

Universidad de las Ciencias Informáticas

Facultad 2



Título: Plataforma de mensajería paging.

Trabajo de Diploma para optar por el título de

Ingeniero Informático

Autor(es): Yanileidis Luna Solis

Otoniel Berroa Arias

Tutor(es): Ing. David Rodríguez*, Ing. Denys Buedo**

Co-tutor: Ing. Dailenis Zamora Zorrilla***

La Habana

2012



Pensamiento

“El futuro de nuestra patria tiene que ser necesariamente un futuro de hombres de ciencia, tiene que ser un futuro de hombres de pensamiento, porque precisamente es lo que más estamos sembrando; lo que más estamos sembrando son oportunidades a la inteligencia (...)”

Fidel Castro

DECLARACIÓN DE AUTORÍA

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

Para que así conste firmo la presente a los ____ días del mes de _____ del año _____.

Autores

Otoniel Berroa Arias

Yanileidis Luna Solis

Tutores

David Rodríguez

Denys Buedo

Dailenis Zamora

DATOS DE CONTACTO

Síntesis del Tutor:

- * Graduado de Ingeniero en Ciencias Informáticas en el 2008 en la Universidad de las Ciencias Informática. Jefe Departamento Docente de Telecomunicaciones de la facultad 2, con 4 años de experiencias.
- ** Graduado de Ingeniero en Ciencias Informáticas en el 2007 en la Universidad de las Ciencias Informática. Director del centro Telemática, con 5 años de experiencias.
- *** Graduado de Ingeniero en Ciencias Informáticas en el 2011 en la Universidad de las Ciencias Informática. Recién Graduado en Adiestramiento de la facultad 2, con 1 año de experiencia.



Agradecimientos

AGRADECIMIENTOS COMUNES

*A nuestros tutores David, Daílenis y Denys que nos apoyaron siempre, por brindarnos su amistad, su apoyo y su entrega, por nunca haber dicho **No** cuando lo necesitamos.*

A todos los profesores que han influido en nuestra formación y a los que nos ayudaron a luchar, a estudiar más, a prepararnos, a tomar decisiones, a enfrentarnos a los retos y a sobreponernos ante las dificultades.

A la profesora Bárbara y al oponente Adonys por el buen trabajo y deseos de ayudar, siempre oportunos con sus señalamientos.

Al profesor Rene por toda la ayuda brindada.

A la Universidad por formarnos como profesionales.

Luna y Oto



Agradecimientos

AGRADECIMIENTOS

Luna

A mi mamá Norma por quererme, apoyarme, comprenderme y malcriarme tanto, por ser la luz de mi vida, si hoy me he convertido en la persona que soy es gracias a su entrega total como madre.

A mi papito Elíaxi por su apoyo, su cariño, por confiar siempre en mí y mostrarme con su ejemplo como ser una mejor persona cada día.

A mis hermanas Lai, Yane, Yarita y Yordalís por estar siempre atentas y cuidar de su hermanita más chiquita, por ser el mejor regalo que me ha dado la vida.

A mi tía Cristina por ser la mano derecha de mi mamá apoyándome, cuidándome, por su cariño y todo su atención.

A mi familia que siempre está ahí cuando la necesito.

A mi novio Bonne por su amor, su comprensión, por soportar mi malcriadez, por cuidar de mí y ser parte de mi vida.

A mis amigos Yaile, Yuya, Tata y Nelsito por estar conmigo en las buenas y en las malas, por aguantarme durante todo este tiempo.

A mi compañero de tesis, que aunque es bastante jodedor, se esforzó mucho para la realización de este trabajo, y ha sido un buen amigo.

A todas las buenas amistades que fueron partícipes de mi vida.



Agradecimientos

Otoniel

A todos los que de un modo u otro han ayudado a ser de mí una mejor persona, muchas gracias.

También hay que mencionar algunas personas por su activa participación en nuestra formación como ingenieros, por lo que es necesario agradecer.

A los amigos que nos acompañan hoy y que sobre todo creyeron en nuestro trabajo.

A mi compañera de tesis que mucho se ha esforzado en la realización de este trabajo, por ser más que una compañera de tesis, una buena amiga.

A mi familia que siempre está ahí cuando la necesito.

A mi hermano muy preocupado siempre por lo que me pasa.

A mis padres que sí hoy defendiendo el título de ingeniero, sí soy la persona en la que me he convertido es gracias a ellos, a su apoyo, su ejemplo y su total entrega como padres.

A mi novia que compartió los mismos sufrimientos que yo.



DEDICATORIA

Luna

A mis padres por ser la fuente que me incita cada día a seguir adelante, por confiar en mis decisiones, por apoyarme toda la vida.

A mis hermanas que me vieron crecer y verme convertir en una profesional.

Otoniel

A todos mis amigos que siempre me brindaron buenos y malos momentos.

A mi familia muy atenta siempre y preocupada.

A todos los que apoyaron esta idea y no dudaron de nuestro trabajo.

A mi hermano siempre pensando en mí.

A mis padres por su entera confianza y apoyarme siempre.



RESUMEN

La aparición y desarrollo de Internet y la telefonía celular en las últimas décadas, ha originado un uso intensivo de los servicios de comunicación que sobre ellas se brindan. El correo electrónico, la mensajería instantánea y la mensajería de texto SMS (del inglés: *Short Message Service*) han revolucionado las comunicaciones y el mercado de las aplicaciones y los servicios en esta rama. Aunque el uso de servicios tradicionales como correo ordinario, fax y paging, aún están presentes en muchas partes del mundo. Un paging o beeper es un simple dispositivo para la localización de personas a través de mensajes de texto corto.

Una de las líneas de trabajo que tiene el sector de las telecomunicaciones, es la convergencia de servicios en un mismo dispositivo. Con el fin de aumentar la usabilidad de los beepers, en esta investigación se propone un sistema que integre en una plataforma el envío de mensajes paging a través de las vías de comunicación SMS, correo electrónico e Internet.

Se incluye un estudio del arte de los sistemas paging en Cuba y en el mundo, las tecnologías y herramientas utilizadas para su desarrollo, las características del sistema, su planificación, implementación y la estrategia de pruebas que se llevó a cabo. Además el trabajo contiene una amplia investigación que se recomienda como referencia para futuros trabajos sobre el tema abordado.

Palabras claves: Convergencia, Correo electrónico, Internet, SMS, Paging.

TABLA DE CONTENIDOS

AGRADECIMIENTOS COMUNES.....	I
AGRADECIMIENTOS.....	II
DEDICATORIA.....	IV
RESUMEN.....	V
ÍNDICE DE FIGURAS.....	X
INTRODUCCIÓN.....	1
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA.....	4
1.1 INTRODUCCIÓN.....	4
1.2 ESTADO DEL ARTE DE LOS SISTEMAS PAGING.....	4
1.2.1 Protocolos utilizados en los sistemas paging.....	5
1.2.2 Soluciones existentes que brindan servicios paging.....	6
1.3 APLICACIONES WEB.....	7
1.4 TECNOLOGÍAS UTILIZADAS PARA EL DESARROLLO DEL SISTEMA PAGING.....	8
1.4.1 Sistemas de Gestión de Contenidos (CMS).....	8
1.4.1.1 Drupal.....	8
1.4.1.2 Joomla.....	8
1.4.1.3 CMS seleccionado.....	9
1.4.2 Lenguajes de Programación. PHP. JavaScript.....	9
1.4.3 Lenguajes de estructura y estilo.....	10
1.4.4 Metodologías de desarrollo.....	10
1.4.4.1 Metodologías Tradicionales o Pesadas.....	10
1.4.4.2 Metodologías ágiles.....	11
1.4.4.3 Metodología seleccionada.....	12
1.4.5 Lenguaje de modelado. UML (Lenguaje Unificado de Modelado).....	12
1.5 HERRAMIENTAS.....	12
1.5.1 Herramienta de modelado. Visual Paradigm 6.4.....	12
1.5.2 Servidores Web.....	12

1.5.2.1	Internet Information Server (IIS).....	12
1.5.2.2	Apache.....	13
1.5.2.3	Servidor web seleccionado.....	13
1.5.3	Sistemas Gestores de Bases de Datos.....	13
1.5.3.1	PostgreSQL.....	13
1.5.3.2	MySQL.....	13
1.5.3.3	Sistema gestor seleccionado.....	14
1.5.4	Herramientas para el diseño de bases de datos.....	14
1.5.4.1	ER/studio.....	14
1.5.4.2	DBDesigner.....	14
1.5.4.3	Herramienta para el diseño de bases de datos seleccionada.....	14
1.5.5	Plataforma de Desarrollo.....	14
1.5.5.1	PhpDesigner.....	14
1.5.5.2	Zend Studio.....	15
1.5.5.3	Plataforma de desarrollo seleccionada.....	15
1.5.6	Servidores de correo.....	15
1.5.6.1	Sendmail.....	15
1.5.6.2	Postfix.....	15
1.5.6.3	Servidor de correo seleccionado.....	16
1.5.7	Herramienta de gestión de módem GSM. Gammu 1.30.....	16
1.5.8	Herramienta de control de versiones. SVN (Subversion) 1.6.....	16
1.6	CONCLUSIONES PARCIALES.....	17
CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA, EXPLORACIÓN Y PLANIFICACIÓN.....		18
2.1	INTRODUCCIÓN.....	18
2.2	PROBLEMÁTICA.....	18
2.3	OBJETO DE AUTOMATIZACIÓN.....	18
2.4	PROPUESTA DEL SISTEMA.....	19
2.5	SEGURIDAD EN EL SISTEMA.....	20
2.6	PERSONAS RELACIONADAS CON EL SISTEMA.....	22
2.7	ESPECIFICACIÓN DE REQUISITOS NO FUNCIONALES DEL SISTEMA.....	22

2.7.1	Requisitos no funcionales.....	23
2.8	FASE DE EXPLORACIÓN Y PLANIFICACIÓN.....	24
2.8.1	Exploración.....	24
2.8.1.1	Historias de Usuario.....	24
2.8.2	Planificación.....	29
2.8.2.1	Estimación de esfuerzo por Historias de Usuario.....	29
2.8.2.2	Plan de iteraciones.....	30
2.8.2.3	Plan de duración de las iteraciones.....	31
2.8.2.4	Plan de entregas.....	32
2.9	CONCLUSIONES PARCIALES.....	33
CAPÍTULO 3: DISEÑO DEL SISTEMA.....		34
3.1	INTRODUCCIÓN.....	34
3.2	ESTILO ARQUITECTÓNICO CLIENTE-SERVIDOR.....	34
3.3	PATRÓN DE ARQUITECTURA. MODELO-VISTA-CONTROLADOR (MVC).....	35
3.4	ARQUITECTURA DEL SISTEMA.....	36
3.5	PATRONES DE DISEÑO.....	37
3.5.1	Patrones GRASP (<i>General Responsibility Assignment Software Patterns</i>).....	38
3.5.1.1	Experto.....	38
3.5.1.2	Creador.....	39
3.5.2	Patrones GOF (Gang of Four).....	39
3.5.2.1	Factory.....	39
3.6	DISEÑO DEL SISTEMA.....	40
3.6.1	Componentes de Joomla.....	40
3.7	TARJETAS COLABORADOR - RESPONSABILIDAD – CLASE (CRC).....	41
3.8	DISEÑO DE LA BASE DE DATOS.....	42
3.9	CONCLUSIONES PARCIALES.....	44
CAPÍTULO 4: IMPLEMENTACIÓN Y PRUEBA.....		45
4.1	INTRODUCCIÓN.....	45
4.2	FASE DE IMPLEMENTACIÓN.....	45
4.2.1	Iteraciones.....	45

4.2.1.1	Primera iteración.....	46
4.2.1.2	Segunda iteración.....	46
4.2.2	Tareas de la ingeniería.....	47
4.3	DIAGRAMA DE DESPLIEGUE.....	49
4.4	PRUEBAS.....	50
4.4.1	Pruebas de aceptación.....	51
4.4.2	Pruebas de seguridad.....	54
4.5	CONCLUSIONES PARCIALES.....	55
CONCLUSIONES GENERALES		56
RECOMENDACIONES.....		57
REFERENCIAS BIBLIOGRÁFICAS.....		58
BIBLIOGRAFÍA.....		61

ÍNDICE DE FIGURAS.

Figura 1: Esquema de un sistema paging. Fuente [4].....	5
Figura 2: Propuesta del sistema.	20
Figura 3: Arquitectura cliente-servidor. Fuente [47].	34
Figura 4: El patrón MVC. Fuente[49].	36
Figura 5: Arquitectura del sistema.	37
Figura 6: Muestra del patrón Experto en el sistema.....	38
Figura 7: Muestra del patrón Creador en el sistema.	39
Figura 8: Muestra del patrón Factory en el sistema.	40
Figura 9: Modelo de datos.....	43
Figura 10: Diagrama de despliegue.....	50



INTRODUCCIÓN

El desarrollo de las telecomunicaciones ha propiciado el surgimiento de nuevas tecnologías que han eliminado el espacio físico como una barrera para comunicarse. La aparición, desarrollo y expansión de Internet y la telefonía celular en las últimas décadas, han originado el uso intensivo de los servicios de comunicación que sobre ellas se brindan. El correo electrónico, la mensajería instantánea y la mensajería de texto (SMS) han revolucionado el mundo de las telecomunicaciones, incluyendo el mercado de las aplicaciones y los servicios.

El uso masivo de estas tecnologías ha deteriorado la utilización de los servicios tradicionales (Voz, FAX y Paging o Beeper), aunque aún están presentes en muchas partes del mundo.

Un Paging o Beeper es un simple dispositivo personal para mensajes de texto corto, los cuales fueron utilizados inicialmente por los médicos para ser alertados en caso de emergencia. Su desarrollo continuó aumentando, adoptando así tecnologías digitales, las cuales permitieron hacer más ágil y barato el servicio, y a su vez más accesible para diferentes tipos de personas e instituciones, como son el personal de la seguridad, profesionales de la informática, equipos de mantenimientos y reparaciones, entre otras empresas. [1]

Actualmente una de las líneas más importantes de investigación y desarrollo que tiene el sector de las telecomunicaciones es la convergencia de servicios en un dispositivo, con tal objetivo se han llevado a cabo muchas iniciativas encaminadas a permitir el acceso a varios servicios desde dispositivos y tecnologías diferentes.

Cuba cuenta con una infraestructura de telecomunicaciones donde la telefonía fija y fija alternativa cubren casi la totalidad del país. Además cuenta con una red celular GSM (del inglés: *Global System for Mobile Communications*) y con el servicio de mensajería paging, que si bien no están en la cima del desarrollo de las tecnologías a nivel mundial, garantiza cierta calidad de servicio a la población.

Movitel es la empresa cubana encargada de brindar los servicios paging. Entre ellos se encuentran el envío de mensajes paging desde la web, correo electrónico o vía telefónica, la venta de productos paging a las diferentes empresas que lo necesiten, además de posibilitar la programación de mensajes para la fecha y hora que se desee vía telefónica, la cual consiste en llamar al centro de operadoras, y pedir el envío de un mensaje recordatorio los días y a la hora que se desee. [2]

Sin embargo la programación de mensajes para la fecha y hora deseada no se encuentra automatizada y no se permite el envío de mensaje a un beeper mediante la vía SMS; elementos que influyen negativamente en la usabilidad del beeper, debido a que se requiere de un ordenador con Internet o de algún teléfono fijo para proceder a enviar un mensaje.



En la UCI existe un servicio de mensajería paging que solo permite el envío de mensajes vía web, no mantiene un registro de los mensajes enviados por un usuario determinado y los campos obligatorios carecen de validación.

A partir de la problemática planteada surge el siguiente **problema a resolver**: ¿Cómo permitir la convergencia de las vías de comunicación, SMS, correo electrónico e Internet para envío de mensajes paging en una misma plataforma?

Definiéndose como **objeto de estudio** los sistemas de mensajerías.

Siendo así el **campo de acción** la convergencia de las vías de comunicación SMS, correo electrónico e Internet en el envío de mensajes paging.

El **objetivo general** es desarrollar un sistema informático que permita la gestión y envío de mensajes paging a partir de la convergencia de los servicios de telecomunicaciones, SMS, correo electrónico e Internet.

Para dar cumplimiento al objetivo planteado, se proponen los **objetivos específicos** siguientes:

- ✓ Gestionar el envío de mensajes paging vía email.
- ✓ Permitir la persistencia de los mensajes, para que tengan constancia y puedan ser leído posteriormente.
- ✓ Permitir la programación de mensajes paging para la fecha y hora deseada.
- ✓ Integrar en una plataforma las diferentes vías de envío de mensajes paging planteadas anteriormente.

Con el propósito de cumplir los objetivos trazados, se definieron **las tareas de la investigación** siguientes:

- ✓ Realizar el diseño teórico de la investigación.
- ✓ Realizar un estado del arte de los principales sistemas existentes a nivel mundial.
- ✓ Seleccionar las herramientas y lenguajes a utilizar para el desarrollo del sistema.
- ✓ Diseñar e implementar una arquitectura que satisfaga los objetivos planteados.
- ✓ Aplicar los patrones de diseño para el desarrollo del sistema.
- ✓ Instalar un servidor de correos para gestionar el envío de mensajes paging por esta vía.
- ✓ Validar las funcionalidades aplicando pruebas que permitan comprobar el correcto funcionamiento del sistema.

Se plantearon las **preguntas científicas** siguientes:

¿Qué herramientas y metodología utilizar para el desarrollo del sistema?

¿Qué patrones de diseño y arquitectura se ajustan más al desarrollo de la solución informática?

¿Qué elementos posibilitan el funcionamiento de un sistema paging?

¿Cómo extraer los datos de envío de los mensajes SMS y registrarlos en una base de datos?



¿Cuáles son las herramientas disponibles para la interacción con un modem GSM?

¿Cómo lograr un alto nivel de seguridad en la transferencia de información?

El presente trabajo está estructurado en 4 capítulos, de los cuales se da una breve descripción a continuación:

Capítulo 1: “Fundamentación teórica”, se describen los aspectos teóricos relacionados con el desarrollo del trabajo, realizando un estudio del estado del arte de los sistemas paging, los antecedentes de los mismos, así como las metodologías, herramientas y plataforma con las cuales se desarrolla la aplicación.

Capítulo 2: “Características del sistema”, se muestra el modelo del sistema, describiéndose la propuesta de desarrollo del mismo, los requisitos no funcionales que debe cumplir, se plantean las historias de usuarios, la estimación de esfuerzo por cada una de las historias de usuarios y los procesos que serán objetos de automatización.

Capítulo 3: “Diseño del sistema”, se obtiene como principal salida las tarjetas Colaborador - Responsabilidad – Clase como técnica de diseño, se describe la arquitectura, el estilo arquitectónico y los patrones de diseño empleados, así como el diseño de la base de datos y su descripción.

Capítulo 4: “Implementación y Prueba”, se describen las tarjetas de la ingeniería generadas por cada historia de usuario que conformarán así la implementación, se representa el diagrama de despliegue, además se exponen los casos de pruebas realizadas al sistema.



CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

1.1 Introducción.

El desarrollo de un sistema que permita la convergencia de las vías de comunicación mencionadas anteriormente para el envío de mensajes paging, aumentará la usabilidad de los beeper. En el capítulo, se realizará el estudio del funcionamiento del envío de mensajes paging, los protocolos utilizados y las soluciones existentes que brindan servicios de mensajería paging. También se fundamentan las herramientas, metodologías y lenguajes a utilizar para desarrollo del sistema.

1.2 Estado del arte de los sistemas paging.

Un sistema paging tiene como misión principal la transmisión inalámbrica de mensajes cortos a una terminal móvil, en este caso el beeper. El desarrollo de la industria paging se inicia con los primeros sistemas de radio móvil iniciados por el Departamento de Policía de los Estados Unidos en 1921, cuando aparece el concepto de radiodifusión de una sola vía, usando transmisores para radiodifundir mensajes de voz desde un sitio estacionario (estación base) a una unidad móvil. [3]

Al inicio un sistema paging funcionaba solamente vía telefónica, el envío de mensajes era realizado por usuarios que llaman a una central telefónica pública, la operadora contestaba y solicitaba el número del beeper y el mensaje que deseaba enviar, luego la operadora enviaba el mensaje al número de beeper solicitado a través de satélites o radiobases, conocidos como beeper de una sola vía, que son aquellos que se limitan solamente a recibir mensajes.

Actualmente los sistemas paging han evolucionado grandemente, debido a que los mensajes podrían ser enviados a través de una página en Internet mediante una cuenta reconocida por el proveedor de paging, donde solo se tiene que ingresar el número de beeper en la página y el mensaje a enviar, y en pocos segundos el mensaje es recibido por el dispositivo móvil. También pueden ser enviados a través de un correo electrónico, de modo parecido al de Internet. [4]

Los mismos a su vez no se podían limitar a recibir solo mensajes, sino que también se pudieran enviar, por lo que se desarrollaron dispositivos móviles que pudieran recibir y enviar mensajes, conocidos como beeper de dos vías, y constan de un teclado para escribir el mensaje. [4]

El sistema paging está compuesto por tres elementos principales, como se muestra en la Figura 1:

- ✓ Dispositivos de entrada: Son el medio por el cual se introduce el mensaje que se desea transmitir, ejemplo la computadora, teléfono y el beeper.
- ✓ Satélites y radiobases: Son los medios a través de los cuales se realiza la transmisión inalámbrica del mensaje.

- ✓ Terminales móviles: Son los dispositivos portátiles que reciben el mensaje y se encargan de alertar al propietario de que ha recibido un mensaje. [4]

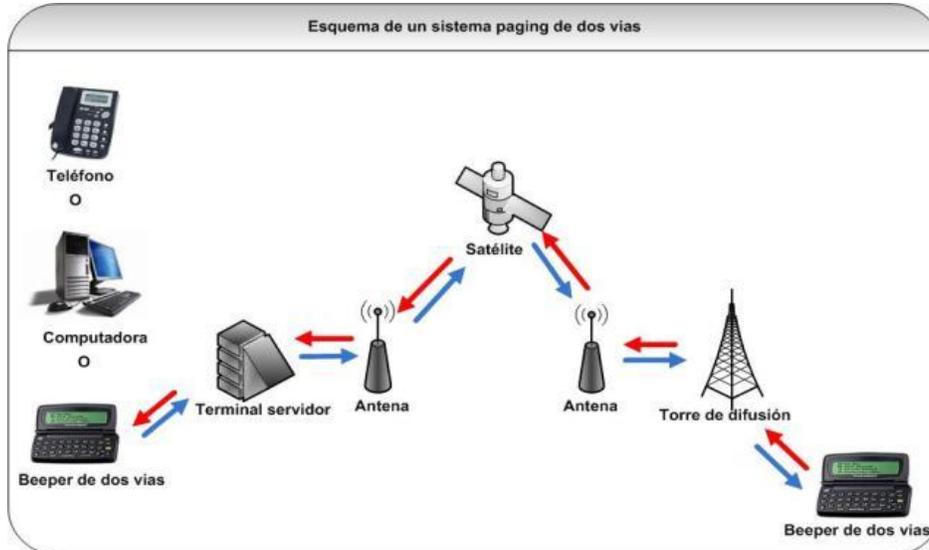


Figura 1: Esquema de un sistema paging. Fuente [4].

1.2.1 Protocolos utilizados en los sistemas paging.

Existen diversos protocolos utilizados en los sistemas paging, como el TAP (del inglés: *Test All Protocol*), POCSAG (del inglés: *Post Office Code Standardization Advisory Group*), Flex y ReFlex. Los protocolos más utilizados son el Flex y el ReFlex, siendo los más recientes e importantes en el desarrollo de los sistemas paging. En Cuba los protocolos utilizados para los servicios paging son Flex y POCSAG. [5]

El protocolo Flex viene de un grupo de tecnologías inalámbricas creadas por Motorola Inc., es un protocolo para beeper de una sola vía, fue diseñado para satisfacer necesidad de incrementar las capacidades y características de los canales de transmisión de los sistemas paging. Opera en un canal de 25 kHz a velocidades de 1600, 3200 o 6400 bps, soportando más de un billón de direcciones, ofreciendo así la más alta capacidad de un canal en el mundo con velocidades de mensajería rápidas, a su vez es robusto, ofrece mayor integridad y confidencialidad de los datos e incrementa la vida de la batería cinco veces más que cualquier otro protocolo de paging. [3] [4]

Por su parte el protocolo ReFlex es una mejora del Flex conservando todas las ventajas del mismo, incorpora los atributos de mensajería de dos vías, proporcionando comunicaciones de dos vías. Un beneficio de ReFlex es la habilidad de confirmar automáticamente la recepción de los mensajes a la central de control del sistema. [4]

POCSAG se consideró un protocolo de alta velocidad cuando se desarrolló. Puede manejar más de dos millones de direcciones y soporta mensajes de tipo numérico, alfanumérico y de tono. Opera a 512, 1200 y



2400 bps, y es el protocolo de buscapersonas más extensamente difundido. Posee un sistema robusto de detección y corrección de errores. La sincronización se efectúa mediante códigos de datos llamados claves, y cada buscapersonas podrá tener una clave diferente, así se pueden dividir en grupos de buscapersonas en el momento de la sincronización y mejorar la vida de la batería. [6] [7]

1.2.2 Soluciones existentes que brindan servicios paging.

Existen empresas a nivel mundial que cuentan con un software que permiten brindar servicios de mensajería paging, las mismas poseen sus sitios en la red, para darle publicidad a sus productos, servicios y brindar información al cliente.

INDICOM es una empresa argentina que ofrece desde 1993 servicio de paging a individuos y empresas líderes con exigentes necesidades de comunicación. Se encarga de la venta de productos paging y envío de mensajes a beeper el día y hora que lo desee, realizando un pago establecido por la empresa. [8]

SkyTel es una compañía argentina que ofrece servicios de telecomunicaciones y paging desde el 2004. Ofrece los mismos servicios mencionados anteriormente, además del envío de mensajes grupales y cíclicos. [9]

Electronic Engineering es una compañía que ofrece servicios de electrónica y comunicación, la cual permite enviar mensajes a beeper de forma gratuita desde la web y a través de correo electrónico, incluyendo además la venta de productos paging. [10]

Movitel es la empresa cubana que ofrece servicios paging, permite el envío de mensajes mediante la web, correo electrónico o vía telefónica, ofrece la venta de productos paging a las diferentes empresas que lo necesiten, además de posibilitar la programación de mensajes para la fecha y hora que se desee vía telefónica y la revisión de mensajes utilizando la web o llamando al centro de operadora, para lo que necesitará un número y una contraseña [2]; al mismo tiempo presenta la limitación de no poseer la programación de mensajes para la fecha y hora deseada automatizada, lo cual tiene como inconveniente que exista una alta demanda de llamadas telefónicas para programar mensajes, lo que puede conllevar a la congestión de las líneas telefónicas y gastos innecesarios en la contratación de personal para atender las llamadas realizadas.

La **UCI** posee un servicio de mensajería paging, donde solo permite el envío de mensajes vía web. El mismo carece del envío de mensajes a través de correo electrónico, no permite la programación de mensajes, y no mantiene un registro de los mensajes enviados por un usuario determinado y los campos obligatorios carecen de validación; además de mantener al usuario dependiente de una pc con Internet o de un teléfono fijo.

Luego de analizar las soluciones expuestas anteriormente se llega a la conclusión que permiten de manera



general el envío de mensajes paging vía web, y algunos posibilitan la programación de mensajes para la fecha y hora deseada. A su vez los servicios paging brindados por las compañías internacionales son ofrecidos de manera regional, es decir, solo se benefician los usuarios que se encuentren en el país o región que abarca la señal de envío de las compañías, lo que conlleva a que no sean de utilidad en Cuba. De ahí la necesidad de desarrollar un sistema que supla las limitaciones encontradas en las soluciones cubanas, permitiendo la integración de diferentes vías de comunicación para el envío de mensajes paging y automatice su programación. El mismo contribuirá a mejorar la calidad de comunicación en Cuba, optimizando el servicio brindado por la empresa cubana Movitel, y proporcionando un medio más de comunicación en la universidad, siendo de mucha ayuda para la localización del personal.

1.3 Aplicaciones web.

Las aplicaciones web son aquellas aplicaciones que los usuarios pueden utilizar accediendo a un Servidor web a través de Internet o de una intranet mediante un navegador. En otras palabras, es una aplicación (Software) que se codifica en un lenguaje soportado por los navegadores web, en la que se confía la ejecución al navegador. Las aplicaciones web poseen características como independencia del Sistema operativo, así como facilidad para actualizar y mantener sin distribuir e instalar software a miles de usuarios potenciales. [11]

Se decide desarrollar una aplicación web gracias a los beneficios y ventajas que las mismas ofrecen, a continuación se plantearán algunos.

- ✓ Para utilizar la aplicación web solo se necesita un computador con navegador Web y conexión a Internet, y no se necesitan tener conocimientos previos de informática.
- ✓ Múltiples usuarios concurrentes: Aplicaciones que pueden ser utilizada por múltiples usuarios al mismo tiempo.
- ✓ Ahorra tiempo: Se pueden realizar tareas sencillas sin necesidad de descargar ni instalar ningún programa. No hay problemas de compatibilidad: Basta tener un navegador actualizado para poder utilizarlas. No ocupan espacio en el disco duro.
- ✓ Multiplataforma: Se pueden usar desde cualquier sistema operativo porque sólo es necesario tener un navegador.
- ✓ Portables: Es independiente del ordenador donde se utilice (un PC de sobremesa, un portátil...) porque se accede a través de una página web (sólo es necesario disponer de acceso a Internet).

[11] [12]



1.4 Tecnologías utilizadas para el desarrollo del sistema paging.

La adecuada selección de las tecnologías para la elaboración de aplicaciones web, proporciona calidad final y reduce el tiempo de desarrollo del producto. Actualmente existen diversas tecnologías, pero es necesario utilizar las que sean estables y sólidas para asegurar un desarrollo simple y eficaz de la aplicación. Para lograr el buen desarrollo del sistema se llevó a cabo un estudio para la selección de las principales tecnologías a utilizar.

1.4.1 Sistemas de Gestión de Contenidos (CMS).

Un CMS (del inglés: Content Management System) es una herramienta que permite a un editor crear, clasificar y publicar cualquier tipo de información en una página web. Generalmente los CMS trabajan contra una base de datos, de modo que el editor simplemente actualiza una base de datos, incluyendo nueva información o editando la existente. [13]

1.4.1.1 Drupal.

Drupal es un CMS para sitios web basado en PHP. Es software libre, se compone de una infraestructura base y un conjunto de módulos que ofrecen una amplia variedad de funciones, permitiendo publicar artículos, imágenes, u otros archivos y servicios añadidos como foros, encuestas, votaciones, blogs, administración de usuarios y permisos, administración de listas de correo electrónico entre otras. [14]

Características generales de Drupal: [15]

- ✓ **Ayuda on-line:** un robusto sistema de ayuda online y páginas de ayuda para sus módulos.
- ✓ **Módulos:** la comunidad ha contribuido muchos módulos que proporcionan diversas funcionalidades.
- ✓ **Personalización:** un robusto entorno de personalización está implementado en su núcleo. Tanto el contenido como la presentación pueden ser individualizados de acuerdo las preferencias definidas por el usuario.
- ✓ **Autenticación de usuarios:** los usuarios se pueden registrar e iniciar sesión de forma local o utilizando un sistema de autenticación externo.
- ✓ **Permisos basados en roles:** los administradores no tienen que establecer permisos para cada usuario. En lugar de eso, pueden asignar permisos a un 'rol' y agrupar los usuarios por roles.

1.4.1.2 Joomla.

Joomla es un CMS dinámico que permite crear sitios web de alta interactividad, profesionalidad y eficiencia basado en PHP. Es de uso gratuito, de libre distribución, y de código abierto (*OpenSource*) y fácil de usar.



Con Joomla el desarrollador sólo se debe ocupar de la información que se desea publicar, ya que el sistema gestiona todos los demás detalles técnicos y administrativos. [16]

Características generales de Joomla:

- ✓ **Organización del sitio web:** Está preparado para organizar eficientemente los contenidos del sitio en secciones y categorías, lo que facilita la navegabilidad para los usuarios y permite crear una estructura sólida, ordenada y sencilla para los administradores.
- ✓ **Publicación de Contenidos:** Se puede crear páginas ilimitadas y editarlas desde un sencillo editor, que permite formatear los textos con los estilos e imágenes deseados. Los contenidos son totalmente editables y modificables.
- ✓ **Escalabilidad e implementación de nuevas funcionalidades:** Ofrece la posibilidad de instalar, desinstalar y administrar componentes y módulos, que agregarán servicios de valor a los visitantes de su sitio web.
- ✓ **Administración de usuarios:** Permite almacenar datos de usuarios registrados. La administración de usuarios es jerárquica, y los distintos grupos de usuarios poseen diferentes niveles de facultades o permisos dentro de la gestión y administración del sitio.
- ✓ **Diseño y aspecto estético del sitio:** Es posible cambiar todo el aspecto del sitio web tan solo con un par de clicks, gracias al sistema de templates que utiliza Joomla. [17]

1.4.1.3 CMS seleccionado.

Las principales ventajas que conllevaron al uso de Joomla en su versión 2.5.4 son:

- ✓ Permite construir sitios web potentes y de forma rápida.
- ✓ Está hecho para diseños simples, sitios pequeños con menor cantidad de páginas.
- ✓ Cumple con la necesidad de buscar una solución de código abierto.
- ✓ Su curva de aprendizaje es corta y presenta una de las mayores comunidades de desarrollo online.
- ✓ Posee un panel administrativo robusto.

1.4.2 Lenguajes de Programación. PHP. JavaScript.

Se decidió usar **Hypertext Pre-processor (PHP)** para la programación del sistema, ya que es un lenguaje interpretado de alto nivel, concentrado en la programación de script del lado del servidor, caracterizado por simplicidad y potencia, permite crear páginas web dinámicas de forma rápida y fácil. Soporta gran variedad de funciones para la explotación de bases de datos, es multiplataforma, permite la conexión al servidor de bases de datos a utilizar, consume pocos recursos, posee rapidez de ejecución, y es completamente gratuito. [18] Otra razón que conlleva a su uso es que el CMS seleccionado está basado en el lenguaje.



Se utilizará **JavaScript** como lenguaje de programación interpretado del lado del cliente basado en acciones, puesto que es utilizado principalmente para crear páginas web dinámicas. Es un lenguaje independiente de la plataforma, donde se puede crear todo tipo de programa que puede ser ejecutado en cualquier ordenador del mercado: Linux, Windows, Apple, etc. No requiere de compilación debido a que el lenguaje funciona del lado del cliente, y los navegadores son los encargados de interpretar el código. Permite que exista interactividad dinámica entre el usuario y la página. Su principal característica es que permitirá ejecutar instrucciones como respuesta a las acciones del usuario. [19]

1.4.3 Lenguajes de estructura y estilo.

Se hará uso de **Hyper Text Markup Language (HTML)** debido a que es un lenguaje de composición de documentos y es utilizado únicamente para dar estructura a las páginas, pues indica cómo desplegar el contenido del documento, incluyendo texto, gráficos, sonido o vídeo y otros medios soportados. [20]

Será utilizado **CascadingStyle Sheets (CSS)** como lenguaje de hojas de estilos para controlar el aspecto y la presentación estética de los documentos electrónicos definidos con HTML. La definición de un buen estilo aumenta considerablemente la usabilidad y accesibilidad al sistema. Facilita la reutilización de código, mejora la accesibilidad del documento, y reduce la complejidad de su mantenimiento. [21]

1.4.4 Metodologías de desarrollo.

1.4.4.1 Metodologías Tradicionales o Pesadas.

Teniendo en cuenta la filosofía de desarrollo de las metodologías, aquellas con mayor énfasis en la planificación y control del proyecto, en especificación precisa de requisitos y modelado, reciben el apelativo de Metodologías Tradicionales o Pesadas.

Las metodologías tradicionales imponen una disciplina de trabajo sobre el proceso de desarrollo del software, con el fin de conseguir un software más eficiente. Se centran especialmente en el control del proceso, mediante una rigurosa definición de roles, actividades, artefactos, herramientas y notaciones para el modelado y documentación detallada. Además, las metodologías tradicionales no se adaptan adecuadamente a los cambios, por lo que no son métodos adecuados cuando se trabaja en un entorno, donde los requisitos no pueden predecirse o bien pueden variar. [22]

Entre las metodologías tradicionales o pesadas se puede citar:

Rational Unified Procces (RUP), sus principales características son:

- ✓ **Guiado por Casos de uso:** sirven para describir el comportamiento del sistema y elaborar los casos de prueba con los que se comprueba que el sistema desarrollado *“hace lo que el cliente quiere”*.



- ✓ **Centrado en la arquitectura:** la arquitectura del sistema es la columna vertebral de todo el desarrollo del mismo. Cada iteración gira en torno a la misma, fortaleciendo y corrigiendo sus características.
- ✓ **Iterativo e incremental:** en cada ciclo de iteración se produce una nueva versión del software. Utiliza UML como lenguaje de modelado y cuenta con varias fases de trabajo en las cuales se desarrolla una serie de flujos fundamentales del desarrollo del proyecto. [23]

1.4.4.2 Metodologías ágiles.

Las metodologías ágiles son una buena elección cuando se trabaja con requisitos desconocidos o variables. Si no existen requisitos estables, no existe una gran posibilidad de tener un diseño estable y de seguir un proceso totalmente planificado, que no vaya a variar ni en tiempo ni en dinero. En estas situaciones, un proceso adaptativo será mucho más efectivo que un proceso predictivo. Por otra parte, los procesos de desarrollo adaptativos también facilitan la generación rápida de prototipos y de versiones previos a la entrega final, lo cual agradará al cliente. [22]

Entre las metodologías ágiles se puede citar:

Extreme Programming (XP), sus principales características son:

La metodología XP se ajusta estrictamente a una serie de reglas que se centran en la necesidad del cliente para lograr un producto de buena calidad en poco tiempo. A su vez es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en el desarrollo del software, promueve el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, y pensada para proyectos de corta duración y equipos pequeños. [24]

XP presenta buenas prácticas basadas en la simplicidad, la comunicación y el reciclado continuo de código; las fundamentales que conllevan al uso de esta metodología son:

- ✓ **El juego de la planificación:** Hay una comunicación frecuente entre el cliente y los programadores. El equipo técnico realiza una estimación del esfuerzo requerido para la implementación y los clientes deciden sobre el ámbito y tiempo de las entregas y de cada iteración.
- ✓ **Diseño simple:** Se debe diseñar la solución más simple que pueda funcionar y ser implementada en un momento determinado del proyecto.
- ✓ **Programación en parejas:** Toda la producción de código debe realizarse con trabajo en parejas de programadores. Proporcionando ventajas implícitas (menor tasa de errores, mejor diseño, mayor satisfacción de los programadores).
- ✓ **Propiedad colectiva del código:** Cualquier programador puede cambiar cualquier parte del código en cualquier momento.



- ✓ **Estándares de programación:** XP enfatiza que la comunicación de los programadores es a través del código, con lo cual es indispensable que se sigan ciertos estándares de programación para mantener el código legible.

1.4.4.3 Metodología seleccionada.

Se decido utilizar la metodología XP debido a que se adapta en gran medida tanto al tipo de proyecto que se va a desarrollar, como a las condiciones de trabajo. El equipo de trabajo es pequeño, es un proyecto de corta duración, el cliente forma parte del equipo ya que está presente en la universidad, y se programará en parejas; además de que los requisitos del sistema cambian frecuentemente.

1.4.5 Lenguaje de modelado. UML (Lenguaje Unificado de Modelado).

El lenguaje de modelado **UML** permite generar diseños que capturen las ideas en una forma convencional y fácil de comprender para comunicarlas a otras personas. Permite la modelación de sistemas con tecnología orientada a objetos. La clave está en organizar el proceso de diseño de tal forma que los analistas, clientes, desarrolladores y otras personas involucradas en el desarrollo del sistema lo comprendan y convengan con él. [25] [26] Proporcionando UML tal organización, será utilizado para realizar el modelado del diagrama de despliegue del sistema.

1.5 Herramientas.

1.5.1 Herramienta de modelado. Visual Paradigm 6.4.

Se hará uso de la herramienta CASE (de las siglas en inglés: *Computer Aided Software Engineering*) **Visual Paradigm** para modelar el Diagrama de despliegue, ya que soporta el ciclo de vida completo del desarrollo de software, permite modelar todo tipo de diagramas, permite modelar con mayor rapidez, mayor exactitud, mejor trabajo en equipo y fácil de utilizar. [27] Esta es una herramienta propietaria, pero se hará uso de ella debido a que la universidad cuenta con la licencia de la misma.

1.5.2 Servidores Web.

1.5.2.1 Internet Information Server (IIS).

Es un potente servidor web que ofrece una infraestructura de gran fiabilidad, capacidad de manejo y escalabilidad para aplicaciones web sobre todas las versiones de Windows Server 2003. IIS hace posible que las organizaciones aumenten la disponibilidad de sus sitios y aplicaciones web y a la vez reducir sus costes administrativos. IIS 0.6 soporta la Iniciativa de Sistemas Dinámicos de Microsoft (DSI) con



monitorización de estado de salud automático, aislamiento de procesos y capacidades de gestión mejoradas. IIS solo funciona sobre sistemas Windows, y debe ser usado bajo licencia. [28]

1.5.2.2 Apache.

Apache es el servidor web donde su configurabilidad, robustez y estabilidad hacen que cada vez millones de servidores reiteren su confianza en él.

Sus principales características son: [29] [30]

- ✓ Corre en una multitud de Sistemas Operativos, lo que lo hace prácticamente universal.
- ✓ Es una tecnología gratuita de código abierto.
- ✓ Es altamente configurable de diseño modular.
- ✓ Trabaja con gran cantidad de PHP y otros lenguajes de script, teniendo todo el soporte que se necesita para tener páginas dinámicas.
- ✓ Permite personalizar la respuesta ante los posibles errores que se puedan dar en el servidor.
- ✓ Gestiona solicitudes y respuestas HTTP (del inglés: *Hypertext Transfer Protocol*).

1.5.2.3 Servidor web seleccionado.

Para atender y responder a las diversas peticiones de los navegadores se hará uso del servidor web Apache en su versión 2.0.53, es sencillo de configurar, fiable, gratis y de código abierto, trabaja de forma eficiente con PHP y otros lenguajes de script.

1.5.3 Sistemas Gestores de Bases de Datos.

1.5.3.1 PostgreSQL.

PostgreSQL es un sistema de gestión de bases de datos objeto-relacional, de código abierto. Funciona muy bien con grandes cantidades de datos y una alta concurrencia de usuarios accediendo a la vez al sistema. Disponible para Linux y UNIX en todas sus variantes y Windows 32/64bit. Posee funciones/procedimientos de almacenados en numerosos lenguajes de programación y completa documentación. [31]

1.5.3.2 MySQL.

MySQL es un sistema de gestión de bases de datos relacional de código abierto, multiplataforma, fácil de usar y administrar. Es multihilos y capaz de almacenar una enorme cantidad de datos de gran variedad y de distribuirlos para cubrir las necesidades de cualquier tipo de organización. Incluye todos los elementos



necesarios para preparar diferentes niveles de acceso de usuario, administrar el sistema y protegerlo. [32]

1.5.3.3 Sistema gestor seleccionado.

Como sistema gestor de base de datos se utilizará MySQL en su versión 5.1.37 para desplegar el sistema, debido a que es el que trae por defecto el CMS seleccionado. Además es fácil de configurar y aprender, presenta un rendimiento alto, debido a que es rápido y posee gran velocidad al conectar con el servidor, es gratis y multiplataforma.

1.5.4 Herramientas para el diseño de bases de datos.

1.5.4.1 ER/studio

ER/studio es una herramienta de modelado de datos fácil de usar y multinivel, para el diseño y construcción de bases de datos a nivel físico y lógico. Direcciona las necesidades diarias de los administradores de bases de datos, desarrolladores y arquitectos de datos que construyen y mantienen aplicaciones de bases de datos grandes y complejas. Está equipado para crear y manejar diseños de bases de datos funcionales y confiables. [33]

1.5.4.2 DBDesigner

DBDesigner, es un sistema totalmente visual de diseño de bases de datos, que combina características y funciones profesionales con un diseño simple, claro y fácil de usar, a fin de ofrecer un método efectivo para gestionar las bases de datos. Es una herramienta libre, licenciada bajo la GPL (del inglés: *General Public License*). [34]

1.5.4.3 Herramienta para el diseño de bases de datos seleccionada.

Para el diseño de las bases de datos se hará uso de la herramienta DBDesigner en su versión 4.0.5.6, herramienta de software libre que permite administrar la base de datos, diseñar tablas, hacer peticiones SQL manuales, guarda los proyectos como modelos XML (del inglés: *eXtensible Markup Language* o Lenguaje de marcas extensible) y posee soporte para la función drag-and-drop (arrastrar y soltar). El programa dispone además de una interfaz profesional y de detallados manuales de uso. [34]

1.5.5 Plataforma de Desarrollo.

1.5.5.1 PhpDesigner.

PhpDesigner, es un completo entorno de desarrollo y programación especialmente diseñado para los gurús



de PHP, aunque también permite trabajar con comodidad en otros lenguajes como HTML, XHTML, CSS y SQL. Posee completamiento de código. Tiene un corrector de sintaxis PHP online, que funciona a la vez que escribes código. Resulta muy sencillo de utilizar como un simple editor de PHP. [35]

1.5.5.2 Zend Studio.

Permite agilizar el desarrollo web y permite simplificar proyectos complejos. Posibilita la administración avanzada de proyectos, soporta múltiples lenguajes, incorpora el Framework de Zend, PHP Documentor, manual de PHP. Integración con subversión, los navegadores, integración avanzada con FTP. Soporte para Web Services, PHP4, PHP5 y SQL. [36]

1.5.5.3 Plataforma de desarrollo seleccionada.

Se utilizará como IDE, Zend Studio en su versión 7.0.1 como plataforma para el desarrollo del sistema, ya que es compatible con las plataformas Linux y Windows, e integrado para el lenguaje de programación PHP. Incluye editor, análisis, depuración, optimizadores de código y herramientas de base de datos. [36]

1.5.6 Servidores de correo.

1.5.6.1 Sendmail.

Sendmail es un Agente de transporte de correo (MTA, o Mail Transport Agent) diseñado para entregar correctamente todos los mensajes. Es confiable, puede ser configurado para manejar entornos complejos, incluyendo múltiples redes, comprueba el contenido de una dirección, así como su sintaxis para determinar cuál correo se utilizará. [37]

1.5.6.2 Postfix.

Postfix es un Agente de Transporte de Correo (MTA) de código abierto, o lo que es lo mismo, un programa informático para el enrutamiento y envío de correo electrónico. Fue creado como alternativa a Sendmail, buscando un servidor que fuera más rápido, fácil de administrar y seguro. [38]

Las principales características de Postfix son: [38]

- ✓ Soporte para TLS (Transport Layer Security).
- ✓ Soporte para mbox, maildir y dominios virtuales.
- ✓ Capacidad de manejar altos volúmenes de correo.
- ✓ Facilidad de configuración: Con los dos ficheros de configuración: main.cf y master.cf se controla todo, y además están muy bien explicados.



- ✓ Uso sencillo de listas negras.
- ✓ Abundante documentación, y de calidad.

1.5.6.3 Servidor de correo seleccionado.

Se utilizará como servidor de correo Postfix ya que permite la gestión de grandes cantidades de correo, es seguro, de uso gratuito, consume pocos recursos y posee rapidez de ejecución. Incluye soporte para TSL y posee abundante documentación.

1.5.7 Herramienta de gestión de módem GSM. Gammu 1.30.

Para el trabajo con mensajes SMS se utilizará un módem GSM como dispositivo físico, posibilita la recepción y manipulación de mensajes SMS. Se hará uso de la herramienta Gammu para la administración de las funcionalidades del módem GSM. La misma opera a nivel de consola y contiene una biblioteca para trabajar con teléfonos móviles de varias marcas. Permite su uso para la recuperación, copia de respaldo y envío de SMS, facilitando el manejo de llamadas. El programa puede funcionar con contactos, mensajes (SMS), calendario, listas de tareas, sistema de archivos, radio integrada, cámara, etc. Gammu cuenta con una serie de configuraciones que son especificadas en un archivo de lectura, en el cual se indica el puerto por el cual se va comunicar con el teléfono celular y los datos necesarios para conectarse con la base de datos y poder interactuar. [39] [40]

La herramienta Gammu en modo demonio¹, escanea periódicamente el módem GSM para capturar los mensajes recibidos y los mismos son almacenados en la base de datos definida, también envía hacia los celulares los mensajes que se encuentran en cola de almacenamiento. Es una herramienta perfecta para manejar grandes cantidades de mensajes enviados o recibidos y procesarlos automáticamente. Actualmente se pueden almacenar mensajes recibidos en varias bases de datos, donde MySQL está soportado por esta herramienta. [41]

1.5.8 Herramienta de control de versiones. SVN (Subversion) 1.6.

La herramienta **SVN (Subversion)** es un sistema de control de versiones de código abierto que permite el mantenimiento de versiones y revisiones antiguas en los archivos de un software, brinda la posibilidad de que varios usuarios en un equipo de desarrollo estén trabajando sobre un mismo archivo a través de la red

¹ Un demonio es un tipo especial de programa que se ejecuta forma continua (infinita) en segundo plano en vez de ser controlado directamente por el usuario.



y se contemplen sin bloqueos las modificaciones realizadas por cada uno, la recuperación de versiones anteriores de la información que se trata o examinar el historial de cambios de dicha información.

Las razones fundamentales para el uso de esta herramienta en el desarrollo de la aplicación es que la misma: [42]

- ✓ Gestiona las modificaciones durante el desarrollo del software.
- ✓ Permite que varias personas trabajen sobre los mismos ficheros.

1.6 Conclusiones Parciales.

En este capítulo se realizó un estudio sobre las principales tecnologías presentes en los sistemas paging, los protocolos utilizados para el envío de mensajes, así como las soluciones existentes en Cuba y el mundo que brindan el servicio de mensajería paging. Quedo definido para el desarrollo del sistema la utilización del CMS Joomla y lenguajes de programación PHP y Javascript. Como metodología de desarrollo se definió XP; las herramientas seleccionadas fueron servidor web Apache, sistema gestor de base de datos MySQL, herramienta para el diseño de la base de datos DBDesigner, plataforma de desarrollo Zend Studio, servidor de correo Postfix y para la gestión de mensajes SMS se definió la herramienta Gammu.



CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA, EXPLORACIÓN Y PLANIFICACIÓN

2.1 Introducción.

En este capítulo se realizará el modelamiento del sistema, planteándose la propuesta del mismo, se especificarán los requisitos no funcionales que debe cumplir el producto. Se describirá el objeto de automatización, y se definirán las historias de usuario con la estimación de esfuerzo necesario para implementar cada una de ellas.

2.2 Problemática.

Los servicios paging son de gran ayuda para muchas personas que necesitan mantenerse localizables. En Cuba las vías para el envío de mensajes paging son a través de Internet, correo electrónico o llamando al centro de operadoras, lo que obliga a requerir de una pc o teléfono fijo. Dado el avance de las tecnologías de la información y las telecomunicaciones, el uso de dispositivos móviles en Cuba se ha incrementado en gran medida, propiciando una vía de comunicación factible. Por lo que se puede aprovechar el auge de la telefonía móvil, en conjunto con las vías que brinda la empresa cubana Movitel para el envío de mensajes paging, como una excelente alternativa, para que todo aquel que posea un celular, Internet o correo electrónico, pueda mantener comunicación con personas que dispongan de un beeper, conllevando a un avance en el desarrollo de las telecomunicaciones del país y a un aumento en la usabilidad del dispositivo. Se propone entonces el desarrollo de un sistema que permita la convergencia de los medios de comunicación SMS, correo electrónico e Internet para facilitar el envío de mensajes paging, mejorando los servicios de comunicación que ya se brindan en Cuba y la universidad.

2.3 Objeto de automatización.

Se pretende automatizar el envío de mensajes paging a través de las vías de comunicación, SMS, Internet y correo electrónico, así como posibilitar la programación de mensajes para la fecha y hora deseada desde la web. Es necesario automatizar también la persistencia de mensajes, para que puedan ser leídos posteriormente por cada usuario registrado en el sistema.



2.4 Propuesta del sistema.

Se propone desarrollar un sistema que haga converger las distintas vías de comunicación (Internet, correo electrónico y SMS) para el envío de mensajes paging. Dispondrá de una aplicación web que podrá ser accedida por el usuario mediante una interfaz sencilla y amigable. La misma tendrá un sistema de autenticación basado en roles (RBAC del inglés: *Role Based Access Control*) donde el usuario después de autenticado podrá acceder a las operaciones permitidas según el rol que tenga asignado, garantizando que cada usuario tenga acceso solo a la información que le compete. Mediante la aplicación web el usuario podrá enviar mensajes paging y programar mensajes para la fecha y hora deseada, sin necesidad de estar autenticado en el sistema. La aplicación permitirá la gestión de mensajes, contactos, y gestionar las cuentas de usuarios, además de incluir el envío de mensajes desde SMS y correo electrónico.

Para el envío de mensajes paging a través SMS, se utilizará para recibirlos un módem GSM; y deberán ser enviados al número de la tarjeta sim² (del inglés: Subscriber Identity Module) del módem, con el siguiente formato: **@númeroBeeper textoMensaje**, ejemplo: @11111 Estamos en tesis.

Luego se hará uso de la herramienta Gammu en modo demonio, que se encarga de escanear periódicamente el módem GSM para capturar el mensaje recibido y almacenarlo en su base de datos, posteriormente se extrae el mensaje, se procede a enviarlo mediante correo electrónico utilizando el servicio de Movitel, y después es almacenado en la base de datos del sistema.

Para el envío de mensaje desde correo electrónico, deberá ser enviado a la siguiente dirección de correo **pager@pager.uci.cu**, poniendo en el texto del mensaje **@númeroBeeper textoMensaje**, ejemplo: @11111 Estamos en tesis.

Los mensajes serán recopilados del directorio de almacenamiento del correo, se extrae el texto del mensaje, se procede a enviarlo mediante correo electrónico utilizando el servicio de Movitel, y después es almacenado en la base de datos del sistema.

² Una tarjeta SIM (Subscriber Identity Module, en español Módulo de Identificación del Suscriptor), es una tarjeta desmontable usada en teléfonos móviles y módems. Almacenan de forma segura la clave de servicio del suscriptor usada para identificarse ante la red.

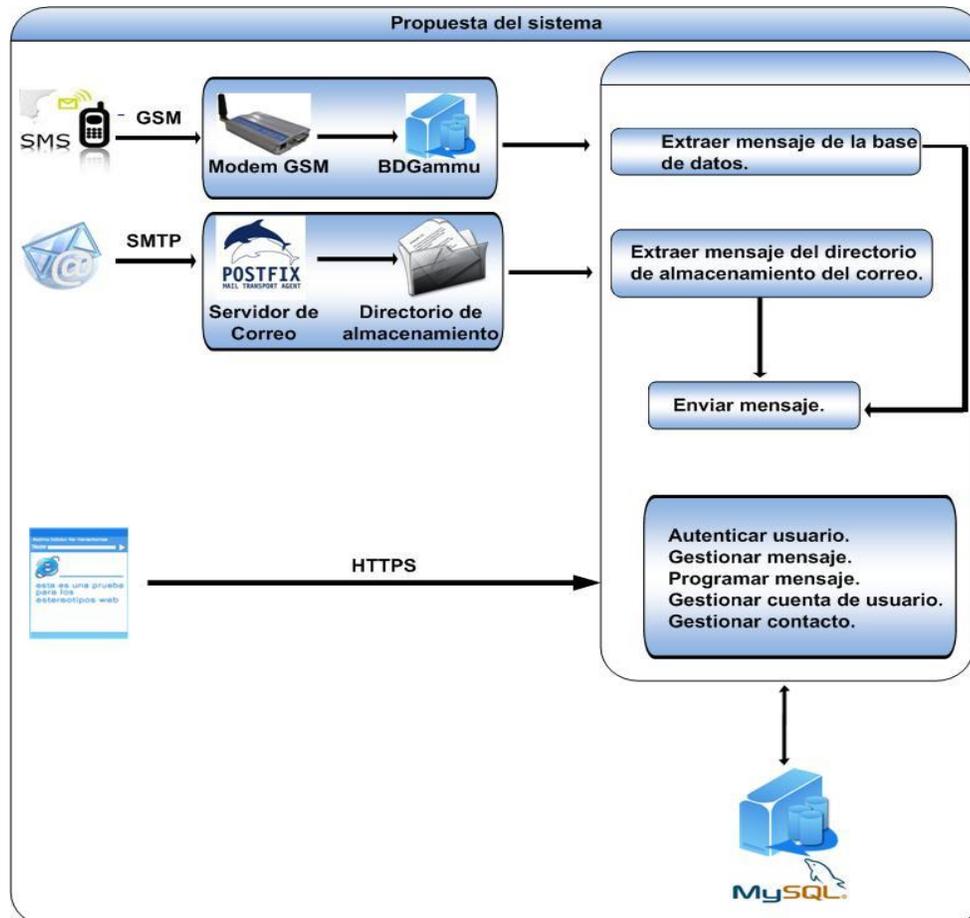


Figura 2: Propuesta del sistema.

2.5 Seguridad en el sistema.

Con el objetivo de garantizar la seguridad en el sistema, se instaló el componente Joomla Securitycheck, el cual realiza una comprobación de las versiones de todos los componentes instalados, comparándolos con su base de datos para mostrar los que son vulnerables y estables. Esta extensión evita tener que probar de forma individual cada componente para comprobar vulnerabilidades.

Securitycheck tiene dos características principales: un componente y un plugin. El plugin permite su configuración para la protección del sitio web, el componente muestra los registros capturados por el plugin y los componentes instalados en el sistema que sean vulnerables.

Capítulo 2: Características del sistema. Exploración y Planificación



La pantalla principal del componente brinda la siguiente información: los componentes instalados en el sitio web Joomla, se puede ver si uno de ellos es vulnerable a través de un código de colores fácil de interpretar: [43]

- ✓ Rojo: Vulnerable.
- ✓ Amarillo: Existe una vulnerabilidad para el componente pero no se especifica la versión de Joomla a la que afecta.
- ✓ Verde: No se le conocen vulnerabilidades.

Con el plugin se puede establecer un nivel de protección contra más de 90 tipos diferentes de Inyección SQL (del inglés: *Structured Query Language*), XSS (del inglés: *Cross-site scripting*), LFI (del inglés: *Local File Include*), RFI (del inglés: *Remote File Include*) y otros patrones de ataques. Algunas de las características que provee para la protección del sitio web son: [43]

- ✓ Lista negra y blanca: donde las negras son los IP's que serán rechazados y la blanca IP's a las cuales no se aplicarán los filtros.
- ✓ Registro de todos los ataques que se le puedan realizar al sitio, reporte que se aprecia en los nuevos registros (Logs), que pueden ser vistos por los administradores de CMS, donde los logs reportan el ip de donde se accedió y la fecha y hora de acceso.
- ✓ Redirección a una página por defecto si se detecta un ataque.
- ✓ Protección de sesiones, donde si un usuario se intenta loguear por una sección que se encuentra activa, automáticamente deslogue al otro usuario en forma de aviso de que alguien intenta iniciar su sección, y redirecciona a una página de error al segundo usuario.

Otras técnicas utilizadas para la protección del sitio fueron:

- ✓ La utilización del protocolo seguro HTTPS (del inglés: *Hypertext Transfer Protocol Secure*), para asegurar la integridad y la confidencialidad de los datos en el intercambio información entre el usuario y la aplicación.
- ✓ La utilización CAPTCHAs, un captcha es una imagen de confirmación humana, que sirve para saber que es una persona la que está escribiendo un formulario y no una máquina. [44] Se utilizó con fin de proteger el sitio contra los ataques de denegación de servicios, para evitar el abuso por programas automatizados.



- ✓ La utilización de una palabra clave, como parámetro para crearse una cuenta, y para solicitar una nueva contraseña.
- ✓ Se aplicó que pasado 15 minutos de inactividad del usuario, se cierre la sección.

2.6 Personas relacionadas con el sistema.

Se define como persona relacionada con el sistema aquel que obtiene un resultado de la ejecución de uno o varios procesos en el mismo y aquellos que se encuentran involucrados en la administración de dichos procesos. Las personas relacionadas con el sistema son:

Tabla 1: Personas relacionadas con el sistema.

Personas relacionadas con el sistema.	Justificación.
Invitado.	Es la persona que navega por el sistema sin haber creado una cuenta en el mismo, trabaja en este sin acceder a los recursos definidos para usuarios registrados.
Usuario autenticado.	Es la persona que se encuentra autenticado en el sitio, que tendrá acceso a diferentes recursos debido al rol que cumple dentro de la aplicación.
Operadora.	Es la persona encargada de gestionar los mensajes, así como gestionar los usuarios registrados en el sitio.
Administrador.	Es la persona que dispone de posibilidades ilimitadas para ejecutar las funciones administrativas del sistema.

2.7 Especificación de requisitos no funcionales del sistema.

Los requisitos son capacidades o condiciones que el sistema debe tener para satisfacer un contrato o documento formal. Dentro de ella están las acciones que podrán ser ejecutadas por el usuario, las operaciones ocultas que debe realizar el sistema y las características determinadas para el funcionamiento del sistema.



2.7.1 Requisitos no funcionales.

Los requisitos no funcionales son cualidades o propiedades que debe tener el producto con el fin de hacerlo confiable, usable, rápido y atractivo. Los presentes en el sistema son:

Apariencia o interfaz externa.

La aplicación debe tener una interfaz sencilla y amigable, fácil de utilizar por el usuario, que le permita al mismo trabajar de forma cómoda y ágil.

Usabilidad.

La aplicación gracias al ambiente sencillo y fácil de usar que posee podrá ser utilizada por usuarios con conocimientos básicos en computación o sistemas informáticos.

Confidencialidad.

Ningún usuario no registrado accederá a la información almacenada, procurando así que no ocurra ningún cambio ni divulgación de la misma y que los datos sean confiables.

Soporte.

El software deberá ser actualizado por los administradores y se debe garantizar el mantenimiento de la aplicación.

Software.

El sistema requiere para su funcionamiento:

Por parte del cliente:

Sistema operativo Windows (98 o superior) o Linux. Navegador web con capacidad de interpretación de JavaScript y CSS, compatible con Internet Explorer o Mozilla Firefox.

Por parte del servidor:

Servidor web Apache 2.0.53. Servidor de base de datos MySQL 5.1.37.

Hardware.

Por parte del cliente:

Las PC clientes deberán contar como mínimo con 128 MB de memoria RAM, con velocidad de procesamiento 450MHz o superior.

Por parte del servidor:

La PC servidora deberá contar como mínimo con 512 MB de memoria RAM, con velocidad de procesamiento 2GHz o superior y poseer un módem GSM.



2.8 Fase de Exploración y Planificación.

2.8.1 Exploración.

La metodología XP define como primera, la fase de exploración. Esta es la fase donde los clientes plantean a grandes rasgos su propuesta de trabajo, al mismo tiempo el equipo de desarrollo se familiarizará con las herramientas, tecnologías y prácticas que se utilizarán en el proyecto. Es donde queda definido el alcance real del proyecto y sobre todo se definen las Historias de Usuario (en lo adelante HU), las cuales describen brevemente las características que el sistema debe poseer y el cliente establece la prioridad a cada HU.

2.8.1.1 Historias de Usuario.

Las Historias de Usuario son la técnica utilizada por XP para especificar las características y funcionalidades requeridas para el sistema que se desarrollará, las mismas son redactadas por el cliente, asignándole una prioridad o valor a cada HU, éstas no deben ser descritas en más de tres líneas, el contenido que ellas abarcan debe ser concreto y sencillo, escritas en el lenguaje del usuario. El equipo de desarrollo evalúa cada HU y le asigna un costo, el cual se mide en semanas de desarrollo. A medida que avance el desarrollo del sistema, el cliente puede agregar HU, modificar el valor de historias existentes, dividir las o eliminarlas. [45]

Clasificación de la prioridad de las HU.

Alta: Se le otorga esta prioridad a las HU que el cliente define como funcionalidades principales en el desarrollo del sistema y fundamentales para control general del sistema.

Media: Se le otorga a las HU que el cliente define como funcionalidades menos significativas, sin que afecten el desarrollo del sistema.

Baja: Se le otorga a las HU que se definen como funcionalidades de ayuda a la estructura y control de elementos asociados al desarrollo del sistema.

Las HU se representarán mediante tablas divididas por las siguientes secciones:

- ✓ Número de la HU.
- ✓ Nombre de la HU.
- ✓ Usuario involucrado en el desarrollo de la HU.
- ✓ Prioridad de la HU.



- ✓ Complejidad de desarrollo.
- ✓ Estimación del tiempo de desarrollo de la HU.
- ✓ Iteración en la que será desarrollada la HU.
- ✓ Descripción de la HU.
- ✓ Observaciones realizadas a la HU.
- ✓ Prototipo de interfaz si se aplica.

Las Historias de Usuario de prioridad media y baja se pueden consultar en el Anexo 1.

Tabla 2: Historias de Usuarios.

No: 1.	Nombre: Autenticar usuario.
Usuarios:	Otoniel Berroa Arias, Yanileidis Luna Solis
Prioridad: Alta	Complejidad: alta
Estimación: 1	Iteración: 1
Descripción:	Permite al usuario autenticarse en el sitio.
Observaciones:	
Prototipo de Interfaz:	



No: 2.	Nombre: Registrar usuario.
Usuarios:	Otoniel Berroa Arias, Yanileidis Luna Solis
Prioridad: Alta	Complejidad: media
Estimación: 1	Iteración: 1
Descripción:	Permite al usuario registrarse en el sistema, para poder acceder a los servicios de los usuarios autenticados.
Observaciones:	
Prototipo de Interfaz:	
No: 7.	Nombre: Enviar mensaje desde la web.
Usuarios:	Otoniel Berroa Arias, Yanileidis Luna Solis
Prioridad: Alta	Complejidad: alta
Estimación: 1	Iteración: 1
Descripción:	El usuario podrá enviar mensajes paging facilitando la comunicación con otras personas.
Observaciones:	



Prototipo de Interfaz:	
No: 8.	Nombre: Enviar mensaje a través de SMS.
Usuarios:	Otoniel Berroa Arias, Yanileidis Luna Solis
Prioridad: Alta	Complejidad: alta
Estimación: 2	Iteración: 1
Descripción:	El usuario envía un SMS al número de sim del módem GSM, con el formato siguiente: @númeroPager textoMensaje. Ejemplo: @11111 Estamos en tesis.
Observaciones:	
Prototipo de Interfaz:	
No: 9.	Nombre: Enviar mensaje a través de correo electrónico.
Usuarios:	Otoniel Berroa Arias, Yanileidis Luna Solis
Prioridad: Alta	Complejidad: alta
Estimación: 2	Iteración: 1
Descripción:	El usuario envía un mensaje al correo definido con el formato siguiente: pager@pager.uci.cu , poniendo el texto del mensaje @númeroPager textoMensaje. Ejemplo: @11111 Estamos en tesis.
Observaciones:	

Capítulo 2: Características del sistema. Exploración y Planificación



Prototipo de Interfaz:									
No: 10.	Nombre: Listar mensajes.								
Usuarios:	Otoniel Berroa Arias, Yanileidis Luna Solis								
Prioridad: Alta	Complejidad: media								
Estimación: 2	Iteración: 1								
Descripción:	El usuario podrá leer sus mensajes paging desde su buzón privado.								
Observaciones:									
Prototipo de Interfaz:	<table border="1"> <thead> <tr> <th>No. </th> <th>De</th> <th>Recibido</th> <th>Mensaje</th> </tr> </thead> <tbody> <tr> <td>1 </td> <td>11111</td> <td>06/02/2012</td> <td>Prueba</td> </tr> </tbody> </table>	No.	De	Recibido	Mensaje	1	11111	06/02/2012	Prueba
No.	De	Recibido	Mensaje						
1	11111	06/02/2012	Prueba						
No: 12.	Nombre: Programar mensajes para la fecha y hora deseada								
Usuarios:	Otoniel Berroa Arias, Yanileidis Luna Solis								
Prioridad: Alta	Complejidad: alta								
Estimación: 2	Iteración: 1								
Descripción:	El usuario podrá programar mensajes para la fecha y hora que desee.								
Observaciones:									



Prototipo de Interfaz:

MENSAJE AUTOMÁTICO

Pager ID:* (5 dígitos)

Mensaje:* (120 caracteres)

Caracteres:*

Fecha Inicial* Hora

Fecha Final*

June 2012

Su	Mo	Tu	We	Th	Fr	Sa
					1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30

2.8.2 Planificación.

La planificación es la fase que plantea XP como un permanente diálogo entre el cliente y los desarrolladores del proyecto. En esta fase los desarrolladores realizan una estimación del esfuerzo que costará implementar de cada una de las HU. Luego de realizar la estimación, se organizan las HU en las iteraciones correspondiente, según la prioridad establecida por el cliente y el tiempo de desarrollo de cada una.

2.8.2.1 Estimación de esfuerzo por Historias de Usuario.

Tabla 3: Estimación de esfuerzo por HU.

Historias de Usuario.	Estimación (semanas).
Autenticar usuario.	1
Registrar usuario.	1
Listar usuarios.	1
Modificar cuenta de usuario.	1
Asignar rol al usuario.	1



Eliminar cuenta de usuario.	1
Enviar mensaje desde la web.	1
Enviar mensaje a través de SMS.	2
Enviar mensaje a través de correo electrónico.	2
Listar mensajes.	2
Eliminar mensaje.	1
Programar mensaje para la fecha y hora deseada.	2
Insertar contacto.	1
Listar contactos.	1
Modificar contacto.	1
Eliminar contacto.	1
Crear cuenta de usuario	1

2.8.2.2 Plan de iteraciones.

Luego de ser identificadas las HU y haber realizado la estimación de esfuerzo de cada una de ellas, se procede a realizar la planificación de la etapa de implementación del sistema, elaborando el plan de iteraciones. En el plan se especifica la prioridad con que se irán desarrollando las HU organizadas por iteraciones.

Una iteración no es más que un mini-proyecto donde se obtiene un resultado del software con valor para el cliente. Es preciso aclarar que no se obtiene la versión completa del producto, ya que concluye al finalizar la última iteración.

Iteración 1.

Esta iteración tiene como objetivo la implementación de las HU de prioridad alta y prioridad media. Propiciando esta iteración al finalizar, tener implementadas las peticiones del cliente descritas en las HU, contando con las funcionalidades principales para el funcionamiento del sistema.



Iteración 2.

En esta iteración serán implementadas las HU de prioridad baja, las mismas tienen como propósito brindarles comodidad al cliente en la gestión de otras tareas asociadas a las de prioridad alta. Al terminar esta iteración se obtendrá una versión 1.0 del producto final, listo para poner en funcionamiento y ser evaluado.

2.8.2.3 Plan de duración de las iteraciones.

Continuando con el desarrollo de la metodología XP, se pasa a crear el plan de duración de las iteraciones. El plan se encarga de detallar el orden en que se implementarán las HU dentro de cada iteración, así como la estimación completa de la misma.

Tabla 4: Plan de duración de las iteraciones.

Iteraciones.	Orden de las HU a implementar.	Duración de las iteraciones.
Iteración 1.	Autenticar usuario. Registrar usuario. Listar usuarios. Modificar cuenta de usuario. Asignar rol al usuario. Eliminar cuenta de usuario. Enviar mensaje desde la web. Enviar mensaje a través de SMS. Enviar mensaje a través de correo electrónico. Listar mensajes. Eliminar mensaje. Programar mensaje para la fecha y hora deseada.	16 semanas.
Iteración 2.	Insertar contacto. Listar contactos. Modificar contacto. Eliminar contacto.	5 semanas.



	Crear cuenta de usuario.	
--	--------------------------	--

2.8.2.4 Plan de entregas.

El plan de entregas establece que HU serán agrupadas para conformar una entrega, y el orden de las mismas, así como la fecha fin de cada iteración. El plan acopla las funcionalidades referentes a un mismo tema en el módulo que serán implementadas.

Tabla 5: Plan de duración de entregas.

Historia de Usuario.	Fin de la 1ra iteración. 4ta semana de abril	Fin de la 2ra iteración. 2da semana de mayo
Autenticar usuario.	V 0.1	Finalizado
Registrar usuario.	V 0.1	Finalizado
Listar usuarios.	V 0.1	Finalizado
Modificar cuenta de usuario.	V 0.1	Finalizado
Asignar rol al usuario.	V 0.1	Finalizado
Eliminar cuenta de usuario.	V 0.1	Finalizado
Enviar mensaje desde la web.	V 0.1	Finalizado
Enviar mensaje a través de SMS.	V 0.1	Finalizado
Enviar mensaje a través de correo electrónico.	V 0.1	Finalizado
Listar mensajes.	V 0.1	Finalizado
Eliminar mensaje.	V 0.1	Finalizado
Programar mensaje para la fecha y hora deseada.	V 0.1	Finalizado
Insertar contacto.	-	V 0.1
Listar contactos.	-	V 0.1
Modificar contacto.	-	V 0.1
Eliminar contacto.	-	V 0.1
Crear cuenta de usuario.	-	V 0.1



2.9 Conclusiones Parciales.

En este capítulo quedó detallada la propuesta del sistema y los requerimientos no funcionales del sistema. Se describieron las HU que son los artefactos generados en la fase exploración y planificación de la metodología XP, se realizó una estimación de esfuerzo por cada una de las historias, y quedó definido el plan de duración de entregas.



Capítulo 3: Diseño del sistema

CAPÍTULO 3: DISEÑO DEL SISTEMA

3.1 Introducción.

En este capítulo se describe la fase de diseño propia de la metodología de desarrollo XP, donde el diseño ofrece una guía de implementación para una historia de usuario, se definen los patrones de arquitectura, el estilo arquitectónico y los patrones de diseño a utilizar para la implementación del sistema. Se procederá al diseño y descripción de la base de datos. Quedan definidas las tarjetas CRC (Colaborador - Responsabilidad – Clase) como técnica de diseño.

3.2 Estilo arquitectónico cliente-servidor.

La arquitectura cliente-servidor es un modelo en el que las transacciones se dividen en procesos independientes que cooperan entre sí para intercambiar información, servicios o recursos. Admite al usuario en una máquina llamada cliente, requerir algún tipo de servicio de una máquina a la que está conectado, llamado servidor, mediante una red de área local o una red de área mundial.

Se denomina cliente al proceso que inicia el diálogo o solicita los recursos y servidor al proceso que responde a las solicitudes. En este modelo las aplicaciones se dividen de forma que el servidor contiene la parte que debe ser compartida por varios usuarios, y en el cliente permanece solo lo particular de cada usuario.

Los servicios solicitados pueden ser peticiones de datos de una base de datos, de información contenida en archivos o los archivos en sí mismos. Este tipo de arquitectura es la base para todas las aplicaciones que hoy se pueden encontrar en Internet y además, en la mayoría de las Intranet corporativas que hoy existen en muchas de las empresas. [46]

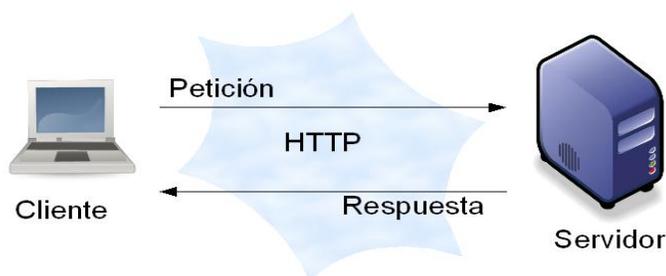


Figura 3: Arquitectura cliente-servidor. Fuente [47].



3.3 Patrón de arquitectura. Modelo-Vista-Controlador (MVC)

Los patrones arquitectónicos expresan un esquema organizativo estructural fundamental para sistemas de software a desarrollar. Se seguirá para desarrollar el sistema el patrón de arquitectura para aplicaciones web que sigue el CMS Joomla, Modelo-Vista-Controlador.

Modelo-Vista-Controlador (en adelante denominado como MVC) es un patrón de arquitectura de software que se puede utilizar para organizar el código de tal manera que la lógica de negocios y presentación de datos están separados. Las tres funciones son la base para el MVC Joomla dentro de un componente, describiendo su estructura a continuación.

El Modelo: es la parte del componente que encapsula los datos de la aplicación. Proporcionan rutinas para manejar y recuperar datos en una forma significativa.

La vista: es la parte del componente que se utiliza para representar los datos del modelo. Para una aplicación basada en web, el punto de vista general, sería una página HTML que se devuelve al usuario. La vista extrae los datos del modelo (que se pasa a ella desde el controlador) y alimenta los datos en una plantilla que se rellena y presenta al usuario, mostrando los datos recuperados a partir del modelo.

El controlador: se encarga de responder a las acciones del usuario, determinará lo que se realiza en una solicitud hecha por el usuario y responde de forma adecuada mediante la activación del modelo para manipular los datos adecuadamente. El controlador realiza los cambios apropiados en el modelo o en la vista, y a continuación muestra los datos de la modelo en la vista. [48]

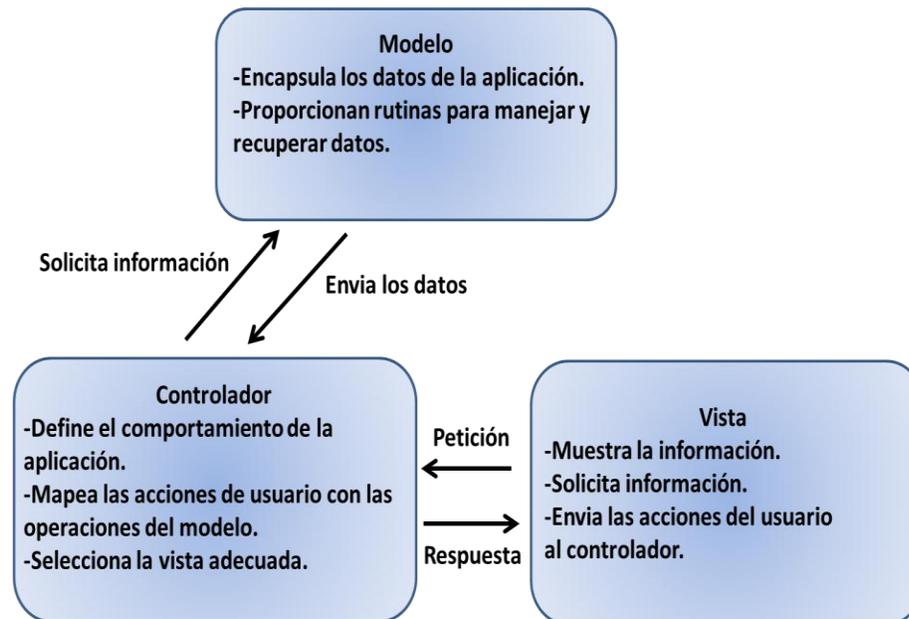


Figura 4: El patrón MVC. Fuente[49].

3.4 Arquitectura del sistema.

La arquitectura del sistema se estructura de manera general con un modelo de aplicación distribuida Cliente-Servidor, y en su componente servidor se aplica el estilo arquitectónico Llamada y Retorno, donde se ven presentes los patrones arquitectónicos Modelo Vista Controlador y Arquitectura en Capas.

En la Arquitectura en Capas la información fluye a través de las diferentes capas, donde la lógica de aplicación se reparte en diferentes capas ubicadas entre el cliente y los datos. Cada capa proporciona servicios a la capa inmediatamente superior y se sirve de las prestaciones que le brinda la inmediatamente inferior; donde el patrón Modelo Vista Controlador está presente en la capa superior, controlando el flujo de información obtenida a partir de las vistas.

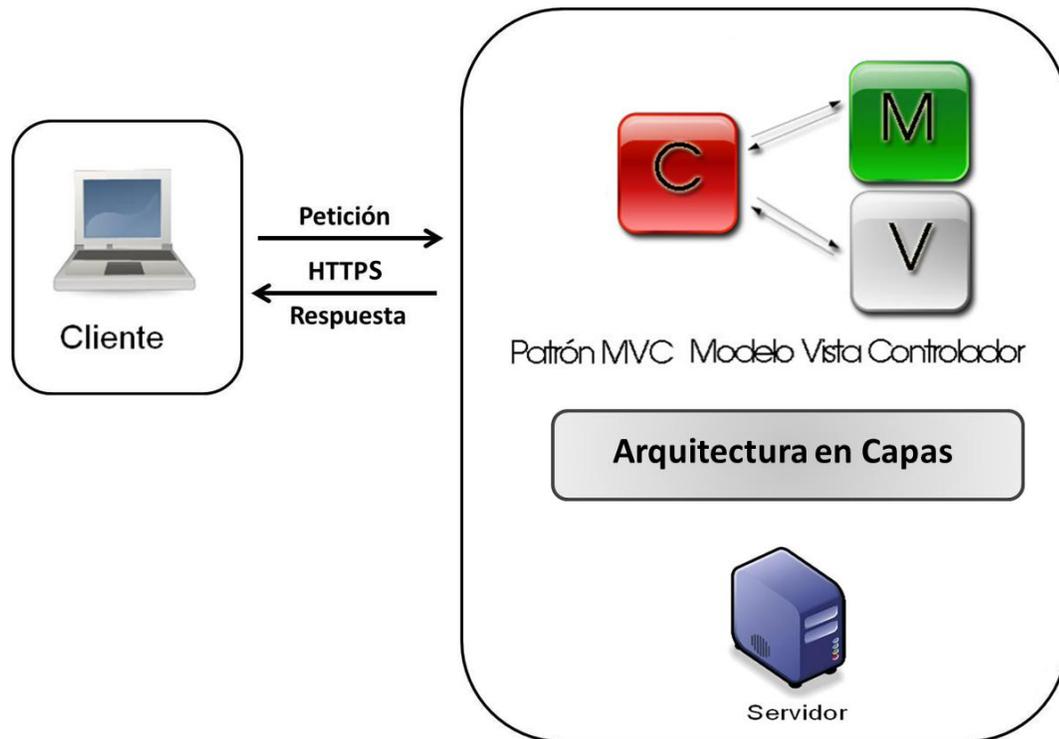


Figura 5: Arquitectura del sistema.

La integración de las arquitecturas en el desarrollo del sistema proporciona las siguientes ventajas:

- ✓ Mayor flexibilidad al añadir nuevos componentes para otorgar al sistema nuevas funcionalidades.
- ✓ Posibilita los recursos centralizados, debido a que el servidor es el centro de la red; se puede administrar los recursos que son comunes a todos los usuarios.
- ✓ Facilita un mantenimiento y soporte más sencillo.
- ✓ Los datos y los servicios web aparecen separados.
- ✓ Facilidad para separar datos de la lógica de negocio.
- ✓ El usuario recibe los datos y la información de forma indirecta a través del servidor.

3.5 Patrones de diseño.

Los patrones de diseño describen un problema que ocurre en un contexto, así como la solución del mismo, proporcionando la reutilización de esta solución. El uso de los patrones de diseño favorece desarrollar sistemas más robustos y fáciles de mantener. Permite además estandarizar el modo en que se va a realizar el diseño y formalizar un vocabulario entre diseñadores.



Capítulo 3: Diseño del sistema

Para el desarrollo de la aplicación se utilizaron los patrones de diseño siguientes:

3.5.1 Patrones GRASP (*General Responsibility Assignment Software Patterns*).

3.5.1.1 Experto.

Cumple con el principio básico de asignar responsabilidades a las clases que cuentan con la información necesaria para cumplirlas. Proporcionando con su uso un sistema fácil de mantener y de reutilizar. [25]

La figura 6 expone un ejemplo de cómo se encuentra presente el patrón experto en el sistema, la clase UsuarioDAO contará con la información necesaria para cumplir cada una de las responsabilidades implementadas para el acceso a los datos. La clase en cuestión contiene métodos para añadir, eliminar, leer y actualizar la información del usuario en la base de datos, incluirá además los datos necesarios para realizar las acciones establecidas.

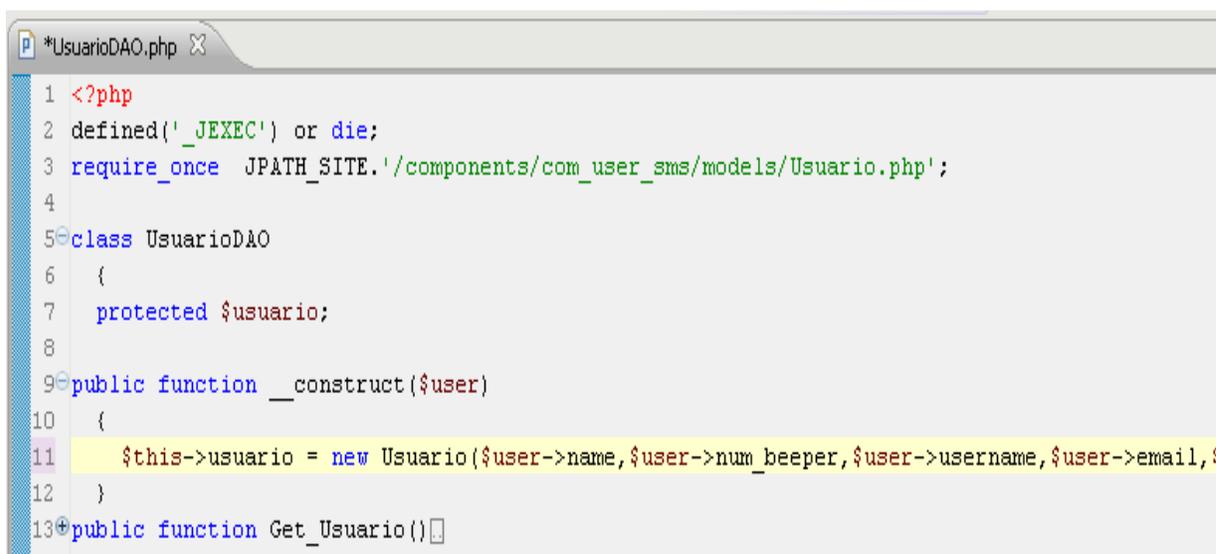
```
UsuarioDAO.php X
1 <?php
2 defined('_JEXEC') or die;
3 require_once JPATH_SITE.'/components/com_user_sms/models/Usuario.php';
4
5 class UsuarioDAO
6 {
7     protected $usuario;
8
9     public function __construct($user) {}
13    public function Get_Usuario() {}
15
16    public function InsertarUsuario($name,$num_beeper,$username,$email,$password,$usertype,$block,
20
21    public function ModificarUsuario($id,$name,$num_beeper,$username,$email,$password,$usertype,$b
25
26    public function EliminarUsuario($id) {}
31
32    public function BuscarUsuario($id) {}
38
39    public function BuscarUsuarioBeeper($num_beeper) {}
45
46    public function BuscarUsuarioTodos() {}
51
52    public function BuscarUsuarioRoll($user_id) {}
58
59    public function Asignar_Grupo($id,$rol) {}
63
64    public function CantidadUser() {}
69 }
70 ?>
```

Figura 6: Muestra del patrón Experto en el sistema.

3.5.1.2 Creador.

Indica que clase es responsable de crear una nueva instancia de otra clase. Su uso en el desarrollo de la aplicación brinda mejores oportunidades de reutilización y mantenimiento. [25]

La figura 7 expone un ejemplo de cómo se encuentra presente el patrón creador en el sistema, la clase UsuarioDAO se encarga de crear instancias de la clase Usuario que serán necesarias en el transcurso de la implementación para realizar su función. Donde la instancia creada le proporciona a la clase encargada de crearla facilidad para trabajar con los datos del usuario.



```
*UsuarioDAO.php
1 <?php
2 defined('_JEXEC') or die;
3 require_once JPATH_SITE.'/components/com_user_sms/models/Usuario.php';
4
5 class UsuarioDAO
6 {
7     protected $usuario;
8
9     public function __construct($user)
10    {
11        $this->usuario = new Usuario($user->name,$user->num_beeper,$user->username,$user->email,$
12    }
13    public function Get_Usuario()
```

Figura 7: Muestra del patrón Creador en el sistema.

3.5.2 Patrones GOF (Gang of Four).

3.5.2.1 Factory.

El **patrón de la fábrica (Factory)** es un patrón que se utiliza para construir y devolver objetos. Joomla nos proporciona la clase estática JFactory, que implementa el patrón fábrica. Esta clase es importante porque permite acceder de forma sencilla a instancias de los objetos globales. [50]

El ejemplo muestra cómo se puede acceder a algunos de los objetos globales utilizando JFactory en la clase:



```
link_user.php X
1 <?php
2 defined('_JEXEC') or die;
3 session_start();
4 require_once JPATH_SITE.'/components/com_user_sms/controller/user_controller.php';
5 $user =& JFactory::getUser();//usuario
```

Figura 8: Muestra del patrón Factory en el sistema.

3.6 Diseño del sistema.

Para el diseño de la aplicación y con el objetivo de una mejor comprensión del trabajo se realizará una breve explicación del funcionamiento del CMS Joomla.

En el CMS se muestran los contenidos de forma que los mismos puedan ser utilizados y modificados por cualquier desarrollador, tienen propiedades tales como título, texto introductorio y completo, autor, fecha de creación entre otros atributos, proporcionando a su vez Joomla los mecanismos para la creación, edición y publicación de contenidos.

Joomla proporciona un sistema de seguridad, se encarga además de la creación de usuarios, así como del control de acceso a los diferentes componentes según los permisos que sean definidos por el administrador.

3.6.1 Componentes de Joomla.

Los componentes son extensiones que amplían las funcionalidades del sistema que se esté desarrollando. Son pequeñas aplicaciones independientes entre sí, que gestionan la información, añaden nuevas funcionalidades que realizan una función específica, con disímiles propósitos dependiendo del objetivo con que se desarrollan.

Un componente consta de uno o más ficheros entre los cuales pueden estar ficheros php, js, y css, todo de acuerdo a la funcionalidad que se vaya a implementar. Los ficheros generalmente se encargarán de mostrar la interfaz visual al usuario donde su trabajo es actuar como enlace a todo el componente a la hora de construir la página web y gestionar el contenido. [51]

A continuación se mostrarán los componentes del sistema en cuestión:

Componentes:

com_user_sms.

com_send_sms.

com_search_sms.



Capítulo 3: Diseño del sistema

com_contact_sms.

com_admin_sms.

Los componentes descritos anteriormente dan soporte a distintas funcionalidades de Joomla, además de los desarrollados en el trabajo. Se modelará cada componente por separado, puesto que son independientes uno del otro en cuanto a funcionalidad. La forma de diseño y organización que se utilizará es diseñar una tarjeta CRC por cada uno de los componentes que brindan una funcionalidad directa al negocio.

3.7 Tarjetas Colaborador - Responsabilidad – Clase (CRC).

La fase de diseño en la metodología XP sigue de manera rigurosa el principio de mantenerlo simple, como parte de la misma se propone el modelado de las tarjetas CRC, proporcionando un medio sencillo para identificar y organizar las clases relevantes para los requisitos del sistema. Donde el objetivo de las tarjetas es desarrollar una representación organizada de las clases.

Las responsabilidades son los atributos y las operaciones relevantes para la clase, de manera más simple una responsabilidad es cualquier cosa que una clase sabe o hace. Los colaboradores son aquellas clases que se requieren para que una clase reciba la información necesaria para completar una responsabilidad, implicando en general ya sea una solicitud de información o de alguna acción. [45]

Las tarjetas CRC están divididas en tres secciones:

- ✓ Nombre del componente.
- ✓ Responsabilidades.
- ✓ Colaboradores.

Tabla 6: Tarjetas CRC.

Componente com_user_sms.	
Responsabilidad.	Colaborador.
Autenticar.	UsuarioDAO
Registrar usuario.	
Modificar cuenta de usuario.	
Eliminar cuenta de usuario.	



Componente com_send_sms.	
Responsabilidad.	Colaborador.
Enviar mensaje desde la web.	sms_controller
Programar mensaje para fecha y hora deseada.	
Componente com_search_sms.	
Responsabilidad.	Colaborador.
Listar mensajes.	UsuarioDAO.
Eliminar mensaje.	MensajeDAO
Componente com_contact_sms.	
Responsabilidad.	Colaborador.
Insertar contacto.	UsuarioDAO ContactoDAO
Listar contactos.	
Modificar contacto.	
Eliminar contacto.	
Componente com_admin_sms.	
Responsabilidad.	Colaborador.
Asignar rol al usuario.	UsuarioDAO, MensajeDAO
Listar usuarios.	ContactoDAO, user_controller
Crear cuenta de usuario.	

3.8 Diseño de la base de datos.

El diseño de la base de datos es una de las tareas más importantes en la construcción de un sistema que la utilice. Es fundamental, puesto que tiene como objetivo brindar la persistencia al modelado descrito anteriormente, elaborándose con el fin de acceder a la información de manera eficiente. Quedando definido el modelo de datos siguiente:

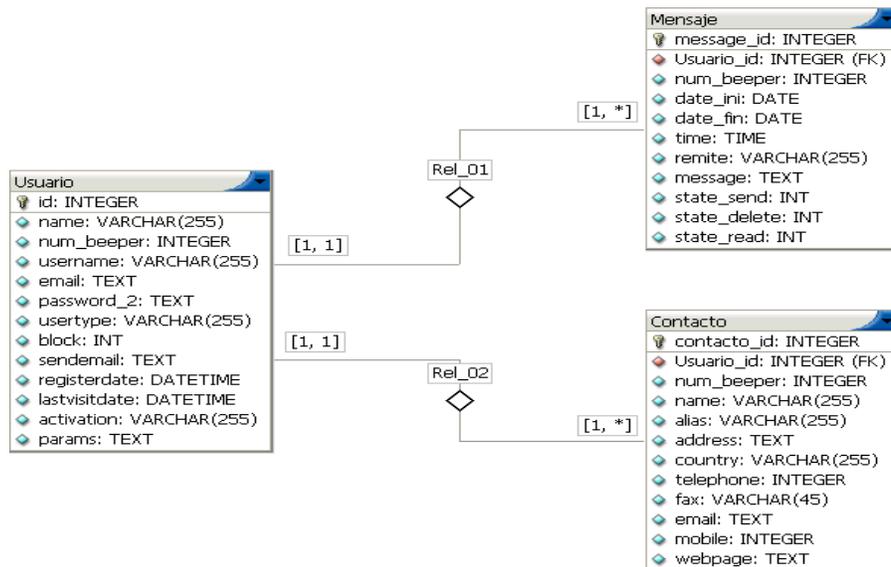


Figura 9: Modelo de datos.

La entidad **Mensaje** se encarga de almacenar los datos del mensaje enviado, los datos almacenados serán el id del mensaje de tipo numérico, el número de beeper a quien se lo envían, la fecha y hora de envío, el remitente, el texto del mensaje y el estado en que se encuentra el mensaje (leído, borrado o guardado). Dicha entidad posee una relación con la entidad Usuario, propiciando que cada usuario tenga una asociación con sus mensajes.

La entidad **Usuario** tiene la responsabilidad de almacenar los datos del usuario registrado en el sistema. Almacenándose el id del usuario, el nombre, número de beeper, correo electrónico si posee, la contraseña encriptada, la fecha de registro y la palabra secreta.

La entidad **Contacto** se encarga de guardar la información del contacto, almacena el id del contacto, el número de beeper, nombre, alias, dirección, teléfono, correo electrónico, y celular si posee alguno. Debido a la relación que existe con la entidad Usuario, se logra que existan contactos relacionados con usuarios, que los agregaron a su lista en un momento determinado.



Capítulo 3: Diseño del sistema

3.9 Conclusiones Parciales.

En este capítulo quedaron definidos los patrones de arquitectura y estilo arquitectónico presente para el desarrollo del sistema, detallándose además los patrones de diseño aplicados. También quedaron definidos los componentes del sistema en cuestión, así como las tarjetas CRC que son los artefactos generados en la fase de diseño de la metodología XP. Realizándose además el diseño y descripción de la base de datos.



Capítulo 4: Implementación y Prueba

CAPÍTULO 4: IMPLEMENTACIÓN Y PRUEBA

4.1 Introducción.

En este capítulo se describen las fases de implementación y prueba propias de la metodología XP. Se detallan las iteraciones planteadas en la fase de Planificación, exponiéndose las tareas de la ingeniería generadas por cada HU, se representa además el diagrama de despliegue, así como las pruebas de aceptación efectuadas sobre el sistema.

4.2 Fase de Implementación.

La metodología XP plantea que la implementación de un software debe realizarse de forma iterativa, obteniendo al finalizar cada iteración un producto funcional que debe ser probado y mostrado al cliente. En esta fase se realiza la implementación de cada una de los HU durante el transcurso de las iteraciones, creándose las tareas de la ingeniería para ayudar a organizar la implementación exitosa de la HU, las cuales son asignadas a los programadores que son los responsables de la implementación. Las tareas pueden escribirse en un lenguaje técnico, al contrario de las HU que son escritas en el lenguaje del cliente. [45]

Las tareas de la ingeniería serán representadas mediante tablas divididas por las siguientes secciones:

- ✓ Número de la tarea.
- ✓ Número de la historia de usuario a la que pertenece la tarea.
- ✓ Nombre de la tarea.
- ✓ Tipo de la tarea, puede ser de Desarrollo, Corrección, Mejora u Otra (especificar).
- ✓ Puntos de estimación, que son el tiempo estimado que se le asignará para su desarrollo.
- ✓ Fecha de inicio del desarrollo de la tarea.
- ✓ Fecha de fin del desarrollo de la tarea.
- ✓ Programador responsable.
- ✓ Descripción de la tarea.

4.2.1 Iteraciones.

Teniendo en cuenta la planificación realizada anteriormente, donde quedaron definidas 2 iteraciones de desarrollo sobre el sistema, con el objetivo de obtener un producto con todas las características deseadas para ser utilizado. Detallándose cada una de las iteraciones a continuación.



4.2.1.1 Primera iteración.

En la 1ra iteración se implementaron las historias de usuario de prioridad alta y media, con el fin de obtener una versión del producto que cuente con las funcionalidades críticas para el funcionamiento del sistema. A continuación se detallan los componentes abordados en la primera iteración.

Tabla 7: Componentes abordados en la 1ra iteración.

Componente.	Historia de Usuario.	Tiempo de implementación.	
		Estimación.	Real.
com_user_sms.	Autenticar usuario.	1	3/6
	Registrar usuario.	1	6/6
	Modificar cuenta de usuario.	1	3/6
	Eliminar cuenta de usuario.	1	2/6
com_send_sms.	Enviar mensaje desde la web.	1	6/6
	Programar mensaje para la fecha y hora deseada.	2	6/12
com_search_sms.	Listar mensajes.	2	6/12
	Eliminar mensaje.	1	2/6
com_admin_sms.	Asignar rol al usuario.	1	1/6
	Listar usuarios.	1	3/6

4.2.1.2 Segunda iteración.

En la 2da iteración se implementaron las historias de usuario de prioridad baja, con el fin de brindarle comodidad al cliente en la gestión de otras tareas asociadas a las de prioridad alta. A continuación se detallan los componentes abordados en la segunda iteración.

Tabla 8: Componentes abordados en la 2da iteración.

Componente.	Historia de Usuario.	Tiempo de implementación.	
		Estimación.	Real.
com_contact_sms.	Insertar contacto.	1	2/6
	Listar contactos.	1	2/6
	Modificar contacto.	1	3/6



	Eliminar contacto.	1	2/6
com_admin_sms	Crear cuenta de usuario.	1	4/6

4.2.2 Tareas de la ingeniería.

Se describirán las tareas de la ingeniería pertenecientes a las HU de prioridad alta, efectuadas para cada uno de los componentes implementados. Las tareas de prioridad media y baja podrán ser consultadas en el [Anexo 2](#).

Tabla 9: Tareas de la ingeniería de la 1ra iteración.

Tareas.	
No. Tarea: 1.	No. HU: 1.
Nombre de la tarea: Autenticar usuario.	
Tipo de tarea: Desarrollo.	Puntos de estimación: 3/6
Fecha inicio: 20/02/2012	Fecha fin: 22/02/2012
Programador responsable: Yanileidis Luna Solis.	
Descripción: se especificará el modo de autenticación del usuario, los datos a insertar son: Usuario y Contraseña, siendo campos obligatorios.	
No. Tarea: 2.	No. HU: 2.
Nombre de la tarea: Registrar usuario.	
Tipo de tarea: Desarrollo.	Puntos de estimación: 6/6
Fecha inicio: 23/02/2012	Fecha fin: 29/02/2012
Programador responsable: Yanileidis Luna Solis.	
Descripción: se especificarán los datos necesarios para que el usuario pueda registrarse en el sistema, los datos a insertar son: Nombre y/o Apellidos*, Usuario*, Número de beeper*, Palabra secreta*, Correo electrónico y Texto de la imagen generada con captcha*, donde los campos con * son obligatorios y el número de beeper es de tipo numérico de 5 dígitos.	
No. Tarea: 7.	No. HU: 7.
Nombre de la tarea: Enviar mensaje desde la web.	



Capítulo 4: Implementación y Prueba

Tipo de tarea: Desarrollo.	Puntos de estimación: 6/6
Fecha inicio: 01/03/2012	Fecha fin: 07/03/2012
Programador responsable: Otoniel Berroa Arias.	
<p>Descripción: se especificarán los datos necesarios para que el usuario pueda enviar mensajes a través de la web, los datos a insertar son: PagerID*, Mensaje*, y Texto de la imagen generada con captcha*, donde los campos con * son obligatorios y el número de beeper es de tipo numérico de 5 dígitos. El usuario introduce los datos del mensaje, posteriormente se verifica si los datos son correctos, cumpliendo con esta condición se procede a enviarlo mediante correo electrónico utilizando el servicio de Movitel, luego se encarga de enviarlo hacia el beeper y seguidamente se almacena el mensaje en la base de datos general del sistema.</p>	
No. Tarea: 8.	No. HU: 8.
Nombre de la tarea: Enviar mensaje a través de SMS.	
Tipo de tarea: Desarrollo.	Puntos de estimación: 12/12
Fecha inicio: 08/03/2012	Fecha fin: 21/03/2012
Programador responsable: Otoniel Berroa Arias.	
<p>Descripción: el proceso comienza cuando se capturan los datos de los mensajes enviados hacia el módem GSM utilizando la herramienta Gammu en modo demonio. Haciendo uso de un script que se ejecutará cada 1minuto mediante una tarea programada en Linux, se obtendrán los mensajes de la base de datos de gammu, luego verifica si el mensaje posee el formato correcto, cumpliendo con esta condición se procede a enviarlo mediante correo electrónico utilizando el servicio de Movitel, posteriormente se encarga de enviarlo hacia el beeper, y luego se almacena el mensaje en la base de datos general del sistema.</p>	
No. Tarea: 9.	No. HU: 9.
Nombre de la tarea: Enviar mensaje a través de correo electrónico.	
Tipo de tarea: Desarrollo.	Puntos de estimación: 12/12
Fecha inicio: 22/03/2012	Fecha fin: 04/04/2012
Programador responsable: Otoniel Berroa Arias.	
<p>Descripción: el proceso comienza cuando se capturan los datos de los mensajes enviados hacia el</p>	



<p>correo. Haciendo uso de un script que se ejecutará cada 1minuto mediante una tarea programada en Linux, se obtendrán los mensajes del directorio de almacenamiento del correo, luego se verifica si el mensaje posee el formato correcto, cumpliendo con esta condición se procede a enviarlo mediante correo electrónico utilizando el servicio de Movitel, posteriormente se encarga de enviarlo hacia el beeper, y luego se almacena el mensaje en la base de datos general del sistema.</p>	
No. Tarea: 10.	No. HU: 9.
Nombre de la tarea: Listar mensajes.	
Tipo de tarea: Desarrollo.	Puntos de estimación: 6/12
Fecha inicio: 05/04/2012	Fecha fin: 11/04/2012
Programador responsable: Otoniel Berroa Arias.	
Descripción: se especificarán los datos necesarios para que el usuario autenticado pueda visualizar los mensajes almacenados en su buzón.	
No. Tarea: 12.	No. HU: 11.
Nombre de la tarea: Programar mensaje para la fecha y hora deseada.	
Tipo de tarea: Desarrollo.	Puntos de estimación: 6/12
Fecha inicio: 12/04/2012	Fecha fin: 18/04/2012
Programador responsable: Otoniel Berroa Arias.	
Descripción: se especificarán los datos necesarios para que el usuario pueda programar un mensaje para la fecha y hora que desee, los datos a insertar son: PagerID*, Mensaje*, Fecha inicial*, Fecha final*, Hora* y Texto de la imagen generada con captcha*, donde los campos con * son obligatorios. En el proceso se hará uso de un script que se ejecutará cada 1minuto mediante una tarea programada en Linux, para que obtenga de la base de datos general, los mensajes que estén programados para la fecha y hora actual, cumpliendo con esta condición se procede a enviarlo mediante correo electrónico utilizando el servicio de Movitel, posteriormente se encarga de enviarlo hacia el beeper.	

4.3 Diagrama de despliegue.

El diagrama de despliegue muestra las relaciones físicas entre componentes de hardware y software en el sistema, es decir permite apreciar de forma visual como se encuentran relacionados físicamente los componentes del sistema. [52]

La metodología XP no plantea el modelado de este artefacto, pero para un mejor entendimiento de donde se desplegará el sistema se decide su realización.

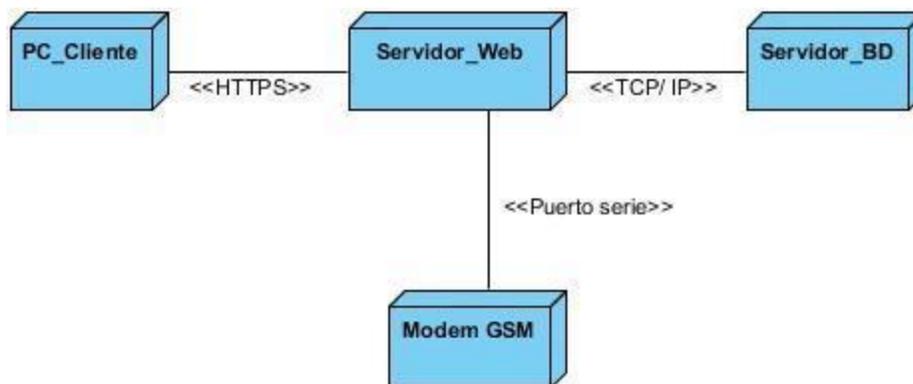


Figura 10: Diagrama de despliegue.

PC_Cliente: Es la pc que se conecta a la aplicación para visualizar los componentes web desarrollados.

Servidor_Web: El servidor tendrá instalado el sistema operativo Ubuntu, en su versión 11.10, contiene la aplicación web y software del sistema; también se tendrá montado el servidor de correo utilizado y la herramienta Gammu para el trabajo con SMS. El servidor se encarga de controlar, ejecutar y gestionar todos los pedidos de acciones que el usuario realiza a través de la PC_Cliente. La aplicación se comunica con un sistema de gestión de base de datos MySQL, empleándose como protocolo de comunicación entre el cliente y el servidor web el HTTPS por el puerto 443.

Servidor_BD: Es el servidor que almacena toda la información que maneja el sistema para cumplir sus funcionalidades, el cual se comunica con el servidor web mediante el protocolo TCP/ IP (del inglés: *Transmission Control Protocol/ Internet Protocol*) por el puerto 3306, ya que permite establecer una conexión e intercambiar datos entre los mismos, garantizando la entrega de los datos y que no se pierdan durante la transmisión.

Módem GSM: Es el dispositivo que se utilizará para recibir los mensajes SMS, el cual estará conectado al servidor web mediante el puerto serie.

4.4 Pruebas.

Uno de los principios fundamentales de XP es el proceso de prueba, donde los desarrolladores realizan pruebas constantemente tanto como sea posible; de esta manera se reduce el número de errores no



Capítulo 4: Implementación y Prueba

detectados, así como el tiempo entre la introducción de este en el sistema y su detección. Contribuyendo a elevar la calidad del producto desarrollado. [53]

XP divide las pruebas en dos grupos:

Pruebas unitarias, desarrolladas por los programadores, encargadas de verificar el código de forma automática. Cada uno de los desarrolladores tiene que ir probando constantemente lo que va obteniendo en el transcurso de la implementación de un sistema, para garantizar que las funcionalidades exigidas por el cliente estén siendo implementadas correctamente.

Las pruebas unitarias fueron desarrolladas constantemente cada vez que se terminaba de implementar alguna funcionalidad probándola directamente en el entorno real, introduciéndoles juegos de datos para asegurar que dicha funcionalidad realizaba lo que deseaba el cliente. A continuación se describe el juego de datos utilizado para uno de los procedimientos, donde para los restantes se llevó a cabo la misma estrategia.

Procedimiento insertar usuario en la base de datos:

Para insertar un usuario es necesario insertar un nombre y/o apellidos, un usuario, número de beeper, palabra secreta, el texto de una imagen, y si deseaba un correo electrónico, se entraron los siguientes datos (Otoniel Berroa, oto, 11111, gato, sd1254ss).

La salida será verdadera cuando el procedimiento sea capaz de validar los datos e insertarlos en la tabla encargada almacenar la información del usuario, en la base de datos.

La salida será falsa si se insertan los datos (111,_,ssks,_,_), y procedimiento no permita la entrada de los datos, mostrando un mensaje, especificando el tipo de error cometido.

Pruebas de aceptación, las cuales son destinadas a evaluar bloques más grandes de la funcionalidad del sistema, tales como las historias de usuarios, verificando que al final de una iteración se obtuvo la funcionalidad requerida, además de que dicha funcionalidad sea la esperada por el cliente. [53]

4.4.1 Pruebas de aceptación.

Las pruebas de aceptación también llamadas pruebas del cliente son creadas a partir de las Historias de Usuarios, donde durante las iteraciones las HU seleccionadas serán sometidas a dichas pruebas. En dichas pruebas se especifican, desde la perspectiva del cliente, los escenarios para probar que una HU ha sido implementada correctamente. Una HU puede tener todas las pruebas de aceptación que necesite para asegurar su correcto funcionamiento. Una HU no se considera completa hasta que no ha pasado por



Capítulo 4: Implementación y Prueba

las pruebas de aceptación. Siendo el objetivo final de las pruebas, garantizar que los requerimientos han sido cumplidos y que el sistema es aceptable. [53]

Las pruebas de aceptación correspondiente a cada una de las funcionalidades de la Plataforma de mensajería paging 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.

Los casos de pruebas restantes podrán ser consultados en el [Anexo 3](#).

Tabla 10: Casos de prueba de aceptación.

Caso de prueba de aceptación.	
Código: HU1-P1.	No.HU: 1.
Nombre: Autenticar usuario.	
Descripción: la aplicación debe brindar la posibilidad de autenticarse.	
Condiciones de ejecución: El usuario debe estar registrado en el sistema.	
Entrada/Paso de ejecución: el usuario accede a la interfaz inicial, escoge la opción de autenticarse e introduce los datos siguientes: Usuario y Contraseña, donde los campos son obligatorios.	
Resultado esperado: Correcto: Si los datos entrados son válidos, la aplicación verificará que exista como usuario dentro la tabla y el rol que cumple dentro del sistema, de ser así permitirá iniciar la sección, cargando solo las funcionalidades correspondientes a su rol. Incorrecto: Si el usuario insertado es incorrecto, se muestra el mensaje de error “El usuario no existe”, y le permite volver a intentarlo.	



<p>Si la contraseña insertada es incorrecta, se muestra el mensaje de error “Contraseña inválida”, y le permite volver a intentarlo.</p> <p>Si el campo de la contraseña está vacío, se muestra el mensaje de error “No se permite una contraseña vacía”, y le permite volver a intentarlo.</p> <p>De lo contrario lo invitará a registrarse.</p>	
<p>Evaluación de la prueba: Satisfactoria.</p>	
<p>Caso de prueba de aceptación.</p>	
<p>Código: HU2-P2.</p>	<p>No.HU: 2.</p>
<p>Nombre: Registrar usuario.</p>	
<p>Descripción: la aplicación debe brindar la posibilidad de registrarse en el sistema.</p>	
<p>Condiciones de ejecución:</p> <p>El usuario debe poseer un beeper, debido a que la contraseña para activar su cuenta le será enviada al mismo.</p>	
<p>Entrada/Paso de ejecución: el usuario accede a la interfaz inicial, escoge la opción de Crear una cuenta e introduce los datos siguientes: Nombre y/o Apellidos*, Usuario*, Número de beeper*, Correo electrónico, Palabra Clave* y Texto de la imagen generada con captcha*.</p>	
<p>Resultado esperado:</p> <p>Correcto: Si los datos entrados son válidos, la aplicación lo insertará dentro de la tabla usuario, se le enviará un mensaje al beeper con su contraseña, de manera que mediante la misma active su cuenta, se muestra el mensaje “Cuenta creada con éxito... un mensaje ha sido enviado a su beeper”.</p> <p>Incorrecto: Si existen campos obligatorios vacíos, se muestra el mensaje de error “Hay campos obligatorios”.</p> <p>Si el número de beeper insertado contiene letras, se muestra el mensaje de error “El beeper no puede contener letras”.</p> <p>Si el número de beeper insertado no es de 5 dígitos, se muestra el mensaje de error “El número de beeper es de 5 dígitos”.</p> <p>Si el campo del texto de la imagen generada con captcha está vacío o no se encuentra escrito de igual manera, se muestra el mensaje de error “Error: Texto de la imagen incorrecto, inténtelo nuevamente”.</p>	
<p>Evaluación de la prueba: Satisfactoria.</p>	



Capítulo 4: Implementación y Prueba

No conformidades.

Luego de realizarse 3 iteraciones para las pruebas de aceptación se obtiene un total de 13 no conformidades, donde al final de las iteraciones 13 fueron resueltas, 0 no procede y no quedó ninguna pendiente; y todas las no conformidades fueron no significativas.

Tabla 11: No conformidades.

Sistema.	Plataforma de mensajería paging.
No conformidades.	13
Resueltas.	13
No proceden.	0
Pendientes.	0

4.4.2 Pruebas de seguridad.

Las pruebas de seguridad que se llevaron a cabo fueron las pruebas de control de accesos, las mismas garantizan que los usuarios están restringidos a funciones específicas o su acceso está limitado únicamente a los datos que estén autorizados a acceder. También que solo aquellos usuarios autorizados a acceder al sistema son capaces de ejecutar las funciones del sistema. [54]

Las técnicas usadas para llevar a cabo dicha prueba fueron identificar cada tipo de usuario, las funciones y datos a los que se debe autorizar; y se crearon las pruebas para verificar los permisos específicos para cada tipo de usuario.

Resultados de las prueba de control de acceso.

Los resultados de las pruebas de control de acceso correspondientes a cada una de las funcionalidades de la Plataforma de mensajería paging serán representadas mediante tablas divididas por las secciones siguientes:

- ✓ Funcionalidades del sistema.
- ✓ Roles definidos.



Tabla 12: Resultados de pruebas de control de acceso.

Funcionalidades.	Roles.			
	Invitado.	Registrado.	Operadora.	Administrador.
Autenticar usuario.		X	X	X
Registrar usuario.	X			
Listar usuarios.			X	X
Modificar cuenta de usuario.		X	X	X
Asignar rol al usuario.				X
Eliminar cuenta de usuario.		X	X	X
Enviar mensaje desde la web.	X	X	X	X
Enviar mensaje a través de SMS.	X	X	X	X
Enviar mensaje a través de correo electrónico.	X	X	X	X
Listar mensajes.		X	X	X
Eliminar mensaje.		X	X	X
Programar mensaje para la fecha y hora deseada.	X	X	X	X
Insertar contacto.		X	X	X
Listar contactos.		X	X	X
Modificar contacto.		X	X	X
Eliminar contacto.		X	X	X
Crear cuenta de usuario				X

4.5 Conclusiones Parciales.

En este capítulo se abordaron los temas referentes a la implementación del sistema, quedaron descritas las tareas de la ingeniería correspondientes a cada HU listas para su implementación; y quedó diseñado el modelo de despliegue. Se plantearon además las pruebas a seguir durante la elaboración del mismo, donde se definieron las pruebas de aceptación que propician al cliente conformidad y seguridad ante el sistema. Con el fin del capítulo se da por terminada la propuesta que trae este trabajo.



CONCLUSIONES GENERALES

Con el desarrollo de este trabajo se puede concluir con la satisfacción de haber obtenido una aplicación donde se implementaron 6 extensiones entre componentes y el tema (Plantilla), permitiendo que la aplicación desarrollada brinde las funcionalidades definidas previamente, erradique las limitaciones que presentan los sistemas mencionados anteriormente y que apoye en gran medida a fortalecer las comunicaciones en Cuba.

Quedó demostrada la necesidad de desarrollar un sistema que fuese capaz de hacer converger las vías de comunicación SMS, correo electrónico e Internet para apoyar las comunicaciones entre las personas que lo necesitan.

Se realizaron las diferentes funcionalidades, añadidas como extensiones de Joomla, lo que permite que la aplicación sea configurable y adaptable, y pueda ser utilizada en otras instituciones si así se considera, sin efectuarse cambios significativos.

A su vez, el uso de la metodología de desarrollo XP ha permitido construir, en un plazo de corto tiempo, un producto bastante maduro. Producto que responde a los requisitos planteados, permitiendo:

- ✓ Gestionar el envío de mensajes paging por las vías SMS y correo electrónico.
- ✓ La persistencia de los mensajes, para que tengan constancia y puedan ser leídos posteriormente.
- ✓ La programación de mensajes paging para la fecha y hora deseada.
- ✓ Integrar en una plataforma las diferentes vías de envío de mensajes paging planteadas anteriormente.



RECOMENDACIONES

Como parte del proceso investigativo que se llevó a cabo surgió una idea que se puede tener en cuenta para un mejoramiento del trabajo realizado:

- ✓ Se recomienda trabajar en la integración con el sistema encargado de brindar servicios paging que ofrece la empresa cubana Movitel, debido a que esta puede proporcionar la base de datos que contiene toda la información relacionada con los beepers existentes en Cuba, para poseer así mayor seguridad de que los números de beeper sean reales.



Referencias Bibliográficas

REFERENCIAS BIBLIOGRÁFICAS

- [1] Ingeniatic. (28/ 11/ 2011). Available: <http://www.ingeniatic.net/index.php/tecnologias>
- [2] Movitel. (25/ 11/ 2011). Available: <http://www.movitel.co.cu/index.php?page=enaios&type=4>
- [3] H. T. Tapia, "Maestria en ciencias con especialidad en Sistemas digitales. Diseño de una red de radiocomunicaciones móvil NPCS de dos vías. ," Centro de Investigación y Desarrollo de Tecnología Digital., Tijuana, Baja california, México.
25 / 11/ 2011). Capitulo 3 Sistema Paging. Available:
http://catarina.udlap.mx/u_dl_a/tales/documentos/lep/martinez_m_c/capitulo_3.html#
- [5] J. Luna. 25/ 11/ 2011). Las telecomunicaciones en Cuba, presente y retos del futuro. Available:
<http://www.slideshare.net/javier.luna/fibra-optica>
- [6] C. G. G. VILLAGRÁN, "SISTEMA INTEGRADO DE MENSAJERÍA," FACULTAD DE INGENIERÍA, UNIVERSIDAD DE SAN CARLOS DE GUATEMALA, GUATEMALA.
(30/ 11/ 2011). *Sistemas Buscapersonas.* Available:
<http://www.salohogar.com/ciencias/tecnologia/beepers/pocsag.htm>
- [8] INDINCOM. (25/ 11/ 2011). Available: <http://www.indicom.com.ar/>
- [9] SkyTel. (25/ 11/ 2011). Available: <http://www.skytel.com.ar/>
- [10] B. Clark. (10/ 01/ 2012). *Electronic Engineering, Pager gateway.* Available:
<http://www.beeperpeople.com/paginggateway/>
- [11] Ecured. (10/ 01/ 2012). *Aplicaciones web.* Available:
http://www.ecured.cu/index.php/Aplicaci%C3%B3n_web
- [12] AMH2WEB. (10/ 01/ 2012). *Ventajas y beneficios de aplicaciones web.* Available:
<http://www.amh2web.com/ventajas-aplicaciones-web.html>
- [13] M. Á. Álvarez. 16/ 01/ 2012). Definición y ejemplos de sistemas CMS: Content Manager System (Sistema gestor de contenidos). Available: <http://www.desarrolloweb.com/articulos/que-es-un-cms.html>
- [14] Drupalcommunity. (16/ 01/ 2012). *Que-es-drupal* Available:
<http://drupalcommunity.org/?q=content/que-es-drupal>
- [15] D. H. Comunidad de usuarios de Drupal. (16/ 01/ 2012). *Características de Drupal.* Available:
<http://drupal.org.es/caracteristicas>
- [16] Joomlaos.net. (16/ 01/ 2012). *¿Qué es Joomla?* Available: <http://www.joomlaos.net/ique-es-joomla>
- [17] Joomlaos.net. (16/ 01/ 2012). *Características-de-joomla.* Available:
<http://www.joomlaos.net/caracteristicas-de-joomla>
- [18] A. Aulbach and Egon Schmid. (25/ 11/ 2011). *Manual de PHP* Available:
<http://www.php.net/manual/es/>
- [19] D. P. Valdés. 25/ 11/ 2011). *¿Qué es Javascript?* Available:
<http://www.maestrosdelweb.com/editorial/%C2%BFque-es-javascript/>
- [20] Ecured. (25/ 11/ 2011). *HTML.* Available: <http://www.ecured.cu/index.php/Html>
- [21] J. E. Pérez. (28/ 11/ 2011). *Introduccion a CSS.* Available: <http://www.librosweb.es/css>
- [22] K. B. Acuña. (2009, 29/ 11/ 2011). *SELECCIÓN DE METODOLOGÍAS DE DESARROLLO PARA APLICACIONES WEB EN LA FACULTAD DE INFORMÁTICA DE LA UNIVERSIDAD DE CIENFUEGOS.* Available:
<http://www.eumed.net/libros/2009c/584/Metodologias%20tradicionales%20y%20metodologias%20agiles.htm>
- [23] P. Kruchten. (29/ 11/ 2011). *A Software Development Process for a Team of One.*



Referencias Bibliográficas

- [24] P. L. y. M. C. P. José H. Canós. (29/ 11/ 2011). Metodologías Ágiles en el Desarrollo de Software. Available: www.willydev.net/descargas/prev/TodoAgil.Pdf
- [25] C. Larman. (25/ 11/ 2011). *UML y Patrones, Introducción al análisis y diseño orientado a objetos*.
- [26] J. Schmuller, *Aprediendo UML en 24 horas*.
- [27] V. Paradigm. (29/ 11/ 2011). *Un repositorio Central de los diseños y documentos*. Available: <http://www.visual-paradigm.com/product/vpuml/>.
- [28] (09/ 01/ 2012). *Internet Information Services*. Available: <http://www.microsoft.com/spain/windowsserver2003/technologies/webapp/iis.msp>
- [29] C. Linux. (09/ 01/ 2012). *Una Introducción a APACHE*. Available: http://linux.ciberaula.com/articulo/linux_apache_intro/
- [30] U. d. Oviedo. (09/ 01/ 2012). *Tema 4 Introducción a Apache 2.0.X*. Available: <http://ocw.uniovi.es/mod/resource/view.php?id=1242>
- [31] Postgresql-es. (09/ 01/ 2012). *Sobre PostgreSQL*. Available: http://www.postgresql.org.es/sobre_postgresql
- [32] I. Gilfillan, *La biblia de MySQL*.
- [33] Ecured. (25/ 11/ 2011). *ERStudio*. Available: <http://www.ecured.cu/index.php/ER/Studio>
- [34] Softonic. (25/ 11/ 2011). *DBDesigner*. Available: <http://dbdesigner.softonic.com/>
- [35] Ecured. (25/ 11/ 2011). *PhpDesigner* Available: <http://www.ecured.cu/index.php/PhpDesigner>
- [36] Ecured. (30/ 11/ 2011). *Zend Studio*. Available: http://www.ecured.cu/index.php/Zend_Studio
- [37] Oracle. (16/ 01/ 2012). *System Administration Guide: Network Services* Available: <http://docs.oracle.com/cd/E19253-01/816-4555/mailrefer-79/index.html>
- [38] Postfix. (16/ 01/ 2012). Available: <http://doc.ubuntu-es.org/Postfix>
- [39] Ecured. (30/ 11/ 2011). *Gammu*. Available: <http://www.ecured.cu/index.php/Gammu>
- [40] D. Paquetes. (30/ 11/ 2011). *Debian GNU/Linux - Paquetes*. Available: <http://packages.debian.org/es/sid/gammu>
- [41] [GW]ammu. (14/ 03/ 2012). *Gammu SMSD*. Available: <http://es.wammu.eu/smsd/>
- [42] B. W. F. Ben Collins-Sussman , C. Michael Pilato. (09/ 01/ 2012). *Version Control with Subversion*. Available: <http://svnbook.red-bean.com/index.es.html>
- [43] L. M. Alejo. 03/05/2012). *Seguridad en Joomla! - Listando extensiones vulnerables en Joomla! 2.5 con Securitycheck*. Available: <http://www.webempresa.com/blog/item/872-seguridad-en-joomla-listando-extensiones-vulnerables-en-joomla-25-con-securitycheck.html>
- [44] M. A. Alvarez. 03/05/2012). *Una de las captchas de PHP más completas, gratuita y fácil de integrar en las aplicaciones PHP*. Available: <http://www.desarrolloweb.com/articulos/captcha-php-securimage.html>
- [45] R. S. Pressman. (12/ 01/ 2012). *Ingeniería del Software: Un Enfoque Práctico (sexta ed.)*.
- [46] Ecured. (16/ 01/ 2012). *Arquitectura Cliente / Servidor*. Available: <http://www.ecured.cu/index.php/Cliente-Servidor>
- [47] 16/ 01/ 2012). *PHP: Comunicación entre servidor y cliente*. Available: <http://www.wp-monkeys.com/php-comunicacion-entre-servidor-y-cliente.html>
- [48] J. documentation. (25/ 01/ 2012). *Developing a Model-View-Controller Component - Part 1*. Available: http://docs.joomla.org/Developing_a_Model-View-Controller_Component_-_Part_1
- [49] Netbeans. (5/ 01/ 2012). Available: http://netbeans.org/kb/docs/javaee/ecommerce/design_ja.html
- [50] W. U. Ayllapan. 25/ 01/ 2012). *El patrón de la fábrica (Factory)*. Available: <http://www.joomlaos.net/framework-joomla-15x/106-el-patron-de-la-fabrica-factory>



Referencias Bibliográficas

- [51] L. Barriocanal. (16/ 01/ 2012). ¿Qué son los componentes de Joomla? Available: http://www.edujoomla.es/index.php?option=com_content&view=article&id=53:los-componentes-en-joomla&catid=28:componentes-de-joomla&Itemid=96
- [52] Ecured. (14/ 03/ 2012). *Diagrama de despliegue*. Available: http://www.ecured.cu/index.php/Diagrama_de_despliegue
- [53] T. H. Lisa Crispin. (05/ 04/ 2012). *Testing Extreme Programming*. Available: http://www.ebooksbay.org/The_Digital_Library/2008/01/23/Testing_Extreme_Programming_The_XP_Series
- [54] I. Blank., *et al.* (05/ 04/ 2012). *Pruebas Funcionales*. Available: http://carolina.terna.net/ingsw3/datos/Pruebas_Funcionales.pdf



BIBLIOGRAFÍA

- [1] R. S. Pressman. (12/ 01/ 2012). *Ingeniería del Software: Un Enfoque Práctico (sexta ed.)*. Capítulos 04_Desarrollo_ágil_Parte_1_XP y 08_Modelado_del_Análisis_Parte_3.
- [2] Joskowicz. Reglas y Prácticas en eXtreme Programming. 2008.
- [3] C. Larman. (25/ 11/ 2011). *UML y Patrones, Introducción al análisis y diseño orientado a objetos*. PARTE IV FASE DE DISEÑO. Capítulo 18 GRASP: PATRONES PARA ASIGNAR RESPONSABILIDADES.
- [4] A. Aulbach and Egon Schmid. (25/ 11/ 2011). *Manual de PHP* Available: <http://www.php.net/manual/es/>
- [5] M. Hernán Ruiz. Programación web avanzada. Capítulo 10 PHP.
- [6] T. H. Lisa Crispin. (05/ 04/ 2012). *Testing Extreme Programming*. Available: http://www.ebooksbay.org/The_Digital_Library/2008/01/23/Testing_Extreme_Programming_The_XP_Series_.
- [7] M. A. Alvarez. (03/05/2012). Una de las captchas de PHP más completas, gratuita y fácil de integrar en las aplicaciones PHP. Available: <http://www.desarrolloweb.com/articulos/captcha-php-securimage.html>.