el sistema, como el envío masivo de SMS.

A partir de la situación descrita se deriva el siguiente **problema a resolver** ¿Cómo lograr el envío masivo de SMS utilizando MODEMs GSM?

Partiendo de lo expresado en el problema a resolver es posible establecer como **objeto de estudio** los sistemas que implementen mecanismos de envío de SMS utilizando MODEMs GSM enmarcando al **campo de acción** a los sistemas que utilicen como base el Gammu SMSD para lograr el envío masivo de SMS.

El **objetivo general** de la investigación es desarrollar una aplicación que permita la gestión de servicios de envío de mensajes masivos SMS que utilice como base el sistema Gammu SMSD. Se derivan de este los siguientes **objetivos específicos**:

- Agregar al sistema de almacenamiento de datos del sistema Gammu la posibilidad de permitir funcionalidades de administración de mensajes, usuarios, servicios y dispositivos.
- Crear la posibilidad de gestionar usuarios, servicios, mensajes y dispositivos utilizando el sistema Gammu y sus modificaciones, a través de una interfaz web.
- Crear un sistema sencillo de control de los procesos de envío de mensajes del sistema a través de una interfaz web.

- Lograr mostrar la consulta de estadísticas del envío de mensajes a través de una interfaz web.
- Crear un servicio web que permita la utilización de las funcionalidades del sistema desde un ambiente distribuido.
- Implementar un mecanismo de seguridad para el acceso a las funcionalidades del sistema.

Para guiar el cumplimiento de los objetivos propuestos se definen como **tareas de la investigación**:

- Análisis sobre el funcionamiento del sistema de almacenamiento de datos de la aplicación Gammu.
- Análisis de las acciones que debe realizar el sistema de gestión para garantizar las gestiones de usuario, servicios, mensajes y dispositivos.
  - Selección de las distintas funcionalidades a incluir en la aplicación.
- Análisis del funcionamiento del proceso de envío de mensajes en el sistema Gammu.
- Análisis del funcionamiento del framework Codeigniter para la creación de aplicaciones web de gestión y servicio web.
- Análisis de los parámetros a representar en la generación de estadísticas de mensajes.
  - Selección de los parámetros a representar en las estadísticas.
- Análisis de las distintas características del funcionamiento de los servicios web en PHP.
  - Selección de la opción más viable de integración de servicio web con el framework Codeigniter.
- Análisis de la ejecución de líneas de comando en Linux desde PHP.
  - Selección de los comandos adecuados para la interacción con Gammu desde PHP.
- Análisis los métodos de seguridad adecuados para implementar en la aplicación web y en el servicio web de la misma.

Se ha determinado establecer las siguientes **preguntas científicas**:

¿Existe algún sistema de administración compatible con Gammu que se pueda utilizar para la solución?

¿Cuáles son las funciones similares de las Gateway que se pueden utilizar en la aplicación?

¿Cómo controlar los procesos del Gammu desde una aplicación externa?

¿Cómo PHP interactúa con la consola de Linux?

¿Cómo implementar las medidas necesarias para la seguridad de los servicios brindados por una aplicación web y su servicio web?

Para la realización de la investigación se utilizaron varios **métodos teóricos** como el **analítico-sintético** que se utilizó para analizar el servicio de envío masivo de SMS requerido por el Centro de Telemática e identificar los diferentes aspectos a tener en cuenta a la hora de crear un sistema que cubriera sus necesidades. También se utilizó el **inductivo-deductivo** para lograr entender los procesos realizados por la aplicación Gammu para el envío masivo de mensajes SMS. Otro método científico utilizado fue el **análisis histórico-lógico** para aclarar conceptos e incertidumbres referentes a la aplicación Gammu a partir del avance histórico de las diferentes versiones de dicha aplicación. También se utilizaron **métodos empíricos** como la **observación**, el cual fue muy útil a la hora de adquirir experiencia y conocimiento sobre el funcionamiento de los distintos sistemas de envío masivo de SMS, y el **experimento** que aportó al equipo de trabajo, diversas experiencias para la utilización adecuada de tecnologías como servicio web y ORM (del inglés: Object Relational Mapper).

El contenido está estructurado de la siguiente forma:

**Capítulo 1 Fundamentación Teórica:** Contiene un análisis a partir de la investigación realizada sobre las tecnologías inmersas en la solución. Además se fundamentan los conceptos, metodologías, lenguajes, y herramientas utilizadas.

**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 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:** Contiene un estudio de factibilidad del sistema, teniendo en cuenta fundamentalmente el factor tiempo de trabajo.

## **CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA**

### **1.1 Introducción**

En este capítulo se describen las diversas tecnologías y conceptos asociados al tema del presente trabajo como son los sistemas de envío de mensajes por GSM, las pasarelas de mensajería móvil o SMS Gateway, la aplicación Gammu y GAMMU SMSD, los sistemas de envío de SMS basados en GAMMU SMSD y el mecanismo de acceso a datos servicio web. También se fundamentan las distintas herramientas, técnicas, metodologías y lenguaje utilizados para dar solución al problema planteado anteriormente.

### **1.2 MODEM GSM**

Los MODEMs GSM son tipos especializados de los MODEMs que funcionan sobre redes inalámbricas basados en suscripción, de forma similar a un teléfono móvil. Un MODEM GSM acepta una tarjeta SIM (del inglés: Subscriber Identity Module) y básicamente, actúa como un teléfono móvil para un equipo. El mismo pudiera incluso ser un teléfono móvil dedicado para que el equipo utilizara las capacidades de la red GSM.

Los MODEMs tradicionales se unen a los equipos para permitir conexiones de acceso telefónico a otros sistemas informáticos, un módem GSM opera de manera similar, excepto que envía y recibe datos a través de las ondas de radio en lugar de una línea telefónica. Este tipo de MODEM puede ser un dispositivo externo conectado a través de un USB (del inglés: Universal Serial Bus) o un cable serie.

Similar a la forma en que lo hacen los MODEMs convencionales, que convierten las señales analógicas en digitales y viceversa, los MODEMs GSM convierten los datos digitales de los SMS para enviar y recibir dichos mensajes a través de la red inalámbrica. [24]

### **1.3 Sistemas de envío masivo de SMS utilizando MODEM GSM**

Los sistemas de envío masivo de SMSs, tienen en internet una amplia aceptación y utilización hoy en día. Las distintas empresas que cuentan con extensas cantidades de personal acuden a los servicios de estos sistemas para agilizar los procesos de entrega de mensajes a su personal o incluso para hacer gestiones comerciales de gran envergadura. De los muchos existentes en este momento en internet se decidió

analizar uno en específico que cuenta con características similares a las que JAMS quiere lograr, este sistema es SMS Masivos, ubicado en Argentina. [6]

### **1.3.1 SMS masivos**

SMS masivos es una plataforma Web para envío y recepción de mensajes de texto personalizados, que radicada en Argentina. Esta plataforma brinda al cliente la posibilidad de enviar y recibir mensajes de texto SMS de forma masiva desde cualquier computadora o dispositivo móvil con acceso a Internet. El sistema cuenta con dos tipos de planes de servicio, el plan abierto que consta de un abono mensual, el cual se acredita en SMS a la cuenta del cliente mes a mes y una vez consumidos los SMS incluidos en el paquete el cliente podrá seguir enviando mensajes libremente a precio excedente y el plan pre-pago, que brinda la posibilidad de administrar los consumos de acuerdo con sus necesidades, sin abono mensual, se habilitará la cantidad de SMS contratados y finalizará el servicio cuando se cumpla el vencimiento del paquete o cuando se haya consumido la totalidad de los mensajes (lo que primero suceda). Ambos planes cuentan con paquetes de 500 hasta 10000 mensajes SMS mensuales. [6]

Entre las ventajas de este sistema se encuentran:

- Personalización de los mensajes
- Envío por grupos de contacto
- Programación de envío día/hora
- Sistema de envío sencillo vía web
- Simulación de envío de mensajes
- Visualización de respuestas
- Cobertura nacional sin límites
- Importación de datos
- Seguridad de entrega y lectura

Este sistema cuenta con clientes empresariales y gubernamentales que utilizan sus servicios a diario para el envío de miles de mensajes de SMS e integran dichos servicios con las mayores empresas de redes de telefonía celular. [6]

#### **1.4 GAMMU y GAMMU SMSD**

El sistema Gammu descrito por su creador Michal Cihar como el proyecto de una utilidad de línea de comandos, que se puede utilizar para controlar el teléfono celular y dispositivos similares. Gammu es un sistema construido en C y con excelentes prestaciones para el manejo de dispositivos que utilizan el estándar de GSM. Incluido en el paquete que comprende GAMMU se puede encontrar el GAMMU SMSD definido por su creador como un programa que explora periódicamente los MODEM GSM conectados a su sistema, para obtener los mensajes recibidos por los mismos, almacenarlos en un medio definido y también enviar los mensajes en la cola de salida de dicho medio de almacenamiento. Es la herramienta perfecta para gestionar grandes cantidades de mensajes recibidos o enviados y automatizar dichos procesos. Disponible solo para sistemas basados en UNIX, GAMMU SMSD cuenta con un desarrollo bien fuerte por parte de sus creadores, sacando versiones que corrigen errores casi una vez por mes y contando con una estructura simple que agiliza los procesos del programa. El paquete de GAMMU es compatible con la mayoría de los sistemas de base de datos que existen actualmente, contiene nativas las librerías para la interacción con MySQL y PostgreSQL y puede conectarse con casi todos sistemas de base de datos a través de la librería libdbi.[13]

Entre las principales características del sistema GAMMU al interactuar con dispositivos móviles se pueden encontrar: [13]

- Anuncio, inicialización y manejo de llamadas.
- Recibo, envío y copias de respaldo de mensajes SMS.
- Recuperación de información de dispositivos y sus redes disponibles.
- Accesos a los almacenamientos de los dispositivos conectados al sistema.
- Manejo de múltiples dispositivos de GSM

#### **1.5 Aplicaciones de envío masivo de SMS que utilizan GAMMU SMSD**

Existen varias aplicaciones de interfaz para interactuar con GAMMU y con GAMMU SMSD como son Kalkun y PlaySMS de estas, es muy recomendado el uso de Kalkun, por los propios creadores de GAMMU, por lo cual fue analizada para definir si ajustando sus funcionalidades podía ser utilizada o



reutilizada para el cumplimiento de los objetivos antes propuestos en la presente investigación. Algunas de las características presentes en el sistema Kalkun en el momento de esta investigación son: [14]

- Permite interacción con varios tipos de almacenamiento de datos como MySQL y SQLite3
- Permite el envío programado de SMS
- Permite el uso del sistema de listas de direcciones integrados en GAMMU para el envío de SMS a varios destinatarios
- Interacción con el proceso de envío de SMS
- Administración de usuarios
- Administración de grupos

Estas características del Kalkun a pesar de ajustarse a varias de las necesidades del sistema que se desea desarrollar para el Centro de Telemática, también presentan ciertos inconvenientes que hacen que dicho sistema no sea aplicable a la solución, estos inconvenientes son: [14]

- No permite la interacción con el sistema de base de datos PostgreSQL que es el actualmente utilizado en el centro de telemática
- El sistema de listas de direcciones es administrado a través del sistema GAMMU y no de la propia aplicación
- El envío programado de SMS solo presenta funciones de inicio y fin de envío de manera automática y no permite establecer otras especificaciones
- No permite la creación por parte de los usuarios de sus propias listas de envío de SMS
- La administración de grupos y usuarios esta delegada a los propios usuarios y al administrador, ambos tienen funcionalidades que les permiten gestionar estas características

Además el sistema también tiene otros inconvenientes implícitos que no permiten su uso para la realización de la solución de esta investigación, como son: [14]

- No cuenta con un sistema para balancear la carga del envío de SMS distribuyendo entre los diversos dispositivos con los que cuenta el GAMMU SMSD
- El administrador no puede definir servicios con distintas características para agrupar a los distintos usuarios

Aun así Kalkun es un buen punto de referencia para la ejecución de funciones básicas como la interacción con el envío de SMS del GAMMU SMSD y la gestión de usuarios.

## **1.6 Servicios Web**

Durante los últimos años, las empresas han utilizado la tecnología para mejorar sus procesos de negocio, incorporando sistemas de gestión, atención al cliente y otras aplicaciones que automatizan estos procesos. Pero no es suficiente, actualmente es necesario contar con la capacidad necesaria para permitir que los sistemas conversen entre sí, estén integrados y operen de forma amigable, como si fueran uno solo, por lo que, uno de los mecanismos para lograr tal comunicación son los Servicios Web.

Existen varias definiciones de Servicios Web, esto muestra una complejidad a la hora de dar una adecuada definición que englobe lo que son y lo que implican; por ejemplo, se pueden definir como un conjunto de aplicaciones o de tecnologías para intercambiar información en la Web; las cuales intercambian datos entre sí con el objetivo de ofrecer servicios. Los proveedores ofrecen sus servicios como procedimientos remotos y los clientes solicitan un servicio determinado llamando a estos procedimientos a través de la web. [7]

Estos servicios proporcionan mecanismos de comunicación estándares entre diferentes aplicaciones, que interactúan entre sí para presentar una información dinámica al cliente.

De acuerdo con la W3C (del inglés: World Wide Web Consortium), un Servicio Web es “...un sistema de software diseñado para soportar interacción interoperable máquina a máquina sobre una red. Este tiene un interface descrita en un formato procesable por una máquina, específicamente WSDL (del inglés: Web Services Description Language). Otros sistemas interactúan con el servicio web en una manera prescrita por su descripción usando mensajes SOAP (del inglés: Simple Object Access Protocol), típicamente enviados usando mensajes HTTP (del inglés: Hypertext Transfer Protocol), con una serialización XML (del inglés: Extensible Markup Language) en relación con otros estándares relacionados con la web.” [7]

Los Servicios Web aportan interoperabilidad entre aplicaciones de software independientemente de sus propiedades o de las plataformas sobre las que se instalen. Fomentan los estándares y protocolos basados en texto, que hacen más fácil acceder a su contenido y entender su funcionamiento. Al apoyarse en HTTP, los Servicios Web pueden aprovecharse de los sistemas de seguridad *firewall* sin necesidad de cambiar las reglas de filtrado. Permiten que servicios y softwares de diferentes compañías ubicadas en diferentes lugares geográficos puedan ser combinados fácilmente para proveer servicios integrados.

Permiten la interoperabilidad entre plataformas de distintos fabricantes por medio de protocolos estándares y abiertos; las especificaciones de dichos servicios son gestionadas la W3C, por tanto no hay secretismos por intereses particulares de fabricantes concretos y se garantiza la plena interoperabilidad entre aplicaciones.

Para el correcto funcionamiento de un Servicio Web es necesaria la intervención de varias tecnologías que hacen posible la circulación de la información en la red. Por un lado estaría el protocolo SOAP que especifica el formato de los mensajes. Dicho protocolo está basado en el lenguaje XML, que permite la interacción entre varios dispositivos y que tiene la capacidad de transmitir información compleja. Los datos pueden ser transmitidos a través de los protocolos HTTP, SMTP (*Simple Mail Transfer Protocol*), etc.

Por todas estas características se escogió a los Servicios Web para el intercambio de información y brindar servicio de acceso datos. [7] [8]

### **1.7 Metodologías de desarrollo de software**

Hacer un software resulta algo complejo, para ello se requiere de una guía o un plan de trabajo que el equipo de desarrollo se debe trazar o planificar, y así de tal forma que el producto o resultado del trabajo tenga éxito, o sea que se cumplan los objetivos y los requerimientos, se entregue en tiempo, forma y con la calidad que requiere o que el cliente exija. Para lograr cumplir todo esto con una mayor facilidad de trabajo, existen las metodologías de desarrollo.

Una metodología de desarrollo de software en ingeniería de software es un marco de trabajo usado para estructurar, planificar y controlar el proceso de desarrollo en sistemas de información. [8]

Una metodología puede seguir uno o varios modelos de ciclo de vida, es decir, el ciclo de vida indica qué es lo que hay que obtener a lo largo del desarrollo del proyecto pero no cómo hacerlo. La metodología indica cómo hay que obtener los distintos productos parciales y finales. Debe ajustarse a los objetivos y cubrir el ciclo entero de desarrollo de software. [9]

La metodología de desarrollo de software debe: [9]

- Integrar las distintas fases del ciclo de desarrollo
- Incluir la realización de validaciones.
- Soportar la determinación de la exactitud del sistema a través del ciclo de desarrollo.
- Ser la base de una comunicación efectiva.

- Funcionar en un entorno dinámico orientado al usuario
- Especificar claramente los responsables de los resultados
- Poder emplearse en un entorno amplio de proyectos software
- Soportar la eventual evolución del sistema.
- Contener actividades que conducen a mejorar el proceso de desarrollo de software.

Cada aproximación al desarrollo de software está basada en ciertos objetivos, dichos objetivos deben estar implementados en la metodología de que se decida utilizar para desarrollar el software. Existen fundamentalmente dos tipos de metodologías para el desarrollo de software, las ágiles y las tradicionales, con ciertas diferencias en entre las mismas, ver Tabla1.

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

Tabla 1. Comparación entre las metodologías ágiles y las tradicionales. [9]

Entre las metodologías tradicionales o pesadas, se encuentra RUP (del inglés: Rational Unified Process), un proceso iterativo e incremental. Esta metodología presenta desventajas para su aplicación. Algunas de estas son:

- La generación de una gran cantidad de documentación que no genera valor respecto a la calidad del desarrollo;
- La necesidad de incluir a más personas en el equipo de desarrollo, tales como especialistas en los diseños de los casos de usos. [10]

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 (del inglés: Extreme Programming) y SCRUM. [10]

### **1.7.1 Metodologías de desarrollo de software: Extreme Programing**

Extreme Programing 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. 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. Su fortaleza radica en tener como parte del equipo, al usuario final, pues es uno de los requisitos para llegar al éxito del proyecto. [9]

Esta metodología consta de 7 fases: [9]

- Exploración: En esta fase, los clientes plantean a grandes rasgos las historias de usuario que son de interés para la primera entrega del producto, mientras que el equipo de desarrollo se familiariza con las herramientas, tecnologías y prácticas que se utilizarán en el proyecto.
- Planificación de la Entrega: En esta fase el cliente establece la prioridad de cada historia de usuario, mientras que los programadores realizan una estimación del esfuerzo necesario de cada una de ellas.
- Iteraciones: Esta fase incluye las iteraciones que se van a realizar sobre el sistema, antes de ser entregado.

- **Producción:** En esta fase se requieren pruebas adicionales y revisiones de rendimiento antes de que el sistema sea trasladado al entorno del cliente.
- **Mantenimiento:** Esta fase contempla el mantenimiento del sistema luego de que la primera versión se encuentre en producción.
- **Muerte del Proyecto:** Esta fase es la última y presenta vital importancia ya que el cliente no tiene más historias para ser incluidas en el sistema, por lo que se deben satisfacer otras necesidades de este, como son el rendimiento y confiabilidad del sistema.

### **1.7.2 Lenguaje de modelado BPMN**

El mundo de los procesos de negocio ha cambiado dramáticamente en los últimos años. Los procesos pueden ser coordinados por detrás, dentro y sobre las fronteras naturales de las organizaciones. Un proceso de negocio ahora se extiende por varios participantes y la coordinación puede ser compleja. BPMN (del inglés: Business Process Modeling Notation) ha sido desarrollado para proporcionar a los usuarios de la misma con una notación libre de derechos. Esto beneficiará a los usuarios de manera similar que el UML estándar del mundo de la ingeniería de software.

El BPMN es una notación gráfica que muestra los pasos de un proceso de negocio, en otras palabras, representa de extremo a extremo un proceso de negocio. La notación ha sido diseñada específicamente para coordinar la secuencia de los procesos y los mensajes que fluyen entre los diferentes participantes del proceso en un conjunto de actividades relacionadas.

BPMN está dirigido a un alto nivel a usuarios de negocio y en un nivel inferior para los ejecutores del proceso. Los usuarios de negocios deben ser capaces de leer y comprender fácilmente un diagrama del proceso del negocio BPMN. El ejecutor de procesos debe ser capaz de adornar un diagrama de procesos de negocio con el mayor detalle posible con el fin de representar el proceso en la implementación física.

BPMN está dirigido a usuarios, proveedores y prestadores de servicios que necesitan comunicar los procesos del negocio de una manera estándar. [22]

### **1.7.2 Metodología utilizada**

Para la realización del sistema se decidió utilizar una fusión de la metodología XP con un modelamiento de los procesos a automatizar a través de BPMN.

XP como metodología guiará el proceso de desarrollo. Esta fusión responde a la creación de un mejor entendimiento por parte del cliente de la evolución del desarrollo del sistema y por supuesto a la creación de una base más sólida que guíase dicho desarrollo.

### **1.8 Lenguajes de programación**

Para el desarrollo de la aplicación se ha decidido la utilización del lenguaje de programación PHP (del inglés: Hypertext Pre-Processor) del lado del servidor, mientras que del lado del cliente se propone utilizar los lenguajes HTML (del inglés: HyperText Markup Language) para definir los componentes de las páginas web y JavaScript para realizar validaciones de datos, crear efectos para interactuar con el usuario y otras funciones básicas.

PHP es un lenguaje de script interpretado en el lado del servidor utilizado para la generación de páginas Web dinámicas, similar al ASP (del inglés: Active Server Page) de Microsoft o el JSP (del inglés: Java Server Page) de Sun, embebido en páginas HTML y ejecutado en el servidor.

La mayor parte de su sintaxis ha sido tomada de C, Java y Perl con algunas características específicas de sí mismo. La meta del lenguaje es permitir rápidamente a los desarrolladores la generación dinámica de páginas. No es un lenguaje de marcas como podría ser HTML, XML o WML (del inglés: Wireless Markup Language). [11]

Dispone de una gran cantidad de características que lo convierten en una buena herramienta para la creación de páginas web dinámicas como son: [11]

- Soporte para una gran cantidad de bases de datos: MySQL, PostgreSQL, Oracle, MS SQL Server, Sybase mSQL, Informix, entre otras.
- Integración con varias bibliotecas externas, permite generar documentos en PDF (documentos de Acrobat Reader) hasta analizar código XML.
- Ofrece una solución simple y universal para las paginaciones dinámicas de la Web de fácil programación.

- Perceptiblemente más fácil de mantener y poner al día que el código desarrollado muchos otros lenguajes de características similares.
- Soportado por una gran comunidad de desarrolladores, como producto de código abierto, PHP goza de la ayuda de un gran grupo de programadores, permitiendo que los fallos de funcionamiento se encuentren y reparen rápidamente.
- El código se pone al día continuamente con mejoras y extensiones de lenguaje para ampliar las capacidades de PHP.
- Con PHP se puede hacer cualquier cosa que podemos realizar con un script CGI, como el procesamiento de información en formularios, foros de discusión, manipulación de cookies y páginas dinámicas.

### **1.9 Framework Doctrine**

Doctrine está definido por sus creadores como un conjunto de bibliotecas de PHP que se centra principalmente en la prestación de servicios de persistencia y sus funcionalidades relacionadas. Sus componentes principales son un ORM (del inglés: Object Relational Mapper) que funciona como la parte superior de un DBAL (del inglés: Database Abstraction Layer). El ORM presenta como característica fundamental la posibilidad de escribir consultas de base de datos en un dialecto orientado a objetos y con propiedades de SQL llamado DQL (del inglés Doctrine Query Language), inspirado en HQL (del inglés Hibernate Query Language). Esto proporciona a los desarrolladores una poderosa alternativa a SQL que mantiene la flexibilidad sin necesidad de duplicar código innecesario. Así mismo el DBAL de Doctrine está caracterizado como una potente capa de abstracción de base de datos con muchas ventajas para la introspección y gestión de distintos esquemas de bases de datos, basado en PDO (del inglés: PHP Data Object). A pesar de todas estas características de gran complejidad técnica, Doctrine cuenta con una gran comunidad de desarrolladores y documentadores que ofrecen una extensa referencia de su API (del inglés: Application Programming Interface) y todas sus funcionalidades, acortado el tiempo de realización de aplicaciones con varias técnicas que ejemplifican su uso y que demuestran la sencillez de las tareas realizadas con dicho sistema. Estas características lo presentan como un buen componente para la realización de proyectos con PHP que necesiten un alto nivel de compatibilidad con distintos sistemas de almacenamiento y que gestionen grandes volúmenes de contenidos y que a su vez necesiten



ser realizados en tiempo record; características que se evidencian en el presente trabajo por lo que se decidió el uso del mismo. [15]

### **1.10 Framework Codeigniter**

CodeIgniter está definido por sus creadores como un potente framework de PHP con un tamaño bastante pequeño, construido para programadores de PHP que necesitan una herramienta simple y elegante para crear aplicaciones web con todo tipo de funcionalidades y que cuentan con estrechas fechas de realización y entrega.[16]

Algunas de las características de Codeigniter son: [16]

- Un framework de trabajo de pequeño formato.
- Rendimiento excepcional.
- Amplia compatibilidad con el estándar de cuentas de alojamiento que se ejecutan una variedad de versiones de PHP y configuraciones.
- Requiere casi cero configuraciones.
- No requiere el uso de la línea de comandos.
- No requiere que se adhieran a las normas restrictivas de codificación.
- No está basado en las bibliotecas monolíticas a gran escala como PEAR.
- No es obligado a utilizar un lenguaje de plantillas (aunque un analizador de plantilla está opcionalmente disponible si usted desea uno).
- Evita la complejidad, favoreciendo las soluciones simples.
- Documentación clara y completa.

Todas estas características lo presentan como una buena opción para la realización de aplicaciones que tengan establecidos tiempos de realización de corta duración y que tengan que cumplir con cierta calidad para su entrega.

### **1.11 Zend Framework**

Zend Framework se trata de un conjunto de clases para el desarrollo de aplicaciones Web y servicios Web con PHP, que brinda soluciones para construir aplicaciones modernas, robustas y seguras. Además, es Open Source y trabaja con PHP 5. [24]

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 como servicios web, que de otro modo resultan ciertamente más complejos 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, patrocinadora de PHP, lo que asegura su continuidad futura tanto como la del propio lenguaje PHP. [25]

Para la realización del sistema se utilizaron componentes de este framework en la construcción del servicio web, pues ofrece clases dedicadas a ese propósito que facilitan y aseguran el trabajo que se necesita para crear dicho servicio.

### **1.12 Entorno de desarrollo: Zend Studio 8**

Zend Studio está definido por sus creadores como un IDE (del inglés: Integrated Development Environment) de PHP de nivel profesional. Diseñado para maximizar la productividad de los desarrolladores lo que permite desarrollar y mantener el código más rápido, resolver los problemas de aplicación de forma rápida y mejorar la colaboración en equipo. Zend Studio ofrece las capacidades necesarias para desarrollar aplicaciones de negocio. Características como la refactorización, la generación de código, el código de ayuda y análisis semántico se combinan para permitir el desarrollo rápido de aplicaciones tanto en el lado del servidor (en PHP) como en el lado del navegador (en JavaScript y HTML). Zend Studio proporciona depuración completa y pruebas de apoyo que ayudan a analizar y resolver problemas rápidamente. A través del apoyo tanto para la depuración local como la remota de scripts el IDE integra la inspección de código y soluciones rápidas para la generación de pruebas y presentación de informes, cuenta con herramientas necesarias para diagnosticar y resolver problemas de aplicación de forma rápida al tiempo que garantiza que el código cuenta con una correcta sintaxis, por lo que se decidió utilizar como entorno de desarrollo para la realización de la posible solución.

### **1.13 Herramienta CASE: BizAgi Process Modeler**

Bizagi, según sus creadores, es una solución de BPM (del inglés: Business Process Modeler) que permite diseñar, modelar, integrar, automatizar y monitorear los procesos de negocio por medio de un ambiente gráfico.

El Modelador de Procesos BizAgi es una herramienta de gestión de procesos ágil y fácil de utilizar, con su apariencia única y el "intelliSense behavior" es posible representar procesos rápidamente sin tener que esperar la rutina de validación al final de cada diagrama. Con BizAgi se puede diagramar y documentar procesos en el estándar BPMN sin ningún costo extra de esfuerzo, el sistema sigue una aproximación de "cero programación" la cual le permitirá crear fácilmente aplicaciones basadas en procesos, coordinando personas y sistemas, además permite crear documentos de procesos con alta calidad en Word o PDF y exportar o importar procesos desde Visio o XML.

BizAgi Process Modeler está liberado bajo licencia Freeware por lo cual su utilización, descarga y aplicación están libres de todo costo.

### **1.14 Servidor Web: Apache 2**

El Apache HTTP Server es un servidor web libre y de código abierto, es un esfuerzo de desarrollo de software de colaboración destinado a crear una implementación de código fuente robusto. El proyecto es administrado conjuntamente por un grupo de voluntarios ubicados en todo el mundo que se comunican a través de Internet para planear y desarrollar el servidor y su documentación. Este proyecto forma parte de la Apache Software Foundation. Además, cientos de usuarios han contribuido con ideas, código y documentación para el proyecto.

Apache es usado principalmente para enviar páginas web estáticas y dinámicas en la World Wide Web. Muchas aplicaciones web están diseñadas asumiendo como ambiente de implantación al servidor Apache, o para utilizar características propias de este servidor web.

Apache es el componente de servidor web en la popular plataforma de aplicaciones LAMP, junto a MySQL y los lenguajes de programación PHP/Perl/Python.

Apache es usado para muchas otras tareas donde el contenido necesita ser puesto a disposición en una forma segura y confiable. Un ejemplo es al momento de compartir archivos desde una computadora

personal hacia Internet. Un usuario que tiene Apache instalado en su escritorio puede colocar arbitrariamente archivos en la raíz de documentos de Apache, desde donde pueden ser compartidos.

La arquitectura del servidor Apache es muy modular. El servidor consta de una sección *core* y diversos módulos. [17]

Algunos de estos módulos son: [17]

- `mod_ssl` - Comunicaciones Seguras vía TLS.
- `mod_rewrite` - reescritura de direcciones (generalmente utilizado para transformar páginas dinámicas como php en páginas estáticas html para así engañar a los navegantes o a los motores de búsqueda en cuanto a cómo fueron desarrolladas estas páginas).
- `mod_auth_ldap` - Permite autenticar usuarios contra un servidor LDAP.
- `mod_proxy_ajp` - Conector para enlazar con el servidor Jakarta Tomcat de páginas dinámicas en Java.

Algunas ventajas: [17]

- Modular
- Código abierto
- Multi-plataforma
- Extensible
- Popular (fácil conseguir ayuda/soporte)

Todas estas características hacen que Apache sea el contenedor web escogido para el despliegue de la aplicación que se mostrará como la posible solución al problema tratado en la presente investigación.

### **1.15 Sistema gestor de bases de datos: PostgreSQL 8.4.7**

PostgreSQL es un sistema de gestión de bases de datos objeto-relacional, distribuido bajo licencia BSD (del inglés: Berkeley Software Distribution) y con su código fuente disponible libremente. Es el sistema de gestión de bases de datos de código abierto más potente del mercado.

PostgreSQL utiliza un modelo cliente/servidor y usa *multiprocesos* en vez de *multihilos* para garantizar la estabilidad del sistema. Un fallo en uno de los procesos no afectará el resto y el sistema continuará funcionando.

PostgreSQL funciona muy bien con grandes cantidades de datos y una alta concurrencia de usuarios accediendo a la vez al sistema.[18]

Algunas de las características más importantes y soportadas por PostgreSQL: [18]

- Integridad referencial
- Replicación asíncrona
- Copias de seguridad en caliente
- Unicode
- Juegos de caracteres internacionales
- Multi-Versión
- Múltiples métodos de autenticación
- Acceso encriptado vía SSL
- Actualización integrada
- Completa documentación
- Licencia BSD
- Disponible para Linux y UNIX en todas sus variantes y Windows 32/64bit.

Desarrollo: [18]

- Funciones/procedimientos almacenados en numerosos lenguajes de programación, entre otros PL/pgSQL, PL/Perl, PL/Python y PL/Tcl.
- Bloques anónimos de código de procedimientos.
- Numerosos tipos de datos y posibilidad de definir nuevos tipos. Además de los tipos estándares en cualquier base de datos, tipos geométricos, de direcciones de red, de cadenas binarias, XML, matrices, etc.
- Soporta el almacenamiento de objetos binarios grandes (gráficos, videos y sonido).
- APIs para programar en C/C++, Java, .Net, Perl, Python, Ruby, PHP, Lisp, Scheme, Qt y muchos otros.

Existe también una gran comunidad de usuarios, programadores y administradores que colaboran activamente en numerosos aspectos y actividades relacionadas con el proyecto, con informes y soluciones de problemas, comprobación del funcionamiento, aportaciones de nuevas ideas, discusiones sobre características y problemas, documentación y fomento de PostgreSQL. [18]

Por las características antes descritas se decidió escoger a PostgreSQL como sistema gestor de bases de datos, aunque cabe decir que el sistema a desarrollar tendrá la posibilidad de interactuar con diferentes gestores, ya que PHP y el Framework Codeigniter proveen soporte para diferentes bases de datos como MySQL, Oracle, MS SQL Server, Sybase mSQL, Informix, entre otras.

### **1.16 Conclusiones**

En el capítulo se analizaron varios sistemas de envío masivos de SMS y se determinó que aunque ninguno satisface las características que se necesitan para dar solución a la problemática planteada, si se pueden aplicar algunas de sus funcionalidades o al menos el diseño y la forma de comportarse de dichos sistemas, en la solución que se desea obtener. Así mismo se determinó la metodología a utilizar para el desarrollo del sistema y las herramientas necesarias.

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

### **2.1 Introducción**

En el presente capítulo se describen procesos y se definen los que requieren ser automatizados así como se aborda el trabajo realizado en las fases de exploración y planificación pertenecientes a la metodología de desarrollo XP, la cual forma parte de la metodología utilizada para el desarrollo del sistema propuesto. Se presenta la estructura y funcionamiento de la nueva solución mediante la propuesta del sistema. Además se definen las cualidades y características que el sistema debe cumplir, se presentan también detalladamente las historias de usuarios, para lograr mejor entendimiento de lo que se va a desarrollar y seguidamente se describen las iteraciones previstas que tendrá el proceso de desarrollo.

### **2.2 Objeto de Automatización**

El envío de SMSs desde la línea de comandos de GAMMU es un proceso engorroso y no permite el envío masivo a distintos destinatarios. Por esta razón se decidió desarrollar un sistema que facilite los procesos de administración de GAMMU y GAMMU SMSD esenciales para crear y posibilitar envíos masivos de SMS, apoyados por la creación de servicios de envíos, listas de contactos, gestión de usuarios y control de estadísticas de los mensajes.

### **2.3 Propuesta del Sistema**

El sistema tiene como propósito la creación de una aplicación web y un servicio web que permitan interactuar con el sistema gestor de bases de datos (SGDB) del sistema GAMMU para, a través de los MODEMs GSM gestionados por el mismo, crear envíos masivos de SMSs, ver Figura 1.

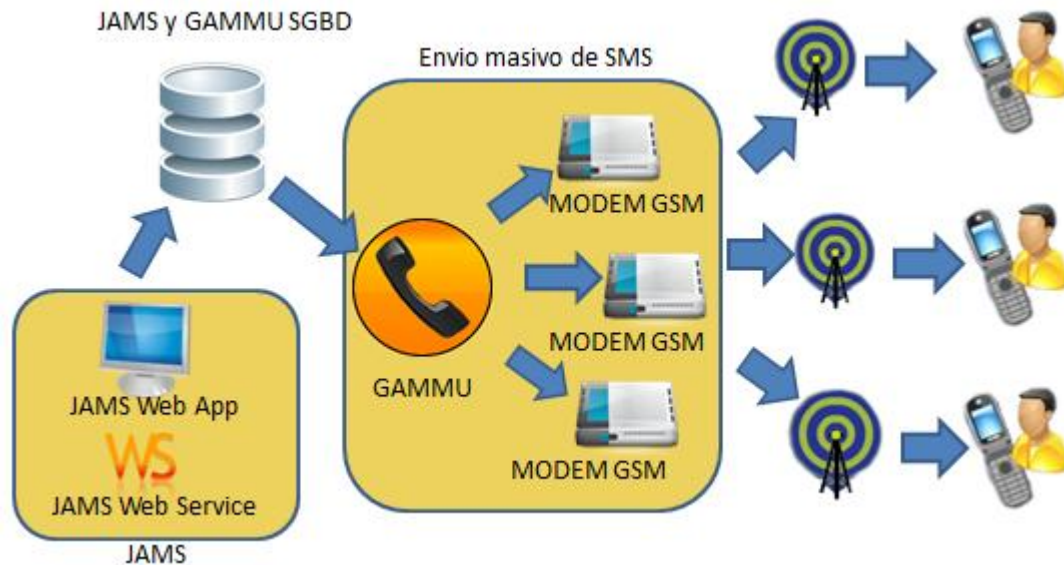


Figura 1. Propuesta general del sistema

Desde un punto de vista más objetivo, con la realización de este sistema se intenta resolver, no solo el envío masivo de SMSs utilizando para ello las listas de envío, sino que también se pretende permitir la administración de servicios para el control de dichos envíos, la posibilidad de controlar los archivos de configuración de GAMMU SMSD y por consiguiente su funcionamiento, la gestión de las listas de envío para los mensajes masivos, la presentación de gráficas sobre el comportamiento de los mensajes del sistema y gestionar los usuarios del mismo. Todas estas funcionalidades deben ser permitidas a los usuarios de la aplicación web, mientras que el servicio web ofrecerá solo una parte de ellas a sistemas externos. Estas funcionalidades se deben realizar sobre la base de la interacción con el sistema gestor de bases de datos (SGDB) de JAMS y GAMMU, ver Figura 2.



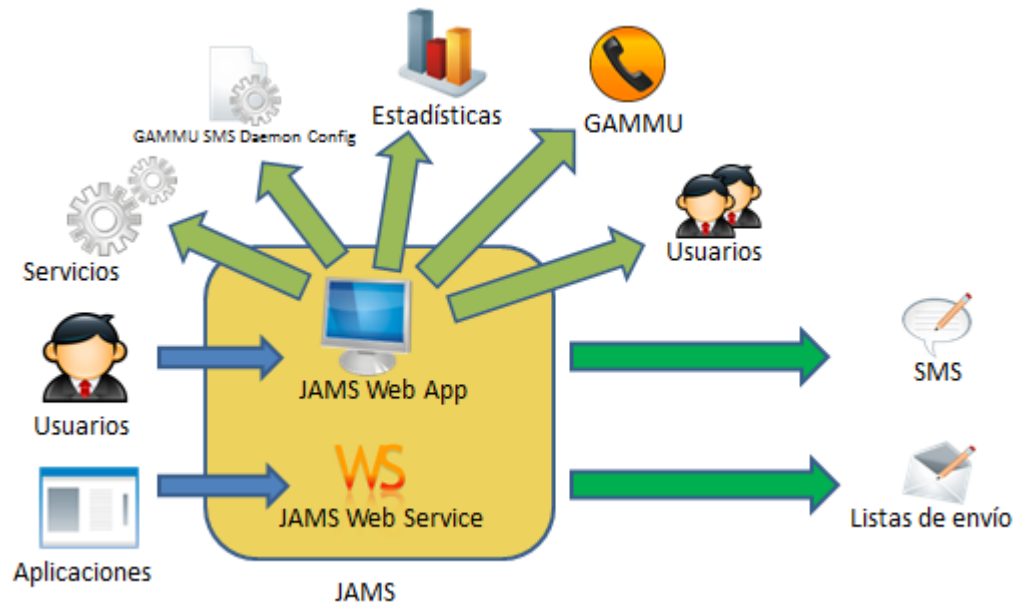
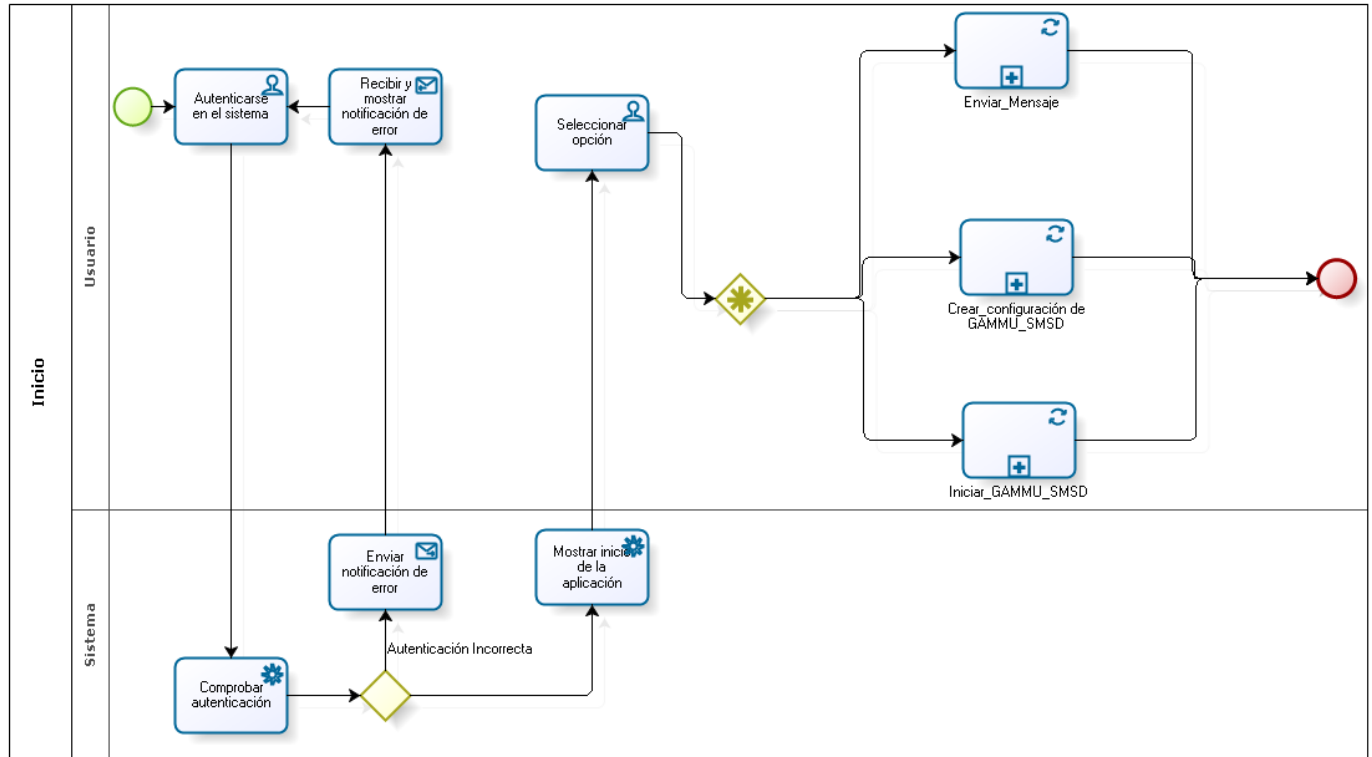


Figura2. Propuesta específica del sistema

#### 2.4 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 opción que desee de los permitidos para su rol. A continuación se muestra el proceso principal descrito anteriormente, los diagramas de los sub-procesos principales se pueden observar en el Anexo 1 del presente documento.



powered by  
BizAgi  
Process Modeler

Figura3. Propuesta específica del sistema

## 2.5 Usuarios relacionados con el sistema

Usuarios relacionados con el sistema	Justificación.
Cliente	Es el usuario que tiene acceso a las funcionalidades de envío de mensajes y gestión de listas de envío de mensajes.
Administrador	Es el usuario que gestiona todas las características del sistema y gestiona a los usuarios clientes.

Tabla 2. Personas relacionadas con el sistema.

## 2.6 Características no funcionales del sistema

Las características no funcionales son propiedades o cualidades que el producto debe tener, debe pensarse en estas como las que hacen al producto atractivo, usable, rápido o confiable.

Estas forman una parte significativa de la especificación. Son importantes para que clientes y usuarios puedan valorar las propiedades del producto, pues si se conoce que el mismo cumple con la toda la funcionalidad requerida, las propiedades no funcionales pueden marcar la diferencia entre un producto bien aceptado y uno con poca aceptación. [19][20]

### 2.6.1 Software

Servidores	Especificaciones
<b>Servidor Aplicaciones Web</b>	<ul style="list-style-type: none"> <li>• Sistema Operativo: Ubuntu Server o cualquiera compatible</li> <li>• Servidor Web: Apache 2.0 o superior</li> <li>• Librerías Adicionales: PHP 5 o superior</li> </ul>
<b>Servidor de Base de Datos</b>	<ul style="list-style-type: none"> <li>• Sistema Operativo: Ubuntu Server o cualquiera compatible</li> <li>• Sistema Gestor de Base de Datos: PostgreSQL 8.3.8 o superior o cualquiera compatible con Gammu 1.27.92 y con Codeigniter 1.7.2</li> </ul>
<b>Sistema GAMMU</b>	<ul style="list-style-type: none"> <li>• Paquete de aplicaciones GAMMU 1.27.92</li> </ul>

Tabla 3. Requerimientos de software (Servidor).

Clientes	Especificaciones
<b>Cliente</b>	<ul style="list-style-type: none"> <li>• Navegador Web: Mozilla Firefox v3.6 o superior o compatible</li> </ul>
<b>Servidor de Base de Datos</b>	<ul style="list-style-type: none"> <li>• Sistema Operativo: Ubuntu Server o cualquiera compatible</li> <li>• Sistema Gestor de Base de Datos: PostgreSQL 8.3.8</li> </ul>

	o superior o cualquiera compatible con Gammu 1.27.92 y con Codeigniter 1.7.2
<b>Sistema GAMMU</b>	• Paquete de aplicaciones GAMMU 1.27.92

Tabla 4. Requerimientos de software (Cliente).

### 2.6.2 Hardware

Servidores	Especificaciones
<b>Servidor Aplicaciones Web</b>	<ul style="list-style-type: none"> <li>• Procesador: 3.00 GHZ</li> <li>• RAM: 1GB</li> <li>• Disco duro: 160 GB</li> <li>• UPS: 1</li> <li>• Tarjeta de Red: 1</li> </ul>
<b>Servidor de Base de Datos</b>	<ul style="list-style-type: none"> <li>• Procesador:3.00 GHZ</li> <li>• RAM: 1GB</li> <li>• Disco duro: 160 GB</li> <li>• UPS: 1</li> <li>• Tarjeta de Red: 1</li> </ul>
<b>Modem GSM</b>	• Cantidad: 1 o más

Tabla 5. Requerimientos de Hardware.

### 2.6.3 Apariencia o interfaz externa

El sistema debe contar con una interfaz amigable, y fácil de usar. El diseño de la interfaz debe posibilitar el acceso rápido a cada una de las funcionalidades del sistema, mediante la utilización de un diseño que muestre en una sola interfaz, el contenido de la sección que el usuario se encuentre utilizando en el momento, evitando las barras de desplazamiento, tanto verticales como horizontales y con el uso de un menú accesible en todo momento para facilitar el acceso a las distintas funcionalidades del sistema.

### 2.6.4 Seguridad

El acceso a cada una de las funcionalidades del sistema estará restringido en dependencia del rol del usuario que acceda al mismo. El uso de un usuario y una contraseña será siempre obligatorio para

acceder a los servicios de la aplicación. Las contraseñas de cada usuario se almacenarán con el uso de un algoritmo de encriptación que imposibilite la lectura clara de dichas contraseñas. El servicio web implementará un sistema de seguridad sencillo pero que agregue cierta confidencialidad, el uso de una llave autogenerada y que se mantenga variando constantemente, para el acceso a cada uno de sus métodos será la base para la seguridad de esta parte del sistema, mientras que el uso de listas de control de acceso asegurarán para la parte web dicha seguridad.

#### **2.6.4.1 Confidencialidad**

La información manejada en el sistema debe de estar protegida de acceso no autorizado y cada usuario con acceso al sistema solo debe acceder a los datos que le correspondan por su rol y cuenta registrada.

En el sistema, el rol de administrador tendrá garantizado permiso a todas las funcionalidades del sistema, mientras que el de cliente, solo contará con permisos sobre el envío de mensajes y la administración de sus listas de contactos

#### **2.6.4.1 Integridad**

Se realizarán validaciones a las entradas de datos de los usuarios del lado del cliente y del servidor, asegurando que no existan incongruencias en dichos datos. El almacenamiento de los mismos o de sus referencias, en el caso de los ficheros, en un sistema gestor de base de datos, asegurarán su persistencia.

#### **2.6.4.1 Disponibilidad**

Mediante la autenticación al sistema, los usuarios tendrán el acceso garantizado a las funcionalidades que les correspondan, siempre que los datos introducidos para dicha autenticación sean los correctos y que la disponibilidad de conexión con el servidor sea la adecuada.

### **2.7 Fase de Exploración**

La fase de exploración es la primera fase de desarrollo de la metodología XP, durante la misma los clientes plantean y describen las historias de usuario que son de gran utilidad para el desarrollo y entrega del producto. De manera simultánea el equipo de trabajo gana conocimiento acerca de las herramientas, tecnologías y prácticas que se utilizarán para el desarrollo del software. La duración de esta fase depende principalmente del grado de experiencia y conocimiento que tengan los miembros del equipo acerca de las tecnologías a utilizar. [21]

### 2.7.1 Historias de Usuarios

Las Historias de Usuarios son las técnicas utilizadas en XP para especificar los requisitos del software. Estas se escriben desde la perspectiva del cliente donde describen brevemente las características que el sistema debe de realizar aunque también los desarrolladores pueden brindar también su ayuda en la identificación de las mismas. El contenido de estas debe ser concreto, sencillo, dinámico y flexible. Cada Historia de Usuario es lo suficientemente comprensible, como para que los programadores puedan estimar con un reducido margen de riesgo, su tiempo de desarrollo.

Durante la fase de exploración se identificaron 25 historias de usuario, de las cuales 10 fueron determinadas como las más importantes y son descritas a continuación y las otras 15 se describen en el Anexo 2 del presente documento.

<b>Historia de Usuario</b>	
<b>Número:</b> 1	<b>Usuario:</b> Usuario del sistema
<b>Nombre historia:</b> Autenticar usuario	
<b>Prioridad en negocio:</b> Alta	<b>Riesgo en desarrollo:</b> Alto
<b>Puntos estimados:</b> 1	<b>Iteración asignada:</b> 1
<b>Programador responsable:</b> José Alexis Hernández Ricardo	
<b>Descripción:</b> El sistema le exigirá a cada usuario que desee acceder al mismo un usuario y una contraseña, luego de verificar si los datos del usuario son correctos concederá acceso a las funcionalidades definidas para el rol del usuario. En caso de datos incorrectos se informará al usuario del error y se permitirá reintentar el acceso.	
<b>Observaciones:</b>	
<ul style="list-style-type: none"> <li>- Los campos de contraseña y usuario serán requeridos.</li> <li>- El dato de la contraseña será siempre mayor de 7 caracteres.</li> </ul>	

Tabla 6. Historia de usuario, Autenticar usuario.

Historia de Usuario	
<b>Número:</b> 2	<b>Usuario:</b> Usuario del sistema
<b>Nombre historia:</b> Enviar mensaje	
<b>Prioridad en negocio:</b> Alta	<b>Riesgo en desarrollo:</b> Alto
<b>Puntos estimados:</b> 2	<b>Iteración asignada:</b> 1
<b>Programador responsable:</b> José Alexis Hernández Ricardo	
<p><b>Descripción:</b> El sistema ofrecerá al usuario una interfaz requiriendo los datos para el envío de un SMS a un número específico, a una lista de contactos de las disponibles para el usuario, previamente almacenada o a una lista rápida de números creada en el momento del envío del mensaje. Luego de los datos ser introducidos, el sistema validará los mismos y en caso de ser correctos agregará el mensaje a la cola de mensajes para enviar de GAMMU SMSD, en caso de existir datos incorrectos, se le informará al usuario del error.</p>	
<p><b>Observaciones:</b></p> <ul style="list-style-type: none"> <li>- Solo el campo de mensaje es requerido, sin embargo si no se especifica un número de destino ni una lista de contactos, ni una lista rápida de números, se mostrará un mensaje de error requiriendo uno de dichos datos.</li> <li>- El texto del mensaje solo permitirá 160 caracteres.</li> </ul>	

Tabla 7. Historia de usuario, Enviar mensaje.

Historia de Usuario	
<b>Número:</b> 3	<b>Usuario:</b> Usuario del sistema
<b>Nombre historia:</b> Adicionar lista de contactos	
<b>Prioridad en negocio:</b> Alta	<b>Riesgo en desarrollo:</b> Alto
<b>Puntos estimados:</b> 3	<b>Iteración asignada:</b> 1

<b>Programador responsable:</b> José Alexis Hernández Ricardo
<b>Descripción:</b> El sistema ofrecerá al usuario una interfaz requiriendo los datos para la creación de una lista de contactos. Luego de los datos ser introducidos, el sistema validará los mismos y en caso de ser correctos se agregará la nueva lista a las listas del usuario, en caso de existir datos incorrectos, se le informará al usuario del error.
<b>Observaciones:</b> <ul style="list-style-type: none"> <li>- Solo el campo de nombre de la lista es requerido.</li> <li>- Los contactos de la lista se agregarán de uno en uno antes de crear la lista.</li> </ul>

Tabla 8. Historia de usuario, Adicionar lista de contactos.

Historia de Usuario	
<b>Número:</b> 4	<b>Usuario:</b> Administrador del sistema
<b>Nombre historia:</b> Adicionar usuario	
<b>Prioridad en negocio:</b> Alta	<b>Riesgo en desarrollo:</b> Alto
<b>Puntos estimados:</b> 2	<b>Iteración asignada:</b> 1
<b>Programador responsable:</b> José Alexis Hernández Ricardo	
<b>Descripción:</b> El sistema ofrecerá al administrador una interfaz requiriendo los datos para la creación de un usuario. Luego de los datos ser introducidos, el sistema validará los mismos y en caso de ser correctos se agregará un nuevo usuario a los usuarios del sistema, en caso de existir datos incorrectos, se le informará al administrador del error.	
<b>Observaciones:</b> <ul style="list-style-type: none"> <li>- Todos los datos del usuario son requeridos.</li> <li>- Solo se podrán definir un rol para el usuario, cliente con un nivel mínimo de acceso y administrador con todos los privilegios del sistema.</li> </ul>	

Tabla 9. Historia de usuario, Adicionar usuario.



Historia de Usuario	
<b>Número:</b> 5	<b>Usuario:</b> Administrador del sistema
<b>Nombre historia:</b> Eliminar usuario	
<b>Prioridad en negocio:</b> Media	<b>Riesgo en desarrollo:</b> Alto
<b>Puntos estimados:</b> 2	<b>Iteración asignada:</b> 1
<b>Programador responsable:</b> José Alexis Hernández Ricardo	
<b>Descripción:</b> El sistema ofrecerá al administrador una interfaz mostrando los datos del usuario a eliminar. El usuario puede seleccionar eliminar o cancelar la acción, si selecciona eliminar, se mostrará un mensaje de confirmación.	
<b>Observaciones:</b>	

Tabla 10. Historia de usuario, Eliminar usuario.

Historia de Usuario	
<b>Número:</b> 6	<b>Usuario:</b> Administrador del sistema
<b>Nombre historia:</b> Adicionar servicio	
<b>Prioridad en negocio:</b> Alta	<b>Riesgo en desarrollo:</b> Alto
<b>Puntos estimados:</b> 4	<b>Iteración asignada:</b> 2
<b>Programador responsable:</b> José Alexis Hernández Ricardo	
<b>Descripción:</b> El sistema ofrecerá al administrador una interfaz requiriendo los datos para la creación del servicio. Luego de los datos ser introducidos, el sistema validará los mismos y en caso de ser correctos se agregará un nuevo servicio a la lista de servicios del sistema, en caso de existir datos incorrectos, se le informará al administrador del error.	

**Observaciones:**

- Solo el nombre del servicio será requerido a la hora de la creación del mismo.
- En dependencia de los datos introducidos para el servicio será la posibilidad de uso del sistema para los clientes.

*Tabla 11. Historia de usuario, Adicionar servicio.*

Historia de Usuario	
<b>Número:</b> 7	<b>Usuario:</b> Administrador del sistema
<b>Nombre historia:</b> Eliminar servicio	
<b>Prioridad en negocio:</b> Media	<b>Riesgo en desarrollo:</b> Alto
<b>Puntos estimados:</b> 2	<b>Iteración asignada:</b> 2
<b>Programador responsable:</b> José Alexis Hernández Ricardo	
<b>Descripción:</b> El sistema ofrecerá al administrador una interfaz mostrando los datos del usuario a eliminar. El usuario puede seleccionar eliminar o cancelar la acción, si selecciona eliminar, se mostrará un mensaje de confirmación.	
<b>Observaciones:</b>	

*Tabla 12. Historia de usuario, Eliminar servicio.*

Historia de Usuario	
<b>Número:</b> 8	<b>Usuario:</b> Administrador del sistema
<b>Nombre historia:</b> Adicionar archivo de configuración de GAMMU SMSD	
<b>Prioridad en negocio:</b> Alta	<b>Riesgo en desarrollo:</b> Alto
<b>Puntos estimados:</b> 2	<b>Iteración asignada:</b> 2

<b>Programador responsable:</b> José Alexis Hernández Ricardo
<b>Descripción:</b> El sistema ofrecerá al administrador una interfaz requiriendo los datos para la creación del archivo. Luego de los datos ser introducidos, el sistema validará los mismos y en caso de ser correctos se agregará una referencia al nuevo archivo a la lista de archivos del sistema y se creará un fichero en el disco duro, en caso de existir datos incorrectos, se le informará al administrador del error.
<b>Observaciones:</b> - La creación del fichero no asegura que realmente funcione el servicio de GAMMU SMSD, si los datos del fichero no son totalmente correctos.

Tabla 13. Historia de usuario, Adicionar archivo de configuración de GAMMU SMSD.

Historia de Usuario	
<b>Número:</b> 9	<b>Usuario:</b> Administrador del sistema
<b>Nombre historia:</b> Iniciar GAMMU SMSD	
<b>Prioridad en negocio:</b> Alta	<b>Riesgo en desarrollo:</b> Alto
<b>Puntos estimados:</b> 1	<b>Iteración asignada:</b> 2
<b>Programador responsable:</b> José Alexis Hernández Ricardo	
<b>Descripción:</b> El sistema ofrecerá al administrador una interfaz mostrando los archivos de configuración que aún no hayan sido utilizados para iniciar un proceso de GAMMU SMSD. El administrador elegirá uno y seleccionará iniciar GAMMU SMSD.	
<b>Observaciones:</b> - La inicialización del proceso no asegura que realmente funcione el servicio de GAMMU SMSD, si los datos del fichero no son totalmente correctos.	

Tabla 14. Historia de usuario, Iniciar GAMMU SMSD.

Historia de Usuario	
<b>Número:</b> 10	<b>Usuario:</b> Administrador del sistema
<b>Nombre historia:</b> Mostrar estadísticas generales de mensajes	
<b>Prioridad en negocio:</b> Medio	<b>Riesgo en desarrollo:</b> Alto
<b>Puntos estimados:</b> 4	<b>Iteración asignada:</b> 2
<b>Programador responsable:</b> José Alexis Hernández Ricardo	
<b>Descripción:</b> El sistema mostrará una serie de estadísticas del comportamiento de los mensajes en la aplicación.	
<b>Observaciones:</b>	

Tabla 15. Historia de usuario, Mostrar estadísticas generales de mensajes.

## 2.8 Fase de Planificación

En esta fase se establece la prioridad de cada historia de usuario, se realiza una estimación del esfuerzo necesario para 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 en la que el equipo de desarrollo labora de forma ininterrumpida.

### 2.8.1 Estimación de esfuerzo por historias de usuarios

Para el desarrollo de la aplicación 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:

Historia de Usuario	Puntos de estimación
Autenticar	1
Enviar mensaje	2

Adicionar lista de contactos	2
Adicionar usuario	2
Eliminar usuario	2
Adicionar servicio	3
Eliminar servicio	2
Adicionar archivo de configuración de GAMMU SMSD	2
Iniciar GAMMU SMSD	1
Mostrar estadísticas generales de mensajes	3
Mostrar mensajes de usuario	1
Mostrar mensajes recibidos	1
Mostrar mensajes en cola para enviar	1
Mostrar mensajes enviados	1
Mostrar listas de contactos	2
Modificar lista de contactos	3
Eliminar lista de contactos	2
Mostrar usuarios	1
Modificar usuario	3
Modificar servicio	3
Mostrar servicios	1
Mostrar archivos de configuración de GAMMU SMSD	1
Eliminar archivo de configuración de GAMMU SMSD	2
Mostrar detalles de archivo de configuración de GAMMU	2
Detener GAMMU SMSD.	2

Tabla 16. Estimación de esfuerzo por historia de usuario.

### **2.8.2 Plan de Iteraciones**

Como parte de las tareas de esta etapa, luego de estimar el esfuerzo dedicado a la realización de cada una de las historias de usuario, se procede a realizar la planificación de las iteraciones previstas para la etapa de desarrollo, las cuales se describen a continuación.

#### **Iteración 1.**

La iteración 1 será la encargada de la implementación de las historias de usuario 1, 2, 3, 4 y 5. Esta iteración servirá para la creación de las funcionalidades fundamentales para iniciar el servicio aunque solo con las posibilidades más básicas, debido a la gran necesidad del cliente de contar con dicho servicio.

#### **Iteración 2.**

La iteración 2 será la encargada de la implementación de las historias de usuario 6, 7, 8, 9 y 10. Al finalizar esta iteración se contará con una versión de la aplicación que no solo permitirá el envío de SMSs a una persona sino a listas de masivos contactos y permitirá observar gráficas del comportamiento de estos envíos masivos.

#### **Iteración 3.**

La iteración 3 será la encargada de la implementación de las historias de usuario 11, 12, 13, 14 y 15. Con la terminación de esta iteración se podrán observar los estados de todos los mensajes de la aplicación así como las distintas listas de envío de cada usuario. Luego de esta iteración estará lista la primera versión de prueba de la aplicación.

#### **Iteración 4.**

La iteración 4 será la encargada de la implementación de las historias de usuario 16, 17, 18, 19 y 20. Con la terminación de esta iteración se concretarán las acciones de gestión de los usuarios y los servicios del sistema.

#### **Iteración 5.**

La iteración 5 será la encargada de la implementación de las historias de usuario 21, 22, 23, 24 y 25. Con la terminación de esta iteración se culminará el proceso de gestión de los servicios de la aplicación y de control del servicio de GAMMU SMSD, se implementarán las medidas de seguridad requeridas que aún se

necesiten y se tendrá una segunda versión del sistema lista para pruebas, esta vez con todas sus funcionalidades.

### 2.8.3 Plan de duración de las iteraciones

Iteración	Orden de la Historias de usuario a implementar.	Duración total
1	Autenticar Enviar mensaje Adicionar lista de contactos Adicionar usuario Eliminar usuario	2 semanas
2	Adicionar servicio Eliminar servicio Adicionar archivo de configuración de GAMMU SMSD Iniciar GAMMU SMSD Mostrar estadísticas generales de mensajes	3 semanas
3	Mostrar mensajes de usuario Mostrar mensajes recibidos Mostrar mensajes en cola para enviar Mostrar mensajes enviados Mostrar listas de contactos	2 semana
4	Modificar lista de contactos Eliminar lista de contactos Mostrar usuarios Modificar usuario Modificar servicio	3 semanas
5	Mostrar servicios Mostrar archivos de configuración de GAMMU SMSD	2 semanas

	Eliminar archivo de configuración de GAMMU SMSD Mostrar detalles de archivo de configuración de GAMMU SMSD Detener GAMMU SMSD.	
--	--	--

Tabla 17. Plan de duración de iteraciones.

### 2.8.4 Plan de entregas

Al finalizar cada iteración se debe presentar el producto desarrollado hasta la versión prevista para la misma, la cual irá incrementando al transcurrir el tiempo, durante las etapas definidas en el plan de iteraciones. A continuación se muestra el plan de entrega de versiones para cada iteración.

Módulo	Final Iteración 1 (15 - marzo - 2011)	Final Iteración 2 (5 - abril - 2011)	Final Iteración 3 (20 - abril - 2011)	Final Iteración 4 (12 - mayo - 2011)	Final Iteración 5 (10 - junio - 2011)
JAMS	Versión 0.1	Versión 0.2	Versión 0.3	Versión 0.4	JAMS 1.0

Tabla 18. Plan de entregas.

### 2.9 Conclusiones

Como parte del presente capítulo se abordó todo lo referente a las fases de exploración y planificación desarrolladas para la implementación de la aplicación, describiendo la duración de cada una de las iteraciones necesarias para la creación del mismo y de los artefactos generados a partir de dichas iteraciones.



## **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.

### **3.2 Arquitectura**

La arquitectura de software consiste en el diseño de componentes de una aplicación, generalmente utilizando patrones de arquitectura. Un diseño arquitectónico describe en general el cómo se construirá una aplicación de software. [23]

### 3.2.1 Arquitectura Utilizada

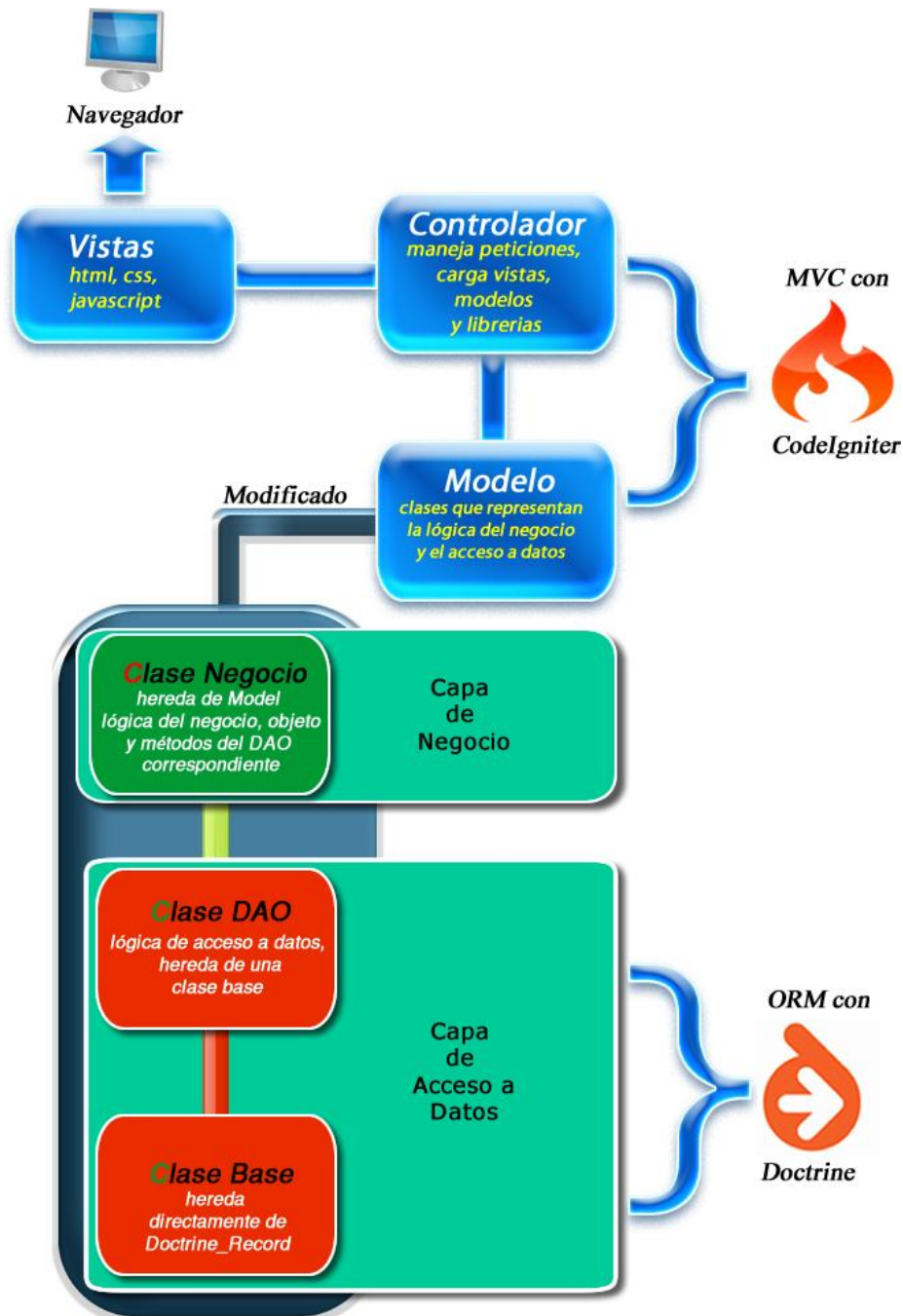


Figura 4. Propuesta específica del sistema.

La arquitectura aplicada para el sistema cuenta con una combinación del patrón MVC (del inglés: Model View-Controller) fusionado con el modelo propuesto de utilización del framework Doctrine por sus propios creadores, estableciendo una arquitectura en capas para toda la aplicación.

La arquitectura en capas está definida como una organización jerárquica tal que cada capa proporciona servicios a la capa inmediatamente superior y se sirve de las prestaciones que le brinda la inmediatamente inferior. [23]

Esta arquitectura tiene como objetivo que el sistema cuente con una escalabilidad y mantenibilidad mejorada sin afectar el rendimiento de la misma, haciendo esfuerzos por agilizar el proceso de implementación del mismo.

### **3.3 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. [23]

Para la realización de la solución se utilizó el patrón MVC implementado por el framework Codeigniter, además de los patrones GRASP y GOF.

#### **3.3.1 Patrones para Asignar Responsabilidades (GRASP)**

- Experto: El uso de este patrón posibilita la asignación de responsabilidad, la clase tiene la información necesaria para la realización de la asignación. [23]

Su uso permitirá que las clases controladoras del sistema manejen las peticiones del cliente, las clases del negocio realicen las validaciones correspondientes de dichas peticiones y las de acceso a datos se encarguen de la gestión de los datos entregados desde la lógica del negocio, esto proporciona que se conserve el encapsulamiento y se soporte el bajo acoplamiento y la alta cohesión.

- Creador: Es el responsable de la creación de una nueva instancia de alguna clase. [23]

Cada clase del negocio debe contener una instancia de su acceso a datos correspondiente y cada clase controladora debe tener una instancia de su negocio, garantizando que se cumpla con la correcta utilización de la arquitectura en capas.

- Alta cohesión: Es el responsable de mantener la complejidad manejable. Este patrón permitirá tener clases fáciles de mantener, entender y reutilizar. [23]

La utilización de comentarios en cada método, el uso de rutinas sencillas de entender y los nombres sugerentes son muestras del uso de este patrón en la solución.

- 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 y la mantenibilidad del mismo. [23]

- Controlador: Es el responsable de gestionar los eventos de entrada al sistema. [23]

En el sistema este patrón se utiliza para controlar las peticiones al sistema, cada clase controladora del framework Codeigniter utiliza este patrón.

### **3.3.2 Patrones GOF**

- 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. [23]

El framework Doctrine 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. [23]

El framework Codeigniter posee una instancia única del controlador frontal disponible mediante este patrón para lograr una vía de entrada única a las solicitudes. Además para el uso de una sola instancia de usuario autenticado se utilizó también este patrón.

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

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. Para la realización del sistema se identificaron 21 tarjetas CRC o de Cargo sin incluir en estas definiciones las encargadas del acceso a datos del sistema, pues estas basan sus responsabilidades en las definidas para el negocio, de ellas mostramos 2 a continuación y el resto se encuentran especificadas en el Anexo 3 del presente documento.

Clase: User	
Responsabilidad	Colaboración
Adicionar usuario Modificar usuario Obtener usuarios Eliminar usuario Mostrar vistas correspondientes	JamsUser

Tabla 19. Tarjeta CRC Clase: User.

Clase: JamsUser	
Responsabilidad	Colaboración
Adicionar usuario Modificar usuario Obtener y devolver usuarios Eliminar usuario Realizar validaciones correspondientes	JamsUserDao

Tabla 20. Tarjeta CRC Clase: JamsUser.

### 3.5 Diseño de la Base de datos

El diseño de la base de datos es una de las tareas más importantes en la construcción de un sistema que utilice base de datos. El modelo de datos permite describir los elementos de la realidad que intervienen en un problema dado y la forma en que se relacionan estos elementos entre sí.

### 3.5.1 Modelo Entidad-Relación

El modelo entidad-relación permite esquematizar la estructura de datos de forma que se pueda entender mejor lo que se está diseñando. El modelo de datos del JAMS muestra las distintas entidades y sus relaciones, agregadas al sistema de almacenamiento de GAMMU para cubrir las funcionalidades del sistema.

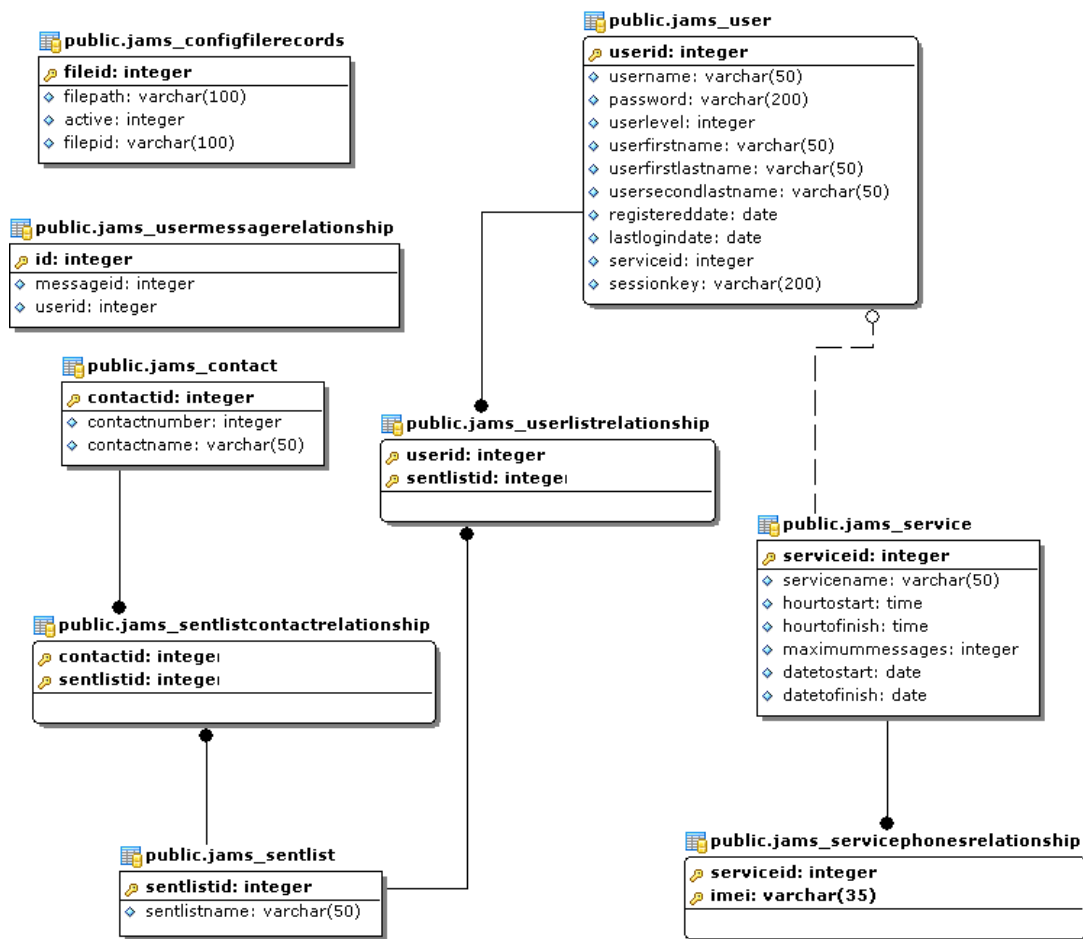


Figura 5. Modelo Entidad Relación.

### 3.6 Conclusiones

En el presente capítulo se determinaron varios aspectos fundamentales para el desarrollo de la solución. Se determinó la arquitectura a utilizar, los patrones, las historias de usuario y el modelo de datos. Cada

uno de estos aspectos se describió, se justificó su utilización y además se detalló la manera en que se usó en la solución.

## CAPÍTULO 4: IMPLEMENTACIÓN Y PRUEBAS

### 4.1 Introducción

El presente capítulo describe las iteraciones realizadas en la etapa de implementación del sistema, las pruebas de aceptación efectuadas sobre el mismo y las tareas de ingeniería generadas en cada iteración.

### 4.2 Fase de Implementación

Durante el transcurso de las iteraciones se realiza la implementación de las CRC que responden a cada una 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. [27]

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 abordan algunas de las historias de usuario de mayor prioridad para el cliente y se construye la base de la arquitectura del sistema con el fin de obtener un producto para ser mostrado al cliente y obtener sugerencias de este. En la siguiente tabla se muestran las historias de usuario que se desean realizar en la iteración, su tiempo de realización estimado y el real.

Historia de Usuario	Estimación	Real
Autenticar	1	0.5
Enviar mensaje	2	1
Adicionar lista de contactos	2	2
Adicionar usuario	2	1
Eliminar usuario	2	1

*Tabla 21. Historias de Usuario desarrolladas en la primera iteración.*

#### Tareas de las historias de usuario desarrolladas en la primera iteración



Las tareas de las historias de usuarios a desarrollar en la primera iteración se encuentran desde el Anexo 4.1 al 4.5.

#### 4.2.2 Segunda Iteración

En esta iteración se desarrollan las tareas que complementaron las desarrolladas en la iteración anterior y se comienza con las primeras funcionalidades que interactúan directamente con el GAMMU SMSD. En la siguiente tabla se muestran las historias de usuario que se desean realizar en la iteración, su tiempo de realización estimado y el real.

Historia de Usuario	Estimación	Real
Adicionar servicio	3	2
Eliminar servicio	2	1
Adicionar archivo de configuración de GAMMU SMSD	2	2
Iniciar GAMMU SMSD	1	1
Mostrar estadísticas generales de mensajes	3	2

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

#### Tareas de las historias de usuario desarrolladas en la segunda iteración

Las tareas de las historias de usuarios a desarrollar en la segunda iteración se encuentran desde el Anexo 4.6 al 4.10

#### 4.2.3 Tercera Iteración

En esta tercera iteración se asumen las tareas de muestras de mensajes generales de todo el sistema. Además se implementa la funcionalidad de muestra de las listas de contactos de cada usuario permitiendo así el envío masivo de SMS.

Historia de Usuario	Estimación	Real
Mostrar mensajes de usuario	1	0.5
Mostrar mensajes recibidos	1	0.5
Mostrar mensajes en cola para enviar	1	0.5
Mostrar mensajes enviados	1	0.5
Mostrar listas de contactos	2	1

Tabla 23. Historias de Usuario desarrolladas en la tercera iteración.

#### 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 desde el Anexo 4.11 al 4.15.

#### 4.2.4 Cuarta Iteración

En esta iteración se realizan las funcionalidades de modificación de datos que complementan las funcionalidades ya creadas previamente.

Historia de Usuario	Estimación	Real
Modificar lista de contactos	3	2
Eliminar lista de contactos	2	1
Mostrar usuarios	1	1
Modificar usuario	3	2
Modificar servicio	3	2

Tabla 24. Historias de Usuario desarrolladas en la cuarta iteración.

#### 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 desde el Anexo 4.16 al 4.20.

#### 4.2.5 Quinta Iteración

Iteración final que entrega la primera versión con todas las funcionalidades implementadas. Se completan características de la aplicación que ya se habían comenzado a construir en las iteraciones anteriores y se hacen ajustes de seguridad del sistema. En la siguiente tabla se muestran las historias de usuario que se desean realizar en la iteración, su tiempo de realización estimado y el real.

Historia de Usuario	Estimación	Real
Mostrar servicios	1	1
Mostrar archivos de configuración de GAMMU SMSD	1	1
Eliminar archivo de configuración de GAMMU SMSD	2	1
Mostrar detalles de archivo de configuración de GAMMU SMSD	2	1
Detener GAMMU SMSD.	2	2

*Tabla 25. Historias de Usuario desarrolladas en la quinta iteración.*

#### 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 desde el Anexo 4.21 al 4.25.

#### 4.3 Fase Prueba

El objetivo de las pruebas de software es detectar el mayor número posible de errores. Para lograrlo existen diferentes métodos de prueba. Pese a que se ha realizado una división entre pruebas de caja blanca y caja negra, no deben considerarse métodos 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. [28]

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 [27]

#### **4.3.1 Prueba Unitaria**

Todos los módulos del sistema 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. Para que funcione la propiedad colectiva del código, dicho código debe arrojar resultados correctos de sus correspondientes pruebas unitarias. 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. [29]

Estas pruebas no generan artefactos y no son directamente palpables para el cliente.

#### **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” (en inglés: 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. [29]

Para la aplicación JAMS el cliente determinó la realización de pruebas a las 3 funcionalidades descritas en las historias de usuario que el mismo cliente consideró más importantes. Como resultados de dichas

pruebas se generaron artefactos descritos en las tablas situadas en el Anexo 4, las cuales constan 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 esperado por el cliente.
- Insatisfactoria: Cuando el resultado de la prueba genera un error de codificación en la aplicación, trayendo consigo la invalidación de la historia de usuario.

Los resultados de dichas pruebas fueron satisfactorios, resultados con los cual el cliente quedó complacido.

#### **4.4 Conclusiones**

En el presente capítulo se implementaron cada una de las funcionalidades descritas en las tarjetas CRC respondiendo así a las historias de usuario previamente definidas. Igualmente se realizaron pruebas a la aplicación y se ejemplificaron los resultados de dichas pruebas.

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

### **5.1 Introducción**

En el presente capítulo se realiza un estudio de la factibilidad del desarrollo del sistema propuesto. Se hace una estimación de los recursos, esfuerzo y otros elementos necesarios para su realización, ya que a partir de un detallado análisis se puede determinar con cierto grado de certeza los recursos necesarios para la implementación y correcta elaboración del producto a entregar.

### **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. [30]

La factibilidad de este sistema se determinó 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 tiene poco conocimiento acerca de la magnitud 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 a usar.

### **5.3 Características del Proyecto**

#### **5.3.1 Entradas Externas**

Las entradas externas consisten en la 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. La aplicación desarrollada cuenta con 9 entradas externas que se muestran a continuación:

Nombre de la entrada externa.	Cantidad de ficheros.	Cantidad de elementos de datos.	Clasificación (Simple, Media, Compleja)
Autenticar	1	2	Simple
Enviar Mensaje	2	3	Simple
Adicionar Lista de Contactos	3	3	Simple
Adicionar Usuario	1	7	Simple
Adicionar Servicio	4	8	Compleja
Adicionar Archivo de Configuración	1	17	Media
Modificar Lista de Contactos	3	3	Simple
Modificar Usuario	1	7	Simple
Modificar Servicio	4	8	Compleja
Eliminar Lista de Contactos	3	1	Simple
Eliminar Usuario	1	1	Simple
Eliminar Servicio	4	1	Media
Eliminar Archivo de Configuración	1	1	Simple

Tabla 26. Entradas Externas.

### 5.3.2 Salidas Externas

Salida de datos de usuario o de control que deja el límite del sistema de software. La aplicación desarrollada cuenta con 2 salidas externas que se muestran a continuación:

Nombre de la salida externa.	Cantidad de ficheros.	Cantidad de elementos de datos.	Clasificación (Simple, Media, Compleja)
Estadística total de mensajes por modem	2	2	Simple
Estadística total de mensajes en los últimos 15 días	2	2	Simple
Estadística total de mensajes en las últimas 24 horas	2	2	Simple



Estadística de mensajes por día en los últimos 15 días	2	2	Simple
Estadística de resumen de mensajes	3	3	Simple
Archivo de Configuración	1	17	Compleja
Mostrar Lista de Contactos	1	1	Simple
Mostrar Usuarios	1	0	Simple
Mostrar Servicios	2	0	Simple
Mostrar Detalles Archivo de Configuración	1	1	Simple

Tabla 27. Salidas Externas.

### 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). La aplicación desarrollada no presenta consultas externas.

Nombre de la Petición.	Cantidad de ficheros.	Cantidad de elementos de datos.	Clasificación (Simple, Media, Compleja)
Datos de MODEMs	1	1	Simple
Datos de usuarios	1	1	Simple
Datos de usuarios asociados a servicio	2	1	Simple
Datos de MODEMs asociados a servicio	2	1	Simple
Datos de listas asociadas a usuario	2	1	Simple

Tabla 28. Consultas Externas.

### 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.

Nombre de la petición.	Cantidad de ficheros.	Cantidad de elementos de datos.	Clasificación (Simple, Media, Compleja)
Contactos	1	2	Simple
Referencia de archivos de configuración	1	3	Simple
Relación mensaje usuario	1	3	Simple
Relación lista contacto	1	2	Simple
Lista de contactos	1	2	Simple
Relación lista usuario	1	2	Simple
Usuario	1	11	Simple
Servicio	1	7	Simple
Relación lista servicio	1	2	Simple

Tabla 29. Fichero lógico interno.

### 5.3.5 Archivos de interfaz externos

Archivos transferidos o compartidos entre sistemas de software. La aplicación desarrollada no cuenta con archivos de interfaz externos.

### 5.3.6 Puntos de función desajustados

A continuación se presenta la tabla basada en las características del sistema anteriormente descritas. 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.

Elementos.	Simple		Medio		Complejo		Subtotal.
	No.	Peso	No.	Peso	No.	Peso	
Entradas externas.	9	3	2	4	2	6	47
Salidas externas.	9	4	0	5	1	7	43
Consultas externas.	5	3	0	4	0	6	15
Fichero lógico interno.	9	7	0	10	0	15	63
Fichero interfaz interno.	0	5	0	7	0	10	0
<b>Total (UFP):</b>							<b>168</b>

Tabla 30. Puntos de función desajustados.

#### 5.4 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 siguiente ecuación:

$$\text{SLOC} = 168 * 59$$

$$\text{SLOC} = 9912$$

Donde:

**SLOC** (Cantidad de instrucciones fuente)

**UFP** (Puntos de función desajustados): 168.

**ratio** (Conversión de puntos de función desajustados a líneas de código): Para el lenguaje PHP es 59.

Partiendo de la ecuación anterior se obtiene el siguiente resultado:

**KSLOC** : Tamaño del software a desarrollar en miles de instrucciones fuente.

$$\text{KSLOC} = 9,912$$

Las características del sistema y valores obtenidos se muestran en la siguiente tabla:

Características	Valor
Puntos de función desajustados.	168
Lenguaje (PHP).	59
Instrucciones fuentes por puntos de función (SLOC).	9912
Miles de instrucciones fuentes (KSLOC)	9,912

Tabla 31. Características del sistema.

### 5.5 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.

Donde:

Esfuerzo nominal requerido en meses / hombres.

Tamaño estimado del software en miles de instrucciones fuente.

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

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

$$B = 0.91 + 0.01 \times 6,32$$

$$B = 0.97$$

Donde:

: Variables escalares que indican las características que el proyecto presenta respecto a su complejidad y entorno de desarrollo, las cuales se analizan a continuación:

**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, complejidad 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 aspectos 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 teniendo en cuenta el anexo A.6.2. y las características del proyecto.

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

Tabla 32. Factores de Escala del Sistema.

## 5.6 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 (EM<sub>i</sub>) que representan las características del proyecto y expresan su impacto en el desarrollo total del producto de software.

### 5.6.1 Ajuste del esfuerzo nominal clasificado

Los 7 multiplicadores de esfuerzo utilizados en el modelo de Diseño Temprano divididos por categorías 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 en cuanto a lenguaje, herramientas, plataforma, etc.

- Del Proyecto.

FCIL: Facilidades.

SCED: Cronograma de Desarrollo Requerido.

La siguiente tabla muestra los valores asignados a cada una de los multiplicadores de esfuerzo teniendo en cuenta el anexo A.6.3. y las características del proyecto.

Nombre	Valor	Justificación
RCPX	1,30	La complejidad que presenta la aplicación es alta.
RUSE	1,00	La aplicación reutiliza código.
PDIF	1,00	Uso de memoria y almacenamiento normal, plataforma estable.
PREX	0,71	La experiencia del equipo con la herramienta y lenguaje es alta.
PERS	1,00	La capacidad del equipo de desarrollo es alto.
FCIL	0,73	La aplicación es fácil de utilizar.
SCED	1,00	El desarrollo del sistema se realizó en el tiempo planificado.
<b>Total(EM)</b>	0,67	

Tabla 33. Multiplicadores de esfuerzo del sistema.

—  
—

### 5.6.2 Cálculo del tiempo de desarrollo del software

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

\_\_\_\_\_

Donde:

**C:** Constante con valor 3.64.

\_\_\_\_\_

Donde:

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

### 5.6.3 Cálculo del costo total del proyecto.

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

\_\_\_\_\_

**C = 1822**

Donde:

**C:** Costo total.

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



Donde:

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

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

\_\_\_\_\_

\_\_\_\_\_

Pero como para la solución del sistema solo existe una persona, entonces ajustamos el tiempo de desarrollo quedando finalmente como valor:

**TDEV = 18.22 meses**

#### 5.6.4 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.

Cálculo de:	Valor
Esfuerzo.	18,22 meses/hombre
Tiempo de desarrollo.	19 meses
Cantidad de hombres.	1 hombre
Salario medio.	100 pesos
Costo.	1822 pesos

Tabla 34. Resultados.

### **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.**

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 JAMS es viable.

## **CONCLUSIONES**

Con el presente trabajo de diploma se presentó el sistema JAMS, el cual consiste en una aplicación encargada de enviar SMSs masivos y de interactuar con el sistema GAMMU para la gestión de dichos mensajes.

Para la construcción y desarrollo del sistema se utilizaron las herramientas y tecnologías seleccionadas como son: lenguaje de programación PHP, los frameworks Codeigniter, Doctrine y Zend, para el acceso a datos el gestor de base de datos PostgreSQL, la herramienta de modelado utilizada fue el BizAgi Process Modeler y además se desarrolló el sistema en base a la metodología de desarrollo seleccionada que permitió crear un software de numerables funcionalidades en un corto período de tiempo.

Objetivamente se logró que:

- Se modificara el sistema GAMMU y GAMMU SMSD para cumplir con las necesidades del sistema JAMS.
- Se creara un sistema que integra una interfaz web y un servicio web para el acceso a sus funcionalidades
- Se estableciera un nivel de seguridad adecuado para proteger la información del sistema
- Se cumpliera con los objetivos propuestos en la presente investigación.

## **RECOMENDACIONES**

Aunque el sistema JAMS resuelve la problemática planteada en la presente investigación, existen varias recomendaciones que garantizarían una mejora en su funcionamiento. Estas recomendaciones son:

- Mejora de la seguridad del sistema, implementando un mecanismo más riguroso de control de acceso de para los usuarios y para el servicio web.
- Creación de un mejorado sistema de roles, permisos y grupos gestionables y almacenados en el sistema gestor de base de datos utilizado, para garantizar mejor escalabilidad del sistema.
- Mejorar el mecanismo de alertas y errores para brindar una mejor información a la hora de la depuración del sistema.
- Aumentar las funcionalidades del servicio web para permitir a los clientes hacer uso de las mismas funcionalidades disponibles a través de la interfaz web.
- Crear un mecanismo para evitar el mal uso del servicio web como peticiones infinitas o envíos excesivos de datos que afecten la estabilidad del servicio web y del sistema en general.

Cada una de estas recomendaciones se centra en mejorar el funcionamiento del sistema haciéndolo más flexible a cambios, reutilizable y seguro.

## REFERENCIAS BIBLIOGRÁFICAS

1. Dell'Ordine, José Luis. Historia de la comunicación. [En línea: 10 de Noviembre de 2008][Citado el: 4 de abril del 2011].  
Disponible en: <http://www.rppnet.com.ar/hiscomunicacion.htm>
2. Farley, Tom. The Cell-Phone Revolution [En línea: 1 de Octubre de 2007][Citado el: 5 de abril del 2011].  
Disponible en: <http://www.americanheritage.com/events/articles/web/20070110-cell-phone-att-mobile-phone-motorola-federal-communications-commission-cdma-tdma-gsm.shtml>
3. SMSCenter[En línea] [Citado el: 5 de abril del 2011]  
Disponible en: <http://www.smscenter.com.uy>
4. Developershome.com. What is an SMS Gateway? [En línea: 6 de abril del 2011] [Citado el: 6 de abril del 2011] Disponible en:  
[http://www.developershome.com/sms/sms\\_tutorial.asp?page=smsGateway](http://www.developershome.com/sms/sms_tutorial.asp?page=smsGateway).
5. GAMMU [En línea] [Citado el: 6 de abril del 2011]  
Disponible en: <http://wammu.eu/gammu/>
6. SMS masivos - Plataforma web para envío y recepción de mensajes de texto personalizados [En línea: 6 de abril del 2011][Citado el: 9 de abril del 2011] Disponible en:  
<http://www.smsmasivos.com.ar/>.
7. Madrid Gorelov, Vadim Paz y De Paz Santana, Juan Francisco. *Servicios Web*. Salamanca: Universidad de Salamanca. Doctorado Informática y Automática.
8. W3C. Extensible Markup Language (XML). [En línea: 14 de marzo de 2010][Citado el: 9 de abril de 2011]  
Disponible en: <http://www.w3.org/XML>
8. Alarcos. Ingeniería de Software 1. Tema04.pdf [En línea] [Citado el: 17 de abril del 2011]  
Disponible en: <http://alarcos.inf-cr.uclm.es/doc/ISOFTWAREI/Tema04.pdf>.
9. Jorge H. Canós, Patricio Letelier, María del Carmen Panadés. Metodologías ágiles de desarrollo de software. Universidad Politécnica de Valencia. 2004.

10. Coad P., Lefebvre E., De Lucas. Metodologías de desarrollo (FDD). Java Modeling In Color With UML: Enterprise Components and Process. Prentice Hall, 1999.
11. Linuxcentro.net [En línea] [Citado el: 20 de Abril del 2011]  
<http://www.linuxcentro.net/linux/staticpages/index.php?page=CaracteristicasPHP>
12. Clickatell.com [En línea: 6 de abril del 2011][Citado el: 9 de abril del 2011] Disponible en:  
<http://www.clickatell.com/>
13. GAMMU [En línea: 25 de febrero del 2011][Citado el: 9 de abril del 2011] Disponible en:  
<http://wammu.eu/download/gammu/>
14. Kalkun.sourceforge.net [En línea][Citado el: 20 de abril del 2011] Disponible en:  
<http://kalkun.sourceforge.net/about.php>
15. Doctrine-project.org [En línea][Citado el: 22 de abril del 2011] Disponible en: <http://www.doctrine-project.org>
16. Codeigniter.com [En línea][Citado el: 22 de abril del 2011] Disponible en: <http://codeigniter.com/>
17. Apache.org [En línea][Citado el: 27 de abril del 2011] Disponible en:  
[http://httpd.apache.org/ABOUT\\_APACHE.html](http://httpd.apache.org/ABOUT_APACHE.html)
18. Postgresql.org [En línea][Citado el: 28 de abril del 2011] Disponible en:  
[http://www.postgresql.org.es/sobre\\_postgresql/](http://www.postgresql.org.es/sobre_postgresql/)
19. PRESSMAN, Roger S. "Ingeniería de Software. Un enfoque Práctico" 6ta edición.2007
20. SOMMERVILLE, I. "Ingeniería del Software" 7ma Edición. 2005.
21. Extremeprogramming.org [En línea][Citado el: 28 de abril del 2011] Disponible en:  
<http://www.extremeprogramming.org/>
22. BPMN.org [En línea][Citado el: 29 de abril del 2011] Disponible en: <http://www.bpmn.org>
23. Pimentel González, Luis Alberto, Pérez Rivero, Iósev y Haro Pérez, Madelín. ArBaWeb: ARQUITECTURA BASE SOBRE LA WEB. 2007
24. nowsms.com [En línea][Citado el: 30 de abril del 2011] Disponible en: <http://www.nowsms.com>
25. Leopoldo, Carlos. Zend Framework, una introducción. [En línea] [Citado el: 27 de Noviembre de 2007] Disponible en: <http://techtastico.com/post/zend-framework-una-introduccion/>.
26. 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/>.

27. 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.
28. Tuya, Javier, Ramos Román, Isabel y Dolado Cosín, Javier. Técnicas cuantitativas para la gestión en la ingeniería del software. 2007.
29. Joskowicz, Ing. José. Reglas y Prácticas en eXtreme Programming. [En línea] [Citado el: 10 de Febrero de 2008]. Disponible en: <http://ie.fing.edu.uy/~josej/docs/XP%20-%20Jose%20Joskowicz.pdf>
30. Moreno, Ana M<sup>a</sup>. Estimación de Proyectos Software: COCOMO II. [En línea] [http://www.eici.ucm.cl/Academicos/ygomez/descargas/Ing\\_Sw2/apuntes/cocomo\\_manual\\_espanol.pdf](http://www.eici.ucm.cl/Academicos/ygomez/descargas/Ing_Sw2/apuntes/cocomo_manual_espanol.pdf).

## **GLOSARIO DE TÉRMINOS**

**GSM:** Del inglés Global System for Mobile Communications. Sistema Global para las Comunicaciones Móviles. Sistema estándar definido para la comunicación mediante teléfonos móviles que incorporan tecnología celular.

**IDE:** Del inglés Integrated Development Environment. Entorno de Desarrollo Integrado. Es un entorno de programación que ha sido empaquetado como un programa de aplicación, es decir, consiste en un editor de código, un compilador, un depurador y un constructor de gráficas GUI. Los IDEs pueden ser aplicaciones por sí solas o pueden ser parte de otras aplicaciones existentes.

**XML:** Del inglés Extensible Markup Language. Lenguaje de Marcas Extensibles. Constituye un metalenguaje extensible de etiquetas desarrollado por el World Wide Web Consortium (W3C) que permite definir la gramática de lenguajes específicos.

**LAMP:** Conjunto de herramientas de código abierto que trabajando juntas nos permiten disponer de un servidor web. Las herramientas que aglutina el término LAMP son Linux, Apache, MySQL y PHP, Perl, o Python , con lo cual conseguimos todo lo que debe tener una plataforma web, que es un sistema operativo, un servidor web, una base de datos, y un lenguaje de programación.