

Universidad de las Ciencias Informáticas

Facultad 2



**Título: Versión 2.0 de la Plataforma de Mensajería SMS
utilizando módems GSM.**

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

Autores: Lisandra Soler Ponce

Eduardo Noel Rosales García

Tutor: Ing. Damián Ilizastegui Arriba

Co-Tutor: Ing. Lex Karel Zayas Hernández

La Habana, Junio de 2012

“Año 54 de la Revolución”

Declaración de autoría

Declaramos que somos los únicos autores de este trabajo y autorizamos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo. Para que así conste firmamos la presente a los __ días del mes de __ del año ____.

Lisandra Soler Ponce

Firma de la autora

Eduardo Noel Rosales García

Firma del autor

Ing. Damián Ilizastegui Arriba

Firma del tutor

Ing. Lex Karel Zayas Hernández

Firma del tutor



“La Revolución hay que hacerla a ritmo violento, el que se cansa tiene derecho a cansarse, pero no tiene derecho a ser un hombre de vanguardia”.

A stylized, handwritten signature in black ink, located in the bottom right corner of the page. The signature is cursive and appears to be the name 'Che'.

RESUMEN

El acelerado desarrollo que se evidencia en la actualidad, ha impulsado entre otros aspectos la comunicación a distancia, específicamente la telefonía celular, al punto de incluir funciones extras en los dispositivos móviles, como es el caso de la reproducción de música y videos, los juegos y SMS, entre otros. En Cuba se evidencia el esfuerzo y la dedicación para promover y ampliar el uso de la telefonía móvil. En el centro de Telemática de la Universidad de las Ciencias Informáticas se desarrolló la propuesta de solución informática “Sistema de servicio de SMS basado en Gammu (JAMS)”, con el objetivo de lograr el envío masivo de mensajes SMS, utilizando módems GSM.

Esta investigación tiene como meta implementar las funcionalidades necesarias para el desarrollo de la versión 2.0 del Sistema de servicio de SMS basado en Gammu, con el objetivo de mejorar el proceso de administración del mismo.

PALABRAS CLAVES

Comunicación, desarrollo, Gammu, GSM, sistema, SMS, telefonía celular.

ÍNDICE

INTRODUCCIÓN	11
CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA	16
1.1. <i>Introducción</i>	16
1.2. <i>Conceptos fundamentales relacionados con el dominio del problema</i>	16
1.2.1. <i>Módem GSM</i>	16
1.2.2. <i>GAMMU y GAMMU SMSD</i>	16
1.3. <i>Sistemas que permiten la ejecución de comandos de forma remota en el servidor</i>	17
1.3.1. <i>Sistema WDS para la Administración remota de servidores</i>	17
1.4. <i>Aplicaciones de envío masivo de SMS que funcionan como SMS GATEWAY</i>	18
1.4.1. <i>Activa SMS</i>	18
1.4.2. <i>Altiria</i>	19
1.4.3. <i>Blue Eye</i>	19
1.5. <i>Servicios web</i>	20
1.6. <i>Metodologías de desarrollo de software</i>	21
1.6.1. <i>Extreme Programming</i>	22
1.7. <i>Lenguaje de modelado UML (Unified Modeling Language o Lenguaje Unificado de Modelado)</i>	23
1.8. <i>Lenguajes de programación</i>	24
1.9. <i>Framework Doctrine</i>	25
1.10. <i>Framework CodeIgniter v1.7.2</i>	25
1.11. <i>Zend Framework v1.7.8</i>	26
1.12. <i>Entorno de desarrollo: Netbeans 7.0 para PHP</i>	26
1.13. <i>Herramienta de Modelado: Visual Paradigm 6.4</i>	27
1.14. <i>Servidor Web: Apache v2.2.12</i>	27

1.15.	<i>Sistema Gestor de Bases de Datos: PostgreSQL 8.4.7</i>	28
1.16.	<i>Conclusiones</i>	28
CAPÍTULO 2. CARACTERÍSTICAS DEL SISTEMA, EXPLORACIÓN Y PLANIFICACIÓN		30
2.1.	<i>Introducción</i>	30
2.2.	<i>Problemática</i>	30
2.3.	<i>Objeto de automatización</i>	31
2.4.	<i>Propuesta del sistema</i>	31
2.5.	<i>Características no funcionales del sistema</i>	33
2.5.1.	<i>Usabilidad</i>	33
2.5.2.	<i>Disponibilidad</i>	33
2.5.3.	<i>Apariencia o Interfaz Externa</i>	33
2.5.4.	<i>Seguridad</i>	33
2.5.5.	<i>Confidencialidad</i>	34
2.5.6.	<i>Requisitos de Hardware</i>	34
2.5.7.	<i>Requisitos de Software</i>	35
2.6.	<i>Usuarios relacionados con el sistema</i>	36
2.7.	<i>Fase de Exploración</i>	36
2.7.1.	<i>Historias de Usuarios</i>	36
2.8.	<i>Planificación</i>	40
2.8.1.	<i>Estimación de esfuerzo por Historia de Usuario</i>	41
2.8.2.	<i>Plan de iteraciones</i>	41
2.8.3.	<i>Plan de duración de las iteraciones</i>	42
2.8.4.	<i>Plan de entregas</i>	42
2.9.	<i>Conclusiones</i>	43
CAPÍTULO 3. ANÁLISIS Y DISEÑO DEL SISTEMA		44

3.1.	<i>Introducción</i>	44
3.2.	<i>Arquitectura</i>	44
3.2.1.	Estilo de arquitectura: Modelo Vista Controlador	44
3.3.	<i>Patrones de Diseño</i>	46
3.3.1.	Patrones para Asignar Responsabilidades (GRASP)	46
3.3.2.	Patrones GOF	46
3.4.	<i>Tarjetas Clase - Responsabilidad - Colaborador (CRC)</i>	47
3.5.	<i>Diseño de la Base de Datos</i>	48
3.5.1.	Modelo Entidad - Relación	48
3.6.	<i>Conclusiones</i>	49
CAPÍTULO 4.	IMPLEMENTACIÓN Y PRUEBA	50
4.1.	<i>Introducción</i>	50
4.2.	<i>Fase de Implementación</i>	50
4.2.1.	Iteraciones	50
4.3.	<i>Diagrama de Despliegue</i>	55
4.4.	<i>Pruebas</i>	55
4.4.1.	Pruebas Unitarias	56
4.4.2.	Pruebas de Aceptación	56
4.5.	<i>Conclusiones</i>	67
CONCLUSIONES GENERALES	68
RECOMENDACIONES	69
REFERENCIAS BIBLIOGRÁFICAS	70
BIBLIOGRAFÍA	72
ANEXOS	73
GLOSARIO DE TÉRMINOS	76

ÍNDICE DE TABLAS

Tabla 2.1 Requerimientos de Hardware	35
Tabla 2.2 Requerimientos de Software (Servidor)	35
Tabla 2.3 Requerimientos de Software (Cliente)	35
Tabla 2.4 Usuarios relacionados con el sistema	36
Tabla 2.5 HU: Enviar mensaje de respuesta hacia entidad externa	39
Tabla 2.6 HU: Enviar notificación de posible error.....	40
Tabla 2.7 HU: Reiniciar aplicación Gammu.....	40
Tabla 2.8 Estimación de esfuerzo por Historias de Usuarios.....	41
Tabla 2.9 Plan de duración de iteraciones	42
Tabla 2.10 Plan de entrega de las iteraciones	43
Tabla 3.1 Clase Gateway	48
Tabla 3.2 Clase Notificación.....	48
Tabla 4.1 Historias abordadas en la primera iteración.....	51
Tabla 4.2 Tarea #1 de la HU: Enviar mensaje de respuesta hacia entidad externa.....	51
Tabla 4.3 Tarea #2 de la HU: Enviar notificación de posible error	52
Tabla 4.4 Tarea #3 de la HU: Reiniciar aplicación Gammu	53
Tabla 4.5 Historias abordadas en la segunda iteración	53
Tabla 4.6 Tarea #6 de la HU: Gestionar saldo de alerta en los Módems.....	54
Tabla 4.7 Tarea #8 de la HU: Gestionar precio del mensaje	54
Tabla 4.8 HU1-P1: Enviar mensaje de respuesta hacia entidad externa.....	57
Tabla 4.9 HU1-P2: Envío fallido de mensaje de respuesta hacia entidad externa.....	58
Tabla 4.10 HU2-P1: Enviar notificación de posible error en el servidor	59
Tabla 4.11 HU2-P2: Envío fallido de notificación de posible error en el servidor	60
Tabla 4.12 HU3-P1: Reiniciar aplicación Gammu	61

Tabla 4.13 HU3-P2: Reinicio fallido de la aplicación Gammu.....	62
Tabla 4.14 HU4-P1: Mostrar estadísticas generales de los mensajes.....	62
Tabla 4.15 HU5-P1: Exportar reportes a formato PDF	63
Tabla 4.16 HU6-P1: Gestionar Saldo de Alerta en los módems	63
Tabla 4.17 HU6-P2: Modificación fallida del saldo de Alerta en los módems	64
Tabla 4.18 HU7-P1: Gestionar Saldo de módem	65
Tabla 4.19 HU7-P2: Modificación fallida del saldo del módem	65
Tabla 4.20 HU8-P1: Gestionar precio del mensaje	66
Tabla 4.21 HU8-P2: Modificación fallida del precio del mensaje	66
Tabla A1- HU4: Mostrar estadísticas generales de los mensajes.....	73
Tabla A6 - HU9: Enviar Notificación de Recarga de módem	74
Tabla A7: Clase PhoneC.....	74
Tabla A10: Clase Statistics	75
Tabla A11: Clase WebService	75
Tabla A12 - Tarea #4 de la HU: Mostrar estadísticas generales de los mensajes	76
Tabla A14 - Tarea #7 de la HU: Gestionar saldo de módem	76

ÍNDICE DE FIGURAS

Figura 2.1 Propuesta General del Sistema.....	32
Figura 2.2 Propuesta Específica del Sistema.....	32
Figura 3.1 Propuesta específica del sistema.....	45
Figura 3.2 Modelo Entidad - Relación	49
Figura 4.1 Diagrama de despliegue	55
Figura 4.2 Estadísticas de las pruebas	67

INTRODUCCIÓN

Desde tiempos remotos el hombre se ha enfrentado a la necesidad de vencer las distancias y para resolverlo comenzó a utilizar sus propios medios de comunicación. Cuando en su camino encontró obstáculos como montañas, ríos, lagos, mares y océanos, puso en juego todas sus energías para dominar estos hechos del medio geográfico y logró vencerlos a base de grandes sacrificios, surgidos siempre por una gran necesidad.

Con la evolución del hombre y el refinamiento de sus instrumentos de trabajo, se establecen nuevas técnicas y tecnologías que influyen en el proceso de emitir las ideas, dando lugar en la actualidad, a los modernos sistemas de comunicación, pudiendo mencionar entre estos: el telégrafo, el correo, la radio, la televisión, el cable submarino y el teléfono.

La creación del teléfono en el año 1876 revolucionó el mundo de las comunicaciones, convirtiéndose rápidamente en uno de los medios a distancia más populares y usados en todo el mundo. Casi un siglo después, en el año 1973, el Dr. Martin Cooper, gerente general de sistemas de Motorola, crea el primer teléfono celular, surgiendo así una nueva era: La era de las comunicaciones inalámbricas y la telefonía celular. (Hernández Ricardo, 2011)

Los teléfonos celulares se han convertido en una herramienta de uso necesario para las personas a nivel mundial, haciéndolas sentir más seguras y productivas. Su principal función es la comunicación de voz, como el teléfono convencional, pero su rápido desarrollo ha incorporado otras funciones como juegos, reproducción de música y videos, correo electrónico, fotografía, navegación por Internet, SMS¹ entre otras. (Fernández Jo, y otros, 2011)

En cuanto a la producción de estos dispositivos móviles, la prestación de servicios y la creación de software de avanzada para el usuario, se encuentran encabezando la lista los países más desarrollados, dominando el mercado con sus nuevas tecnologías que avanzan rápidamente e imponen estándares que sirven de guía al resto de las naciones.

Cuba a pesar de ser un país subdesarrollado, ha comenzado a dar sus primeros pasos en lo que respecta a la telefonía celular, incorporando nuevas soluciones informáticas en esta

¹ **SMS:** Short Message Service o Servicio de Mensajes Cortos.

rama. En el centro de Telemática de la Universidad de las Ciencias Informáticas (UCI) se desarrollan varios productos relacionados con la telefonía celular, entre ellos se encuentra la Plataforma de servicio de SMS basado en Gammu, creada con el propósito de lograr el envío masivo de mensajes SMS. Además de permitir el envío de mensajes a múltiples destinatarios, esta plataforma permite gestionar los usuarios registrados, y brinda la posibilidad a cada usuario de crear y modificar sus propias listas de contactos. Los administradores pueden llevar un control de las estadísticas de los mensajes que se envían y reciben a través de los módems conectados al sistema, esto se refleja mediante la generación de diferentes gráficas que ilustran estos resultados. Aunque esta plataforma cumple con los objetivos para los que inicialmente fue diseñada, la misma carece de algunas funcionalidades necesarias para mejorar el proceso de administración de la misma.

Las estadísticas de los mensajes enviados, recibidos y en cola, se pueden observar en forma gráfica dentro de la aplicación, pero no es posible exportar reportes con esta información en formatos como PDF². Avisos importantes como un posible fallo en Gammu, Gammu Smsd, o la necesidad de recargar alguno de los módems conectados al sistema no están soportados actualmente. No existe la posibilidad de ejecutar comandos vía SMS en el servidor, de manera que permita a los administradores realizar acciones de forma remota. No es posible gestionar el saldo de los módems conectados al sistema, ni el precio de los mensajes que se envían a través de éstos. Esta aplicación soporta la interacción con aplicaciones externas mediante un servicio web que permite el envío de mensajes SMS, pero no es posible entregar los mensajes de respuesta que se generan a estas entidades externas.

A partir de esta situación se deriva el siguiente **problema a resolver**:

¿Cómo resolver las limitantes de la Plataforma de servicio de SMS basado en Gammu para facilitar el proceso de administración de la misma?

² **PDF**: Portable Document Format o Formato de Documento Portable.

En correspondencia con el problema, el **objeto de estudio** lo constituyen las Plataformas de Mensajerías Móviles, enmarcando en el **campo de acción** a las Plataformas que utilicen el Gammu y los módems GSM³ para el envío de los mensajes.

El **objetivo general** de la investigación es desarrollar la versión 2.0 de la Plataforma de mensajería SMS utilizando módems GSM. Se derivan de éste, los siguientes **objetivos específicos**:

- Agregar al sistema un módulo que permita gestionar el saldo de los diferentes módems, y el precio de los mensajes que se envían, utilizando el sistema Gammu y las modificaciones realizadas, a través de una interfaz web amigable y sencilla.
- Crear un módulo que permita exportar un reporte en formato PDF con las estadísticas de los mensajes que se envían y reciben a través de la aplicación.
- Posibilitar el envío de un correo electrónico a los administradores cuando sea necesario la recarga de alguno de los módems conectados.
- Implementar un mecanismo de notificaciones vía SMS a los administradores ante cualquier posible fallo o problema en Gammu.
- Incluir un módulo que posibilite la ejecución de comandos remotos en el servidor vía SMS.
- Crear un servicio web que permita a las aplicaciones externas enviar mensajes SMS, y en caso de producirse una respuesta, entregar el mensaje recibido a la entidad a la cual va dirigido.

La **idea a defender** plantea que el desarrollo de la versión 2.0 de la Plataforma de Mensajería SMS utilizando módems GSM permitirá mejorar el proceso de administración de la aplicación.

Para dar cumplimiento a los objetivos planteados con anterioridad se definen como **tareas de investigación** las siguientes:

³ **GSM**: Global System for Mobile Communications o Sistema Global para Comunicaciones Móviles.

- Definir los métodos teóricos y empíricos que serán empleados en el diseño metodológico de la investigación.
- Realizar un estudio sobre las aplicaciones que posean características similares a las que se esperan lograr con esta investigación en el mundo y en Cuba.
- Realizar un estudio sobre el funcionamiento de CodeIgniter, Doctrine y Zend como frameworks de desarrollo para la creación de servicios y aplicaciones web.
- Realizar una comparación entre las metodologías de desarrollo de software y definir la más conveniente para el proyecto.
- Analizar el funcionamiento del sistema para lograr el almacenamiento de datos de la aplicación Gammu.
- Seleccionar los patrones de arquitectura y diseño más apropiados para redefinir la arquitectura del producto de software.
- Seleccionar los nuevos parámetros a representar en la generación de estadísticas de los mensajes.
- Seleccionar los comandos adecuados para interactuar con Gammu desde el lenguaje de programación PHP.
- Realizar las pruebas a cada una de las funcionalidades correspondientes a la versión 2.0 de la Plataforma de Mensajería SMS utilizando módems GSM.

Para desarrollar la investigación se utilizaron **métodos empíricos** como la **observación**, que sirvió de base para comprender el funcionamiento de los distintos sistemas de envío masivo de SMS que implementan función de pasarela, y el comportamiento de las aplicaciones que son manejadas remotamente, a través de mensajes SMS.

También se usaron **métodos teóricos** como el **inductivo-deductivo** para entender los procesos realizados por Gammu y Gammu Smsd para lograr el envío de los mensajes y el **analítico-sintético** que valió para analizar las modificaciones y las funcionalidades que necesitan incluirse en la Plataforma para cumplir con el objetivo propuesto.

El presente trabajo 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; se describen también 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 ejecución de las pruebas a realizar.

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

1.1. Introducción

En este capítulo se abordan los diferentes conceptos y tecnologías que han sido objeto de análisis y estudio en el transcurso de la investigación como son: las pasarelas de mensajería móvil o SMS Gateway, la aplicación Gammu y Gammu Smsd, y el mecanismo de acceso a datos mediante servicios web. Se fundamentan también las distintas herramientas, metodologías y lenguaje utilizados para dar solución a la problemática presentada con anterioridad.

1.2. Conceptos fundamentales relacionados con el dominio del problema

1.2.1. Módem GSM

Un módem GSM es un tipo especial de módem, al cual se le incluye una tarjeta SIM⁴ y el mismo actúa de forma similar a un teléfono móvil. Conectando este dispositivo a una computadora, se logra la comunicación entre esta última y la red móvil. Muchos de estos módems son utilizados hoy en día para enviar y recibir mensajes SMS. (NOWSMS)

1.2.2. GAMMU y GAMMU SMSD

Gammu es una utilidad de línea de comandos, que se puede utilizar para controlar un teléfono móvil, está escrito en C y presenta excelente compatibilidad para manejar dispositivos que utilicen el estándar GSM. Proporciona el acceso a una amplia gama de funciones del teléfono, existiendo diferencias entre estas funciones al cambiar de un teléfono a otro. Dentro del Gammu se puede encontrar el Gammu Smsd, cuya función es buscar entre los módems que se encuentren conectados al sistema, para obtener los mensajes almacenados en estos medios, almacenarlos en una zona definida y enviar los mensajes que se encuentren en la cola de salida de estos medios. Se encuentra disponible para los sistemas basados en UNIX y posee una amplia comunidad de desarrolladores que sacan nuevas versiones con errores corregidos periódicamente. (Gammu)

⁴ **SIM:** Subscriber Identity Module o Módulo de Identificación del Suscriptor.

1.3. Sistemas que permiten la ejecución de comandos de forma remota en el servidor

Generalmente la administración de servicios informáticos se lleva a cabo de forma presencial, debido a que en cualquier momento un servidor puede verse comprometido y quedar total o parcialmente fuera de servicio, por un determinado factor o por el propio entorno donde esté ubicado; afectando así la calidad y el prestigio de la empresa que lo administra y provocando daños que pueden ser irreversibles en el peor de los casos. Casi siempre al frente de esta tarea se encuentran uno o varios administradores, con el propósito de velar por el buen funcionamiento de los servidores que se encuentran trabajando. Si uno de estos se viera afectado, el responsable deberá solucionarlo de forma presencial y en el menor tiempo posible; lo que implica un mayor costo financiero en cuanto a recursos humanos se refiere, por tener un personal trabajando las 24 horas del día, los 7 días de la semana. Por esta razón actualmente existen en todo el mundo varias soluciones con el objetivo de administrar servidores de manera remota, haciendo uso de la tecnología inalámbrica por medio de mensajes de texto SMS.

1.3.1. Sistema WDS⁵ para la Administración remota de servidores

Es un sistema capaz de monitorear y controlar los servicios que presta uno o más servidores por medio de mensajes de texto SMS, mediante un teléfono celular. Esta solución fue desarrollada en la Universidad Francisco Gavidia ubicada en San Salvador, El Salvador.

Está basada en el proyecto de código abierto Gammu, abarcando aplicaciones, scripts y drivers para administrar varias funciones de teléfonos celulares y posee también un esquema de base de datos para el manejo de los mensajes SMS. (Rivera Rivera, y otros, 2008)

Entre las ventajas de WDS se encuentran:

- El administrador puede monitorear y controlar el servidor desde cualquier lugar y a cualquier hora.
- En términos de seguridad este proyecto ofrece mecanismos para verificar la estructura que debe poseer el mensaje a ser enviado por el administrador, de forma que WDS ejecutará aquellos mensajes que poseen dicha estructura y si no, los ignora, manejándolos como un error.

⁵ **WDS:** Watch Dog System o Sistema de Perro Guardián.

Esta solución ofrece grandes ventajas, ya que no necesita conectividad a Internet, ni inversión en infraestructura de comunicaciones y además ahorra los costos en recursos humanos.

1.4. Aplicaciones de envío masivo de SMS que funcionan como SMS GATEWAY

En el mercado actual existe un gran número y variedad de pasarelas SMS Gateway, cada una de ellas con sus especificaciones, que responden a los diferentes propósitos para los que fueron creadas. A continuación se describen las principales características de algunas plataformas que reflejan cierta semejanza entre su comportamiento y lo que se espera lograr con esta investigación.

1.4.1. Activa SMS

Radicada en España, ofrece la posibilidad de enviar y recibir SMS a móviles de una manera sencilla, rápida y económica a través del servicio de pasarela (SMS Gateway). El envío se realiza desde cualquier computadora con conexión a Internet; para utilizarla únicamente es necesario disponer de un cliente HTTP⁶ que se comuniquen con la pasarela, utilizando un protocolo descrito en una API⁷ que provee la aplicación. (ACTIVASMS)

Algunas de las características presentes en el sistema Activa SMS en el momento de esta investigación son:

- Envíos de mensajes: Múltiples o Individuales.
- Gestión de Lista de Contactos.
- Historial de Envíos.
- Informes detallados de envío.
- Aviso de saldo.

Se desarrolló en el año 2009 y brinda cobertura a 900 operadoras en un total de 214 países. Esta plataforma no brinda información sobre las herramientas que se utilizaron en su desarrollo.

⁶ **HTTP:** Hypertext Transfer Protocol o Protocolo de Transferencia de Hipertexto.

⁷ **API:** Application Programming Interface o Interfaz de Programación de Aplicaciones.

1.4.2. Altiria

Esta aplicación radicada en Madrid, España, ofrece la posibilidad de integrar en el software o web del cliente, funciones de envío de SMS, haciendo función de pasarela. El usuario necesita un cliente HTTP, que se comuniquen con la pasarela utilizando un protocolo descrito en una API que provee dicha aplicación, de este modo, se elimina el costo y la complejidad de la conexión directa con los operadores; Esta pasarela está implementada a través de servidores seguros con balanceo de carga y servicios duplicados para garantizar un servicio de alta disponibilidad. (Altiria)

El funcionamiento se describe a continuación:

El cliente, desde su aplicación, compone el SMS que desea enviar y selecciona los números de teléfono destinatarios.

Luego se conecta a la pasarela y envía una petición HTTP POST, incluyendo el texto del mensaje y los destinatarios como parámetros.

De esta forma Altiria recibe la petición del cliente y la reencamina hacia la operadora de telefonía móvil adecuada.

Finalmente, la operadora recibe la información y envía el mensaje SMS a los destinatarios seleccionados, que lo reciben en su móvil.

1.4.3. Blue Eye

Es una plataforma que radica en Cuba, destinada a desarrollar servicios de valor agregado que funcionan mediante mensajes de texto.

En general, el escenario más común de uso consiste en que los suscriptores clientes de las aplicaciones BlueEye envían un mensaje de texto a un número corto de servicio, ofrecido mediante el operador celular local. La primera palabra de este SMS se considera una palabra clave, y en dependencia de la misma, el mensaje recibido por la plataforma es dirigido hacia una de las aplicaciones asociadas a esa palabra. La aplicación en cuestión procesa el pedido del usuario, cuyos detalles vienen en el SMS, y devuelve una respuesta apropiada vía SMS al suscriptor que solicitó el servicio específico inicialmente. (BlueEye)

BlueEye está basado en Java, EJB3, Hibernate y otras tecnologías que han probado su madurez en el mundo empresarial y los servicios que brinda se encuentran divididos en tres grandes grupos:

- Servicios a usuarios: Brinda la oportunidad de consultar diferentes servicios de gran utilidad, como es el caso de las Noticias, el Horóscopo Diario, Cartelera, El tiempo, entre otros.
- Servicios a empresas: Permite a los centros culturales mediante las Carteleras SMS, ofrecer la programación de sus actividades diarias en un espacio de 250 caracteres por día, para ser consultada desde los móviles.
- Servicios en desarrollo (avance de lo que se encuentra en producción): Muestra los servicios que están próximos a lanzar, en fase de pruebas, brindando la posibilidad a los usuarios de usarlos y enviar sus opiniones y sugerencias al respecto.

Esta solución informática brinda sus servicios a toda persona natural o jurídica que sea titular de un contrato de servicio de telefonía móvil con ETECSA (Empresa de Telecomunicaciones de Cuba) y cuenta con varios años de experiencia y un gran número de profesionales trabajando para mejorar día a día los servicios que se brindan. Esta aplicación funciona mediante una conexión directa con el SMS Center de ETECSA, esto le da la ventaja de poder usar números cortos de servicio y tener una mayor velocidad de conexión.

1.5. Servicios web

Con los avances tecnológicos experimentados en los últimos tiempos en el campo de la informática y las comunicaciones, y el creciente mercado, se hace necesario que los sistemas interactúen entre sí, que la comunicación entre ellos se logre de una manera estándar y sencilla; una de las formas de lograr lo anteriormente expuesto lo constituyen los Servicios Web.

Existen varias bibliografías que tratan diferentes conceptos sobre lo que es un Servicio Web, esto hace que el asunto de dar una definición acertada sea algo relativamente complejo.

Una posible definición sería: Conjunto de aplicaciones o tecnologías con capacidad para interoperar en la Web. Estas aplicaciones o tecnologías intercambian datos entre sí con el objetivo de ofrecer sus servicios. Los proveedores ofrecen sus servicios como procedimientos remotos y los usuarios solicitan un servicio llamando a estos procedimientos a través de la web. (W3C)

La W3C⁸ define un Servicio Web como: “...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 WSDL⁹. Otros sistemas interactúan con el servicio web en una manera prescrita por su descripción usando mensajes SOAP¹⁰, típicamente enviados usando mensajes HTTP, con una serialización XML¹¹ en relación con otros estándares relacionados con la web.” (W3C)

Estos servicios proporcionan mecanismos de comunicación estándares entre diferentes aplicaciones, que interactúan entre sí para presentar información dinámica al usuario, fomentan los estándares y protocolos basados en texto, lo que hace más fácil acceder al contenido y entender su funcionamiento. Al apoyarse en el protocolo HTTP, se pueden aprovechar los sistemas de seguridad firewall sin necesidad de cambiar las reglas de filtrado. Permiten además que servicios y software de diferentes compañías ubicadas en diferentes lugares geográficos puedan ser combinados fácilmente para proveer servicios integrados.

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; está basado en el lenguaje XML, que permite la interacción entre varios dispositivos y tiene la capacidad de transmitir información compleja. Los datos pueden ser transmitidos a través de los protocolos HTTP, SMTP¹², etc.

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

1.6. Metodologías de desarrollo de software

Son un conjunto de métodos, reglas, que sirven como un modelo para desarrollar el software y que guían a la dirección del proyecto y a los componentes del desarrollo a realizar ciertas comprobaciones sistemáticas de modo que el producto o resultado del trabajo final tenga éxito, o

⁸ **W3C:** World Wide Web Consortium.

⁹ **WSDL:** Web Services Description Language o Lenguaje de Descripción de Servicios Web.

¹⁰ **SOAP:** Simple Object Access Protocol o Protocolo de Acceso a Objeto Simple.

¹¹ **XML:** Extensible Markup Language o Lenguaje de Marcas Extensibles.

¹² **SMTP:** Simple Mail Transfer Protocol o Protocolo Simple de Transferencia de Correo.

sea, que se cumplan los objetivos y los requerimientos, se entregue en tiempo, forma y cumpla con la calidad que requiere o que el cliente exija. (Quispe Carita, y otros, 2011)

Existen fundamentalmente dos tipos de metodologías para contribuir al desarrollo de software: las tradicionales y las ágiles.

Las tradicionales están basadas en normas provenientes de estándares seguidos por el entorno de desarrollo; ofrecen por lo general cierta resistencia a los cambios que puedan producirse durante el ciclo de desarrollo del producto, y es un proceso muy bien controlado con muchas políticas y normas. (Quispe Carita, y otros, 2011)

En este grupo se encuentra RUP¹³, un proceso iterativo e incremental, lo que facilita en cada iteración tener una nueva versión del sistema, garantizando así la calidad durante todo el ciclo de vida del proyecto, aunque presenta algunos inconvenientes para su aplicación en algunos proyectos como son:

La generación de un gran cúmulo de documentación que no genera valor respecto a la calidad del desarrollo y que necesita incluir un mayor número de personas al equipo de desarrollo, tales como especialistas en casos de uso. (Rumbaugh, y otros, 2000)

Por otro lado las metodologías ágiles están basadas en heurísticas provenientes de prácticas de producción de código, están especialmente preparadas para sufrir cambios durante el proyecto y son un proceso menos controlado y con pocos principios, por lo que desarrollan software en cortos períodos de tiempo y exigen poca documentación. Dentro de las metodologías ágiles se encuentra Extreme Programming (XP). (Carmen Penadés, y otros, 2004)

1.6.1. Extreme Programming

Es una metodología ágil, que centra su atención en potenciar las relaciones interpersonales como clave para el éxito en el desarrollo de software, para esto promueve el trabajo en equipo, se preocupa por el aprendizaje de los desarrolladores, y propicia un buen clima de trabajo para los mismos. Se basa en la retroalimentación continua entre el cliente y el equipo de desarrollo, en la comunicación fluida entre todos los participantes y en el coraje para enfrentar los cambios. Está diseñada para mitigar los riesgos en proyectos con corto tiempo de entrega y con continuos cambios de requerimientos. Este enfoque muestra su efectividad en proyectos con requisitos muy

¹³ **RUP:** Rational Unified Process.

cambiantes y cuando se exige reducir drásticamente el tiempo de desarrollo pero manteniendo siempre una alta calidad. (Carmen Penadés, y otros, 2004)

La metodología Extreme Programming (XP) consta de 6 fases:

- **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 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, como son: el rendimiento y la confiabilidad del sistema.

Para la realización del sistema se decide emplear la metodología XP, por ser este un proyecto pequeño donde todo el trabajo es realizado por una pareja de programadores, en un ambiente de trabajo donde el riesgo de desarrollo es elevado debido al corto tiempo de entrega planteado y a los continuos cambios de requerimientos; además el cliente forma parte del equipo de desarrollo, de esta manera se logra una mejor retroalimentación, corrección de errores y se garantiza la entrega de un producto final con la calidad requerida. XP es una metodología diseñada para mitigar los riesgos en proyectos con estas características.

1.7. Lenguaje de modelado UML (Unified Modeling Language o Lenguaje Unificado de Modelado)

UML es un lenguaje de representación visual que permite combinar diversos elementos gráficos y crear diagramas. Se usa para modelar sistemas y usa tecnología orientada a objetos. El lenguaje unificado de modelado describe lo que hará un sistema pero no dice cómo implementarlo. Su objetivo es visualizar, especificar, construir y documentar los artefactos que se crean durante el proceso de desarrollo. Involucra todo el ciclo de vida del proyecto y está pensado para varios lenguajes y plataformas.

1.8. Lenguajes de programación

Para desarrollar las funciones necesarias a incluir en la aplicación se ha decidido la utilización del lenguaje de programación PHP v5.2.5 del lado del servidor, mientras que del lado del cliente se propone utilizar los lenguajes HTML 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, la mayor parte de su sintaxis ha sido tomada de C, Java y Perl con algunas características específicas de sí mismo. La meta de este lenguaje es permitir rápidamente a los desarrolladores la generación dinámica de páginas web. (DesarrolloWebPHP)

Dispone de una gran cantidad de características que lo convierten en un buen lenguaje para la creación de páginas web dinámicas como son: (DesarrolloWebPHP)

- Soporte para una gran cantidad de bases de datos: MySQL, PostgreSQL, Oracle, MS SQL Server, Sybase mSQL, Informix, entre otras.
- Permite generar documentos en PDF y analizar código XML.
- Es un producto de código abierto, soportado por una gran comunidad de desarrolladores permitiendo que los fallos de funcionamiento y seguridad se encuentren y reparen rápidamente.
- El código se pone al día continuamente con mejoras y extensiones del lenguaje para ampliar sus capacidades.

JavaScript es un lenguaje de programación que se utiliza principalmente para crear páginas web dinámicas. Se diseñó con una sintaxis similar al lenguaje de Programación C, aunque adopta nombres y convenciones del lenguaje de programación Java. Sin embargo Java y JavaScript no están relacionados y tienen semánticas y propósitos diferentes.

Técnicamente, JavaScript es un lenguaje de programación interpretado, por lo que no es necesario compilar los programas para ejecutarlos. En otras palabras, los programas escritos con JavaScript se pueden probar directamente en cualquier navegador sin necesidad de procesos intermedios. (Eguíluz Pérez, 2008)

1.9. Framework Doctrine

Doctrine está definido como un conjunto de bibliotecas de PHP que se centra principalmente en la prestación de servicios de persistencia y sus funcionalidades relacionadas. Es un ORM¹⁴ que funciona como la parte superior de un DBAL¹⁵. 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 DQL¹⁶. A su vez el DBAL de Doctrine está caracterizado como una potente capa de abstracción de base de datos con muchas ventajas para la gestión de distintos esquemas de bases de datos. Cuenta con una gran comunidad de desarrolladores y documentadores que ofrecen una extensa referencia de su API y todas sus funcionalidades, acortando así 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. (Doctrine)

1.10. Framework CodeIgniter v1.7.2

CodeIgniter es un 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. (CodeIgniter, 2011)

Algunas de las características de CodeIgniter en el momento de la investigación son las siguientes:

- Un framework de trabajo de pequeño formato.

¹⁴ **ORM:** Object-Relational Mapping o Mapeo Relacional de Objetos.

¹⁵ **DBAL:** Database Abstraction Layer o Capa de Abstracción de Base de Datos.

¹⁶ **DQL:** Doctrine Query Language.

- Rendimiento excepcional, pues va cargando dinámicamente las librerías en el momento que lo requieran.
- No requiere de mucha configuración.
- Evita la complejidad, favoreciendo las soluciones simples.
- Documentación clara y completa.

Por todas estas características se escoge a CodeIgniter como framework para el desarrollo de la solución.

1.11. Zend Framework v1.7.8

Zend Framework se trata de un conjunto de clases para el desarrollo de aplicaciones y servicios web con PHP, que brinda soluciones para construir aplicaciones modernas, robustas y seguras. (ZendFramework)

Entre las principales ventajas de su uso se encuentran:

- Ofrece muchas facilidades para el acceso a recursos avanzados como servicios web, que de otro modo resultan 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.

Todas estas características hacen de éste un potente marco de trabajo, del cual es posible utilizar sólo los componentes requeridos para brindar en este caso el servicio web y lograr el envío de correos electrónicos.

1.12. Entorno de desarrollo: Netbeans 7.0 para PHP

NetBeans IDE¹⁷ es una herramienta que permite escribir, compilar, depurar y ejecutar programas. Está escrito en Java, pero puede servir para cualquier otro lenguaje de programación. Tiene

¹⁷ **IDE:** Integrated Development Environment o Entorno de Desarrollo Integrado.

soporte para AJAX y para la creación de aplicaciones orientadas a servicios (SOA), incluyendo herramientas de esquemas XML, un editor WSDL, y un editor BPEL¹⁸ para servicios web.

Desde julio de 2006, NetBeans IDE es licenciado bajo la Common Development and Distribution License (CDDL), una licencia basada en la Mozilla Public License (MPL). Lo que hace que cada vez más desarrolladores de aplicaciones, estén utilizando NetBeans como IDE de desarrollo. (NetBeans)

1.13. Herramienta de Modelado: Visual Paradigm 6.4

Visual Paradigm es una herramienta CASE¹⁹ que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño, construcción, pruebas y despliegue. Permite modelar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación. (Ecured)

Dentro de sus principales características se encuentran:

- Disponibilidad en varias plataformas (Windows, Linux).
- Capacidades de ingeniería directa e inversa.
- Soporta aplicaciones web.
- Soporta UML versión 2.1.

Estas características unidas a la experiencia que posee el equipo de desarrollo con esta herramienta y que la Universidad de las Ciencias Informáticas posee la licencia comercial para su uso, ayudan a decidir que esta sea la herramienta de modelado utilizada en el desarrollo de la solución.

1.14. Servidor Web: Apache v2.2.12

El Apache HTTP Server es un servidor web de código abierto; 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.

¹⁸ **BPEL:** Business Process Execution Language o Lenguaje de Ejecución de Procesos de Negocio.

¹⁹ **CASE:** Computer Aided Software Engineering o Ingeniería de Software Asistida por Computadora.

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 para muchas otras tareas donde el contenido necesita ser puesto a disposición en una forma segura y confiable. Permite la comunicación segura mediante TLS²⁰ y autenticar usuarios a través de un servidor LDAP²¹, entre otras.

Posee una arquitectura modular, de código abierto y multiplataforma y es fácil de conseguir ayuda y soporte por la amplia comunidad de desarrolladores y seguidores en todo el mundo. (APACHE)

1.15. Sistema Gestor de Bases de Datos: PostgreSQL 8.4.7

PostgreSQL es un sistema de gestión de bases de datos relacional orientado a objetos, distribuido bajo la licencia Berkeley Software Distribution (BSD), permitiendo así que el código fuente pueda ser usado en software no libre.

Incluye la mayor parte de los tipos de datos especificados en los estándares SQL92 y SQL99 y presenta soporte de protocolo de comunicación encriptado por SSL²², extensiones para alta disponibilidad, nuevos tipos de índices y minería de datos. Permite crear, editar, copiar, extraer y bajar todo objeto de las bases de datos tales como esquemas, tablas, vistas, funciones, dominios, reglas, secuencias, idiomas y operadores; construye consultas visualmente, ejecuta consultas y scripts SQL, visualiza y edita datos, representa datos como diagramas, exporta e importa datos desde y hacia los formatos de archivos de uso más popular. Además administra roles, usuarios, grupos y sus privilegios. Está soportado por una amplia comunidad de desarrolladores en todo el mundo denominada PGDG (PostgreSQL Global Development Group). (PostgreSQL)

1.16. Conclusiones

En este capítulo se analizaron varios sistemas que ejecutan procedimientos similares a los que se desean lograr en la aplicación, y se determinó que ninguno de ellos es aplicable a la problemática planteada, pero el análisis de su comportamiento y diseño pueden servir como objeto de estudio y

²⁰ **TLS:** Transport Layer Security o Seguridad de la Capa de Transporte

²¹ **LDAP:** Lightweight Directory Access Protocol o Protocolo Ligero de Acceso a Directorios

²² **SSL:** Secure Sockets Layer o Capa de Conexión Segura.

punto de partida en la solución que se desea obtener. Así mismo se determinó la metodología a utilizar para el desarrollo del sistema y las herramientas necesarias para su implementación.

CAPÍTULO 2. CARACTERÍSTICAS DEL SISTEMA, EXPLORACIÓN Y PLANIFICACIÓN

2.1. Introducción

En este capítulo se describen las nuevas funcionalidades que serán incluidas en el sistema. Se hace alusión a las fases de Exploración y Planificación, propias de la metodología de desarrollo utilizada, donde se confeccionan las historias de usuarios importantes para cada iteración definida por el equipo de desarrollo.

2.2. Problemática

En la actualidad existen numerosas aplicaciones web que permiten el envío de mensajes SMS, y otras que cumplen función de pasarela entre las diferentes operadoras y el cliente. Cada una de ellas tiene sus propias características, pero todas persiguen el objetivo de automatizar el proceso de envío y recepción de mensajes SMS. En el capítulo anterior se realizó un estudio de los sistemas que permiten la ejecución de comandos remotos en el servidor vía SMS; así como aquellos que permiten el envío masivo de mensajes de texto y los que hacen función de pasarela SMS. Se realizó un análisis funcional de cada uno de ellos, con el fin de adquirir experiencia para el desarrollo de la propuesta.

El centro de Telemática de la UCI cuenta con la Plataforma de servicio de SMS basado en Gammu, creada con el propósito de posibilitar el envío masivo de SMS. Esta plataforma necesita incluir un mecanismo de notificaciones, debido a que avisos importantes como un fallo crítico en la aplicación Gammu o la necesidad de recargar alguno de los módems conectados al sistema, no están soportados actualmente. Otra limitante que presenta la solución es que no brinda la posibilidad de ejecutar comandos vía SMS en el servidor, de manera que permita a los administradores realizar acciones de forma remota. Las estadísticas de los mensajes enviados, recibidos y en cola, se pueden observar en forma gráfica dentro de la aplicación, pero no es posible exportar reportes en formato PDF con esta información. No es posible gestionar el precio de los módems conectados al sistema, ni el precio de los mensajes que se envían a través de éstos. Por último se cuenta con una interfaz que permite la interacción de aplicaciones externas mediante peticiones HTTP, pero no es posible entregar los mensajes recibidos a estas aplicaciones externas.

2.3. Objeto de automatización

Se pretende automatizar los procesos que intervienen en la generación de reportes, que contengan la información relacionada con las estadísticas de los mensajes que fluyen a través de la aplicación; así como el envío de notificaciones vía SMS a los administradores del sistema, con información referida a posibles errores en el servidor donde se encuentra corriendo la aplicación o la necesidad de recargar alguno de los módems conectados al sistema; brindando la posibilidad de ejecutar comandos remotos vía SMS de forma segura. Se espera automatizar también los procesos relacionados con la recepción y entrega de mensajes a entidades externas, haciendo función de pasarela SMS.

2.4. Propuesta del sistema

La presente investigación tiene como objetivo dotar de nuevas funcionalidades al Sistema de servicio de SMS basado en Gammu, con el propósito de hacer de ésta un mejor uso por parte de los administradores, brindando así mejores resultados, ver Figura 2.1.

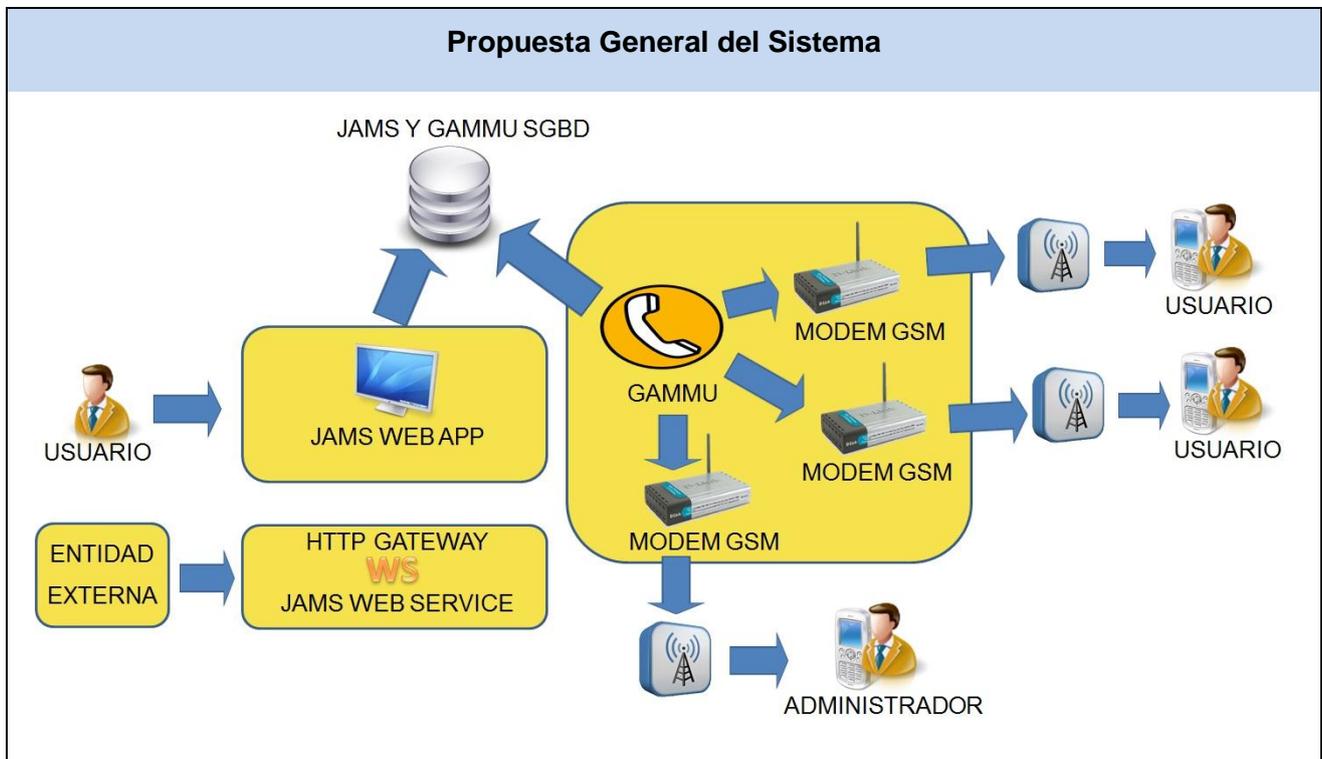


Figura 2.1 Propuesta General del Sistema

Desde un punto de vista más objetivo, con la nueva versión del sistema se pretende adicionar nuevas estadísticas relacionadas con el envío y la recepción de los mensajes de texto que fluyen a través de la aplicación; se procura además exportar éstas estadísticas recogidas a formato PDF, para su posterior análisis y almacenamiento fuera del ámbito de la aplicación. Se espera gestionar el saldo disponible en cada uno de los módems que se encuentre conectado, así como el precio de los mensajes que se envían a través de éstos y un saldo mínimo establecido para lograr, en el momento que algún módem quede con un saldo inferior al saldo mínimo establecido, generar una notificación a los administradores del sistema mediante correo electrónico, con el aviso de que es necesaria la recarga del módem en cuestión. El sistema cumplirá la función de pasarela SMS, permitiendo a entidades externas enviar y recibir mensajes a través de la aplicación mediante un servicio web; se espera que el sistema notifique vía SMS a los administradores ante la presencia de algún posible error en el programa Gammu, de esta forma se podrán ejecutar comandos en el servidor de manera remota, mediante un SMS de una forma segura, ver Figura 2.2.

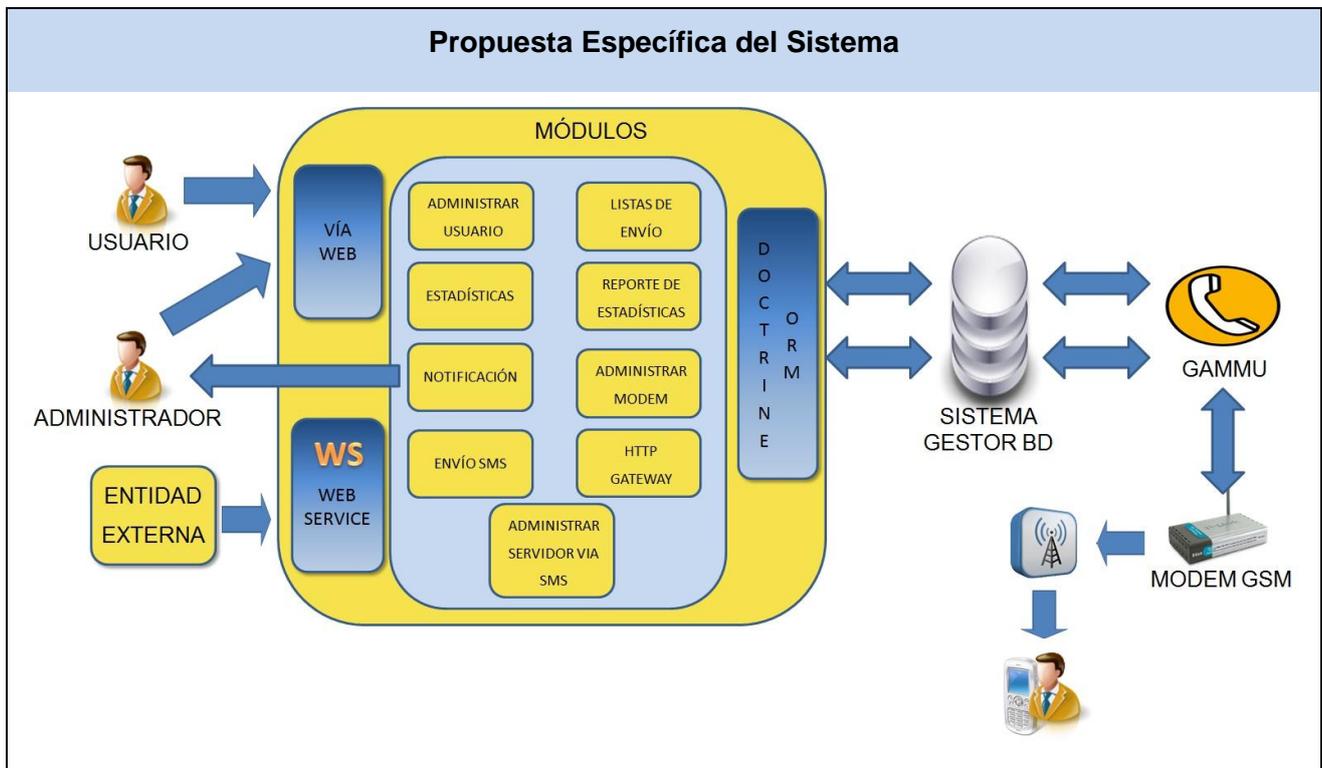


Figura 2.2 Propuesta Específica del Sistema

2.5. Características no funcionales del sistema

Los requisitos no funcionales son cualidades o propiedades que el producto debe tener, estas características hacen al producto atractivo, usable, rápido y confiable. Son importantes para que clientes y usuarios puedan valorar las propiedades de un producto, pues si se conoce que el mismo cumple con toda la funcionalidad requerida, las propiedades no funcionales pueden marcar la diferencia entre un producto bien aceptado y uno con poca aceptación.

2.5.1. Usabilidad

Se requiere una preparación previa para interactuar con el sistema, aunque el manejo de la aplicación es sencillo, permitiendo la fácil comprensión por el usuario.

2.5.2. Disponibilidad

Una vez que el usuario se autentique en el sistema, tendrá el acceso garantizado a las funcionalidades que le correspondan según su rol, siempre que los datos introducidos en la autenticación sean los correctos y que la conexión con el servidor esté disponible.

2.5.3. Apariencia o Interfaz Externa

El sistema contará con una interfaz amigable y sencilla. El diseño se realiza siguiendo las pautas de la interfaz web en su versión inicial; un diseño que muestra en una sola interfaz, el contenido de la sección que el usuario se encuentra utilizando en el momento, evitando las barras de desplazamiento, tanto verticales como horizontales y el uso de un menú accesible en todo momento para facilitar el acceso a las distintas funcionalidades del sistema.

2.5.4. Seguridad

El acceso a las funcionalidades estará determinado por los permisos establecidos por el administrador según el tipo de usuario. Autenticarse será obligatorio para el uso de cualquier servicio de la aplicación. Las contraseñas de cada usuario se almacenarán con el uso de un algoritmo de encriptación que imposibilite su lectura en claro.

La ejecución de comandos remotos en el servidor sólo se podrá efectuar si el SMS entrante tiene el formato establecido previamente, especificando un usuario y una contraseña válidos con permisos de administración. Se verifica que el número de teléfono del emisor corresponda con el número de teléfono del administrador que proporciona su usuario y contraseña en el mensaje, en caso de que no se cumpla alguna de estas condiciones, se reporta en el Log de la aplicación la fecha, la hora, el mensaje y el número de teléfono que envió el mensaje.

Si se recibe un mensaje, de un número de teléfono al cual no se le ha enviado ningún mensaje desde la aplicación, se procede a registrar en el Log: la fecha, la hora, el mensaje y el teléfono emisor.

Si se recibe una petición HTTP invocando al método Request o al método RunOnError, y además esta petición proviene de un IP diferente al IP del servidor donde está corriendo la aplicación, se procede a escribir en el Log: la fecha, la hora y el IP que hace la petición. Estas funciones sólo pueden ser llamadas por el propio servidor, ninguna PC cliente tiene acceso a estos métodos.

En el formulario Enviar Mensaje se muestra una imagen con un código que debe ser insertado en un campo de texto para su verificación, esto se logra con el uso de la función CAPTCHA de CodeIgniter.

Se utilizan las funciones strip_tags y filter_var de PHP para filtrar los campos de entrada manual de los formularios y así evitar que se introduzcan etiquetas HTML y a su vez eliminar cualquier carácter que no sea letras o números. De esta manera se evita cualquier intento de inyección XSS (Cross Site Scripting). También se utiliza la función addslashes de PHP para escapar las comillas simples de los campos de texto, con el fin de evitar cualquier intento de inyección SQL.

2.5.5. Confidencialidad

La información manejada en el sistema estará protegida de acceso no autorizado y cada usuario con acceso al sistema sólo debe acceder a los datos que le correspondan por su rol y cuenta registrada. El rol de administrador tendrá acceso a todas las funcionalidades del sistema, mientras que el cliente, solo contará con permisos sobre el envío de mensajes y la gestión de sus listas de contactos.

2.5.6. Requisitos de Hardware

Servidores	Especificaciones
Servidor de Aplicaciones Web	<ul style="list-style-type: none"> • RAM: 1 GB o superior. • Disco Duro: 160 GB o superior. • UPS: 1.
Servidor de Base de Datos	<ul style="list-style-type: none"> • RAM: 1 GB o superior. • Disco Duro: 160 GB o superior.

	<ul style="list-style-type: none"> • UPS: 1.
Módem GSM	<ul style="list-style-type: none"> • Cantidad: 1 o más.

Tabla 2.1 Requerimientos de Hardware

2.5.7. Requisitos de Software

Servidores	Especificaciones
Servidor de Aplicaciones Web	<ul style="list-style-type: none"> • Sistema Operativo: Ubuntu 10.10 o cualquiera que tenga instalado el Gammu v1.27.92. • Servidor Web: Apache 2.0 o superior. • Librerías Adicionales: PHP 5 o superior. <ul style="list-style-type: none"> : php5-curl : php5-gd : libapache2-mod-php5 • Paquete de aplicaciones GAMMU 1.27.92.
Servidor de Base de Datos	<ul style="list-style-type: none"> • Sistema Operativo: Windows o Linux o cualquiera compatible con PostgreSQL 8.4.7. • Sistema Gestor de Base de Datos: PostgreSQL 8.4.7 o superior o cualquiera compatible con Gammu 1.27.92 y con CodeIgniter 1.7.2.
Gammu	<ul style="list-style-type: none"> • Paquete de aplicaciones GAMMU 1.27.92.

Tabla 2.2 Requerimientos de Software (Servidor)

Cliente	Especificaciones
Cliente Web	<ul style="list-style-type: none"> • Navegador Web: Mozilla Firefox 2.0+, Safari 3+, Opera 10.6+ Google Chrome 8+.

Tabla 2.3 Requerimientos de Software (Cliente)

2.6. Usuarios relacionados con el sistema

Se define como usuarios relacionados con el sistema a aquellos que interactúan de una u otra forma con la plataforma de mensajería y obtienen un resultado de todos los procesos que se ejecutan en ella.

Usuario	Justificación
Administrador	Es la persona encargada de gestionar el saldo disponible en cada uno de los módems disponibles, así como el saldo mínimo requerido para notificar la necesidad de recargarlos. En caso de existir algún problema en el Gammu, el administrador podrá ejecutar comandos remotos mediante mensajes de texto. Además, llevará un control de las estadísticas de los mensajes y podrá exportar reportes con esa información.
Cliente	Es la persona que realiza el envío de SMS y gestiona sus listas de contactos.
Entidad externa	Es la encargada de interactuar con el sistema a través de un servicio web, para enviar y recibir los mensajes SMS.

Tabla 2.4 Usuarios relacionados con el sistema

2.7. Fase de Exploración

La fase de Exploración es la primera fase definida por la metodología XP. Esta fase comienza por la creación de una serie de historias, llamadas Historias de Usuarios (HU), las cuales definen mediante su redacción qué es lo que verdaderamente necesita el cliente. Es aquí donde se define el alcance real del sistema, permitiendo una familiarización del equipo de desarrollo con las herramientas y tecnologías a usar en la solución.

2.7.1. Historias de Usuarios

Las Historias de Usuarios son las técnicas que utiliza la metodología XP para especificar los requisitos del software. Estas son escritas desde la perspectiva del cliente, donde describen de

forma breve las características que el sistema debe realizar, aunque también los desarrolladores pueden brindar su ayuda en la identificación de las mismas. El contenido de las historias debe ser concreto, sencillo, dinámico y flexible. Cada una de ellas es lo suficientemente comprensible, como para que los programadores puedan estimar con un reducido margen de riesgo, su tiempo de desarrollo. Es el cliente el encargado de asignarle una prioridad a cada Historia de Usuario y es el equipo de desarrollo el encargado de asignarle un costo, este se traduce en las semanas que llevará el desarrollo de las mismas. Por otra parte es bueno destacar que las Historias de Usuarios nuevas pueden describirse en cualquier momento, con esto se comprueba la flexibilidad de la metodología.

2.7.1.1. Clasificación de las Historias de Usuarios

La prioridad en el negocio:

Alto: Se le otorga a las HU que resultan funcionalidades fundamentales en el desarrollo del sistema, a las que el cliente define como principales para el control integral del sistema.

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

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

El riesgo en su desarrollo:

Alto: Cuando en la implementación de las HU se consideran la posible existencia de errores que lleven la inoperatividad del código.

Medio: Cuando pueden aparecer errores en la implementación de la HU que puedan retrasar la entrega de la versión.

Bajo: Cuando pueden aparecer errores que serán tratados con relativa facilidad sin que traigan perjuicios para el desarrollo del proyecto.

Las HU serán representadas mediante tablas divididas por las siguientes secciones:

- **Número:** Número de la historia de usuario incremental en el tiempo.
- **Nombre de Historia de Usuario:** El nombre de la historia de usuario sería para identificarlas mejor entre los desarrolladores y el cliente.

- Usuario: Personas involucradas en el desarrollo de la HU.
- Iteración Asignada: Número de la iteración.
- Prioridad en negocio: Alto, Medio o Bajo.
- Riesgo en Desarrollo: Alto, Medio o Bajo.
- Puntos estimados: Tiempo estimado que se demorará el desarrollo de la HU; cada punto es considerado como una semana de trabajo.
- Puntos Reales: Tiempo que se demoró en realidad el desarrollo de la HU.
- Descripción: Breve descripción de la HU.
- Observaciones: Señalamiento o advertencia del sistema.
- Prototipo de interfaz: Prototipo de interfaz si aplica.

Se muestran a continuación las Historias de Usuarios de prioridad alta, el resto se describen en el Anexo 1 del presente documento:

Historia de Usuario	
Número: 1	Nombre de Historia de Usuario: Enviar mensaje de respuesta hacia entidad externa.
Usuario: Lisandra Soler Ponce Eduardo N. Rosales	Iteración Asignada: 1
Prioridad en negocio: Alto	Puntos estimados: 2
Riesgo en Desarrollo: Alto	Puntos Reales: 2
<p>Descripción: El sistema permitirá a entidades externas enviar y recibir mensajes a través de la aplicación mediante un servicio web. Si un usuario envía un mensaje de texto utilizando el servicio web, y como consecuencia se produce un mensaje de respuesta, el sistema enviará el mensaje y el número de teléfono que responde hacia una web del cliente (entidad externa). El usuario que utilice este servicio debe proporcionar los datos requeridos para realizar la entrega y recepción de sus mensajes.</p>	

<p>Observaciones:</p> <ul style="list-style-type: none"> • El campo URL de la entidad externa es requerido. • El campo Mensaje es requerido: Es la variable en la cual se le enviará el mensaje a la entidad externa. • El campo No. De Telf. es requerido: Es la variable en la cual se le enviará el número de teléfono que respondió al mensaje.
<p>Prototipo de interfaz:</p>

Tabla 2.5 HU: Enviar mensaje de respuesta hacia entidad externa

Historia de Usuario	
Número: 2	Nombre de Historia de Usuario: Enviar notificación de posible error.
Usuario: Lisandra Soler Ponce Eduardo N. Rosales	Iteración Asignada: 1
Prioridad en negocio: Alto	Puntos estimados: 2
Riesgo en Desarrollo: Alto	Puntos Reales: 2
<p>Descripción: Se analiza el fichero log de Gammu, en caso de encontrar algún error se procede a enviar un SMS a los administradores con la información del error producido.</p>	
<p>Observaciones:</p> <ul style="list-style-type: none"> • Sólo se le enviará este tipo de mensaje a los administradores que previamente hayan configurado su cuenta para recibir este tipo de notificación. • El intervalo mínimo de tiempo transcurrido entre un envío de mensaje de error y otro, es de 5 minutos, con la finalidad de no sobrecargar el sistema con un gran número de errores en un corto período de tiempo. • No se tendrán en cuenta errores de tipo "Warning", en el fichero log. Porque se considera que este tipo de error no es crítico para el funcionamiento y estabilidad del Gammu. 	

Prototipo de interfaz:

Tabla 2.6 HU: Enviar notificación de posible error

Historia de Usuario	
Número: 3	Nombre de Historia de Usuario: Reiniciar aplicación Gammu.
Usuario: Lisandra Soler Ponce Eduardo N. Rosales	Iteración Asignada: 1
Prioridad en negocio: Alto	Puntos estimados: 1
Riesgo en Desarrollo: Alto	Puntos Reales: 1
Descripción: Se implementarán las funcionalidades necesarias para reiniciar la aplicación Gammu mediante un SMS.	
Observaciones: El SMS debe cumplir las siguientes condiciones: <ul style="list-style-type: none"> • Formato correcto. • Especificar un usuario y una contraseña válida y con permisos de administración. • Ser enviado desde el propio teléfono de cada administrador. 	
Prototipo de interfaz:	

Tabla 2.7 HU: Reiniciar aplicación Gammu

2.8. Planificación

En la fase de planificación en la metodología XP es donde se le concede la prioridad a cada Historia de Usuario, describiendo sus características y funcionalidades requeridas para el desarrollo del software. Durante la planeación se realiza una estimación del esfuerzo que costará implementar cada Historia de Usuario, este se expresa utilizando como medida el punto, el cual es considerado como una semana de trabajo, donde el equipo de desarrollo labora a tiempo

completo. Esta estimación incluye todo el esfuerzo asociado a la implementación de la Historia de Usuario.

2.8.1. Estimación de esfuerzo por Historia de Usuario

A continuación se muestra la estimación del esfuerzo por cada Historia de Usuario descrita anteriormente para el desarrollo del sistema llegando a los siguientes resultados:

Historia de usuario	Estimación
Enviar mensaje de respuesta hacia entidad externa.	2
Enviar notificación de posible error en el servidor.	2
Reiniciar aplicación Gammu.	1
Exportar reportes a formato PDF.	1
Enviar notificación de Recarga de Módem.	1
Mostrar estadísticas generales de los mensajes.	2
Gestionar Saldo de Alerta en los módems.	2
Gestionar Saldo de Módem.	2
Gestionar precio del Mensaje.	2

Tabla 2.8 Estimación de esfuerzo por Historias de Usuarios

2.8.2. Plan de iteraciones

Luego de ser identificadas y descritas las Historias de Usuarios, además de estimar su esfuerzo, se procede a la planificación de la fase de implementación, donde se realizarán dos iteraciones, las cuales se describen a continuación:

Iteración 1

Esta iteración tiene como objetivo la implementación de las Historias de Usuarios 1, 2 y 3 de prioridad alta. Esta iteración será el inicio de las nuevas funcionalidades para la aplicación que el cliente necesita.

Iteración 2

En esta iteración se tendrán en cuenta las Historias de Usuarios 4, 5, 6, 7, 8 y 9 que facilitarán un mejor funcionamiento de la aplicación para los usuarios que interactúen en ella. Concluida esta iteración se obtendrá la versión 2.0 del producto, listo para ser evaluado y puesto en práctica en un entorno de trabajo real.

2.8.3. Plan de duración de las iteraciones

Este plan se encarga de mostrar las Historias de Usuarios en el orden en que se implementarán en cada iteración, así como la duración estimada.

Iteración	Orden de las HU a implementar	Duración total
1	Enviar mensaje de respuesta hacia entidad externa.	5 semanas
	Enviar notificación de posible error en el servidor.	
	Reiniciar aplicación Gammu.	
2	Exportar reportes a formato PDF.	10 semanas
	Enviar Notificación de Recarga de módem.	
	Mostrar estadísticas generales de los mensajes.	
	Gestionar Saldo de Alerta en los módems.	
	Gestionar Saldo de Módem.	
	Gestionar precio del Mensaje.	

Tabla 2.9 Plan de duración de iteraciones

2.8.4. Plan de entregas

En este plan se detalla la fecha fin de cada iteración, mostrando una versión desarrollada del producto en ese momento hasta lograr el producto final en la fecha establecida. A continuación se muestra el plan de entrega para cada iteración:

FPA: Funcionalidades con prioridad alta.

FPM: Funcionalidades con prioridad media.

Módulo	Fin de la iteración 1 (5 de marzo de 2012)	Fin de la iteración 2 (12 de mayo de 2012)
JAMS	FPA	FPM (JAMS V2.0)

Tabla 2.10 Plan de entrega de las iteraciones

2.9. Conclusiones

En este capítulo quedó definida la propuesta del sistema. Se identificaron las funcionalidades y los requisitos que el mismo debe cumplir. Además, se realizó la descripción de las Historias de Usuarios divididas por iteraciones y la planificación del esfuerzo dedicado al desarrollo de cada una de ellas, en el orden en que se les dará cumplimiento.

CAPÍTULO 3. ANÁLISIS Y DISEÑO DEL SISTEMA

3.1. Introducción

En este capítulo se describe la fase de diseño del sistema propia de la metodología XP. Se define el patrón de arquitectura a seguir, así como los patrones de diseño a utilizar para la implementación del sistema. Se propondrá un nuevo diseño de la base de datos para satisfacer los nuevos requerimientos del software y quedarán definidas las tarjetas CRC (Colaborador - Responsabilidad - Clase) como técnica de diseño.

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. (de la Torre, y otros, 2011)

3.2.1. Estilo de arquitectura: Modelo Vista Controlador

Es una arquitectura estructurada por tres capas: modelo, vista y controlador, donde cada una tiene un objetivo específico:

- Vista: Presenta información dinámica al usuario en una página web.
- Controlador: Responde a eventos, que son acciones del usuario, e invoca peticiones al modelo.
- Modelo: Representa la información con la que trabaja la aplicación, es decir, su lógica de negocio. Contiene funciones para insertar, consultar y actualizar información de la base de datos.

A continuación se describe la arquitectura presente en la solución, ver Figura 3.

Arquitectura utilizada

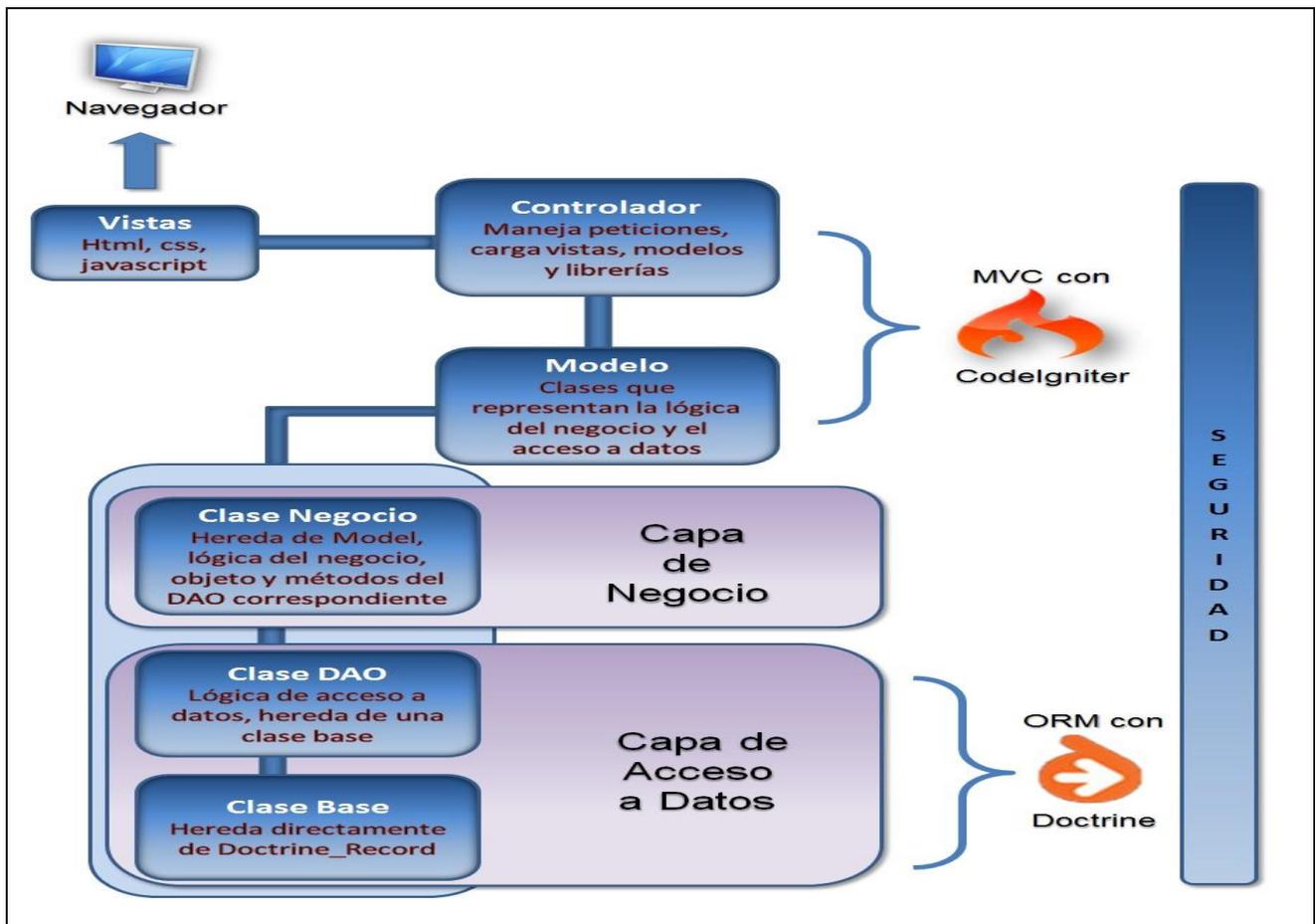


Figura 3.1 Propuesta específica del sistema

La arquitectura aplicada en el sistema cuenta con una combinación del patrón MVC, 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. (de la Torre, y otros, 2011)

Esta arquitectura tiene como objetivo que el sistema cuente con una escalabilidad y mantenibilidad mejorada, haciendo esfuerzos por agilizar el proceso de implementación. Separando la lógica del negocio de las vistas, el mantenimiento de la aplicación se vuelve un proceso relativamente sencillo. Si por ejemplo una misma aplicación debe ejecutarse tanto en un navegador estándar como en un dispositivo móvil, basta crear una vista nueva para cada caso, manteniendo el controlador y el modelo original.

3.3. Patrones de Diseño

Los patrones de diseño son una solución ya probada a un problema o situación. Estos hacen que el sistema esté bien estructurado, por lo que se pueden emplear tantos patrones como se considere necesario. Cada patrón tiene una descripción de la solución según el problema a tratar, donde esa solución puede ser empleada una y otra vez sin repetir las mismas acciones.

3.3.1. Patrones para Asignar Responsabilidades (GRASP)

Los patrones GRASP describen los elementos esenciales de la asignación de responsabilidades a objetos, donde son expresados en forma de patrones.

Se emplearon en el sistema los siguientes patrones GRASP:

Experto

Este patrón se encarga de asignar una responsabilidad al experto en información, o sea, aquella clase que cuenta con la información necesaria para cumplir la responsabilidad. En la solución las clases controladoras manejan las peticiones del cliente, las clases del negocio realizan las validaciones correspondientes de dichas peticiones y las de acceso a datos se encargan de la gestión de los datos entregados desde la lógica del negocio.

Creador

Este patrón es el responsable de asignarle a una clase la responsabilidad de crear una instancia de otra. En la solución las clases del negocio contienen una instancia de su acceso a datos correspondiente y cada clase controladora tiene una instancia de su negocio.

Controlador

Este patrón asigna la responsabilidad del manejo de los eventos de un sistema a una o varias clases. Es usado en cada clase controladora del framework CodeIgniter. Cada petición que se genera es manejada por una clase controladora, ésta invoca a las clases de su negocio que necesita para cumplir con la petición, recibe la respuesta y construye la vista correspondiente para mostrar finalmente la respuesta al usuario.

3.3.2. Patrones GOF

Factory

Se utiliza cuando es necesario tener diferentes objetos, donde todos pertenecen a la misma familia. Este patrón es utilizado en el framework Doctrine en la implementación de un conjunto de clases para el acceso a datos.

Singleton

Con la utilización de este patrón se garantiza una única instancia de una clase. En este caso el framework CodeIgniter posee una instancia única del controlador frontal disponible, para lograr una vía de entrada única a las solicitudes. Además se usó para garantizar una sola instancia de usuario autenticado.

3.4. Tarjetas Clase - Responsabilidad - Colaborador (CRC)

En la fase de diseño de la metodología XP se propone el modelado de Clase-Responsabilidad-Colaborador (CRC), lo que permite un modelo simple para organizar las clases más relevantes en el sistema. El objetivo de este modelado es utilizar tarjetas para la representación organizada de las clases, las cuales tienen los siguientes datos:

- Nombre
- Responsabilidades
- Colaboraciones

Una responsabilidad es lo que sabe o hace la clase. Los colaboradores son aquellas clases que se requieren para que una clase reciba la información necesaria para completar una responsabilidad. Las tarjetas determinan el comportamiento de cada actividad.

Para la nueva versión de la plataforma se identificaron 7 tarjetas CRC, sin incluir las encargadas del acceso a datos del sistema, pues estas basan sus responsabilidades en las definidas para el negocio. A continuación se muestran algunas de estas tarjetas CRC y el resto se especifican en el Anexo 2 del presente documento.

Clase Gateway	
Responsabilidad	Colaborador

Entregar el mensaje recibido a la entidad externa.	Inbox Jamsgateway
Reiniciar aplicación Gammu.	Jamssmsd

Tabla 3.1 Clase Gateway

Clase Notificación	
Responsabilidad	Colaborador
Notificar vía SMS posible error en el servidor.	Outbox JamsUser

Tabla 3.2 Clase Notificación

3.5. Diseño de la Base de Datos

En un sistema donde se almacena información la tarea más importante es el diseño de la base de datos. Con el modelo se pueden describir los elementos existentes en un problema dado, y la forma en que ellos se relacionan entre sí.

3.5.1. Modelo Entidad - Relación

El modelo Entidad-Relación facilita tener una estructura de datos de forma tal que se pueda entender fácilmente lo que se está diseñando. A continuación se muestran las distintas entidades y sus relaciones, agregadas al sistema de almacenamiento de GAMMU para cubrir las funcionalidades del sistema.

Modelo Entidad - Relación

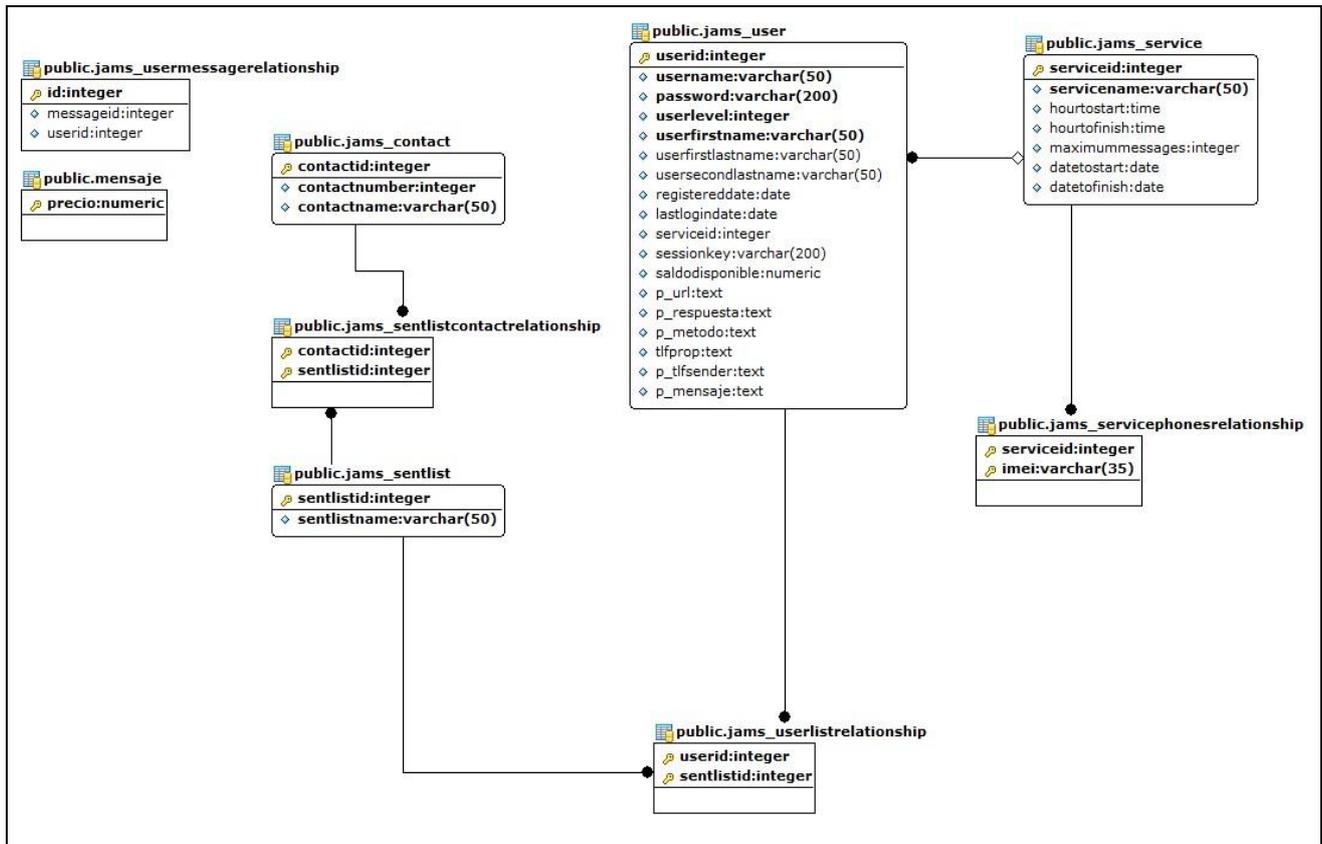


Figura 3.2 Modelo Entidad - Relación

3.6. Conclusiones

En este capítulo se realizó un análisis sobre el patrón arquitectónico utilizado para el desarrollo de la aplicación, así como el estudio de los patrones de diseños empleados para obtener la solución del sistema. También quedaron definidos los módulos, los componentes del sistema y las tarjetas CRC. Se realizó además un nuevo diseño de la base de datos que responde a los nuevos requerimientos del producto.

CAPÍTULO 4. IMPLEMENTACIÓN Y PRUEBA

4.1. Introducción

La metodología XP plantea que la implementación de un software debe realizarse de forma iterativa, obteniendo al culminar cada iteración un producto funcional que debe ser probado y mostrado al cliente para retroalimentar a los desarrolladores con su opinión. En el presente capítulo se describen las iteraciones realizadas correspondientes a la etapa de implementación del sistema, así como las pruebas de aceptación efectuadas sobre el mismo y las tareas de ingeniería generadas en cada una de estas iteraciones.

4.2. Fase de Implementación

Durante el transcurso de las iteraciones se realiza la implementación de las historias de usuario seleccionadas para cada una de éstas. Al inicio de las mismas, se lleva a cabo una revisión del plan de iteraciones y se modifica de ser necesario (Beck, y otros, 2000). Como parte de este plan, se descomponen las HU en tareas de desarrollo, asignando posteriormente cada una de éstas a un equipo (o una persona) responsable de su implementación. Estas tareas son para el uso de los programadores, pueden escribirse utilizando un lenguaje técnico y no necesariamente deben ser entendibles para el cliente.

4.2.1. Iteraciones

Ajustándose a la planificación realizada, se llevaron a cabo dos iteraciones de desarrollo sobre el sistema, obteniéndose al finalizar, un producto listo para su puesta en funcionamiento. A continuación se detallan cada una de las iteraciones.

4.2.1.1. Iteración 1

Durante esta iteración se abordaron las historias de usuario de prioridad alta, con el fin de obtener una versión del producto que cuente con las funcionalidades definidas por el cliente. A continuación se muestran las historias de usuario que se desean realizar en esta iteración, su tiempo de realización estimado y el real.

Componente	Historia de usuario	Estimación	Real
Ejecutar comando remoto.	Enviar notificación de posible error.	2	2

	Reiniciar aplicación Gammu.	1	1
Pasarela HTTP.	Enviar mensaje de respuesta hacia entidad externa.	2	2

Tabla 4.1 Historias abordadas en la primera iteración

4.2.1.1.1. Tareas de Ingeniería

A continuación se muestran las tareas de ingeniería correspondientes a las historias de usuario de prioridad alta abordadas en esta iteración, el resto de prioridad media se encuentran expuestas en el Anexo 3 del presente documento.

Tareas de ingeniería	
No. Tarea: 1	No. Historia de Usuario: 1
Nombre de la Tarea: Enviar mensaje de respuesta hacia entidad externa.	
Tipo de Tarea: Desarrollo	Puntos de estimación: 2/2
Fecha de Inicio: 02/02/2012	Fecha de Fin: 16/02/2012
Programador Responsable: Lisandra Soler Ponce Eduardo Noel Rosales	
Descripción: Se implementarán las funcionalidades necesarias para que el sistema reciba un mensaje, encuentre el destinatario y lo envíe por el método GET o POST (según la configuración del usuario) a una entidad externa (web del cliente) que estará esperando por él.	

Tabla 4.2 Tarea #1 de la HU: Enviar mensaje de respuesta hacia entidad externa

Tareas de ingeniería	
No. Tarea: 2	No. Historia de Usuario: 2
Nombre de la Tarea: Enviar notificación de posible error.	

Tipo de Tarea: Desarrollo	Puntos de estimación: 2/2
Fecha de Inicio: 17/02/2012	Fecha de Fin: 03/03/2012
Programador Responsable: Lisandra Soler Ponce Eduardo Noel Rosales	
Descripción: Se implementarán las funcionalidades necesarias para que el sistema detecte posibles errores en el fichero de log del Gammu y envíe un SMS a los administradores que soliciten recibir los mensajes en sus teléfonos con la descripción del posible error.	

Tabla 4.3 Tarea #2 de la HU: Enviar notificación de posible error

Tareas de ingeniería	
No. Tarea: 3	No. Historia de Usuario: 3
Nombre de la Tarea: Reiniciar aplicación Gammu.	
Tipo de Tarea: Desarrollo	Puntos de estimación: 1/1
Fecha de Inicio: 04/03/2012	Fecha de Fin: 11/03/2012
Programador Responsable: Lisandra Soler Ponce Eduardo Noel Rosales	
Descripción: Se implementarán las funcionalidades necesarias para que el sistema reciba un SMS con un formato predeterminado: (@reiniciar@ @user:usuario @passw:mipassword). Se analizará el contenido del mensaje, en caso de tener el formato correcto, se procederá a verificar que provenga de un teléfono autorizado para esta operación y por último se verificará que el usuario y la contraseña brindados sean correctos y posean permisos administrativos. Si todo lo anterior es correcto, entonces se procederá a reiniciar el Gammu.	

Tabla 4.4 Tarea #3 de la HU: Reiniciar aplicación Gammu

4.2.1.2. Iteración 2

Durante el transcurso de la presente iteración se concluyó la implementación de las funcionalidades con prioridad media. A continuación se detallan las Historias de Usuario concernientes a esta iteración.

Componente	Historia de usuario	Estimación	Real
Gestionar Módem.	Gestionar Saldo de Alerta en los módems.	2	2
	Gestionar Saldo de Módem.	2	2
	Mostrar estadísticas generales de los mensajes.	1	1
	Exportar reportes a formato PDF.	2	2
	Gestionar precio del Mensaje.	2	2
	Enviar Notificación de Recarga de módem.	1	1

Tabla 4.5 Historias abordadas en la segunda iteración

4.2.1.2.1. Tareas de Ingeniería

A continuación se muestran dos de las tareas de ingeniería correspondientes a las historias de usuario de prioridad media abordadas en esta iteración, el resto se encuentran expuestas en el Anexo 3 del presente documento.

Tareas de ingeniería	
No. Tarea: 6	No. Historia de Usuario: 6
Nombre de la Tarea: Gestionar saldo de alerta en los módems.	
Tipo de Tarea: Desarrollo	Puntos de estimación: 2/2

Fecha de Inicio: 10/04/2012	Fecha de Fin: 24/04/2012
Programador Responsable: Lisandra Soler Ponce Eduardo Noel Rosales	
Descripción: Se mostrará una interfaz con la lista de módems disponibles, así como el saldo necesario para generar una alerta en cada uno de ellos, y brindará la opción de modificar ese saldo. Luego se muestra otra interfaz requiriendo al usuario que introduzca el nuevo saldo para generar una alerta, en caso de existir datos incorrectos, se mostrará un mensaje con el error producido.	

Tabla 4.6 Tarea #6 de la HU: Gestionar saldo de alerta en los Módems

Tareas de ingeniería	
No. Tarea: 8	No. Historia de Usuario: 8
Nombre de la Tarea: Gestionar precio del mensaje.	
Tipo de Tarea: Desarrollo	Puntos de estimación: 2/2
Fecha de Inicio: 10/05/2012	Fecha de Fin: 24/06/2012
Programador Responsable: Lisandra Soler Ponce Eduardo Noel Rosales	
Descripción: El sistema ofrecerá al administrador una interfaz con la opción de modificar el precio actual que se cobra por causa del envío de mensajes, en caso de existir datos incorrectos, se mostrará un mensaje con el error producido.	

Tabla 4.7 Tarea #8 de la HU: Gestionar precio del mensaje

4.3. Diagrama de Despliegue

Un diagrama de despliegue representa la distribución física del sistema en términos de cómo se distribuirán las funcionalidades entre los nodos, cada nodo representa un recurso de cómputo, siendo estos procesadores o dispositivos hardware que se necesitarán para el despliegue del sistema.

La arquitectura en tiempo de ejecución de la solución propuesta se modela haciendo uso de tres elementos de hardware: computadora cliente, servidor de aplicaciones y servidor de base de datos, además de contar con un dispositivo de conexión, en este caso un módem.

La comunicación entre los nodos se rige por los protocolos HTTP para la asociación entre la computadora cliente y el servidor de aplicaciones y TCP/IP para la conexión entre los dos servidores. El servidor de aplicaciones y el módem se conectan a través del puerto USB o SERIAL.

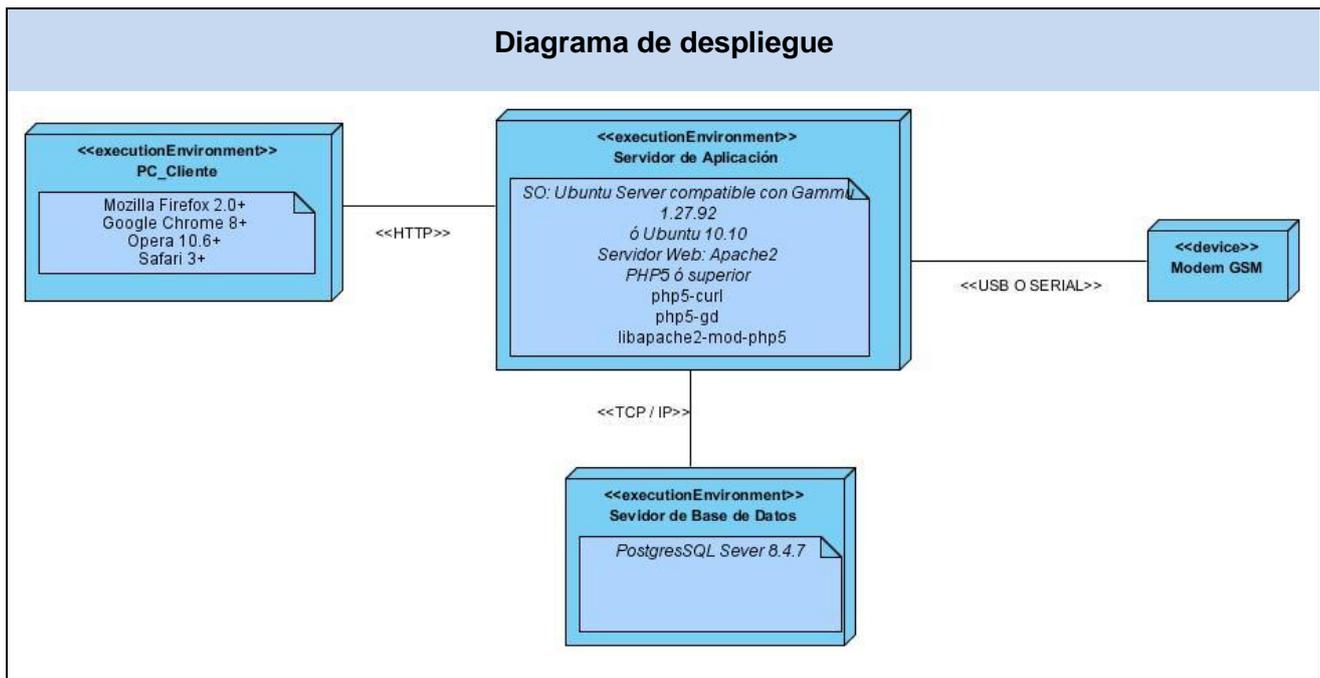


Figura 4.1 Diagrama de despliegue

4.4. Pruebas

El proceso de pruebas es fundamental en el ciclo de vida de un software. En este los desarrolladores comprueban constantemente y tanto como sea posible el código, con el objetivo

de reducir al mínimo el número de errores no detectados. Lo que contribuye a elevar la calidad de los productos desarrollados.

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. (Crispin, y otros, 2002)

4.4.1. Pruebas Unitarias

Las pruebas unitarias son una actividad fundamental en la metodología XP, pues asegura que un determinado módulo cumpla con un comportamiento esperado en forma aislada antes de ser integrado al sistema. Estas pruebas brindan al programador una inmediata retroalimentación de cómo está realizando su trabajo, permitiéndole realizar cambios de forma segura respaldado por efectivos casos de prueba.

Las pruebas unitarias fueron empleadas durante todo el proceso de desarrollo del producto, en cada una de las funcionalidades implementadas, probando el código con casos de prueba creados por el equipo de desarrollo y verificando si los resultados brindados eran los esperados para cada funcionalidad.

4.4.2. Pruebas de Aceptación

Las pruebas de aceptación son pruebas de caja negra que se crean a partir de las historias de usuario. Su principal objetivo es validar que el sistema cumple con el funcionamiento esperado y permitir al cliente determinar su aceptación, desde el punto de vista de su funcionalidad y rendimiento. Una historia de usuario no se considera lista hasta que haya pasado todas sus pruebas de aceptación.

Las pruebas de aceptación correspondiente a cada una de las funcionalidades serán representadas mediante tablas divididas por las secciones siguientes:

- Código de la prueba de aceptación.
- Número de la historia de usuario a la que se le realiza la prueba.
- Nombre de la funcionalidad.
- Descripción de la funcionalidad.
- Condiciones de ejecución de la funcionalidad.

- Entrada y pasos de ejecución que realiza el usuario con el objetivo de obtener el resultado esperado.
- Resultado esperado.
- Evaluación de la prueba.

Caso de prueba de aceptación	
Código: HU1-P1	No. Historia de Usuario: 1
Nombre: Enviar mensaje de respuesta hacia entidad externa.	
Descripción: Prueba para la funcionalidad Enviar mensaje de respuesta hacia entidad externa.	
Condiciones de ejecución:	
<ul style="list-style-type: none"> • La llegada de un nuevo mensaje a la bandeja de entrada del Gammu. 	
Entrada/Pasos de ejecución: No existen datos de entrada para esta prueba.	
<p>El sistema obtiene el IP que realiza la petición. Verifica que ese IP corresponda con el IP del servidor. Luego busca el último usuario de tipo Gateway que envió un mensaje al número de teléfono que está respondiendo; se verifica que ese usuario tenga habilitada la opción de recibir mensajes de respuesta y finalmente se le envía el mensaje y el número de teléfono que está respondiendo por el método GET o POST (según la configuración del usuario) hacia la URL que haya definido previamente ese usuario en su configuración.</p>	
Resultado esperado: El mensaje de respuesta es entregado sin errores.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 4.8 HU1-P1: Enviar mensaje de respuesta hacia entidad externa

Caso de prueba de aceptación	
Código: HU1-P2	No. Historia de Usuario: 1
Nombre: Envío fallido de mensaje de respuesta hacia entidad externa.	

Descripción: Prueba para la funcionalidad Enviar mensaje de respuesta hacia entidad externa.
Condiciones de ejecución: <ul style="list-style-type: none"> La llegada de un nuevo mensaje a la bandeja de entrada del Gammu.
Entrada/Pasos de ejecución: No existen datos de entrada para esta prueba. El sistema obtiene el IP que realiza la petición. Verifica que ese IP corresponda con el IP del servidor. En caso que el IP que está realizando la petición no coincida con el IP del servidor, se procede a guardar en el Log de la aplicación la fecha, la hora y el IP que intentó acceder a la URL y se le muestra una página de error. Luego el sistema busca el último usuario de tipo Gateway que envió un mensaje al número de teléfono que está respondiendo; se verifica que ese usuario tenga habilitada la opción de recibir mensajes de respuesta, si este usuario no desea recibir mensajes de respuesta el mensaje es ignorado. Si no existe ningún usuario Gateway que haya enviado un mensaje al número de teléfono que responde, se guarda en el Log de la aplicación la fecha, la hora y el número de teléfono que envió el mensaje.
Resultado esperado: El mensaje de respuesta es ignorado.
Evaluación de la prueba: Prueba satisfactoria.

Tabla 4.9 HU1-P2: Envío fallido de mensaje de respuesta hacia entidad externa

Caso de prueba de aceptación	
Código: HU2-P1	No. Historia de Usuario: 2
Nombre: Enviar notificación de posible error en el servidor.	
Descripción: Prueba para la funcionalidad Enviar notificación de posible error en el servidor.	
Condiciones de ejecución: <ul style="list-style-type: none"> El registro de algún error en el fichero de log de Gammu. 	

<p>Entrada/Pasos de ejecución: No existen datos de entrada para esta prueba.</p> <p>El sistema obtiene el IP que realiza la petición. Verifica que ese IP corresponda con el IP del servidor.</p> <p>Se busca en el fichero de log de Gammu, el último mensaje de error perteneciente al día en curso. Si se ha reportado más de un error, se verifica que hayan transcurrido más de 5 minutos entre el anterior error del log y éste, para evitar saturar el sistema con varios errores en un corto período de tiempo. Se verifica si la cantidad de mensajes de error en la bandeja de salida del Gammu no es mayor que 5, en caso positivo se procede a buscar los administradores del sistema que desean recibir este tipo de notificación en sus teléfonos. Luego se añade el mensaje de error a la bandeja de salida del Gammu. En el caso de que hayan 5 o más mensajes en la bandeja de salida del Gammu, se elimina el último mensaje de error de la base de datos y se sustituye por éste; con el objetivo de no insertar numeroso mensajes de error en el sistema.</p>
<p>Resultado esperado: El mensaje de posible error es enviado satisfactoriamente.</p>
<p>Evaluación de la prueba: Prueba satisfactoria.</p>

Tabla 4.10 HU2-P1: Enviar notificación de posible error en el servidor

Caso de prueba de aceptación	
Código: HU2-P2	No. Historia de Usuario: 2
Nombre: Envío fallido de notificación de posible error en el servidor.	
Descripción: Prueba para la funcionalidad Enviar notificación de posible error en el servidor.	
<p>Condiciones de ejecución:</p> <ul style="list-style-type: none"> El registro de algún error en el fichero de log de Gammu. 	
<p>Entrada/Pasos de ejecución: No existen datos de entrada para esta prueba.</p> <p>El sistema obtiene el IP que realiza la petición. Verifica que ese IP corresponda con el IP del servidor. En caso que el IP que está realizando la petición no coincida con el IP del servidor, se procede a guardar en el Log de la aplicación la fecha, la hora y el IP que intentó acceder a la URL</p>	

<p>y se le muestra una página de error.</p> <p>Se busca en el fichero de log de Gammu, el último mensaje de error perteneciente al día en curso. Si se ha reportado más de un error, se verifica que hayan transcurrido más de 5 minutos entre el anterior error del log y éste, para evitar saturar el sistema con varios errores en un corto período de tiempo. En caso negativo, entonces se descarta este error.</p>
<p>Resultado esperado: El mensaje de posible error no es enviado.</p>
<p>Evaluación de la prueba: Prueba satisfactoria.</p>

Tabla 4.11 HU2-P2: Envío fallido de notificación de posible error en el servidor

Caso de prueba de aceptación	
Código: HU3-P1	No. Historia de Usuario: 3
Nombre: Reiniciar aplicación Gammu.	
Descripción: Prueba para la funcionalidad Reiniciar aplicación Gammu.	
<p>Condiciones de ejecución:</p> <ul style="list-style-type: none"> La llegada de un nuevo mensaje a la bandeja de entrada del Gammu con el formato: @reiniciar@ @user:miusuario @pass:mipassword. 	
<p>Entrada/Pasos de ejecución: No existen datos de entrada para esta prueba.</p> <p>El sistema obtiene el IP que realiza la petición. Verifica que ese IP corresponda con el IP del servidor.</p> <p>El sistema comprueba que el formato del mensaje corresponda con el especificado. Luego se verifica que el usuario brindado tenga permisos de administración, luego se verifica que la contraseña brindada corresponde con la del usuario en cuestión y por último se verifica que el número de teléfono del que proviene este mensaje corresponda con el número de teléfono de ese administrador. Esto permitirá que cada administrador solo pueda reiniciar el Gammu si envía el mensaje desde su propio teléfono. Finalmente se procede a reiniciar el Gammu Smsd.</p>	

Resultado esperado: Se reinicia la aplicación Gammu.
Evaluación de la prueba: Prueba satisfactoria.

Tabla 4.12 HU3-P1: Reiniciar aplicación Gammu

Caso de prueba de aceptación	
Código: HU3-P2	No. Historia de Usuario: 3
Nombre: Reinicio fallido de la aplicación Gammu.	
Descripción: No se reinicia la aplicación Gammu.	
Condiciones de ejecución: <ul style="list-style-type: none"> • La llegada de un nuevo mensaje a la bandeja de entrada del Gammu. 	
Entrada/Pasos de ejecución: No existen datos de entrada para esta prueba. El sistema obtiene el IP que realiza la petición. Verifica que ese IP corresponda con el IP del servidor. En caso que el IP que está realizando la petición no coincida con el IP del servidor, se procede a guardar en el Log de la aplicación la fecha, la hora y el IP que intentó acceder a la URL y se le muestra una página de error. El sistema comprueba el formato del mensaje. Si el formato no es el correcto no se reinicia el Gammu SMSD. Si el formato es correcto pero el usuario brindado no tiene permisos de administración, no se reinicia el Gammu SMSD. Si el usuario tiene privilegios administrativos se verifica que la contraseña brindada es la correcta, en caso contrario no se reinicia el Gammu SMSD. Por último se verifica si el número de teléfono del que proviene este mensaje corresponde con el número de teléfono de ese administrador, en caso contrario no se reinicia el Gammu SMSD.	
Resultado esperado: No se reinicia la aplicación Gammu.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 4.13 HU3-P2: Reinicio fallido de la aplicación Gammu

Caso de prueba de aceptación	
Código: HU4-P1	No. Historia de Usuario: 4
Nombre: Mostrar estadísticas generales de los mensajes.	
Descripción: La aplicación deberá brindar la posibilidad de mostrar las estadísticas de los mensajes en forma de gráficas.	
Condiciones de ejecución: <ul style="list-style-type: none"> • El usuario deberá estar autenticado en el sistema con el rol de Administrador. El usuario deberá seleccionar en el Menú la opción: Mostrar Estadísticas. 	
Entrada/Pasos de ejecución: No existen datos de entrada para esta prueba. El sistema busca la cantidad de mensajes enviados, recibidos y en cola.	
Resultado esperado: Se muestran las gráficas correspondientes a los mensajes enviados, recibidos y en cola.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 4.14 HU4-P1: Mostrar estadísticas generales de los mensajes

Caso de prueba de aceptación	
Código: HU5-P1	No. Historia de Usuario: 5
Nombre: Exportar reportes a formato PDF.	
Descripción: La aplicación deberá brindar la posibilidad de crear un reporte en formato PDF.	
Condiciones de ejecución: <ul style="list-style-type: none"> • El usuario deberá estar autenticado en el sistema con el rol de administrador. • El usuario deberá seleccionar la opción del menú: Crear Reporte de Estadísticas. 	

<p>Entrada/Pasos de ejecución: No existen datos de entrada para esta prueba.</p> <p>El sistema busca la cantidad de mensajes enviados, recibidos y en cola y genera un reporte en formato PDF con esa información.</p>
<p>Resultado esperado: Se crea el PDF con todas las estadísticas de los mensajes.</p>
<p>Evaluación de la prueba: Prueba satisfactoria.</p>

Tabla 4.15 HU5-P1: Exportar reportes a formato PDF

Caso de prueba de aceptación	
Código: HU6-P1	No. Historia de Usuario: 6
Nombre: Gestionar Saldo de Alerta en los módems.	
Descripción: La aplicación deberá brindar la posibilidad de modificar el saldo de alerta en cada uno de los módems conectados al sistema.	
<p>Condiciones de ejecución:</p> <ul style="list-style-type: none"> El usuario deberá estar registrado en el sistema con el rol de Administrador. El usuario deberá seleccionar en el menú la opción: Modificar Saldo de Alerta. 	
Entrada/Pasos de ejecución: El usuario accede a la interfaz inicial, escoge la opción de Modificar Saldo de Alerta e introduce el nuevo saldo correspondiente.	
Resultado esperado: Se modifica el saldo de alerta en el módem seleccionado.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 4.16 HU6-P1: Gestionar Saldo de Alerta en los módems

Caso de prueba de aceptación	
Código: HU6-P2	No. Historia de Usuario: 6

Nombre: Modificación fallida del saldo de Alerta en los módems.
Descripción: No se modifica el saldo de la alerta.
Condiciones de ejecución: <ul style="list-style-type: none"> El usuario deberá estar registrado en el sistema con el rol de Administrador. El usuario deberá seleccionar la opción Modificar Saldo de Alerta.
Entrada/Pasos de ejecución: El usuario accede a la interfaz inicial, escoge la opción de Modificar Saldo de Alerta e introduce el nuevo saldo correspondiente. Se comprueba que el dato brindado sea un número.
Resultado esperado: No se modifica el saldo de alerta en el módem seleccionado.
Evaluación de la prueba: Prueba satisfactoria.

Tabla 4.17 HU6-P2: Modificación fallida del saldo de Alerta en los módems

Caso de prueba de aceptación	
Código: HU7-P1	No. Historia de Usuario: 7
Nombre: Gestionar Saldo de módem.	
Descripción: La aplicación deberá brindar la posibilidad de modificar el saldo de cada uno de los módems conectados al sistema.	
Condiciones de ejecución: <ul style="list-style-type: none"> El usuario deberá estar registrado en el sistema con el rol de Administrador. El usuario deberá seleccionar la opción Modificar Saldo de Módem. 	
Entrada/Pasos de ejecución: El usuario accede a la interfaz inicial, escoge la opción de Modificar Saldo de Módem e introduce el nuevo saldo correspondiente.	
Resultado esperado: Se modifica el saldo del módem seleccionado.	

Evaluación de la prueba: Prueba satisfactoria.

Tabla 4.18 HU7-P1: Gestionar Saldo de módem

Caso de prueba de aceptación	
Código: HU7-P2	No. Historia de Usuario: 7
Nombre: Modificación fallida del saldo del módem.	
Descripción: No se modifica el saldo del módem.	
Condiciones de ejecución:	
<ul style="list-style-type: none"> El usuario deberá estar registrado en el sistema con el rol de Administrador. El usuario deberá seleccionar la opción Modificar Saldo de módem. 	
Entrada/Pasos de ejecución: El usuario accede a la interfaz inicial, escoge la opción de Modificar Saldo de módem e introduce el nuevo saldo correspondiente. Se comprueba que el dato brindado sea un número mayor que cero.	
Resultado esperado: No se modifica el saldo del módem seleccionado.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 4.19 HU7-P2: Modificación fallida del saldo del módem

Caso de prueba de aceptación	
Código: HU8-P1	No. Historia de Usuario: 8
Nombre: Gestionar precio del mensaje.	
Descripción: La aplicación deberá brindar la posibilidad de modificar el precio de los mensajes que se envían.	
Condiciones de ejecución:	

<ul style="list-style-type: none"> • El usuario deberá estar registrado en el sistema con el rol de Administrador. • El usuario deberá seleccionar la opción Modificar Precio del Mensaje.
Entrada/Pasos de ejecución: El usuario accede a la interfaz inicial, escoge la opción de Modificar Precio del Mensaje e introduce el nuevo precio correspondiente.
Resultado esperado: Se modifica el precio del mensaje.
Evaluación de la prueba: Prueba satisfactoria.

Tabla 4.20 HU8-P1: Gestionar precio del mensaje

Caso de prueba de aceptación	
Código: HU8-P2	No. Historia de Usuario: 8
Nombre: Modificación fallida del precio del mensaje.	
Descripción: No se modifica el precio del mensaje.	
Condiciones de ejecución: <ul style="list-style-type: none"> • El usuario deberá estar registrado en el sistema con el rol de Administrador. • El usuario deberá seleccionar la opción Modificar precio del mensaje. 	
Entrada/Pasos de ejecución: El usuario accede a la interfaz inicial, escoge la opción de Modificar Precio del Mensaje e introduce el nuevo precio correspondiente. Se comprueba que el dato brindado sea un número mayor que cero.	
Resultado esperado: No se modifica el precio del mensaje.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 4.21 HU8-P2: Modificación fallida del precio del mensaje

Luego de realizarse las pruebas de aceptación se obtienen un total de 6 no conformidades donde 6 fueron resueltas, 0 no procedían y quedaron 0 pendientes por resolver, ilustrando estos resultados en la siguiente gráfica:



Figura 4.2 Estadísticas de las pruebas

4.5. Conclusiones

En este capítulo se implementaron las funcionalidades de acuerdo a las historias de usuario anteriormente definidas, además se realizaron las pruebas de aceptación al producto final y se ilustraron los resultados de dichas pruebas.

CONCLUSIONES GENERALES

- El estudio realizado sobre las Plataformas de Mensajería existentes a nivel internacional y nacional, evidenció que existe una gran variedad en cuanto a los servicios que brindan estas plataformas, en dependencia de los objetivos para los que fueron creadas. Ninguna de las pasarelas estudiadas funcionan a través de módems GSM, sino que utilizan una conexión directa con las operadoras.
- Para el desarrollo de las funcionalidades de la versión 2.0 de la Plataforma de Mensajería, se asimiló la arquitectura propuesta en la versión anterior. El diseño propuesto y las tecnologías empleadas se basaron en dicha arquitectura.
- El sistema propuesto cuenta con un mecanismo de notificación mediante el envío de correos electrónicos y SMS a los administradores, garantizando la retroalimentación continua, así como un mejor mantenimiento y soporte a la aplicación.
- El uso de la metodología XP permitió crear un software que cumple con las funcionalidades requeridas, en un corto período de tiempo, con la calidad y la documentación necesarias para posteriores investigaciones.
- La implementación de la Versión 2.0 de la Plataforma de Mensajería SMS utilizando módems GSM, resuelve las limitantes presentadas en la versión anterior y pone en manos del centro de Telemática un sistema que permite mejorar el proceso de envío masivo de mensajes de texto, con nuevas funcionalidades que hacen de éste un producto más usable y acabado.

RECOMENDACIONES

- Incrementar 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 un mal uso del servicio web como peticiones infinitas que afecten la estabilidad del servicio web y del sistema en general.
- Aumentar las capacidades del sistema para evitar la sobrecarga de alguno de los módems conectados, implementando un servicio de balanceo de carga entre los módems.

REFERENCIAS BLIBLIOGRÁFICAS

Altiria. Altiria. [En línea] [Citado el: 15 de 11 de 2011.]
http://www.altiria.com/web/sms_mms/pasarela_envio_sms.

APACHE, HTTP SERVER PROJECT. APACHE, HTTP SERVER PROJECT. [En línea] [Citado el: 17 de 11 de 2011.] http://httpd.apache.org/ABOUT_APACHE.html.

Beck, Kent y Flower, Martin. 2000. *Planning Extreme Programming*. s.l.: Addison Wesley, 2000.

BlueEye. Blue Eye. [En línea] [Citado el: 16 de 1 de 2012.] <http://entumovil.cu/>.

Carmen Penadés, María, H. Canós, José y Letelier, Patricio. 2004. *Métodologías Ágiles en el Desarrollo de Software*. Universidad Politécnica de Valencia: s.n., 2004.

Codelgniter. 2011. *Codelgniter Guía del Usuario en Español*. 2011.

Crispin, Lisa y House, Tip. 2002. *Testing Extreme Programming*. s.l.: Addison Wesley, 2002.

de la Torre, César, Zorrilla, Unai y Ramos, Miguel Ángel . 2011. *Guía de Arquitectura N-Capas DDD .NET 4.0*. s.l.: Krassis Press, 2011.

DesarrolloWeb. Desarrollo Web. [En línea] [Citado el: 23 de 11 de 2011.]
<http://www.desarrolloweb.com/articulos/392.php>.

Doctrine. Doctrine. [En línea] [Citado el: 11 de 11 de 2011.] <http://docs.doctrine-project.org/projects/doctrine-orm/en/latest/reference/introduction.html>.

Ecured. Ecured. [En línea] [Citado el: 24 de 1 de 2012.]
http://www.ecured.cu/index.php/Visual_Paradigm.

Eguíluz Pérez, Javier. 2008. *Introducción a JavaScript*. 2008.

Fernández Jo, Zussel y Rodríguez Suarez, Santiago. 2011. *Sistema de Administración de la Plataforma de Servicios de Mensajería SMS PLASERV*. Habana: s.n., 2011. pág. 85.

Framework, Zend. Zend Framework. [En línea] [Citado el: 18 de 11 de 2011.]
<http://spanish.zendfw.com/que-es-zend-framework/>.

Gammu. GWammu. [En línea] [Citado el: 13 de 11 de 2011.] <http://www.wammu.eu/gammu/>.

Hernández Ricardo, José Alexis. 2011. *Sistema de servicio de SMS basado en Gammu (JAMS)*. Habana: s.n., 2011. pág. 80.

- NetBeans.** NetBeans. [En línea] [Citado el: 25 de 10 de 2011.] http://netbeans.org/index_es.html.
- NOWSMS.** NOWSMS. [En línea] [Citado el: 1 de 11 de 2011.] <http://www.nowsms.com>.
- PostgreSQL.** PostgreSQL. [En línea] [Citado el: 30 de 11 de 2011.] http://www.postgresql.org.es/sobre_postgresql/.
- Quispe Carita, Vilma, Huamantuco Solorzano, Dante Harry y Vargas Yupanqui, José Luis. 2011.** *METODOLOGIA RUP (RATIONAL UNIFIED PROCESS)*. Perú: s.n., 2011.
- Rivera Rivera, Leyla Dinora, Colato Rodríguez, Hugo Miguel y Antonio Tesorero, Nelson. 2008.** *Sistema WDS para la Administración remota de servidores*. El Salvador: s.n., 2008.
- Rumbaugh, James, Booch, Grady y Jacobson, Ivar. 2000.** *El Proceso Unificado de Desarrollo de Software*. s.l.: Pearson Addisson-Wesley, 2000.
- SMS, Activa Global.** Activa Global SMS. [En línea] [Citado el: 11 de 6 de 2011.] <http://www.activasms.com/>.
- W3C.** W3C. [En línea] [Citado el: 16 de 11 de 2011.] <http://www.w3c.es/>.

BIBLIOGRAFÍA

1. **Beck, Kent y Flower, Martin. 2000.** Planning Extreme Programming. s.l.: Addison Wesley, 2000.
2. **Carmen Penadés, María, H. Canós, José y Letelier, Patricio. 2004.** Metodologías Ágiles en el Desarrollo de Software. Universidad Politécnica de Valencia: s.n., 2004.
3. **Carrillo Pérez, Isaías, Pérez González, Rodrigo y Rodríguez Martin, Aureliano David.** Metodologías de Desarrollo de Software. 2008.
4. **Crispin, Lisa y House, Tip. 2002.** Testing Extreme Programming. s.l.: Addison Wesley, 2002.
5. **Eguíluz Pérez, Javier. 2008.** Introducción a JavaScript. 2008.
6. **Fernández Jo, Zussel y Rodríguez Suarez, Santiago. 2011.** Sistema de Administración de la Plataforma de Servicios de Mensajería SMS PLASERV. Habana: s.n., 2011. pág. 85.
7. **Hernández León, Rolando Alfredo y Coello González, Sayda. 2002.** EL paradigma cuantitativo de la Investigación Científica. La Habana : EDUNIV: Editorial Universitaria, 2002. 959-16-0343-6.
8. **Hernández Ricardo, José Alexis. 2011.** Sistema de servicio de SMS basado en Gammu (JAMS). Habana: s.n., 2011. pág. 80.
9. **Izaguirre Remón, Rafael Claudio y Concepción Obregón, Dr. Tebelio.** El problema científico y su papel en la arquitectura del proceso de investigación. pág. 12.
10. **Larman, Craig.** UML y patrones. Introducción al análisis y diseño orientado a objetos. 2004.
11. **Quispe Carita, Vilma, Huamantuco Solorzano, Dante Harry y Vargas Yupanqui, José Luis. 2011.** Metodología RUP (Rational Unified Process). Perú: s.n., 2011.

ANEXOS

Anexo I

Historia de Usuario																																								
Número: 4	Nombre de Historia de Usuario: Mostrar estadísticas generales de los mensajes.																																							
Usuario: Lisandra Soler Ponce Eduardo N. Rosales	Iteración Asignada: 2																																							
Prioridad en negocio: Medio	Puntos estimados: 2																																							
Riesgo en Desarrollo: Alto	Puntos Reales: 2																																							
<p>Descripción: El sistema mostrará las estadísticas en forma de gráficas de la cantidad de mensajes enviados y recibidos en el último año (detallado por meses) y la cantidad de mensajes enviados y recibidos en las últimas 24 horas (detallados por hora).</p>																																								
<p>Observaciones:</p>																																								
<p>Prototipo de interfaz:</p>  <p>The graph displays the number of messages sent and received over a 12-month period. The Y-axis represents the number of messages (0 to 20), and the X-axis represents the months (1 to 12). The 'Recibidos' (Received) series is shown in blue with circular markers, and the 'Enviados' (Sent) series is shown in red with diamond markers. A callout box points to the data for month 12, indicating 14 received messages.</p> <table border="1"> <caption>Mensajes en los últimos 12 meses</caption> <thead> <tr> <th>Mes</th> <th>Recibidos</th> <th>Enviados</th> </tr> </thead> <tbody> <tr><td>1</td><td>5</td><td>8</td></tr> <tr><td>2</td><td>0</td><td>13</td></tr> <tr><td>3</td><td>0</td><td>0</td></tr> <tr><td>4</td><td>0</td><td>0</td></tr> <tr><td>5</td><td>0</td><td>0</td></tr> <tr><td>6</td><td>0</td><td>0</td></tr> <tr><td>7</td><td>0</td><td>0</td></tr> <tr><td>8</td><td>0</td><td>0</td></tr> <tr><td>9</td><td>0</td><td>0</td></tr> <tr><td>10</td><td>1</td><td>0</td></tr> <tr><td>11</td><td>1</td><td>0</td></tr> <tr><td>12</td><td>14</td><td>8</td></tr> </tbody> </table>		Mes	Recibidos	Enviados	1	5	8	2	0	13	3	0	0	4	0	0	5	0	0	6	0	0	7	0	0	8	0	0	9	0	0	10	1	0	11	1	0	12	14	8
Mes	Recibidos	Enviados																																						
1	5	8																																						
2	0	13																																						
3	0	0																																						
4	0	0																																						
5	0	0																																						
6	0	0																																						
7	0	0																																						
8	0	0																																						
9	0	0																																						
10	1	0																																						
11	1	0																																						
12	14	8																																						

Tabla A1- HU4: Mostrar estadísticas generales de los mensajes

Historia de Usuario	
Número: 9	Nombre de Historia de Usuario: Enviar Notificación de Recarga de módem.

Usuario: Lisandra Soler Ponce Eduardo N. Rosales	Iteración Asignada: 2
Prioridad en negocio: Medio	Puntos estimados: 1
Riesgo en Desarrollo: Alto	Puntos Reales: 1
Descripción: El sistema enviará una notificación mediante correo electrónico a los administradores del sistema cuando sea necesaria la recarga de algunos de los módems conectados.	
Observaciones: <ul style="list-style-type: none"> Es necesario recargar un módem cuando su saldo es menor o igual que el saldo mínimo para generar una alerta. 	
Prototipo de interfaz:	

Tabla A6 - HU9: Enviar Notificación de Recarga de módem

Anexo II

Clase PhoneC	
Responsabilidad	Colaborador
Listar módems conectados.	Phones Message
Actualizar saldo del módem.	
Actualizar el saldo mínimo requerido para generar alerta de recarga.	
Actualizar el precio de los mensajes.	

Tabla A7: Clase PhoneC

Clase Statistics	
Responsabilidad	Colaborador
Obtener mensajes en las últimas 24 horas.	

Obtener mensajes en los últimos 15 días.	SentItems Inbox Phones OutBox
Obtener mensajes en el último mes.	
Obtener mensajes en el último año.	
Obtener mensajes enviados, recibidos y en cola históricamente.	
Crear PDF.	

Tabla A10: Clase Statistics

Clase WebService	
Responsabilidad	Colaborador
Brindar el servicio web para permitir el envío de mensajes SMS.	OutBox

Tabla A11: Clase WebService

Anexo III

Tareas de ingeniería	
No. Tarea: 4	No. Historia de Usuario: 4
Nombre de la Tarea: Mostrar estadísticas generales de los mensajes.	
Tipo de Tarea: Desarrollo	Puntos de estimación: 2/2
Fecha de Inicio: 12/03/2012	Fecha de Fin: 24/03/2012
Programador Responsable: Lisandra Soler Ponce Eduardo Noel Rosales	
Descripción: Se implementarán las funcionalidades necesarias para mostrar gráficamente en el sistema las estadísticas relacionadas con la cantidad de mensajes enviados y recibidos en el último año (detallado por meses) y en las últimas 24 horas (detallado por horas).	

Tabla A12 - Tarea #4 de la HU: Mostrar estadísticas generales de los mensajes

Tareas de ingeniería	
No. Tarea: 7	No. Historia de Usuario: 7
Nombre de la Tarea: Gestionar saldo de módem.	
Tipo de Tarea: Desarrollo	Puntos de estimación: 2/2
Fecha de Inicio: 25/04/2012	Fecha de Fin: 09/05/2012
Programador Responsable: Lisandra Soler Ponce Eduardo Noel Rosales	
Descripción: El sistema ofrecerá al administrador una interfaz con la lista de módems disponibles, así como el saldo disponible en cada uno de ellos, y brindará la opción de recargar ese saldo. Luego se muestra otra interfaz requiriendo al usuario que introduzca el saldo con que recargará el módem. En caso de existir datos incorrectos, se mostrará un mensaje con el error producido.	

Tabla A14 - Tarea #7 de la HU: Gestionar saldo de módem

GLOSARIO DE TÉRMINOS

AJAX: Asynchronous JavaScript And XML o JavaScript asincrónico y XML. Es una técnica de desarrollo web para crear aplicaciones interactivas. Estas aplicaciones se ejecutan en el cliente, es decir, en el navegador de los usuarios, mientras se mantiene la comunicación asíncrona con el servidor en segundo plano. De esta forma es posible realizar cambios sobre las páginas sin necesidad de recargarlas, lo que aumenta la interactividad, velocidad y usabilidad de las aplicaciones.

API: Application Programming Interface o Interfaz de Programación de Aplicaciones. Es un conjunto de convenciones internacionales que definen cómo debe invocarse una determinada función de un programa desde una aplicación.

BPEL: Business Process Execution Language o Lenguaje de Ejecución de Procesos de Negocio. Es un lenguaje estandarizado para la composición de servicios web.

BSD: Es la licencia de software otorgada principalmente para los sistemas BSD (Berkeley Software Distribution). Es una licencia libre permisiva y al contrario de la licencia GPL permite el uso del código fuente en software no libre.

CAPTCHA: Completely Automated Public Turing test to tell Computers and Humans Apart, en español "Prueba de Turing pública y automática para diferenciar a máquinas y humanos". Es una prueba utilizada en la computación para determinar si el que interactúa con el sistema es o no humano.

A grandes rasgos la prueba consiste en la inserción de un código generado en una imagen distorsionada para su comprobación, teniendo en cuenta la dificultad de una máquina para la comprensión de la secuencia de forma correcta, no siendo así generalmente para los humanos.

DOM: Document Object Model o Modelo de Objeto del Documento. Es una plataforma que proporciona un conjunto estándar de objetos a los programas y scripts para acceder y actualizar dinámicamente el contenido, estructura y estilo de los documentos. El documento puede ser procesado y los resultados del procesamiento se pueden incorporar de nuevo a la página presentada.

GOF: Gang of Four o Banda de los Cuatro. Es el nombre con el que se conoce comúnmente a los autores del libro Design Patterns o Diseño de Patrones.

GNU: General Public License o Licencia Pública General es una licencia que está orientada principalmente a proteger la libre distribución, modificación y uso de software. Su propósito es declarar

que el software cubierto por esta licencia es software libre y protegerlo de intentos de apropiación que restrinjan esas libertades a los usuarios.

GRASP: General Responsibility Assignment Software Patterns o Patrones de Principios Generales para Asignar Responsabilidades. Son una serie de buenas prácticas de aplicación recomendable en el diseño de software.

GSM: Global System for Mobile Communications o Sistema Global para Comunicaciones Móviles. Es un sistema estándar definido para la comunicación mediante teléfonos móviles que incorporan tecnología celular.

HTTP: Hypertext Transfer Protocol o Protocolo de Transferencia de Hipertexto. Es un protocolo que define la sintaxis y la semántica que utilizan los elementos de software de la arquitectura web (clientes-servidores) para comunicarse. Es un protocolo orientado a transacciones y sigue el esquema petición-respuesta entre un cliente y un servidor.

IDE: Integrated Development Environment o 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.

LDAP: Lightweight Directory Access Protocol o Protocolo Ligero de Acceso a Directorios. Protocolo a nivel de aplicación que permite el acceso a un servicio de directorio ordenado y distribuido para buscar diversa información en un entorno de red.

MIT: Massachusetts Institute of Technology o Instituto de Tecnología de Massachusetts. Institución universitaria técnica con sede en Cambridge, Massachusetts (EEUU).

ORM: Object-Relational Mapping o Mapeo Relacional de Objetos. Es una técnica de programación para convertir datos entre el sistema de tipos utilizado en un lenguaje de programación orientado a objetos y el utilizado en una base de datos relacional, utilizando un motor de persistencia.

PDF: Portable Document Format o Formato de Documento Portátil. Es el formato de archivos desarrollado por Adobe Systems. Esta tecnología ha tenido éxito estandarizando el formato de los documentos que se utilizan y transfieren en la red.

PDO: PHP Data Object. Es una extensión que provee una capa de abstracción de acceso a datos para PHP 5, con lo cual se consigue hacer uso de las mismas funciones para hacer consultas y obtener datos de distintos manejadores de bases de datos.

SMTP: Simple Mail Transfer Protocol o Protocolo Simple de Transferencia de Correo. Protocolo de red basado en texto utilizado para el intercambio de mensajes en correos electrónicos entre computadoras o distintos dispositivos.

SOAP: Simple Object Access Protocol o Protocolo de Acceso a Objeto Simple. Es un protocolo para intercambiar mensajes, basado en XML, de un amplio uso en servicios web.

SSL: Secure Sockets Layer o Capa de Conexión Segura. Es un protocolo criptográfico que proporciona comunicaciones seguras por una red.

XML: Extensible Markup Language o 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.